Contents

Abstract. Enforcing privacy in modern day database has become more and more one of the main aspect of study in the database community. Storing and processing information had led to an explosion of data collection, many organizations such as hospitals,companies and even the State itself with the census, collect analyse and distribute personal information in order to return services that are more useful,appealing or more efficient. However the collected data are the result of tracking public or private lives of individuals this therefore lead to a great privacy risk of unauthorized or malevolent disclosure.

During the process of analyzing integrated databases we suddenly understand that the risks are even more concrete than a normal database because of the structure and composition itself of an integrated database, while normally in a database there are information coming from just one source and referring to just one aspect of the life of the indvidual in a integrated database the sources for information are mutiple and usually lead to a greater risk in privacy disclosure.

In a clinical database are not only stored all the information of the patient but also the clinical history which can lead, if disclosed, not only to a loss of privacy but also to all sorts of harassment and discrimination towards the patient therefore a more strong concept of anonimicity is required.

In the first part of this thesis we give some basic notions of privacy such as k-anonimity and l-diversity we then move forward to a more "state of the art" approach to the problem, since the case we are considering requires so much privacy, and therefore we present the differential-privacy as well as some mechanism that preserve it; we also give some legal notion on the argument to frame the problem to create better solutions

In the second part of the thesis we present the case of study of a integrated clinical database of Italian Lymphoma Foundation "FIL"

Italian Summary of the Thesis

(Sintesi della tesi in italiano)

Prefazione

Questa sintesi della tesi in italiano ha un fine esclusivamente illustrativo, in quanto la complessità e la vastità dell' argomento hanno reso impossibile descrivere correttamente, in così poche pagine, le varie fasi di ricerca e lo sviluppo del software *Anon-Alyzer*.
In particolar modo è caldamente consigliata la lettura della tesi in inglese, in quanto più accurata, dettagliata e corretta. Per tale motivo è stato più opportuno procedere da un sunto della parte teorica riguardante le tematiche basilari del mondo dell'anonimizzazione.

Introduzione

Negli ultimi anni si è assistito ad un aumento esponenziale dei dati personali prodotti e di conseguenza immagazzinati, si pensi che nel solo 2007 tali dati ammontavano a 8-10gb di dati giornalieri pubblici e circa 4tb di dati "privati" [1]
Ogni giorno i dati di milioni di invidivui, prodotti ad ogni loro azione,vengono immagazzinati, ad esempio quando si usa moneta elettronica per pagamenti, ma anche quando si utilizza moneta cartacea magari abbinata a tessere per sconti personalizzate, quando ci si reca in strutture sanitarie pubbliche o private ed automaticamente vengono registrati tutti i nostri dati, quando si comunica con altre persone attraverso social network. Si può dire che in un certo senso ogni aspetto della nostra vita, anche privata, viene immagazzinato e catalogato all'interno di un database.

Per privacy si intende, comunemente, il diritto della persona di controllare che le informazioni che la riguardano vengano trattate o usate da altri solo in caso di necessità. La privacy è un concetto che inizialmente veniva riferito solo alla sfera della vita privata, mentre negli ultimi decenni ha subito un'evoluzione e si è esteso fino ad arrivare ad indicare il diritto al controllo sui propri dati personali [5]

Lo scopo di questa tesi è quello studiare lo *state of the art* sull'anonimizzazione in maniera critica e di produrre una *web application* che si interfacci ad un database clinico integrato e fornisca informazioni anonimizzate che rispettino la *privacy* degli individui all'interno del database.

Il contributo originale sta proprio nell'affrontare tematiche così attuali, ancora non trattate in nessun corso di studio, come la difesa della *privacy* degli utenti presenti nei dataset, e nel fornirne esempi

concreti di implementazioni possibili con un occhio critico in grado di discernere fra le varie metodologie quali, nel nostro caso specifico, siano le più efficaci.

Legislazione sulla Privacy

La recente diffusione delle nuove tecnologie ha contribuito ad un assottigliamento della barriera della privacy, ad esempio la tracciabilità dei cellulari o la relativa facilità a reperire gli indirizzi di posta elettronica delle persone. Oggi, con la nascita del Laboratorio Privacy Sviluppo presso il Garante per la protezione dei dati personali, la privacy viene anche intesa come "*sovranità su di sé*", in un'accezione del tutto nuova, non più limitata, come in passato, ad un diritto alla "non intromissione nella propria sfera privata", ma ponendosi come indiscutibile strumento per la salvaguardia della libera e piena autodeterminazione della persona.

Già la Convenzione europea dei diritti dell'uomo, all'art. 8, stabiliva che non può esservi ingerenza di una autorità pubblica nell'esercizio di tale diritto a meno che tale ingerenza sia prevista dalla legge e costituisca una misura che, in una società democratica, è necessaria per la sicurezza nazionale, per la pubblica sicurezza, per il benessere economico del paese, per la difesa dell'ordine e per la prevenzione dei reati, per la protezione della salute o della morale, o per la protezione dei diritti e delle libertà altrui. Oltre che negli Accordi di Schengen, il concetto è stato riportato nella Carta dei diritti fondamentali dell'Unione europea all'art. 8, che recita:

> *Ogni individuo ha diritto alla protezione dei dati di carattere personale che lo riguardano. Tali dati devono essere trattati secondo il principio di lealtà, per finalità determinate e in base al consenso della persona interessata o a un altro fondamento legittimo previsto dalla legge. Ogni individuo ha il diritto di accedere ai dati raccolti che lo riguardano e di ottenerne la rettifica.*
> *Il rispetto di tali regole è soggetto al controllo di un'autorità indipendente.*

Le fonti comunitarie rilevanti sono contenute nella Direttiva del Parlamento europeo e del Consiglio del 24 ottobre 1995, contrassegnata dalla sigla 95/46/CE, pubblicata nella GUCEL 281 del 23.11.1995 (p. 31).

Per quanto attiene alla legislazione italiana, i fondamenti costituzionali sono ravvisabili negli art. 14, 15 e 21 Cost., rispettivamente riguardanti il domicilio, la libertà e segretezza della corrispondenza, e la libertà di manifestazione del pensiero; ma si può fare anche riferimento all'art.

2 Cost., incorporando la riservatezza nei diritti inviolabili dell'uomo. I diritti della persona vengono riconosciuti nella Dichiarazione Universale dei Diritti Umani e nella Convenzione Europea dei Diritti dell'Uomo in maniera internazionale.Prima della Legge sulla privacy, la fonte di diritto principale in materia era costituita dalla Corte di Cassazione. Questa, con la sent. n. 4487 del 1956, nega inizialmente la presenza di un diritto alla riservatezza. Il riferimento all'art. 2 Cost. di cui sopra arriva invece solo nel 1975, con la sent. Cass. 27 maggio 1975 n. 2129, con cui la stessa Corte identifica tale diritto nella tutela di quelle situazioni e vicende strettamente personali e familiari, le quali, anche se verificatesi fuori dal domicilio domestico, non hanno per i terzi un interesse socialmente apprezzabile contro le ingerenze che, sia pure compiute con mezzi leciti, per scopi non esclusivamente speculativi e senza offesa per l'onore, la reputazione o il decoro, non sono giustificati da interessi pubblici preminenti.[6] Questa affermazione è fondamentale per il bilanciamento col diritto di cronaca.
La casistica in materia è ampia; in particolare, il Tribunale di Roma, nella sent. Del 13 febbraio 1992, aveva notato che chi ha scelto la notorietà come dimensione esistenziale del proprio agire, si presume abbia rinunciato a quella parte del proprio diritto alla riservatezza direttamente correlato alla sua dimensione pubblica.

La linea di demarcazione tra il diritto alla riservatezza e il diritto all'informazione di terzi sembra quindi essere la popolarità del soggetto. Tuttavia, anche soggetti molto popolari conservano tale diritto, limitatamente a fatti che non hanno niente a che vedere con i motivi della propria popolarità.
Un ulteriore passo avanti nella formazione di una normativa adeguata, anche se notevolmente in ritardo, viene fatto per rispetto di obblighi internazionali: con la legge n. 98 del 21 febbraio 1989[5], è infatti ratificata la Convenzione di Strasburgo (adottata nel 1981), sulla protezione delle persone rispetto al trattamento automatizzato di dati di carattere personale.In Italia è attualmente in vigore il Decreto legislativo 30 giugno 2003, n. 196, Codice in materia di protezione dei dati personali, che ha abrogato la Legge sulla privacy del 1996.

Privacy non è infatti più considerata quale diritto a che nessuno invada il "nostro mondo" precostituito bensì è anche intesa quale diritto a che ciascuno possa liberamente esprimere le proprie aspirazioni più profonde e realizzarle, attingendo liberamente e pienamente ad ogni propria potenzialità.In questo senso si parla di privacy come "autodeterminazione e sovranità su di sé" (Stefano Rodotà) e "diritto a essere io" (Giuseppe Fortunato), riconoscersi parte attiva e non passiva di un sistema in evoluzione, che deve portare necessariamente ad un diverso rapporto con le istituzioni, declinato attraverso una presenza reale, un bisogno dell'esserci, l'imperativo del dover contare, nel rispetto reciproco delle proprie libertà.

Ed è proprio seguendo questo principio fondamentale di autodeterminazione che molti dei concetti più moderni di privacy nell'ambito informatico ci portano ad interrogarci non solo esclusivamente sulla diffusione dei dati privati dell'individuo ma anche sulla sua presenza o meno all'interno dello stesso database insomma qualsiasi informazione anche quella che non è contenuta nel database è da considerarsi privata.

Caso in esame

Il nostro caso riguarda nello specifico un database clinico integrato, frutto cioè della convoluzione di informazioni provenienti da più fonti eterogenee, precisamente di un database integrato attraverso MOMIS.

MOMIS[18] (Mediator envirOnment for Multiple Information Sources) e' un framework per l'estrazione e l'integrazione di informazioni per sorgenti dati strutturate e semistrutturate. Per compiere l'estrazione viene introdotto un linguaggio object- oriented con una semantica basata su di una Description Logics chiamato ODL-I3 derivato dallo standard ODMG. L'integrazione di informazioni viene compiuta in modo semi-automatico, utilizzando la conoscenza presente in un Common Thesaurus (definito utilizzando il framework) e le descrizioni ODL-I3 degli schemi sorgenti e con tecniche di clustering e di Description Logics. Il processo di integrazione definisce un vista virtuale integrata degli schemi sottostanti (il Global Schema) nella quale sono specificate regole di mapping e vincoli di integrita' per la gestione delle eterogeneita'. Il sistema MOMIS, basato sull'architettura wrapper/mediator, fornisce le modalita' e tool aperti per il data management in Internet-based information systems utilizzando un'interfaccia compatibile CORBA-2. Lo sviluppo di MOMIS e' iniziato nell'ambito del progetto nazionale INTERDATA, e nell'ambito del progetto D2I, sotto la direzione della Professoressa S. Bergamaschi. L'attività di ricerca è continuata all'interno del progetto di ricerca europeo IST "SEWASIE: Semantic Webs and Agents in Integrated Economies" (2002/2005). E' stato inoltre esteso nell'ambito del progetto MUR "NeP4B: Networked Peers for Business" (2006/2009) e del progetto IST-EU RDT "STASIS (SofTware for Ambient Semantic Interoperable Services)" (2006-2009)

Il database in questione appartiene alla Fondazione Fil[19] . La Fondazione Italiana Linfomi ONLUS è un organo di coordinamento delle attività svolte in Italia nel campo dei linfomi da oltre 120 Centri distribuiti su tutto il territorio nazionale, con lo scopo di migliorare la loro capacità di ricerca e di assistenza. Non ha scopo di lucro in quanto persegue esclusivamente finalità di solidariertà sociale, svolgendo infatti attività di ricerca scientifica nel campo dei linfomi. La Fondazione Italiana Linfomi è stata la naturale evoluzione dell'Intergruppo Italiano Linfomi, che è sorto nel 1993 con la prima riunione svoltasi a Firenze come gruppo di cooperazione spontanea tra clinici e ricercatori italiani impegnati nello studio e nella terapia dei linfomi. La sua esistenza e la sua attività sono state successivamente ufficializzate nel luglio 2004 con atto notarile che ha determinato la nascita di una Fondazione dotata di personalità giuridica e iscritta al registro delle ONLUS. La Fondazione Italiana Linfomi è nata con lo scopo di far collaborare gruppi attivi nello studio dei linfomi; succesivamente alcuni gruppi si sono fusi tra loro e l'IIL si è proposto come punto di riferimento per la loro collaborazione. La FIL ONLUS è nata ad Alessandria il 30 settembre 2010 con un atto notarile che ha sancito la trasformazione dello Statuto con la fusione di tutti i Gruppi all'interno di un'unica grande organizzazione. La FIL promuove studi prospettici e retrospettivi per rispondere a quesiti che richiedano casistiche molto numerose. Intende favorirequindi forme di collaborazione con organismi internazionali, verso i quali si configura come interlocutore d'elezione. Si prefigge inoltre di organizzare e migliorare i servizi per la diagnosi e la terapia di altre malattie linfoproliferative e promuove la formazione del Registro Italiano dei Linfomi (RIL).

La FIL ONLUSsviluppa iniziative di tipo informativo sui linfomi, con l'intento di far conoscere il problema e aiutare pazienti e parenti, coordinare gruppi di ricerca nella lotta contro i linfomi, costituire la base scientifica, organizzativa e legale per la conduzione di studi clinici sui linfomi, coordinare gli sforzi dei ricercatori per creare un unico grande gruppo cooperativo italiano per la lotta contro i linfomi e collaborare con gruppi europei negli studi internazionali sui linfomi. La FIL ONLUS coopera con l'International Extranodal Lymphoma Study Group (IELSG). Sono in fase di sviluppo ulterioriore studi su fattori prognostici internazionali di cui la FIL è il promotore (F2, T-cell project, validazione early-PET) e sono inziate nuove collaborazioni con l'European Organisation for Research and Treatment of Cancer (EORTC) per gli Hodgkin localizzati e con il Mantle Cell Network per i linfomi Mantellari.

Il database conterrà perciò, oltre ai dati privati del paziente anche le cartelle cliniche e la storia clinica del paziente, per tale motivo è di fondamentale importanza l'assoluta certezza dell'anonimato dell'individuo qualora i dati venissero rilasciati per scopi statistici.

Anonimizzazione vs Incertezza

Nel 2002 il governo del Massachussetts ha rilasciato dati clinici di pazienti del loro sistema sanitario, presumendo erroneamente che fossero anonimizzati correttamente, tuttavia fu provato che quasi l'87% degli individui all'interno dei dati erano riconoscibili e ri-identificabili univocamente. [3] Nel 2004 la Choicepoint ha rilasciato informazioni private finanziarie su quasi 145.000 persone a criminali che li stavano truffando. [2] Infine nel 2006 America OnLine "AOL" ha rilasciato log, credendo fossero anonimi, di ricerche effettuate tramite il motore di ricerca. collezionate da utenti per aiutare l'information retrieval in ambito accademico, ciò nonostante l'identità degli individui fu facilmente scoperta e diversi aspetti della vita privata degli stessi rivelata al mondo intero [4]

Questa è solo una breve lista degli episodi degli ultimi anni, probabilmente i più famosi, che non hanno visto l'utilizzo di mezzi esterni per appropriarsi in maniera indebita di dati di terzi ma bensì sono stati gli stessi proprietari dei dati a rilasciarli convinti fossero stati anonimizzati correttamente. Per tale motivo è stata d'uopo la definizione più teorica di ciò che è o non è correttamente anonimizzato.

Uno dei modi più comuni di concepire l'anonimizzazione è quello di vedere il tale processo come l'aggiunta di incertezza ai dati veri per essere sicuri che un malintenzionato non possa discernere correttamente quale sia l'associazione che porti a dati privati. E' perciò importante utilizzare strumenti e modelli di incertezza:

- per quantificare l'incertezza di un aggressore
- per capire l'impatto che può avere una conoscenza di background
- per permettere query efficienti e accurate sui dati anonimizzati

Dati incerti rappresentano molteplicità di mondi o scenari possibili, ogni mondo corrisponde ad un database, o ad un grafo ed un modello di incertezza può collegare ad ogni mondo una certa probabilità; le query concettualmente dovranno poter spaziare su tutti i mondi possibili in modo da fornire all'utente che faccia richiesta dati anonimizzati.

In particolar modo, con riferimento ai dataset, è necessario definire alcune terminologie per permettere una maggiore capacità di sintesi e al contempo una maggiore accuratezza nell'affrontare questa specifica tematica:

Identificatore: un identificatore univoco , ad esempio il SSN (social secury number, l'equivalente del nostro codice fiscale)

Quasi-Identificatore (QI) : Un dato che può identificare parzialmente un individuo in un dataset ad esempio la data di nascita (DOB), il sesso, il codice postale o ZIP code

Attributo Sensibile (SA) : l'associazione che vogliamo nascondere, ad esempio il salario è da considerarsi un dato sensibile, non sempre un SA è ben definibile.

Nello specifico se si considera un dataset come il seguente [10]

SSN	*DOB*	*Sex*	*ZIP*	*Salary*
11-1-111	0	0	53715	50
22-2-222	0	0	53715	55
33-3-333	0	0	53703	60
44-4-444	0	0	53703	65
55-5-555	0	0	53706	70
66-6-666	0	0	53706	75

Rilasciare l'informazione SSN in associazione con l'informazione del salario è una violazione di un SA. Ciò che molti gestori di dati fanno in questo caso è sopprimere l'identificatore univoco ma ciò non è bastato nel caso di AOL. La prima tipologia di "attacco" alla privacy sui nostri database infatti viene denominata "linking-attack" o attacco al collegamento e si riferisce per l'appunto al collegamento fra due dati quali, nell'esempio precedente, sono DOB, Sex e ZIP, è stato infatti dimostrato che basta conoscere questi tre dati di una persona per identificarla univocamente nell'87% dei casi [7].

E' perciò obbligatorio definire la k-anonimicità come metodo per difenderci dagli attacchi ai collegamenti di quasi identificatori

> *"La tabella T soddisfa la k-anonimicità riguardo i QI se e solo se ogni tupla nel multiset T[QI] compare almeno k volte"* [8]

Una tabella T' k-anonima perciò rappresenta l'insieme di tutti i casi possibili di tabelle T e in tal modo si definisce T' k-anonimizzazione di T

Uno degli algoritmi che implementa la k-anomicità è quello denominato Incognito, [9] ideato da Kristen LeFevre, David J. DeWitt e Raghu Ramakrishnan nel 2005.

Tale algoritmo computa tutte le "minime" generalizzazioni del nostro dominio prendendo idee dalla data cube computation e dalle regole di associazione nel datamining.

Ogni generalizzaaione del nostro intero dominio viene descritta da un "vettore del dominio"

B0={1/21/76, 2/28/76, 4/13/86} → B1={76-86}

1S0={M, F} → S1={*}

Z0={53715,53710,53706,53703} → Z1={5371*,5370*} → Z2={537**}

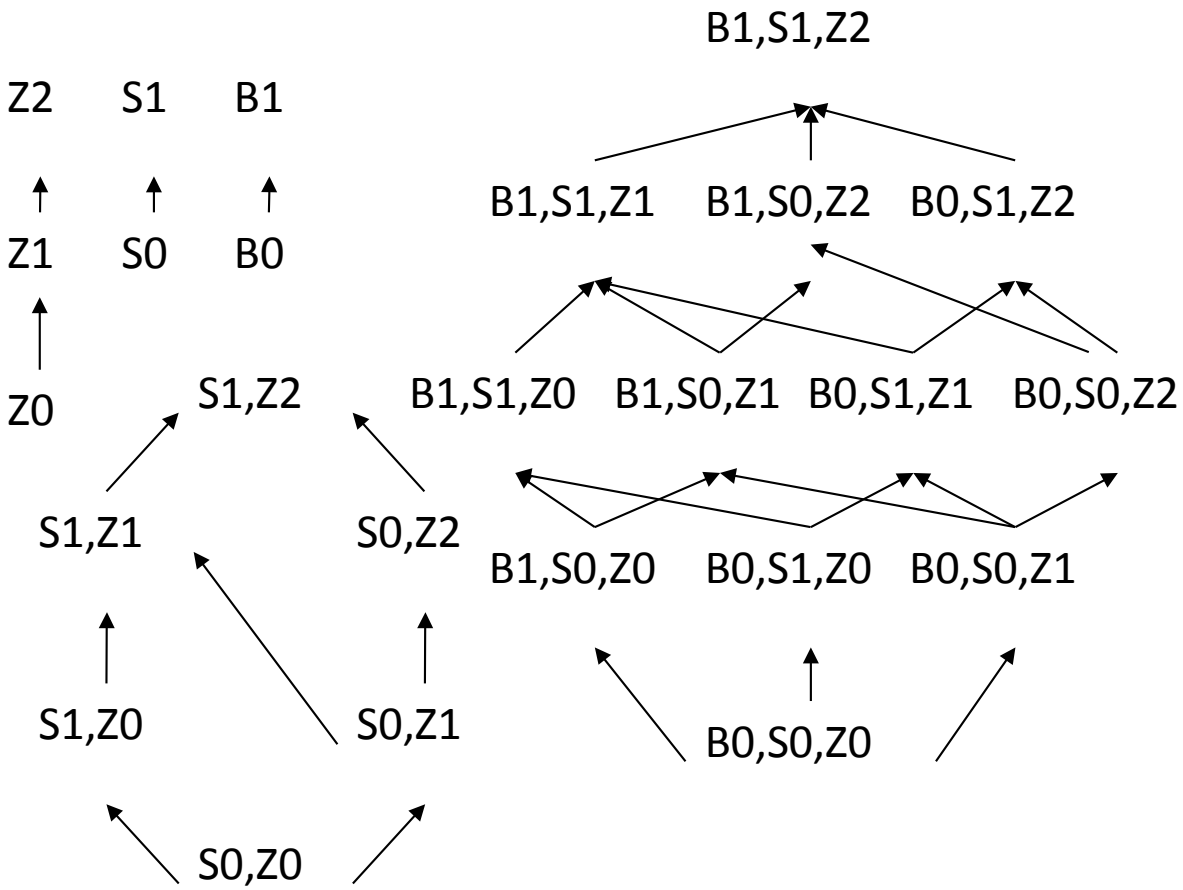
In tale modo la nostra tabella iniziale diventerà applicando il vettore B0,S1,Z2[13]

DOB	Sex	ZIP	Salary
1/21/76	M	53715	50,000
4/13/86	F	53715	55,000
2/28/76	M	53703	60,000
1/21/76	M	53703	65,000
4/13/86	F	53706	70,000
2/28/76	F	53706	75,000

B0, S1, Z2 →

DOB	Sex	ZIP	Salary
1/21/76	*	537**	50,000
4/13/86	*	537**	55,000
2/28/76	*	537**	60,000
1/21/76	*	537**	65,000
4/13/86	*	537**	70,000
2/28/76	*	537**	75,000

La griglia di vettori per k-anonimizzare il nostro dataset avrà, in riferimento a questo esempio specifico, il seguente aspetto:



Si può dedurre già da questa semplice immagine che la complessità dell'algoritmo renderà piuttosto oneroso il lavoro di conversione del nostro database in uno anonimizzato. Attraverso questo metodo si stima, infatti, che il sopracitato algoritmo abbia una complessità totale proporzionale a $\theta(2^{|QI|})$ [10]

Anche se le recenti implementazioni dell'algoritmo Incognito sono state adattate al concetto di l-diversità, un algoritmo puramente k-anonimizzante non garantisce però una privacy certa, giacché esiste una classe di attacchi alla privacy denominata homogeneity-attack, attacco alla omogeneità, che mina la sicurezza della privacy degli utenti.

Infatti se tutti, o quasi tutti, i valori dei SA in un gruppo di QI sono uguali c'è perdita di privacy.
Il problema risiede nella scelta dei raggruppamenti, non nei dati infatti per certi raggruppamenti non c'è perdita di privacy.

