INFORMATION SYSTEMS PROJECT

RAYS RENTALS

Maryam Elgahmi | 12009935

Mark Bellingham | 14032098

Janet D'Souza | 14059185

Contents

Introduction

The purpose for Ray Rentals report is to solve the problems with the paper based system which was inefficient and had problems with stock control and double bookings. This report includes Ray Rentals system requirements for a future computerised system. In addition, it includes the different type of management reports and data enquires. This report aims to establish the flow of data through the Ray's Rentals system and to explain the different aspects via the use of Use Case Diagrams, Entity Relationship Diagrams from both top-down and bottom-up perspectives, and through Relational Data Analysis or Normalisation. The entity relationship diagram represents the conceptual level and relational database is the logical level for the database. The database has been created and populated with the business details and some useful queries included. The group members have equally participated in this project.

A comprehensive description of the problems with the current paper based system at Ray's Rentals

- Current system is paper based.
- Has potential to be lost or mislaid. Some records take up more than one sheet increasing risk.
- Notes are hand-written which can sometimes be difficult to read.
- No centralised way to determine which bikes need servicing, which means some are missed.
- Slow to deal with enquiries from the customer, which could result in lost sales.
- Slow to deal with enquiries because someone has to call back the customer with answer to query which costs a significant amount of time, and money for calls
- Information given to customer is hand-written, which looks unprofessional and could be hard to decipher.
- No way to issue reminders of advance bookings, relying on memory and someone checking the records, which can result in double bookings.
- Price lists and other information cannot be updated and printed in-house which leads to hand-written amendments, which can be inaccurate and/or hard to read and also looks unprofessional.
- Current system relies on paperwork being updated but if that paperwork is temporarily unavailable for any reason, updates could be missed.
- Paper is fragile, it can easily be damaged and deteriorates over time.
- Paper records take up a lot of space.
- Security issue with taking customer bank details.
- Business security because paperwork cannot be encrypted like computer files.
- Inefficient system means that bikes are not rotated properly.
- Inefficient system means that the stockroom is not organised properly leading to overstock on some parts and other parts are missing.
- Stock ordering system is not centralised leading to problems with accounting.
- Inefficient filing system, which is slow and sometimes problematic for accounting.
- No way to analyse business records to see where improvements can be made.
- No way to bring up records of customers for purposes of sending out special offers etc.

System requirements for a proposed new computerised database system

- Able to keep records of:
    - each bicycle
    - the customers

  - rental bike sales
  - bike maintenance
  - booking system
- all items used to repair the bicycles, and which items are in or out of stock
- parts ordered and received and to check one against the other
- Browsing to check the availability of booking for a given date.
- Ordered and available stock information is easily accessible.
- Issue reminders when bike maintenance is due.
- Not allow customers to book bikes that are due in the workshop.
- Not allow double bookings.
- Print off details of bookings and issue receipts to the customer.
- Create reports for date ranges.
- Stockroom is searchable to see which items are out of stock so that they can be reordered.
- Print an up-to-date price list to give to customers who are just enquiring.
- Create regular, automatic, offsite backups of the database.
- User friendly, easy to use for people who are not computer literate.
- Keep a record of how much each bike is used so that all bikes are used regularly.
- Be secured by means of password to keep customer data safe.

Data enquiries, management reports and the different types of management reports that are used.

Data Enquiries:

A data enquiry is a query, a request for a specific piece of information to deal with a particular issue. It is often utilised as an application from a business, to be completed by a client or customer. They may need to know if a particular item of stock is available. Data Enquiries are searches of day to day planning information related to the product of the organisation and are undertaken by the operational level staff. Before we can carry out any work on behalf of any organisation, finding out the required data to proceed with the undertaken work is very important. Enquiries can be made by key or other search terms if the key is not known. The result will probably be shown on the screen [Whiteley, D. 2013] but can also be printed.

Management Reports:

Managerial reports are specifically designed to aid management in decision making. They should not contain too much detail, they should be analytical by nature and they should link up with other applications where possible [Eccles, M., Julyan, F., Boot, G. and van Belle, J. 2004:570]. They will help to determine where the business needs to cut expenses and focus on developing future products or services. Brand awareness and marketing reports may include detailed information about customers or an organisation's profit and loss by department, clients, products and geographical regions. Management reports should be released as often as is practical and as soon after the reporting period as possible. They should highlight both good and bad performance and only include things which can be controlled [Curry, A., Flett, P. and Hollingsworth, I. 2006].

Different types of management reports:

Analysis report

Analysis reports are basic reports that show mainly numerical information in a table. This information could be monthly or quarterly, showing for example, sales in a particular region. They are useful to give a quick overview of performance but a drawback is that they don't tell the background story so it's difficult to tell from the analysis report alone why one area is doing well or another area is failing. Analysis reports are relatively easy to program but they can take a long time to run so it is usually a good idea to schedule them to run overnight.

Key Target Report

Key target reports are used to show how actual performance compares with the target or prediction. They can be used to show individual progress or that of a group. Key targets should be limited and achievable [Whiteley, D. 2013]. An example of a key target report would be whether a sales target has been achieved in a particular month.

Exception Reports

An exception report would be designed to highlight any data which does not fall into a normal or defined range. Exception reports can be generated as the exception happens and send an alert to the manager by text or email so to help them find problems as or even before they occur, so that they can take corrective action. It will be short and to the point. Examples of exception reports are outstanding accounts, overdue deliveries or errors such as an increase in scrap being produced from a process. They can also highlight stock which is not selling and price irregularities [Nagpal, D. 2011].

Ad-Hoc Reports

Ad hoc analysis is a business intelligence process designed to answer a single, specific business question. The product of ad hoc analysis is typically a statistical model, analytic report, or other type of data summary. Ad-hoc reports are one-off reports, which are created as and when the user requires. They are not scheduled and can be built to the user's individual requirements. They can also be created or modified from existing reports but always fulfil an irregular need and provide information, which is not available by regular analysis [Nagpal, D. 2011]. An example might be when dealing with a complaint from a customer or information needed for a new marketing strategy. Ad-hoc reports can be expensive and time consuming but this can be mitigated by using specialised database software such as SQL.

Scheduled Reports

Scheduled reports are automatically produced at regular intervals, which can be daily, weekly or monthly, and they will be distributed to all interested parties including users, staff and investors. They will contain a large amount of information which may not all be relevant. Scheduled reports may not be used now as much as in the past because as machine readability improves, people can more easily access the direct and specific information that they require [Curry, A., Flett, P. and Hollingsworth, I. 2006].

Data enquiries and management reports that may be of use to Ray's Rentals, including diagrams and figures.

Data Enquiries

- Which bikes are available for reservation
- Which bikes are due for servicing
- Which bikes are due to be sold
- To print off an up-to-date price list
- To list prices by hour, day and other time periods to inform the customer
- Which parts are not in stock
- Which parts are no longer being used

Customers who have enquired but not made a booking

Management Reports

Management reports that would be useful to Rays Rentals include:

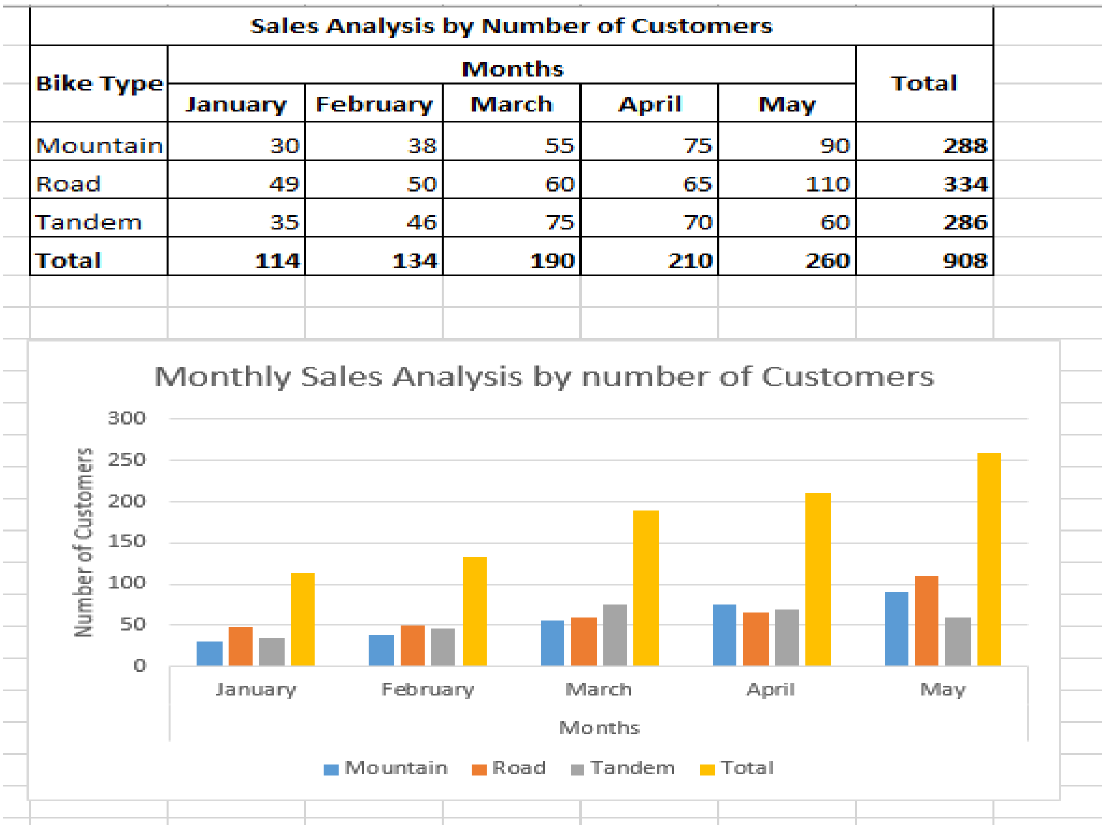
Analysis Report:

The analysis report could provide data on such as:

- Number of customers by week/month
- Analysis of which bikes are most popular
- Frequency of bike faults by manufacturer
- Seasonal trends by number of customers and type of bike

In this way, the performance of the business can be better tracked to help decide which areas could use more or less investment.

Examples of one of the possible types of analysis reports that would be available include:



Sales Analysis by Number of Customers

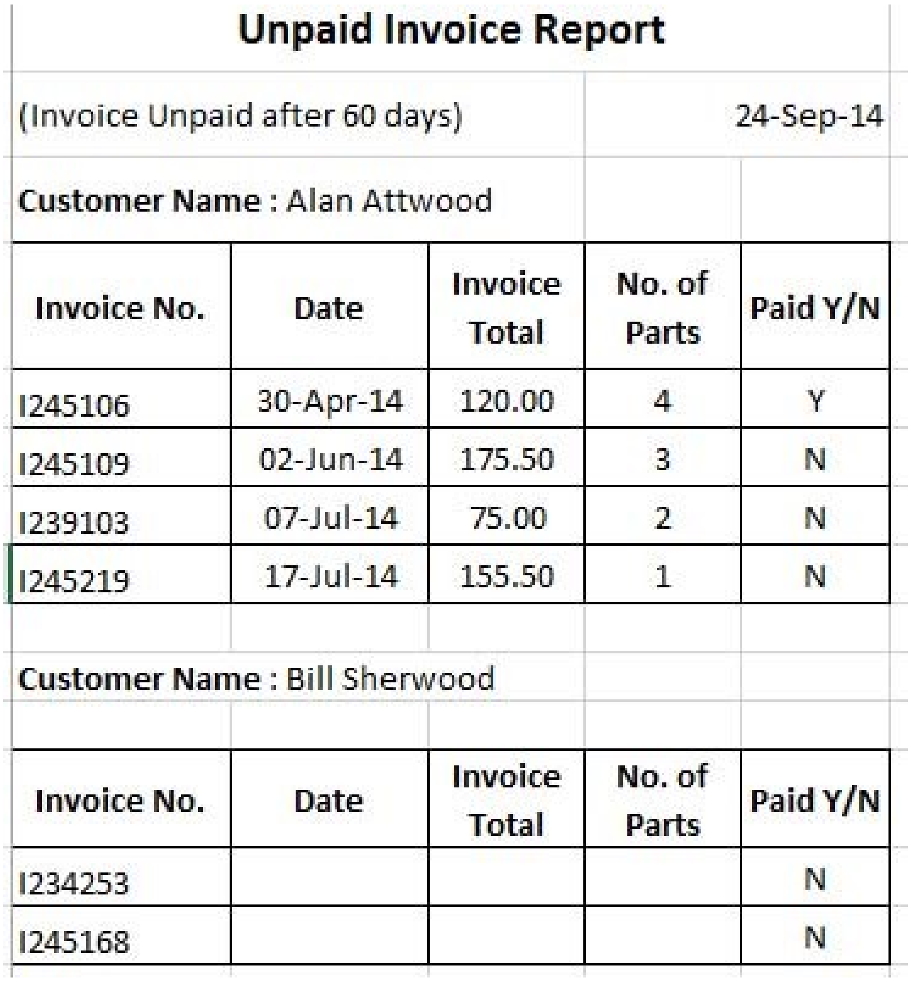
Bike Type	Months					Total
	January	February	March	April	May	
Mountain	30	38	55	75	90	**288**
Road	49	50	60	65	110	**334**
Tandem	35	46	75	70	60	**286**
Total	**114**	**134**	**190**	**210**	**260**	**908**

Exception Reports:

Exception reports could be used to:

- Show which invoices have not been paid.
- Identify which parts have been ordered but not delivered.
- Show bikes which need an excessive amount of maintenance.

An example of how unpaid invoices could be presented under the new system:



Unpaid Invoice Report

(Invoice Unpaid after 60 days) 24-Sep-14

Customer Name : Alan Attwood

Invoice No.	Date	Invoice Total	No. of Parts	Paid Y/N
I245106	30-Apr-14	120.00	4	Y
I245109	02-Jun-14	175.50	3	N
I239103	07-Jul-14	75.00	2	N
I245219	17-Jul-14	155.50	1	N

Customer Name : Bill Sherwood

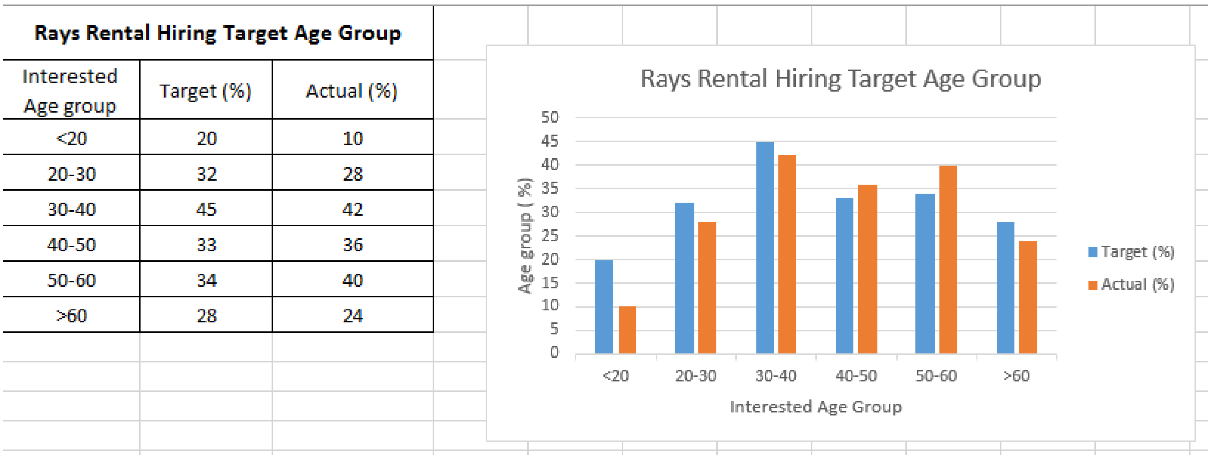
Invoice No.	Date	Invoice Total	No. of Parts	Paid Y/N
I234253				N
I245168				N

Scheduled Reports:

- To identify which bikes need to be serviced each week.
- To regularly show which customers have expressed interest but not been followed up.

