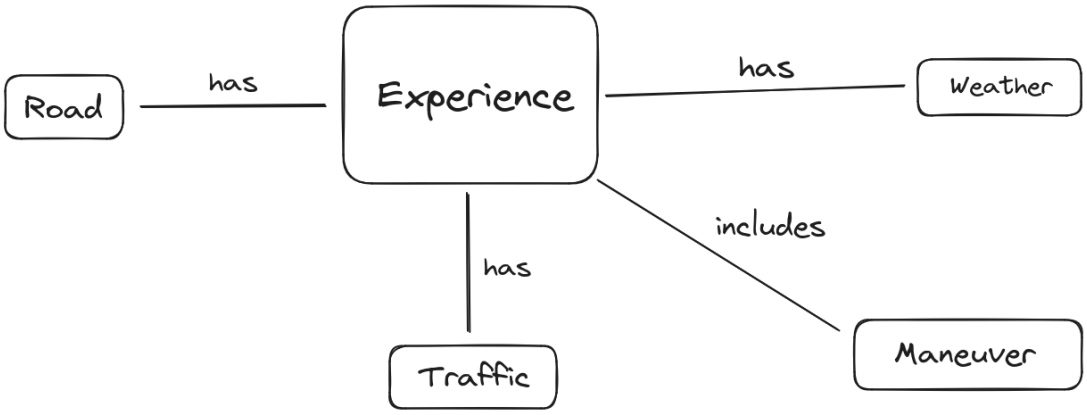
Databases Homework Project by Alikamran Rzayev

Step 1: Guidelines, rules, and CDM

Entities

- Experience
- Weather
- Road
- Traffic
- Maneuver

CDM



CDM in text format

[Experience](has)[Weather]
[Experience](has)[Road]
[Experience](has)[Traffic]
[Experience](includes)[Maneuver]

Information System Rules

A driving experience has one and only one weather type
A weather has 0 or many driving experiences

A driving experience has one and only one road type

A road type has 0 or many driving experiences

A driving experience has one and only one traffic type
A traffic type has 0 or many driving experiences

A driving experience has 0 or many Maneuvers
A maneuver has 0 or many driving experiences

Step 2: Data Dictionary and LDM

Data Dictionary

Attribute/Experience	Description	Type	Size
idExperience	Unique identifier	INT	4 bytes
date	Date at which experience was	DATE	3 bytes
departureTime	Time of experience departure	TIME	3 bytes
arrivalTime	Time of experience arrival	TIME	3 bytes
distance	Distance travelled in kilometers	FLOAT	8 bytes

Attribute/Traffic	Description	Type	Size
idTraffic	Unique identifier	INT	4 bytes
traffic	Label for traffic type	VARCHAR	50 chars

Attribute/Weather	Description	Type	Size
idWeather	Unique identifier	INT	4 bytes
weather	Label for weather type	VARCHAR	50 chars

