

Cognome e nome: \_\_\_\_\_ Matricola: \_\_\_\_\_ Turno: \_\_\_\_\_

Riportare sui fogli i seguenti dati: cognome, nome, matricola e turno di laboratorio.

## Esame di SQL

Punteggi massimi:

- Domande 1 e 2 svolte perfettamente: 23;
- Domande 1 e 3 svolte perfettamente: 25;
- Domande 2 e 3 svolte perfettamente: 28;
- Domande 1, 2 e 3 svolte perfettamente: 33.

Lo svolgimento corretto di una sola domanda non permette il raggiungimento della sufficienza.

Le seguenti relazioni definiscono una base di dati “AllYouCanFit” per gestire esercizi in palestra. Gli attributi sottolineati sono le chiavi primarie delle relazioni.

UTENTE(Nome, Cognome, Eta, Sesso)

VOCESCHEDA(Nome, Cognome, Ordine, Esercizio, NumRipetizioni, Durata)

ESERCIZIO(Nome, Tipo, Attrezzo\*)

Vincoli di integrità referenziale:

VOCESCHEDA(Nome, Cognome) referencia UTENTE(Nome, Cognome),

VOCESCHEDA(Esercizio) referencia ESERCIZIO(Nome).

VOCESCHEDA contiene le voci della scheda dell'utente con i vari esercizi, Ordine indica l'ordine in cui l'esercizio deve essere eseguito (1, 2, 3, ...) nella scheda, Tipo può assumere i valori aerobico, posturale, tonicità; Attrezzo può assumere i valori cyclette, manubrio, elastico ecc. e può essere NULL. I rimanenti attributi sono autoesplicativi.

Con riferimento alla base di dati AllYouCanFit esprimere in SQL le seguenti interrogazioni (**commentare ogni soluzione proposta spiegando le varie parti della query**).

### Domanda 1 (bassa complessità).

Elencare, senza duplicati, i tipi di esercizi fatti da maschi che non hanno nessuna vocale nel cognome.

#### Soluzione 1.

```
select distinct e.tipo
from utente u join vocescheda v on u.nome=v.nome and u.cognome=v.cognome
join esercizio e on v.esercizio=e.nome
where sesso='maschio' and not(u.cognome like '%a%' or u.cognome like '%A%' or
u.cognome like '%e%' or u.cognome like '%E%' or u.cognome like '%i%' or u.cognome
like '%I%' or u.cognome like '%o%' or u.cognome like '%O%' or u.cognome like '%u%'
or u.cognome like '%U%');
```

La query effettua un join tra utente, vocescheda e esercizio sulle chiavi esterne per collegare i dati degli utenti con gli esercizi che effettuano e seleziona gli utenti maschi escludendo ogni vocale presente nel cognome sia minuscola che maiuscola. I duplicati vengono eliminati con la distinct.

### Domanda 2 (media complessità).

Elencare gli utenti e la durata massima dei loro esercizi per gli utenti che hanno tra 20 e 30 anni e hanno fatto in media tra 20 e 30 ripetizioni. **Non usare sottoquery.**

## Soluzione 2.

```
select v.nome, v.cognome, max(v.durata)
from utente u join vocesccheda v on u.nome=v.nome and u.cognome=v.cognome where
u.eta>=20 and u.eta<=30
group by v.nome, v.cognome
having avg(v.numripetizioni)>=20 and avg(v.numripetizioni)<=30;
```

La query effettua un join tra utente e vocesccheda sulle chiavi esterne per collegare i dati degli utenti con i dati degli esercizi e seleziona gli utenti che hanno tra 20 e 30 anni. Raggruppa per utente (attraverso la chiave primaria) per calcolare il numero medio di ripetizioni per ogni utente selezionando solo una media compresa tra 20 e 30 ripetizioni. Nel risultato riportiamo il nome e il cognome dell'utente con la durata massima dell'esercizio per ogni utente.

## Domanda 3 (alta complessità).

Tra gli utenti che hanno fatto solo esercizi con più di 30 ripetizioni (tranne che con la cyclette, per cui il numero di ripetizioni può essere qualunque), trovare l'utente più vecchio e il numero totale delle ripetizioni dei suoi esercizi. Si presti attenzione al fatto che non tutti gli esercizi usano attrezzi e che possono esserci più utenti con la stessa età.

