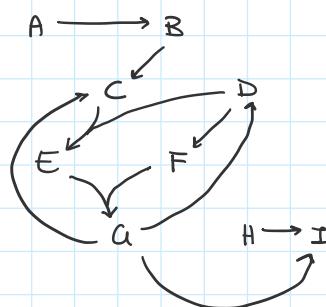


$$\left\{ \begin{array}{l} A \rightarrow B \\ B \rightarrow C \\ CD \rightarrow E \\ D \rightarrow F \\ EF \rightarrow G \\ G \rightarrow CD \\ H \rightarrow I \end{array} \right.$$



AH devono sfare sicuro nella chiave:

$$AH^+ = \{ A B C \}$$

$$AHD^+ = \{ A B C D E F G H I \} \rightarrow \text{chiave}$$

$$AHE^+ = \{ A B C E H I \} \rightarrow \text{NO}$$

$$AHF^+ = \{ A B C F H I \} \rightarrow \text{NO}$$

$$AHG^+ = \{ A B C D E F G H I \} \rightarrow \text{chiave}$$

$$\text{Chiavi} = \{ AHD, AHG \}$$

## APPUNTI DI INGEGNERIA INFORMATICA

GAIA BERTOLINO

Non è in BNFC:

$$A^+ = \{ A B C \}$$

$$x = A$$

$$y = ABC$$

$$z = DEFGHI$$

$$(ABC)$$

$$(ADEFGHI)$$

$$\left\{ \begin{array}{l} A \rightarrow B \\ B \rightarrow C \end{array} \right.$$

$$\left\{ \begin{array}{l} D \rightarrow F \\ EF \rightarrow G \\ G \rightarrow I \\ G \rightarrow D \\ H \rightarrow I \end{array} \right.$$

$$A^+ = \{ A B C \}$$

$$B^+ = \{ BC \}$$

$$x = B$$

$$y = BC$$

$$z = BA$$

$$(BC) \quad (BA)$$

$$CD \rightarrow E$$

sì perde

$$G \rightarrow C$$

sì perde

$$D^+ = \{ DF \}$$

$$x = D$$

$$y = DF$$

$$z = ADEGHI$$

$$(DF)$$

$$(ADEGHI)$$

$$\begin{array}{l} EF \rightarrow G \\ \downarrow \\ ED \rightarrow G \end{array}$$

$(BC)^*$

$$\{B \rightarrow C\}$$

$$B^+ = \{BC\}$$

$\underbrace{\hspace{1cm}}_{BCNF}$

${}^*(BA)$

$$\{A \rightarrow B\}$$

$$A^+ = \{AB\}$$

$\underbrace{\hspace{1cm}}_{BCNF}$

$\downarrow$   
 $(DF)$

$$\{D \rightarrow F\}$$

$$D^+ = \{DF\}$$

$\underbrace{\hspace{1cm}}_{BCNF}$

$(ADEGHIT)$

$$\boxed{ED \rightarrow G}$$

$$G \rightarrow I$$

$$G \rightarrow D$$

$$H \rightarrow I$$

$$G^+ = \{A \sqsupseteq D\}$$

$$H^+ = \{H \sqsupseteq I\}$$

$$EF \rightarrow G$$

$$\downarrow$$
  
 $ED \rightarrow G$

$\downarrow$   
si perde

$$x = H$$

$$y = HI$$

$$z = ADEGH$$

$(HI)$

$$\{H \rightarrow I\}$$



$\downarrow$   
 $(ADEGH)$

$$\begin{cases} ED \rightarrow G \\ G \rightarrow D \end{cases}$$

$G \rightarrow I \rightarrow$  si perde  
 $\downarrow$   
 $ED \rightarrow G$

$$ED^+ = \{EDG\}$$

$$G^+ = \{$$

# APPUNTI DI INGEGNERIA INFORMATICA

GAIA BERTOLINO

22 feb 2019

martedì 1 febbraio 2022 21:00



APPELLO\_B  
ASI\_DI\_D...

**Corso di Laurea in Ingegneria Informatica (DM 270)**  
**Esame di Basi di Dati**  
**22 febbraio 2019**

**Esercizio 1: Progettazione concettuale e logica di una Base di Dati (40 minuti)**

a) Si progetti una base di dati di supporto alla gestione delle prove d'esame svolte all'interno di una scuola di specializzazione.

Le figure professionali di cui tenere traccia nel sistema sono dette *esperti*, e ciascuno di loro è identificato dalla coppia <nome, cognome>, oltre che essere caratterizzato da un recapito. Alcuni esperti sono detti *docenti*, e per essi occorre indicare un titolo di studio in loro possesso che li renda idonei all'insegnamento, assieme alla lista dei *corsi* di specializzazione che hanno curato.

Ciascun corso di specializzazione è caratterizzato dal nome, dall'anno in cui è stato erogato, dal docente che lo ha curato, dalla lista degli *argomenti* trattati nel corso, e dalla lista delle *sedute* d'esame che si sono tenute relativamente al corso. Possono esistere più corsi omonimi, ma non esistono corsi con lo stesso nome erogati nello stesso anno.

Gli argomenti sono identificati da un nome e sono caratterizzati anche da una descrizione. Uno stesso argomento può essere trattato in corsi diversi, e nell'indicare che un argomento è stato trattato in un corso deve essere possibile specificare il numero di ore di lezione dedicate nel corso all'argomento. Deve essere possibile rappresentare anche la propedeuticità tra argomenti: un argomento può essere propedeutico ad un numero qualunque di argomenti, e ad esso possono essere propedeutici un numero qualsivoglia di argomenti.

Ogni *seduta* d'esame è caratterizzata, oltre che dal corso, dalla data, dal luogo di svolgimento, dal *test* che è stato assegnato ai candidati, e può essere associata all'esperto che ha presenziato allo svolgimento della stessa. La data di una seduta la identifica rispetto a quelle relative allo stesso corso. Si noti che uno stesso esperto può avere presenziato ad un numero qualunque di sedute.

Ogni *test* è caratterizzato da un codice (che ne è identificante) e dalla data di creazione. Uno stesso test può essere stato utilizzato in un numero qualunque di sedute d'esame. Ciascun test appartiene esattamente ad una delle seguenti tipologie: "RC" (ossia, test a *risposte chiuse*), o "misto". Occorre indicare la lista dei *quesiti* che compongono i test, facendo attenzione a quanto indicato nel seguito (come vedremo, a seconda del tipo, un test è composto da quesiti di tipi diversi).

Ogni *quesito* è identificato da un codice ed è anche caratterizzato dal testo del quesito stesso e dall'esperto che lo ha definito. Si noti che uno stesso esperto può avere definito un numero qualunque di quesiti. L'insieme dei quesiti è partitionato in due classi: i quesiti a *risposta chiusa* e quelli a *risposta aperta*. I quesiti a risposta chiusa sono caratterizzati dalla risposta corretta, e da due risposte alternative (le risposte corrette ed alternative sono codificate da stringhe di testo). Invece, ogni quesito a risposta aperta è caratterizzato da un'unica stringa di testo che rappresenta una risposta corretta al quesito stesso. Detto ciò, la composizione dei test d'esame segue la seguente specifica: i test RC sono composti da un numero qualunque di quesiti a risposta chiusa, mentre i test misti sono composti da un numero qualunque di quesiti, di qualunque tipo. Ogni quesito può essere presente in un numero qualunque di test (nel rispetto della specifica sopra descritta).

b) Si definisca uno schema relazionale corrispondente allo schema E-R.

**Esercizio 2: Progettazione logica e dipendenze funzionali (30 minuti)**

Si consideri lo schema relazionale  $\langle R(A,B,C,D,E,F,G,H), \mathcal{F} \rangle$ , dove  $\mathcal{F}$  è il seguente insieme di dipendenze funzionali:

AB  $\rightarrow$  E  
CD  $\rightarrow$  F  
E  $\rightarrow$  ACH  
EF  $\rightarrow$  G  
F  $\rightarrow$  BD  
G  $\rightarrow$  E (prosegue nel foglio successivo)

$AG \rightarrow D$   
 $AF \rightarrow C$

Definire:

1. una copertura minimale per  $\mathcal{F}$ ;
2. le chiavi candidate di  $R$ ;
3. una decomposizione in BCNF di  $R$  che sia senza perdita di informazione;
4. una decomposizione in 3NF di  $R$  che sia senza perdita di informazione, senza perdita di dipendenze funzionali e di dimensione minima.

### Esercizio 3: Indici (20 minuti)

Si consideri un sistema con le seguenti caratteristiche:

- i. pagine fisiche di 6 KB;
- ii. intestazione fissa delle pagine fisiche di 144 byte;
- iii. puntatori di 4 byte.

Si consideri la relazione:

*CORSO* (nome, descrizione, docente)

e si assuma che il numero di byte destinato alla rappresentazione di ciascun attributo sia:

- a) 64 byte per l'attributo *nome*;
- b) 232 byte per l'attributo *descrizione*;
- c) 32 byte per l'attributo *docente*.

Si assume altresì che sia stato costruito un indice basato su struttura B-tree per consentire la valutazione efficiente di interrogazioni del tipo:

```
SELECT descrizione  
FROM Corso  
WHERE nome=X;
```

dove  $X$  è un parametro fornito in ingresso.

Supponendo che il B-tree sia composto da tre livelli, ricavare il numero di tuple indicizzate ed il numero di pagine del B-tree nel caso in cui:

- 1) La radice ha riempimento massimo;
- 2) Il 50% dei nodi del secondo livello hanno riempimento minimo, e il restante 50% hanno riempimento massimo;
- 3) I nodi all'ultimo livello sono tutti pieni al 75%.

