Samuel Bloxham, Zack Hawkins, Sebastian Michalak, Josie Wood

Word count:  1490

ISAD: Group Report

# Database Location

Our database created as a group is located on the following Xserve account:

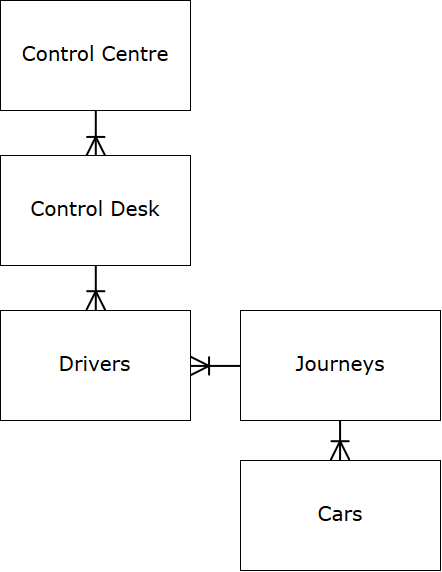
Username: isad156\_22205323

User: Zack Hawkins

Server: Xserve

# Modelling

Initial ERD



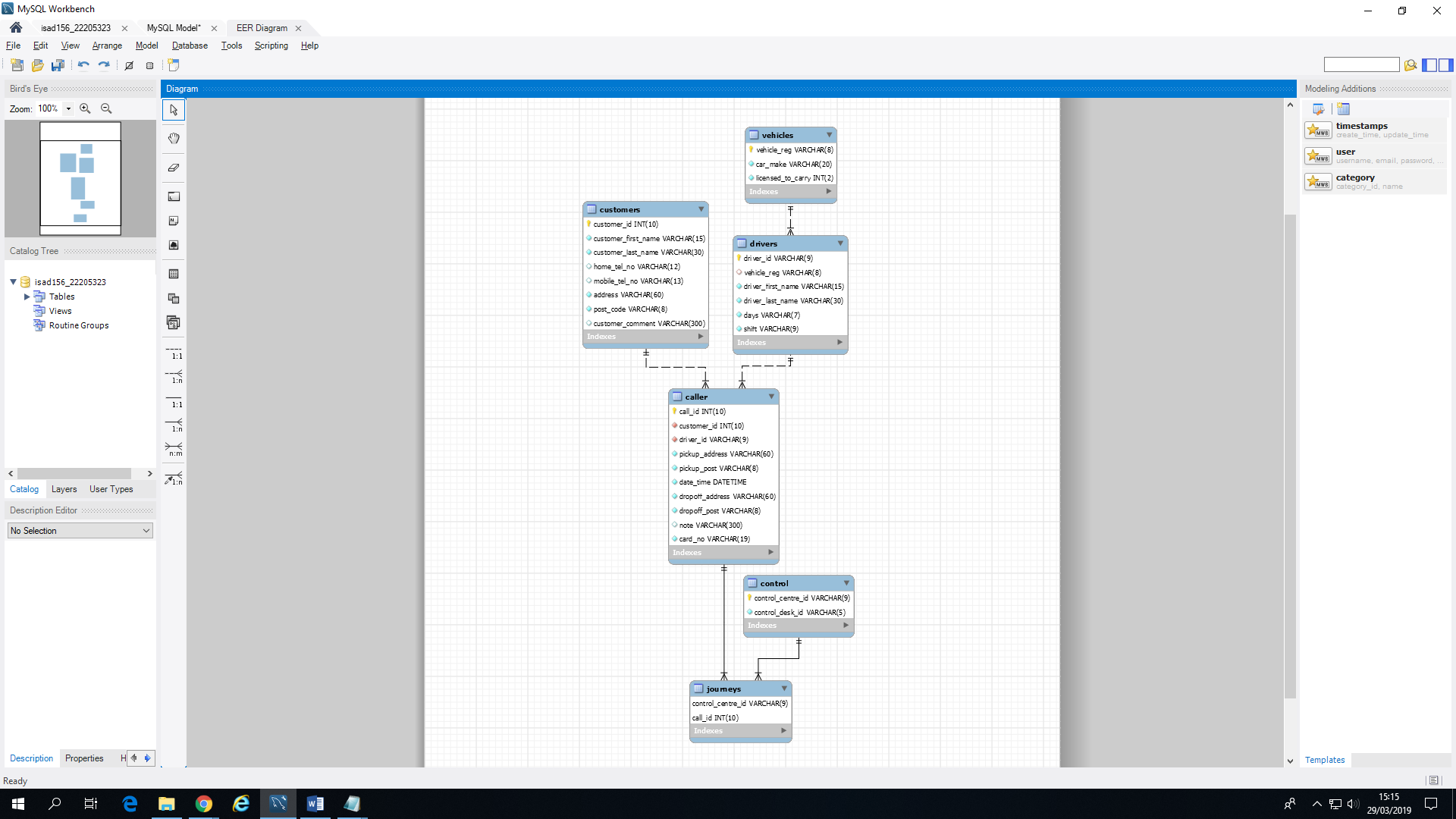
Assumptions:

* Journeys are made by cars, not drivers

Normalisation

|  |  |  |  |
| --- | --- | --- | --- |
| **UNF** | **1NF** | **2NF** | **3NF** |
| **Control Centre** | **Control centre** | **Control centre** | **Control Centre** |
| Control Desk | Control desk | Control desk | Control Desk |
| (Call ID | Vehicle reg | Vehicle reg |  |
| Customer first name | Car make | Car make | **Control Centre** |
| Customer last name | Licensed to Carry | Licensed to Carry | **Call ID** |
| Home tel. # |  |  |  |
| Mobile tel. # | **Control centre** | **Control centre** | **Call ID** |
| Address | **Call ID** | **Call ID** | *\*Customer ID* |
| Postcode | Customer ID |  | *\*Driver ID* |
| Comment) | Driver ID | **Call ID** | Pick up Address |
|  | Pick up address | Customer ID | Pick up Postcode |
|  | Pick up postcode | Driver ID | Date/Time |
| (Customer ID | Drop off address | Pick up address | Drop off Address |
| Pick up address | Drop off postcode | Pick up postcode | Drop off Postcode |
| Pick up postcode | Note | Date/time | Note |
| Drop off address | Card # | Drop off address | Card # |
| Drop off postcode | Date-Time | Drop off postcode |  |
| Date/time |  | Note | **Control Centre** |
| Note | **Control centre** | Card # | **Customer ID** |
| Card #) | **Customer ID** |  |  |
|  | Customer first name | **Control centre** | **Customer ID** |
| (Driver ID | Customer last name | **Customer ID** | Customer first name |
| Driver first name | Home tel.# |  | Customer last name |
| Driver last name | Mobile tel. # | **Customer ID** | Home tel. # |
| Days working | Address | Customer first name | Mobile # |
| Shift | Postcode | Customer last name | Address |
| Vehicle Reg | Comment | Home tel. # | Postcode |
| Car make |  | Mobile # | Comment |
| Licensed to Carry) | **Control centre** | Address |  |
|  | **Driver ID** | Postcode | **Control Centre** |
|  | Driver first name | Comment | **Driver ID** |
|  | Driver last name |  |  |
|  | Days working | **Control centre** | **Driver ID** |
|  | Shift | **Driver ID** | *\*Vehicle reg* |
|  |  |  | Driver first name |
|  |  | **Driver ID** | Driver last name |
|  |  | Driver first name | Days working |
|  |  | Driver last name | Shifts |
|  |  | Days working |  |
|  |  | shifts | **Vehicle Reg** |
|  |  |  | Car make |
|  |  |  | Licensed to carry |

