

Esempio di esercizi per la prima prova in itinere 2011-2012

Questo documento contiene alcuni esempi di esercizi di preparazione alla prima prova 2011-2012. Per un'adeguata preparazione si suggerisce di svolgerli tutti, tenendo presente che gli esercizi per i vari argomenti sono elencati in ordine di difficoltà.

Algebra relazionale e calcolo su tuple

Dato il seguente schema relazionale¹

FILM(idf,nomef,budget,nazione,annof)
CONF(film,premio,annoc)
PREMI(idp,nomep)

con i seguenti vincoli referenziali

CONF.film -> FILM.idf
CONF.premio-> PREMI.idf

scrivere le seguenti interrogazioni in algebra relazionale con theta-join e in calcolo su tuple. Svolgere le interrogazioni in sequenza aiuta nel trovare la soluzione, visto che alcune interrogazioni possono essere scritte usando i risultati delle precedenti.

1. Trovare i premi oscar conferiti a film italiani, mostrando nome del film e anno di conferimento del premio,
2. Trovare il nome di tutte le nazioni che hanno vinto un premio prima del 2010
3. Trovare il nome di tutte le nazioni, fra quelle in cui si producono film, che non hanno vinto un premio prima del 2010
4. Trovare il nome di tutte le nazioni in cui sono stati prodotti almeno due film
5. Trovare il nome di tutte le nazioni in cui è stato prodotto esattamente (non di più né di meno) un solo film
6. Trovare il nome di tutte le nazioni in cui sono stati prodotti almeno tre film
7. Trovare il nome dei FILM con il budget massimo
8. Trovare il nome del FILM che in ordine lessicografico ascendente segue subito dopo 'matrix' (E' il FILM che si trova subito dopo matrix se ordiniamo i film per nome dalla A alla Z. Si osservi che due nomi possono essere confrontati usando i normali operatori, <,>,<=,>=).

¹ Si osservi che per rendere più brevi le interrogazioni, sono usati nomi abbreviati per tabelle e attributi. CONF è una tabella che contiene i conferimenti dei premi, idf è un attributo che contiene il codice di un film, idp il codice di un premio e così via.

SQL: data definition language

1. Dato il seguente schema logico e i seguenti vincoli, si scriva il codice SQL che crea le tabelle SQL corrispondenti. Si ricordi di creare le tabelle nell'ordine richiesto dal linguaggio SQL. Si osservi che la matricola del dipendente non è unica all'interno della tabella dipendenti in quanto dipendenti di aziende diverse potrebbero avere la stessa matricola.

aziende(id, nome, PIVA)

dipendenti(idAz, matricola, nome, cognome, CF, dataNascita, dataAssunzione, stipendio)

telefoni(idAz, matricolaDip, numero, cellulare)

Vincoli integrità referenziale

dipendenti.idAz -> aziende.id

telefoni.idAz, telefoni.matricolaDip -> dipendenti.idAz, dipendenti.matricola

Altri vincoli

Il codice fiscale è unico per ciascun dipendente ed è lungo 16 caratteri

La partita iva è unica per ciascuna azienda ed è lunga 11 caratteri

Nessun attributo può essere nullo eccetto lo stipendio

Lo stipendio deve permettere di rappresentare valori fino a 999.999,99 e deve avere due cifre decimali.

Il campo cellulare può valere 0 o 1.

Il valore di default per il campo cellulare è 0.

La data di assunzione deve essere successiva a quella di nascita.

SQL: data manipulation language language

Dato il seguente schema logico

```
aziende(id, nome, nazione, numeroDipendenti, dataNascita)
dipendenti(idAz, matricola, nome, cognome, dataAssunzione, stipendio)
telefoni(idAz, matricolaDip, numero)
```

con i seguenti vincoli di integrità referenziale

```
dipendenti.idAzienda -> aziende.id
telefoni.idAz, telefoni.matricolaDip -> dipendenti.idAz, dipendenti.matricola
```

si implementino in SQL le seguenti interrogazioni. Per le interrogazioni da 1 a 10 non si usino sottointerrogazioni e viste.

