

Esercitazione 3

SQL

c.vallati@iet.unipi.it

DB Riferimento

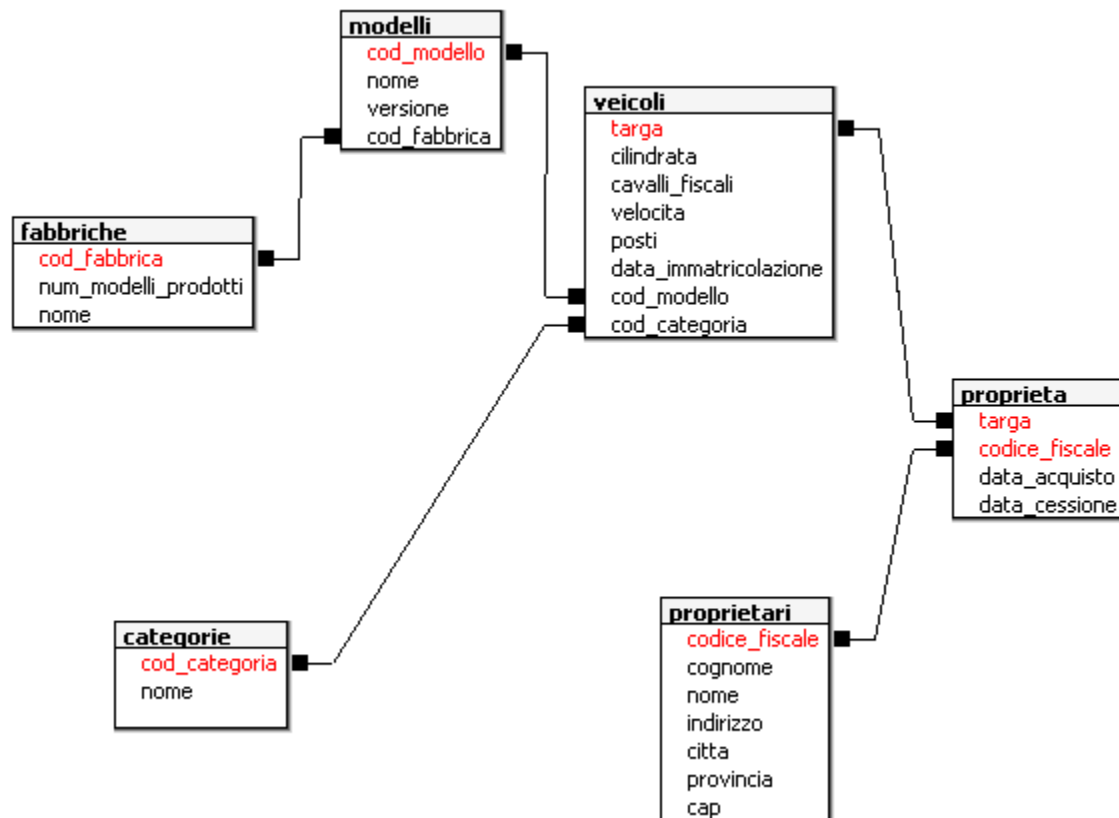
Consideriamo i seguenti schemi di una base di dati relazionale:

- MODELLI (cod_modello,nome,versione,cod_fabbrica)
- VEICOLI
(targa,cilindrata,cavalli_fiscali,velocita,posti,data_immatricolazione,cod_modello,cod_categoria)
- FABBRICHE (cod_fabbrica,nome,num_modelli_prodotti)
- CATEGORIE (cod_categoria,nome)
- PROPRIETARI
(codice_fiscale,cognome,nome,indirizzo,citta,provincia,cap)
- PROPRIETA (targa,codice_fiscale,data_acquisto,data_cessione)

Viene data la seguente istanza della base di dati:

DB Riferimento

- Il database ha il seguente schema relazionale:



Selezionare nome e codice di tutti i Modelli

```
SELECT nome, cod_modello  
FROM modelli
```

Rinominare le colonne nome, cod_modello in 'Nome Modello' e 'Codice Modello'

```
SELECT nome AS "Nome Modello", cod_modello AS "Codice Modello"  
FROM modelli
```

E' equivalente a:

```
SELECT nome "Nome Modello", cod_modello "Codice Modello"  
FROM modelli
```