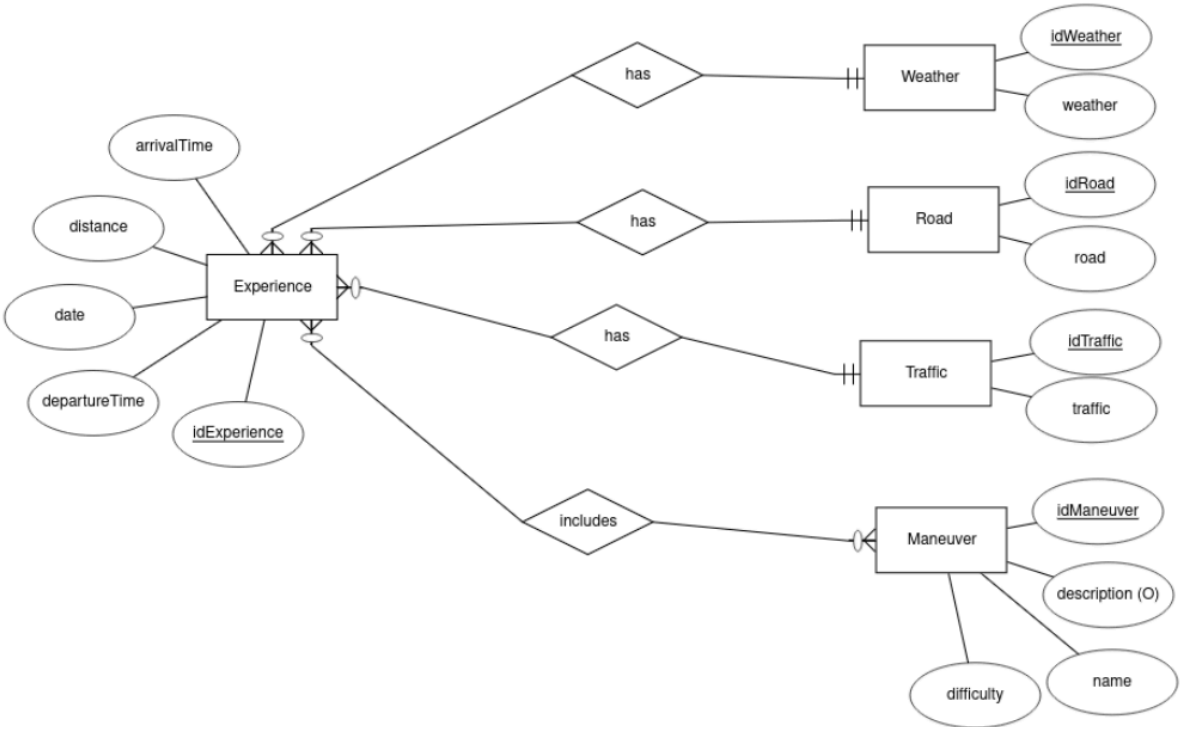
Attribute/Maneuver	Description	Type	Size
idManeuver	Unique identifier	INT	4 bytes
name	Name of maneuver	VARCHAR	50 chars
description	Detailed description of maneuver	VARCHAR	255 chars
difficulty	Difficulty of maneuver	TINYINT	1 bytes

Attribute/Road	Description	Type	Size
idRoad	Unique identifier	INT	4 bytes
road	Label for road type	VARCHAR	50 chars

Data Dictionary in text format

```
[Experience](idExperience, date, departureTime, arrivalTime,
distance)<INT, DATE, TIME, TIME, FLOAT>

[Weather](idWeather, weather)<INT, VARCHAR 50>

[Road](idRoad, road)<INT, VARCHAR 50>

[Traffic](idTraffic, traffic)<INT, VARCHAR 50>

[Maneuver](idManeuver, name, description, difficulty)<INT, VARCHAR 50,
VARCHAR 255, TINYINT>
```

