



## **DATABASE MANAGEMENT SYSTEMS**

**SWER351**

**DriveSmart Insurance Company Database Project**

**Done By:**

**Emran Alheeh**

**Feras Bannoura**

**Marah Jabr**

**Supervised By:**

**Dr.Mohammed Sharaf**

**SWER351 Project**



## **Introduction:**

DriveSmart is a car insurance company that provides insurance services on all types of vehicles. In this project, we implemented a database design for the company that would reflect their new business model, which is based on monitoring drivers and rewarding them those whose driving is safe, the database contains six tables ,

1. Customer : to store customer information like name ,email address and phone number so it helps the company to contact the customer when they need.

Customer table has Cus\_ID as a primary key

2. Vehicle: contains information about vehicle and the tracking device information

Why do we need a Tracking Device in a vehicle?

The insurance company tracks their customers to see their trips , so they can know their total trips , and give them a score ,so they manage customer driving habits ,so they advise bad drivers to improve their driving habits and thank them also they give them a special reward .

Vehicle table has vehicle\_ID as primary key

has Cus\_ID as foreign key

3. InsurancePolicy :this Table contains the insurance policy information like type of insurance , premium cost ,start date, and expiry date of current policy.

Insurance Policy table has ins\_ID as primary key

Cus\_ID , Veh\_ID as foreign key

4. DailyScore: every customer has three individual scores in every day : speed score ,braking score , night driving score then they take the average of these 3 scores and give the customer daily Score . Customer sees his Score in Mobile application that is connected with the database .

DailyScore Table has Score\_ID as primary key

Veh\_ID as foreign key

5. Trip : stores the trips done by customer , length of trips and time of trips

Trip table has Trip\_ID as primary Key

Veh\_ID as foreign

6. Speed events :It stores the violation of Speed : location ,speed limit , and actual speed

Speed events table has speed\_event\_ID as primary Key

Veh\_ID as forign Key

## **THE ENTITY RELATIONSHIP DIAGRAM**

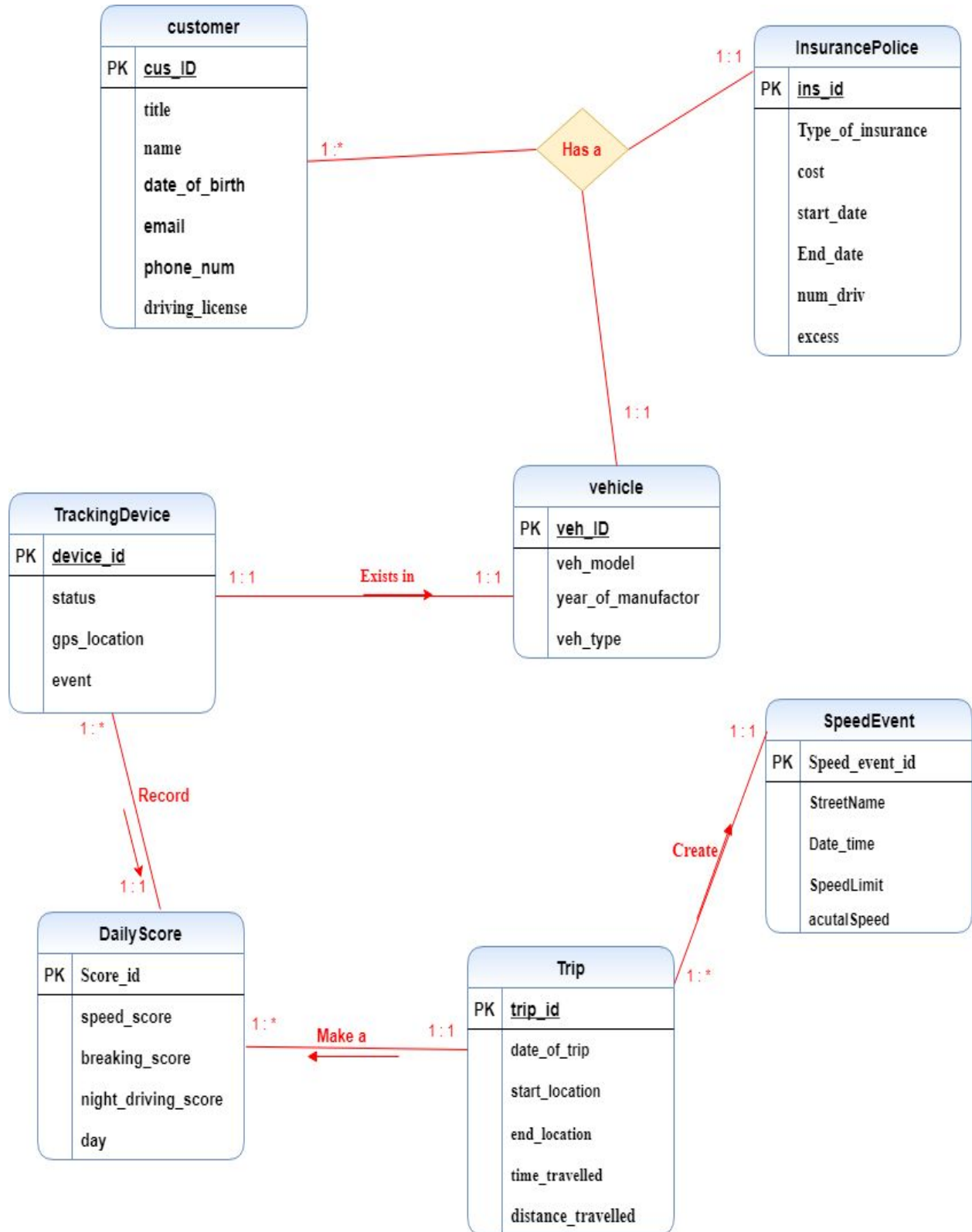
An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database.

