**DATABASE SYSTEMS:5330**

**PROJECT 1**

**DATABASE FOR MANAGEMENT OF YOUR MUSIC COLLECTION**

**NAME:**

1)SUJAY NATARAJAN(1001086537)

2)VINCENT

3)SAMSON YERRANGUNTLA

4)KINSHUK KASHYUP

5)POONAM SATHE DEEPAK

**TASK1**

**A.** **Make a list of the entities for the above process, explaining why each is a separate entity.**

**1)ALBUM:**

The entity album contains list of albums and their respective id’s. The core of a music collection is the list of albums that I have. It has the following fields album\_name, album\_id, description and rating so that there is only one rating per album.

**2)ARTIST**:

Artist entity contains the details about the different artists with a description and an id. The artist entity is chosen to provide details about the numerous artists that my collection can contain.

**3)GENRE**:

Genre describes the various genres that an album can be associated with. Genre is a separate entity because multiple genres can be associated to an album and it is easier to keep a list of genres separately. Genre contains genre\_id, description, genre category.

**4)ALBUM\_ARTIST**:

Album\_artist is a separate entity which contains the details about albums with its associated artists. It contains album\_id and artist\_id to associate different artists to different albums.

**5)ALBUM\_GENRE**:

This entity deals with associating different genres to different albums. This is separate because there can be many genres to same album and that is kept track using the album\_id and genre\_id.

**6)MY\_COLLECTION**:

This entity is the core of my music collection and contains various details about albums,release and record date etc. My\_Collection describes the collection of albums that I have in my collection which will be able to point out if I have two or more copies of the same album in different conditions or with different disktypes. This entity is chosen because it can help keep track of the albums with a collection id.

**B.** **For each entity define the required attributes and the data type of each of those attributes. You can include any attributes you believe that should be there, based on the mini-world description.**

**1)ALBUM:**

|  |  |
| --- | --- |
| ATTRIBUTE | DATATYPE |
| Album\_id | Integer |
| Album\_name | Varchar |
| Album\_decription | Varchar |
| Rating | Integer |

**2)ARTIST:**

|  |  |
| --- | --- |
| ATTRIBUTE | DATATYPE |
| Artist\_id | Integer |
| Artist\_name | Integer |

**3)GENRE:**

|  |  |
| --- | --- |
| ATTRIBUTE | DATATYPE |
| Genre\_id | Integer |
| Genre | Varchar |
| Genre\_description | Varchar |

**4)ALBUM\_ARTIST:**

|  |  |
| --- | --- |
| ATTRIBUTE | DATATYPE |
| Album\_id | Integer |
| Artist\_id | Integer |

