

# Progetto Basi di Dati Politecnico di Milano

## Realizzazione di un programma automatizzato per la creazione e l'analisi statistica di questionari in ambito economico

Francesco De Liva    Luca Di Minervino

### 0. Introduzione

La traccia del progetto nasce dalla volontà di migliorare il database creato da FDL<sup>1</sup> per il progetto PLAN D: "Euroregion: youth, information and economic identity in the new Europe" attuato da R.U.E. - Risorse Umane Europa, che nello specifico di questo progetto ha creato un sito apposito chiamato ECLAB - Economic Lab visibile all'indirizzo: [www.eclab.eu](http://www.eclab.eu).

Il database originario era composto da un'unica tabella e circa 260 attributi.

Quando abbiamo iniziato a progettare il nuovo database abbiamo voluto impegnarci per costruire un sistema che non fosse solamente compatibile con il database passato, ma che fosse più possibile adattabile a esigenze future.

Le idee alla base erano che il database doveva poter contenere più questionari e più annate di compilazione. Il database è stato strutturato per distinguere nettamente il visitatore, ovvero l'impresa che vuole compilare il questionario dalle sessioni in cui compila il questionario, difatti un'impresa a distanza di anni può compilare il questionario più volte (per questo la tabella annata). Quindi un visitatore a cui è associata una sessione e un questionario compila il questionario che è composto di *SuperDomande* (il titolo della domanda) che a sua volta sono composte da *Domande* che possono essere *singlechoice*, *multiple choice* o *matriciali* (ovvero mix di *single* e *multiple choice*). Ogni *Domanda* ha delle *risposte possibili* e ogni sessione può avere associato alla tabella *risposte date* delle risposte possibili.

Il primo e forse più impegnativo dopo aver progettato e creato le tabelle (di cui si allega lo schema ER) è stato popolare il database. Difatti era opportuno che le applicazioni che andavamo a sviluppare avessero un continuo test sui dati. La popolazione del database svolta interamente da LDM ha creato un numero impressionante di tuple nella tabella *risposte date*, circa 40 mila.

Superato il popolamento del database il lavoro si è svolto autonomamente, FDL ha seguito la parte relativa alle analisi statistiche sui dati, mentre LDM ha seguito la realizzazione dell'interfaccia per amministrare i questionari.

### 1.1 Database

Ecco una visione più approfondita del database effettuata in base alla necessità di creare un sistema di creazione, amministrazione e analisi di questionari economici.

Un questionario difatti è formato da più pagine numerate con un nome e un titolo.

Una pagina è formata da un elenco di superdomande numerate con un nome ed un titolo.

Una superdomanda è caratterizzata da un testo, da un modello predefinito e da un elenco di lunghezza variabile di domande: il modello è un attributo che indica che tipo di domande sono contenute nella

---

<sup>1</sup> FDL e LDM sono studenti meritevole regolarmente iscritto al terzo anno di Ingegneria Informatica al Politecnico di Milano.

superdomanda e in che modo devono essere visualizzate dall'interfaccia grafica, (i modelli possono essere ad esempio: matrice di multiple-choice, gruppi di single-choice, matrice mista, ecc.).

Una domanda è caratterizzata da un testo, un tipo ed un elenco di opzioni; il tipo può indicare: multiple-choice (radio button, menu a tendina) oppure elenco di single-choice (cioè elenco di checkbox).

Ogni questionario può essere valido per diverse annate, per ognuna delle quali gli utenti sono invitati a rispondere nuovamente al questionario. Un'annata è caratterizzata da data di apertura e data di chiusura.

Si vuole memorizzare "CHI RISPONDE COSA, QUANDO".

E' richiesto che sia disponibile un archivio di superdomande separate dai questionari.

Un amministratore di questionari deve poter creare superdomande (sempre a partire da uno dei modelli predefiniti) e inserirle nell'archivio oppure prelevare dall'archivio una superdomanda precedentemente salvata, copiarla, aggiungere delle domande o delle risposte e salvare la nuova superdomanda ottenuta.

Quando l'amministratore crea un nuovo questionario, lo fa copiando le superdomande dall'archivio.

Deve essere presente un attributo che indica che l'amministratore ha confermato il questionario creato e questo non è più cancellabile, ma soltanto parzialmente modificabile.

E' richiesto che l'archivio e i questionari siano virtualmente in DB diversi, perché le superdomande possono essere cancellate dall'archivio. In realtà si ha a disposizione un solo DB.

