

Esercizio 1 – Prodotti

rif. esercizi 6.1.3 (query 1-6), 6.2.2 (query 7-12)

Sia dato lo schema:

- product (model, maker, type)
- pc (model, speed, ram, hd, price)
- laptop (model, speed, ram, hd, screen, price)
- printer (model, color, type, price)

Esprimere in SQL le seguenti query:

1. Trovare modello, memoria e velocità per laptop di almeno 1200\$
2. Elencare le stampanti a colori (tornare tutti i dati a disposizione)
3. Elencare modello e HD dei PC di almeno 3.0 GHz e con prezzo inferiore a 1000\$
4. Elencare modello, velocità e dimensione HD per i PC con prezzo minore di 800\$
5. Come nella query (1), ma ridenominando *ram* in *gigabytes* e *speed* in *gigahertz*
6. Ritornare i produttori di laptop
7. Elencare i produttori che vendono PC ma non laptop
8. Trovare produttore e velocità dei laptop con HD di almeno 100 GB
9. Trovare modello e prezzo di tutti i prodotti venduti dal produttore C
10. Trovare le coppie di PC con stessa RAM e HD (evitare ripetizioni)
11. Trovare le velocità di CPU in comune a due o più PC
12. Trovare i produttori di almeno due computer (PC o laptop) veloci almeno 2 GHz

Esercizio 1 – Soluzioni (1/6)

1. Trovare modello, memoria e dimensione schermo per laptop di almeno 1200\$

```
SELECT model, ram, screen
FROM prod.laptop
WHERE price >= 1200;
```

2. Elencare le stampanti a colori (tornare tutti i dati a disposizione)

```
SELECT *
FROM prod.printer
WHERE color;
```

3. Elencare modello e HD dei PC di almeno 3.0 GHz e con prezzo inferiore a 1000\$

```
SELECT model, hd
FROM prod.pc
WHERE speed >= 3.0 AND price < 1000;
```

4. Elencare modello, velocità e dimensione HD per i PC con prezzo minore di 800\$

```
SELECT model, speed, hd
FROM prod.pc
WHERE price < 800;
```

Esercizio 1 – Soluzioni (2/6)

5. Come nella query (1), ma ridenominando ram in gigabytes e speed in gigahertz

```
SELECT model, ram AS gigabytes, speed AS gigahertz  
FROM prod.laptop  
WHERE price >= 1200;
```

6. Ritornare i produttori di laptop

```
SELECT DISTINCT maker  
FROM prod.product  
WHERE type = 'laptop';
```

7. Elencare i produttori che vendono PC ma non laptop

```
( SELECT maker  
  FROM prod.product  
  WHERE type = 'pc' )  
EXCEPT  
( SELECT maker  
  FROM prod.product  
  WHERE type = 'laptop' );
```

Esercizio 1 – Soluzioni (3/6)

8. Trovare produttore e velocità dei laptop con HD di almeno 100 GB

```
SELECT maker, speed  
FROM prod.product p, prod.laptop l  
WHERE hd >= 100 AND p.model = l.model;
```

oppure:

```
SELECT maker, speed  
FROM prod.product NATURAL JOIN prod.laptop  
WHERE hd >= 100;
```

oppure:

```
SELECT maker, speed  
FROM prod.product p JOIN prod.laptop l ON p.model = l.model  
WHERE hd >= 100;
```

Esercizio 1 – Soluzioni (4/6)

9. Trovare modello e prezzo di tutti i prodotti venduti dal produttore C

```
( SELECT model, price
  FROM prod.pc NATURAL JOIN prod.product
  WHERE maker = 'C' )
UNION
( SELECT model, price
  FROM prod.laptop NATURAL JOIN prod.product
  WHERE maker = 'C' )
UNION
( SELECT model, price
  FROM prod.printer NATURAL JOIN prod.product
  WHERE maker = 'C' )
```

10. Trovare le coppie di PC con stessa RAM e HD (evitare ripetizioni)

```
SELECT pc1.model AS model1, pc2.model AS model2
FROM prod.pc pc1, prod.pc pc2
WHERE pc1.ram = pc2.ram AND pc1.hd = pc2.hd AND pc1.model <
pc2.model;
```

