



Esercizi svolti in aula e/o proposti

Riflettiamo ... (25/9/2025)

- Dato il seguente schema relazionale:

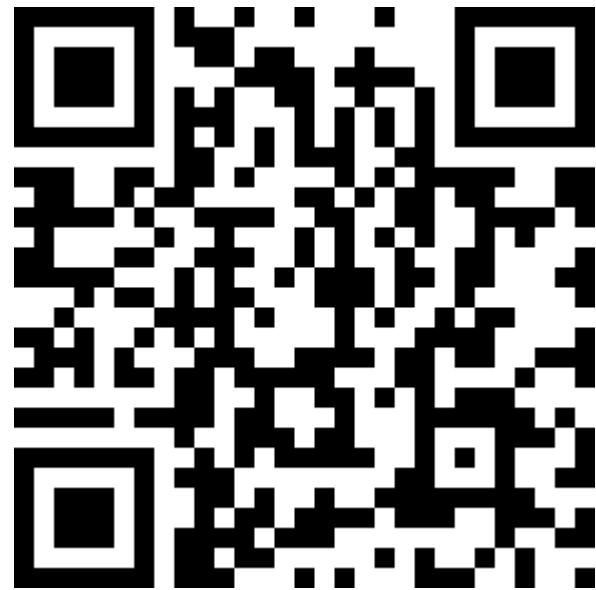
EDITORE(CodE, NomeEditore, Indirizzo, Città)

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 militato 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 B1.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');
```

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



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

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

... il cui codice non appartiene all'insieme dei codici dei film proiettati a Torino

```
SELECT Titolo
```

```
FROM FILM
```

```
WHERE CodFilm NOT IN
```

```
(SELECT CodFilm
```

```
    FROM PROIEZIONI P, SALE S
```

```
    WHERE P.CodSala = S.CodSala
```

```
    AND Città = "Torino") ;
```

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
... per cui non esiste una proiezione di quel film a Torino

```
SELECT Titolo
FROM FILM F
WHERE NOT EXISTS
  (SELECT *
   FROM SALA S, PROIEZIONE P
   WHERE Città = 'Torino'
   AND S.CodSala = P.CodSala
   AND P.CodFilm = F.CodFilm);
```

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

... per cui Torino non appartiene all'insieme delle città in cui è stato proiettato
quel film

```
SELECT Titolo
FROM FILM F
WHERE 'Torino' NOT IN
  (SELECT Città
   FROM SALE S, PROIEZIONI P
   WHERE S.CodSala = P.CodSala
   AND P.CodFilm = F.CodFilm);
```

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 nomi dei calciatori che non hanno vinto nessun premio
- Mostrare i codici e i cognomi dei calciatori che non hanno mai segnato un goal con una squadra fondata prima del 1970
- Trovare il nome delle squadre i cui giocatori non hanno mai vinto un premio nel 2018
- (Trovare i calciatori che non hanno mai segnato alcun 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)

- Trovare i nomi dei calciatori che non hanno vinto nessun premio

```
SELECT Nome  
FROM CALCIATORE  
WHERE CodC NOT IN  
(SELECT CodC  
FROM PREMIO);
```

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)

- Mostrare i codici e i cognomi dei calciatori che non hanno mai segnato un goal con una squadra fondata prima del 1970

```
SELECT CodC, Cognome  
FROM CALCIATORE  
WHERE CodC NOT IN  
(SELECT CodC  
    FROM CALCIATORE-SQUADRA CS, SQUADRA S  
    WHERE CS.CodS = S.CodS AND NumGol >= 1  
    AND AnnoFondazione < 1970);
```

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 nome delle squadre i cui giocatori non hanno mai vinto un premio nel 2018

