**CdL in Informatica Triennale – A.A. 2020-2021**
**Basi di Dati**
**Proff. S. Alaimo - A. Pulvirenti**
**Prova scritta 17 giugno 2021**

| Matricola | Nome e Cognome | Valutazione |
|---|---|---|
| | | |

a) Si consideri il seguente schema relazionale relativo alla gestione delle riunioni di un dipartimento:

Riunione(id, iddipartimento data, ora)

Odg(idriunione, idodg, descrizione, docente)

Docente(id,iddipartimento nome, cognome)

presentiRiunione(idriunione, iddocente)

1. identificare le chiavi primarie ed esterne dello schema [0 punti corretta, -1 sbagliata]
2. Rispondere alle seguenti query in algebra relazionale
    a. Trovare i docenti, mostrando nome e cognome, che hanno partecipato a tutte le riunioni [4 punti]
    b. Trovare le riunioni che non hanno esattamente due record in odg [4 punti]
3. Rispondere alle seguenti query in SQL
    a. Per ogni docente trovare il numero di riunioni a cui non ha partecipato [4 punti]
    b. Trovare gli anni in cui il numero complessivo di partecipanti alle riunioni è stato maggiore di quello medio [4 punti]
4. Implementare una asserzione che eviti di far inserire nella tabella “presentiRiunione” docenti che appartengono a dipartimenti differenti rispetto a quello dove la riunione è stata indetta [5 punti]
5. Dedurre uno schema ER che descriva lo schema logico [3 punti]

b) Si consideri il seguente schema relazionale R(A,B,C,D,E,G,H), F={A→B, B→CA, C→ D,D→A, A→G, H→A}
    a. Identificare le chiavi dello schema [2 punti]
    b. Decomporre lo schema in BCNF [4 punti]

**CdL in Informatica Triennale – A.A. 2020-2021**
**Basi di Dati**
**Proff. S. Alaimo - A. Pulvirenti**
**Prova scritta 8 luglio 2021**

| Matricola | Nome e Cognome | Valutazione |
|---|---|---|
| | | |

Si consideri il seguente schema relazionale:

Piazzola(id, dimensione, tipologia)

Prenotazione(id, idpiazzola, data_arrivo, data_partenza, numero_adulti, numero_bambini)

DettagliPrenotazione(idprenotazione, cf)

Stagione(id,data_inizio, data_fine, prezzo_piazzola, tipologia, prezzo_adulti, prezzo_bambini)

Cliente(cf,nome,cognome, data_nascita)

Dove l'attributo tipologia indica: Tenda, Camper, Roulotte;

1. Definire le chiavi primarie ed esterne dello schema (0 corretta, -1 errata)
2. Rispondere alle seguenti query in algebra relazionale
    a. Identificare le piazzole di dimensione massima che non sono state mai prenotate (2 punti);
3. Rispondere alle seguenti query in SQL
    a. Per ogni tipologia di piazzola indicare gli incassi per ogni stagione (6 punti);
    b. Trovare le stagioni per le quali il numero di prenotazione è stato inferiore a quello medio di ogni stagione (6 punti);
4. Ipotizzare di avere la relazione fattura(id,id_prenotazione, costo complessivo). Implementare un trigger che al checkout (inserimento data_partenza nella tabella prenotazione) calcola il costo complessivo del soggiorno ed inserisce il corrispondente record nella tabella fattura (4 punti);
5. Definire uno schema ER che rappresenta lo schema relazionale (4 punti)

Si consideri il seguente schedule S: r1(x)w1(y)r2(z)w2(x)r3(y)r4(x)r1(t)r4(z)w2(t)

1. Stabilire se lo schedule e' VSR o CSR (4 punti);
2. Se passato ad uno scheduler 2PL causa dei deadlock (4 punti).

Esame

Studente(matricola,nome,cognome)

Materia(id,titolo,descrizione)

Esercizi(id,testo,soluzione, materia,numerosoluzioni)

Risolto(idesercizio,idstudente,data)

-Definire le chiavi primarie ed esterne (ITINERE) **[1 punto]**

Algebra

-Trovare gli studenti che non hanno risolto esercizi per la materia "basi di dati" **[3 punti]**

$$R1 = MATERIA \bowtie_{MATERIA.id=ESERCIZI.materia} ESERCIZI \bowtie_{ESERCIZI.id=RISOLTO.idesercizio} RISOLTO$$
$$R2 = \pi_{matricola}(STUDENTE) - \delta_{idstudente \to matricola}\left(\pi_{idstudente}\left(\sigma_{titolo="BASI\ DI\ DATI"}(R1)\right)\right)$$
$$STUDENTE \bowtie R1$$

-Trovare le materie per cui sono stati risolti tutti gli esercizi (ITINERE) **[3 punti]**

$$MATERIA - \delta_{materia \to id}\left(\pi_{materia}\left(\sigma_{numerosoluzioni=0}(ESERCIZI)\right)\right)$$

SQL

-Per ogni materia contare il numero di esercizi disponibili e quelli risolti **[4 punti]**

```
SELECT t1.materia,numeroesercizi,numeroesercizirisolti
FROM (
    SELECT count(*) numeroesercizi, materia
    FROM ESERCIZI
    GROUP BY MATERIA
  ) AS t1,(
    SELECT count(*) numeroesercizirisolti, materia
    FROM ESERCIZI
    WHERE numerosoluzioni>0
    GROUP BY MATERIA
  ) AS t2
WHERE t1.materia=t2.materia
```

-Trovare gli esercizi che contengono la parola "SQL" che non sono stati risolti (ITINERE) **[4 punti]**

```
SELECT *
FROM Esercizi
WHERE testo LIKE '%SQL%' AND numerosoluzioni = 0
```

-Implementare un trigger in SQL che ogni qualvolta viene inserita una soluzione per un esercizio nella relazione Risolto aggiorna il campo numerosoluzioni della tabella esercizi (ITINERE) **[4 punti]**

```
CREATE TRIGGER T1
AFTER INSERT ON RISOLTO
FOR EACH ROW
UPDATE Esercizi
SET numerosoluzioni = numerosoluzioni + 1
WHERE id = NEW.idesercizio
```

ER **[5 punti]**

Si supponga di avere le seguenti operazioni:

q1 - Inserisci soluzione nella relazione Risolto 100 volte al giorno

q2 - Dai il numero di soluzioni proposte per un esercizio

Valutare se conviene mantenere l'attributo "numerosoluzioni"

| Con ridondanza | Senza ridondanza |
|---|---|
| Q1<br>1 Scrittura Risolto<br>1 Scrittura Esercizi<br>TOTALE = 2 S = 4 L -> 400<br>Q2<br>1 Lettura Esercizi<br>TOTALE = 1 * numero operazioni<br><br>COSTO TOTALE = 400 + 1*x | Q1<br>1 Scrittura Risolto<br>TOTALE = 1 S = 2 L -> 100<br>Q2<br>1 Lettura Esercizio<br>10 Letture Risolto<br>TOTALE = 11 * numero operazioni<br><br>COSTO TOTALE = 200 + 11 * x |

Se 200 + 11*x > 400 + x converrà mantenere la ridondanza
Ovvero 10*x -200 > 0
Quindi se x > 20 converrà mantenere la ridondanza

Normalizzazione **[6 punti]**

Dato lo schema Impiegato (Nome, Livello, Stipendio) con le seguenti dipendenze funzionali
F={Nome→Livello, Stipendio, Livello→ Stipendio}

-Dire se lo schema è in 3NF o BCNF

Chiave Nome

Non è in 3NF ne BCNF

- Se non è in BCNF decomporlo in modo tale che rispetti la BCNF

BCNF: R1(Livello,Stipendio) R2(Nome,Livello)

Itinere 1

Persona(cf,nome,cognome)

Libro(id,titolo,descrizione,autore,numerodilettori,datauscita,saga,volume)

Recensione(id,libro, testo,data,persona)

Letto(libro,persona,data)