A

**Corso di Laurea in Ingegneria Informatica (DM 270)**  
**Esame di Basi di Dati**  
**22 febbraio 2019**

**Esercizio 4: Interrogazioni su un database relazionale (50 minuti)**

Sia dato il seguente schema di database:

- AGENZIA-TURISTICA(Codice, nome, Indirizzo, Città, Cap)
- PACCHETTO-VACANZA(CodPacchetto, Prezzo, Nome, Descrizione, AgenziaTuristica)
- CLIENTE(CodiceFiscale, Nome, Cognome, Città, Età)
- ACQUISTO(CodAcquisto, PacchettoVacanza, Cliente, Data)

con i seguenti vincoli di chiave esterna:

- PACCHETTO-VACANZA[AgenziaTuristica]  $\sqsubseteq_{FK}$  AGENZIA-TURISTICA[Codice];
- ACQUISTO [PacchettoVacanza]  $\sqsubseteq_{FK}$  PACCHETTO-VACANZA[CodPacchetto];
- ACQUISTO[Cliente]  $\sqsubseteq_{FK}$  CLIENTE[CodiceFiscale];

Lo studente formuli in SQL e, dove possibile, in Algebra Relazionale le seguenti interrogazioni:

- 1) Le agenzie turistiche della città di Roma o di Milano che vendono almeno due pacchetti vacanza, ciascuno dei quali è stato acquistato dopo l'anno 2014 da almeno due clienti distinti di età compresa tra 20 e 35 anni.
- 2) Il nome delle agenzie turistiche situate nella città di Salerno che non hanno mai venduto un pacchetto vacanza ad un cliente della città di Napoli o Avellino.
- 3) Le coppie di clienti <C1, C2> di età superiore ai 35 anni tali che l'insieme dei pacchetti vacanza acquistati da C1 nell'anno 2018 è lo stesso dell'insieme dei pacchetti vacanza acquistati da C2 nello stesso anno.
- 4) Per ogni città i cui clienti hanno un'età media inferiore a 40 anni, restituire i clienti che nell'anno 2017 hanno speso di più (rispetto ai clienti della stessa città) considerando tutti i loro acquisti di pacchetti vacanza di nome "Mini-crociera".

**Esercizio 5: Interrogazioni su un database XML (20 minuti)**

Si consideri un documento "Workshop.xml" conforme al seguente DTD:

```
<!ELEMENT Workshop(Sessione+)
<!ELEMENT Sessione (Sala, Presentazione+, Organizzatore+)
<!ELEMENT Presentazione (Nome, Autore+)
<!ELEMENT Organizzatore (Nome)
<!ELEMENT Autore (Codice, Nome)
<!ELEMENT Sala (#PCDATA)
<!ELEMENT Codice (#PCDATA)
<!ELEMENT Nome (#PCDATA)>
```

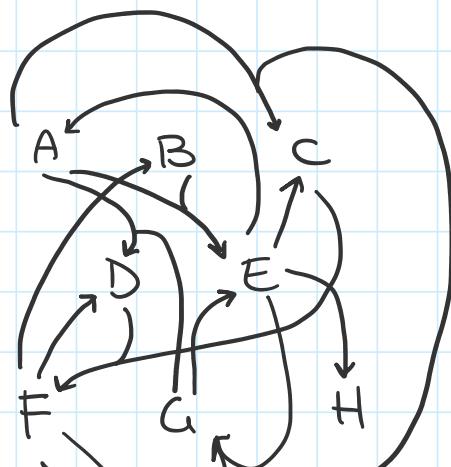
Supponendo che organizzatori omonimi siano la stessa persona, si formuli in XQuery la seguente interrogazione su "Workshop.xml": Restituire un documento XML contenente, per ogni sessione, la lista delle presentazioni che la compongono (assieme alla cardinalità della lista), e, per ciascuna presentazione, il numero dei suoi autori. In particolare, il documento XML restituito deve essere conforme al seguente DTD:

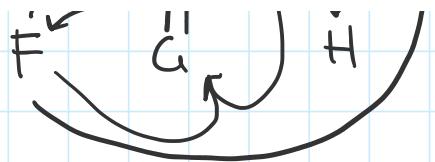
```
<!ELEMENT Risultato (Sessione+)
<!ELEMENT Sessione (Sala, ListaPresentazioni)
<!ELEMENT ListaPresentazioni(Presentazione+)
<!ATTLIST ListaPresentazioni numPresentazioni #CDATA>
<!ELEMENT Presentazione(Nome, NumeroAutori)
<!ELEMENT Nome (#PCDATA)
<!ELEMENT NumeroAutori (#PCDATA)>
```

2) 21:00

$$\left\{ \begin{array}{l} AB \rightarrow E \\ CD \rightarrow F \\ E \rightarrow ACH \\ EF \rightarrow G \\ F \rightarrow BD \\ G \rightarrow E \\ AG \rightarrow D \\ AF \rightarrow C \end{array} \right.$$

A B C D E F G H





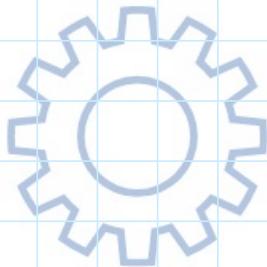
$$E^+ = \{ A \quad C \quad E \quad H \} \rightarrow \text{NO}$$

$$F^+ = \{ B \quad D \quad F \} \rightarrow \text{NO}$$

$$G^+ = \{ A \quad B \quad C \quad D \quad E \quad F \quad G \quad H \} \rightarrow \text{chiave}$$

Dunque in AG è ridondante la A:

$$\left\{ \begin{array}{l} AB \rightarrow E \\ CD \rightarrow F \\ E \rightarrow ACH \\ EF \rightarrow G \\ F \rightarrow BD \\ G \rightarrow E \\ G \rightarrow D \\ AF \rightarrow C \end{array} \right.$$



$$AB^+ = \{ A \quad B \quad C \quad E \quad H \} \rightarrow \text{NO}$$

**APPUNTI DI INGEGNERIA  
INFORMATICA  
GAIA BERTOLINO**

Dunque ora è in  $BNF$

$$CD^+ = \{ B \quad C \quad D \quad F \} \rightarrow \text{NO}$$

$$EF^+ = \{ A \quad B \quad C \quad D \quad E \quad F \quad G \quad H \} \rightarrow \text{chiave}$$

$$AF^+ = \{ A \quad B \quad C \quad D \quad E \quad F \quad G \quad H \} \rightarrow \text{chiave}$$

$$ABC^+ = \{ A \quad B \quad C \quad D \quad E \quad F \quad G \quad H \} \rightarrow \text{chiave}$$

$$ABD^+ = \{ A \quad B \quad C \quad D \quad E \quad F \quad G \quad H \} \rightarrow \text{chiave}$$

$$ABE^+ = \{ A B C \quad E \quad H \} \rightarrow \text{no}$$

$$ABF^+ = \{ A B C \quad D \quad E \quad F \quad G \quad H \} \rightarrow \text{closure}$$

$$ACD^+ = \{ A B C \quad D \quad E \quad F \quad G \quad H \} \rightarrow \text{closure}$$

$$ACE^+ = \{ A \quad C \quad E \quad H \} \rightarrow \text{no}$$

$$ACF^+ = \{ A B C \quad D \quad E \quad F \quad G \quad H \} \rightarrow \text{closure}$$

Closure candidate  $\{ G, EF, ABC, ABD, ABF, ACD, ACF \}$

$$x = CD$$

$$y = BCDF$$

$$z = AEGH$$

$$(BCDF)$$

$$(ACDEGH)$$

$$\left\{ \begin{array}{l} CD \rightarrow F \\ F \rightarrow BD \end{array} \right.$$

$$CD^+ = \{ CDFB \}$$

$$F^+ = \{ FBD \}$$

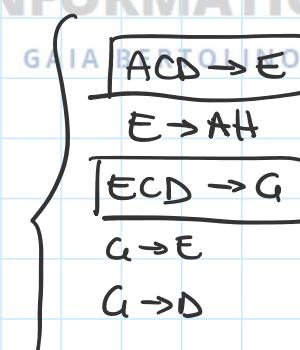
$$x = F$$

$$y = FBD$$

$$z = C$$

$$R_1 (FBD) \quad R_2 (FC)$$

$$\{ F \rightarrow BD \quad \{ FB \} \}$$



$$ACD^+ = \{ ACDEH \} \checkmark$$

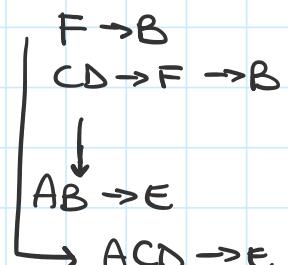
$$G^+ = \{ GEHD \} x$$

$$x = G$$

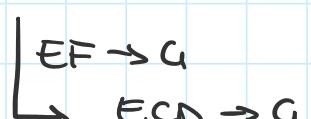
$$y = GEHD$$

$$z = C$$

$$\{ \quad \quad \quad \downarrow \quad \quad \quad \downarrow \}$$

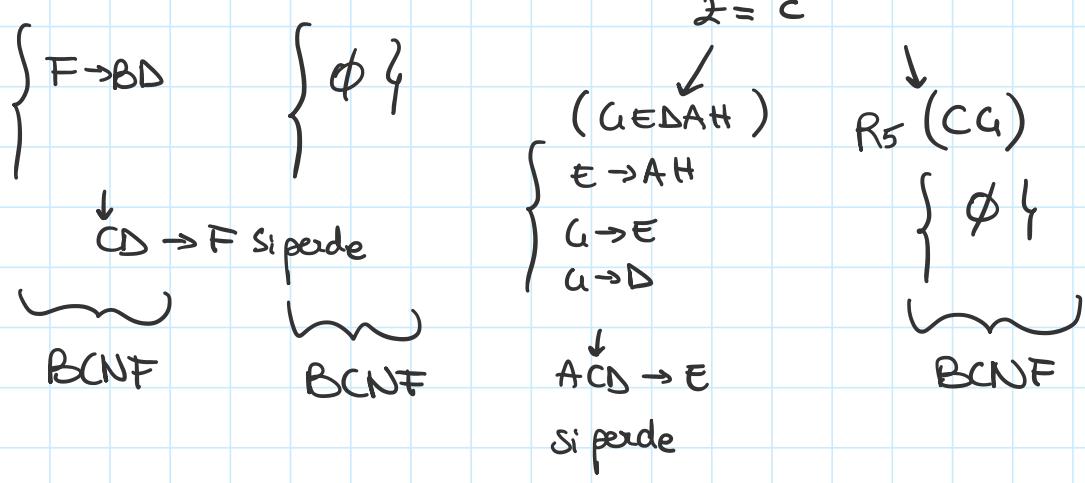


$E \rightarrow C$  si perde



$AF \rightarrow C$

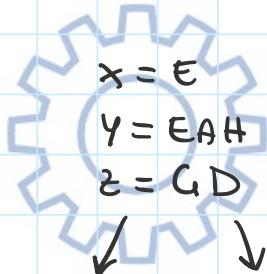
si perde



si perde

$$U^+ = \{ GE\Delta AH \}$$

$$E^+ = \{ EAH \}$$



APPUNTI DI INGEGNERIA  
INFORMATICA  
GAIA BERTOLINO

$$\left\{ \begin{array}{l} R_3(EAH) \\ E \rightarrow AH \end{array} \right\}$$

$$R_4(EGD)$$

$$\left\{ \begin{array}{l} E^+ = \{ EAH \} \\ \text{BCNF} \end{array} \right\}$$

$$\left\{ \begin{array}{l} U \rightarrow E \\ U \rightarrow D \end{array} \right\}$$

$$\left\{ \begin{array}{l} U^+ = \{ EGD \} \\ \text{BCNF} \end{array} \right\}$$

$$R_1(FBD) = \left\{ \begin{array}{l} F \rightarrow BD \\ \phi \end{array} \right\}$$