Di un utente si vuole conoscere nome dell'impresa, e-mail, città, dimensione dell'impresa, codice utente che gli viene assegnato. Ogni volta che un utente compila un questionario si deve poter memorizzare la percentuale di domande alla quale ha risposto, se ha risposto a tutte le domande, e se le risposte che ha dato verranno considerate nelle analisi statistiche.

## 1.2 Modello logico:

La chiave primaria di ogni tabella è un identificativo di nome "id" che si autoincrementa, in modo da avere maggiore semplicità e maggiore garanzia che i vincoli di unicità non vengano violati.

Tutte le gerarchie collasano verso l'alto, questo provoca qualche campo inutilizzato, ma si hanno query più semplici e compatte.

L'attributo "tipo" è ridondante, perché si ripete in "domanda", "risp\_possibile" e "risp\_data", ma questo dà la possibilità di eliminare delle JOIN da query molto frequenti.

```
QUESTIONARIO(id,nome,titolo,isConfermato);
PAGINA(id,titolo,numero ,id_questionario)
SUPERDOMANDA(id,
testo,modello,nRighe,nColonne,nomeMatrice,numero,coloreSfondo,id_pagina)
NOMERIGHECOLONNE(id,nome,isRow,numero,id_superdomanda)
DOMANDA(id,testo,tipo,obbligatoria,numero,id_superdomanda)
RISP_POSSIBILE(id,numero,stringa,tipo,id_Domanda)
RISP_DATA(id,tipo,risposta,id_sessione,id_risp_possibile)
SESSIONE(id,data,percentuale_risposte,user_usata,pwd_usata,nSessioniPrecedenti,n
BuoneCompilazioni,isBuonaCompilazione,isConsiderato,id_annata,id_visitatore)
ANNATA(id,dataApertura,dataChiusura,isAperta,numero,id_questionario)
VISITATORE(id,nomeImpresa,dimensione,citta,email,user_attuale,pwd_attuale)
AMMINISTRATORE(id,user,pwd)
```

```
ARC_SUPERDOMANDA(id, testo,modello,nRighe,nColonne,nomeMatrice,numero,sfondo)
ARC_NOMERIGHECOLONNE(id,nome,isRow,numero,id_superdomanda)
ARC_DOMANDA(id,testo,tipo,obbligatoria,numero,id_superdomanda)
ARC_RISP_POSSIBILE(id,numero,stringa,tipo,id_Domanda)
```

"Arc" sta per archivio

## 2.1 Amministrazione

Accedendo con un browser alla pagina "autenticazione.php" un amministratore si può autenticare e quindi può accedere alla pagina di amministrazione. Dalla pagina di amministrazione è possibile:

- 1) Vedere l'elenco dei questionari presenti nel DB e i relativi URL per la compilazione.
- 2) Vedere l'elenco di superdomande presenti in archivio, che possono essere utilizzate per la creazione di altri questionari.

Le superdomande che hanno modello 1 sono destinate automaticamente alla prima pagina (o pagina introduttiva) di un questionario, le superdomande con modello successivo al primo sono destinate automaticamente alle pagine successive di un questionario.

- 3) Creare una nuova superdomanda da inserire in archivio:

modello 1: single-choice per pagina introduttiva, è una single-choice che sarà automaticamente obbligatoria in fase di compilazione.

modello 2: matrice di single-choice con numero di righe e di colonne variabile

modello 3: matrice di single-choice più una colonna di checkbox

modello 4: single-choice per pagina successiva alla prima

modello 5: elenco di gruppi di multiple-choice

modello 6: elenco di gruppi di multiple-choice divisi su due colonne

modello 7: una o più single-choice

Il testo di una sottodomanda è sempre opzionale.

Quando una domanda è obbligatoria, il sistema non accetta la risposta se è quella di default e ripropone la domanda.

Ogni volta che si crea una single-choice (in qualsiasi modello) la prima "risposta possibile" che viene inserita sarà considerata automaticamente risposta di default, e non verrà considerata dall'algoritmo di analisi delle risposte. Nel caso in cui la single-choice sia obbligatoria la prima "risposta possibile" verrà considerata come risposta non data, e nel caso questa sia selezionata il sistema riproporrà la domanda.

Si Consigliava pertanto di settare la prima risposta di una single-choice come ad esempio: "Scegli...", o "Non voglio rispondere", o "Non conosco la risposta".