-Trovare le chiavi primarie ed esterne dello schema

Algebra

-Trovare i libri che hanno almeno 2 recensioni ma che non sono stati letti

PROJ_libro (SEL_ libro=libro1 AND id>id_1(Recensione x RID_id->id_1,libro->libro1(Recensione))) – PROJ_libro(letto)

-Trovare le persone che hanno letto tutti i libri di “JK Rowling”

libriJKR := RID_id->libro(PROJ_id(SEL_autore=”JK Rowling”(libro)))

PROJ_libro,persona / libirJKR

SQL

-Implementare un vincolo che non consenta di inserire in Letto un libro di una saga se non nel corretto ordine cronologico (V1, V2,…)

```
CREATE TRIGGER T1 AFTER INSERT ON Letto

Declare int X,Y,Z

For each row

SELECT volume into x, saga into y

FROM libro WHERE id = new.libro

If( X IS NOT NULL AND x > 1)

        SELECT id into Z FROM LIBRO WHERE saga = Y and volume = X-1;

        If(NOT EXISTS (SELECT * FROM letto where libro = Z and persona = new.persona)

                Delete from letto WHERE data = new.data..
```

-Per ogni autore contare il numero di libri e il numero di lettori distinti e il numero di recensioni avute da persone distinte.

```
CREATE VIEW V1 AS

SELECT count(DISTINCT persone) lettori,autore FROM letto, libro WHERE libro=ID GROUP BY autore
```

CREATE VIEW V2 AS

SELECT count(DISTINCT persone) recesioni,autore FROM recensione, libro WHERE libro=ID GROUP BY autore

CREATE VEW V3 AS

SELECT COUNT(*) libri, autore FROM libro group by autore

SELECT libri, recesioni, lettori FROM V1 Natural join V2 natural join V3

Itinere 2

AutoPosseduta(targa,idauto ,costorifornimenti,dataimmatricolazione)

Auto(id,marca, alimentazione,cilindrata)

Rifornimento(targa,data,prezzolitro,litri)

Manutanezione(targa,data,descrizione,costo)

-Identificare le chiavi primarie ed esterne

Algebra

-Trovare le auto, indicando marca e modello, che non sono state vendute

AUTO JOIN (PROJ_ID(AUTO) – REN_idauto->id(PROJ(AUTOPOSSEDUTA)))

-Trovare per ogni marca trovare le auto con la cilindrata maggiore

PROJ_cilindrata,marca(auto) -PROJ_cilindrata,marca(SEL_cilindrata < c1 and marca=marca1(AUTO X REN_ID->ID1,marca->marca1,almentazione->a1,cilindarata->c1(AUTO)))

SQL

-Trovare le marche di automobili che hanno venduto tutti i modelli

SELECT DISTINCT marca

FROM auto a1

WHERE not exists (SELECT * FROM aUTO a2 WHERE a1.marca =a2.marca AND

Not exists SELECT * FROM autoposseduta WHERE a2.id=idauto))

-Per ogni autoposseduta mostrare quelle per il quale il numero di manutenzioni effettuate è maggiore di quello medio. Per queste visualizzare pure il costo totale della manutenzione

SELECT count(*) nmat, targa FROM manutenzione

GROUP BY targa

HAVING nmat >= (Select AVG(NUMMANUT) FROM ( Select count(*) nummanut from manutenzione group by targa))

-Implementare un trigger che ogni qualvolta viene inserito un rifornimento in Rifornimento aggiorna il costo complesivo in AutoPosseduta

CREATE TRIGGER t1 after insert on rifornimento

For each row

Update autposseduata set costotale = costotoale + new.costo where targa= new.targa

Itinere 3

Libro(id,titolo,descrizione,autore ,datauscita,sequeldi ,genere)

CopiaLibro(collocazione,idlibro)

Persona(id,nome,cognome,prestiti)

Presitito(libro,persona,dataprestito,datarestituzione,restitutito)

-Identificare le chiavi primarie ed esterne dello schema

-Trovare le persone che non hanno mai chiesto libri di “Stephen King” ma hanno chiesto almeno un libro di “Joseph Conrad”

(PROJ_persona(PRESTITO) - PROJ_persona(PRESTITO NATURAL JOIN Ren_collocazione->libro(Proj_collocazione(SEL_autore=“Stephen king”(Copialibro JOIN_idlibro=id LIBRO))))) NATURAL JOIN Ren_collocazione->libro(Proj_collocazione(SEL_autore=“Joseph Conrad”(Copialibro JOIN_idlibro=id LIBRO))))

-Trovare le persone che hanno preso in prestito tutti i libri del genere “Fantasy”

R1 := Proj_id(sel_genere=fantasy(libro))

R2 :=PROJ_idlibro,persona(R1 JOIN_id=idlibro COPIALIBRO JOIN_libro=collocazione PRESTITO)

R2 / R1

SQL

-Trovare le coppie di libri che hanno almeno 2 sequel

WITH RECURSIVE sequel

(SELECT id, sequeldi FROM libro

UNION ALL

SELECT libro.id,sequel.sequeldi

FROM libro, sequel WHERE sequel.sequeldi= libro.id FROM sequel, libro)

SELECT id

FROM sequel

GROUP BY id

HAVING count(*) >=2

-Trovare libri che hanno avuto piu’ prestiti di quelli medi avuti da ogni libro (!)

SELECT count(*) prestiti, libro

FROM prestito,copialibro WHERE libro= collocazione

GROUP BY libro

HAVING prestiti >= (SELECT AVG(prestiti) FROM  (SELECT count(*) prestiti, libro

FROM prestito,copialibro WHERE libro= collocazione

GROUP BY libro))

-Implementare un vincolo che non consenta di inserire un nuovo prestito per una persona che ha ancora un libro prestato e non restituito

CREATE TRIGGER T1 after insert on presitito

When (1 < select count(*) from prestito where persona = new.persona and restituito IS NULL)

DELETE FROM prestito WHERE libro = new.libro, persona=new.persona, data= new.data

Esame

Studente(matricola,nome,cognome)

Materia(id,titolo,descrizione)

Esercizi(id,testo,soluzione, materia,numerosoluzioni)

Risolto(idesercizio,idstudente,data)

-Definire le chiavi primarie ed esterne (ITINERE) **[1 punto]**

Algebra

-Trovare gli studenti che non hanno risolto esercizi per la materia "basi di dati" **[3 punti]**

$$R1 = MATERIA \bowtie_{MATERIA.id=ESERCIZI.materia} ESERCIZI \bowtie_{ESERCIZI.id=RISOLTO.idesercizio} RISOLTO$$
$$R2 = \pi_{matricola}(STUDENTE) - \delta_{idstudente \to matricola}\left(\pi_{idstudente}\left(\sigma_{titolo="BASI\ DI\ DATI"}(R1)\right)\right)$$
$$STUDENTE \bowtie R1$$

-Trovare le materie per cui sono stati risolti tutti gli esercizi (ITINERE) **[3 punti]**

$$MATERIA - \delta_{materia \to id}\left(\pi_{materia}\left(\sigma_{numerosoluzioni=0}(ESERCIZI)\right)\right)$$

SQL

-Per ogni materia contare il numero di esercizi disponibili e quelli risolti **[4 punti]**

```
SELECT t1.materia,numeroesercizi,numeroesercizirisolti
FROM (
    SELECT count(*) numeroesercizi, materia
    FROM ESERCIZI
    GROUP BY MATERIA
  ) AS t1,(
    SELECT count(*) numeroesercizirisolti, materia
    FROM ESERCIZI
    WHERE numerosoluzioni>0
    GROUP BY MATERIA
  ) AS t2
WHERE t1.materia=t2.materia
```

-Trovare gli esercizi che contengono la parola "SQL" che non sono stati risolti (ITINERE) **[4 punti]**

```
SELECT *
FROM Esercizi
WHERE testo LIKE '%SQL%' AND numerosoluzioni = 0
```

-Implementare un trigger in SQL che ogni qualvolta viene inserita una soluzione per un esercizio nella relazione Risolto aggiorna il campo numerosoluzioni della tabella esercizi (ITINERE) **[4 punti]**

```
CREATE TRIGGER T1
AFTER INSERT ON RISOLTO
FOR EACH ROW
UPDATE Esercizi
SET numerosoluzioni = numerosoluzioni + 1
WHERE id = NEW.idesercizio
```

ER **[5 punti]**

Si supponga di avere le seguenti operazioni:

q1 - Inserisci soluzione nella relazione Risolto 100 volte al giorno

q2 - Dai il numero di soluzioni proposte per un esercizio

Valutare se conviene mantenere l'attributo "numerosoluzioni"

| Con ridondanza | Senza ridondanza |
|---|---|
| Q1<br>1 Scrittura Risolto<br>1 Scrittura Esercizi<br>TOTALE = 2 S = 4 L -> 400<br>Q2<br>1 Lettura Esercizi<br>TOTALE = 1 * numero operazioni<br><br>COSTO TOTALE = 400 + 1*x | Q1<br>1 Scrittura Risolto<br>TOTALE = 1 S = 2 L -> 100<br>Q2<br>1 Lettura Esercizio<br>10 Letture Risolto<br>TOTALE = 11 * numero operazioni<br><br>COSTO TOTALE = 200 + 11 * x |

Se 200 + 11*x > 400 + x converrà mantenere la ridondanza
Ovvero 10*x -200 > 0
Quindi se x > 20 converrà mantenere la ridondanza

Normalizzazione **[6 punti]**

Dato lo schema Impiegato (Nome, Livello, Stipendio) con le seguenti dipendenze funzionali
F={Nome→Livello, Stipendio, Livello→ Stipendio}

-Dire se lo schema è in 3NF o BCNF

Chiave Nome

Non è in 3NF ne BCNF

- Se non è in BCNF decomporlo in modo tale che rispetti la BCNF

BCNF: R1(Livello,Stipendio) R2(Nome,Livello)

Itinere 1

Persona (cf, nome, cognome)

Libro (id, titolo, descrizione, autore, numerodilettori, datauscita, saga, volume)

Recensione (id, ~~libro~~, testo, data, ~~persona~~)

Letto(~~libro, persona~~, data)

-Trovare le chiavi primarie ed esterne dello schema **[1 punto]**

Algebra

-Trovare i libri che hanno almeno 2 recensioni ma che non sono stati letti **[3 punti]**

$$R1 = R2 = Recensione$$
$$\pi_{R1.libro}(R1 \bowtie_{R1.libro=R2.libro \ \wedge R1.id>R2.id} R2) - \pi_{libro}(Letto)$$

-Trovare le persone che hanno letto tutti i libri di "JK Rowling" **[3 punti]**

$$LibriJKR \coloneqq \delta_{id \to libro}\left(\pi_{id}\left(\sigma_{autore="JK\ Rowling"}(Libro)\right)\right)$$
$$\pi_{libro,persona}(Letto)/LibriJKR$$

SQL

-Implementare un vincolo che non consenta di inserire in Letto un libro di una saga se non nel corretto ordine cronologico (V1, V2,…) **[4 punti]**

```
CREATE TRIGGER T1 AFTER INSERT ON Letto
FOR EACH ROW
DECLARE X,Y,Z INT
BEGIN
  SELECT volume INTO X, saga INTO Y FROM libro WHERE id = NEW.libro
  IF (X IS NOT NULL AND X > 1) THEN
    SELECT id INTO Z FROM LIBRO WHERE saga = Y and volume = X-1;
    IF(NOT EXISTS (SELECT * FROM letto where libro = Z and persona = new.persona) THEN
      DELETE FROM letto WHERE data = new.data
    END IF
  END IF
END
```

-Per ogni autore contare il numero di libri e il numero di lettori distinti e il numero di recensioni avute da persone distinte. **[4 punti]**

```
CREATE VIEW V1 AS SELECT COUNT(DISTINCT persone) AS lettori,autore FROM Letto, Libro
WHERE libro=ID GROUP BY autore

CREATE VIEW V2 AS SELECT COUNT(DISTINCT persone) AS recesioni,autore FROM recensione,
libro WHERE libro=ID GROUP BY autore

CREATE VEW V3 AS SELECT COUNT(*) libri, autore FROM libro group by autore

SELECT libri, recesioni, lettori FROM V1 NATURAL JOIN V2 NATURAL JOIN V3
```

Itinere 2

AutoPosseduta(<u>targa</u>,idauto ,costorifornimenti,dataimmatricolazione)
Auto(<u>id</u>,marca, alimentazione,cilindrata)
Rifornimento(<u>targa,data</u>,prezzolitro,litri)
Manutanezione(<u>targa,data</u>,descrizione,costo)

-Identificare le chiavi primarie ed esterne **[1 punto]**

Algebra

-Trovare le auto, indicando marca e modello, che non sono state vendute **[2 punti]**

$$Auto \bowtie \left(\pi_{id}(Auto) - \delta_{idauto \to id}\left(\pi_{idauto}(AutoPosseduta)\right)\right)$$

-Trovare per ogni marca trovare le auto con la cilindrata maggiore **[3 punti]**

$$A1 = A2 = Auto$$
$$R1 = \pi_{A1.cilindrata,A1.marca}(A1 \bowtie_{A1.cilindrata<A2.cilindrata \wedge A1.marca=A2.marca} A2)$$
$$\pi_{cilindrata,marca}(Auto) - R1$$

SQL

-Trovare le marche di automobili che hanno venduto tutti i modelli **[3 punti]**

```
SELECT DISTINCT marca
FROM auto a1
WHERE NOT EXISTS (SELECT * FROM Auto a2 WHERE a1.marca=a2.marca AND
NOT EXISTS (SELECT * FROM AutoPosseduta WHERE a2.id=idauto))
```

-Per ogni autoposseduta mostrare quelle per il quale il numero di manutenzioni effettuate è maggiore di quello medio. Per queste visualizzare pure il costo totale della manutenzione **[4 punti]**

```
SELECT COUNT(*) AS nmat,SUM(costo), targa FROM manutenzione GROUP BY targa
HAVING nmat >= (SELECT AVG(NumManut) FROM (SELECT COUNT(*) AS NumManut FROM manutenzione
GROUP BY targa))
```

-Implementare un trigger che ogni qualvolta viene inserito un rifornimento in Rifornimento aggiorna il costo complesivo in AutoPosseduta **[2 punti]**

```
CREATE TRIGGER t1 AFTER INSERT ON Rifornimento
FOR EACH ROW
UPDATE AutoPosseduata
SET costorifornimenti = costorifornimenti + NEW.costo
WHERE targa = NEW.targa
```

Itinere 3

Libro(<u>id</u>,titolo,descrizione,autore ,datauscita,sequeldi ,genere)

CopiaLibro(<u>collocazione</u>,idlibro)

Persona(<u>id</u>,nome,cognome,prestiti)

Presitito(<u>libro,persona,dataprestito</u>,datarestituzione,restitutito)

-Identificare le chiavi primarie ed esterne dello schema **[1 punto]**

-Trovare le persone che non hanno mai chiesto libri di “Stephen King” ma hanno chiesto almeno un libro di “Joseph Conrad” **[3 punti]**

$$R1 = \delta_{collocazione \to libro}(CopiaLibro \bowtie_{idlibro=id} Libro)$$
$$R2 = \sigma_{autore="Stephen\ King"}(R1)$$
$$R3 = \sigma_{autore="Joseph\ Conrad"}(R1)$$
$$\left(\pi_{persona}(Prestito) - \pi_{persona}(Prestito \bowtie R2)\right) \bowtie R3$$

-Trovare le persone che hanno preso in prestito tutti i libri del genere “Fantasy” **[2 punti]**

$$R1 := \pi_{id}\left(\sigma_{genere="fantasy"}(Libro)\right)$$
$$R2 := \pi_{id,persona}(R1 \bowtie_{id=idlibro} CopiaLibro \bowtie_{libro=collocazione} Prestito)$$
$$R2 \div R1$$

