PROJECT MOVIE LIBRARY

Coding Ninjas

Table: Movie_Library

Customer_name	Title	Address	Movie_rented
Lokesh Daga	Mr.	403-B Morwadi Nagar, Nokha, Rajasthan	Race 2, Radhe, Bharat
Neelabh Shukla	Mr.	419-M Kota Nagar, Nokha, Rajasthan	Daddy's little girls
Lokesh Daga	Mr.	403-B Ashok Nagar, Bikaner, Rajasthan	The Notebook
Rashi Sharma	Mrs.	109 Ram Nagar, Napasar, Rajasthan	Fanna, The Notebook

Q1. Identify the basic issue with the table currently and convert it into 1NF normal form and explain your reasons for doing this. Also show how the new table looks ?

Ans. The attribute Address is a composite valued attribute, A table containing one field for an address would not be in 1NF because it stores the street address, city and state. Three different bits of information in one field and the attribute Movie_rented is a multivalued attribute. So both attributes violate 1-NF Normalization of the table Movie_Library. Also every table in a relational database should have a primary key, which is a unique identifier for each record in the table. The first normal form requires that each row in a table should be uniquely identifiable, and having a primary key is one way to achieve this. Without a primary key, it would be difficult to identify and update specific records in the table, which could lead to data inconsistencies and other issues.

To fix the attribute address we can Separate that into Street address , City , State and add a primary key Customer_id. After that the table will have these much attributes

Movie_Library

Customer_id
Customer_name
Title
Street_address
City
State
Movie_rented

After the normalization in the address attribute the table would be like:

Customer_id	Customer_Name	Title	Street_Address	City	State	Movie_rented
C1	Lokesh Daga	Mr.	403-B Morwadi Nagar	Nokha	Rajasthan	Race 2, Radhe, Bharat
C2	Neelabh Shukla	Mr.	419-M Kota Nagar	Nokha	Rajasthan	Daddy's little girls
C3	Lokesh Daga	Mr.	403-B Ashok Nagar	Bikaner	Rajasthan	The Notebook
C4	Rashi Sharma	Mrs.	109 Ram Nagar	Napasar	Rajasthan	Fanna, The Notebook

Now there are two more attributes that still violate 1-NF. The State attribute contains all the columns with the same value. The first normal form requires that each column in a table must contain atomic values, which means that each value in a column should be indivisible. Having a column with the same value for all rows violates this rule because it means that the column contains redundant information, and it could be removed from the table without losing any information. Also the attribute movie_rented as discussed contains multiple values. If a column contains multiple values for a single instance of an entity or a record, it violates this rule. In other words, a column in a table should only contain one value for each record. So we will seperate those two attributes to a different table

The States table would be

Table: States

State_id	State_Name
S1	Rajasthan

Since we move movies to another table it will be the new library to hold the movie details. So Movies Library table would look like this

Table: Movie_Library

Movie_id	rent_id	Movie
L1	C1	Race 2
L2	C1	Radhe
L3	C1	Bharat
L4	C2	Daddy's little girls
L5	C3	The Notebook
L6	C4	Fanna
L7	C4	The Notebook

And then holds the customer details. So the Customers table would be

Table: Customers

Customer_id	Customer_name	Title	Street_Address	City
C1	Lokesh Daga	Mr.	403-B Morwadi Nagar	Nokha
C2	Neelabh Shukla	Mr.	419-M Kota Nagar	Nokha
C3	Lokesh Daga	Mr.	403-B Ashok Nagar	Bikaner
C4	Rashi Sharma	Mrs.	109 Ram Nagar	Napasar

Q2. Identify the keys that uniquely identify the tuples in the table you formulated in Q1. ?

Ans. Primary keys are used to uniquely identify tuples or records in a table. A primary key is a column or a set of columns that uniquely identifies each record in the table.

- Movie_library has Movie_id as Primary Key
- States has State_id as Primary Key
- Customers has Customer_id as Primary Key

Q3. Convert the table you formulated into its next normal form (2NF) and state your reason for doing so and also create the new tables.

Ans. To determine whether a table is in second normal form (2NF), you need to check two things:

1. The table must be in first normal form (1NF). This means that each cell in the table should contain only atomic values and there should be no repeating groups or arrays.
2. There should be no partial dependencies. This means that every non-key attribute (i.e., any attribute that is not part of the primary key) should depend on the entire primary key and not on only a part of it.

