Ride4U Database System

TOWER HAMLETS COLLEGE | [Company address]

GHEORGHE MITREA 1118495

Database Design Concepts

2016

Contents

[*INTRODUCTION* 1](#_Toc462951297)

[TASK 1 1](#_Toc462951298)

[1.1And 1.2 Basics of database management system 1](#_Toc462951299)

[TASK 2 6](#_Toc462951300)

[2.1 Database design method and model 6](#_Toc462951301)

[2.2 Normalization and Entity relationship model 9](#_Toc462951302)

[TASK 3 16](#_Toc462951303)

[3.1 Development environment and Solutions 16](#_Toc462951304)

[3.2 Testing of database system 35](#_Toc462951305)

[3.3 Validation and Verification techniques 49](#_Toc462951306)

[3.4 User manual and technical documentation 51](#_Toc462951307)

[CONCLUSION 53](#_Toc462951308)

[Bibliography 55](#_Toc462951309)

***Table of Figures***

[**Figure 1: 1st Tier Architecture** 2](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951310)

[Figure 2: 2nd Tier architecture 3](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951311)

[Figure 3: N-tier architecture 4](#_Toc462951312)

[Figure 4 Analyse the relationship between tables 10](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951313)

[Figure 5 ER diagram for ride4u database 10](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951314)

[Figure 6 example of a non-1st NF 11](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951315)

[Figure 7 Example of a non-2nd NF 13](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951316)

[Figure 8 will be sub divided into two tables when 2NF is applied on it 13](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951317)

[Figure 9 will be sub divided into two tables when 2NF is applied on it 14](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951318)

[Figure 10 A non-3rd NF 14](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951319)

[Figure 11 A 3rd NF of my application 15](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951320)

[Figure 12 The input data of customer table 17](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951321)

[Figure 13 Output data for Customer table 18](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951322)

[Figure 14 The structure of Customer table 18](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951323)

[Figure 15 output data of customer invoice 19](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951324)

[Figure 16 Input data of customer invoice table 19](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951325)

[Figure 17 Input data for Customer Order table 20](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951326)

[Figure 18 The structure of Customer Order table 21](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951327)

[Figure 19 The output data for Customer Order table 21](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951328)

[Figure 20 The structure of the Customer Order Detail 22](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951329)

[Figure 21 The output data for Customer Order Detail 22](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951330)

[Figure 22 The code for Customer Order Detail 22](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951331)

[Figure 23 The input data for Driver table 23](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951332)

[Figure 24 The output data for Driver table 23](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951333)

[Figure 25 The structure of Driver table 24](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951334)

[Figure 26 Input data for Employee table 24](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951335)

[Figure 27 Output data for Employee table 25](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951336)

[Figure 28 Structure of Employee table 25](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951337)

[Figure 29 Input data for Employee Role table 26](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951338)

[Figure 30 Output data for Employee Role table 26](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951339)

[Figure 31 Structure of Employee Role table 26](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951340)

[Figure 32 Input data for Office table 27](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951341)

[Figure 33 Output data for Office table 27](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951342)

[Figure 34 Office table structure 28](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951343)

[Figure 35 Role table with input data 28](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951344)

[Figure 36 Structure of Role table 28](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951345)

[Figure 37 Output data for Role table 29](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951346)

[Figure 38 input data for Vehicle table 29](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951347)

[Figure 39 output data for Vehicle table 30](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951348)

[Figure 40 Vehicle Structure 30](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951349)

[Figure 41 the input data for Vehicle Driver table 30](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951350)

[Figure 42 Structure of Vehicle Driver table 31](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951351)

[Figure 43 Input data for VehicleMot table 31](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951352)

[Figure 44 Input data for VehicleMot table 32](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951353)

[Figure 45 The output data for VehicleMot table 32](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951354)

[Figure 46 Structure of VehicleMot table 33](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951355)

[Figure 47 Insert data into Office table 33](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951356)

[Figure 48 Insert data into Employee table 34](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951357)

[Figure 49 insert data for CustomerOrder 35](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951358)

[Figure 50 Update data for customer id 1 with the old value for the phone no. 36](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951359)

[Figure 51 Message regarding the update which I have updated for column phone\_no from customer id 1 in customer table 37](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951360)

[Figure 52 Correct data after update data for phone no. of customer id 1 37](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951361)

[Figure 53 The names and phone numbers of the Managers at each office location 39](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951362)

[Figure 54 The names of all drivers based in the different offices 40](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951363)

[Figure 55 The total number of staff at each office 40](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951364)

[Figure 56 The details of all taxis at the different offices 41](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951365)

[Figure 57 The number of drivers allocated to each taxi 41](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951366)

[Figure 58 The name and number of owners with more than one taxi 42](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951367)

[Figure 59 The full address of all clients 42](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951368)

[Figure 60 The details of all Taxis with MOT details 43](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951369)

[Figure 61 The details of jobs undertaken by a driver on a given day 44](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951370)

[Figure 62 The names and addresses of clients who have hired a taxi more than three times 44](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951371)

[Figure 63 The average number of jobs per driver 45](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951372)

[Figure 64 The total number of jobs allocated to each car 46](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951373)

[Figure 65 The total number of jobs allocated to each driver 47](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951374)

[Figure 66 The total number of jobs and miles driven for a given contract 48](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951375)

[Figure 67: Customer table with data types 49](#_Toc462951376)

[Figure 68 Validation data when I try to duplicate the data in a employee table b introducing a new employee 50](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951377)

[Figure 69 cannot delete or update a parent row for a foreign key 51](file:///C:\Users\mitrea\Desktop\Final_Assingment_%20Database%20.docx#_Toc462951378)

[Figure 70: Login screen 52](#_Toc462951379)

[Figure 71: Relationship among attributes 53](#_Toc462951380)

### *INTRODUCTION*

Database system is computer software which is used to interact with database whereas database is a collection of organized and structured data in a file. The report will describe the basics about database, its architecture and role of administrators. Later, the report will emphasize on SSADM as database development methodology for design of database for Ride4U. ER diagrams and normalization will be described for database design. The report will suggest the database design and structure along with number of useful queries for Ride4U organization which is a taxi providing organization. Also designed database will be tested for errors and acceptability. The report will provide the user manual and technical specification for database system.

# TASK 1

## 1.1And 1.2 Basics of database management system

**Data consistency**

Data is at core of any organization to regulate the business operations. Data consistency means the usability of data in a database so it can not violate the rules defined to store data. For instance, if an attribute is configured to accept only the values in format male and female then database must need to roll back the transaction and report errors to user. Data consistency ensures that data will be stored and retrieved in a consistent manner. (Adnan, 2010). For instance, in Ride4U, if database is designed well and secure then user will be unable to enter incorrect data format and data in queries to update and store the values in database. Also the right values can be received each time the query is made on database.

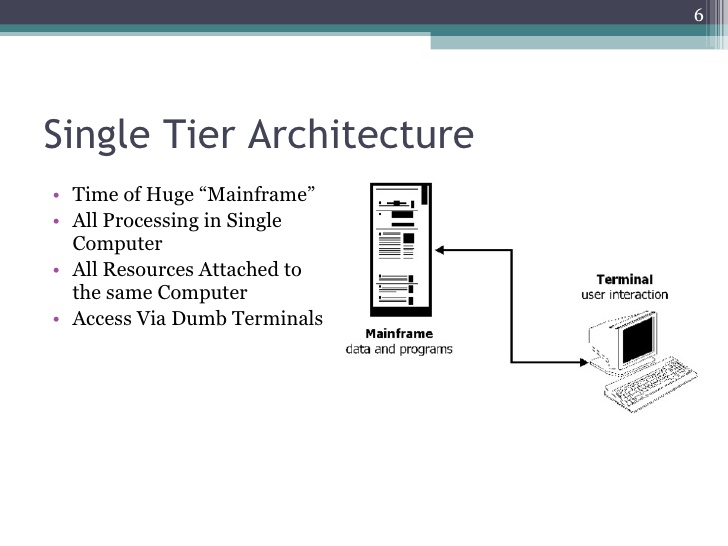
**Normalization**

The term normalization in database refers to the process of eliminating redundant data and anomalies from database. Anomalies may be in insertion, update or delete operation. With normalization, following benefits can be achieved:

* Data can be arranged in logical groups so that each one represents small part of database rather than whole.
* Space can be saved by removing duplicate data from database.
* Organizing the data in a manageable form so that changes needs to be done only at one place rather than data being reflected elsewhere in database.
* For accessibility and manipulation in quick manner without loss of data integrity (Halpin, 2010).

**Database architectures**

It is the way to address the needs of Ride4U to deploy effective database. Application logic is used to define the presentation, store and processing configurations in database implementation. On the basis of various application logics, database architecture can be categorized and designed as according to demands:

* 1-tier: In it, DBMS is directed operated by user and any of the changes on DBMS will be reflected with the core interface of DBMS. For instance, access running on local machine is in 1-tier architecture. It is normally preferred by programmers and developers in order to test and optimize the database(Özsu, 2011). It is beneficial when dealing with small amount of data which is centric to single user maintenance.

**Figure 1: 1st Tier Architecture**

Source:3 Tier Architecture, http://www.slideshare.net/sanjeevwebx/3-tier-architecture-2410697

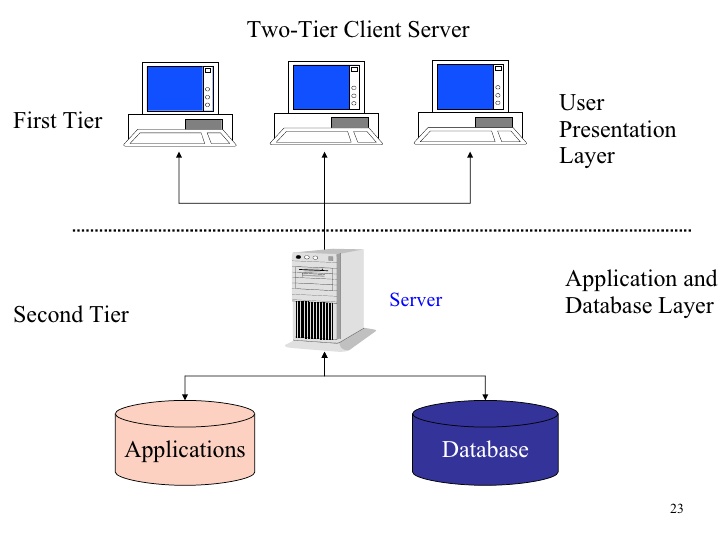
* 2-tier: When user uses an application program to work with DBMS, then architecture is said to be in 2-tier. Common example is web based interface to access database entries
* where user makes use of GUI to operate on database. Main database system has two tiers: one is client system and second is application program.

Figure 2: 2nd Tier architecture

(Source: Infs2005 Accounting Information Systems, http://www.slideshare.net/guest3bd2a12/infs2005-accounting-information-systems-presentation)

* N-tier architecture: In n-tier architecture, user and DBMS is connected through multiple layers which helps DBMS by translating client queries and data into appropriate form. A typical n-tier may contain web server and application server as middle tiers. N-tier architecture is capable to handle demands of enterprises to provide most robust entries and queries by simplifies them at each tier for better performance at database system.

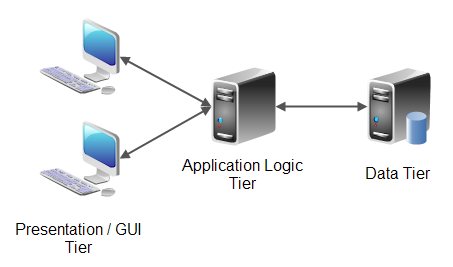


Figure 3: N-tier architecture

(Source: n-tier architecture, 2016)

* Peer-to-peer architecture: In this model, one server device works dedicatedly for another device to provide services and data. For instance, interconnected devices may be shared for processing power, storage and data applications but their connection to each other are dedicated and no other device can same resources until the host free it in network(Varshney, 2012). Peer to peer architecture is used mainly for backup and file sharing in DBMS within the network. It is effective to provide security and consistency in database.

**Methods of data access, data recovery and data security**

Various indexes and hash functions like B-tree and multi valued index can be used to access data from database system. Addition to it, sequential and random method is used to access data from various devices in database network. Also, DBMS needs to provide data recovery in case of failures and data loss. For that, DBMS can rollover all the transactions back to consistent state or can use recovery techniques to restore previous data.Data security can be provided in application as back up programs and data front end applications may be affected with malware and other bugs to meld the normal operations. Unauthorised users normally use the application program’s weak security and integrity with system. Application needs to provide abstraction to database to provide security. Database also can be configured well to receive only one transaction and value at a time to protect the fields against the abnormal values. Data types and attributes can be reviewed to error proof the fields in database. User permission and accounts on database can be logged only at administrator to monitor the security easily and effectively. Addition to it, servers can be safeguarded for physical and logical access or damage for data security. Also anti-theft programs can be installed on servers to monitor transactional attacks(Elmasri, 2011).

**Role of database administrator in keeping data integrity**

Administrator is responsible to monitor unauthorised access and activities on database. Administrator in Ride4U needs to do following functions to ensure data integrity in system:

* Update privileges and permission to user accounts.
* Monitors activities in DBMS
* Action against the illegal attempts on DBMS.
* Configure and setup database system to ensure consistency and integrity in operations.

**Role of Industry standards**

Selection of DBMS system is typical difficult because most of DBMS products are similar in functions but are different in term of technology used and vendor supports. This is the job of DBA group to identify the right DBMS which can fulfil the requirements of organization along with hardware configuration of system.