$\uparrow$  parte di chiavi

$$R_2(FC) = \{ \phi \}$$

$$R_3(EAH) = \{ E \rightarrow AH \}$$

$$R_4(EGD) = \{ G \rightarrow E, G \rightarrow D \}$$

$\uparrow$  parti di chiavi:  
 $G \rightarrow E, G \rightarrow D$

$$R_4(EAD) = \{ A \rightarrow E, C \rightarrow D \} \quad \left\{ \begin{array}{l} A \rightarrow E, C \rightarrow D \\ \text{parte di chiavi} \end{array} \right.$$

$$R_5(CA) = \{ \emptyset \}$$

Si perdono:

$$AB \rightarrow E, CD \rightarrow F, E \rightarrow C, EF \rightarrow G, AC \rightarrow D, AF \rightarrow C$$

$$R_6(ABE) = \{ AB \rightarrow E \} \quad \left\{ \begin{array}{l} AB \rightarrow E, CD \rightarrow F \\ \text{parti di chiavi} \end{array} \right.$$

$$R_7(CDF) = \{ CD \rightarrow F \}$$

$$R_8(EC) = \{ E \rightarrow C \} \quad \left\{ \begin{array}{l} E \rightarrow C, EF \rightarrow G \\ \text{parte di chiavi} \end{array} \right.$$

$$R_9(EFG) = \{ EF \rightarrow G \}$$

$$R_{10}(AGD) = \{ AG \rightarrow D \} \quad \left\{ \begin{array}{l} AG \rightarrow D, AF \rightarrow C \\ \text{parti di chiavi} \end{array} \right.$$

$$R_{11}(AFC) = \{ AF \rightarrow C \}$$

APPUNTI DI INGEGNERIA

INFORMATICA

GAIA BERTOLINO

$$R_{12}(CBDF) = \{ F \rightarrow BD \}$$

$$R_3(EAH) = \{ E \rightarrow AH \}$$

$$R_{14}(EAD) = \{ A \rightarrow E, C \rightarrow D \}$$

$$R_{13}(EABCDF) = \{ AB \rightarrow E, CD \rightarrow F \}$$

$$R_{14}(ECFG) = \{ E \rightarrow C, EF \rightarrow G \}$$

$$R_{15}(ACDF) = \{ AC \rightarrow D, AF \rightarrow C \}$$

17 febbraio 2011

mercoledì 2 febbraio 2022 12:01



17 febbraio 2011

**Facoltà di Ingegneria – Corso di Laurea in Ingegneria Informatica (DM 270)**  
**Esame di Basi di Dati**  
**17 febbraio 2011**

**Esercizio 1: Progettazione concettuale e logica di una Base di Dati (40 minuti)**

- a) Si progetti lo schema Entità-Relazione di una base di dati di supporto alla gestione di pratiche di contabilità presso gli uffici della regione Calabria.

La gestione delle pratiche coinvolge diverse categorie di persone: i *direttori* d'ufficio, i *contabili*, i *supervisori* e i *consulenti esterni*. Ogni persona è caratterizzata dal proprio codice fiscale (che è identificativo della persona stessa), dal nome, dalla data di nascita, e dal titolo di studio. Una persona riveste esattamente uno dei ruoli sopra elencati. I direttori sono anche caratterizzati da un'anzianità di servizio (espressa in anni), ed i contabili da una qualifica. I titoli di studio sono caratterizzati da un codice (identificativo) e da un grado (formazione primaria, secondaria, universitaria). Nell'indicare il titolo di studio in possesso di ogni persona, deve essere possibile specificare l'Istituto che ha rilasciato tale titolo.

Per ogni pratica gestita occorre tenere traccia del numero ad essa assegnato, dell'anno in cui essa è stata avviata, di una descrizione, e del numero di scaffale in cui è contenuto il raccolto dei documenti ad essa associata. Possono esistere pratiche con lo stesso numero, ma devono essere state avviate in anni diversi.

Ciascuna pratica è affidata ad un unico ufficio, che è caratterizzato da un nome (identificativo) e dall'indicazione dell'edificio in cui esso è ubicato. Un ufficio può essere affidatario di un numero qualsiasi di pratiche.

Per ogni ufficio occorre tenere traccia dell'indicazione del suo (unico) direttore, e della lista dei contabili che operano al suo interno. Si noti che i direttori e i contabili possono afferire ad un numero qualsiasi di uffici.

Le pratiche sono soggette a verifiche da parte dei supervisori. Per ogni pratica va memorizzata la lista dei supervisori a cui è stata affidata, e per ogni revisore di tale lista deve essere possibile indicare la data di inizio e di fine dell'incarico di supervisione (si assuma che un supervisore può ricevere al più un incarico di verifica per una stessa pratica). Si noti che i supervisori, a differenza dei direttori e dei contabili, non hanno un'afferenza agli uffici.

Per ogni pratica, occorre tenere traccia delle consulenze che si sono rese necessarie per il suo espletamento. Ciascuna consulenza è caratterizzata dall'unico consulente esterno che l'ha redatta, dalla pratica a cui si riferisce, dalla data in cui è stata depositata, e da una descrizione. Si noti che non possono esistere più consulenze depositate per una stessa pratica ed in una stessa data.

Infine, si deve memorizzare l'elenco dei documenti che compongono ogni pratica. Ciascun documento è caratterizzato da un numero progressivo (che identifica il documento in coppia con la pratica in cui è inserito) e da una descrizione.

- b) Si definisca uno schema relazionale corrispondente allo schema E-R.

**Esercizio 2: Progettazione logica e dipendenze funzionali (30 minuti)**

Si consideri lo schema relazionale  $\langle R(A,B,C,D,E,F,G,H), \delta \rangle$ , dove  $\delta$  è il seguente insieme di dipendenze funzionali:

$$\begin{array}{l} A \rightarrow CD \\ BCD \rightarrow E \\ ABD \rightarrow H \\ E \rightarrow B \\ D \rightarrow F \\ EF \rightarrow H \\ H \rightarrow ABG \\ F \rightarrow G \end{array}$$

GNERIA  
CA

Definire:

1. una copertura minimale per  $\beta$ ;
2. le chiavi candidate di R;
3. una decomposizione in BCNF di R che sia senza perdita di informazione;
4. una decomposizione in 3NF di R che sia senza perdita di informazione, senza perdita di dipendenze funzionali e di dimensione minima.

**Esercizio 3: Indici (20 minuti)**

Si consideri un sistema con le seguenti caratteristiche:

- i. pagine fisiche di 4 KB;
- ii. intestazione fissa delle pagine fisiche di 128 byte;
- iii. puntatori di 4 byte.

Si consideri la relazione: *Ufficio* (nome, edificio, direttore)

e si assuma che il numero di byte destinato alla rappresentazione di ciascun attributo sia:

- a) 64 byte per l'attributo *nome*;
- b) 4 byte per l'attributo *edificio*;
- c) 16 byte per l'attributo *direttore*.

Si assume altresì che sia stato costruito un indice basato su struttura B-tree per consentire la valutazione efficiente di interrogazioni del tipo:

```
SELECT edificio, direttore  
FROM Ufficio  
WHERE nome=X;
```

dove *X* è un parametro fornito in ingresso.

Sapendo che il fattore di caricamento è pari al 75% e che il B-tree occupa 75 895 pagine, stimare l'altezza del B-tree ed il numero di tuple in esso indicizzate.

GNERIA  
INFORMATICA  
GAIA BERTOLINO

**Esercizio 4: Interrogazioni su un database relazionale (60 minuti)**

Sia dato il seguente schema relazionale:

- Ufficio ( nome, edificio, città)
- Pratica ( numero, anno, ufficio)
- Documento(numero,pratica)
- Supervisore ( cf, nome, dataNascita, città)
- Verifica (supervisore, pratica)

Si considerino i seguenti vincoli di chiave esterna sullo schema relazionale:

Pratica[ufficio] ⊆<sub>FK</sub> Ufficio[nome];

Documento[pratica] ⊆<sub>FK</sub> Pratica[numero];

Verifica[supervisore] ⊆<sub>FK</sub> Supervisore[cf];

Verifica[pratica] ⊆<sub>FK</sub> Pratica[numero].

