

1. Project Title

Car Parking Management System

2. Team

- Saif Ali (MySQL Expert)
- Karen Yao (MS SQL Server Expert)

3. Weekly Meeting Hours

We will meet and work on the project every Wednesday from 10:30 am to 12:30 pm.

4. Project Description

Consider a big company with several car parking lots on its premises. We create a database for the management and ticketing of these car parking lots.

The data requirements are as follows:

- A parking lot is identified by a unique id and can have one or more blocks/ parking grounds.
- A block has many parking stalls and a block code.
- A stall has a number, size(small cars/large cars), and type (for handicaps, for bicycles, reserved for police...)
- A customer can have weekly, monthly, or yearly passes (these passes are known as regular passes).
- If a customer does not have a pass, they can pay on a daily basis for which vehicle's license plate number and duration of parking are required.
- A regular pass has its total cost and duration of the pass.
- For the daily customer, a receipt is provided which shows the cost, duration, and license plate of the parked vehicle.
- If a car is parked in a stall without a valid pass, a ticket is issued for the car.
- The ticket includes the date, time, info about the car, stall number, fine amount, and the employee ID of the person who issued the ticket

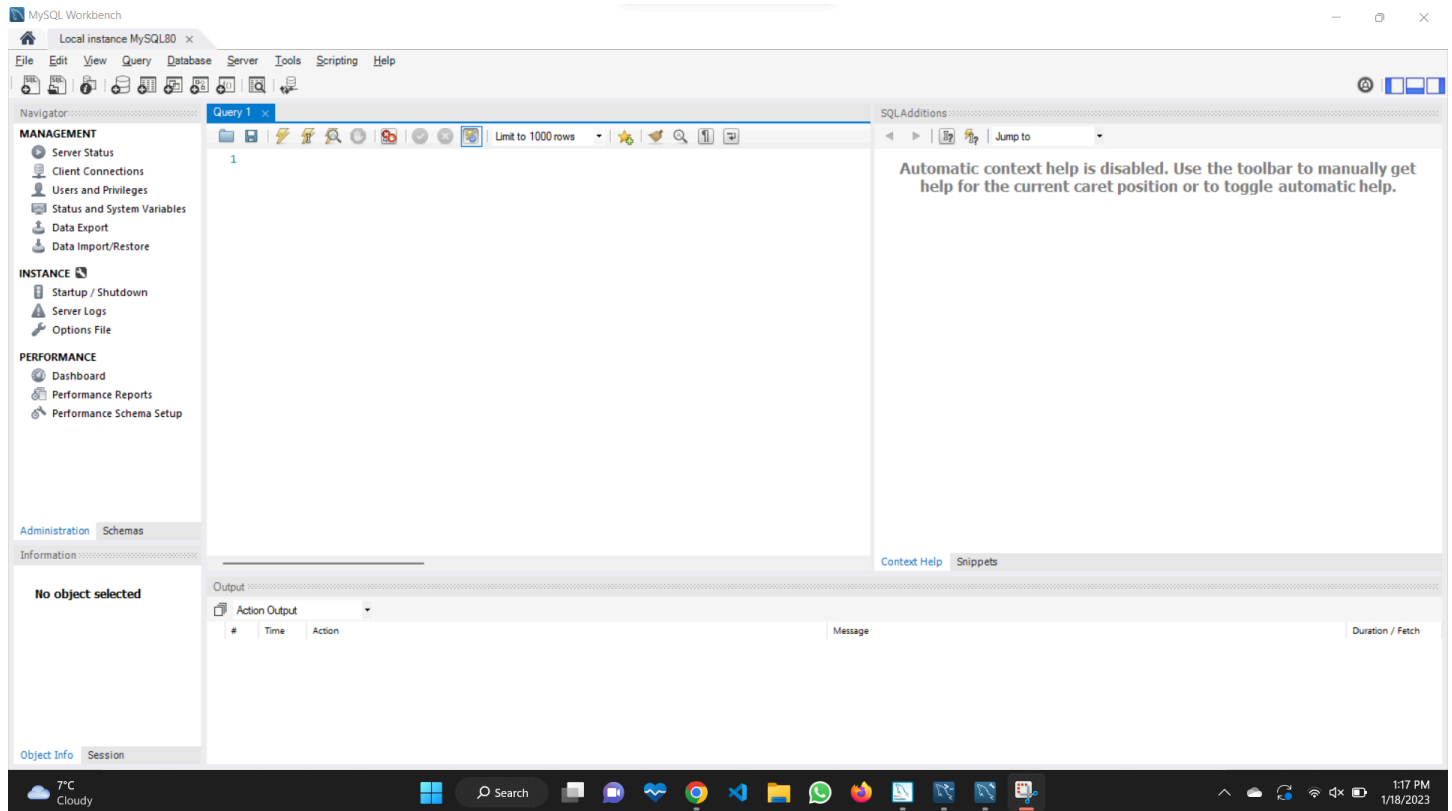
Entities: Parking lot, blocks, stall, customer, pass, tickets, car, receipt

