

### --Query 1

#### Richiesta:

Contare il numero di lingue in cui le release contenute nel database sono scritte (il risultato deve contenere soltanto il numero delle lingue, rinominato "Numero\_Lingue").

#### Commento:

Nella prima query si può constatare che il risultato restituito dalla interrogazione è corretto semplicemente contando il numero di lingue e confrontandolo con quelle realmente presenti, come chiave abbiamo una unica chiave esterna nell'entità RELEASE che corrisponde a language.

```
select count(distinct language) as Numero_Lingue
from release
```

---

### -- Query 2

#### Richiesta:

Elencare gli artisti che hanno cantato canzoni in italiano (il risultato deve contenere il nome dell'artista e il nome della lingua).

#### Commento:

La seconda interrogazione ci permette di ricavare quali artisti hanno cantato una canzone in lingua italiana e le entità usate sono artista e lingua.

Le entità Artist e Language hanno come primary key ID e Artist ha come foreign key gli attributi type, gender e area.

```
select artist.name as Nome_artista, language.name as Lingua_canzone
from artist, language
where language.name = 'Italiano'
```

---

### -- Query 3

#### Richiesta:

Elencare le release di cui non si conosce la lingua (il risultato deve contenere soltanto il nome della release).

#### Commento:

La risoluzione della seguente query richiedeva che la lingua della canzone fosse sconosciuta quindi si è potuto confrontare il risultato dell'interrogazione con i dati della tabella e assicurarsi che entrambe avessero nel campo lingua un valore NULL. le chiavi sono le stesse della query 1\*/

```
select name
from release
where language is NULL
```

---

#### --Query 4

##### **Richiesta:**

Elencare gli artisti il cui nome contiene tutte le vocali ed è composto da una sola parola (il risultato deve contenere soltanto il nome dell'artista).

##### **Commento:**

Per questa richiesta, abbiamo selezionato il nome degli artisti dalla tabella Artist ,quindi, tramite il comando like abbiamo constatato che effettivamente nel nome di un artista fossero presenti tutte le vocali richieste.

```
select artist.name
from artist
where artist.name like '%a%'
      and artist.name like '%e%'
      and artist.name like '%i%'
      and artist.name like '%o%'
      and artist.name like '%u%'
      and artist.name not like '% %'
```

---

#### --Query 5

##### **Richiesta:**

Elencare tutti gli pseudonimi di Prince con il loro tipo, se disponibile (il risultato deve contenere lo pseudonimo dell'artista, il nome dell'artista (cioè Prince) e il tipo di pseudonimo (se disponibile)).

##### **Commento:**

Per la risoluzione della seguente query è risultato sufficiente assicurarsi che ogni tupla presente nel risultato contenesse nel nome la parola "Prince".

```
select sort_name,name,type
from artist
where name like '%Prince%'
```

---

#### --Query 6

##### **Richiesta :**

Elencare le release di gruppi inglesi ancora in attività (il risultato deve contenere il nome del gruppo e il nome della release e essere ordinato per nome del gruppo e nome della release)

##### **Commento:**

Nella seguente query abbiamo utilizzato le entità Artist, release e area per ricavare i gruppi Inglesi in attività verificando che l'area fosse uguale a "United Kingdom" e che il valore dell'attributo artist.ended è diverso da true.

```
select distinct artist.name, release.name
from artist,release,area join area_type on (area.type = area_type.id)
where area_type.id = '1' and area.name = 'United Kingdom'
      and artist.ended <> 'true' and release.artist_credit = artist.id
order by artist.name ASC
```

---

#### --Query 7

##### **Richiesta:**

Trovare le release in cui il nome dell'artista è diverso dal nome accreditato nella release (il risultato deve contenere il nome della release, il nome dell'artista accreditato (cioè artist\_credit.name) e il nome dell'artista(cioè artist.name)).

##### **Commento:**

Nella risoluzione della settima query viene stampato il nome accreditato dell'artista il quale però deve avere un nome diverso dal proprio. per farlo abbiamo innanzitutto eseguito un join tra le relazioni interessate e successivamente abbiamo imposto come vincolo appunto che artist\_credit.name <> artist.name.

```
select distinct artist_credit.name,artist.name,release.name
from ((artist_credit join artist_credit_name
      on artist_credit_name.artist_credit = artist_credit.id)
join artist on artist_credit_name.artist = artist.id)
join release on(release.artist_credit = artist_credit.id)
where artist_credit.name <> artist.name
```

---

#### --Query 8

##### **Richiesta:**

Trovare gli artisti con meno di tre release (il risultato deve contenere il nome dell'artista ed il numero di release).

