DBMS MIDSEM GROUP 58

Arham Ali(2019411), Ayush Arpit(2019029), Manvik Arya(2019432)

WOW - Done

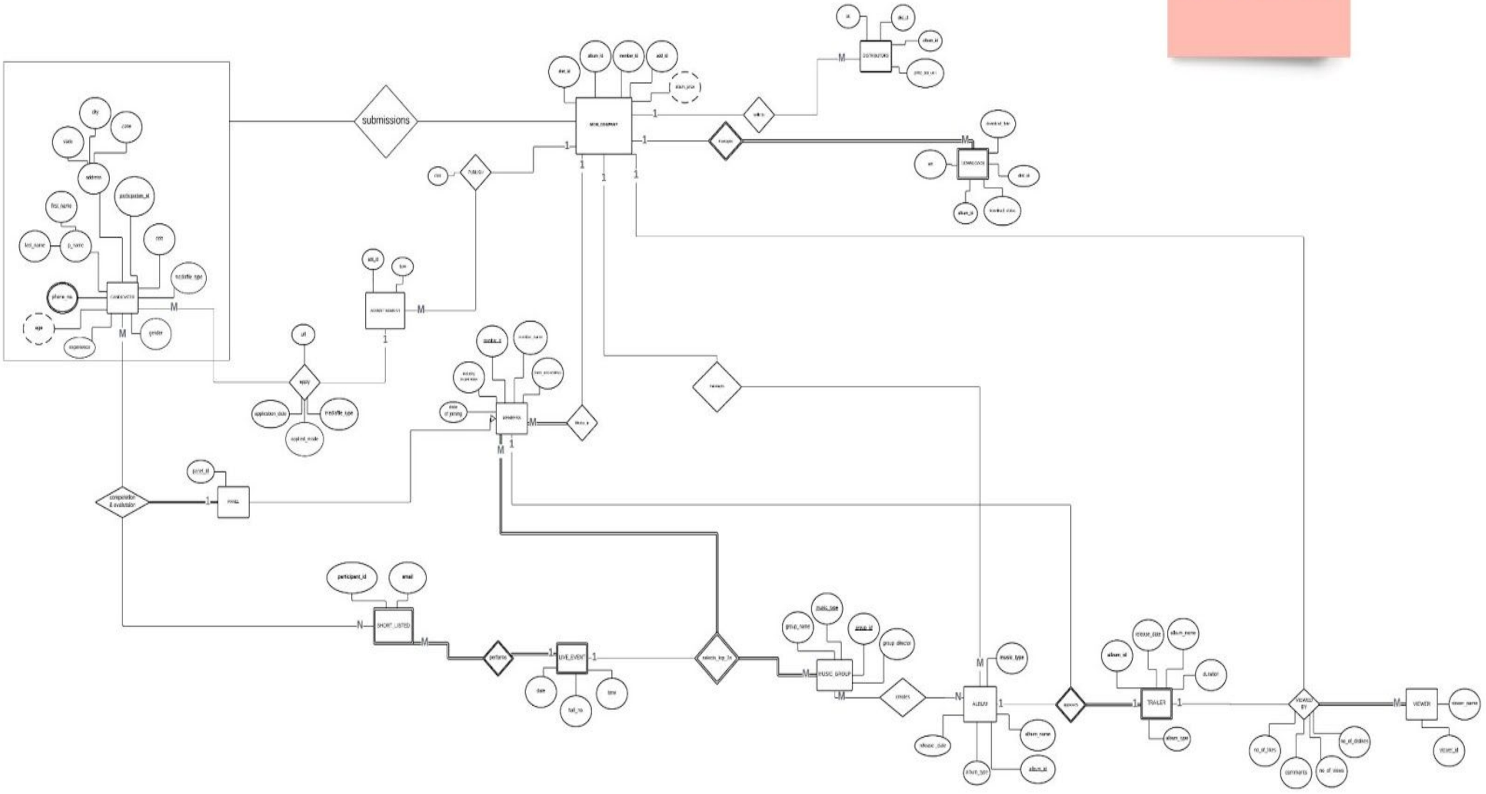
Q1.