1. Trovare telefoni di tutti i dipendenti assunti dopo il '1/1/1990', mostrando numero di telefono, nome, cognome del dipendente e nome dell'azienda. Si ordino i risultati, dalla A alla Z, rispetto al nome dell'azienda, al cognome del dipendente e al nome del dipendente.
2. Trovare tutti i dipendenti dell'azienda 'de Paperoni spa', restituendo il cognome, il nome lo stipendio netto e lo stipendio lordo del dipendente. Si assuma che il valore dello stipendio del dipendente presente nell'archivio sia quello lordo e che il netto venga calcolato approssimativamente decrementando il lordo del 40%. Si ordino i risultati, nell'ordine, rispetto al cognome del dipendente dalla Z alla A e al nome del dipendente dalla A alla Z
3. Trovare tutti i dipendenti delle aziende il cui nome inizia per "p" mostrando nome, cognome del dipendente e nome dell'azienda.
4. Ripetere l'interrogazione precedente, avendo cura di mostrare anche le aziende che non hanno dipendenti.
5. Si trovi le coppie di aziende nate nella stessa data restituendone il nome delle aziende e la data. Questa interrogazione può essere implementata sia facendo comparire ogni coppia di aziende due volte, cioè 'azienda1 azienda2' e 'azienda2 azienda1', sia una sola volta, ad esempio, restituendo solo le coppie dove l'id dell'azienda di sinistra precede quella di destra.. Mostrare entrambe le soluzioni.
6. Si trovi il numero di aziende presenti nell'archivio .
7. Si ripeta l'interrogazione precedente avendo però cura di contare una sola volta le aziende con lo stesso nome.
8. Si trovi il numero di aziende presenti nell'archivio per nazione, mostrando il nome della nazione e il numero di aziende.
9. Si ripeta l'interrogazione precedente usando nel conteggio solo le aziende con più di 15 dipendenti e mostrando solo le nazioni con più di 20 aziende.
10. Si trovi lo stipendio medio pagato dalle aziende, mostrandolo insieme al nome dell'azienda e facendo attenzione al fatto che l'archivio potrebbe contenere aziende con lo stesso nome
11. Si crei una vista che contiene il risultato della precedente interrogazione
12. Si usi la vista precedente per calcolare lo stipendio medio lordo e quello medio netto. Si assuma che il valore dello stipendio del dipendente presente nell'archivio sia quello lordo e che il netto venga calcolato approssimativamente decrementando il lordo del 40%.
13. Si trovino le aziende senza dipendenti, mostrandone il nome e l'id. Questa interrogazione può essere implementata sia usando le sottointerrogazioni (operatore IN) che l'operatore differenza except. Si mostrino entrambe le soluzioni.
14. Si trovino i nomi delle aziende i cui dipendenti sono tutti stati assunti alla data di nascita dell'azienda. Si usi le sottointerrogazioni con l'operatore EXISTS..
15. Si trovi l'azienda con più dipendenti, mostrandone il nome. Questa interrogazione può essere implementata sia usando le viste e il join sia con le sottointerrogazioni. Si mostrino entrambe le soluzioni.
16. Si trovi la nazione con la media dipendenti per azienda più alta, mostrandone il nome.

SOLUZIONI

Esercizi algebra relazionale e calcolo su tuple

1. Trovare i premi oscar conferiti a film italiani, mostrando nome del film e anno di conferimento del premio,

$$\circ \{f.nomef, c, anno \mid f(FILM), c(CONF), p(PREMI) \mid \\ f.idf=c.film \wedge c.premio=p.idp \wedge p.nomep='oscar' \wedge f.nazione='italia' \}$$

che può essere scritta, ad esempio, anche come

$$\{f.nomef, c, anno \mid f(FILM), c(CONF) \mid \\ f.idf=c.film \wedge f.nazione='italia' \wedge \exists p(PREMI) c.premio=p.idp \wedge p.nomep='oscar' \}$$

$$\circ \pi_{nomef,annoc} (\sigma_{nazione='italia'} (FILM) \triangleright \triangleleft_{idf=film} CONF \triangleright \triangleleft_{premio=idp} \sigma_{nomep='oscar'} (PREMI))$$