Lo studente formuli in SQL e, dove possibile, in Algebra Relazionale le seguenti interrogazioni:

- Le città dei supervisori che hanno effettuato verifiche di almeno due pratiche diverse del 2010 gestite da uffici di Cosenza.
- Il nome dei supervisori che non hanno mai effettuato verifiche su pratiche antecedenti al 1999 gestite da uffici di Roma.
- Le pratiche che sono state supervisionate da tutti i supervisori di Milano.
- Per ogni supervisore che ha effettuato più di 10 verifiche, restituire la pratica avente il maggior numero di documenti tra quelle su cui il supervisore ha effettuato una verifica (NB: le tuple del risultato rappresentano coppie <supervisore, pratica> )

**Esercizio 5: Interrogazioni su un database XML (20 minuti)**

Si consideri il seguente DTD:

```
<!ELEMENT libri (genere+)>
<!ELEMENT genere (nome, libro+)>
<!ELEMENT libro (nome, autore*)>
<!ELEMENT autore (#PCDATA)>
<!ELEMENT nome (#PCDATA)>
<!ATTLIST libro anno CDATA
codice ID
cita IDREF >
```

Si assume che nei documenti che rispettano tale DTD, il valore dell'attributo *cita* sia il codice di un elemento libro

Si formuli in XQuery la seguente interrogazione sul documento "libri.xml" conforme a tale DTD:

Per ogni autore, restituire ogni suo libro con la lista dei libri che lo citano. In particolare, il documento XML restituito deve essere conforme al seguente DTD:

```
<!ELEMENT risultato (autore+)>
<!ELEMENT autore (nome, libro+)>
<!ELEMENT libro (nome, genere, citatoDa*)>
<!ELEMENT citatoDa (nome)>
<!ELEMENT genere (nome)>
<!ELEMENT nome (#PCDATA)>
```

Non sono presenti elementi *citatoDa* se il libro non è citato da alcun libro

GNERIA

INFORMATICA

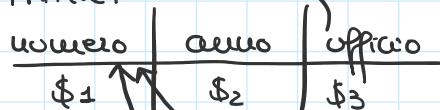
GAIA BERTOLINO

4) 60 minuti → 12:25

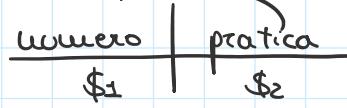
UFFICIO



PRATICA



DOCUMENTO



SUPERVISORE



cf	nome	dataNascita	città
\$1	\$2	\$3	\$4
VERIFICA			
Supervisore   pratica			
\$1   \$2			

### 1) Città di supervisori

verifiche di due pratiche diverse che risalgono al 2010 e gestite da un ufficio di Cesena

$$P = \prod_{\$1} [ \underset{\substack{6\$6='CS' \\ \$2=\$1}}{\wedge} (\text{Pratica} \bowtie \text{Ufficio}) ] \rightarrow \text{numero delle pratiche}$$

$\$2 = 2010$

$$\text{Almeno2} = \prod_{\$1} [ ( \text{Verifica} \bowtie P ) \bowtie ( \text{Verifica} \bowtie P ) ] \rightarrow \text{uomini supervisori con due pratiche}$$

$\begin{array}{c} \$2=\$1 \\ \$1=\$1 \\ \$2=\$1 \\ \wedge \\ \$2 \neq \$2 \end{array}$

$$\text{Query} = \prod_{\$4} [ \text{Almeno2} \bowtie \text{Supervisore} ]$$

$\begin{array}{c} \$1=\$1 \end{array}$

CREATE VIEW Filtrate AS  
 (SELECT Pratica.numero  
 FROM Pratica, Ufficio  
 WHERE Pratica.ufficio = Ufficio.nome  
 AND Ufficio.citta = 'CS'  
 AND Pratica.anno = 2010);

SELECT Supervisore.citta  
 FROM Filtrate AS F1, Filtrate AS F2, Supervisore, Verifica AS V1, Verifica AS V2  
 WHERE V1.supervisore = V2.supervisore  
 AND V1.pratica ≠ V2.pratica  
 AND V2.pratica = F1.numero  
 AND V2.pratica = F2.numero  
 AND Supervisore.cf = V1.supervisore

### 2) Nome supervisori - Supervisori che hanno fatto pratiche prima del 1999 in ufficio di Roma

$$\text{SUPNO} = \prod_{\$2} \left\{ \begin{array}{l} 6 \\ \$12='ROM' \end{array} \right\} \left\{ \begin{array}{l} 6 \\ \$8 < 1999 \end{array} \right\} \left[ ( \text{Supervisore} \bowtie \text{Verifica} ) \bowtie \text{pratica} \right] \left\{ \begin{array}{l} \bowtie \text{Ufficio} \\ \$9=\$1 \end{array} \right\}$$

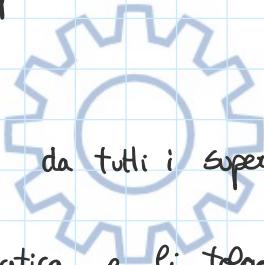
$SUPNO = \Pi_{S_2} \left\{ \begin{array}{l} S_1 \\ S_2 = 'Roma' \end{array} \right\} \left\{ \begin{array}{l} S_6 < 1999 \\ S_1 = S_1 \end{array} \right\} (Supervisori \bowtie Verifica) \bowtie Pratica ] \left\{ \begin{array}{l} \bowtie Ufficio \\ S_9 = S_1 \end{array} \right\}$

Query =  $\Pi_{S_2} (\text{Supervisori}) - SUPNO$

```

SELECT Supervisori.nome
FROM Supervisori
EXCEPT
SELECT Supervisori.nome
FROM Supervisori, Verifica, Pratica, Ufficio
WHERE Supervisore.cf = Verifica.supervisore
AND Verifica.pratica = Pratica.numero
AND Pratica.ufficio = Ufficio.nome
AND Ufficio.citta = 'Roma'
AND Pratica.anno < 1999

```



3) Codice pratiche supervisionate da tutti i supervisori di Milano

=

Prendo i supervisori di una pratica e li tolgo ai supervisori di milano. Se ne rimane qualcuno allora è da escludere

## INFORMATICA

$SUPMilano = \Pi_{S_2} [ \left\{ \begin{array}{l} S_1 \\ S_2 = 'Milano' \end{array} \right\} (\text{Supervisori}) ]$

```

SELECT
FROM
WHERE

```

P <sub>1</sub>	
S <sub>1</sub>	C <sub>1</sub>
S <sub>2</sub>	C <sub>2</sub>

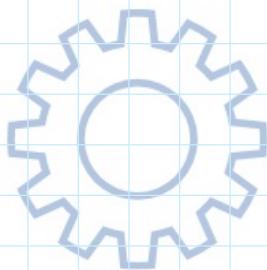
```

CREATE VIEW Filtro AS
(SELECT Verifica.pratica
FROM Verifica, Supervisore
WHERE Supervisore.cf = Verifica.supervisore
AND Supervisore.citta = 'Milano'
GROUP BY Verifica.pratica)

```

P <sub>2</sub>	
S <sub>3</sub>	C <sub>3</sub>
S <sub>4</sub>	C <sub>4</sub>
S <sub>5</sub>	C <sub>5</sub>

```
CREATE VIEW SopMilano AS  
(SELECT Supervisore.cf  
FROM Supervisore  
WHERE Supervisore.citta = 'Milano');
```



# APPUNTI DI INGEGNERIA INFORMATICA

GAIA BERTOLINO

13:16

## CONFERENZA

codice	uomo	città	dataInizio	durata	inizio Registratione
\$1	\$2	\$3	\$4	\$5	\$6

## PERSONA

CF	uomo	copiuomo	data-uoscila	città
\$1	\$2	\$3	\$4	\$5

## REGISTRAZIONE



## RELATORE

persona	conference	giorno	ora	durataPresentazione
\$1	\$2	\$3	\$4	\$5

- 1) CF delle persone registrate in due conferenze almeno e tali conferenze hanno almeno due relatori diversi da P ma della stessa città di P

$$2\text{conf} = \left( \begin{array}{l} \text{Reg} \bowtie \text{Reg} \\ \$1=\$1 \\ \$2 \neq \$2 \end{array} \right)$$

$$\text{Query} = \text{TBS} \left[ \left[ \left[ \left[ \begin{array}{l} 2\text{conf} \bowtie (\text{Relatore} \bowtie \text{Registration}) \\ \$1 \neq \$1 \\ \$2 = \$2 \end{array} \right] \bowtie (\text{Relatore} \bowtie \text{Registration}) \right] \right] \right] \bowtie \text{Persona} \left\{ \begin{array}{l} \text{Reg} \\ \$1=\$1 \\ \$2 = \$2 \end{array} \right\}$$
$$\left\{ \begin{array}{l} \text{Relatore} \\ \$1 \neq \$1 \\ \$2 = \$2 \end{array} \right\} \bowtie \text{Persona} \left\{ \begin{array}{l} \text{Registration} \\ \$1=\$1 \\ \$2 = \$2 \end{array} \right\}$$

$$\left\{ \begin{array}{l} \text{Persona} \\ \$5=\$5 \\ \$12=\$1 \end{array} \right\}$$

$$\wedge$$

$$\$23 = \$28$$

$$\wedge$$

$$\$23 = \$33$$

APPUNTI DI INGEGNERIA  
INFORMATICA  
GAIA BERTOLINO

08 gen 2020

mercoledì 2 febbraio 2022

18:28



08 gen  
2020

**Corso di Laurea in Ingegneria Informatica (DM 270)**  
**Esame di Basi di Dati**  
**8 gennaio 2020**

**Esercizio 4: Interrogazioni su un database relazionale (50 minuti)**

Sia dato il seguente schema di database:

COMPAGNIA (Codice, Nome, Indirizzo, Città, CAP)  
POLIZZA(CodPolizza, Prezzo, Nome, Tipologia, CompagniaAssicurativa)  
CLIENTE(CodiceFiscale, Nome, Cognome, Città, Età)  
ACQUISTO(CodAcquisto, Polizza, Cliente, Data)

con i seguenti vincoli di chiave esterna:

POLIZZA[CompagniaAssicurativa]  $\sqsubseteq_{FK}$  COMPAGNIA [Codice];  
ACQUISTO [PacchettoPolizza]  $\sqsubseteq_{FK}$  POLIZZA[CodPacchetto];  
ACQUISTO[Cliente]  $\sqsubseteq_{FK}$  CLIENTE[CodiceFiscale];

Lo studente formuli in SQL e, dove possibile, in Algebra Relazionale le seguenti interrogazioni:

1. Le compagnie assicurative della città di Torino o di Trieste che vendono almeno due polizze, ciascuna delle quali è stata acquistata tra gli anni 2011 e 2018 da almeno due clienti distinti di età superiore a 40 anni.
2. Il nome delle compagnie assicurative situate nella città di Roma le cui polizze non sono mai state vendute a clienti di Milano.
3. Le coppie di clienti  $\langle C1, C2 \rangle$  di età inferiore ai 28 anni tali che l'insieme delle polizze acquistate da C1 dall'anno 2018 al 2019 è lo stesso dell'insieme delle polizze acquistate da C2 nello stesso intervallo.
4. Per ogni città i cui clienti hanno un'età minima superiore a 45 anni, restituire i clienti che nell'anno 2013 hanno speso di più (rispetto ai clienti della stessa città) considerando tutti i loro acquisti di pacchetti polizza di tipologia "assicurazione vita".