Key Target Reports

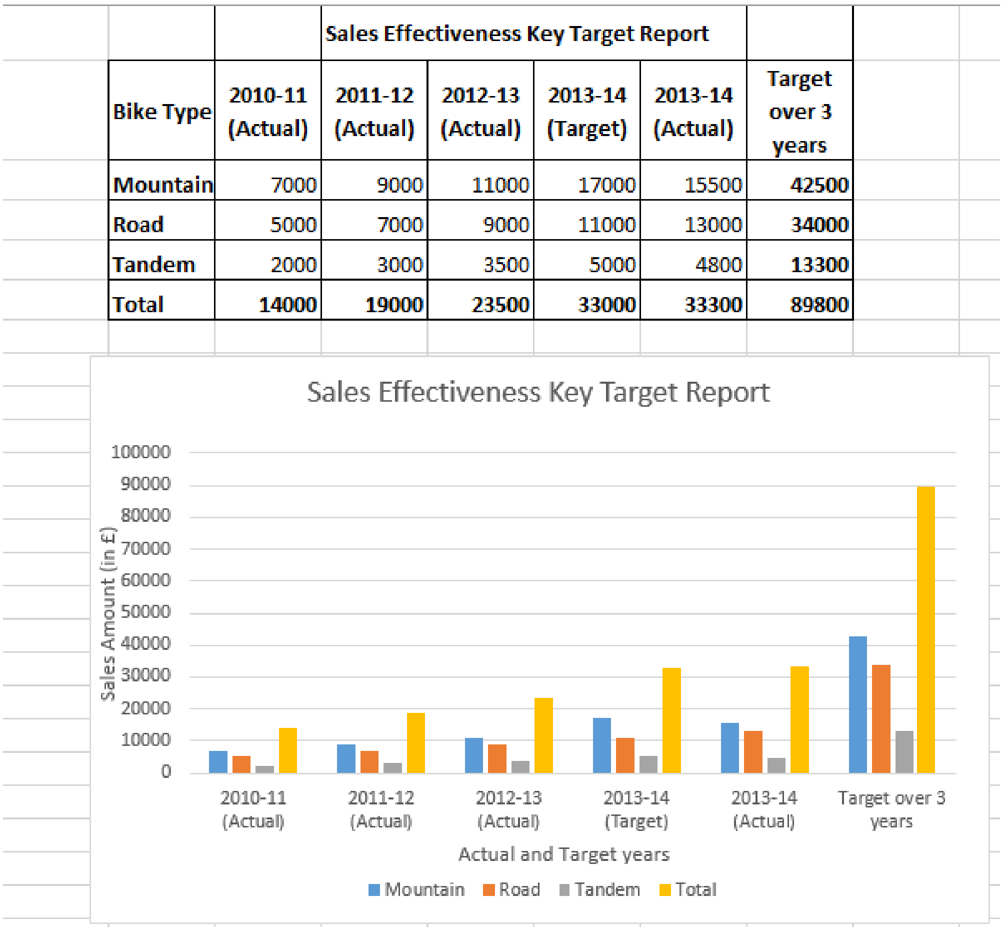
To find out the progress toward the goal in each season every year – a percent increase in our renting of various types of bikes, different age groups hiring various types of bikes. The percent improvement in one of several areas of business – either by reducing the overall material used, including return customers by giving good customer service and cut down staff or keep the bikes in maintained in regular intervals to avoid huge loss.



Rays Rental Hiring Target Age Group

Interested Age group	Target (%)	Actual (%)
<20	20	10
20-30	32	28
30-40	45	42
40-50	33	36
50-60	34	40
>60	28	24

Actual and Target Sales Report has been depicted in excel graphical format.



			Sales Effectiveness Key Target Report			
Bike Type	**2010-11 (Actual)**	**2011-12 (Actual)**	**2012-13 (Actual)**	**2013-14 (Target)**	**2013-14 (Actual)**	**Target over 3 years**
Mountain	7000	9000	11000	17000	15500	**42500**
Road	5000	7000	9000	11000	13000	**34000**
Tandem	2000	3000	3500	5000	4800	**13300**
Total	**14000**	**19000**	**23500**	**33000**	**33300**	**89800**

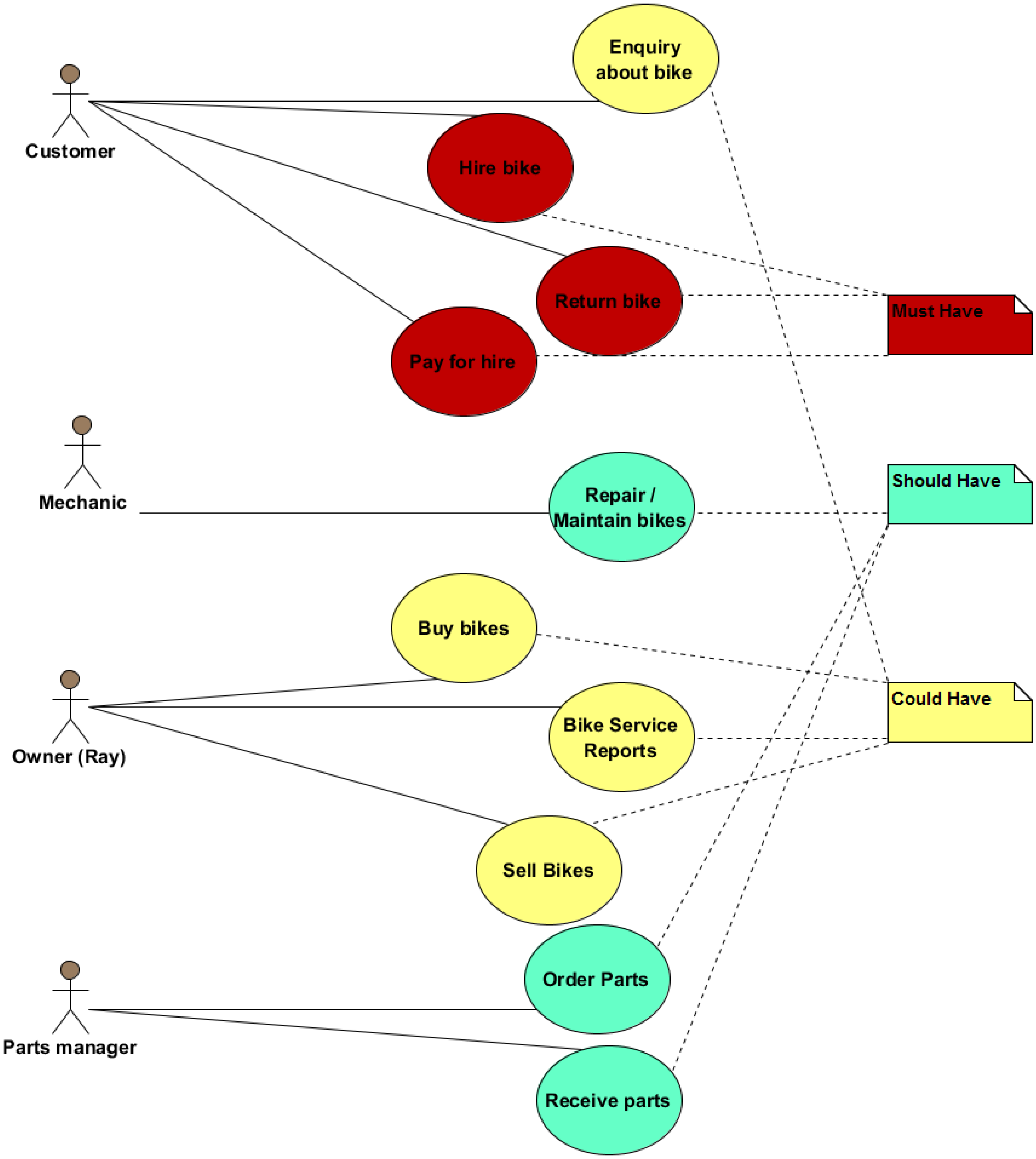
Conclusion

This report introduced the difficulties Ray’s Rentals had with recording their data on paper. In conclusion, it is recommended that a computerised database could benefit and improve Rays Rentals business, helping the business to be better organised, by reducing the risk of missing out or duplicating data. They will be able to quickly search for and find errors in their data. It will also become easier to manage both data and operations with additional help from different types of management reports.

Part 2 - Introduction

In part 2 the different use cases within the business have been analysed and a Use Case Diagram created which is prioritised using the MoSCoW (Must have, Should have, Could have, Won't have) system of prioritisation. The core use cases have been further analysed and a detailed specification produced from them. Relational data analysis has been undertaken to normalise the data from the documents provided and a bottom up ERD produced from them. Finally the top down and bottom up ERDs have been merged to create a final one which covers the whole system.

Use Case Diagram (UCD) of the new system (allowing for the activities that take place in the current system), using the 'MoSCoW' system of prioritisation



Commentary explaining the decisions made when creating the UCD and a summary of what has been learned in the process

To begin with, all the possible actors were identified from the Case Study Summary for Ray's Rentals. Then their roles within the system were identified. This was done by deciding whether they were supplying or receiving information, or initiating a use case. The different Use Cases were also organised using the MoSCoW (Must have, Should have, Could have or Won't have) system of prioritisation. This method separates the core parts of the system, which it absolutely cannot exist without, from those which make the system more efficient, and those which are simply useful. In this way, it was decided that hiring, returning and paying for hire are the core parts of the business. Maintaining and repairing the bikes are necessary for efficient running of the business and buying and selling bikes, while useful, are not fundamental aspects. Though it could be argued that without the ability to buy bikes there is no business, it was decided that the business already exists and already has everything it needs to function.

Next, the different actors and use cases were analysed to determine whether there was any duplication. It was clear that the use cases played by the rental staff were already covered by the use cases assigned to the customer and that checking deliveries should be included as part of receiving them. This reduced the number of actors by one and the number of use cases from 15 to 10.

In creating the use case diagram, we have learnt about the most important flows of data through the system, and who they interact with.

Use Case Specifications covering the core use cases, each with a supporting ERD.

Use Case Diagrams Specification designed by	
Enquiries about bike Hire bike Return bike Pay for bike	**Maryam Elgahmi**
Buy bikes Bike Service Reports Sell Bikes	**Mark Bellingham**
Repair/Maintain Bikes Order Parts Receive Parts	**Janet D'Souza**

Use Case: Hire bike

Owner: customer

Pre-Conditions

Customer chooses a bike and informs a staff

Post-Conditions

Customer leaves with bike with due date and time after it has been checked out in database from a staff.

Primary Path

Customer provides staff with their contact details, chooses type of bike and hire date

Staff records customers' information and save it in database and checks availability.

Customer hires bike

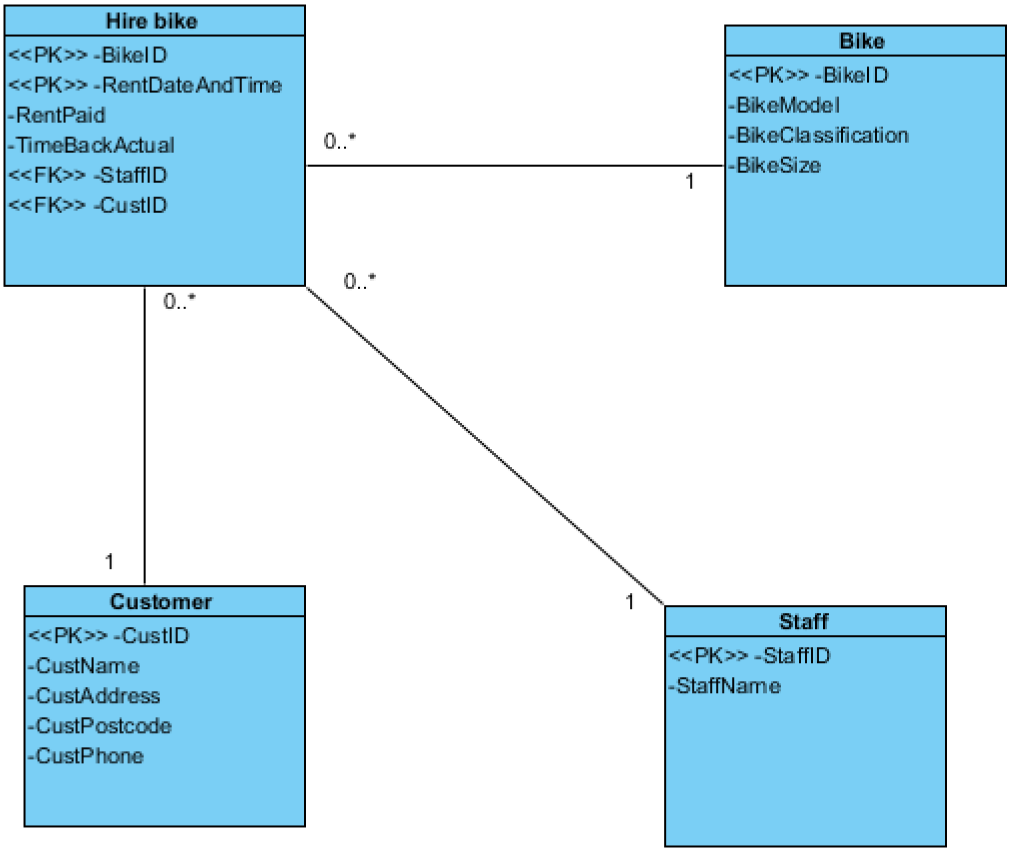
Customer makes payment

Alternate Path

Customer makes reservation in advance

Staff updates reservation for bike record.

Notes


Hire bike
<<PK>> -BikeID
<<PK>> -RentDateAndTime
-RentPaid
-TimeBackActual
<<FK>> -StaffID
<<FK>> -CustID
Bike
<<PK>> -BikeID
-BikeModel
-BikeClassification
-BikeSize
0..*
1
0..*
0..*
1
1
Customer
<<PK>> -CustID
-CustName
-CustAddress
-CustPostcode
-CustPhone
Staff
<<PK>> -StaffID
-StaffName

Use Case: Bike Service Reports

Owner: Ray

Pre-Conditions

Find out which bikes need servicing

Post-Conditions

Details of which bikes need servicing have been passed to the maintenance department

Primary Path

Check bike records

Create report where last service date is more than one month ago

List of bikes is passed to the maintenance department

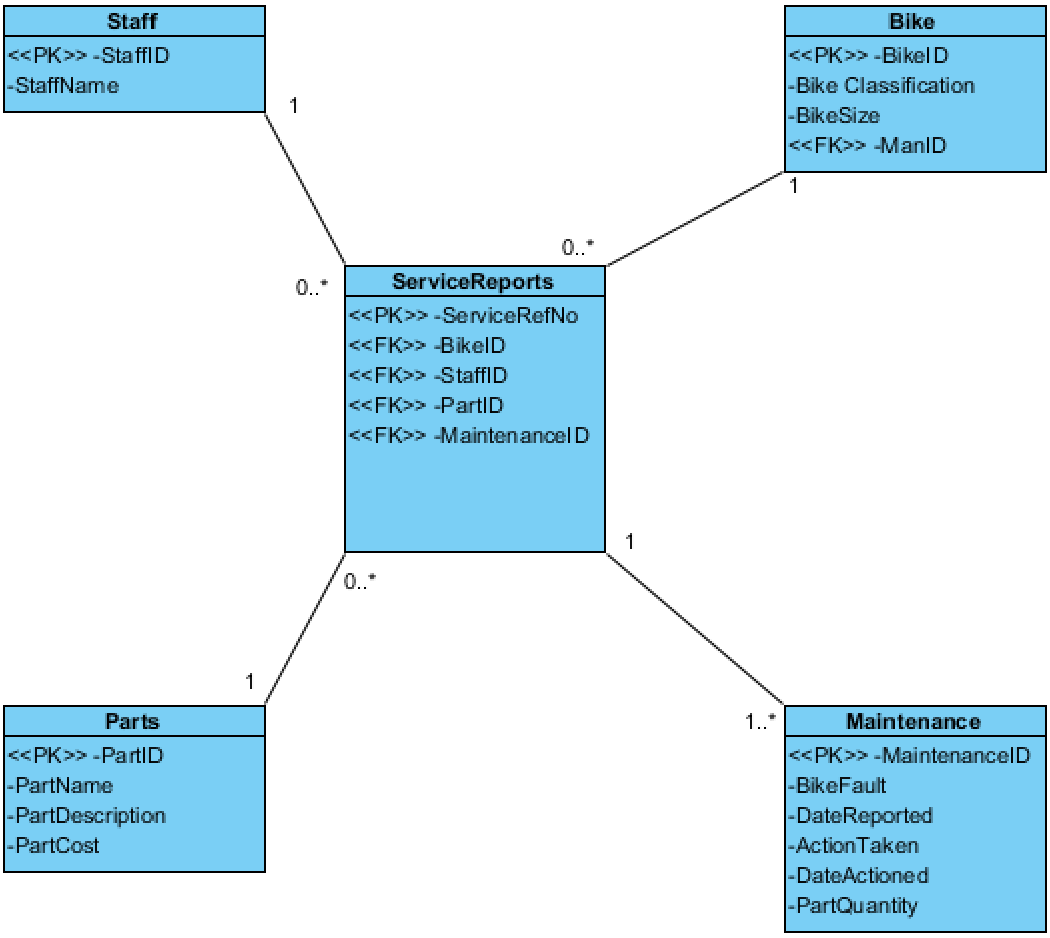
Alternate Path

Customer complains about bike fault

Reception updates a list of bikes with faults

List of bikes is passed to the maintenance department

Notes


Staff
<<PK>> -StaffID
-StaffName
Bike
<<PK>> -BikeID
-Bike Classification
-BikeSize
<<FK>> -ManID
ServiceReports
<<PK>> -ServiceRefNo
<<FK>> -BikeID
<<FK>> -StaffID
<<FK>> -PartID
<<FK>> -MaintenanceID
Parts
<<PK>> -PartID
-PartName
-PartDescription
-PartCost
Maintenance
<<PK>> -MaintenanceID
-BikeFault
-DateReported
-ActionTaken
-DateActioned
-PartQuantity
1
0..*
0..*
1
0..*
1
1
1..*

Use Case: Order Parts

Owner: Parts Manager

Pre-Conditions

Checking the frequency of necessary parts not in stock

Checking the parts that are over-ordered and left lying around the workshop for long that they either go rusty or become obsolete.

Post-Conditions

Parts have been ordered from suppliers

Primary Path

Parts Manager has to keep a track of parts which are frequently used

Creates a report not in stock or low on stock.

