QUERY:

1-select count(distinct language.name) as Numero\_Lingue

from release join language

on release.language=language.id

QUERY 2

**select distinct artist.name,**

**language.name**

**from release join language**

**on release.language=language.id**

**join artist\_credit**

**on artist\_credit.id=release.artist\_credit**

**join artist\_credit\_name**

**on artist\_credit.id=artist\_credit\_name.artist\_credit**

**join artist**

**on artist\_credit\_name.artist=artist.id**

**where language.name='English'**

**1)** Nel risultato vengono richiesti gli artisti e la lingua (attributo name della tabella artist e attributo name della tabella language)della canzone che cantano. (si è scelto Inglese perché mancava Italiano).

**2)** Per questa query si è partiti con un join tra release e language sull’attributo language della tabella release e sull’attributo id della tabella language.Il join è tra la chiave esterna Language della tabella release e la chiave primaria id della tabella Language. Viene poi fatto un altro join è tra l’attributo id della tabella artist\_credit e l’attributo artist\_credit nella tabella release. Un altro join è fatto tra la chiave esterna Artist\_credit di release e la chiave primaria id nella tabella artist\_credit. Il prossimo join è tra l’attributo id della tabella artist\_credit e l’attributo artist\_credit in artist\_credit\_name.Il join è quindi fatto sulle due chiavi primarie delle due tabelle. L’ ultimo è tra l’attributo id della tabella artist e l’attributo artist in artist\_credit\_name.Il join è fatto sulle due chiavi primarie delle due tabelle.

**3)** Nella query non sono presenti sotto problemi.

**4)** Per la gestione dei duplicati si è utilizzato il distinct, i valori nulli non vengono considerati perché viene scelta solo la lingua Inglese. Sull’attributo Language di Release sono presenti valori nulli ma non interessano al risultato.

**5)** Non sono richieste due versioni della query.

**6)**

**select distinct artist.name,language.name**

**from release join language**

**on release.language=language.id**

**join artist\_credit**

**on artist\_credit.id=release.artist\_credit**

**join artist\_credit\_name**

**on artist\_credit.id=artist\_credit\_name.artist\_credit**

**join artist**

**on artist\_credit\_name.artist=artist.id**

**where language.name not in**

**(**

**select distinct language.name**

**from release join language**

**on release.language=language.id**

**join artist\_credit**

**on artist\_credit.id=release.artist\_credit**

**join artist\_credit\_name**

**on artist\_credit.id=artist\_credit\_name.artist\_credit**

**join artist**

**on artist\_credit\_name.artist=artist.id**

**where language.name='English'**

**)**

Con la query di verifica prendo tutti gli artisti che non hanno cantato canzoni in Inglese e li cerco dove ci sono tutti quelli che hanno invece cantato canzoni in Inglese e non trovo risultati di artisti che hanno cantatom canzoni in inglese quindi significa che la query è corretta.

3- select release.name

from release

where release.language is null

QUERY 4

**select release.name**

**from release**

**where release.name like '%Love%'**

**1)** Nella query viene richiesto solo il nome della release (attributo name della tabella release) che contiene la parola Love.

**2)** Non si sono usati join per questa query.

**3)** Non esistono sotto problemi per questa query.

**4)** I valori nulli non sono presenti sull’attributo name di release.

**5)** Non sono richieste due versioni della query.

**6)**

**select release.name**

**from release**

**where release.name not in**

**(**

**select release.name**

**from release**

**where release.name like '%Love%'**

**)**

La query permette di verificare il risultato perché sto cercando le release che non hanno la parola Love e, cercandole dove effettivamente c’è la parola Love, non trovo risultati che contengano la parola Love. Quindi significa che la query è correta.

5- select artist\_alias.name, artist.name, artist\_alias\_type.name

from artist\_alias left join artist

on artist.id=artist\_alias.artist

join artist\_alias\_type

on artist\_alias.type=artist\_alias\_type.id

where artist.name='Prince'

QUERY 6

**select distinct artist.name,release.name**

**from artist join artist\_type**

**on artist.type=artist\_type.id**

**join artist\_credit\_name**

**on artist\_credit\_name.artist=artist.id**

**join artist\_credit**

**on artist\_credit.id=artist\_credit\_name.artist\_credit**

**join release**

**on artist\_credit.id=release.artist\_credit**

**join area**

**on artist.area=area.id**

**where artist\_type.name='Group'**

**and artist.ended=false**

**and area.name='United Kingdom'**

**order by artist.name, release.name**

**1)** Nella query è richiesto il nome del gruppo (attributo name della tabella artist) e il nome della release (atributo name della tabella release) di gruppi inglesi ancora in attività. Il tutto deve essere ordinato per nome della release e nome dell’artista.