```
SELECT NomeS  
FROM SQUADRA  
WHERE CodS NOT IN  
(SELECT CodS  
  CALCIATORE-SQUADRA CS, PREMIO P  
  WHERE CS.CodC = P.CodC  
  AND DataConsegna >= '2018-01-01'  
  AND DataConsegna <= '2028-12-31');
```

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 nome delle squadre i cui giocatori non hanno mai vinto un premio nel 2018

```
SELECT NomeS  
FROM SQUADRA S  
WHERE NOT EXISTS  
(SELECT *  
    FROM PREMIO P, CALCIATORE_SQUADRA CS  
    AND P.CodC = CS.CodC AND CS.CodS = S.CodS  
    AND DataConsegna >= '2018-01-01'  
    AND DataConsegna <= '2028-12-31');
```

Esercizi proposti

CORSO (CodCorso, NomeC, Anno, Semestre)

ORARIO-LEZIONI (CodCorso, GiornoSettimana, Oralnizio, OraFine, Aula)

- a) Trovare le aule in cui non si tengono mai lezioni di corsi del primo anno
- b) Trovare le aule in cui si tengono solo lezioni di corsi del primo anno

Esercizi proposti

CORSO (CodCorso, NomeC, Anno, Semestre)

ORARIO-LEZIONI (CodCorso, GiornoSettimana, Oralnizio, OraFine, Aula)

- a) Trovare le aule in cui non si tengono mai lezioni di corsi del primo anno

```
SELECT DISTINCT Aula
FROM ORARIO-LEZIONI OL1
WHERE NOT EXISTS
  (SELECT *
   FROM CORSO C, ORARIO-LEZIONI OL
   WHERE C.CodCorso = OL.CodCorso
   AND Anno = 1
   AND OL.Aula = OL1.Aula);
```

Esercizi proposti

CORSO (CodCorso, NomeC, Anno, Semestre)

ORARIO-LEZIONI (CodCorso, GiornoSettimana, Oralnizio, OraFine, Aula)

- a) Trovare le aule in cui non si tengono mai lezioni di corsi del primo anno

```
SELECT DISTINCT Aula
FROM ORARIO-LEZIONI
WHERE Aula NOT IN
    (SELECT Aula
     FROM CORSO C, ORARIO-LEZIONI OL
      WHERE C.CodCorso = OL.CodCorso
        AND Anno = 1);
```

Esercizi proposti

CORSO (CodCorso, NomeC, Anno, Semestre)

ORARIO-LEZIONI (CodCorso, GiornoSettimana, Oralnizio, OraFine, Aula)

- a) Trovare le aule in cui si tengono solo lezioni di corsi del primo anno

```
SELECT DISTINCT Aula
FROM ORARIO-LEZIONI
WHERE Aula NOT IN
    (SELECT Aula
     FROM CORSO C, ORARIO-LEZIONI OL
      WHERE C.CodCorso = OL.CodCorso
        AND Anno <> 1);
```

Esercizi proposti

SALA_RIUNIONI (CodS, NumeroMaxPosti, Proiettore)

PRENOTAZIONE_SALA (CodS, Data, OraInizio, OraFine, CodDip)

DIPENDENTE (CodDip, Nome, Cognome, DataNascita, Città)

- Visualizzare il codice e il numero massimo di posti delle sale dotate di proiettore che sono state prenotate almeno 15 volte per riunioni che iniziano prima delle ore 15:00, ma non sono mai state prenotate per riunioni che cominciano dopo le ore 20:00

Esercizi proposti

SALA_RIUNIONI (CodS, NumeroMaxPosti, Proiettore)

PRENOTAZIONE_SALA (CodS, Data, Oralnizio, OraFine, CodDip)

DIPENDENTE (CodDip, Nome, Cognome, DataNascita, Città)

- Visualizzare il codice e il numero massimo di posti delle sale dotate di proiettore che sono state prenotate almeno 15 volte per riunioni che iniziano prima delle ore 15:00, ma non sono mai state prenotate per riunioni che cominciano dopo le ore 20:00