* MySQL and Oracle is available to Linux and UNIX where as Microsoft SQL server and Access Database is only indented to use on Windows platforms.
* Also, Postgre and NoSQL databases are in trend for new requirements in organizations. Proper decision can be made to select the right choice.
* Oracle, DB2 and SQL server are capable to handle 1-tier DBMS whereas Informix dynamic server, adaptive server enterprise.
* Teradata and MySQL are good choices for 2-tier architecture. But all this depended on capabilities of employees and available resources in workplace.

# TASK 2

## 2.1 Database design method and model

DBTech has recommended the use of SQL database servers for implementation in Ride4U. An effective and good design methodology for database development on SQL server may include following features in it:

* Complete set of stages to achieve objective.
* Integrity of data with data processes to interconnect the modules.
* Flexibility to arrange modules and demands as according to needs.
* Responsive to deliver specific results after the completion of stage (Rock-Evans, 2014).
* Accessibility in reuse of already achieved information and results to generate new results.

SQL server is recommended due to high backed support and easy to use services. Also SQL server has a big storage space and divided core process for faster and smarter use within a network system. All this features and cost effectiveness make it suitable for Ride4U organization. It is free and effective to use in network but the little problem comes in creation of reports and forms.

**SSADM- Structured Systems Analysis and Design Methodology**

It is one of database design methodology which is widely accepted due to following features:

* Modularization: entire method is sub divided into more manageable and target specific modules which help to optimize the use of resources in development process. Modularization will be helpful to divide the designing and development part which are easy to parallelize the work in Ride4U.
* Visualization: Visual diagrams and trees help to achieve contribution of non-technical staff in Ride4U organization.
* Faster-smarter: Each stage in model is clear and target directed to accomplish the objectives of Ride4U organization.

However, SSADM may be less effective because of redundant analysis of requirements and larger design of database by developers. These mistakes should be overcome to make use of SSADM methodology.

**STAGES IN SSADM**

**Stage 0: feasibility study**

DBTech developer needs to consider all the financial, technical and business elements those may affect the development of new system(PAŞCU, 2010). Proposed requirements are studied to ensure to practical and future scope of new design for organization. Ride4U also need to support in feasible study to provide necessary information to DBTech developers. This stage will answer the realistic and achievability of new design.

**Stage 1: Investigation of current environment**

Current working environments including practices and skill sets along the physical implementation of business is determined in this stage. Staff of Ride4U can be used to gather required information with questions and interviews. Their activities can be observed for a time to develop understanding with current system. It helps to achieve timely and measurable requirements of Ride4U.

**Stage 2: Business system options (BSO)**

DBTech developers will prepare a list of all possible solutions to meet the requirements of Ride4U. Organization can select of one of solution as according to requirements and investment constraints. Ride4U may provide the information on following points to support the team:

* Interface according to skills of users
* Resources distribution manner
* Cost and benefits with selected option.

**Stage 3: Requirement specification**

In this stage, BSO is analysed to find what the functionality will be of new development design in Ride4U organization. This stage mainly deals with the documentation of requirements those are specified by organization to achieve. It should be specific to design, measurable to features and investment and realistic. Addition to it, it should be prepared within time and achievable manner(Sun, 2013).Ride4U can help to develop:

* Demands and information catalogues.
* Events and functionality matrices in documents.
* Diagrams reflecting data flow in system.

**Stage 4: Technical system options**

Now developer needs to identify the technical specification of design. For that, Ride4U can be consulted to know the skills and capabilities in organization about:

* Hardware and software requirements
* Cost and time in which new system is demanded
* Implementation factors in place
* Network structure
* Frontend GUI to users.

**Stage 5: Logical design**

Once technical as well as BSO is clear to DBTech, it can move on logical designing of new system which includes the data as data operations to implements, diagrams to shown interconnectivity and logical flow of information for user activities. This can be said as outline for Ride4U to develop the new design with sketches.

**Stage 6: Physical design**

In physical design, all the elements of logical designs are converted to programming practices and codes to implement the database. Logical schemas, diagrams and flow charts are practically implemented to prepare the system for use with sample data. At the end of this stage, a working system come into existence for testing and maintenance.

**Advantages of SSADM in Ride4U**

SSADM is selected as the database development methodology for Ride4U organization because it is faster and modularised to use all the available skill sets in organization. Also the better representation of information and intermediate stages helps and encourages employees to become part of development. This methodology is step forwarded and capable to finish the development within minimum time. Each clear and target directed step of development is helpful for Ride4U to achieve and add functionality.

## 2.2 Normalization and Entity relationship model

Entity-relationship(ER) model is a graphical representation of relationship among various entities in database. Key components of ER diagram are:

|  |  |  |  |
| --- | --- | --- | --- |
| **Component** | **definition** | **Symbol** | **Example** |
| Entity | Atomic data unit in table | Rectangle | Taxi, customers etc. |
| Attribute | Property of entity | Eclipse | Customer ID, taxi number etc. |
| Key attribute | Unique attribute to retrieve entire record | Underline text in eclipse | Taxi number, invoice number etc. |
| Multi valued attribute | Attribute with composition of values | Double eclipse | Stations, owned taxi number etc. |
| Relationship | How one entity related to other in database(Thalheim, 2013) | Diamond shape | Driver of, dated on etc. |

**Relationship**

Entities in database may be related in following manner:

*One-to-one relationship*: For instance, only a driver can drive the taxi of specific number.

*One-to-many relationship*: An owner can hold more than one cab(Halpin, 2010).

*Many-to-many relationship*: many owners may have multiple cabs.

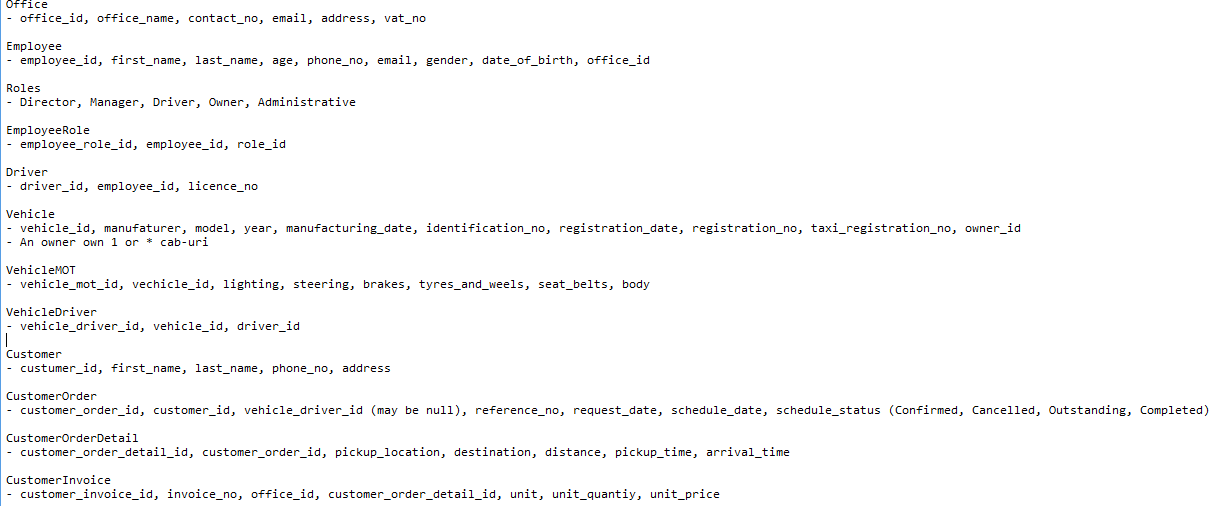
In below relationships, vehicle and owner are entities in different tables of database where as owning is relationship between them.

Figure 4 Analyse the relationship between tables

 ER diagram is given below and illustrates the relationships among vehicles, customers, employee, invoices details for ride4u database.

Figure 5 ER diagram for ride4u database

**Normalization**

Normalization means to remove redundant data values and anomalies from database. It is possible that a database contains multiple columns storing same information which result into difficulties in update process. Three types of normalization forms are used to deal with data redundancy and anomalies.

**First normal form (1NF):** First normal form always contains the atomic data values and each attribute is capable to hold single entity only. Domain is set of all possible values those can fit into data field. Below given table has customer ID attribute which is an auto incremental attribute. Both name fields are of text type and telephone numbers are of number type with length 20. But no attribute, from these two text fields, is of type primary key.

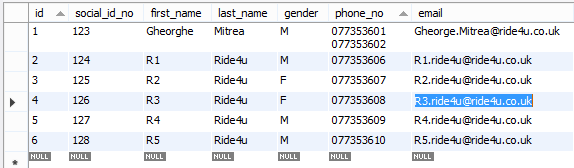


Figure 6 example of a non-1st NF

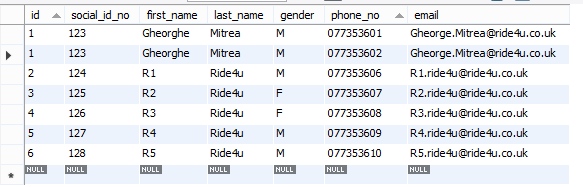
Note that telephone column contains multiple values for a single record. 1NF table is:

Figure 7 example of a 1st NF

**Second normal form (2NF):**

A database relation which is already in 1NF and has no dependency of non-prime key attributes on candidate keys is known as relation in 2NF (Bog, 2014). Candidate key are attributes those combined to act as unique parameter to find the record in database whereas prime keys are self-capable to determine the record uniquely. In table CustomerOrder, the yellow highlighted columns are candidate attributes and the red underline columns are non-prime attributes.

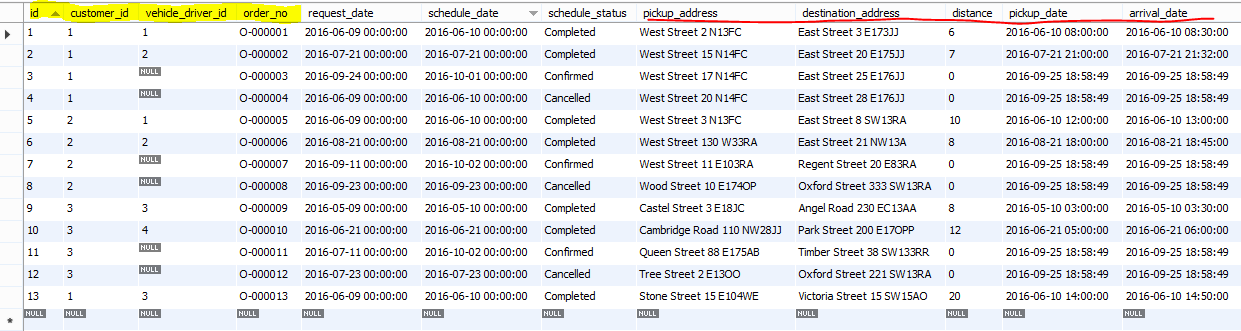


Figure 7 Example of a non-2nd NF

The below figures will be subdivided into two tables when 2NF is applied on it.

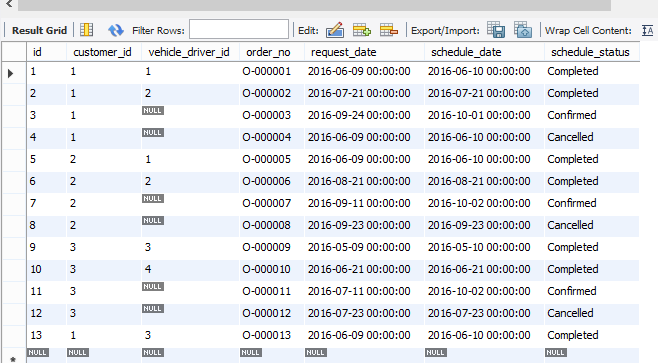


Figure 8 will be sub divided into two tables when 2NF is applied on it

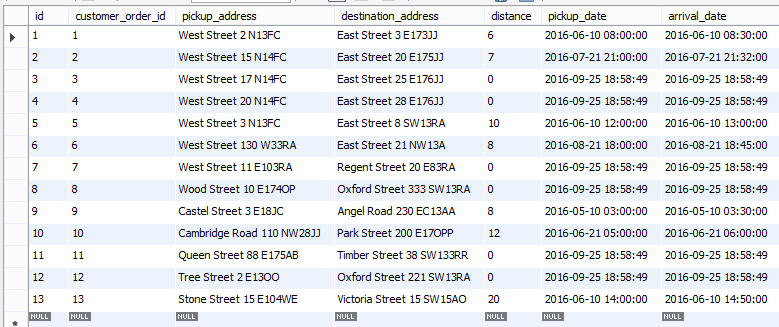


Figure 9 will be sub divided into two tables when 2NF is applied on it

**Third normal form (3NF):** It is used to remove anomalies leaved in 2NF in the form of redundant data and integrity issues. A database relation is said to be 3NF if it is already in 2NF and all attributes are only dependent on candidate key. 3NF is effective to make use of storage and cost associated with database implementation (Fattah, 2013).

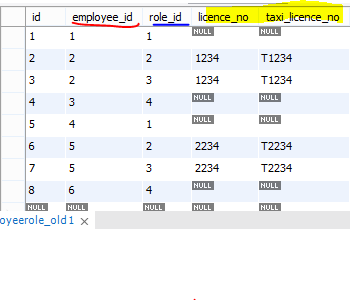


Figure 10 A non-3rd NF

In the above figure we have licence\_no and taxi\_licence\_no, which are non-prime attributes functionally depending on role\_id, which is a prime attribute of this table. In turn the role\_id attribute functionally depends on employee\_id which is the super key for this table. That means that both non-prime attributes (license\_no and taxi\_license\_no) are transitively dependent on super key employee\_id.

