

Due on or before 19<sup>th</sup> February

**Name:** Kishan Kumar Zalavadia

**UID:** 11685261

1. Let's design an Entity-Relationship (ER) model for a Car Rental System with ten entities, considering various relationships, attributes, and constraints:

This Car Rental System scenario involves entities representing customers, cars, car categories, rentals, employees, branches, insurance, payments, pickup locations, and drop-off locations. The relationships between these entities capture the complexities of a real-world car rental system, considering factors such as customer rentals, car categories, employee management, branch oversight, insurance coverage, and payment transactions.

Customer: Attributes: CustomerID (Primary Key), Name, ContactNumber, Email  
Customer may have many rentals.

Car: Attributes: CarID (Primary Key), Model, Manufacturer, Year, RentalRate  
One car can be rented many times by many customers in each instance.  
Car belongs to a one car category and many cars can be in one car category.

CarCategory: Attributes: CategoryID (Primary Key), CategoryName  
CarCategory can have many cars in it. Each rental has a car category.

Rental: Attributes: RentalID (Primary Key), RentalDate, ReturnDate, TotalCost  
Customer may have many car rentals. Rental may have one or many cars.

Employee: Attributes: EmployeeID (Primary Key), Name, Position  
Employee processes the rental. One employee may process many rentals. One employee manages several branches. Some employees do not manage any branch.

Branch: Attributes: BranchID (Primary Key), Location  
One branch can store (park) many cars. In a branch, there can be many employees working for that branch.

Insurance: Attributes: InsuranceID (Primary Key), PolicyNumber, CoverageDetails  
Insurance can cover many rentals.

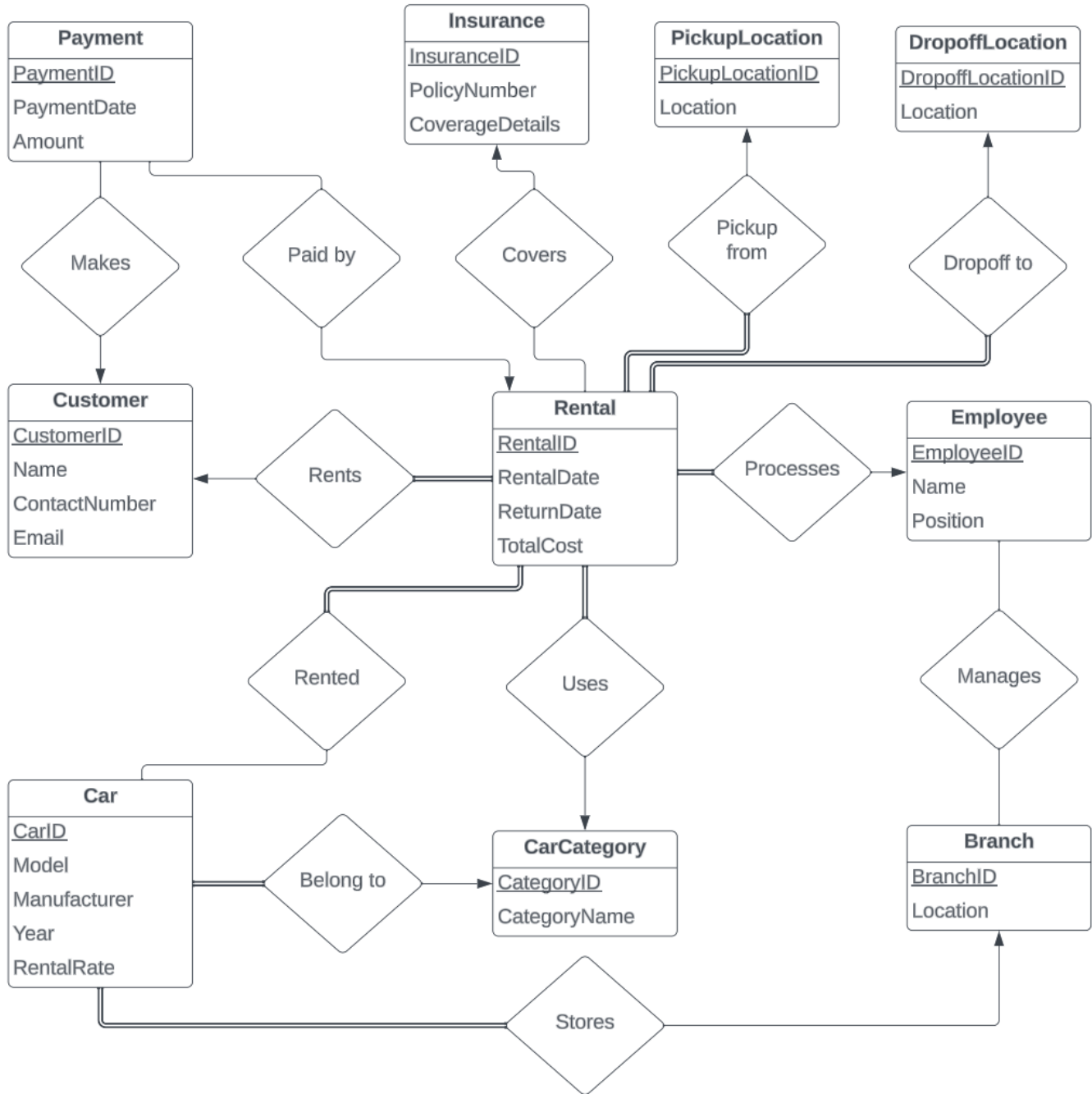
Payment: Attributes: PaymentID (Primary Key), PaymentDate, Amount  
Customer may have many payments. One rental can be made in many payments.