**2)** Viene fatto un join tra la chiave esterna type della tabella artist e la chiave primaria id della tabella artist\_type. Il prossimo join è tra la chiave primaria id della tabella artist e la chiave esterna artist della tabella artist\_credit\_name. Il successivo è fatto sulle chiavi primarie delle tabelle artist\_credit e artist\_credit\_name. Poi c’è il join tra la chiave primaria di artist\_credit e la chiave esterna della tabella release artist\_credit. L’ultimo join è tra la chiave primaria della tabella area e la chiave esterna area della tabella artist.

**3)** Non sono presenti sotto problemi.

**4)** Per eliminare i duplicati si è usato il distinct, i valori nulli non si sono considerati in quanto non presenti sugli attributi.

**5)** Non sono richieste due versioni della query.

**6)**

**select distinct artist.name,release.name**

**from artist join artist\_type**

**on artist.type=artist\_type.id**

**join artist\_credit\_name**

**on artist\_credit\_name.artist=artist.id**

**join artist\_credit**

**on artist\_credit.id=artist\_credit\_name.artist\_credit**

**join release**

**on artist\_credit.id=release.artist\_credit**

**join area**

**on artist.area=area.id**

**where artist\_type.name='Group'**

**and area.name='United Kingdom'**

**and artist.name not in**

**(**

**select distinct artist.name**

**from artist join artist\_type**

**on artist.type=artist\_type.id**

**join artist\_credit\_name**

**on artist\_credit\_name.artist=artist.id**

**join artist\_credit**

**on artist\_credit.id=artist\_credit\_name.artist\_credit**

**join release**

**on artist\_credit.id=release.artist\_credit**

**join area**

**on artist.area=area.id**

**where artist\_type.name='Group'**

**and artist.ended=true**

**and area.name='United Kingdom'**

**order by artist.name**

**)**

La query permette di verificare il risultato perché, cercando i gruppi nell’insieme di quelli che hanno già concluso la loro attività, trovo tutti quelli che non l’hanno ancora conclusa. Posso così dire che la query è giusta.

7- select distinct release.name,

artist.name,

artist\_credit.name

from artist join artist\_credit\_name

on artist.id=artist\_credit\_name.artist

join artist\_credit

on artist\_credit.id=artist\_credit\_name.artist\_credit

join release

on release.artist\_credit=artist\_credit.id

where artist\_credit.name!=artist.name

QUERY 8

**select artist.name, count(release.name)**

**from artist join artist\_credit\_name**

**on artist.id=artist\_credit\_name.artist**

**join artist\_credit**

**on artist\_credit.id=artist\_credit\_name.artist\_credit**

**join release**

**on artist\_credit.id=release.artist\_credit**

**group by artist.name**

**having count(release.name)<3**

**1)** Nella query viene richiesto il nome dell’artista (attributo name della tabella artist) e il numero delle sue release.

**2)** Viene fatto il join tra la chiave primaria id della tabella artist con la chiave esterna artist della tabella artist\_credit\_name. L’altro join è tra le chiavi primarie di artist\_credit e artist\_credit\_name. L’ultimo join è tra la chiave primaria di artist\_credit e la chiave esterna artist\_credit della tabella release. Non sono presenti valori nulli sugli attributi su cui si è fatto il join.

**3)** Non esistono sotto problemi.

**4)** Non sono presenti duplicati quindi non si è usato il distinct, i valori nulli non sono stati presi in considerazione.