```
SELECT S.CodS, NumeroMaxPosti  
FROM SALA_RIUNIONI S, PRENOTAZIONE_SALA P  
WHERE Proiettore = 'sì'  
AND S.CodS = P.CodS AND Oralnizio < 15:00  
AND S.CodS NOT IN  
(SELECT CodS FROM PRENOTAZIONE_SALA  
WHERE Datalnizio > 20:00)  
GROUP BY S.CodS, NumeroMaxPosti  
HAVING COUNT(*) >= 15;
```

Esercizi proposti

SALA_RIUNIONI (CodS, NumeroMaxPosti, Proiettore)

PRENOTAZIONE_SALA (CodS, Data, Oralnizio, OraFine, CodDip)

DIPENDENTE (CodDip, Nome, Cognome, DataNascita, Città)

- Visualizzare il codice e il numero massimo di posti delle sale dotate di proiettore che sono state prenotate almeno 15 volte per riunioni che iniziano prima delle ore 15:00, ma non sono mai state prenotate per riunioni che cominciano dopo le ore 20:00

```
SELECT CodS, NumeroMaxPosti  
FROM SALA_RIUNIONI  
WHERE Proiettore = 'sì'  
AND CodS IN  
    (SELECT CodS  
     FROM PRENOTAZIONE_SALA  
     WHERE Oralnizio < 15:00  
     GROUP BY CodS  
     HAVING COUNT(*) >= 15)  
AND CodS NOT IN  
    (SELECT CodS FROM PRENOTAZIONE_SALA  
     WHERE Datalnizio > 20:00);
```

Esercizi proposti

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

ATTORI (CodAttore, Nome, AnnoNascita, Nazionalità)

RECITA (CodAttore, CodFilm, Ruolo)

- Trovare il nome degli attori che hanno recitato in tutti i film di Tarantino.

```
SELECT Nome  
FROM ATTORI A, RECITA R, FILM F  
WHERE A.CodAttore = R.CodAttore AND F.CodFilm = R.CodFilm  
AND Regista = 'Tarantino'  
GROUP BY A.CodAttore , Nome  
HAVING COUNT(*) =  
    (SELECT COUNT(*) FROM FILM  
WHERE Regista = 'Tarantino');
```

Riflettiamo ... (22/10/2025)

- 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 il testo di un'interrogazione richieda l'utilizzo dell'operazione di «divisione»



Esercizi proposti

UTENTE (CodUtente, NomeUtente, DataIscrizione, TipoAbbonamento)

BRANO (CodBrano, Titolo, Artista, Album, Genere, Durata)

ASCOLTO (CodUtente, CodBrano, TimestampAscolto, DurataAscolto)

- Visualizzare il nome e il tipo di abbonamento degli utenti hanno ascoltato **tutti** i brani dell'album “By the Way” dei “Red Hot Chili Peppers”, ma che **non** hanno ascoltato nessun brano dell'album “I'm with You” (sempre dei “Red Hot Chili Peppers”)

Esercizi proposti

UTENTE (CodUtente, NomeUtente, Dataiscrizione, TipoAbbonamento)

BRANO (CodBrano, Titolo, Artista, Album, Genere, Durata)

ASCOLTO (CodUtente, CodBrano, TimestampAscolto, DurataAscolto)

```
SELECT NomeUtente, TipoAbbonamento
FROM UTENTE U, ASCOLTO A, BRANO B
WHERE U.CodUtente = A.CodUtente AND A.CodBrano = B.CodBrano
AND Artista = 'Red Hot Chili Pepper' AND Album = 'By the Way'
AND CodUtente NOT IN
    (SELECT CodUtente
     FROM ASCOLTO A1, BRANO B1
     WHERE A1.CodBrano = B1.CodBrano
     AND Artista = 'Red Hot Chili Pepper' AND Album = "I'm with you" )
GROUP BY U.CodUtente, NomeUtente, TipoAbbonamento
HAVING COUNT(DISTINCT CodBrano) =
    (SELECT COUNT(*) FROM BRANO
     WHERE Artista = 'Red Hot Chili Pepper' AND Album = 'By the Way' );
```

Visualizzare il nome e il tipo di abbonamento degli utenti hanno ascoltato tutti i brani dell'album "By the Way" dei "Red Hot Chili Peppers", ma che non hanno ascoltato nessun brano dell'album "I'm with You" (sempre dei "Red Hot Chili Peppers")

Esercizi proposti

GUIDA (CodGuida, Nome, Cognome, Nazionalità)

TIPO-VISITA (CodTipoVisita, Monumento, Durata, Città)

GRUPPO (CodGR, NumeroPartecipanti, Lingua)

VISITA-GUIDATA-EFFETTUATA (CodGR, Data, Oral, CodTipoVisita, CodGuida)

- Tra i monumenti per cui sono state effettuate almeno 10 visite guidate, visualizzare il monumento che è stato **visitato complessivamente dal maggior numero di persone**

Monumento	Tot Part	SUM (part sugli visit)
Colosseo		

↑ MAX

Esercizi proposti

GUIDA (CodGuida, Nome, Cognome, Nazionalità)

TIPO-VISITA (CodTipoVisita, Monumento, Durata, Città)

GRUPPO (CodGR, NumeroPartecipanti, Lingua)

VISITA-GUIDATA-EFFETTUATA (CodGR, Data, Oral, CodTipoVisita, CodGuida)

Tra i monumenti per cui sono state effettuate almeno 10 visite guidate, visualizzare il monumento che è stato visitato complessivamente dal maggior numero di persone

```
SELECT Monumento, SUM(NumPartecipanti) AS TotPart
FROM TIPO-VISITA TV, VISITA-GUIDATA-EFFETTUATA VG, GRUPPO G
WHERE TV.CodTipoVisita = VG.CodTipoVisita AND VG.CodGR = G.CodGR
GROUP BY Monumento
HAVING COUNT(*) >= 10
```

Esercizi proposti

GUIDA (CodGuida, Nome, Cognome, Nazionalità)

TIPO-VISITA (CodTipoVisita, Monumento, Durata, Città)

GRUPPO (CodGR, NumeroPartecipanti, Lingua)

VISITA-GUIDATA-EFFETTUATA (CodGR, Data, Oral, CodTipoVisita, CodGuida)

Tra i monumenti per cui sono state effettuate almeno 10 visite guidate, visualizzare il monumento che è stato visitato complessivamente dal maggior numero di persone

WITH

VISITE-MONUMENTO AS (

```
SELECT Monumento, SUM(NumeroPartecipanti) As TotPart
FROM VISITA-GUIDATA-EFFETTUATA V, TIPO-VISITA T, GRUPPO G
WHERE V.CodTipoVisita=T.CodTipoVisita AND V.CodGR=G.CodGR
GROUP BY Monumento
HAVING COUNT(*) >= 10,
```

SELECT Monumento

FROM VISITE-MONUMENTO

WHERE TotPart = (

```
SELECT MAX(TotPart) AS MaxTotPart
FROM VISITE-MONUMENTO);
```

Esercizi proposti

GUIDA (CodGuida, Nome, Cognome, Nazionalità)

TIPO-VISITA (CodTipoVisita, Monumento, Durata, Città)

GRUPPO (CodGR, NumeroPartecipanti, Lingua)

VISITA-GUIDATA-EFFETTUATA (CodGR, Data, Oral, CodTipoVisita, CodGuida)

Tra i monumenti per cui sono state effettuate almeno 10 visite guidate, visualizzare il monumento che è stato visitato complessivamente dal maggior numero di persone

WITH

VISITE-MONUMENTO AS (