##### **Commento:**

Qui abbiamo eseguiti vari join tra le rispettive relazioni, eseguito un raggruppamento delle release degli artisti per nome dato che ogni artista doveva aver pubblicato almeno 3 release ed infine viene imposto come vincolo quello di avere 3 release sotto il proprio nome.

```
select artist.name,count(release.name)
from ((artist_credit join artist_credit_name
      on artist_credit_name.artist_credit = artist_credit.id)
join artist on artist_credit_name.artist = artist.id)
```

```
--Query 9
```

--Version A

[illegible]

```
and R2.length>recording.length
and R2.artist_credit = artist_credit.id
and artist.id=artist_credit.id
and artist.gender=gender.id
and gender.name='Female'
and R2.length is not null)
```

---

#### --Query 10

##### Richiesta:

Elencare le lingue cui non corrisponde nessuna release (il risultato deve contenere il nome della lingua, il numero di release in quella lingua, cioè 0, e essere ordinato per lingua) (scrivere due versioni della query).

#### --Versione A

##### Commento:

Per la versione A prima ci ricaviamo una tabella con tutte le lingue e contiamo le rispettive release. a questa sottraiamo (con il comando except) tutte le lingue che hanno una release.

```
select distinct language.name,count (distinct release.id)
from language left join release on (release.language=language.id)
group by language.name
except
select language.name,count (distinct release.id)
from language,release
where language.id=release.language
group by language.name
```

#### -- Versione B

##### Commento:

Nella seconda versione selezioniamo il nome della lingua e contiamo gli id delle release facendo un left join tra language e release verificando che l'id della release sia diverso da null, infine li raggruppiamo per nome della lingua.

```
select distinct language.name,count (distinct release.id)
from language left join release on (release.language = language.id)
where release.id is null
group by language.name
```

---

## --Query 11

### Richiesta:

Ricavare la seconda registrazione per lunghezza di un artista uomo (il risultato deve comprendere l'artista accreditato, il nome della registrazione e la sua lunghezza) (scrivere due versioni della query).

### -Versione A

#### Commento:

Per la versione A sono state usate tre query annidate, con la prima otteniamo tutti i nomi degli artisti e le registrazioni con i rispettivi nomi e lunghezze, di questa con il comando not con la seconda query prendiamo solo la seconda registrazione più lunga togliendo la registrazione più lunga con la terza query.

```
select distinct recording.name, recording.length/60000 as MaxLenght, artist.name
from recording join artist_credit on recording.artist_credit = artist_credit.id
      join artist_credit_name on artist_credit_name.artist_credit = artist_credit.id
      join artist on artist_credit_name.artist = artist.id
where recording.length in (
      select max(recording.length)
      from recording join artist_credit on recording.artist_credit = artist_credit.id
            join artist_credit_name on artist_credit_name.artist_credit = artist_credit.id
            join artist on artist_credit_name.artist = artist.id
      where artist.gender = 1 and recording.length not in
            (select max(distinct recording.length) as MaxDurata
            from recording join artist_credit on recording.artist_credit = artist_credit.id
                  join artist_credit_name
                  on artist_credit_name.artist_credit = artist_credit.id
                  join artist on artist_credit_name.artist = artist.id
            where artist.gender = 1 ) )
```

### --Versione B

#### Commento :

Anche qui utilizziamo lo stesso ragionamento della versione precedente, con la differenza che troviamo la seconda registrazione più lunga, verificando che la lunghezza della registrazione della terza query (che è quella più lunga in assoluto) sia maggiore di tutte le lunghezze di registrazione della seconda (ottenendo quindi la seconda più lunga), a questo punto non ci rimane che estrarre con la prima select nome dell'artista, nome e lunghezza della registrazione con lunghezza registrazione uguale al risultato della seconda sottoquery.

```
select distinct recording.name, recording.length/60000 as MaxLenght, artist.name
from recording join artist_credit on recording.artist_credit = artist_credit.id
      join artist_credit_name on artist_credit_name.artist_credit = artist_credit.id
      join artist on artist_credit_name.artist = artist.id
where recording.length in (
      select max(recording.length)
```

```

from recording join artist_credit on recording.artist_credit = artist_credit.id
    join artist_credit_name on artist_credit_name.artist_credit = artist_credit.id
    join artist on artist_credit_name.artist = artist.id
where artist.gender = 1
and recording.length < (
    select max(distinct recording.length) as MaxDurata
    from recording join artist_credit on recording.artist_credit = artist_credit.id
        join artist_credit_name on artist_credit_name.artist_credit = artist_credit.id
        join artist on artist_credit_name.artist = artist.id
    where artist.gender = 1 ) )

