DATABASE DESIGN

PROJECT REPORT

CAR RENTAL SYSTEM

TEAM MEMBERS:

Israt Jarin – 18103167

Nishat Tasnim – 19203058

Md Mehedi Hasan – 19303001

Fabiha Khan – 19303004

Oyaon Sultan – 20303003

INDEX1. Introduction

2. Functionalities of the system  
3. Entities   
4. Relationships, Cardinality Ratio, and Participation  
5. ER Diagram   
6. Possible Data Fields  
7. Normalization Of the Data Fields  
8. Tables and Fields with Data Values  
9. SQL Statements   
10. Conclusion

**1.INTRODUCTION**

A car rental management system is a self-contained system that represents the best opportunity for people who cannot afford to buy a car for their family. Under this system, you can rent a car for a certain period and continue your main business while paying the rental fee.

people need to move to another location for official business, domestic, or other important reasons. For long-distance travel, a car is required as it is convenient apart from other public transport options. That's why people prefer cars over public transport.

Public transport is not very flexible and unstable like trains, buses, etc. Therefore, people are choosing cars instead of other options. With the advancement of technology and the desire to go to different places away from the place of residence, every family needs transportation.

The proposed system offers opportunities for middle-class people who don't know how to drive but can't afford a car. The system offers driver options while minimizing equipment charges. Depending on the number of people who want to trek in one car, users can rent a car according to their wishes and budget.

Car rental companies typically rent vehicles and other motor vehicles for an hourly or daily fee. For these reasons, the car rental industry is growing and most car rental companies now have several branches in different cities. That`s why car rental companies need well-designed and intuitive systems to manage their employees, vehicles, and drivers. The project includes a car rental system that can be used by small, medium or large car rental companies

In our system, customers can rent a car depending on the make and model. Customers can purchase optional car rental insurance and use up to one coupon on their final bill.

**2. Functionalities of the system:**

a) The car rental company must have a collection of cars.

b) Each car belongs to a specific car category and each car’s special place.

c) Customers rent cars based on location and vehicle category preferences.

d) Based on his location and vehicle category preferences, a list of available cars for rent is created. Along with available dates and times (start and end dates).

e) The customer should be able to select a car from the offer and reserve it for rental.

f) If a customer reserves a car, it must be an option to purchase a car. Must be an insurance plan and be able to apply at most one discount code.

g) If you are also a member of a car rental company and have a membership card Standard his 10% discount is applied on top of any applied discount code. Therefore, rate according to the applied discount code the total discount rate is 10 plus the specified discount rate.

h) Claims will be made upon return of the vehicle.

i) Customer may return before or on the due date or return.

It's too late.

j) If the customer returns the car after the due date, additional late fees will be charged and added.

on the bill.

k) Standard tax of 8.25% will be applied to the amount. This includes late charges and this. Tax is added to get the discounted total amount and the final amount

receive the amount.

l) Once the car has been returned, it will be available for booking.

m) Appointments can be canceled up to five days before his actual pickup.

n) A company may have multiple discount plans, Weekend discounts, corporate discounts, etc.

o) Car prices are calculated based on the make and model selected.

**3. ENTITIES**

**a) Admin:**

Administrators are responsible for the overall development of the system by managing it efficiently and monitoring any errors that occur in the system. Administrators also manage accounts for other actors and update account data. Since administrators may have multiple people, the system separates accounts for each administrative unit.

**b) User:**

The customer will be the person using the car rental system to book a car. He can be a system member or a non-system member. System members will have Member ID. The customer organization will store details such as the customer's driver's license number, e-mail, address, name, and phone number.

**c) Cars:**

The car entity will have a list of cars available in the system. Each vehicle will be associated with an auto and car categories will have attributes like make, model, mileage, and registration Numbers. Vehicles will also have their flag to check vehicle availability.

**d) Car Category:**

Each vehicle has a vehicle type. Vehicle prices are calculated according to the type of vehicle. Car catalog will have attributes like number of people, number of bags, name, cost per day, and late rate per hour.

**e) Availability:**

Cars are available or not that user ad admin can see by availability

**f): Reservation:**

Each reservation will be tracked in the entity called booking. Pre-order will have attributes such as car ID, booking date and time, return date and time and the actual date and time of the return of the reservation, and the status of the reservation. Book this place This amount may also include rental insurance and a discount code.