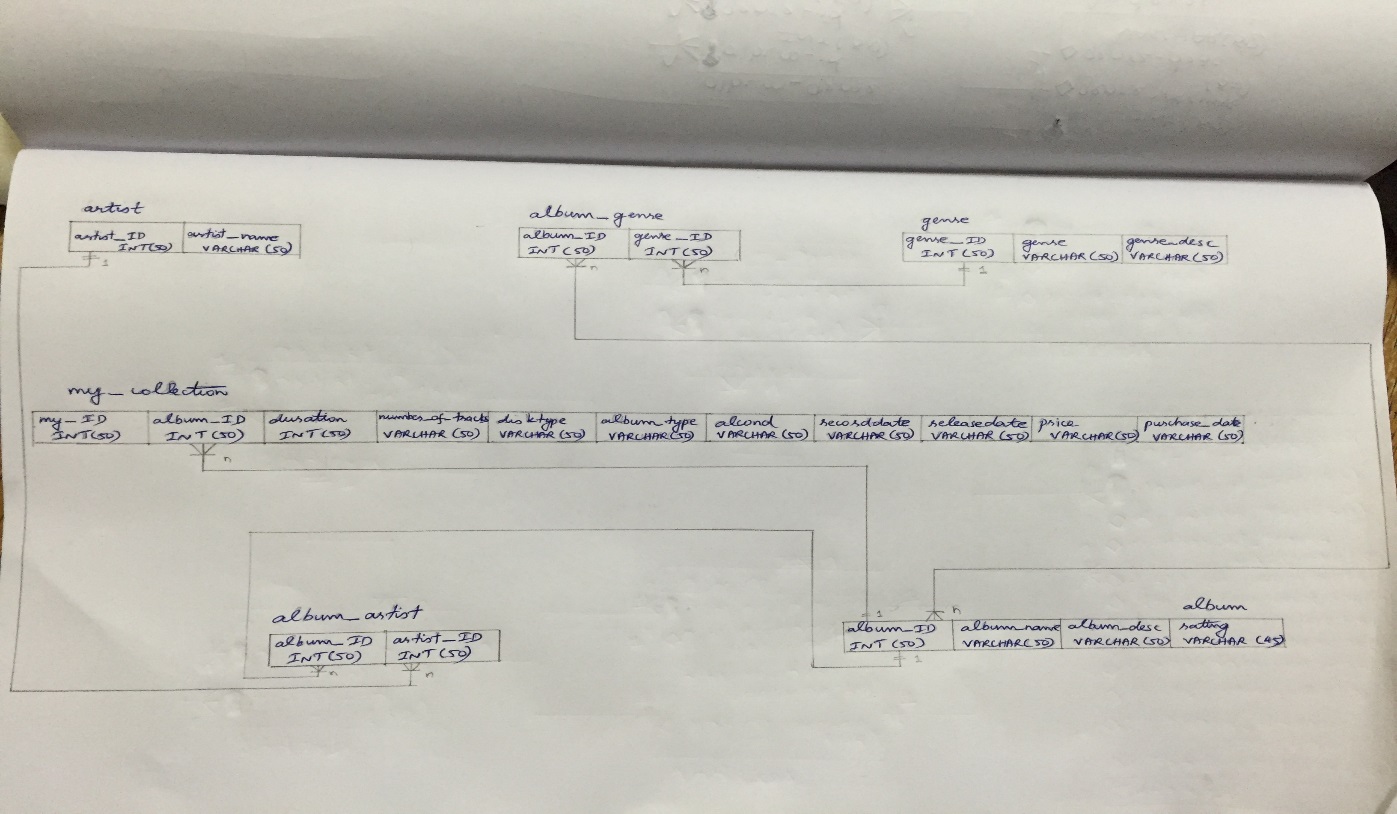
**5)ALBUM\_GENRE:**

|  |  |
| --- | --- |
| ATTRIBUTE | DATATYPE |
| Album\_id | Integer |
| Genre\_id | Integer |

**6)MY\_COLLECTION:**

|  |  |
| --- | --- |
| ATTRIBUTE | DATATYPE |
| My\_id | Integer |
| Album\_id | Integer |
| Duration | Integer |
| Number\_of\_Tracks | Varchar |
| Disktype | Varchar |
| Album\_type | Varchar |
| Alcond | Varchar |
| Recorddate | Varchar |
| Releasedate | Varchar |
| Price | Integer |
| Purchase\_date | Varchar |

**TASK2**

**A.** **Design the database schema (diagram) using arrows to define the references, based on the description. Your schema should be handwritten at this point.**

**B.** **Explain why you took specific decisions regarding: ♣ The relations (tables) you have created. ♣ The cardinality of the relationships and the references you made.**

Album table has attribute album\_id which is the primary key and is set to not null and uniquely identifies the albums. Artist entity contains primary key artist\_id and is also set to not null. The album\_artist tables contains the album\_id and artist\_id as a composite key to signify that each album can have many artists and thus album table and album\_artist table is connected in a one to many fashion. The Genre entity contains genre\_id as primary key which contains unique ids of different genres. The album\_genre entity contains both album\_id and genre\_id as a composite key to show that each album can have different genres and different albums can be assigned to one genre. Thus this has a many to many relationship. My\_collection has my\_id and album\_id as a composite key which will be helpful in determining that mutiplecopies of the same album can be present in a collection and the composite key uniquely identifies that the same album can be present in the collection in different conditions like CD/Vinyl. Thus my\_collection and album are related in a one to many manner. The album\_id in my\_collection,album\_genre,album\_artist is a foreign key referencing the album\_id in Album table. Similarly Genre\_id in album\_genre and artist\_id in album\_artist are foreign keys referencing genre,artist entities repesctively.

**C.** **Based on your decisions, make a list of the three fundamental types of constraints that are existing on your schema, mentioning the type of each of the constraints. Refer to the schema constraints only.**

The three fundamental constraints existing in my schema are :

1)Primary Key

2)foreign Key

3) Set Not Null

And my schema also has set default values if no value is provided .

Album\_id, genre\_id, artist\_id are three different primary keys as explained above for album, genre, artist tables repectively.

Album\_id, Genre\_id of the Album\_genre table are foreign keys referencing to the album\_id and genre\_id of tables album,genre\_repectively. Similarly Album\_id, Artist\_id of album\_artist table are foreign key references to album\_id and artist\_id of albm, artist tables repecrively.

Album\_id of the album table is set to not null and so no null values can be provided for album\_id and similarly has been set for artist\_id and genre\_id.

Recorddate field of my\_collection table has additional constraint of setting default value to null if no value is provided.

