Ripasso – Operatori aggregati

- Gli aggregati si usano nella **SELECT** o nella **HAVING**
- Funzioni disponibili
  - **COUNT** (*)
  - **COUNT** ([**DISTINCT**] espressione)
  - **SUM** ([**DISTINCT**] espressione)
  - **AVG** ([**DISTINCT**] espressione)
  - **MIN** ([**DISTINCT**] espressione)
  - **MAX** ([**DISTINCT**] espressione)
- **DISTINCT** elimina i valori duplicati prima di operare l'aggregazione
- Tranne **COUNT**, gli altri operatori ignorano valori **NULL**
- Se l'aggregazione è applicata ad un elenco vuoto di valori (dopo aver tolto i **NULL**), il risultato è:
  - 0, nel caso di **COUNT**
  - **NULL** per tutti gli altri operatori aggregati

Ripasso – GROUP BY / HAVING

- Forma generale

```
 SELECT   exp_select_1, ... , exp_select_M
 FROM     ...
[WHERE    ...]
 GROUP BY exp_groupby_1, ..., ex_groupby_N
[HAVING   condizione]
[ORDER BY exp_orderby_1, ..., exp_orderby_Q]
```

- Vincolo:
  - le espressioni exp_select_i della clausola SELECT,
  - le espressioni $exp\_orderby_j$ della clausola ORDER BY e
  - la *condizione* della HAVING

  possono fare uso soltanto di
  - attributi/espressioni $exp\_groupby_k$ citate nella GROUP BY
  - operatori aggregati applicati a qualsiasi attributo (o espressioni di attributi) appartenente a relazioni nella FROM

Ripasso – Sub-queries

- Possono comparire in clausole **SELECT**, **FROM** e **WHERE**
  - **SELECT** (*scalar sub-query*) [**AS**] *var*
  - **FROM** (subquery) [**AS**] *var*
  - **WHERE** ... [**NOT**] exp *operator* (*unary subquery*)
    ... [**NOT**] (exp_1, ..., exp_N) **IN**|**NOT IN** (*subquery*)
    ... [**NOT**] **EXISTS** (*subquery*)
  - *operator* può essere:
    - **<|<=|=|>=|>|<>|LIKE** [**ANY**|**ALL**] → senza **ANY**|**ALL** solo per sub-query scalari
    - **IN** → corrisponde a **= ANY**
    - **NOT IN** → corrisponde a **<> ALL**
- Tipi rilevanti di sotto-query (non mutuamente esclusivi):
  - **unaria**: ritornano tuple con un solo attributo
  - **scalare**: ritornano esattamente una tupla con un solo attributo
  - **correlate**: se referenziano attributi di tabelle (via query variable o nome tabella) definite dalla query esterna

Esercizio 1 – Prodotti

rif. esercizio 6.4.6 (query 2-8)

Sia dato lo schema:

product (<u>model</u>, maker, type)
pc (<u>model</u>, speed, ram, hd, price)
laptop (<u>model</u>, speed, ram, hd, screen, price)
printer (<u>model</u>, color, type, price)

Esprimere in SQL le seguenti query usando funzioni aggregate, GROUP BY e/o HAVING:

1. Trovare la dimensione minima, media e massima degli HD dei PC
2. Trovare il prezzo medio dei laptop veloci almeno 3.0 GHz
3. Trovare il prezzo medio dei PC venduti dal produttore A
4. Trovare, per ogni prezzo, la velocità media dei PC associati
5. Elencare i produttori che vendono almeno tre diversi modelli di PC
6. Trovare la velocità massima dei PC venduti da ciascun produttore
7. Trovare, per ogni velocità di PC sopra 2.5 GHz, la dim. media di HD
8. Trovare, per ogni produttore, la velocità media dei suoi laptop
9. Trovare le velocità di CPU in comune a due o più PC
10. Stilare una classifica dei produttori per numero di prodotti diversi venduti
11. Elencare per ogni produttore quanti tipi diversi di prodotto (pc/laptop/…) vende
12. Trovare il numero di modelli per ciascun produttore e tipologia di prodotto

Esercizio 1 – Soluzioni (1/4)

1. Trovare la dimensione minima, media e massima degli HD dei PC

```
SELECT MIN(hd) AS min_hd, AVG(hd) AS avg_hd, MAX(hd) AS max_hd
FROM   prod.pc
```

