

Original Data - Unnormalized Movie Rental Store Scenario

customer name	address	email	phone 1 (land)	phone 2 (cell)	movies rented	category
Bob Beltcher	12-94 Ocean Avenue	bob@burgers.com	980-463-9966		Good Fellas	Drama
Homer Simpson	742 Evergreen Terrace	hsimpson@radioactive.com	635-962-1365	635-962-1366	Alita: Battle Angel, Batman	Fantasy, Action
Jessica Jones	123 Broadway	jj@superdetective.com	212-165-8865		The Big Sick	Romance

For movies rented and category, notice that the category corresponds to the movie in order i.e. Alita is fantasy and

1st Normal Form (1NF)

Add comments and tables (with data)

Customer Table

customer id	customer name	address	email
1	Bob Beltcher	12-94 Ocean Avenue	bob@burgers.com
2	Homer Simpson	742 Evergreen Terrace	hsimpson@radioactive.com
3	Jessica Jones	123 Broadway	jj@superdetective.com

Contact Table

customer id	phone number	phone type
1	980-463-9966	land line
2	635-962-1365	land line
2	635-962-1366	cell
3	212-165-8865	land line

Movie Rented Table

customer id	movie id	movies rented	category
1	12	Good Fellas	Drama
2	13	Alita: Battle Angel	Fantasy
2	14	Batman	Action
3	15	The Big Sick	Romance

For the first normal form I created three tables : Customer Table, Contact Table, and Movie Rented Table. The Customer Table contains "customer id", "name", "address", and "email". Since these values are atomic and have no repeating attributes, they remain together in a table containing customer info. I created Contact Table to eliminate the repeating attributes phone 1(land) and phone 2(cell). This table lists "phone numbers" as a column and "phone type" as a column. This means there is only one column containing phone numbers while the other column identifies whether the number is a land line or cell number. The key for this table is a composite key (customer id and phone number) since the primary key for the first table is customer id (no duplicate keys allowed). Finally, I created a Movie Rented Table to make sure the values for the movies rented column are atomic. It lists "customer id", "movie id", "movies rented", and "category" as columns, which allows for each movie rented to be listed in a separate row with its respective movie genre. The primary key for this table was a composite key(customer id and movie id) to identify who rented what movie.

2nd Normal Form (2NF)

Add comments and tables (with data)

Customer Table

customer id	customer name	address	email
1	Bob Beltcher	12-94 Ocean Avenue	bob@burgers.com
2	Homer Simpson	742 Evergreen Terrace	hsimpson@radioactive.com
3	Jessica Jones	123 Broadway	jj@superdetective.com

Contact Table

customer id	phone number	phone type
1	980-463-9966	land line
2	635-962-1365	land line
2	635-962-1366	cell
3	212-165-8865	land line

Movie Rented Table

customer id	movie id
1	12
2	13
2	14
3	15

Movie Table

movie id	movies	category
12	Good Fellas	Drama
13	Alita: Battle Angel	Fantasy
14	Batman	Action
15	The Big Sick	Romance

I based my 2NF off of my 1NF tables since 2NF has to be in 1NF and also every non-key attribute must be fully dependent on the key. My Customer Table satisfies these requirements since “customer name”, “address”, and “email” all depend on the unique customer id (primary key). The Contact Table also satisfies these requirements since phone type depends on both the customer id and phone number. A customer can have multiple phone numbers and each phone number needs to be matched to a Customer to identify if it’s their home phone or land line. However, the Movie Rented Table had to be split since the attribute “category” was only dependent on “movie id” and not the full primary key. I created Movie Rented Table to identify who rented what movie with the columns “customer id” and “movie id” (making up the key and whole table). The Movie Table contained the attributes “movie id”, “movies”, and “category” which satisfied the 2NF requirement since both “movies” and “categories” fully depend on the key “movie id”.

3rd Normal Form (3NF)

Add comments and tables (with data)

Customer Table

<u>customer id</u>	customer name	address	email
1	Bob Beltcher	12-94 Ocean Avenue	bob@burgers.com
2	Homer Simpson	742 Evergreen Ter	hsimpson@radioactive.com
3	Jessica Jones	123 Broadway	jj@superdetective.com

Contact Table

<u>customer id</u>	<u>phone number</u>	phone type
1	980-463-9966	land line
2	635-962-1365	land line
2	635-962-1366	cell
3	212-165-8865	land line

Movie Rented Table

<u>customer id</u>	<u>movie id</u>
1	12
2	13
2	14
3	15

Movie Table

<u>movie id</u>	movies	category
12	Good Fellas	Drama
13	Alita: Battle Angel	Fantasy
14	Batman	Action
15	The Big Sick	Romance

My 3NF Tables are the same as my 2NF Tables since every non-key attribute in each table is non-transitively dependent on their respective keys. For the Movie Rented Table, the attributes in the table make up the composite key so there are no non-key attributes. The Customer Table’s attributes are only dependent on the key because there can be many customers with the same name, many customers living at the same address (a family), and people with the same email address (because emails are recycled over time, I googled this since at first I felt the email should go in a separate table). Therefore, there is no unique identifier for these attributes other than the “customer id” (primary key). “Phone type” attribute only has one choice to reference in the table and that is the primary key so it satisfies the 3NF requirement. Finally, The Movie Table contains attributes “movies” and “category” which only depend on the key “movie id” since there are movies with the same name (remakes) and movies with the same category.