**Task 3**

**3a)** **Write down SQL code to define the above mentioned database schema. Each one of the relations should have the appropriate attributes.**

**ALBUM:**

CREATE TABLE `album` (   `album\_id` int(50) NOT NULL,   `album\_name` varchar(50) DEFAULT NULL,   `album\_desc` varchar(50),   `rating` varchar(45) DEFAULT NULL,   PRIMARY KEY (`album\_id`) );

**ARTIST:**

CREATE TABLE `artist` (   `artist\_id` int(50) NOT NULL,   `artist\_name` varchar(50) DEFAULT NULL,   PRIMARY KEY (`artist\_id`) ) ;

**GENRE:**

CREATE TABLE `genre` (   `genre\_id` int(50) NOT NULL,   `genre` varchar(50),   `genre\_desc` varchar(50) DEFAULT NULL,   PRIMARY KEY (`genre\_id`) );

**ALBUM\_ARTIST:**

CREATE TABLE `album\_artist` (

`album\_id` int(50) NOT NULL DEFAULT '0',

`artist\_id` int(50) NOT NULL DEFAULT '0',

PRIMARY KEY (`album\_id`,`artist\_id`),

CONSTRAINT FOREIGN KEY (`album\_id`) REFERENCES `album` (`album\_id`) on delete restrict,

CONSTRAINT FOREIGN KEY (`artist\_id`) REFERENCES `artist` (`artist\_id`) on update no action

) ;

**ALBUM\_GENRE:**

CREATE TABLE `album\_genre` (

`album\_id` int(50) NOT NULL DEFAULT '0',`genre\_id` int(50) NOT NULL DEFAULT '0',

PRIMARY KEY (`album\_id`,`genre\_id`),

CONSTRAINT `album\_genre\_ibfk\_1` FOREIGN KEY (`album\_id`) REFERENCES `album` (`album\_id`) on delete restrict ,

CONSTRAINT `album\_genre\_ibfk\_2` FOREIGN KEY (`genre\_id`) REFERENCES `genre` (`genre\_id`) on delete restrict,

) ;

**MY\_COLLECTION:**

CREATE TABLE `my\_collection` (

`my\_id` int(50) NOT NULL DEFAULT '0',

`album\_id` int(50) NOT NULL DEFAULT '0',

`duration` int(50),

`number\_of\_tracks` varchar(50),

`disktype` varchar(50),

`album\_type` varchar(50),

`alcond` varchar(50),

`recorddate` datetime

`releasedate` datetime DEFAULT NULL,

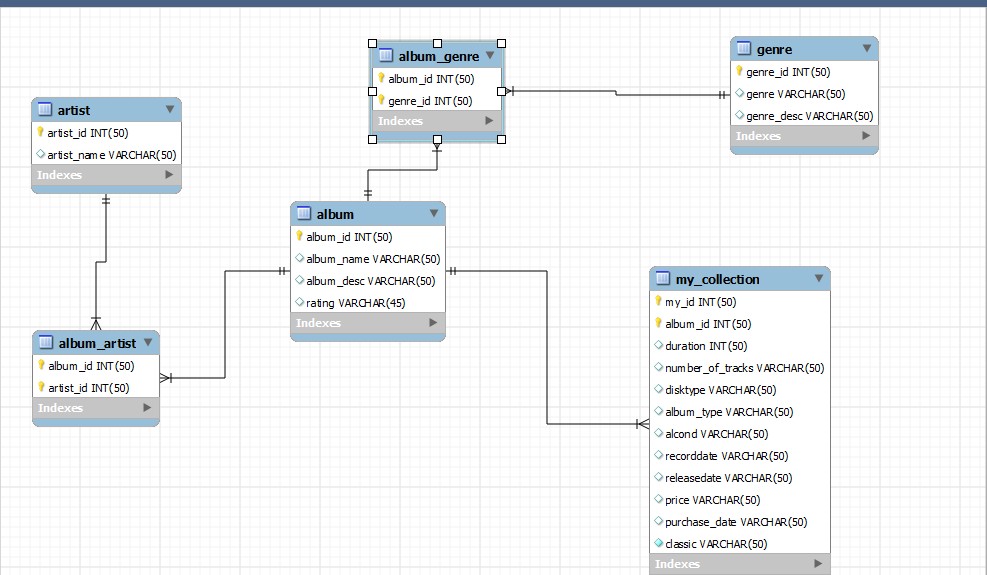
`price` varchar(50) DEFAULT NULL,

`purchase\_date` datetime DEFAULT NULL,

PRIMARY KEY (`my\_id`,`album\_id`),

CONSTRAINT ` FOREIGN KEY (`album\_id`) REFERENCES `album`’(`album\_id`) on update no action

) ;



**ON UPDATE AND ON DELETE:**

1)ALBUM\_ARTIST:

CONSTRAINT FOREIGN KEY (`album\_id`) REFERENCES `album` (`album\_id`) on update cascade on delete restrict,

CONSTRAINT FOREIGN KEY (`artist\_id`) REFERENCES `artist` (`artist\_id`) on update cascade on delete restrict

On update of album\_id and artist\_id has been set to cascade so that if there is any changes in album or artist table it will be cascaded and on delete operation is restricted so that no values from artist or album is deleted.