5. Assumptions about Cardinality and Participation

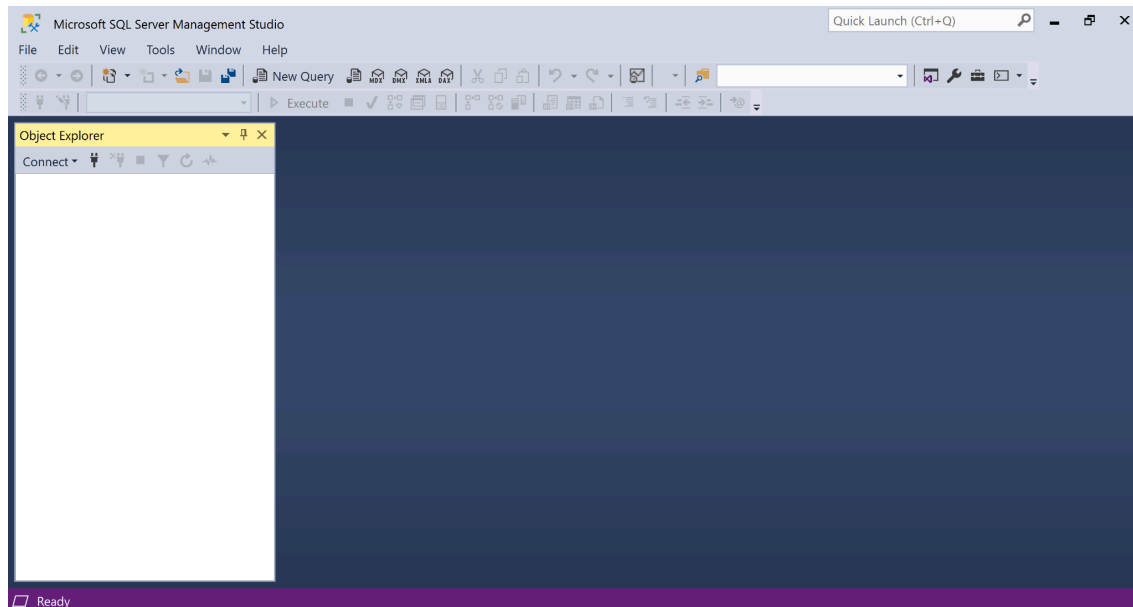
You can write all the assumptions about Cardinality and Participation (total/partial) here.