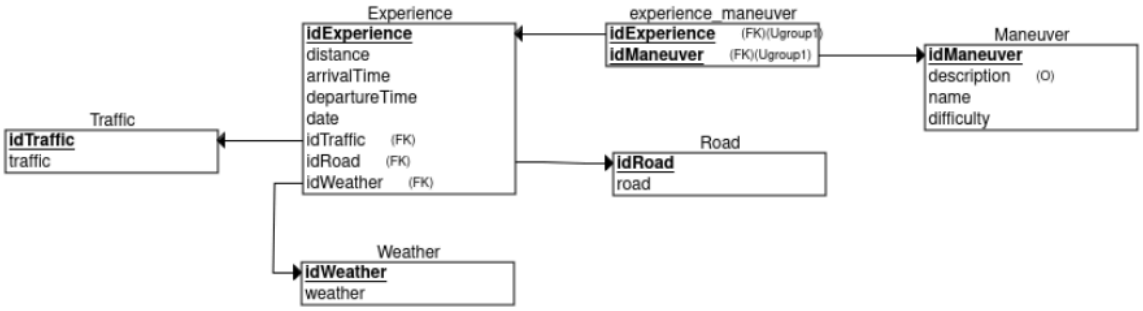
LDM

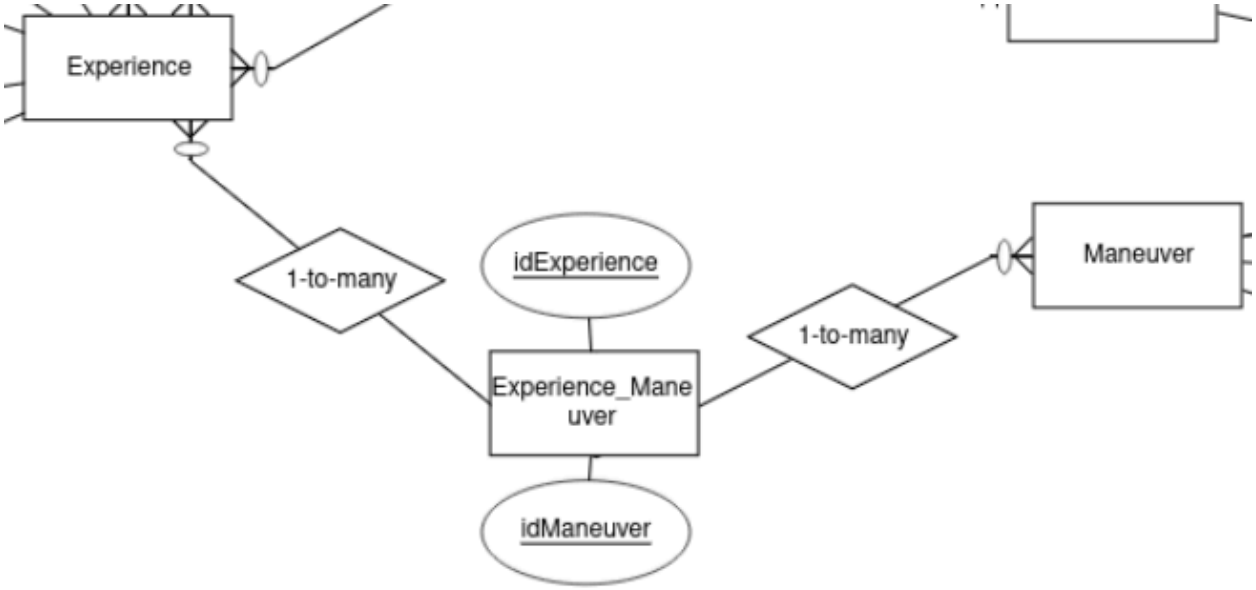


LDM in text format

[Experience](>011)[Weather](1-to-many)
[Experience](>011)[Road](1-to-many)
[Experience](>011)[Traffic](1-to-many)
[Experience](>00<)[Maneuver](many-to-many)

Step 3: PDM





PDM in text format

[Experience, PK: idExperience, FK: idWeather](>011)[Weather, PK: idWeather](1-to-many)
[Experience, PK: idExperience, FK: idRoad](>011)[Road, PK: idRoad](1-to-many)
[Experience, PK: idExperience, FK: idTraffic](>011)[Traffic, PK: idTraffic](1-to-many)

[Experience, PK: idExperience](110<)[Experience_Maneuver, FK: idExperience, FK: idManeuver]
[Experience_Maneuver, FK: idExperience, FK: idManeuver](>011)[Maneuver, PK: idManeuver]

Schema in text format

Experience (PK: idExperience, distance, arrivalTime, departureTime, date, FK: idWeather, FK: idRoad, FK: idTraffic)
Weather (PK: idWeather, weather)
Road (PK: idRoad, road)
Traffic (PK: idTraffic, traffic)
Maneuver (PK: idManeuver, name, description, difficulty)
Experience_Maneuver (FK: idExperience, FK: idManeuver)

Step 4

MySQL query to create the database structure

```
CREATE TABLE Weather
(
  idWeather INT NOT NULL,
  weather VARCHAR(50) NOT NULL,
```

```
  PRIMARY KEY (idWeather)
);

CREATE TABLE Road
(
  idRoad INT NOT NULL,
  road VARCHAR(50) NOT NULL,
  PRIMARY KEY (idRoad)
);

CREATE TABLE Traffic
(
  idTraffic INT NOT NULL,
  traffic VARCHAR(50) NOT NULL,
  PRIMARY KEY (idTraffic)
);

CREATE TABLE Maneuver
(
  idManeuver INT NOT NULL,
  description VARCHAR(255),
  name VARCHAR(50) NOT NULL,
  difficulty TINYINT NOT NULL,
  PRIMARY KEY (idManeuver)
);

CREATE TABLE Experience
(
  distance FLOAT NOT NULL,
  arrivalTime TIME NOT NULL,
  departureTime TIME NOT NULL,
  date DATE NOT NULL,
  idExperience INT NOT NULL,
  idTraffic INT NOT NULL,
  idRoad INT NOT NULL,
  idWeather INT NOT NULL,
  PRIMARY KEY (idExperience),
  FOREIGN KEY (idTraffic) REFERENCES Traffic(idTraffic),
  FOREIGN KEY (idRoad) REFERENCES Road(idRoad),
  FOREIGN KEY (idWeather) REFERENCES Weather(idWeather)
);

CREATE TABLE Experience_Maneuver
(
  idExperience INT NOT NULL,
  idManeuver INT NOT NULL,
  PRIMARY KEY (idExperience, idManeuver),
  FOREIGN KEY (idExperience) REFERENCES Experience(idExperience),
  FOREIGN KEY (idManeuver) REFERENCES Maneuver(idManeuver),
  UNIQUE (idExperience, idManeuver)
);
```