**g) Payment:**

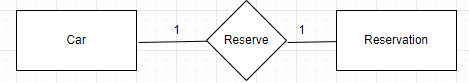
After booking a car for a certain period, also select a private driver option and an estimated cost is displayed on the customer's screen to be paid by the customer at the time of delivery. In addition, the customer must pay the advance amount to make the reservation through the system in the online mode for the administrator. This will help to book a car for a specific customer.

**h) Driver:**

Customers can choose to rent a car with a chauffeur or self-drive option depending on their use.

Driver fees are an additional charge on top of renting a car, so it's up to you whether you need to spend more money on a driver or you can drive yourself to your desired destination.

**4. Relationship and Cardinality Ration:**



**Relationship:**

**Left To Right**

Car reserve reservation.

**Right To Left**

Reservation reserved by car.

**Cardinality Ratio:**

**Left To Right**

1 car reserve 1 reservation.

**Right To Left**

1 Reservation is reserved by 1 car.

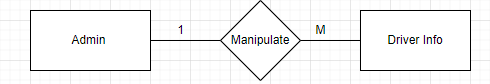
**Participation:**

**Left To Right**

Car may/may not reserve a reservation.

**Right To Left**

Reservation may/may not be reserved by car.



**Relationship:**

**Left To Right**

Admin Manipulate Driver Info.

**Right To Left**

Driver Info manipulated by Admin.

**Cardinality Ratio:**

**Left To Right**

1 admin manipulates Many driver info.

**Right To Left**

1 driver’s info was manipulated by 1 admin.

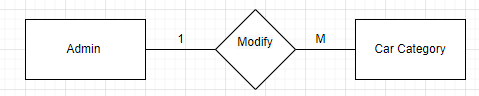
**Participation:**

**Left To Right**

Admin may/may not manipulate driver info.

**Right To Left**

Driver info may/may not be manipulated by the admin.



**Relationship:**

**Left To Right**

Admin Modify Car Category.

**Right To Left**

Car Category modified by Admin.

**Cardinality Ratio:**

**Left To Right**

1 admin modify many car categories.

**Right To Left**

1 car category modified 1 admin.

**Participation:**

**Left To Right**

Admin may/may not modify car category.

**Right To Left**

Car category may/may not be modified by the admin.



**Relationship:**

**Left To Right**

Admin visit user.

**Right To Left**

User visited by admin.

**Cardinality Ratio:**

**Left To Right**

1 Admin visit 1 user.

**Right To Left**

1 user visited by 1 admin.

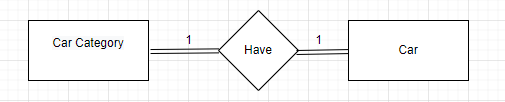
**Participation:**

**Left To Right**

Admin must visit user.

**Right To Left**

The user must be visited by Admin.



**Relationship:**

**Left To Right**

Car Category has a car.

**Right To Left**

The car has a car category.

**Cardinality Ratio:**

**Left To Right**

1 car category has 1 car.

**Right To Left**

1 car has 1 car category.

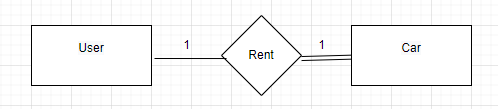
**Participation:**

**Left To Right**

The car category must be a car.

**Right To Left**

The car must have a car category.

 **Relationship:**

**Left To Right**

User rent Car.

**Right To Left**

Car rent by the user.

**Cardinality Ratio:**

**Left To Right**

1 user rents 1 car.

**Right To Left**

1 car rent 1 user.

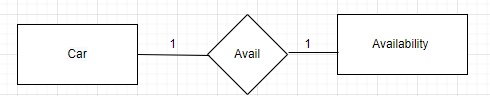
**Participation:**

**Left To Right**

The user may/may not rent a car.

**Right To Left**

The car must be sent by the user.



**Relationship:**

**Left To Right**

Car avail Availability.

**Right To Left**

Availability availed car.

**Cardinality Ratio:**

**Left To Right**

1 car avail 1 availability.

**Right To Left**

1 availability availed 1 car.

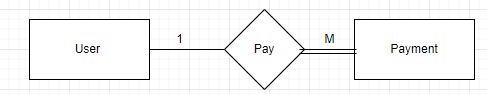
**Participation:**

**Left To Right**

Car may/may not avail available.