To make the table with third normal form, split the table in more manageable tables:

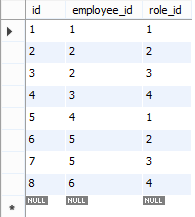
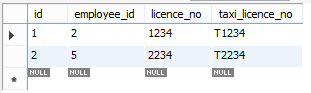


Figure 11 A 3rd NF of my application



# TASK 3

## 3.1 Development environment and Solutions

In order to implement DBTech needs to revisit the requirements specified by Ride4U and working environment. Following requirements are found in context of development of database system:

* Digitalization of organization as it is moving from paper work to digital world.
* Organization has different locations, taxis, customers and drivers(Febbraro, 2011). Thus there is need of at list 12 tables.
* Invoice generating facility.
* Mailing function.
* Data validation and duplication check-up service in database.
* Report generation etc.

***Implementation steps***

The life cycle of a database is the set of steps, techniques, methods and tools used for transposing the data model into a physical model.

Next steps to be followed to create a logic model:

* defining the specific requirements tables in the domain that shape (as established in the conceptual model);
* determining relationships between tables;
* determination (columns) each table;
* normalising tables until, at least, three normal form;
* determining the primary keys;
* determining the values of specific columns

1. ***Tables***

A table is an object database that you we to store data about an object, such as a ***customer*** or employee. A table consists of records and fields. Each record contains data about an instance of a topic in the table, such as a particular employee or customer in my example. A record is also called row or court. Each field contains information about one aspect of table subject, such as first name or e-mail. A field is also known as the column or attribute such as id, first\_name, last\_name, phone\_no, email, address.

***customer table:*** contains columns such as: *`id`, `first\_name`, `last\_name`, `phone\_no`, `email`, `address`,* where table contains indexes as *primary key (‘id’)* and *unique key* (‘*uk\_first\_name\_last\_name\_email’***)**

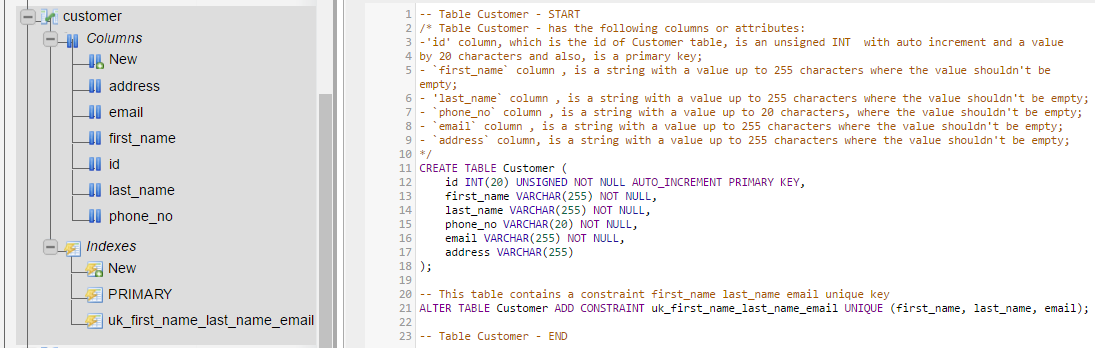
******

Figure 12 The input data of customer table

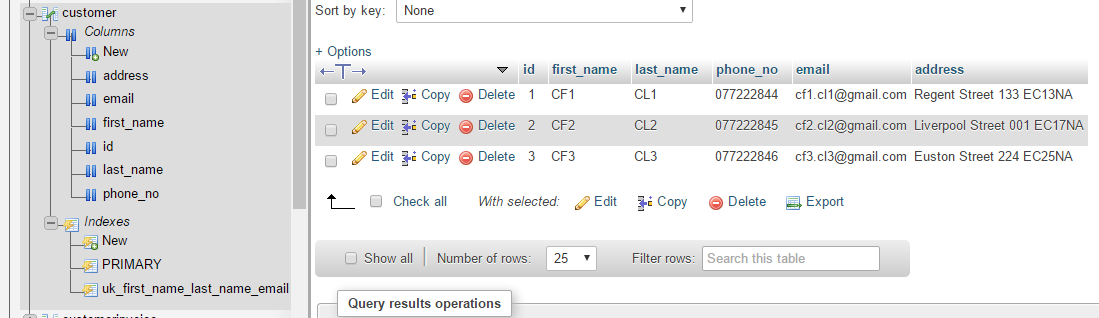


Figure 13 Output data for Customer table

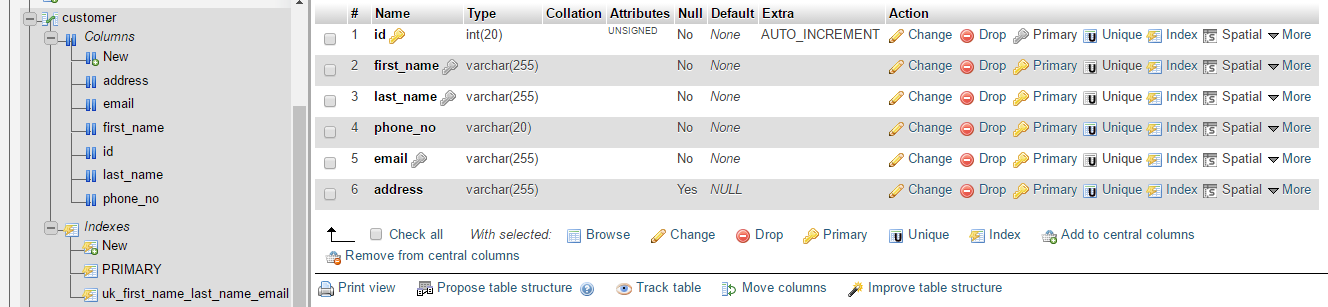


Figure 14 The structure of Customer table

***customerInvoice table:*** contains columns such as: `id`, `invoice\_no`, `customer\_order\_detail\_id`, `unit`, `unit\_quantiy`, `unit\_price`, `invoice\_status`, where table contains indexes as *primary key (‘id’)* and *unique key* and *unique key* (‘uk\_customer\_order\_detail\_id’ and ‘uk\_invoice\_no*’***):**

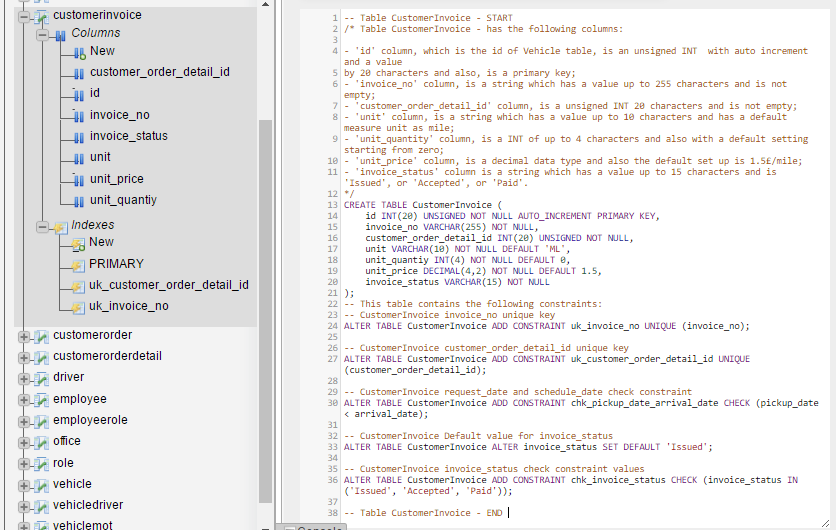
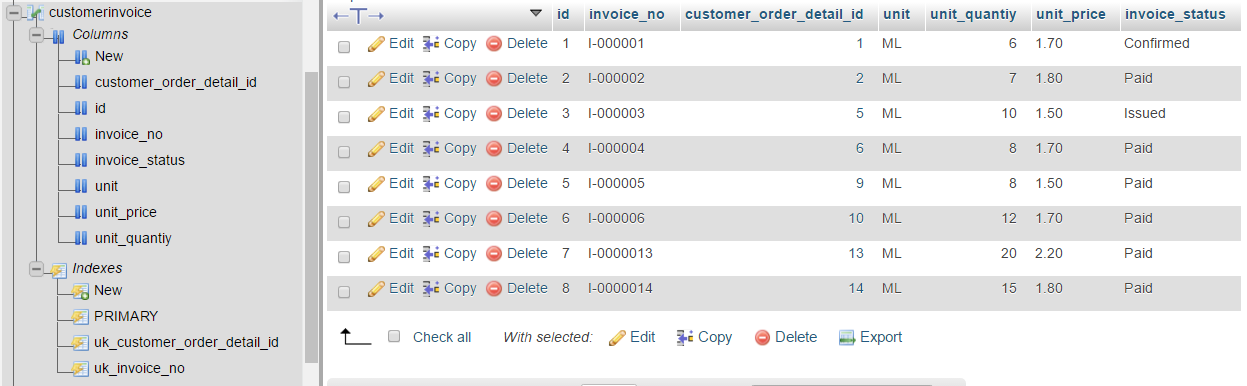
****

Figure 15 output data of customer invoice

Figure 16 Input data of customer invoice table

****

***Customer Order table*** contains columns such as: `id`, `customer\_id`, `vehicle\_driver\_id`, `order\_no`, `request\_date`, `schedule\_date`, `schedule\_status`, where table contains indexes as primary key (‘id’), unique key (‘uk\_order\_no’) and foreign keys (‘fk\_customer\_order\_customer\_customer\_id’) ‘fk\_customer\_order\_vehicle\_driver\_vehicle\_driver\_id’):

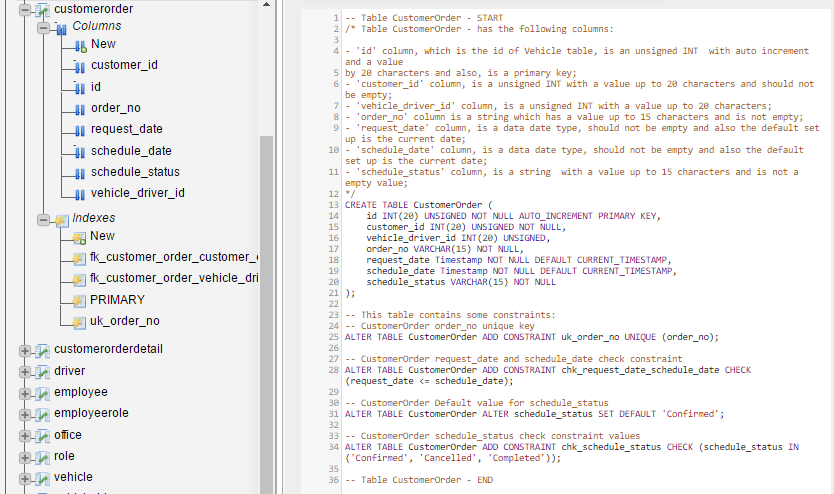


Figure 17 Input data for Customer Order table

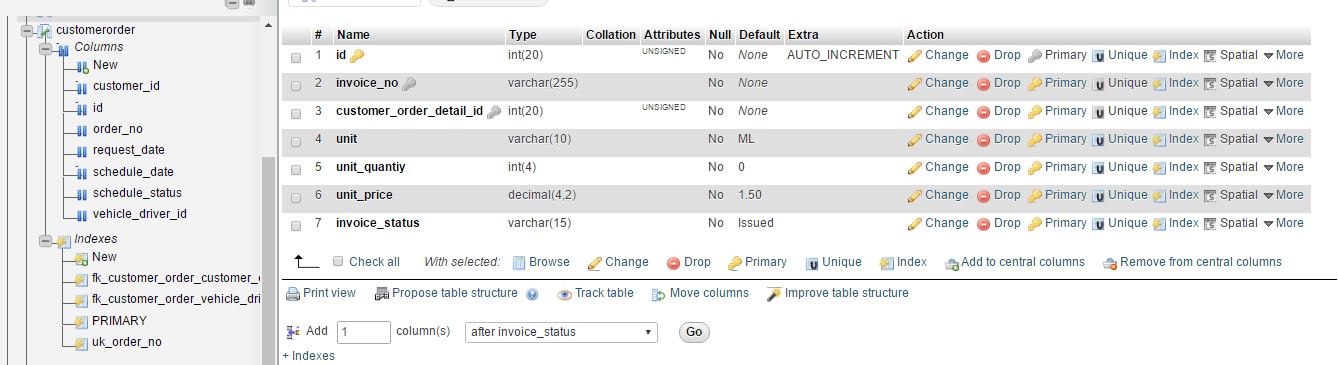
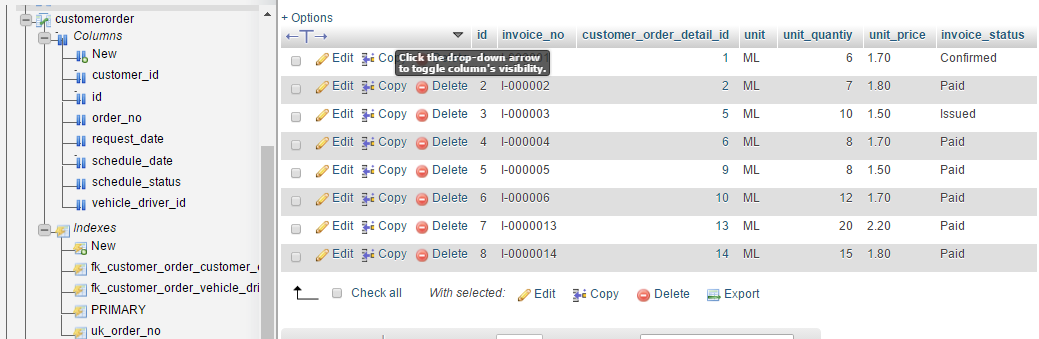