SQL

-Trovare le coppie di libri che hanno almeno 2 sequel **[3 punti]**

```
WITH RECURSIVE sequel (
  SELECT id, sequeldi FROM libro
  UNION ALL
  SELECT libro.id,sequel.sequeldi
  FROM libro, sequel WHERE sequel.sequeldi = libro.id FROM sequel, libro
) SELECT id FROM sequel GROUP BY id HAVING count(*) >=2
```

-Trovare libri che hanno avuto piu’ prestiti di quelli medi avuti da ogni libro (!) **[4 punti]**

```
SELECT count(*) AS prestiti, libro FROM prestito,copialibro
WHERE libro = collocazione GROUP BY libro
HAVING prestiti >= (SELECT AVG(prestiti) FROM (SELECT count(*) AS prestiti, libro
FROM prestito, copialibro WHERE libro = collocazione GROUP BY libro))
```

-Implementare un vincolo che non consenta di inserire un nuovo prestito per una persona che ha ancora un libro prestato e non restituito **[2 punti]**

```
CREATE TRIGGER T1 AFTER INSERT ON Prestito
WHEN (1 < SELECT count(*) FROM prestito where persona=NEW.persona and restituito IS NULL)
DELETE FROM prestito WHERE libro = NEW.libro, persona=NEW.persona, data=NEW.data
```

# Esame Basi di Dati
# 29 marzo 2021

a) Si supponga di avere il seguente schema relazionale:

viaggi(id,id_cliente,id_guidatore,id_citta,stato,data_richiesta)

utenti(id_utente,bannato,ruolo)

dove:

- l'attributo stato è un tipo enumerato che assume i valori "completato", "cancellato_dal_guidatore","cancellato_dal_cliente".

- Bannato assume i valori V/F

- Ruolo assume i valori "Cliente", "Guidatore"

1. Identificare le chiavi primarie ed esterne dello schema [0 punti corretta, -1 sbagliata]
2. Rispondere alle seguenti query in algebra:
   a. Trovare le date per cui esiste almeno un viaggio per tutti i clienti [4 punti];

      proj_(id_cliente, data_richiesta) (viaggi) / proj_(id_cliente) viaggi

3. Rispondere alle seguenti query in SQL
   b. Per ogni data, trovare il tasso di cancellazione (numero di corse cancellate su numero di corse totali) di tutti gli utenti non bannati [5 punti];

```
Select cc.corse_cancelate/ct.corse_totali, data
FROM (
    SELECT count(*) as corse_totali, data_richiesta as data
    FROM VIAGGI v JOIN utenti u ON u.id_utente=v.id_cliente
    WHERE u.bannato='F' GROUP BY data_richiesta
) AS ct, (
    SELECT count(*) as corse_cancellate, data_richiesta as data
    FROM VIAGGI v JOIN utenti u ON u.id_utente=v.id_cliente
    WHERE u.bannato='F' AND v.stato='cancellato_dal_cliente' GROUP BY data_richiesta
) AS cc
WHERE ct.data=cc.data
```

   c. Trovare i clienti che hanno fatto il numero massimo di viaggi in due giorni consecutivi [5 punti];

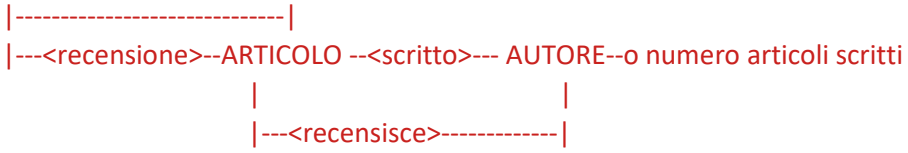
```
CREATE VIEW ViaggiConsecutivi AS SELECT COUNT(V1.id) + COUNT(V2.id) AS vtot, v1.id_cliente
FROM viaggi v1, viaggi v2 WHERE v1.id_cliente=v2.id_cliente AND
v1.data_richiesta<v2.data_richiesta AND v2.data_richiesta – v1.data_richiesta=1
GROUP BY v1.id_cliente

SELECT id_cliente
FROM ViaggiConsecutivi
WHERE vtot = (SELECT MAX(vtot) FROM ViaggiConsecutivi)
```

# Esame Basi di Dati
# 29 marzo 2021

b) Si vuole progettare un database per la memorizzazione di articoli di giornale. Ogni articolo ha un titolo, un contenuto, una descrizione e appartiene ad una categoria (cronaca, gossip, scienze, etc.). Gli articoli sono scritti da autori, per gli autori si tiene traccia del numero di articoli scritti. Questi possono scrivere recensioni anche per articoli scritti da altri autori. Per ogni articolo si tiene traccia del numero di recensioni.

1. Disegnare uno schema con 2 entità e 3 relazioni [3 punti];




2. Estendere lo schema introducendo una gerarchia che consenta un partizionamento degli articoli in base alla categoria [2 punti];
3. Si supponga di avere 100000 articoli e 500 autori. Si considerino le seguenti operazioni
   i. Inserimento nuovo articolo, 1000 volte al giorno;
   ii. Lettura del numero di articoli scritti da un autore 200 al giorno;

   Analizzare le due operazioni e stabilire se conviene mantenere l'attributo "numero articoli scritti" all'interno della tabella autori [4 punti].

   Con ridondanza i) 1S in autore e 1S in articolo = 2S = 4L * 1000v/gg = 4000L/gg
   ii) 1L in autore * 200v/gg = 200L/gg
   Totale: 4200L/gg
   Senza ridondanza i) 1S in articolo = 1S = 2L * 1000v/gg = 2000L/gg
   ii) 1L in scritto * 200 articoli in media ad autore = 200L * 200v/gg = 40000L/gg
   Totale: 42000L/gg

   **Conviene mantenere la ridondanza**

4. Trasformare lo schema sviluppato al punto b in uno schema relazionale [3 punti].

c) Si consideri il seguente schema di relazione R(A,B,C,D,E) F={A→B; C→ A; D→C; E→ A}
   1. Identificare le chiavi dello schema [1 punti];
      Chiave: DE
   2. Decomporre lo schema in BCNF [3 punti].

# Esame Basi di Dati
# 29 marzo 2021

a) Si supponga di avere il seguente schema relazionale:

viaggi(id,id_cliente,id_guidatore,id_citta,stato,data_richiesta)

utenti(id_utente,bannato,ruolo)

dove:

- l'attributo stato è un tipo enumerato che assume i valori "completato", "cancellato_dal_guidatore","cancellato_dal_cliente".

- Bannato assume i valori V/F

- Ruolo assume i valori "Cliente", "Guidatore"

1. Identificare le chiavi primarie ed esterne dello schema [0 punti corretta, -1 sbagliata]
2. Rispondere alle seguenti query in algebra:
   a. Trovare le date per cui esiste almeno un viaggio per tutti i clienti [4 punti];
3. Rispondere alle seguenti query in SQL
   b. Per ogni data, trovare il tasso di cancellazione (numero di corse cancellate su numero di corse totali) di tutti gli utenti non bannati [5 punti];
   c. Trovare i clienti che hanno fatto il numero massimo di viaggi in due giorni consecutivi [5 punti];

b) Si vuole progettare un database per la memorizzazione di articoli di giornale. Ogni articolo ha un titolo, un contenuto, una descrizione e appartiene ad una categoria (cronaca, gossip, scienze, etc.). Gli articoli sono scritti da autori, per gli autori si tiene traccia del numero di articoli scritti. Questi possono scrivere recensioni anche per articoli scritti da altri autori. Per ogni articolo si tiene traccia del numero di recensioni.
   1. Disegnare uno schema con 2 entità e 3 relazioni [3 punti];
   2. Estendere lo schema introducendo una gerarchia che consenta un partizionamento degli articoli in base alla categoria [2 punti];
   3. Si supponga di avere 100000 articoli e 500 autori. Si considerino le seguenti operazioni
      i. Inserimento nuovo articolo, 1000 volte al giorno;
      ii. Lettura del numero di articoli scritti da un autore 200 al giorno;

      Analizzare le due operazioni e stabilire se conviene mantenere l'attributo "numero articoli scritti" all'interno della tabella autori [4 punti].