ES 1

1. Selezionare cognome, nome e indirizzo dei proprietari residenti nella provincia di Pisa
2. Fornire targa, nome e cognome dei proprietari, dei veicoli i cui proprietari risiedono nella provincia di Milano
3. Fornire targa, nome e cognome dei proprietari, dei veicoli i cui proprietari risiedono nella provincia di Torino, trovare solo i veicoli attualmente in possesso

Selezionare cognome, nome e indirizzo dei proprietari residenti nella provincia di Pisa:

```
SELECT cognome, nome, indirizzo  
FROM proprietari  
WHERE provincia = 'PI'
```

Fornire targa, nome e cognome dei proprietari, dei veicoli i cui proprietari risiedono nella provincia di Milano:

```
SELECT proprieta.targa, proprietari.nome, proprietari.cognome
FROM proprieta, proprietari
WHERE proprieta.codice_fiscale = proprietari.codice_fiscale
        AND proprieta.codice_fiscale IN (
                                SELECT codice_fiscale
                                FROM proprietari
                                WHERE provincia = 'MI')
```

Oppure:

```
SELECT proprieta.targa, proprietari.nome, proprietari.cognome
FROM proprieta, proprietari
WHERE proprieta.codice_fiscale = proprietari.codice_fiscale
        AND provincia = 'MI';
```

Oppure:

```
SELECT proprieta.targa, proprietari.nome, proprietari.cognome
FROM proprieta INNER JOIN proprietari ON proprieta.codice_fiscale =
proprietari.codice_fiscale
WHERE proprietari.provincia = 'MI';
```

Oppure:

```
SELECT proprieta.targa, proprietari.nome, proprietari.cognome
FROM proprieta NATURAL JOIN proprietari
WHERE proprietari.provincia = 'MI';
```


ES 2

1. Calcolare il numero dei modelli prodotti da ciascuna fabbrica, ma solo per le fabbriche con almeno 2 modelli
2. Visualizzare per ciascun veicolo la targa, il codice e il nome della categoria
3. Visualizzare il nome delle fabbriche che producono il modello 'PUNTO'

Calcolare il numero dei modelli prodotti da ciascuna fabbrica, ma solo per le fabbriche con almeno 2 modelli:

```
SELECT cod_fabbrica, COUNT(*) "Numero modelli"  
FROM modelli  
GROUP BY cod_fabbrica  
HAVING COUNT(*) >= 2
```

Visualizzare per ciascun veicolo la targa, il codice e il nome della categoria:

```
SELECT targa, categorie.cod_categoria, categorie.nome "Nome Categoria"  
FROM veicoli, categorie  
WHERE veicoli.cod_categoria = categorie.cod_categoria
```

Oppure

```
SELECT targa, categorie.cod_categoria, categorie.nome "Nome Categoria"  
FROM veicoli NATURAL JOIN categorie
```

Oppure

```
SELECT targa, categorie.cod_categoria, categorie.nome "Nome Categoria"  
FROM veicoli JOIN categorie ON veicoli.cod_categoria = categorie.cod_categoria;
```

Visualizzare il nome delle fabbriche che producono il modello 'PUNTO':

```
SELECT fabbriche.nome  
FROM modelli, fabbriche  
WHERE modelli.cod_fabbrica = fabbriche.cod_fabbrica  
AND modelli.nome = 'PUNTO'
```

Oppure

```
SELECT fabbriche.nome  
FROM modelli INNER JOIN fabbriche  
ON modelli.cod_fabbrica = fabbriche.cod_fabbrica  
WHERE modelli.nome = 'PUNTO'
```

Perché' in questo caso il NATURAL JOIN non può essere usato??
Perché tra le relazioni modelli e fabbriche ci sono attributi con lo stesso nome che non sono chiave.

ES 3

1. Visualizzare per ogni categoria il numero di veicoli
2. Visualizzare le targhe dei veicoli immatricolati dopo il 1 gennaio 1995
3. Visualizzare le targhe dei veicoli immatricolati nel 2001

Visualizzare per ogni categoria il numero di veicoli:

```
SELECT categorie.cod_categoria, COUNT(*) AS "Numero veicoli"  
FROM veicoli, categorie  
WHERE veicoli.cod_categoria = categorie.cod_categoria  
GROUP BY categorie.cod_categoria
```

Oppure

```
SELECT categorie.cod_categoria, COUNT(*) AS "Numero veicoli"  
FROM veicoli INNER JOIN categorie  
ON veicoli.cod_categoria = categorie.cod_categoria  
GROUP BY categorie.cod_categoria
```

Oppure

```
SELECT categorie.cod_categoria, COUNT(*) AS "Numero veicoli"  
FROM veicoli NATURAL JOIN categorie  
GROUP BY categorie.cod_categoria
```

Visualizzare le targhe dei veicoli immatricolati dopo il 1 gennaio 1995

```
SELECT targa
FROM veicoli
WHERE data_immatricolazione > '1995-01-01'
```

Visualizzare le targhe dei veicoli immatricolati nel 2001

```
SELECT targa
FROM veicoli
WHERE data_immatricolazione < '2001-12-31' AND data_immatricolazione > '2001-01-01'
```

Oppure

```
SELECT targa
FROM veicoli
WHERE data_immatricolazione BETWEEN '2001-01-01' AND '2001-12-31'
```

Oppure

```
SELECT targa
FROM veicoli
WHERE EXTRACT(YEAR FROM data_immatricolazione) = 2001
```

NOTA:

La funziona EXTRACT(YEAR FROM x) estrae il valore dell'anno da un valore x di tipo DATE

INTERSECT/EXCEPT

Selezionare i cognomi dei proprietari che sono anche nomi (di qualche proprietario)

```
SELECT cognome  
FROM proprietari  
INTERSECT  
SELECT nome  
FROM proprietari
```



Selezionare i nomi dei proprietari che non sono anche cognomi (di qualche proprietario)

```
SELECT nome  
FROM proprietari  
EXCEPT  
SELECT cognome  
FROM proprietari
```



INTERSECT/EXCEPT in MySQL

- MySQL non implementa le operazioni INTERSECT/EXCEPT
- Queste operazioni possono essere comunque implementate usando IN o NOT IN

Selezionare i cognomi dei proprietari che sono anche nomi (di qualche proprietario)

```
SELECT cognome  
FROM proprietari  
WHERE cognome IN (SELECT nome  
                    FROM proprietari)
```

Selezionare i nomi dei proprietari che non sono anche cognomi (di qualche proprietario)

```
SELECT nome  
FROM proprietari  
WHERE nome NOT IN (SELECT cognome  
                    FROM proprietari)
```

ES 4

1. Selezionare i veicoli la cui targa ha un 2 come quarto carattere
2. Selezionare i veicoli la cui targa inizia o termina con la lettera A
3. Trovare i modelli con cilindrata inferiore a 1400 che non sono prodotti dalla fabbrica con codice 001
4. Selezionare tutti i veicoli con cilindrata **inferiore a tutte** le cilindrature dei veicoli con modello 002 (quindi inferiore alla più bassa di queste cilindrature)

Selezionare i veicoli la cui targa ha un 2 come quarto carattere:

```
SELECT *  
FROM veicoli  
WHERE targa LIKE '___2%'
```

Selezionare i veicoli la cui targa inizia o termina con la lettera A:

```
SELECT *  
FROM veicoli  
WHERE targa LIKE 'A%' OR targa LIKE '%A'
```

NOTA:

_ rappresenta un confronto con un carattere arbitrario

% rappresenta un confronto con una stringa di lunghezza arbitraria (eventualmente nulla)

Trovare i modelli con cilindrata inferiore a 1400 che non sono prodotti dalla fabbrica con codice 001

```
SELECT cod_modello
FROM veicoli
WHERE cilindrata < 1400
AND cod_modello <> ALL [NOT IN] (SELECT cod_modello
                                FROM modelli
                                WHERE cod_fabbrica='001')
```

NOTA: <>**ALL** è equivalente a **NOT IN**

Selezionare tutti i veicoli con cilindrata **inferiore a tutte** le cilindrature dei veicoli con modello 002 (quindi inferiore alla più bassa di queste cilindrature)

```
SELECT *  
FROM veicoli  
WHERE cilindrata < ALL (SELECT cilindrata  
                        FROM veicoli  
                        WHERE cod_modello='002')
```

```
SELECT *  
FROM veicoli  
WHERE cilindrata < (SELECT MIN(cilindrata)  
                    FROM veicoli  
                    WHERE cod_modello='002')
```

ES 5

1. Fornire l'elenco dei veicoli del proprietario con codice fiscale 'RSSBNA71G02A403J' specificando la targa, il nome della categoria ed il nome del modello
2. Fornire l'elenco dei veicoli **attualmente** in possesso del proprietario con codice fiscale 'RSSBNA71G02A403J' specificando la targa, il nome della categoria ed il nome del modello

Fornire l'elenco dei veicoli del proprietario con codice fiscale 'RSSBNA71G02A403J' specificando la targa, il nome della categoria ed il nome del modello:

```
SELECT v.targa "Targa", m.nome "Nome Modello", c.nome "Nome categoria"  
FROM veicoli v, modelli m, categorie c  
WHERE v.cod_modello = m.cod_modello AND  
        v.cod_categoria = c.cod_categoria AND  
        v.targa IN  
                (SELECT targa  
                 FROM proprieta  
                 WHERE codice_fiscale = 'RSSBNA71G02A403J')
```

Oppure:

```
SELECT v.targa "Targa", m.nome "Nome Modello",  
        c.nome "Nome Categoria"  
FROM veicoli v INNER JOIN modelli m ON v.cod_modello = m.cod_modello  
        INNER JOIN categorie c ON v.cod_categoria = c.cod_categoria  
WHERE v.targa IN  
        (SELECT targa  
         FROM proprieta  
         WHERE codice_fiscale = 'RSSBNA71G02A403J')
```

Fornire l'elenco dei veicoli **attualmente** in possesso del proprietario con codice fiscale 'RSSBNA71G02A403J' specificando la targa, il nome della categoria ed il nome del modello

```
SELECT v.targa "Targa", c.nome "Nome Categoria ", m.nome "Nome Modello"  
FROM veicoli v, modelli m, categorie c  
WHERE v.cod_modello = m.cod_modello  
AND v.cod_categoria = c.cod_categoria  
AND v.targa IN  
        (SELECT targa  
        FROM proprieta  
        WHERE codice_fiscale = 'RSSBNA71G02A403J' AND  
        data_cessione IS NULL)
```

Oppure:

```
SELECT v.targa "Targa", c.nome "Nome Categoria ", m.nome "Nome Modello"  
FROM veicoli v JOIN modelli m ON v.cod_modello = m.cod_modello  
JOIN categorie c ON v.cod_categoria = c.cod_categoria  
WHERE v.targa IN  
        (SELECT targa  
        FROM proprieta  
        WHERE codice_fiscale = 'RSSBNA71G02A403J' AND  
        data_cessione IS NULL)
```


ES 6

1. Selezionare i nomi dei modelli che hanno cilindrata maggiore della media
2. Selezionare il codice fiscale di tutti gli attuali possessori di veicoli che hanno un numero di cavalli fiscali maggiore della media
3. Selezionare il nome dei proprietari e la targa dei veicoli che hanno avuto due proprietari
4. Trovare i nomi di tutte le categorie per cui è presente almeno un veicolo

Selezionare i nomi dei modelli che hanno cilindrata maggiore della media

```
SELECT DISTINCT nome  
FROM modelli NATURAL JOIN veicoli  
WHERE cilindrata >  
                (SELECT AVG(cilindrata) FROM veicoli)
```

Oppure:

```
SELECT DISTINCT nome  
FROM modelli JOIN veicoli USING(cod_modello)  
WHERE cilindrata >  
                (SELECT AVG(cilindrata) FROM veicoli)
```

Selezionare il codice fiscale di tutti gli attuali possessori di veicoli che hanno un numero di cavalli fiscali maggiore della media

```
SELECT codice_fiscale  
FROM proprieta JOIN veicoli USING(targa)  
WHERE data_cessione IS NULL AND cavalli_fiscali >  
                (SELECT AVG(cavalli_fiscali) FROM veicoli)
```

Oppure:

```
SELECT codice_fiscale  
FROM proprieta NATURAL JOIN veicoli  
WHERE data_cessione IS NULL AND cavalli_fiscali >  
                (SELECT AVG(cavalli_fiscali) FROM veicoli)
```

Selezionare il nome dei proprietari, la targa dei dei veicoli e l'inizio e la fine di ciascun periodo di proprietà per i veicoli che hanno avuto due o più proprietari, ordinati dal primo all'ultimo

```
SELECT nome, targa, data_acquisto AS inizio, data_cessione AS fine
FROM proprieta NATURAL JOIN proprietari
WHERE targa IN
    (SELECT targa FROM proprieta GROUP BY targa HAVING COUNT(*) >= 2)
ORDER BY data_acquisto
```

Oppure:

```
SELECT nome, targa, data_acquisto AS inizio, data_cessione AS fine
FROM proprieta AS pro NATURAL JOIN proprietari AS prop
WHERE EXISTS (SELECT * FROM proprieta WHERE targa = pro.targa HAVING COUNT(*) >= 2)
ORDER BY data_acquisto
```

Trovare i nomi di tutte le categorie per cui è presente almeno un veicolo

```
SELECT nome  
FROM categorie AS c  
WHERE EXISTS (SELECT *  
                FROM veicoli AS v  
                WHERE c.cod_categoria = v.cod_categoria )
```

```
SELECT nome  
FROM categorie  
WHERE cod_categoria =ANY (SELECT cod_categoria  
                          FROM veicoli)
```

```
SELECT nome  
FROM categorie  
WHERE cod_categoria IN (SELECT cod_categoria  
                        FROM veicoli)
```

ES 7

1. Recuperare le fabbriche che producono almeno un modello di auto che ha una velocità maggiore di 200
2. Recuperare il valore medio del numero di passaggi di proprietà per ogni veicolo
3. Elencare (se esistono) i proprietari che hanno almeno un autoveicolo (cat 01) e un motoveicolo (cat 02)
4. Recuperare il tempo medio dopo cui un veicolo viene venduto

Recuperare il nome delle fabbriche che producono almeno un modello di auto che ha una velocità maggiore di 200

```
SELECT DISTINCT f.nome  
FROM fabbriche AS f INNER JOIN modelli AS m ON f.cod_fabbrica = m.cod_fabbrica  
WHERE EXISTS(  
    SELECT *  
    FROM veicoli AS v  
    WHERE m.cod_modello = v.cod_modello  
    AND v.velocita >= 200  
)
```

Oppure:

```
SELECT DISTINCT f.nome  
FROM fabbriche AS f INNER JOIN modelli AS m ON f.cod_fabbrica = m.cod_fabbrica  
WHERE m.cod_modello in (  
    SELECT v.cod_modello  
    FROM veicoli AS v  
    WHERE v.velocita >= 200  
)
```

Recuperare il valor medio del numero di passaggi di proprietà per ogni veicolo

```
SELECT AVG(conto.passaggi)  
FROM (  
    SELECT v.targa, COUNT(*) AS passaggi  
    FROM veicoli AS v NATURAL JOIN proprieta AS p  
    GROUP BY v.targa  
) AS conto
```


Elencare (se esistono) i proprietari che hanno almeno un autoveicolo (cat 01) e un motoveicolo (cat 02)

```
SELECT cognome, nome
FROM proprietari AS p
WHERE EXISTS(
    SELECT *
    FROM veicoli NATURAL JOIN proprieta
    WHERE p.codice_fiscale = codice_fiscale AND cod_categoria = '01'
)
AND
EXISTS(
    SELECT *
    FROM veicoli NATURAL JOIN proprieta
    WHERE p.codice_fiscale = codice_fiscale AND cod_categoria = '02'
)
```

Recuperare il tempo medio dopo cui ogni modello viene venduto. Mostrare solamente il cod_modello e il numero di giorni

```
SELECT cod_modello, AVG(numero_giorni_proprieta)
FROM veicoli NATURAL JOIN (
    SELECT targa, data_cessione - data_acquisto AS numero_giorni_proprieta
    FROM proprieta
    WHERE data_cessione IS NOT NULL
) AS giorni_vendita
GROUP BY cod_modello
```

Oppure:

```
SELECT cod_modello, AVG(data_cessione - data_acquisto) AS
numero_giorni_proprieta
FROM proprieta NATURAL JOIN veicoli
WHERE data_cessione IS NOT NULL
GROUP BY cod_modello
```