- Every Parking Lot has a total participation and have multiple Parking Blocks, but every parking block has a total participation and must have one parking stall
- Every Parking Block has a total participation and must have multiple Parking Stall, but every stall must have a total participation and have one parking block
- Each stall has a partial participation and may have parking stalls
- Each car has a partial participation and may park in multiple stalls
- Each car has a partial participation and may have multiple tickets
- Each ticket is associated with only one car
- Each Employee can issue multiple tickets, but each ticket must be only issued by one employee
- Each vehicle can have multiple passes, but each pass is only associated with one car
- Each pass must come with a receipt, and each receipt must come with a pass
- Each ticket is only associated with one stall, each stall may have multiple tickets associated to it

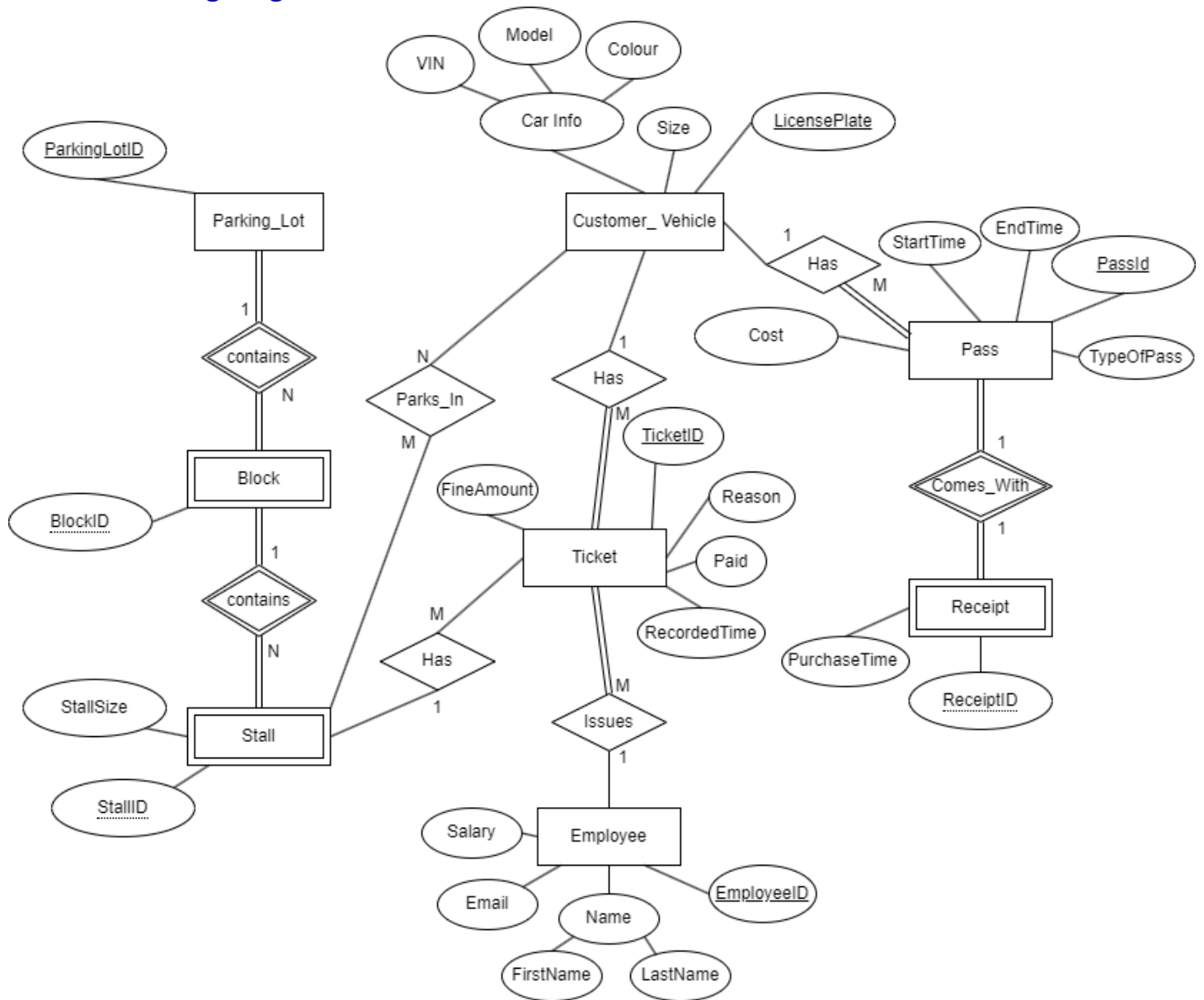
MySQL install screenshot



MS SQL install screenshot



6. EER Modeling Diagram



7. ER-Model Mapping to Database Relational Schema

The relational Schema is written here

Parking_Lot(ParkingLotID)

