**TV show Case Study**

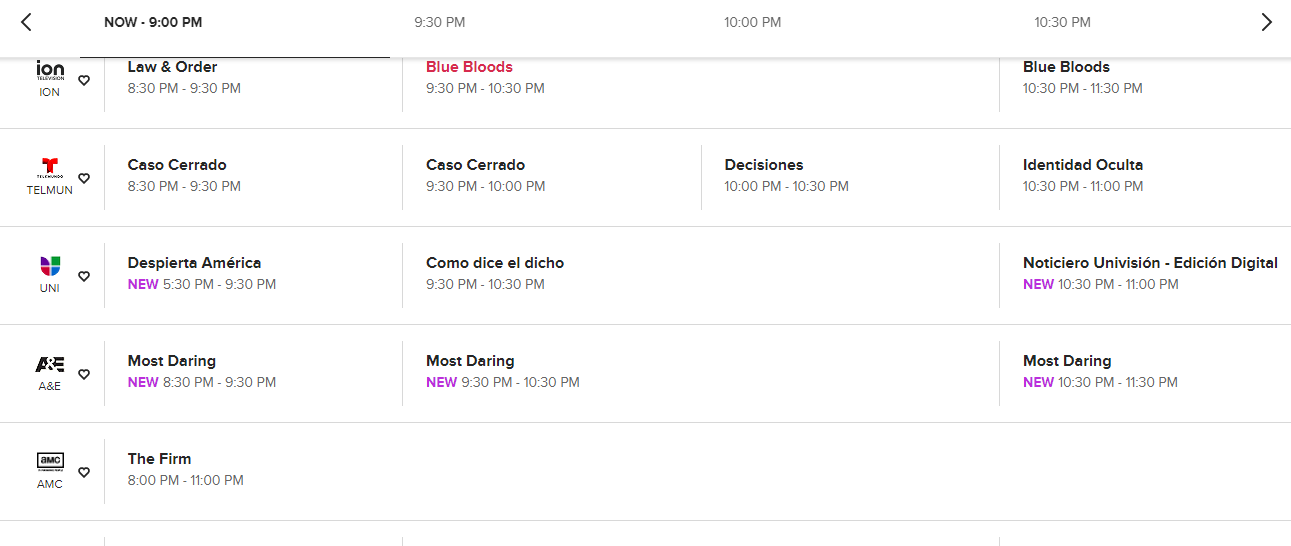
**INTRODUCTION**

In this world of data, it important to also generate metadata. Metadata is basically information about data (viz-data of data). In this case study we are going to talk about the metadata for various TV SHOWS.

A Tv Show consist of various components as the name of program being aired, on the channel in which it is going to air, on what date and time it is going to be aired etc so where will we get this information from? The answer might vary for various domains as for some programs we might get it from our partner channels, from some it might be from network operators whereas for some it might be Internet.

Now, after we have got all our component, the pattern won’t be the same for all as shown below

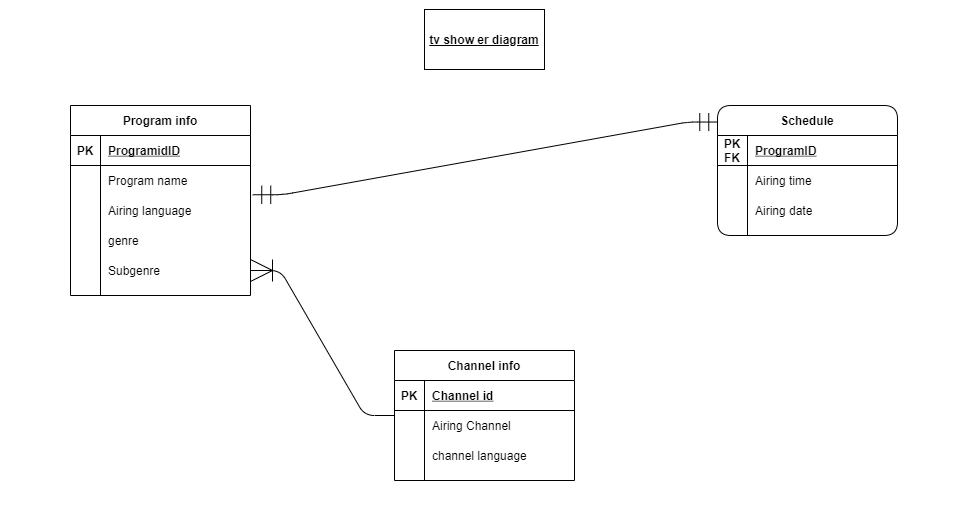


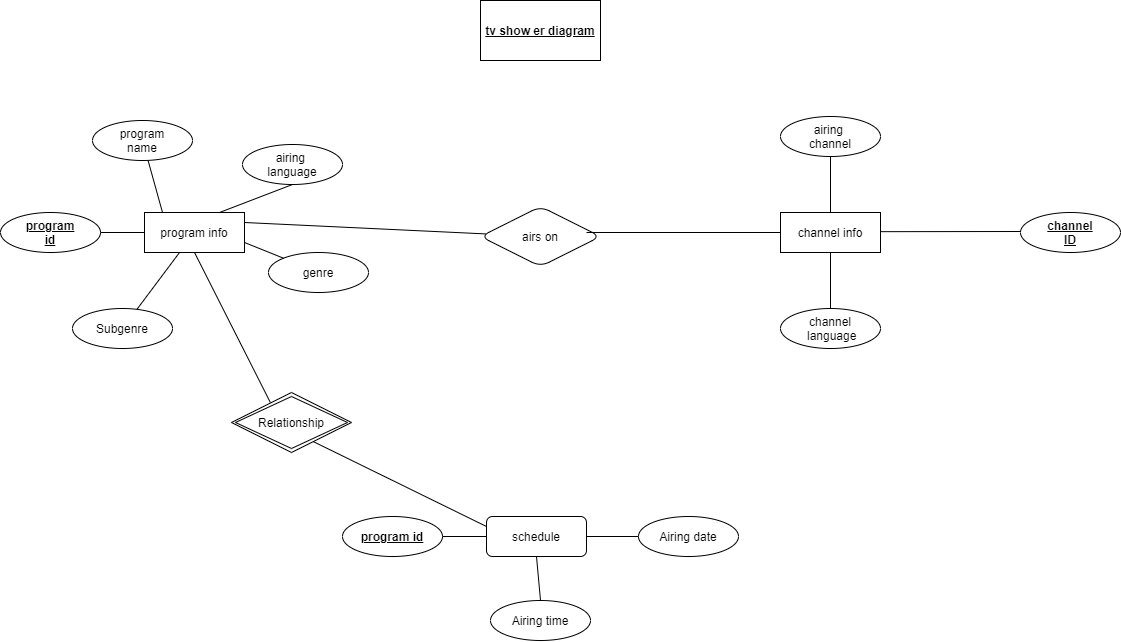


So now let us make use of wonderful concept of Rdbms which consist of various tables with a systematic arrangement of data with various er diagrams and concepts.

**E-R diagrams**

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research. Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes.





Above is the Er diagram from tv show which represents various attribute and their relationship which each other. In our case program info consist of 4 attributes and as program id as its primary key whereas channel info consist of 3 attributes and has channel id as its primary key but in schedule module we have programid which is a foreign key been borrowed by program info hence we can say schedule is a week entity.

Relation

Program info – Schedule (one to one)

As only one program can be scheduled at a time so the relation will be one to one.

Program info – channel info(many to one)

As many programs can be aired on one channel relation here is many to one.

**Normalisation**

The data added to the table may consist of many duplication and arrangment might not be at its best so now lets normalise the data. Normalization is the process of reorganizing data in a database so that it meets two basic requirements i.e There is no redundancy of data, all data is stored in only one place. Data dependencies are logical ,all related data items are stored together.The main purpose of normalisation is to avoid analomies.

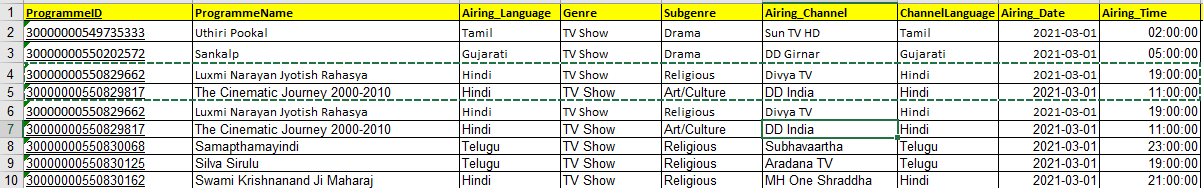
**Analomies-**

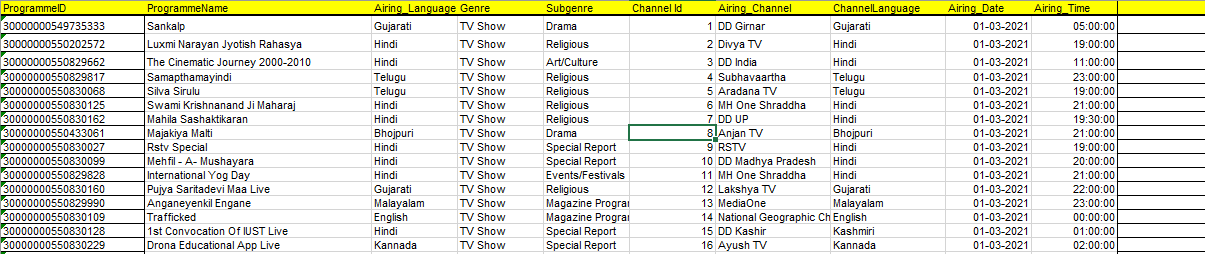
Analomies can occur in three form i.e insertion,deletion and update analomies.

Insertion analomy - It leads when a user is not able to insert data due to havoc created in table by adding multiple data into a single cell and no specification of data whereas in normalised database a single cell consist of a single data which makes data insertion easy for a user

Deletion analomy - It may occur when a user has yo delete particular data he will have to visit various tables to delete particular data whereas in normalised database we can delete a data on single table and that data will be deleted from all other tables as well.

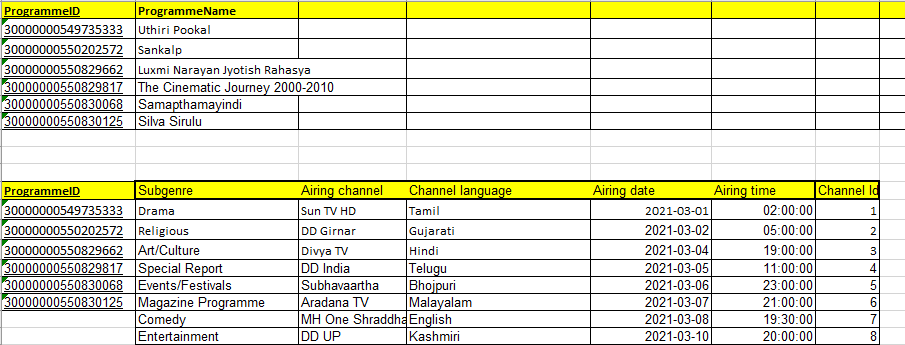
Update analomy- It may occur when a user wants to update data. Here, a user may be trying to delete or insert data as well but he might not be able to update data as updating data may not update precise information from all the data whereas in normalised data a user can easily update data without affecting other tables.





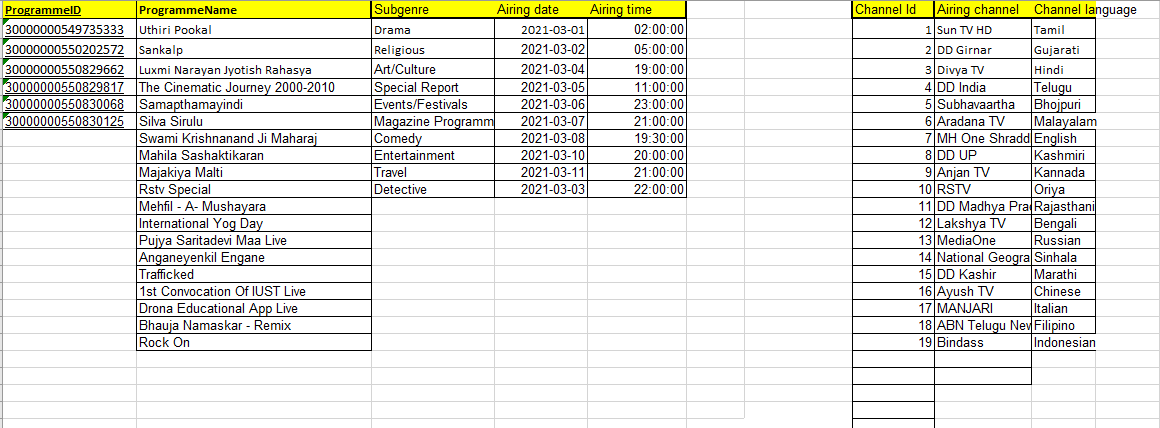
In above table we can see that there is data duplication resulting in failure of assigning primary key so we need to bring this table in 1nf

1nf-



The above table is in 1nf as there is no repetitive data and each cell consist of one and only one data. So we can assign programmeid as our primary key but we can see that the data is not fully dependant on programmeid and it may lead in insertion analomy so further moving to 2nf

2nf-



Here we can see table is in 2nf as it satisfy criteria of 1nf and schedule table is fully dependant on its primary key channel id and there is no partial dependency. But as we see the program info data airing date and airing time is dependants of program name whereas program name is depended on programmeid so further bringing table into 3nf

3nf-

In above table we can see the table is in 3nf as it satisfies the condition of 2nf and there is no transitive dependency because all the attribute of programme info is fully dependant on programmeid also, other tables are fully dependant on their primary keys channelid and programmeid.

**Data Dictionary**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Data type** | **description** | **examples** |
| ProgrammeID | bigint | Unique id for all tv shows | 30000000550829662 |
| ProgrammeName | Varchar | Name of tv show | Uthiri Pookal |
| Airing\_Language | Varchar | Language of tv show | Gujarati |
| Genre | Varchar | Type of genre being used | Tvshow,movie |
| Subgenre | Varchar | Type of genre used in tvshow | drama |
| Airing\_Channel | Varchar | Name of channel | DD Girnar |
| ChannelLanguage | Varchar | Language used in channel | Marathi |
| Airing\_Date | Date | Date on which program will air | 01-03-2021 |
| Airing\_Time | Time | Time on which program airs | 19:00:00 |

Permissions