```

---

## --Query 12

### Richiesta:

Per ogni stato esistente riportare la lunghezza totale delle registrazioni di artisti di quello stato (il risultato deve comprendere il nome dello stato e la lunghezza totale in minuti delle registrazioni (0 se lo stato non ha registrazioni) (scrivere due versioni della query).

### --Versione A

#### Commento:

Per la prima versione della query abbiamo innanzitutto ricavato l'ID di tutti gli artisti che hanno rilasciato una release. poi con la query esterna eseguo dei join tra le opportune tabelle con il vincolo area\_type.name = 'Country' ed estraggo quindi solo gli id degli artisti presenti nel risultato della seconda sottoquery.

```

select area.name, coalesce(sum(recording.length)/60000,0)
from area left join artist on artist.area = area.id
    left join artist_credit_name on artist_credit_name.artist=artist.id
    left join artist_credit on artist_credit.id = artist_credit_name.artist_credit
    left join recording on recording.artist_credit=artist_credit.id
    left join area_type on area_type.id=area.type
where area_type.name = 'Country' and artist.id in(
    select artist.id
    from artist left join artist_credit_name on artist.id = artist_credit_name.artist
        join artist_credit on artist_credit.id = artist_credit_name.artist_credit
        join recording on recording.artist_credit = artist_credit.id
    group by artist.id)
group by area.name

```

### --Versione B

**Commento:**

Nella seconda versione eseguiamo dei join tra le opportune tabelle verificando che il tipo dell'area sia uguale a "stato" e infine facciamo la group by dei nomi dell'area.

```
select area.name nome_nazione,coalesce (sum(recording.length)/60000, 0)
somma_registrazioni
from ((artist_credit join recording on recording.artist_credit = artist_credit.id)
      join artist on artist_credit.id = artist.id)
      join area on artist.area = area.id
      join area_type on area_type.id=area.type
where area_type.name = 'Country'
group by area.name
```

---

**--Query 13****Richiesta:**

Ricavare gli artisti britannici che hanno pubblicato almeno 10 release (il risultato deve contenere il nome dell'artista, il nome dello stato (cioè United Kingdom) e il numero di release) (scrivere due versioni della query).

**--Versione A****Commento:**

La prima versione esegue una select su due join fra le query contenenti le informazioni a noi necessarie ed in seguito filtra il risultato applicando le condizioni specificate dalla consegna.

```
select artist_credit.name,area.name,count(release.id)
from release join artist_credit on release.artist_credit = artist_credit.id
      join artist on artist_credit.id = artist.id
      join area on artist.area = area.id
where area.name = 'United Kingdom'
group by artist_credit.name,area.name
having count(release.id) >= 10
```

**--Versione B****Commento:**

La prima select ossia quella esterna seleziona gli artisti con più di 10 release, di questi prende solo coloro (quindi gli artisti) che compaiono nella lista degli artisti inglesi data dalla seconda query.

```
select artist.name,count(release.id),area.name
from release join artist_credit on artist_credit.id = release.artist_credit
      join artist_credit_name on artist_credit_name.artist_credit = artist_credit.id
      join artist on artist.id=artist_credit_name.artist
      join area on artist.area=area.id
```



```

group by artist.name,area.name
having count(release.id)>=10 and artist.name in(
    select artist.name
    from release join artist_credit on artist_credit.id = release.artist_credit
    join artist_credit_name on artist_credit_name.artist_credit = artist_credit.id
    join artist on artist.id=artist_credit_name.artist
    join area on artist.area=area.id
    where area.name = 'United Kingdom' and artist.id = release.artist_credit
    group by artist.name,area.name)

```

---

#### --Query 14

##### Richiesta:

Considerando il numero medio di tracce tra le release pubblicate su CD, ricavare gli artisti che hanno pubblicato esclusivamente release con più tracce della media (il risultato deve contenere il nome dell'artista e il numero di release ed essere ordinato per numero di release discendente) (scrivere due versioni della query).

