



Politecnico
di Torino



Esercizi svolti in aula e/o proposti

Riflettiamo ... (25/9/2025)

- Dato il seguente schema relazionale:

EDITORE(CodE, NomeEditore, Indirizzo, Città)

PUBBLICAZIONE(CodP, Titolo, NomeAutore, NumPagine*, CodE)

LIBRERIA(CodL, NomeLibreria, Indirizzo, Città,)

VENDITA(CodP, CodL, Data, CopieVendute)



- Scrivere:
 - a) Il testo di un'interrogazione che siamo in grado di risolvere con le istruzioni introdotte finora
 - b) Il testo di un'interrogazione che NON siamo in grado di risolvere con le istruzioni introdotte finora

Esercizi proposti (da voi)

EDITORE(CodE, NomeEditore, Indirizzo, Città)

PUBBLICAZIONE(CodP, Titolo, NomeAutore, NumPagine*, CodE)

LIBRERIA(CodL, NomeLibreria, Indirizzo, Città,)

VENDITA(CodP, CodL, Data, CopieVendute)

- Codice libro e nome editore dei libri con più di 500 pagine.
- Fornire i titoli delle pubblicazioni vendute a luglio 2025.
- Il nome libreria in cui sono stati venduti libri di Italo Calvino.
- Ordinare in ordine alfabetico i titoli dei libri dell'editore Sellerio che sono stati venduti.
- Trova i titoli delle pubblicazioni dell'editore Sellerio vendute nelle librerie di Torino.
- Trovare gli editori e le librerie che hanno lo stesso indirizzo, visualizzando i loro nomi.

Esercizi proposti (da voi)

EDITORE(CodE, NomeEditore, Indirizzo, Città)

PUBBLICAZIONE(CodP, Titolo, NomeAutore, NumPagine*, CodE)

LIBRERIA(CodL, NomeLibreria, Indirizzo, Città,)

VENDITA(CodP, CodL, Data, CopieVendute)

- Trovare il codice del libro e il nome dell'editore dei libri con più di 500 pagine **o che potrebbe averne più di 500**.

```
SELECT P.CodP, E.NomeEditore
FROM EDITORE AS E, PUBBLICAZIONE P
WHERE E.CodE = P.CodE
AND P.NumPagine > 500 OR P.NumPagine IS NULL;
```

EDITORE join PUBBLICAZIONE (CodP, Titolo, NomeAutore, NumPagine*,
CodE, NomeEditore, Indirizzo, Città)

Esercizi proposti (da voi)

EDITORE(CodE, NomeEditore, Indirizzo, Città)

PUBBLICAZIONE(CodP, Titolo, NomeAutore, NumPagine*, CodE)

LIBRERIA(CodL, NomeLibreria, Indirizzo, Città,)

VENDITA(CodP, CodL, Data, CopieVendute)

- Trovare il codice del libro e il nome dell'editore dei libri con più di 500 pagine **venduti il 12 settembre 2025**.

```
SELECT P.CodP, E.NomeEditore
FROM EDITORE AS E, PUBBLICAZIONE P, VENDITA V
WHERE E.CodE = P.CodE AND P.CodP = V.CodP
AND P.NumPagine > 500 AND V.Data = "2025-09-12" ;
```

Esercizi proposti (da voi)

EDITORE(CodE, NomeEditore, Indirizzo, Città)

PUBBLICAZIONE(CodP, Titolo, NomeAutore, NumPagine*, CodE)

LIBRERIA(CodL, NomeLibreria, Indirizzo, Città,)

VENDITA(CodP, CodL, Data, CopieVendute)

- Fornire i titoli delle pubblicazioni vendute a luglio 2025.

```
SELECT P.Titolo
```

```
FROM PUBBLICAZIONE P, VENDITA V
```

```
WHERE P.CodP = V.CodP
```

```
AND V.Data >= "2025-07-01" AND V.Data <= "2025-07-31" ;
```

Esercizi proposti (da voi)

EDITORE(CodE, NomeEditore, Indirizzo, Città)

PUBBLICAZIONE(CodP, Titolo, NomeAutore, NumPagine*, CodE)

LIBRERIA(CodL, NomeLibreria, Indirizzo, Città,)

VENDITA(CodP, CodL, Data, CopieVendute)