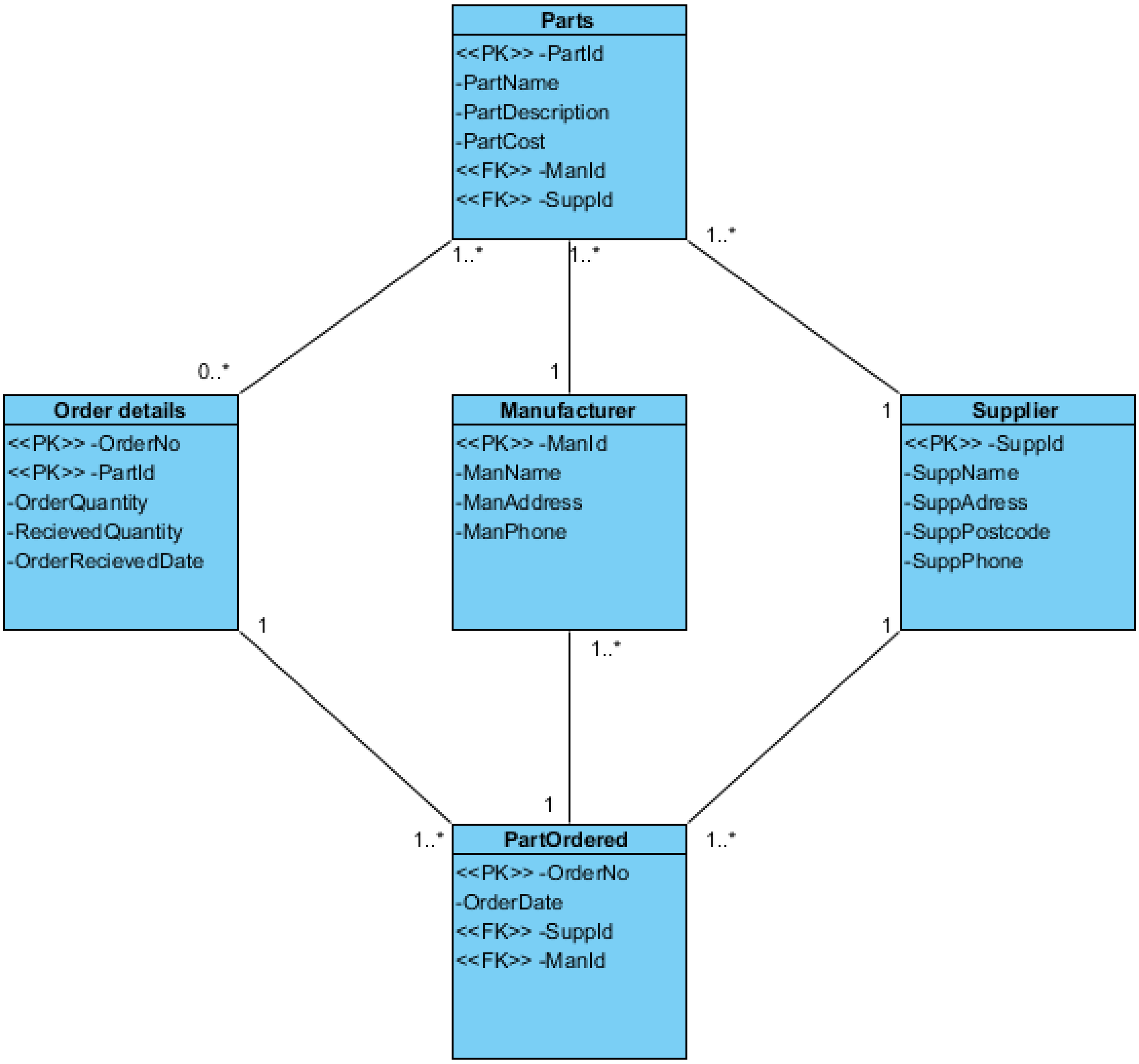
Parts are ordered

Part Order file updated with number of parts ordered from which supplier

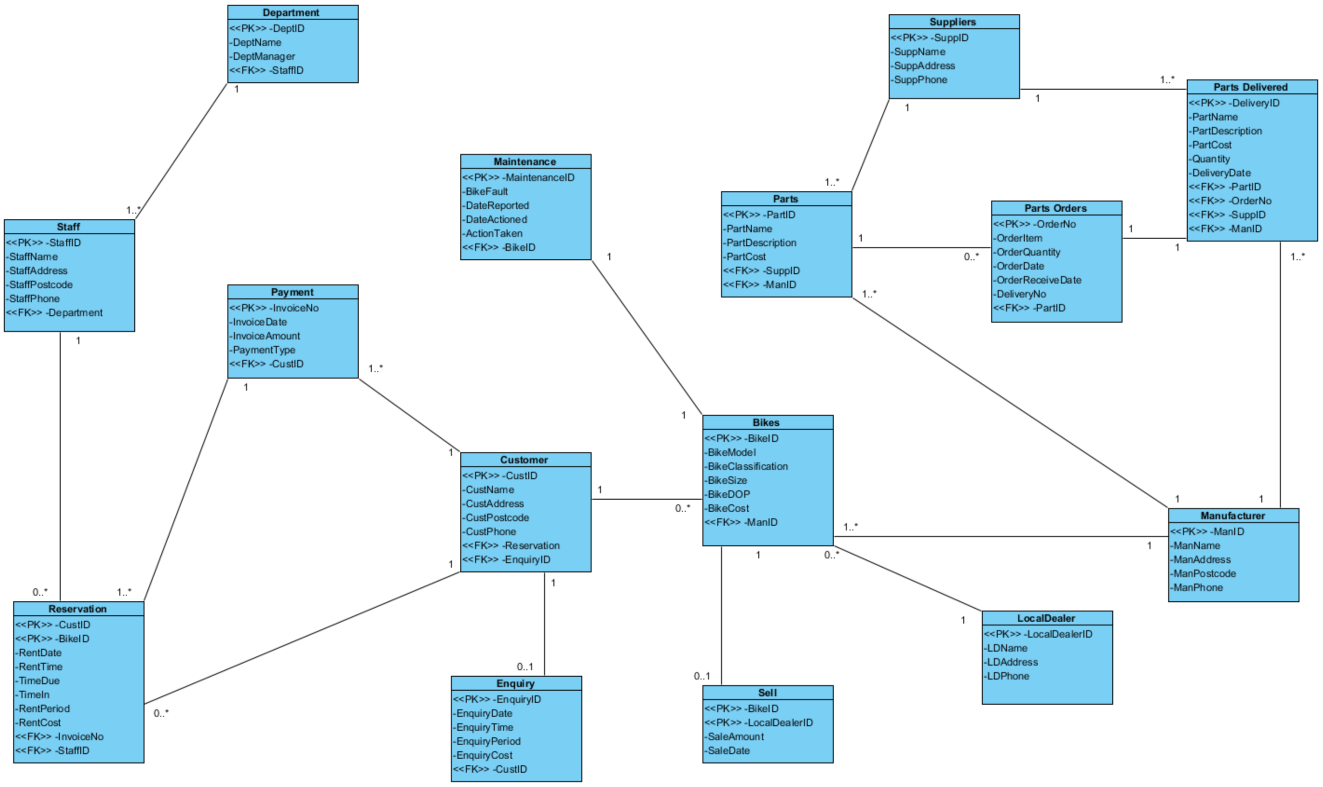
Alternate Path

Parts can be obtained from several other trusted suppliers.

Notes


Parts
<<PK>> -PartId
-PartName
-PartDescription
-PartCost
<<FK>> -ManId
<<FK>> -SuppId
Order details
<<PK>> -OrderNo
<<PK>> -PartId
-OrderQuantity
-RecievedQuantity
-OrderRecievedDate
Manufacturer
<<PK>> -ManId
-ManName
-ManAddress
-ManPhone
Supplier
<<PK>> -SuppId
-SuppName
-SuppAdress
-SuppPostcode
-SuppPhone
PartOrdered
<<PK>> -OrderNo
-OrderDate
<<FK>> -SuppId
<<FK>> -ManId
1..*
1..*
1..*
0..*
1
1
1
1..*
1
1
1..*
1..*

Completed top-down ERD of the system

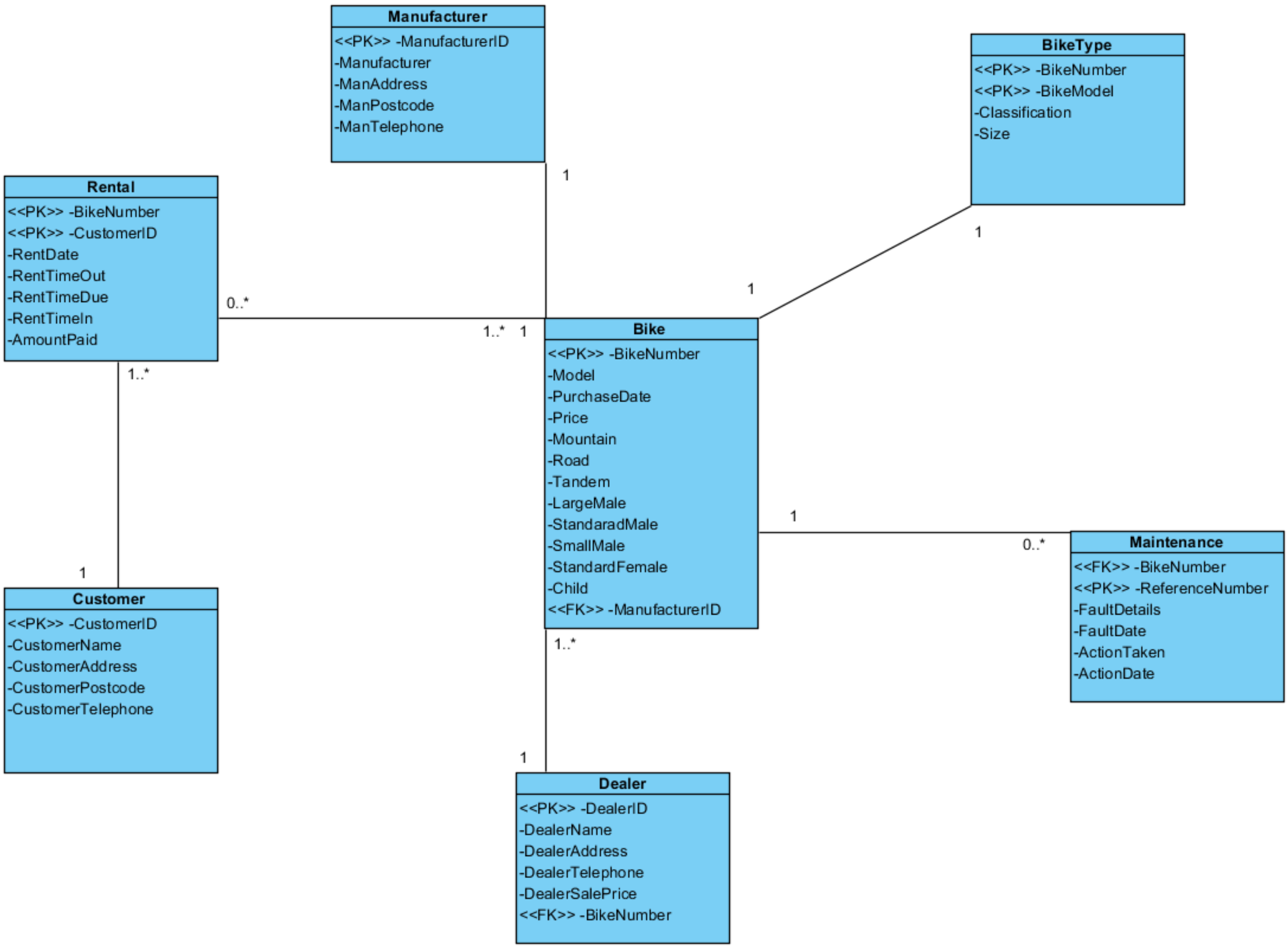


Completed RDA of each the two documents provided in the case study and a bottom up ERD of the merged RDAs

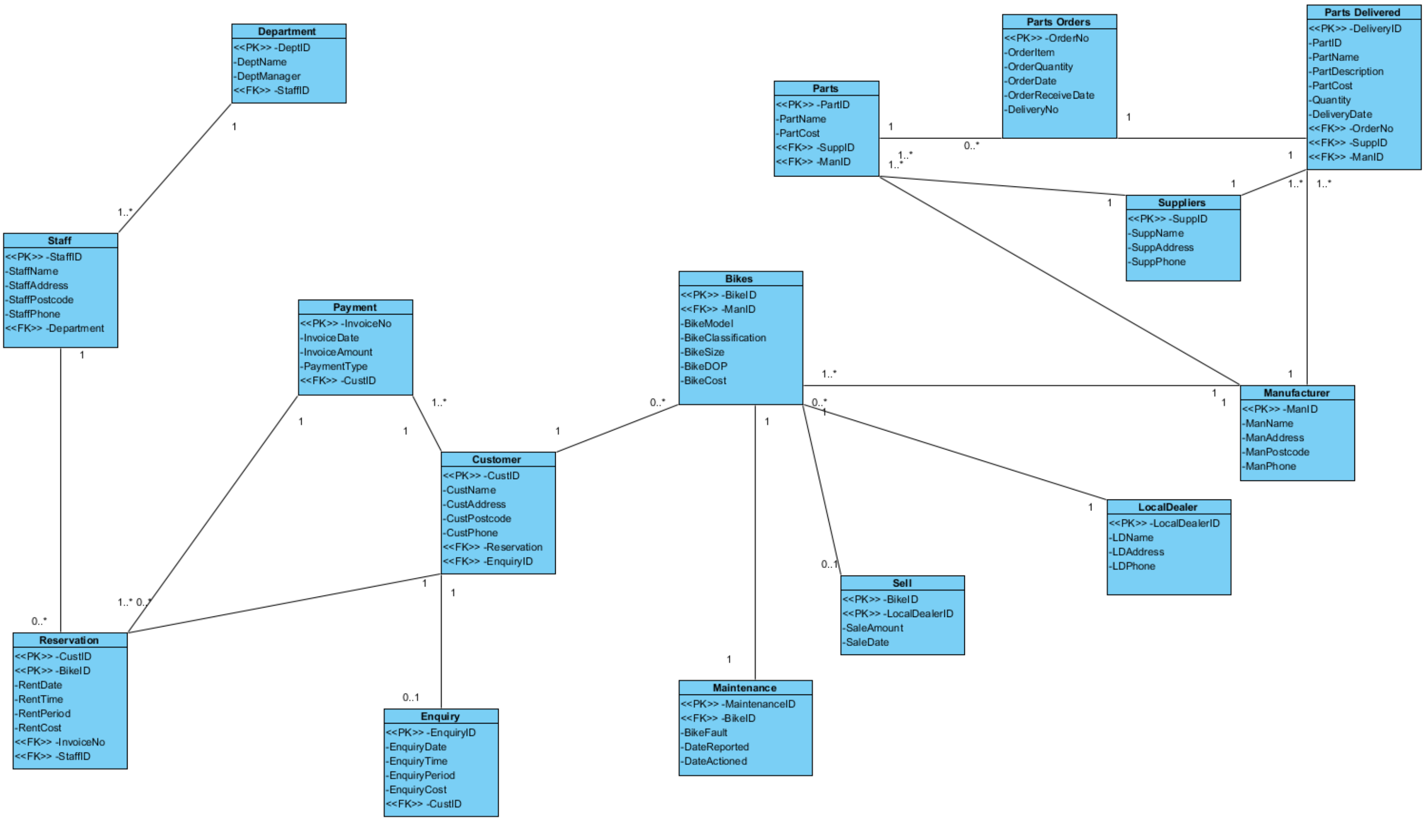
Appendix A - Bike Record								
Unnormalised		**1NF**		**2NF**		**3NF**		**Entities**
Bike Number		**Bike Number**		**Bike Number**		**Bike Number**		
Model		Model		Model		Model		
Purchase Date		Purchase Date		Purchase Date		Purchase Date		
Price		Price		Price		Price		
Mountain		Mountain		Mountain		Mountain		
Road		Road		Road		Road		
Tandem		Tandem		Tandem		Tandem		Bike
Large Male		Large Male		Large Male		Large Male		
Standard Male		Standard Male		Standard Male		Standard Male		
Small Male		Small Male		Small Male		Small Male		
Standard Female		Standard Female		Standard Female		Standard Female		
Child		Child		Child		Child		
Manufacturer		Manufacturer		Manufacturer		ManufacturerID (FK)		
Man Address		Man Address		Man Address				
Man Postcode		Man Postcode		Man Postcode		**ManufacturerID**		
Man Telephone		Man Telephone		Man Telephone		Manufacturer		
Sale Date		Sale Date		Sale Date		Man Address		Manufacturer
Dealer Name		Dealer Name		Dealer Name		Man Postcode		
Dealer Address		Dealer Address		Dealer Address		Man Telephone		
Dealer Telephone		Dealer Telephone		Dealer Telephone				
Dealer Sale Price		Dealer Sale Price		Dealer Sale Price		**DealerID**		
Reference No						Dealer Name		
Fault Details		**Bike Number**		**Bike Number**		Dealer Address		
Fault Date		**Reference No**		**Reference No**		Dealer Telephone		Dealer
Action Taken		Fault Details		Fault Details		Dealer Sale Price		
Action Date		Fault Date		Fault Date		Sale Date		
		Action Taken		Action Taken		BikeNumber (FK)		
		Action Date		Action Date				
						Reference No		
						Fault Details		
						Fault Date		
						Action Taken		Maintenance
						Action Date		
						Bike Number(FK)		

Appendix B - Rental Record

Unnormalised		**1NF**		**2NF**		**3NF**		**Entities**
Bike Number		**Bike Number**		**Bike Number**		**Bike Number**		Bike
Bike Model		Bike Model		Bike Model		Bike Model		
Classification		Classification		Classification		Classification		
Size		Size		Size		Size		
Rent Date								
Rent Time Out		**Bike Number**		**Bike Number**		**Bike Number**		Rental
Rent Time Due		**Rent Date**		**Rent Date**		**Rent Date**		
rent Time In		**Rent Time Out**		**Rent Time Out**		**Rent Time Out**		
Customer Name		Rent Time Due		Rent Time Due		Rent Time Due		
Customer Address		rent Time In		rent Time In		rent Time In		
Customer Postcode		Customer Name		Amount Paid		Amount Paid		
Customer Phone		Customer Address						
Amount Paid		Customer Postcode		**Customer ID**		**Customer ID**		Customer
		Customer Phone		Customer Name		Customer Name		
		Amount Paid		Customer Address		Customer Address		
				Customer Postcode		Customer Postcode		
				Customer Telephone		Customer Telephone		


Manufacturer
<<PK>> -ManufacturerID
-Manufacturer
-ManAddress
-ManPostcode
-ManTelephone
BikeType
<<PK>> -BikeNumber
<<PK>> -BikeModel
-Classification
-Size
Rental
<<PK>> -BikeNumber
<<PK>> -CustomerID
-RentDate
-RentTimeOut
-RentTimeDue
-RentTimeIn
-AmountPaid
1
1
1
0..*
1..*
1
Bike
<<PK>> -BikeNumber
-Model
-PurchaseDate
-Price
-Mountain
-Road
-Tandem
-LargeMale
-StandaradMale
-SmallMale
-StandardFemale
-Child
<<FK>> -ManufacturerID
1..*
1
1
0..*
Customer
<<PK>> -CustomerID
-CustomerName
-CustomerAddress
-CustomerPostcode
-CustomerTelephone
Maintenance
<<FK>> -BikeNumber
<<PK>> -ReferenceNumber
-FaultDetails
-FaultDate
-ActionTaken
-ActionDate
1..*
1
Dealer
<<PK>> -DealerID
-DealerName
-DealerAddress
-DealerTelephone
-DealerSalePrice
<<FK>> -BikeNumber

Finalised group ERD (including both top down and bottom up perspectives) covering the complete system



Commentary explaining the decisions made when creating the finalised ERD and a summary of what has been learned in the process.