**Corso di Laurea in Ingegneria Informatica (DM 270)**  
**Esame di Basi di Dati**  
**7 febbraio 2019**

**Esercizio 4: Interrogazioni su un database relazionale (50 minuti)**

Sia dato il seguente schema di database:

*Tragitto* (codice, portoPartenza, portoDestinazione, distanza)  
*Nave* (codice, tragittoCoperto, tipologiaNave, data, durata)  
*Porto* (codice, nome, città, nazione)

con i seguenti vincoli di chiave esterna:

*Tragitto*[portoPartenza]  $\subseteq_{FK}$  *Porto*[codice];  
*Tragitto*[portoDestinazione]  $\subseteq_{FK}$  *Porto*[codice];  
*Nave*[tragittoCoperto]  $\subseteq_{FK}$  *Tragitto*[codice].

Lo studente formuli in SQL e, dove possibile, in Algebra Relazionale le seguenti interrogazioni:

1. I nomi dei porti che sono il punto di partenza di almeno due tragitti, ciascuno dei quali è coperto da almeno due navi di uguale tipologia
2. Le tipologie navali delle quali non esiste alcuna nave che copre un tragitto che parte dalla città di Genova e che copre una distanza superiore a 100 km.
3. Le coppie di porti tali che gli insiemi di tipologie di navi che partono da ciascuno di essi sono uguali.
4. Per ogni città in cui esiste un numero di porti superiore alla città di Salerno, restituire ogni tipologia di nave che, sommando i tragitti percorsi dalle navi di tale tipologia, ricopre la massima distanza cumulativa.

**CORSO DI LAUREA IN INGEGNERIA INFORMATICA (DM 270)**  
**Esame di Basi di Dati**  
**6 marzo 2018**

**Esercizio 4: Interrogazioni su un database relazionale (50 minuti)**

Sia dato il seguente schema di database:

- CONCESSIONARIA(Codice, nome, Indirizzo, Città, Cap, Nazione)
- AUTOMOBILE(CodAutomobile, Prezzo, Nome, Marca, Concessionaria)
- GUIDATORE(CodiceFiscale, Nome, Cognome, Città, DataNascita, Cittadinanza)
- ACQUISTO(CodAcquisto, Automobile, Guidatore, Anno)

con i seguenti vincoli di chiave esterna:

- AUTOMOBILE [Concessionaria]  $\subseteq_{FK}$  CONCESSIONARIA[Codice];
- ACQUISTO [Automobile]  $\subseteq_{FK}$  AUTOMOBILE[CodAutomobile];
- ACQUISTO[Guidatore]  $\subseteq_{FK}$  GUIDATORE[CodiceFiscale];

Lo studente formuli in SQL e, dove possibile, in Algebra Relazionale le seguenti interrogazioni:

- 1) Le concessionarie della città di Torino o Napoli che hanno venduto almeno due automobili, ciascuna delle quali è stata acquistata da almeno due guidatori distinti nati tra il 1984 e il 1988.
- 2) Il nome delle concessionarie situate in Germania che non hanno mai venduto un'automobile ad un guidatore nato nel 1998.
- 3) Le coppie di guidatori  $\langle G1, G2 \rangle$  di città diverse tali che l'insieme delle marche delle automobili acquistate da G1 è lo stesso di quello di G2.
- 4) Per ogni città, restituire i guidatori che tra il 1998 e il 2015 hanno acquistato il numero massimo di automobili di marca Fiat rispetto ai guidatori della stessa città.

**Corso di Laurea in Ingegneria Informatica (DM 270)**  
**Esame di Basi di Dati**  
**28 aprile 2017**

**Esercizio 4: Interrogazioni su un database relazionale (50 minuti)**

Sia dato il seguente schema di database:

- AUTO (Targa, Marca, Cilindrata, Potenza, Proprietario, Assicurazione);
- PROPRIETARIO (CF, Nome, Cognome, Residenza)
- ASSICURAZIONE (Codice, Nome, Sede)
- SINISTRO (Codice, Località, Data)
- AUTOINVOLTE (Sinistro, Auto, ImportoDelDanno)

con i seguenti vincoli di chiave esterna:

- AUTOINVOLTE[Sinistro]  $\subseteq_{FK}$  SINISTRO[Codice];
- AUTOINVOLTE [Auto]  $\subseteq_{FK}$  AUTO[Targa];
- AUTO[Assicurazione]  $\subseteq_{FK}$  ASSICURAZIONE[Codice];
- AUTO[Proprietario]  $\subseteq_{FK}$  PROPRIETARIO[CF].

Lo studente formuli in SQL e, dove possibile, in Algebra Relazionale le seguenti interrogazioni:

- 1) Le auto di marca "Fiat" che sono state coinvolte in almeno due sinistri, in ciascuno dei quali sono state coinvolte almeno due auto di cilindrata superiore a 1600cc .
- 2) Il nome ed il cognome dei proprietari di un'auto di marca "Suzuki" che non è coperta da una polizza contratta con un'assicurazione di Roma o Milano e che non è stata mai coinvolta in alcun sinistro.
- 3) Le coppie di proprietari <P1, P2> tali che l'insieme delle città in cui si è verificato almeno un sinistro in cui è stata coinvolta un'auto di P1 è lo stesso dell'insieme delle città in cui si è verificato almeno un sinistro in cui è stata coinvolta un'auto di P2.
- 4) Tra i sinistri avvenuti nel 2010, restituire quelli per cui è massimo il numero di auto coinvolte con cilindrata superiore a 1000 e la cui assicurazione ha sede a Torino o a Milano.



08 gen  
2020 1

**Corso di Laurea in Ingegneria Informatica (DM 270)**  
**Esame di Basi di Dati**  
**8 gennaio 2020**

**Esercizio 1: Progettazione concettuale e logica di una Base di Dati (40 minuti)**

a) Si progetti una base di dati di supporto ad un negozio di cucine componibili, in cui venga tenuta traccia dei clienti, dei modelli di cucine in vendita, e dei progetti di cucine elaborati per i clienti.

I *modelli* di cucina in vendita sono caratterizzati dal nome, dalla marca, da una descrizione dei materiali con cui la cucina è realizzata, e dalla lista dei *pezzi* che è possibile scegliere nel comporre una cucina del modello. Possono esistere modelli con lo stesso nome di marche diverse, ma non modelli omonimi della stessa marca. Ogni pezzo si riferisce ad un unico modello, ed è caratterizzato dal nome, dalla tipologia, e dalle dimensioni. Esistono pezzi diversi con lo stesso nome, ma in tale caso devono riferirsi a modelli di cucina diversi. L'insieme dei pezzi è partizionato in tre categorie: *colonne*, *basi* e *pensili*. Per ogni colonna devono essere riportate le dimensioni minime e massime del frigorifero in essa installabile, e per ogni base deve essere riportata la lista degli *elettrodomestici* da incasso in essa installabili. Ogni elettrodomestico è identificato dal codice (identificativo), dalla marca e dalla tipologia, ed è installabile in un numero qualunque di basi.

I *clienti* sono identificati dal codice fiscale e sono anche caratterizzati dal nome, da un recapito, e dalla lista dei *progetti* di cucine da essi richiesti. Ogni progetto, oltre che dal cliente richiedente, è caratterizzato dalla data in cui è stato realizzato, da una descrizione, e da una lista di *composizioni lineari*, ciascuna riguardante un lato della cucina. Uno stesso cliente può richiedere più progetti, ma i progetti di uno stesso cliente devono differire nella data di richiesta.

Ciascuna composizione lineare riguarda un unico progetto ed è caratterizzata da un numero progressivo che la identifica rispetto alle composizioni dello stesso progetto. Ciascuna composizione è caratterizzata da una descrizione ed appartiene ad esattamente una delle seguenti classi: composizione *base* e composizione *pensile*. Per ciascuna delle sottoclassi, occorre indicare la lista di pezzi da cui essa è composta, secondo quanto indicato nel seguente. Una composizione base può contenere un numero qualunque di pezzi di tipo "colonna" ed un numero qualunque di pezzi di tipo "base", mentre una composizione pensile può contenere un numero qualunque di pezzi di tipo "pensile". Nell'indicare l'appartenenza di un pezzo di tipo "colonna", di tipo "base" e di tipo "pensile" ad una composizione (base o pensile, a seconda dei casi) occorre indicare il colore del pezzo. Si noti che uno stesso pezzo può essere presente in un numero qualunque di composizioni.

Ogni nuovo progetto può essere la revisione di un progetto già esistente, per cui occorre tenere traccia di tale relazione tra i progetti: un progetto **può** discendere da **al più un** progetto, e da esso discende un numero qualunque di altri progetti.

Infine, per ogni progetto per cui si è concretizzato l'acquisto, deve essere indicato il *pagamento*. Un pagamento si riferisce ad un unico progetto (che identifica il pagamento stesso) ed è anche caratterizzato dalla data in cui è stato effettuato e dalle modalità. Si osservi che non per tutti i progetti esiste l'acquisto corrispondente.

## Corso di Laurea in Ingegneria Informatica (DM 270)

### Esame di Basi di Dati

7 febbraio 2019

#### Esercizio 1: Progettazione concettuale e logica di una Base di Dati (40 minuti)

a) Si progetti una base di dati di supporto alla gestione di un museo, nelle cui sale vengono ospitate esposizioni permanenti e mostre temporanee di opere d'arte.

Il museo è composto da diverse *sedi*, identificate da un codice ed anche caratterizzate dal nome e dall'indirizzo. Ogni sede contiene più sale, ciascuna delle quali è caratterizzata da un numero, dal nome e dall'area, oltre che dalla sede in cui è ubicata. Il numero di una sala la identifica rispetto alle sale presenti nella stessa sede. Inoltre, per ogni sala deve essere possibile indicare la lista delle sale accessibili da essa (a tale proposito, si noti che non c'è limite sul numero di sale a cui si può accedere da una stessa sala, né sul numero di sale da cui si può accedere alla stessa sala).