- Il nome libreria in cui sono stati venduti libri di Italo Calvino
- Cioè: [...] è stato venduto almeno un libro di Italo Calvino

```
SELECT NomeLibreria
FROM PUBBLICAZIONE P, LIBRERIA L, VENDITA V
WHERE P.CodP = V.CodP AND L.CodL = V.CodL
AND P.NomeAutore = "Italo Calvino";
```

Esercizi proposti (da voi)

EDITORE(CodE, NomeEditore, Indirizzo, Città)

PUBBLICAZIONE(CodP, Titolo, NomeAutore, NumPagine*, CodE)

LIBRERIA(CodL, NomeLibreria, Indirizzo, Città,)

VENDITA(CodP, CodL, Data, CopieVendute)

- Ordinare in ordine alfabetico i titoli dei libri dell'editore Sellerio **che sono stati venduti.**

```
SELECT P.Titolo
FROM PUBBLICAZIONE P, EDITORE E, VENDITA V
WHERE P.CodE = E.CodE AND V.CodP = P.CodP
AND E.NomeEditore = "Sellerio"
ORDER BY P.Titolo ASC;
```


Esercizi proposti (da voi)

EDITORE(CodE, NomeEditore, Indirizzo, Città)

PUBBLICAZIONE(CodP, Titolo, NomeAutore, NumPagine*, CodE)

LIBRERIA(CodL, NomeLibreria, Indirizzo, Città,)

VENDITA(CodP, CodL, Data, CopieVendute)

- Trovare gli editori e le librerie che hanno lo stesso indirizzo, visualizzando i loro nomi.

```
SELECT E.NomeEditore, L.NomeLibreria  
FROM EDITORE E, LIBRERIA L  
WHERE E.Indirizzo = L.Indirizzo  
AND E.Città = L.Città;
```

Esercizi proposti (da voi)

EDITORE(CodE, NomeEditore, Indirizzo, Città)

PUBBLICAZIONE(CodP, Titolo, NomeAutore, NumPagine*, CodE)

LIBRERIA(CodL, NomeLibreria, Indirizzo, Città,)

VENDITA(CodP, CodL, Data, CopieVendute)

- Calcolare quante pubblicazioni ha ciascun editore.
- Trovare il libro con il numero massimo di pagine /
Il libro con meno pagine.
- Dimmi il titolo con il numero di copie vendute /
Trovare per ogni libro il numero totale di copie vendute.
- Seleziona da vendita e libreria e editore il numero di vendite per ogni libreria divise per editore.

Esercizi proposti (da voi)

EDITORE(CodE, NomeEditore, Indirizzo, Città)

PUBBLICAZIONE(CodP, Titolo, NomeAutore, NumPagine*, CodE)

LIBRERIA(CodL, NomeLibreria, Indirizzo, Città,)

VENDITA(CodP, CodL, Data, CopieVendute)

- Calcolare quante pubblicazioni ha ciascun editore.

```
SELECT CodE, COUNT(*)  
FROM PUBBLICAZIONE  
GROUP BY CodE;
```

```
SELECT CodE, COUNT(CodP)  
FROM PUBBLICAZIONE  
GROUP BY CodE;
```

Esercizi proposti (da voi)

EDITORE(CodE, NomeEditore, Indirizzo, Città)

PUBBLICAZIONE(CodP, Titolo, NomeAutore, NumPagine*, CodE)

LIBRERIA(CodL, NomeLibreria, Indirizzo, Città,)

VENDITA(CodP, CodL, Data, CopieVendute)

- Calcolare quante pubblicazioni ha ciascun editore, tenendo solo conto degli editori che hanno almeno 15 pubblicazioni.

```
SELECT CodE, COUNT(*)  
FROM PUBBLICAZIONE  
GROUP BY CodE  
HAVING COUNT(*) >= 15;
```

Riflettiamo ... (1/10/2025)

- Dato il seguente schema relazionale:

CALCIATORE(CodC, Nome, Cognome, Indirizzo, CittàResidenza)

SQUADRA(CodS, NomeS, Categoria, AnnoFondazione)

PREMIO(CodP, CodC, DataConsegna, LuogoConsegna, Tipologia)

CALCIATORE-SQUADRA(CodC, CodS, DataInizio, DataFine, NumPresenze, NumGol)

- Scrivere:
 - a) Il testo di un'interrogazione che richieda GROUP BY (senza HAVING)
 - b) Il testo di un'interrogazione che richieda GROUP BY...HAVING



Esercizi proposti (da voi)

CALCIATORE(CodC, Nome, Cognome, Indirizzo, CittàResidenza)