#### *Final ERD made in mySQL*



# Database development

|  |  |  |
| --- | --- | --- |
| **Task** | | **Query** |
| **QUERIES TO PRODUCE THE DATABASE** | | |
| Creating each table with columns, primary and foreign keys | Control | CREATE TABLE control (  control\_centre\_id VARCHAR(9) NOT NULL,  control\_desk\_id VARCHAR(5) NOT NULL,  primary key(control\_centre\_id)); |
| Customers | CREATE TABLE customer\_details (  customer\_id INT(10) NOT NULL AUTO\_INCREMENT,  customer\_first\_name VARCHAR(15) NOT NULL,  customer\_last\_name VARCHAR(30) NOT NULL,  home\_tel\_no INT(11),  mobile\_tel\_no INT(12) NOT NULL,  address VARCHAR(60) NOT NULL,  postcode VARCHAR(8) NOT NULL,  customer\_comment VARCHAR(300),  primary key (customer\_id)); |
| Caller | CREATE TABLE caller(  call\_id INT(10) NOT NULL,  customer\_id INT(10)NOT NULL,  driver\_id VARCHAR(9) NOT NULL,  pickup\_address VARCHAR (60) NOT NULL,  pickup\_post VARCHAR (8) NOT NULL,  date\_time datetime NOT NULL,  dropoff\_address VARCHAR(60) NOT NULL,  dropoff\_post VARCHAR(8) NOT NULL,  note VARCHAR (300),  card\_no INT(19) NOT NULL,  primary key (call\_id),  foreign key(customer\_id) REFERENCES customers(customer\_id)  ON DELETE CASCADE ON UPDATE CASCADE,  foreign key(driver\_id) REFERENCES drivers(driver\_id)  ON DELETE CASCADE ON UPDATE CASCADE); |
| Drivers | CREATE TABLE driver\_details(  driver\_id VARCHAR(9) NOT NULL,  vehicle\_reg VARCHAR(8) NOT NULL,  driver\_first\_name VARCHAR (15) NOT NULL,  driver\_last\_name VARCHAR (30) NOT NULL,  days VARCHAR (7) NOT NULL,  shift VARCHAR (9) NOT NULL,  primary key (driver\_id),  foreign key(vehicle\_reg) REFERENCES vehicles(vehicle\_reg)  ON DELETE CASCADE ON UPDATE CASCADE); |
| Vehicles | CREATE TABLE vehicles(  vehicle\_reg VARCHAR(8) NOT NULL,  car\_make VARCHAR(20) NOT NULL,  licensed\_to\_carry INT(2),  primary key (vehicle\_reg)); |
| Journeys | CREATE TABLE journeys(  control\_centre\_id VARCHAR(9) NOT NULL,  call\_id INT (10) NOT NULL,  primary key (control\_centre\_id),  foreign key(call\_id) REFERENCES caller(call\_id));  foreign key (control\_centre\_id) REFERENCES control(control\_centre\_id)); |
| Reading data from .txt files into table | Customers.txt | LOAD DATA LOCAL INFILE ‘/path/customers.txt’ INTO TABLE customers |
| Drivers.txt | LOAD DATA LOCAL INFILE ‘/path/drivers.txt’ INTO TABLE drivers |
| Journey\_plans.txt | LOAD DATA LOCAL INFILE ‘/path/journey\_plans.txt’ INTO TABLE caller |
| Remove records from database  (deleting customer with id 007) | | DELETE FROM customers WHERE customer\_id = ‘007’; |
| Editing record in database  (changing customer last name given cust id) | | UPDATE customers SET ‘customer\_last\_name = ‘Smith’ WHERE customer\_id = 007; |
| **QUEREIS TO RETRIEVE HISTORICAL DETAILS FROM DATABASE** | | |
| Retrieve list of customers served in previous week | | SELECT customer\_ID FROM caller WHERE date\_time >= NOW() – INTERVAL 1 WEEK; |
| Retrieve list of customers picked up or dropped off at the Railway Station | | SELECT customer\_first\_name, customer\_last\_name FROM caller WHERE dropoff\_address LIKE ‘%Railway Station%’ OR pickup\_address LIKE ‘%Railway Station%’; |
| Retrieve number of taxis hired per shift on particular date, week, or period of time  Needs linked search | Date  (AM shift of 20/03/19) | SELECT COUNT(DISTINCT call\_id) FROM caller  WHERE (DATE(date\_time) = ‘2018-03-20’) AND TIME(date\_time) BETWEEN ’01:00:00’ AND ’12:00:00’;; |
| Week  (AM shift between 20/03/19 and 27/03/19) | SELECT COUNT(DISTINCT call\_id) FROM caller  WHERE (DATE(date\_time) BETWEEN ‘2018-03-10’ AND ‘2018-03-27’) AND TIME(date\_time) BETWEEN ’01:00:00’ AND ’12:00:00’; |
| Period of time  (between 10 am on 20/03/19 and 9pm on 21/03/19) | SELECT COUNT(DISTINCT call\_id) FROM caller  WHERE (DATE(date\_time) BETWEEN ‘2018-03-10’ AND ‘2018-04-10’) AND TIME(date\_time) BETWEEN ‘01:00:00’ AND ‘12:00:00’; |
| Retrieve list of most frequent customers  (defining frequent as customers within 3 month time period who have booked more than 2 taxis) | | SELECT caller.customer\_id, customers.customer\_first\_name FROM caller INNER JOIN customers ON caller.customer\_id = customers.customer\_id WHERE DATE(date\_time) BETWEEN ‘2018-01-01’ AND ‘2018-03-31’ GROUP BY caller.customer\_id HAVING COUNT(caller.customer\_id) > 2; |
| Retrieve list of non-returning customers | | SELECT customers.customer\_first\_name, customers.customer\_ last\_name FROM caller INNER JOIN customers ON caller.customer\_id = customers.customer\_id WHERE date\_time <= NOW() – INTERVAL 1 YEAR GROUP BY caller.customer\_id; |

