

Basi di dati I — Prova di autovalutazione — 1 novembre 2016

Soluzioni

Domanda 1 Si consideri una base di dati sulle relazioni

- $R_1(\underline{A}, B, C)$
- $R_2(\underline{D}, \underline{E}, F)$

Scrivere interrogazioni in SQL equivalenti alle seguenti espressioni dell'algebra relazionale:

1. $\pi_{BC}(\sigma_{C>10}(R_1))$

```
SELECT DISTINCT B, C
FROM R1
WHERE C > 10
```

2. $\pi_B(R_1 \bowtie_{C=D} \sigma_{F=2}(R_2))$

```
SELECT DISTINCT B
FROM R1 JOIN R2 ON C = D
WHERE F = 2
```

Domanda 2 Con riferimento alla base di dati nella domanda 1 scrivere espressioni dell'algebra relazionale equivalenti alle seguenti interrogazioni SQL

1. SELECT DISTINCT A , B
FROM R1, R2
WHERE C = D AND E > 100

$$\pi_{AB}(R_1 \bowtie_{C=D} \sigma_{E>100}(R_2))$$

2. SELECT DISTINCT A , B
FROM R1 X1
WHERE NOT EXISTS
(SELECT *
FROM R1 Y1, R2
WHERE Y1.C = D AND X1.A = Y1.A AND F>10)

$$\pi_{AB}(R_1) - \pi_{AB}(R_1 \bowtie_{C=D} \sigma_{F>10}(R_2))$$

Nota bene, se la chiave di R_1 fosse stata AB , allora l'interrogazione sarebbe stata più complessa:

$$\pi_{AB}(\pi_A(R_1) - \pi_A(R_1 \bowtie_{C=D} \sigma_{F>10}(R_2))) \bowtie_{A=A'} \rho_{A' \leftarrow A}(R_1)$$

Domanda 3 Ancora con riferimento alla base di dati nella domanda 1, indicare, per ciascuna delle seguenti interrogazioni, se la parola chiave DISTINCT è necessaria

1. l'interrogazione 1 nella domanda 2 *Risposta:* sì
2. l'interrogazione 2 nella domanda 2 *Risposta:* no
3. SELECT DISTINCT A , B
FROM R1, R2
WHERE B = D AND C = E *Risposta:* no
4. SELECT DISTINCT B , C
FROM R1, R2
WHERE B = D AND C = E *Risposta:* sì

Domanda 4 Considerare la base di dati relazionale definita per mezzo delle seguenti istruzioni:

```
create table studenti (matricola numeric not null primary key,
                      cognome char(20) not null,
                      nome char(20) not null,
                      età numeric);
create table corsi (codice numeric not null primary key,
                  titolo char(20) not null,
                  CFU numeric not null);
create table esami (corso numeric not null references corsi(codice),
                  studente numeric not null references studenti(matricola),
                  data date not null,
                  voto numeric not null,
                  primary key (corso, studente));
```

Supponendo che le relative relazioni abbiano rispettivamente le cardinalità $S = 10.000$ (studenti), $C = 1.000$ (corsi) e $E = 40.000$ (esami), indicare le cardinalità minime e massime (in simboli e numeri) dei risultati delle seguenti interrogazioni:

	Min (simboli)	Max (simboli)	Min (valore)	Max (valore)
SELECT matricola, codice FROM studenti, corsi	$S \times C$	$S \times C$	10.000.000	10.000.000
SELECT * FROM studenti, esami WHERE matricola = studente	E	E	40.000	40.000
SELECT matricola, codice FROM studenti, esami, corsi WHERE matricola = studente AND corso = codice	E	E	40.000	40.000

Domanda 5 Con riferimento alla base di dati usata nella domanda precedente formulare le seguenti interrogazioni in algebra relazionale:

1. trovare matricole e cognomi degli studenti che hanno preso almeno un trenta

$$\pi_{matricola, cognome}(\text{studenti} \bowtie_{matricola=studente} \sigma_{voto=30}(\text{esami}))$$

2. trovare le matricole degli studenti che hanno sostenuto almeno due esami

$$\pi_{studente}(\sigma_{corso \neq corso'}(\text{esami} \bowtie_{studente=studente'} \rho_{X \leftarrow X'}(\text{esami})))$$

Domanda 6 Con riferimento alla base di dati usata nelle domande precedenti, formulare le seguenti interrogazioni in SQL

1. trovare codici e titoli di corsi nei cui esami è stato assegnato almeno un trenta

```
SELECT DISTINCT codice, titolo
FROM corsi JOIN esami ON codice = corso
WHERE voto = 30
```