> *L-diversità: una tabella è l-diversa se ognuno dei suoi gruppi di quasi identificatori contiene almeno l valori ben rappresentati per i SA*[11]

Intuitivamente quello che viene enunciato è che se in una grossa frazione degli scenari possibili alcuni fatti sono veri allora la privacy può essere violata.
Sostanzialmente un algoritmo che può implementare la l-diversità è un algoritmo che implementa la k-diversità ma che sostituisce al test della k-anonimità il test per la l-diversità, ed è abbastanza facile da realizzare poichè basta controllare i contatori dei SA all'interno dei gruppi di QI.
Tuttavia ci sono limiti all'l-diversità, il limite più evidente è che nonostante due valori SA siano distinti essi possono essere semanticamente simili, si prenda ad esempio questo caso[10]:

DOB	Sex	ZIP	Salary
1/21/76	*	537**	50,000
4/13/86	*	537**	55,000
2/28/76	*	537**	60,000
1/21/76	*	537**	50,001
4/13/86	*	537**	55,001
2/28/76	*	537**	60,001

SSN	DOB	Sex	ZIP
11-1-111	1/21/76	M	53715

Diviene pertanto necessario definire la t-vicinanza[12] (t-Closeness) come:

> *Una tabella soddisfa la t-vicinanza se in ognuno dei gruppi QI la distanza fra la distribuzione dei valori dei SA nei gruppi e nell'intera tabella non è più grande di un valore di soglia t*

Differential privacy e meccanismo di Laplace

Ci si è accorti nello sviluppo di nuove tecniche di anonimizzazione che alla base delle nuove forme di attacco alle stessa era necessario ri-definire l'obiettivo.
Il motivo è da ricercarsi nella troppa generalità del termine "anonimizzato", la comunità scientifica ha quindi elaborato innanzitutto una nuova formalità per l'obiettivo da ricercare definendola "privacy" o più correttamente privacy differenziale.
La Privacy Differenziale è divenuta di recente uno standard di fatto per il rilascio di dati privati, ciò permette di garantire forti garanzie teoriche sulla privacy e sull'utilità dei dati rilasciati.
Il concetto alla base della privacy differenziale è che ciò che possiamo apprendere dai dati rilasciati non differisce sostanzialmente da ciò che avremmo appreso con o senza i dati di un particolare individuo al suo interno, in pratica se suddetto individuo avesse o non avesse i propri dati all'interno del dataset. Questo ovviamente ha lo scopo di rassicurare le persone garantendo che l'utilizzo dei loro dati privati non porti a rivelare informazioni su di loro.
Il problema perciò non sarà più di garantire l'anonimato ma bensì di rispettare la privacy differenziale, garantendo perciò che i risultati siano privati, accertandosi che i risultati siano "utili"

Siano D1 e D2 due dataset vicini, tali che D1 e D2 differiscano solo in una tupla t, scritto $||D1-D2||= 1$ (in alcuni casi questo significa che la tupla t è presente solo in uno dei due dataset.. in altri che le due tuple differiscano in alcuni valori, ed entrambe le definizioni garantiscono lo stesso livello di privacy)

Definizione: Siano D1,D2 due dataset vicini, sia A un algoritmo stocastico sul dataset e sia S un set arbitrario di possibili output di A. L'algoritmo A sarà **ε -differentially** private se

$$Pr[A(D1)\in S] \leq e^{\varepsilon}Pr[A(D2)] \in S]$$

Intuitivamente la privacy differenziale garantisce che nessuna tupla individuale possa influire sulle informazioni divulgate, la tecnica più comune per la realizzazione di un algoritmo differentially-private fu proposta in [13] ed è un meccanismo per la somma di rumore così definito:

Definizione: (Laplace Mechanism) Sia F(D) una funzione numerica su di un dataset D.
Un meccanismo ε -differentially private che permetta alla funzione di rilasciare informazioni dovrà essere così costruito L (D) = F(D) + X, dove X è una variabile stocastica campionata dalla

distribuzione Laplaciana Lap (s(f)/ ε)
Il valore di s(f), chiamato sensibilità di f, è la massima variazione di f quando una singola tupla di D cambia.
Formalmente:

$$s(f) = \max_{D1;D2:||D1-D2||=1} ||f(D1)- f(D2)||$$

Se per esempio f è un contatore allora la sensibilità s(f) sarà 1 per ogni due dataset vicini D1 e D2 la cui cardinalità differisca solo di 1.
Inoltre, come dimostrato nella sezione delle dimostrazioni di questa tesi, un rumore di grandezza θ (Δ/ε) campionato dalla distribuzione di Laplace soddisfa la ε-differential privacy
Ovviamente tale metodologia si applica bene su dati numerici mentre applicarla ad una query del tipo "Qual'è la nazionalità più diffusa nel database" produrrebbe risultati privi di senso, in aiuto ci viene, come vedremo, il meccanismo esponenziale.

Meccanismo Esponenziale

Il meccanismo esponenziale è una tecnica sviluppata da Frank McSherry e Kunal Talwar per lo sviluppo di algoritmi differentially-private in particolar luogo per estendere il concetto di privatezza differenziale anche a quei casi in cui il meccanismo di Laplace produce risultati privi di senso.
L'idea alla base è quella di perturbare la risposta ad una query di un ammontare proporzionale alla sensibilità della query stessa,

Un Meccanismo M: $\mathbb{N}^{|X|} \times R \rightarrow \mathbb{R}$ per qualche intorno astratto R.
Sia D il dominio degli input del dataset.
Sia R il range di valori con rumore.
Sia $\mathbb{R}$ l'insieme dei numeri reali.
Iniziamo definendo il voto di una funzione che restituisca voti: $D \times R \rightarrow \mathbb{R}$ che prenda come input un dataset $A \in D$ e restituisca un output $r \in \mathbb{R}$, in particolare ci restituirà un punteggio pesato, questo punteggio ci dice quando sia "buono" l'output per il dataset A, ovviamente più alto sarà il punteggio meglio sarà.

Il meccanismo esponenziale ε prenderà in input la funzione di punteggio e un parametro del dataset A e farà :

$$\varepsilon(A,\text{punteggio}, \varepsilon) = \text{output r con probabilità proporzionale a } \exp((\varepsilon/2)\text{punteggio}(A, r))$$

La sensibilità di una funzione che darà voti ci dice il massimo cambiamento nella funzione di voti per ogni coppia di dataset A e B tali che $|A \circ B|=1$. Più formalmente:

$$\Delta = \max_{r,A,B \text{ dove } |A \circ B|=1.} || \text{punteggio}(A,r) - \text{punteggio}(B,r) ||$$

In particolare la privacy del nostro meccanismo dipenderà dalla sensibilità in modo tale che sarà rispettata la $\varepsilon\Delta$ - differential privacy. In questa maniera non sarà più necessario progettare meccanismi che rispettino la differential privacy ma basterà progettare la funzione di punteggio e poi lavorando sulla sensibilità della funzione di score potremo determinare quale differential privacy staremo garantendo per le nostre query.
Difatti lo stesso meccanismo di Laplace non è nientaltro che un meccanismo esponenziale in cui la nostra funzione di punteggio sarà:

$$\text{punteggio}(A, r) = 2 \mid \text{contatore}(A) - r \mid$$

Processo decisionale

Nostro scopo è quello di realizzare una web application che, presi dati dal nostro database integrato, li renda sufficientemente anonimi applicando le metodologie e le tecniche più opportune.
Forti delle conoscenze acquisite sullo *state of the art* dell'anonimizzazione nella comunità scientifica e delle tecniche proposte e analizzate dai nomi più autorevoli a livello internazionale di questo campo.
Per tale motivo e per ridimensionare il nostro problema tramite un approccio più concreto è bene definire alcune priorità strutturali.

- Le risposte date dal nostro programma devono dare le garanzie più forti di anonimizzazione. Il motivo è legato interamente alla natura dei dati, alla legislazione italiana che, come abbiamo visto poc'anzi, tutela ogni forma di privacy persino per coloro i quali abbiano rinunciato alla stessa tramite scelte di vita (i.e. Personaggi famosi, stelle del cinema, musicisti, politici etc) ciò ci spinge verso l'utilizzo del concetto di privacy differenziale.

- La nostra applicazione web sarà accessibile per definizione tramite web e perciò, conoscendo quanto possa essere vulnerabile la nostra posizione e allo stesso tempo mascherabile l'identità di un eventuale aggressore siamo ulteriormente spinti verso la scelta di un meccanismo che utilizzi la privacy differenziale.

- La nostra applicazione web inoltre ci darà come vantaggio, non trascurabile, la possibilità di andare ad interagire direttamente sulle query in ingresso, per analizzarle ancor prima che queste interroghino il database cioè prima che l'informazione, anonimizzata, giunga all'utente. Ciò è di grande importanza perchè ci permette di scegliere un approccio "differenziale" per il nostro caso in esame. Ricordiamo infatti che tutti i meccanismi che garantiscono la privacy differenziale, o suoi derivati, sono meccanismi che vanno ad interagire direttamente sulle queries rendendo i risultati anonimi.

- La scelta e la successiva implementazione di un algoritmo che anonimizzi il nostro database completamente sono problemi estremamente onerosi sia dal punto di vista computazionale, legato al processo stesso di anonimizzazione, che dal punto di vista analitico in quanto non solo estremamente complesso in maniera proporzionale alla complessità del database ma anche "fine a se stesso" in quanto ad ogni successiva modifica, anche minima, dei nostri dati all'interno dello stesso saremmo costretti a ri-computare tutto il processo di anonimizzazione ed, in casi particolari, a rivedere alcune delle tecniche utilizzate.

- L'approccio ingegneristico ai problemi inoltre ci spinge a pensare al futuro ed è di indubbia certezza che, allo stato attuale della conoscenza sull'argomento, la privacy differenziale sia l'alternativa più concreta e sulla quale vengono proposte quotidianamente nuovi spunti di discussione atti a creare nuove tecniche più efficienti o a migliorare e raffinare tecniche già esistenti. La maggior parte del lavoro e della ricerca pubblicata dal 2006 ad oggi sull'argomento dell'anonimizzazione porta nel suo incipit la definizione sostanziale della privacy-differenziale.

- L'importanza della modularità nell'approccio ingegneristico ad un problema, esso non sia solo di natura informatica è sempre stata di importanza sostanziale. La capacità, qualora si debba modificare anche in parti minime un progetto, di adattarsi ai cambiamenti repentinamente, di migliorarne le singole parti, con beneficio del progetto globale, forti di un approccio semplificato e la possibilità di utilizzare un modulo progettato per altri scopi. La modularità è stata studiata ampiamente e le proprie implicazioni e peculiarità sono parte integrante del percorso di studio di ogni ingegnere.

- L'ambiente col quale andremo a interagire, nello specifico MOMIS, può sia effettuare una integrazione virtuale di più database, cioè integrare solamente il risultato di una query agli occhi dell'utente, che materializzarla e perciò renderlo un database "nuovo" frutto della convoluzione delle informazioni provenienti da più sistemi. Questo è di fondamentale importanza per la nostra scelta giacchè ci da una ulteriore spinta verso l'utilità che un approccio modulare potrebbe far guadagnare al nostro software.

- Anche in ottica di progetti futuri l'idea di sviluppare un modulo che renda dati, qualunque sia il database di loro provenienza, ci sembra quella più concreta anche se di maggior difficoltà progettuale ma un approccio step by step ci tornerà sicuramente utile.

- Infine essendo i dati strutturati di tipo clinico dati densi, cioè con bassa percentuale di 0, possiamo evitare di utilizzare tecniche avanzate per il campionamento che ci avrebbero provocato non pochi problemi in fase di sintesi per limitare la mole e i tempi computazionali.

Ambiente di Sviluppo

L'ambiente che si sceglierà di utilizzare per lo sviluppo di questo software è il Java. In particolare si farà utilizzo di Eclipse come ambiente di programmazione/framework. L'interfaccia grafica verrà anch'essa sviluppata in Swing.
Il database integrato dal MOMIS sarà materializzato e istanziato come un database MySQL in particolare verrà utilizzato il server Wamp (wampserver) queste scelte sono state fatte, oltre che per l'utilità e la conoscenza dei sistemi anche per la loro gratuità degli stessi essendo entrambi progetti opensource.
Infine per l'interconnessione fra il database e il nostro programma verrà utilizzato il JDBC, e nello specifico il driver J, per andare ad interagire con i dati tramite Java.

Java

Java è un linguaggio di programmazione orientato agli oggetti, creato da James Gosling e altri ingegneri di Sun Microsystems. Java è un marchio registrato di Oracle. [23]
Java è stato creato a partire da ricerche effettuate alla Stanford University agli inizi degli anni Novanta. Nel 1992 nasce il linguaggio Oak (in italiano "quercia"), prodotto da Sun Microsystems e realizzato da un gruppo di esperti sviluppatori capitanati da James Gosling.[24] Tale nome fu successivamente cambiato in Java a causa di un problema di copyright (il linguaggio di programmazione Oak esisteva già).[25] Per facilitare il passaggio a Java ai programmatori old-

fashioned, legati in particolare a linguaggi come il C++, la sintassi di base (strutture di controllo, operatori e così via) è stata mantenuta pressoché identica a quella del C++[26]; tuttavia, non sono state introdotte caratteristiche ritenute fonti di una complessità non necessaria a livello di linguaggio e che favoriscono l'introduzione di determinati bug durante la programmazione, come l'aritmetica dei puntatori, l'ereditarietà multipla delle classi, e l'istruzione goto. Per le caratteristiche orientate agli oggetti del linguaggio ci si è ispirati al C++ e soprattutto all'Objective C.[27] In un primo momento Sun decise di destinare questo nuovo prodotto alla creazione di applicazioni complesse per piccoli dispositivi elettronici; fu solo nel 1993 con l'esplosione di internet che Java iniziò a farsi notare come strumento per iniziare a programmare per internet. Contemporaneamente Netscape Corporation annunciò la scelta di dotare il suo allora omonimo e celeberrimo browser della Java Virtual Machine (JVM). Questo segna una rivoluzione nel mondo di Internet: grazie alle applet, le pagine web diventarono interattive a livello client (ovvero le applicazioni vengono eseguite direttamente sulla macchina dell'utente di internet, e non su un server remoto). Gli utenti poterono per esempio utilizzare giochi direttamente sulle pagine web ed usufruire di chat dinamiche e interattive. Java fu annunciato ufficialmente il 23 maggio 1995 a SunWorld. Il 13 novembre 2006 la Sun Microsystems ha distribuito la sua implementazione del compilatore Java e della macchina virtuale (virtual machine) sotto licenza GPL. Non tutte le piattaforme java sono libere. L'ambiente Java libero si chiama IcedTea. L'8 maggio 2007 Sun ha pubblicato anche le librerie (tranne alcune componenti non di sua proprietà) sotto licenza GPL, rendendo Java un linguaggio di programmazione la cui implementazione di riferimento è libera.Il linguaggio è definito da un documento chiamato The Java Language Specification (spesso abbreviato JLS). La prima edizione del documento è stata pubblicata nel 1996[28]. Da allora il linguaggio ha subito numerose modifiche e integrazioni, aggiunte di volta in volta nelle edizioni successive. Ad oggi, la versione più recente delle specifiche è la Java SE 7 Edition (quarta).[29] La nostra scelta ricade su Java non solo per motivi legati alla diffusione di questo linguaggio di programmazione ma anche perchè la sua iterazione con MySQL è consolidata e funzionale soprattutto dopo l'acquisizione della Sun da parte della Oracle.

MySQL

MySQL, definito Oracle MySQL, è un Relational database management system (RDBMS), composto da un client con interfaccia a riga di comando e un server, entrambi disponibili sia per sistemi Unix o Unix-like come GNU/Linux che perWindows, anche se prevale un suo utilizzo in

ambito Unix. Dal 1996 supporta la maggior parte della sintassi SQL e si prevede in futuro il pieno rispetto dello standard ANSI. Possiede delle interfacce per diversi linguaggi, compreso un driver ODBC, due driver Java, un driver per Mono e .NET ed unalibreria per python. Il codice di MySQL venne sviluppato fin dal 1979 dalla ditta TcX ataconsult, poi rinominata MySQL AB, ma è solo dal 1996 che viene distribuita una versione che supporta SQL, prendendo spunto da un altro prodotto: mSQL. MySQL AB è stata rilevata da Sun Microsystems nel 2008, mentre nel 2010 quest'ultima è stata acquisita da Oracle Corporation. MySQL fa parte di pacchetti come piattaforma LAMP e WAMP usati per sviluppare in locale siti web, anche in congiunzione con pacchetti software (CMS) come per esempio WordPress, Drupal, Joomla o altri. Fino a qualche anno fa lo sviluppo del programma era opera soprattutto dei suoi sviluppatori iniziali: David Axmark, Allan Larsson e Michael Widenius. Quest'ultimo era il principale autore del codice - oltre che principale socio della società - e tuttora coordina il progetto, tra l'altro vagliando i contributi che pervengono dai volontari. I contributi vengono accettati a condizione che il loro autore condivida i diritti d'autore con la società. Da luglio 2007 la società impiega un centinaio di sviluppatori a tempo pieno. [30]

WAMPserver [31]

WampServer è un pacchetto software che implementa la piattaforma WAMP composta dunque da Apache, MySQL e PHP per Microsoft Windows. Rilasciato per la prima volta il 21 novembre 2007, WampServer è gratuito e libero ed è rilasciato sotto la GNU General Public License. È disponibile nelle due versioni installer (auto-installante) e zip (che non richiede l'installazione), quest'ultima utile per poter avviare WampServer da una chiave USB. Ha come prerequisito un s.o. Microsoft ma ne esiste una versione del tutto uguale,seppure dal nome diverso, anche per altri sistemi operativi quasi Linux o Mac.

JDBC

JDBC (Java DataBase Connectivity)[32], è un connettore per database che consente l'accesso alle basi di dati da qualsiasi programma scritto con il linguaggio di programmazione Java, indipendentemente dal tipo di DBMS utilizzato. È costituita da una API, raggruppata nel package java.sql, che serve ai client per connettersi a un database. Fornisce metodi per interrogare e

modificare i dati. È orientata ai database relazionali ed è Object Oriented. La piattaforma Java 2 Standard Edition contiene le API JDBC, insieme all'implementazione di un bridge JDBC-ODBC, che permette di connettersi a database relazionali che supportino ODBC. Questo driver è in codice nativo e non in Java. [33]

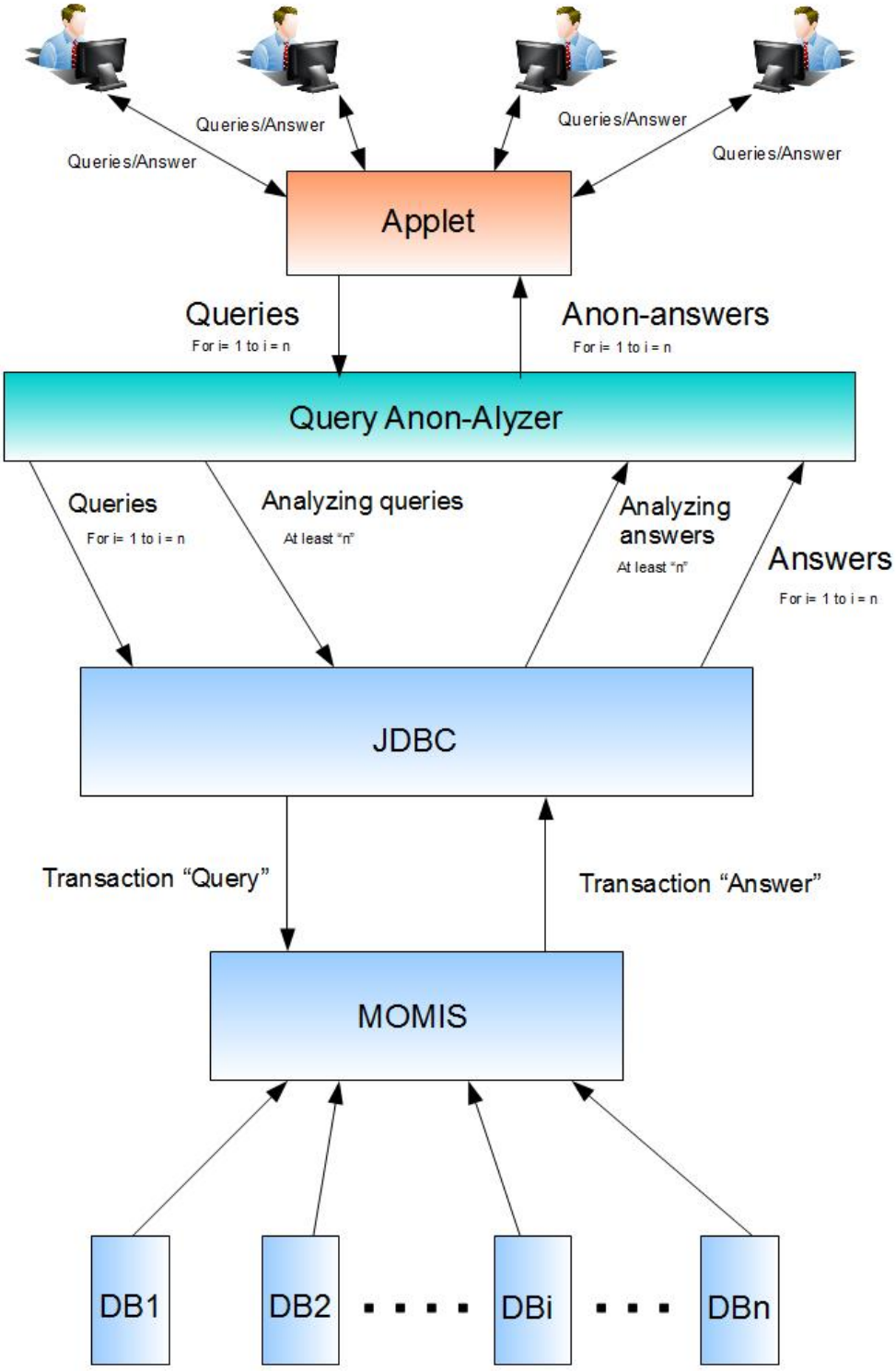
L'architettura di JDBC, così come quella di ODBC, prevede l'utilizzo di un "driver manager", che espone alle applicazioni un insieme di interfacce standard e si occupa di caricare a "run-time" i driver opportuni per "pilotare" gli specifici DBMS. Le applicazioni Java utilizzano le "JDBC API" per parlare con il JDBC driver manager, mentre il driver manager usa le JDBC driver API per parlare con i singoli driver che pilotano i DBMS specifici. Esiste un driver particolare, il "JDBC-ODBC Bridge", che consente di interfacciarsi con qualsiasi driver ODBC in ambiente Windows. JDBC ammette che esistano diverse implementazioni e vengano utilizzate dalla stessa applicazione. L'API fornisce un meccanismo che carica dinamicamente i driver appropriati e li registra nel JDBC Driver Manager. Esso funge da fabbrica di connessioni. Le connessioni JDBC supportano la creazione e l'esecuzione delle istruzioni. Esse possono essere comandi SQL come INSERT, UPDATE, DELETE, interrogazioni come SELECT o chiamate a stored procedure. I tipi di istruzioni supportati sono:

- Statement - l'istruzione viene inviata al database di volta in volta;
- Prepared Statement - l'istruzione viene compilata una sola volta, in modo che le chiamate successive siano più efficienti;
- Callable Statement - usati per chiamare le stored procedure.

I comandi di scrittura come INSERT, UPDATE e DELETE restituiscono un valore che indica quante righe sono state affette (inserite, modificate, cancellate) dall'istruzione. Essi non restituiscono altre informazioni. Le interrogazioni (query) restituiscono un result set (classe ResultSet). È possibile spostarsi nel result set riga per riga (tramite il metodo next()). Si può accedere alle colonne di ogni singola riga chiamandole per nome o per numero. Il result set può essere costituito da un numero qualsiasi di righe. Esso comprende dei metadati che indicano il nome, il tipo e le dimensioni delle colonne. Esiste un'estensione di JDBC che permette, tra le altre cose, l'uso di result set scorribili e di cursori lato client. Si veda la documentazione di Sun Microsystems per maggiori informazioni.

Diagramma concettuale

Troviamo importante cercare di esplicitare graficamente l'idea alla base del nostro programma per anonimizzare dati clinici come primo passo per analizzare e sviluppare il nostro software.



Conclusioni e prospettive future.

Il software sviluppato, come dimostrato ampiamente nella parte in inglese della tesi, affronta la tematica dell'anonimizzazione con un approccio nuovo ma al contempo già sufficientemente collaudato dall'intera comunità scientifica che si sta occupando a pieno regime di questo specifico ramo dell'information technology sin dal 2006.
Nello specifico il programma realizzato anonimizza correttamente diverse tipologie di queries mantenendo un ottimo livello di utilità nelle risposte e proteggendo la privacy attraverso meccanismi laplaciani e esponenziali correttamente implementati in Java.
Sono state inoltre proposte due GUI, una per utenti futuri e una utilizzata per i test e il perfezionamento del software Anon-Alyzer.

Per quanto concerne il futuro, dal punto di vista prettamente teorico, lo sguardo è, a nostro avviso, da rivolgersi alle metodologie di divulgazione di database sintetici ottenuti tramite metodologie di privacy differenziale, sui quali sia possibile operare queries senza preoccuparsi più della sensibilità delle stesse. Ciò potrebbe essere svolto a partire dalle stesse tecniche proposte in questa tesi con particolare attenzione all'analisi di meccanismi esponenziali sufficientemente efficienti.

Per quanto invece concerne il nostro software una implementazione futura, che ne accrescerebbe la dinamicità e l'utilità dovrebbe essere focalizzata sullo studio di tecniche rivolte all'analisi semantica delle query, in grado di fornire in modo più dinamico la sensibilità e la tipologia delle stesse.

Chapter 1 : Introduction to the problem.

1.1 Introduction

> *I don't want to write an autobiography because I would become public property with no privacy left.*
> *- Stephen Hawking*

One of the biggest problems in modern day information technology is the definition and enforcing of privacy, therefore it's logical and practical to search for meanings of privacy in the source of the data, where it's stored, thus in databases.
While being able, in 1997, to produce more transistors in one year that the total numbers of Ants, that ranges from 10^6 to 10^7, we were also able to increase dramatically the quantity and availability of informations stored to the point that in 2007 the estimated user data generated per day reached 8-10 GB of public content and approximately 4TB of "private" content. [1]
This has been made possible by breakthroughs in hardware, faster processing speeds, massive storage capabilities and reliable communication, coupled with rapid advances in information retrieval, data management and data integration.