In fase di creazione di una superdomanda non è possibile annullare le operazioni effettuate, ma è possibile cancellare la superdomanda creata, e creare una nuova superdomanda, oppure creare direttamente una nuova superdomanda (anche uguale).

Quando si clicca sul "link" "COMPLETA E TERMINA CREAZIONE" o si abbandona la fase di creazione, la superdomanda è inserita in archivio e non può più essere modificata, ma può essere cancellata con l'apposita funzione.

- 4) Cancellare una superdomanda dall'archivio. La cancellazione non influirà sui questionari che sono "virtualmente" in un DB separato dall'archivio. La cancellazione lascerà delle tuple orfane nel DB per motivi di sicurezza dei dati, ma queste non creeranno nessun problema, saranno invisibili agli amministratori e ai visitatori.

- 5) Cancellare un questionario creato o in fase di creazione. Anche qui la cancellazione lascia delle tuple orfane per motivi di sicurezza, ma saranno invisibili per amministratori e visitatori e non creeranno alcun problema.

- 6) Visualizzare l'elenco delle imprese che hanno compilato almeno una volta almeno un questionario e i relativi codici utenti.

- 7) Creare un questionario: è possibile creare diverse pagine, inserendo in ogni pagina delle superdomande presenti in archivio, una alla volta.

Per la creazione della prima pagina il sistema propone automaticamente superdomande di modello 1, per le pagine successive il sistema propone tutte le altre superdomande presenti.

Le superdomande verranno presentate a chi compila nell'ordine di inserimento.

Quando si clicca sul pulsante "CREA NUOVA PAGINA" la pagina precedente viene automaticamente salvata e non sarà più modificabile.

Le operazioni effettuate in fase di creazione di un questionario non possono essere annullate, ma è possibile cancellare il questionario creato e cominciare a creare un nuovo questionario, oppure creare direttamente un nuovo questionario, anche uguale.

Quando si clicca sul pulsante "COMPLETA E TERMINA CREAZIONE" il questionario non è più modificabile, ma può essere cancellato con l'apposita funzione presentata nella pagina.

ATTENZIONE: cancellare un questionario vuol dire non poter più vedere le risposte date. In caso di cancellazioni accidentali sarà possibile recuperare facilmente TUTTI i dati agendo direttamente sul DB in linguaggio SQL.

### 3.1 Compilare un questionario

Dall'amministrazione, nella sezione "visualizza elenco questionari" è possibile prelevare l'URL per la compilazione del questionario che si vuole somministrare. Si consiglia di creare un link con l'URL interessato in un'altra pagina web.

Quando si accede all'URL per la compilazione di un questionario viene presentato un form per la registrazione dell'impresa se il visitatore non ha mai compilato nessun questionario in passato, ed un form per l'inserimento del "codice utente" se il visitatore ha già compilato altri questionari. Il codice utente deve essere comunicato dall'amministratore che invita l'impresa a compilare un questionario, perché è il solo a poter vedere i codici utenti.

Successivamente il visitatore accede alla prima pagina del questionario.

Il sistema calcola automaticamente la percentuale di risposte date se il questionario viene compilato solo parzialmente.

### 4.1 Analisi Statistica

E' doveroso ricordare che l'applicazione per le analisi dei dati non ha lo scopo di essere distribuita ad un qualsiasi utente, ma ad una tipologia di utente che è un professionista nel campo dell'analisi statistica, per cui non è prevista nessuna spiegazione sugli output, poiché per un esperto sono di facile interpretazione. Nonostante ciò abbiamo cercato di semplificare la vita all'utente finale con una grafica volta alla semplicità.

Il software scelto per l'implementazione di questa parte è stato JAVA (1.6). Il primo problema che si è posto è stato quello relativo alla connessione dell'applicazione al Database. Difatti il database originariamente era ospitato all'interno del sito di Eclab, ma poiché il server su cui è ospitato questo dominio non consente l'accesso remoto, è stato necessario spostare il database all'interno del dominio dell'associazione R.U.E.

Una delle prime cose che abbiamo testato sul database (MySQL 4.1.22) è stata la compatibilità di questo con le query studiate a lezione. Con rammarico abbiamo notato che solo i costrutti SQL più semplici venivano interpretati dal server, mentre quelli più complicati, anche a volte semplicemente di una "GROUP BY", non venivano accettati. Questo difatti non è stato un problema da poco, poiché l'applicazione durante le association rules aveva la necessità di query veloci che siamo stati costretti a risolvere con "SELECT" annidate come unica soluzione funzionante rispetto a molte altre viste durante il Corso di Basi di Dati.