2)ALBUM\_GENRE:

CONSTRAINT `album\_genre\_ibfk\_1` FOREIGN KEY (`album\_id`) REFERENCES `album` (`album\_id`) on delete restrict on update cascade ,

CONSTRAINT `album\_genre\_ibfk\_2` FOREIGN KEY (`genre\_id`) REFERENCES `genre` (`genre\_id`) on delete restrict on update cascade,

On delete operation on Album\_id and genre\_id is restricted so that no deletion occurs in album or genre table and if there is any update it is cascaded.

3)MY\_COLLECTION:

CONSTRAINT ` FOREIGN KEY (`album\_id`) REFERENCES `album`’(`album\_id`) on update no action

On update of my collection album id then no action is taken on album\_id in album table so that no values are modified.

**3b) Once your schema is ready, provide the appropriate SQL code to populate the database with sample data.**

**ALBUM:**

INSERT INTO `album` (`album\_id`,`album\_name`,`album\_desc`,`rating`) VALUES (1,'Master of puppets','favorite','5');

INSERT INTO `album` (`album\_id`,`album\_name`,`album\_desc`,`rating`) VALUES (2,'Load','another brick in the wall','5');

INSERT INTO `album` (`album\_id`,`album\_name`,`album\_desc`,`rating`) VALUES (3,'Mall','parkit','3');

INSERT INTO `album` (`album\_id`,`album\_name`,`album\_desc`,`rating`) VALUES (4,'Slumdog','BGM','2');

INSERT INTO `album` (`album\_id`,`album\_name`,`album\_desc`,`rating`) VALUES (5,'back in black','highway to hell','2');

INSERT INTO `album` (`album\_id`,`album\_name`,`album\_desc`,`rating`) VALUES (6,'Powerslave','favorite','1');

INSERT INTO `album` (`album\_id`,`album\_name`,`album\_desc`,`rating`) VALUES (7,'Roja','ARR','4');

INSERT INTO `album` (`album\_id`,`album\_name`,`album\_desc`,`rating`) VALUES (8,'Death Magnetic','Rock','5');

INSERT INTO `album` (`album\_id`,`album\_name`,`album\_desc`,`rating`) VALUES (9,'Ride the lightening','awesome','3');

INSERT INTO `album` (`album\_id`,`album\_name`,`album\_desc`,`rating`) VALUES (10,'another brick in the wall','no education','4');

**ARTIST:**

INSERT INTO `artist` (`artist\_id`,`artist\_name`) VALUES (1,'Metallica');

INSERT INTO `artist` (`artist\_id`,`artist\_name`) VALUES (2,'pink floyd');

INSERT INTO `artist` (`artist\_id`,`artist\_name`) VALUES (3,'linkin park');

INSERT INTO `artist` (`artist\_id`,`artist\_name`) VALUES (4,'Slumdog');

INSERT INTO `artist` (`artist\_id`,`artist\_name`) VALUES (5,'acdc');

INSERT INTO `artist` (`artist\_id`,`artist\_name`) VALUES (6,NULL);

INSERT INTO `artist` (`artist\_id`,`artist\_name`) VALUES (7,NULL);

INSERT INTO `artist` (`artist\_id`,`artist\_name`) VALUES (8,'ARR');

INSERT INTO `artist` (`artist\_id`,`artist\_name`) VALUES (9,NULL);

INSERT INTO `artist` (`artist\_id`,`artist\_name`) VALUES (10,'Iron Maiden');

**GENRE:**

INSERT INTO `genre` (`genre\_id`,`genre`,`genre\_desc`) VALUES (1,'hard rock','headbang');

INSERT INTO `genre` (`genre\_id`,`genre`,`genre\_desc`) VALUES (2,'pop','have fun');

INSERT INTO `genre` (`genre\_id`,`genre`,`genre\_desc`) VALUES (3,'metal','listen');

INSERT INTO `genre` (`genre\_id`,`genre`,`genre\_desc`) VALUES (4,'classical','rythym');

INSERT INTO `genre` (`genre\_id`,`genre`,`genre\_desc`) VALUES (5,'country','normal');

INSERT INTO `genre` (`genre\_id`,`genre`,`genre\_desc`) VALUES (6,'techno','dj');

**ALBUM\_GENRE**

INSERT INTO `album\_genre` (`album\_id`,`genre\_id`) VALUES (1,1);

INSERT INTO `album\_genre` (`album\_id`,`genre\_id`) VALUES (6,1);

INSERT INTO `album\_genre` (`album\_id`,`genre\_id`) VALUES (3,2);

INSERT INTO `album\_genre` (`album\_id`,`genre\_id`) VALUES (8,2);

INSERT INTO `album\_genre` (`album\_id`,`genre\_id`) VALUES (2,3);

INSERT INTO `album\_genre` (`album\_id`,`genre\_id`) VALUES (5,3);

INSERT INTO `album\_genre` (`album\_id`,`genre\_id`) VALUES (10,3);

INSERT INTO `album\_genre` (`album\_id`,`genre\_id`) VALUES (4,4);

INSERT INTO `album\_genre` (`album\_id`,`genre\_id`) VALUES (7,4);