Queries

Show maneuvers performed in each experience

This query aims to retrieve all maneuvers performed in each driving experience along with other relevant details such as distance, arrival time, departure time, date, weather, road, and traffic conditions.
MySQL syntax explanation:

- `JOIN`: Joins the "Experience" table with "Weather", "Road", "Traffic", "Experience_Maneuver", and "Maneuver" tables based on their respective foreign keys.
- `GROUP_CONCAT`: Concatenates the names of maneuvers performed in each experience separated by a comma.
- `GROUP BY`: Groups the result set by the unique identifier of each experience ("idExperience").

```
SELECT
        e.idExperience,
        e.distance,
        e.arrivalTime,
        e.departureTime,
        e.date,
        w.weather,
        r.road,
        t.traffic,
        GROUP_CONCAT(m.name ORDER BY m.name SEPARATOR ', ') AS maneuvers_performed
FROM Experience e
JOIN Weather w ON e.idWeather = w.idWeather
JOIN Road r ON e.idRoad = r.idRoad
JOIN Traffic t ON e.idTraffic = t.idTraffic
JOIN Experience_Maneuver em ON e.idExperience = em.idExperience
JOIN Maneuver m ON em.idManeuver = m.idManeuver
GROUP BY e.idExperience;
```

idExperience	distance	arrivalTime	departureTime	date	weather	road	traffic	maneuvers_performed
1	20.5	08:30:00	08:00:00	2024-04-01	Cloudy	Highway	Heavy	Lane Change
2	15.2	12:00:00	11:30:00	2024-04-05	Sunny	City Street	Moderate	Left Turn
3	30.7	17:45:00	17:00:00	2024-04-10	Rainy	Mountain Road	Standstill	Highway Merge, Lane Change
4	10.3	09:15:00	09:00:00	2024-04-15	Sunny	Country Road	Light	Parallel Parking
5	25.8	14:20:00	13:45:00	2024-04-20	Foggy	Coastal Road	Moderate	Lane Change, U-turn
6	18.9	10:45:00	10:15:00	2024-04-25	Cloudy	City Street	Moderate	Parallel Parking
7	40.2	16:30:00	15:45:00	2024-05-02	Rainy	Highway	Heavy	Highway Merge, Left Turn
8	12.8	11:10:00	10:45:00	2024-05-05	Sunny	City Street	Light	Lane Change
9	28.5	13:40:00	13:00:00	2024-05-10	Snowy	Coastal Road	Standstill	Lane Change, U-turn
10	22.3	08:20:00	07:45:00	2024-05-15	Cloudy	Country Road	Heavy	Left Turn, Parallel Parking

Show the total number of experiences for each maneuver

This query calculates the total number of times each maneuver has been performed across all experiences.

MySQL syntax explanation:

- `LEFT JOIN`: Joins the "Maneuver" table with the "Experience_Maneuver" table using a left join to ensure that all maneuvers are included in the result set, even if they haven't been performed in any experience.
- `COUNT`: Counts the occurrences of each maneuver in the "Experience_Maneuver" table.
- `GROUP BY`: Groups the result set by the unique identifier of each maneuver ("idManeuver").

```
SELECT
    m.name AS maneuver_name,
    COUNT(em.idExperience) AS total_experiences
FROM
    Maneuver m
LEFT JOIN
    Experience_Maneuver em ON m.idManeuver = em.idManeuver
GROUP BY
    m.idManeuver;
```

maneuver_name	total_experiences
Lane Change	5
U-turn	2
Parallel Parking	3
Highway Merge	2
Left Turn	3

Calculate the total distance traveled on each road

This query computes the total distance traveled on each road, aggregating the distances from all driving experiences on each road.
MySQL syntax explained:

- `JOIN`: Joins the "Experience" table with the "Road" table based on their respective foreign keys.
- `SUM`: Calculates the total distance traveled on each road by summing up the distances from all experiences.
- `GROUP BY`: Groups the result set by the unique identifier of each road ("idRoad").