Risolto il problema della connessione dell'applicazione Java al database i problemi maggiori di tutto lo sviluppo dell'applicazione sono stati di realizzare un'applicazione più astratta possibile rispetto al questionario e costruire query SQL che avessero il supporto per tutte le future funzioni del questionario.

L'applicazione, chiamata StatTOOL è composta da 6 package (come si può vedere nel diagramma UML allegato): *Database* gestisce tutte le interazioni dell'applicazione con il server (per completezza l'IP statico: 66.71.188.1) , *Statistica* contiene tutte le informazioni su uno specifico database e contiene delle classi che sono lo stampino per ogni superdomanda, inoltre ha le classi che riguardano la selezione degli utenti, *Interfacce* contiene le classi che servono a fare ponte fra le varie package e inoltre contiene il main, *AssociationRules* è composto da una classe, un po' slegata dalla struttura, difatti per motivi di velocità e esecuzione grafica è stata creata un'unica classe che svolge tutti i suoi compiti usando solamente l'interfaccia connessione, infine ci sono *Grafica* e *Grafici* che rappresentano l'interfaccia grafica per l'utente.

L'applicazione consente all'utente di collegarsi al database, selezionare il questionario e le sessioni interessate e poi generare statistiche relative a questi.

## 4.2 Statistiche offerte

Le statistiche sono di tre tipi:

1. Sulla singola domanda
2. Sulla singola sessione
3. Regole di associazione

Per quanto riguarda le **statistiche sulle singole domande** bisogna effettuare un'ulteriore distinzione, ovvero ci sono tre modi di visualizzare i risultati: *Normale* ovvero per ogni singola domanda si visualizzano diagrammi a torta o a barre che interessano tutte le risposte date a quella specifica SuperDomanda, *ATECO* si sceglie di filtrare i dati in base alla tipologia di settore in cui opera l'impresa e *Dimensione* ovvero filtrare i dati in base alle dimensioni delle imprese.

In totale questa sezione consente di effettuare più di 300 diversi grafici di tipo semplice o comparativo sui dati in possesso.

La sezione **statistiche sulle singole sessioni** è molto semplice ed è stata creata essenzialmente per garantire i vincoli sull'integrità del database ovvero per consentire di visualizzare tutte le risposte date in una singola sessione e poter garantire che quella sessione sia utile per l'analisi dei dati.

La sezione **Regole di associazione** è stata la più complicata da realizzare, difatti le conoscenze a noi in possesso erano insufficienti per affrontare il tema ed è stato necessario attingere a corsi della specialistica, nello specifico "Tecniche di Apprendimento Automatico per Applicazioni di Data Mining". Essenzialmente l'applicazione (in accordo all'algoritmo Apriori) seleziona tutte le risposte date tra quelle possibili e attribuisce a ciascuna di queste un contatore in base a quanti visitatori nelle varie sessioni hanno accordato la preferenza. Questo contatore rappresenta il supporto. Successivamente si selezionano tutte quelle risposte che hanno il contatore maggiore di una certa soglia. Fatto questo primo passaggio si creano insiemi formati da più elementi dove l'elemento è la risposta. Per ogni insieme si calcola il contatore in base all'intersezione delle risposte fra di loro e si selezionano sempre quegli insiemi che contengono un contatore maggiore del supporto minimo.

Ottenuti tutti questi insiemi (anche di un solo elemento che chiamiamo frequent itemset I) di risposte date con contatore appropriato vada cercare le associazioni tra di loro con il seguente algoritmo:

```
for every frequent itemset Ij
  for every Ih ⊆ Ij, Ih ≠ ∅, Ih ≠ Ij
  {
    if freqT(Ij) / freqT(Ih) ≥ min_conf
    output the rule Ih ⇒ (Ij - Ih)
  }
}
```

Durante l'implementazione di questi algoritmi si sono verificati notevoli problemi dal punto di vista della durata dell'esecuzione dell'analisi. Difatti ogni singola query eseguita sul server ha un tempo di circa 0.0015 secondi (usando una connessione a fibra ottica).