**Right To Left**

Availability may/may not avail car.



**Relationship:**

**Left To Right**

User pays payment.

**Right To Left**

Payment paid by the user.

**Cardinality Ratio:**

**Left To Right**

1 user pays Much payment.

**Right To Left**

1 payment is paid by 1 user.

**Participation:**

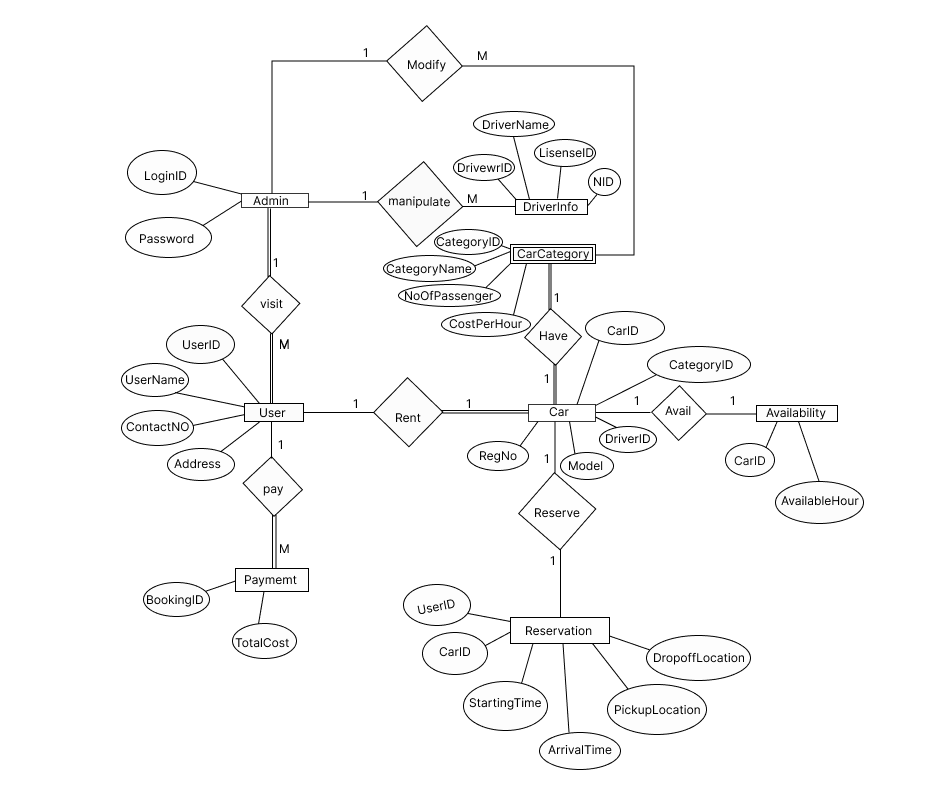
**Left To Right**

The user may/may not pay the payment.

**Right To Left**

payment must be paid by the user.

**6. ER Diagram**



**7. Possible data fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UserID | Username | ContactNo | Address | LoginID | Password | CarID |
| Model | RegNo | CategoryID | CategoryName | No\_of\_passenger | Costperhour | latefeeperhour |
| AvailableHour | Startingtime | EndingTime | Pickuplocation | Dropofflocation | BookingID | Totalcost |
| DriverID | DriverName | NID | LicenceNo |  |  |  |

**8. Normalization**

**1NF:**

**Step1:** Eliminating the repeating groups.

**Step2**: Identifying the primary key.

UserID, LoginID, CarID, CategoryID, BookingID, DriverID

**Step3**: Identifying all dependencies.

UserID (Username, ContactNo, Address) [Partial key dependency]

LoginID Password [Partial key dependency]

CarID (Model, RegNo) [Partial key dependency]

CategoryID (CategoryName, No\_of\_passenger, Costperhour,

Latefeeperhour) [Partial key dependency]

CarID AvailableHour [Partial key dependency]

BookingID (Startingtime, EndingTime, Pickuplocation, Dropofflocation)

[Partial key dependency]

DriverID (DriverName,NID,LicenceNo) [Partial key dependency]

Startingtime, EndingTime,Pickuplocation, Dropofflocation Totalcost

[Transitivekeydependency]