Esercizio 1 – Soluzioni (5/6)

11. Trovare le velocità di CPU in comune a due o più PC

```
SELECT DISTINCT pc1.speed AS speed  
FROM prod.pc pc1, prod.pc pc2  
WHERE pc1.model <> pc2.model AND pc1.speed = pc2.speed;
```

12. Trovare i produttori di almeno due computer (PC o laptop) veloci almeno 2 GHz

```
( SELECT p1.make  
  FROM   prod.product p1, prod.product p2,  
         prod.pc pc1, prod.pc pc2  
  WHERE  p1.make = p2.make AND p1.model = pc1.model  
         AND p2.model = pc2.model and p1.model <> p2.model  
         AND pc1.speed >= 2.0 AND pc2.speed >= 2.0 )  
UNION  
( SELECT p1.make  
  FROM   prod.product p1, prod.product p2,  
         prod.pc pc1, prod.laptop lp2  
  WHERE  p1.make = p2.make AND p1.model = pc1.model  
         AND p2.model = lp2.model and p1.model <> lp2.model  
         AND pc1.speed >= 2.0 AND lp2.speed >= 2.0 )
```

[continua]

Esercizio 1 – Soluzioni (6/6)

12. Trovare i produttori di almeno due computer (PC o laptop) veloci almeno 2 GHz

[continua]

UNION

```
( SELECT p1.make
  FROM   prod.product p1, prod.product p2,
         prod.laptop lp1, prod.pc pc2
  WHERE  p1.make = p2.make AND p1.model = lp1.model
         AND p2.model = pc2.model and p1.model <> p2.model
         AND lp1.speed >= 2.0 AND pc2.speed >= 2.0 )
```

UNION

```
( SELECT p1.make
  FROM   prod.product p1, prod.product p2,
         prod.laptop lp1, prod.laptop lp2
  WHERE  p1.make = p2.make AND p1.model = lp1.model
         AND p2.model = lp2.model and p1.model <> p2.model
         AND lp1.speed >= 2.0 AND lp2.speed >= 2.0 );
```

Esercizio 2 – Navi da battaglia

rif. esercizi 6.1.4 (query 1-6), 6.2.3 (query 7-12)

Sia dato lo schema:

- classes (class, type, country, num_guns, bore, displacement)
- ships (name, class, launched)
- outcomes (ship, battle, result)
- battles (name, date)

Esprimere in SQL le seguenti query:

1. Elencare nome classe e paese di tutte le classi con almeno 12 cannoni
2. Elencare i nomi delle navi varate prima del 1915, cambiando *name* in *shipName*
3. Elencare i nomi delle navi che iniziano con 'M'
4. Elencare gli affondamenti in battaglia (nome nave e nome battaglia)
5. Elencare le navi che hanno lo stesso nome della rispettiva classe
6. Elencare i nomi delle navi il cui nome è composto da almeno tre parole
7. Elencare nome, stazza e numero cannoni delle navi della battaglia di Guadalcanal
8. Elencare le navi con stazza maggiore di 37000 t
9. Elencare le navi presenti nel DB (attenzione a navi in Outcomes non in Ships)
10. Trovare le battaglie con almeno due navi dello stesso paese
11. Trovare i paesi che avevano sia corazzate (bb) che incrociatori (bc)

Esercizio 2 – Soluzioni (1/4)

1. Elencare nome classe e paese di tutte le classi con almeno 12 cannoni

```
SELECT class, country
FROM ships.classes
WHERE num_guns >= 12;
```

2. Elencare i nomi delle navi varate prima del 1915, cambiando name in shipName

```
SELECT name AS "shipName"
FROM ships.ships
WHERE launched < 1915;
```

3. Elencare i nomi delle navi che iniziano con 'M'

```
SELECT name
FROM ships.ships
WHERE name LIKE 'M%';
```

4. Elencare gli affondamenti in battaglia (nome nave e nome battaglia)

```
SELECT ship, battle
FROM ships.outcomes
WHERE result = 'sunk';
```

Esercizio 2 – Soluzioni (2/4)