Assumptions:

- Every Music group produces at least one album.
- Every approved album is sold to a distributor having at least 1 view.
- Panelists are the members of the McM company.
- Members finalize top 2n from shortlisted participants (Given).
- McM Director is a member of the McM Company.
- Each Distributor has a unique URL.
- Panel is an entity that consists of McM company members and their department is "panel".
- Each Music group has a specific music type (such as rock, jazz, pop, etc.).
- album_price is a derived attribute that is calculated from the # of likes,dislikes.
- Views may/may not come, so price may/may not be sold.
- Only those albums that are sold to the distributors can be downloaded.
- Album type can be audio, video or both.

- We are assuming that candidates are connected directly to the MCM company. (aggregation)

ER DIAGRAM:



note:
underline = Primary key
bold = foreign key

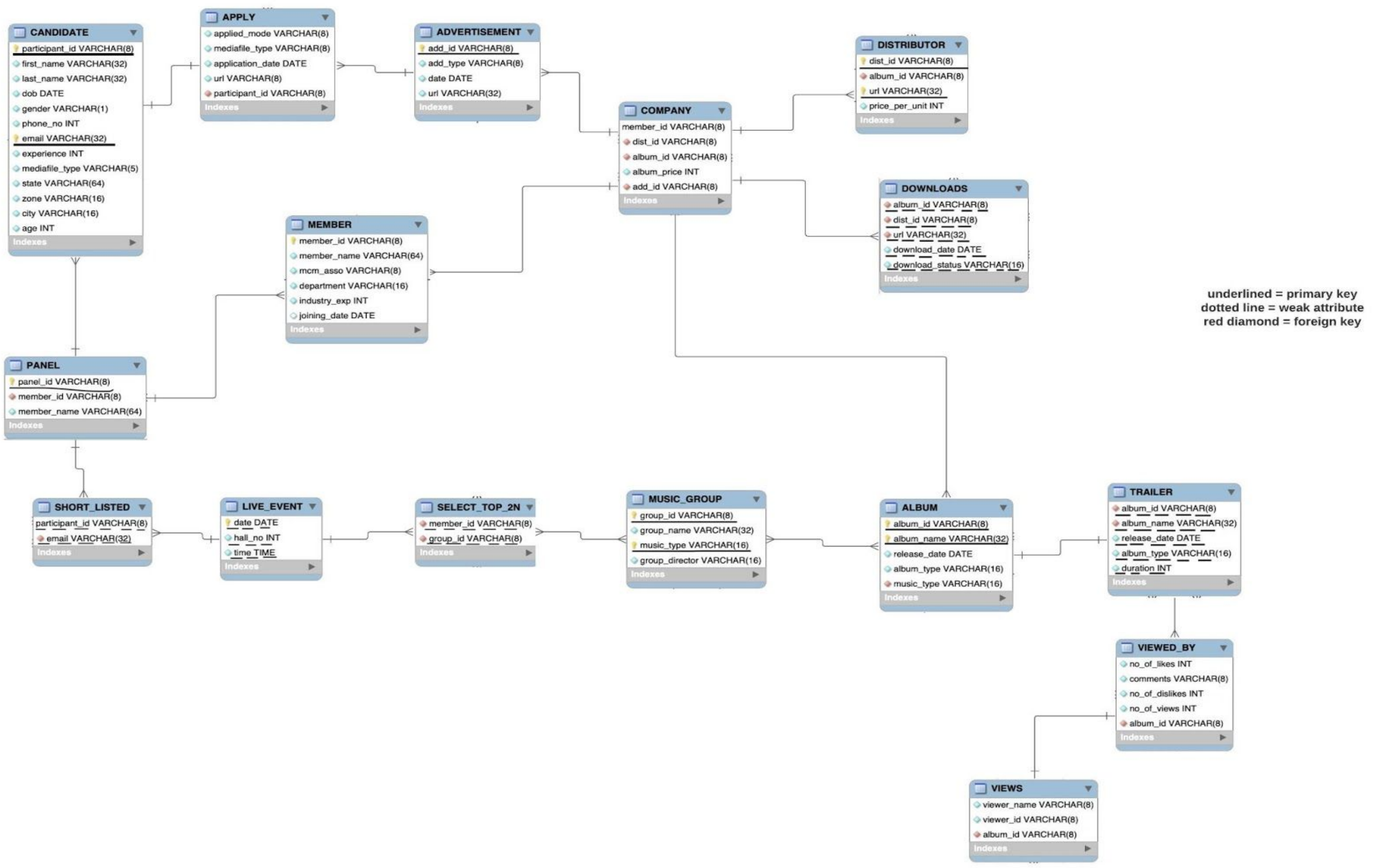
Link to ER DIAGRAM PDF:ER diagram in PDF

