

Normalisation steps:

Step-1:

Initially, columns of our table contain multiple values. For 1NF domain of each column should contain atomic(indivisible) values. Hence after 1NF, the domain of each attribute is made atomic even though few values are getting repeated. Using the First Normal Form, data redundancy increases, as there will be many columns with the same data in multiple rows but each row as a whole will be unique.

Step-2:

Primary key for the table obtained after 1NF is (Client_Id, Client_Phone_Number, Boat_Number, From_Date, Spots) and there are partial some partial dependencies.

If we consider Client_Name, city, country and DOB they only depend on Client_id which is a part of primary key. To remove this partial dependency Client_info table is created to store only client details.

Similarly, if we consider boat details such as boat_capacity, service type, owner_id etc they depend only on boat_id which is a part of primary key and hence to remove this partial dependency, boat table is created to store details of boat.

At this stage, all entities are in 2NF.

Step-3:

Primary key for Boat Table after 2NF is BoatNumber but boat has services such as Bar, Cuisine, Wifi, Rate where they depend on Service_type of the boat, which is a non-prime attribute that gives transitive dependency. To remove this transitive dependency a separate table is created for Services provided by the boat which includes boat number and facilities.

Similarly, Owner details depend on Owner id which is a non-prime attribute. To remove this transitive dependency a separate table is created to store owner details.

At this stage, all entities are in 3NF.

Step-4:

Redundancy caused by multi-valued attributes such as client_phone_number and sightseeing spots in each route can be removed by creating separate tables for phone numbers and sightseeing spots.

Now, all entities are in 4NF.

Relational schema after 1NF:

HouseBoat(client_id, client_phone_number, boat_number, from_date, spots, client_first_name, client_last_name, client_city, client_country, client_dob, boat_capacity, service_type, owner_id, wifi_bandwidth, bar, cuisine, rate, owner_first_name, owner_last_name, license_no, owner_dob, to_date, route_id, source, destination, spots, payment_type, payment_status, amount)

Relational schema after 2NF:

Client(client_id, client_first_name, client_last_name, client_city, client_country, client_dob)

Phone_numbers(client_id, client_phone_number) (AFTER 4NF)

Boat(boat_number, boat_capacity, service_type, owner_id, wifi_bandwidth, bar, cuisine, rate, owner_first_name, owner_last_name, license_no, owner_dob)

Booking(client_id, boat_number, from_date, to_date, route_id, payment_type, payment_status, amount)

Route(route_id, source, destination)

Sightseeing Spot(route_id, spots) (AFTER 4NF)

Relational schema after 3NF:

Client(**client_id**, client_first_name, client_last_name, client_city, client_country, client_dob)

Phone_numbers(**client_id, client_phone_number**) (AFTER 4NF)

Boat(**boat_number**, boat_capacity, service_type, owner_id)

Services(**service_type**, wifi_bandwidth, bar, cuisine, rate)

Owner(**owner_id**, owner_first_name, owner_last_name, license_no, owner_dob)

Booking(**client_id, boat_number, from_date**, to_date, route_id, payment_type, payment_status, amount)

Route(**route_id**, source, destination)

Sightseeing Spot(**route_id, spots**) (AFTER 4NF)