

Laurea in Ingegneria Informatica – SAPIENZA Università di Roma

Insegnamento di Basi di Dati

Esercitazione:

Esercizi di esame su SQL

Domenico Fabio Savo

Esercizio 1

La relazione **Dirige(azienda,direttore)** memorizza, per ogni azienda, il direttore.

La relazione **Hobby(persona,hobby)** specifica quali hobby abbiano le varie persone.

Si chiede di esprimere in SQL le seguenti interrogazioni:

1. Mostrare i direttori che non hanno alcun hobby.
2. Mostrare le coppie di direttori che hanno almeno un hobby in comune.
3. Mostrare i direttori che hanno meno hobby del numero di aziende che dirigono.

Soluzione 1.1

Un direttore non ha alcun hobby se non appare nella tabella “Hobby”.

```
SELECT direttore
FROM Dirige
WHERE direttore NOT IN (
    SELECT persona
    FROM Hobby)
```

Soluzione 1.2

Si calcola il join tra le relazioni “Dirige” ed “Hobby” ed ancora tra “Dirige” ed “Hobby” con una selezione per individuare le coppie di direttori tra loro diversi e che condividono il medesimo hobby.

```
SELECT d1.direttore, d2.direttore
FROM Dirige d1, Dirige d2, Hobby h1, Hobby h2
WHERE d1.direttore = h1.persona AND
      d2.direttore = h2.persona AND
      d1.direttore <> d2.direttore AND
      h1.hobby = h2.hobby
```

Soluzione 1.3

È sufficiente aggregare il join tra “Dirige” e “Hobby” su “direttore”, ed usare la clausola “having” per filtrare solo i direttori che hanno meno hobby delle aziende che dirigono

```
SELECT direttore
FROM Dirige, Hobby
WHERE direttore = persona
GROUP BY direttore
HAVING
count(distinct hobby) < count(distinct azienda)
```

Esercizio 2

La relazione **Member(codp,codg,anno)** memorizza a quale gruppo è affiliata ogni persona, insieme all'anno dal quale inizia tale affiliazione. In particolare, una tupla **<p,g,a>** nella relazione **Member** indica che la persona **p** è affiliata al gruppo **g** a partire dall'anno **a**.

La relazione **Pratica(codp,sport)** specifica quali sport praticano le varie persone.

Si chiede di esprimere in SQL le seguenti interrogazioni:

1. Mostrare il codice dei gruppi che hanno almeno due membri e tali che almeno uno di questi membri pratica almeno uno sport.
2. Per ogni gruppo, indicare la persona (o le persone, se sono più d'una) che sono affiliate a tale gruppo da meno tempo.
3. Chiamiamo “omogeneo” un gruppo G se esiste almeno uno sport praticato da tutti i membri del gruppo G. Mostrare tutti i gruppi omogenei.
4. Mostrare i gruppi formati solo da persone che non praticano alcuno sport.

Soluzione 2.1

Per mostrare il codice dei gruppi con almeno due membri e tali che almeno uno di questi pratichi uno sport eseguiamo il join tra: le relazioni “Pratica” e “Member” (in modo da ottenere i membri di un gruppo che praticano almeno uno sport) ed ancora con la relazione “Member” sul campo “codg” con una selezione per individuare coppie di persone tra loro diverse (per ottenere gruppi con almeno due membri).

```
SELECT distinct m1.codg
FROM Member m1, Member m2, Pratica p
WHERE m1.codg = m2.codg AND
      m1.codp <> m2.codp AND
      m1.codp = p.codp
```

Soluzione 2.2 (1 / 2)

Per individuare la persona che, per ogni gruppo, è iscritta da meno tempo calcoliamo il join sui campi “codg” ed “anno” tra la relazione “Member” e la relazione ottenuta eseguendo la query sulla relazione “Member” che, per ogni gruppo, restituisce la data di iscrizione più recente ed il relativo gruppo.

```
SELECT m1.codg
FROM   Member m1,
       (SELECT codg, max(anno) AS maxanno
        FROM Member
        GROUP BY codg) m2
WHERE  m1.codg = m2.codg AND
       m1.anno = m2.maxanno
```


Soluzione 2.2 (2/2)

Il medesimo risultato si può ottenere creando una vista:

```
CREATE VIEW grupmaxanno(codg, maxanno) AS
SELECT codg, max(anno)
      FROM Member
      GROUP BY codg
```

```
SELECT m1.codp
FROM   Member m1,
      grupmaxanno m2
WHERE  m1.codg = m2.codg AND
      m1.anno = m2.maxanno
```

Soluzione 2.3

Usando delle query nella clausola “FROM” calcoliamo due nuove relazioni: la prima, “temp1”, contiene il numero di persone appartenenti ad ogni gruppo, la seconda, “temp2”, contiene il numero di persone che appartengono allo stesso gruppo e praticano il medesimo sport. Calcolando poi il join tra “temp1” e “temp2” sui campi “codg” ed il numero di persone otteniamo i gruppi “omogenei”.

```
SELECT temp1.codg
FROM (SELECT codg, count(distinct codp) AS tot1
      FROM Member
      GROUP BY codg) temp1,
      (SELECT m.codg, p.sport,
            count(distinct m.codp) AS tot2
      FROM Member m, Pratica p
      WHERE m.codp = p.codp
      GROUP BY m.codg, p.sport) temp2
WHERE temp1.codg = temp2.codg AND
      temp1.tot1 = temp2.tot2
```