View Access and Permissions:

- Candidates have the view access to their form applied.
- Members can view who all are other members that are part of the company.
- Distributors can view albums information like who created it, views, etc.
- Company has all the information about downloads of albums and user information.
- Anyone can view the Trailer of albums and also view its statistics

Q2.

RELATIONAL SCHEMA DIAGRAM:



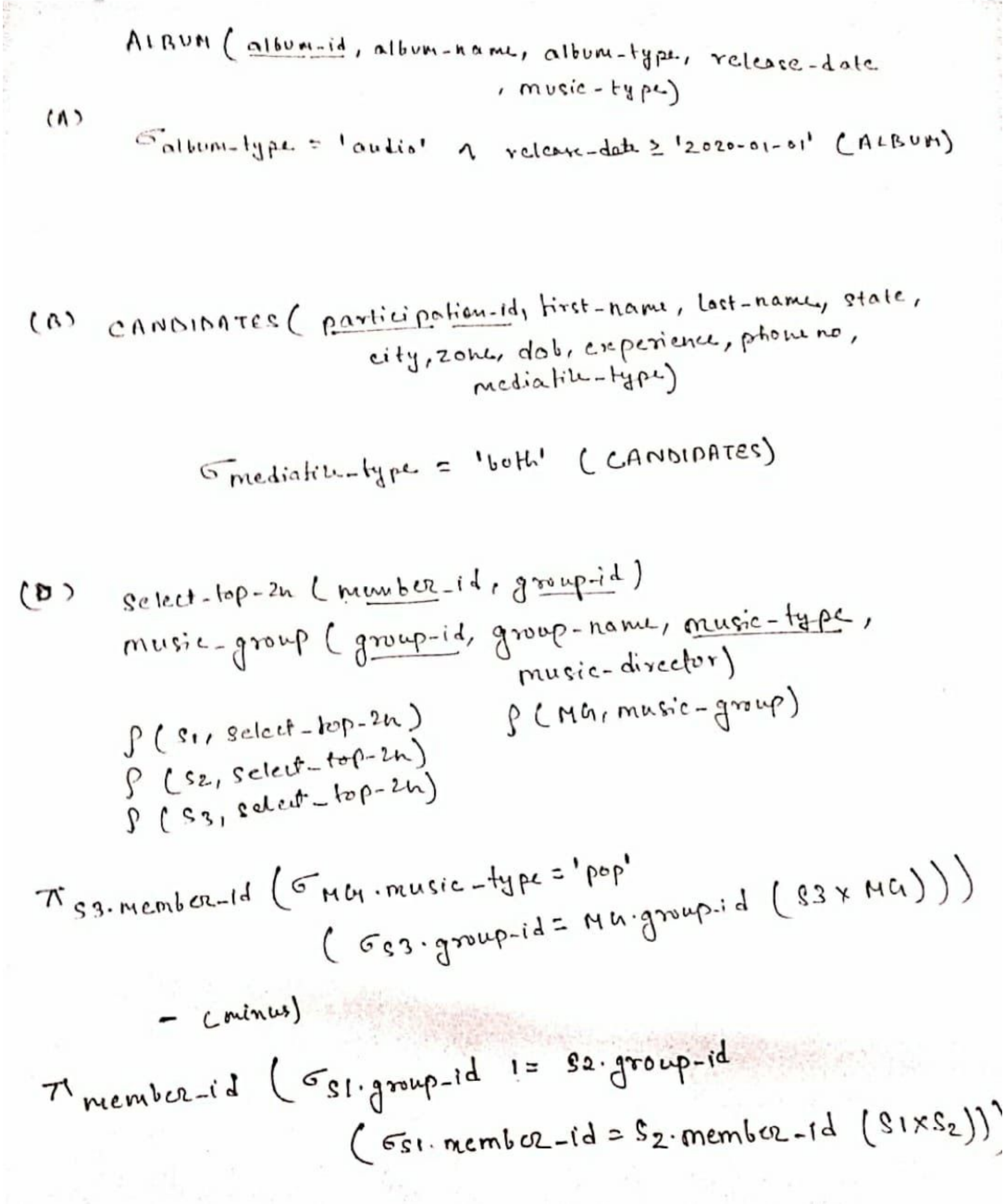
Link to RELATIONAL SCHEMA DIAGRAM PDF:relational schema in PDF