PickupLocation: Attributes: PickupLocationID (Primary Key), Location  
Many rentals can be picked up from a single PickupLocation

DropoffLocation: Attributes: DropoffLocationID (Primary Key), Location

Many rentals can be returned to a DropoffLocation

Ans:

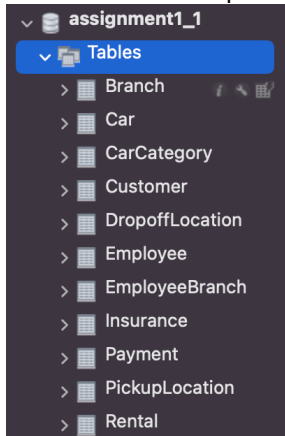


### Assumptions/Total Participations:

- Customer-Rental: Every Rental has a customer associated with it.
- Car-Rental: Every rental must have a car that is been rented.
- Car-CarCategory: Every car must belong to a car category.
- Rental-CarCategory: Each rental has a carCategory
- Employee-Rental: Every rental is processed by a customer.
- Branch-Car: Every car is stored at some or the other branch.
- Rental-PickupLocation: Every rental has a pickupLocation associated with it.
- Rental-DropoffLocation: Every rental has a dropoffLocation associated with it.

2. Answer the following questions based on the above ER model.

a. Write an SQL query to perform the following tasks.



i. To create the table Customer-10

Ans:

```
CREATE TABLE Customer (  
    CustomerID INT PRIMARY KEY,  
    Name VARCHAR(255),  
    ContactNumber VARCHAR(20),  
    Email VARCHAR(255)  
);
```

Result table:

Field	Type	Null	Key	Default	Extra
CustomerID	int	NO	PRI	NULL	
Name	varchar(255)	YES		NULL	
ContactNumber	varchar(20)	YES		NULL	
Email	varchar(255)	YES		NULL	

ii. To retrieve the rental history of a given customer (Customer Id is provided)-10

Ans:

```
SELECT RentalID, RentalDate, ReturnDate, TotalCost  
FROM Rental  
WHERE CustomerID = <CustomerID>;
```

Ex: SELECT RentalID, RentalDate, ReturnDate, TotalCost FROM Rental WHERE CustomerID = 2;

Result table:

RentalID	RentalDate	ReturnDate	TotalCost
2	2024-02-16	2024-02-19	300.00
5	2024-02-19	2024-02-26	450.00
NULL	NULL	NULL	NULL

Customer table:

CustomerID	Name	ContactNumb...	Email
1	John Doe	123-456-7890	john.doe@example.com
2	Jane Smith	987-654-3210	jane.smith@example.com
3	Alice Johnson	555-555-5555	alice.johnson@example.com
4	Bob Williams	666-666-6666	bob.williams@example.com
NULL	NULL	NULL	NULL

Rental table:

RentalID	RentalDate	ReturnDate	TotalCost	CustomerID	EmployeeID
1	2024-02-15	2024-02-20	250.00	1	1
2	2024-02-16	2024-02-19	300.00	2	2
3	2024-02-17	2024-02-22	400.00	3	3
4	2024-02-18	2024-02-25	500.00	4	4
5	2024-02-19	2024-02-26	450.00	2	2
NULL	NULL	NULL	NULL	NULL	NULL

- iii. To find the employee details with the branch he/she is working at and the rentals that he/she has processed so far. The employee Id is provided. **-10**

Ans:

```

SELECT
    e.EmployeeID,
    e.Name AS EmployeeName,
    e.Position,
    b.Location AS BranchLocation,
    r.RentalID,
    r.RentalDate,
    r.ReturnDate,
    r.TotalCost,
    c.Name AS CustomerName
FROM
    Employee AS e
JOIN
    EmployeeBranch AS eb ON e.EmployeeID = eb.EmployeeID
JOIN
    Branch AS b ON eb.BranchID = b.BranchID
JOIN
    Rental AS r ON e.EmployeeID = r.EmployeeID
JOIN
    Customer AS c ON r.CustomerID = c.CustomerID
WHERE
    e.EmployeeID = <employee_id>;