#### --Versione A

##### Commento:

La query interna trova tutti gli artisti con un numero di tracce pubblicate su CD (format =1) inferiore alla media. Quella esterna elimina da tutte le release degli artisti quelle tuple della seconda che sono appunto inferiori alla media, ottenendo quindi solo coloro che hanno pubblicato esclusivamente release con più tracce della media

```

select count(release.id),artist_credit.name
from release join artist_credit on release.artist_credit = artist_credit.id
join medium on medium.release = release.id
where medium.format = 1 and artist_credit.name not in
    (--Gli artisti che hanno fatto release con numero track inferiore a media
    select artist_credit.name
    from release join artist_credit on release.artist_credit = artist_credit.id
    join medium on medium.release = release.id
    where medium.track_count <= (
        select avg(medium.track_count)
        from medium
        where medium.format=1) )
group by artist_credit.name
order by count(release.id) desc

```

#### --Versione B

**Commento:**

Quattro sottoquery.

La prima select più interna ricava la media delle tracce delle release su cd.

La 2 ricava gli artisti che hanno pubblicato release con numero tracce minore o uguale alla media.

La 3 esclude da tutti gli artisti, quelli che hanno rilasciato un numero di release inferiori alla media.

La 4 esegue una proiezione del nome degli artisti che si trovano nella sottoquery ossia hanno un numero di release superiore alla media.

```
select count(release.id),artist_credit.name
from release join artist_credit on release.artist_credit = artist_credit.id
      join medium on medium.release = release.id
where medium.format = 1 and artist_credit.name in
      (--Gli artisti che hanno fatto release con numero track maggiore a media
      select artist_credit.name
      from release join artist_credit on release.artist_credit = artist_credit.id
            join medium on medium.release = release.id
      except
      select artist_credit.name
      from release join artist_credit on release.artist_credit = artist_credit.id
            join medium on medium.release = release.id
      where medium.track_count <=(
            select avg(medium.track_count)
            from medium
            where medium.format=1)) )
group by artist_credit.name
order by count(release.id) desc
```

---

**--Query 15****Richiesta:**

Ricavare il primo artista morto dopo Louis Armstrong (il risultato deve contenere il nome dell'artista, la sua data di nascita e la sua data di morte) (scrivere due versioni della query).

**--Versione A****Commento:**

Nella prima versione abbiamo 3 sottoquery dove abbiamo :

Prima select: Cerchiamo la data di nascita e di morte dell'artista tale che sia nulla.

Seconda select: Troviamo la data di nascita immediatamente successiva a quella della terza sottoquery.

Terza Sottoquery: Troviamo la data di morte di Louis Armstrong.

Quindi combinando i risultati delle sottoquery abbiamo il primo artista morto dopo Louis Armstrong.

```

select artist.name nome_artista,
make_date(artist.begin_date_year,artist.begin_date_month,artist.begin_date_day) as
data_nascita,
make_date(artist.end_date_year,artist.end_date_month,artist.end_date_day) as data_morte
from artist
where artist.begin_date_year is not null
      and artist.begin_date_month is not null
      and artist.begin_date_day is not null
      and artist.end_date_year is not null
      and artist.end_date_month is not null
      and artist.end_date_day is not null
and make_date(artist.end_date_year,artist.end_date_month,artist.end_date_day) in
(select
  min(make_date(artist.end_date_year,artist.end_date_month,artist.end_date_day))
from artist
where artist.begin_date_year is not null
      and artist.begin_date_month is not null
      and artist.begin_date_day is not null
      and artist.end_date_year is not null
      and artist.end_date_month is not null
      and artist.end_date_day is not null
      and
make_date(artist.end_date_year,artist.end_date_month,artist.end_date_day) >
      (select
make_date(artist.end_date_year,artist.end_date_month,artist.end_date_day)
from artist
where artist.name = 'Louis Armstrong'))

```

--Versione B

**Commento:**

Nella seconda versione troviamo tutte le date di nascita e di morte dell'artista tale che siano nulle e intersechiamo il risultato con la tabella data da:

Prima select: Cerchiamo la data di nascita e di morte dell'artista tale che sia nulla.

Seconda Select: Troviamo la data di morte immediatamente successiva a quella della terza sottoquery.

