

Esercizi SQL

1) Semplicissimi esercizi introduttivi

Impiegato(nome, titolo, età, salario, dip)

Acquisto(idCliente, dataDiOrdinazione,
item, quantità, prezzo)

Cliente(id, nome, cognome, città, stato)

1) Selezionare il nome, l'età e il salario degli impiegati con più di 50 anni

```
SELECT nome, età, salario  
FROM impiegato  
WHERE età > 50
```

2) Estrarre tutte le informazioni relative agli acquisti di tende (cioè dove item=Tenda)

```
SELECT *  
FROM acquisto  
WHERE item = 'Tenda'
```

3) *Di tutti i prodotti comprati dal cliente con id=10449 mostrare nome, prezzo e id del cliente*

```
SELECT id, item, prezzo  
FROM acquisto  
WHERE id=10449
```

4) *Trovare il nome, il titolo e il dipartimento degli impiegati il cui titolo comincia con “Ing”*

```
SELECT nome, titolo, dip  
FROM impiegato  
WHERE titolo LIKE 'Ing%'
```

5) *Nome, titolo e salario di tutti coloro che hanno il titolo di programmatore e un salario ≥ 50000*

```
SELECT nome, titolo, salario  
FROM impiegato  
WHERE salario  $\geq$  50000  
AND Titolo = 'Programmatore'
```

6) *Nome e salario di tutti coloro che lavorano al dipartimento 'Vendite' o 'Programmazione'*

```
SELECT nome, salario  
FROM impiegato  
WHERE dip = 'Vendite'  
OR dip = 'Programmazione'
```

7) Selezionare tutte le diverse età degli impiegati

```
SELECT distinct età  
FROM impiegato
```

8) Selezionare lo stipendio medio degli impiegati

```
SELECT avg(salario)  
FROM impiegato
```

*12) Estrarre i dati degli impiegati del dipartimento
“Vendite” in ordine (ascendente) di stipendio*

```
SELECT *  
FROM impiegato  
WHERE dip = 'Vendite'  
ORDER BY salario
```

*13) Estrarre i dati degli impiegati del dipartimento
“Vendite” in ordine ascendente di stipendio e
discendente di età (a pari stipendio)*

```
SELECT *  
FROM impiegato  
WHERE dip = 'Vendite'  
ORDER BY salari, età DESC
```


*14) Estrarre tutti i dati degli impiegati
'Hernandez', 'Jones', 'Roberts', 'Ruiz'*

```
SELECT *  
FROM impiegato  
WHERE nome IN ('Hernandez', 'Jones',  
'Roberts', 'Ruiz')
```

Oppure

```
WHERE nome='Hernandez' OR nome='Jones'  
OR nome='Roberts' OR nome='Ruiz'
```

Si noti che

*15) Estrarre tutti i dati di tutti gli impiegati **tranne**
'Hernandez', 'Jones', 'Roberts' e 'Ruiz'*

Si può esprimere come...

```
SELECT *  
FROM impiegato  
WHERE nome NOT IN  
('Hernandez', 'Jones', 'Roberts', 'Ruiz')
```

16) Estrarre tutti i dati degli impiegati di età compresa tra 30 e 40 anni

```
SELECT *  
FROM impiegato  
WHERE età BETWEEN 30 AND 40
```

Oppure

```
WHERE età >= 30 AND età <= 40
```

17) Estrarre, per ogni acquisto, il nome del cliente e il prezzo del prodotto

```
SELECT nome, prezzo  
FROM cliente, acquisto  
WHERE cliente.id = acquisto.idcliente
```

oppure

```
SELECT nome, prezzo  
FROM cliente join acquisto  
      on cliente.id = acquisto.idcliente
```

2)

AEROPORTO (Città, Nazione, NumPiste)

VOLO (IdVolo, GiornoSett, CittàPart, OraPart,
CittàArr, OraArr, TipoAereo)

AEREO (TipoAereo, NumPasseggeri, QtaMerci)

18) Trovare le città da cui partono voli diretti a Roma, ordinate alfabeticamente

```
SELECT distinct CittàPar  
FROM Volo  
WHERE CittàArr= 'Roma'  
ORDER BY CittàPar
```

19) Trovare le città con un aeroporto di cui non è noto il numero di piste

```
SELECT Città  
FROM Aeroporto  
WHERE NumPiste IS NULL
```

