

# DBMS ASSIGNMENT

Nikki Gautam

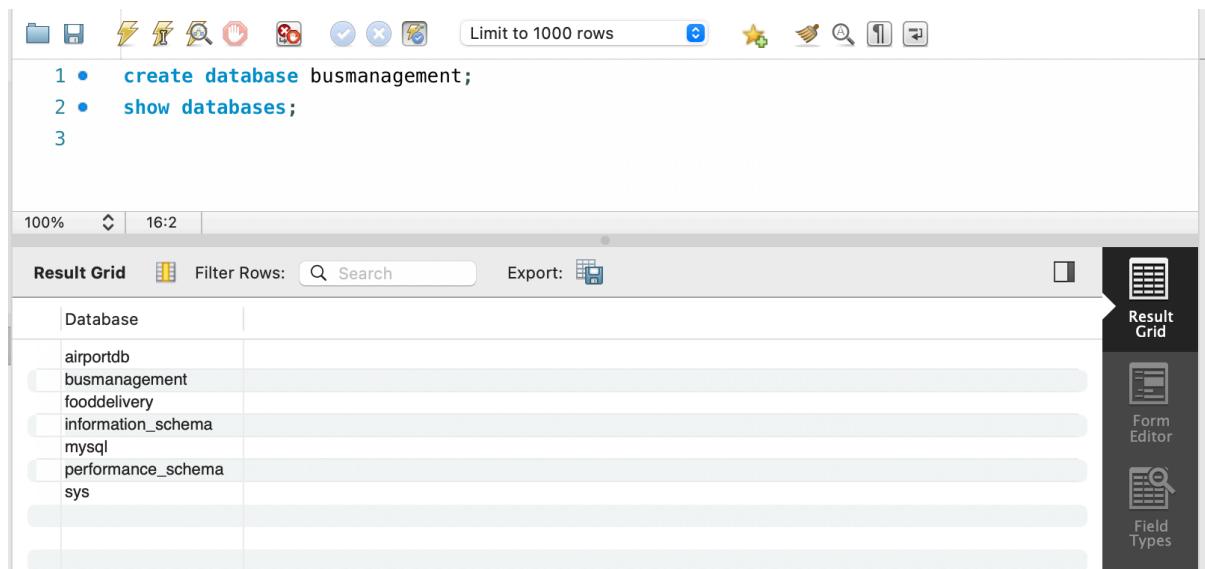
## Question 2:

Language Used (for querying): **SQL**

### Part A:

For the ER Model created in the earlier question, create the database tables, and normalize them. You are free to modify the original ER diagram with new findings / changes. You are also free to leave some tables in a denormalized state if you can justify it in the final report (part C).

**Database Created:** busmanagement



The screenshot shows the MySQL Workbench interface. At the top, there is a toolbar with various icons for database management. Below the toolbar, a SQL editor window displays the following commands:

```
1 •  create database busmanagement;
2 •  show databases;
3
```

Below the SQL editor is a results grid titled "Result Grid". The grid shows a list of databases:

Database
airportdb
busmanagement
fooddelivery
information_schema
mysql
performance_schema
sys

On the right side of the interface, there is a vertical sidebar with three tabs: "Result Grid" (which is currently selected), "Form Editor", and "Field Types".

## Database Tables:

### 1. User Table

Used to store information of the users who interact with the system

```
CREATE TABLE User (
    user_id INT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL,
    phone_number VARCHAR(15) UNIQUE NOT NULL,
    password VARCHAR(100) NOT NULL,
    user_type ENUM('passenger', 'owner', 'admin') NOT
NULL
);
```

The screenshot shows the MySQL Workbench interface. In the SQL tab, the following code is displayed:

```
1 •  use busmanagement;
2
3 •  CREATE TABLE User (
4      user_id INT PRIMARY KEY,
5      name VARCHAR(100) NOT NULL,
6      email VARCHAR(100) UNIQUE NOT NULL,
7      phone_number VARCHAR(15) UNIQUE NOT NULL,
8      password VARCHAR(100) NOT NULL,
9      user_type ENUM('passenger', 'owner', 'admin') NOT
NULL
10 );
11
12 •  describe User;
13
```

In the Results tab, the table structure is shown in a grid:

Field	Type	Null	Key	Default	Extra
user_id	int	NO	PRI	NULL	
name	varchar(100)	NO		NULL	
email	varchar(100)	NO	UNI	NULL	
phone_number	varchar(15)	NO	UNI	NULL	
password	varchar(100)	NO		NULL	
user_type	enum('passenger','owner','admin')	NO		NULL	

The Results tab has a sidebar with icons for Result Grid, Form Editor, and Find.