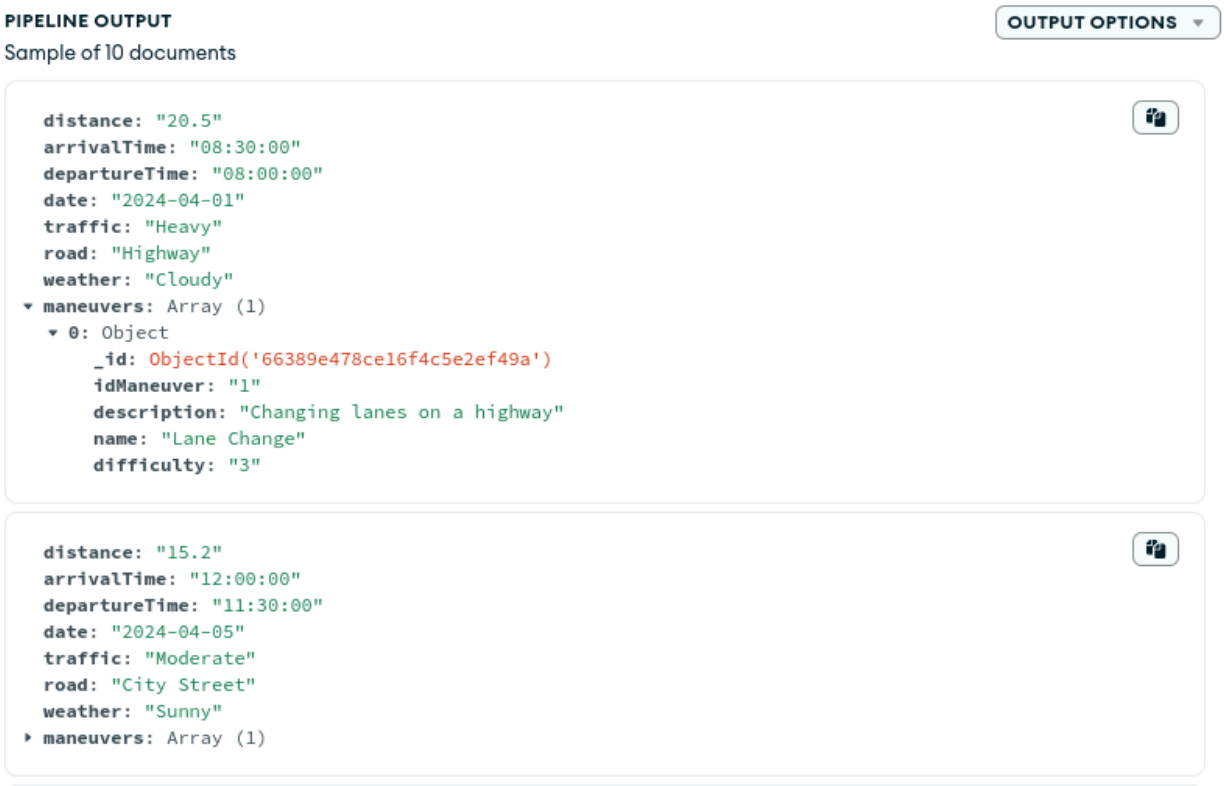
```
SELECT
       r.road,
       SUM(e.distance) AS total_distance
FROM Experience e
JOIN Road r ON e.idRoad = r.idRoad
GROUP BY r.idRoad;
```

```

| road | total_distance |
|---|---|
| Highway | 60.70000076293945 |
| City Street | 46.89999961853027 |
| Country Road | 32.59999942779541 |
| Mountain Road | 30.700000762939453 |
| Coastal Road | 54.29999923706055 |

### Determine the average difficulty level of maneuvers performed in each experience

This query calculates the average difficulty level of maneuvers performed in each driving experience.
MySQL syntax explained:

- `JOIN`: Joins the "Experience" table with the "Experience_Maneuver" table and the "Maneuver" table based on their respective foreign keys.
- `AVG`: Computes the average difficulty level of maneuvers performed in each experience.
- `GROUP BY`: Groups the result set by the unique identifier of each experience ("idExperience").