> *"If US healthcare were to use big data creatively and effectively to drive efficiency and quality, the sector could create more than $300 billion in value every year. "*
> *- McKinsey Global Institute Report*

Almost every action performed by individuals are electronically logged, web searches they make, people they know on social media, places they have been, food they eat and doctors they visit. Companies, institutions and even individuals benefits immensely from this informations through the use of data-mining they can improve services, "understand" better public tendencies and try to accommodate; critical services such as health-care have greatly improved with the digitalization of medical records by both save up money from huge expenses due to physical storing the data to improve the health-care level by studying better the clinical cases with instruments from statistics; Government bases their economic policies on demographic information collected by Census, companies and search engines track user actions in return for personalized web services.[58]

As a side effect the elecronical logs are literally tracking the private lives of individuals and of the entire public, this poses a truly great privacy risk of unauthorized disclosure [2]
In 2002 the governor of Massachussetts released clinical data of patient that used their health-care system, by wrongly assuming the data was anonymized correctly; instead it was proven that almost 87% of the people inside those data could be uniquely identified [3]
In 2004 Choicepoint released private finantial information regarding approximately 145.000 individuals to criminals operating a scam [2]
In 2006 America OnLine "AOL" released logs intended to be anonymous of search queries collected from users to aid information retrieval research in academia however identities of individuals were easily uncovered and by doing so revealed private lives to the whole world [4]

The word privacy came from the latin *"privatus"* that means separated from the rest, deprived of something and the right not to be subjected to unsanctioned invasion, and deprivation, of privacy by the government, corporations or individuals is part of many countries' privacy laws and in some cases costitutions.
Thus the main goal of us as computer scientist is to fulfil the responsability towards those people whose privacy can be impaired while maximizing the utility of the informations we gonna disclose through queries answering.

1.2 Structure of the Thesis

This Thesis will consist of an abstract followed by a summary,which will be the only part wholly written in italian, of the thesis followed by five main chapters plus an Appendix with some theorems demonstrations and descriptions of the software we will be utilizing.
The end of the Thesis will consist of both Acknowledgments and References chapters.

The first chapter will be about an introduction to the theme of anonymization and privacy with a description of what our goal and original contribute will be.
The chapter will then continue with a legislation approach to the matter of privacy in order to focus better on what are the bounds imposed by the italian law.
We will then have an initial brief analysis of the case of study followed by an intuitive approach to anonymization and uncertainty.
The chapter will then end with a brief analysis of the utility that strictly correlates to the matter of anonymization such as no anonymization can be made without preserving utility of the anonymized data.

The second chapter will be about a critical analysis of the "State of the Art" on anonymization, we will initially be giving some basis and rudiments followed by the main approaches to the anonymization such as k-anonimity, l-diversity and t-closeness, we will then focus our attention on differential-privacy and a more generic approach to it, the exponential mechanism.
In the end of the chapter we will then be analysing the state of the art and making some crucial decisions on the approach we will be giving to our case of study.

The third chapter will be about the description of the database focusing on technology choices implied, this chapter in particular will have many references in the Appendix C. The chapter will then go on with a decision making paragraph where we summarize both the knowlede acquired by the analysis of the state of the art and the knowledge on the analysis of the database finally we will be describing the process of designing our project and we'll be giving a name to it accordingly to its main characteristics.

The fourth chapter will be on the algorithm we are utilizing, with transcription of code from the actual software we are developing, focusing on various aspects of anonymization through differential privacy and an example of an exponential mechanism we have implemented.
Moreover there will be query examples and a description and analysis of them.
The chapter will then describe the GUI interface we are using and propose an GUI more user-friendly for future implementations of the software.
Finally there will be examples of queries correctly anonymized by our Anon-Alyzer.

The fifth, and final, chapter will be about conclusions and future works on both the anonymization field of study and future implementation of our software.

1.3 On goal and original contribute

There will be different goals in this thesis. The first goal of this thesis is to initially analyse the state of the art of the anonymization and then a concrete case of study, which will be the FIL's integrated through MOMIS database.
After this initial step the next goal will be to provide a concrete implementation of a web application that will enforce anonymization on some queries while preserving the utility of the data disclosed. We have to clarify that our goal will not be to fully anonymize a database, which is, as we are acquiring knowledge on the anonymization, a very specific, complex and extremely time-spending process that will produce just a specific and very "subjective" anonymization but rather to provide concrete implementation of generic, but really common, type of queries on a concrete database that can be applied to many cases and that will therefore provide a good approximation of a fully anonymization.
Meanwhile the original contribute will be to provide itself a concrete implementation of the state of the art on anonymization while considering a concrete case that needs anonymization. While being one of the major concerns in the international community, expecially in the scientific one related to database management, there's no trace of recent works and implementation, with given extremely rare exceptions, in the italian IT community.

1.4 Privacy Legislation

The recent diffusion of new tecnology has contributed to a dampening of the barrier of privacy, i.e. the traceability of mobile phones or the easy way to recover mails addresses of someone. Nowadays with the birth of Laboratorio Privacy Sviluppo at the Garante for the protection of personal data, the privacy starts being known as "sovereignty over ourselves" an all new interpretation, no longer limited, as in the past, by a right of "not receiving intrusions in our private sphere of life"

Withe the European Convention of human rights, art 8, it was established that there can be no intrusion of a public authority in the handling of this right unless that this intrusion is in accordance with the law and that it's demanded for national security, for the economic wealth of the country, for the defence of the order and for the prevention of crimes, the protection of wealth and morality or the protection of freedom and rights of others. This concept was reported not only in the Schengen Agreement but also n the Charter of Fundamental Rights of the European Union in art. 8 that says:

> ***Article 8. Protection of personal data***
>
> *1.Everyone has the right to the protection of personal data concerning him or her.*
>
> *2.Such data must be processed fairly for specified purposes and on the basis of the consent of the person concerned or some other legitimate basis laid down by law. Everyone has the right of access to data which has been collected concerning him or her, and the right to have it rectified.*
>
> *3.Compliance with these rules shall be subject to control by an independent authority.*

The community source are contained the Directive of European Parliament and the Council 24th october 1995, 95/46/CE pubblished in the GUCEL 281 of 23.11.1995 (p.31)

For what concerns the italian legislation, costitutional fundamentals are in the art 14,15 and 21 Cost, that are about residence, freedon and secrecy of correspondence, freedom of manifestation of personal opinion; but they can also redirect to art 2 of the Costitution,incorporating privacy in inviolable right of a mankind. Rights similar to those of the italian Costituzion can be seen in the

Charter of Fundamental Rights of the European Union, then the Corte di Cassazione with the sent. N4487 of 1956 initially deny the presence of a right to the privacy but then in 1975 with the sentence of Cass. 27 of may 1975 n. 2129 it confirm that right and the Cassazione identify that right in the protection of situations strictly personal and familiar even if outside of the residence [6] This phrase in particular is fundamental for the balance with the right of news.

Cases are many but in 1992 the Tribunal of Rome declare that those who have chosen to be famous have renounced to that part of the right of privacy strictly correlated to the pubblic dimension. So the fundamental line between privacy and pubblic seems to be the popularity of the subject. Nevertheless even subjects very popular retain this right in the meaning of facts that doesn't correlate with the reasons of their popularity.

Another big step forward was in the creation of a suitable set of laws that with the law n°98 of the 21 of february 1989 [5] it's then ratified the Convention of Strasburg adopted in 1981 on the protection and respect and automated processing of private data. In Italy it's still in vigor the Decreto legislativo 30 july 2003 n 196 private data protection that has abrogated the law on privacy of 1996.The Privacy is no longer considered to be the right that nobody intrude "our world" already being determined but it's also intended as the right that everybody can express freely his deepest aspirations and to fullfill them, by drawing freely and completely from his own potential. In this way we are talking about privacy as "self-determination and sovereignty for itself" (Stefano Rodotà) and "the right to be myself" (Giuseppe Fortunato), to be active part and no longer passive of a system evolving, that must bring necessarly to a different relation with the istitutions in the respect of the freedom of eachothers.

While just following this fundamental principle of self-determination the most modern concepts of privacy bring us in the IT to ponder not only on the disclosure of private data of the person but also on the presence or not inside the database of any information even the information that nothing of it is inside the dataset is still to be considered private.

1.5 Case of study

In our specific case we are dealing with a integrated clinical database, product of the convolution of informations coming from many heterogeneous sources, in the specific a database integrated through MOMIS.
The MOMIS (Mediator envirOnment for Multiple Information Sources) is a framework to perform information extraction and integration from both structured and semistructured data sources. An object-oriented language, with an underlying Description Logic, called ODL-I3, derived from the standard ODMG is introduced for information extraction. Information integration is then performed in a semi-automatic way, by exploiting the knowledge in a Common Thesaurus (defined by the framework) and ODL-I3 descriptions of source schemas with a combination of clustering techniques and Description Logics. This integration process gives rise to a virtual integrated view of the underlying sources (the Global Schema) for which mapping rules and integrity constraints are specified to handle heterogeneity. The MOMIS system, based on a conventional wrapper/mediator architecture, provides methods and open tools for data management in Internet-based information systems by using a CORBA-2 interface. MOMIS development started within the INTERDATA and the D2I italian national research project, under the direction of Professor S. Bergamaschi. The research activity continued within the IST project "SEWASIE: Semantic Webs and Agents in Integrated Economies" (2002/2005). MOMIS will be used and extended within the MUR "NeP4B: Networked Peers for Business" project (2006/2009) and the IST-EU RDT project "STASIS (SofTware for Ambient Semantic Interoperable Services)" (2006-2009).

The database we are considering belongs to the Fil Foundation (Fondazione Italiana Linfomi)
The Italian Lymphoma Foundation is an NPO which coordinates the activities carried out in Italy in the field of lymphoma by more than 120 centers located throughout the country. Our aim is improving the centers skills in terms of research and assistance. FIL conducts scientific research activities in the field of lymphoma and, as a non-profit organization, its purposes only concern social solidarity.The Italian Lymphoma Foundation has been the natural evolution of the Italian Lymphoma Intergroup, which was founded in 1993 with the first meeting held in Florence as a group of spontaneous cooperation between clinicians and Italian researchers involved in the study and treatment of lymphoma. Its existence and its assets were later formalized in July 2004 with a notary deed that led to the creation of a foundation with legal status and to the enrolment in the register of NPOs.

The Italian Lymphoma Foundation was born with the aim of make active groups in the study of lymphoma cooperate; afterwards groups have merged and the IIL has decided to become a reference point for their cooperation. FIL Onlus was born in Alessandria on September 30, 2010 with a notarial deed that has marked the transformation of the statute, with the fusion of all groups within one large organization. FIL encourages prospective and retrospective studies in order to answer questions that require a large case report. Therefore it intends to foster partnerships with international organizations, to which it is configured as a partner of choice. FIL also seeks to organize and improve services for the diagnosis and treatment of other lymphoproliferative disorders and promotes the formation of the Italian Registry of lymphoma (RIL). FIL Onlus develops projects for the diffusion of information on lymphoma, in order to raise the awareness of the problem and help patients and relatives; coordinate research groups in the fight against lymphoma; constitute the scientific, organizational and legal framework for managing clinical trials on lymphoma; coordinate the efforts of researchers to create one large Italian cooperative group for the fight against lymphoma and collaborate with European groups as for international studies on lymphoma. FIL Onlus cooperates with the International Extranodal Lymphoma Study Group (IELSG). Other studies on international prognostic factors are in progress and FIL is the promoter (F2, T-cell project, early-PET validation); new collaborations with the European Organisation for

Research and Treatment of Cancer (EORTC) for localized Hodgkin and with Mantle Cell Network for the mantle lymphoma have started.
Therefore the database will include not only private data of the patient but the documents of patient being hospitalized and also the entire case clinical history of the patient. For this reason it's mandatory the absolute certainty of anonymity of the person in the case of disclosure of data for statistical purpose.

1.6 Anonymization vs Uncertainty

"... Last week AOL did another stupid thing ... but, at least it was in the name of science..."
Alternet, August 2006

In 2002 the governor of Massachusset has discosed data of clinical patient of their health-care system, assuming errouneosly that the data were correctly anonymized but it was proved that roughly 87% of the people inside the data could be identified and univocally de-anonymized.[3]
In 2004 was found that Choicepoint had given private finantial informations on more than 145.000 people to criminals operating a scam [2]
In 2006 America OnLine "AOL" released logs, believing they were correctly anonymized, of websearches nevertheless the identity of people was easily uncovered and many aspects of the private life ot them disclosed to the whole world [4]
In 2008 Netflix discosed over 100M of ratings from over 480k users on 18k movies in order to predict ratings of unlabeled examples. All direct customer information was removed, only subset of full data; dates modified; some ratings deleted, movie title and year published in full but it was claimed vulnerable with attack links to IMDB where same users also rated movies [14]

This is just a brief list of cases, easily the most famous, of violation of privacy without the use of any external instruments in order for attackers to gain private data, famous and of data itself that released data without correctly anonymizing it; and the loss while being mostly on privacy for people was also a big loss of credibility and money for the owner of the db.

The objectives for Anonymization is to prevent, with high confidence, inference of associations, to prevent linking of sensitive information to an individual but also to prevent inference of presence of an individual in the data set.This cannot be analysed alone but we have also to assume that the attacker can have background knowledge, such as facts about the dataset, and domain knowledge as in broad properties of data itself.

Moreover the uncertainty on data represents multiple possible worlds, each possible world corresponds to a database with a given probability attached to it.
There can be a possibilistic interpretation or a probabilistic interpretation and as we can se we will rely on the second of this interpretation to define a more solid and mathematical definition of what we call anonymous.

1.7 Utility

Another crucial point of debate lies around the concept of utility, without utility the whole anonymization process is meaningless. An empy data set respect perfectly privacy but has no utility at all while the original data has full utility but no privacy.
What we know for sure is that the concept of utility strictly correlates with privacy/anonymity, for instance a completely anonymous dataset has no utility at all while a dataset with full utility has no privacy preserved at all.
The meaning of utility, as for our settings, depends mostly on the application itself, we can define surrogates measures and try to optimize them maybe as functions of the loss of information between the original data and the anonymized one or we can just compute the utility by empircally evaluating with reasonable workload or sample of the workload on the results then we can compare results on the anonymized data with the results on the original data.

Chapter 2: Anonymization and Privacy

2.1 Technical Terms definition

It's imperative to define some technical terms in order to describe better more complex concept:

Identifier: it's an attribute that uniquely identifies a person inside the data, e.g. SSN (Social Secury Number).
Usually the first step is to remove those attribute but in many cases, such the AOL case, it's certainly not enough.

Quasi-Identifier (QI) : Are attributes whose composition partially identifies an individual in a dataset, e.g. DOB+Sex+ZIP is an unique composition for 87% of US Resident and therefore identifies them in 87% of cases.

Sensible attribute (SA) : The association of this attribute it's that which we want to hide, it depends on the data and on the type of database e.g. Salary in Census is considered sensitive, the Case History in a clinical database[10].

2.2 K-anonymity

SSN	*ZIP*	*Age*	*Nationality*	*Disease*
555-81-5347	12423	22	Italian	Heart
661-91-5348	17832	32	Russian	Viral
555-31-3478	12378	23	Japanese	Viral
856-31-1254	15340	19	American	Heart
511-71-1110	14450	44	American	Cancer
522-31-3232	19700	59	American	Heart
345-31-2323	13547	44	Indian	Cancer
555-21-2321	13458	60	American	Heart
125-31-4222	17459	81	Japanese	Cancer
537-31-3273	14907	57	Italian	Viral
432-11-8293	15439	92	Russian	Heart

Removing SSN, which is a unique identifier, will bring us to this situation:

ZIP	*Age*	*Nationality*	*Disease*
12423	22	Italian	Heart
17832	32	Russian	Viral
12378	23	Japanese	Viral
15340	19	American	Heart
14450	44	American	Cancer
19700	59	American	Heart
13547	44	Indian	Cancer
13458	60	American	Heart
17459	81	Japanese	Cancer
14907	57	Italian	Viral
15439	92	Russian	Heart

But is this new configuration sufficiently anonymous ? Of course it isn't has we have seen in the AOL case just removing the unique identifier, by suppressing it or generalizing it,isn't enough because at least 87% of the people in the database can be re-identified by just linking DOB.Sex and Zip code and similar attacks can be made in order to re-identifies persons with the data we have, for instance Nationality Age and Zip.
This technique is called Linking-attack or Linkage-attack and it's what we've seen in cases such as the Massachusset and AOL cases.

In order to prevent this kind of attack we will rely on the concept of k-anonymity, in SQL a table T is k-anonymous if each

```
SELECT COUNT (*)
FROM T
GROUP BY Quasi-Identifier
```

is $\geq K$

This paramenter indicates the level of anonymity of the table.

In a more formal way we definine:

k-anonymity: Table T satisfies k-anonymity with regard to quasi-identifier QI iff each tuple in (the multiset) T[QI] appears at least k times

And we define:

k-anonymization: Table T' is a k-anonymization of T if T' is a generalization/suppression of T , and T' satisfies k-anonymity

Using the previous example of a table we can se how this new table, obtained by the generalizaion of the last 3 digits of the ZIP code is a k-anonymity table , wrt T[ZIP] .

ZIP	Age	Nationality	Disease
12***	22	Italian	Heart
17***	32	Russian	Viral
12***	23	Japanese	Viral
15***	19	American	Heart
14***	44	American	Cancer
19***	59	American	Heart
13***	44	Indian	Cancer
13***	60	American	Heart
17***	81	Japanese	Cancer
14***	57	Italian	Viral
15***	92	Russian	Heart

In particular this new table it's a 2-anonymization of the original table.
A k-anonymized table T' represents the set of all “possible worlds“ tables Ti s.t. T' is k-anonymization of Ti and the table T, from which T' was originally derived is just one of the possible worlds.

ZIP	Age	Nationality	Disease
12***	22	Italian	Heart
17***	32	Russian	Viral
12***	23	Japanese	Viral
15***	19	American	Heart
14***	44	American	Cancer
19***	59	American	Heart
13***	44	Indian	Cancer
13***	60	American	Heart
17***	81	Japanese	Cancer
14***	57	Italian	Viral
15***	92	Russian	Heart

→

ZIP	Age	Nationality	Disease
12423	22	Italian	Heart
17832	32	Russian	Viral
12378	23	Japanese	Viral
15340	19	American	Heart
14450	44	American	Cancer
19700	59	American	Heart
13547	44	Indian	Cancer
13458	60	American	Heart
17459	81	Japanese	Cancer
14907	57	Italian	Viral
15439	92	Russian	Hear

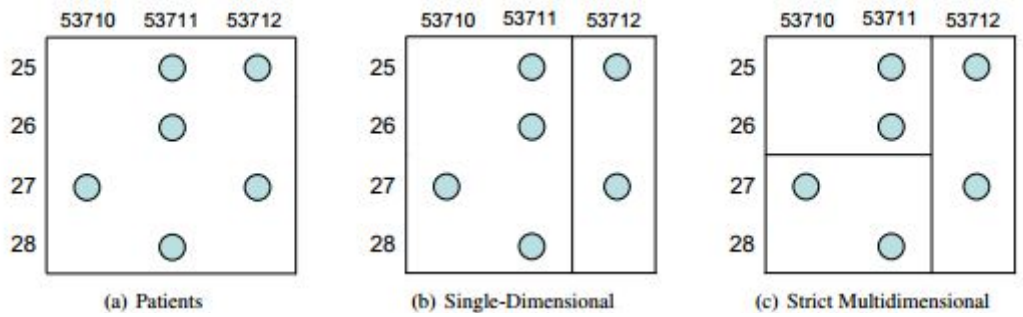
In particular we can observe that if Q = {B, Z, S} is a quasi-identifier, then Q + {N} is also a quasi-identifier. We need to guarantee k-anonymity against the largest set of quasi-identifiers

2.3 Two k-anonimous algorithms

There are many algorithms that satisfy k-anonymity such as Mondrian[15] and Incognito[9].

Mondrian takes idea from spatial kd-tree construction so that QI tuples points in a multi-dimensional space, there are hyper-rectangles with ≥ k points in order to infer k-anonymous groups, we then choose axis-parallel line to partition point-multiset at median.

We only need to compute one good-enough dimensional generalization, it will use local recoding to explore a larger search space and will treat all attributes as ordered chosing partition boundaries[10].



(a) Patients (b) Single-Dimensional (c) Strict Multidimensional

While Incognito relies on a generalization described by a domain vector, it computes all minimal full-domain generalization taking ideas from data cube computation and association rule mining The main intuition is that, in order to have an efficient computation,two properties must be fullfilled:

- *Subset Property* : If table T is k-anonymous wrt a set of attributes Q , then T is k-anonymous wrt any set of attributes that is a subset of Q
- *Generalization Property* : If table T_2 is a generalization of table T_1 , and T_1 is k-anonymous, then T_2 is k-anonymous.

As we can see from an example:

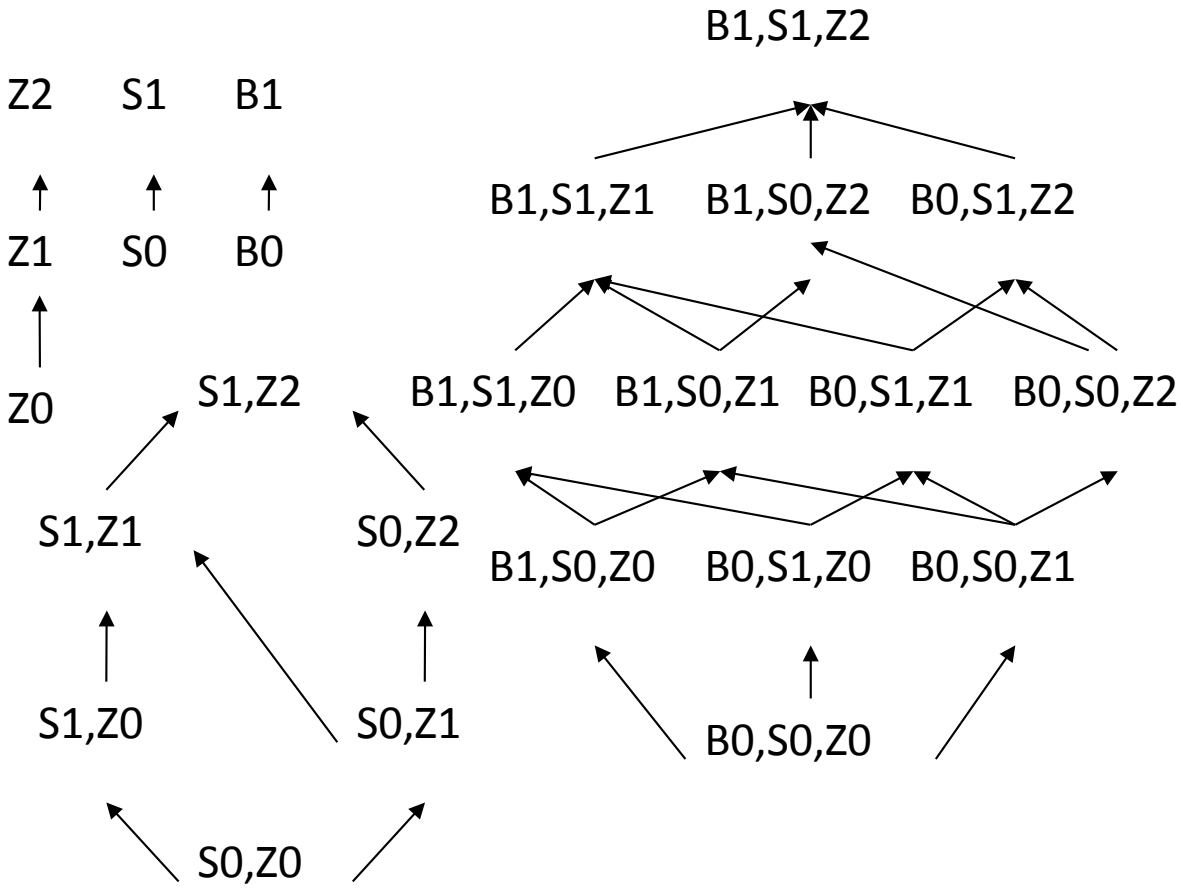
DOB	Sex	ZIP	Salary
1/21/76	M	53715	50,000
4/13/86	F	53715	55,000
2/28/76	M	53703	60,000
1/21/76	M	53703	65,000
4/13/86	F	53706	70,000
2/28/76	F	53706	75,000

B0, S1, Z2 →

DOB	Sex	ZIP	Salary
1/21/76	*	537**	50,000
4/13/86	*	537**	55,000
2/28/76	*	537**	60,000
1/21/76	*	537**	65,000
4/13/86	*	537**	70,000
2/28/76	*	537**	75,000

- B0={1/21/76, 2/28/76, 4/13/86} → B1={76-86}
- S0={M, F} → S1={*}
- Z0={53715,53710,53706,53703}→ Z1={5371*,5370*}→ Z2={537**}

We have different vectors taking us from an initial “status“ where we are not anonymous to a final statous where the graph respect is k-anonymous.



Status S0,Z0,Z1,B0 aren't "anonymous" while Z2,S1,B1are the way we are going to reach them depends "just" on the path taken.

As for our case, we can notice how, with this two kind of algorithms we are obliged to compute a full database anonymization in order to infer a k-anonymity on our data.

Moreover as we will see the main problem is that what we thought of being anonymous isn't.

2.4 L-diversity

The main idea behind l-diversity is that the big problem of a k-anonymous database is that there wont be so much diversity in the T[QI], lack of diversity of SA values implies that in large fraction of possible worlds, some fact is true, which can violate privacy.

ZIP	*Age*	*Nationality*	*Disease*
12***	22	Italian	Viral
17***	32	Russian	Viral
12***	23	Japanese	Viral
15***	19	American	Heart
17***	44	American	Cancer
15***	59	American	Heart
13***	44	Indian	Cancer
13***	60	American	Heart
17***	81	Japanese	Cancer
12***	22	Italian	Viral
15***	92	Russian	Heart

For instance in a table like this one we can see how the first patient is italian, has age 22, a zip generalized in 12*** and had a Viral disease.This is a violation of privacy cause we now know that all the italian in this dataset 22 years old had viral disease.

> *l -Diversity Principle: a table is l -diverse if each of its QI groups contains at least l "well-represented" values for the SA.* [11]

There can be different definitions of l-diversity that are all based on the formalization on the definition of "well preserved".

- *Entropy l-diversity*: for each QI group g , entropy(g) ≥ log(*l*)
- *Recursive (c,l)-diversity* : for each QI group g with m SA values, and r_i the i 'th highest f

$$\text{frequency, } r_1 < c\,(r_l + r_{l+1} + \ldots + r_m)$$

- *Folk l-diversity* : for each QI group g , no SA value should occur more than 1/*l* fraction of the time = Recursive(1/*l* , 1)-diversity

In order to make a k-algorithm to respect l-diversity we can just change the k-anonymity test with a generalizaded version with the l-diversity test.

2.5 T-Closeness

T-closeness came out in 2007 as an observation on the fact that even if SA values are distinct they can still be sematically similar.

ZIP	*Age*	*Nationality*	*Salary*
12***	22	Italian	12000
17***	32	Russian	13000
12***	23	Japanese	50000
15***	19	American	123000
17***	44	American	13000
15***	59	American	32000
13***	44	Indian	40000
13***	60	American	92000
17***	81	Japanese	22000
12***	22	Italian	12001
15***	92	Russian	22000

In this example the first and the last italian in the table have ZIP correctly generalized but having Salary 12000 and 12001. Those are two distinct number for our database but still we can imply that both have a salary ~50000 and that's a loss of privacy.

t-Closeness Principle : a table has t-closeness if in each of its QI groups, the distance between the distribution of SA values in the group and in the whole table is no more than threshold t [16]

2.6 Permutation

Another viable way lies around the permutation, since generalizing/suppressing adds a lot of uncertainty to our data resulting in inaccurate aggregate analysis [17]
The key idea is that we have to weaken the link between QI and SA association so we'll partition the private data into groups of tuples, permute SA values with respect to QI values of each group and in the end for individuals known to be in private data we have the same privacy guarantee as a generalization.

2.7 Differential Privacy [13]

The main idea of Differential Privacy is that, since we have proven over and over that our "anonymity" concept can be broken by various attacks, and since new kind of attacks are discovered every day, we need to change the definition of what is anonymous for us.
Therefore we will rely on a more mathematical approach, derived mostly from Cryptography.
Differential privacy has recently emerged as the de facto standard for private data release.
This makes it possible to provide strong theoretical guarantees on the privacy and utility of released data.[20]

Differential privacy is achieved by introducing randomnessinto query answers. The original algorithm for achieving differential privacy, commonly called the Laplace mechanism ,returns the sum of the true answer and random noise drawn from a Laplace distribution. The scale of the distribution is determined by a property of the query called its sensitivity: roughly the maximum possible change to the query answer induced by the addition or removal of one tuple. Higher sensitivity queries are more revealing about individual tuples and must receive greater noise.If an analyst requires only the answer to a single query about the database, then the Laplace mechanism has re-cently been shown optimal in a strong sense [21]

The idea behind this concept is quite simple and yet really fundamental: If one individual is considering lying about her data to a data collector, the result of the anonymization algorithm will not be very different, whether or not the individual is in our database.

We model the database as a vector of n entries from some domain D. We typically consider domains D of the form {0, 1}d or Rd. The Hamming distance dH(·, ·) over Dn is the number of entries in which two databases differ. Our basic definition of privacy requires that close databases correspond to close distributions on the transcript. Specifically, for every transcript, the probabilities of it being produced with the two possible databases are close. We abuse notation somewhat and use Pr[A = a] to denote probability density for both continuous and discrete random variables.

Definition 1: *A mechanism is -indistinguishable if for all pairs x, x′ ∈ Dn which differ in only one entry, for all adversaries A, and for all transcripts t:*

$$\left| \ln\left(\frac{\Pr[\mathcal{T}_{\mathcal{A}}(\mathbf{x}) = t]}{\Pr[\mathcal{T}_{\mathcal{A}}(\mathbf{x}') = t]}\right) \right| \leq \epsilon.$$

We sometimes call the leakage. When is small, ln(1+) , and so the definition is roughly equivalent to requiring that for all transcripts t, the requirement of Definition 1 is much more stringent than statistical closeness: one can have a pair of distributions whose statistical difference is arbitrarily small, yet where the ratio in Eqn. 1 is infinite (by having a point where one distribution assigns probability zero and the other, non-zero). We chose the more stringent notion because (a) it is achievable at very little cost, and (b) more standard distance measures do not yield meaningful guarantees in our context, since, as we will see, the leakage must be non-negligible. As with statistical closeness, Definition 1 also has more “semantic” formulations;

As we will next show, it is possible to release quite a lot of “global” information about the database while satisfying Definition 1. We first define the Laplace distribution, Lap(). This distribution has density function h(λ) / exp (−|y|/λ), mean 0, and standard deviation .
Suppose x ∈ {0, 1}n, and the user wants to learn f(x) =Pi xi, the total number of 1’s in the database. Consider adding noise to f(x) according to a Laplace distribution:

$$\mathcal{T}(x_1, \ldots, x_n) = \sum_i x_i + Y, \quad \text{where } Y \sim \text{Lap}(1/\epsilon).$$

This mechanism is -indistinguishable. To see why, note that for any real numbers y, y' we have

$$\frac{h(y)}{h(y')} \leq e^{\epsilon|y-y'|}.$$

For any two databases x and x' which differ in a single entry, the sums f(x) and f(x') differs by one. Thus, for t∈R, the ratio

$$\frac{\Pr(\mathcal{T}(\mathbf{x})=t)}{\Pr(\mathcal{T}(\mathbf{x}')=t)} = \frac{h(t-f(\mathbf{x}))}{h(t-f(\mathbf{x}'))}$$

is at most $e^{\varepsilon|f(x)-f(x0)|} \leq e^{\varepsilon}$, as desired.

> *<u>Definition 2:</u> Sensitivity The L_1 sensitivity of a function f : Dn ! R^d is the smallest number S(f) such that for all x, x′ ∈ Dn which differ in a single entry,*
>
> $$\frac{\|f(\mathbf{x})-f(\mathbf{x}')\|_1}{\mathrm{d}_H(\mathbf{x},\mathbf{x}')} \leq S(f).$$

One can define sensitivity with respect to any metric on the output space.
Intuitively, it's the sum of the worst case difference in answers that can be caused by adding or removing someone from a data set [22].
This sensitivy is of much importance to us cause, as we will see afterwards, it's the key to analyse and calibrate our noise in order to enforce ε-differential privacy on our queries.

Recall that if the noise Y is drawn from the Laplace distribution, then h(y)/h(y') is at most $e^{|y-y'/\lambda}$. A similar phenomenon holds in higher dimension. If Y is a vector of d independent Laplace variables, the density function at y is proportional to exp(−||y||1/λ). A simple but important consequence is that the random variables z + Y and z' + Y are close in the sense of the definition of differential privacy: for all t R^d,

$$\frac{\Pr(z + Y = t)}{\Pr(z' + Y = t)} \in \exp(\pm \frac{\|z - z'\|_1}{\lambda}).$$

Thus, to release a (perturbed) value f(x) while satisfying privacy, it suffices to add Laplace noise with standard deviation S(f)/ε in each coordinate!

Proposition: For all $f : D^n \rightarrow \mathbb{R}^d$, the following mechanism is ε-indistinguishable:
Sanf (x) = f(x) + (Y1, ..., Yd) where the Yi are drawn i.i.d. from Lap(S(f)/ε)

The proposition is actually a special case of the privacy of a more general, possibly adaptive, interactive process. Before continuing with our discussion, we will need to clarify some of the notation to highlight subtleties raised by adaptivity. Specifically, adaptivity complicates the nature of the "query function", which is no longer a predetermined function, but rather a strategy for producing queries based on answers given thus far. For example, an adaptive histogram query might ask to refine those regions with a substantial number of respondents, and we would expect the set of such selected regions to depend on the random noise incorporated into the initial responses[13].

Recalling our notation, a transcript t = [Q1, a1,Q2, a2 . . . ,Qd, ad] is a sequence of questions and answers. For notational simplicity, we will assume that Qi is a well defined function of a1, . . . , ai−1, and that we can therefore truncate our transcripts to be only a vector t = [a1, a2, . . . , ad]. For any transcript t, we will let ft : D → R^d be the function whose ith coordinate reflects the query Qi, which we assume to be determined entirely by the first i−1 components of t. As we now see, we can bound the privacy of an adaptive series of questions using the largest diameter among the functions f_t.

Consider a trusted server, holding x, which receives an adaptive sequence of queries f1, f2, f3, ..., fd, where each fi : D^n → R. For each query, the server San either (a) refuses to answer, or (b) answers fi(x)+Lap(λ). The server can limit the queries by refusing to answer when S(ft) is above a certain threshold. Note that the decision whether or not to respond is based on S(ft), which can be computed by the user, and hence is not disclosive.

Theorem **:** *For an arbitrary adversary A, let ft(x) : $D^n \rightarrow R^d$ be its query function as parameterized by a transcript t. If = maxt S(ft)/λ , the mechanism above is ε-indistinguishable.*[1]

This is of a great importance to us because we can now model our laplacian noise, in order to calibrate our algorithm's utility, confident that the result will still enforce ε-indistinguishable or we can say *ε-differential privacy* on the queries.

In the end we can define a more "closer", as in closer to our problem. definition of differential privacy.

We defne two neighbour datasets D1 and D2 such as D1 and D2 differs by just one tuple t, written ||D1-D2||= 1 (in some cases it means that the tuple t is present just in one of the dataset in others that it differ from one dataset to another in some values and that both definition enforce the same level of privacy)

Definition: : A is ε-differentially private if, for all neighbors x, x', for all subsets S of outputs

$$Pr[A(D1) \in S] \leq e^{\varepsilon} Pr[A(D2)] \in S]$$

We can now design an algorithm that evaluate the sensitivity of a query or of a group of queries and then, based on that parameter, sample from a Laplacian distribution a random value that added to our "true" query answer gives us an answer that satisfy ε-differential privacy.

1 The proof for this theorem will be in the theorem proofs appendix at the end of the Thesis.

2.8 Exponential Mechanism [35]

The exponential mechanism is a technique for designing differentially private algorithms developed by Frank McSherry and Kunal Talwar. [34]
Most of the initial research in the field of differential privacy revolved around real valued functions which have relatively low sensitivity to change in the data of a single individual and whose usefulness is not hampered by small additive perturbations. A natural question is what happens in the situation when one wants to preserve more general sets of properties. The Exponential Mechanism helps to extend the notion of differential privacy to address these issues. Moreover, it describes a class of mechanisms that includes all possible differentially private mechanisms.

Let D be the domain of input datasets.
Let R be the range of \noisy" outputs.
Let $\mathbb{R}$ be the real numbers.
We start by defining a scoring function score : D x R $\rightarrow \mathbb{R}$ that takes in a dataset A $\in$ D and output r $\in$ R and returns a real-valued score; this score tells us how "good" this output r is for this dataset A, with the understanding that higher scores are better.
The exponential mechanism E takes in the scoring function score and a dataset A parameter ε and does the following:

$$E(A;score;\varepsilon) = \text{output r with probability proportional to } \exp\left(\frac{\varepsilon}{2}\ score(A,r)\right)$$

Next we need to define the sensitivity of a scoring function. The sensitivity Δ tells us the maximum change in the scoring function for any pair of datasets A, B such that $|A \circ B| = 1$. More formally:

$$\Delta = \max_{r,A,B \text{ dove } |A \circ B|=1.} || score(A,r) - score(B,r) ||$$

We can now demonstrate that the privacy of the exponential mechanism depends on the sensititivy Δ .

Theorem. If the score function has sensitivity Δ , the mechanism E(A; score; ε) is εΔ -differentially private.

Proof. First, since we know the score has sensitivity Δ , we know that for any A, B where $|A \circ B| = 1$, it follows that

$$-\Delta \leq \text{score}(A; r) - \text{score}(B; r) \leq \Delta$$

What if we have X, Z such that $|X \circ Z| = a$? How can we bound the score function?
Well, imagine we had a + 1 intermediate databases, where $Y_1 = X$ and $Y_{a+1} = Z$ and for each i = 1 ... a we have that $|Y_i \circ Y_{i+1}| = 1$. Then, we can take the telescoping sum:

$$\text{score}(X,r) - \text{score}(Z,r) = (\text{score}(Y_1,r) - \text{score}(Y_2,r)) + (\text{score}(Y_2,r) - \text{score}(Y_3,r)) + ... + (\text{score}(Y_a,r) - \text{score}(Y_{a+1},r))$$

Repeatedly applying the first disequation to the telescoping sum we get:

$$-a\Delta \leq \text{score}(X; r) - \text{score}(Y; r) \leq a\,\Delta$$

Now, let's take two databases X, Y where $|X \circ Y| = a$. We can write:

$$\begin{aligned}
\frac{\Pr[M(X) = r]}{\Pr[M(Y) = r]} &= \frac{\frac{\exp(\frac{\epsilon}{2}\text{score}(X,r))}{\int_{\rho \in \mathcal{R}} \exp(\frac{\epsilon}{2}\text{score}(X,\rho))d\rho}}{\frac{\exp(\frac{\epsilon}{2}\text{score}(Y,r))}{\int_{\rho \in \mathcal{R}} \exp(\frac{\epsilon}{2}\text{score}(Y,\rho))d\rho}} \\
&= \frac{\exp(\frac{\epsilon}{2}(\text{score}(X,r) - \text{score}(Y,r)))}{\int_{\rho \in \mathcal{R}} \exp(\frac{\epsilon}{2}(\text{score}(X,\rho) - \text{score}(Y,\rho)))d\rho} \\
&\leq \frac{\exp(\frac{\epsilon}{2}a\Delta)}{\exp(-\frac{\epsilon}{2}a\Delta)\int_{\rho \in \mathcal{R}} d\rho} \qquad \text{(Apply equation (1))} \\
&\leq \exp(\epsilon a \Delta) \qquad \left(\text{If } \int_{\rho \in \mathcal{R}} d\rho \geq 1\right) \\
&= \exp(\epsilon \Delta |X \oplus Y|)
\end{aligned}$$

which gives us εΔ-differential privacy. So we're done[36]. □

By proving this theorem from now on we no longer have to design mechanisms, we need only to design the scoring function! Then, working out the sensitivity of the scoring function, we can determine the differential privacy guarantee.

Even the Laplace mechanism can be seen as a instance of the exponential mechanism.
We can capture the laplace noise mechanism by setting the score function to be

$$score(A, r) = -2|count(A)\ \ r|$$

To see how this works, notice first that with this scoring function, the exponential mechanism becomes

$$E(A;score;\varepsilon) = \text{output r with probability proportional to } \exp(\varepsilon\,|score(A) - r|)$$

We can equivalently write this as

$$Pr[E(A),score,\varepsilon) = r] \ \propto\ \exp(-\varepsilon|count(A)\ - r|)$$

Meanwhile recall that for the laplace mechanism we have

$$Pr[count(A) + Lap(\frac{1}{\varepsilon}) = r] = Pr[Lap(\frac{1}{\varepsilon}) = r - count(A)] = \frac{\varepsilon}{2}\exp(-\varepsilon|count(A) - r|)$$

and so we can see the two are the same.

2.9 Critical analysis of the “State of the Art”

Given the knowledge acquired so far we are able to analyse the majority of the techniques developed to enforce anonymization on data disclosed.
First of all we can observe that there are two distinct approaches to the problem of anonymization that we can distinguish into :

- approaches towards persistent database's anonymization
- approaches towards dynamic queries response anonymization

In the first category we can place methods such as the algorithm that enforce k-anonimity, l-diversity, t-closeness to a dataset.
These methods are persistent techniques that alter the nature itself of the database in order to disclose to a third party the whole data, confidents that its anonymicity is both robust and resilient and yet the utility of the anonymized data is still good-enough.
While there are concrete implementations of these methods for almost each one of them are presented possible or even real attacks examples, moreover it's to be noted that, while searching for anonymization methodologies, after 2006 almost all the international scientific community has changed its course towards differential-privacy.
Finally these methods imply large-scale processing directly proportional to the size of the databse itself, therefore in cases such as regional or national database they could require not only high-end workstation but a directly proportional time at any given time the databse is changed and needs a new anonymization.

The second category comprehends methods that revolve around the differential-privacy concept, these methods take a distinct parting from the past.
Firstly they change the concept itself of what we are seeking, the "abrupt" change from Anonymity to Privacy relies on the principle that if, while trying to pursue an objective, your goal takes more distance at every step maybe it's the goal that needs to be redefined not the process to pursue it.So it's the concept of Anonymity that needs to be redefined to a more generic concept of Privacy.

For instance in 1977 Dalenius T. defined what's provability for a database that doesn't disclose any information on the individuals inside of it by "*Anything that can be learned about a respondent from the statistical database can be learned without access to the database.*"[55] which is *"Unhappily, Unachievable, both for not serios and serious reasons"*[56] because it cannot be achieved unless it's from a database with no utility at all.

Moreover this category includes methods that operates on queries rather than enforcing anonymicity on a dataset in a persistent way, they enforce differential-privacy on the data disclosed by the query itself perturbing the answer with stochastic noise.

Differential-privacy enforce anonymity not only on individual that are present in the data but also to those that aren't in the data.

While enforcing privacy by acting on queries, rather than on the dataset, we have a more dynamic approach that can be utilized on multiple systems once all queries typologies are examined, but not only on multiple systems also on multiple type of systems such as integrated both virtually and materialized until the query language is the same as the one processed with differential privacy.

Finally, since 2006, the year when the differential-privacy concept emerged from the international scientific community, the whole community is focused almost entirely on verifying first and then trying to improve these methods and presenting entirely new ones at every seminar. Even multinational such as Microsoft has started projects and are developing software in this direction such as PINQ[50] that stands for privacy integrated queries and that uses differential privacy as his main istrument to enforce privacy on disclosed data.

Differential-privacy has become a de facto standard for most of the community for both its dynamically and resilience to data re-identification.

Finally we came to know that, on behalf of (ln3)-differentially private algorithms[57] :

- Randomized Response has an Error $\sim \Theta\left(\frac{1}{\sqrt{n}}\right)$
- Laplacian Mechanism has an Error $\sim \Theta\left(\frac{1}{n}\right)$

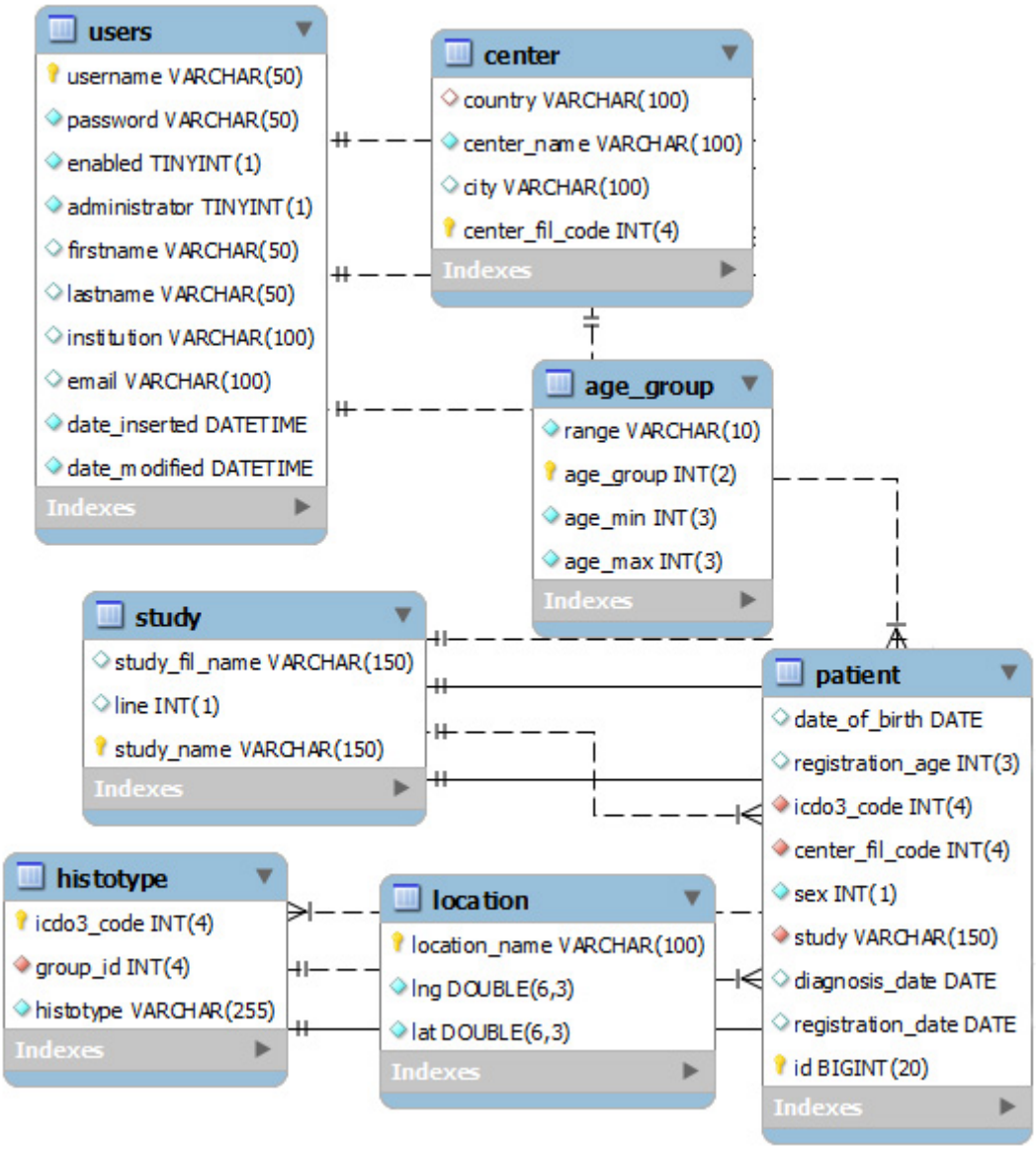
So as the number of tuples grows the Laplacian Mechanism algorithm error is exponentially lower than the error of Randomized Response algorithm with both having the exactly same amount of privacy.

Chapter 3: Decision Making

3.1 Description of the Database

The integrated clinical database we are going to work on, as we already have said, is a materialization of an integration process, made by MOMIS, on multiple data source and belongs to FIL. This is of a great relevance to us because through the materialization of a integrated database we will be able to analyse its Entity–relationship model

Thus it will contain anagraphical and clinical data of the patients, for our develop point of view this translates in many alphanumerical and numerical attributes but also date attributes.
As we can see from an image taken from MySQL workbench, the portion of the ERR we will be focus on will be :



The more crucial attributes that we will be going to disclose are the dates attribute:{ diagnosis date, registration_date, date_inserted} the alphanumerical attributes {firstname,lastname,istitution,center_name,city,histotype} and the numerical attributes {age_group,age_min,age_max} those will be the main attributes on which we are going to analyse and anonymize queries.
We can also analyse the sensitivity of those attributes by analyzing their cardinality and enstablish which is the "worst case scenario" that we have to consider in order to produce a coherent sensibility and enforce differential privacy correctly.
On the other hand we can see that the develop of an algorithm that will enforce on the dataset for instance a k-anonimity could be a very difficult and complex problem expecially with regards to the final utility of the dataset disclosed as "sufficiently" anonymous.

3.2 Data Integration

The Data integrations concept involves the combination of data residing in different sources and that provides to the user a unified view of the data[51]. This process becomes significant in a variety of situations, which include both commercial (when two similar companies need to merge their databases) and scientific (combining research results from different bioinformatics repositories, for example) domains. Data integration appears with increasing frequency as the volume and the need to share existing data explodes.It has become the focus of extensive theoretical work, and numerous open problems remain unsolved. In management circles, people frequently refer to data integration as "Enterprise Information Integration" (EII). [52]
Consider a web application where a user can query a variety of information about cities (such as crime statistics, weather, hotels, demographics, etc.). Traditionally, the information must be stored in a single database with a single schema. But any single enterprise would find information of this breadth somewhat difficult and expensive to collect. Even if the resources exist to gather the data, it would likely duplicate data in existing crime databases, weather websites, and census data.

A data-integration solution may address this problem by considering these external resources as materialized views over a virtual mediated schema, resulting in "virtual data integration". This means application-developers construct a virtual schema— the mediated schema — to best model the kinds of answers their users want. Next, they design "wrappers" or adapters for each data source, such as the crime database and weather website.

These adapters simply transform the local query results (those returned by the respective websites or databases) into an easily processed form for the data integration solution. When an application-user queries the mediated schema, the data-integration solution transforms this query into appropriate queries over the respective data sources. Finally, the virtual database combines the results of these queries into the answer to the user's query.[52]

Modern enterprises are often organized as "virtual networks" where the nodes, i.e, enterprises, operate through inter-enterprise cooperative processes.The enterprises hold pripirietary infromation systems,i.e,legacy systems, thus the problem of data exchange among autonomous,possibly heterogeneous, data sources must be faced. A Key issue, in managing inter-enterprise processes and data exchange systems is mediating among the many heterogeneous information systems. Data Integration is the best technological solution to perform mediation.[53]

The main concern in data-integration problematic relies on the heterogeneity of semantic and structural surces. Structutal heterogeneity arises when the data models are conceptually different while semantic heterogeneity derives from different meaning and interpretations of data that can be the same and thus definy the obligation of a *structural reconciliation*.[54]

3.3 Technology Choices

One of the goal of this thesis is to develop a software that is accessible by web while being able to access a database whose materialization or virtualization can run under MySQL[1].
In order to achieve this we must either develop our project with a specific, already known, programming language and develop then a driver to interconnect it to MySQL or we can rely on a more efficient, already designed, connector such as JDBC driver J[2] that will take care of the more specific problematics around interconnections with databases and also supply us with new and useful utilities and methods to easen up and refine our application.
The JDBC driver is entirely programmed under Java[3] and therefore we take this opportunity to learn this object-oriented language, almost entirely new for us, as an further optional goal for us.
Learning an object-oriented language such as Java will be very important both to us and to the implementation of this software cause of its intuitivity and inclination to produce better software with given new properties of dynamic dispatch, encapsulation, subtype polymorphism, object

1 A brief description of MySQL technology will be available in Appendix B.1
2 A brief description of JDBC technology will be available in Appendix B.2
3 A brief description of Java technology will be available in Appendix B.3

inheritance and open recursion that will make our software more resilient, efficient and modular. All given characteristic that fits perfectly with the idea we have in mind.

The GUI, graphical user interface, will be also developed under Java that supply us with a toolkit such as Swing, an API for providing a graphical user interface (GUI) for Java programs.
Finally we need a server in order to run phisically our database and we will be relying on WAMPServer[4] that suits our project for "user friendliness" and for intuitiveness supplying us with an browser-based interface named phpMyAdmin that will make our tests on queries way easier and quicker while developing.

Confidend that these choices of software are all high-level and high-ended we can now focus on the design and test of our application.

4 A brief description of WAMPServer technology will be available in Appendix B.4

3.4 Decision making

"Information is not knowledge."

- Albert Einstein

Our purpose is to create a web application that, after taking data from our integrated database, anonymizes them adequately by applying the more suitable methods and techniques.
We are confident of the knowledge acquired on the state of the art of anonymization, from international scientific community and from techniques proposed and analysed by the most renown names internationally in this field of study.
For this reason and to get things into perspective, we analyse our problem with a more concrete approach in order to define some structural priorities.

- Answers disclosed by our database must give the strongest guarantees of anonymization. The reason it strictly bound to the nature of data, to the italian legislation that, as we have just seen, uphold any kind of privacy even for those who have already renounced to it by lifestyle (i.e. Vips, Movie Stars,Musicians, Politicians)

- Our web application will be accessible for definition by the web and so, knowing how much this can make it vulnerable and at the same time stealthy the identity of potential attackers, we are strongly pushed towards the choice of a differential-privacy mechanism.

- Our web application will give us a big advantage, the ability interact directly with incoming queries, in order to analyse before these questions the database so before the information, anonymized, reach the user. This is of great importance to us because it allows us to choose a differentially-private approach for our problem. Reminding that our mechanisms that enforce differential-privacy are mechanism that interact directly with the queries enforcing anonymity on the results.

- The Engineering approach to the problems moreover pushes us to think to the future and it's without a shadow of a doubt that, for the state of the art, the differential-privacy is the more concrete option. It is on this field that the majority of publications are published everyday as well as the majority of research are made in order to improve methods already know as well as to produce new ones. It's comforting knowing that since 2006 the majority of published research as differential-privacy as a keyword.

- The importance of modularity in the engineering approach to a problem, even while not considering an IT one, it's always of crucial importance. The capability, whenever we need to modify even the tiniest a project, with a big advantage for the whole project, to adapt swiftly to changes; confident of a more easier and dedicated approach and the capability to use a developed module for other purposes. Modularity has been extensively studied and implications and advantages are firmly part of the field of study of an engineer.

- The framework with whom we are going to interact, in this specific case MOMIS, can either compute a virtualization of a database, i.e. integrating only the result of a query to the eyes of an user, or compute a concrete integrated database, a new database whose properties are the products of a convolution of informations coming from different systems. This is of fundamental importance to us for our choice because it gives us another push in the direction of a modularity approach that will give our software more utility.

- Even with regards to future projects the idea of developing a module that anonymizes data, whatever the datasource is, it seems the best choice even if it will be the harder one to develop for us.

- Finally being structured clinical data not-sparse, i.e. with less probability of zeros, we can avoid to rely on advanced techniques for sampling data that would've been crucial for anonymization on a sparse dataset[37] and that would've been of a great impact on our workload.

3.5 Development Environment

The Development Environment that we chose to use for the creation of this software it's Java. In particular we are going to use Eclipse as a programming workspace.
Eclipse is a multi-language Integrated development environment comprising a base workspace and an extensible plug-in system for customizing the environment
The Database that was integrated by MOMIS will be instanced as a MySQL database, in particular we will use WAMPserver as a server. These choices have been made not only for the utility and the knowledge of these systems but also for they are widespread, free and opensource.
Finally for the interconnection between the database and our program we'll use JDBC and in particular the driver J in order to interact with the data in Java.
The testing phase will mostly be done under eclipse framework for debugging and perfectioning the software while the queering will all be done under phpMyAdmin.
For the applet testing we are going to utilize Chrome as browser as it is one of the most common and lighter browser.

3.6 Designing the project

First of all we need to focus on what is our purpose, and our goal is this:

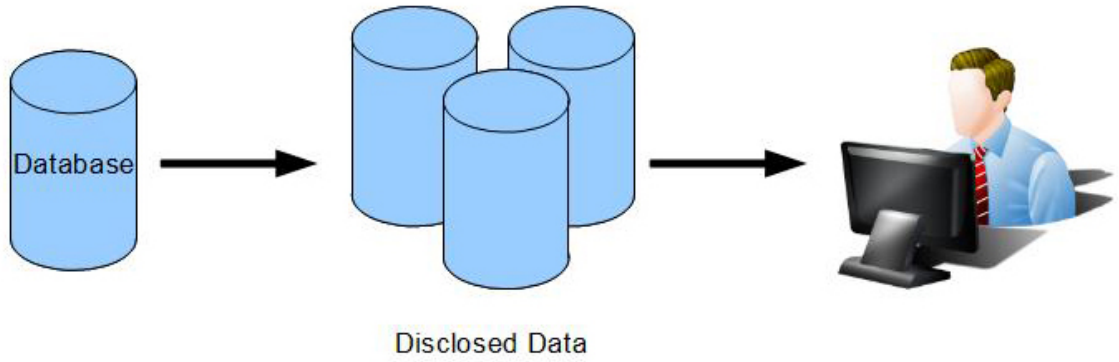


Figure 1: the User perspective.

An user wants to have access to the data we have disclosed in order to use them for his purpose. Whether it's for statistical clinical purpose such in this case or, for a more generic approach, for any kind of reasons such as economical,political,scientific or cultural.

The "clinical" part of our process of decision making will just rely on the choice of the anonymization mechanism, in order to ensure us the strongest level of privacy, while preserving a good level of accuracy in the disclosed data.

In order to do this he will be in the position of not knowing what's the tecnology under it but still being able to fulfil, with a good utility, his purpose.

Taking a look at our perspective as designers of this problem we can see that

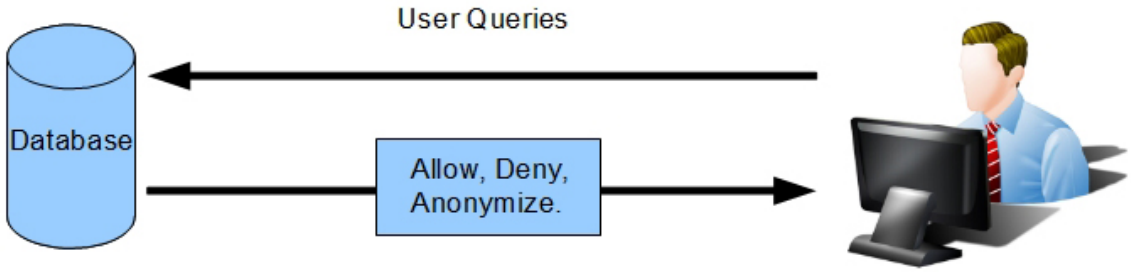


Figure 2 : the Designers perspective.

The user will make queries, or more generical questions translated into queries, that will be posed to our database, we will then be able to answer his questions by allowing, denying or anonymizing them.

As we decided while analyzing our problem in the choices chapter of this thesis, we will be using differential privacy and this scheme we just show will be a good approximation to our real project. The "Allow,Deny,Anonymize" box will be a *Black Box* for the user and for those that want to utilize it on their databases that will judge whether or not to answer/anonymize or deny the queries to the database.

Moreover as we discovered by studying differential-privacy and the Exponential and Laplace mechanisms we know that, in order to calibrate our global sensibility, or the score function and his sensibility, we will need the queries to be "processed" into our *black box* so a more accurate perspective of fig. 2 will be this new one.

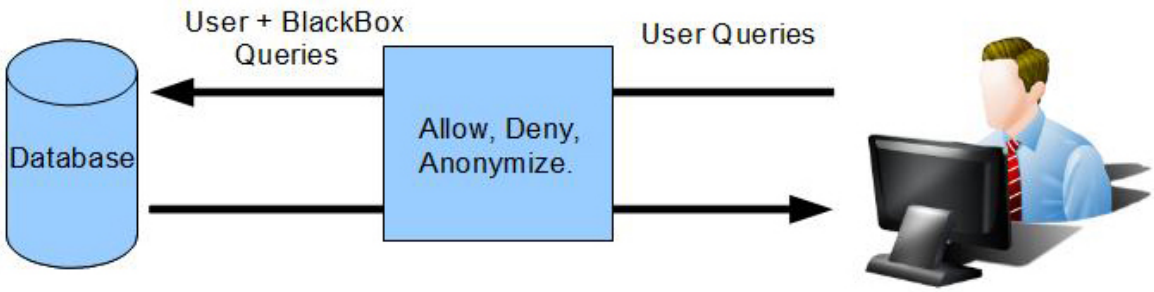


Figure 3: Defining the need of more queries.

We will need our Black Box, in order to disclose differentially-private answers, to take as input the user's queries and, in order to know whether or not to answer to them or to "apply" the correct amount of differential-privacy to them, to make new ones to the database.
So what we will get back as a result from the database will be not only the original answers to the queries but also the answers to our new synthetic queries.

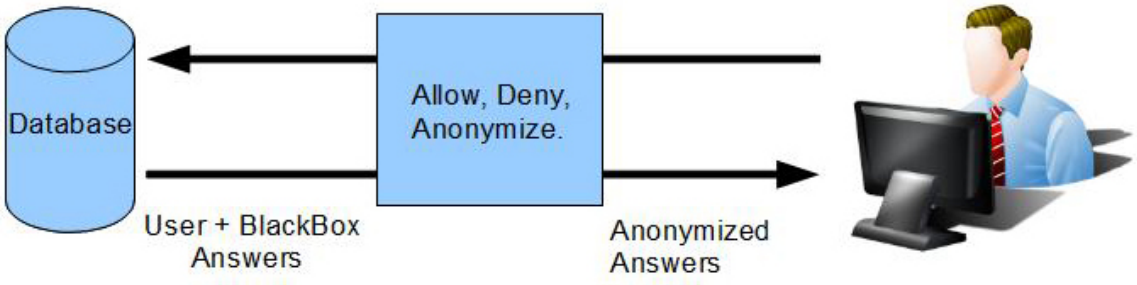


Figure 4: Answers to the queries.

The answers we will get (figure 4) from the database will be of two kind, real answers to the original queries and answers to our synthetic ones that we will need to apply the correct amount of noise on the real answers to enforce differential privacy on them in order to disclose anonymized queries to the user.

Our black box will need a way to communicate with the database's server and that will be our JDBC driver J that will ensure two things:

- That the queries will all reach the database server, that there will be no errors and if there will be, such as syntax errors on the query itself will prompt us back.
- That all the queries will be made to the server at the same time in order to achieve and higher level of resilience and performance through the use of batch queries and transactions.

For the database we are taking into consideration all the queries will be seen as one big flux of queries with no need, server side, to changes. Enforcing in this way modularity.

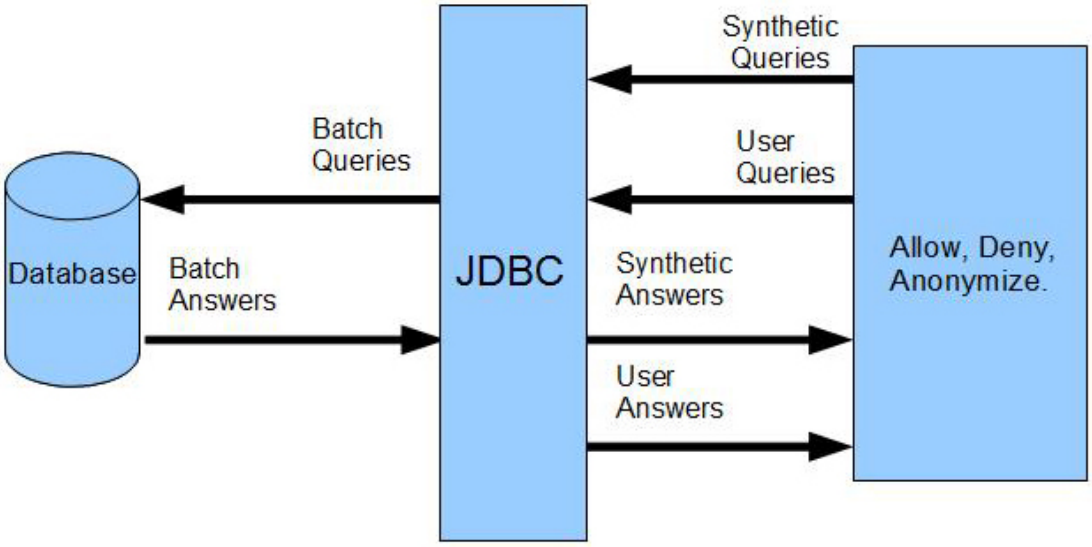


Figure 5: modelling JDBC in our project.

MOMIS on the other hand is capable of producing, with the convolution or multiple source data, a virtualization of a database or even a concrete new one. In our project it will be seen as another blackbox capable of the integration properties that we will need for the modularity of our project. Nevertheless it's of great importance that, in the decision making process, we decided which kind of mechanisms suited best an integrated-by MOMIS database, such that we were able to chose between mechanisms that enforce differential-privacy for non-sparse data.

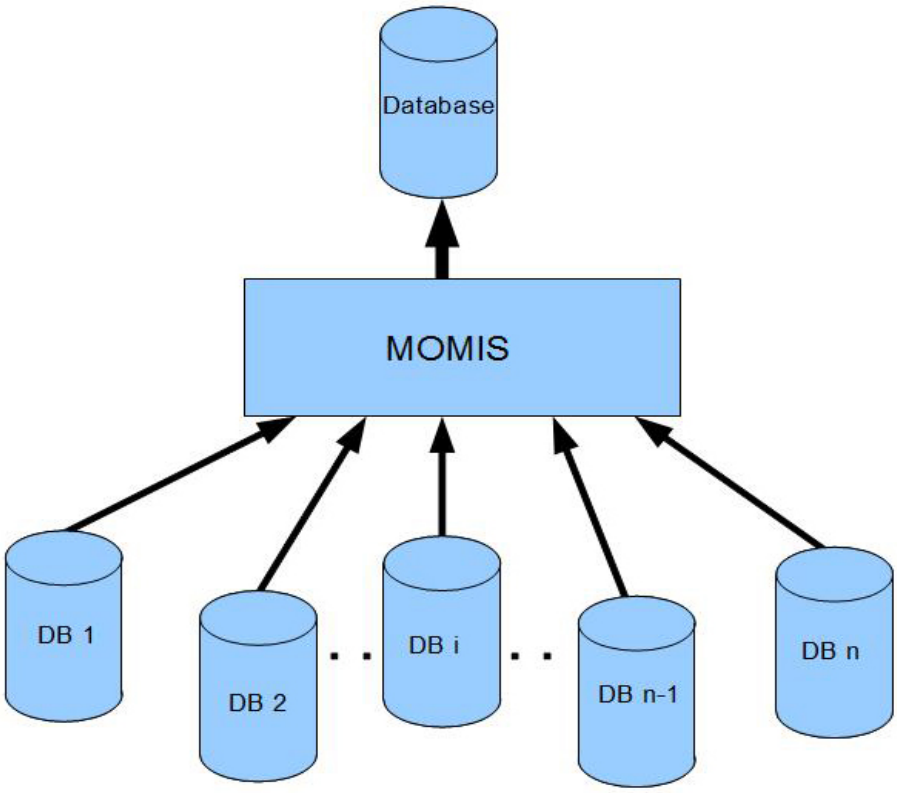


Figure 6: modelling MOMIS interaction in our project.

Finally in regards of the web-application part of our project we schematize our applet in regards of the Anon-Alyzer and the users that will use it for interrogating the FIL integrated database.

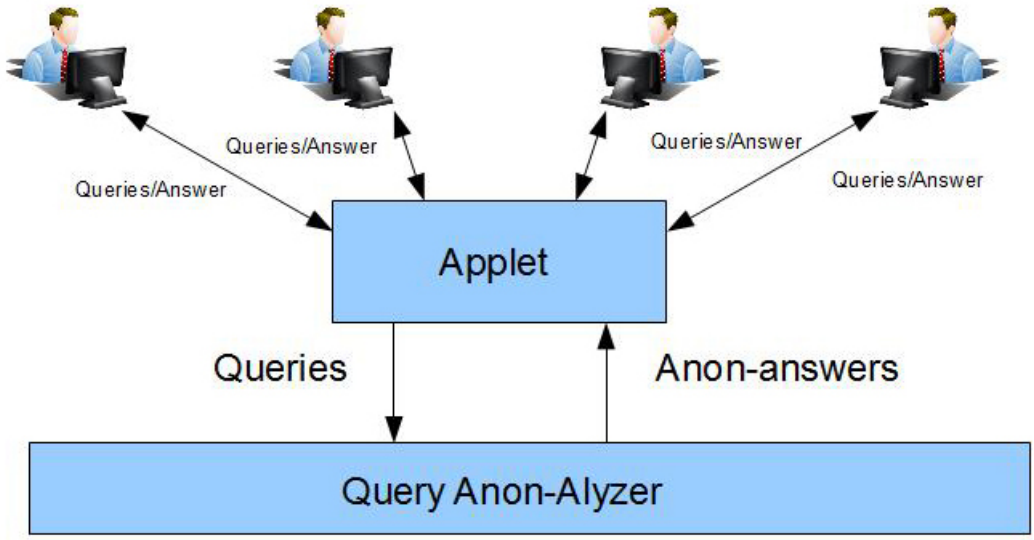


Figure 7: Applet schematization.

Confident of the decision making's choices we made and of this analysis of our problem we will now present the final schema of our Anon-Alyzer which will give us a good starting point for the design of our software.

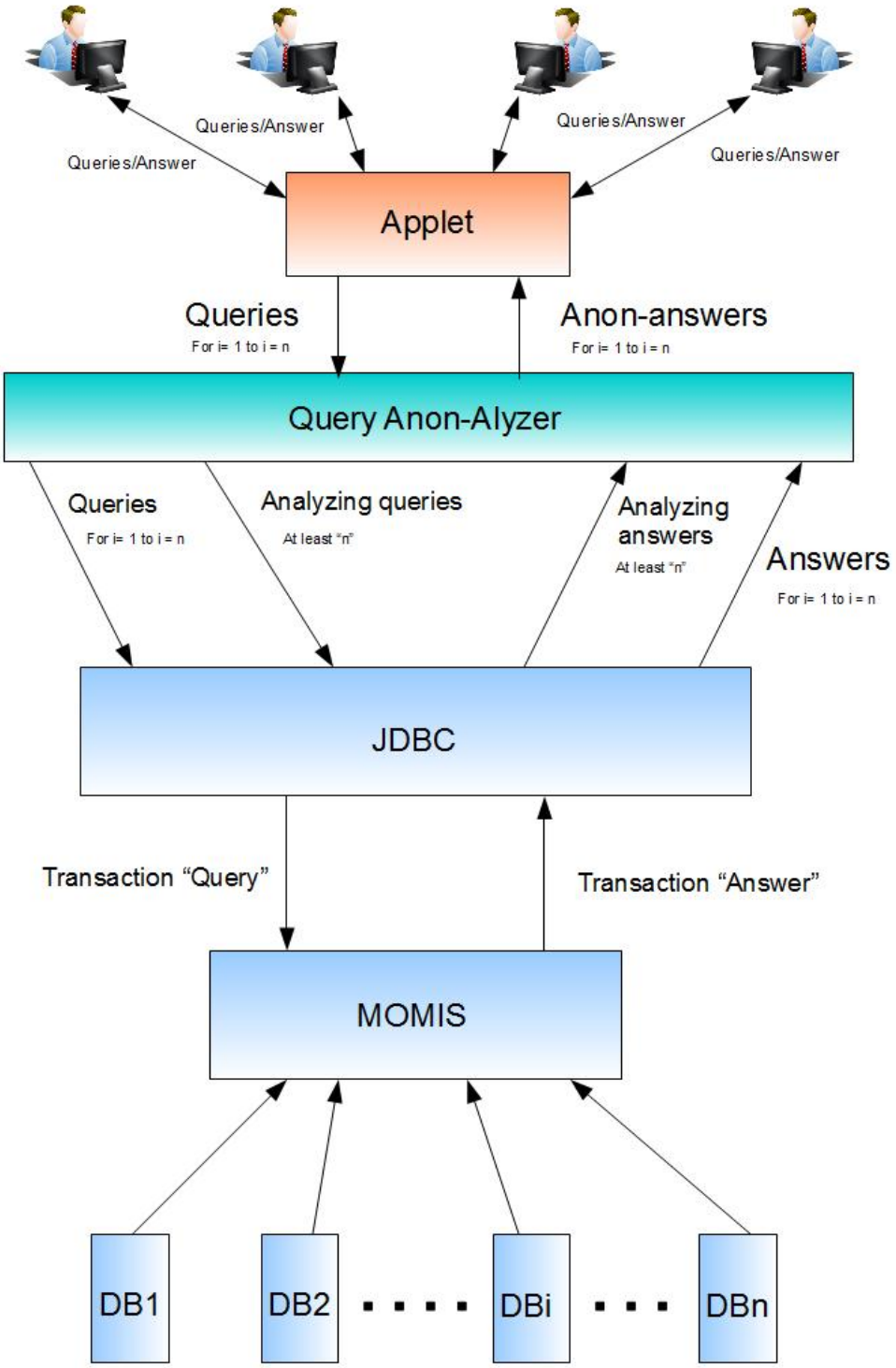


Figure 7: Schema of Anon-Alyzer interactions

Chapter 4: Coding the Anon-Alyzer.

4.1 Generic differentially-private algorithm

In this thesis we'll gonna use a generic differentially-private algorithm, described as follows

```
Algorithm  Generic differentially-private
for all incoming queries do

     Let qi be a i-th incoming query.
     Let ai be the i-th answer to query qi
     Let σi be the sensitivity of the i-th query.
     Let λ be the scale of a laplacian distribution.
     Let  ε be the parameter of the  ε-differential-privacy level
          we want to obtain
     Get qi typology.
     Chose mechanism accordingly to the typology of query.
     Chose λ accordingly to λ =σi / ε to enforce  ε-differential-
                                                  privacy
     Get  ai by queering the database
     Pass ai , λ, to the mechanism and sample a stochastic value
                                 from the parameterized distribution.
     Return ri , being ri the differentially-private answer to qi.

end for
```

4.2 Programming "Bottom Up"

As the entity of this software is still not completely discerned we decided to implement a programming approach "bottom up" starting from the most specialized classes going up to the more generic ones and to the core of our program.

It's a long-standing principle of programming style that the functional elements of a program should not be too large. If some component of a program grows beyond the stage where it's readily comprehensible, it becomes a mass of complexity which conceals errors as easily as a big city conceals fugitives. Such software will be hard to read, hard to test, and hard to debug.

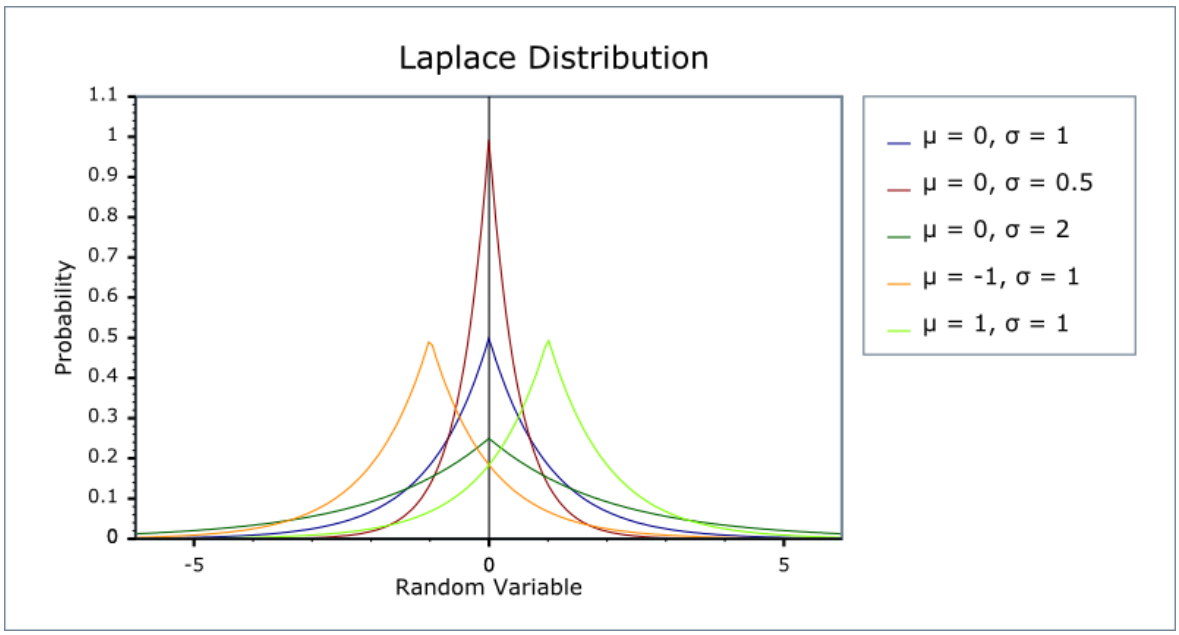
In accordance with this principle, a large program must be divided into pieces, and the larger the program, the more it must be divided.

