

## **SIT- 103 DATABASE FUNDAMENTALS**

### **MINIPROJECT PART 1 - DATABASE DESIGN AND NORMALISATION**

**JASVEENA-224001588**

#### **UBER:**

Uber is a sophisticated taxi booking application that operates via its mobile platform, enabling users to request a cab from virtually any location. The system efficiently matches customers with available drivers based on proximity and availability. Active drivers who are ready and willing to accept booking requests receive notifications through the platform, allowing them to confirm and secure the ride for the customer. This seamless process ensures that customers can quickly find and book a cab, while drivers can efficiently manage and fulfill ride requests.

Uber's database system is designed to manage and organize the various components involved in the taxi booking process. Here is a detailed breakdown of how the system handles different data elements:

#### **1. Customer Management:**

- **Customer Table:**
  - **Primary Key:** Customer\_ID (unique identifier for each customer)
  - **Attributes:** Name, Age, Email, Phone\_Number
- **Description:** Each customer is uniquely identified by a Customer\_ID. The system maintains a record of customer details including their full name, age, email address, and phone number. This information is used for identification, communication, and service customization.

#### **2. Booking Process:**

- **Booking Table:**
  - **Primary Key:** Booking\_ID
  - **Attributes:** Booking\_Date, Current\_Location, Destination\_Location, Customer\_ID (FK), Driver\_ID (FK)
- **Description:** When a customer requests a ride, the system generates a unique Booking\_ID for each request. It records the date of the booking, the current location (pickup point), and the destination location. The Customer\_ID links

the booking to the specific customer who made the request. Driver\_ID references the driver who accepted the booking.

### 3. Driver Information:

- **Driver Table:**
  - **Primary Key:** Driver\_ID
  - **Attributes:** Name, Email, Phone\_Number, Car\_Number
- **Description:** Each driver is uniquely identified by a Driver\_ID. The system records key details such as the driver's name, email address, phone number, and vehicle registration number. This information helps in matching drivers with customer bookings and facilitating communication.

### 4. Payment Records:

- **Payment Table:**
  - **Primary Key:** Payment\_ID
  - **Attributes:** Payment\_Date, Amount, Booking\_ID (FK)
- **Description:** After the ride, the customer makes a payment, which is recorded in the Payment table. Each payment is assigned a unique Payment\_ID, and details such as the payment date and amount are stored. The Booking\_ID links the payment to the specific ride it pertains to.

### 5. Review and Rating System:

- **Review Table:**
  - **Primary Key:** Review\_ID
  - **Attributes:** Review\_Date, Description, Driver\_ID, Booking\_ID(FK)
- **Description:** Customers can leave reviews and ratings for drivers after a ride. Each review is assigned a unique Review\_ID, and includes the review date and textual feedback. The Driver\_ID links the review to the specific driver, and the Booking\_ID associates the review with the relevant ride.

NORMALISATION STATUS :

#### 1. First Normal Form (1NF)

1NF requires that:

- Each table has a clear primary key.

- All attributes contain atomic (indivisible) values.
- Each column contains values of a single type.

**Analysis:**

- **Customer Table:** Contains Customer\_id, Name, Age, Email, Phone\_number. All attributes are atomic, and Customer\_id is the primary key.
- **Driver Table:** Contains Driver\_id, Name, Email, Phone\_number, Car\_number. All attributes are atomic, and Driver\_id is the primary key.
- **Booking Table:** Contains Booking\_id, Booking\_date, Curr\_location, Dest\_location. All attributes are atomic, and Booking\_id is the primary key.
- **Review Table:** Contains Review\_id, Review\_date, Description. All attributes are atomic, and Review\_id is the primary key.
- **Payment Table:** Contains Payment\_id, Payment\_date, Amount. All attributes are atomic, and Payment\_id is the primary key.

**Status:** All tables appear to be in 1NF.

## 2. Second Normal Form (2NF)

2NF requires that:

- The table is in 1NF.
- All non-key attributes are fully functionally dependent on the entire primary key (i.e., no partial dependencies).

**Analysis:**

- **Customer Table:** Customer\_id is the primary key, and all other attributes depend fully on Customer\_id.
- **Driver Table:** Driver\_id is the primary key, and all other attributes depend fully on Driver\_id.
- **Booking Table:** Booking\_id is the primary key, and all other attributes depend fully on Booking\_id.
- **Review Table:** Review\_id is the primary key, and all other attributes depend fully on Review\_id.
- **Payment Table:** Payment\_id is the primary key, and all other attributes depend fully on Payment\_id.

**Status:** All tables appear to be in 2NF.

### 3. Third Normal Form (3NF)

3NF requires that:

- The table is in 2NF.
- All non-key attributes are non-transitively dependent on the primary key (i.e., no transitive dependencies).

**Analysis:**

- **Customer Table:** Customer\_id is the primary key. Attributes like Name, Age, Email, and Phone\_number are directly dependent on Customer\_id, with no transitive dependencies.
- **Driver Table:** Driver\_id is the primary key. Attributes like Name, Email, Phone\_number, and Car\_number are directly dependent on Driver\_id, with no transitive dependencies.
- **Booking Table:** Booking\_id is the primary key. Attributes like Booking\_date, Curr\_location, and Dest\_location are directly dependent on Booking\_id, with no transitive dependencies.
- **Review Table:** Review\_id is the primary key. Attributes like Review\_date and Description are directly dependent on Review\_id, with no transitive dependencies.
- **Payment Table:** Payment\_id is the primary key. Attributes like Payment\_date and Amount are directly dependent on Payment\_id, with no transitive dependencies.

**Status:** All tables appear to be in 3NF.

Therefore, the schema for the Uber taxi booking application is normalized up to the Third Normal Form (3NF).