SQUADRA(CodS, NomeS, Categoria, AnnoFondazione)

PREMIO(CodP, CodC, DataConsegna, LuogoConsegna, Tipologia)

CALCIATORE-SQUADRA(CodC, CodS, DataInizio, DataFine, NumPresenze, NumGol)

- Per ogni squadra, trovare il numero di giocatori
- Trovare il numero di goal totali "segnati" da una squadra
- Quanti gol una certa squadra ha fatto in totale
- Trovare i calciatori che hanno segnato più di 10 gol in totale
- Visualizzare il numero di gol, se questo è maggiore di 3, fatti da ogni giocatore in un periodo di tempo fissato
- **Trovare il capocannoniere di ciascuna squadra (il giocatore con più gol)**

Esercizi proposti (da voi)

CALCIATORE(CodC, Nome, Cognome, Indirizzo, CittàResidenza)

SQUADRA(CodS, NomeS, Categoria, AnnoFondazione)

PREMIO(CodP, CodC, DataConsegna, LuogoConsegna, Tipologia)

CALCIATORE-SQUADRA(CodC, CodS, DataInizio, DataFine, NumPresenze, NumGol)

- Per ogni squadra, trovare il numero di giocatori
- Trovare il numero di goal totali "segnati" da una squadra
- Quanti gol una certa squadra ha fatto in totale
- Trovare i calciatori che hanno segnato più di 10 gol in totale
- Visualizzare il numero di gol, se questo è maggiore di 3, fatti da ogni giocatore in un periodo di tempo fissato
- Trovare i calciatori che hanno militato sia nella Juventus sia nel Torino
- Trovare il capocannoniere di ciascuna squadra (il giocatore con più gol)

Esercizi proposti (da voi)

CALCIATORE(CodC, Nome, Cognome, Indirizzo, CittàResidenza)

SQUADRA(CodS, NomeS, Categoria, AnnoFondazione)

PREMIO(CodP, CodC, DataConsegna, LuogoConsegna, Tipologia)

CALCIATORE-SQUADRA(CodC, CodS, DataInizio, DataFine, NumPresenze, NumGol)

- Per ogni squadra, trovare il numero di giocatori

```
SELECT CodS, COUNT(DISTINCT CodC)
FROM CALCIATORE-SQUADRA CS
GROUP BY CodS;
```

```
SELECT NomeS, COUNT(DISTINCT CodC)
FROM CALCIATORE-SQUADRA CS, SQUADRA S
WHERE CS.CodS = S.CodS
GROUP BY S.CodS, NomeS;
```


Esercizi proposti (da voi)

CALCIATORE(CodC, Nome, Cognome, Indirizzo, CittàResidenza)

SQUADRA(CodS, NomeS, Categoria, AnnoFondazione)

PREMIO(CodP, CodC, DataConsegna, LuogoConsegna, Tipologia)

CALCIATORE-SQUADRA(CodC, CodS, DataInizio, DataFine, NumPresenze, NumGol)

- Trovare il numero di goal totali "segnati" da una squadra
- Quanti gol una certa squadra ha fatto in totale

```
SELECT NomeS, SUM(NumGoal)
FROM CALCIATORE-SQUADRA CS, SQUADRA S
WHERE CS.CodS = S.CodS
GROUP BY S.CodS, NomeS;
```

Esercizi proposti (da voi)

CALCIATORE(CodC, Nome, Cognome, Indirizzo, CittàResidenza)

SQUADRA(CodS, NomeS, Categoria, AnnoFondazione)

PREMIO(CodP, CodC, DataConsegna, LuogoConsegna, Tipologia)

CALCIATORE-SQUADRA(CodC, CodS, DataInizio, DataFine, NumPresenze, NumGol)

- Trovare il numero di goal totali "segnati" da ogni squadra
- Quanti gol il Torino ha fatto in totale

```
SELECT NomeS, SUM(NumGoal)
FROM CALCIATORE-SQUADRA CS, SQUADRA S
WHERE CS.CodS = S.CodS
GROUP BY S.CodS, NomeS;
```

```
SELECT SUM(NumGoal)
FROM CALCIATORE-SQUADRA CS, SQUADRA S
WHERE CS.CodS = S.CodS AND NomeS = «Torino»;
```

Esercizi proposti (da voi)

CALCIATORE(CodC, Nome, Cognome, Indirizzo, CittàResidenza)