After looking at Ray Rentals case study, a bottom-up entity relationship diagram was drawn using visual paradigm, for the required system database. After creating the entity relationship diagram, the group decided to delete some attributes, because some were similar to each other and others were not considered important in this system database. Bike record entity, has been removed because the attributes: bike size, classification, model, bike number (bike ID), purchase date (bike DOP) have been added in the bikes entity, and other attributes like Maintenance that were left as an entity and had the attributes 'bike fault' and 'date action taken' included. Maintenance now has a one to one relationship with bikes entity and a bike ID is a foreign key in Maintenance.

Ray the manager has been changed from an entity to an attribute called depmanager in the Department entity, which has a one-to-many relationship with staff entity, because some other staff are also linked with the department. Model and faults has also been changed to an attribute. Model has been included in bikes entity as bike model, which will contain all the details. Faults has been added to maintenance entity, which will include the details of the bike fault. Tyres, brake blocks, cables and lubrication were left as attributes in Parts as Parts name. For rental record and hiring, the group has decided to leave it, because it had the same attributes as reservation, which had a composite primary key of customer ID and bike ID. Pete, Sheila, Megan, Alf and Bert have been included as attributes in Staff entity, which is linked with department entity. Last entity that has been changed from entity to an attribute was receipt, added to payment, which had relationship with customer and reservation.

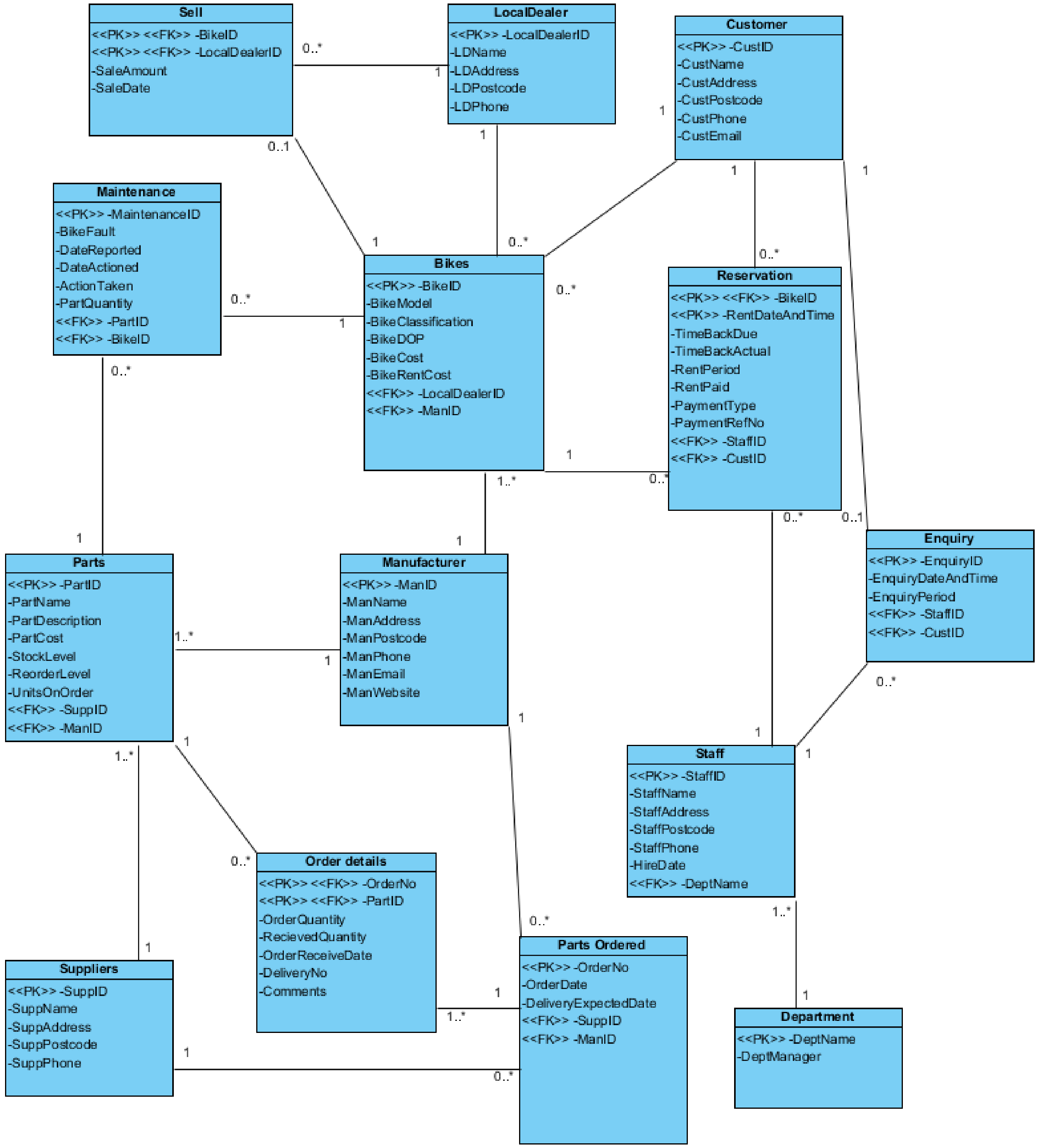
After merging the bottom up and top down diagrams, a third was created to remove repetition and include anything else which was missed. The attributes that have been included were: in the parts order entity, a foreign key has been added to be linked with parts ID. Reservation entity had time due and time in attributes added from rental entity which had similar attributes. For maintenance entity, action taken attribute has been included.

After this process, the group has understood the relationship between the entities and how they link together.

Part 3 – Introduction

For this section, the ERD has been amended to add and delete some attributes. The data dictionary shows the data types and values for each of the attributes. After creating and inserting records into the tables, screenshots have been included to show the result of each designed query.

Amended ERD


Sell
<<PK>> <<FK>> -BikeID
<<PK>> <<FK>> -LocalDealerID
-SaleAmount
-SaleDate
LocalDealer
<<PK>> -LocalDealerID
-LDName
-LDAddress
-LDPostcode
-LDPhone
Customer
<<PK>> -CustID
-CustName
-CustAddress
-CustPostcode
-CustPhone
-CustEmail
Maintenance
<<PK>> -MaintenanceID
-BikeFault
-DateReported
-DateActioned
-ActionTaken
-PartQuantity
<<FK>> -PartID
<<FK>> -BikeID
Bikes
<<PK>> -BikeID
-BikeModel
-BikeClassification
-BikeDOP
-BikeCost
-BikeRentCost
<<FK>> -LocalDealerID
<<FK>> -ManID
Reservation
<<PK>> <<FK>> -BikeID
<<PK>> -RentDateAndTime
-TimeBackDue
-TimeBackActual
-RentPeriod
-RentPaid
-PaymentType
-PaymentRefNo
<<FK>> -StaffID
<<FK>> -CustID
Enquiry
<<PK>> -EnquiryID
-EnquiryDateAndTime
-EnquiryPeriod
<<FK>> -StaffID
<<FK>> -CustID
Parts
<<PK>> -PartID
-PartName
-PartDescription
-PartCost
-StockLevel
-ReorderLevel
-UnitsOnOrder
<<FK>> -SuppID
<<FK>> -ManID
Manufacturer
<<PK>> -ManID
-ManName
-ManAddress
-ManPostcode
-ManPhone
-ManEmail
-ManWebsite
Staff
<<PK>> -StaffID
-StaffName
-StaffAddress
-StaffPostcode
-StaffPhone
-HireDate
<<FK>> -DeptName
Order details
<<PK>> <<FK>> -OrderNo
<<PK>> <<FK>> -PartID
-OrderQuantity
-RecievedQuantity
-OrderReceiveDate
-DeliveryNo
-Comments
Parts Ordered
<<PK>> -OrderNo
-OrderDate
-DeliveryExpectedDate
<<FK>> -SuppID
<<FK>> -ManID
Suppliers
<<PK>> -SuppID
-SuppName
-SuppAddress
-SuppPostcode
-SuppPhone
Department
<<PK>> -DeptName
-DeptManager
0..*
1
0..1
1
1
0..*
1
1
0..*
1
0..*
1
0..*
0..*
1
1..*
0..*
0..*
0..1
0..*
1
1
1
1..*
1
1
0..*
1
1
1..*
1
0..*
1
1..*
0..*
1
1..*
1
1
0..*

Data dictionary

DEPARTMENT							
Attribute Name	**Data Type**	**Length**	**Key**	**Constraint**	**Constraint Name**	**FK Table**	**FK Column**
DeptName	Varchar2	30	PK		DeptNmPK		
DeptManager	Varchar2	30					

STAFF							
Attribute Name	**Data Type**	**Length**	**Key**	**Constraint**	**Constraint Name**	**FK Table**	**FK Column**
StaffId	Number	5	PK		StfIdPK		
StaffName	Varchar2	30		NOT NULL	StfNmNN		
StaffAddress	Varchar2	60		NOT NULL	StfAdrsNN		
StaffPostCode	Varchar2	15		NOT NULL	StfPCNN		
StaffPhone	Varchar2	20		NOT NULL	StfPhNN		
HireDate	Date			NOT NULL	StfHDtNN		
DeptName	Varchar2	30	FK	NOT NULL		Department	DeptId

ENQUIRY							
Attribute Name	Data Type	Length	Key	Constraint	Constraint Name	FK Table	FK Column
EnquiryId	NUMBER	10	PK		EnqIDPK		
EnquiryDateNTime	DATE			NOT NULL	EnqDtNN		
EnquiryPeriod	NUMBER	2		NOT NULL	EnqPrdNN		
StaffId	NUMBER	5	FK			STAFF	StaffId
CustId	NUMBER	10	FK			CUSTOMER	CustId

RESERVATION							
Attribute Name	**Data Type**	**Length**	**Key**	**Constraint**	**Constraint Name**	**FK Table**	**FK Column**
BikeId	NUMBER	6	PK,FK			RR_BIKES	BikeId
RentDateAndTime	DATE		PK		ResrvRDPK		
TimeBackDue	DATE						
TimeBackActual	DATE						
RentPeriod	Number	2		NOT NULL	ResrvRntPrdNN		
RentPaid	CHAR	1		Check RentPaid ('Y','N')	ResrvRntPdCk		
PaymentType	Varchar2	15		Check paymentType in('Cash','card' , 'cheque')	InvPyTypChk		
PaymentRefNo	NUMBER	10					
StaffId	NUMBER	5	FK			RR_STAFF	StaffId
CustId	NUMBER	10	FK			RR_CUSTOMER	CustId

CUSTOMER							
Attribute Name	Data Type	Length	Key	Constraint	Constraint Name	FK Table	FK Column
CustId	NUMBER	10	PK		CustIdPK		
CustName	Varchar2	25		NOT NULL	CustNmNN		
CustAddress	Varchar2	60		NOT NULL	CustAdrNN		
CustPostCode	Varchar2	15		NOT NULL	CustPCNN		
CustPhone	Varchar2	14		NOT NULL	CustPhNN		
CustEmail	Varchar2	40					

MAINTENANCE							
Attribute Name	Data Type	Length	Key	Constraint	Constraint Name	FK Table	FK Column
MaintenanceId	NUMBER	10	PK		MainIdPK		
BikeFault	Varchar2	30					
DateReported	Date						
DateActioned	Date						
ActionTaken	Varchar2	30					
PartQuantity	NUMBER	2					
PartId	NUMBER	4	FK			RR_Parts	PartId
BikeId	NUMBER	6	FK			RR_Bikes	BikeId

PARTSORDERED							
Attribute Name	Data Type	Length	Key	Constraint	Constraint Name	FK Table	FK Column
OrderNo	NUMBER	10	PK		PartsOrderNoPK		
OrderDate	Date			Default SYSDATE			
DeliveryExpectedDate	Date						
SuppId	NUMBER	3	FK			RR_Suppliers	SuppId
ManId	NUMBER	3	FK			RR_Manufacturer	ManId

BIKES							
Attribute Name	Data Type	Length	Key	Constraint	Constraint Name	FK Table	FK Column
BikeId	NUMBER	6	PK		BikeIdPK		
BikeModel	Varchar2	20		NOT NULL	BikeModelNNChk		
BikeClassification	Varchar2	10		Check(BikeClassification in(‘mountain’,’road’,’tandem’)	BikeClassChk		
BikeDOP	Date			NOT NULL	BikeDOPNN		
BikeCost	NUMBER	6,2		CHECK(BikeCost>0)	BikeCstChk		
BikeRentCost	NUMBER	4,2		NOT NULL	BikeRtCstNN		
LocalDealerId	Varchar2	10	FK			RR_LocalDealer	LocalDealerId
ManId	Varchar2	3	FK			RR_Manufacturer	ManId

MANUFACTURER							
Attribute Name	Data Type	Length	Key	Constraint	Constraint Name	FK Table	FK Column
ManId	Varchar2	3	PK		ManIdPK		
ManName	Varchar2	25		NOT NULL	ManNmNN		
ManAddress	Varchar2	80		NOT NULL	ManAddNN		
ManPostCode	Varchar2	15		NOT NULL	ManPCNN		
ManPhone	Varchar2	20		NOT NULL	ManPhNN		
ManEmail	Varchar2	40					
ManWebsite	Varchar2	30					

LOCALDEALER							
Attribute Name	Data Type	Length	Key	Constraint	Constraint Name	FK Table	FK Column
LocalDealerId	Varchar2	10	PK		LocalDealerIdPK		
LDName	Varchar2	25		NOT NULL	LDNmNN		
LDAddress	Varchar2	80		NOT NULL	LDAdrNN		
LDPostCode	Varchar2	15		NOT NULL	LDPCNN		
LDPhone	Varchar2	14		NOT NULL	LDPhone		

SELL							
Attribute Name	Data Type	Length	Key	Constraint	Constraint Name	FK Table	FK Column
BikeId	NUMBER	6	PK FK			RR_BIKES	BikeId
LocalDealerId	NUMBER	10	PK FK			RR_LocalDealer	LocalDealerId
SaleAmount	NUMBER	6,2		CHECK(SaleAmount>=0)	SaleAmtChk		
SaleDate	Date						

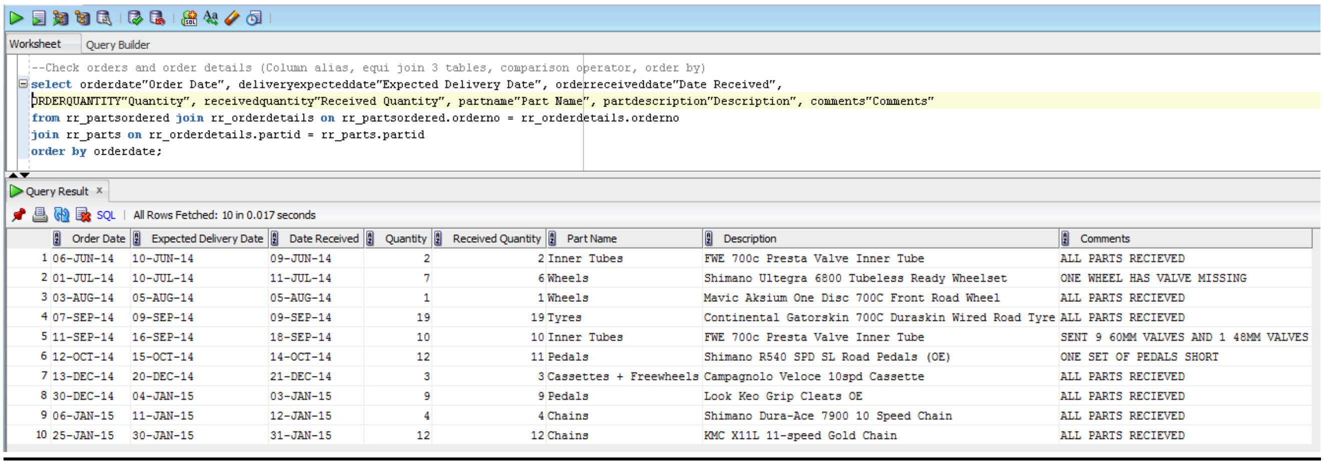
SUPPLIERS							
Attribute Name	Data Type	Length	Key	Constraint	Constraint Name	FK Table	FK Column
SuppId	NUMBER	3	PK		SuppIdPK		
SuppName	Varchar2	25		NOT NULL	SuppNmNN		
SuppAddress	Varchar2	80		NOT NULL	SuppAdrNN		
SuppPostCode	Varchar2	15		NOT NULL	SuppPCNN		
SuppPhone	Varchar2	14		NOT NULL & UNIQUE	SuppPhNNUQ		