5. Elencare le navi che hanno lo stesso nome della rispettiva classe

```
SELECT name
FROM ships.ships
WHERE name = class;
```

6. Elencare i nomi delle navi il cui nome è composto da almeno tre parole

```
( SELECT name
  FROM ships.ships
  WHERE name LIKE '% % %' )
UNION
( SELECT ship AS name
  FROM ships.outcomes
  WHERE ship LIKE '% % %' );
```

7. Elencare nome, stazza e numero cannoni delle navi della battaglia di Guadalcanal

```
SELECT s.name, c.displacement, c.num_guns
FROM    ships.outcomes o JOIN ships.ships s ON o.ship = s.name
        JOIN ships.classes c ON c.class = s.class
WHERE   o.battle = 'Guadalcanal';
```

Esercizio 2 – Soluzioni (3/4)

8. Elencare le navi con stazza maggiore di 37000 t

```
SELECT s.name  
FROM ships.ships s NATURAL JOIN ships.classes c  
WHERE c.displacement > 37000;
```

9. Elencare le navi presenti nel DB (attenzione a navi in Outcomes non in Ships)

```
( SELECT name  
  FROM ships.ships )  
UNION  
( SELECT ship AS name  
  FROM ships.outcomes );
```

Esercizio 2 – Soluzioni (4/4)

10. Trovare le battaglie con almeno due navi dello stesso paese

```
SELECT DISTINCT o1.battle
FROM    ships.ships s1, ships.classes c1, ships.outcomes o1,
          ships.ships s2, ships.classes c2, ships.outcomes o2
WHERE    s1.class = c1.class AND s1.name = o1.ship AND
          s2.class = c2.class AND s2.name = o2.ship AND
          s1.name <> s2.name AND o1.battle = o2.battle
```

11. Trovare i paesi che avevano sia corazzate (bb) che incrociatori (bc)

```
SELECT DISTINCT c1.country
FROM    ships.classes c1, ships.classes c2
WHERE    c1.country = c2.country AND
          c1.type = 'bb' AND c2.type = 'bc';
```

Esercizio 3 – Visite mediche

paziente (id_paziente, codice_asl, provincia_asl, nome, cognome, data_nascita, provincia_nascita)

visita (id_visita, id_paziente, id_medico, data, peso, altezza, pressione_minima, pressione_massima)

medico (id_medico, nome, cognome, sesso, codice_asl, provincia_asl, indirizzo_ambulatorio)

Esprimere in SQL le seguenti query:

1. Per ciascuna visita, elencare data, cognome paziente e peso misurato
2. Trovare nome e cognome dei pazienti che hanno avuto pressione max ≥ 140
3. Elencare le date in cui la dott.ssa RUSSI ha visitato pazienti
4. Elencare nome e cognome dei pazienti visitati nell'agosto 2010
5. Elencare le visite nelle ASL di Milano: per ognuna, ritornare cognome paziente, data visita, peso e altezza misurati, ordinando per cognome e data visita
6. Elencare, per ciascun medico, i pazienti visitati
7. Ritornare nome e cognome di tutti i pazienti dell'ASL di Reggio Emilia, includendo peso e altezza misurati nelle varie visite se ci sono dati disponibili
8. Elencare nome e cognome di tutti i medici con i pazienti visitati (se ce ne sono)
9. Elencare i pazienti che non hanno effettuato visite
10. Elencare i pazienti che sono stati visitati da dottoresse

Esercizio 3 – Soluzioni (1/3)

1. Per ciascuna visita, elencare data, cognome paziente e peso misurato

```
SELECT data, cognome, peso  
FROM med.visita NATURAL JOIN med.paziente;
```

2. Trovare nome e cognome dei pazienti che hanno avuto pressione max ≥ 140

```
SELECT DISTINCT nome, cognome  
FROM med.paziente NATURAL JOIN med.visita  
WHERE pressione_massima  $\geq 140$ ;
```

3. Elencare le date in cui la dott.ssa RUSSI ha visitato pazienti

```
SELECT DISTINCT data  
FROM med.visita NATURAL JOIN med.medico  
WHERE cognome = 'RUSSI';
```