SQUADRA(CodS, NomeS, Categoria, AnnoFondazione)

PREMIO(CodP, CodC, DataConsegna, LuogoConsegna, Tipologia)

CALCIATORE-SQUADRA(CodC, CodS, DataInizio, DataFine, NumPresenze, NumGol)

- Trovare i calciatori che hanno segnato più di 10 gol in totale
- Trovare i calciatori che hanno militano più volte nella stessa squadra

```
SELECT Nome, Cognome, SUM(NumGol)
FROM CALCIATORE-SQUADRA CS, CALCIATORE C
WHERE CS.CodC = C.CodC
GROUP BY C.CodC, Nome, Cognome
HAVING SUM(NumGol) > 10;
```

Esercizi proposti (da voi)

CALCIATORE(CodC, Nome, Cognome, Indirizzo, CittàResidenza)

SQUADRA(CodS, NomeS, Categoria, AnnoFondazione)

PREMIO(CodP, CodC, DataConsegna, LuogoConsegna, Tipologia)

CALCIATORE-SQUADRA(CodC, CodS, DataInizio, DataFine, NumPresenze, NumGol)

- Trovare i calciatori che hanno militato sia nella Juventus sia nel Torino

```
SELECT Nome, Cognome
```

```
FROM CALCIATORE C, CALCIATORE-SQUADRA CS1, SQUADRA S1,
```

```
    CALCIATORE-SQUADRA CS2, SQUADRA S2
```

```
WHERE CS1.CodS=S1.CodS AND CS2.CodS=S2.CodS
```

```
AND C.CodC=CS1.CodC AND CS1.CodC = CS2.CodC (il giocatore è lo stesso)
```

```
AND S1.NomeS=«Juve» AND S2.NomeS= «Toro»;
```

Il punto

- Quante ore ho dedicato finora a Basi Dati (escluse le ore di lezione)?
- Quanti esercizi ho fatto?
- Quali concetti/istruzioni non mi sono chiare?



Esercizi proposti

VELISTI (Vid, VNome, Esperienza, DataNascita)

PRENOTAZIONI (Vid, Bid, Data)

BARCHE (Bid, BNome, Colore*)

Trovare i codici e i nomi dei velisti che hanno prenotato una barca rossa e una barca verde

```
SELECT V.Vid, VNome
FROM PRENOTAZIONI P1, BARCHE B1, PRENOTAZIONI P2, BARCHE B2, VELISTI V
WHERE P1.Bid = B1.Bid AND P2.Bid = B2.Bid AND P1.Vid = P2.Vid
AND V.Vid = P1.Vid
AND B1.Colore = 'Rosso' AND B2.Colore = 'Verde';
```

Esercizi proposti

VELISTI (Vid, VNome, Esperienza, DataNascita)

PRENOTAZIONI (Vid, Bid, Data)

BARCHE (Bid, BNome, Colore*)

Trovare i codici e i nomi dei velisti che hanno prenotato una barca rossa e una barca verde

```
SELECT Vid, VNome
FROM VELISTI V
WHERE Vid IN (SELECT Vid
              FROM PRENOTAZIONI P, BARCHE B
              WHERE B.Bid = P.Bid AND Colore='Rosso')
AND Vid IN (SELECT Vid
            FROM PRENOTAZIONI P, BARCHE B
            WHERE B.Bid = P.Bid AND Colore='Verde' );
```

Esercizi proposti

IMPIEGATO (CodImp, Cognome, Nome, CodDip, Stipendio)

DIPARTIMENTO (CodDip, Denominazione, Città)

Selezionare il codice del dipartimento in cui lavorano sia Rossi che Bianchi

```
SELECT CodDip
FROM IMPIEGATO
WHERE Cognome = 'Rossi'
AND CodDip IN (SELECT CodDip
                FROM IMPIEGATO
                WHERE Cognome = 'Bianchi' ) ;
```


Esercizi proposti

IMPIEGATO (CodImp, Cognome, Nome, CodDip, Stipendio)

DIPARTIMENTO (CodDip, Denominazione, Città)

Selezionare i codici dei dipartimenti che non erogano stipendi maggiori di 10.000 euro

-> Selezionare i codici dei dipartimenti che non appartengono all'insieme dei dipartimenti che erogano almeno uno stipendio maggiore di 10.000

Selezionare i codici dei dipartimenti che erogano solo stipendi maggiori di 10.000 euro

-> Selezionare i codici dei dipartimenti che (a) non appartengono all'insieme dei dipartimenti che erogano almeno uno stipendio minore di 10.000 e (b) che erogano stipendi

Esercizi proposti

IMPIEGATO (CodImp, Cognome, Nome, CodDip, Stipendio)

DIPARTIMENTO (CodDip, Denominazione, Città)

Selezionare i codici dei dipartimenti che non erogano stipendi maggiori di 10.000 euro

-> Selezionare i codici dei dipartimenti che non appartengono all'insieme dei dipartimenti che erogano almeno uno stipendio maggiore di 10.000

```
SELECT DISTINCT CodDip
FROM IMPIEGATO
WHERE CodDip NOT IN (SELECT CodDip
                     FROM IMPIEGATO
                     WHERE Stipendio > 10000) ;
```

Esercizi proposti

IMPIEGATO (CodImp, Cognome, Nome, CodDip, Stipendio)

DIPARTIMENTO (CodDip, Denominazione, Città)

Selezionare i codici dei dipartimenti che erogano solo stipendi maggiori di 10.000 euro

-> Selezionare i codici dei dipartimenti che (a) non appartengono all'insieme dei dipartimenti che erogano almeno uno stipendio minore di 10.000 e (b) che erogano stipendi

```
SELECT DISTINCT CodDip
FROM IMPIEGATO
WHERE CodDip NOT IN (SELECT CodDip
                     FROM IMPIEGATO
                     WHERE Stipendio < 10000);
```

Esercizi proposti

IMPIEGATO (CodImp, Cognome, Nome, CodDip, Stipendio)

DIPARTIMENTO (CodDip, Denominazione, Città)

Selezionare i codici dei dipartimenti che erogano solo stipendi maggiori di 10.000 euro

-> Selezionare i codici dei dipartimenti che (a) non appartengono all'insieme dei dipartimenti che erogano almeno uno stipendio minore di 10.000 e (b) che erogano stipendi

```
SELECT CodDip
FROM DIPARTIMENTO
WHERE CodDip NOT IN (SELECT CodDip
                     FROM IMPIEGATO
                     WHERE Stipendio < 10000)
AND CodDip IN (SELECT CodDip
              FROM IMPIEGATO) ;
```

Esercizi proposti

VELISTI (Vid, VNome, Esperienza, DataNascita)

PRENOTAZIONI (Vid, Bid, Data)

BARCHE (Bid, BNome, Colore*)

Trovare i codici e i nomi dei velisti che non hanno mai prenotato una barca rossa

```
SELECT Vid, Vnome
FROM VELISTI V
WHERE NOT EXISTS (SELECT *
                   FROM PRENOTAZIONI P, BARCHE B
                   WHERE P.Bid = B.Bid AND Colore = 'Rosso'
                   AND P.Vid = V.Vid );
```

Esercizi proposti

VELISTI (Vid, VNome, Esperienza, DataNascita)

PRENOTAZIONI (Vid, Bid, Data)

BARCHE (Bid, BNome, Colore*)

Trovare i codici e i nomi dei velisti che non hanno mai prenotato una barca rossa

```
SELECT Vid, VNome
FROM VELISTI V
WHERE Vid NOT IN
    (SELECT Vid
     FROM PRENOTAZIONI P, BARCHE B
     WHERE B.Bid = P.Bid
     AND Colore='Rosso');
```

Esercizi proposti

FILM (CodFilm, Titolo, AnnoProduzione, Nazionalità, Regista, Genere)

PROIEZIONI (CodProiezione, CodFilm, CodSala, Incasso, DataProiezione)

SALE (CodSala, Posti, Nome, Città)

Trovare i titoli dei film che non sono mai stati proiettati a Torino

Riflettiamo ... (15/10/2025)

- Dato il seguente schema relazionale:

CALCIATORE(CodC, Nome, Cognome, Indirizzo, CittàResidenza)

SQUADRA(CodS, NomeS, Categoria, AnnoFondazione)

PREMIO(CodP, CodC, DataConsegna, LuogoConsegna, Tipologia)

CALCIATORE-SQUADRA(CodC, CodS, DataInizio, DataFine, NumPresenze, NumGol)

- Scrivere:
 - a) Il testo di un'interrogazione che richieda l'utilizzo di NOT o NOT EXISTS
 - ~~b) Il testo di un'interrogazione che richieda la correlazione fra interrogazioni (ma non EXISTS/NOT EXISTS)~~