2. Trovare il nome di tutte le nazioni che hanno vinto un premio prima del 2010

$$\circ \{f.nazione \mid f(FILM), c(CONF) \mid \\ f.idf=c.film \wedge c.anno < 2010\}$$

che può essere scritta, ad esempio, anche come

$$\{f.nazione \mid f(FILM) \mid \\ \exists c(CONF) f.idf=c.film \wedge c.anno < 2010\}$$

$$\circ \pi_{nazione} (FILM \triangleright \triangleleft_{idf=film} \sigma_{annoc < 2010} (CONF))$$

3. Trovare il nome di tutte le nazioni, fra quelle in cui si producono film, che non hanno vinto un premio prima del 2010

$$\circ \{f.nazione \mid f(FILM) \mid \\ \neg \exists c2(CONF) \exists f2(FILM) f.nazione=f2.nazione \wedge f2.idf=c2.film \wedge c2.anno < 2010\}$$

$$\circ NAZIONI_VINCENTI = \pi_{nazione} (FILM \triangleright \triangleleft_{idf=film} \sigma_{annoc < 2010} (CONF)) \\ \pi_{nazione} (FILM) - NAZIONI_VINCENTI$$

4. Trovare il nome di tutte le nazioni in cui sono stati prodotti almeno due film

- $\{f.nazione \mid f(FILM) \mid \exists f2(FILM) f.idf \neq f2.idf \wedge f.nazione = f2.nazione \}$

- $FILM2 = \rho_{idf2, nomef2, budget2, nazione2, annof2} \leftarrow idf, nomef, budget, nazione, annof (FILM)$
 $\pi_{nazione} (FILM \triangleright \triangleleft_{idf \neq idf2 \wedge nazione = nazione2} (FILM2))$

5. Trovare il nome di tutte le nazioni in cui è stato prodotto esattamente (non di più né di meno) un solo film

- $\{f.nazione \mid f(FILM) \mid \neg \exists f2(FILM) f.idf \neq f2.idf \wedge f.nazione = f2.nazione \}$
- $FILM2 = \rho_{idf2, nomef2, budget2, nazione2, annof2} \leftarrow idf, nomef, budget, nazione, annof (FILM)$
 $NAZIONE_DUE_VITTORIE = \pi_{nazione} (FILM \triangleright \triangleleft_{idf \neq idf2 \wedge nazione = nazione2} (FILM2))$
 $\pi_{nazione} (FILM) - NAZIONE_DUE_VITTORIE$

6. Trovare il nome di tutte le nazioni in cui sono stati prodotti almeno tre film

- $\{f.nazione \mid f(FILM) \mid \exists f2(FILM) \exists f3(FILM) f.idf \neq f2.idf \wedge f.idf \neq f3.idf \wedge f2.idf \neq f3.idf \wedge f.nazione = f2.nazione \wedge f.nazione = f3.nazione \}$
- $FILM2 = \rho_{idf2, nomef2, budget2, nazione2, annof2} \leftarrow idf, nomef, budget, nazione, annof (FILM)$
 $FILM3 = \rho_{idf3, nomef3, budget3, nazione3, annof3} \leftarrow idf, nomef, budget, nazione, annof (FILM)$
 $\pi_{nazione} ((FILM \triangleright \triangleleft_{idf \neq idf2 \wedge nazione = nazione2} FILM2) \triangleright \triangleleft_{idf \neq idf3 \wedge idf2 \neq idf3 \wedge nazione = nazione3} FILM3)$