PARTS							
Attribute Name	**Data Type**	**Length**	**Key**	**Constraint**	**Constraint Name**	**FK Table**	**FK Column**
PartId	NUMBER	4	PK		PartIdPK		
PartName	Varchar2	25		NOT NULL	PartNmNN		
PartDescription	Varchar2	60					
PartCost	NUMBER	6,2		CHECK(PartCost>0)	PartCostChk		
StockLevel	NUMBER	3					
ReOrderLevel	NUMBER	2					
UnitsOnOrder	NUMBER	3					
SuppId	NUMBER	3	FK			RR_Suppliers	SuppId
ManId	NUMBER	3	FK			RR_Manufacturer	ManId

ORDERDETAILS							
Attribute Name	**Data Type**	**Length**	**Key**	**Constraint**	**Constraint Name**	**FK Table**	**FK Column**
OrderNo	NUMBER	10	PKFK			RR_PARTSORDERED	OrderNo
PartId	NUMBER	3	PKFK			RR_PARTS	PartId
OrderQuantity	NUMBER	3		CHECK(OrderQuantity > 0)	OrderDetQtyChk		
ReceivedQuantity	NUMBER	3					
OrderReceivedDate	DATE						
DeliveryNo	Varchar2	15					
Comments	Varchar2	50					

Screen Shots of Queries created by group members:

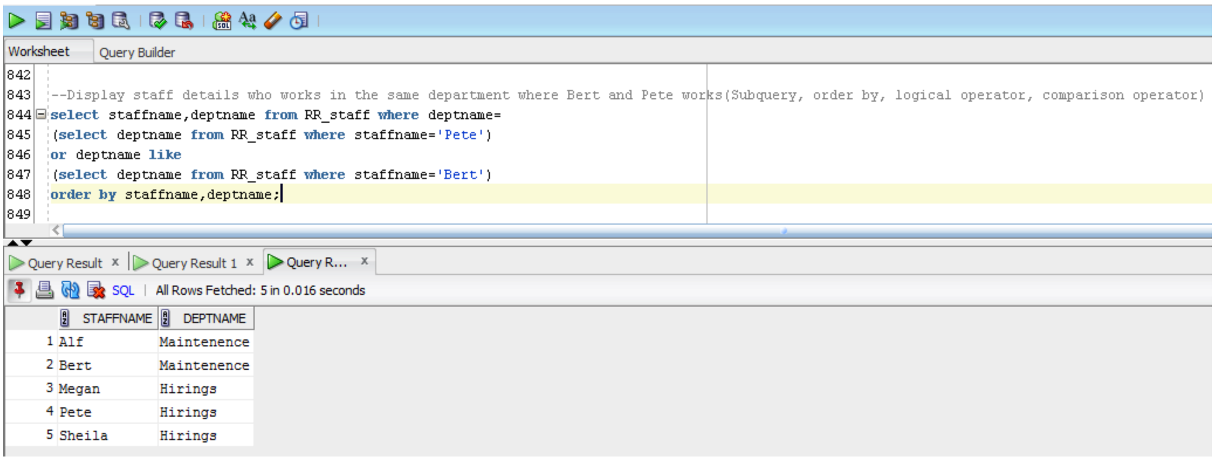
Mark Bellingham



```
--Check orders and order details (Column alias, equi join 3 tables, comparison operator, order by)
select orderdate"Order Date", deliveryexpecteddate"Expected Delivery Date", orderreceiveddate"Date Received",
ORDERQUANTITY"Quantity", receivedquantity"Received Quantity", partname"Part Name", partdescription"Description", comments"Comments"
from rr_partsordered join rr_orderdetails on rr_partsordered.orderno = rr_orderdetails.orderno
join rr_parts on rr_orderdetails.partid = rr_parts.partid
order by orderdate;
```

All Rows Fetched: 10 in 0.017 seconds

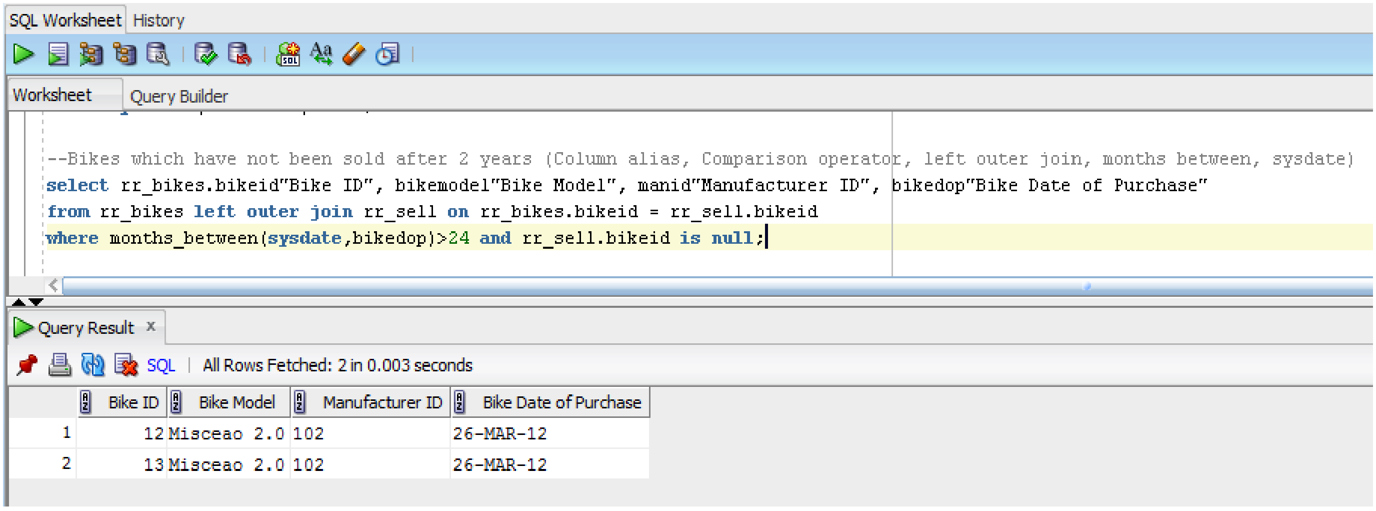
	Order Date	Expected Delivery Date	Date Received	Quantity	Received Quantity	Part Name	Description	Comments
1	06-JUN-14	10-JUN-14	09-JUN-14	2	2	Inner Tubes	FWE 700c Presta Valve Inner Tube	ALL PARTS RECIEVED
2	01-JUL-14	10-JUL-14	11-JUL-14	7	6	Wheels	Shimano Ultegra 6800 Tubeless Ready Wheelset	ONE WHEEL HAS VALVE MISSING
3	03-AUG-14	05-AUG-14	05-AUG-14	1	1	Wheels	Mavic Aksium One Disc 700C Front Road Wheel	ALL PARTS RECIEVED
4	07-SEP-14	09-SEP-14	09-SEP-14	19	19	Tyres	Continental Gatorskin 700C Duraskin Wired Road Tyre	ALL PARTS RECIEVED
5	11-SEP-14	16-SEP-14	18-SEP-14	10	10	Inner Tubes	FWE 700c Presta Valve Inner Tube	SENT 9 60MM VALVES AND 1 48MM VALVES
6	12-OCT-14	15-OCT-14	14-OCT-14	12	11	Pedals	Shimano R540 SPD SL Road Pedals (OE)	ONE SET OF PEDALS SHORT
7	13-DEC-14	20-DEC-14	21-DEC-14	3	3	Cassettes + Freewheels	Campagnolo Veloce 10spd Cassette	ALL PARTS RECIEVED
8	30-DEC-14	04-JAN-15	03-JAN-15	9	9	Pedals	Look Keo Grip Cleats OE	ALL PARTS RECIEVED
9	06-JAN-15	11-JAN-15	12-JAN-15	4	4	Chains	Shimano Dura-Ace 7900 10 Speed Chain	ALL PARTS RECIEVED
10	25-JAN-15	30-JAN-15	31-JAN-15	12	12	Chains	KMC X11L 11-speed Gold Chain	ALL PARTS RECIEVED



```
--Display staff details who works in the same department where Bert and Pete works(Subquery, order by, logical operator, comparison operator)
select staffname,deptname from RR_staff where deptname=
(select deptname from RR_staff where staffname='Pete')
or deptname like
(select deptname from RR_staff where staffname='Bert')
order by staffname,deptname;
```

All Rows Fetched: 5 in 0.016 seconds

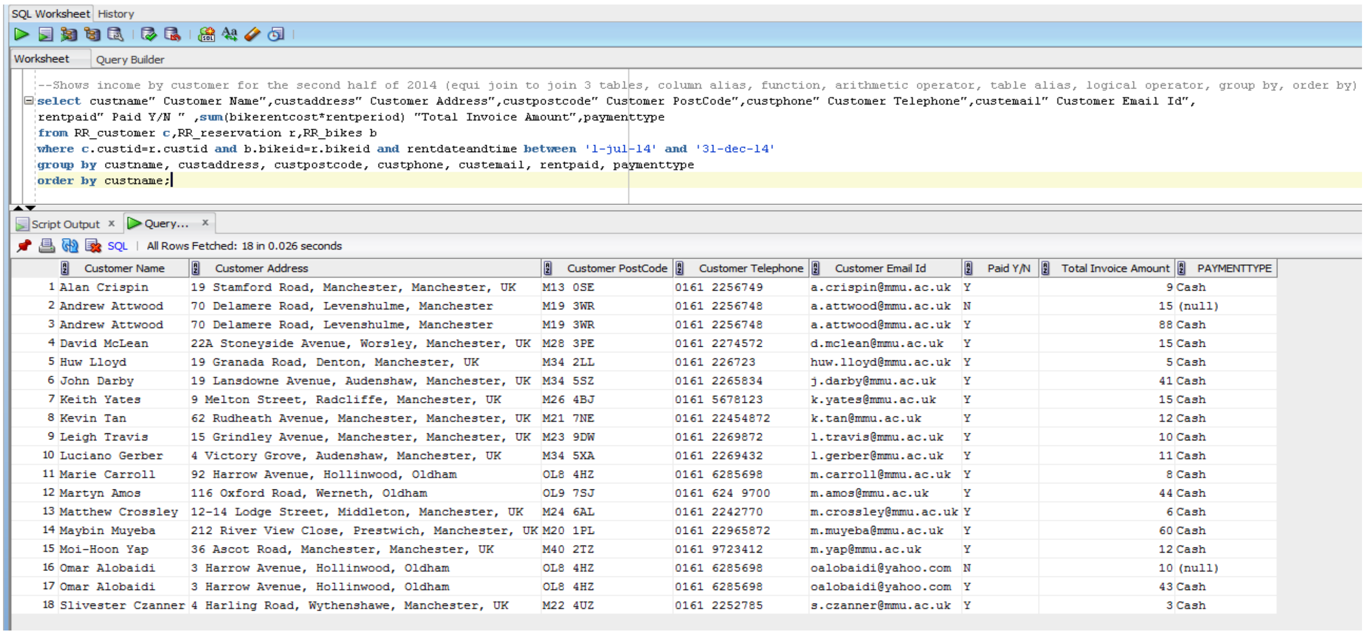
	STAFFNAME	DEPTNAME
1	Alf	Maintenence
2	Bert	Maintenence
3	Megan	Hirings
4	Pete	Hirings
5	Sheila	Hirings



```
--Bikes which have not been sold after 2 years (Column alias, Comparison operator, left outer join, months between, sysdate)
select rr_bikes.bikeid"Bike ID", bikemodel"Bike Model", manid"Manufacturer ID", bikedop"Bike Date of Purchase"
from rr_bikes left outer join rr_sell on rr_bikes.bikeid = rr_sell.bikeid
where months_between(sysdate,bikedop)>24 and rr_sell.bikeid is null;
```

All Rows Fetched: 2 in 0.003 seconds

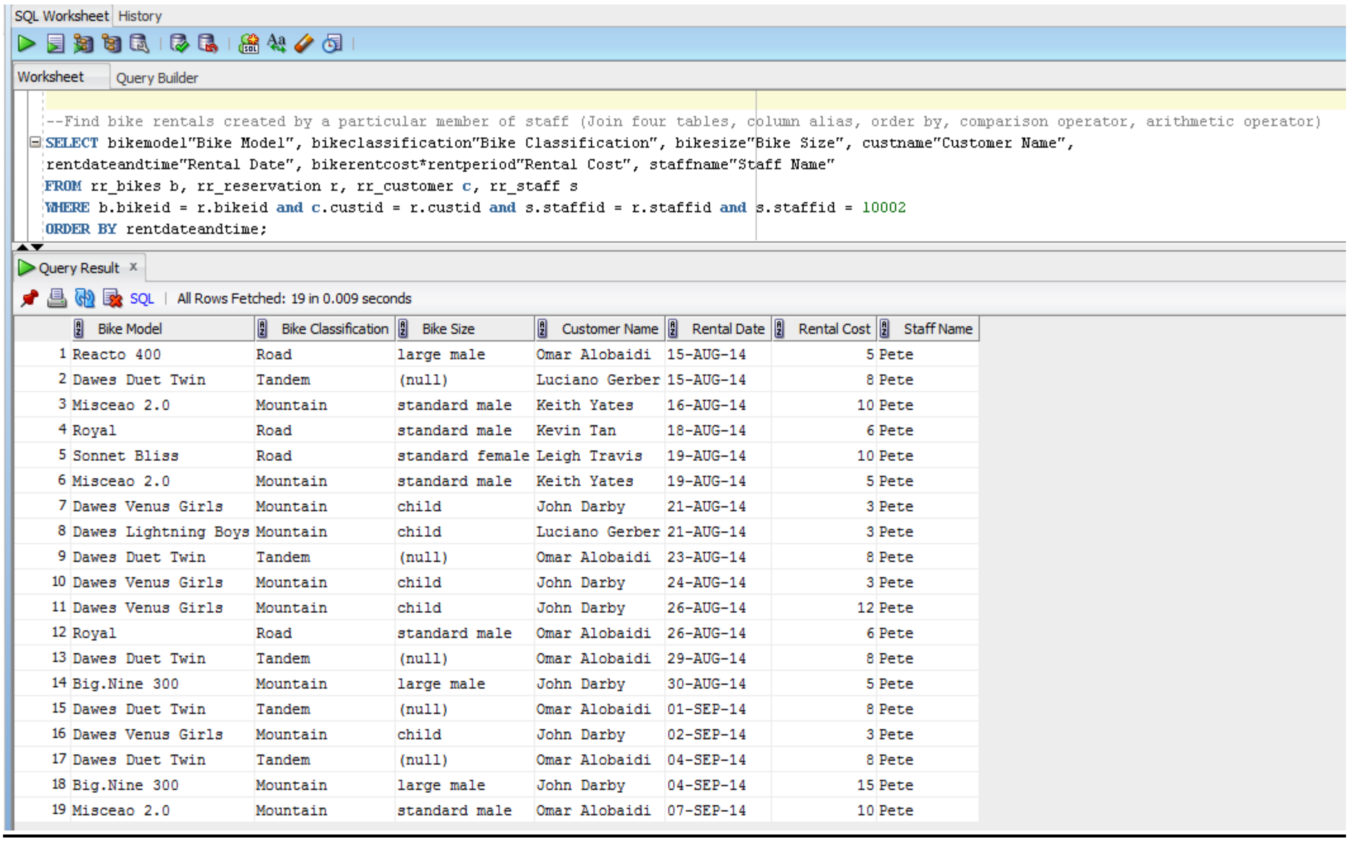
	Bike ID	Bike Model	Manufacturer ID	Bike Date of Purchase
1	12	Misceao 2.0	102	26-MAR-12
2	13	Misceao 2.0	102	26-MAR-12



```sql
--Shows income by customer for the second half of 2014 (equi join to join 3 tables, column alias, function, arithmetic operator, table alias, logical operator, group by, order by)
select custname" Customer Name",custaddress" Customer Address",custpostcode" Customer PostCode",custphone" Customer Telephone",custemail" Customer Email Id",
rentpaid" Paid Y/N " ,sum(bikerentcost*rentperiod) "Total Invoice Amount",paymenttype
from RR_customer c,RR_reservation r,RR_bikes b
where c.custid=r.custid and b.bikeid=r.bikeid and rentdateandtime between '1-jul-14' and '31-dec-14'
group by custname, custaddress, custpostcode, custphone, custemail, rentpaid, paymenttype
order by custname;
```