**5)** Non sono richieste due versioni della query.

**6)**

**select artist.name, count(release.name)**

**from artist join artist\_credit\_name**

**on artist.id=artist\_credit\_name.artist**

**join artist\_credit**

**on artist\_credit.id=artist\_credit\_name.artist\_credit**

**join release**

**on artist\_credit.id=release.artist\_credit**

**where artist.name not in**

**(**

**select artist.name**

**from artist join artist\_credit\_name**

**on artist.id=artist\_credit\_name.artist**

**join artist\_credit**

**on artist\_credit.id=artist\_credit\_name.artist\_credit**

**join release**

**on artist\_credit.id=release.artist\_credit**

**group by artist.name**

**having count(release.name)<3**

**)**

**group by artist.name**

Con questa query di controllo cerco tutti gli artisti con un numero di release superiori a 3 e non ne trovo nessuno che ne abbia un numero inferiore a 3. Si deduce quindi che la query sia giusta.

9a- select artist.name,

recording.name,

recording.length/60000

from artist join gender

on artist.gender=gender.id

join artist\_credit\_name

on artist\_credit\_name.artist=artist.id

join artist\_credit

on artist\_credit\_name.artist\_credit=artist\_credit.id

join recording

on recording.artist\_credit=artist\_credit.id

where gender.name='Female'

and recording.length=

(

select max(recording.length)

from artist join gender

on artist.gender=gender.id

join artist\_credit\_name

on artist\_credit\_name.artist=artist.id

join artist\_credit

on artist\_credit\_name.artist\_credit=artist\_credit.id

join recording

on recording.artist\_credit=artist\_credit.id

where gender.name='Female'

)

9b- select \* from

(

select artist.name,recording.name,recording.length/60000 as length

from artist join gender

on artist.gender=gender.id

join artist\_credit\_name

on artist\_credit\_name.artist=artist.id

join artist\_credit

on artist\_credit\_name.artist\_credit=artist\_credit.id

join recording

on recording.artist\_credit=artist\_credit.id

where gender.name='Female'

and recording.length is not null

)query

where query.length>= all

(

select recording.length/60000 as length

from artist join gender

on artist.gender=gender.id

join artist\_credit\_name

on artist\_credit\_name.artist=artist.id

join artist\_credit

on artist\_credit\_name.artist\_credit=artist\_credit.id

join recording

on recording.artist\_credit=artist\_credit.id

where gender.name='Female'

and recording.length is not null

)

QUERY 10 ( versione 1)

**select language.name, count(release.name)**

**from language left join release**

**on language.id=release.language**

**group by language.name**

**having count(release.name)=0**

**order by language.name**

**1)** Il risultato chiede il nome della lingua (attributo name della tabella Language) e il numero delle release.

**2)** Viene fatto un left join tra la chiave primaria della tabella Language e la chiave esterna della tabella release. Questo viene fatto per considerare i valori nulli che ha l’attributo language della tabella release. Quindi, grazie a ciò, mette count = 0 alle lingue cui non corrisponde nessuna release.

**3)** Non sono presenti sotto problemi.

**4)** Non è necessario usare distinct per i duplicati e per quanto riguarga i valori nulli, ho già spiegato al punto precedente.

**5)** E’ richiesta una seconda versione della query.

**6)**

**select language.name,count(release.name)**

**from language join release**

**on language.id=release.language**

**group by language.name**

**having count(release.name) !=0**

**order by language.name**

Questa query cerca tutte le lingue cui corrisponde almeno una release e non compaiono le lingue cui non corrisponde nessuna release. Si conclude che la query richiesta è giusta.

QUERY 10 (versione 2)

**select language.name,**

**0 as count**

**from language**

**where language.name not in**

**(**

**select distinct language.name**

**from language join release**

**on language.id = release.language**

**)**

**group by language.name**

**order by language.name**

**1)** Il risultato chiede il nome della lingua (attributo name della tabella Language) e il numero delle release.

**2)** Il join viene fatto nel sotto problema tra la chiave primaria della tabella language e la chiave esterna language della tabella release.

**3)** E’ presente un sotto problema in cui vengono considerate tutte le lingue cui corrisponde almeno una release. Si cercano poi qui tutte le lingue che non sono contenute in questo insieme e si mette a queste il count=0, perché sicuramente queste lingue non avranno release.

**4)** Duplicati e valori nulli non vengono considerati in questa query.

**5)** Si è già fatta una prima versione della query.

**6)**

Per controllare la query ho già i risultati della prima versione, che cincidono, la verifica è già stata fatta nella prima versione quindi la query è giusta.