Per far girare l'algoritmo Apriori bisogna, rispetto all'insieme di tutte le singole risposte selezionate (la cui cardinalità chiameremo M), eseguire un numero di query equivalente alle combinazioni a due a due degli M elementi. E successivamente le combinazioni a tre a tre, che però sebbene abbiamo abbassato il più possibile il supporto minimo non sono comparsi insiemi di tre elementi, per cui abbiamo deciso di togliere questa parte dell'algoritmo e limitarli ad insiemi di cardinalità uno e due.<sup>2</sup>

---

<sup>2</sup> Questo poiché l'algoritmo a priori si può sviluppare bene se le risposte date sono in un numero molto elevato rispetto alle risposte possibili, mentre nel nostro caso sebbene avessi più di 28 mila risposte, queste non erano sufficienti ad avere un'implementazione con insiemi con cardinalità superiore a 2.

Quindi l'applicazione svolge  $M! / ((M-2)! * 2!)$  operazioni che vanno moltiplicate per la durata di una query. Si capisce che il tempo di durata dell'algoritmo dipende dal supporto in forma esponenziale, difatti maggiore è il supporto minore è la cardinalità M e di conseguenza le sue combinazioni.

### 4.3 Vincoli sulla Applicazione

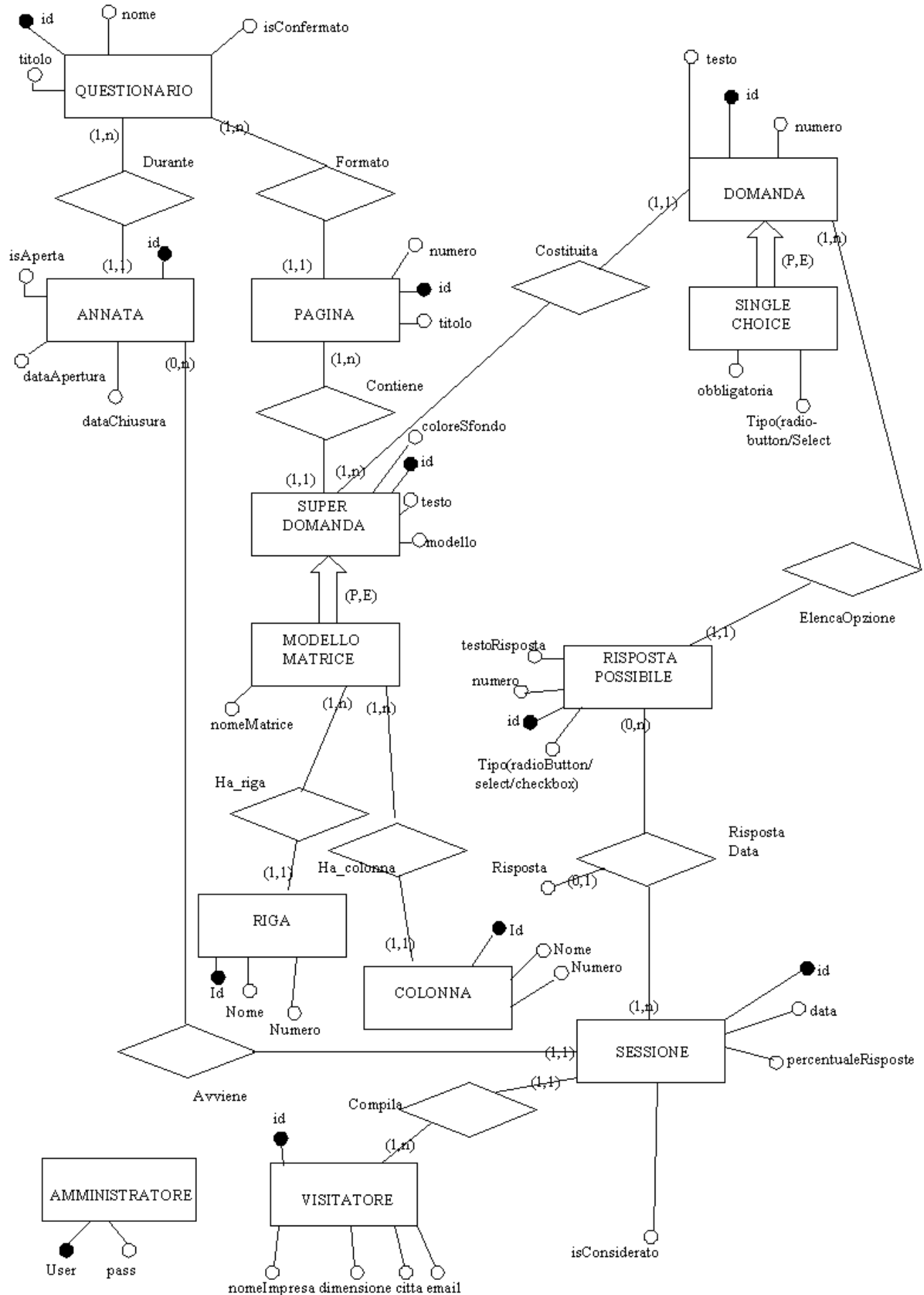
Durante lo sviluppo dell'applicazione abbiamo dovuto risolvere il problema dei vincoli sui dati. Per ottenere la maggiore indipendenza possibile da possibili errori, per ogni singola statistica il programma accede prima alle SuperDomande del questionario selezionato, poi ottiene gli Id delle domande associate, entra nelle domande e impugna gli Id associati alle risposte possibili, successivamente entra in risposte possibili e ottiene ulteriormente gli Id per le risposte date. Non si scorrono i database andando al Id successivo, sempre usando un riferimento diretto all'Id interessato.

