**Database Report**

**Student Banner ID’s:**

**B00189659 - Michael Clark**

**B00199011 - Ian Stevenson**

**B00187021 - Glen Jones**

**Table Of Contents**

[Assumptions 2](#_Toc290983843)

[ER Diagram 3](#_Toc290983844)

[Relational Schema 4](#_Toc290983845)

[Data Dictionary 5](#_Toc290983846)

[SQL Server Database and Tables 7](#_Toc290983847)

[Customer 7](#_Toc290983848)

[Job 7](#_Toc290983849)

[Driver 8](#_Toc290983850)

[Staff 8](#_Toc290983851)

[Taxi 8](#_Toc290983852)

[Service History 8](#_Toc290983853)

[Database Diagram 9](#_Toc290983854)

[SQL Server Database Queries 10](#_Toc290983855)

## Assumptions

1. A staff table should be included so there is information about the people who work for the company.
2. The members of staff need to record the details of all jobs.
3. A driver can own taxis but not drive them.
4. A driver can drive taxis but not own them.
5. All jobs are accepted.
6. Assuming that the driver’s picture will be included in the database.
7. Assumption that the 'appearance' of the taxi will be required.
8. A customer must be registered to order a taxi – they can’t get a taxi without being a member.

## ER Diagram

**Customer**

cID {PK}

cEmail

cPassword

cName

cTitle

cFirstName

cLastName

cGender

cDob

cAddress

cAddressLine1

cAddressLine2

cCity

cPostcode

cHomeNumber

cMobileNumber

/cFullName

**Driver**

dLicenceNumber {PK}

dName

dTitle

dFirstName

dLastName

dPicture

**Taxi**

tVehicleLicenceNumber {PK}

tVehicleMake

tVehicleModel

tPassengerCapacity

tStorageCapacity

tCurrentMileage

/tAppearance

**1..1**

**1..\***

**makesA**

**0..\***

**drivesA**

**1..1**

**acceptedBy**

**1..1**

**1..\***

**1..1**

**isOwnedBy**

**0..\***

**Staff**

sNumber {PK}

sName

sTitle

sFirstName

sLastName

sSalary

**records**

**1..\***

**1..1**

**ServiceHistory**

shNumber {PPK}

shDate

shMileage

shType

shDescription

**has**

**1..1**

**0..\***

**Job**

jID {PK}

jType

jNumberAdults

jNumberChildren

jLuggage

jSpecialReqs

jDiscount

jPickupAddress

jPickupAddressLine1

jPickupAddressLine2

jPickupCity

jPickupPostcode

jDropoffAddress

jDropoffAddressLine1

jDropoffAddressLine2

jDropoffCity

jDropoffPostcode

jPickupDate

jPickupTime

jMileageBefore

jMileageAfter

jPayment

jCancellationReason

/jTotalMileage

/jTotalPassengers

## Relational Schema

**Customer** (cID, cEmail, cPassword, cTitle, cFirstName,cLastName, cGender, cDob, cAddressLine1, cAddressLine2, cCity, cPostcode, cHomeNumber,cMobileNumber)

Primary Key: cID

Derived attribute:

**Job** (jID, jType, jNumberAdults, jNumberChildren, jLuggage, jSpecialReqs, jDiscount, jPickupAddressLine1, jPickupAddressLine2, jPickupCity, jPickupPostcode, jDropoffAddressLine1, jDropoffAddressLine2, jDropoffCity, jDropoffPostcode, jPickupDate, jPickupTime, jMileageBefore, jMileageAfter, jPayment,jCancellationReason, cID, dLicenceNumber, sNumber)

Primary Key:jID

Foreign key: dLicenceNumber **References** Driver (dLicenceNumber) **on delete**  **on update**

Foreign Key: cID **References** Customer (cID) **on** **delete** **on update**

Foreign Key: sNumber **References** Staff (sNumber) **on delete on update**

Derived attribute: jTotalMileage, jTotalPassengers

**Driver** (dLicenceNumber, dTitle, dFirstName, dLastName, dPicture, tVehicleLicenceNo)

Primary key dLicencenumber

Foreign key: tVehicleLicenceNo **References** Taxi (tVehicleLicenceNo) **on delete on update**

Derived attribute:

**Taxi** (tVehiclelicenceNo, tVehicleMake, tVehicleModel, tPassengerCapacity, tStorageCapacity, tCurrentMileage, dLicenceNumber)

Primary Key: tVehicleLicenceNumber

Foreign Key: dLicenceNumber **References** Driver (dLicenceNumber) **on delete on update**

Derived attribute: tAppearance

**ServiceHistory** (shNumber, shDate, shType, shDescription, tVehicleLicence/number)

Primary Key: shNumber, tVehicleLicenceNumber

Foreign Key: tVehicleLicenceNo **References** Taxi (tVehicleLicenceNo) **on delete on update**

Derived attribute:

**Staff** (sNumber, sTitle, sFirstName, sLastName, sSalary)

Primary Key: sNumber

Derived attribute

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Dictionary | | | | |
| **Attribute name** | **Description(meaning)** | **Data type (length)** | **Nulls** | **Constraint(PK,AK,FK)** |
| cID | Customers Identification | Nvarchar(8) | No | PK, FK |
| cEmail | Customers email | Nvarchar(30) | No |  |
| cPassword | Customers password | Nvarchar(12) | No |  |
| cFullName | Customers full name | Derived | No |  |
| cTitle | Customers Title | Nvarchar(15) | No |  |
| cFirstName | Customers first name | Nvarchar(50) | No |  |
| cLastName | Customers last name | Nvarchar(50) | No |  |
| cGender | Customers gender (Male or female) | Char(1) | No | M,F |
| cDob | Customers date of birth | Date | No |  |
| cAddressLine1 | Customer address line 1 | Nvarchar(50) | No |  |
| cAddressLine2 | Customer address line 2 | Nvarchar(50) | No |  |
| cCity | Customers city | Nvarchar(30) | No |  |
| cPostcode | Customers postcode | Nvarchar(9) | No |  |
| cHomeNumber | Customers home number | Nvarchar(11) | Yes |  |
| cMobileNumber | Customers Mobile number | Nvarchar(11) | Yes |  |
| jID | Job Identification | Nvarchar(8) | No | PK |
| jType | Job type (online or Telephone) | Char(1) | No | O/T |
| jTotalPassengers | Total number of passengers | Derived |  |  |
| jNumberOfAdults | Number of adults for the booking | Int | No |  |
| jNumberOfChildren | Number of children for the booking | Int | No |  |
| jLuggage | Customers luggage requirements | Int | No | Defaults to 0 |
| jSpecialReqs | Customers special requirements | Nvarchar(50) | No |  |
| jDiscount | Job discount included (Yes or No) | Char(1) | No | Y/N |
| jPickupAddressLine1 | Jobs pickup address line 1 | Nvarchar(50) | No |  |
| jPickupAddressLine2 | Jobs pickup address line 2 | Nvarchar(50) | No |  |
| jPickupCity | Jobs pickup city | Nvarchar(30) | No |  |
| jPickupPostcode | Jobs pickup postcode | Nvarchar(9) | No |  |
| jDropoffAddressLine1 | Jobs dropoff address line 1 | Nvarchar(50) | No |  |
| jDropoffAddressLine2 | Jobs dropoff address line 2 | Nvarchar(50) | No |  |
| jDropoffCity | Jobs dropoff city | Nvarchar(30) | No |  |
| jDropoffPostcode | Jobs dropoff postcode | Nvarchar(9) | Yes |  |
| jPickupDate | Jobs pickup date | Date | No |  |
| jPickupTime | Time of the job | Time(7) | No |  |
| jMileageBefore | Cars mileage before job | Int | No | Defaults to 0 |
| jMileageAfter | Cars mileage after job | Int | No | Defaults to 0 |
| jPayment | Jobs payment value | Money | No | Defaults to 0 |
| jCancellationReason | Reason for the jobs cancellation | Nvarchar(50) | Yes |  |
| jTotalMileage | The total mileage travelled for the job | Int | No |  |
| dLicenceNumber | Drivers licence number | Nvarchar(18) | No | PK, FK |
| dTitle | Drivers title | Nvarchar(15) | No |  |
| dFirstName | Drivers first name | Nvarchar(50) | No |  |
| dLastName | Drivers last name | Nvarchar(50) | No |  |
| dPicture | Drivers picture | Image | Yes |  |
| tVehicleLicenceNumber | Taxis vehicle licence number | Nvarchar(8) | No | PK, FK |
| tVehicleMake | Taxis car make | Nvarchar(20) | No |  |
| tVehicleModel | Taxis car model | Nvarchar(20) | No |  |
| tPassengerCapacity | Taxis passenger capacity | Int | No | Min of 4 Max of 12 |
| tStorageCapacity | Taxis storage capacity | Int | No | Min of 4 Max of 12 |
| tCurrentMileage | Taxis current mileage | Int | No |  |
| sNumber | Staff’s id number | Nvarchar(8)s | No | PK |
| sTitle | Staff’s title | Nvarchar(15) | No |  |
| sFirstName | Staff’s first name | Nvarchar(50) | No |  |
| sLastName | Staff’s last name | Nvarchar(50) | No |  |
| sSalary | Staff’s salary | Money | No |  |
| shNumber | Service history’s id number | Nvarchar(3) | No | PPK |
| shDate | Service history’s date of service | Date | No |  |
| shMileage | Vehicle mileage at point of service | Int | No |  |
| shType | Type of service | Nvarchar(7) | No |  |
| shDescription | Important info regarding the service | Nvarchar(100) | No |  |

**Derived attributes:**

jTotalMileage = jMileageAfter – jMileageBefore

jTotalPassengers = jNumberAdults + jNumberChildren

tAppearance = tVehicleLicenceNumber + tVehicleMake + tVehicleModel

## SQL Server Database and Tables

### Customer



### Job



### Driver



### Staff



### Taxi



### Service History



## Database Diagram



## SQL Server Database Queries

1. **Retrieve all records and all columns from a single table.**

SELECT \*

FROM Driver;