Figure 18 The structure of Customer Order table

Figure 19 The output data for Customer Order table

******

***Customer Order Detail Table*** contains columns such as: `id`, `customer\_order\_id`, `pickup\_address`, `destination\_address`, `distance`, `pickup\_date`, `arrival\_date`, where table contains indexes as primary key (‘id’), a unique key (‘**uk\_customer\_order\_id’**) and foreign key (‘**fk\_customer\_order\_detail\_customer\_order\_customer\_order\_id’**):

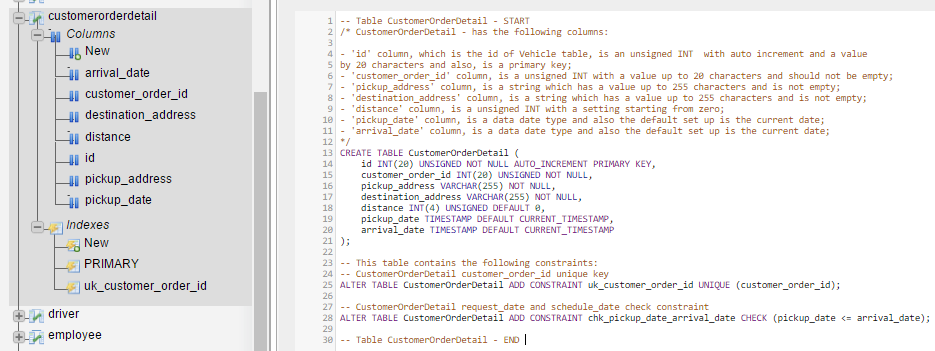
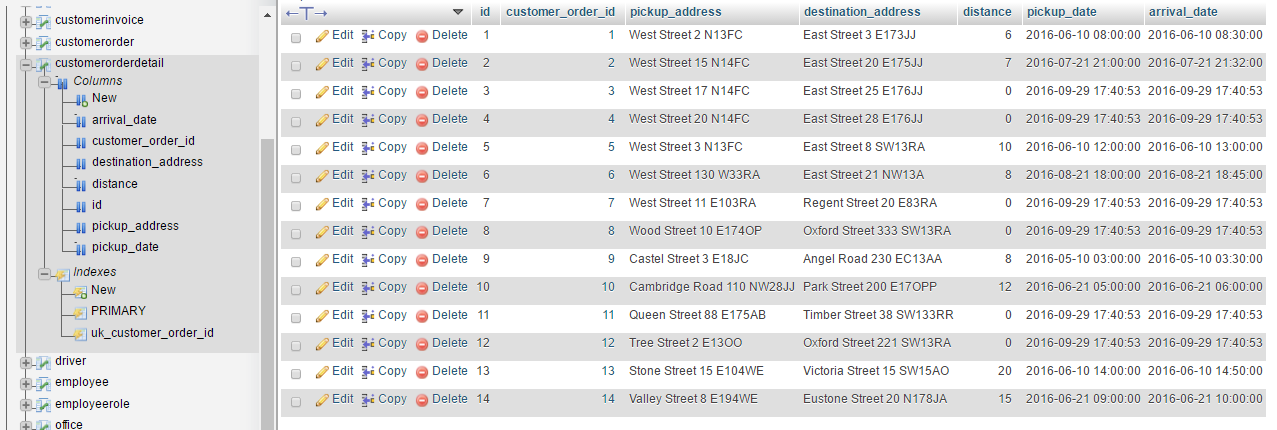
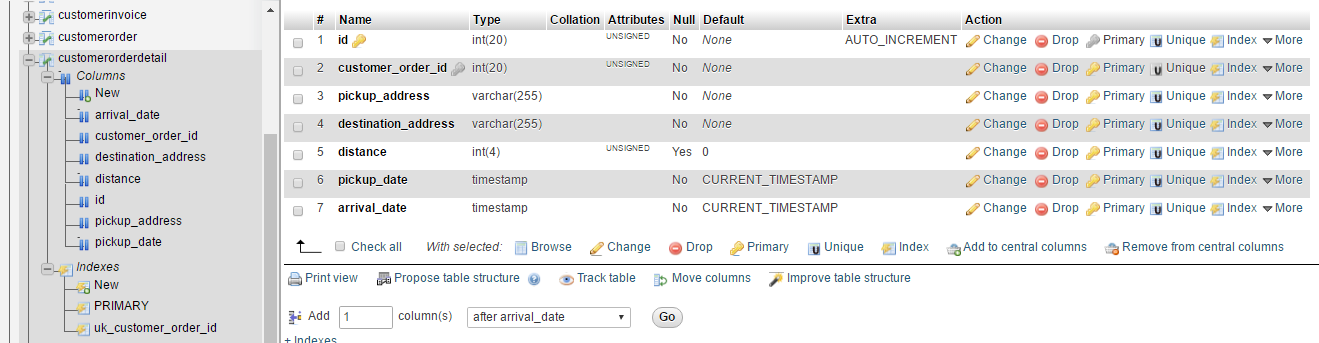


Figure 20 The structure of the Customer Order Detail

Figure 21 The output data for Customer Order Detail

Figure 22 The code for Customer Order Detail



***Driver table*** contains columns such as: `id`, `employee\_id`, `licence\_no`, `taxi\_licence\_no`, where table contains indexes as primary key (‘id’), some unique keys ('uk\_employee\_id', 'uk\_licence\_no' and 'uk\_taxi\_licence\_no') and foreign key ('fk\_driver\_employee\_employee\_id'):

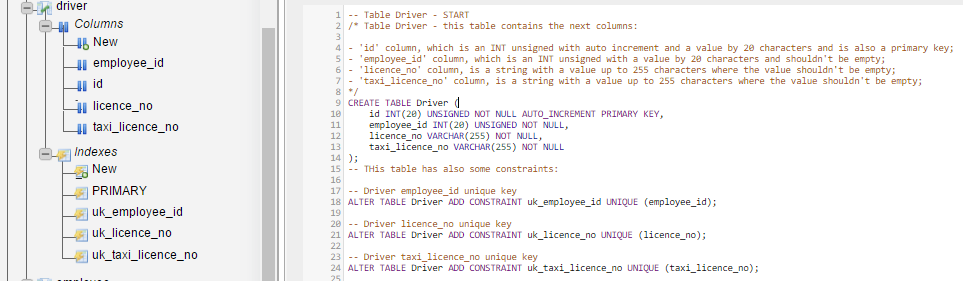


Figure 23 The input data for Driver table

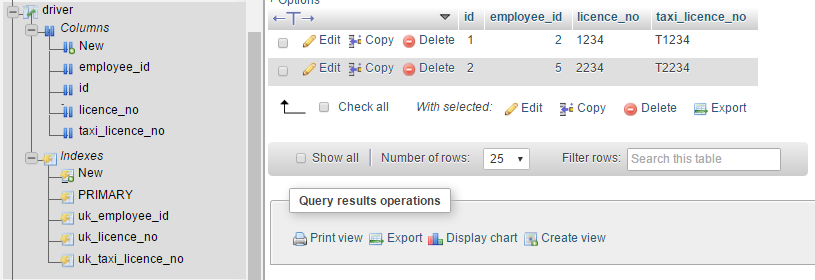


Figure 24 The output data for Driver table

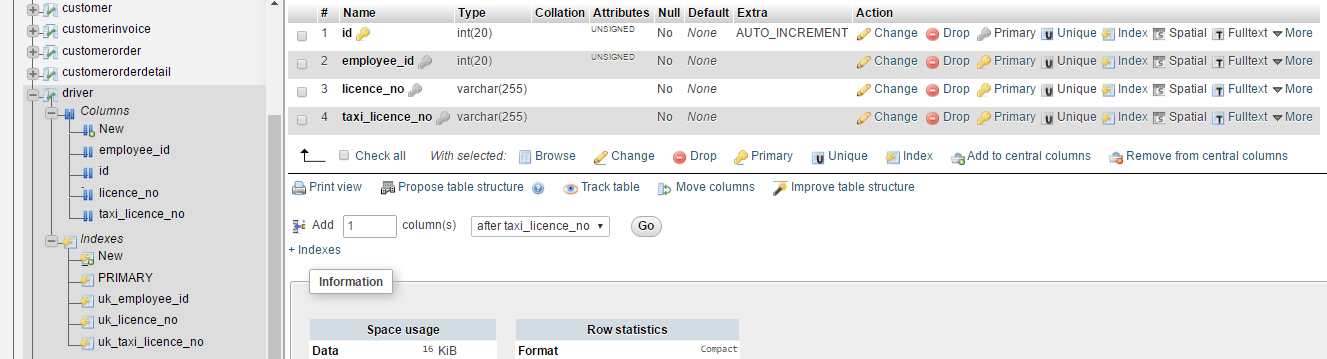


Figure 25 The structure of Driver table

***Employee table*** contains columns such as: `id`, `social\_id\_no`, `first\_name`, `last\_name`, `gender`, `phone\_no`, `email`, `address`, `birth\_place`, `birth\_date`, `office\_id`, where table contains indexes as primary key (‘id’), some unique keys ('uk\_social\_id\_no' and ‘uk\_social\_id\_no\_office\_id’) and foreign key (‘fk\_employee\_office\_office\_id’):

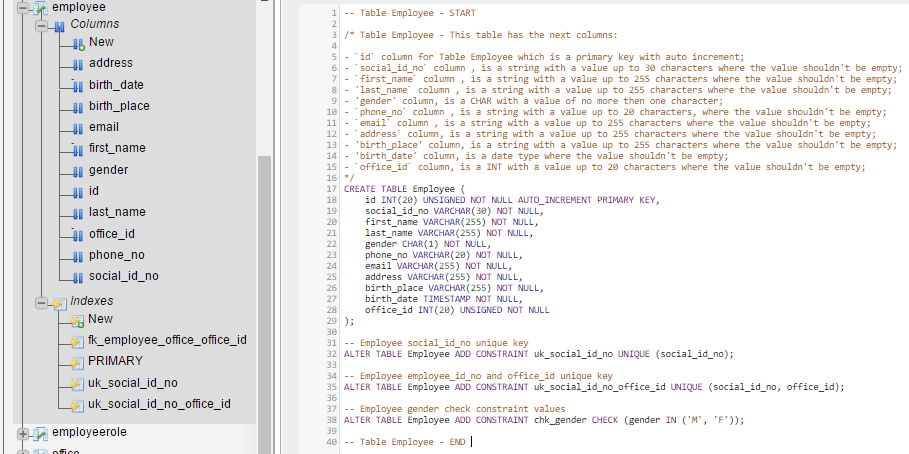


Figure 26 Input data for Employee table

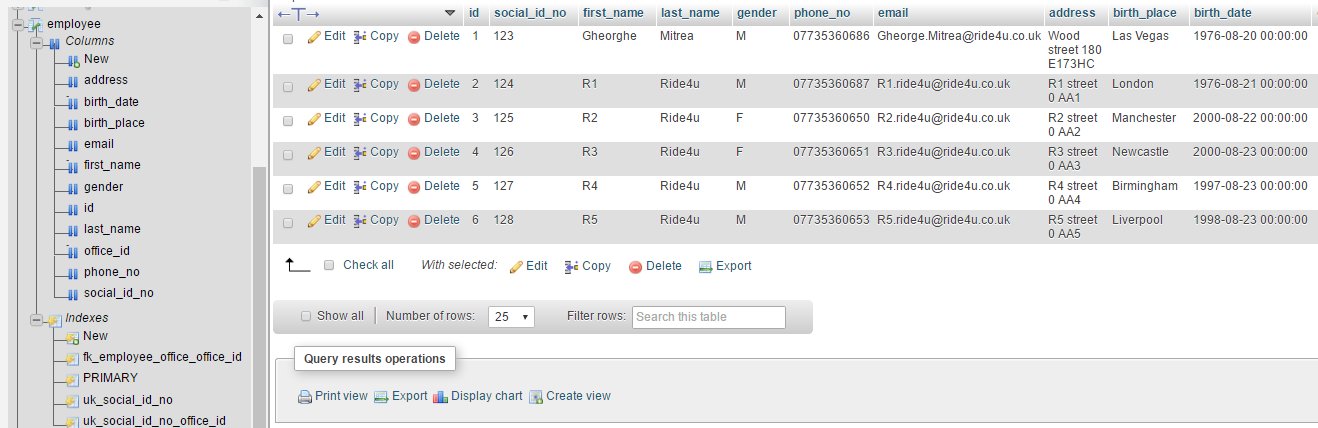


Figure 27 Output data for Employee table

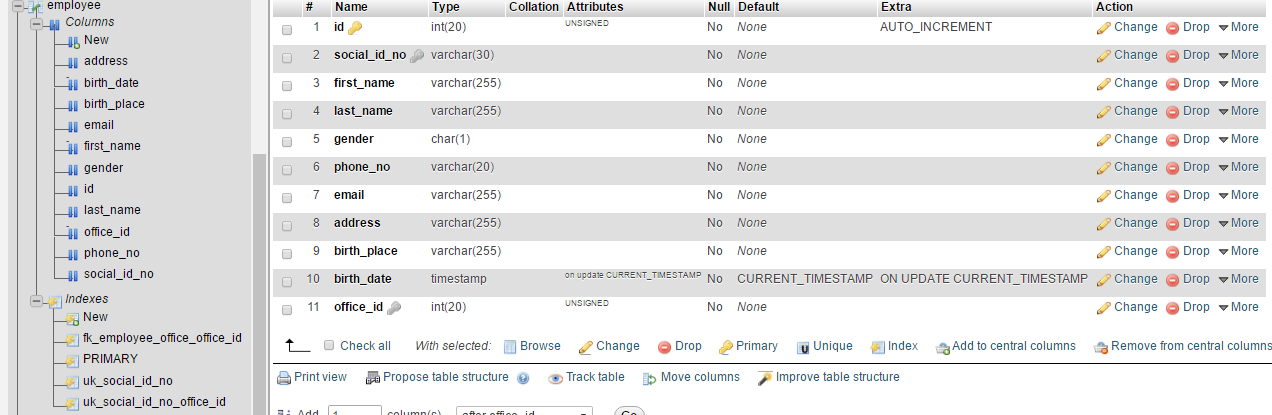


Figure 28 Structure of Employee table

***EmployeeRole table*** contains columns such as `id`, `employee\_id`, `role\_id`, where table contains indexes as primary key (‘id’), a unique key (‘uk\_employee\_id\_role\_id’) and foreign key (‘fk\_employee\_role\_role\_role\_id):