## 2. Bus Table

Used to store details about the buses owned by users

```
CREATE TABLE Bus (
    bus_id INT PRIMARY KEY,
    bus_number VARCHAR(20) UNIQUE NOT NULL,
    type ENUM('AC', 'Non-AC', 'Sleeper') NOT NULL,
    capacity INT NOT NULL,
    owner_id INT NOT NULL,
    amenities VARCHAR(255),
    FOREIGN KEY (owner_id) REFERENCES User(user_id)
ON DELETE CASCADE
);
```

The screenshot shows the MySQL Workbench interface. At the top, there is a toolbar with various icons for database management. Below the toolbar, the SQL editor window displays the creation of the 'Bus' table and its description:

```
1 • CREATE TABLE Bus (
2     bus_id INT PRIMARY KEY,
3     bus_number VARCHAR(20) UNIQUE NOT NULL,
4     type ENUM('AC', 'Non-AC', 'Sleeper') NOT NULL,
5     capacity INT NOT NULL,
6     owner_id INT NOT NULL,
7     amenities VARCHAR(255),
8     FOREIGN KEY (owner_id) REFERENCES User(user_id) ON DELETE CASCADE
9 );
10
11 • describe Bus;
```

Below the SQL editor is the Results grid, which shows the structure of the 'Bus' table:

Field	Type	Null	Key	Default	Extra
bus_id	int	NO	PRI	NULL	
bus_number	varchar(20)	NO	UNI	NULL	
type	enum('AC','Non-AC','Sleeper')	NO		NULL	
capacity	int	NO		NULL	
owner_id	int	NO	MUL	NULL	
amenities	varchar(255)	YES		NULL	

On the right side of the results grid, there is a vertical toolbar with three icons: 'Result Grid' (selected), 'Form Editor', and 'Field'.

### 3. Route Table

Used to store details of routes the buses take

```
CREATE TABLE Route (
    route_id INT PRIMARY KEY,
    source VARCHAR(100) NOT NULL,
    destination VARCHAR(100) NOT NULL,
    total_distance DECIMAL(10, 2) NOT NULL,
    estimated_time TIME NOT NULL
);
```

The screenshot shows the MySQL Workbench interface. At the top, there is a toolbar with various icons. Below the toolbar, a code editor window displays the SQL code for creating the Route table and its description. The code is color-coded for syntax highlighting. In the bottom half of the interface, there is a 'Result Grid' window showing the table structure with five columns: Field, Type, Null, Key, and Default. The table has five rows corresponding to the columns defined in the CREATE TABLE statement. On the right side of the interface, there is a vertical panel with three tabs: 'Result Grid' (which is currently selected), 'Form Editor', and 'Field'.

Field	Type	Null	Key	Default	Extra
route_id	int	NO	PRI	NULL	
source	varchar(100)	NO		NULL	
destination	varchar(100)	NO		NULL	
total_distance	decimal(10,2)	NO		NULL	
estimated_time	time	NO		NULL	

## 4. Trip Table

Used to store link buses and routes to specific trips

```
CREATE TABLE Trip (
    trip_id INT PRIMARY KEY,
    bus_id INT NOT NULL,
    route_id INT NOT NULL,
    trip_date DATE NOT NULL,
    departure_time TIME NOT NULL,
    arrival_time TIME NOT NULL,
    status ENUM('Scheduled', 'Ongoing', 'Completed')
NOT NULL,
    FOREIGN KEY (bus_id) REFERENCES Bus(bus_id) ON
DELETE CASCADE,
    FOREIGN KEY (route_id) REFERENCES Route(route_id)
ON DELETE CASCADE
);
```

The screenshot shows the MySQL Workbench interface with the following details:

- Script Editor:** Displays the SQL code for creating the Trip table.
- Result Grid:** Shows the structure of the Trip table with the following columns and their properties:

Field	Type	Null	Key	Default	Extra
trip_id	int	NO	PRI	NULL	
bus_id	int	NO	MUL	NULL	
route_id	int	NO	MUL	NULL	
trip_date	date	NO		NULL	
departure_time	time	NO		NULL	
arrival_time	time	NO		NULL	
status	enum('Scheduled','Ongoing','Completed')	NO		NULL	

- Right Panel:** Contains three tabs: "Result Grid", "Form Editor", and "Field".