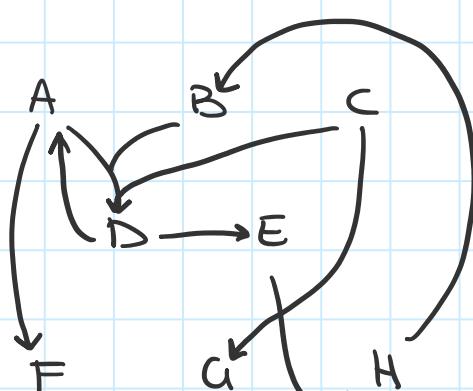
Ciascuna *opera* d'arte esposta è caratterizzata da un codice, dal nome, e dall'artista che l'ha realizzata. L'insieme delle opere d'arte è partitionato in due classi: le opere di *proprietà* e quelle in *prestito*. Per ogni opera di proprietà occorre tenere traccia della lista delle *dislocazioni* che l'opera ha avuto nel tempo. Ogni dislocazione è caratterizzata dall'opera a cui si riferisce, dalla data a partire dalla quale ha validità e dalla sala in cui viene dislocata l'opera. Non possono esistere più dislocazioni della stessa opera aventi la stessa data di decorrenza. Inoltre, in una stessa sala possono essere dislocate un numero qualunque di opere di proprietà. Per quanto riguarda le opere in prestito, invece, ognuna di esse va associata alla *mostra* temporanea nell'ambito della quale essa viene esposta. Ogni mostra è caratterizzata dal nome (attributo identificante), dalle date di inizio e fine, e dalla lista delle sale per essa riservate. Si noti che in una stessa mostra temporanea possono essere esposte un numero qualunque di opere in prestito, e che non c'è alcun limite sul numero di mostre che possono essere allestite in una stessa sala.

Per ogni opera di proprietà può essere anche indicato l'*atto di acquisizione* dell'opera stessa. Ogni atto è caratterizzato da un numero progressivo, dalla data di stipula, e da una descrizione. Uno stesso atto di acquisizione può riferirsi a più opere di proprietà. Atti di acquisizione aventi la stessa data devono differire nel numero.

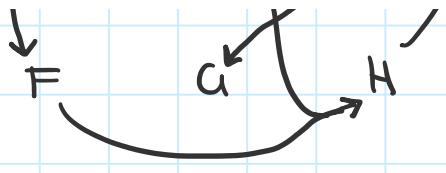
Infine, occorre tenere traccia della lista degli *interventi* di restauro delle opere svolti all'interno del museo (tali interventi possono riguardare le opere indipendentemente dal fatto di essere di proprietà o meno). Ogni intervento si riferisce ad un'unica opera, ed è caratterizzato dalla data di inizio, dalla durata, e dal restauratore che lo ha eseguito. Una stessa opera può subire un numero qualunque di interventi di restauro. Non possono esistere più interventi di restauro effettuati sulla stessa opera ed iniziati nella stessa data data.

18:30

$$\left\{ \begin{array}{l} ABC \rightarrow D \\ A \rightarrow F \\ EF \rightarrow H \\ H \rightarrow B \\ D \rightarrow AE \end{array} \right.$$



$$\begin{cases} H \rightarrow B \\ D \rightarrow AE \\ C \rightarrow G \end{cases}$$



$$A^+ = \{ A \}$$

$$F \}$$

$$B^+ = \{ B \}$$

$$C^+ = \{ C \}$$

$$ABC^+ = \{ A B C D E F G H \} \rightarrow \text{closure}$$

$$DC^+ = \{ A B C D E F G H \} \rightarrow \text{closure}$$

$$EF^+ = \{ B E F H \}$$

$$D^+ = \{ A B D E F H \}$$

$$x = A$$

$$y = AF$$

$$z = BCDEGH$$



$$(AF)$$

$$(BCDEGH)$$

$$\left\{ \begin{array}{l} A \rightarrow F \end{array} \right.$$

$$\left\{ \begin{array}{l} \end{array} \right.$$

$$ABC \rightarrow D$$

$$EF \rightarrow H$$

$$AE \rightarrow H$$

06 set 2019

mercoledì 2 febbraio 2022 19:13

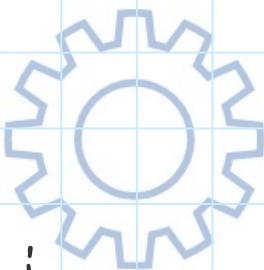
P Coglia = 4 KB

Interfaccia = 128 byte

Portatori = 4 byte

Giocatore (CF, nome, tipoAbbraccamento)  
16 byte

SELECT tipoAbbraccamento  
FROM giocatore  
WHERE CF = x;



Ricerca sulla chiave :

APPUNTI DI INGEGNERIA  
INFORMATICA  
GAIA BERTOLINO

**Corso di Laurea in Ingegneria Informatica (DM 270)**  
**Esame di Basi di Dati**  
**6 settembre 2019**

**Esercizio 3: Indici (20 minuti)**

Si consideri un sistema con le seguenti caratteristiche:

- iv. pagine fisiche di 4 KB;
- v. intestazione fissa delle pagine fisiche di 128 byte;
- vi. puntatori di 4 byte.

Si consideri la relazione:

*Giocatore (codiceFiscale, nome, tipoAbbonamento)*

e si assuma che il numero di byte destinato alla rappresentazione di ciascun attributo sia:

- d) 16 byte per l'attributo *codiceFiscale*;
- e) 32 byte per ciascuni degli attributi *nome* e *tipoAbbonamento*.

Si assuma altresì che sia stato costruito un indice basato su struttura B-tree per consentire la valutazione efficiente di interrogazioni del tipo:

```
SELECT tipoAbbonamento  
FROM Giocatore  
WHERE nome=X;
```

dove *X* è un parametro fornito in ingresso.

Supponendo che il B-tree sia composto da tre livelli, ricavare il numero di tuple indicizzate ed il numero di pagine del B-tree nel caso in cui il secondo livello sia costituito da 30 nodi, ciascuno dei quali ha riempimento minimo, mentre i nodi al terzo livello hanno tutti un fattore di riempimento pari al 90%.

**APPUNTI DI INGEGNERIA**

**INFORMATICA**

GAIA BENTOLINO

$$\begin{array}{cccc} 4096 & 128 & 4 \cdot u & (16+4)(u-1) \\ \underbrace{\quad}_{\text{Pag}} & \underbrace{\quad}_{\text{Iut}} & \underbrace{\quad}_{\text{p.u}} & \text{p.u} + (k+p)(u-1) \end{array}$$

• Oggi vedo cosa è una pagina

1) capire come è fatta una pagina

$$2988 > 24u \Rightarrow u < 166,17$$

Si associa all'intero più vicino

Si associa all' intero più vicino



$$\boxed{\lceil u = 166 \rceil}$$

Dunque A uodo il grado massimo è 166.

Il grado minimo sarà  $\lceil \frac{u}{2} \rceil = 83$

Il grado minimo della radice è 1 per definizione

2) Capire come sono definiti i singoli livelli



- Primo livello = 1 uodo con 29 tuple e 30 figli
- Secondo livello = 30 uodi a rimp. minimo  
 $(83 \text{ figli}) = (30 \times 83) - 1 \text{ tuple}$
- Terzo livello =  $30 \times 83 - (166 \cdot 0,9) - 1 \text{ tuple}$

APPUNTI DI INGEGNERIA  
INFORMATICA

GAIA BERTOLINO

06 mar 2018

mercoledì 2 febbraio 2022 19:40



06 mar  
2018

Corso di Laurea in Ingegneria Informatica (DM 270)  
Esame di Basi di Dati  
6 marzo 2018

Esercizio 1: Progettazione concettuale e logica di una Base di Dati (40 minuti)

a) Si progetti una base di dati di supporto ad un'azienda agricola specializzata nella produzione in serre.

Ciascuna delle *serre* in cui i prodotti dell'azienda vengono coltivati è caratterizzata dalla dimensione, dall'indicazione del *terreno* in cui è installata, e da un numero progressivo, che identifica la serra rispetto alle altre nello stesso terreno. I terreni sono caratterizzati dal comune in cui si trovano, dalle coordinate catastali, e dall'estensione. Non esistono terreni diversi siti nello stesso comune e che coincidano nelle coordinate catastali. Si noti che in un terreno possono essere situate diverse serre.

Nelle serre è installato un sistema di monitoraggio con il quale viene rilevata la temperatura al loro interno. In tale riguardo, deve essere possibile memorizzare la lista delle *rilevazioni* eseguite su ciascuna serra. Ogni rilevazione è caratterizzata, oltre che dalla serra, dalla data/ora in cui è avvenuta la rilevazione, e dal valore di temperatura rilevato. Non esistono rilevazioni contemporanee nella stessa serra.

Deve essere possibile memorizzare gli *interventi* effettuati sulle serre. In particolare, ogni intervento è identificato da un codice ed è anche caratterizzato dalla data in cui è stato effettuato e dal motivo che lo ha reso necessario. L'insieme degli interventi è partitionato nelle due seguenti classi: *interventi di installazione* ed *interventi di manutenzione*. Per i primi, occorre indicare una descrizione del materiale utilizzato per il montaggio della serra. Ogni intervento di installazione si riferisce ad un'unica serra, ed ogni serra può essere associata ad al più un intervento di installazione (possono dunque esistere serre per le quali non è specificato l'intervento di installazione con cui sono state costruite). Riguardo agli interventi di manutenzione, ogni intervento di tale tipo riguarda un'unica serra, la quale, a sua volta, può essere associata ad un numero qualunque di interventi di manutenzione diversi.  $\exists$

Indipendentemente dal fatto di essere di installazione o manutenzione, ogni intervento è esattamente di uno dei seguenti tipi: *interno* o *in outsourcing*. Gli interventi interni sono espletati dal personale dell'azienda, per cui ciascun intervento interno è associato alla lista degli operai della stessa azienda che lo hanno effettuato. Ogni operario è caratterizzato dal numero di matricola (identificativo) e dal nome. Nell'indicare la sua partecipazione ad un intervento interno, deve essere possibile specificare il ruolo svolto. Non c'è alcun limite sul numero di interventi interni a cui può partecipare un operario.  $\forall$

Infine, ogni intervento in outsourcing può essere associato alla ditta esterna che lo ha effettuato (tale ditta potrebbe quindi non essere indicata). Le ditte sono identificate dalla coppia *<nome, indirizzo>*, e ciascuna di esse può avere effettuato un qualunque numero di interventi in outsourcing.

b) Si definisca uno schema relazionale corrispondente allo schema E-R.

Esercizio 2: Progettazione logica e dipendenze funzionali (30 minuti)

Si consideri lo schema relazionale  $R(A,B,C,D,E,F,G)$ , dove  $\beta$  è il seguente insieme di dipendenze funzionali:

$$\begin{aligned} BDC &\rightarrow F \\ F &\rightarrow C \\ E &\rightarrow B \\ AB &\rightarrow D \\ A &\rightarrow EF \\ EF &\rightarrow DG \\ G &\rightarrow A \end{aligned}$$



# INGEGNERIA

Grado = numero di figli di un nodo

## BERTOLINO

- Definire:  
1. una copertura minima per  $\beta$ ;  
2. le chiavi candidate di R;  
3. una decomposizione in BCNF di R che sia senza perdita di informazione;  
4. una decomposizione in 3NF di R che sia senza perdita di informazione, senza perdita di dipendenze funzionali e di dimensione minima.

Esercizio 3: Indici (20 minuti)

Si consideri un sistema con le seguenti caratteristiche:  
i. pagine fisiche di 4 KB;  
ii. intestazione fissa delle pagine fisiche di 128 byte;  
iii. puntatori di 4 byte.

Si consideri la relazione:

*Intervento* (*codice*, serra, data, motivazione)

e si assuma che il numero di byte destinato alla rappresentazione di ciascun attributo sia:  
a) 16 byte per l'attributo *codice*;  
b) 4 byte per ciascuno degli attributi *serra* e *data*;  
c) 112 byte per l'attributo *motivazione*.

Si assuma altresì che sia stato costruito un indice basato su struttura B-tree per consentire la valutazione efficiente di interrogazioni del tipo:

```
SELECT serra
  FROM Intervento
 WHERE codice=X;
```

dove X è un parametro fornito in ingresso.

Supponendo che il B-tree sia composto da tre livelli, ricavare il numero di tuple indicizzate ed il numero di pagine del B-tree nel caso in cui valgano contemporaneamente le seguenti due condizioni:

- a) La radice ha 5 figli;  
b) i nodi al terzo livello hanno riempimento minimo, ed il loro numero è pari al grado massimo del B-tree moltiplicato per il numero di nodi al secondo livello.

$$1) \text{ Pag} > \text{Iut} + p \cdot u + (p+k)(u-1)$$

$$1096 > 128 + 6u + 20(u-1)$$

1° livello  $\rightarrow$  1 nodo con 5 figli  $\nearrow$  1 pagina  
 $\searrow$  4 tuple

2° livello  $\rightarrow$  5 nodi con grado massimo  $\nearrow$  5 pagine  
 $5 \cdot 165$  tuple  $\searrow$  825 tuple

3° livello  $\rightarrow$   $(5 \cdot 166)$  nodi con riempimento minimo  
 $\searrow$  5 nodi  
 $(5 \cdot 166) \cdot 82$  tuple  
 $\searrow$  numero di chiavi

Se ha grado 83 nodi dire che  
ogni nodo ha 82 tuple

$\nearrow$  830 nodi  
 $\searrow$  68060 tuple

$$7) \text{ Tag} > \text{Lut} + p \cdot u + (p+1)(u-1)$$

$$4096 > 128 + 6u + 20(u-1)$$

$$4096 - 128 + 20 > 26u$$

$$u < \frac{3988}{24} < 166$$

$$u = 166 \quad \lceil \frac{u}{2} \rceil = 83$$

$\hookrightarrow$  grado max       $\hookrightarrow$  grado minimo

2) Primo livello: 1 nodo con 5 figli e 4 tuple

Secondo livello:  $5 \text{ nodi} \cdot (166-1) = 829 \text{ tuple}$

Terzo livello:  $166 \cdot 5 = 830 \text{ nodi} \times (83-1) \text{ tuple}$

$$1 + 5 + 830 \text{ nodi}$$

$$1 + 829 + 68060 \text{ tuple}$$

Se ha grado 83 vuol dire che ogni nodo ha 82 tuple

Nodi: 826

TUPLE: 68869

830 nodi  
68060 tuple

#### Corso di Laurea in Ingegneria Informatica (DM 270) Esame di Basi di Dati 6 marzo 2018

##### Esercizio 4: Interrogazioni su un database relazionale (50 minuti)

Sia dato il seguente schema di database:

- CONCESSIONARIA(Codice, nome, Indirizzo, Città, Cap, Nazione)
- AUTOMOBILE(CodAutomobile, Prezzo, Nome, Marca, Concessionaria)
- GUIDATORE(CodiceFiscale, Nome, Cognome, Città, DataNascita, Cittadinanza)
- ACQUISTO(CodAcquisto, Automobile, Guidatore, Anno)

con i seguenti vincoli di chiave esterna:

- AUTOMOBILE [Concessionaria]  $\hookrightarrow$  CONCESSIONARIA[Codice];
- ACQUISTO [Automobile]  $\hookrightarrow$  AUTOMOBILE[CodAutomobile];
- ACQUISTO[Guidatore]  $\hookrightarrow$  GUIDATORE[CodiceFiscale];

Lo studente formuli in SQL e, dove possibile, in Algebra Relazionale le seguenti interrogazioni:

- 1) Le concessionarie della città di Torino o Napoli che hanno venduto almeno due automobili, ciascuna delle quali è stata acquistata da almeno due guidatori distinti nati tra il 1984 e il 1988.
- 2) Il nome delle concessionarie situate in Germania che non hanno mai venduto un'automobile acquistata da un guidatore nato nel 1998.
- 3) Le coppie di guidatori  $\langle G_1, G_2 \rangle$  di città diverse che hanno acquistato automobili dello stesso insieme di marche.
- 4) Per ogni città, restituire i guidatori che tra il 1998 e il 2015 hanno acquistato il numero massimo di automobili di marca Fiat rispetto ai guidatori della stessa città.

##### Esercizio 5: Interrogazioni su un database XML (20 minuti)

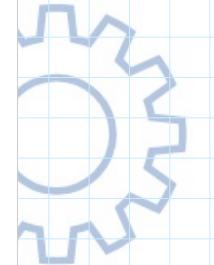
Si consideri un documento "gestioneAutomobili.xml" conforme al seguente DTD:

<!ELEMENT CasaAutomobilistica (Concessionaria)*>	?	0-1
<!ELEMENT Concessionaria (Nome, Automobile+, Impiegato*)>	+	1-u
<!ELEMENT Automobile (Codice, Nome, Accessorio*)>	*	0-1
<!ELEMENT Impiegato (Nome)*>		
<!ELEMENT Accessorio (Codice, Nome)*>		
<!ELEMENT Nome (#PCDATA)>		
<!ELEMENT Codice (#PCDATA)>		

Si formuli in XQuery la seguente interrogazione su "gestioneAutomobili.xml":

Restituire un documento XML contenente, per ogni concessionaria, il nome della concessionaria e la lista delle automobili presenti (assesti alla cardinalità della lista), e, per ciascuna automobile, il suo nome e il numero di accessori al suo interno. In particolare, il documento XML restituito deve essere conforme al seguente DTD:

```
<!ELEMENT Risultato (Concessionaria+)>
<!ELEMENT Concessionaria (Nome, ListaAutomobili)>
<!ELEMENT ListaAutomobili (Automobile+)>
<!ATTLIST ListaAutomobili numAutomobili Ceduta>
<!ELEMENT Automobile (Nome, NumeroAccessori)>
<!ELEMENT Nome (#PCDATA)>
<!ELEMENT NumeroAccessori (#PCDATA)>
```



I INGEGNERIA  
INFORMATICA  
BERTOLINO

29 giu 2018

mercoledì 2 febbraio 2022 19:55



29 giu 2018

**Corso di Laurea in Ingegneria Informatica (DM 270)**  
**Esame di Basi di Dati**  
**29 giugno 2018**

**Esercizio 1: Progettazione concettuale e logica di una Base di Dati (40 minuti)**

a) Si progetti una base di dati di supporto alla gestione di una scuola di Judo.

Le persone sulle quali occorre rappresentare informazioni per supportare la gestione della scuola sono di due tipologie: maestri ed allievi. Indipendentemente dal tipo, le persone sono caratterizzate dal nome, dalla data di nascita, e dall'indirizzo di residenza. Possono esistere persone con lo stesso nome, ma in tale caso non possono essere nate nello stesso giorno. I maestri sono caratterizzati dal "dan" (un valore numerico che certifica le loro abilità ed esperienza nella pratica del judo) e dalla data in cui hanno ricevuto il dan. Per ogni maestro, occorre indicare il maestro che ha certificato l'attribuzione del dan. Ogni maestro può essere certificatore di un numero qualunque di altri maestri. I maestri sono inoltre suddivisi in due categorie: interni ed esterni. Come sarà più chiaro nel seguito, i maestri interni sono quelli che insegnano l'arte del judo agli allievi, mentre quelli esterni sono i certificatori delle promozioni degli allievi.

dei maestri

Per quanto riguarda gli allievi, per ciascuno di essi deve essere possibile tenere traccia della lista delle iscrizioni annuali da lui sottoscritte. Ogni iscrizione è caratterizzata dall'allievo, dall'anno solare a cui si riferisce, e dalle caratteristiche del certificato medico presentato (nome del medico, data di rilascio). Per ogni allievo e per ogni anno solare, esiste al più un'iscrizione. Ogni iscrizione è inoltre associata al maestro interno che si occupa della formazione dell'allievo nell'anno solare ricoperto dall'iscrizione. Un maestro interno può essere associato ad un numero qualunque di iscrizioni in termini di curatore della formazione degli allievi.

Per ogni iscrizione vanno inoltre memorizzati i pagamenti rateali effettuati dall'allievo. Ciascun pagamento è caratterizzato, oltre che dall'iscrizione, dalla data in cui è avvenuto, dall'importo versato, e dal periodo ricoperto, descritto da una data di inizio ed un numero di mesi. Non possono esistere, relativamente alla stessa iscrizione, più pagamenti che ricoprono periodi con la stessa data di inizio. Ciascun pagamento può essere associato ad al più una convenzione, che ha determinato l'applicazione di uno sconto. Ogni convenzione può essere associata ad un numero qualunque di pagamenti ed è identificata da un codice, oltre che essere caratterizzata da un nome, da una descrizione, e dalla percentuale di sconto a cui dà diritto.