**2NF:**

**Step1:** Eliminating Partial key dependency

|  |  |  |  |
| --- | --- | --- | --- |
| UserID | Username | ContactNo | Address |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CarID | CategoryID | DriverID | Model | RegNo |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CategoryID | CategoryName | No\_of\_passenger | Costperhour | Latefeeperhour |

|  |  |
| --- | --- |
| CarID | AvailableHour |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| BookingID | UserID | CarID | Startingtime | EndingTime | Pickuplocation | Dropofflocation |

|  |  |  |  |
| --- | --- | --- | --- |
| DriverID | DriverName | NID | LicenceNo |

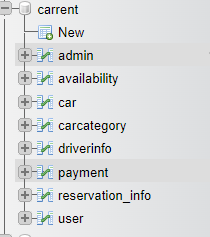
**3NF:**

**Step1:** Eliminating Transitive key dependency

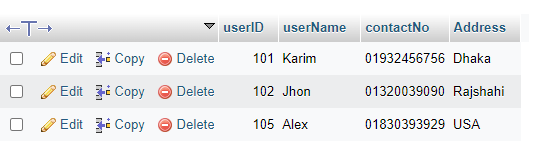
|  |  |
| --- | --- |
| BookingID | Totalcost |

**9. Tables and Fields with Data Values**

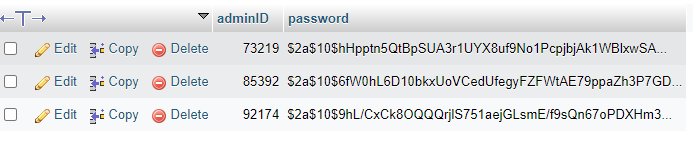
In our project, all tables are created after the normalization