An entity in this context is an object, a component of data. An entity set is a collection of similar entities. These entities can have attributes that define its properties.

By defining the entities, their attributes, and showing the relationships between them, an ER diagram illustrates the logical structure of databases.

ER diagrams are used to sketch out the design of a database.

### **INSURANCE COMPANY ER DIAGRAM :**



## Schema

### 1. Customer(

```
cus_ID      CHAR(10),
title       VARCHAR(255),
name        VARCHAR(255),
date_of_birth  date,
email       VARCHAR(255),
phone_num    CHAR(10),
driving_license  VARCHAR(255),
PRIMARY KEY (cus_ID)
);
```

### 2. Vehicle(

```
cus_ID      CHAR(10),
ins_ID       CHAR(10),
veh_ID       CHAR(10),
veh_model    VARCHAR(255),
year_of_manufacture  VARCHAR(255),
veh_type     VARCHAR(255),
status       VARCHAR(255),
gps_location VARCHAR(255),
event        VARCHAR(255),
PRIMARY KEY (veh_ID),
FOREIGN KEY (cus_ID ) REFERENCES customer,
);
```

### 3. InsurancePolicies(

```
cus_ID      CHAR(10),
ins_ID       CHAR(10),
veh_ID       CHAR(10),
Type_of_insurance  VARCHAR(255),
cost         NUMERIC(8,2),
start_date   date,
End_date     date,
num_driv     char(5),
excess       VARCHAR(255)
PRIMARY KEY (ins_ID),
```

```

FOREIGN KEY (cus_ID ) REFERENCES customer,
FOREIGN KEY (veh_ID ) REFERENCES Vehicle
);

```

**4.** DailyScore (

```

    Score_id          int,
    veh_ID            CHAR(10),
    speed_score       NUMERIC(8,2),
    breaking_score    NUMERIC(8,2),
    night_driving_score NUMERIC(8,2),
    day              date,
    PRIMARY KEY (score_ID),
    FOREIGN KEY (veh_ID ) REFERENCES Vehicle,
);

```

**5.** Trip (

```

    Trip_id          int,
    veh_ID          CHAR(10),
    date_of_trip     DATE,
    start_location   VARCHAR(255),
    end_location     VARCHAR(255),
    time_travelled   TIME,
    distance_travelled NUMERIC(8,2)
    PRIMARY KEY (trip_ID),
    FOREIGN KEY (veh_ID ) REFERENCES vehicle,
);

```

**6.** SpeedEvents (

```

    Speed_event_id    int,
    veh_ID            CHAR(10),
    StreetName        VARCHAR(255),
    Date_Time         datetime,
    SpeedLimit        NUMERIC(8,2),
    actualSpeed       NUMERIC(8,2),
    FOREIGN KEY (veh_ID ) REFERENCES vehicle,
);