```
```
SELECT
	e.idExperience,
	AVG(m.difficulty) AS average_difficulty
FROM Experience e
JOIN Experience_Maneuver em ON e.idExperience = em.idExperience
JOIN Maneuver m ON em.idManeuver = m.idManeuver
GROUP BY e.idExperience;
```
```
```

idExperience	average_difficulty
1	3.0000
2	4.0000
3	2.5000
4	4.0000
5	4.0000
6	4.0000
7	3.0000
8	3.0000
9	4.0000
10	4.0000

Part 2: MongoDB Denormalized Database

We'll denormalize the MySQL database into a MongoDB database using an aggregation pipeline. Transforming a normalized structure into a denormalized structure means related information is merged into a single document for each experience. The following is the sample output of the aggregation pipeline:



Detailed explanation of each stage of the pipeline:

1. **$lookup (Traffic):**
    - This stage performs a lookup in the "Traffic" collection based on the "idTraffic" field in the "Experience" collection.
    - It retrieves documents from the "Traffic" collection where the "idTraffic" matches the "idTraffic" in the "Experience" collection.
    - The result is stored in an array field called "traffic".
2. **$unwind ("traffic"):**
    - Since the previous stage returns an array ("traffic"), this stage unwinds the array, creating a separate document for each element in the array.
    - This is necessary to continue processing each document individually in the pipeline.
3. **$lookup (Road):**
    - Similar to the first $lookup stage, this stage performs a lookup in the "Road" collection based on the "idRoad" field in the "Experience" collection.
    - It retrieves documents from the "Road" collection where the "idRoad" matches the "idRoad" in the "Experience" collection.
    - The result is stored in an array field called "road".
4. **$unwind ("road"):**
    - Again, this stage unwinds the array field "road" to create separate documents for each element.
5. **$lookup (Weather):**
    - This stage performs a lookup in the "Weather" collection based on the "idWeather" field in the "Experience" collection.
    - It retrieves documents from the "Weather" collection where the "idWeather" matches the "idWeather" in the "Experience" collection.
    - The result is stored in an array field called "weather".
6. **$unwind ("weather"):**
    - Similar to previous unwind stages, this stage unwinds the array field "weather".
7. **$lookup (Experience_Maneuver):**
    - This stage performs a lookup in the "Experience_Maneuver" collection based on the "idExperience" field in the "Experience" collection.
    - It retrieves documents from the "Experience_Maneuver" collection where the "idExperience" matches the "idExperience" in the "Experience" collection.
    - The result is stored in an array field called "maneuvers".
8. **$lookup (Maneuver):**
    - This stage performs a lookup in the "Maneuver" collection based on the "idManeuver" field in the documents of the "maneuvers" array.
    - It retrieves documents from the "Maneuver" collection where the "idManeuver" matches the "idManeuver" in the "maneuvers" array.
    - The result is stored in an array field called "maneuverDetails".
9. **$project:**

- This stage reshapes the documents by including or excluding fields as needed.
- Here, we keep fields like "distance", "arrivalTime", "departureTime", "date", and extract values from the lookup results for "traffic", "road", "weather", and "maneuverDetails".
- The result is a denormalized document containing all relevant information for each experience.

The code for the aggregation pipeline in full is given in Appendix C.

Appendix A: Sample data queries for MySQL

```
```
-- Weather table
INSERT INTO Weather (idWeather, weather)
VALUES
(1, 'Sunny'),
(2, 'Cloudy'),
(3, 'Rainy'),
(4, 'Foggy'),
(5, 'Snowy');

-- Road table
INSERT INTO Road (idRoad, road)
VALUES
(1, 'Highway'),
(2, 'City Street'),
(3, 'Country Road'),
(4, 'Mountain Road'),
(5, 'Coastal Road');

-- Traffic table
INSERT INTO Traffic (idTraffic, traffic)
VALUES
(1, 'Light'),
(2, 'Moderate'),
(3, 'Heavy'),
(4, 'Standstill');

-- Maneuver table
INSERT INTO Maneuver (idManeuver, description, name, difficulty)
VALUES
(1, 'Changing lanes on a highway', 'Lane Change', 3),
(2, 'Making a U-turn at an intersection', 'U-turn', 5),
(3, 'Parallel parking on a busy street', 'Parallel Parking', 4),
(4, 'Merging onto a highway from a ramp', 'Highway Merge', 2),
(5, 'Making a left turn on a busy intersection', 'Left Turn', 4);

-- Experience table
INSERT INTO Experience (distance, arrivalTime, departureTime, date, idExperience, idTraffic,
idRoad, idWeather)
VALUES
```
```

```
(20.5, '08:30:00', '08:00:00', '2024-04-01', 1, 3, 1, 2),
(15.2, '12:00:00', '11:30:00', '2024-04-05', 2, 2, 2, 1),
(30.7, '17:45:00', '17:00:00', '2024-04-10', 3, 4, 4, 3),
(10.3, '09:15:00', '09:00:00', '2024-04-15', 4, 1, 3, 1),
(25.8, '14:20:00', '13:45:00', '2024-04-20', 5, 2, 5, 4);

-- Experience_Maneuver table
INSERT INTO Experience_Maneuver (idExperience, idManeuver)
VALUES
(1, 1),
(2, 5),
(3, 4),
(3, 1),
(4, 3),
(5, 2),
(5, 1);

-- Additional Experience records
INSERT INTO Experience (distance, arrivalTime, departureTime, date, idExperience, idTraffic,
idRoad, idWeather)
VALUES
(18.9, '10:45:00', '10:15:00', '2024-04-25', 6, 2, 2, 2),
(40.2, '16:30:00', '15:45:00', '2024-05-02', 7, 3, 1, 3),
(12.8, '11:10:00', '10:45:00', '2024-05-05', 8, 1, 2, 1),
(28.5, '13:40:00', '13:00:00', '2024-05-10', 9, 4, 5, 5),
(22.3, '08:20:00', '07:45:00', '2024-05-15', 10, 3, 3, 2);

-- Additional entries in the Experience_Maneuver table
INSERT INTO Experience_Maneuver (idExperience, idManeuver)
VALUES
(6, 3), -- Parallel Parking
(7, 4), -- Highway Merge
(7, 5), -- Left Turn
(8, 1), -- Lane Change
(9, 2), -- U-turn
(9, 1), -- Lane Change
(10, 5), -- Left Turn
(10, 3); -- Parallel Parking
```