Parking_Block(ParkingLotID, BlockID)

Parking_Stall(ParkingLotID, BlockID, StallID, StallSize, StallNumber)

Customer_Vehicle(LicensePlate, VIN, Model, Color, Size)

Customer_ParksIn_Stall((ParkingLotID, BlockID, StallID), LicensePlate)

Pass(PassID, Cost, TypeOfPass, StartTime, EndTime, LicensePlate)

Receipt(PassID, ReceiptID, PurchaseTime)

Employee(EmployeeID, FirstName, LastName, Email, Salary)

Ticket(TicketID, Reason, FineAmount, RecordedTime, Paid, LicensePlate, EmployeeID, (ParkingLotID, BlockID, StallID))

8. Normalization

All relations are normalized up to First Normal Form since there are no multi-valued attributes.

All relations are normalized up to Second Normal Form since there are no strong entities using composite keys, therefore there are no partial functional dependencies. There are however composite keys in the Parking_Block, Parking_Stall, and Receipt weak entities. StallSize and StallNumber can only be identified using the full “**ParkingLotID, BlockID, StallID**” key and for the receipt weak entity, the date and time of the issuing of the receipt can only be determined by using the full “**PassID, ReceiptID**” key.

All relations are normalized up to Third Normal Form; it satisfies both First and Second forms and there are no transitive functional dependencies. No attributes can be obtained by attributes, only keys for each relation. For example, in the relation Pass. Cost cannot determine StartTime, only the PassID.

All relations are normalized up to BCNF since no non-prime attributes can determine any part of the keys. For example, VIN, Model, and Color cannot determine the License Plate.

9. Determining Data Types (Domain) and Constraints

You explain why you choose a certain data type for a field and why you apply certain constraints:

ParkingLotID INT -> Number ID

BlockID INT -> Number ID

StallID INT -> Number ID

StallSize CHAR(1) -> 'L' for Large Cars, 'S' for Small Cars

StallNumber INT -> Number ID

LicensePlate VARCHAR(7)-> License Plate Numbers vary from 6-7 but depend on the location

VIN VARCHAR(17) -> VIN numbers are always 17 characters universally

Model CHAR-> Size may vary as there are many different models

Size CHAR(1) -> can only be a small 'S' or large 'L' car

Color CHAR-> Size may vary as there are many different colors

PassID INT -> Number ID

Cost INT -> Numeric(5,2)

TypeOfPass CHAR(1) -> 'D' for Daily pass, 'R' for Regular pass

StartTime DATETIME -> showcases both the date and the time in the format: YYYY-MM-DD HH:MI:SS

EndTime DATETIME -> showcases both the date and the time in format: YYYY-MM-DD HH:MI:SS

ReceiptID INT -> Number ID

PurchaseTime -> showcases both the date and the time in the format: YYYY-MM-DD HH:MI:SS

EmployeeID INT -> Number ID

FirstName VARCHAR(30) -> may vary vastly in size

LastName VARCHAR(30) -> may vary vastly in size

Email VARCHAR(100) -> may vary vastly in size

Salary -> Numeric(4,2)

TicketID INT -> Number ID

Reason VARCHAR(255) -> may vary vastly in size

FineAmount INT -> Money owed measured with int

RecordedTime DATETIME -> showcases both date and the time in the format: YYYY-MM-DD HH:MI:SS

Paid BOOL -> true or false value to know if the customer has paid ticket fees or not

10.Creating Database and Tables - SQL DDL

File Name: Create.sql

11.Inserting Values in Tables

File Name: Insert.sql

12.SQL Queries

File Name: Queries.sql