4.3 LaplaceNoise class

Therefore it seemed logical to start from the programming and definition of the Laplace noise.

```
package AnonAlyzer;
import jsc.distributions.Laplace;
public class LaplaceNoise
{
 double mu= 0;
 double scale;
 double err;
 double mu;
 public LaplaceNoise (double mu,double scale)
      {
           this.number=number;
           this.scale = scale;
      }
 public double getNoised()
      {
           Laplace Lap = new Laplace(mu,scale);
           err=Lap.random();
           return err+number;
      }
}
```

We decided to utilize a distribution class of the famous jsc api which will , given two number, return us a probabilistic distribution. In particular a probability distribution with μ = to "mu" and σ = to "scale" we will have laplace distributions similar to :



Being *number* the number on whom we wanna enforce differential privacy, through the laplace mechanism, the method *getNoised* will return a value, double, which will be equal to *err+number.*

4.4 Laplacian Mechanism class

Our Laplacian Mechanism is a class taking in three numbers a number that will become our μ in the Laplace Noise class and NoisedUp() method, a sensibility which will refer to the sensibility of the query we are answering and a epsilon = ε that will be the parameter with whom we will define how differentially private is our answer.

```
package AnonAlyzer;
public class LaplacianMechanism {

 double number=0;
 double sensibility= 1;
 double epsilon = 0.5;
 double mu=0;
 double scale=sensibility/epsilon;
 double ndp=0;
 public LaplacianMechanism(double number, double
```

```
sensibility, double epsilon)
 {
      this.numero=numero;
      this.sensibility=sensibility;
      this.epsilon=epsilon;
 }
 public double getdiffPriv ()
 {
      this.scale=sensibility/epsilon; // sigma =
sensibility/epsilon
      LaplaceNoise Lap = new LaplaceNoise(numero,
scale); // give the values to the LaplaceNoise
      ndp = Math.round(Lap.getNoised()); // the value that
will ensure diff-priv will be the noised up closer integer.
      return ndp;
 }

}
```

In particular we need to underline the fact that our "sigma" σ, since we have proven that in order our result to be enforcing ε-differential privacy we will need our σ to be the result of the sensibility of the query divided by ε (more formally $\frac{\Delta}{\varepsilon}$)

The method getDiffPriv() will return the original answer noised up by a noise sampled from his Laplacian distribution and rounded to the nearest integer in order to validate the result.

4.5 LaplacianCount

The LaplacianCount class scope is to correctly enforce differential privacy over counting queries that revolves around many attributes, such as `GROUP BY` query, and that cannot be anonymized by just adding noise.
This new class it's our first solution attempting to implement an Exponential Mechanism in our Anon-Alyzer. The scoring function will be the initial distribution of occurrence of any value in a given attribute, with the clarification that more occurrence is better. Since we are considering for the moment attributes that are biunivocally connected to an unique identifier the maximum change in the attribute is 1.
In order to do this we will be needing a new set of object, that we will call ResMatrix that will work a buffer for us taking values of the queries and letting us work on the ResultSet values without working directly on it.

```
public class ResMatrix
{
     int i=0;
     int arrSize;
     int loc;
     String _matrix="";
     public ResLine[] reslines;
     public ResMatrix(int arrSize)
     {
          reslines = new ResLine[arrSize];
          this.arrSize=arrSize;
     }

     public void setArrEle(int loc, int count, String name)
     {
          reslines[loc] = new ResLine(count, name);
     }

     public void printElement(int loc)
```

```
	{
		System.out.println(+reslines[loc].toStr());
	}
	public String printAll()
	{
		for(i=0;i<arrSize;i++)
		{
			loc=i;
		_matrix = _matrix+ "\n" +reslines[loc].toStr();
	}
		return _matrix;
	}
}
```

With this class as a buffer we can pass the ResultSet values through it, change the occurrence of each count, apply laplacian noise to it, and render the result set, with a correct amount of sensibility, differentially private.

```
package AnonAlyzer;
import java.sql.ResultSet;
import java.sql.SQLException;
public class LaplacianCount
{
	public LaplacianCount(ResultSet res, Double sensibility,
Double epsilon)
				{
					this.res=res;
					this.sensibility=sensibility;
					this.epsilon=epsilon;
				}
		public ResMatrix getNoised()
			{
			CountRowsRS cr = new CountRowsRS(res);
			int nrows=cr.getRn();
```

```
                ResMatrix rs = new ResMatrix(nrows);
                try {
                     while (res.next())
                          {
                          count=res.getInt(1);
                          name=res.getString(2);
          LaplacianMechanism LM = new
LaplacianMechanism( count,sensibility,epsilon);
                          ndp = LM.getdiffPriv();
                          rs.setArrEle(i, ndp.intValue(), name);
                          i++;
                          }
                     }catch(SQLException ignore){}
                     return rs;
                }
     }
```

The LaplacianCount class will take as an input the ResultSet, result of the query, instantiate a ResMatrix with a correct amount of lines in order to contain all of his values, call the LaplacianMechanism with the right values on each count result inside the ResMatrix and return to the web application the correct ResMatrix filled with the values that enforce differential privacy.

4.6 Query on a Date attribute

The query on an attribute of a date type can be very tricky.
Although the date represent ideally a number, the computation of a differentially private date is strictly connected to the level of noise we wanna add to the data.
Applying a laplacian noise to the whole date can result in really inaccurate dates to the point that they are neither useful neither private.
In clinical data the time for events can be one of the most important information in order to define patterns, create models of data and even analyzing the history of a pathology, while the day can be in many cases "not that important" in the majority of cases the most important value is the year, followed by the month, in order to define connections between seasons and pathologies for instance.

In order to enforce a strictly correct anonymization we could risk in many cases to have a complete useless anonymized data. Moreover the data correctly anonymized will result to be quite the opposite a loss of privacy cause some dates, maybe in the future or in a too distant past, can be easily marked to be false by a canny analyst.
Therefore we decided to apply a laplacian noise only to the day resulting in a good compromise between privacy and utility.
Another tricky point will be the validation of a correct data, in order not to fall on an "easy" errors of months with too many days or days of the week which doesn't correlate correctly to the date.
We will call this class DateValidator:

```
public class DateValidator {
     public boolean isThisDateValid(String dateToValidate, String
dateFromat){
          if(dateToValidate == null){
               return false;
          }
          SimpleDateFormat sdf = new SimpleDateFormat(dateFromat);
          sdf.setLenient(false);
          try {
               //if not valid, it will throw ParseException
               Date date = sdf.parse(dateToValidate);
          } catch (ParseException e) {
               e.printStackTrace();
               return false;
          }
          return true;
     }
}
```

This class will return us a boolean value on the validity of the date, false will be a data which is not real and true will be a realistic data.
Then we will need a new mechanism to enforce differential privacy that will affect only the day of a date.

Thus we'll introduce the LaplacianDate class:

```
public class LaplacianDate {
     java.util.Date date = new Date();
     java.util.Date anonDate = new Date();

     public LaplacianDate (Date date, double sensibility, double
epsilon)
     {
          this.sensibility=sensibility;
          this.date=date;
          this.epsilon=epsilon;
     }

     public Date getAnonDate()
     {
          GregorianCalendar calendar = new GregorianCalendar();
          calendar.setTime(date);
          day=calendar.get(calendar.DAY_OF_MONTH);
          month=calendar.get(calendar.MONTH);
          year=calendar.get(calendar.YEAR);
          calendar.set(calendar.YEAR,year);
          calendar.set(calendar.MONTH,month);
          LaplacianMechanism ln = new
LaplacianMechanism(day,sensibility,epsilon);
          anonday=ln.getdiffPriv();
          calendar.set(calendar.DAY_OF_MONTH, anonday.intValue());
          anonDate = calendar.getTime();
          return anonDate;
     }

}
```

This class will takes as inputs a date a sensibility and an epsilon and will return us a date anonymized with the laplacian noise applied to the day.

4.7 Query on a single alphanumeric attribute

In order to enforce differential privacy on a query that will question our database to produce the most common alphanumeric value in a specific set or with specific join, we need to produce a mechanism that will enforce differential privacy but we cannot only rely on the laplacian noise. Simply adding noise to an alphanumeric attribute, that therefore has a meaning, such as the name of a sickness or the name of a location, will result in an answer without no meaning and moreover no privacy.

Therefore we should use a method, such as the LaplacianCount that generates a scoring function. In our case it's the original distribution of the attribute values such as the less present value will have less probability to be picked and the most common value will have the highest probability to be picked. As we have seen in the LaplacianCount paragraph the initial distribution of probability will be "noised up" of a Laplacian value so we'll have a "differential privacy" on the occurrence of each value. Then we'll have to provide only the first value with the correctly noised occurrence that can or cannot be the original most common one.

In order to do this, we can utilize for the most part the LaplacianCount class but we must implement inside the ResMatrix a new method to sort our objects array, we decide to use the BubbleSort algorithm studied in many courses and we feel confident that this will be a good compromise between efficiency and utility.

Before we can apply a BubbleSort method to sort our object we must define inside the object itself a method to identify which object of the same type is "bigger" then the other.

So in the ResLine class we implement a new method called compareTO():

```
public int compareTo(ResLine aResLine)
          {
               if(aResLine.getCount()<this.getCount())
               {
                    return -1;
               }
               else if (aResLine.getCount()>this.getCount())
               {
                    return 1;
               }
```

```
            else {return 0;}
        }
```

This method will return us a -1 if one object is bigger s.t. "has a bigger count value" than another. Then we can finally implement the BubbleSort algorithm in the ResMatrix class:
public void sort()

```
for(i=0;i<arrSize;i++)
     {
     for(j=arrSize-1 ; j > i; --j){
          if (reslines[j].compareTo(reslines[j-1])==-1)
               {
                    ResMatrix.swap(reslines,reslines, j, j-1);
               }
               }
     }
```

Now we just need a method to swap the two objects if one has a bigger count value than the other.

```
static void swap ( ResLine a[], ResLine b[] , int first, int
second )
    {
      int qC;
      int wC;
      String qN;
      String wN;

      qC = a[first].getCount();
      wC = a[second].getCount();
      qN = a[first].getName();
      wN = a[second].getName();

      b[first].setCount(wC);
```

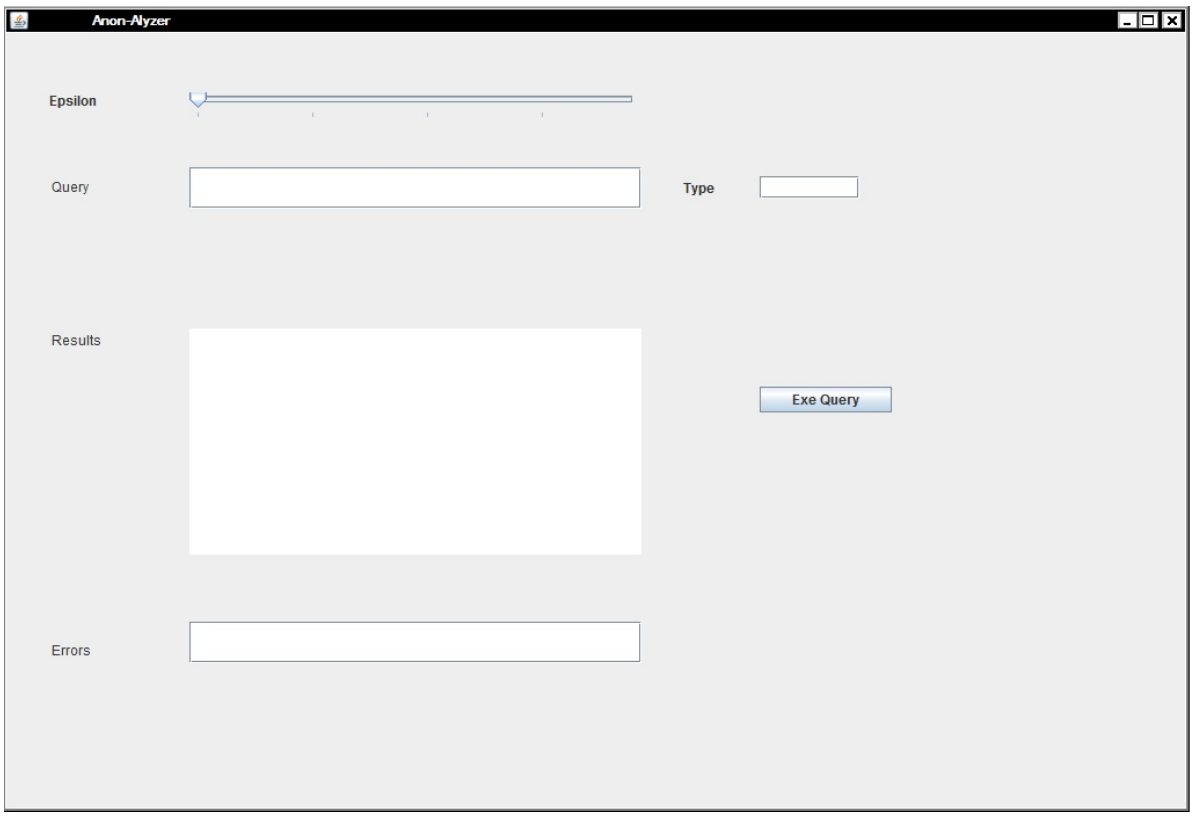
```
        b[second].setCount(qC);
        b[first].setName(wN);
        b[second].setName(qN);

    }
```

Finally we'll pass to the result text area of our application just the first value of this modified LaplacianCount in order to produce a correctly differentially private answer.
We would like to make a point here, while the result is differentially private the process in which the values of the query are analysed and modified aren't but we feel that, given the case and the prospect of a higher utility while preserving privacy this is a good estimate of the original method cause the final result is still indistinguishable from both a result with a given individual taking part in the data or without it.

4.8 Graphical "tester" Interface

The AnonAlyzer application will have a graphical user interface (GUI) made in a Java's Jframe, An extended version of java.awt.Frame that adds support for the JFC/Swing component architecture. Our main goal is to produce firstly a GUI for our tests on the database, in order to get results and to debug the software, and then to propose another GUI more user-friendly for future implementations.

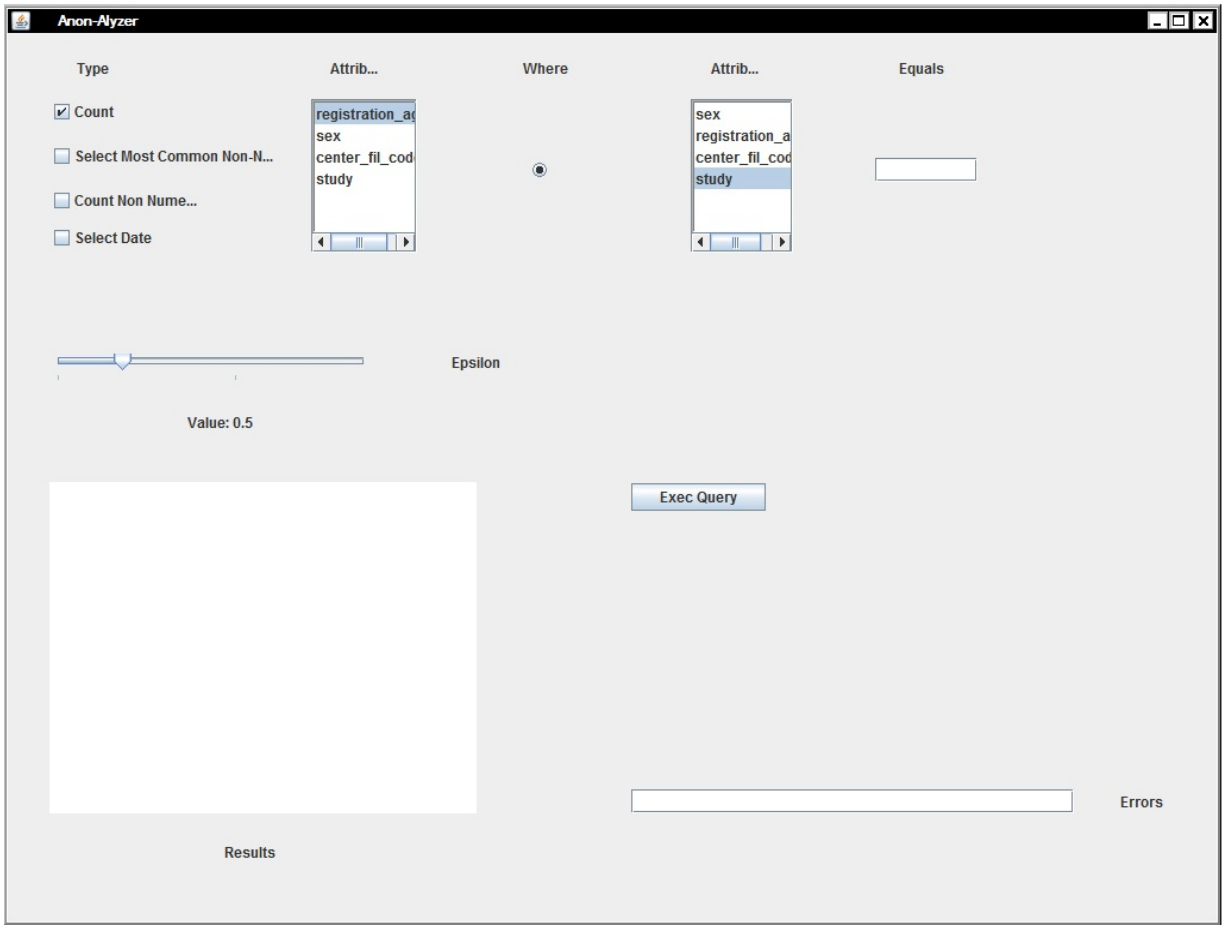
The testing GUI will have a text field, in which the user will insert a query (string), a texfield where the user will insert the type (int) of a query, a Jslider that will provide the epsilon value (between 0.5 and 2), a textfield area where the results will be parsed, a textfield for errors that may or may not occur while querying the server and finally a button to execute the query.



This is a good compromise between utility and clarity, we can now chose dynamically the epsilon value in order to test what's the better value for each situation and what will give us a good-enough ε-differential-privacy.

4.9 Graphical User Interface

For the final user of the Anon-Alyzer application we will need something more user friendly.
In particular we will need something that makes the user query the database without being forced to know the database's specific language.
Moreover the user must be limited in the choices by the number and type of attributes he can query on and on the number of "primitive-query".
To clarify our meaning of primitive-query we must say that we intend a primitive-query to be a simple query of one attribute on one table i.e. a join query between two tables will be seen by our software as two distinct query.
This is very important for us because on the number of primitive query and on the cardinality of the attribute will be crafted the global sensibility of the query and therefore passed to the mechanism the correct value to enforce differential-privacy on the results.



We have decided to put a limit of two, for our initial release of this application, to the number of primitive-query, the anon-Alyzer will parse in order to simplify our task that mainly resides on the study of the anonymization of results rather than complete anonymization of just a specific world. We will then need in our GUI two selectable lists of attributes, one for the specific type of query and another for the "where" confront parameter with a value that the user will put in a textfield. The lists will change the attributes displayed accordingly to the choices made in the checkboxes such as:

```
     list_1 = new JList(attributes);
     scrollPane.setViewportView(list_1);
list_1.setBorder(new BevelBorder(BevelBorder.LOWERED, null, null,
null, null));
     list_1.setSelectedIndices(new int[] {1});

     list_1.setSelectedIndex(1);

     list_1.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);

     list_1.setVisibleRowCount(10);

     list_1.addListSelectionListener(

     new ListSelectionListener()
          {
               public void  valueChanged(ListSelectionEvent event)
               {
               _attribute_1=(String) list_1.getSelectedValue();

               }
          }

     );
```

So that to the list will be passed everytime a string with the correct values of attributes that can be selected.

We'll have 4 checkboxes, and only one at a time can be selected, that will give us four typologies of queries that will change the parameters in the two attributes list that we can select.

We will mantain the epsilon slider,the text area for results, the text field for eventual errors and of course the button to execute our queries.

In the end the "where" button can or cannot be selected.

The combination of the checkboxes, the where-button and attribute selections will create our query in the QueryAnalyser class while the epsilon parameter will be passed directly to the Querier class.

4.10 QueryAnalyser

The QueryAnalyser is responsible of :

- Determining the correct typology of the query, by logically analyzing the choices in the user GUI, and to pass that correct value to the Querier.
- Constructing the query by putting together the various attributes and the value we must make comparisons to,in a specific way that depends on the typology of the query itself.
- Passing the constructed query to the Querier with the correct sensibility attached to it.

In order to do this we'll have to implements two methods getQuery() that will return the query constructed by the QueryAnalyser and getSensibility() that will return the correct value of sensibility given the constructed query.
The getQuery() method will look like this:

```
public String getQuery()
{
      if(type==1&&whereStatus==false)
      {
      _query="Select Count(Distinct " + attribute_1+" ) From patient";
            return _query;
      }
      if(type==1&&whereStatus==true)
      {
      _query1="Select Count(Distinct" + attribute_1 + " )  From patient Where ";
      _query2=""+ attribute_2 +"="+equalsValue+"";
      _query= _query1+_query2;
      return _query;
      }
      if(type==2&&whereStatus==false)
      {

                                    ...
                                    ...
      }

            return _query;
      }
```

parsing each logical case and producing a correct query for each of it.

While the getSensibility() method for our specific case will just return the sensibility of each attribute multiplied by the coefficient 1 if it's a primitive-query or 2 if it's a join query.

4.11 Querier

The Querier is the core of the Anon-Alyzer application. It's main objectives are:

- Taking the queries as input from the web application or the QueryAnalyser.
- Analyse the queries in order to establish which method to use and what informations must be passed to those classes or , in the case we are utilizing the user GUI, get from the QueryAnalyser the correct values of sensibility, epsilon and the constructed query.
- Query the database and retrieve the real answer to it.
- Correctly anonymize it and return it to the applet.

Nevertheless the Querier is responsible for the interconnection between the Anon-Alyzer and the DB server and, in order to do that, will utilize the JDBC connector J.

```
public class Querier {
	String _driverName = "com.mysql.jdbc.Driver";
	String _url = "url or  path for the server";
	String _user = "username";
	String _pwd = "password";
	String _query;
	String errmsg = "no errors";
	String _matrix;
	int type =1;
	double epsilon = 0.1;
	double sensibility= 1;

	public Querier(String query,int type,double epsilon)
	{
		this.epsilon=epsilon;
		_query=query;
		this.type=type;
		this.sensibility=sensibility;
	}
```

The querier will get from the QueryAnalyser the sensibility, the query and the type while from the GUI will be given the epsilon chosen by the user.
The querier will utilize static parameter such as *_user* , that will contain the name of the username of the database user, *_password* as the password to access the database, *_url* as the url of the location of it, in order to gain an authenticated access to the database.

```
String getAnswer()
{
     Connection conn = null;
     Statement state = null;
     ResultSet res = null;
     int contatore = 0;
     double ndp=0;
     switch(type){
case 1 :{  // case 1 single query on single attribute numeric
     try
     {
          Class.forName(_driverName);
     conn= DriverManager.getConnection(_url,_user,_pwd);
     state= conn.createStatement();
     {
          res = state.executeQuery(_query);
          while (res.next())
          {
          contatore = res.getInt(1);
          LaplacianMechanism LM = new
LaplacianMechanism( contatore,sensibility,epsilon);
          ndp = LM.getdiffPriv();
          }
     }
     catch(Exception error)
     {
     errmsg=error.getMessage();
```

```
          }
          finally
               {
               If(conn!=null) try {conn.close();}
catch(SQLException ignore){}
               if(state!=null) try {state.close();}
catch(SQLException ignore){}
               }
               return "The answer is: " +ndp;
               }  // end case 1 works!
```

The First typology of query will be a single counting query that should return the most common numeric value inside a set.
In order to do this the Querier will firstly create a connection and a statement then execute the query finally making the resultset advance in order to get the first result that will be in our case the answer itself.
Taken that answer it will apply the Laplacian noise to it and return to the GUI the correctly anonymized value.
Finally it will close both the connection and the statement returning eventual error in the querying process or in the connection process to the GUI error textfield.

```
case 2 :{  // case 2
try
     {
          Class.forName(_driverName);
          conn= DriverManager.getConnection(_url,_user,_pwd);
          state= conn.createStatement();
          res = state.executeQuery(_query);
          LaplacianCount lc = new
LaplacianCount(res,sensibility,epsilon);
          ResMatrix rm = lc.getNoised();
          _matrix=rm.printAll();
```

```
     }
     catch(Exception error)
          {
          errmsg=error.getMessage();
          }
          finally
          {
          if(conn!=null) try {conn.close();} catch(SQLException
ignore){}
                                                  if(state!=null)
try {state.close();} catch(SQLException ignore){}
          }
          return "the answer is: \n" +_matrix; // works!
     }  // end case 2
```

The second case will be a generic counting query on non-numeric attributes, that will imply that we will use our method LaplacianCount in order to enforce differential privacy on more results.
We will once again open a connection and a statement, query the server and passing all the information in the resultset to our ResMatrix class built for the purpose of storing information in order to work in the results.
We will then apply LaplacianNoise to each value and insert the noised value inside the ResMatrix and return the final corrected object to the result area.

```
case 3 :{
     try
     {
     Class.forName(_driverName);
     conn= DriverManager.getConnection(_url,_user,_pwd);
     state= conn.createStatement();
     res = state.executeQuery(_query);
     while (res.next())
          {
          date = res.getDate(2);
          DateValidator dv = new DateValidator();
          do// apply a laplacian mechanism to the date
          {
          LaplacianDate LD = new
LaplacianDate( date,sensibility,epsilon);
               anonDate =LD.getAnonDate();
               reportDate = df.format(anonDate);
          }
while (dv.isThisDateValid(reportDate, _dateFormat)==false);   //
verify the date is a valid date
```

The third case will be a query on a date attribute, we will open a connection and a statement pass the results of the query to the resultset and then while there is results in the resultset apply the LaplacianDate method to each value.
In particular the noised date will still be a correct date cause of the use of the DateValidator method that will return us if the date is valid or not and until it's valid we will generate a new one.