11a- select recording.name,recording.length

from artist join artist\_credit\_nameon artist\_credit\_name.artist=artist.id

join artist\_crediton artist\_credit\_name.artist\_credit=artist\_credit.id

join recordingon artist\_credit.id=recording.artist\_credit

join genderon artist.gender=gender.id

where gender.name='Male' and recording.length =(

select max(query.length) from(

select recording.name,recording.length

from artist join artist\_credit\_nameon artist\_credit\_name.artist=artist.id

join artist\_crediton artist\_credit\_name.artist\_credit=artist\_credit.id

join recordingon artist\_credit.id=recording.artist\_credit

join genderon artist.gender=gender.id

where gender.name='Male'

except

select recording.name,recording.length

from artist join artist\_credit\_nameon artist\_credit\_name.artist=artist.id

join artist\_credit on artist\_credit\_name.artist\_credit=artist\_credit.id

join recording on artist\_credit.id=recording.artist\_credit

join gender on artist.gender=gender.id

where gender.name='Male'and recording.length=(

select max(recording.length)

from artist join artist\_credit\_name on artist\_credit\_name.artist=artist.id

join artist\_credit on artist\_credit\_name.artist\_credit=artist\_credit.id

join recording on artist\_credit.id=recording.artist\_credit

join gender on artist.gender=gender.id

where gender.name='Male'

)

) query

)

11b- viste

create view MaxRegUomo as // crea vista con la registrazione più lunga di un uomo

select max(recording.length)

from recording join artist

on artist.id=recording.artist\_credit

join gender on artist.gender=gender.id

join artist\_credit\_name on artist\_credit\_name.artist= artist.id

join artist\_credit on artist\_credit.id = artist\_credit\_name.artist\_credit

where gender.name = 'Male'

create view AllReg as //trova tutte le registrazioni di un uomo

select artist\_credit.name,recording.name as recName,recording.length as length

from recording join artist

on artist.id=recording.artist\_credit

join gender on artist.gender=gender.id

join artist\_credit\_name on artist\_credit\_name.artist= artist.id

join artist\_credit on artist\_credit.id = artist\_credit\_name.artist\_credit

where gender.name= 'Male'

and recording.length is not null

select AllReg.name,AllReg.recName,length

from AllReg

where length=

(

select max(query.length)

from

(

select AllReg.name,AllReg.length

from AllReg

except

select AllReg.name, AllReg.length

from AllReg

where AllReg.length=

(

select \* from MaxRegMen

)

)query

)

QUERY 12 (versione 1)

**select area.name, sum(recording.length)/60000**

**from area join area\_type**

**on area.type=area\_type.id**

**join artist**

**on artist.area=area.id**

**join artist\_credit\_name**

**on artist\_credit\_name.artist=artist.id**

**join artist\_credit**

**on artist\_credit\_name.artist\_credit=artist\_credit.id**

**join recording**

**on recording.artist\_credit=artist\_credit.id**

**where area\_type.name='Country'**

**group by area.name**

**having sum(recording.length) is not null**

**union**

**select area.name, 0 as length**

**from area join area\_type**

**on area.type=area\_type.id**

**join artist**

**on artist.area=area.id**

**join artist\_credit\_name**

**on artist\_credit\_name.artist=artist.id**

**join artist\_credit**

**on artist\_credit\_name.artist\_credit=artist\_credit.id**

**join recording**

**on recording.artist\_credit=artist\_credit.id**

**where area\_type.name='Country'**

**group by area.name**

**having sum(recording.length) is null**

**1)** Il risultato chiede il nome dello stato (attributo name della tabella area) e la somma delle lunghezze delle registrazioni degli artisti di quello Stato.

**2)** Il primo join è tra la chiave primaria della tabella area\_type e la chiave esterna type della tabella area, il successivo è tra la chiave primaria della tabella area e la chiave esterna area della tabella artist. Viene poi fatto il join tra la chiave primaria id della tabella artist con la chiave esterna artist della tabella artist\_credit\_name. L’altro join è tra le chiavi primarie di artist\_credit e artist\_credit\_name. L’ultimo join è tra la chiave esterna artist\_credit della tabella recording e la chiave primaria della tabella artist\_credit.

Nella query viene utilizzato l’operatore insiemistico union per mettere insieme i risultati degli stati che hanno 0 come somma delle registrazioni con quelli degli stati con somma delle registrazioni maggiore di 0.