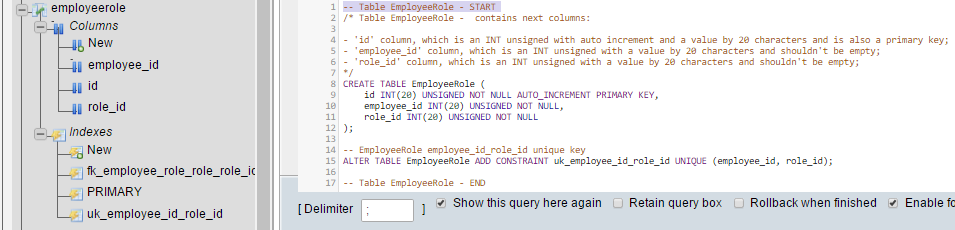


Figure 29 Input data for Employee Role table

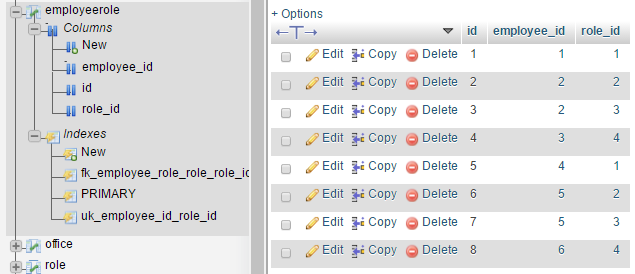


Figure 30 Output data for Employee Role table

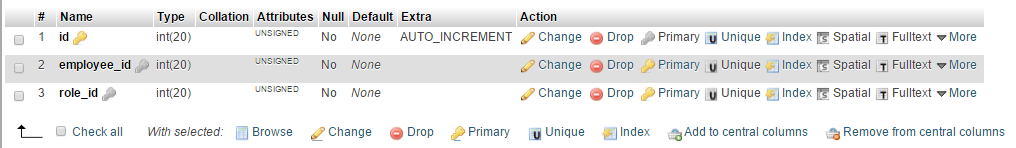


Figure 31 Structure of Employee Role table

***Office tablle:*** contains columns such as `id`, `office\_registration\_id`, `contact\_no`, `email`, `address`, `vat\_no`, where table contains indexes as primary key (‘id’), and a unique key (‘uk\_office\_registration\_id’):

Figure 32 Input data for Office table



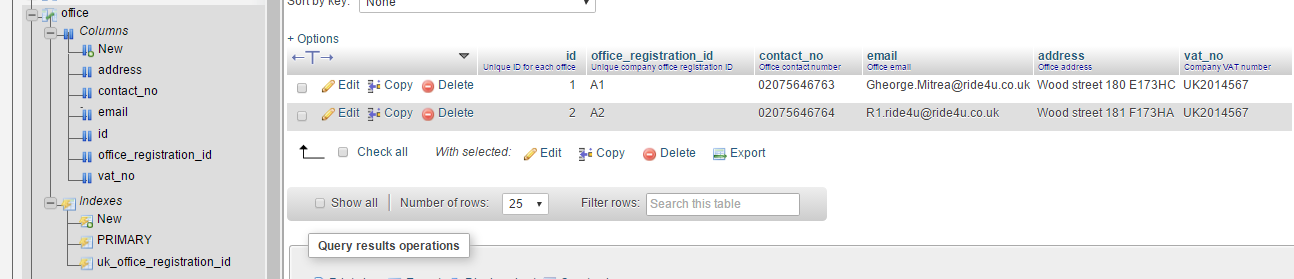


Figure 33 Output data for Office table

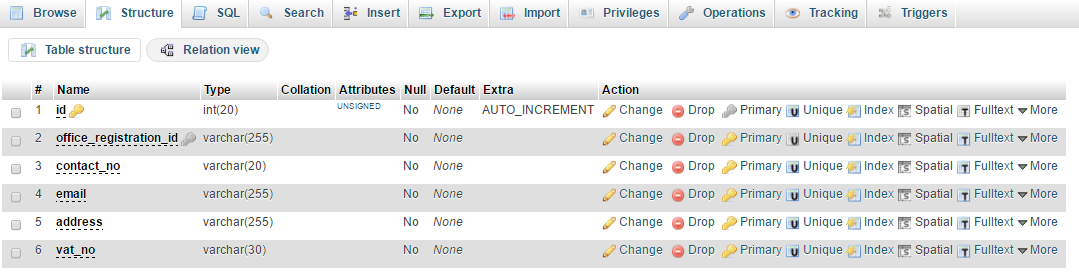
******

Figure 34 Office table structure

***Role table:*** contains columns such as `id`, `name`, where table contains indexes as primary key (‘id’), and a unique key (‘uk\_name’):

******

Figure 35 Role table with input data

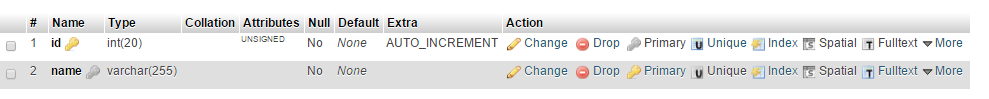
******

Figure 36 Structure of Role table

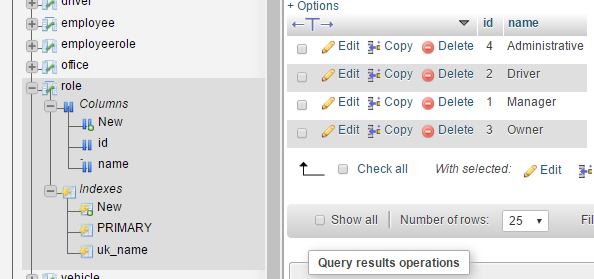
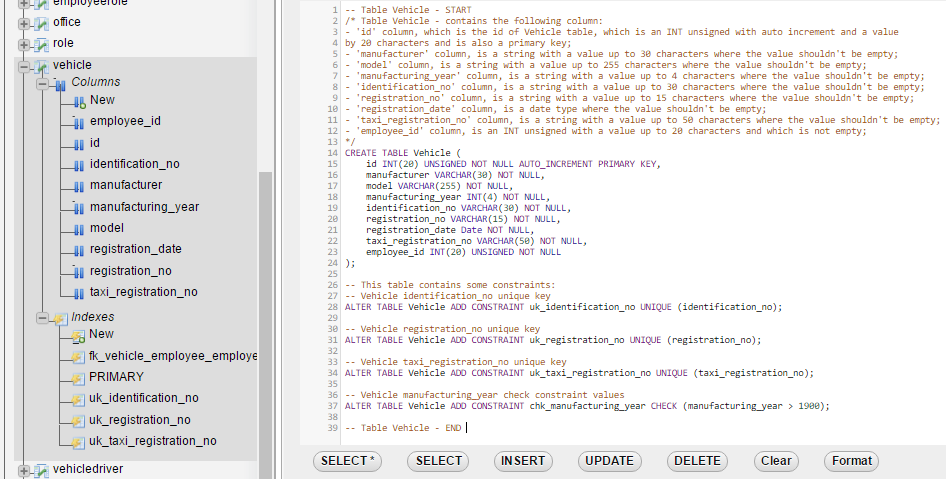


Figure 37 Output data for Role table

***Vehicle table:*** contains columns such as `id`, `manufacturer`, `model`, `manufacturing\_year`, `identification\_no`, `registration\_no`, `registration\_date`, `taxi\_registration\_no`, `employee\_id`, where table contains indexes as primary key (‘id’), and a unique key ('uk\_identification\_no', 'uk\_registration\_no' and 'uk\_taxi\_registration\_no'):

Figure 38 input data for Vehicle table



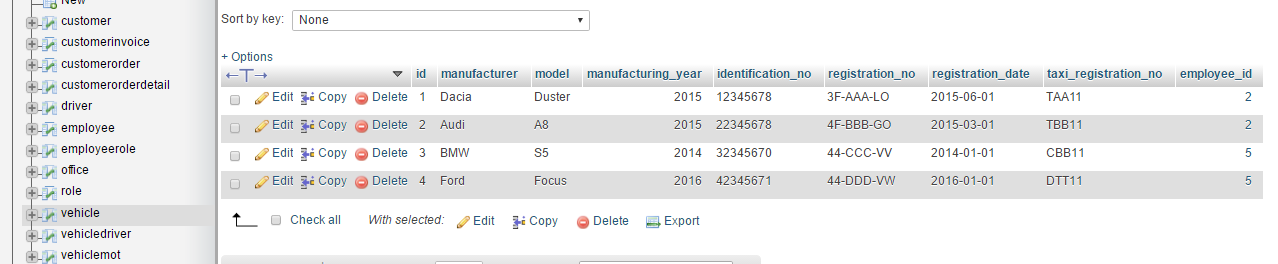


Figure 39 output data for Vehicle table

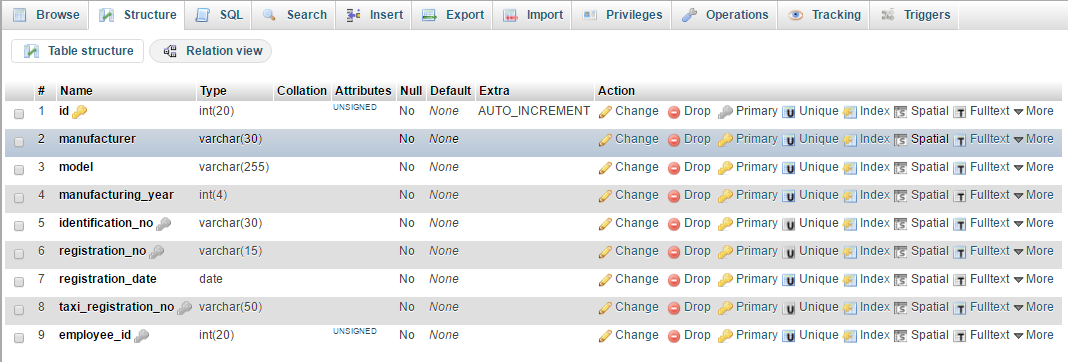


Figure 40 Vehicle Structure

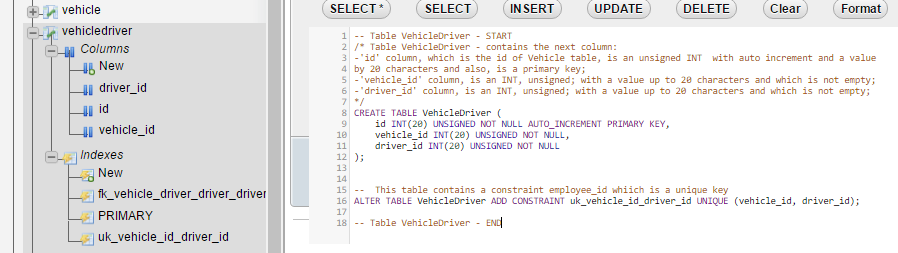
***VehicleDriver table:*** contains columns such as `id`, `vehicle\_id`, `driver\_id`, where table contains indexes as primary key (‘id’), a unique key (‘uk\_vehicle\_id\_driver\_id’) and a foreign key (‘fk\_vehicle\_driver\_driver\_driver\_id’):