4.12 Queries example

We now present you with some queries on whom our Anon-Alyzer correctly enforce ε-differential-privacy, these are just some typologies that can be replicated with different sets, tables and even with different databases, providing a quick correction on the url and username/password static attributes in the Querier.
In order to present you both the real answer and the noised up ones while preserving the anonymity of individual inside the FIL database we'll have to utilize a different database, with similar attributes and values, filled by us with random and not-real values.

We'll use, to ease up our explanation, a single tableset called *filtest* of this type:

id	*name*	*surname*	*pathology*	*dob*	*city*
...	...	...	...	...	...

Id: an int, primary key,auto increment not null attribute
Name and Surname: the name and surname of a fictional individual ,both name and surname will be randomly chosen by a list of names.
Pathology: a pathology chosen from a list of generic pathologies.
Dob: the date of birth of the patient
City: the city where the patient lives (chosen randomly from a list of county seat*)*

A random sample from our database of ten thousand tuples will be:

id	*name*	*surname*	*pathology*	*dob*	*city*
23	Domenico	Gallo	Respiratory	18/01/53	Ancona
24	Walter	Bianchi	Neoplastic	19/03/41	Roma
25	Stefano	Rizzo	Infective	23/10/22	Bologna
26	Sergio	Ferrari	Respiratory	06/03/02	Genova
27	Maria	Catellani	Traumatic	28/10/80	Napoli

Now with this given approximation of our true database we can feel free to perform our tests while preserving the anonymity of the true individuals inside the FIL database.

- First example query:

Let's consider the first typology of query we have already coded a single request on a single attribute numeric: a simple counting query

```
Select Count(Distinct id)  From filtest Where pathology =
"Cardiovascular"
```

The true answer to this query in our database would've been: 1883

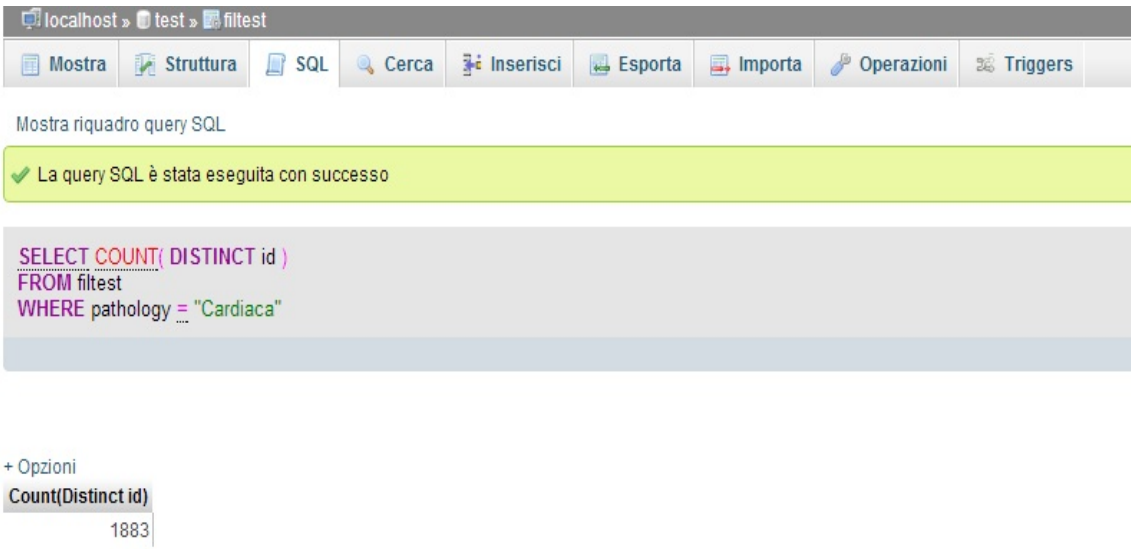


Figure 1: phpMyAdmin query on filtest server

The answer is 1883 total entries of Cardiovascular pathologies, let's see how our Anon-Alyzer will process this query.

We insert the query in the textfield, chose an amount of epsilon to enforce a generic ln3- differential privacy. The Anon-Alyzer will then receive the correct amount of epsilon, the query, will determine that since our typology of query the sensibility will be one and will utilize the correct method of LaplacianNoise to enforce differential privacy on the results, that will be:

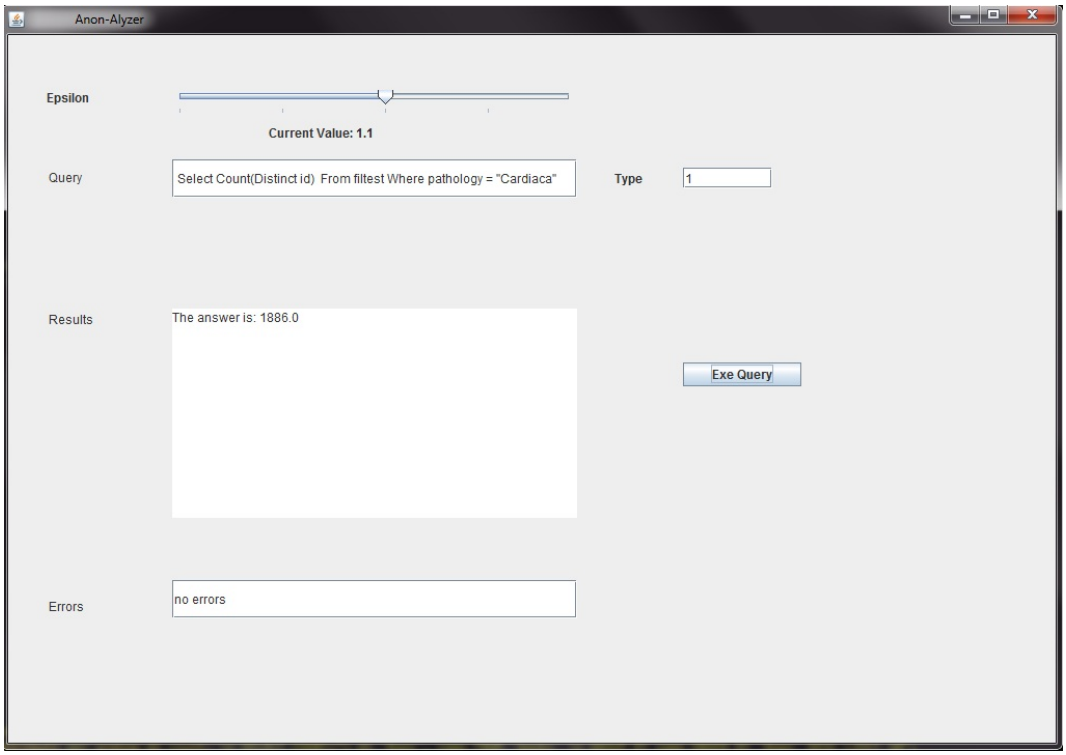


Figure 2: Query answer in Anon-Alyzer GUI

For our Anon-Alyzer a 1886 if we repeat this query, since it's a stochastic algorithm it will produce different values such as 1889,1880,1798,1901 and so on. This is one of the example more "easy" for our Anon-Alyzer since counting queries are the easiest to code into differential-privacy.

- Second example query:

Four our second example query we will take into consideration something a little bit complicated, a counting query that will return us the value counted for each pathology.

```
SELECT COUNT(*) AS `Rows`, pathology FROM `filtest` GROUP BY
pathology ORDER BY `Rows` DESC LIMIT 10
```

Let's see how it's true answer will be by queering directly the database:

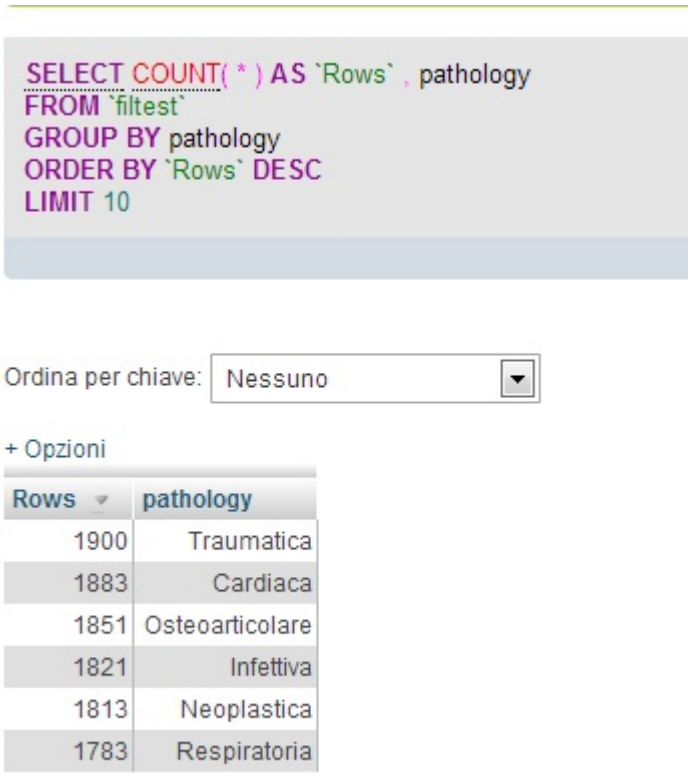


Figure 3: phpMyAdmin query on multiple counting rows

Count	*Pathology*
1900	Traumatic
1883	Cardiovascular
1851	Osteorarticular
1821	Infective
1813	Neoplastic
1783	Respiratory

Given this true answer let's see if our second methodology of laplacian mechanism will enforce differential privacy. As we can see there are even occurrence that maintained the same values as before the anonymization this isn't to be seen as a fault but rather a success cause enforcing correctly differential privacy means to enforce it on every value even the original one.

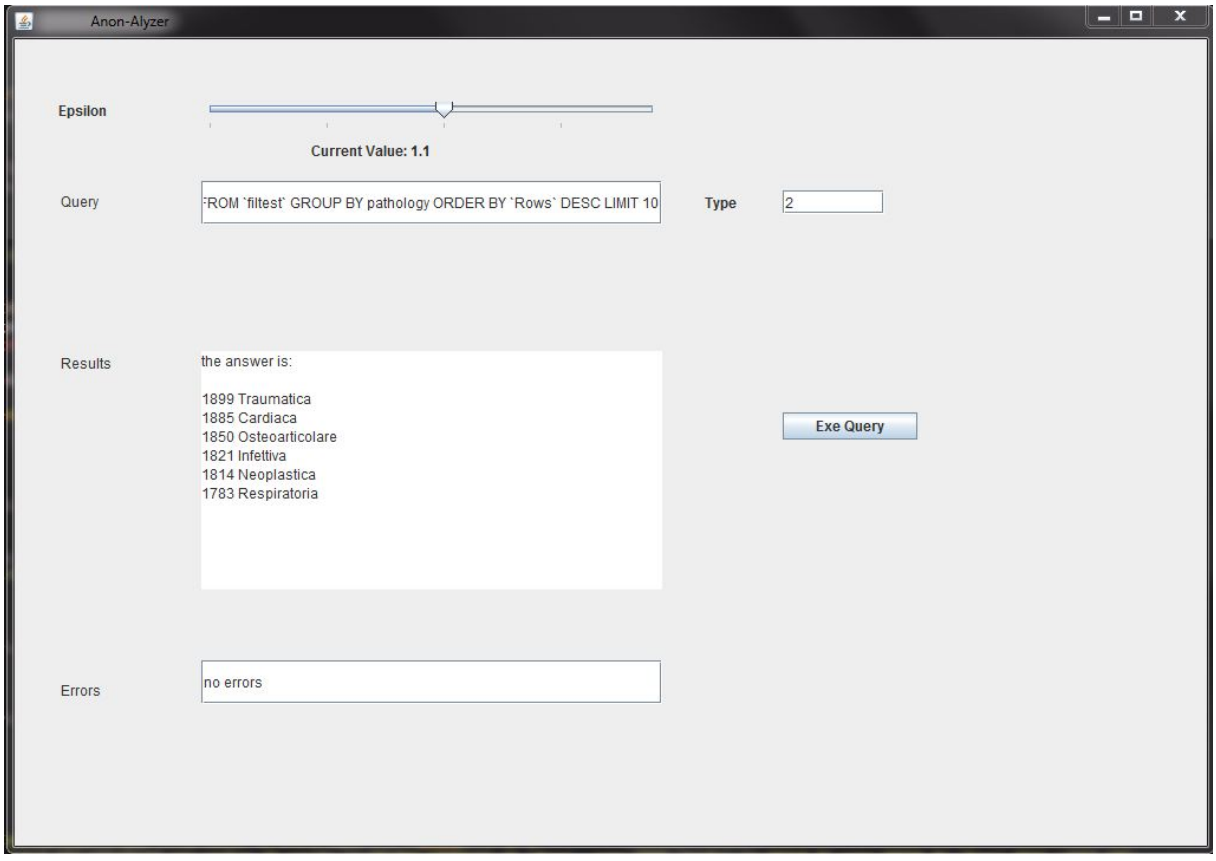


Figure 4: query on multiple counting rows on Anon-Alyzer

Count	*Pathology*
1899	Traumatic
1885	Cardiovascular
1850	Osteorarticular
1821	Infective
1814	Neoplastic
1783	Respiratory

In this example too the results are correctly randomized, subsequent attempt to query the database again with the same question will produce different responses every time

Count	*Pathology*
1910	Traumatic
1873	Cardiovascular
1821	Osteorarticular
1821	Infective
1812	Neoplastic
1785	Respiratory

Count	*Pathology*
1900	Traumatic
1883	Cardiovascular
1851	Osteorarticular
1822	Infective
1813	Neoplastic
1784	Respiratory

Count	*Pathology*
1890	Traumatic
1900	Cardiovascular
1853	Osteorarticular
1821	Infective
1823	Neoplastic
1782	Respiratory

- Third example query:

Let's try it on a query that shall return a date value, let's try with the first date inserted in the database for example.

```
SELECT dob FROM  `filtest` ORDER BY  `filtest`.`id` ASC  LIMIT 1
```

Again the true answer will be presented first in the phpMyAdmin panel:

Figure 5: phpMyAdmin panel on single date query

The result will be a date representing 21th October 1977
Let's see how the Anon-Alyzer behave.

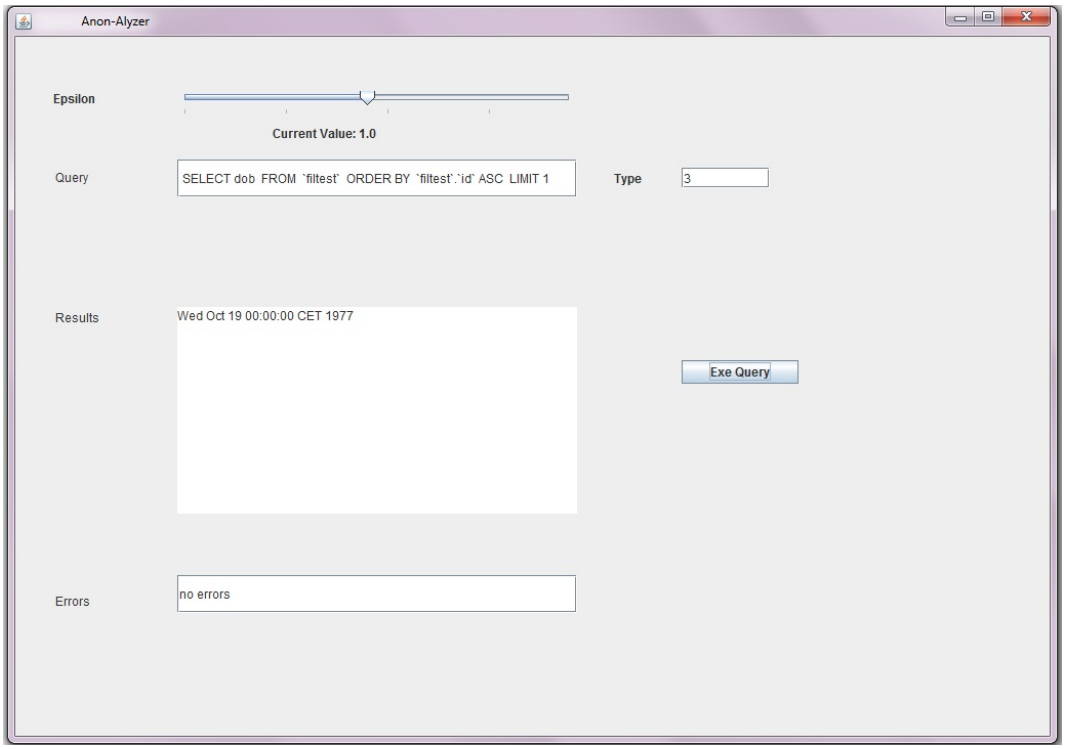


Figure 6: anon-Alyzer on a date query.

The answer from Anon-Alyzer is Wednesday 19 October 1977, subsequent queries will produce new dates, restricted to changing just the day, while being valid date due to the DateValidator.

Fourth example query:

Finally let's see how the Anon-Alyzer behave with a query on the most common alphanumeric attribute, we can't now utilize laplacian noise directly on the value but we can enforce differential privacy through our exponential mechanism described before.
The query will be just the most common pathology in the dataset:

```
SELECT COUNT(*) AS `Rows`, pathology FROM `filtest` GROUP BY
pathology ORDER BY `Rows` DESC LIMIT 1
```

```
SELECT COUNT( * ) AS `Rows` , pathology
FROM `filtest`
GROUP BY pathology
ORDER BY `Rows` DESC
LIMIT 1
```

+ Opzioni

Rows	pathology
1900	Traumatica

Figure 7: phpMyAdmin query on single alphanumeric attribute "most common"

Let's see if Anon-Alyzer enforce differential privacy even on this kinda queries:

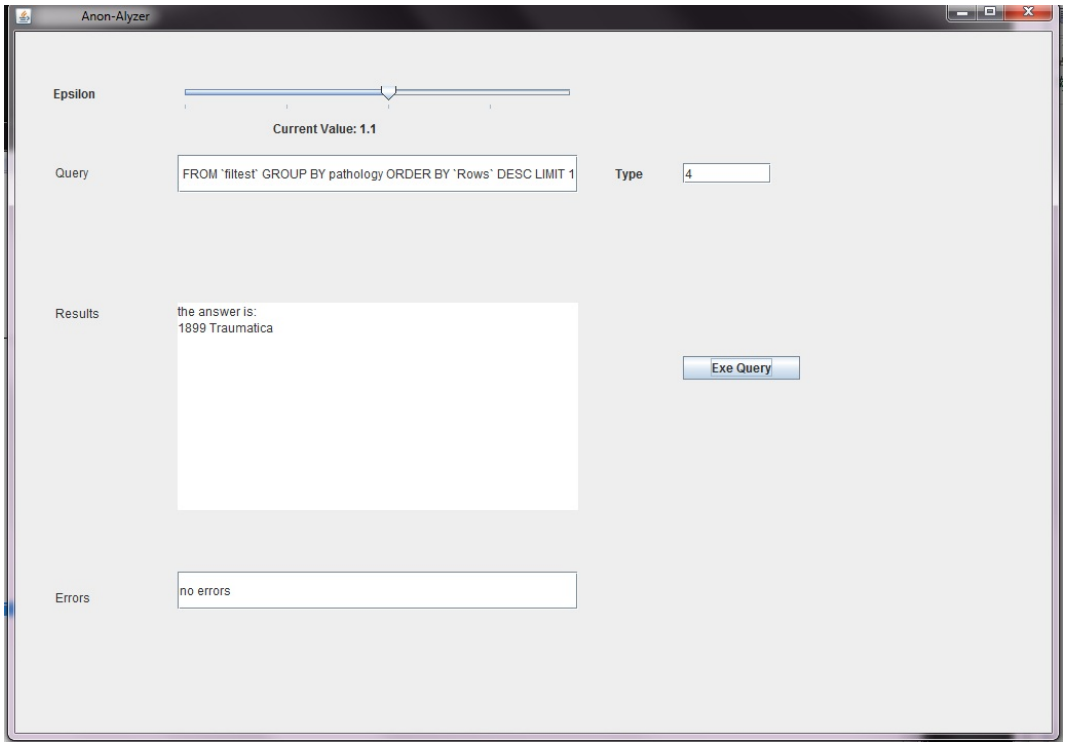


Figure 8: Anon-Alyzer response on a single attribute alphanumeric query

As we can see the result is still Traumatic but with a different value of 1899 occurrences, subsequent attempt on the same query will produce different values even with different pathologies such as the other listed.

We have now demonstrated so far that our Anon-Alyzer can correctly enforce differential privacy on 4 kinds of queries, the most common.

5 Analysis of Results

5.1 Foreword

Although the analysis of the results isn't in the goals of our thesis we feel that a brief analysis will both benefit our project and our research and will help us elaborate more concrete conclusions on our Anon-Alyzer.
We will be analysing a simple counting query, on a single numeric value, something like the query:

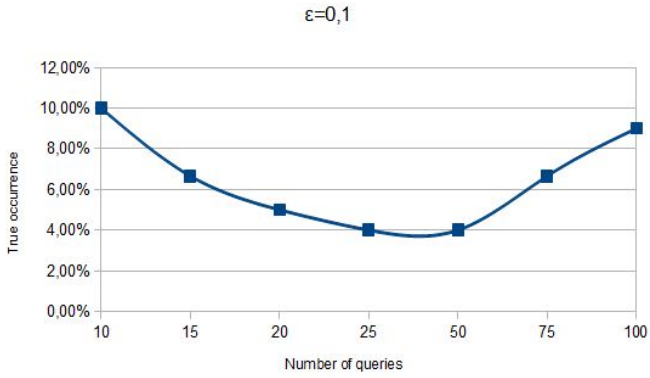
```
Select Count(Distinct id)  From filtest Where pathology =
                                          "Cardiovascular"
```

we have already taken into consideration in the query example paragraph.
In order to produce a correct and meaningful analysis we need to take into consideration if and how the variation of our ε will influence the accuracy of our results. We decide to elaborate results on three different values of ε: 0.1 0.5 and 1.1 ~ln(3), the choice of this last value is determined by the analysis of errors on ln(3) differentially private algorithms made by Cynthia Dwork that we have already talked about in the 2.9 paragraph.
We also know that the accuracy of a laplacian mechanism is strictly connected to the occurrence of the specific value so we will be analyzing attributes in the three *buckets* of 10, 100 and 1000 occurrences.
Finally, since the occurrence of the real value isn't enough to define how accurate our mechanism is, we need to define a new parameter we will be calling *precision* defined as the true value,perturbed value ratio that will help us analysing how close our perturbed value is to the original one.

5.2 Analysis

We will start by analysing a query that will return us a value of the order of tens.

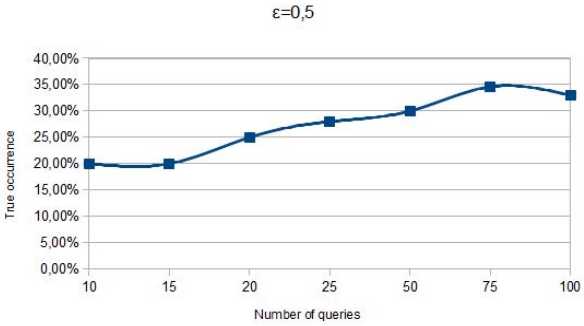
Let's start, in particular, with the analysis of the true occurrence of the true answer inside the values of laplacian perturbed ones.

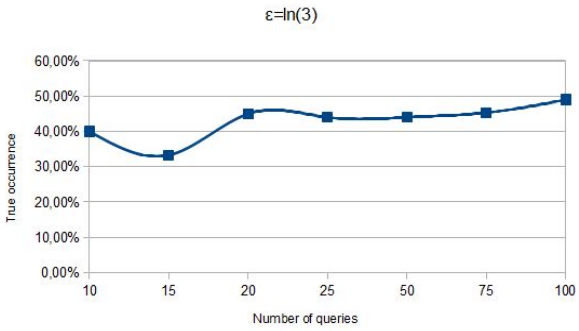


With an ε of 0.1 and considering a query that return us a value of the order of tens, the occurrence of the real value inside the perturbed bucket fluctuate betwen 10% and 4% stabilizing at 8% for higher repetitions of the query.