**3)** Non sono presenti sotto problemi.

**4)** Valori duplicati non sono presenti, per quanto riguarda i valori nulli si tengono in considerazione nella seconda parte della query e vengono considerati zero, cioè viene messo il valore 0 dove la somma delle registrazioni è null.

**5)** La seconda versione della query è realizzata con le viste.

**6)**

**select area.name, sum(recording.length)/60000**

**from area join area\_type**

**on area.type=area\_type.id**

**join artist**

**on artist.area=area.id**

**join artist\_credit\_name**

**on artist\_credit\_name.artist=artist.id**

**join artist\_credit**

**on artist\_credit\_name.artist\_credit=artist\_credit.id**

**join recording**

**on recording.artist\_credit=artist\_credit.id**

**where area\_type.name!='Country'**

**group by area.name**

**having sum(recording.length) is null**

**and area.name in**

**(**

**select area.name**

**from area join area\_type**

**on area.type=area\_type.id**

**join artist**

**on artist.area=area.id**

**join artist\_credit\_name**

**on artist\_credit\_name.artist=artist.id**

**join artist\_credit**

**on artist\_credit\_name.artist\_credit=artist\_credit.id**

**join recording**

**on recording.artist\_credit=artist\_credit.id**

**where area\_type.name='Country'**

**group by area.name**

**having sum(recording.length) is null**

**)**

Con questa query si verifica che non ci siano aree che non sono Stati in cui ci sia somma registrazioni=0.

**select area.name, sum(recording.length)/60000**

**from area join area\_type**

**on area.type=area\_type.id**

**join artist**

**on artist.area=area.id**

**join artist\_credit\_name**

**on artist\_credit\_name.artist=artist.id**

**join artist\_credit**

**on artist\_credit\_name.artist\_credit=artist\_credit.id**

**join recording**

**on recording.artist\_credit=artist\_credit.id**

**where area\_type.name!='Country'**

**group by area.name**

**having sum(recording.length) is not null**

**and area.name in**

**(**

**select area.name**

**from area join area\_type**

**on area.type=area\_type.id**

**join artist**

**on artist.area=area.id**

**join artist\_credit\_name**

**on artist\_credit\_name.artist=artist.id**

**join artist\_credit**

**on artist\_credit\_name.artist\_credit=artist\_credit.id**

**join recording**

**on recording.artist\_credit=artist\_credit.id**

**where area\_type.name='Country'**

**group by area.name**

**having sum(recording.length) is not null**

**)**

Con questa query si verifica che non ci siano aree che non sono Stati in cui ci sia somma registrazioni>0.

La verifica contemporanea di queste due query permette di dire che la query di partenza è corretta.

QUERY 12 (versione 2)

**create view RegNotNull as**

**select area.name, sum(recording.length)/60000**

**from area join area\_type**

**on area.type=area\_type.id**

**join artist**

**on artist.area=area.id**

**join artist\_credit\_name**

**on artist\_credit\_name.artist=artist.id**

**join artist\_credit**

**on artist\_credit\_name.artist\_credit=artist\_credit.id**

**join recording**

**on recording.artist\_credit=artist\_credit.id**

**where area\_type.name='Country'**

**group by area.name**

**having sum(recording.length) is not null**

**create view RegNull as**

**select area.name, 0 as length**

**from area join area\_type**

**on area.type=area\_type.id**

**join artist**

**on artist.area=area.id**

**join artist\_credit\_name**

**on artist\_credit\_name.artist=artist.id**

**join artist\_credit**

**on artist\_credit\_name.artist\_credit=artist\_credit.id**

**join recording**

**on recording.artist\_credit=artist\_credit.id**

**where area\_type.name='Country'**

**group by area.name**

**having sum(recording.length) is null**

**select \* from**

**RegNotNull**

**union**

**select \* from**

**RegNull**

**1)** I risultati della query sono tutti gli attributi della vista RegNotNull e di quella RegNull.

**2)** I join sono i medesimi della query senza viste.

**3)** Non sono presenti sotto problemi. Sono state create 2 viste, una con gli Stati con somma registrazioni = 0, una con somma registrazioni > 0.

**4)** Il discorso sui duplicati e sui valori nulli vale lo stessso fatto per la versione 1.

**5)** E’ stata fatta una prima versione della query.

**6)** La query è costruita sulla base di quella della versione 1, quindi i controlli fatti per la precedente valgono anche per questa.