```

Ex:

```

SELECT

```

```

e.EmployeeID, e.Name AS EmployeeName, e.Position, b.Location AS
BranchLocation, r.RentalID, r.RentalDate, r.ReturnDate, r.TotalCost, c.Name AS
CustomerName
FROM
Employee AS e JOIN EmployeeBranch AS eb ON e.EmployeeID = eb.EmployeeID
JOIN Branch AS b ON eb.BranchID = b.BranchID JOIN Rental AS r ON e.EmployeeID =
r.EmployeeID JOIN Customer AS c ON r.CustomerID = c.CustomerID
WHERE
e.EmployeeID = 2;

```

Result table:

	EmployeeID	EmployeeName	Position	BranchLocation	RentalID	RentalDate	ReturnDate	TotalCost	CustomerName
	2	Emily Davis	Sales Representative	Main Street	2	2024-02-16	2024-02-19	300.00	Jane Smith
	2	Emily Davis	Sales Representative	Broadway Avenue	2	2024-02-16	2024-02-19	300.00	Jane Smith
	2	Emily Davis	Sales Representative	Main Street	5	2024-02-19	2024-02-26	450.00	Jane Smith
	2	Emily Davis	Sales Representative	Broadway Avenue	5	2024-02-19	2024-02-26	450.00	Jane Smith

iv. The amount of money each customer spent on rentals for the past month. -15

Ans:

```

SELECT c.CustomerID, c.Name AS CustomerName, SUM(r.TotalCost) AS
TotalAmountSpent
FROM Customer c
JOIN Rental r ON c.CustomerID = r.CustomerID
WHERE r.RentalDate >= DATE_SUB(CURRENT_DATE(), INTERVAL 1 MONTH)
GROUP BY c.CustomerID, c.Name;

```

Result table:

	CustomerID	CustomerName	TotalAmountSp...
	1	John Doe	250.00
	2	Jane Smith	750.00
	3	Alice Johnson	400.00
	4	Bob Williams	500.00

Customer table:

	CustomerID	Name	ContactNumb...	Email
	1	John Doe	123-456-7890	john.doe@example.com
	2	Jane Smith	987-654-3210	jane.smith@example.com
	3	Alice Johnson	555-555-5555	alice.johnson@example.com
	4	Bob Williams	666-666-6666	bob.williams@example.com
	NULL	NULL	NULL	NULL

Rental table:

RentalID	RentalDate	ReturnDate	TotalCost	CustomerID	EmployeeID
1	2024-02-15	2024-02-20	250.00	1	1
2	2024-02-16	2024-02-19	300.00	2	2
3	2024-02-17	2024-02-22	400.00	3	3
4	2024-02-18	2024-02-25	500.00	4	4
5	2024-02-19	2024-02-26	450.00	2	2
NULL	NULL	NULL	NULL	NULL	NULL

### 3. Install and run your queries on MySQL -5 Marks

Steps

1. Install MySQL server and MySQL Workbench.
2. Load the sample dataset from <https://www.db-book.com/>.
3. Run this SQL statement. **select \* from instructor;**
4. Submit the result table as a screenshot.

Ans:

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' panel displays the 'assignment1' database structure, including tables like 'advisor', 'classroom', 'course', 'department', 'instructor', 'prereq', 'section', 'student', 'takes', 'teaches', and 'time\_slot'. The main query editor contains the following SQL code:

```

10 select * from student;
11 select * from takes;
12 select * from teaches;
13 select * from time_slot;
14
15 # Question: 3
16 use assignment1;
17 select * from instructor;

```

The 'Result Grid' at the bottom displays the results of the query 'select \* from instructor;'. It shows 12 rows of data with columns: ID, name, dept\_name, and salary.

ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65000.00
12121	Wu	Finance	90000.00
15151	Mozart	Music	40000.00
22222	Einstein	Physics	95000.00
32343	El Said	History	60000.00
33456	Gold	Physics	87000.00
45565	Katz	Comp. Sci.	75000.00
58583	Califleri	History	62000.00
76543	Singh	Finance	80000.00
76768	Crick	Biology	72000.00
83821	Brandt	Comp. Sci.	92000.00
98345	Kim	Elec. Eng.	80000.00
NULL	NULL	NULL	NULL

The 'Action Output' panel at the bottom shows the execution details: 'select \* from instructor LIMIT 0, 1000' returned 12 rows in 0.0013 seconds.

**You must state the assumptions that you made during the design process. But you must design all the requirements mentioned above.**

**Note:** Plagiarism is strictly enforced, and identical solutions get ZERO marks and no negotiations.

**You Must use a software (Lucid Chart or equivalent) to create ER diagram. Handwritten diagrams will not be graded.**

Please contact graders if you have any questions regarding the E-R diagrams, mapping into relations, or normalization.