Soluzione 2.3

Usando delle query nella clausola “FROM” calcoliamo due nuove relazioni: la prima, “temp1”, contiene il numero di persone appartenenti ad ogni gruppo, la seconda, “temp2”, contiene il numero di persone che appartengono allo stesso gruppo. Poi, nella clausola “WHERE”, selezioniamo i gruppi “omogenei”.

```
SELECT temp1.codg, temp1.tot1
FROM (SELECT m.codg, COUNT(p.sport) tot1
      FROM M
      GROUP BY m.codg) temp1
      (SELECT m.codg, COUNT(p.sport) tot2
      FROM M
      WHERE m.codg = p.codg
      GROUP BY m.codg, p.sport) temp2
WHERE temp1.codg = temp2.codg AND
      temp1.tot1 = temp2.tot2
```

**Anche in questo caso la soluzione
può essere ottenuta definendo due
viste**

Soluzione 2.4

Si calcola il complemento dell'insieme dei gruppi che hanno almeno un membro che pratica almeno uno sport.

```
SELECT distinct codg
FROM Member
WHERE codg NOT IN (
    SELECT distinct m.codg
    FROM Member m, Pratica p
    WHERE m.codp = p.codp)
```

Esercizio 3

La relazione **Opera(codo,anno,museo)** memorizza in quali musei sono state custodite le varie opere d'arte nei diversi anni (la tupla **<c,a,m>** indica che nell'anno **a** l'opera con codice **c** è stata custodita nel museo con codice **m**).

La relazione **Museo(codm,nazione)** specifica in quali nazioni hanno sede i musei.

Si chiede di esprimere in SQL le seguenti interrogazioni:

1. Mostrare le opere custodite almeno una volta per due anni consecutivi in musei italiani (non necessariamente lo stesso museo nei due anni).
2. Per ogni museo francese e per ogni anno mostrare il numero di opere custodite in quell'anno in quel museo, ma solo se tale numero supera 10.
3. Mostrare le opere che non sono mai state custodite in musei italiani.

Soluzione 3.1

Calcoliamo il join tra le relazioni Opera, Museo ed ancora Opera e Museo imponendo la condizione che la nazione dei musei deve essere “Italia” . Si selezionano poi le coppie della stessa opera con la condizione che l’anno di esposizione della prima sia pari all’anno di esposizione della seconda più 1.

```
SELECT o1.codo
FROM opera o1, opera o2, "Museo" m1, "Museo" m2
WHERE o1.museo = m1.codm AND
      m1.nazione = 'italia' AND
      o2.museo = m2.codm AND
      m2.nazione = 'italia' AND
      o1.codo = o2.codo AND
      o1.anno = o2.anno + 1
```

Soluzione 3.2

Calcoliamo il join tra la relazione “Opera” e “Museo” con la condizione che la nazione del museo sia la ‘Francia’. Calcoliamo poi una aggregazione sulla coppia di attributi “museo” ed “anno” ed usiamo la clausola “having” per filtrare solo i musei che hanno esposto più di 10 opere in quell’anno.

```
SELECT      anno, museo, count(distinct codo)
FROM        Opera o, Museo m
WHERE       o.museo = m.codm AND
            m.nazione = 'francia'
GROUP BY    o.museo, o.anno
HAVING      count(distinct o.codo) > 10
```

Soluzione 3.3

Calcoliamo il complemento dell'insieme delle opere che sono state esposte almeno una volta in 'Italia'.

```
SELECT codo
FROM Opera
WHERE codo <> ALL
      (SELECT codo
       FROM Opera, Museo
       WHERE museo = codm AND
        nazione = 'italia')
```


Esercizio 4 (per casa)

La relazione **Volo(cod,partenza,arrivo,numpass,comp)** memorizza, per ogni volo aereo effettuato, il codice, l'aeroporto di partenza, quello di arrivo, il numero di passeggeri e la compagnia aerea, mentre la relazione **Compagnia(codice,nazione)** specifica in quali nazioni hanno sede le compagnie aeree.

Si chiede di esprimere in SQL le seguenti interrogazioni:

1. Mostrare i dati dei voli di compagnie italiane.
2. Mostrare gli aeroporti dai quali non partono voli di compagnie francesi.
3. Per ogni aeroporto a, mostrare il numero complessivo di passeggeri di voli partiti da a, ma solo se il numero di voli partiti da a è maggiore di 4.

Nella sezione “materiale didattico” della pagina

<http://www.dis.uniroma1.it/~savo/bd1011.html>

È possibile scaricare lo script SQL (esercizio-4.sql) per creare e popolare un DB su MySQL per provare le query di questo esercizio.