13a- select artist.name, area.name, count(release.name)

from artist join area

on area.id=artist.area

join artist\_credit\_name

on artist\_credit\_name.artist=artist.id

join artist\_credit

on artist\_credit\_name.artist\_credit=artist\_credit.id

join release

on release.artist\_credit=artist\_credit.id

where area.name='United Kingdom'

group by artist.name, area.name

having count(release.name)>=10

13b-

create view britainRelease as

select artist.name as artName, area.name as areaName, release.name as relName

from artist join area

on area.id=artist.area

join artist\_credit\_name

on artist\_credit\_name.artist=artist.id

join artist\_credit

on artist\_credit\_name.artist\_credit=artist\_credit.id

join release

on release.artist\_credit=artist\_credit.id

where area.name='United Kingdom'

select artName,areaName,count(relName)

from britainRelease

group by artName, areaName

having count(relName)>=10

QUERY 14 (versione 1)

**select distinct artist.name,count(release.name)**

**from release**

**join artist\_credit**

**on release.artist\_credit=artist\_credit.id**

**join artist\_credit\_name**

**on artist\_credit.id=artist\_credit\_name.artist\_credit**

**join artist**

**on artist\_credit\_name.artist=artist.id**

**join medium**

**on release.id=medium.release**

**join medium\_format**

**on medium.format=medium\_format.id**

**where medium\_format.name='CD'**

**and artist.name not in**

**(**

**select distinct artist.name**

**from medium join track**

**on track.medium=medium.id**

**join medium\_format**

**on medium.format=medium\_format.id**

**join release**

**on medium.release=release.id**

**join artist\_credit**

**on release.artist\_credit=artist\_credit.id**

**join artist\_credit\_name**

**on artist\_credit.id=artist\_credit\_name.artist\_credit**

**join artist**

**on artist\_credit\_name.artist=artist.id**

**where medium\_format.name='CD'**

**and medium.track\_count<=all**

**(**

**select avg(medium.track\_count)**

**from medium**

**join track**

**on track.medium=medium.id**

**join medium\_format**

**on medium.format=medium\_format.id**

**join release**

**on medium.release=release.id**

**where medium\_format.name='CD'**

**)**

**)**

**group by artist.name**

**order by count(release.name) desc**

**1)** Nel risultato vengono richiesti il nome dell’artista ( attributo name della tabella artist) e il numero delle release, ordinato per numero delle release decrescente.

**2**) Il primo join è tra la chiave primaria della tabella medium e la chiave esterna medium ella tabella track, il successivo è tra la chaive esterna format della tabella medium e la chiave primaria della tabella medium\_format. In seguito c’è il join tra la chiave esterna release della tabella medium e la chiave primaria della tabella release. C’è poi il join tra la chiave primaria di artist\_credit e la chiave esterna artist\_credit della tabella release. L’altro join è tra le chiavi primarie di artist\_credit e artist\_credit\_name.L’ultimo è il join tra la chiave primaria id della tabella artist con la chiave esterna artist della tabella artist\_credit\_name.

Per quanto riguarda il calcolo della media sono stati fatti gli stessi join a parte quelli con le tabelle artist,artist\_credit,artist\_credit\_name.

**3)** In questa query sono presenti 2 sotto problemi, il più interno serve per calcolare la media delle tracce, il secondo sotto problema più interno serve invece per trovare tutti gli artisti che hanno release con lunghezza minore della media.

**4)** Si è usato il distinct per i duplicati

**5)** Sono richieste due versioni della query.

**6)**

**select distinct artist.name,count(release.name)**

**from release**

**join artist\_credit**

**on release.artist\_credit=artist\_credit.id**

**join artist\_credit\_name**

**on artist\_credit.id=artist\_credit\_name.artist\_credit**

**join artist**

**on artist\_credit\_name.artist=artist.id**

**join medium**

**on release.id=medium.release**

**join medium\_format**

**on medium.format=medium\_format.id**

**where medium\_format.name='CD'**

**and artist.name not in**

**(**

**select distinct artist.name**

**from medium join track**

**on track.medium=medium.id**

**join medium\_format**

**on medium.format=medium\_format.id**

**join release**

**on medium.release=release.id**

**join artist\_credit**

**on release.artist\_credit=artist\_credit.id**

**join artist\_credit\_name**

**on artist\_credit.id=artist\_credit\_name.artist\_credit**

**join artist**

**on artist\_credit\_name.artist=artist.id**

**where medium\_format.name='CD'**

**and medium.track\_count <= all**

**(**

**select avg(medium.track\_count)**

**from medium**

**join track**

**on track.medium=medium.id**

**join medium\_format**

**on medium.format=medium\_format.id**

**join release**

**on medium.release=release.id**

**where medium\_format.name='CD'**

**)**

**)**

**group by artist.name**

**intersect**

**select distinct artist.name,count(release.name)**

**from release**

**join artist\_credit**

**on release.artist\_credit=artist\_credit.id**

**join artist\_credit\_name**

**on artist\_credit.id=artist\_credit\_name.artist\_credit**

**join artist**

**on artist\_credit\_name.artist=artist.id**

**join medium**

**on release.id=medium.release**

**join medium\_format**

**on medium.format=medium\_format.id**

**where medium\_format.name='CD'**

**and artist.name in**

**(**

**select distinct artist.name**

**from medium join track**

**on track.medium=medium.id**

**join medium\_format**

**on medium.format=medium\_format.id**

**join release**

**on medium.release=release.id**

**join artist\_credit**

**on release.artist\_credit=artist\_credit.id**

**join artist\_credit\_name**

**on artist\_credit.id=artist\_credit\_name.artist\_credit**

**join artist**

**on artist\_credit\_name.artist=artist.id**

**where medium\_format.name='CD'**

**and medium.track\_count <= all**

**(**

**select avg(medium.track\_count)**

**from medium**

**join track**

**on track.medium=medium.id**

**join medium\_format**

**on medium.format=medium\_format.id**

**join release**

**on medium.release=release.id**

**where medium\_format.name='CD'**

**)**

**)**

**group by artist.name**

Questa query cerca l’intersezone tra l’insieme degli artisti che hanno tutte le registrazioni maggiori della media e gli altri. L’intersezione è nulla quindi la query è giusta.

QUERY 14 (versione 2)

**create view Media\_t as**

**select avg(medium.track\_count)**

**from medium**

**join track**

**on track.medium=medium.id**

**join medium\_format**

**on medium.format=medium\_format.id**

**join release**

**on medium.release=release.id**

**where medium\_format.name='CD'**

**create view ArtMinore as**

**select distinct artist.name**

**from medium join track**

**on track.medium=medium.id**

**join medium\_format**

**on medium.format=medium\_format.id**

**join release**

**on medium.release=release.id**

**join artist\_credit**

**on release.artist\_credit=artist\_credit.id**

**join artist\_credit\_name**

**on artist\_credit.id=artist\_credit\_name.artist\_credit**

**join artist**

**on artist\_credit\_name.artist=artist.id**

**where medium\_format.name='CD'**

**and medium.track\_count<=all**

**(**

**select \* from Media\_t**

**)**

**group by artist.name**

**create view AllArt as**

**select distinct artist.name, count(release.name)**

**from release**

**join artist\_credit**

**on release.artist\_credit=artist\_credit.id**

**join artist\_credit\_name**

**on artist\_credit.id=artist\_credit\_name.artist\_credit**

**join artist**

**on artist\_credit\_name.artist=artist.id**

**join medium**

**on release.id=medium.release**

**join medium\_format**

**on medium.format=medium\_format.id**

**where medium\_format.name='CD'**

**group by artist.name**

**select \* from AllArt**

**where AllArt.name not in**

**(**

**select \* from ArtMinore**

**)**

**order by count desc**

**1)** Le richieste sono le stesse della versione 1.

**2)** I join sono gli stessi fatti per la versione 1.

**3)** I sotto problemi sono gli stessi della prima query, sono state create 3 viste, una per calcolare la media(Media\_t), una per trovare tutti gli artist con il numero delle release(ArtAll) e un’altra per trovare tutti gli artisti con registrazioni sotto la media(ArtMinore.

**4)** Il discorso sui duplicati e sui valori nulli vale lo stessso fatto per la versione 1.

**5)** E’ già stata fatta una prima versione della query.

**6)** La query è stata fatta con le viste sulla base della versione 1, quindi la verifica fatta per la versione 1 vale anche per questa .

