

---Seleziona tutti i viaggi svolti da 'Amedeo Barbieri'

```
select v.*
from AUTOMOBILISTA a
join VIAGGIO v on a.COD_F = v.AUTOMOBILISTA
where nome = 'Amedeo'
and cognome = 'Barbieri'
```

---Seleziona le tratte sulle quali non ha mai viaggiato un automobilista di sesso maschile ➤

```
Select * from TRATTA t
where not exists (
    select *
    from AUTOMOBILISTA a
    join VIAGGIO v on a.COD_F = v.AUTOMOBILISTA
    where sesso = 'M'
    and v.TRATTA=t.COD_T)
```

---Selezionare, per ogni compagnia autostradale, il codice, il nome e l'incasso complessivo ➤

```
select s.COD_S, s.NOME, sum(t.pedaggio) as somma_pedaggio
from SOCIETA s
join TRATTA t on s.COD_S = t.SOC_COMPETENZA
join VIAGGIO v on t.COD_T = v.TRATTA
join AUTOMOBILISTA a on v.AUTOMOBILISTA = a.COD_F
group by s.COD_S, s.NOME;
```

---Trova per ogni automobilista qual'è la società autostradale di cui ha percorso più tratte nei suoi viaggi ➤

```
select a1.COD_F, SOCIETA.COD_S, count(cod_t)
from automobilista a1
join viaggio v on v.AUTOMOBILISTA = a1.COD_F
join tratta t on t.COD_T=v.TRATTA
join societa s on s.COD_S=t.SOC_COMPETENZA
group by a1.COD_F, SOCIETA.COD_S
having count(cod_t) >= all (
    select count(cod_t)
    from automobilista a
    join viaggio v on v.AUTOMOBILISTA = a.COD_F
    join tratta t on t.COD_T=v.TRATTA
    join societa s on s.COD_S=t.SOC_COMPETENZA
    where a.COD_F=a1.COD_F
    group by a.COD_F, s.COD_S )
```

---Seleziona gli automobilisti che hanno percorso almeno una tratta di ogni società autostradale ➤

---(Suggerimento: si vogliono gli utenti che siano in relazione con tutte le società) ➤

```
select *
from AUTOMOBILISTA a
where not exists ( select *
                  from SOCIETA s
                  where not exists (
                      select *
                      from VIAGGIO v
                      join TRATTA t on v.TRATTA = t.COD_T
                      where t.SOC_COMPETENZA = s.COD_S
                      and a.COD_F = v.AUTOMOBILISTA))
```

---secondo esame:

---Inserire 'Italy' come valore di default di nazione in tabella Conferenza.

```
alter table Conferenza
add constraint def
default 'Italy' for nazione
```

---Selezionare gli autori che hanno scritto e inviato almeno un articolo ad una conferenza tenutasi in Italia

```
Select A.*, Conf.nazione
from Autore A join Scrive S on S.idautore=a.idautore
join Articolo Art on Art.idarticolo=S.idarticolo
join Conferenza Conf on Conf.idconferenza=Art.idconferenza
where Conf.nazione='Italy'
```

---Selezionare gli autori che hanno scritto almeno un articolo insieme a un coautore con diversa afferenza

```
Select distinct A1.*
from Autore A1
join Scrive S1 on S1.idautore=A1.idautore
join Scrive S2 on S1.idarticolo=S2.idarticolo
join Autore A2 on S2.idautore=A2.idautore
where A1.afferenza<>A2.afferenza
```

---Creare una vista AutoreStatistiche che riporti per ogni autore,  
---i suoi dati anagrafici,  
---il numero di articoli scritti e inviati a conferenze,  
---il numero di articoli accettati e il tasso di successo espresso in percentuale  
---(calcolato come il rapporto tra articoli accettati e articoli inviati).

```
Create view AutoreStatistiche as
select A.nome,a.cognome,a.idautore,count(Art.idarticolo) as
    NumeroArticoliScritti,
SUM(Art.accettato) as NumeroArticoliAccettati,
TassoSuccesso=(SUM(Art.accettato)*100/count(Art.idarticolo))
from Autore A join Scrive S on S.idautore=a.idautore
join Articolo Art on Art.idarticolo=S.idarticolo
group by A.nome,A.cognome,a.idautore
```

---Selezionare, per l'anno 2022, il titolo e la nazione della conferenza alla  
---quale sono stati inviati il minor numero di articoli.

```
select c.titolo, c.nazione,  
count(a.idarticolo) as num_articoli  
from Conferenza c  
left join Articolo a on c.idconferenza = a.idconferenza  
where year(c.datainizio) = 2022  
group by c.titolo, c.nazione  
having count(a.idarticolo)<=all(  
    select count(a1.idarticolo)  
    from Conferenza c1  
    left join Articolo a1 on c1.idconferenza = a1.idconferenza  
    where year(c1.datainizio) = 2022  
    group by c1.idconferenza)
```

---terzo esame

---Selezionare tutti i dati dei mutui richiesti da ROBERTA LOIACONO come  
 richiedente2,  
---riportare i dati sia delle richieste in corso che dei finanziamenti attivati

```
Select *  
from Mutuo M join Persona P on (M.richiedente2=P.id)  
where stato IN (0,1)  
and nome='ROBERTA LOIACONO'
```