   4. Trasformare lo schema sviluppato al punto b in uno schema relazionale [3 punti].

c) Si consideri il seguente schema di relazione R(A,B,C,D,E) F={A→B; C→ A; D→C; E→ A}
   1. Identificare le chiavi dello schema [1 punti];
   2. Decomporre lo schema in BCNF [3 punti].

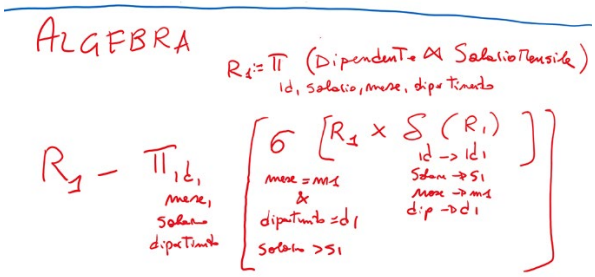
Dipartimento(id,nome)

Dipendente(id,salarioannuale, nome, dipartimento)

SalarioMensileDipendente(id,salario,mese)

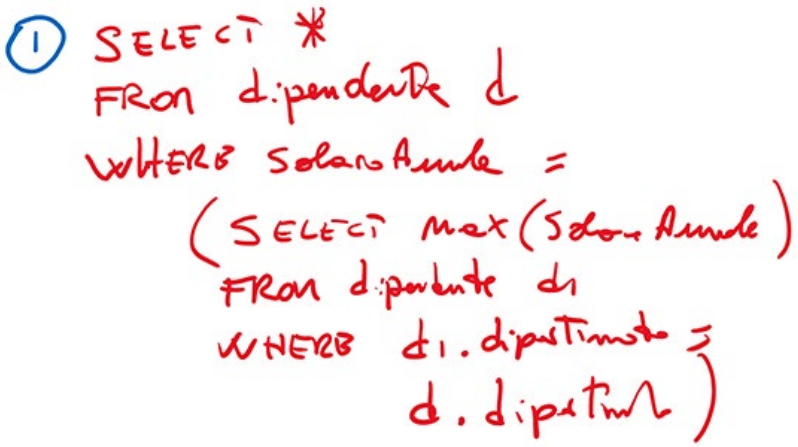
Algebra

Per ogni dipartimento e per ogni mese trovare i dipendenti con il salario minimo (4 punti)

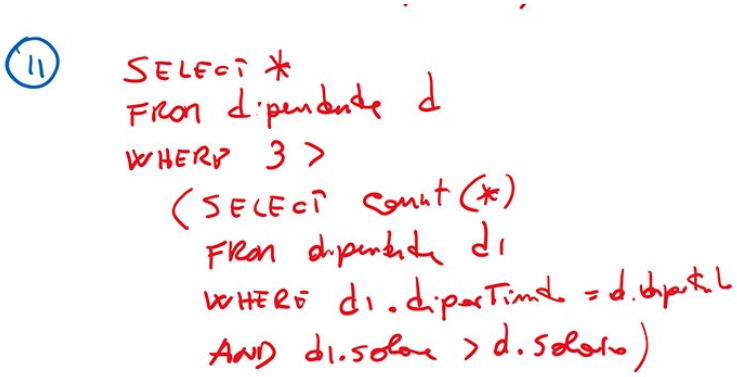



SQL

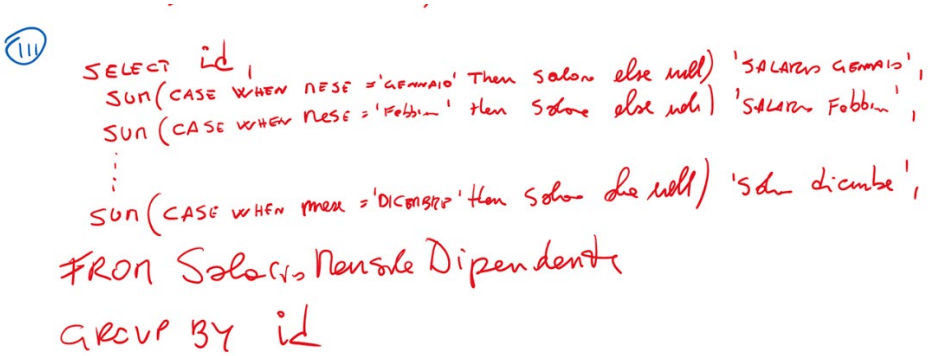
Per ogni dipartimento trovare i dipendenti con il salario annuale massimo, visualizzando nome, salario e dipartimento (2 punti)




Per ogni dipartimento trovare i tre dipendenti con il salario annuale massimo, visualizzando dipartimento, nome dipendente e salario (6 punti).



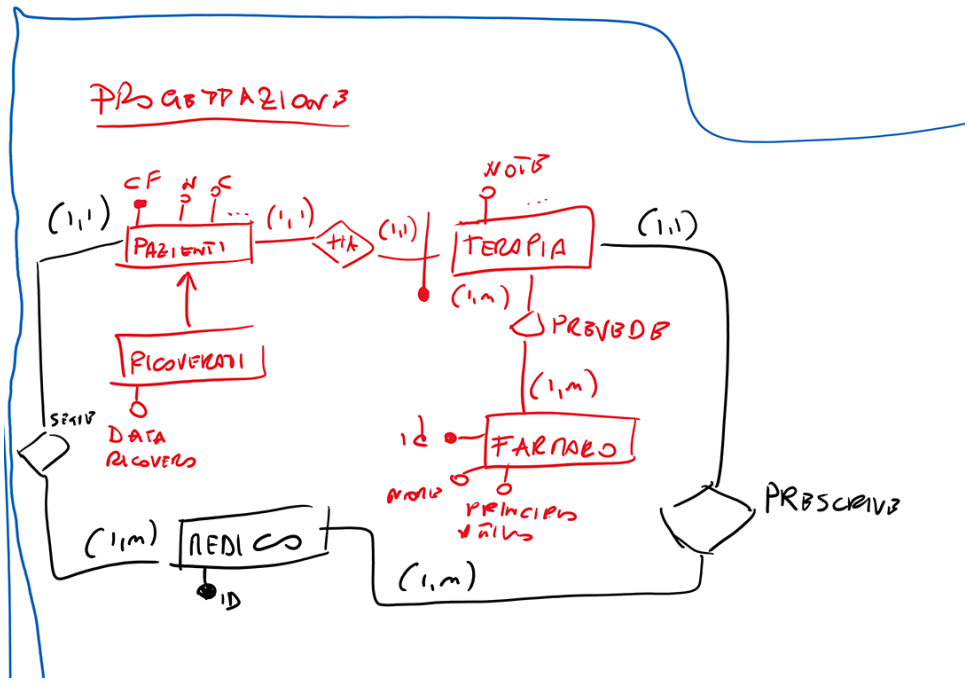
Riformattare la tabella salarioMesileDipendente inserendo le seguenti colonne (Id,gen,feb,mar,apr,mag,giu,lug,ago,set,ott,nov,dic) ogni riga della tabella dovrà contenere la somma dei salari del rispettivo mese  per ogni dipendente(4 punti)




Progettazione.