INSERT INTO `album\_genre` (`album\_id`,`genre\_id`) VALUES (4,5);

INSERT INTO `album\_genre` (`album\_id`,`genre\_id`) VALUES (7,5);

INSERT INTO `album\_genre` (`album\_id`,`genre\_id`) VALUES (9,5);

INSERT INTO `album\_genre` (`album\_id`,`genre\_id`) VALUES (5,6);

**ALBUM\_ARTIST:**

INSERT INTO `album\_artist` (`album\_id`,`artist\_id`) VALUES (1,1);

INSERT INTO `album\_artist` (`album\_id`,`artist\_id`) VALUES (8,1);

INSERT INTO `album\_artist` (`album\_id`,`artist\_id`) VALUES (1,2);

INSERT INTO `album\_artist` (`album\_id`,`artist\_id`) VALUES (2,2);

INSERT INTO `album\_artist` (`album\_id`,`artist\_id`) VALUES (9,2);

INSERT INTO `album\_artist` (`album\_id`,`artist\_id`) VALUES (10,2);

INSERT INTO `album\_artist` (`album\_id`,`artist\_id`) VALUES (3,3);

INSERT INTO `album\_artist` (`album\_id`,`artist\_id`) VALUES (2,4);

INSERT INTO `album\_artist` (`album\_id`,`artist\_id`) VALUES (7,4);

INSERT INTO `album\_artist` (`album\_id`,`artist\_id`) VALUES (3,6);

INSERT INTO `album\_artist` (`album\_id`,`artist\_id`) VALUES (5,7);

INSERT INTO `album\_artist` (`album\_id`,`artist\_id`) VALUES (2,8);

INSERT INTO `album\_artist` (`album\_id`,`artist\_id`) VALUES (4,8);

INSERT INTO `album\_artist` (`album\_id`,`artist\_id`) VALUES (5,8);

INSERT INTO `album\_artist` (`album\_id`,`artist\_id`) VALUES (5,9);

INSERT INTO `album\_artist` (`album\_id`,`artist\_id`) VALUES (6,9);

INSERT INTO `album\_artist` (`album\_id`,`artist\_id`) VALUES (5,10);

INSERT INTO `album\_artist` (`album\_id`,`artist\_id`) VALUES (6,10);

**MY\_COLLECTION:**

INSERT INTO `my\_collection` (`my\_id`,`album\_id`,`duration`,`number\_of\_tracks`,`disktype`,`album\_type`,`alcond`,`recorddate`,`releasedate`,`price`,`purchase\_date`,`numberofcopies`,`classic`) VALUES (1,1,1510,'5','vinyl','band','good','12-1-1988','2-2-1990','20','1-10-1995',1,'1');

INSERT INTO `my\_collection` (`my\_id`,`album\_id`,`duration`,`number\_of\_tracks`,`disktype`,`album\_type`,`alcond`,`recorddate`,`releasedate`,`price`,`purchase\_date`,`numberofcopies`,`classic`) VALUES (2,1,460,'2','Vinyl','Band','Mint','1-11-985','2-3-1988','25','10-12-1994',2,'1');

INSERT INTO `my\_collection` (`my\_id`,`album\_id`,`duration`,`number\_of\_tracks`,`disktype`,`album\_type`,`alcond`,`recorddate`,`releasedate`,`price`,`purchase\_date`,`numberofcopies`,`classic`) VALUES (3,10,1410,'10','vinyl','Compilation','Mint','10-5-1988','2-4-2000','23','8-8-1990',1,'1');

INSERT INTO `my\_collection` (`my\_id`,`album\_id`,`duration`,`number\_of\_tracks`,`disktype`,`album\_type`,`alcond`,`recorddate`,`releasedate`,`price`,`purchase\_date`,`numberofcopies`,`classic`) VALUES (4,7,2010,'10','Vinyl','Compilation','mint','5-10-1991','10-5-1995','15',NULL,4,'0');

INSERT INTO `my\_collection` (`my\_id`,`album\_id`,`duration`,`number\_of\_tracks`,`disktype`,`album\_type`,`alcond`,`recorddate`,`releasedate`,`price`,`purchase\_date`,`numberofcopies`,`classic`) VALUES (5,9,440,'1','CD','single','good','1-12-2012',NULL,'30','9-10-2015',3,'1');

INSERT INTO `my\_collection` (`my\_id`,`album\_id`,`duration`,`number\_of\_tracks`,`disktype`,`album\_type`,`alcond`,`recorddate`,`releasedate`,`price`,`purchase\_date`,`numberofcopies`,`classic`) VALUES (6,10,1410,'2','Vinyl','Band','poor','2-3-2015','10-10-2015','0.5',NULL,5,'0');