This can be seen as a really bad result but we must point out that it was intended, since the first goal of our software is to produce an anonymized response and with an ε so low, while the privacy is very high, the variance grow directly with it.

Let's consider now the same query and the same answer but with ε=0.5 and ε=ln(3):

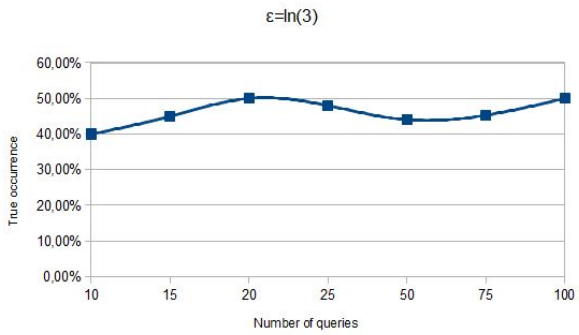


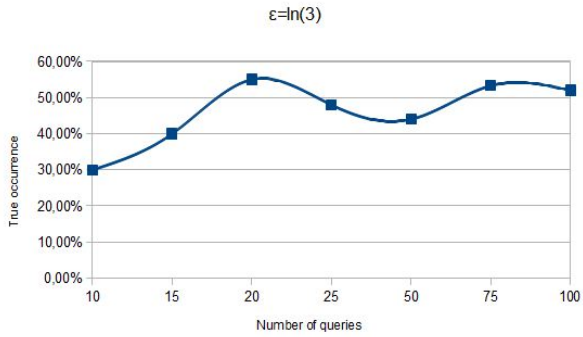


We can see how, with growing ε the occurrence of the real answer will grow as well. In particular in ε=0,5 will stabilize around 30% and around 49% for ε=ln(3).

Even if this is just a software for theoretical implementation of the Laplacian Mechanism we alreay feel that the results are really good and promising and we understand better why the differential privacy has been taken into so much consideration in the last years.

For attributes values of the order of hundreds and thousands we empirically produce similar graphs:

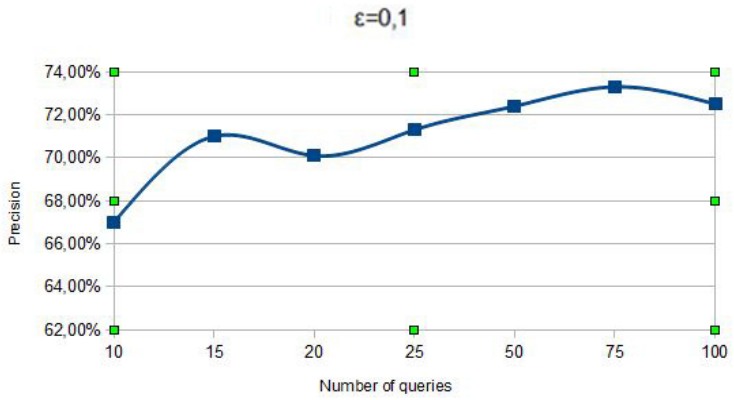




The occurrence of the real answer will be respectively around 50% and 52%.
This is a result that could've been predicted cause the occurrence of the true value is not dependent on the result's value but rather just on the ε value.

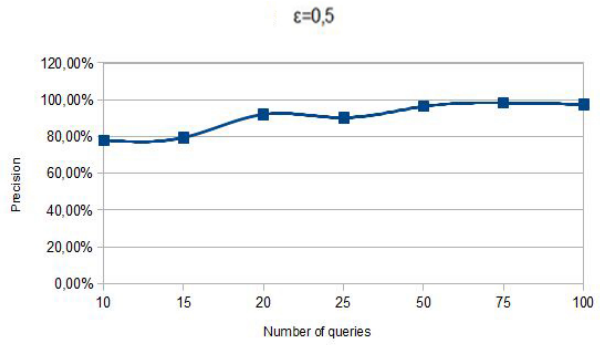
5.3 Another critical point of view

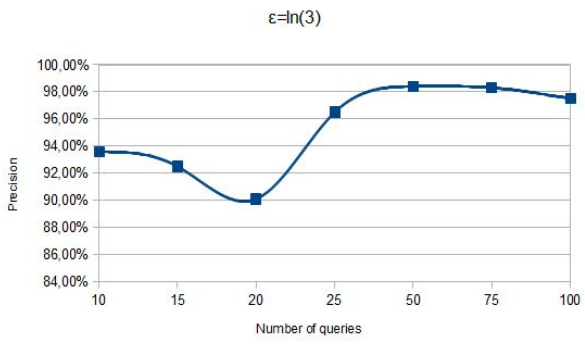
We feel although that we must add some more criticism to our analysis.
While the occurrence of the true value fluctuates from 49% to 52% for ln(3)-differentially private mechanism for attributes values respectively of the order of tens, hundreds and thousands, how much accurate are the other results? We need to verify our precision for all of the ε we have already considered.
For an attribute value of the order of tens we empirically produce these results:



These results are really comforting, giving with the maximum variance and the minimum attribute value a high utility already of the order of ~70%.

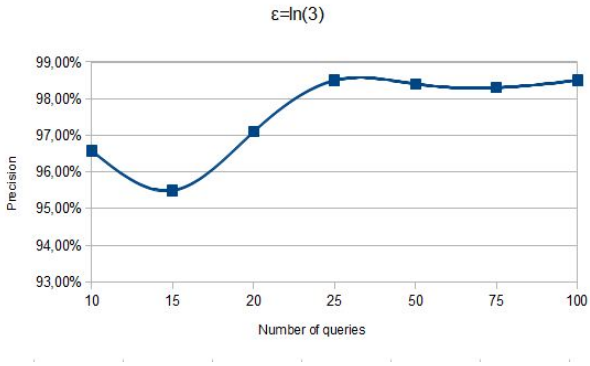
Let's see for ε=0,5 and ε=ln(3):

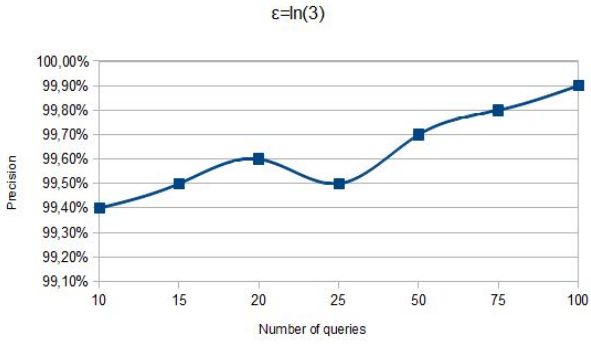




Needless to say the results are great, while losing very few privacy our utility will grow to ~80% and around ~90-96% while enforcing a ln(3)-differential privacy which is considered a standard to measurements.

Finally let's see how our software behave with attributes of the order of hundreds and thousands for just ε=ln(3):





As we can see the results are encouraging, while for the order of hundreds our Anon-Alyzer has an precision around ~97-98% for the order of thousands the Anon-Alyzer reaches a precision almost total with a precision that ranges from 99.4% to 99.9%

While these is just a brief analysis of one of the methods of our software and while these are just empirical results on really low numbers of iterations we feel that given the time and the effort put on the development of it we have prove to have achieved a really high utility.

6 Summary and Conclusions

We've presented you the Anon-Alyzer a query anonymizer that utilize the differential-privacy studied so far to enforce anonymity on data disclosed.
While it's just an application made mostly to apply the state of the art on anonymization to a concrete scenario it has provided us with really encouraging results.
Not only we have demonstrated that it's possible to utilize the Laplacian Mechanism and Exponential Mechanism with an object-oriented language on a concrete scenario but also that, by enforcing differential privacy, the results of counting queries preserves many of their utility while still being indistinguishable from the results of a similar scenario whether or not the individual is in the dataset.

The future of application such as Anon-Alyzer relies on new ways to analyse and parse queries without being able to control them, as we did, from a web-application, by developing code and algorithms able to understand what are the attributes inside a query, once identified to analyse on the database the cardinality of those attributes and return a sensibility in a more dynamic way. Moreover the future lies in the study of Exponential Mechanism, and in particular scoring functions. By doing so we'll have the capacity to refine the utility of the results, while still enforcing the same amount of differential-privacy, and improve the efficiency of our mechanisms.

Finally a different approach can be, for clinical databases, the release, in a differentially private way, of an entirely new synthetical dataset originated form the real database. Techniques such as this one already exists such as K-core clustering [57] and a researched as a brand new way of disclosing dataset in a differentially private way.

7 Appendix

A: proof of Theorem "ε-indistinguishable."

Theorem **:** *For an arbitrary adversary A, let ft(x) : $D^n \rightarrow R^d$ be its query function as parameterized by a transcript t. If = maxt S(ft)/λ , the mechanism above is ε-indistinguishable.*

Proof. Using the law of conditional probability, and writing ti for the indices of t,

$$\frac{\Pr[\mathsf{San}_f(\mathbf{x}) = t]}{\Pr[\mathsf{San}_f(\mathbf{x}') = t]} = \prod_i \frac{\Pr[\mathsf{San}_f(\mathbf{x})_i = t_i | t_1, \ldots, t_{i-1}]}{\Pr[\mathsf{San}_f(\mathbf{x}')_i = t_i | t_1, \ldots, t_{i-1}]}$$

For each term in the product, fixing the first i − 1 coordinates of t fixes the values of ft(x)i and ft(x0)i. As such, the conditional distributions are simple laplacians, and we can bound each term and their product as

$$\begin{aligned}\prod_i \frac{\Pr[\mathsf{San}_f(\mathbf{x})_i = t_i | t_1, \ldots, t_{i-1}]}{\Pr[\mathsf{San}_f(\mathbf{x}')_i = t_i | t_1, \ldots, t_{i-1}]} &\leq \prod_i \exp(|f_t(\mathbf{x})_i - f_t(\mathbf{x}')_i|/\lambda) \\ &= \exp(\|f_t(\mathbf{x}) - f_t(\mathbf{x}')\|_1/\lambda)\end{aligned}$$

We complete the proof using the bound S(ft) ≤ λε for all t. □

B: Software utilized

B.1 Java

Java is a general-purpose, concurrent, class-based, object-oriented computer programming language that is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that code that runs on one platform does not need to be recompiled to run on another. Java applications are typically compiled to bytecode (class file) that can run on any Java virtual machine (JVM) regardless of computer architecture. Java is, as of 2012, one of the most popular programming languages in use, particularly for client-server web applications, with a reported 10 million users[38][39]. Java was originally developed byJames Gosling at Sun Microsystems (which has since merged into Oracle Corporation) and released in 1995[40] as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++, but it has fewer low-level facilities than either of them.

The original and reference implementation Java compilers, virtual machines, and class libraries were developed by Sun from 1991 and first released in 1995. As of May 2007, in compliance with the specifications of the Java Community Process, Sun relicensed most of its Java technologies under the GNU General Public License. Others have also developed alternative implementations of these Sun technologies, such as the GNU Compiler for Java and GNU Classpath.

One characteristic of Java is portability, which means that computer programs written in the Java language must run similarly on any hardware/operating-system platform. This is achieved by compiling the Java language code to an intermediate representation called Java bytecode, instead of directly to platform-specific machine code. Java bytecode instructions are analogous to machine code, but they are intended to be interpreted by a virtual machine (VM) written specifically for the host hardware. End-users commonly use a Java Runtime Environment (JRE) installed on their own machine for standalone Java applications, or in a Web browser for Java applets.

Standardized libraries provide a generic way to access host-specific features such as graphics, threading, and networking.A major benefit of using bytecode is porting. However, the overhead of interpretation means that interpreted programs almost always run more slowly than programs compiled to native executables would. Just-in-Time (JIT) compilers were introduced from an early

stage that compile bytecodes to machine code during runtime.In particular we are going to use Java SE 7.

Reasons why we are chosing this language are to be found in it's widespread usage in the IT community, in the "almost" native interaction with MySQL and in the nature of this language that make it architecture-neutral. dynamic and portable.

B.2 MySQL

MySQL , is (as of 2008) the world's most widely used[41][42] open source relational database management system (RDBMS)[43] that runs as a server providing multi-user access to a number of databases. It is named after co-founder Michael Widenius' daughter, My. TheSQL phrase stands for Structured Query Language.[44]

The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation.

MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack (and other'AMP' stacks). LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python." Free-software-open source projects that require a full-featured database management system often use MySQL.

For commercial use, several paid editions are available, and offer additional functionality. Applications which use MySQL databases include: TYPO3, Joomla, WordPress, phpBB,MyBB, Drupal and other software. MySQL is also used in many high-profile, large-scale websites, including Wikipedia,[45] Google[46] (though not for searches),Facebook,[47] Twitter, Flickr, and YouTube.

MySQL is a relational database management system (RDBMS), and ships with no GUI tools to administer MySQL databases or manage data contained within the databases. Users may use the included command line tools, or use MySQL "front-ends", desktop software and web applications that create and manage MySQL databases, build database structures, back up data, inspect status, and work with data records. The official set of MySQL front-end tools, MySQL Workbench is actively developed by Oracle, and is freely available for use.[48]

We chose this as for Java for the sinergy within the two, for it's the most widespread RDBMS and because the MOMIS virtualization and synthetization works both with it.

B.3 WAMPserver[31]

As for the server part, in order to perform instructions on the DB and to test/develop our software we are relying on WAMPserver which is easy-to-use and freeware.

WAMPs are packages of independently created programs installed on computers that use a Microsoft Windows operating system. WAMP (computing) is an acronym formed from the initials of the operating system Microsoft Windows and the principal components of the package: Apache, MySQL and one of PHP, Perl or Python. Apache is a web server. MySQL is an open-source database. PHP, Perl and Python are scripting languages that can manipulate information held in a database and generate web pages dynamically each time content is requested by a browser. Other programs may also be included in a package, such as phpMyAdmin which provides a graphical user interface for the MySQL database manager.

B.4 JDBC

A JDBC driver is a software component enabling a Java application to interact with a database.[49] JDBC drivers are analogous to ODBC drivers, ADO.NET data providers, and OLE DB providers.

To connect with individual databases, JDBC (the Java Database Connectivity API) requires drivers for each database. The JDBC driver gives out the connection to the database and implements the protocol for transferring the query and result between client and database. JDBC technology drivers fit into one of four categories.

The JDBC type 1 driver, also known as the JDBC-ODBC bridge, is a database driver implementation that employs the ODBC driver to connect to the database. The driver converts JDBC method calls into ODBC function calls. The driver is platform-dependent as it makes use of ODBC which in turn depends on native libraries of the underlying operating system the JVM is running upon. Also, use of this driver leads to other installation dependencies; for example, ODBC

must be installed on the computer having the driver and the database must support an ODBC driver. The use of this driver is discouraged if the alternative of a pure-Java driver is available. The other implication is that any application using a type 1 driver is non-portable given the binding between the driver and platform. This technology isn't suitable for a high-transaction environment. Type 1 drivers also don't support the complete Java command set and are limited by the functionality of the ODBC driver.

Sun provides a JDBC-ODBC Bridge driver: sun.jdbc.odbc.JdbcOdbcDriver. This driver is native code and not Java, and is closed source. If a has been written so that loading it causes an instance to be created and also calls DriverManager.registerDriver with that instance as the parameter (as it should do), then it is in the DriverManager's list of drivers and available for creating a connection. It may sometimes be the case that more than one JDBC driver is capable of connecting to a given URL. For example, when connecting to a given remote database, it might be possible to use a JDBC-ODBC bridge driver, a JDBC-to-generic-network-protocol driver, or a driver supplied by the database vendor. In such cases, the order in which the drivers are tested is significant because the DriverManager will use the first driver it finds that can successfully connect to the given URL.

First the DriverManager tries to use each driver in the order it was registered. (The drivers listed in jdbc.drivers are always registered first.) It will skip any drivers that are untrusted code unless they have been loaded from the same source as the code that is trying to open the connection. It tests the drivers by calling the method Driver.connect on each one in turn, passing them the URL that the user originally passed to the method DriverManager.getConnection. The first driver that recognizes the URL makes the connection.

Connections in JDBC support creation and esecution of instructions. They can be SQL statement such as INSERT, UPDATE,DELETE, interrogations such as SELECT or calls to a stored procedure. There are 3 kinds of instructions supported.

- Statement – instructions will be sent to the database once at a time
- Prepared Statement – instructions will be compiled just once such that the later calls are more efficient
- Callable Statement – used for calling stored procedures.

Write commands such as INSERT, UPDATE and DELETE give us back a value that indicates how many lines are affected from the instructions. Queries give us back a result set (class ResultSet) It's also possible to move in the result over each line.

This will be our choice, in particular the JDBC J driver which will create the connection between our software and the MySQL database.

8 Acknowledgements

I would like to thank my supervisor, Prof.sa. Sonia Bergamaschi, for the patient guidance, encouragement and advice he has provided throughout my time as his student. I have been extremely lucky to have a supervisor who cared so much about my work, and who responded to my questions and queries so promptly, but mostly I must thank her for her sympathy and her positive thinking that encouraged me even more.

I must express my gratitude to Mariarosa, my fiancé, for her continued support and encouragement. I was continually supported and encouraged by her willingness to psychologically sustain and motivate me in these last years of study and by her confidence in my capacities. I would also like to thank my mother for her patience in these years so hard and my father for continuous support, both experienced all of the ups and downs of my years in the University.

I would like to express my gratitude to Divesh Srivastava for his insight and helpful thoughts on the anonymization matter and for his tutorials on anonymization that worked for me guide in this matter of study, Christine Task of the Purdue University for her bright full presentation on Differential Privacy and for her willingness to answer so many questions. Finally I would like to thank Aaron Roth and Ashwin Machanavajjhala for both their courses on anonymization that ease up the study of this brand new and difficult matter.

I would also like to thank all the members of staff at Datariver who helped me in the process of defining, analysing solving the problem, in particular I would like to thank all of those who always answered to my questions and request for informations politely during the course of analysing the database of FIL and perfectioning of my software.

Sincerely thanks again to everybody,

Gabriele Trombetta

9 References

[1] [Ramakrishnan 2007] : https://www.cs.duke.edu/courses/fall09/cps116/lectures/09-privacy-notes.pdf

[2] Privacy rights clearinghouse: A chronology of privacy breaches, http://www.privacyrights.org/ar/chrondatabreaches.htm.

[3] [Sweeney IJUFKS 2002] Massachussetts privacy breach

[4] Aol privacy breach: http://www.washingtonpost.com/wp-dyn/content/article/2006/08/07/ar2006080700790.html

[5] Mantelero, Privacy, in Contratto e Impresa, 2008, pp. 757-779 17-04-2010

[6] Cass. 27 maggio 1975 n. 2129

[7] Latanya Sweeney. k-Anonymity: a model for protecting privacy. International journal of uncertainty, fuzziness, and knowledge-based systems 2002.

[8] Pierangela Samarati, Latanya Sweeney. Generalizing data to provide anonymity when disclosing information. In ACM PODS 1998

[9] Kristen LeFevre, David J. DeWitt, Raghu Ramakrishnan. Incognito: efficient full-domain k-anonymity. In ACM SIGMOD Conference 2005.

[10] Graham Cormode , Divesh Srivastava, Anonymized data: generation, models, usage, Proceedings of the 35th SIGMOD international conference on Management of data, June 29-July 02, 2009, Providence, Rhode Island, USA

[11] Ashwin Machanavajjhala, Johannes Gehrke, Daniel Kifer, Muthuramakrishnan Venkitasubramaniam. l-Diversity: privacy Beyond k-anonymity. In ICDE 2006

[12] Ninghui Li, Tiancheng Li and Suresh Venkatasubramanian. t-closeness: privacy beyond k-anonymity and l-diversity. In ICDE 2007

[13] C. Dwork, F. Mcsherry, K. Nissim, and A. Smith. Calibrating Noise to Sensitivity in Private Data Analysis. In Theory of Cryptography Conference, 2006.

[14] A. Narayanan, V. Shmatikov. Robust de-anonymization of large sparse datasets (How to break anonymity of the Netflix prize dataset). In S&P 2008.

[15] Kristen LeFevre, David J. DeWitt, Raghu Ramakrishnan. Mondrian multidimensional k-anonymity. In ICDE 2006.

[16] Ninghui Li, Tiancheng Li and Suresh Venkatasubramanian. t-closeness: privacy beyond k-anonymity and l-diversity. In ICDE 2007

[17] Xiaokui Xiao, Yufei Tao. Anatomy: simple and effective privacy preservation. In VLDB 2006

[18] MOMIS : http://www.dbgroup.unimo.it/Momis/

[19] FIL : http://www.filinf.it/

[20] Graham Cormode, Magda Procopiuc, Divesh Srivastava: Differentially Private Spatial Decompositions

[21] A. Ghosh, T. Roughgarden, and M. Sundararajan. Universally utility-maximizing privacy mechanisms. InSTOC, 2009.

[22] Christine Task - Purdue University - A Practical Beginner's Guide to Differential Privacy - CERIAS presentation.

[23] http://www.oracle.com/us/corporate/press/044428

[24] http://docs.oracle.com/javase/specs/

[25] So why did we decided to call it java : http://www.javaworld.com/javaworld/jw-10-1996/jw-10-javaname.html

[26] http://www.oracle.com/technetwork/java/index.html#349

[27] Q: What components of the JDK software are you open sourcing today? A: We're open sourcing the Java programming language compiler ("javac"), and the Java HotSpot virtual machine."Free and Open Source Java FAQ; the source is being released via the OpenJDK project.

[28] http://docs.oracle.com/javase/specs/

[29] Excerpts from http://it.wikipedia.org/wiki/Java_(linguaggio_di_programmazione)#cite_note-JLS.2C_prefazione_alla_prima_edizione-3

[30] Excerpts from http://it.wikipedia.org/wiki/MySQL

[31] All the informations are in the http://www.wampserver.com/en/ documentation.

[32] http://www.oracle.com/technetwork/java/javase/tech/index-jsp-136101.html

[33] Excerpts from http://it.wikipedia.org/wiki/JDBC#cite_note-1

[34] F.McSherry and K.Talwar. Mechasim Design via Differential Privacy. Proceedings of the 48th Annual Symposium of Foundations of Computer Science, 2007

[35] Based on notes of Aaron Roth's course on "The Algorithmic Foundations of Data Privacy "

http://www.cis.upenn.edu/~aaroth/courses/slides/Lecture3.pdf

[36] Notes on the Exponential Mechanism Boston University CS 558. Sharon Goldberg phd and assistant professor in the BU Computer Science Department.

[37] Graham Cormode, Magda Procopiuc, Divesh Srivastava - Differentially Private Publication of Sparse Data

[38] "Programming Language Popularity". 2009. Retrieved 2009-01-16.

[39] "TIOBE Programming Community Index". 2009. Retrieved 2009-05-06.

[40] "The History of Java Technology". Retrieved October 6, 2012.

[41] "Market Share".Why MySQL?. Oracle. Retrieved 17 September 2012.

[42] "DB-Engines Ranking". Retrieved 26 February 2013.

[43] Schumacher, Robin; Lentz, Arjen. "Dispelling the Myths". MySQL AB. Archived from the original on 6 June 2011. Retrieved 17 September 2012.

[44] What is MySQL?". MySQL 5.1 Reference Manual. Oracle. Retrieved 17 September 2012. "The official way to pronounce “MySQL” is “My Ess Que Ell” (not “my sequel”)"

[45] "Wikimedia servers — System architecture". Wikimedia Meta-Wiki. Wikimedia Foundation. Retrieved 17 September 2012.

[46] Urlocker, M. Zack (13 December 2005). "Google Runs MySQL".The Open Force. M. Zack Urlocker. Retrieved 3 August 2010. "AdWords was built using the MySQL database"

Claburn, Thomas (24 April 2007). "Google Releases Improved MySQL Code". InformationWeek (CPM Media). Retrieved 30 November 2008.

[47] Callaghan, Mark (13 April 2010). "MySQL at Facebook". YouTube(Google). Retrieved 3 August 2010. "x,000 servers, ... Master-slave replication, InnoDB"

Sobel, Jason (21 December 2007). "Keeping Up". The Facebook Blog. Facebook. Retrieved 30 October 2008.

Malik, Om (25 April 2008). "Facebook’s Insatiable Hunger for Hardware". GigaOM. GigaOmniMedia. Retrieved 30 October 2008.

[48] MySQL Workbench, MySQL Downloads

[49] "Java SE Technologies - Database"

[50] PINQ http://research.microsoft.com/en-us/projects/PINQ/

[51] Maurizio Lenzerini (2002). "Data Integration: A Theoretical Perspective". PODS 2002. pp. 233–246.

[52] Excerpt from Wikipedia page on "Data Integration"

[53] Excerpt from Chapter 14 of " Handbook of Conceptual Modeling - Theory, Practice, and Research Challenges. David W. Embley and B. Thalheim, Eds. Springer, 1st Edition., 2011, XX, 587 p., ISBN: 978-3-642-15864-3. , from Sonia Bergamaschi, Domenico Beneventano, Francesco Guerra, and Mirko Orsini.

[54] Noy NF,Doan A, Halevy AY (2005) Semantic integration. AI Mag 26(I):7-10

[55] Dalenius T. Towards a methodology for statistical disclosure control. Statistisk Tidskrift, 5:429–444, 1977

[56] [Dwork and Naor 2006]

[57] Dan Feldman, Amos Fiat, Haim Kaplan, and Kobbi Nissim. 2009. Private coresets. InProceedings of the 41st annual ACM symposium on Theory of computing (STOC '09). ACM, New York, NY

[58] Ashwin Kumar V. Machanavajjhala"Defining and Enforcing Privacy in Data Sharing" June 26, 2008