Figure 41 the input data for Vehicle Driver table

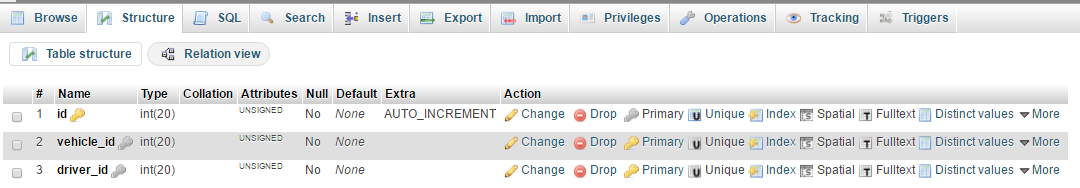
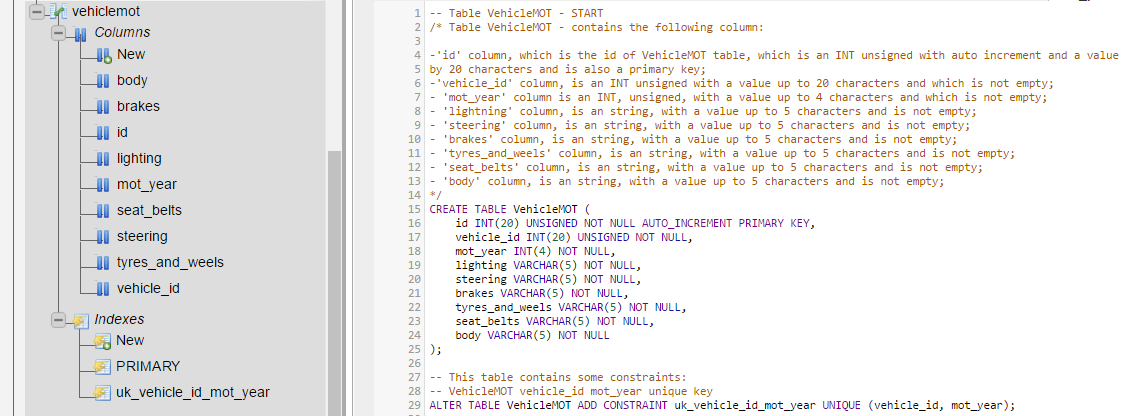
******

Figure 42 Structure of Vehicle Driver table

***VehicleMot table:*** contains columns such as `id`, `vehicle\_id`, `mot\_year`, `lighting`, `steering`, `brakes`, `tyres\_and\_weels`, `seat\_belts`, `body`, where table contains indexes as primary key (‘id’), a unique key ('uk\_vehicle\_id\_mot\_year') and a foreign key (‘fk\_vehicle\_mot\_vehicle\_vehicle\_id’):

Figure 43 Input data for VehicleMot table

******

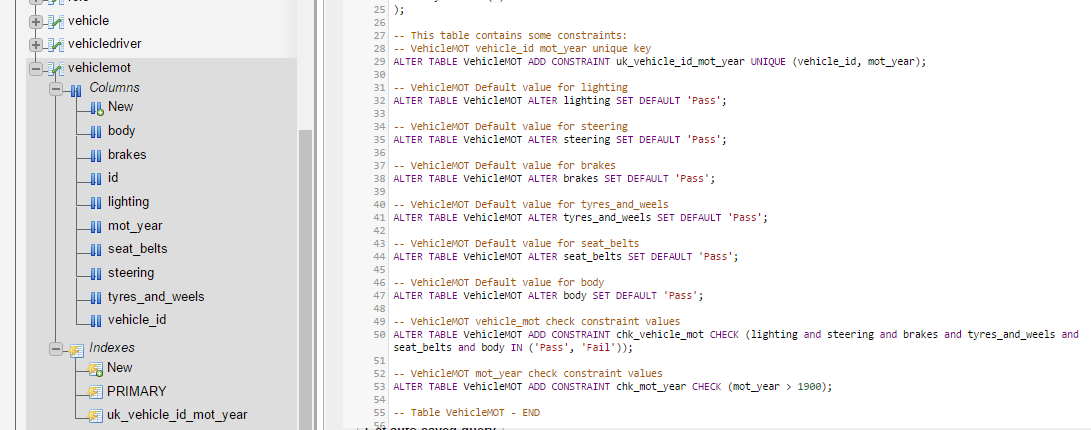
******

Figure 44 Input data for VehicleMot table

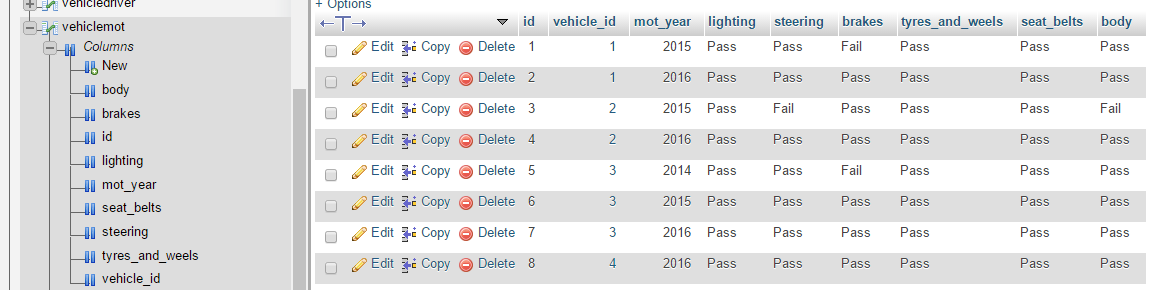
******

Figure 45 The output data for VehicleMot table

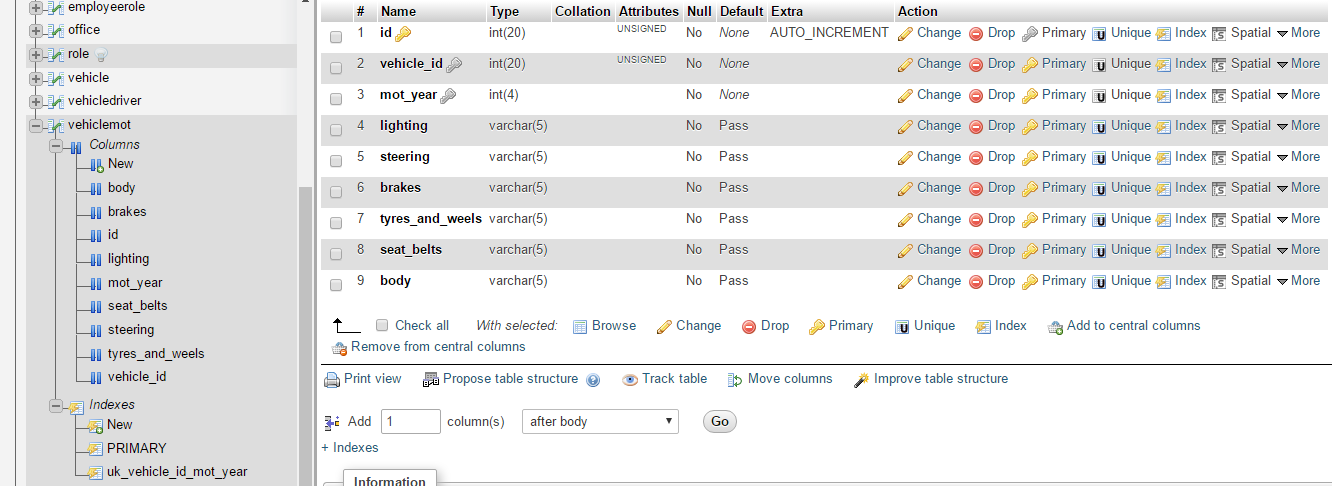
******

Figure 46 Structure of VehicleMot table

1. ***Populate the tables by inserting the data***

INSERT is the command for entering data into the database. The most common way is taking their input in a form suitable database structure. The syntax for entering data into the database table is:

INSERT INTO tablename (column\_1, column\_2 ..., column\_n) values ('value\_1', 'value\_2', ..., 'value\_n');

Examples of introducing data in my MySQL database:

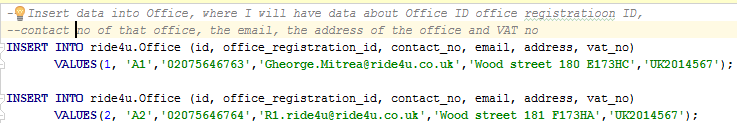
* Insert data into Office, where I will have data about Office ID office registration ID,  
  contact no. of that office, the email, the address of the office and VAT no.:

Figure 47 Insert data into Office table

* Insert data into Employee table where I will have data about employees such as id, social\_id\_no, first\_name, last\_name, gender, phone\_no, email, address, birth\_place, birth\_date, office\_id. I have six employees, so I have to introduce data for each of them:

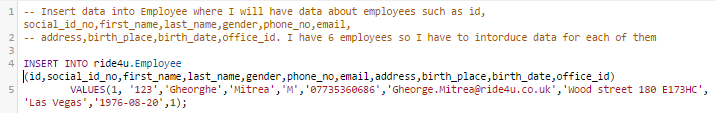


Figure 48 Insert data into Employee table

* Insert data into CustomerOrder will contain data related to the order of a customer when he books a taxi. All these data are related to customer order id, the customer id, the vehicle driver id inside the company, the order no, the request date for that taxi booking, the scheduled date for that order and the status of that customer order which can be cancelled when the customer appointed a cab but he did not attend it; completed status when the company completed the payment process and confirmed status when the customer took the cab, but the payment process did not complete moreover, is still ongoing for the complete status:



Figure 49 insert data for CustomerOrder

## 3.2 Testing of database system

Ride4U organization needs to test the database for error and lack of functionality so that improvements can be made to achieve desired objectives with new system. Various types of testing can be implemented to ensure the fulfilment of database objectives for Ride4U.

**Black Box testing**

During the black box testing phase, interfaces and integration of various database modules are tested for their interoperability and functioning(Curino, 2011). Black box testing is related to test a system for major objectives without dealing with internal processing of system. Following steps are followed to ensure the right functioning of database:

* Data mapping in processes
* Verification of input data
* Query based checking of output results
* Process monitoring
* Impact of module integration on other module of system.

***Test regarding Update data in ride4u database***

We can change the value of a registration database for a particular id with the following syntax:

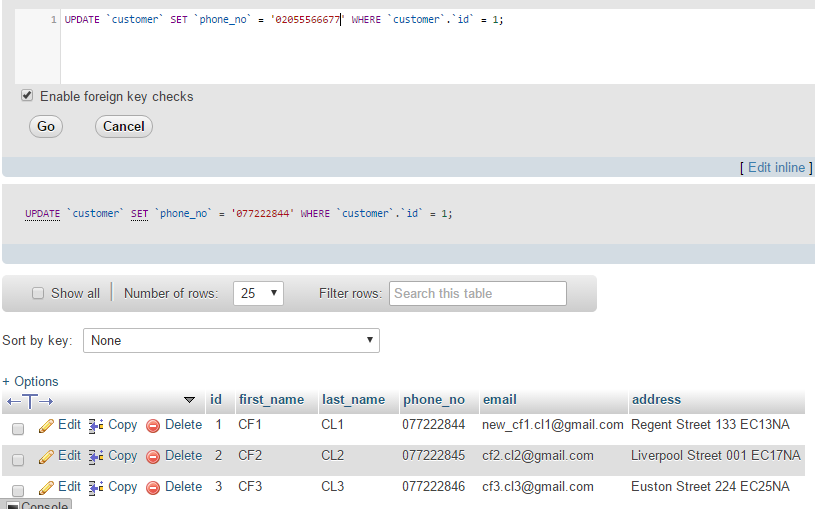


Figure 50 Update data for customer id 1 with the old value for the phone no.

After inserting the code for updating the data of phone\_no from customer id 1 we will have a message where it tells us that “Your SQL query has been executed successfully. 1 row affected”:



Figure 51 Message regarding the update which I have updated for column phone\_no from customer id 1 in customer table

Then I will check if the update is set up to the new data which I wanted and is showing me the correct data of the update:



Figure 52 Correct data after update data for phone no. of customer id 1

Black box testing is fairly simpler and cost effective in comparison of other testing technologies but is also less capable to find the right functioning of modules to a specific task. It is not used to check performance and quality related results but preferred to obtain the working system with desired result sets.

**White Box testing**

In comparison of black box testing, white box testing is deep testing technology, mainly related to validate the internal working of database system. Users are normally not aware to this type of testing. Following steps may be followed during white box testing of database system:

* Testing of logical views and events on data participating in data refactoring process
* Each module is tested for the database objectives like functions, data, processing, queries etc. are validated and checked for right results from each(Coronel, 2016).
* Validation of data types and tables, models and schemas to ensure the data consistency.
* Checking of data integrity in each operation on database.

It is comparatively slower and costly but much effective to ensure the database consistency and integrity with transactions and operability.

**WHODATE Technique**

It is advance form of traditional white box testing in which SQL statements are transformed into GPL statements to generate test cases. WHODATE stands for white box database application technique. Each SQL statement is independently converted to GPL to ensure the query semantics and test results. Ride4U will make use of WHODATE to check the database for errors and accidentally generated right results.

Addition to above testing methods, these steps also can be followed for various improvement purposes:

**Coding-error:** Most of coding error will be identified during black box and white box testing. If any of the remains, it can be eliminated by giving number of required parameters to system in generation of test results. So coding error will be easily identified during testing when data is not received in accurate form or not stored in consistent manner.

**Functional requirements:** designed system can be tested for functional requirements. For that all the parts or module of DBMS can be joined together to achieve organizational goals of Ride4U. Organization needs to use the system with real users and data to identify the errors and requirements those are lacking from the system design(Xu, 2010). This is required to find major features from whole system rather than to modules.

**Usability:** New system can be tested for its usability with changes coming in communication and management technologies. Thus, new system can be used for a few months can identify the real usability of system as it will uncover the fields like:

* Integration with technology in communication
* Handling of more data
* Robustness to erroneous activities of users.
* Security form cyber threats etc.