INSERT INTO `my\_collection` (`my\_id`,`album\_id`,`duration`,`number\_of\_tracks`,`disktype`,`album\_type`,`alcond`,`recorddate`,`releasedate`,`price`,`purchase\_date`,`numberofcopies`,`classic`) VALUES (7,6,440,'1','cd','single','Mint','20-12-1976','5-5-1980','12','11-12-1988',4,'0');

INSERT INTO `my\_collection` (`my\_id`,`album\_id`,`duration`,`number\_of\_tracks`,`disktype`,`album\_type`,`alcond`,`recorddate`,`releasedate`,`price`,`purchase\_date`,`numberofcopies`,`classic`) VALUES (8,7,440,'1','cd','single','acceptable','1-5-1976','6-6-1980','10','12-11-1984',5,'1');

INSERT INTO `my\_collection` (`my\_id`,`album\_id`,`duration`,`number\_of\_tracks`,`disktype`,`album\_type`,`alcond`,`recorddate`,`releasedate`,`price`,`purchase\_date`,`numberofcopies`,`classic`) VALUES (9,5,540,'1','CD','single','mint','5-5-200','10-6-2002','10','8-5-2003',4,'0');

INSERT INTO `my\_collection` (`my\_id`,`album\_id`,`duration`,`number\_of\_tracks`,`disktype`,`album\_type`,`alcond`,`recorddate`,`releasedate`,`price`,`purchase\_date`,`numberofcopies`,`classic`) VALUES (10,2,480,'1','CD','Band','acceptable','4-10-2003','11-7-2004','10','7-7-2005',1,'0');

INSERT INTO `my\_collection` (`my\_id`,`album\_id`,`duration`,`number\_of\_tracks`,`disktype`,`album\_type`,`alcond`,`recorddate`,`releasedate`,`price`,`purchase\_date`,`numberofcopies`,`classic`) VALUES (11,3,1200,'4','Vinyl','Compilation','Poor','6-10-1949','8-10-1922','17','7-5-2001',2,'0');

INSERT INTO `my\_collection` (`my\_id`,`album\_id`,`duration`,`number\_of\_tracks`,`disktype`,`album\_type`,`alcond`,`recorddate`,`releasedate`,`price`,`purchase\_date`,`numberofcopies`,`classic`) VALUES (12,4,1530,'5','CD','Compilation','Good','6-11-1927','8-8-1949','30','5-9-1992',4,'0');

INSERT INTO `my\_collection` (`my\_id`,`album\_id`,`duration`,`number\_of\_tracks`,`disktype`,`album\_type`,`alcond`,`recorddate`,`releasedate`,`price`,`purchase\_date`,`numberofcopies`,`classic`) VALUES (13,8,1000,'3','Vinyl','Compilation','poor','10-11-1998','10-9-1998','0.2','7-11-1998',5,'1');

INSERT INTO `my\_collection` (`my\_id`,`album\_id`,`duration`,`number\_of\_tracks`,`disktype`,`album\_type`,`alcond`,`recorddate`,`releasedate`,`price`,`purchase\_date`,`numberofcopies`,`classic`) VALUES (14,8,1000,'3','Vinyl','Compilation','mint','11-10-1998','8-10-1998','35','26-11-1998',4,'1');

INSERT INTO `my\_collection` (`my\_id`,`album\_id`,`duration`,`number\_of\_tracks`,`disktype`,`album\_type`,`alcond`,`recorddate`,`releasedate`,`price`,`purchase\_date`,`numberofcopies`,`classic`) VALUES (15,2,4030,'2','CD','Single','mint','12-12-2015','10-12-2015','100','9-11-2015',5,'1');

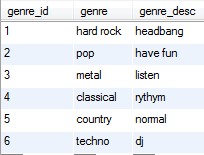
INSERT INTO `my\_collection` (`my\_id`,`album\_id`,`duration`,`number\_of\_tracks`,`disktype`,`album\_type`,`alcond`,`recorddate`,`releasedate`,`price`,`purchase\_date`,`numberofcopies`,`classic`) VALUES (16,2,1230,'1','cd','band','mint','1-2-2014','12-4-2014','10','20-12-2015',2,'1');

**3C) What would be the INITIAL STATE of your database? You can answer by providing an example.**

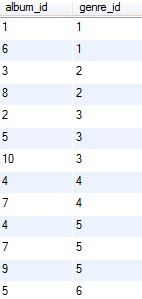
The Initial state of Album table contains 10 albums with their respective rating and description. The album\_id is numbered from 1-10 and is unique.



The Initial state of Genre table contains the values of the six different genres provided with the description.



The Album\_genre table contains the id’s of albums’s and genre’s associating many genres to an album

.

**TASK4:**

1. Show a list of all the albums that you own and the condition of each album

select album\_name,album.album\_id,alcond from album,my\_collection where album.album\_id=my\_collection.album\_id;



2)Show the total number of your albums that you have in your collection.

select count(album\_id) from my\_collection;

G:\2.jpg

3) Show the total number of your UNIQUE albums that you have in your collection.

select distinct album.album\_id,album\_name from my\_collection,album where album.album\_id=my\_collection.album\_id;



4) Show for each released album the number of times you own it. Re-releases of a same album count as different releases.

select

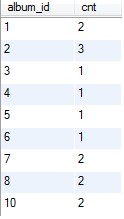
album\_id,count(1) as cnt

from

my\_collection

where releasedate is not null

group by album\_id;



5) Show the total cost of your collection.

select sum(price) from my\_collection;

G:\5.jpg

6) Show only those albums that you own more than once

select

album.album\_id,album\_name, count(1) as cnt

from

album,

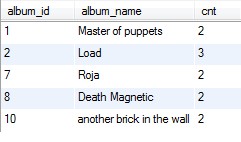
my\_collection

where

album.album\_id = my\_collection.album\_id

group by album\_id

having cnt>1;



\*7) Show the artist for whom you have the largest number of items (counting duplicates).

select

album.album\_id, artist.artist\_id,artist\_name,count(1) as cnt

from

album\_artist,

album,

artist,

my\_collection

where

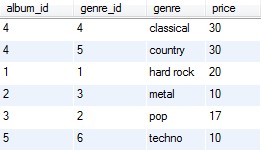
album.album\_id = my\_collection.album\_id

and artist.artist\_id = album\_artist.artist\_id and album.album\_id=album\_artist.album\_id

group by artist\_id;

8) Show your most expensive album per music genre category

select album.album\_id,genre.genre\_id,genre,price from genre,album\_genre,album,my\_collection where album.album\_id=my\_collection.album\_id and album.album\_id=album\_genre.album\_id and genre.genre\_id=album\_genre.genre\_id group by genre having max(price) ;



9) Show the albums purchased after a given year (you decide) with a music genre category “country” or “techno”.

select album.album\_id,album\_name,genre,purchase\_date from album,my\_collection,album\_genre,genre where album.album\_id=my\_collection.album\_id and genre.genre\_id=album\_genre.genre\_id and album.album\_id=album\_genre.album\_id and (genre like'country' or genre like 'classical') having purchase\_date>'1980';



10) Show a list of your albums with more than one music genre categories.

select

album.album\_id,album\_name,count(1) as cnt

from

album,

genre,

album\_genre

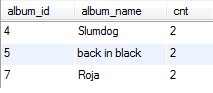
where

album.album\_id = album\_genre.album\_id

and genre.genre\_id = album\_genre.genre\_id

group by album\_id

having cnt>1;



11) Show a list of your albums with more than one music genre categories, which are NOT compilations.

select

album.album\_id,album\_name,count(1) as cnt,album\_type

from

album,

genre,

album\_genre,

my\_collection

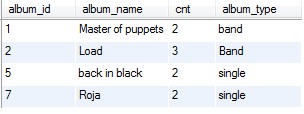
where

album.album\_id = album\_genre.album\_id

and genre.genre\_id = album\_genre.genre\_id and album.album\_id=my\_collection.album\_id and album\_type not like'compilation'

group by album\_id

having cnt>1;



12) Show your top 5 rated vinyl albums which are in mint condition.

select album.album\_id,album\_name,alcond,rating,disktype from album,my\_collection where album.album\_id=my\_collection.album\_id and disktype like 'vinyl'having alcond like'mint' order by rating desc limit 5;



13) Show the average price you have paid per format -CD/vinyl- (counting duplicates)

select avg(price),disktype from my\_collection group by disktype;



14) What is your favorite re-released album?

\*15) Show the least expensive albums which is in mint condition

select price,album.album\_id,album\_name,alcond from album,my\_collection where album.album\_id=my\_collection.album\_id and (price=(select min(price) from my\_collection)) and alcond like 'mint';

16) Show how many mint albums you own per each recording year.

select count(album.album\_id),album\_name,recorddate from album,my\_collection where album.album\_id=my\_collection.album\_id and alcond like 'mint' group by recorddate;



\*17) Show the albums which have NO artist stored into the database.

select album.album\_id,album\_name,artist.artist\_id,artist\_name from my\_collection,album,artist,album\_artist where album.album\_id=my\_collection.album\_id and album.album\_id=album\_artist.album\_id and artist.artist\_id=album\_artist.artist\_id and artist\_name is null;

18) Show all the albums with all their artists. The result should show all albums and all artists stored into the database, even if, i.e., an album is missing its artist information.

select album.album\_id,artist.artist\_id,album\_name,artist\_name from album,artist,album\_artist where album.album\_id=album\_artist.album\_id and artist.artist\_id=album\_artist.artist\_id;



19) What percentage (%) of your single-artist albums that you rated with more than three stars?

select

(got/total)\*100 as percentage

from

(select

count(album\_id) as total

from

my\_collection where album\_type = 'single') as t,

(select

count(album.album\_id) as got

from

my\_collection, album

where

album.album\_id = my\_collection.album\_id

and album\_type = 'single'

and rating > 3) as y;