Q3.

We consider that our group has made a good database design because of the following points:-

- All the relevant information and data is stored in the database.
- The relationships between the entities are well defined and logical.
- Redundant data is not being stored in the database which will reduce the chance of integrity problems and increase the efficiency of the database. Application of concepts of specialization, weak entities, etc. will contribute to the reduction of redundancy in the database.
- All the entities are unambiguously defined as each entity has a unique key that defines it.
- Complex attributes are used to reduce ambiguity in the data of an entity.
- The database has a variety of attributes like simple, composite, multi-valued, derived etc.
- Weak entities and attributes are well defined in the ER diagram.
- Specialization is also used for inheritance in the database.
- Total participation and partial participation is also used between every entity to enforce well defined relationships between entities..
- Cardinality is used to make sure that the relation between two entities is unambiguous.

Q4.



ALBUM ($\underline{\text{album-id}}$, album-name, album-type, release-date, music-type)

(A)

$\sigma_{\text{album-type} = \text{'audio'} \wedge \text{release-date} \geq \text{'2020-01-01'}}$ (ALBUM)

(B) CANDIDATES ($\underline{\text{participation-id}}$, first-name, last-name, state, city, zone, dob, experience, phone no, mediatile-type)

$\sigma_{\text{mediatile-type} = \text{'both'}}$ (CANDIDATES)

(D) select-top-2n ($\underline{\text{member}}$-id, $\underline{\text{group-id}}$)

music-group ($\underline{\text{group-id}}$, group-name, $\underline{\text{music-type}}$, music-director)

ρ (S1, select-top-2n) ρ (MG, music-group)

ρ (S2, select-top-2n)

ρ (S3, select-top-2n)

$\pi_{\text{S3.member-id}}$ ($\sigma_{\text{MG.music-type} = \text{'pop'}}$ ($\sigma_{\text{S3.group-id} = \text{MG.group-id}}$ (S3 × MG)))

- (minus)

$\pi_{\text{member-id}}$ ($\sigma_{\text{S1.group-id} \mathrel{!=} \text{S2.group-id}}$ ($\sigma_{\text{S1.member-id} = \text{S2.member-id}}$ ($S1 \times S_2$)))

(c)

$\text{Select\_top-2n} \ (\underline{\text{member-id}}, \ \underline{\text{group-id}})$

$\rho \ (S1, \ \text{Select-top-2n})$

$\rho \ (S2, \ \text{select-top-2n})$

$$\pi_{\text{member-id}} \left(\sigma_{S1.\text{group-id} \ != \ S2.\text{group-id}} \left(\sigma_{S1.\text{member-id} = S2.\text{member-id}} \ (S_1 \times S_2) \right) \right)$$