# Evaluation/ Critical review

Problems encountered

* When we received the data set for customers we found that it had a home postcode for each customer, which is an attribute we didn’t pick out from the brief. This means we need to alter our database from our 3NF as we need to add a postcode column to the customers table, because otherwise when we read the data there’s more than we have allocated space for, causing it to error.
* We had trouble importing data from the .txt files into our database without creating new tables from the data or converting the files into .csv format, both of which don’t appear to apply with our briefing. As such, we have discussed our issue with Chris and found our issue using the Show Logs function, which informed us that some of the data being read in didn’t comply with the restrictions of the table, eg we set our card num value as an INT(16), but in the table each 4 digits is followed by a dash, so it couldn’t be saved as a n int. We fixed this issue by changing the card num value to hold a VARCHAR(19) to account for non-int characters such as the dash 3 times within the card number. By editing all the data types where required, we were able to successfully read in the data to our tables.
* The way data was given in the data didn’t match up to how we had planned to arrange it in our database; for the car vehicle reg, for example, the data didn’t specify so we’ve had to create our own to fill the database
* Another issue we found was that the customer data we were given didn’t contain customer IDs. This meant we had to procedurally create an ID for each customer in the database. As well as this, there were more customer IDs in the journeys list than customers themselves, so we had to create another 200 customer records on top of the given 100 in order to match up with the 300 customers referred to in the journeys table.

Suitability and completeness of solution

* Our solution works well and completes all the queries as expected and required. Solutions to problems seem suitable and overcome them well. The database as a whole functions properly and is complete in itself.
* We can query our database as needed to retrieve all the information required, which indicates well to us that the solution would work well in a real life situation as a tool to analyse use of database and receive any additional details as required

What would we do differently?

* If we undertook this project again we would probably end up with a similar end product, as we are happy with how it functions, but some additional features may be useful to allow further manipulation of the data
* One improvement would be to convert the days of the week that the drivers are working into numeric values, such as Monday = 1, Tuesday = 2 and so on. This may make it easier to demonstrate the selection of days drivers are working, especially the range of days such as Monday to Wednesday, as it can instead be shown as 1 to 3, aka 1, 2, 3
* As far as project management goes, in future we may opt to equally share the report writing aspect of the project. This stage allows reflection throughout the project, which may be best done by all team members in order to ensure that everyone is caught up with our progress and fully understands what has been done.

# Group involvement

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Activity** | **Expected Start Date** | **Estimated Hours (per person)** | **Actual Start Date** | **Actual Hours Spent on this activity** | **Work Carried Out by** |
| **Data Model (ERD)** | 11 February | 10 | 19/02/19 | 2 | Sam |
| 2 | Seb |
| 2 | Zack |
| 2 | Josie |
| **Data Model (Normalisation)** | 18 February | 10 | 19/02/19 | 7 | Sam |
| 7 | Seb |
| 7 | Zack |
| 7 | Josie |
| **Development (MySQL)** | 25 February | 30 | 12/03/19 | 5 | Sam |
| 5 | Seb |
| 7 | Zack |
| 5 | Josie |
| **Report Writing** | 11 February | 10 | 26/02/19 | - | Sam |
| - | Seb |
| - | Zack |
| 5 | Josie |