Implementare uno schema ER per rappresentare un insieme di soggetti con una patologia, evidenziare un sott'insieme di pazienti ricoverati con indicazione della data ricovero, mantenere per tutti i pazienti i dettagli della terapia farmacologica.  (5 punti)

Modificare lo schema per rappresentare anche il medico in relazione al paziente e alla terapia farmacologica. (3 punti)




Normalizzazione

R(A,B,C,D,E,F,G) F={A→BC , C→EG  G→F D->FG}

1. Identificare le chiavi dello schema (2 punti)
2. Decomporlo in BCNF (4 punti)

# Normalizzazione

1) CHIAVI

$A \to B$

$A \to C \to E$

$\quad \to D\,G \to F$

$A \to BCEFG$

$AD^+ = \{ABCDEFG\}$

AD chiave

II) BCNF

$G \to F$

$R_1(G,F)$ $\quad \Pi_{GF}(\{G \to F\}) = \{G \to F\}$

$R_2(A,B,C,D,E,G)$ $\quad \Pi_{A\ldots EG}(\ldots) = \{A \to BC,\ C \to EG,\ D \to G\}$

$R_3(DG)$

$R_4(ABCDE)$ $\quad \{A \to BC \quad C \to E\}$

$R_5(C,E)$

$R_6(ABCD)$ $\quad \{A \to BC\}$

$R_7(ABC)$ $\quad R_8(AD)$

$R_1(G,F)$ $\{G \to F\}$ $\quad R_5(C,E)$ $\{C \to E\}$

$R_3(D,G)$ $\{D \to G\}$ $\quad R_7(ABC)$ $\{A \to B,C\}$

$R_8(AD)$

**CdL in Informatica Triennale – A.A. 2020-2021**
**Basi di Dati**
**Proff. S. Alaimo - A. Pulvirenti**

**Prova scritta 15 settembre 2021**

| Matricola | Nome e Cognome | Valutazione |
|---|---|---|
| | | |

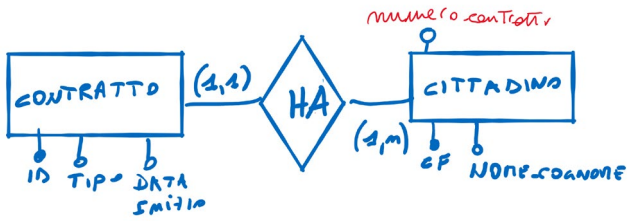
1) Si consideri il seguente schema relazionale:

   candidato(CF, Nome, Cognome)
   prova(Data, ID)
   domanda(numero, IDProva, Testo, RispostaCorretta, puntiDomanda)
   CandidatoProva(CF, IDProva, Esito, punteggio)

   L'attributo Esito assume i valori "Positivo", "Negativo";

   a. Identificare le chiavi primarie ed esterne dello schema [0 punti corretta, -1 errata];
   b. Rispondere alle seguenti query in algebra relazionale:
      1. Identificare i candidati che hanno partecipato a tutte le prove [3 punti];
      2. Identificare i candidati che hanno avuto esito "Positivo" e punteggio minimo [4 punti];
   c. Rispondere alle seguenti query in SQL:
      1. Trovare i candidati che hanno avuto un punteggio superiore a quello medio, , tra quelli che hanno ottenuto un esito positivo [4 punti];
      2. Trovare le prove che sono state superate da un numero minimo di candidati, restituire il punteggio medio ottenuto [4 punti].

2) Si consideri il seguente schema:




   Si supponga di avere le seguenti operazioni:

   1. Inserimento di un nuovo contratto, dato un cittadino, 40 volte al giorno;
   2. Calcolo del numero di contratti di un cittadino, 5 volte al giorno;

Ipotizzando di avere, 10000 cittadini e 30000 contratti, stabilire se conviene mantenere l'attributo, numero contratti [5 punti].

3) Si consideri il seguente schema relazionale R(A,B,C,D,E,F) con F={A→B, B→C, A→C,D→EF}
   1. Identificare le chiavi [2 punti];
   2. Decomporre lo schema in BCNF [4 punti];
   3. Identificare un ricoprimento minimale [4 punti].

Esame

Si consideri il seguente schema relazionale

`Negozio (id, ragionesociale, idcategoria, numero_visite)`

`CategorieNegozi (id, titolo, descrizione)`

`Città (codiceIstat, nome, provincia, regione)`

`PresenzaNegozio (codicecitta, idinegozio, via, cap)`

`Persona (cf, nome, cognome)`

`visitaNegozio (cf, idnegozio, data)`

1. Identificare le chiavi primarie ed esterne dello schema (1 punto)
2. Algebra
   a. Trovare le categorie di negozi presenti in tutte le provincie (3 punti);
   b. Trovare le regioni in cui mancano alcune categorie di negozi (3 punti);
3. SQL
   a. Trovare i CAP e le relative Città che hanno un numero totale di negozi minore di quello medio della relativa regione (5 punti);
   b. Contare per ogni categoria quanti negozi hanno nella ragione sociale la parola “risparmio” (4 punti);
4. Progettazione
   a. Estrapolare uno schema ER che descrive lo schema logico (3 punti)
   b. L’attributo numero_visite della tabella negozio è ridondante. Ipotizzando di avere 10000 negozi e 1000000 di persone, verificare se conviene mantenerlo sulla base delle seguenti operazioni (4 punti)

      O1. Visita di un negozio da parte di una persona, 100 volte al giorno

      O2. Conta il numero di persone che hanno visitato il negozio 2 volte al giorno.

5. Normalizzazione

Dato lo schema `Giocatore (nome, squadra, stadio, campionato, goal, assist)`

| nome | Squadra | Stadio | Campionato | Goal | Assist |
|---|---|---|---|---|---|
| Messi | Barcellona | Camp Nou | Liga spagnola | 10 | 23 |
| C. Ronaldo | Juventus | Delle Alpi | Serie A | 8 | 12 |
| Chiellini | Juventus | Delle Alpi | Serie A | 1 | 11 |
| … | … | … | … | … | … |

   a. Quali anomalie presenta lo schema? Supponiamo che la Juventus vuole fare un nuovo stadio e chiamarlo “Allianz Stadium”. Quali sono i problemi che avrà questo schema? (3 punti)
   b. Proporre una soluzione, tenendo conto delle dipendenze funzionali che possono essere dedotte dallo schema (4 punti).

XML

Supponiamo di avere il seguente DTD

```
<?xml version=”1.0”>

<!DOCTYPE libri [

<!ELEMEMT libro(autore+, titolo, isbn, introduzione, capitoli)>
<!ELEMENT autore(#PCDATA)>
<!ELEMENT titolo(#PCDATA)>
<!ELEMENT isbn(#PCDATA)>
<!ELEMENT introduzione(#PCDATA)>
<!ELEMENT capitoli(capitolo+)>
<!ELEMENT capitolo(numero, titolo, testo)>
<!ELEMENT numero(#PCDATA)>
<!ELEMENT testo(#PCDATA)>

]>
```

1. Dare un file xml d’esempio con 2 libri, il primo con 1 autore, e 2 capitoli, il secondo libro 1 autore e 3 capitoli (2 punti)
2. Rispondere alle seguenti query:
    a. Per ogni libro conteggiare il numero di capitoli
    b. Per ogni autore conteggiare il numero di libri scritti

Itinere 1

1. Si vuole progettare un database per la memorizzazione di prodotti, ogni prodotto è costituito da una serie di componenti. I componenti sono di tre tipologie: legno (per il quale si mantiene il tipo), metallo (per il quale si mantiene il peso), vetro (per il quale si mantiene la tipologia).
    a. Implementare uno schema ER che modella i requisiti dati sopra con 2 entità, una gerarchia (collegata ad una delle due entità) e 1 associazione (2 punti)
    b. Ristrutturare lo schema e tradurlo in modello relazionale (2 punti).

2. Si consideri il seguente schedule:
   r1(x)r2(y)r1(y)r3(x)r1(z)w1(y)w2(z)r4(z)r1(t)w2(t)
    a. Dire se lo schedule è CSR o VSR (3 punti).
    b. Se passato ad un 2PL scheduler si verificheranno dei deadlock (3 punti)?