Appendix B: Sample data queries for MongoDB

```
// Import JSON data into MongoDB collections
db.createCollection("Weather");
db.createCollection("Road");
db.createCollection("Traffic");
db.createCollection("Maneuver");
db.createCollection("Experience");
db.createCollection("Experience_Maneuver");
```

```
// Insert JSON data into corresponding collections
db.Weather.insertMany([
  {"idWeather":"1","weather":"Sunny"},
  {"idWeather":"2","weather":"Cloudy"},
  {"idWeather":"3","weather":"Rainy"},
  {"idWeather":"4","weather":"Foggy"},
  {"idWeather":"5","weather":"Snowy"}
]);

db.Road.insertMany([
  {"idRoad":"1","road":"Highway"},
  {"idRoad":"2","road":"City Street"},
  {"idRoad":"3","road":"Country Road"},
  {"idRoad":"4","road":"Mountain Road"},
  {"idRoad":"5","road":"Coastal Road"}
]);

db.Traffic.insertMany([
  {"idTraffic":"1","traffic":"Light"},
  {"idTraffic":"2","traffic":"Moderate"},
  {"idTraffic":"3","traffic":"Heavy"},
  {"idTraffic":"4","traffic":"Standstill"}
]);

db.Maneuver.insertMany([
  {"idManeuver":"1","description":"Changing lanes on a highway","name":"Lane
Change","difficulty":"3"},
  {"idManeuver":"2","description":"Making a U-turn at an
intersection","name":"U-turn","difficulty":"5"},
  {"idManeuver":"3","description":"Parallel parking on a busy street","name":"Parallel
Parking","difficulty":"4"},
  {"idManeuver":"4","description":"Merging onto a highway from a ramp","name":"Highway
Merge","difficulty":"2"},
  {"idManeuver":"5","description":"Making a left turn on a busy intersection","name":"Left
Turn","difficulty":"4"}
]);

db.Experience.insertMany([

{"distance":"20.5","arrivalTime":"08:30:00","departureTime":"08:00:00","date":"2024-04-01","idExp
erience":"1","idTraffic":"3","idRoad":"1","idWeather":"2"},

{"distance":"15.2","arrivalTime":"12:00:00","departureTime":"11:30:00","date":"2024-04-05","idExp
erience":"2","idTraffic":"2","idRoad":"2","idWeather":"1"},

{"distance":"30.7","arrivalTime":"17:45:00","departureTime":"17:00:00","date":"2024-04-10","idExp
erience":"3","idTraffic":"4","idRoad":"4","idWeather":"3"},

{"distance":"10.3","arrivalTime":"09:15:00","departureTime":"09:00:00","date":"2024-04-15","idExp
erience":"4","idTraffic":"1","idRoad":"3","idWeather":"1"},

{"distance":"25.8","arrivalTime":"14:20:00","departureTime":"13:45:00","date":"2024-04-20","idExp
erience":"5","idTraffic":"2","idRoad":"5","idWeather":"4"},
```

```
{"distance":"18.9","arrivalTime":"10:45:00","departureTime":"10:15:00","date":"2024-04-25","idExp
erience":"6","idTraffic":"2","idRoad":"2","idWeather":"2"},

{"distance":"40.2","arrivalTime":"16:30:00","departureTime":"15:45:00","date":"2024-05-02","idExp
erience":"7","idTraffic":"3","idRoad":"1","idWeather":"3"},

{"distance":"12.8","arrivalTime":"11:10:00","departureTime":"10:45:00","date":"2024-05-05","idExp
erience":"8","idTraffic":"1","idRoad":"2","idWeather":"1"},

{"distance":"28.5","arrivalTime":"13:40:00","departureTime":"13:00:00","date":"2024-05-10","idExp
erience":"9","idTraffic":"4","idRoad":"5","idWeather":"5"},

{"distance":"22.3","arrivalTime":"08:20:00","departureTime":"07:45:00","date":"2024-05-15","idExp
erience":"10","idTraffic":"3","idRoad":"3","idWeather":"2"}
]);

db.Experience_Maneuver.insertMany([
  {"idExperience":"1","idManeuver":"1"},
  {"idExperience":"2","idManeuver":"5"},
  {"idExperience":"3","idManeuver":"1"},
  {"idExperience":"3","idManeuver":"4"},
  {"idExperience":"4","idManeuver":"3"},
  {"idExperience":"5","idManeuver":"1"},
  {"idExperience":"5","idManeuver":"2"},
  {"idExperience":"6","idManeuver":"3"},
  {"idExperience":"7","idManeuver":"4"},
  {"idExperience":"7","idManeuver":"5"},
  {"idExperience":"8","idManeuver":"1"},
  {"idExperience":"9","idManeuver":"1"},
  {"idExperience":"9","idManeuver":"2"},
  {"idExperience":"10","idManeuver":"3"},
  {"idExperience":"10","idManeuver":"5"}
]);
```

Appendix C: Aggregation pipeline for denormalization

```
[
  {
        $lookup: {
        from: "Traffic",
        localField: "idTraffic",
        foreignField: "idTraffic",
        as: "traffic",
        },
  },
  {
        $unwind: "$traffic",
  },
```

```
{
      $lookup: {
      from: "Road",
      localField: "idRoad",
      foreignField: "idRoad",
      as: "road",
      },
},
{
      $unwind: "$road",
},
{
      $lookup: {
      from: "Weather",
      localField: "idWeather",
      foreignField: "idWeather",
      as: "weather",
      },
},
{
      $unwind: "$weather",
},
{
      $lookup: {
      from: "Experience_Maneuver",
      localField: "idExperience",
      foreignField: "idExperience",
      as: "maneuvers",
      },
},
{
      $lookup: {
      from: "Maneuver",
      localField: "maneuvers.idManeuver",
      foreignField: "idManeuver",
      as: "maneuverDetails",
      },
},
{
      $project: {
      _id: 0,
      distance: 1,
      arrivalTime: 1,
      departureTime: 1,
      date: 1,
      traffic: "$traffic.traffic",
      road: "$road.road",
      weather: "$weather.weather",
      maneuvers: "$maneuverDetails",
      },
},
{
      $out:
```

```
        "Denormalized_Experience",
  },
]
```