7. Trovare il nome dei FILM con il budget massimo

- $\{f.nomef \mid f(FILM) \mid \neg \exists f2(FILM) f2.budget > f.budget \}$

che può essere riscritta, ad esempio, come

$$\{f.nomef \mid f(FILM) \mid \forall f2(FILM) f2.budget \leq f.budget \}$$

- $FILM2 = \rho_{idf2, nomef2, budget2, nazione2, annof2} \leftarrow idf, nomef, budget, nazione, annof (FILM)$
 $BUDGET_NON_MASSIMO = \pi_{idf, nomef} (FILM \triangleright \triangleleft_{idf \neq idf2 \wedge budget < budget2} FILM2)$
 $\pi_{nomef} (\pi_{idf, nomef} (FILM) - BUDGET_NON_MASSIMO)$

8. Trovare il nome del FILM che in ordine lessicografico ascendente segue subito dopo 'matrix' (E' il FILM che si trova subito dopo matrix se ordiniamo i film per nome dalla A alla Z. Si osservi che due nomi possono essere confrontati usando i normali operatori, <, >, =).

- $\{f.nomef \mid f(FILM) \mid f.nome > 'matrix' \wedge \neg \exists f2(FILM) f2.nome > 'matrix' \wedge f2.nome < f.nome \}$

- $\text{DOPO_MATRIX} = \sigma_{\text{nomef} > \text{'matrix'}} (\text{FILM})$
 $\text{DOPO_MATRIX2} = \rho_{\text{idf2, nomef2, budget2, nazione2, annof2} \leftarrow \text{idf, nomef, budget, nazione, annof}} (\text{DOPO_MATRIX})$
 $\text{NON_SUBITO_DOPO} = \pi_{\text{idf, nomef}} (\text{DOPO_MATRIX} \bowtie_{\text{idf} \neq \text{idf2} \wedge \text{nome} < \text{nome2}} \text{DOPO_MATRIX2})$
 $\pi_{\text{nomef}} (\pi_{\text{idf, nomef}} (\text{DOPO_MATRIX}) - \text{NON_SUBITO_DOPO})$

SQL: data definition language

2. Dato il seguente schema logico e i seguenti vincoli, si scriva il codice SQL che crea le tabelle SQL corrispondenti. Si ricordi di creare le tabelle nell'ordine richiesto dal linguaggio SQL. Si osservi che la matricola del dipendente on è unica all'interno della tabella dipendenti in quanto dipendenti di aziende diverse potrebbero avere la stessa matricola.

aziende(id, nome, PIVA)
dipendenti(idAz, matricola, nome, cognome, CF, dataNascita, dataAssunzione, stipendio)
telefoni(idAz, matricolaDip, numero, cellulare)

Vincoli integrità referenziale

dipendenti.idAz -> aziende.id
telefoni.idAz, telefoni.matricolaDip -> dipendenti.idAz, dipendenti.matricola

Altri vincoli

Il codice fiscale è unico per ciascun dipendente ed è lungo 16 caratteri

La partita iva è unica per ciascuna azienda ed è lunga 11 caratteri

Nessun attributo può essere nullo eccetto lo stipendio

Lo stipendio deve permettere di rappresentare valori fino a 999.999,99 e deve avere due cifre decimali.

Il campo cellulare può valere 0 o 1.

Il valore di default per il campo cellulare è 0.

La data di assunzione deve essere successiva a quella di nascita

```
CREATE TABLE aziende(  
    id serial not null primary key,  
    nome varchar(20) not null,  
    PIVA char(11) not null unique  
)
```

```
CREATE TABLE dipendenti (  
    idAz integer not null references aziende(id),  
    matricola int not null,  
    nome varchar(20) not null,  
    cognome varchar(20) not null,  
    CF char(16) not null unique,  
    dataNascita date not null,  
    dataAssunzione date not null,  
    stipendio decimal(8,2) ,  
    primary key(idAz,matricola),  
    check (dataNascita<dataAssunzione)  
)
```

```
CREATE TABLE telefoni (  
    idAz integer not null,  
    matricolaDip integer not null,  
    numero varchar(20) not null primary key,  
    cellulare bit default 0,  
    foreign key (idAz,matricolaDip) references dipendenti(idAz,matricola)  
)
```

SQL: data manipulation language language

Dato il seguente schema logico

```
aziende(id, nome, nazione, numeroDipendenti, dataNascita)
dipendenti(idAz, matricola, nome, cognome, dataAssunzione, stipendio)
telefoni(idAz, matricolaDip, numero)
```

con i seguenti vincoli di integrità referenziale

```
dipendenti.idAzienda -> aziende.id
telefoni.idAz, telefoni.matricolaDip -> dipendenti.idAz, dipendenti.matricola
```

si implementino in SQL le seguenti interrogazioni. Per le interrogazioni da 1 a 10 non si usino sottointerrogazioni e viste.