G:\19.jpg

20) If a typical CD weights 4 oz. and a typical vinyl disk weighs 9 oz. what is the weight of your entire collection in pounds?

select cd\_weight+vinyl as sum from

(select

count(disktype)\*4 as CD\_WEIGHT

from

my\_collection

where

disktype='CD') as t,

(select

count(disktype)\*9 as vinyl

from

my\_collection

where

disktype='vinyl') as y;

G:\20.jpg

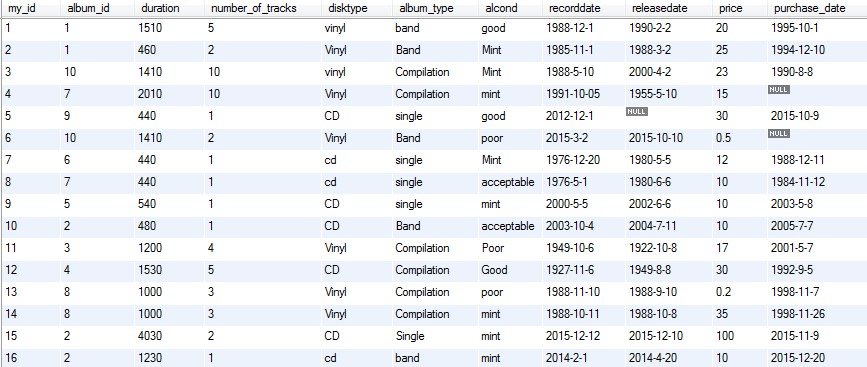
**TASK-5**

1) Write a query which would update the album duration: increase by 10 seconds the duration of all your CD albums

update my\_collection

set duration=duration+10

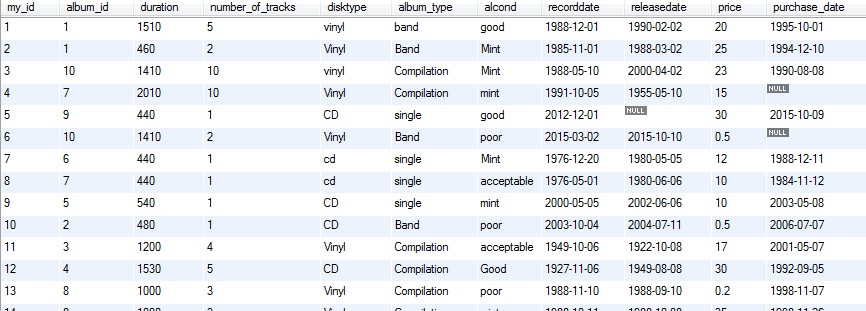
where disktype= 'cd';



2) Write a query which would delete all albums that you bought in 2005, their condition is "poor" and cost less than $1.00

delete from my\_collection

where purchasedate like'%2005' and alcond='poor' and price<1;



3) Write a query which would delete the relation which stores the information about the artists

Drop table artist;

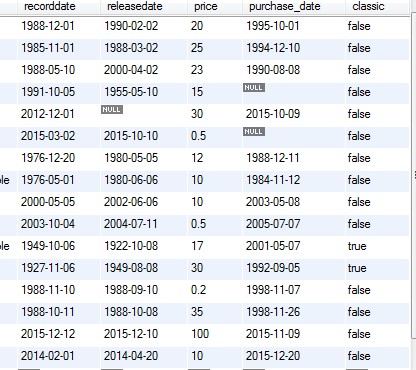
4) Write a query which would create an extra attribute in the appropriate relation. Give the name ‘classic’ to this attribute. The column then should be populated with data: values ‘true’ if the album was recorded before the year 1950 and ‘false’ if the album was recorded after 1950

alter table my\_collection add column classic varchar(10);

Update my\_collection

Set

Classic=if(year(recorddate)>1950,’false’,’true’);

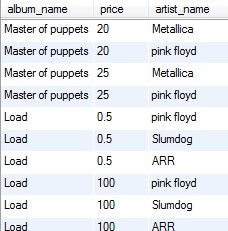


5) The album title with the money paid and its artist(s).

create or replace view v2 as

select album\_name,price,artist\_name from album,artist,album\_artist,my\_collection where album.album\_id=my\_collection.album\_id and album.album\_id=album\_artist.album\_id and artist.artist\_id=album\_artist.artist\_id;

select \* from v2;

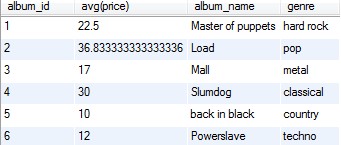


6) The album title with the music genre category and the condition.

create or replace view v1 as

select album\_name,genre,alcond from album,genre,my\_collection,album\_genre where album.album\_id=my\_collection.album\_id and genre.genre\_id=album\_genre.genre\_id and album.album\_id=album\_genre.album\_id;

select \* from v1;



7)