```

### List of assumptions

- Each customer MAY **have** more than one insurance policy for more than one vehicle.
- In each vehicle, there **exists** a tracking device so that their driving is continuously monitored.
- Each vehicle has a tracking device which **records** the customer movement information.
- Each score was recorded for customer trips. Therefore, these trips **make** the daily score.
- The relationship between vehicle and tracking device is 1 to 1 so we put them in one table , because tracking device is in car so we make one table vehicle for vehicle and tracking device
- We connected insurance policy table ,vehicle , trip , speed events and daily score in one foreign key (VehicleID ) because vehicle make the trip , speed events ,and daily score ,also the insurance policy is for the vehicle

### REPORTING ASSUMPTIONS:

- For calculating score we assumed date of score so when we want to calculate score in specific month we put a condition that date of score is in this specific month.
- In report 2.4 the report should return the trips by customer who his gps locator is off, but how we can know trips if the gps locator is off , so we assumed that we can't know the trips ,when gps locator is off.



- in report 2.5 we gives all the scores for customer who achieve the report condition (Score >70 ) in last 2 months

## **Reports results & screenshot**

### **2.1. Renewals report:**

#### **Scenario For Report 2.1**

Feras Bannoura is a customer of DriveSmart Insurance Company. He owns a car as it was insured on 1/1/2020 for an annual amount of 1,500 shekels in exchange for comprehensive insurance for the car, which will expire at the end of this month. While Marah Jabr bought a new car and insured it for a period of six months in the company on 25/6/2020 at an amount of 1,000 shekels in exchange for the third-party insurance. Therefore, the insurance expires at the end of this month. In addition, Emran Alheeh renewed the insurance on his truck on 1/12/2020, with an annual amount of 7000 shekels in exchange for comprehensive insurance. Moreover, muhmod Ayash owns an old truck, which was insured for an annual amount of 7000 shekels on 23/8/2020 in exchange for comprehensive insurance. Mohammad also comprehensively insured his car, at an amount of 1,500 shekels on 15/8/2020.

The screenshot displays the Azure Data Studio interface. The main window shows a SQL query in the editor, which is executed against a local database. The query selects customer details and calculates an average score based on daily driving scores, filtered for high-risk drivers. The results are displayed in a table below the query.

```

1 SELECT c.title, c.name, c.date_of_birth, InsurancePolicies.End_date, InsurancePolicies.cost, AVG(DailyScore.speed_score+DailyScore.breaking_score+DailyScore.night_score) AS AvgScore
2 FROM Customer AS c INNER JOIN InsurancePolicies on c.cus_ID = InsurancePolicies.cus_ID
3 INNER JOIN Vehicle on c.cus_ID = Vehicle.cus_ID
4 INNER JOIN DailyScore on Vehicle.veh_ID = DailyScore.veh_ID
5 WHERE DailyScore.(day) IN (
6   SELECT MAX((day))
7   FROM DailyScore)
8 WHERE DailyScore.veh_ID = Vehicle.veh_ID AND InsurancePolicies.End_date BETWEEN GETDATE() and DATEADD(MONTH,1,GETDATE())
9
10 GROUP BY c.title, c.name, c.date_of_birth, InsurancePolicies.End_date, InsurancePolicies.cost
11 ORDER BY InsurancePolicies.End_date
12
13
14 -- Report 2.2
15 -- SELECT DISTINCT Customer.title, Customer.name, Vehicle.veh_ID, SpeedEventr.Date, Time, SpeedEventr.SpeedLimit, SpeedEventr.actualSpeed

```

	title	name	date_of_birth	End_date	cost	score
1	ms	Marah Jabr	2000-08-15	2020-12-25	1000.00	76.000000
2	mr	Feras Bannoura	2000-08-15	2021-01-01	1500.00	63.000000

## 2.2. High risk drivers report:

### Scenario For Report 2.2

Feras is a reckless driver, accustomed to driving quickly as he does not adhere to the speed limit on the roads. As he exceeded the speed limit 3 times during the last two weeks, on Al-Amal Street, where the speed limit was 40 km / h and his speed was 70 km/ h,\. The second time in Beit Sahour Street Where the specified speed was 70 km/h and its speed was 100 km/ h, and the third time on Al-Quds Street, where the specified speed was 50 km/ h and its speed was 70 km/h.

