# Project Movie Library

**Q. 1 Identify the issues with the table currently and convert it into 1 NF normal form and explain your reasons for doing this. Also show how the new table looks.**

Currently, the address attribute is composite, and movie rented attribute is multi-valued. This will increase complexity of basic queries such as filtering/grouping records based on state, city or movies rented.

Hence, address attribute should be broken into multiple attributes like address, city and state, while there should be different row for each movie rented

| customer_name | title | Address | City | State | Movies_Rented |
|---|---|---|---|---|---|
| Lokesh Daga | Mr. | 403-B Marwadi Nagar | Nokha | Rajasthan | Race 2 |
| Lokesh Daga | Mr. | 403-B Marwadi Nagar | Nokha | Rajasthan | Radhe |
| Lokesh Daga | Mr. | 403-B Marwadi Nagar | Nokha | Rajasthan | Bharat |
| Neelabh Shukla | Mr. | 419-M Kota Nagar | Nokha | Rajasthan | Daddy's Little Girls |
| Lokesh Daga | Mr. | 57 Ashok Nagar | Bikaner | Rajasthan | The Notebook |
| Rashi Sharma | Ms. | 109 Ram Nagar | Napasar | Rajasthan | Fanna |
| Rashi Sharma | Ms. | 109 Ram Nagar | Napasar | Rajasthan | The Notebook |

Here, we will make customer name and address as our composite key

**Q. 2 Identify the keys that uniquely identify the tuples in the table you formulated in Q1.**

In the above table, no attribute can uniquely identify a row by itself. Hence, we need a composite key. Customer name and address combined, can uniquely identify each row as two customers can have the same name but not the same address.

Hence, customer name, Address, city and state will be our super key, while customer name and address will be both our candidate and primary key.

**Q. 3 Convert the table you formulated into the next normal form (2 NF) and state your reasons for doing so and create the new tables.**

As 2 NF requires that there be no partial dependencies, the city, state and title columns need to be removed as the first two are interdependent only with address while title depends only on the customer's name.

Hence, we should add the title to the name itself and remove the column and create a separate table for address with a numerical (surrogate key) primary key and add that key to the customer table as a foreign key. Hence, we will have the following two tables:

1. Customer: customer name, address id, movies rented
2. Address: address id, address, city, state

The Address table has a synthetic key as multiple cities can have addresses with the same name and multiple states can have cities with the same name. Hence, it is better to keep a surrogate key

**Q. 4 Now suppose someone wants to add new movies like The Jungle Book, Fast and Furious 9 into the DB, which are not yet rented by anyone, can we do that?**

No, as rented movies must have a corresponding customer in our current schema otherwise, we will end up adding null values for all the other attributes. Moreover, we already have a composite key (customer name, address id), which means we cannot have null values in them even if we tried.

Hence, to do this task, we need a separate table named Movies which will keep a list of all the available movies. As movies and customers share a "many to many" relationship, we will need a separate joining table named "Movies_Rented" where movie name, customer and address will be referenced from their respective table and act as a composite key in this table. Hence, we will now have the following four tables:

1. Customer: customer name, address id
2. Address: address id, address, city, state
3. Movies: movie
4. Movies_Rented: movie, customer name, address id

**Q. 5 Convert the tables you have formulated after converting to 2 NF into its next possible Normal Form and state your reasons for doing so.**

3 NF states that there must not be any transitive dependencies i.e., a non-key attribute must not determine any other non-key attribute.

In the above tables, the customer table has only one non-key attribute while in Address, address, city and state combined form the candidate key. If we consider the Movie table, it has only one attribute while the movies_rented table has no non-key attribute either.

Hence, our tables are already in 3 NF

**Q. 6 Convert the tables you have formulated in the previous question into the next possible Normal Form and state your reasons for doing so.**

The next normal form BCNF, states that for every nontrivial functional dependency A -> B, A must be a super key, i.e. A cannot be a non-key

In the schema proposed in Q3, we have only one non-key attribute, which does not determine any attribute, while the schema proposed in Q4 has no non-key attributes at all

Hence, our tables are already in Boyce Codd Normal Form (BCNF).