15a-

select query.name,query.begin\_date\_day,query.begin\_date\_month,query.begin\_date\_year,query.end\_date\_day,query.end\_date\_month,query.end\_date\_year

from

(

select artist.name as name,artist.begin\_date\_day,artist.begin\_date\_month,artist.begin\_date\_year,artist.end\_date\_day,artist.end\_date\_month,artist.end\_date\_year

from artist

where artist.end\_date\_day >

(

select artist.end\_date\_day

from artist

where artist.name = 'Louis Armstrong'

)

and artist.end\_date\_month >

(

select artist.end\_date\_month

from artist

where artist.name = 'Louis Armstrong'

)

and artist.end\_date\_year >

(

select artist.end\_date\_year

from artist

where artist.name = 'Louis Armstrong'

)

)query

where query.end\_date\_day =

(

select min(query.end\_date\_day)

from

(

select artist.name as name,artist.end\_date\_day,artist.end\_date\_month,artist.end\_date\_year

from artist

where artist.end\_date\_day >

(

select artist.end\_date\_day

from artist

where artist.name = 'Louis Armstrong'

)

and artist.end\_date\_month >

(

select artist.end\_date\_month

from artist

where artist.name = 'Louis Armstrong'

)

and artist.end\_date\_year >

(

select artist.end\_date\_year

from artist

where artist.name = 'Louis Armstrong'

)

)query

)

and query.end\_date\_month =

(

select min(query.end\_date\_month)

from

(

select artist.name as name,artist.end\_date\_day,artist.end\_date\_month,artist.end\_date\_year

from artist

where artist.end\_date\_day >

(

select artist.end\_date\_day

from artist

where artist.name = 'Louis Armstrong'

)

and artist.end\_date\_month >

(

select artist.end\_date\_month

from artist

where artist.name = 'Louis Armstrong'

)

and artist.end\_date\_year >

(

select artist.end\_date\_year

from artist

where artist.name = 'Louis Armstrong'

)

)query

)

and query.end\_date\_year =

(

select min(query.end\_date\_year)

from

(

select artist.name as name,artist.end\_date\_day,artist.end\_date\_month,artist.end\_date\_year

from artist

where artist.end\_date\_day >

(

select artist.end\_date\_day

from artist

where artist.name = 'Louis Armstrong'

)

and artist.end\_date\_month >

(

select artist.end\_date\_month

from artist

where artist.name = 'Louis Armstrong'

)

and artist.end\_date\_year >

(

select artist.end\_date\_year

from artist

where artist.name = 'Louis Armstrong'

)

)query

)

15b-

create view DayDeath as

select artist.end\_date\_day

from artist

where artist.name = 'Louis Armstrong'

create view MonthDeath as

select artist.end\_date\_month

from artist

where artist.name = 'Louis Armstrong'

create view YearDeath as

select artist.end\_date\_year

from artist

where artist.name = 'Louis Armstrong'

create view DeathAfterArm as

select artist.name as name,artist.end\_date\_day,artist.end\_date\_month,artist.end\_date\_year

from artist

where artist.end\_date\_day >

(

select \*

from DayDeath

)

and artist.end\_date\_month >

(

select \*

from MonthDeath

)

and artist.end\_date\_year >

(

select \*

from YearDeath

)

select query.name,query.begin\_date\_day,query.begin\_date\_month,query.begin\_date\_year,query.end\_date\_day,query.end\_date\_month,query.end\_date\_year

from

(

select artist.name as name,artist.begin\_date\_day,artist.begin\_date\_month,artist.begin\_date\_year,artist.end\_date\_day,artist.end\_date\_month,artist.end\_date\_year

from artist

where artist.end\_date\_day >

(

select \*

from DayDeath

)

and artist.end\_date\_month >

(

select \*

from MonthDeath

)

and artist.end\_date\_year >

(

select \*

from YearDeath

)

)query

where query.end\_date\_day =

(

select min(DeathAfterArm.end\_date\_day)

from DeathAfterArm

)

and query.end\_date\_month =

(

select min(DeathAfterArm.end\_date\_month)

from DeathAfterArm

)

and query.end\_date\_year =

(

select min(DeathAfterArm.end\_date\_year)

from DeathAfterArm

)

­­­­