Marah is a good driver. But during the last two weeks she exceeded the speed limit twice and it was unintentionally, the first time in Al-Khader Street, where the speed limit was 40 km / h and the speed was 60 km/ h, and the second time in Al-Madbasa Street, where the specified speed was 30 km / h Its speed is 50 km / h.

The screenshot displays the Azure Data Studio interface. On the left, the 'SERVERS' pane shows a database structure with tables like InsurancePolicies, SpeedEvents, and Vehicle. The main editor shows a SQL query (SQLQuery\_2) that joins Customer, Vehicle, and SpeedEvents tables, filters for speed scores below 30, and groups the results by customer and vehicle. The 'Results' pane at the bottom shows the output of the query, which is a table with 5 rows and 6 columns: title, name, veh\_ID, Date\_Time, SpeedLimit, and actualSpeed.

	title	name	veh_ID	Date_Time	SpeedLimit	actualSpeed
1	mr	Feras Bannoura	2000	2020-12-15 11:30:30.000	40.00	70.00
2	mr	Feras Bannoura	2000	2020-12-17 13:30:30.000	70.00	100.00
3	mr	Feras Bannoura	2000	2020-12-19 09:30:30.000	50.00	70.00
4	ms	Marah Jabr	3000	2020-12-15 13:30:30.000	40.00	60.00
5	ms	Marah Jabr	3000	2020-12-18 12:20:30.000	30.00	50.00

## 2.3. Customer trips report:

### Scenario For Report 2.3

Customer trips are recorded. For example, on 22/5/2020, Feras recorded a trip from Beit Sahour to Nablus, which lasted for 2:23h. On 12/7/2020, Marah recorded a trip from Al-Khader to Hebron, which lasted for 1:12h. On the same day, Feras recorded a trip from Bethlehem to Ramallah and lasted for 1:33h. In addition, Emran recorded a trip on 12/8/2020 from Sourif to Bethlehem, which lasted for 1:14h. etc

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Servers' tree is expanded to show the 'dbo.Vehicle' table. The main pane displays a SQL query and its results.

```

347 SELECT DISTINCT Customer.title, Customer.name, Vehicle.veh_ID, SpeedEvents.Date_Time, SpeedEvents.SpeedLimit, SpeedEvents.actualSpeed
348 FROM Customer
349 INNER JOIN Vehicle on Customer.cus_ID = Vehicle.cus_ID
350 INNER JOIN SpeedEvents on Vehicle.veh_ID = SpeedEvents.veh_ID
351 INNER JOIN DailyScore on Vehicle.veh_ID = DailyScore.veh_ID
352 WHERE DailyScore.speed_score < 30
353 GROUP BY Customer.title, Customer.name, Vehicle.veh_ID, SpeedEvents.Date_Time, SpeedEvents.SpeedLimit, SpeedEvents.actualSpeed
354 HAVING COUNT(DailyScore.speed_score) > 1
355
356 -- Report 2.3 query :
357 WITH
358     dailyScoreAvg
359 AS
360

```

The results table shows the following data:

	name	NumberOfTrips	TotalTraveledDistance	Month	timetotal	AvgScore
1	Feras Bannoura	3	110.00	December 2020	4:31	55.222222
2	Marah Jabr	3	75.00	December 2020	2:12	65.444444
3	Imran Al-heeh	3	119.00	December 2020	4:39	79.888888
4	Noor Heeh	1	40.00	December 2020	1:15	85.444444
5	Noor Heeh	1	40.00	November 2020	1:15	81.000000

## 2.4. Customers misusing tracking device report:

### Scenario For Report 2.4

Some customers try to cheat the tracking device by switching it off to avoid being tracked. Like mohammed who has recently disabled the tracking device, so it became difficult to record all his movement information.

