ASSIGNMENT - 01

C. Maithri
192312382

(1) Fleet management system for Logistics Tracking Develop a database to manage logistics for a fleet of vehicles, including maintenance records, trip history, and fuel usage.

Requirements:

create tables for vehicles, trips, drivers, maintenance, and fuel usage, with relationships linking trips and drivers.

write stored procedures to track trip completion, calculate fuel consumption, and shedule maintenance.

write queries to generate fuel efficiency reports, maintenance shedules, and driver activity logs. Implement triggers to automatically update maintenance records and send alerts for overdue services.

Give conceptual,

A:

Conceptual ERD model:

- correctly identified entities and relationships.
- consider adding on entity for "category" to further normalize the data.

Logical ERD model:

- Well-defined attributes for each entity.
- consider adding a "status" attribute to the Enrollment entity (e.g., "Enrolled", "completed", "inprogress").

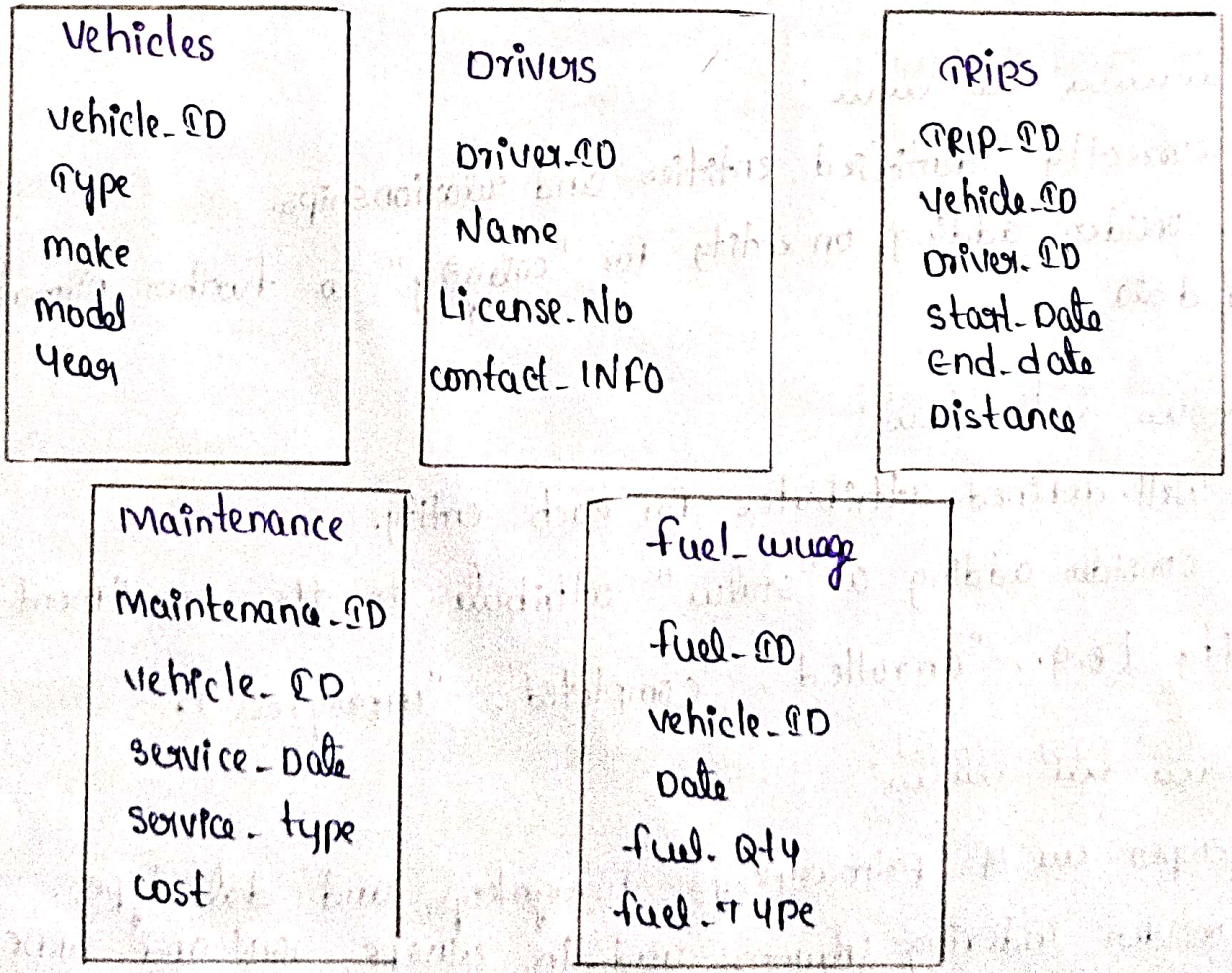
Physical ERD model:

- proper use of primarykeys, foreignkeys, and data types.
- consider indexing columns used in WHERE, JOIN and ORDER BY clauses for improved performance.