1. **Retrieve some records and only some columns from a single table.**

SELECT cEmail, cPassword, cID, cTitle

FROM Customer

WHERE cTitle='Mr';

1. **Retrieve all related records and all columns from two tables.**

SELECT \*

FROM Customer c, Job j

WHERE c.cID=j.cID;

1. **Retrieve some related records and only some columns from two related tables.**

SELECT c.cID, c.cTitle, c.cFirstName, c.cLastName, j.jID, j.jPickupAddressLine1, j.jPickupAddressLine2, j.jPickupCity, j.jPickupPostcode

FROM Customer c, Job j

WHERE c.cID=j.cID AND c.cTitle='mr';

1. **Retrieve all related records and only some columns from three related tables.**

SELECT c.cID, c.cFirstName, c.cLastName, j.jID, j.jPickupCity, j.jPickupPostcode, s.sNumber, s.sFirstName, s.sLastName

FROM Customer c, Job j, Staff s

WHERE c.cID=j.cID AND s.sNumber=j.sNumber;

1. **Retrieve all records and all columns from a single table and include a single calculated column. Give the calculated column an appropriate name.**

SELECT \*, sSalary/12 AS sMonthlySalary

FROM Staff;

1. **Retrieve all records and some columns from two related tables and include a single calculated column based on columns from both tables. Give the calculated column an appropriate name.**

SELECT t.tVehicleLicenceNumber, t.tCurrentMileage, sh.shNumber , sh.shMileage, t.tCurrentMileage-sh.shMileage AS mileageSinceFirstService

FROM Taxi t, ServiceHistory sh

WHERE t.tVehicleLicenceNumber=sh.tVehicleLicenceNumber;

1. **Retrieve some records from a single table using pattern matching.**

SELECT tVehicleLicenceNumber, tVehicleMake,tVehicleModel,tPassengerCapacity

FROM Taxi

WHERE tVehicleMake LIKE '%Vauxhall';

1. **Retrieve records from a single table that contain nulls in one or more columns.**

SELECT \*

FROM Customer

WHERE cMobileNumber IS NULL;

1. **Retrieve records from a single table that do not contain nulls in one or more columns.**

SELECT \*

FROM Customer

WHERE cMobileNumber IS NOT NULL;

1. **Retrieve all related records from a single table that are ordered using two non-primary key columns.**

SELECT \*

FROM Driver

ORDER BY dFirstName, dLastName;

1. **Retrieve one or more aggregate values (such as average or sum) for a single table.**

SELECT MIN(tCurrentMileage) AS lowestTaxiMieage,

MAX(tCurrentMileage) AS maxTaxiMileage,

AVG(tCurrentMileage) AS averageTaxiMileage

FROM Taxi;

1. **Retrieve groups of records from a single table.**

SELECT tVehicleMake, COUNT(tVehicleModel) AS tNumberOfVehicles

FROM Taxi

GROUP BY tVehicleMake

ORDER BY tNumberOfVehicles;

1. **Retrieve some groups of records from two related table.**

SELECT c.cID, j.jType, COUNT(\*) AS jCount

FROM Customer c, Job j

WHERE c.cID = j.cID

GROUP BY c.cID, j.jType

ORDER BY c.cID, j.jType;

1. **Retrieve aggregate values (such as count) for groups of records from a single table.**

SELECT tVehicleMake, COUNT(tVehicleModel) AS tNumberOfVehicles

FROM Taxi

GROUP BY tVehicleMake

ORDER BY tNumberOfVehicles;

1. **Retrieve some records and some columns from a table based on the value(s) retrieved from another table.**

SELECT t.tVehicleLicenceNumber, t.tVehicleMake, t.tVehicleModel

FROM Taxi t, ServiceHistory sh

WHERE sh.shMileage BETWEEN 5000 AND 5800

AND t.tVehicleLicenceNumber=sh.tVehicleLicenceNumber;

1. **Retrieve some records and some columns from a table based on the value(s) retrieved from another table, which are based on the value(s) from another table.**

SELECT c.cTitle, c.cFirstName, c.cLastName

FROM Customer c, Job j, Staff s

WHERE s.sNumber='SN004321' AND c.cID=j.cID AND s.sNumber=j.sNumber;

1. **Retrieve all records from one side of two related tables.**

SELECT c.\*, j.\*

FROM Customer c LEFT JOIN

Job j ON c.cCity=j.jPickupCity AND c.cID=j.cID;

1. **Retrieve all records from both sides of two related tables.**

SELECT c.\*, j.\*

FROM Customer c FULL JOIN

Job j ON c.cCity=j.jPickupCity AND c.cID=j.cID;

1. **Retrieve all records from three related tables.**

SELECT \*

FROM Job j, Staff s, Driver d

WHERE s.sNumber=j.sNumber AND d.dLicenceNumber=j.dLicenceNumber;