1. Trovare telefoni di tutti i dipendenti assunti dopo il '1/1/1990', mostrando numero di telefono, nome, cognome del dipendente e nome dell'azienda. Si ordino i risultati, dalla A alla Z, rispetto al nome dell'azienda, al cognome del dipendente e al nome del dipendente.

```
SELECT aziende.nome AS "nome azienda", dipendenti.nome as "nome dipendente",
       cognome as "cognome dipendente", telefono
FROM aziende, dipendenti, telefoni
WHERE aziende.id=dipendenti.idAz AND dipendenti.matricola=telefoni.matricolaDip
      AND dipendenti.idAz=telefoni.idAz AND dipendente. dataAssunzione > '1/1/1990'
ORDER BY aziende.nome, dipendenti.cognome, dipendenti.nome
```

2. Trovare tutti i dipendenti dell'azienda 'de Paperoni spa', restituendo il cognome, il nome lo stipendio netto e lo stipendio lordo del dipendente. Si assuma che il valore dello stipendio del dipendente presente nell'archivio sia quello lordo e che il netto venga calcolato approssimativamente decrementando il lordo del 40%. Si ordino i risultati, nell'ordine, rispetto al cognome del dipendente dalla Z alla A e al nome del dipendente dalla A alla Z

```
SELECT dipendenti.nome, cognome, stipendio AS "stipendio lordo", stipendio *0.6 AS "stipendio netto"
FROM aziende, dipendenti
WHERE aziende.id=dipendenti.idAz AND aziende.nome like 'de Paperoni spa'.
ORDER BY dipendenti.cognome DESC, dipendenti.nome ASC
```

3. Trovare tutti i dipendenti delle aziende il cui nome inizia per "p" mostrando nome, cognome del dipendente e nome dell'azienda.

```
SELECT aziende.nome AS "nome azienda", dipendenti.nome as "nome dipendente",
       cognome as "cognome dipendente"
FROM aziende, dipendenti
WHERE aziende.id=dipendenti.idAz AND aziende.nome like "p%".
```

4. Ripetere l'interrogazione precedente, avendo cura di mostrare anche le aziende che non hanno dipendenti.

```
SELECT azienda.nome AS "nome azienda", dipendenti.nome as "nome dipendente",
       cognome as "cognome dipendente"
```

```
FROM aziende LEFT JOIN dipendenti ON aziende.id=dipendenti.idAz
WHERE aziende.nome like "p%"
ORDER BY aziende.nome ASC, dipendenti.cognome DESC, dipendenti.nome DESC
```

5. Si trovi le coppie di aziende nate nella stessa data restituendone il nome delle aziende e la data. Questa interrogazione può essere implementata sia facendo comparire ogni coppia di aziende due volte, cioè 'azienda1 azienda2' e 'azienda2 azienda1', sia una sola volta, ad esempio, restituendo solo le coppie dove l'id dell'azienda di sinistra precede quella di destra.. Mostrare entrambe le soluzioni.

- ```
SELECT az1.dataNascita, az1.nome AS "nome azienda 1", az2. "nome azienda 2"
FROM aziende AS az1, aziende as az2
WHERE az1.dataNascita = az2.dataNascita
```
- ```
SELECT az1.dataNascita, az1.nome AS "nome azienda 1", az2. "nome azienda 2"
FROM aziende AS az1, aziende as az2
WHERE az1.dataNascita = az2.dataNascita AND az1.id>az2.id
```