3. Supponiamo di avere la relazione R (A, B, C, D, E, F, G, H, I, J) con le seguenti dipendenze funzionali F = {A,B→C; B,D→E,F; A,D→G,H; A→I; H→J}
    a. Trovare le chiavi (2 punti)
    b. Decomporre in BCNF dettagliando tutti i passaggi (3 punti)

Dato il seguente file xml:

```
<?xml version="1.0">
<libri>
   <libro categoria="Java">
      <titolo lang="en">Learn Java in 24 Hours</titolo>
      <autore>Marsellus Wallace</autore>
      <anno>2005</anno>
      <prezzo>30.00</prezzo>
   </libro>
   <libro categoria="Scala">
      <titolo lang="en">Learn Scala in 15 minutes</titolo>
      <autore>Harry Potter</autore>
      <anno>2018</anno>
      <prezzo>10.00</prezzo>
   </libro>
…
<libri>
```

1. Generare un file xml che restituisce il numero di libri scritti da ogni autore;
2. Generare un file xml che restituisce per ogni ogni anno il numero di libri scritti per ogni categoria

Itinere 2

1. Si vuole progettare un db per mantenere le vaccinazioni, ci sono le persone che chiedono di essere vaccinate, approvata la richiesta per queste vengono effettuati diversi richiami del vaccino sulla base di quelli previsti.
    a. Progettare uno schema ER con 3 relazioni e 3 associazioni (3 punti)
    b. Tradurre lo schema ER nel modello relazionale (2 punti)
    c. Ipotizzare che il vaccino ha un attributo #numeropersonevaccinate. Ipotizzare di avere 100000 persone e 50 diversi tipi di vaccini. Si considerino le seguenti operazioni

       O1 inserimento di una specifica tipologia di vaccinazione 100 volte al giorno;
       O2 conteggio del numero di persone che hanno ricevuto un particolare vaccino 3 volte al giorno;

       Stabilire sulla base delle informazioni date se conviene mantenere la ridondanza (3 punti).

2. Si supponga di avere il seguente schema relazionale R (A, B, C, D, E) con le seguenti dipendenze funzionali F = {A→C; D→E; B,E→A}
    a. Calcolare tutte le chiavi dello schema (2 punti)
    b. Decomporlo in BCNF dettagliando tutti i passaggi (3 punti)

3. Dato il seguente file xml:

```
<?xml version="1.0">
<libri>
   <libro categoria="Java">
      <titolo lang="en">Learn Java in 24 Hours</titolo>
      <autore>Marsellus Wallace</autore>
      <anno>2005</anno>
      <prezzo>30.00</prezzo>
   </libro>
   <libro categoria="Scala">
      <titolo lang="en">Learn Scala in 15 minutes</titolo>
      <autore>Harry Potter</autore>
      <anno>2018</anno>
      <prezzo>10.00</prezzo>
   </libro>
…
<libri>
```

    a. Generare un file xml che restituisce il numero di libri scritti da ogni autore;
    b. Generare un file xml che restituisce per ogni ogni anno il numero di libri scritti per ogni categoria

```
<?xml version="1.0">
<canzoni>
   <canzone categoria="rock">
      <titolo lang="en"> Strawberry Fields Forever</titolo>
      <gruppo>Beatles</gruppo>
      <autore>Beatles</autore>
      <anno>1967</anno>
   </canzone>
   <canzone categoria="rock">
      <titolo lang="en">Help </titolo>
      <autore>Jack Malik</autore>
      <autore>Beatles</autore>
      <anno>2019</anno>
   </canzone>
…
<canzoni>
```

c. Scrivere una query che restituisce un xml con il numero di canzoni scritte da ogni gruppo;
d. Scrivere una query che restituisce per ogni canzone il numero di gruppi e di autori che l'hanno interpretata

Esame

Si consideri il seguente schema relazionale

Negozio (id, ragionesociale, idcategoria, numero_visite)

CategorieNegozi (id, titolo, descrizione)

Città (codiceIstat, nome, provincia, regione)

PresenzaNegozio (codicecitta, idinegozio, via, cap)

Persona (cf, nome, cognome)

visitaNegozio (cf, idnegozio, data)

1. Identificare le chiavi primarie ed esterne dello schema (1 punto)
2. Algebra
   a. Trovare le categorie di negozi presenti in tutte le provincie (3 punti);

$$\delta_{idcategoria \to id}(\pi_{provincia,idcategoria}(\sigma_{idnegozio=id}(\sigma_{codicecitta=codiceistat}(PresenzaNegozio \times Citta) \times Negozio)) \div \pi_{id}(CategorieNegozi)$$

   b. Trovare le regioni in cui mancano alcune categorie di negozi (3 punti);

$$\pi_{regione}(Citta) - \delta_{idcategoria \to id}(\pi_{regione,idcategoria}(\sigma_{idnegozio=id}(\sigma_{codicecitta=codiceistat}(PresenzaNegozio \times Citta) \times Negozio) \div \pi_{id}(CategorieNegozi)$$

3. SQL
   a. Trovare i CAP e le relative Città che hanno un numero totale di negozi minore di quello medio della relativa regione (5 punti);

```
SELECT *
FROM (  SELECT count(*) numeroNegozi, cap, regione
        FROM PresenzaNegozio, citta
        WHERE codicecitta= codiceIstat
        GROUP BY cap) t
WHERE t.numeronegozi >= (
        SELECT AVG(numeronegozi)
        FROM (     SELECT count(*) numeroNegozi, cap, regione
                   FROM PresenzaNegozio, citta
                   WHERE codicecitta= codiceIstat
                   GROUP BY cap) t1

        WHERE t1.regione =t.regione
```

   b. Contare per ogni categoria quanti negozi hanno nella ragione sociale la parola “risparmio” (4 punti);

```
SELECT count(*), cn.id, titolo
FROM categorieNegozi cn, Negozi n
WHERE cn.id = n.idcategoria AND ragionesociale LIKE
'%risparimio%'
GROUP BY cn.id
```

4. Progettazione
   a. Estrapolare uno schema ER che descrive lo schema logico (3 punti)

b. L'attributo numero_visite della tabella negozio è ridondante. Ipotizzando di avere 10000 negozi e 1000000 di persone, verificare se conviene mantenerlo sulla base delle seguenti operazioni (4 punti)

O1. Visita di un negozio da parte di una persona, 100 volte al giorno

O2. Conta il numero di persone che hanno visitato il negozio 2 volte al giorno.

Soluzione:

In presenza della ridondanza

O1

1 S in Visita negozio, 1 Lettura in Negozio, 1 Scrittura in Negozio.

2S +1L = 5L x 100 = 500

O2

2 L

Totale 502 L

Assenza ridondanza

O1

1S in vistanegozio

1S = 2L → 200

O2 in media ogni persona visita 100 negozi

100 L → 200

400 L

Non conviene mantenere la ridondanza.

5. Normalizzazione

Dato lo schema `Giocatore (nome, squadra, stadio, campionato, goal, assist)`

| nome | Squadra | Stadio | Campionato | Goal | Assist |
|---|---|---|---|---|---|
| Messi | Barcellona | Camp Nou | Liga spagnola | 10 | 23 |
| C. Ronaldo | Juventus | Delle Alpi | Serie A | 8 | 12 |
| Chiellini | Juventus | Delle Alpi | Serie A | 1 | 11 |
| ... | … | … | ... | … | … |

a. Quali anomalie presenta lo schema? Supponiamo che la Juventus vuole fare un nuovo stadio e chiamarlo "Allianz Stadium". Quali sono i problemi che avrà questo schema? (3 punti)

Anomalia di cancellazione e aggiornamento.

b. Proporre una soluzione, tenendo conto delle dipendenze funzionali che possono essere dedotte dallo schema (4 punti).