---Per ogni mutuo finanziato, selezionare la somma totale pagata dal richiedente.  
---Riportare anche i dati del mutuo ovvero:  
---richiedente1, dataRichiesta, somma.

```
Select M.richiedente1, M.dataRichiesta,  
somma as somma_del_mutuo,  
sum(sommapagata) as somma_pagata_dal_cliente  
From Mutuo M  
join PagamentoRata P on (P.richiedente1=M.richiedente1 and  
    P.dataRichiesta=M.dataRichiesta)  
where stato=1  
group by M.richiedente1, M.dataRichiesta, somma
```

---Selezionare per ogni cliente,  
---il suo id e nome e il numero di mutui finanziati come secondo richiedente  
 (richiedente2),  
---riportando in ordine crescente dal cliente che non ha avuto un mutuo finanziato  
---a quello che ne ha avuti di più.

```
SELECT p.nome,  
COUNT(m.dataRichiesta) as numero_mutui_finanziati
```

```
FROM Persona p
LEFT JOIN Mutuo m on m.richiedente2 = p.id
WHERE p.tipo = 'richiedente'
and (stato=1 or stato is null)
GROUP BY p.id,p.nome
ORDER BY 2
```

---Selezionare i dati dei clienti richiedenti che non hanno mai ottenuto un finanziamento di un mutuo (come richiedente1): possono essere sia richiedenti che non hanno mai fatto alcuna richiesta di mutuo oppure richiedenti che hanno fatto richieste, ma non ne hanno  
---mai avuta una il cui finanziamento sia stato attivato

```
select *
from persona p1
where p1.tipo='richiedente'
and p1.id not in (
    select richiedente1
    from mutuo m
    where m.stato = 1
)
```

---Selezionare i mutui il cui finanziamento è stato attivato e  
---la cui data di inizio mutuo è nel 2019 e visualizzare il totale  
---fino ad oggi versato e la data dell'ultimo pagamento.

```
select M.richiedente1, M.dataRichiesta,
sum(P.sommapagata) as totale_pagato,
max(P.dataPagamento) as ultima_Data_Pagamento
from Mutuo M
left join PagamentoRata P on ( P.richiedente1 = M.richiedente1 and
    M.dataRichiesta=P.dataRichiesta)
where M.stato = 1
and year(M.dataInizioMutuo) = 2019
group by M.richiedente1, M.dataRichiesta
```

---Inserire un controllo in tabella ISCRITTO che garantisca che una competizione  
---non possa avere più artisti classificati nella stessa posizione.

```
ALTER TABLE ISCRITTO
ADD CONSTRAINT ck_I UNIQUE ([NOMECOMP],[ANNOEDIZIONE],[POSIZIONE])
```

---Inserire un controllo che l'attributo ANNOEDIZIONE in tabella COMPETIZIONE sia  
>= 1980.

```
ALTER TABLE [COMPETIZIONE]
ADD CONSTRAINT check_anno CHECK ([ANNOEDIZIONE] >=1980)
```

---Selezionare i dati del vincitore (primo classificato) di ogni COMPETIZIONE  
svoltasi a Milano.

```
SELECT *
```

```
FROM COMPETIZIONE F JOIN ISCRITTO P ON (P.NOMECOMP=F.NOMECOMP
AND P.ANNOEDIZIONE=F.ANNOEDIZIONE) JOIN INTERPRETE I ON (I.NOME=P.INTERPRETE)
WHERE F.LUOGO='Milano'
AND POSIZIONE=1
```

---Selezionare le competizioni in cui non si è iscritto nessun artista singolo.

```
SELECT *
FROM COMPETIZIONE C
WHERE NOT EXISTS( SELECT *
FROM ISCRITTO P
JOIN INTERPRETE I ON (I.NOME=P.INTERPRETE)
WHERE P.NOMECOMP=C.NOMECOMP
AND P.ANNOEDIZIONE=C.ANNOEDIZIONE
AND I.GRUPPO='no')
```

--oppure

---poteva essere interpretata anche come una divisione:

---Selezionare le competizioni in cui tutte le iscrizioni

--sono state effettuate da gruppi

```
select c.*
from COMPETIZIONE c
where not exists (select *
                  from ISCRITTO isc
                  where isc.NOMECOMP = c.NOMECOMP
                  and isc.ANNOEDIZIONE=c.ANNOEDIZIONE
                  and not exists ( select *
                                from INTERPRETE i
                                where i.GRUPPO = 'si'
                                and i.NOME = isc.INTERPRETE))
```

---Selezionare nome e data di nascita degli interpreti che hanno partecipato a tutte le edizioni di WIND MUSIC ➤

--riscritta con doppia negazione diventa: selezionare nome e data di nascita degli interpreti per cui ➤

-- non esiste una edizione del festival di sanremo a cui non abbiano partecipato

```
SELECT nome, data_nascita
```

```
FROM INTERPRETE I
WHERE NOT EXISTS (
    SELECT *
    FROM COMPETIZIONE F
    WHERE NOMECOMP='WIND MUSIC'
    AND NOT EXISTS (
        SELECT *
        FROM ISCRITTO P
        WHERE I.NOME=P.INTERPRETE
        AND P.NOMECOMP=F.NOMECOMP
        AND P.ANNOEDIZIONE=F.ANNOEDIZIONE
        AND POSIZIONE IS NOT NULL --[questa condizione ➤
        poteva essere omessa]))
```