Terza Sottoquery: Troviamo a data di morte di Louis Armstrong.

```

select artist.name nome_artista,
make_date(artist.begin_date_year,artist.begin_date_month,artist.begin_date_day)
data_nascita,
make_date(artist.end_date_year,artist.end_date_month,artist.end_date_day) data_morte

```

```

from artist
where artist.begin_date_year is not null
        and artist.begin_date_month is not null
        and artist.begin_date_day is not null
        and artist.end_date_year is not null
        and artist.end_date_month is not null
        and artist.end_date_day is not null

intersect

select artist.name nome_artista,
make_date(artist.begin_date_year,artist.begin_date_month,artist.begin_date_day) as
data_nascita,
make_date(artist.end_date_year,artist.end_date_month,artist.end_date_day) as data_morte
from artist
where artist.begin_date_year is not null
        and artist.begin_date_month is not null
        and artist.begin_date_day is not null
        and artist.end_date_year is not null
        and artist.end_date_month is not null
        and artist.end_date_day is not null
and make_date(artist.end_date_year,artist.end_date_month,artist.end_date_day) =
(select
min(make_date(artist.end_date_year,artist.end_date_month,artist.end_date_day))
from artist
where artist.begin_date_year is not null
        and artist.begin_date_month is not null
        and artist.begin_date_day is not null
        and artist.end_date_year is not null
        and artist.end_date_month is not null
        and artist.end_date_day is not null
and make_date(artist.end_date_year,artist.end_date_month,artist.end_date_day) >
(
select make_date(artist.end_date_year,artist.end_date_month,artist.end_date_day)
from artist
where artist.name = 'Louis Armstrong'))

```

---

### --Query 16

**Richiesta:**

**Elencare le coppie di etichette discografiche che non hanno mai fatto uscire una release in comune ma hanno fatto uscire una release in collaborazione con una medesima terza etichetta (il risultato deve comprendere i nomi delle coppie di etichette discografiche) (scrivere due versioni della query).**

--Versione A

**Commento:**

Nella prima select effettuiamo dei join tra release\_label rinominata e label rinominata.

Otteniamo la tabella delle release e delle case discografiche distinte che le hanno prodotte collaborando, da qui escludiamo i risultati delle sottoquery successive.

Nella seconda select selezioniamo tutte le release distinte che si trovano nella terza sottoquery.

Nella terza sottoquery facciamo il join tra la tabella delle label che hanno lavorato alla stessa release e il risultato della quarta sottoquery

Nella quarta troviamo le label diverse che hanno prodotto la medesima release.

```
select distinct l1.name as name_label1, l2.name as name_label2
from release_label r1
    join release_label r2 on(r1.label <> r2.label)
    join label l1 on(l1.id=r1.label)
    join label l2 on (l2.id=r2.label)
where not exists (
    select *
    from release
    where release.id = r1.release
        and r2.release =release.id)
        and (r1.label,r2.label) in (
            select distinct r1.label as label1,x.label1
            from release_label r1
                join release_label r2
                on(r1.label <>r2.label and r1.release = r2.release)
                join(
                    select distinct r1.label as label1,r2.label as label2
                    from release_label r1
                        join release_label r2 on(r1.label <>r2.label and
                            r1.release = r2.release)
                )x on(x.label1<>r1.label and x.label2=r2.label)
            order by r1.label)
and r1.label<r2.label
```

--Versione B

**Commento:**

Nella tabella label\_collaboration, creata con la WITH, vengono inserite tutte le coppie di label che hanno lavorato all'uscita di una stessa release

Nella query principale, si prendono tutte le coppie di label che non appartengono a quelle estratte da label\_collaboration, quindi tutte le coppie di label che non hanno rilasciato una release in comune.

Con la query interna, si controlla se esiste una terza label che ha collaborato con entrambe.

```
with label_collaboration as(
    select distinct r1.label as label1, r2.label as label2
    from release_label r1, release_label r2
    where r1.release = r2.release and r1.label <> r2.label
)

select distinct l1.name as label1, l2.name as label2
from label l1, label l2, release_label r1, release_label r2
where (l1.id, l2.id) not in (select * from label_collaboration)
    and l1.name <> l2.name
    and r1.label = l1.id
    and r2.label = l2.id
    and exists(
        select distinct l3.name
        from label l3 join release_label r3 on l3.id = r3.label
        where r3.release = r1.release
            and l1.name <> l3.name
            and l3.name in(
                select distinct l.name
                from label l join release_label r on l.id = r.label
                where r.release = r2.release
                    and l.name <> l2.name))
```