Additional Suggestions:

1. Consider adding a "Rating" attribute to the course entity to store user ratings.
2. Add a "Timestamp" attribute to the completion time entity to track completion time.
3. Create a seperate table for "Questions" to store individual quiz questions.
4. Use ENUM data type for "category" and "status" attributes.
5. Use VARCHAR(255) for "password" to accommodate hashed passwords.

* Conceptual - ER-diagram :-



Logical-ER-diagram :-

Vehicles
Vehicle_ID (PK)
Type
Make
model
Year

Drivers
Driver_ID (PK)
Name
License_No
contact_INFO

TRIPS
TRIP_ID (PK)
vehicle_ID (FK)
Driver_ID (FK)
start_Date
End_Date
Distance

maintenance
maintenance_ID (PK)
Vehicle_ID (FK)
service_Date
service_Type
cost

fuel_usage
fuel_ID (PK)
vehicle_ID (FK)
Date
fuel_Qty
fuel_type

conceptual-ER-diagram :

```
create table vehicles (
    Vehicle_ID INT PK,
    Type VARCHAR (50),
    make VARCHAR(50),
    model VARCHAR (50),
    Year INT
);
```

```
create TAble Drivers (
    Driver_ID INT PK,
    Name VARCHAR (100),
    License_NO VARCHAR(50),
    contact_INFO VARCHAR (200).
    );
```

```
create table trips (
    TRIP_ID INT PK,
    vehicle_ID INT,
    DRIVER_ID INT,
    start_date DATE,
    END_Date DATE,
    Distance DECIMAL (10,2),
    FOREIGN key(vehicle_ID )
REFERENCES vehicles(vehicle_ID,
    FOREIGN KEY (DRIVER ID) RE-
FERENCES vehicles (vehicle_ID)
);
```

```
create table fuel_log (
    fuel_ID INT PK,
    vehicle_ID INT
    Date DATE,
    fuel_Qty DECIMAL(10,2),
    fuel_type VARCHAR(50),
    FOREIGN KEY (vehicle_ID)
REFERENCES vehicles(vehicle_ID)
);
```

```
create table maintenance (
    maintenance_ID INT PK,
    vehicle_ID INT,
    service_Date DATE,
    service_TYPE VARCHAR(100),
    cost DECIMAL (10,2),
    FOREIGN KEY (vehicle_ID) REFERENCES
    vehicles (vehicle_ID)
);
```

1. Tables and Relationships:

- Vehicles Table: This table holds data on each vehicle in the fleet, including vehicle ID, model, make, year, capacity and status. Each vehicle is uniquely identified and can be linked to various trips and maintenance records.
- Drivers Table: contains driver ID, name, license number, contact information, and status. Each driver can be associated with multiple trips, with a relationship linking trips and drivers.
- Trips Table: Manages trip-related information such as trip ID, vehicle ID, driver ID, start and end locations, start and end times, trip distance, and trip status.
- Maintenance Table: maintenance records with maintenance ID, vehicle ID, maintenance date, type of maintenance and notes.
- fuel usage Table: contains fuel usage records with fuel ID, vehicle ID, trip ID, date, fuel quantity, and fuel cost.

2. Stored procedures:

- Track Trip Completion: A stored procedure to mark trips as completed, updating relevant fields in the trip record and logging the completion time.
- Calculate fuel consumption: This stored procedure can take the vehicle ID and trip ID, calculate fuel consumed per trip, and update fuel efficiency metrics. This allows the fleet manager to asses vehicle performance and adjust as needed.

- schedule maintenance: A procedure to determine the next maintenance date based on mileage or time since the last service. This procedure might run periodically to check if any vehicle requires maintenance soon and flag those records.

(3) Queries :-

- fuel Efficiency Reports: A query to calculate the average fuel consumption of each vehicle over time, providing insight into which vehicles are the most efficient. This can involve taking trip distance and total fuel used to calculate mileage per gallon or kilo meter.
- Maintenance schedules: A query that lists vehicles that are due for maintenance soon, based on mileage or date criteria. This helps the fleet manager keep up with preventative maintenance, reducing unexpected breakdowns.
- Driver Activity Logs: A query to generate logs of driver activities. detailing trips taken, distances driven and hours worked. This is valuable for evaluating driver performance and workload.

4. Triggers:

- Update Maintenance Records: A trigger on the maintenance table could automatically update a "last serviced" date in the vehicles table whenever a new maintenance record is added.

- Alert for overdue services: A trigger can check for overdue maintenance each time a vehicle's mileage is updated. If a vehicle is past due for service based on mileage or time, the trigger could send an alert to the fleet manager, remaining them to schedule maintenance.

This database design enables effective tracking and management of fleet operations, providing valuable insights into vehicle performance, driver activity, and cost control, while ensuring maintenance and fuel efficiency are closely monitored.