*20) Di ogni volo misto (merci e passeggeri)
estrarre il codice e i dati relativi al trasporto*

```
select  IdVolo, NumPasseggeri, QtaMerci  
from    VOLO as V, AEREO as A  
where   V.TipoAereo = A.TipoAereo and  
        NumPasseggeri > 0 and QtaMerci > 0
```

```
select IdVolo, NumPasseggeri, QtaMerci  
from  VOLO V inner join AEREO A  
      on V.TipoAereo = A.TipoAereo  
where NumPasseggeri > 0 and QtaMerci > 0
```

(sintassi equivalente)

21) Le nazioni di partenza e arrivo del volo AZ274

```
SELECT A1.Nazione, A2.Nazione
FROM (AEROPORTO A1 join VOLO
      on A1.Città=CittàArr)
join AEROPORTO A2
   on CittàPar=A2.Città
WHERE IdVolo= 'AZ274'
```


23) Trovare l'aeroporto italiano con il maggior numero di piste

Ad esempio si può usare una query annidata

```
select Città, NumPiste
from AEROPORTO
where Nazione = 'Italia' and
      NumPiste = ( select max(numPiste)
                  from AEROPORTO
                  where Nazione = 'Italia' )
```

27) *Trovare il numero **totale** di partenze internazionali del giovedì da tutti gli aeroporti*

```
SELECT count(*)
FROM   (AEROPORTO as A1 join VOLO
        on CittàPar=A1.Città)
        join AEROPORTO as A2
        on CittàArr=A2.Città
WHERE  A1.Nazione <> A2.Nazione
        and GiornoSett = 'Giovedì'
```

28) Trovare il numero di aeroporti che hanno almeno una partenza internazionale al giovedì

```
SELECT count( distinct CittàPar )  
FROM (AEROPORTO as A1 join VOLO  
on CittàPar=A1.Città)  
join AEROPORTO as A2  
on CittàArr=A2.Città  
WHERE A1.Nazione <> A2.Nazione  
and GiornoSett = 'Giovedì'
```

26) *Gli aeroporti da cui **partono** voli internazionali*

```
SELECT distinct CittàPar
FROM   (AEROPORTO as A1 join VOLO
        on CittàPar=A1.Città)
join AEROPORTO as A2
on CittàArr=A2.Città
WHERE  A1.Nazione <> A2.Nazione
```

Il distinct è “essenziale” per la chiarezza e leggibilità del risultato

18)

CD (CDNumber, Title, Year, Price)

Track (CDNumber, PerformanceCode,
trackNo)

Recording (Performance, SongTitle, Year)

Composer (CompName, SongTitle)

Singer (SingerName, PerformanceCode)

I cantautori (persone che hanno scritto e cantato la stessa canzone).

```
SELECT SingerName
FROM ( Singer S join Recording R on
      S.PerformanceCode=R.Performance )
     join Composer C on R.SongTitle=C.SongTitle
WHERE SingerName=CompName
```

I titoli dei dischi che contengono canzoni di cui non si conosce l'anno di registrazione

```
SELECT Title
FROM CD
    JOIN Track AS T ON
        CD.CDNumber=T.CDNumber
    JOIN Recording AS R ON
        T.PerformanceCode=
            R.PerformanceCode
WHERE R.Year IS NULL
```

I pezzi del disco con numero di serie 78574, ordinati per numero progressivo, con indicazione degli interpreti.

```
SELECT TrackNo, SingerName
FROM Track JOIN Singer ON
        Track.PerformanceCode=
                Singer.PerformanceCode
WHERE CDNumber=78574
ORDER BY TrackNo
```


Gli autori e i cantanti **puri**, ovvero autori che non hanno mai registrato una canzone e cantanti che non hanno mai scritto una canzone

Scriviamo due query separate,
poi effettuiamo l'unione

```
SELECT CompName
FROM Composer
WHERE CompName NOT IN
( SELECT SingerName
  FROM Singer )
```

UNION

```
SELECT SingerName
FROM Singer
WHERE SingerName NOT IN
( SELECT CompName
  FROM Composer)
```

Gli autori che non hanno mai inciso
una canzone **scritta da loro** e i
cantanti che non hanno scritto
nessuna delle canzoni **che hanno
cantato**

*(Si noti che **include** i cantanti e i cantautori puri)*

```
SELECT CompName
FROM Composer
WHERE CompName NOT IN
(SELECT CompName
FROM Composer AS C
JOIN Recording AS R ON
C.SongTitle=R.SongTiltle
JOIN Singer ON
Performance=PerformanceCode
WHERE CompName=SingerName )
```

```
SELECT SingerName
FROM Singer
WHERE SingerName NOT IN
(SELECT SingerName
FROM Singer
JOIN Recording ON
Performance=PerformanceCode
JOIN Composer ON
Recording.SongTitle=
Composer.SongTitle
WHERE CompName=SingerName)
```

```
select CompName
from Composer
where CompName NOT IN
(select CompName
 from Composer C join Recording R on C.SongTitle=R.SongTitle
      join Singer on Performance=PerformanceCode
 where CompName=SingerName )
```

UNION

```
select SingerName
from Singer
where SingerName NOT IN
(select SingerName
 from (Singer join Recording on Performance=PerformanceCode)
      join Composer C on Recording.SongTitle=C.SongTitle
 where CompName=SingerName)
```

6)

Catalogo di prodotti e fornitori

Fornitori (CodiceForn, Nome,
Indirizzo, Città)

Prodotti (CodiceProd, Nome,
Marca, Modello)

Catalogo (CodiceForn,
CodiceProd, Costo)

40) Trovare i nomi dei fornitori di prodotti IBM

```
SELECT DISTINCT F.Nome  
FROM   Fornitori AS F,  
       Prodotti AS P,  
       Catalogo AS C  
WHERE  F.CodiceForn=C.CodiceForn  
       AND C.CodiceProd=P.CodiceProd  
       AND P.Marca='IBM';
```


41) Codici dei prodotti distribuiti da almeno 2 fornitori

```
SELECT distinct C.CodiceProd  
FROM Catalogo C, Catalogo C1  
WHERE C.CodForn <> C1.CodForn  
      AND C.CodiceProd=C1.CodiceProd
```

43) Trovare il codice dei prodotti più costosi tra quelli distribuiti dai fornitori presenti a Roma

```
SELECT DISTINCT C.CodiceProd
FROM Fornitori AS F, Catalogo AS C
WHERE F.CodiceForn=C.CodiceForn
AND F.Città='Roma' AND C.Costo=
(SELECT MAX(Costo)
FROM Fornitori AS F1, Catalogo AS C1
WHERE F1.CodiceForn=C1.CodiceForn
AND F1.Città='Roma')
```

7)

Informazioni relative alle partite dei campionati di serie A, B, C. Ad ogni *partita giocata* è assegnato un codice univoco (per comodità).

Ogni partita è (ovviamente) giocata da due squadre

Partita(Codice, Campionato, Serie, Data, Stadio)

Risultato (Squadra, Partita, Punti, Reti)

*Si noti che la registrazione dei punti è **ridondante**, in quanto deducibile dal confronto delle reti di squadre diverse nella stessa partita, ma molto comoda (occorrono 2 join per confrontare le reti)*

45) *Squadre che hanno giocato sempre e solo in serie A*

```
SELECT distinct Squadra
FROM Risultato
WHERE Squadra NOT IN
( SELECT distinct Squadra
  FROM Risultato join Partita
    on Partita=Codice
  WHERE Serie <> "A")
```

47) *Il calendario, con i risultati, delle 34 giornate del campionato di serie A '01/'02 in ordine cronologico*

```
select      Data, r1.Squadra, r2.Squadra,  
           r1.Reti, r2.Reti, Stadio  
from        Partita, Risultato r1, Risultato r2  
where       Serie='A' and  
           Campionato='01-02' and  
           r1.Partita = Codice and  
           r2.Partita = Codice and  
order by   Data
```

Con uno schema diverso è MOLTO più facile

Partita (SqCasa, SqOsp, Data, RetiCasa,
RetiOsp, Campionato, Serie)

48) Il calendario, con i risultati, delle 34 giornate del campionato di serie A '01/'02 in ordine cronologico

```
select    Data, SqCasa, SqOsp, RetiCasa
from      Partita
where     Serie='A' and Campionato='01-02'
order by  Data
```

8)

REGISTA (Nome, DataNascita, Nazionalità)

ATTORE (Nome, DataNascita, Nazionalità)

INTERPRETA (Attore, Film, Personaggio)

FILM (Titolo, NomeRegista, Anno)

PROIEZIONE (NomeCin, CittàCin, TitoloFilm)

CINEMA (Città, NomeCinema, #Sale, #Posti)

50) Nazionalità dei registi che hanno diretto qualche film nel 1992 ma non hanno diretto film nel 1993

```
SELECT distinct Nazionalità
FROM REGISTA
WHERE Nome IN ( SELECT NomeRegista
                  FROM FILM
                  WHERE Anno = 1992)
AND Nome NOT IN ( SELECT NomeRegista
                    FROM FILM
                    WHERE Anno = 1993)
```


In alternativa si può usare EXCEPT (a patto di discriminare in base alla chiave) ma assolutamente NON un join e la condizione

~~WHERE Anno = 1992 AND Anno <> 1993~~

perché la WHERE agisce a livello di TUPLA

```
SELECT Nazionalità
```

```
FROM REGISTA
```

```
WHERE Nome IN
```

```
( SELECT NomeRegista
```

```
FROM FILM WHERE Anno = 1992
```

```
EXCEPT
```

```
SELECT NomeRegista
```

```
FROM FILM WHERE Anno = 1993 )
```

NON si può usare la EXCEPT direttamente se nella target list non è incluso l'attributo discriminante per l'esclusione

~~SELECT Nazionalità~~

~~FROM FILM join REGISTA on NomeRegista=Nome~~

~~WHERE Anno = 1992~~

~~EXCEPT~~

~~SELECT Nazionalità~~

~~FROM FILM join REGISTA on NomeRegista=Nome~~

~~WHERE Anno = 1993~~

Un solo regista di nazionalità X che abbia girato un film nel '93 farebbe sparire dalla soluzione il valore X, anche se molti altri suoi connazionali fossero tali da soddisfare l'interrogazione.

Attenzione: in SQL gli operatori insiemistici eliminano i duplicati (come se davanti a Nazionalità ci fosse distinct)

51) Le date di nascita dei registi che hanno diretto film in proiezione sia a Torino sia a Milano

```
select distinct DataNascita
from REGISTA join FILM
    on Nome=NomeRegista
where Titolo in ( SELECT TitoloFilm
                  FROM PROIEZIONE
                  WHERE CittàCin='Milano')
AND Titolo in ( SELECT TitoloFilm
                  FROM PROIEZIONE
                  WHERE CittàCin='Torino')
```

54) Trovare gli attori che hanno interpretato più personaggi in uno stesso film (+ di 1 !!)

select distinct P1.Attore

from INTERPRETA P1 , INTERPRETA P2

where P1.Attore = P2.Attore

and P1.Film = P2.Film

and P1.Personaggio <> P2.Personaggio

57) Film di registi italiani in cui non recita nessun italiano

```
select Titolo
from FILM join REGISTA on Nome=NomeRegista
where Nazionalità = Italiana
```

EXCEPT

```
select NomeFilm
from INTERPRETA join ATTORE on Nome=NomeAttore
where Nazionalità = Italiana
```

```
select Titolo
from FILM join REGISTA on Nome=NomeRegista
where Nazionalità = Italiana and
      Titolo NOT IN ( select NomeFilm
                        from INTERPRETA join ATTORE
                        on Nome=NomeAttore
                        where Nazionalità = Italiana )
```

58) Registi che hanno recitato in (almeno) uno dei film che hanno diretto

```
SELECT DISTINCT NomeRegista  
FROM FILM join INTERPRETA  
on Titolo=Film  
WHERE NomeRegista=Attore
```

13)

Dato il seguente schema relazionale

OFFICINA(ID, Sede, Direttore, Fatturato-92)

RIPARAZIONE (
ID_OFFICINA, ID_CLIENTE, ID_RIPAR,
Tipo_Riparazione, Tipo_Auto, Data, Pagamento)

CLIENTE(ID_CLIENTE, Nome, Indirizzo)

Determinare nome e indirizzo dei clienti che hanno pagato qualche riparazione relativa ad auto Fiat il cui importo (individuale) sia superiore a 3000 € in officine fuori Milano

```
SELECT Nome, Indirizzo
FROM Cliente
WHERE ID_cliente IN
  ( SELECT ID_cliente
    FROM Riparazione
    WHERE Pagamento > 3000 AND
      Tipo_auto LIKE “%FIAT%” AND
      ID_officina NOT IN
        ( SELECT Numero
          FROM Officina
          WHERE Sede LIKE “%MILANO”))
```


Determinare nome e indirizzo dei clienti che hanno pagato più di dieci riparazioni nel 1991 ma non hanno pagato alcuna riparazione nel 1992

```
SELECT Nome, Indirizzo
FROM Cliente
WHERE ID_cliente IN
((SELECT ID_cliente
  FROM Riparazione
  WHERE Data between 1.1.1991 and 31.12.1991
  GROUP BY ID_cliente
  HAVING count(*) > 10)
  EXCEPT
  (SELECT ID_cliente
  FROM Riparazione
  WHERE Data between 1.1.1992 and 31.12.1992))
```

15)

Cliente(CodCliente, Nome, Indirizzo, Città)

Ordine(Numero, CodCliente, Data, Importo)

PartiOrdine(NroOrdine, CodProdotto,
Quantità, PrezzoUnitario)

Prodotto(Codice, Descrizione, QtaMagazzino)

Nomi dei clienti che non hanno
mai ordinato prodotti che siano
stati ordinati anche dalla ditta
“Brambilla”

```
SELECT Nome  
FROM Cliente  
WHERE Nome not in
```

```
( SELECT nome  
FROM cliente c, ordine o, partiordine  
WHERE c.codcliente=o.codcliente  
AND numero=nroordine AND codprodotto in  
( SELECT codprodotto  
FROM cliente c, ordine o, partiordine po1  
WHERE nome="Brambilla" AND  
c.codcliente=o.codcliente AND  
numero=nroordine))
```

16) Dato il seguente schema relazionale:

AGENTE(Nome, Percentuale)

ARTICOLO(Nome, Descrizione, Tipo)

CLIENTE(Nome, Indirizzo, Telefonoi)

VENDITA(Nome-Comp, Nome-Art,
Nome-Ag, Data, Quantità, Importo,
Validità)

Selezionare gli Agenti che hanno venduto qualche articolo di tipo “scarpa” ma non hanno venduto nulla a clienti il cui indirizzo è a Milano

```
SELECT V.NomeAg
FROM ARTICOLO A, VENDITA V, CLIENTE C
WHERE A.Nome=V.NomeArt
and C.Nome=V.NomeComp and A.Tipo="scarpa"
and Vendite.NomeAg NOT IN
(SELECT Vendite.NomeAg
FROM Cliente, Vendita
WHERE Cliente.Nome=Vendite.NomeComp
AND Cliente.Indirizzo like “%Milano%” )
```

17) Dato il seguente schema relazionale:

AUTORE(NOME, COGNOME,

Data-N, Nazionalita)

AUTORELIBRO(NOME, COGNOME,

SEGNATURA)

LIBRO(SEGNATURA, Scaffale,

Argomento, Lingua)

Selezionare il COGNOME degli autori tedeschi di libri in italiano con argomento “filosofia” o “logica”

```
SELECT Cognome  
FROM   Autore A, Libro L, Autorelibro AL,  
WHERE  A.Nome=AL.Nome  
and A.Cognome=AL.Cognome  
and A.Segnatura=L.Segnatura  
and Nazionalita="tedesca"  
and Lingua="italiano" and  
(Argomento="filosofia" OR Argomento="logica)
```

Selezionare la data di nascita degli autori italiani di libri in inglese di Argomento “informatica”, che non sono autori di libri di Argomento “matematica”


```
SELECT Data_N
FROM Autore AS A JOIN Autorelibro ON
  (A.Nome=Autorelibro.Nome AND
   A.Cognome=Autorelibro.Cognome)
JOIN Libro ON
  (Autorelibro.Segnatura=Libro.Segnatura)
WHERE Nazionalita="IT" AND Lingua="ING"
AND Argomento="INF" AND A NOT IN
( SELECT *
  FROM Autorelibro AS AL JOIN Libro AS L
  ON (AL.Segnatura=L.Segnatura)
  WHERE A.Nome=AL.Nome
        AND A.Cognome=AL.Cognome
        AND Argomento="MATEMATICA")
```