File Edit View Help

• SqlFinalReport.sql - localhost:master (Integrated) - Azure Data Studio

CONNECTIONS

SERVERS

- Final
  - Tables
    - dbo.Customer
    - dbo.DailyScore
    - dbo.InsurancePolicies
  - Columns
    - cus\_ID (FK, char(10), null)
    - ins\_ID (PK, char(10), not null)
    - veh\_ID (FK, char(10), null)
    - Type\_of\_insurance (varchar(255), null)
    - cost (numeric(8,2), null)
    - start\_date (date, null)
    - End\_date (date, null)
    - num\_driv (char(5), null)
    - excess (varchar(255), null)
  - Keys
  - Constraints
  - Triggers
  - Indexes
  - Statistics
  - dbo.SpeedEvents
  - dbo.Trip
  - dbo.Vehicle
    - cus\_ID (FK, char(10), null)
    - ins\_ID (char(10), null)
    - veh\_ID (PK, char(10), not null)
    - veh\_model (varchar(255), null)
    - year\_of\_manufacture (varchar(255), null)
    - veh\_type (varchar(255), null)
    - status (varchar(255), null)

AZURE

SQL SERVER BIG DATA CLUSTERS

Welcome

schema.sql - localhost:grated

insert.sql - localhost:grated

reportQuery.sql - localhost:grated

SqlFinalReport.sql - localhost:grated

Run Cancel Disconnect Change Connection Final

Run SQLCMD Export as Notebook

```

375 From TripAvg, dailyScoreAvg, Vehicle INNER JOIN Customer on Customer.cus_ID = Vehicle.cus_ID
376 INNER JOIN InsurancePolicies on InsurancePolicies.veh_ID = Vehicle.veh_ID
377 WHERE TripAvg.veh_ID = Vehicle.veh_ID AND TripAvg.veh_ID = dailyScoreAvg.veh_ID AND TripAvg.Month = dailyScoreAvg.Month AND GETDATE() < InsurancePolicies.End_date
378
379
380
381
382 -- Report 2.4 query :
383
384 SELECT Customer.name, Customer.phone_num, Vehicle.[status]
385 FROM Customer INNER JOIN Vehicle on Customer.cus_ID = Vehicle.cus_ID
386 WHERE Vehicle.[status] = 'off'
387
388 -- Report 2.5 query :
389
390

```

Results Messages

	name	phone_num	status
1	MohamadSharaf	0592623001	off

Ln 387, Col 1 (163 selected) Spaces: 4 UTF-8 CRLF SQL Choose SQL Language 1 rows 00:00:00 localhost: Final

## 2.5. Customers Rewards report:

### Scenario For Report 2.5

noor heeh is a very good driver , in the last two months her average score is above 70.

Here we tried to show customers whose daily main scores stayed above 70 for the last two months.

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the database structure, including tables like `dbo.Customer`, `dbo.DailyScore`, and `dbo.Vehicle`. The right pane shows a SQL query and its results.

```
SELECT Customer.name NOT IN (
FROM Customer
INNER JOIN Vehicle on Customer.cus_ID = Vehicle.veh_ID
INNER join DailyScore on Vehicle.veh_ID = DailyScore.veh_ID
GROUP BY Customer.phone_num, Customer.name , Vehicle.veh_ID, DailyScore.speed_score,DailyScore.breaking_score,DailyScore.night_driving_score,DailyScore.[day]
HAVING ( AVG(DailyScore.speed_score+DailyScore.breaking_score+DailyScore.night_driving_score)/3) <70
GROUP BY Customer.phone_num, Customer.name , Vehicle.veh_ID, DailyScore.speed_score,DailyScore.breaking_score,DailyScore.night_driving_score,DailyScore.[day]
HAVING ( AVG(DailyScore.speed_score+DailyScore.breaking_score+DailyScore.night_driving_score)/3) >70
```

The Results pane shows the following data:

	name	phone_num	veh_ID	day	speed_score	breaking_score	night_driving_score	mainScore
1	Mahmoud Ayash	0592623001	1000	2020-12-15	75.00	78.00	60.00	71.000000
2	Noor Heeh	0592623301	6000	2020-11-20	75.00	88.00	80.00	81.000000
3	Noor Heeh	0592623301	6000	2020-12-20	75.00	88.00	80.00	81.000000
4	Noor Heeh	0592623301	6000	2020-12-18	85.00	88.00	90.00	87.666666
5	Noor Heeh	0592623301	6000	2020-12-15	95.00	88.00	80.00	87.666666