Squadra(stadio,squadra)

Giocatore(squadra,giocatore)

Campionato(campionato,giocatore

XML

Supponiamo di avere il seguente DTD

<?xml version=”1.0”>

<!DOCTYPE libri [

<!ELEMEMT libro(autore+, titolo, isbn, introduzione, capitoli)>
<!ELEMENT autore(#PCDATA)>
<!ELEMENT titolo(#PCDATA)>
<!ELEMENT isbn(#PCDATA)>
<!ELEMENT introduzione(#PCDATA)>
<!ELEMENT capitoli(capitolo+)>
<!ELEMENT capitolo(numero, titolo, testo)>
<!ELEMENT numero(#PCDATA)>
<!ELEMENT testo(#PCDATA)>

]>

1. Dare un file xml d’esempio con 2 libri, il primo con 1 autore, e 2 capitoli, il secondo libro 1 autore e 3 capitoli (2 punti)
2. Rispondere alle seguenti query:
    a. Per ogni libro conteggiare il numero di capitoli
    b. Per ogni autore conteggiare il numero di libri scritti

Itinere 1

1. Si vuole progettare un database per la memorizzazione di prodotti, ogni prodotto è costituito da una serie di componenti. I componenti sono di tre tipologie: legno (per il quale si mantiene il tipo), metallo (per il quale si mantiene il peso), vetro (per il quale si mantiene la tipologia).
    a. Implementare uno schema ER che modella i requisiti dati sopra con 2 entità, una gerarchia (collegata ad una delle due entità) e 1 associazione (2 punti)
    b. Ristrutturare lo schema e tradurlo in modello relazionale (2 punti).

2. Si consideri il seguente schedule:
   r1(x)r2(y)r1(y)r3(x)r1(z)w1(y)w2(z)r4(z)r1(t)w2(t)
    a. Dire se lo schedule è CSR o VSR (3 punti).
    b. Se passato ad un 2PL scheduler si verificheranno dei deadlock (3 punti)?

3. Supponiamo di avere la relazione R (A, B, C, D, E, F, G, H, I, J) con le seguenti dipendenze funzionali F = {A,B→C; B,D→E,F; A,D→G,H; A→I; H→J}
    a. Trovare le chiavi (2 punti)
    b. Decomporre in BCNF dettagliando tutti i passaggi (3 punti)

Dato il seguente file xml:

```
<?xml version="1.0">
<libri>
   <libro categoria="Java">
      <titolo lang="en">Learn Java in 24 Hours</titolo>
      <autore>Marsellus Wallace</autore>
      <anno>2005</anno>
      <prezzo>30.00</prezzo>
   </libro>
   <libro categoria="Scala">
      <titolo lang="en">Learn Scala in 15 minutes</titolo>
      <autore>Harry Potter</autore>
      <anno>2018</anno>
      <prezzo>10.00</prezzo>
   </libro>
…
<libri>
```

1. Generare un file xml che restituisce il numero di libri scritti da ogni autore;
2. Generare un file xml che restituisce per ogni ogni anno il numero di libri scritti per ogni categoria

Itinere 2

1. Si vuole progettare un db per mantenere le vaccinazioni, ci sono le persone che chiedono di essere vaccinate, approvata la richiesta per queste vengono effettuati diversi richiami del vaccino sulla base di quelli previsti.
    a. Progettare uno schema ER con 3 relazioni e 3 associazioni (3 punti)
    b. Tradurre lo schema ER nel modello relazionale (2 punti)
    c. Ipotizzare che il vaccino abbia un attributo #numeropersonevaccinate. Ipotizzare di avere 100000 persone e 50 diversi tipi di vaccini. Si considerino le seguenti operazioni

       O1 inserimento di una specifica tipologia di vaccinazione 100 volte al giorno;
       O2 conteggio del numero di persone che hanno ricevuto un particolare vaccino 3 volte al giorno;

       Stabilire sulla base delle informazioni date se conviene mantenere la ridondanza (3 punti).

2. Si supponga di avere il seguente schema relazionale R (A, B, C, D, E) con le seguenti dipendenze funzionali F = {A→C; D→E; B,E→A}
    a. Calcolare tutte le chiavi dello schema (2 punti)
    b. Decomporlo in BCNF dettagliando tutti i passaggi (3 punti)

3. Dato il seguente file xml:

```
<?xml version="1.0">
<libri>
   <libro categoria="Java">
      <titolo lang="en">Learn Java in 24 Hours</titolo>
      <autore>Marsellus Wallace</autore>
      <anno>2005</anno>
      <prezzo>30.00</prezzo>
   </libro>
   <libro categoria="Scala">
      <titolo lang="en">Learn Scala in 15 minutes</titolo>
      <autore>Harry Potter</autore>
      <anno>2018</anno>
      <prezzo>10.00</prezzo>
   </libro>
…
<libri>
```

    a. Generare un file xml che restituisce il numero di libri scritti da ogni autore (1 punto);
    b. Generare un file xml che restituisce per ogni anno il numero di libri scritti per ogni categoria (2 punti).

```
<?xml version="1.0">
<canzoni>
   <canzone categoria="rock">
      <titolo lang="en"> Strawberry Fields Forever</titolo>
      <gruppo>Beatles</gruppo>
      <autore>Beatles</autore>
      <anno>1967</anno>
   </canzone>
   <canzone categoria="rock">
      <titolo lang="en">Help </titolo>
      <autore>Jack Malik</autore>
      <autore>Beatles</autore>
      <anno>2019</anno>
   </canzone>
…
<canzoni>
```

c. Scrivere una query che restituisce un xml con il numero di canzoni scritte da ogni gruppo;
d. Scrivere una query che restituisce per ogni canzone il numero di gruppi e di autori che l'hanno interpretata

**CdL in Informatica Triennale – A.A. 2020-2021**
**Basi di Dati**
**Proff. S. Alaimo - A. Pulvirenti**

**Prova scritta 28 settembre 2021**

| Matricola | Nome e Cognome | Valutazione |
|---|---|---|
| | | |

1) Si consideri il seguente schema relazionale relativo alla gestione di uno studio dentistico:
   Paziente(cf, nome, cognome)
   Interventi(id, nome, descrizione, costo_complessivo_previsto)
   InterventoPaziente(id, cf, idintervento, numero_seduta, idmedico, descrizione, data, costo_seduta)
   AllegatiInterventoPaziente(idintervento, numero_seduta, descrizione, allegato)
   DentistaOdontoiatra(id, nome, cognome, specializzazione)
   Fattura(id, cf, intervento, data, numero_sedute, costo_totale)

Ogni intervento ha un costo complessivo previsto dal tariffario. Un intervento può richiedere più sedute e questo viene identificato dal numero (progressivo), l'id dell'intervento resta lo stesso. L'attributo specializzazione può assumere i valori: "dentista", "odontoiatra". Il costo totale dell'intervento in fattura può differire da quello previsto.

   a. Identificare le chiavi primarie ed esterne dello schema [0 punti corretta, -1 errata];

   b. Rispondere alle seguenti query in algebra relazionale:
      1. Identificare i pazienti che hanno avuto sedute con tutti gli odontoiatri [3 punti];
      2. Identificare i pazienti che hanno avuto interventi con un numero massimo di sedute, visualizzare il nome del paziente e la tipologia di intervento [4 punti];

   c. Rispondere alle seguenti query in SQL:
      1. Per ogni paziente visualizzare il costo complessivo degli interventi che hanno richiesto più sedute della media [4 punti];
      2. Trovare gli interventi che hanno un costo_totale medio inferiore a quello complessivo previsto, indicare per questi anche il numero medio di sedute effettuate [5 punti].

   d. Produrre uno schema ER a partire dallo schema logico [2 punti]

2) Si consideri il seguente schedule r1(x),r2(x),w2(y),r3(y),w4(x),r3(z),w4(z),r1(y),w1(y),w5(z),w4(z)
   1. Stabilire se lo schedule è CSR o VSR [4 punti];
   2. Se lo schedule è passato ad uno scheduler 2PL stabilire se ci saranno dei deadlock [3 punti].

3) A partire dallo schema relazionale del punto 1:
      1. definire un DTD dove per ogni paziente si rappresenta il nome, gli interventi ricevuti, e per ogni intervento il dettaglio della seduta [3 punti]
      2. Implementare una query in xquery per identificare i pazienti che hanno avuto interventi da parte di "odontoiatri".