DEPENDENCY DIAGRAM

table: customer

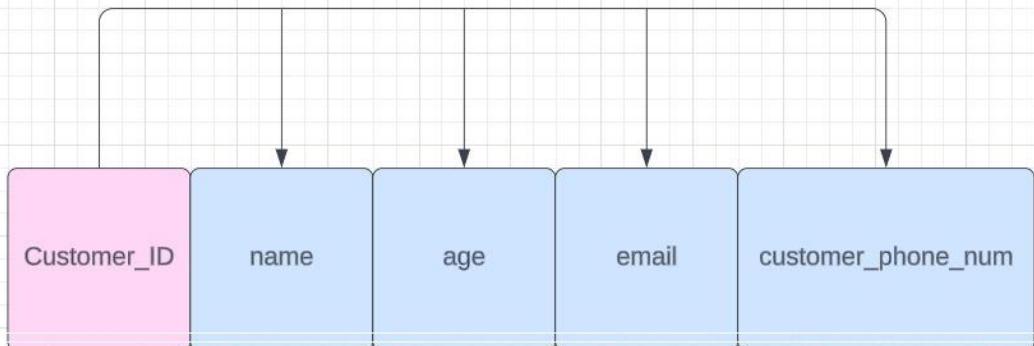


table: driver

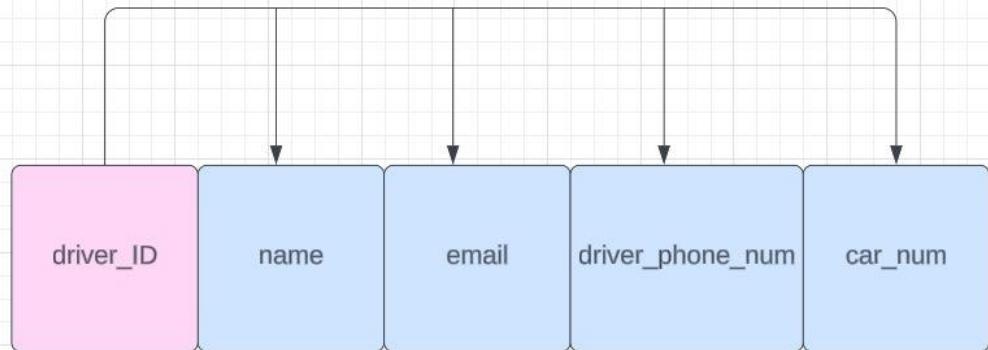


table: review

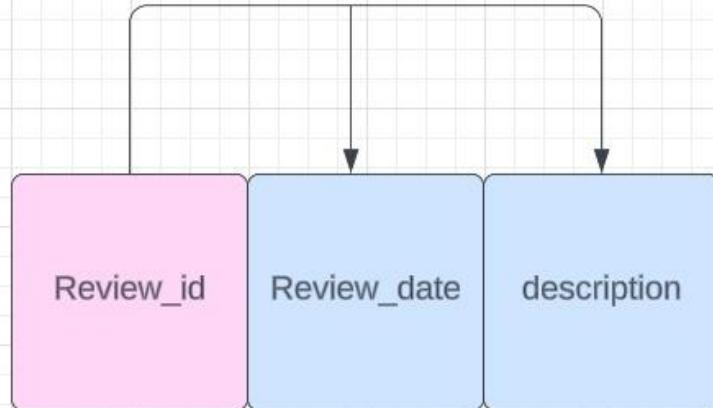


table: booking

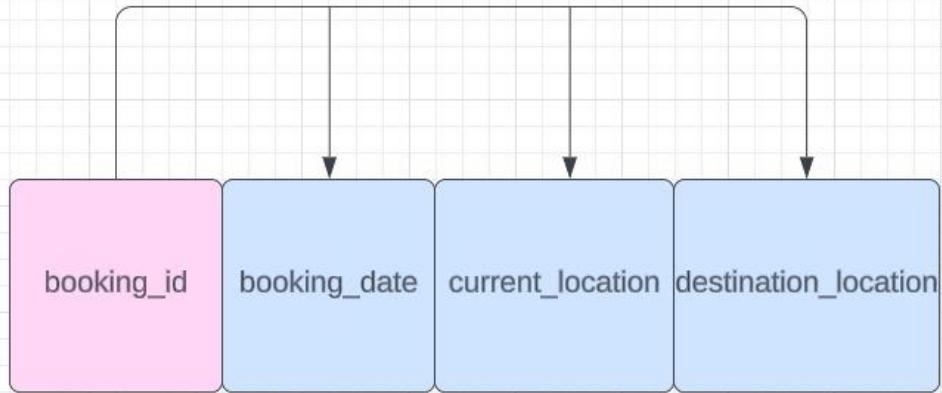
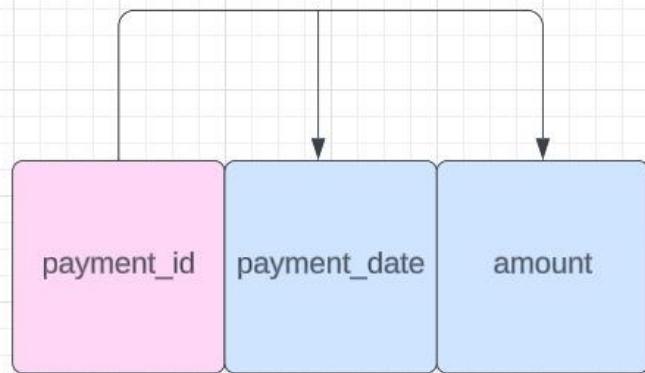


table: payment



ERD:

