

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

La prova verrà discussa in aula, prevedibilmente giovedì 9 novembre. Si consiglia di svolgere gli esercizi “simulando l’esame,” sulla carta e senza ausilio di libri e appunti. Si consiglia poi di eseguire le interrogazioni SQL su un DBMS. Gli studenti interessati a sostenere le prove parziali **debbono** consegnare su Moodle le soluzioni e i risultati dei test (mostrando anche le basi di dati di esempio utilizzate).

**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.  $\pi_{BC}(\sigma_{C>10}(R_1))$
2.  $\pi_B(R_1 \bowtie_{C=D} \sigma_{F=2}(R_2))$

**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
```
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)
```

**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
2. l’interrogazione 2 nella domanda 2
3. 

```
SELECT DISTINCT A , B
FROM R1, R2
WHERE B = D AND C = E
```
4. 

```
SELECT DISTINCT B , C
FROM R1, R2
WHERE B = D AND C = E
```

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

**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				
SELECT * FROM studenti, esami WHERE matricola = studente				
SELECT matricola, codice FROM studenti, esami, corsi WHERE matricola = studente AND corso = codice				

**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
2. trovare le matricole degli studenti che hanno sostenuto almeno due 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
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.
3. trovare lo studente con la media più alta; mostrare i dati dello studente, la media in questione e il numero di esami superati

**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

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

2. Trovare tutte le utenze della provincia che ha “Caserta” come denominazione
3. Trovare, per ogni utenza, la somma degli importi delle relative bollette
4. Trovare, per ogni utenza, la somma degli importi delle bollette pagate
5. Trovare le utenze per le quali c'è almeno una bolletta non pagata, mostrando prefisso e numero telefonico
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
7. Trovare l'utenza con il massimo valore di importi non pagati