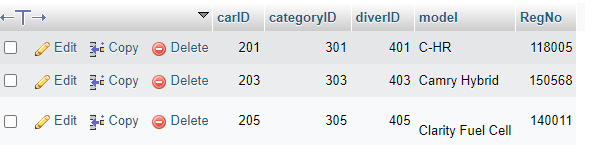
**User table**



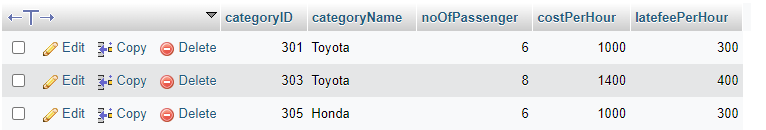
**Admin**



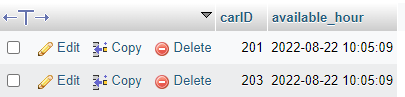
**Car**



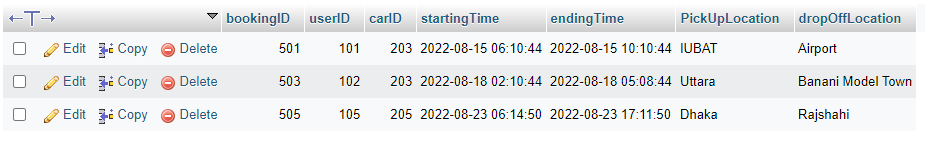
**Car Category**

****

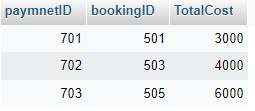
**Availability**

****

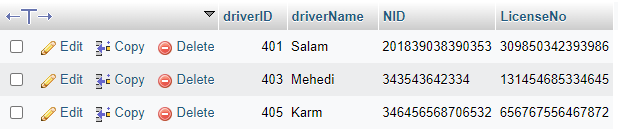
**Reservation\_info**

****

**Payment info**

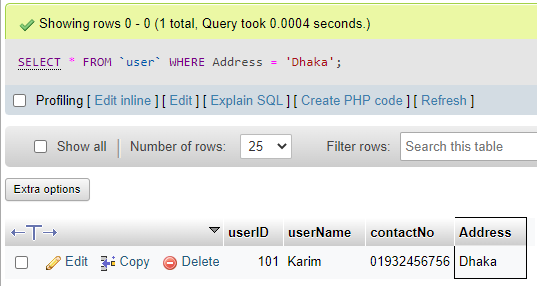
****

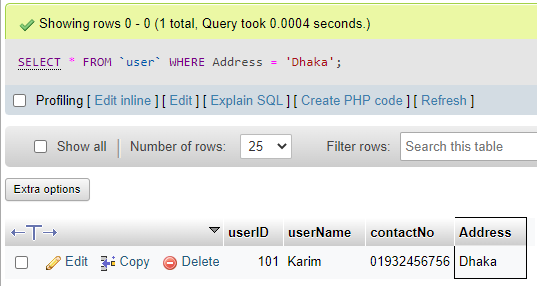
**Driver Info**

****

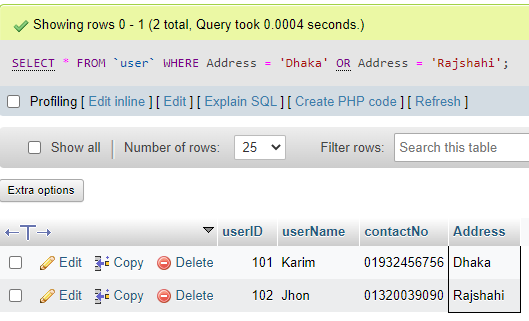
**10. SQL Query: Application**

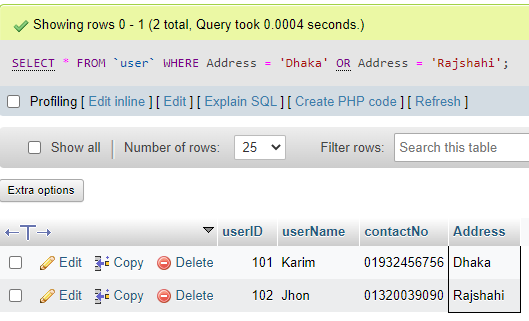
**1)** We want to find users who are from Dhaka