**Consistency:** Database can be checked for its data types in tables. Also various erroneous inputs can be prepared to check the data consistency in multiple transactions in network of Ride4U. This is required to keep all the data values safe and consistent from modifications.

**Integration:** Ride4U and DBTech can check database system for its integrity to other third party application used by organization to regulate the business. Thus, system should be check and validated for integration in current working context.

**Acceptance testing:** Database and interfaces can be tested with user opinions to find the acceptance of system in environment. Ride4U can feedback the system for layout and interface, complexity, functionality and performance to achieve a more acceptable form of current version of database system.

I will come with some tests for the Queries to explain how is working my database

***Queries:***

1. The names and phone numbers of the Managers at each office location.

Figure 53 The names and phone numbers of the Managers at each office location



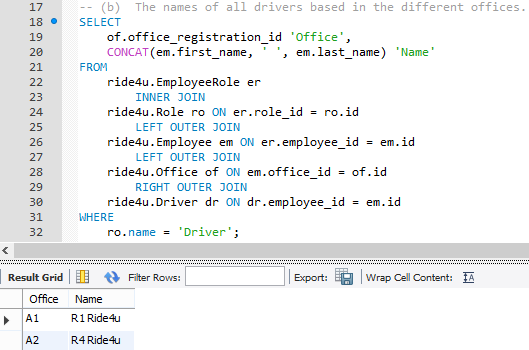
1. **The names of all drivers based in the different offices.

Figure 54 The names of all drivers based in the different offices

1. The total number of staff at each office.

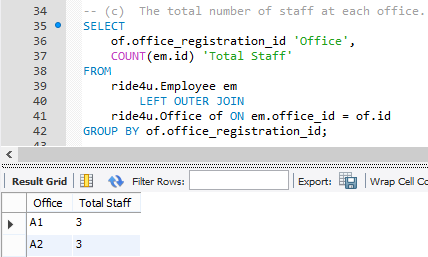
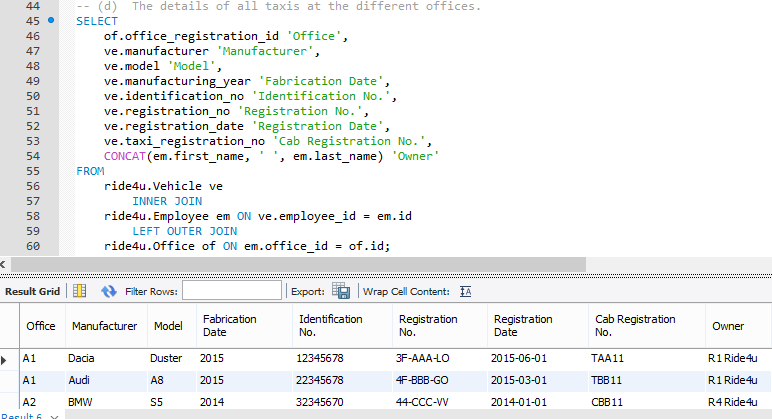
**

Figure 55 The total number of staff at each office

1. The details of all taxis at the different offices.

Figure 56 The details of all taxis at the different offices

**

1. The number of drivers allocated to each taxi.

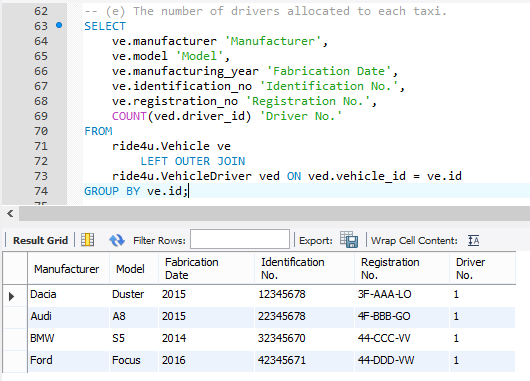
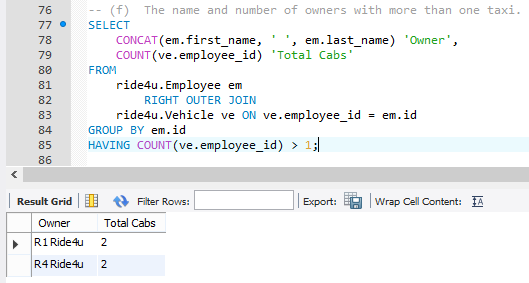
**

Figure 57 The number of drivers allocated to each taxi

1. The name and number of owners with more than one taxi.

Figure 58 The name and number of owners with more than one taxi

**

1. The full address of all clients.

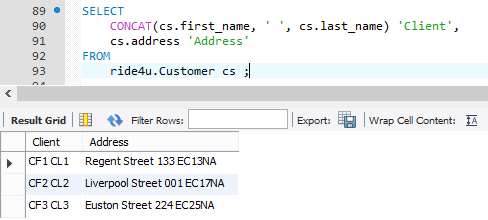
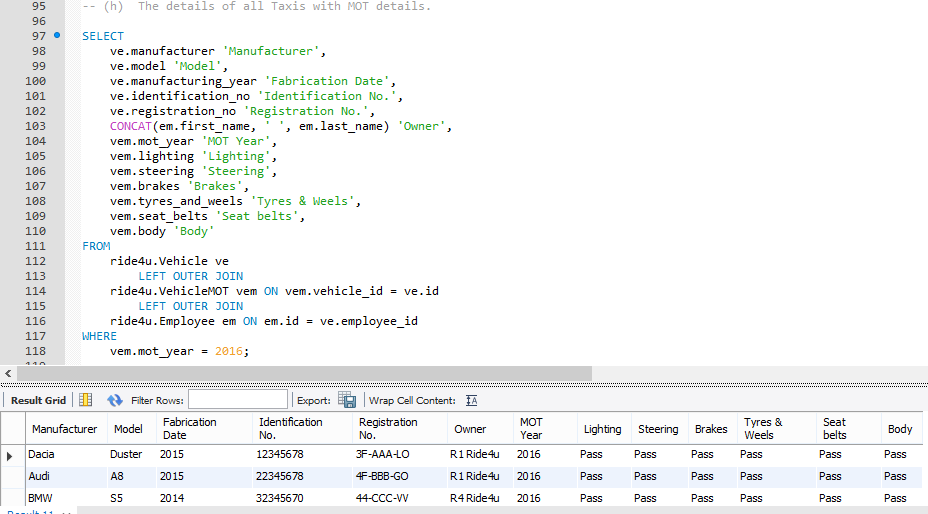


Figure 59 The full address of all clients

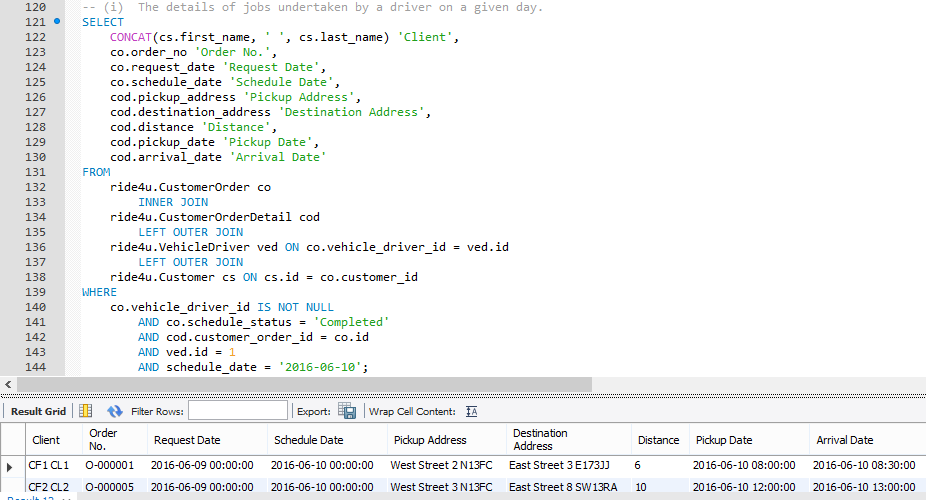
1. The details of all Taxis with MOT details.

Figure 60 The details of all Taxis with MOT details

**

1. The details of jobs undertaken by a driver on a given day.

Figure 61 The details of jobs undertaken by a driver on a given day

**

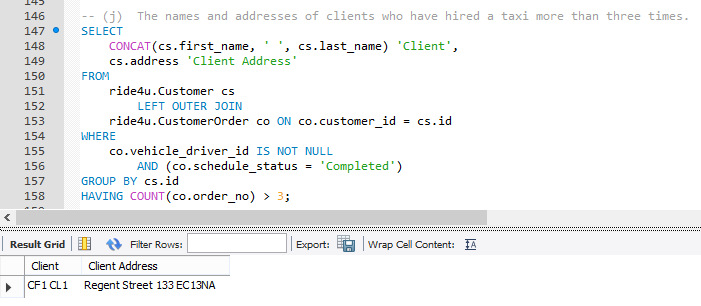
1. The names and addresses of clients who have hired a taxi more than three times.

Figure 62 The names and addresses of clients who have hired a taxi more than three times

1. The average number of jobs per driver.

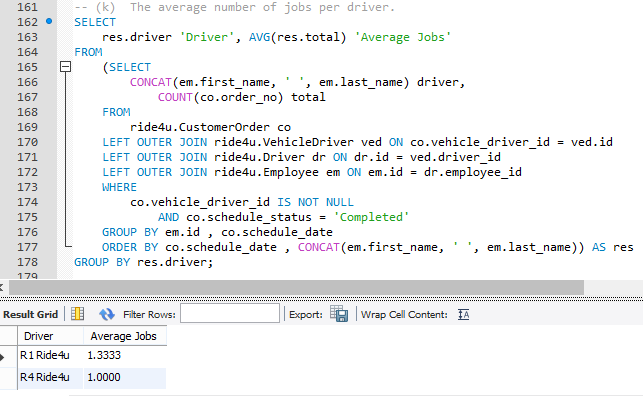
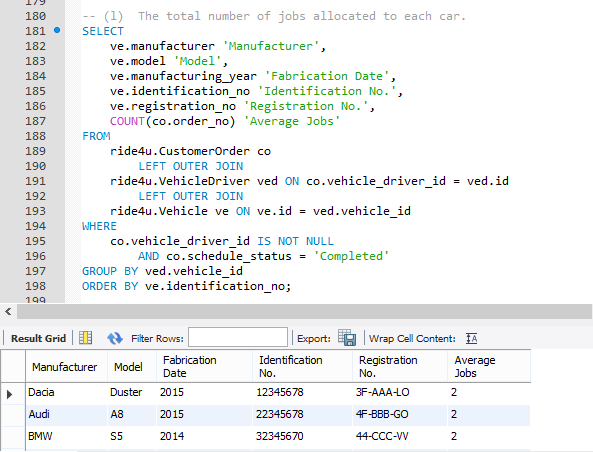


Figure 63 The average number of jobs per driver

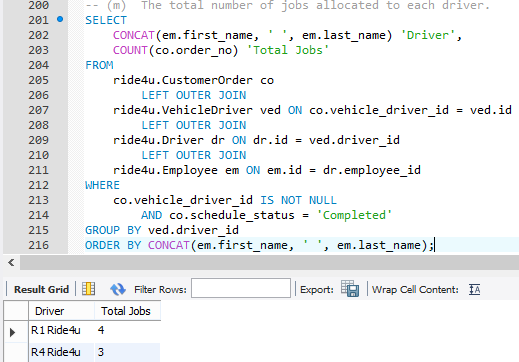
1. The total number of jobs allocated to each car.

Figure 64 The total number of jobs allocated to each car

**

1. The total number of jobs allocated to each driver.

Figure 65 The total number of jobs allocated to each driver

**

1. The total number of jobs and miles driven for a given contract.

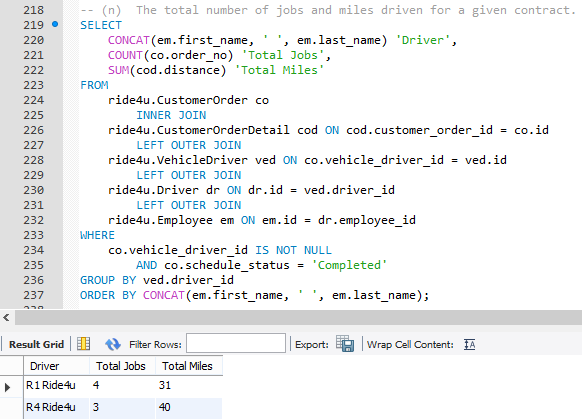


Figure 66 The total number of jobs and miles driven for a given contract

## 3.3 Validation and Verification techniques

Validation and verification techniques prove to be helpful in removing the unwanted data from the database such that the database does not contain any error(Bellatreche, 2011). For example, mail, mobile and date attributes can be typed strongly to accept desired format and values.

