Name: Khush Bothra
Roll No.: 213075
PRN No.: 22320063

DBMS Lab Assignment – 1

Title: ER to Relational Model Conversion

Problem Statement: Draw an ER diagram for the following problem statements and then convert them into database relational schema:

CAR Rental Management

- One customer can rent one car.
- Each car has one manufacturer.
- Each manufacturer can manufacture many types of cars.
- Each car has its own maintenance schedule.
- The revenue is generated based on number of days the car was rented.
- The company has branches in 5 major cities.
- The companies also have 5 employees per branch.
- Initial rental requests are handled by employees.

Theory Questions:

Q1. What are ER Diagrams? What is their significance in DBMS?

Ans. ER diagrams, or Entity-Relationship diagrams, are a visual representation of the data model that describes the structure of a database. They are used to model the entities, attributes, and relationships within a database system. ER diagrams are a crucial component of database design as they provide a clear and concise way to represent the logical structure of a database.

Key components of ER diagrams:

1. **Entities**: Entities represent real-world objects or concepts, such as customers, products, or employees. Each entity is depicted as a rectangle in the diagram.

2. **Attributes**: Attributes are properties or characteristics of entities. They are represented within ovals and are connected to the respective entities. For example, a "Customer" entity may have attributes like "CustomerID," "Name," and "Email."

3. **Relationships**: Relationships illustrate how entities are related to each other. They are shown as lines connecting entities and are labelled to indicate the nature of the relationship.

For instance, a relationship between "Customer" and "Order" entities might be labelled as "places," indicating that a customer places an order.

ER diagrams are significant in Database Management Systems (DBMS) for several reasons:

1. **Visualization**: ER diagrams provide a visual representation of the database structure, making it easier for designers, developers, and stakeholders to understand the relationships between different entities and how they interact.

2. **Communication**: ER diagrams serve as a communication tool between various stakeholders involved in the database design process. Designers can use these diagrams to convey their understanding of the system's requirements and structure to others.

3. **Database Design**: ER diagrams play a vital role in the design phase of a database. They help in identifying entities, their attributes, and the relationships between them. This information is crucial for creating an efficient and well-organized database schema.

4. **Normalization**: ER diagrams assist in the normalization process, which involves organizing data to minimize redundancy and improve data integrity. By visualizing relationships, designers can identify opportunities for normalization, resulting in a more robust and efficient database design.

Q2. Explain the conversion process from ER to Relational.

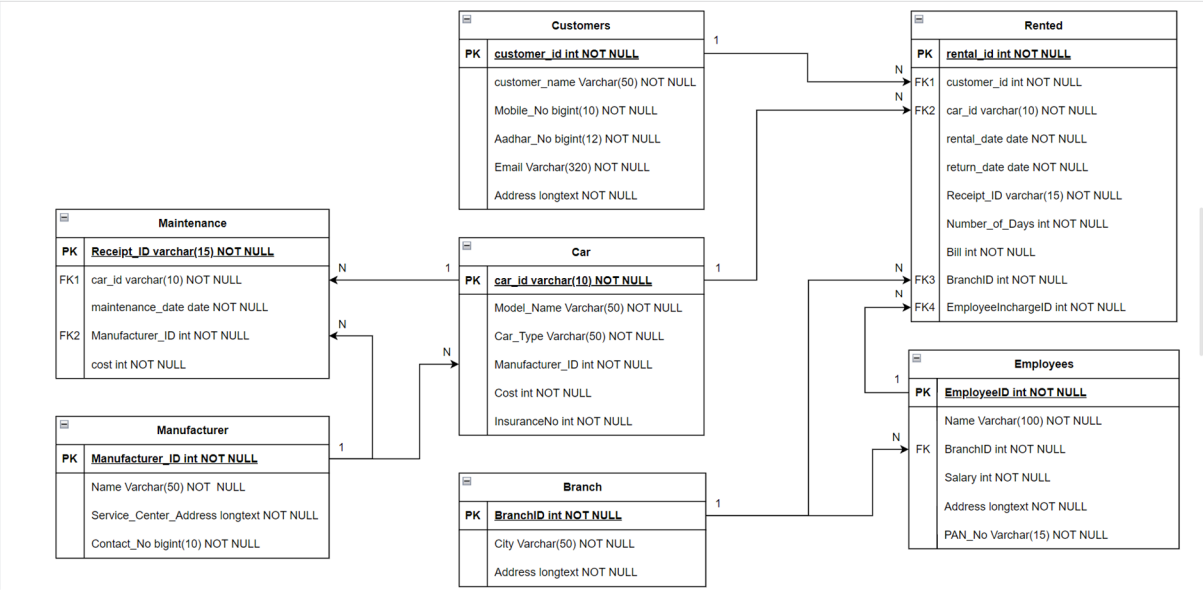
Ans. Converting an Entity-Relationship (ER) model to a relational model involves transforming the conceptual design represented by entities, relationships, and attributes into a set of relational tables. The process typically includes defining tables, attributes, primary keys, and foreign keys to capture the structure and relationships of the database. Here are the general steps in the conversion process:

1. **Identify Entities:**
   - Each entity in the ER diagram becomes a table in the relational model.
2. **Identify Attributes:**
   - Each attribute of an entity becomes a column in the corresponding table.
   - Attributes that are part of the primary key become primary key columns.
3. **Identify Relationships:**
   - Relationships in ER diagrams are represented by foreign keys in relational databases.
   - For each relationship, identify the entities involved and the cardinality (e.g., one-to-one, one-to-many, many-to-many).
4. **Resolve Many-to-Many Relationships:**
   - If there are many-to-many relationships, create a new table (junction table) to represent the association.
   - The junction table will have foreign keys referencing the participating entities.
5. **Determine Primary Keys:**
   - Identify or create primary keys for each table.
   - Primary keys uniquely identify each record in a table.
6. **Specify Foreign Keys:**

- Identify foreign keys that reference the primary keys of other tables to represent relationships.
- Ensure referential integrity by enforcing foreign key constraints.

7. **Normalize Tables:**
   - Apply normalization techniques to minimize data redundancy and improve data integrity.
   - This may involve decomposing tables into smaller tables to meet normalization rules.
8. **Denormalization (if necessary):**
   - In some cases, denormalization may be applied for performance reasons.
   - This involves adding redundancy to tables to reduce the need for joins in certain queries.
9. **Review and Refine:**
   - Review the relational model to ensure it accurately captures the information from the ER diagram.
   - Refine the model based on feedback and optimization considerations.
10. **Document Constraints:**
    - Document any additional constraints, such as unique constraints or check constraints.
11. **Create Data Definition Language (DDL) Statements:**
    - Use DDL statements (e.g., SQL) to create the tables, define columns, primary keys, foreign keys, and other constraints.

Entity Relationship Diagram:



Name: Khush Bothra
Roll No.: 213075
PRN No.: 22320063

Code:

```
use dbms_a1;

CREATE TABLE `customers` (
  `customer_id` INT NOT NULL,
  `customer_name` VARCHAR(50) NOT NULL,
  `Mobile_No` BIGINT NOT NULL,
  `Aadhar_No` BIGINT NOT NULL,
  `Email` VARCHAR(320) NOT NULL,
  `Address` LONGTEXT NOT NULL,
  PRIMARY KEY (`customer_id`));

CREATE TABLE `branch` (
  `branch_id` INT NOT NULL,
  `city` VARCHAR(50) NOT NULL,
  `address` longtext NOT NULL,
  PRIMARY KEY (`branch_id`));

 CREATE TABLE `manufacturer` (
  `manufacturer_id` INT NOT NULL,
  `name` VARCHAR(50) NOT NULL,
  `service_center_address` longtext NOT NULL,
  `contact_no` bigint NOT NULL,
  PRIMARY KEY (`manufacturer_id`));

 CREATE TABLE `employees` (
  `employee_id` INT NOT NULL,
  `name` VARCHAR(100) NOT NULL,
  `branch_id` INT NOT NULL,
  `salary` INT NOT NULL,
  `pan_no` VARCHAR(15) NOT NULL,
  `address` LONGTEXT NOT NULL,
  PRIMARY KEY (`employee_id`),
  foreign key (branch_id) references branch(branch_id));

 CREATE TABLE `car` (
  `car_id` varchar(10) NOT NULL,
  `model_name` VARCHAR(50) NOT NULL,
  `car_type` VARCHAR(50) NOT NULL,
  `manufacturer_id` int NOT NULL,
  `cost` int NOT NULL,
  `insuranceNo` int NOT NULL,
  PRIMARY KEY (`car_id`),
  foreign key (manufacturer_id) references manufacturer(manufacturer_id));

 CREATE TABLE `maintenance` (
  `receipt_id` varchar(15) NOT NULL,
  `car_id` VARCHAR(10) NOT NULL,
  `maintenance_date` date NOT NULL,
  `manufacturer_id` int NOT NULL,
  `cost` int NOT NULL,
  PRIMARY KEY (`receipt_id`),
  foreign key (manufacturer_id) references manufacturer(manufacturer_id),
  foreign key (car_id) references car(car_id));

 CREATE TABLE `rented` (
  `rental_id` int NOT NULL,
  `customer_id` int NOT NULL,
  `car_id` VARCHAR(10) NOT NULL,
  `rental_date` date NOT NULL,
  `return_date` date NOT NULL,
  `receipt_id` varchar(15) NOT NULL,
  `no_of_days` int NOT NULL,
  `bill` int NOT NULL,
  `branch_id` int NOT NULL,
  `employee_incharge_id` int NOT NULL,
  PRIMARY KEY (`rental_id`),
  foreign key (customer_id) references customers(customer_id),
  foreign key (car_id) references car(car_id),
  foreign key (branch_id) references branch(branch_id),
  foreign key (employee_incharge_id) references employees(employee_id));
```

Name: Khush Bothra
Roll No.: 213075
PRN No.: 22320063

Table Views:

Customer:

Field	Type	Null	Key	Default	Extra
customer_id	int	NO	PRI	NULL	
customer_name	varchar(50)	NO		NULL	
Mobile_No	bigint	NO		NULL	
Aadhar_No	bigint	NO		NULL	
Email	varchar(320)	NO		NULL	
Address	longtext	NO		NULL	

Branch:

Field	Type	Null	Key	Default	Extra
branch_id	int	NO	PRI	NULL	
city	varchar(50)	NO		NULL	
address	longtext	NO		NULL	

Manufacturer:

Field	Type	Null	Key	Default	Extra
manufacturer_id	int	NO	PRI	NULL	
name	varchar(50)	NO		NULL	
service_center_address	longtext	NO		NULL	
contact_no	bigint	NO		NULL	

Employees:

Field	Type	Null	Key	Default	Extra
employee_id	int	NO	PRI	NULL	
name	varchar(100)	NO		NULL	
branch_id	int	NO	MUL	NULL	
salary	int	NO		NULL	
pan_no	varchar(15)	NO		NULL	
address	longtext	NO		NULL	

Car:

Field	Type	Null	Key	Default	Extra
car_id	varchar(10)	NO	PRI	NULL	
model_name	varchar(50)	NO		NULL	
car_type	varchar(50)	NO		NULL	
manufacturer_id	int	NO	MUL	NULL	
cost	int	NO		NULL	
insuranceNo	int	NO		NULL	

Name: Khush Bothra
Roll No.: 213075
PRN No.: 22320063

Maintenance:

Field	Type	Null	Key	Default	Extra
receipt_id	varchar(15)	NO	PRI	NULL	
car_id	varchar(10)	NO	MUL	NULL	
maintenance_date	date	NO		NULL	
manufacturer_id	int	NO	MUL	NULL	
cost	int	NO		NULL	

Rented:

Field	Type	Null	Key	Default	Extra
rental_id	int	NO	PRI	NULL	
customer_id	int	NO	MUL	NULL	
car_id	varchar(10)	NO	MUL	NULL	
rental_date	date	NO		NULL	
return_date	date	NO		NULL	
receipt_id	varchar(15)	NO		NULL	
no_of_days	int	NO		NULL	
bill	int	NO		NULL	
branch_id	int	NO	MUL	NULL	
employee_incharge_id	int	NO	MUL	NULL	

Conclusion: While performing this assignment, we have learned about Entity Relationship Diagrams and how they are created. We further learned to implement these ER diagrams into Relational Databases.