All Rows Fetched: 18 in 0.026 seconds

	Customer Name	Customer Address	Customer PostCode	Customer Telephone	Customer Email Id	Paid Y/N	Total Invoice Amount	PAYMENTTYPE
1	Alan Crispin	19 Stamford Road, Manchester, Manchester, UK	M13 0SE	0161 2256749	a.crispin@mmu.ac.uk	Y	9	Cash
2	Andrew Attwood	70 Delamere Road, Levenshulme, Manchester	M19 3WR	0161 2256748	a.attwood@mmu.ac.uk	N	15	(null)
3	Andrew Attwood	70 Delamere Road, Levenshulme, Manchester	M19 3WR	0161 2256748	a.attwood@mmu.ac.uk	Y	88	Cash
4	David McLean	22A Stoneyside Avenue, Worsley, Manchester, UK	M28 3PE	0161 2274572	d.mclean@mmu.ac.uk	Y	15	Cash
5	Huw Lloyd	19 Granada Road, Denton, Manchester, UK	M34 2LL	0161 226723	huw.lloyd@mmu.ac.uk	Y	5	Cash
6	John Darby	19 Lansdowne Avenue, Audenshaw, Manchester, UK	M34 5SZ	0161 2265834	j.darby@mmu.ac.uk	Y	41	Cash
7	Keith Yates	9 Melton Street, Radcliffe, Manchester, UK	M26 4BJ	0161 5678123	k.yates@mmu.ac.uk	Y	15	Cash
8	Kevin Tan	62 Rudheath Avenue, Manchester, Manchester, UK	M21 7NE	0161 22454872	k.tan@mmu.ac.uk	Y	12	Cash
9	Leigh Travis	15 Grindley Avenue, Manchester, Manchester, UK	M23 9DW	0161 2269872	l.travis@mmu.ac.uk	Y	10	Cash
10	Luciano Gerber	4 Victory Grove, Audenshaw, Manchester, UK	M34 5XA	0161 2269432	l.gerber@mmu.ac.uk	Y	11	Cash
11	Marie Carroll	92 Harrow Avenue, Hollinwood, Oldham	OL8 4HZ	0161 6285698	m.carroll@mmu.ac.uk	Y	8	Cash
12	Martyn Amos	116 Oxford Road, Werneth, Oldham	OL9 7SJ	0161 624 9700	m.amos@mmu.ac.uk	Y	44	Cash
13	Matthew Crossley	12-14 Lodge Street, Middleton, Manchester, UK	M24 6AL	0161 2242770	m.crossley@mmu.ac.uk	Y	6	Cash
14	Maybin Muyeba	212 River View Close, Prestwich, Manchester, UK	M20 1PL	0161 22965872	m.muyeba@mmu.ac.uk	Y	60	Cash
15	Moi-Hoon Yap	36 Ascot Road, Manchester, Manchester, UK	M40 2TZ	0161 9723412	m.yap@mmu.ac.uk	Y	12	Cash
16	Omar Alobaidi	3 Harrow Avenue, Hollinwood, Oldham	OL8 4HZ	0161 6285698	oalobaidi@yahoo.com	N	10	(null)
17	Omar Alobaidi	3 Harrow Avenue, Hollinwood, Oldham	OL8 4HZ	0161 6285698	oalobaidi@yahoo.com	Y	43	Cash
18	Slivester Czanner	4 Harling Road, Wythenshawe, Manchester, UK	M22 4UZ	0161 2252785	s.czanner@mmu.ac.uk	Y	3	Cash

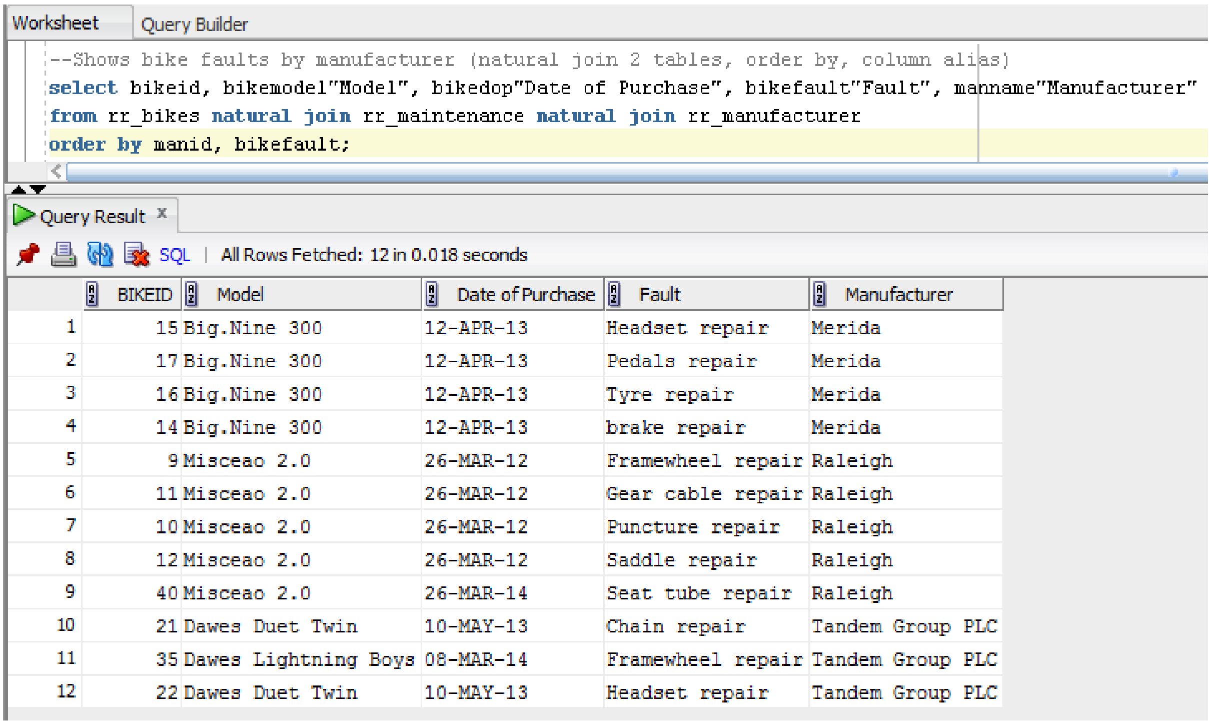


```sql
--Find bike rentals created by a particular member of staff (Join four tables, column alias, order by, comparison operator, arithmetic operator)
SELECT bikemodel"Bike Model", bikeclassification"Bike Classification", bikesize"Bike Size", custname"Customer Name",
rentdateandtime"Rental Date", bikerentcost*rentperiod"Rental Cost", staffname"Staff Name"
FROM rr_bikes b, rr_reservation r, rr_customer c, rr_staff s
WHERE b.bikeid = r.bikeid and c.custid = r.custid and s.staffid = r.staffid and s.staffid = 10002
ORDER BY rentdateandtime;
```

All Rows Fetched: 19 in 0.009 seconds

	Bike Model	Bike Classification	Bike Size	Customer Name	Rental Date	Rental Cost	Staff Name
1	Reacto 400	Road	large male	Omar Alobaidi	15-AUG-14	5	Pete
2	Dawes Duet Twin	Tandem	(null)	Luciano Gerber	15-AUG-14	8	Pete
3	Misceao 2.0	Mountain	standard male	Keith Yates	16-AUG-14	10	Pete
4	Royal	Road	standard male	Kevin Tan	18-AUG-14	6	Pete
5	Sonnet Bliss	Road	standard female	Leigh Travis	19-AUG-14	10	Pete
6	Misceao 2.0	Mountain	standard male	Keith Yates	19-AUG-14	5	Pete
7	Dawes Venus Girls	Mountain	child	John Darby	21-AUG-14	3	Pete
8	Dawes Lightning Boys	Mountain	child	Luciano Gerber	21-AUG-14	3	Pete
9	Dawes Duet Twin	Tandem	(null)	Omar Alobaidi	23-AUG-14	8	Pete
10	Dawes Venus Girls	Mountain	child	John Darby	24-AUG-14	3	Pete
11	Dawes Venus Girls	Mountain	child	John Darby	26-AUG-14	12	Pete
12	Royal	Road	standard male	Omar Alobaidi	26-AUG-14	6	Pete
13	Dawes Duet Twin	Tandem	(null)	Omar Alobaidi	29-AUG-14	8	Pete
14	Big.Nine 300	Mountain	large male	John Darby	30-AUG-14	5	Pete
15	Dawes Duet Twin	Tandem	(null)	Omar Alobaidi	01-SEP-14	8	Pete
16	Dawes Venus Girls	Mountain	child	John Darby	02-SEP-14	3	Pete
17	Dawes Duet Twin	Tandem	(null)	Omar Alobaidi	04-SEP-14	8	Pete
18	Big.Nine 300	Mountain	large male	John Darby	04-SEP-14	15	Pete
19	Misceao 2.0	Mountain	standard male	Omar Alobaidi	07-SEP-14	10	Pete

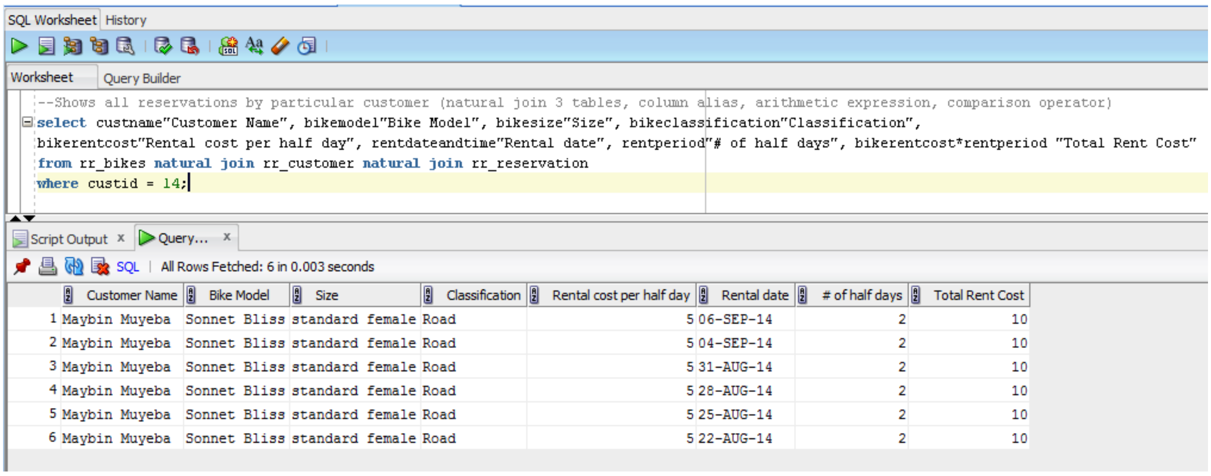
Maryam El-gahmi



```
--Shows bike faults by manufacturer (natural join 2 tables, order by, column alias)
select bikeid, bikemodel"Model", bikedop"Date of Purchase", bikefault"Fault", manname"Manufacturer"
from rr_bikes natural join rr_maintenance natural join rr_manufacturer
order by manid, bikefault;
```

All Rows Fetched: 12 in 0.018 seconds

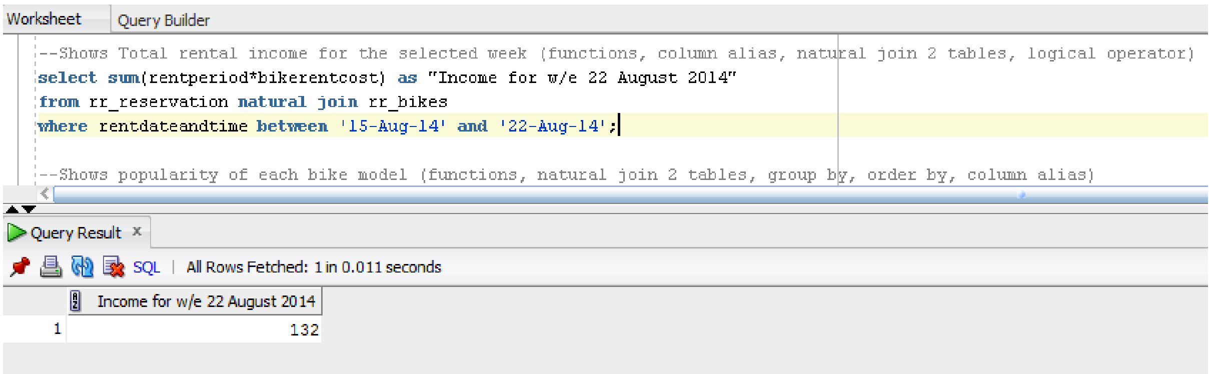
	BIKEID	Model	Date of Purchase	Fault	Manufacturer
1	15	Big.Nine 300	12-APR-13	Headset repair	Merida
2	17	Big.Nine 300	12-APR-13	Pedals repair	Merida
3	16	Big.Nine 300	12-APR-13	Tyre repair	Merida
4	14	Big.Nine 300	12-APR-13	brake repair	Merida
5	9	Misceao 2.0	26-MAR-12	Framewheel repair	Raleigh
6	11	Misceao 2.0	26-MAR-12	Gear cable repair	Raleigh
7	10	Misceao 2.0	26-MAR-12	Puncture repair	Raleigh
8	12	Misceao 2.0	26-MAR-12	Saddle repair	Raleigh
9	40	Misceao 2.0	26-MAR-14	Seat tube repair	Raleigh
10	21	Dawes Duet Twin	10-MAY-13	Chain repair	Tandem Group PLC
11	35	Dawes Lightning Boys	08-MAR-14	Framewheel repair	Tandem Group PLC
12	22	Dawes Duet Twin	10-MAY-13	Headset repair	Tandem Group PLC



```
--Shows all reservations by particular customer (natural join 3 tables, column alias, arithmetic expression, comparison operator)
select custname"Customer Name", bikemodel"Bike Model", bikesize"Size", bikeclassification"Classification",
bikerentcost"Rental cost per half day", rentdateandtime"Rental date", rentperiod"# of half days", bikerentcost*rentperiod "Total Rent Cost"
from rr_bikes natural join rr_customer natural join rr_reservation
where custid = 14;
```

All Rows Fetched: 6 in 0.003 seconds

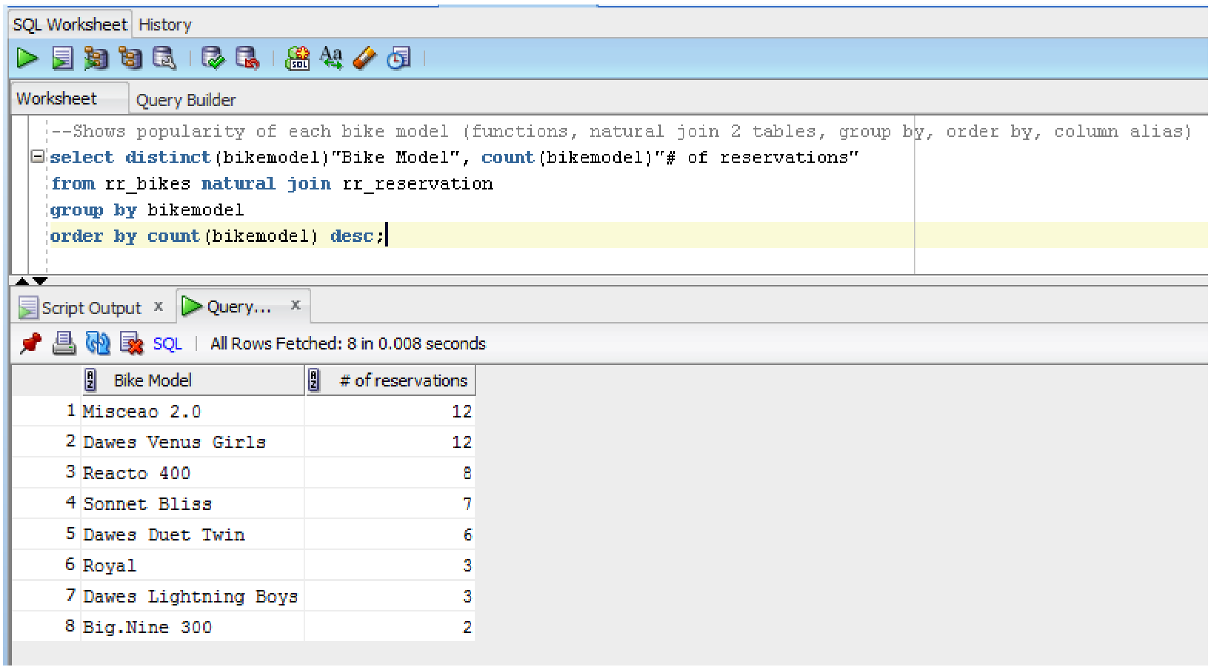
	Customer Name	Bike Model	Size	Classification	Rental cost per half day	Rental date	# of half days	Total Rent Cost
1	Maybin Muyeba	Sonnet Bliss	standard female	Road	5	06-SEP-14	2	10
2	Maybin Muyeba	Sonnet Bliss	standard female	Road	5	04-SEP-14	2	10
3	Maybin Muyeba	Sonnet Bliss	standard female	Road	5	31-AUG-14	2	10
4	Maybin Muyeba	Sonnet Bliss	standard female	Road	5	28-AUG-14	2	10
5	Maybin Muyeba	Sonnet Bliss	standard female	Road	5	25-AUG-14	2	10
6	Maybin Muyeba	Sonnet Bliss	standard female	Road	5	22-AUG-14	2	10



```
--Shows Total rental income for the selected week (functions, column alias, natural join 2 tables, logical operator)
select sum(rentperiod*bikerentcost) as "Income for w/e 22 August 2014"
from rr_reservation natural join rr_bikes
where rentdateandtime between '15-Aug-14' and '22-Aug-14';

--Shows popularity of each bike model (functions, natural join 2 tables, group by, order by, column alias)
```