## 5. Ticket Table

Used to store information of tickets booked by passengers for trips

```
CREATE TABLE Ticket (
    ticket_id INT PRIMARY KEY,
    trip_id INT NOT NULL,
    user_id INT NOT NULL,
    seat_number INT NOT NULL,
    booking_time TIMESTAMP NOT NULL,
    fare DECIMAL(10, 2) NOT NULL,
    status ENUM('Booked', 'Cancelled', 'Checked-in')
NOT NULL,
    FOREIGN KEY (trip_id) REFERENCES Trip(trip_id) ON
DELETE CASCADE,
    FOREIGN KEY (user_id) REFERENCES User(user_id) ON
DELETE CASCADE
);
```

The screenshot shows the MySQL Workbench interface with the SQL editor tab open. The SQL code for creating the `Ticket` table is displayed, along with the `describe` command to view the table's structure.

```
1 • CREATE TABLE Ticket (
2     ticket_id INT PRIMARY KEY,
3     trip_id INT NOT NULL,
4     user_id INT NOT NULL,
5     seat_number INT NOT NULL,
6     booking_time TIMESTAMP NOT NULL,
7     fare DECIMAL(10, 2) NOT NULL,
8     status ENUM('Booked', 'Cancelled', 'Checked-in') NOT NULL,
9     FOREIGN KEY (trip_id) REFERENCES Trip(trip_id) ON DELETE CASCADE,
10    FOREIGN KEY (user_id) REFERENCES User(user_id) ON DELETE CASCADE
11 );
12 • describe Ticket;
```

The results grid shows the following table structure:

Field	Type	Null	Key	Default	Extra
ticket_id	int	NO	PRI	NULL	
trip_id	int	NO	MUL	NULL	
user_id	int	NO	MUL	NULL	
seat_number	int	NO		NULL	
booking_time	timestamp	NO		NULL	
fare	decimal(10,2)	NO		NULL	
status	enum('Booked','Cancelled','Checked-in')	NO		NULL	

On the right side of the interface, there is a vertical toolbar with three buttons: **Result Grid**, **Form Editor**, and **Field**.

## 6. Check-in Table

Used to store check-in information by passengers

```
CREATE TABLE CheckIn (
    checkin_id INT PRIMARY KEY,
    ticket_id INT UNIQUE NOT NULL,
    checkin_time TIMESTAMP NOT NULL,
    FOREIGN KEY (ticket_id) REFERENCES
Ticket(ticket_id) ON DELETE CASCADE
);
```

The screenshot shows the MySQL Workbench interface. At the top, there is a toolbar with various icons. Below the toolbar, a query editor window displays the SQL code for creating the CheckIn table. The code is numbered from 1 to 8. After the creation command, there is a 'describe' command. The main pane below the toolbar shows the table structure in a grid format. On the right side, there is a vertical panel with three tabs: 'Result Grid' (selected), 'Form Editor', and 'Field'. The 'Result Grid' tab shows the columns: checkin\_id, ticket\_id, and checkin\_time, with their respective types, nullability, key status, default values, and extra options.

Field	Type	Null	Key	Default	Extra
checkin_id	int	NO	PRI	NULL	
ticket_id	int	NO	UNI	NULL	
checkin_time	timestamp	NO		NULL	

## 7. Bus Tracking Table

Used to store information to track the location of buses during trips

```
CREATE TABLE BusTracking (
    tracking_id INT PRIMARY KEY,
    trip_id INT NOT NULL,
    current_location VARCHAR(255) NOT NULL,
    last_updated TIMESTAMP NOT NULL,
    FOREIGN KEY (trip_id) REFERENCES Trip(trip_id) ON
DELETE CASCADE
);
```

The screenshot shows the MySQL Workbench interface. At the top, there is a toolbar with various icons. Below the toolbar, a code editor window displays the SQL code for creating the BusTracking table. The code is numbered from 1 to 9. Line 9 shows the 'describe' command being run on the BusTracking table. Below the code editor is a results grid titled 'Result Grid'. The results grid shows the structure of the BusTracking table with four columns: Field, Type, Null, and Key. The table has four rows with the following data:

Field	Type	Null	Key
tracking_id	int	NO	PRI
trip_id	int	NO	MUL
current_location	varchar(255)	NO	
last_updated	timestamp	NO	

On the right side of the results grid, there is a vertical toolbar with three icons: 'Result Grid', 'Form Editor', and 'Field'.