As from Q1 we have made all the tables in first normal form. Now we need to check for partial dependencies in all the tables

The table Movie_Library has

- Movie_id(Primary Key)
- Rent_id
- Movie

As every non- key attribute depends on the primary key. ie, Movie_id -> Rent_id and Movie_id ->Movie, the table follows second normal form

The table Customers has

- Customer_id(Primary Key)
- Customer_name
- Title
- Street_address
- City

Here all the non-key attributes(Customer_name , Title , Street_address , City) are dependent on the entire Primary Key(Customer_id)

The table States has

- State_id(Primary Key)
- State_Name

And the non_key attribute State_Name is dependent on Primary key State_id. ie, State_id -> State_Name.

Since all the tables don't have any partial dependencies and follow the first normal form(1-NF) we can say that the database follows the second normal form(2-NF).

Q4. Now suppose someone wants to add new movies like The Jungle Book, Fast and Furious 9 into the db , which are not yet rented to anyone, can we do that?

Ans. We can add the new movie details into the movie library table and since the movies aren't rented to anyone the rent_id would remain NULL. So the table after inserting the values would be

Table: Movie_Library

Movie_id	rent_id	Movie
L1	C1	Race 2
L2	C1	Radhe
L3	C1	Bharat
L4	C2	Daddy's little girls
L5	C3	The Notebook
L6	C4	Fanna
L7	C4	The Notebook
L8	NULL	The Jungle Book
L9	NULL	Fast and Furious 9

Q5. Convert the tables you have formulated after converting to 2NF into its next possible Normal Form and state your reasons for doing so.

Ans. It seems that there is a transitive dependency between the Title and Customer Name attributes, as both of them depend on the Customer ID, which is the primary key. This means that the Title attribute is indirectly dependent on the primary key through the Customer Name attribute, which violates the third normal form.

To bring the table to 3NF, we need to remove the transitive dependency. One way to do this is to split the table into two tables:

Table 1: Customers

- Customer ID (primary key)
- Customer Name

Table 2: Customer Details

- Customer ID (foreign key)
- Title
- Street Address
- City

Table: Customers

Customer_id	Customer_name
C1	Lokesh Daga
C2	Neelabh Shukla
C3	Lokesh Daga
C4	Rashi Sharma

Table: Customer_Details

Customer_id	Title	Street_Address	City
C1	Mr.	403-B Morwadi Nagar	Nokha
C2	Mr.	419-M Kota Nagar	Nokha
C3	Mr.	403-B Ashok Nagar	Bikaner
C4	Mrs.	109 Ram Nagar	Napasar

Q6. Convert the tables you have formulated in the previous question into its next possible Normal Form and state your reasons for doing so.

Ans. To determine whether a table is in Boyce-Codd Normal Form (BCNF), you need to follow these three steps:

1. Check if the table has a composite primary key:
   BCNF requires that each non-trivial functional dependency in the table must have a determinant that is a candidate key. So, if the table has a composite primary key, then you need to check if all non-trivial functional dependencies in the table have the entire composite key as their determinant. If not, then the table is not in BCNF.
2. Check for any non-trivial dependencies on non-prime attributes:
   If the table has a single-column primary key, or if all functional dependencies have the entire primary key as their determinant, then you need to check for any non-trivial dependencies on non-prime attributes. If there are any such dependencies, then the table is not in BCNF.
3. Decompose the table if necessary:
   .

Since the tables created follow all the conditions the database follows Boycee Codd Normal form.
The final tables in the Movie Library database are as follows

Table: Customers

Customer_id	Customer_name
C1	Lokesh Daga
C2	Neelabh Shukla
C3	Lokesh Daga
C4	Rashi Sharma

Table: Customer_Details

Customer_id	Title	Street_Address	City
C1	Mr.	403-B Morwadi Nagar	Nokha
C2	Mr.	419-M Kota Nagar	Nokha
C3	Mr.	403-B Ashok Nagar	Bikaner
C4	Mrs.	109 Ram Nagar	Napasar

Table: Movie_Library

Movie_id	rent_id	Movie
L1	C1	Race 2
L2	C1	Radhe
L3	C1	Bharat
L4	C2	Daddy's little girls
L5	C3	The Notebook
L6	C4	Fanna
L7	C4	The Notebook
L8	NULL	The Jungle Book
L9	NULL	Fast and Furious 9

Table: States

State_id	State_Name
S1	Rajasthan