(E)

$\text{DISTRIBUTOR} \ (\underline{\text{dist-id}}, \ \underline{\text{url}}, \ \text{album-id}, \ \text{price-per-unit})$

$\text{ALBUM} \ (\underline{\text{album-id}}, \ \text{album-name}, \ \underline{\text{album-type}}, \ \text{release-date}, \ \text{music-type})$

$\rho \ (D, \ \text{DISTRIBUTOR})$

$\rho \ (A, \ \text{ALBUM})$

$$\pi_{D.\text{dist-id}} \left(\left(\sigma_{A.\text{album-type} = \text{'both'}} \left(\sigma_{D.\text{album-id} = A.\text{album-id}} \ (D \times A) \right) \right) \right.$$

$$\cup$$

$$\left(\sigma_{A.\text{album-type} = \text{'audio'}} \left(\sigma_{D.\text{album-id} = A.\text{album-id}} \ (D \times A) \right) \right)$$

$$\wedge$$

$$\left. \left(\sigma_{A.\text{album-type} = \text{'video'}} \left(\sigma_{D.\text{album-id} = A.\text{album-id}} \ (D \times A) \right) \right) \right)$$

Q5.

A)
```
select * from ALBUM where release_date >= '2020-01-01' and album_type = 'audio';
```

B)
```
SELECT member_id as 'more_than_one' from select_top_2n group by member_id having count(member_id) >1;
```

C)
```
SELECT member_id from select_top_2n where group_id in (select group_id from music_group where music_type = 'pop') and member_id in (SELECT member_id from select_top_2n group by member_id having count(*) = 1);
```

D)
```
SELECT * from CANDIDATE where mediafile_type ='both';
```

E)
```
SELECT applied_mode as 'most effective',count(*) as 'entries' from APPLY group by applied_mode order by COUNT(*) desc LIMIT 1;
```

Q6.

A) To model this data, we have Zone, State, Age , Type_of_album as the attributes of each candidate which then gives aggregated information about candidates to the McM company. Age is a derived attribute that takes reference from DOB(date of birth). Zone and State are composite attributes of address. Now, the company can get aggregated information about total entries/submission as shown in below parts. This helps us in making simple SQL queries to understand as well as execute.

B) **<u>Store:</u>**

```
insert into `CANDIDATE`values('p11','ayush','arpit','2020-4-5','m',99999,'xxx1',10,'audio','Delhi','North','delhi',20);
insert into `CANDIDATE` values('p12','arham','ali','2018-4-5','m',99999,'xxx2',6,'audio','UP','North','lucknow',60);
insert into `CANDIDATE` values('p13','aryan','yadav','2017-4-5','m',9999,'xxx3',13,'audio','WB','East','kolkata',45);
insert into `CANDIDATE` values('p14','suraj','chauhan','2015-4-5','m',99999,'xxx4',9,'both','TN','South','chennai',34);
insert into `CANDIDATE` values('p15','gaurav','gupta','2019-3-5','m',94444,'yyy1',1,'both','AP','South','indore',27);
insert into `CANDIDATE`
        values('p16','abhish','sharma','2008-3-5','m',94444,'yyy2',4,'video','Gujrat','West','jaipur',19);
insert into `CANDIDATE`
        values('p17','manvik','arya','2011-1-9','m',9965,'zzz1','4','audio','Rajasthan','West','jaipur',28);
insert into `CANDIDATE`
        values('p18','vishal','yadav','2011-1-9','m',99446,'zzz2','9','audio','Rajasthan','West','jaipur',72);
insert into `CANDIDATE`
        values('p19','nitin','sharma','2011-1-9','m',99665,'zzz3','7','audio','Delhi','West','jaipur',24);
insert into `CANDIDATE`
        values('p20','akash','patel','2011-1-9','m',99244,'zzz4','3','audio','AP','West','jaipur',42);
```

Analyze :