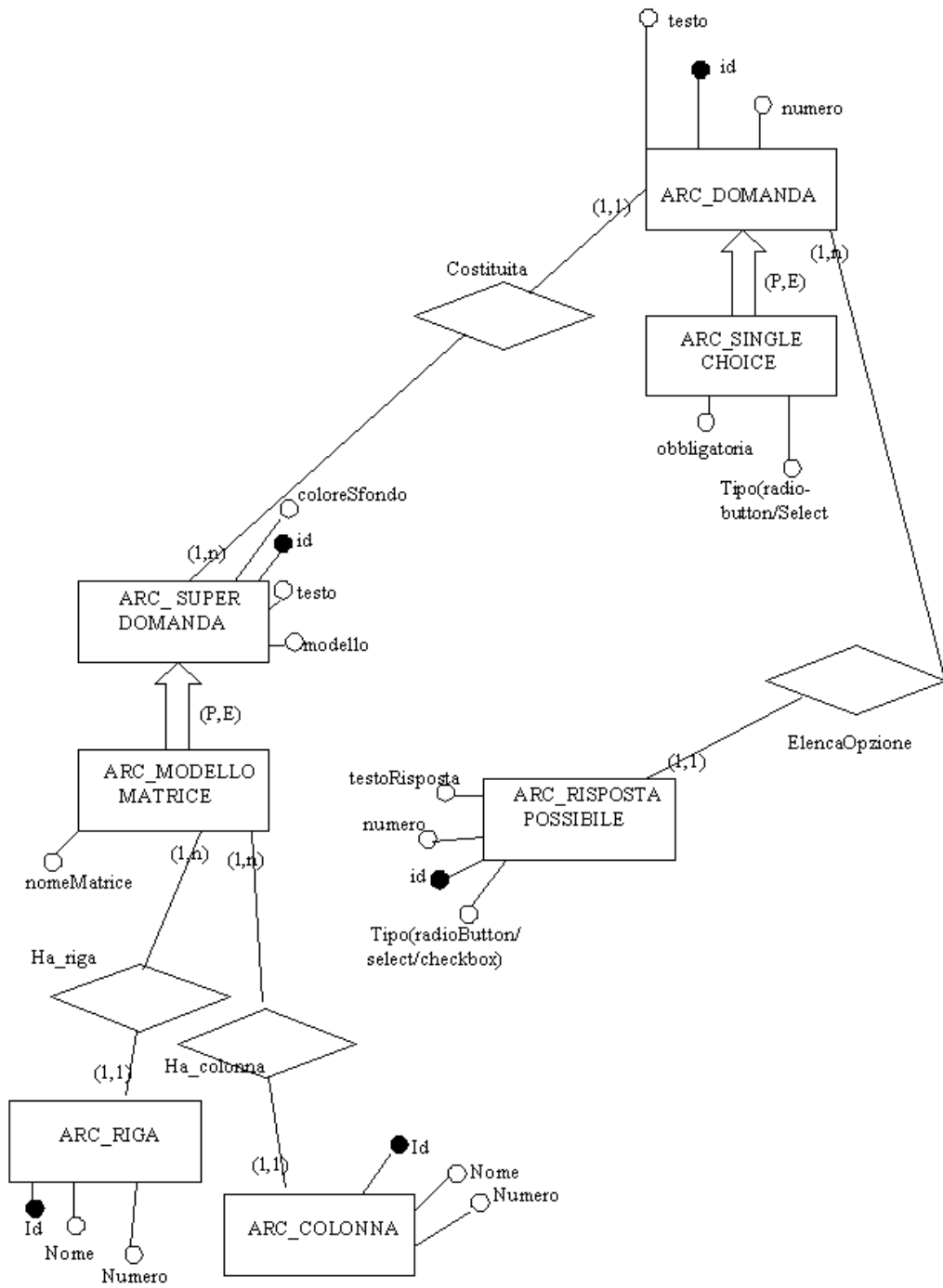
Per quanto riguarda i vincoli sui dati, il sistema, essendo basato interamente su single choice e multiple choice, impedisce di inserire valori che siano fuori dal range di quelli decisi a priori, per cui essenzialmente a meno di errori del database stesso che non sono stati trattati, il sistema non può presentare valori fuori dai range. E nonostante ciò se ci fossero valori fuori dai range questi non verrebbero visti dall'algoritmo di selezioni successive usato. Essenzialmente l'applicazione legge solo quello che conosce già, mentre le tuple "sbagliate" le ignora.