```
SELECT Monumento, SUM(NumeroPartecipanti) As TotPart  
FROM VISITA-GUIDATA-EFFETTUATA V, TIPO-VISITA T, GRUPPO G  
WHERE V.CodTipoVisita=T.CodTipoVisita AND V.CodGR=G.CodGR  
GROUP BY Monumento  
HAVING COUNT(*) >= 10),
```

MAXVISITE AS (

```
SELECT MAX(TotPart) AS MaxTotPart  
FROM VISITE-MONUMENTO )
```

SELECT Monumento

FROM VISITE-MONUMENTO, MAXVISITE

WHERE TotPart = MaxTotPart;

Esercizi proposti

GUIDA (CodGuida, Nome, Cognome, Nazionalità)

TIPO-VISITA (CodTipoVisita, Monumento, Durata, Città)

GRUPPO (CodGR, NumeroPartecipanti, Lingua)

VISITA-GUIDATA-EFFETTUATA (CodGR, Data, Oral, CodTipoVisita, CodGuida)

Tra i monumenti per cui sono state effettuate almeno 10 visite guidate, visualizzare il monumento che è stato visitato complessivamente dal maggior numero di persone

```
SELECT Monumento
FROM TIPO-VISITA TP, VISITA-GUIDATA-EFFETTUATA VGE, GRUPPO G
WHERE TP.CodTipoVisita=VGE.CodTipoVisita AND G.CodGR=VGE.CodGR
GROUP BY Monumento
HAVING COUNT(*) >=10 AND SUM(NumeroPartecipanti) =
(SELECT MAX(TOTPart)
FROM (SELECT SUM(NumeroPartecipanti) AS TOTPart
      FROM TIPO-VISITA TP, VISITA-GUIDATA-EFFETTUATA VGE, GRUPPO G
      WHERE TP.CodTipoVisita=VGE.CodTipoVisita
      AND G.CodGR=VGE.CodGR
      GROUP BY Monumento
      HAVING COUNT(*) >= 10) AS TOTM);
```

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 le pubblicazioni che sono vendute in tutte le librerie
- Trovare il nome degli editori che hanno sede a Torino che hanno pubblicato tutti i libri di Calvino con un numero di pagine maggiore di 200
- Trovare il nome della libreria che ha venduto meno libri fra giugno e dicembre 2024

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 titolo delle pubblicazioni che sono vendute in tutte le librerie

```
SELECT Titolo  
FROM PUBBLICAZIONE P, VENDITA V  
WHERE P.CodP = V.CodP  
GROUP BY CodP, Titolo  
HAVING COUNT(DISTINCT CodL) = (SELECT COUNT(*)  
                                FROM LIBRERIA);
```

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 nome degli editori che hanno sede a Torino che hanno pubblicato tutti i libri di Calvino con un numero di pagine maggiore di 200

```
SELECT NomeEditore
  FROM PUBBLICAZIONE P, EDITORE E
 WHERE P.CodE = E.CodE AND Città = 'Torino'
   AND NomeAutore = 'Calvino' AND NumPagine > 200
 GROUP BY CodE, NomeEditore
 HAVING COUNT(*) =
        (SELECT COUNT(*)
          FROM PUBBLICAZIONE
         WHERE NomeAutore = 'Calvino' AND NumPagine > 200);
```

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 nome della libreria che ha venduto meno libri fra giugno e dicembre 2024

WITH

NUM-LIBRI AS (

```
SELECT CodL, COUNT(DISTINCT CodP) AS LibriVenduti  
FROM LIBRERIA L, VENDITA V  
WHERE L.CodL= V.CodL  
AND Data >= '2024-06-01' AND Data <= '2024-12-01'  
GROUP BY CodL, NomeLibreria )
```

SELECT CodL, NomeLibreria

FROM LIBRERIA L, NUM-LIBRI NL

WHERE L.CodL= NL.CodL

AND NL.LibriVenduti = (SELECT MIN(NumLibriVenduti)

FROM NUM-LIBRI);