4. Elencare nome e cognome dei pazienti visitati nell'agosto 2010

```
SELECT DISTINCT nome, cognome  
FROM med.paziente NATURAL JOIN med.visita  
WHERE data  $\geq$  '2010-08-01' AND data  $\leq$  '2010-08-31';
```

Esercizio 3 – Soluzioni (2/3)

5. Elencare le visite nelle ASL di Milano: per ognuna, ritornare cognome paziente, data visita, peso e altezza misurati, ordinando per cognome e data visita

```
SELECT cognome, data, peso, altezza
FROM med.paziente NATURAL JOIN med.visita
WHERE provincia_asl = 'MI'
ORDER BY cognome, data;
```

6. Elencare, per ciascun medico, i pazienti visitati

```
SELECT DISTINCT m.cognome AS cognome_medico, m.nome AS nome_medico,
    p.cognome AS cognome_paziente, p.nome AS nome_paziente
FROM med.paziente p
    JOIN med.visita v ON p.id_paziente = v.id_paziente
    JOIN med.medico m ON v.id_medico = m.id_medico;
```

7. Ritornare nome e cognome di tutti i pazienti dell' ASL di Reggio Emilia, includendo peso e altezza misurati nelle varie visite se ci sono dati disponibili

```
SELECT nome, cognome, peso, altezza
FROM med.paziente NATURAL LEFT OUTER JOIN med.visita
WHERE provincia_asl = 'RE';
```

Esercizio 3 – Soluzioni (3/3)

8. Elencare nome e cognome di tutti i medici con i pazienti visitati (se ce ne sono)

```
SELECT DISTINCT m.cognome AS cognome_medico, m.nome AS nome_medico,  
                p.cognome AS cognome_paziente, p.nome AS nome_paziente  
FROM med.medico m  
      LEFT OUTER JOIN med.visita v ON m.id_medico = v.id_medico  
      LEFT OUTER JOIN med.paziente p ON v.id_paziente = p.id_paziente;
```

9. Elencare i pazienti che non hanno effettuato visite

```
SELECT cognome, nome  
FROM med.paziente NATURAL LEFT OUTER JOIN med.visita  
WHERE id_visita IS NULL;
```

10. Elencare i pazienti che sono stati visitati da dottoresse

```
SELECT DISTINCT p.cognome, p.nome  
FROM med.paziente p  
      JOIN med.visita v ON p.id_paziente = v.id_paziente  
      JOIN med.medico m ON v.id_medico = m.id_medico  
WHERE m.sesso = 'F';
```


Esercizi dal libro

- Gli esercizi 1 e 2 corrispondono agli esercizi 6.1.3, 6.2.2 e 6.1.4, 6.2.3 del libro
- In aggiunta, si propongono gli esercizi 6.1.2 e 6.2.1

Esercizi dal libro – 6.1.2

Exercise 6.1.2: Write the following queries, based on our running movie database example

```
Movies(title, year, length, genre, studioName, producerC#)
StarsIn(movieTitle, movieYear, starName)
MovieStar(name, address, gender, birthdate)
MovieExec(name, address, cert#, netWorth)
Studio(name, address, presC#)
```

- a) Find Aishwara Rai's birthdate.
- b) Find the address of Film City.
- c) Find all the stars that appeared either in a movie made in 2000 or a movie with "Story" in the title.
- d) Find all the stars who either are female or live in Mumbai (have string Mumbai as a part of their address).
- e) Find all executives worth at least \$20,000,000.

Esercizi dal libro – 6.2.1

Exercise 6.2.1: Using the database schema of our running movie example

```
Movies(title, year, length, genre, studioName, producerC#)
StarsIn(movieTitle, movieYear, starName)
MovieStar(name, address, gender, birthdate)
MovieExec(name, address, cert#, netWorth)
Studio(name, address, presC#)
```

write the following queries in SQL.

- a) Who is the president of Film City?
- b) Who were the male stars in *Monsoon Wedding*?
- c) Which stars appeared in movies produced by Sony in 2005?
- ! d) Which executives are worth more than Subhash Ghai?
- ! e) Which movies are longer than *Bride and Prejudice*?