Ogni allievo è inoltre associato alla lista delle promozioni che ha conseguito. Oltre che dall'allievo, ciascuna promozione è caratterizzata dal colore della cintura che è stata assegnata all'allievo, dal giorno in cui è stato sostenuto l'esame, e dal nome del maestro esterno che ha certificato la promozione. Ogni maestro esterno può certificare un numero qualunque di promozioni. Per ciascun allievo, non possono esistere più promozioni con lo stesso colore di cintura.

b) Si definisca uno schema relazionale corrispondente allo schema E-R.

**Esercizio 2: Progettazione logica e dipendenze funzionali (30 minuti)**

Si consideri lo schema relazionale  $\langle R(A,B,C,D,E,F,G,H), \mathcal{F} \rangle$ , dove  $\mathcal{F}$  è il seguente insieme di

dipendenze funzionali:

$AB \rightarrow C$

$A \not\rightarrow D \rightarrow E$

$DE \rightarrow F$

$G \rightarrow AD$

$F \rightarrow G$

$A \rightarrow H$

$D \rightarrow B$

$E \rightarrow H$

Scanned by CamScanner

Definire:

1. una copertura minimale per  $\mathcal{R}$ ;
2. le chiavi candidate di  $R$ ;
3. una decomposizione in BCNF di  $R$  che sia senza perdita di informazione;
4. una decomposizione in 3NF di  $R$  che sia senza perdita di informazione, senza perdita di dipendenze funzionali e di dimensione minima.

### Esercizio 3: Indici (20 minuti)

Si consideri un sistema con le seguenti caratteristiche:

- i. pagine fisiche di 2 KB;
- ii. intestazione fissa delle pagine fisiche di 128 byte;
- iii. puntatori di 4 byte.

Si consideri la relazione:

*Allievo* (CF, nome, coloreCintura)

e si assuma che il numero di byte destinato alla rappresentazione di ciascun attributo sia:

- a) 16 byte per l'attributo *codiceCognome*;
- b) 64 byte per l'attributo *federazione*;
- c) 2 byte per l'attributo *note*.

Si assume altresì che sia stato costruito un indice basato su struttura B-tree per consentire la valutazione efficiente di interrogazioni del tipo:

```
SELECT coloreCintura  
FROM Allievo  
WHERE CF=X;
```

dove  $X$  è un parametro fornito in ingresso.

Supponendo che il B-tree sia composto da tre livelli, ricavare il numero di tuple indicizzate ed il numero di pagine del B-tree nel caso in cui valgano contemporaneamente le seguenti tre condizioni:

- a) La radice contiene un numero di chiavi pari alla metà del numero minimo di chiavi che possono essere presenti negli altri nodi;
- b) I nodi al secondo livello hanno riempimento massimo;
- c) I nodi al terzo livello hanno fattore di riempimento dell'80%.

Scanned by CamScanner

$$1) \text{ Pag} = 8192 \text{ byte}$$

$$\text{Pag} > \text{Int} + (\text{u} \cdot p) + (\text{k}_{\text{tp}})(\text{u}-1)$$

$$8192 > 128 + (4u) + (20)(u-1)$$

$$8192 - 128 + 20 > 24u$$

$$u < 336,83 \rightarrow u = 336$$

$$\left\lceil \frac{u}{2} \right\rceil = 168$$

$$u = u_{\text{min}} \dots$$

$$\dots u = u_{\text{max}}$$

temp. massimo

$$\left| \frac{u}{2} \right| = u_{\text{mid}}$$

temp. minimo

La radice ha un numero di chiavi pari a  $\frac{168}{2} = 84$

Quindi ha 85 figli e 84 tuple

Il secondo livello avrà 85 nodi ciascuno a  
riempimento massimo quindi  $85(336)^{-1}$  tuple = 28475  
e avrà  $85 \cdot 85$  figli = 7225

Il Terzo livello ha 7225 nodi ciascuno con riempimento  
al 90% quindi  $7225(336 \cdot 0,9)^{-1} = 2184840$

$$\text{Pagine} = 1 + 85 +$$

$$\text{Tuple} = 84 + 28475 +$$

## APPUNTI DI INGEGNERIA INFORMATICA

GAIA BERTOLINO

Corso di Laurea in Ingegneria Informatica (DM 270)  
Esame di Basi di Dati  
29 Giugno 2018

**Esercizio 4: Interrogazioni su un database relazionale (50 minuti)**

Sia dato il seguente schema di database:

- COMUNE(NomeComune, Regione, Provincia, NumeroAbitanti)
- AZIENDA(PIVA, RagSociale, NomeComune, RegioneComune)
- PRODOTTO(CodProdotto, Tipo, Descrizione, Peso)
- LISTINO(CodProdotto, TipoProdotto, Azienda, Prezzo, Sconto)

con i seguenti vincoli di chiave esterna:

- AZIENDA[NomeComune, RegioneComune] ⊑ COMUNE[NomeComune, Regione];
- LISTINO[CodProdotto, TipoProdotto] ⊑ PRODOTTO[CodProdotto, Tipo];
- LISTINO[Azienda] ⊑ AZIENDA[PIVA];

Lo studente formuli in SQL e, dove possibile, in Algebra Relazionale le seguenti interrogazioni:

- 1) I comuni in cui sono situate almeno due aziende, ciascuna delle quali ha in listino almeno due prodotti diversi di prezzo superiore a 20 euro.
- 2) La partita IVA e la Ragione Sociale delle aziende che non hanno in listino prodotti di tipo "detersivo" e con peso inferiore a 500 grammi.
- 3) Le coppie di aziende  $\langle P_1, P_2 \rangle$  che hanno in listino lo stesso insieme di tipi di prodotti.
- 4) Per ogni comune, restituire le aziende che hanno nel listino prodotti con sconto al 50 % ed il cui prezzo complessivo è minimo rispetto alle aziende dello stesso comune.

**Esercizio 5: Interrogazioni su un database XML (20 minuti)**

Si consideri un documento "azienda.xml" conforme al seguente DTD:

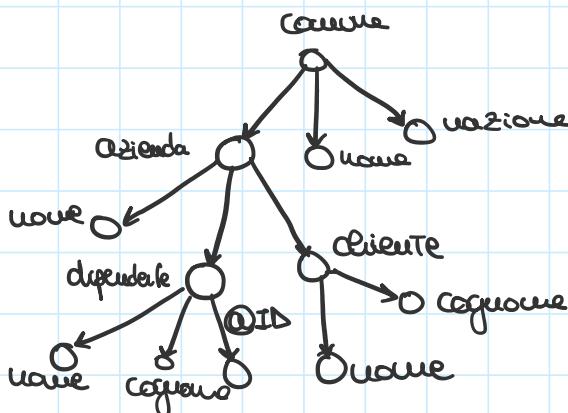
```
<!ELEMENT Comune (Azienda+, Nome, Nazione)>
<!ELEMENT Azienda (Nome, Dipendente+, Cliente+)>
<!ELEMENT Dipendente (Cognome, Nome)>
<!ATTLIST Dipendente id ID #REQUIRED >
<!ELEMENT Cliente (Cognome, Nome)>
<!ELEMENT Cognome (#PCDATA)>
<!ELEMENT Nome (#PCDATA)>
<!ELEMENT Nazione (#PCDATA)>
```

Si formuli in XQuery la seguente interrogazione su "azienda.xml":  
*Restituire un documento XML contenente, per ogni azienda, la lista dei suoi clienti, la lista dei suoi dipendenti, e il numero dei dipendenti.*

In particolare, il documento XML restituito deve essere conforme al seguente DTD:

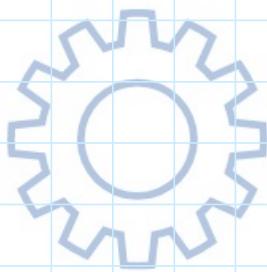
```
<!ELEMENT risultato (Azienda+)>
<!ELEMENT Azienda (Nome, Clienti, Dipendenti) >
<!ELEMENT Clienti(Cliente+)>
<!ELEMENT Cliente (Cognome, Nome)>
<!ELEMENT Dipendenti(Dipendente+)>
<!ELEMENT Dipendente (Cognome, Nome)>
<!ATTLIST Dipendenti numeroDipendenti CDATA>
<!ELEMENT Nome (#PCDATA)>
<!ELEMENT Cognome (#PCDATA)>
```

Scanned by CamScanner



Retorno <risultato> {

document (azienda.xls) /



# APPUNTI DI INGEGNERIA INFORMATICA

GAIA BERTOLINO

# Esercizio

giovedì 3 febbraio 2022 10:41

10:30

$F \rightarrow D$   
 $CDE \rightarrow F$   
 $A \rightarrow C$   
 $B \rightarrow AF$   
 $BC \rightarrow E$   
 $AF \rightarrow EG$   
 $C \rightarrow B$

Verifica la chiave e cerca le chiavi:

$B^+ = \{A, B, C, D, E, F, G\} \rightarrow$  chiave

Quindi la C in BC è ridondante. Il sistema diventa:

$F \rightarrow D$ $CDE \rightarrow F$ $A \rightarrow C$ $B \rightarrow AF$ $B \rightarrow E$ $AF \rightarrow EG$ $G \rightarrow B$	=>	$F \rightarrow D$ $CDE \rightarrow F$ $B \rightarrow AFE$ $AF \rightarrow EG$ $G \rightarrow B$
--	----	---

$G^+ = \{A, B, C, D, E, F, G\} \rightarrow$  chiave

$ECD^+ = \{C, D, E, F\} \rightarrow$  NO dunque è una delle relazioni non in BCNF

$AF^+ = \{A, B, C, D, E, F, G\} \rightarrow$  chiave

Altre possibili chiavi:

Avere chiavi A, C, D, E, F non sono chiavi, la conclude!

$A^+ = \{A, C\}$ $C^+ = \{C\}$ $D^+ = \{D\}$ $E^+ = \{E\}$ $F^+ = \{D, F\}$	}	$ADE^+ = \{A, B, C, D, E, F, G\} \rightarrow$ chiave $CF^+ = \{C, D, F\} \rightarrow$ NO
---	---	---