For state

```
SELECT * FROM CANDIDATE where state= 'Delhi';
SELECT * FROM CANDIDATE where state= 'WB';
SELECT * FROM CANDIDATE where state= 'UP';
SELECT * FROM CANDIDATE where state= 'TN';
SELECT * FROM CANDIDATE where state= 'Gujarat';
```

For type of album

```
SELECT * FROM CANDIDATE where mediafile_type= 'both';
SELECT * FROM CANDIDATE where mediafile_type= 'audio';
SELECT * FROM CANDIDATE where mediafile_type= 'video';
```

For Age group

```
SELECT * FROM CANDIDATE where age >= 15 and age <= 20;
SELECT * FROM CANDIDATE where age >= 21 and age <= 25;
SELECT * FROM CANDIDATE where age >= 26 and age <= 30;
SELECT * FROM CANDIDATE where age >= 31 and age <= 40;
SELECT * FROM CANDIDATE where age >= 41 and age < 100;
```

C) For zone

```
select * FROM CANDIDATE where zone= 'North';
select * FROM CANDIDATE where zone= 'South';
select * FROM CANDIDATE where zone= 'East';
select * FROM CANDIDATE where zone= 'West';
```

Q7.

A. The given schema relation is a bad design because of the following reasons: -

- **Each tuple in a relation should represent one entity or relationship instance:** participant# should be a foreign key in panelist_album_evaluation so panel knows which particular candidate to evaluate.
- **Null values in tuple:** Participant already has a member# even before the evaluation. So, all of the values will be null in member# column until the shortlisted participants are declared.
- **Handling redundant information in tuples and updating anomalies Attributes of different entities (like Instructor, Department) should not be mixed in the same relation**: Album_Type is the same attribute in both participant and member_group_album_trailer. It isn't a primary or foriegn key.

The following anomalies were present in the given database:-

1. Updation Anomaly : This is the phenomenon when we want to update some data in the database but, in order to perform such change we would need to update all the data associated with it multiple times.
   For example: Suppose, a music group has ‘n’ members. This music group releases a brand new album then, the attributes in the entity **Member_Group_Album_Trailer** such as the Album#, Album_Name, Album_Type, Date_of_creation, Album_Description, Album_Approver#, Approval_Date, Trailer_Release_Date, Trailer_Release_URL, Incoming_URL_for_View, View_Date and Comments would need to updated ‘n’ number of times causing updation anomaly which could lead to data inconsistency if any of the row accidentally doesn’t get updated.

2. Deletion Anomaly : This is the phenomenon when some unwanted data has to be deleted but, it causes the deletion of some other data.
   For example: Suppose, a specific album’s information has to be deleted like album#, release_date, album_name etc. from the **Album_Distribution_and_Download** entity and this album is only being sold by only one distributor then, if we delete the album’s data, all the distributor information would get lost too along with the album’s info.

3. Insertion Anomaly : This is the phenomenon when certain data can’t be inserted into the database without the presence of some other data.
   For example: If a new distributor’s details like Distributor#, Distributor_Name, Distributor_Location etc. has to be added in the **Album_Distribution_and_Download** entity then, it could only be added if an album gets associated with this distributor.

B. 1st NF is given already.

For second NF,

Participant1(Member#, Name, Age, City, Phone, Email, Prior_Experience, Advt_Seen, Album_Type, Submission Date)
Participant2(File_Upload_Path, Status_Round1, Status_Round2)

Panelist_Album_Evalutaion1(Panelist#, Panelist_Name, Experience, Association_Month)
Panelist_Album_Evalutaion2(Panelist#, Association_Year, File_Upload_Path)

Member_Group_Album_Trailer1(Member#, Member_Name, Group#, Group_Name, Member-Role, Group_Music_Class, Album#, Album_Name, Album_Type, Date_of_Creation, Album_Description, Group_Leader#)
Member_Group_Album_Trailer2(Member#, Group_Leader_Name, Album_Approver#, Approval_Date, Trailer_Release_Date, Trailer_Release_URL, Incoming_URL_for_View, View_Date, Comments)