## Soluzione 3.

```
with utenti30rip as (
    select nome, cognome, eta, numripetizioni
    from utente u join vocesccheda v1 on u.nome=v1.nome and u.cognome=v1.cognome where
    not exists (
        select *
        from vocesccheda v2 join esercizio e on v2.esercizio=e.nome
        where v2.nome=u.nome and v2.cognome=u.cognome and
        (attrezzo<>'cyclette' or attrezzo is null) and numripetizioni<=30
    )
)
select nome, cognome, sum(numripetizioni)
from utenti30rip
where eta=(
    select max(eta)
    from utenti30rip
)
group by nome, cognome;
```

Con una common table expression (clausola with) costruiamo una sottoquery utenti30rip che ricava gli utenti che hanno fatto solo esercizi con più di 30 ripetizioni (tranne che con la cyclette).

Per farlo escludiamo dall'insieme degli utenti quelli che hanno fatto almeno un esercizio con meno di 30 ripetizioni con un attrezzo diverso dalla cyclette.

Dato che l'attributo attrezzo può essere NULL, per selezionare gli attrezzi diversi dalla cyclette, dobbiamo esplicitamente considerare il caso in cui attrezzo sia NULL.

Nella query principale abbiamo una sottoquery che seleziona l'età massima da utenti30rip e selezioniamo gli utenti che hanno questa età massima ottenendo per ogni utente la somma delle ripetizioni effettuate.

## Esame di Teoria (rispondere su fogli separati rispetto alla parte di SQL)

### Domanda 1 (9 punti).

Con riferimento alla base di dati AllYouCanFit:

A. (4 punti) Esprimere in Algebra Relazionale l'interrogazione

**Elencare nome, cognome e età degli utenti che hanno fatto l'esercizio che è durato di più.**

B. (5 punti) Esprimere, nel calcolo relazionale su tuple con dichiarazione di range, la seguente domanda:

**Elencare nome, cognome e sesso degli utenti che hanno fatto solo esercizi con il manubrio e con più di 30 ripetizioni.**

### Soluzione 1.

A. Una possibile soluzione in algebra relazionale è la seguente:

$$\pi_{\text{utente.nome, utente.cognome, eta}} \left( \begin{array}{l} \text{utente} \bowtie_{\text{utente.nome}=V1.\text{nome} \wedge \text{utente.cognome}=V1.\text{cognome}} \left( \right. \\ \left. \rho_{V1 \leftarrow \text{VOCESCHEDA}}(\text{vocescheda}) - \right. \\ \left. \left. \pi_{V1.*}(\rho_{V1 \leftarrow \text{VOCESCHEDA}}(\text{vocescheda}) \bowtie_{V1.\text{durata} < V2.\text{durata}} \rho_{V2 \leftarrow \text{VOCESCHEDA}}(\text{vocescheda})) \right) \right) \end{array} \right)$$

Ricavo l'esercizio con durata massima escludendo dalle tuple di vocesccheda tutte le tuple minori di almeno un'altra tupla. Dopo, effettuo un equijoin con utente per ricavare i dati dell'utente.

B. Una possibile soluzione è la seguente:

$$\begin{array}{l} \{u.\text{nome}, u.\text{cognome}, u.\text{sesso} \mid u(\text{utente}) \mid \\ \forall v(\text{vocescheda})((v.\text{nome} = u.\text{nome} \wedge v.\text{cognome} = u.\text{cognome}) \Rightarrow \\ (\text{numripetizioni} > 30 \wedge \exists e(\text{esercizio})(e.\text{nome} = v.\text{esercizio} \wedge e.\text{attrezzo} = \text{'manubrio'}))) \} \end{array}$$

Considero tutte le tuple di utente per cui tutte le tuple di vocesccheda relative a loro hanno più di 30 ripetizioni e sono effettuate con un manubrio.

### Domanda 2 (9 punti).

A. Riportare la definizione di chiusura di un insieme di attributi.

B. Dati  $R(A,B,C,D,E)$  e  $F = \{AB \rightarrow BC, CD \rightarrow BC, AC \rightarrow D\}$  dire, motivando la risposta, se  $R$  è in 3FN e, se non lo è, decomporla in relazioni in 3FN esplicitando tutti i passaggi. Il risultato è BCNF? Perché?

### Soluzione 2.

A. Si vedano gli appunti/testo/slide.

B. Per prima cosa, occorre individuare la o le chiavi della relazione  $R$ . Ogni chiave deve contenere  $A$  ed  $E$ , perché non compaiono a destra in nessuna dipendenza funzionale.  $ABE^+ = \{A, B, E \mid C \mid D\}$ , e quindi è chiave; inoltre  $ACE^+ = \{A, C, E \mid D \mid B\}$  è chiave, mentre  $ADE^+ = \{A, D, E\}$  non è chiave.