Un problema è stato gestire tutti i valori NULL che il database possiede in seguito al popolamento a partire da quello precedente. Il programma statistico, difatti necessita che alcuni campi chiave del database, come visitatore e sessione abbiano la gran parte degli attributi Not NULL, ma questo è garantito dal database.

# Allegati

Schema ER:





# DESCRIZIONE INFORMALE CLASSI APPLICAZIONE INTERFACCIA PHP

(Elenco delle classi php e dei nomi dei metodi che contengono)

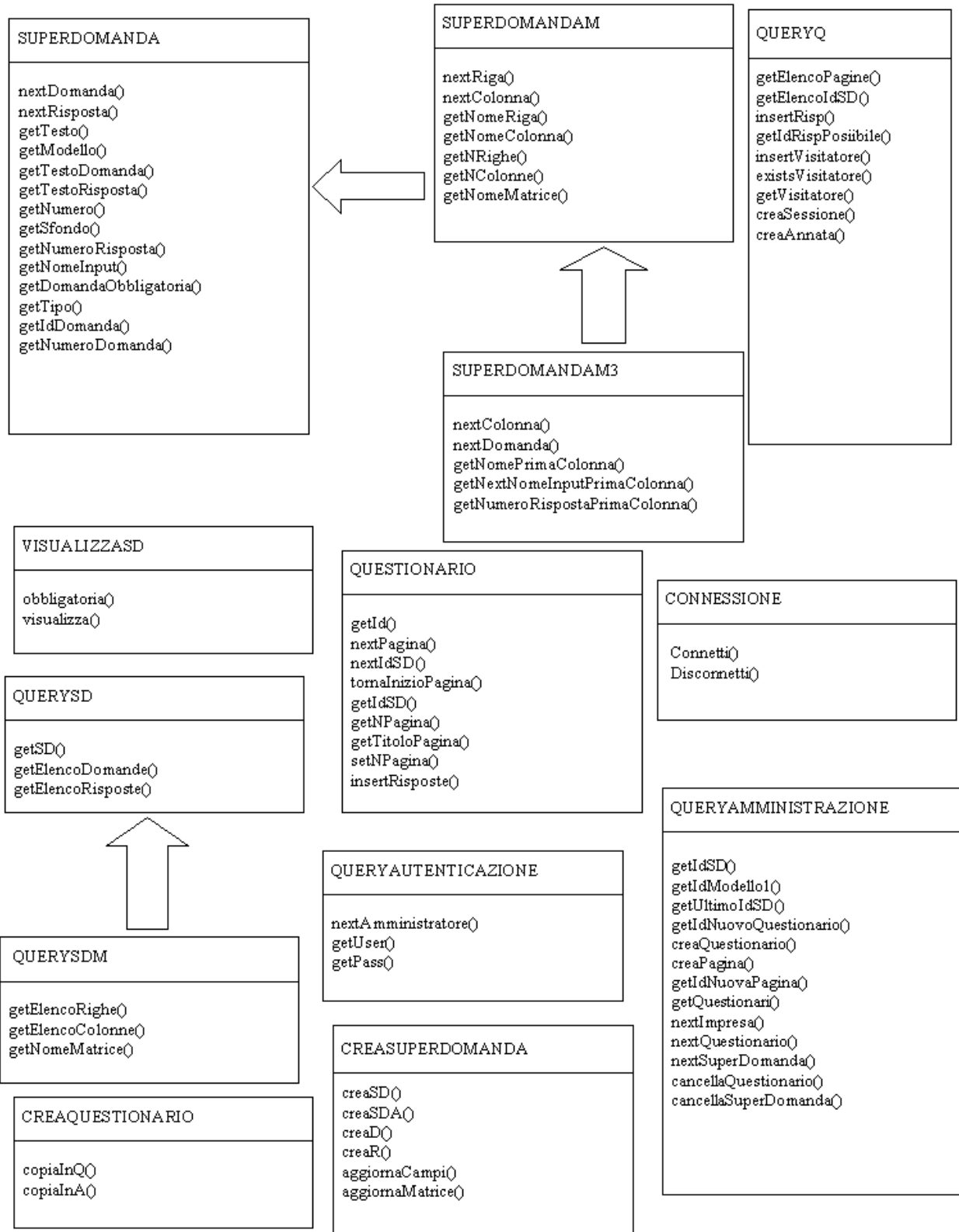


Diagramma UML delle dipendenza tra le Package dell'applicazione JAVA:

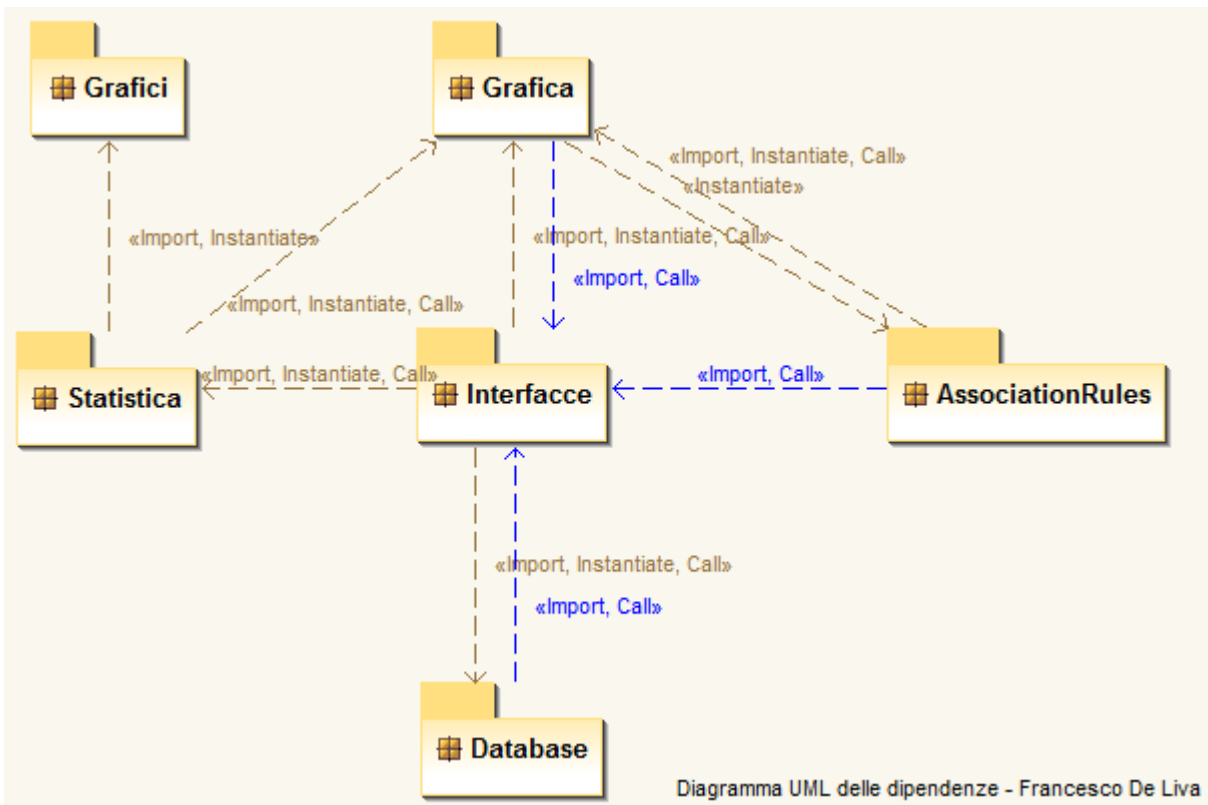


Diagramma UML delle classi della Package Statistica dell'applicazione JAVA:

