

Basi di Dati
Prof. Alfredo Pulvirenti
A.A. 2015-2016
Prova in itinere A – 14 dicembre 2015

Si consideri il seguente schema relazionale relativo ad una compagnia telefonica.

CLIENTE(ID, Nome, Cognome, email)

TIPOLOGIECONTRATTI(codice, Nome, Descrizione)

CONTRATTICLIENTI(cliente, contratto, data_inizio, data_fine)

TARIFFEORARIECONTRATTI(id, tipologia_chiamata, tariffaSecondo, ora_inizio_tariffa, ora_fine_tariffa, idcontratto)

CHIAMATA(clienteChiamante, clienteRicevente, data, ora_inizio, durata)

L'attributo tipologia_chiamata può assumere i seguenti valori {"urbana", "interurbana", "internazionale"}.

Basi di Dati
Prof. Alfredo Pulvirenti
A.A. 2015-2016
Prova in itinere A – 14 dicembre 2015

1. Identificare le chiavi primarie ed esterne [1 punto].

CLIENTE(ID, Nome, Cognome, email)

TIPOLOGIECONTRATTI(codice, Nome, Descrizione)

CONTRATTICLIENTI(cliente, contratto, data_inizio, data_fine)

TARIFFEORARIECONTRATTI(id, tipologia_chiamata, tariffaSecondo, ora_inizio_tariffa, ora_fine_tariffa, idcontratto)

CHIAMATA(clienteChiamante, clienteRicevente, data, ora_inizio, durata)

2. Rispondere alle seguenti query in algebra relazionale:

- a. Trovare per ogni tipologia di contratto, mostrando codice e nome, la tariffa oraria più bassa [2 punti];

$T1 := \text{TARIFFEORARIECONTRATTI}$

$T2 := T1$

$T3$

$:= \pi_{T2.idcontratto, T2, tariffasecondo}(T2)$

$- \pi_{T2.idcontratto, T2, tariffasecondo}(\sigma_{T1.idcontratto=T2.idcontratto \text{ AND } (T1 \times T2))$
 $T1.tariffasecondo < T2.tariffasecondo$

$R := \pi_{codice, nome}(T3 \bowtie_{idcontratto=codice} \text{TIPLOGIECONTRATTI})$

- b. Trovare i clienti, mostrando nome e cognome, che non hanno mai effettuato o ricevuto chiamate in alcune fasce orarie [2 punti];

$T1 := \text{TARIFFEORARIECONTRATTI}$

$C := \text{CHIAMATA}$

$T2 := \sigma_{(oit \leq oi \text{ AND } oft > oi) \text{ OR } (oit \leq oi + durata \text{ AND } oft > oi + durata) \text{ OR } (oi < oit \text{ AND } oi + durata > oft)}(C \times T1)$

$T3 := \delta_{id \leftarrow clientechiamante, oit, oft}(\pi_{clientechiamante, oit, oft}(T2))$
 $\cup \delta_{id \leftarrow clientericevente, oit, oft}(\pi_{clientericevente, oit, oft}(T2))$

$R := \pi_{id}(\pi_{id, oit, oft}(CLIENTE \times T1) - T3)$

- c. Trovare i clienti, mostrando nome e cognome, che hanno sottoscritto tutte le tipologie di contratti [4 punti];

$T1 := \delta_{contratto \leftarrow codice} \pi_{codice}(\text{TIPLOGIECONTRATTO})$

$R := \pi_{cliente, contratto}(\text{CONTRATTICLIENTI}) \div T1$

- d. Trovare i clienti che hanno sottoscritto le stesse tipologie di contratti in intervalli temporali che si sovrappongono [3 punti].

$CC := \text{CONTRATTICLIENTI}$

$T1 := CC \sigma_{cc.cliente \neq cc1.cliente \text{ AND } (CC \times CC1)}(CC \times CC1)$
 $cc.contratt = cc1.contratto \text{ AND } [(cc.di \leq cc1.di \text{ and } cc.df \geq cc1.df) \text{ OR } (cc1.di \leq cc.di \text{ AND } cc1.di \leq cc.df) \text{ OR } (cc.di \leq cc1.df \text{ AND } cc.df \geq cc1.df) \text{ OR } (cc1.di \leq cc.df \text{ AND } cc1.df \geq cc.df)]$

$R := \pi_{cc.cliente, cc1.cliente}(T2)$

3. Rispondere alle seguenti query in SQL

- a. Per ogni tipologia di contratto indicare i costi delle tariffe orarie interurbane dopo le 20.00 [2 punti];

SELECT *

FROM tariffeorariecontratti

WHERE tipologia_chiamata = "interurbana" **AND**

Ora_inizio_tariffa >= 20.00

Basi di Dati
Prof. Alfredo Pulvirenti
A.A. 2015-2016
Prova in itinere A – 14 dicembre 2015

- b. Trovare il cliente, mostrando nome e cognome, che è stato più tempo al telefono, come chiamante o come ricevente [2 punti];

```
CREATE VIEW CM(id,totale) AS
    SELECT clientechiamante as id, SUM(durata) as totale
    FROM chiamata
    GROUP BY clientechiamante
    HAVING SUM(durata) >=ALL (SELECT SUM(durata)
                                FROM chiamata
                                GROUP BY clientechiamante)

    UNION
    SELECT clientericevente as id, SUM(durata) as totale
    FROM chiamata
    GROUP BY clientericevente
    HAVING SUM(durata) >=ALL (SELECT SUM(durata)
                                FROM chiamata
                                GROUP BY clientericevente)

SELECT id
FROM CM
WHERE totale = (SELECT MAX(totale) FROM CM)
```

- c. Trovare il cliente, mostrando nome e cognome, che non ha mai fatto chiamate nella fascia oraria più conveniente del suo contratto [3 punti];

```
CREATE VIEW BESTTOC AS
    SELECT *
    FROM tarifferorariecontratto t
    WHERE tariffaSecondo <= ALL (SELECT tariffaSecondo FROM tariffeorariecontratto t1
                                WHERE t1.idcontratto = t.idcontratto )

    SELECT cli.Nome, cli.Cognome
    FROM CONTRATTICLIENTI cc, CHIAMATA c, CLIENTE cli
    WHERE Cliente = clientechiamante AND cli.ID = Cliente AND
        NOT EXISTS (SELECT * FROM BESTTOC
                    WHERE cc.contratto = idcontratto AND
                        ((oit <= c.oi AND oft >= c.oi+durata) OR
                         (oit < c.oi+c.durata AND oft > c.oi+c.durata) OR
                         (c.oi < oit AND c.oi+c.durata > oft))
```

Basi di Dati
Prof. Alfredo Pulvirenti
A.A. 2015-2016
Prova in itinere A – 14 dicembre 2015

- d. Trovare le coppie di persone che non si sono mai chiamate o che non stanno nella chiusura transitiva del grafo delle chiamate [4 punti].

```
SELECT c1.id as CC, c2.id as CR
FROM CLIENTE C1, CLIENTE c2
WHERE c1.id <> c2.id
EXCEPT
(WITH RECURSIVE call(CC, CR) (
    SELECT clientechiamante as CC, clientericevente as CC
    FROM chiamate
    UNION ALL
    SELECT clientechiamante as CC, CR
    FROM chiamate, call
    WHERE clientericevente = cc)
SELECT * FROM call)
```

4. Definire un vincolo di integrità in SQL che consenta di evitare inserimenti in CHIAMATA con clienteChiamante = ClienteRicevente [3 punti].

```
CHECK clientechiamante <> clientericevente
```

Basi di Dati
Prof. Alfredo Pulvirenti
A.A. 2015-2016
Prova in itinere A – 14 dicembre 2015

5. Si supponga di avere inoltre la seguente relazione:

BOLLETTA(id, idcli,data_inizio_periodo_rif,data_fine_periodo_rif,importoBolletta)

Implementare un trigger che dopo aver aggiornato il campo “durata” della relazione CHIAMATA aggiorna la bolletta del relativo cliente sommando il costo della chiamata all’importo della bolletta [4 punti].

```
CREATE TRIGGER UPB
AFTER UPDATE OF durata ON chiamata
FOR EACH ROW
CREATE CURSOR tariffazione AS
SELECT oit, ofc, ts, oi, oi+durata AS durataChiamata
FROM contratticliente cc, tariffeorariecontratti toc
WHERE cc.cliente = new.clientechiamante AND
      Toc.idcontratto = cc.contratto AND
      [(oit <= oi AND ofc > oi) OR
       (oit <= oi+durata AND ofc > oi+durata) OR
       (oi<oit AND oi+durata = ofc)]
BEGIN
  DECLARE totale integer;
  OPEN tariffazione
  LOOP
    FETCH tariffazione INTO t
    EXIT WHEN tariffazione%NOTFOUND
    Totale = totale + [MIN(t.ofc, t.durataChiamata) -
                      MAX(t.oit, t.oi)] * t.ts
  END LOOP
  UPDATE bolletta SET importobolletta += totale
  WHERE idcli = new.clientechiamante
  AND DIP <= new.data AND DFT > new.data
END;
```

Basi di Dati
Prof. Alfredo Pulvirenti
A.A. 2015-2016
Prova in itinere A – 14 dicembre 2015

Basi di Dati
Prof. Alfredo Pulvirenti

A.A. 2015-2016
Prova in itinere B – 14 dicembre 2015

Si consideri il seguente schema relazionale relativo ad gestionale per una scuola materna.

MAESTRA(id, nome, cognome)
CLASSE(id, nome, descrizione)
BAMBINO(id, nome, cognome, eta)
APPARTENENZA(bambino, classe, anno_scolastico)
MAESTRECLASSI(classe, maestra, anno_scolastico)
PRESENZE(bambino, data, presenza, pasti, tempo_pieno)

Dove all'interno della tabella PRESENZE gli attributi presenza, pasti e tempo_pieno sono tre booleani che indicano rispettivamente che il bambino era presente o assente, che ha usufruito del servizio mensa o che ha usufruito del tempo pieno.

1. Indicare le chiavi primarie ed esterne dello schema relazionale [1 punto].
2. Risolvere le seguenti query in algebra relazionale
 - a. Trovare per ogni classe il bambino più piccolo nell'anno scolastico 2015, mostrando il nome della classe e il nome del bambino [2 punti];
 - b. Trovare le maestre, mostrando nome e cognome, che non hanno mai insegnato in alcune classi [2 punti];
 - c. Trovare i bambini, mostrando nome e cognome, che hanno frequentato tutte le classi [4 punti];
 - d. Trovare le coppie di bambini che frequentano le stesse classi nello stesso anno scolastico [3 punti].
3. Rispondere alle seguenti query in SQL
 - a. Per ogni bambino restituire il numero di presenze fatte ogni mese nell'ultimo anno scolastico [2 punti].
 - b. Trovare il bambino che ha fatto più assenze in ogni classe nell'anno scolastico 2015 [2 punti];
 - c. Trovare i bambini che hanno avuto tutte le maestre [4 punti];
 - d. Trovare le coppie di bambini sono stati compagni di classe in ogni anno scolastico [3 punti]
4. Implementare un vincolo di integrità che eviti di mettere TRUE il campo pasti o tempo_pieno quando il bambino è assente [3 punti].
5. Supponendo di avere la seguente relazione

FATTURA(nFattura, bambino, Data, costo)

Implementare un trigger che dopo aver inserito un record relativo a una fattura nella tabella FATTURA aggiorna lo stesso record della tabella calcolando il costo mensile come 150 euro più il numero di pasti * 3 più il numero di tempo_pieno * 5 [4 punti].