All Rows Fetched: 1 in 0.011 seconds

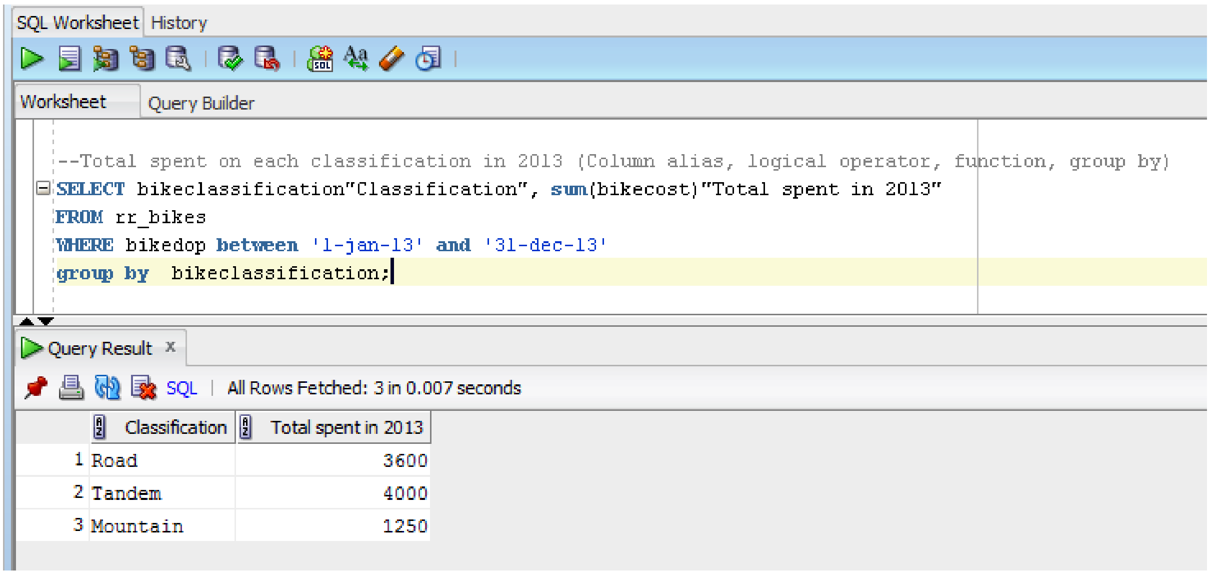
	Income for w/e 22 August 2014
1	132



```
--Shows popularity of each bike model (functions, natural join 2 tables, group by, order by, column alias)
select distinct(bikemodel)"Bike Model", count(bikemodel)"# of reservations"
from rr_bikes natural join rr_reservation
group by bikemodel
order by count(bikemodel) desc;
```

All Rows Fetched: 8 in 0.008 seconds

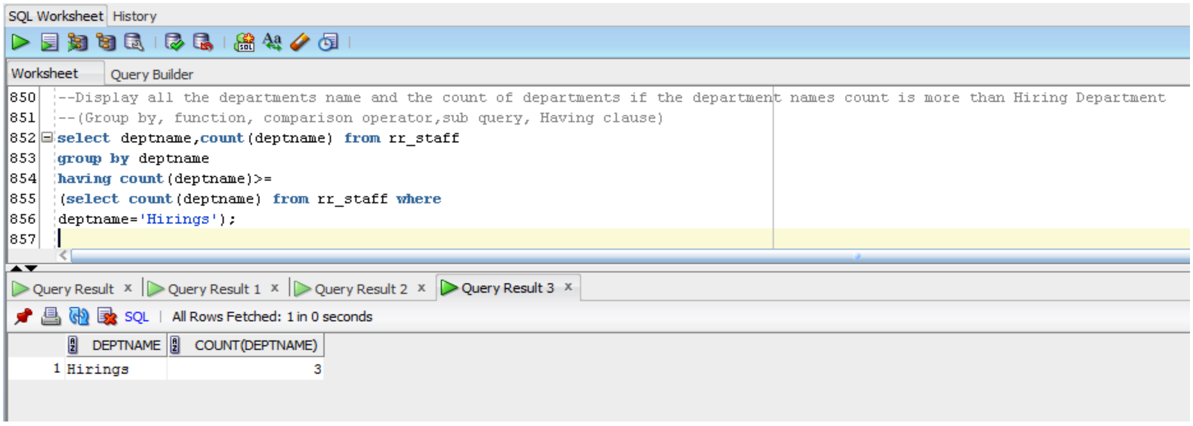
	Bike Model	# of reservations
1	Misceao 2.0	12
2	Dawes Venus Girls	12
3	Reacto 400	8
4	Sonnet Bliss	7
5	Dawes Duet Twin	6
6	Royal	3
7	Dawes Lightning Boys	3
8	Big.Nine 300	2



```
--Total spent on each classification in 2013 (Column alias, logical operator, function, group by)
SELECT bikeclassification"Classification", sum(bikecost)"Total spent in 2013"
FROM rr_bikes
WHERE bikedop between '1-jan-13' and '31-dec-13'
group by  bikeclassification;
```

All Rows Fetched: 3 in 0.007 seconds

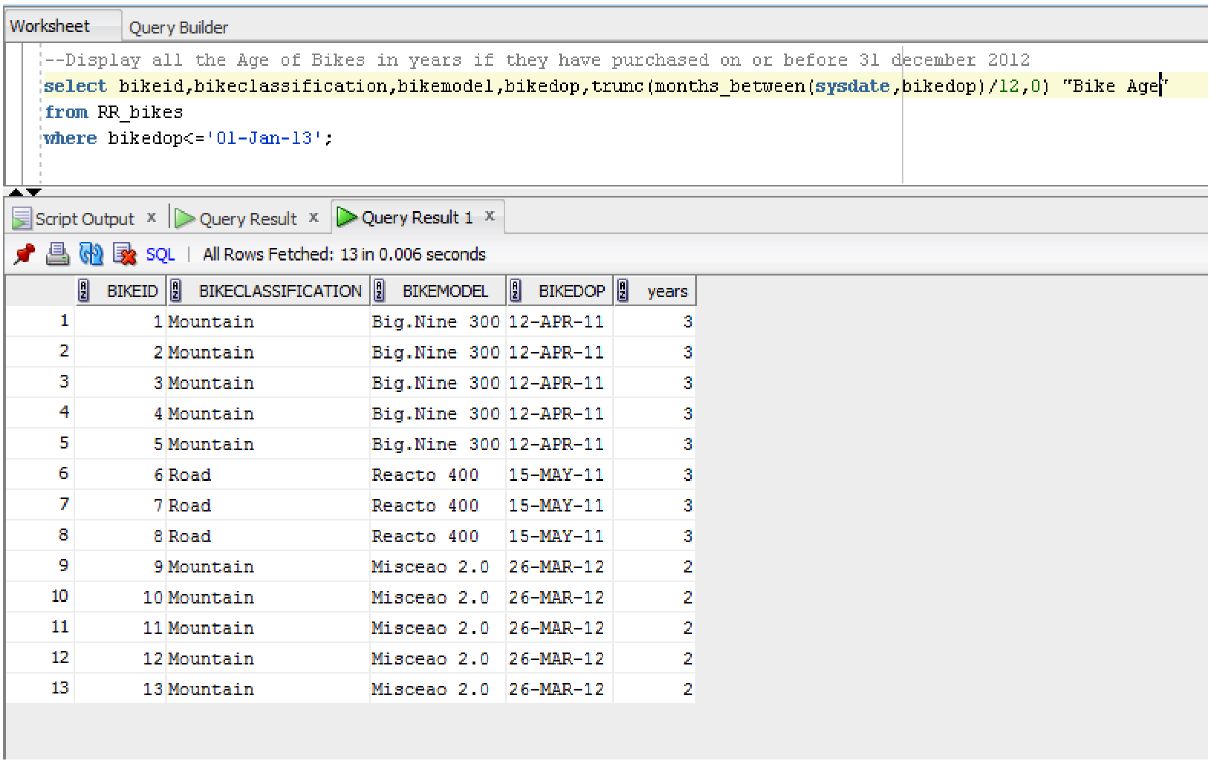
	Classification	Total spent in 2013
1	Road	3600
2	Tandem	4000
3	Mountain	1250



```
--Display all the departments name and the count of departments if the department names count is more than Hiring Department
--(Group by, function, comparison operator,sub query, Having clause)
select deptname,count(deptname) from rr_staff
group by deptname
having count(deptname)>=
(select count(deptname) from rr_staff where
deptname='Hirings');
```

All Rows Fetched: 1 in 0 seconds

	DEPTNAME	COUNT(DEPTNAME)
1	Hirings	3

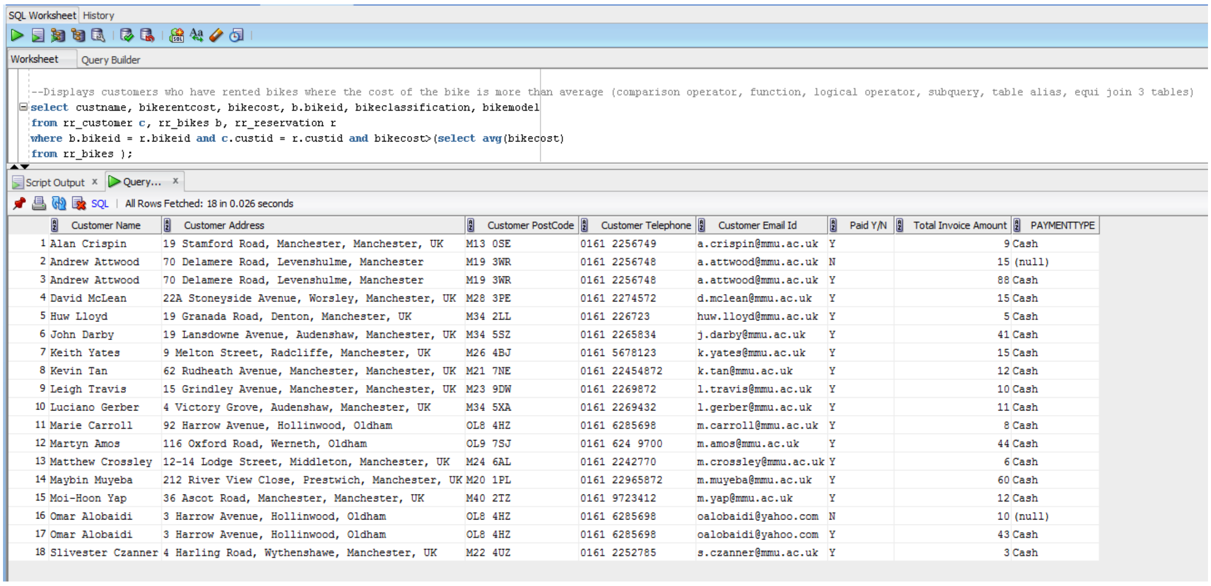


```
--Display all the Age of Bikes in years if they have purchased on or before 31 december 2012
select bikeid,bikeclassification,bikemodel,bikedop,trunc(months_between(sysdate,bikedop)/12,0) "Bike Age"
from RR_bikes
where bikedop<='01-Jan-13';
```

All Rows Fetched: 13 in 0.006 seconds

	BIKEID	BIKECLASSIFICATION	BIKEMODEL	BIKEDOP	years
1	1	Mountain	Big.Nine 300	12-APR-11	3
2	2	Mountain	Big.Nine 300	12-APR-11	3
3	3	Mountain	Big.Nine 300	12-APR-11	3
4	4	Mountain	Big.Nine 300	12-APR-11	3
5	5	Mountain	Big.Nine 300	12-APR-11	3
6	6	Road	Reacto 400	15-MAY-11	3
7	7	Road	Reacto 400	15-MAY-11	3
8	8	Road	Reacto 400	15-MAY-11	3
9	9	Mountain	Misceao 2.0	26-MAR-12	2
10	10	Mountain	Misceao 2.0	26-MAR-12	2
11	11	Mountain	Misceao 2.0	26-MAR-12	2
12	12	Mountain	Misceao 2.0	26-MAR-12	2
13	13	Mountain	Misceao 2.0	26-MAR-12	2

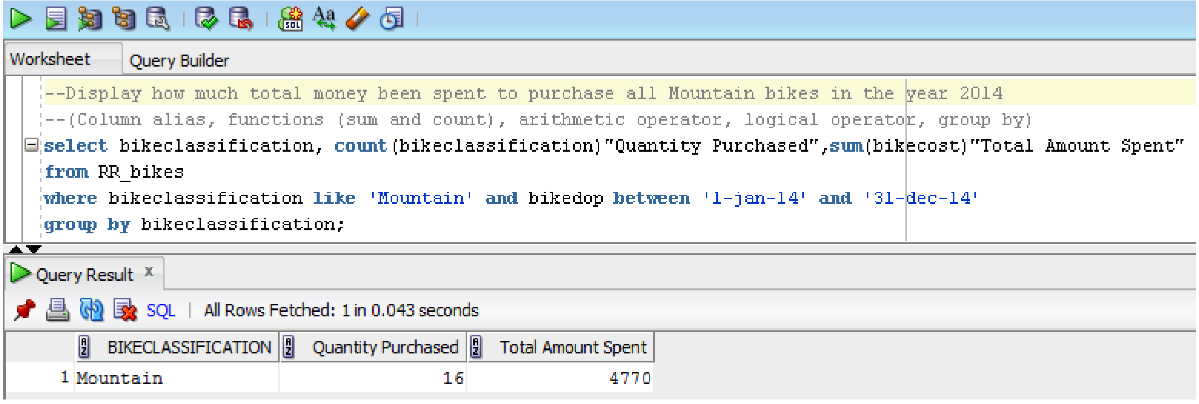
Janet D'souza



```
--Displays customers who have rented bikes where the cost of the bike is more than average (comparison operator, function, logical operator, subquery, table alias, equi join 3 tables)
select custname, bikerentcost, bikecost, b.bikeid, bikeclassification, bikemodel
from rr_customer c, rr_bikes b, rr_reservation r
where b.bikeid = r.bikeid and c.custid = r.custid and bikecost>(select avg(bikecost)
from rr_bikes );
```

All Rows Fetched: 18 in 0.026 seconds

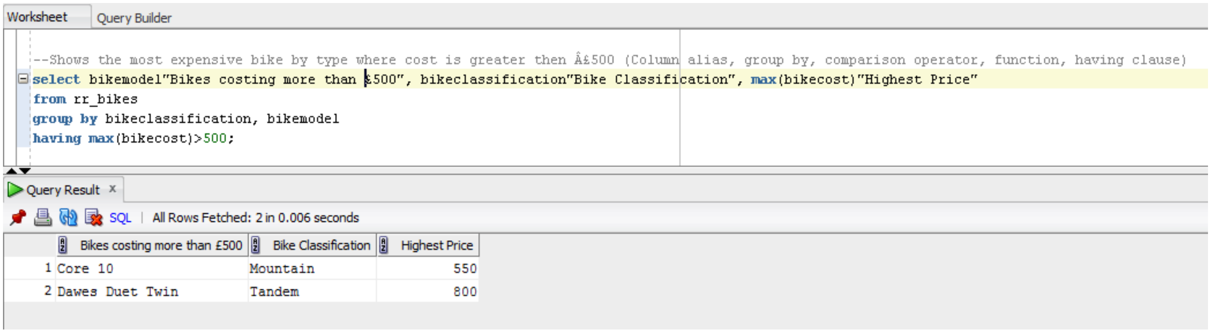
	Customer Name	Customer Address	Customer PostCode	Customer Telephone	Customer Email Id	Paid Y/N	Total Invoice Amount	PAYMENTTYPE
1	Alan Crispin	19 Stamford Road, Manchester, Manchester, UK	M13 0SE	0161 2256749	a.crispin@mmu.ac.uk	Y	9	Cash
2	Andrew Attwood	70 Delamere Road, Levenshulme, Manchester	M19 3WR	0161 2256748	a.attwood@mmu.ac.uk	N	15	(null)
3	Andrew Attwood	70 Delamere Road, Levenshulme, Manchester	M19 3WR	0161 2256748	a.attwood@mmu.ac.uk	Y	88	Cash
4	David McLean	22A Stoneyside Avenue, Worsley, Manchester, UK	M28 3PE	0161 2274572	d.mclean@mmu.ac.uk	Y	15	Cash
5	Huw Lloyd	19 Granada Road, Denton, Manchester, UK	M34 2LL	0161 226723	huw.lloyd@mmu.ac.uk	Y	5	Cash
6	John Darby	19 Lansdowne Avenue, Audenshaw, Manchester, UK	M34 5SZ	0161 2265834	j.darby@mmu.ac.uk	Y	41	Cash
7	Keith Yates	9 Melton Street, Radcliffe, Manchester, UK	M26 4BJ	0161 5678123	k.yates@mmu.ac.uk	Y	15	Cash
8	Kevin Tan	62 Rudheath Avenue, Manchester, Manchester, UK	M21 7NE	0161 22454872	k.tan@mmu.ac.uk	Y	12	Cash
9	Leigh Travis	15 Grindley Avenue, Manchester, Manchester, UK	M23 9DW	0161 2269872	l.travis@mmu.ac.uk	Y	10	Cash
10	Luciano Gerber	4 Victory Grove, Audenshaw, Manchester, UK	M34 5XA	0161 2269432	l.gerber@mmu.ac.uk	Y	11	Cash
11	Marie Carroll	92 Harrow Avenue, Hollinwood, Oldham	OL8 4HZ	0161 6285698	m.carroll@mmu.ac.uk	Y	8	Cash
12	Martyn Amos	116 Oxford Road, Werneth, Oldham	OL9 7SJ	0161 624 9700	m.amos@mmu.ac.uk	Y	44	Cash
13	Matthew Crossley	12-14 Lodge Street, Middleton, Manchester, UK	M24 6AL	0161 2242770	m.crossley@mmu.ac.uk	Y	6	Cash
14	Maybin Muyeba	212 River View Close, Prestwich, Manchester, UK	M20 1PL	0161 22965872	m.muyeba@mmu.ac.uk	Y	60	Cash
15	Moi-Hoon Yap	36 Ascot Road, Manchester, Manchester, UK	M40 2TZ	0161 9723412	m.yap@mmu.ac.uk	Y	12	Cash
16	Omar Alobaidi	3 Harrow Avenue, Hollinwood, Oldham	OL8 4HZ	0161 6285698	oalobaidi@yahoo.com	N	10	(null)
17	Omar Alobaidi	3 Harrow Avenue, Hollinwood, Oldham	OL8 4HZ	0161 6285698	oalobaidi@yahoo.com	Y	43	Cash
18	Slivester Czanner	4 Harling Road, Wythenshawe, Manchester, UK	M22 4UZ	0161 2252785	s.czanner@mmu.ac.uk	Y	3	Cash



```
--Display how much total money been spent to purchase all Mountain bikes in the year 2014
--(Column alias, functions (sum and count), arithmetic operator, logical operator, group by)
select bikeclassification, count(bikeclassification)"Quantity Purchased",sum(bikecost)"Total Amount Spent"
from RR_bikes
where bikeclassification like 'Mountain' and bikedop between '1-jan-14' and '31-dec-14'
group by bikeclassification;
```