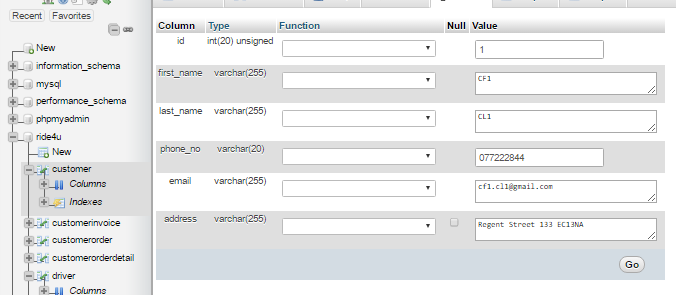


Figure 67: Customer table with data types

Also, forms and check boxes should be configured properly at development end to ensure that data is receiving in right format and prompting the users for wrong values, for this constrain I have an example when I try to introduce a new customer with an already employee of an office Ride4u company, the result will prompt with a message that there is a duplicate of introducing data of employee:

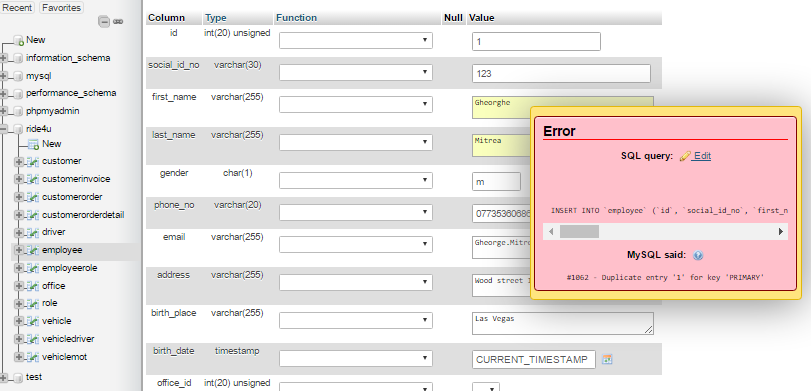


Figure 68 Validation data when I try to duplicate the data in a employee table b introducing a new employee

Another constrain to validate my data is when I try to change the value of a primary key from driver.id is coming with a message that is not possible to change the value for this data because driver.id is a transitory element for employee.id, that means if I want to change the value key for driver.id I need to change first the value of employee.id which is the super key in my database, here is the example:

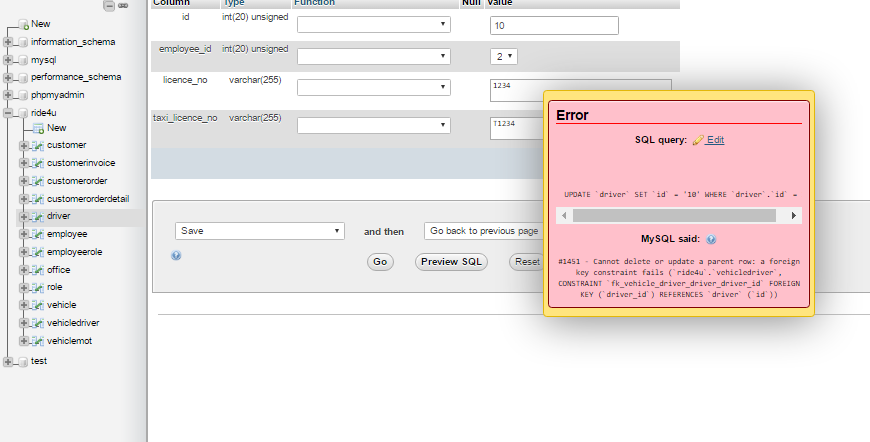


Figure 69 cannot delete or update a parent row for a foreign key

## 3.4 User manual and technical documentation

**User manual**

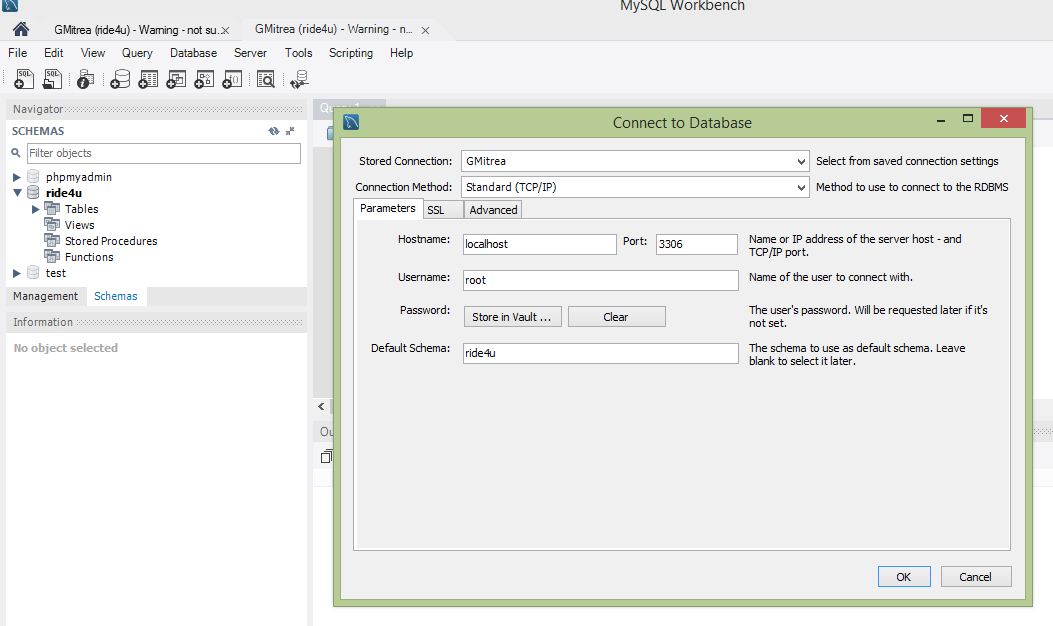
The following manual is designed for database system which is being implemented in the Tom and Jerry organization. The manual will assist the users in starting a new interface and features in the system.

*Chapter 1: Introduction*

The database is designed in the Tom and Jerry organization in order to keep the data systematic and safe. The new implemented database system would help the organization in removing the errors and difficulties in data values and data items(Woudenberg, 2010).

*Chapter 2: Main screens*

To access the database, user ID and password assigned by the administration has to be entered in the login screen. The following is the login screen of the database:

Figure 70: Login screen

*Chapter 3: Common FAQs*

1. How to print an invoice.

Answer: Go to *main screen*, select *find invoice* and enter the invoice number and click *print*.

1. How to find specific customer name and mail address.

Answer: Click *Find by name* or *mail* in main screen and type the parameter(Chodorow, 2013).

**Technical Specifications**

This would help the developer in upgrading and maintaining the database system. It contains the explanations in textual and graphical form. The developers are suggested to append their details if any required changes are made in the database system.

*Chapter 1: Framework details*

Windows 2010 or Linux

Microsoft Access 2010 or later

620 MB RAM

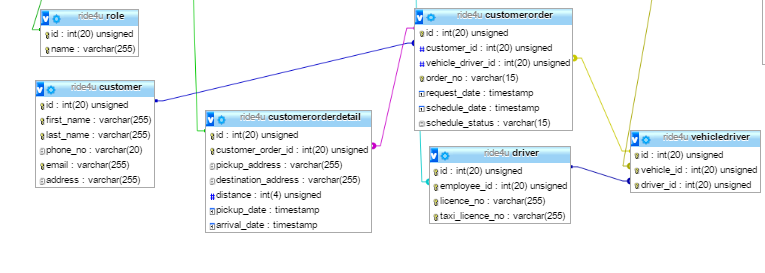
1.3 GHZ processor (64 bit is recommended)

16 GB storage

Internet connection

*Chapter 2: database configuration*

Database contains tables for product, customers, payment and invoice. Each table contains the unique relationship with other as shown in diagram(Aranda, 2013).

Figure 71: Relationship among attributes

*Chapter 3: Security*

The default security of database is the password provided by the administrator and user accounts and groups are configured to access on database and tables. The dual nature security is implemented.

# CONCLUSION

The report has been concluded the basic detail of database management system along with the design of new database system for Ride4U organization. The report has been determined the design mythology for development of database.ER diagrams and normalization process is described for database system. Later, reports have been identified various testing and implementation methods for new system. User manual and technical specification is also provided for newly designed database system.

# Bibliography

Adnan, M. S. A. a. L. P., 2010. *Developing efficient web-based GIS applications,* s.l.: s.n.

Antonakopoulos, S. F. S. a. Z. L., 2010. *Power-aware routing with rate-adaptive network elements,* s.l.: In 2010 IEEE Globecom Workshops (pp. 1428-1432). IEEE.

Aranda, B. A. P. A.-F. Y. A. I. B. A. D. C. F. M. G. A. K. S. K. J. a. K. J., 2013. *The IntAct molecular interaction database in 2010,* s.l.: Nucleic acids research, 38(suppl 1), pp.D525-D531.

Bellatreche, L. B. S. G. A. C. A. a. C. A., 2011. *Verification of partitioning and allocation techniques on teradata DBMS,* s.l.: In International Conference on Algorithms and Architectures for Parallel Processing (pp. 158-169). Springer Berlin Heidelberg..

Bog, A., 2014. *Evaluation of Mixing the Workload and Variation of the Database Schema. ,* s.l.: In Benchmarking Transaction and Analytical Processing Systems (pp. 121-131). Springer Berlin Heidelberg..

Borgatti, S. a. H. D., 2011. *On network theory,* s.l.: Organization Science, 22(5), pp.1168-1181.

Chodorow, K., 2013. *MongoDB: the definitive guide,* s.l.: O'Reilly Media, Inc..

Coronel, C. a. M. S., 2016. *Database systems: design, implementation, & management.,* s.l.: Cengage Learning.

Curino, C. J. E. M. S. a. B. H., 2011. *Workload-aware database monitoring and consolidation,* s.l.: In Proceedings of the 2011 ACM SIGMOD International Conference on Management of data(pp. 313-324). ACM..

Elmasri, R. a. N. S., 2011. *Database systems (Vol. 9),* s.l.: Pearson Education.

Espina, J. F. T. P. A. S. L. M. O. a. Y. G., 2014. *Network topologies, communication protocols, and standards,* s.l.: In Body sensor networks (pp. 189-236). Springer London.

Fattah, S. M. M. a. A.-E. L., 2013. *An Adaptive Hybrid Controller for DBMS Performance Tuning,* s.l.: s.n.

Fattah, S. M. M. a. A.-E. L., 2013. *An Adaptive Hybrid Controller for DBMS Performance Tuning,* s.l.: s.n.

Febbraro, O. R. K. a. R. F., 2011. *ASPIDE: Integrated development environment for answer set programming. In International Conference on Logic Programming and Nonmonotonic Reasoning (pp. 317-330).,* s.l.: Springer Berlin Heidelberg..

Halpin, T. a. M. T., 2010. *. Information modeling and relational databases,* s.l.: Morgan Kaufmann..

Ihm, S. P. K. a. P. V., 2010. *Wide-area Network Acceleration for the Developing World,* s.l.: In USENIX Annual Technical Conference.

Lin, C. T. Y. a. Y. M., 2011. *Green network and green evaluation: mechanism, modeling and evaluation,* s.l.: Jisuanji Xuebao(Chinese Journal of Computers), 34(4), pp.593-612..

Özsu, M. &. V. P., 2011. *Principles of distributed database systems,* s.l.: Springer Science & Business Media.

Pahlavan, K., 2011. *Principles of wireless networks: A unified approach,* s.l.: John Wiley & Sons, Inc.

Pandya, K., 2013. *Network Structure or Topology,* s.l.: International Journal of Advance Research in Computer Science and Management Studies, 1(2)..

PAŞCU, P., 2010. *The Stages of Implementation of the SSADM system in the Government Institutions.,* s.l.: Journal of Applied Computer Science & Mathematics, (8), p.4.

Pathak, P. a. D. R., 2011. *A survey of network design problems and joint design approaches in wireless mesh networks,* s.l.: IEEE Communications surveys & tutorials, 13(3), pp.396-428..

Rock-Evans, R., 2014. *Data Modelling and Process Modelling using the most popular Methods: Covering SSADM, Yourdon, Inforem, Bachman, Information Engineering and'Activity/Object'Diagramming Techniques.,* s.l.: Butterworth-Heinemann.

Sanchez, F. D. Z. a. D. Y., 2012. *Blocking spam by separating end‐user machines from legitimate mail server machines,* s.l.: Security and Communication Networks..

Sehgal, P. T. V. a. Z. E., 2010. *Evaluating Performance and Energy in File System Server Workloads,* s.l.: In FAST (pp. 253-266)..

Shapiro, C. a. V. H., 2013. *. Information rules: a strategic guide to the network economy,* s.l.: Harvard Business Press.

Subramanian, M., 2010. *Network management: principles and practice,* s.l.: Pearson Education India.

Sun, Z., 2013. *User Involvement in System Development Process.,* s.l.: Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE 2013) Paris: Atlantid Press..

Thalheim, B., 2013. *. Entity-relationship modeling: foundations of database technology,* s.l.: Springer Science & Business Media..

Varshney, G., 2012. *Database Management System,* s.l.: Global Vision Publishing House.

Winter, T., 2012. *RPL: IPv6 routing protocol for low-power and lossy networks.,* s.l.: s.n.

Wolf, T. G. J. C. K. D. R. R. G. B. I. a. N. A., 2012. *Choice as a principle in network architecture,* s.l.: ACM SIGCOMM Computer Communication Review, 42(4), pp.105-106..

Woudenberg, S. C. B. O. B. L. E. T. J. a. W. K., 2010. *The Forest Inventory and Analysis Database: Database description and user’s manual version 4.0 for Phase 2,* s.l.: . Gen. Tech. Rep. RMRS-GTR-245. Fort Collins, CO: US Department of Agriculture, Forest Service, Rocky Mountain Research Station..

Xu, Z. T. Y. a. W. X. 2., 2010. *Exploring power-performance tradeoffs in database systems,* s.l.: In 2010 IEEE 26th International Conference on Data Engineering (ICDE 2010) (pp. 485-496). IEEE.