****

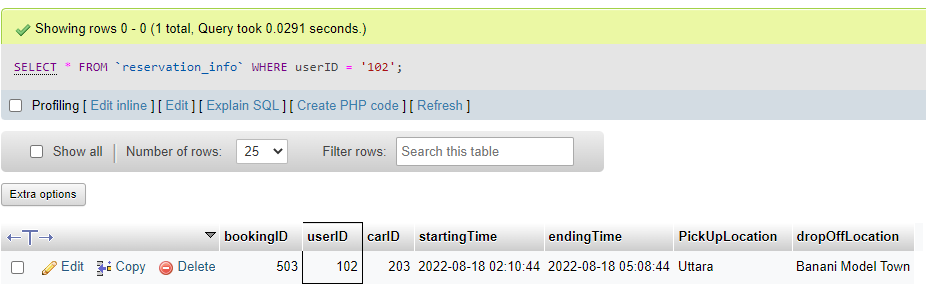
****

**2)** We are going to find out some users register from Dhaka **OR** Rajshahi

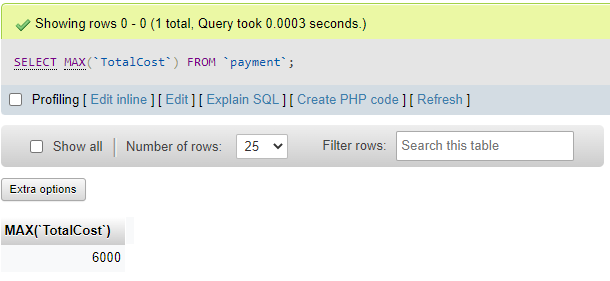
****

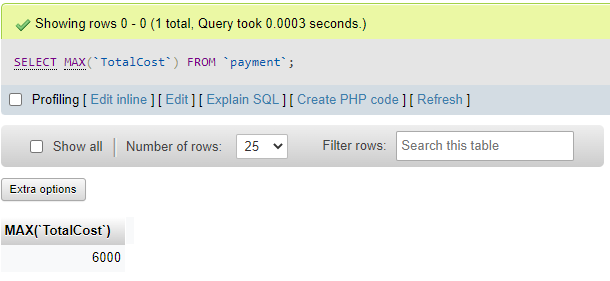
****

**3)** we want to see **user 102** give any booking or not

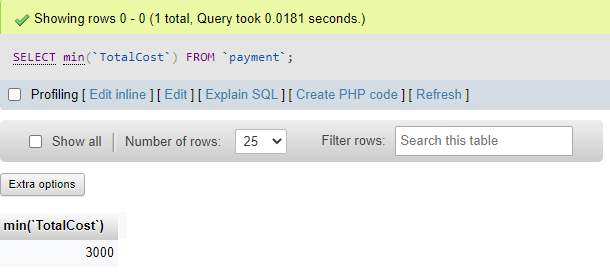
****

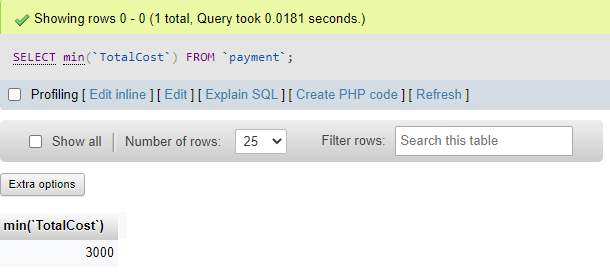
**4) Maximum** cost paid by user or customer we want to know the **amount**

****

****

**5) Minimum** cost pay by user or customer we want to know that **amount**

****

****

**6)** As a customer or user want to see **number of set** capability of a car and how much we have to spend for that **lower** range to **higher** range **cost per hour**

****

****

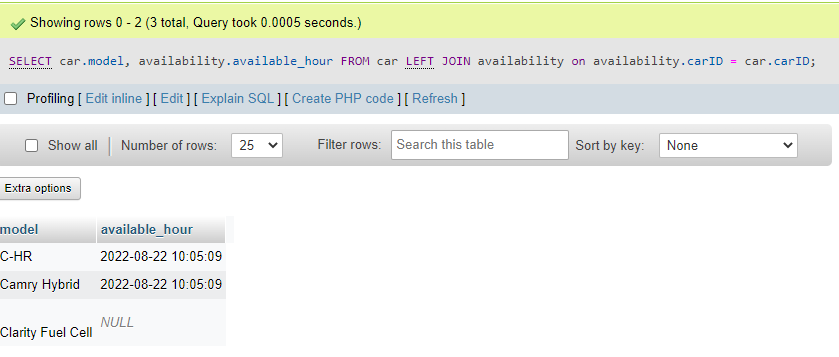
**7**) we are going to use **inner join** to get information from both tables depending on the common field of both tables. In our case, we want to see that the particular **user pay** how much cost to **rent a car** for his or her **journey**

**Reservation\_info** inner join **payment**

****

**8)** we are going to use the **left join** to get information from both tables depend on the common field of both tables and the left table full elements. In our case, we want to see which model of car is available and which care is not available right now

**Car** left join **availability**



**Conclusion:**

In comparison to previous experiences, when every activity related to the vehicle rental business was restricted to a physical place alone, the car rental industry has emerged with new delicacies. Even if the physical location has not been eliminated, the internet's power has altered the nature of functions and how these tasks are accomplished. Customers may now book vehicles online, rent automobiles online, and have the car delivered to their home if they are registered members, or they can travel to the office to pick up the car.

The data can then be easily accessed, managed, modified, updated, controlled, and organized. Most databases use structured query language (SQL) for writing and querying data. But Before Creating a database it is important to design a database first. There are four phases for designing a database. And, we have followed all of them to create the database.

**Requirement specification and analysis:**

requirements specification and analysis identify, analyze, and model the functionality or “what's” of a prospective software system. The requirements specification and analysis phase of a software project is the most important. We started our work by analyzing our system. we find out the problem and opportunities of the system. We collect all the features required.

**Conceptual design:**

It is an early phase of the design process, in which the broad outlines of function and form of something are articulated. It includes the design of interactions, experiences, processes, and strategies. We identify entities that will interact with the system. Then Find out the relation between entities’ cardinality ratio and participation constants. Then we use those Components to create ERD.

**Logical design:**

A logical design is a conceptual, abstract design. We did not deal with the physical implementation details yet; we deal only with defining the types of information that you need. The process of logical design involves arranging data into a series of logical relationships called entities and attributes. So we collect the list of fields and normalized them. We Normalization the field for meeting two basic requirements:

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.

After the normalization, we have a list of tables.

**Physical design:**

Physical design is the process of turning a design into manufacturable geometries. We create the database and fill dummy topples to the table and run SQL queries (Where, AND, OR, Order by, Min)