6. Si trovi il numero di aziende presenti nell'archivio .

```
SELECT COUNT(*) AS "numero aziende"
FROM aziende
```

7. Si ripeta l'interrogazione precedente avendo però cura di contare una sola volta le aziende con lo stesso nome.

```
SELECT COUNT(DISTINCT nome) AS "numero aziende"
FROM aziende
```

8. Si trovi il numero di aziende presenti nell'archivio per nazione, mostrando il nome della nazione e il numero di aziende.

```
SELECT nazione, COUNT(*) AS "numero aziende"
FROM aziende
GROUP BY nazione
```

9. Si ripeta l'interrogazione precedente usando nel conteggio solo le aziende con più di 15 dipendenti e mostrando solo le nazioni con più di 20 aziende.

```
SELECT nazione, COUNT(*) AS "numero aziende"
FROM aziende
WHERE numeroDipendenti>15
GROUP BY nazione
HAVING COUNT(*)>20
```

10. Si trovi lo stipendio medio pagato dalle aziende, mostrandolo insieme al nome dell'azienda e facendo attenzione al fatto che l'archivio potrebbe contenere aziende con lo stesso nome

```
SELECT aziende.nome, AVG(stipendio) AS media
FROM aziende, dipendenti
WHERE aziende.id=dipendenti.idAz
GROUP BY aziende.id, aziende.nome
```

11. Si crei una vista che contiene il risultato della precedente interrogazione

```
CREATE VIEW mediaStipendi(nome,media)
SELECT aziende.nome, AVG(stipendio)
FROM aziende, dipendenti
WHERE aziende.id=dipendenti.idAz
GROUP BY aziende.id, aziende.nome
```

12. Si usi la vista precedente per calcolare lo stipendio medio lordo e quello medio netto. Si assuma che il valore dello stipendio del dipendente presente nell'archivio sia quello lordo e che il netto venga calcolato approssimativamente decrementando il lordo del 40%.

```
SELECT nome, media AS "media lordo", media * 0.6 AS "media netto"
FROM mediaStipendi
```

13. Si trovino le aziende senza dipendenti, mostrandone il nome e l'id. Questa interrogazione può essere implementata sia usando le sottointerrogazioni (operatore IN) che l'operatore differenza except. Si mostrino entrambe le soluzioni.

- ```
SELECT id, nome
FROM aziende
WHERE id NOT IN (SELECT idAz FROM dipendenti)
```

- ```
SELECT id, nome
FROM aziende
EXCEPT
SELECT id, nome
FROM aziende, dipendenti
WHERE aziende.id=dipendenti.idAz
```

14. Si trovino i nomi delle aziende i cui dipendenti sono tutti stati assunti alla data di nascita dell'azienda. Si usi le sottointerrogazioni con l'operatore EXISTS..

```
SELECT nome
FROM aziende
WHERE NOT EXISTS (SELECT *
FROM dipendenti
WHERE dataNascita <> dataAssunzione AND aziende.id=dipendenti.idAz)
```

15. Si trovi l'azienda con più dipendenti, mostrandone il nome. Questa interrogazione può essere implementata sia usando le viste e il join sia con le sottointerrogazioni. Si mostrino entrambe le soluzioni.

- ```
CREATE VIEW massimo(maxDipendenti)
SELECT MAX(numeroDipendenti)
FROM aziende

SELECT nome
FROM aziende, massimo
WHERE aziende.numeroDipendenti=massimo.maxDipendenti
```

- ```
SELECT nome
```

```
FROM aziende, massimo  
WHERE aziende.numeroDipendenti IN (SELECT massimo FROM.maxDipendenti)
```

16. Si trovi la nazione con la media dipendenti per azienda più alta, mostrandone il nome.

```
CREATE VIEW mediaDipendenti(nazione,media)  
SELECT nazione, AVG(numeroDipendenti)  
FROM aziende  
GROUP BY nazione
```

```
SELECT nome  
FROM mediaDipendenti  
WHERE media IN (SELECT MAX(media) FROM. mediaDipendenti)
```