## 8. Payment Table

Used to store information to manage payments for booked tickets

```
CREATE TABLE Payment (
    payment_id INT PRIMARY KEY,
    ticket_id INT UNIQUE NOT NULL,
    payment_time TIMESTAMP NOT NULL,
    amount DECIMAL(10, 2) NOT NULL,
    payment_method ENUM('Credit Card', 'Debit Card',
    'Net Banking', 'UPI', 'Wallet') NOT NULL,
    status ENUM('Successful', 'Failed') NOT NULL,
    FOREIGN KEY (ticket_id) REFERENCES
Ticket(ticket_id) ON DELETE CASCADE
);
```

The screenshot shows the MySQL Workbench interface with the following details:

- SQL Editor:** Displays the SQL code for creating the `Payment` table and its description.
- Result Grid:** Shows the structure of the `Payment` table with the following columns and their properties:

Field	Type	Null	Key	Default	Extra
payment_id	int	NO	PRI	NULL	
ticket_id	int	NO	UNI	NULL	
payment_time	timestamp	NO		NULL	
amount	decimal(10,2)	NO		NULL	
payment_method	enum('Credit Card','Debit Card','Net Banking','UPI','Wallet')	NO		NULL	
status	enum('Successful','Failed')	NO		NULL	

- Right Panel:** A sidebar with three tabs: `Result Grid` (selected), `Form Editor`, and `Field`.

## 9. Feedback Table

Used to store the feedback given by users for trips

```
CREATE TABLE Feedback (
    feedback_id INT PRIMARY KEY,
    user_id INT NOT NULL,
    trip_id INT NOT NULL,
    rating INT CHECK (rating BETWEEN 1 AND 5),
    comment TEXT,
    feedback_time TIMESTAMP NOT NULL,
    FOREIGN KEY (user_id) REFERENCES User(user_id) ON
DELETE CASCADE,
    FOREIGN KEY (trip_id) REFERENCES Trip(trip_id) ON
DELETE CASCADE
);
```

The screenshot shows the MySQL Workbench interface with the following details:

- SQL Editor:** Displays the SQL code for creating the Feedback table. It includes the table definition with columns for feedback\_id, user\_id, trip\_id, rating, comment, and feedback\_time, along with foreign key constraints referencing the User and Trip tables.
- Result Grid:** Shows the description of the Feedback table, listing its fields, their types, nullability, key status, and default values.
- Right Panel:** A sidebar with three tabs: "Result Grid" (selected), "Form Editor", and "Field".

Field	Type	Null	Key	Default	Extra
feedback_id	int	NO	PRI	NULL	
user_id	int	NO	MUL	NULL	
trip_id	int	NO	MUL	NULL	
rating	int	YES		NULL	
comment	text	YES		NULL	
feedback_time	timestamp	NO		NULL	

## **Normalization (Applied in the solution proposed):**

### **First Normal Form (1NF):**

- All attributes are atomic and single-valued
- The table “**Bus**” (amenities) allow a list of amenities. To keep it atomic, a separate table can be created for bus amenities:

```
CREATE TABLE BusAmenities (
    bus_id INT NOT NULL,
    amenity VARCHAR(100) NOT NULL,
    PRIMARY KEY (bus_id, amenity),
    FOREIGN KEY (bus_id) REFERENCES Bus(bus_id)
ON DELETE CASCADE
);
```

### **Second Normal Form (2NF):**

- All non-key attributes are fully dependent on the primary key
- No changes are needed as the ER model already ensures no partial dependencies

### **Third Normal Form (3NF):**

- No transitive dependencies exist
- Tables like “**Ticket**” and “**Payment**” are already normalized, as every non-key attribute is dependent solely on the primary key

## **Denormalized Tables** (Justification for keeping them as they are):

### **1. Ticket Table:**

- **Reason for Denormalization:** Instead of breaking “fare” or “status” into separate tables, keeping them in the “Ticket” table reduces join queries during frequent ticket-related operations.

### **2. BusTracking Table:**

- **Reason for Denormalization:** “current\_location” is updated frequently, and keeping this denormalized avoids joins when fetching the latest location for display in real-time.
- **Since normalization ensures consistency and removes redundancy.** Hence, most tables are normalized to **3NF**
- **Denormalization improves performance** for frequently accessed attributes like “current\_location” and “fare”
- The schema supports scalability, frequent updates, and real-time tracking while ensuring data integrity.

Final glimpse of “busmanagement”:

The screenshot shows the MySQL Workbench interface with the following details:

- Toolbar:** Includes icons for file operations, search, and various database management functions.
- Query Editor:** Shows the command `1 • show tables;`.
- Result Grid:** Displays the results of the query in a tabular format. The table has one row with the header "Tables\_in\_busmanagement" and eight data rows: Bus, BusTracking, CheckIn, Feedback, Payment, Route, Ticket, Trip, and User.
- Right Sidebar:** Contains three panels:
  - Result Grid:** Shows a preview of the result grid icon.
  - Form Editor:** Shows a preview of the form editor icon.
  - Field Types:** Shows a preview of the field types icon.