2. Trovare il prezzo medio dei laptop veloci almeno 3.0 GHz

```
SELECT AVG(price) AS avg_price
FROM   prod.laptop
WHERE  speed >= 3.0
```

3. Trovare il prezzo medio dei PC venduti dal produttore A

```
SELECT AVG(price) AS avg_price
FROM   prod.pc NATURAL JOIN prod.product
WHERE  maker = 'A'
```

4. Trovare, per ogni prezzo, la velocità media dei PC associati

```
SELECT   price, AVG(speed) AS avg_speed
FROM     prod.pc
GROUP BY price
```

Esercizio 1 – Soluzioni (2/4)

5. Elencare i produttori che vendono almeno tre diversi modelli di PC

```
SELECT   maker
FROM     prod.product
WHERE    type = 'pc'
GROUP BY maker
HAVING   COUNT(*) >= 3
```

6. Trovare la velocità massima dei PC venduti da ciascun produttore

```
SELECT   maker, MAX(speed) AS max_speed
FROM     prod.product NATURAL JOIN prod.pc
GROUP BY maker
```

7. Trovare, per ogni velocità di PC sopra 2.5 GHz, la dim. media di HD

```
SELECT   speed, AVG(hd) AS avg_hd
FROM     prod.pc
WHERE    speed > 2.5
GROUP BY speed
```

Nota: la clausola WHERE può essere trasformata in HAVING preservando la semantica della query.

Esercizio 1 – Soluzioni (3/4)

8. Trovare, per ogni produttore, la velocità media dei suoi laptop

```
SELECT   maker, AVG(speed) AS avg_speed
FROM     prod.product NATURAL JOIN prod.laptop
GROUP BY maker
```

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

```
SELECT speed
FROM prod."PC"
GROUP BY speed
HAVING count(*) >= 2
```

10. Stilare una classifica dei produttori per numero di prodotti diversi venduti

```
SELECT   maker, COUNT(*)
FROM     prod.product
GROUP BY maker
ORDER BY COUNT(*) DESC, maker ASC
```

Esercizio 1 – Soluzioni (4/4)

11. Elencare per ogni produttore quanti tipi diversi di prodotto (pc/laptop/…) vende

```
SELECT   maker, COUNT(DISTINCT type)
FROM     prod.product
GROUP BY maker
```

12. Trovare il numero di modelli per ciascun produttore e tipologia di prodotto

```
SELECT   maker, type, COUNT(*) AS num_models
FROM     prod.product
GROUP BY maker, type
```

Esercizio 2 – Prodotti

rif. esercizi 6.3.1 (query 1-6), 6.4.6 (8, 9)

Sempre a partire dallo schema:

product (<u>model</u>, maker, type)
pc (<u>model</u>, speed, ram, hd, price)
laptop (<u>model</u>, speed, ram, hd, screen, price)
printer (<u>model</u>, color, type, price)

Esprimere in SQL le seguenti query facendo uso di sotto-query:

1. Trovare i produttori di laptop veloci almeno 2.0 GHz
2. Trovare la stampante / le stampanti con il prezzo più alto
3. Trovare i PC più lenti del laptop più veloce
4. Elencare il modello / i modelli di prodotto con il prezzo più basso
5. Trovare il produttore della stampante a colori con il prezzo più alto
6. Trovare produttore/i dei PC più veloci tra quelli con la quantità massima di RAM
7. Trovare i produttori di almeno due computer (PC o laptop) veloci almeno 2 GHz
8. Trovare il prezzo medio di PC e laptop venduti dal produttore D
9. Trovare la dim. media degli HD di PC per ogni produttore di laptop
10. Trovare, per ogni prezzo, la velocità media dei PC associati (senza group by)
11. Trovare la velocità max dei PC venduti da ciascun produttore (senza group by)
12. Trovare il prezzo medio per ciascun produttore e tipologia di prodotto

Esercizio 2 – Soluzioni (1/9)

1. Trovare i produttori di laptop veloci almeno 2.0 GHz

```
SELECT DISTINCT maker
FROM   prod.product
WHERE  model IN (SELECT model
                 FROM   prod.laptop
                 WHERE  speed >= 2.0 )
```

Nota: IN può essere sostituito da = ANY. In alternativa si può fare senza sotto-query:

```
SELECT DISTINCT maker
FROM   prod.product p, prod.laptop l
WHERE  p.model = l.model AND l.speed >= 2.0
```

Altra alternativa (non molto utile in generale) è annidare una sotto-query nella FROM:

```
SELECT DISTINCT ( SELECT maker
                  FROM   prod.product p
                  WHERE  p.model = l.model ) AS maker
FROM   prod.laptop l
WHERE  speed >= 2.0
```

Esercizio 2 – Soluzioni (2/9)

2. Trovare la stampanti / le stampanti con il prezzo più alto

```
SELECT model
FROM   prod.printer
WHERE  price = ( SELECT MAX(price)
                 FROM   prod.printer )
```

oppure:

```
SELECT model
FROM   prod.printer
WHERE  price >= ALL ( SELECT price
                      FROM   prod.printer )
```

oppure senza sotto query:

```
( SELECT model
  FROM   prod.printer )
EXCEPT
( SELECT p1.model
  FROM   prod.printer p1, prod.printer p2
  WHERE  p1.price < p2.price )
```

oppure con outer join:

```
SELECT p1.model
FROM   prod.printer p1 LEFT JOIN prod.printer p2 ON p1.price < p2.price
WHERE  p2.model IS NULL
```

Esercizio 2 – Soluzioni (3/9)

3. Trovare i PC più lenti del laptop più veloce

```
SELECT model
FROM   prod.pc
WHERE  speed < ( SELECT MAX(speed)
                 FROM   prod.laptop )
```

oppure:

```
SELECT model
FROM   prod.pc
WHERE  speed < ANY ( SELECT speed
                     FROM   prod.laptop )
```

oppure senza sotto query:

```
SELECT p.model
FROM   prod.pc p,
       prod.laptop l1 LEFT JOIN prod.laptop l2 ON l1.speed < l2.speed
WHERE  p.speed < l1.speed AND l2.model IS NULL
```

Esercizio 2 – Soluzioni (4/9)

4. Elencare il modello / i modelli di prodotto con il prezzo più basso

```
SELECT model
FROM   ( ( SELECT model, price
           FROM   prod.pc )
         UNION
         ( SELECT model, price
           FROM   prod.laptop )
         UNION
         ( SELECT model, price
           FROM   prod.printer ) ) modelprice
WHERE  price <= ALL ( SELECT price
                      FROM   prod.pc )
       AND price <= ALL ( SELECT price
                          FROM   prod.laptop )
       AND price <= ALL ( SELECT price
                          FROM   prod.printer )
```

Esercizio 2 – Soluzioni (5/9)

5. Trovare il produttore della stampante a colori con il prezzo più alto

```
SELECT maker
FROM   prod.product p JOIN prod.printer s ON p.model = s.model
WHERE  color = TRUE AND price = ( SELECT MAX(price)
                                  FROM   prod.printer
                                  WHERE  color = TRUE )
```

oppure:

```
SELECT maker
FROM   prod.product p JOIN prod.printer s ON p.model = s.model
WHERE  color = TRUE AND price >= ALL ( SELECT price
                                       FROM   prod.printer
                                       WHERE  color = TRUE )
```

oppure:

```
SELECT maker
FROM   prod.product
WHERE  model IN ( SELECT model
                  FROM   prod.printer
                  WHERE  color = TRUE AND
                         price = ( SELECT MAX(price)
                                   FROM prod.printer
                                   WHERE color = TRUE ) )
```

Esercizio 2 – Soluzioni (6/9)

6. Trovare produttore/i dei PC più veloci tra quelli con la quantità massima di RAM

```
SELECT maker
FROM   prod.pc NATURAL JOIN prod.product
WHERE  ram = ( SELECT MAX(ram)
               FROM   prod.pc )
       AND speed = ( SELECT MAX(speed)
                     FROM   prod.pc
                     WHERE  ram = ( SELECT MAX(ram)
                                    FROM   prod.pc ) )
```

Notare che si può sostiturie speed = (SELECT MAX ... con speed >= ALL (SELECT …
In alternativa, si può fare senza sotto-query usando l'outer join:

```
SELECT maker
FROM   prod.product p
       NATURAL JOIN prod.pc p1
       LEFT OUTER JOIN prod.pc p2
       ON p1.ram < p2.ram OR (p1.ram = p2.ram AND p1.speed <
p2.speed)
WHERE  p2.model IS NULL
```

Esercizio 2 – Soluzioni (7/9)

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

```
SELECT   maker
FROM     prod.product NATURAL JOIN ( ( SELECT model
                                       FROM prod.pc
                                       WHERE speed >= 2.0 )
                                     UNION
                                     ( SELECT model
                                       FROM prod.laptop
                                       WHERE speed >= 2.0 ) ) s
GROUP BY maker
HAVING   count(*) >= 2;
```

8. Trovare il prezzo medio di PC e laptop venduti dal produttore D

```
SELECT AVG(price) AS avg_price
FROM   prod.product NATURAL JOIN ( ( SELECT model, price
                                     FROM   prod.pc )
                                   UNION
                                   ( SELECT model, price
                                     FROM   prod.laptop )
                                   UNION
                                   ( SELECT model, price
                                     FROM   prod.printer ) ) p
WHERE  maker = 'D'
```

Esercizio 2 – Soluzioni (8/9)

9. Trovare la dim. media degli HD di PC per ogni produttore di laptop

```
SELECT   maker, AVG(hd) AS avg_hd
FROM     prod.product NATURAL JOIN prod.pc
WHERE    maker IN ( SELECT maker
                    FROM   prod.product
                    WHERE  type = 'laptop' )
GROUP BY maker
```

10. Trovare, per ogni prezzo, la velocità media dei PC associati (senza group by)

```
SELECT   DISTINCT p1.price,
         ( SELECT AVG(speed) AS avg_speed
           FROM   prod.pc p2
           WHERE  p2.price = p1.price ) AS avg_speed
FROM     prod.pc p1
```

Esercizio 2 – Soluzioni (9/9)

11. Trovare la velocità max dei PC venduti da ciascun produttore (senza group by)

```
SELECT DISTINCT maker,
       ( SELECT AVG(speed)
         FROM   prod.product p2 NATURAL JOIN prod.pc p3
         WHERE  p2.maker = p1.maker ) AS avg_speed
FROM   prod.product p1
WHERE  p1.type = 'pc'
```

12. Trovare il prezzo medio per ciascun produttore e tipologia di prodotto

```
SELECT maker, type, AVG(price) AS avg_price
FROM   prod.product
       NATURAL JOIN ( ( SELECT model, price
                        FROM   prod.pc )
                      UNION
                      ( SELECT model, price
                        FROM   prod.laptop )
                      UNION
                      ( SELECT model, price
                        FROM   prod.printer ) ) p
GROUP BY maker, type
```

Esercizio 3 – Navi da battaglia

rif. esercizio 6.4.7 (query 1-6)

Sia dato lo schema:

classes (<u>class</u>, type, country, num_guns, bore, displacement)
ships (<u>name</u>, class, launched)
outcomes (<u>ship</u>, <u>battle</u>, result)
battles (<u>name</u>, date)

Esprimere in SQL le seguenti query usando funzioni aggregate, GROUP BY e/o HAVING:

1. Trovare il numero di classi di incrociatori (bc)
2. Trovare il calibro medio dei cannoni delle classi di corazzate (bb), non considerando quanti navi appartengono ad una classe
3. Trovare il calibro medio dei cannoni delle corazzate (bb), mediando su tutte le navi e non solo sulle classi
4. Trovare per ogni classe il numero di navi affondati appartenenti ad essa
5. Trovare per ogni classe l'anno in cui è stata varata l'ultima nave
6. Per ciascun paese, trovare il peso medio in libbre dei proiettili sparati dalle sue navi (mediare sulle navi) – il peso è circa la metà del calibro in pollici al cubo
7. Trovare le battaglie con almeno due navi dello stesso paese

Esercizio 3 – Soluzioni (1/3)

1. Trovare il numero di classi di incrociatori (bc)

```
SELECT COUNT(*)
FROM   ships.classes
WHERE  type = 'bc'
```

2. Trovare il calibro medio dei cannoni delle classi di corazzate (bb), non considerando quanti navi appartengono ad una classe

```
SELECT AVG(bore)
FROM   ships.classes
WHERE  type = 'bb'
```

3. Trovare il calibro medio dei cannoni delle corazzate (bb), mediando su tutte le navi e non solo sulle classi

```
SELECT AVG(bore)
FROM   ships.classes NATURAL JOIN ships.ships
WHERE  type = 'bb'
```

Esercizio 3 – Soluzioni (2/3)

4. Trovare per ogni classe il numero di navi affondati appartenenti ad essa

```
SELECT   class, COUNT(*) AS num_sunk
FROM     ships.ships JOIN ships.outcomes ON name = ship
WHERE    result = 'sunk'
GROUP BY class
```

5. Trovare per ogni classe l'anno in cui è stata varata l'ultima nave

```
SELECT   class, MAX(launched) AS year
FROM     ships.ships
GROUP BY class
```

6. Per ciascun paese, trovare il peso medio in libbre dei proiettili sparati dalle sue navi (mediare sulle navi) – il peso è circa la metà del calibro in pollici al cubo

```
SELECT   country, AVG(0.5 * bore * bore * bore) AS avg_weight
FROM     ships.classes NATURAL JOIN ships.ships
GROUP BY country
```

Esercizio 3 – Soluzioni (3/3)

7. Trovare le battaglie con almeno due navi dello stesso paesedi cannoni

```
SELECT   o.battle
FROM     ships.classes c JOIN ships.ships s ON c.class = s.class
                         JOIN ships.outcomes o ON s.name = o.ship
GROUP BY o.battle, c.country
HAVING   COUNT(*) >= 2
```

Esercizio 4 – Navi da battaglia

rif. esercizi 6.3.2 (query 1-5), 6.4.7 (query 6)

Sia dato lo schema:

classes (<u>class</u>, type, country, num_guns, bore, displacement)
ships (<u>name</u>, class, launched)
outcomes (<u>ship</u>, <u>battle</u>, result)
battles (<u>name</u>, date)

Esprimere in SQL le seguenti query facendo uso di sotto-query:

1. Trovare i paesi con le classi di navi aventi il calibro maggiore di cannoni
2. Trovare i nomi delle navi con 9 cannoni
3. Trovare le battaglie cui hanno partecipato navi della classe 'South Dakota'
4. Trovare le classi di cui almeno una nave è stata affondata
5. Trovare le navi il cui calibro di cannone era il maggiore tra navi aventi lo stesso numero di cannoni
6. Trovare per ogni classe, avente almeno due navi, il numero di navi affondate appartenenti ad essa

Esercizio 4 – Soluzioni (1/3)

1. Trovare i paesi con le classi di navi aventi il calibro maggiore di cannoni

```
SELECT country
FROM   ships.classes
WHERE  bore = ( SELECT MAX(bore)
                FROM   ships.classes )
```

oppure:

```
SELECT country
FROM   ships.classes
WHERE  bore >= ALL ( SELECT bore
                     FROM   ships.classes )
```

2. Trovare i nomi delle navi con 9 cannoni

```
SELECT name
FROM   ships.ships
WHERE  class IN ( SELECT class
                  FROM   ships.classes
                  WHERE  num_guns = 9 )
```

oppure senza sotto-query:

```
SELECT name
FROM   ships.ships NATURAL JOIN ships.classes
WHERE  num_guns = 9
```

Esercizio 4 – Soluzioni (2/3)

3. Trovare le battaglie cui hanno partecipato navi della classe 'South Dakota'

```
SELECT DISTINCT battle
FROM   ships.outcomes
WHERE  ship IN ( SELECT name
                 FROM   ships.ships
                 WHERE  class = 'South Dakota' )
```

Nota: la query si può scrivere anche senza sotto-query, similmente alla query 9.

4. Trovare le classi di cui almeno una nave è stata affondata

```
SELECT DISTINCT class
FROM   ships.ships
WHERE  name IN ( SELECT ship
                 FROM   ships.outcomes
                 WHERE  result = 'sunk' )
```

oppure senza sotto-query:

```
SELECT DISTINCT class
FROM   ships.ships s JOIN ships.outcomes o ON s.name = o.ship
WHERE  result = 'sunk'
```

Esercizio 4 – Soluzioni (3/3)

5. Trovare le navi il cui calibro di cannone era il maggiore tra navi aventi lo stesso numero di cannoni

```
SELECT   name
FROM     ships.ships NATURAL JOIN ships.classes c
WHERE    c.bore = ( SELECT MAX(bore)
                    FROM   ships.classes c2
                    WHERE  c.num_guns = c2.num_guns )
```

6. Trovare per ogni classe, avente almeno due navi, il numero di navi affondate appartenenti ad essa

```
SELECT   class, COUNT(*) AS num_sunk
FROM     ships.ships JOIN ships.outcomes ON name = ship
WHERE    result = 'sunk' AND
         class IN ( SELECT   class
                    FROM     ships.ships
                    GROUP BY class
                    HAVING   COUNT(*) >= 2 )
GROUP BY class
```

Esercizio 5 – 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 usando funzioni aggregate, GROUP BY e/o HAVING:

1. Ritornare il numero di medici presenti nel DB
2. Ritornare le date della prima e ultima visita in ordine di tempo
3. Ritornare altezza e peso medi misurati nelle visite del 2011
4. Determinare in quante giornate del 2010 si sono effettuate visite
5. Trovare altezza e peso medi per ciascun anno in cui si sono svolte delle visite
6. Per ogni paziente, estrarre nome, cognome, numero. di visite, peso minimo e massimo misurati
7. Elencare il numero di visite effettuate da ciascun medico
8. Elencare nome e cognome dei medici che hanno fatto più di 5 visite nel 2010
9. Ritornare le medie di pressione minima e massima per persone di più di 60 anni
10. Contare il numero di visite nel 2010 raggruppate per ASL del medico

Esercizio 5 – Soluzioni (1/3)

1. Ritornare il numero di medici presenti nel DB

```
SELECT COUNT(*)
FROM   med.medico
```

2. Ritornare le date della prima e ultima visita in ordine di tempo

```
SELECT MIN(data) AS prima_data, MAX(data) AS ultima_data
FROM   med.visita
```

3. Ritornare altezza e peso medi misurati nelle visite del 2011

```
SELECT AVG(altezza) AS altezza_media, AVG(peso) AS peso_medio
FROM   med.visita
WHERE  data >= '2011-01-01' AND data <= '2011-12-31'
```

4. Determinare in quante giornate del 2010 si sono effettuate visite

```
SELECT COUNT(DISTINCT data) AS num_giornate
FROM   med.visita
WHERE  data >= '2010-01-01' AND data <= '2010-12-31'
```

Esercizio 5 – Soluzioni (2/3)

5. Trovare altezza e peso medi per ciascun anno in cui si sono svolte delle visite

```
SELECT   EXTRACT(YEAR FROM data) AS anno, AVG(altezza), AVG(peso)
FROM     med.visita
GROUP BY EXTRACT(YEAR FROM data)
```

6. Per ogni paziente, estrarre nome, cognome, numero. di visite, peso minimo e massimo misurati

```
SELECT   nome, cognome, COUNT(*) AS num_visite,
         MIN(peso) AS peso_minimo, MAX(peso) AS peso_massimo
FROM     med.visita NATURAL JOIN med.paziente
GROUP BY id_paziente, nome, cognome
```

7. Elencare il numero di visite effettuate da ciascun medico

```
SELECT   nome, cognome, COUNT(*) AS num_visite
FROM     med.visita NATURAL JOIN med.medico
GROUP BY id_medico, nome, cognome
```

Esercizio 5 – Soluzioni (3/3)

8. Elencare nome e cognome dei medici che hanno fatto più di 5 visite nel 2010

```
SELECT   nome, cognome
FROM     med.medico NATURAL JOIN med.visita
WHERE    data >= '2010-01-01' AND data <= '2010-12-31'
GROUP BY id_medico, nome, cognome
HAVING   COUNT(*) > 5
```

9. Ritornare le medie di pressione minima e massima per persone di più di 60 anni

```
SELECT AVG(pressione_minima), AVG(pressione_massima)
FROM   med.visita NATURAL JOIN med.paziente
WHERE  EXTRACT(YEAR FROM data) - EXTRACT(YEAR FROM data_nascita) >= 60
```

10. Contare il numero di visite nel 2010 raggruppate per ASL del medico

```
SELECT   codice_asl, provincia_asl, COUNT(*) AS num_visite
FROM     med.medico NATURAL JOIN med.visita
WHERE    data >= '2010-01-01' AND data <= '2010-12-31'
GROUP BY codice_asl, provincia_asl
```

Esercizi dal libro

- Gli esercizi 1, 2, 3 e 4 corrispondono agli esercizi 6.3.1, 6.4.6, 6.3.2 e 6.4.7 del libro
- In aggiunta, si propongono gli esercizi
  - 6.3.3, 6.3.4, 6.3.8, 6.3.9 su query annidate
  - 6.4.1, 6.42 su rimozione duplicati e come ripasso (usare tutti i costrutti a disposizione, comprese funzioni aggregate, dove conveniente)

Esercizi dal libro – 6.3.3

! **Exercise 6.3.3:** Write the query of Fig. 6.10 without any subqueries.

```
Movies(title, year, length, genre, studioName, producerC#)
```

```
1)  SELECT title
2)  FROM Movies Old
3)  WHERE year < ANY
4)      (SELECT year
5)       FROM Movies
6)       WHERE title = Old.title
        );
```

Figure 6.10: Finding movie titles that appear more than once

Esercizi dal libro – 6.3.4

! **Exercise 6.3.4:** Write the following queries without using the intersection or difference operators:

a) The intersection query of Fig. 6.5.

b) The difference query of Example 6.17.

```
Movies(title, year, length, genre, studioName, producerC#)
MovieStar(name, address, gender, birthdate)
MovieExec(name, address, cert#, netWorth)
```

```
(SELECT name, address FROM MovieStar)
    EXCEPT
(SELECT name, address FROM MovieExec);
```

Example 6.17

```
1)  (SELECT name, address
2)   FROM MovieStar
3)   WHERE gender = 'F')
4)       INTERSECT
5)  (SELECT name, address
6)   FROM MovieExec
7)   WHERE netWorth > 10000000);
```

Figure 6.5: Intersecting female movie stars with rich executives

Esercizi dal libro – 6.3.8, 6.3.9

! **Exercise 6.3.8:** Using the database schema

```
Product(maker, model, type)
PC(model, speed, ram, hd, rd, price)
Laptop(model, speed, ram, hd, screen, price)
Printer(model, color, type, price)
```

write a SQL query that will produce information about all products — PC's, laptops, and printers — including their manufacturer if available, and whatever information about that product is relevant (i.e., found in the relation for that type of product).

Exercise 6.3.9: Using the two relations

```
Classes(class, type, country, numGuns, bore, displacement)
Ships(name, class, launched)
```

from our database schema of Exercise 2.4.3, write a SQL query that will produce all available information about ships, including that information available in the `Classes` relation. You need not produce information about classes if there are no ships of that class mentioned in `Ships`.

Esercizi dal libro – 6.4.1

Exercise 6.4.1: Write each of the queries in Exercise 2.4.3 in SQL, making sure that duplicates are eliminated.

```
Classes(class, type, country, numGuns, bore, displacement)
Ships(name, class, launched)
Battles(name, date)
Outcomes(ship, battle, result)
```

a) Find the ships launched prior to 1917.

b) Find the ships sunk in the battle of Surigao Strait.

c) The treaty of Washington in 1921 prohibited capital ships heavier than 35,000 tons. List the ships that violated the treaty of Washington.

d) List the name, displacement, and number of guns of the ships engaged in the battle of North Cape.

e) List all the capital ships mentioned in the database. (Remember that all these ships may not appear in the `Ships` relation.)

f) Give the class names and countries of the classes that carried guns of at least 16-inch bore.

! g) Find those countries that had both battleships and battlecruisers.

! h) Find those ships that "lived to fight another day"; they were damaged in one battle, but later fought in another.

! i) Find the classes that had only one ship as a member of that class.

Esercizi dal libro – 6.4.2

Exercise 6.4.2: Write each of the queries in Exercise 2.4.1 in SQL, making sure that duplicates are eliminated.

```
Product(maker, model, type)
PC(model, speed, ram, hd, price)
Laptop(model, speed, ram, hd, screen, price)
Printer(model, color, type, price)
```

a) Find those manufacturers that sell printers, but not PC's.

b) What PC models have a speed of at least 2.50?

c) Which manufacturers make laptops with a hard disk of at least 120GB?

d) Find the model number and price of all products (of any type) made by manufacturer C.

e) Find the model numbers of all black-and-white laser printers.

! f) Find those hard-disk sizes that occur in two or more PC's.

! g) Find those pairs of PC models that have both the same speed and RAM. A pair should be listed only once; e.g., list (i, j) but not (j, i).

!! h) Find those manufacturers of at least two different computers (PC's or laptops) with speeds of at least 2.20.

!! k) Find the manufacturers who sell exactly three different models of PC.

!! j) Find the manufacturer(s) of the computer (PC or laptop) with the highest available speed.

!! k) Find the manufacturers of PC's with at least three different speeds.