Album_Distribution_and_Download1(Album#, Album_Release_Date, Distributor#, Distributor_Name, Distributor_Location, Price, Download#)
Album_Distribution_and_Download2(Incoming_URL_for_Download,Download_Request_Date, Downloaded_Album#, Download_Status)

For third NF,

Participant1(Member#, Name, Age, City, Phone, Email, Prior_Experience)
Participant2(Member#,Advt_Seen, Album_Type, Submission Date)
Participant3(File_Upload_Path, Status_Round1, Status_Round2)

Panelist_Album_Evalutaion1(Panelist#, Panelist_Name, Experience)
Panelist_Album_Evalutaion2(Panelist#, Association_Month)
Panelist_Album_Evalutaion3(Panelist#, Association_Year, File_Upload_Path)

Member_Group_Album_Trailer1(Member#, Member_Name, Group#, Group_Name, Member-Role, Group_Music_Class)
Member_Group_Album_Trailer2(Member#,Album#, Album_Name, Album_Type, Date_of_Creation, Album_Description, Group_Leader#)
Member_Group_Album_Trailer3(Member#, Group_Leader_Name, Album_Approver#, Approval_Date, Trailer_Release_Date, Trailer_Release_URL, Incoming_URL_for_View, View_Date, Comments)
Member_Group_Album_Trailer4(Member#, Trailer_Release_URL, Incoming_URL_for_View, View_Date, Comments)

Album_Distribution_and_Download1(Album#, Album_Release_Date, Distributor#, Distributor_Name, Distributor_Location, Price, Download#)
Album_Distribution_and_Download2(Downloaded_Album#, Download_Status)
Album_Distribution_and_Download3(Incoming_URL_for_Download,Download_Request_Date)

For BCNF NF,

Participant1(Member#, Phone, Email, Prior_Experience)
Participant2(Member#,Advt_Seen, Album_Type, Submission Date)
Participant3(File_Upload_Path, Status_Round2)
Participant4(Member#, Name, Age, City)
Participant5(File_Upload_Path, Status_Round1)

Panelist_Album_Evalutaion1(Panelist#, Experience)
Panelist_Album_Evalutaion2(Panelist#, Association_Month)
Panelist_Album_Evalutaion3(Panelist#, File_Upload_Path)
Panelist_Album_Evalutaion4(Panelist#, Association_Year)
Panelist_Album_Evalutaion5(Panelist#, Panelist_Name)

Member_Group_Album_Trailer1(Member#, Member_Name, Group#, Group_Name, Member-Role, Group_Music_Class)
Member_Group_Album_Trailer2(Member#,Date_of_Creation, Album_Description, Group_Leader#)
Member_Group_Album_Trailer3(Member#, Group_Leader_Name, Album_Approver#, Approval_Date, Trailer_Release_Date, Trailer_Release_URL, Incoming_URL_for_View, View_Date, Comments)
Member_Group_Album_Trailer4(Member#, View_Date, Comments)
Member_Group_Album_Trailer5(Member#,Album#, Album_Name, Album_Type)
Member_Group_Album_Trailer6(Member#, Trailer_Release_URL, Incoming_URL_for_View)

Album_Distribution_and_Download1(Album#, Distributor_Location, Price, Download#)
Album_Distribution_and_Download2(Downloaded_Album#, Download_Status)
Album_Distribution_and_Download3(Album#,Incoming_URL_for_Download,Download_Request_Date)
Album_Distribution_and_Download4(Album#, Album_Release_Date, Distributor#, Distributor_Name)
Album_Distribution_and_Download5(Album#,Incoming_URL_for_Download)

C. Download# in Album_Distribution_and_Download relation a surrogate key because there is no primary key. Also, we have to maintain a record of every download for which we would need a unique value. a surrogate key is required here as an automatically generated primary key by the system itself for every download, and users cannot define it .