2. trovare le coppie di studenti (mostrare le sole matricole) per i quali uno dei due ha riportato un voto più alto di quello riportato dall'altro in tutti gli esami superati da entrambi.

```
SELECT e1.studente, e2.studente
FROM esami e1, esami e2
WHERE e1.voto > e2.voto
AND e1.studente <> e2.studente
AND e1.corso = e2.corso
AND NOT EXISTS (
    SELECT *
    FROM esami e3, esami e4
    WHERE e3.corso = e4.corso
    AND e3.studente = e1.studente
    AND e4.studente = e2.studente
    AND e3.voto <= e4.voto )
```

3. trovare lo studente con la media più alta; mostrare i dati dello studente, la media in questione e il numero di esami superati

```
CREATE VIEW MediaVoti AS SELECT studente, AVG(voto) AS media, COUNT(*) AS numEsami
FROM esami
GROUP BY studente
```

```
SELECT studente, media, numEsami
FROM MediaVoti, studenti
WHERE studente = matricola
AND media = (SELECT MAX(media)
             FROM MediaVoti)
```

oppure

```
SELECT matricola, cognome, nome, AVG(voto), COUNT(*)
FROM esami join studenti on studente = matricola
GROUP BY matricola, cognome, nome
HAVING AVG(voto) >= ALL
    (SELECT AVG(voto)
     FROM esami
     GROUP BY studente)
```

Domanda 7 Considerare lo schema relazionale seguente, che descrive un insieme di utenze telefoniche (ognuna delle quali si trova in un distretto, identificato dal prefisso; ogni distretto, a sua volta, è in una provincia), con le bollette e i pagamenti relativi ad esse (si noti che i pagamenti fanno riferimento alle bollette, supponendo quindi che si tratti di pagamenti dell'intero importo):

- UTENZE(Prefisso, Numero, Categoria)
con vincolo di integrità referenziale
fra *Prefisso* e la relazione *distretti*
- DISTRETTI(Prefisso, Nome, Provincia)
con vincolo di integrità referenziale
fra *Provincia* e la relazione *province*
- PROVINCE(Sigla, Denominazione)
- BOLLETTE(CodiceBolletta, Prefisso, Numero, DataEmissione, Importo)
con vincolo di integrità referenziale
fra gli attributi *Prefisso*, *Numero* e la relazione *utenze*
- PAGAMENTI(CodicePagamento, Bolletta)
con vincolo di integrità referenziale
fra *Bolletta* e la relazione *bollette*

Formulare in SQL le interrogazioni seguenti

1. Produrre lo stesso risultato della seguente interrogazione in algebra relazionale

$$\rho_{\text{Provincia} \leftarrow \text{Denominazione}}(\pi_{\text{Prefisso, Denominazione}}(\text{DISTRETTI} \bowtie_{\text{Provincia}=\text{Sigla}} \text{PROVINCE}))$$

Possibile soluzione:

```
SELECT DISTINCT prefisso, denominazione AS provincia
FROM distretti JOIN province ON (provincia = sigla)
```

2. Trovare tutte le utenze della provincia che ha “Caserta” come denominazione

Possibile soluzione:

```
SELECT u.*
FROM utenze u JOIN distretti d ON (u.prefisso = d.prefisso)
      JOIN province ON provincia = sigla
WHERE denominazione = 'Caserta'
```

3. Trovare, per ogni utenza, la somma degli importi delle relative bollette

Possibile soluzione:

```
SELECT prefisso, numero, sum(importo)
FROM bollette
GROUP BY prefisso, numero
```

Basi di dati I — Prova di autovalutazione — 1 novembre 2016 — Soluzioni

4. Trovare, per ogni utenza, la somma degli importi delle bollette pagate

Possibile soluzione:

```
SELECT prefisso, numero, SUM(importo)
FROM bollette JOIN pagamenti ON codicebolletta = bolletta
GROUP BY prefisso, numero
```

5. Trovare le utenze per le quali c'è almeno una bolletta non pagata, mostrando prefisso e numero telefonico

Possibile soluzione:

```
SELECT DISTINCT prefisso, numero
FROM bollette
WHERE codicebolletta NOT IN (SELECT bolletta FROM pagamenti)
```

6. Trovare le utenze per le quali c'è almeno una bolletta non pagata, mostrando, per ciascuna di esse, oltre a prefisso e numero telefonico, anche il totale complessivo degli importi delle bollette e il totale di quelle non pagate

Possibile soluzione:

```
CREATE VIEW utenzecontotali AS
  SELECT prefisso, numero, SUM(importo) AS totale
  FROM bollette
  GROUP BY prefisso, numero;

CREATE VIEW utenzeconbollettenonpagate AS
  SELECT prefisso, numero, SUM(importo) AS totaledapagare
  FROM bollette
  WHERE codicebolletta NOT IN (SELECT bolletta FROM pagamenti)
  GROUP BY prefisso, numero;

SELECT u.prefisso, u.numero, totale, totaledapagare
FROM utenzecontotali u JOIN utenzeconbollettenonpagate n
  ON u.prefisso=n.prefisso AND u.numero=n.numero
```