La relazione non è in 3NF, infatti  $AC \rightarrow D$  non è riflessiva, non ha a sinistra una superchiave né a destra attributi primi.

Troviamo quindi la copertura minimale di  $F$ , partendo dalla sua forma canonica:  $F' = \{AB \rightarrow B, AB \rightarrow C, CD \rightarrow B, CD \rightarrow C, AC \rightarrow D\}$ .

$A$  è estraneo in  $AB \rightarrow B$  perché  $B$  ovviamente si può ottenere da se stesso e allo stesso modo  $D$  è estraneo in  $CD \rightarrow C$ , quindi:

$F' = \{B \rightarrow B, AB \rightarrow C, CD \rightarrow B, C \rightarrow C, AC \rightarrow D\}$ .

$B \rightarrow B$  e  $C \rightarrow C$  sono ridondanti perché riflessive, quindi la copertura minimale è  $F' = \{AB \rightarrow C, CD \rightarrow B, AC \rightarrow D\}$ .

Possiamo ora scomporre in 3NF e otteniamo le seguenti relazioni dopo aver accorpato le d.f. con lo stesso antecedente:

$R1(\underline{A}, \underline{B}, C)$

$R2(\underline{C}, \underline{D}, B)$

$R3(\underline{A}, \underline{C}, D)$

cui bisogna aggiungere anche una relazione  $R4(\underline{A}, \underline{B}, E)$  contenente una chiave della relazione R.

Il risultato è BCNF perché ogni relazione ha la dipendenza funzionale implicita di chiave primaria, cioè di tipo superchiave.

### Domanda 3 (7 punti).

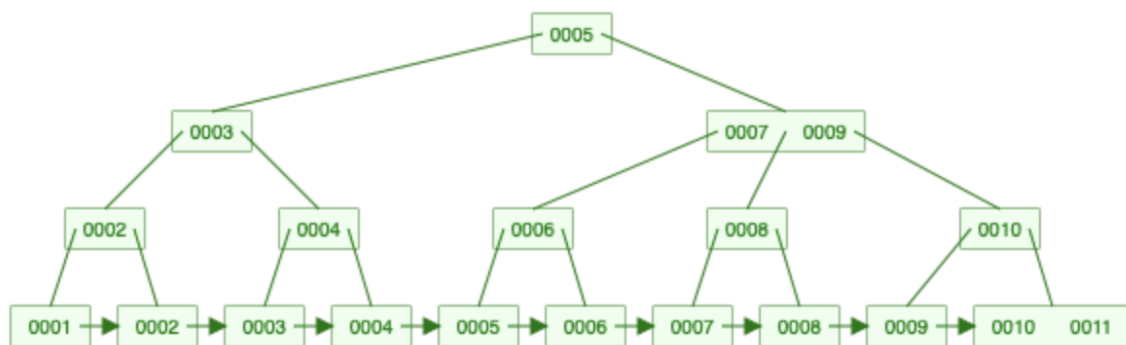
A. Riportare la definizione di B+-tree.

B. Rappresentare un possibile B+-tree contenente le chiavi (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11) per il caso  $m=3$ . Non si richiede di simulare le singole operazioni di inserimento, ma di mostrare un possibile B+-tree con le caratteristiche indicate.

### Soluzione 3.

A. Vedere il libro di testo o le slide.

B. Se  $m=3$ , ogni nodo ha al massimo  $m - 1 = 2$  chiavi, la radice ha almeno 1 chiave, gli altri nodi hanno almeno  $\lceil m/2 - 1 \rceil = 1$  chiave. Le foglie devono contenere tutte le chiavi e essere linkate. Un possibile B+-tree è in figura.



### Domanda 4 (8 punti).

Considerare la seguente storia interfogliata

$S = r1(x), r1(y), r2(x), r3(y), r2(y), w2(x)$  La storia S è compatibile con il protocollo 2PL? Giustificare la risposta.

### Soluzione 4.

La storia S è compatibile con 2PL perché banalmente c'è un'unica scrittura ed è l'ultima operazione della storia, quindi sicuramente le altre transazioni possono rilasciare i loro lock senza avere bisogno di richiederli nuovamente. Un esempio di aggiunta dei lock 2PL è il seguente:

$S = LS1(x), r1(x), LS1(y), r1(y), LS2(x), r2(x), LS3(y), r3(y), LS2(y), r2(y), UL1(x), LX2(x), w2(x)$