All Rows Fetched: 1 in 0.043 seconds

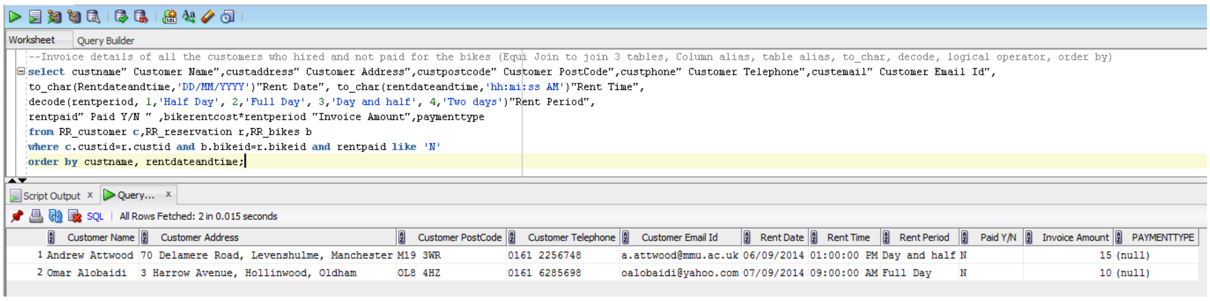
	BIKECLASSIFICATION	Quantity Purchased	Total Amount Spent
1	Mountain	16	4770



```
--Shows the most expensive bike by type where cost is greater then Â£500 (Column alias, group by, comparison operator, function, having clause)
select bikemodel"Bikes costing more than £500", bikeclassification"Bike Classification", max(bikecost)"Highest Price"
from rr_bikes
group by bikeclassification, bikemodel
having max(bikecost)>500;
```

All Rows Fetched: 2 in 0.006 seconds

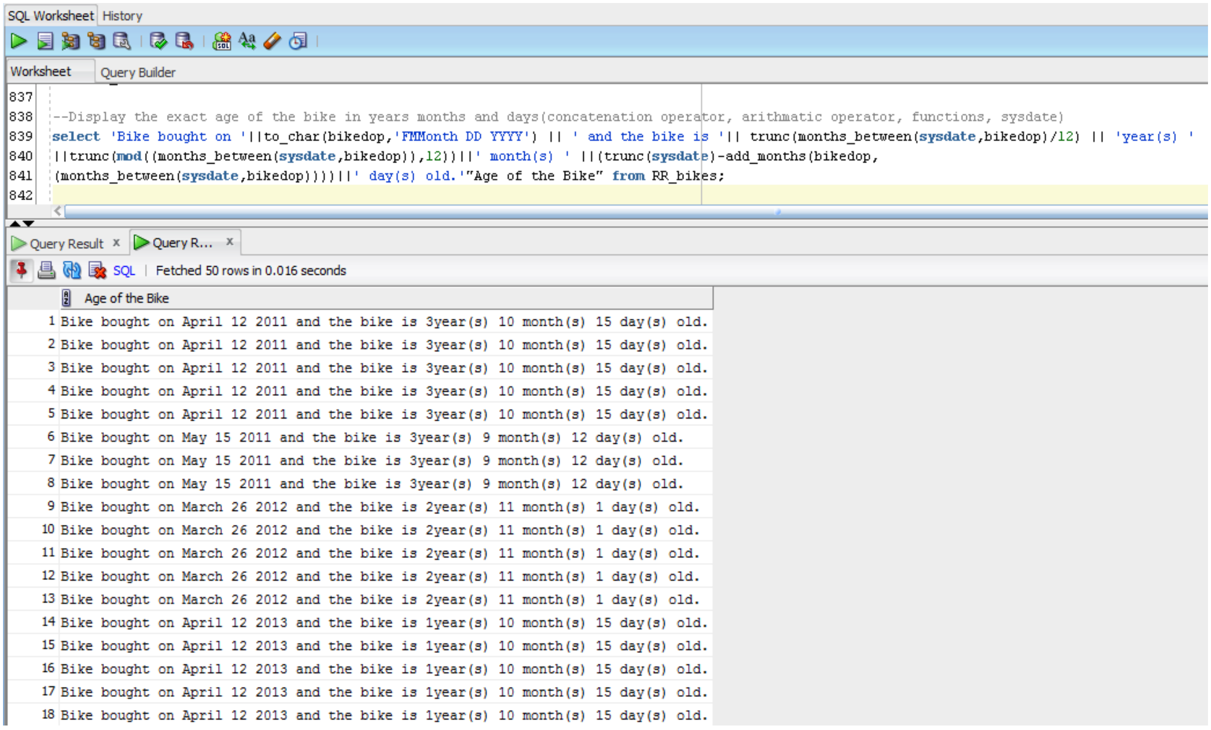
	Bikes costing more than £500	Bike Classification	Highest Price
1	Core 10	Mountain	550
2	Dawes Duet Twin	Tandem	800



```
--Invoice details of all the customers who hired and not paid for the bikes (Equi Join to join 3 tables, Column alias, table alias, to_char, decode, logical operator, order by)
select custname" Customer Name",custaddress" Customer Address",custpostcode" Customer PostCode",custphone" Customer Telephone",custemail" Customer Email Id",
to_char(Rentdateandtime,'DD/MM/YYYY')"Rent Date", to_char(rentdateandtime,'hh:mi:ss AM')"Rent Time",
decode(rentperiod, 1,'Half Day', 2,'Full Day', 3,'Day and half', 4,'Two days')"Rent Period",
rentpaid" Paid Y/N " ,bikerentcost*rentperiod "Invoice Amount",paymenttype
from RR_customer c,RR_reservation r,RR_bikes b
where c.custid=r.custid and b.bikeid=r.bikeid and rentpaid like 'N'
order by custname, rentdateandtime;
```

All Rows Fetched: 2 in 0.015 seconds

	Customer Name	Customer Address	Customer PostCode	Customer Telephone	Customer Email Id	Rent Date	Rent Time	Rent Period	Paid Y/N	Invoice Amount	PAYMENTTYPE
1	Andrew Attwood	70 Delamere Road, Levenshulme, Manchester	M19 3WR	0161 2256748	a.attwood@mmu.ac.uk	06/09/2014	01:00:00 PM	Day and half	N	15	(null)
2	Omar Alobaidi	3 Harrow Avenue, Hollinwood, Oldham	OL8 4HZ	0161 6285698	oalobaidi@yahoo.com	07/09/2014	09:00:00 AM	Full Day	N	10	(null)



```
--Display the exact age of the bike in years months and days(concatenation operator, arithmatic operator, functions, sysdate)
select 'Bike bought on '||to_char(bikedop,'FMMonth DD YYYY') || ' and the bike is '|| trunc(months_between(sysdate,bikedop)/12) || 'year(s) '
||trunc(mod((months_between(sysdate,bikedop)),12))||' month(s) ' ||(trunc(sysdate)-add_months(bikedop,
(months_between(sysdate,bikedop))))||' day(s) old.'"Age of the Bike" from RR_bikes;
```

Fetched 50 rows in 0.016 seconds

	Age of the Bike
1	Bike bought on April 12 2011 and the bike is 3year(s) 10 month(s) 15 day(s) old.
2	Bike bought on April 12 2011 and the bike is 3year(s) 10 month(s) 15 day(s) old.
3	Bike bought on April 12 2011 and the bike is 3year(s) 10 month(s) 15 day(s) old.
4	Bike bought on April 12 2011 and the bike is 3year(s) 10 month(s) 15 day(s) old.
5	Bike bought on April 12 2011 and the bike is 3year(s) 10 month(s) 15 day(s) old.
6	Bike bought on May 15 2011 and the bike is 3year(s) 9 month(s) 12 day(s) old.
7	Bike bought on May 15 2011 and the bike is 3year(s) 9 month(s) 12 day(s) old.
8	Bike bought on May 15 2011 and the bike is 3year(s) 9 month(s) 12 day(s) old.
9	Bike bought on March 26 2012 and the bike is 2year(s) 11 month(s) 1 day(s) old.
10	Bike bought on March 26 2012 and the bike is 2year(s) 11 month(s) 1 day(s) old.
11	Bike bought on March 26 2012 and the bike is 2year(s) 11 month(s) 1 day(s) old.
12	Bike bought on March 26 2012 and the bike is 2year(s) 11 month(s) 1 day(s) old.
13	Bike bought on March 26 2012 and the bike is 2year(s) 11 month(s) 1 day(s) old.
14	Bike bought on April 12 2013 and the bike is 1year(s) 10 month(s) 15 day(s) old.
15	Bike bought on April 12 2013 and the bike is 1year(s) 10 month(s) 15 day(s) old.
16	Bike bought on April 12 2013 and the bike is 1year(s) 10 month(s) 15 day(s) old.
17	Bike bought on April 12 2013 and the bike is 1year(s) 10 month(s) 15 day(s) old.
18	Bike bought on April 12 2013 and the bike is 1year(s) 10 month(s) 15 day(s) old.

What has been learnt in the process of creating the SQL database:

Mark Bellingham

In part three of our project I have learnt how to deal with many-to-many relationships because they cause problems in a relational database so an extra entity or table has to be created to split it into two one-to-many relationships. Creating data dictionaries, identifying different data types, deciding on and choosing appropriate value limits for them. Using constraints to check, verify and limit the allowed data in a field. Primary keys to identify tables, foreign keys which are primary keys linking one table with another and composite keys where a table has more than one primary key, which can be unique or also a foreign key.

In SQL I have learnt how to create tables, create sequences, drop tables (which also deletes the data contained within), insert data into the fields of tables, perform queries to extract data from one or more table. I have been performing arithmetic calculations on data in a table and then displaying the result; compare data from two or more tables and make decisions based on the result. I have been able to rename column headings to something more user-friendly than the attribute name, display data in ascending or descending order, group similar fields together to show totals. I can use the 'having' clause to only show data from the result of queries which meet specified limits, I can use sub-queries to output the data from one query into another query. I am able to use different date formats and manipulate what is shown, compare dates with each other and do all these together with the system date also. I am able to locate and fix errors in table structure, locate and fix errors in table data or show it as an exception query. I can use decode to change how data from a table is displayed.

Maryam Elgahmi

In part three, I have learnt how useful data dictionary is. After describing each table and each attribute, it has helped me create the tables in oracle because the data dictionary included what data type and length it needs for each attribute name, and this has made it easier and quicker for me to create each table. It was also very beneficial to have the constraint and constraint name before creating the table. Data dictionary has helped me to create and insert data in SQL developer easily. I have learnt how to create a constraint check for a sale amount. I also know now why constraint unique can be important when implementing the database design.

After implementing the database design, I have learnt how to reference foreign key with two different attribute names. I have also learnt the composite key where there is an attribute name with a primary key and a foreign key. I have found it more useful to make the data type length long, because if you do not you can find problems later when inserting the data. As a group we have faced a problem with the address attribute name and this was because the attribute name was short, so we had to go back and change the value to 60. It was very beneficial to create a sequence, so then you don't have to keep typing the next number. With date and varchar2 you have to add single quotation, were with number you do not have to, because SQL would not recognise it then. When you want an empty field, you can just add quotation. It was my first time using drop statements. Drop statements were useful when you wanted to delete all the tables. With queries, I have learnt how to select, group by query from table names and how to use natural join to help to find which two columns to compare, I have used order by in my query to show which order I want the query to be. Finally I have learnt if you want a specific query you can use ‘where’, ‘between’, ‘and ‘.

Janet D’Souza

In part3, I have learnt the ER diagram represents the conceptual level and relational database is the logical level for the database. An entity within ER diagram is easy to convert into a table. Each attributes of the entity turned as column or field names in the table considering what type of data and length we are going to store in the individual fields selecting appropriate datatypes. Naming and applying constraints to the fields to restrict the correct data been inserted and to make the data entering in the fields mandatory. The key attribute of the entity can be primary or composite key same logic been transferred in the table. In relational database design, a many-to-many relationship is not allowed, to get around the problem of having a many-to-many relationship we need to break apart the many-to-many relationship into two one-to-many relationships. While inserting records to the table there is a rule to follow in case we do not want to insert data to all the fields of the table. Inserting date and time to the table using function when we want to store the date or time in different formats. Creating and applying the sequence to the table and the importance of it. When there is Primary and foreign key in the tables, Primary key uniquely identify a row in database and a foreign key placed constraint on the data in the related tables to ensure data referential integrity as well as consistency. How to rearrange all the tables and the order in which data needs to be inserted in the tables. Using Update and delete commands to manipulate the data and DROP command to delete the table structure along with the data.

When I was doing queries, I learnt how to use arithmetic calculations, using logical, and comparison operators. DECODE to temporary show the alternate values to the given data, where keyword could not be used with aggregate functions like sum, max and count with group by and ‘Having’ clause can be used. There are different types of Joins available to connect more than one table. Natural join automatically recognises common field between two tables. In Equi Join we have to specify which two fieldnames want to join and if you want to display any field name which is common in both tables, has to be prefixed with table alias or the table name itself. Non Equi Joins used operators other than = and the importance of LEFT outer Join and RIGHT outer join. Date functions are very useful Add_months function is used to add exact number of months to any date it also recognises exact days in the month, months_between function helped to find the age of the bike along with the sysdate from dual table. When and how to use Subqueries, usage of table alias and column alias, concatenating fields with string. Learning backend tool Oracle is a good experience to store the bulk business information into the database.

Project Conclusion

Mark Bellingham

In the course of this assignment I have learnt about different types of data enquiries and management reports and how they can be used. I have learnt how to identify use cases and entities from the case study. I have learnt how to create Use Case Diagrams and assign priorities to each of the use cases. How to create use case specifications from each of the use cases in the UCD, which describe how the use case will work from beginning to end, including any possible alternative routes to get there. How to create a top-down Entity Relationship Diagram from the case study. I have learnt how to use Relational Data Analysis to normalise the attributes, which helps to identify any missing attributes and any other problems with the database which may not be immediately apparent. I have created a bottom-up ERD from existing paperwork using the RDA. I have created data dictionaries, identifying and defining data types.

I have learnt how to convert an ERD into SQL tables, creating those tables and inserting, updating and deleting data from them. I have been able to extract data from one or more tables using queries and then displaying that data in different ways using order by, group by, ascending and descending. I have been able to perform arithmetic and logical calculations with data in SQL, further refine query outputs using 'having' and subqueries, locate and fix errors in table structure and table data. I have learnt how to organise and prepare a presentation for an audience. Finally, I have learnt how to work in a group, organising timetables and sharing responsibilities.

Janet D'Souza

After reading Rays Rental case study, I could identify and visualise what are the problems faced by the current manual system. System requirements of the proposed computerised system shows the different tasks new system can do efficiently and quickly. A use case show activities and can be a collection of possible activities related to a particular goal. In this project, I have learnt to identify actors and the activities involved by the actors and how to prioritise the use cases according to MoSCoW Rule. ER diagram shows the structure of the system and how they associate with each other. How to identify the different types of relationship between entities. In relational database design, a many-to-many relationship is not allowed, to get around the problem of having a many-to-many relationship we need to break apart the many-to-many relationship into two one-to-many relationships.

Creating tables, identifying appropriate data types and size, Naming and applying constraints to the fields to restrict the incorrect data been inserted and to make the data entering in the fields mandatory. Identifying Primary keys to uniquely identify a row in a database and foreign keys which are primary keys linked to the parent table. Using more than one primary key in your table is called composite key and combination of those are unique. I have learnt create sequence, drop tables, insert records and query the database to extract desired output. I have used arithmetic, logical and comparison operators, arithmetic and date functions with the queries. Used order by and group by to display the data in a required format. I have used joins to connect more than one table to extract information from multiple tables. I was confident to use most of the aspects of Oracle in this project

Maryam Elghami

In this project, I have learnt how to assess the needs of Ray rentals business. Checking what the actual problems for the business, made a better plan by setting out suitable requirements and several data enquires to design for the new system. Use case diagrams and use case specification were helpful to create before designing the new system, because you know what the role for some staff like Ray and the parts manager. I have also learnt how to make a good ERD for the system for each entity and its attribute and how some entities are linked, and the primary key and foreign keys.

I have learnt from data dictionary how to create a check constraint for some attribute and what are the data types: number, varchar2, date and char. In oracle I have learnt how to create, insert and drop tables and drop sequence. With queries, I have learnt how to select, group by query from table names and how to use natural join to help me to find which two columns to compare, I have used order by in my query to show which order I want the query to be. If you want a specific query you can use 'where', 'between', 'and '. Finally, I have learnt how sequence are useful to insert in some of the tables.

References

Curry, A., Flett, P. and Hollingsworth, I. (2006). *Managing information and systems*. London: Routledge.

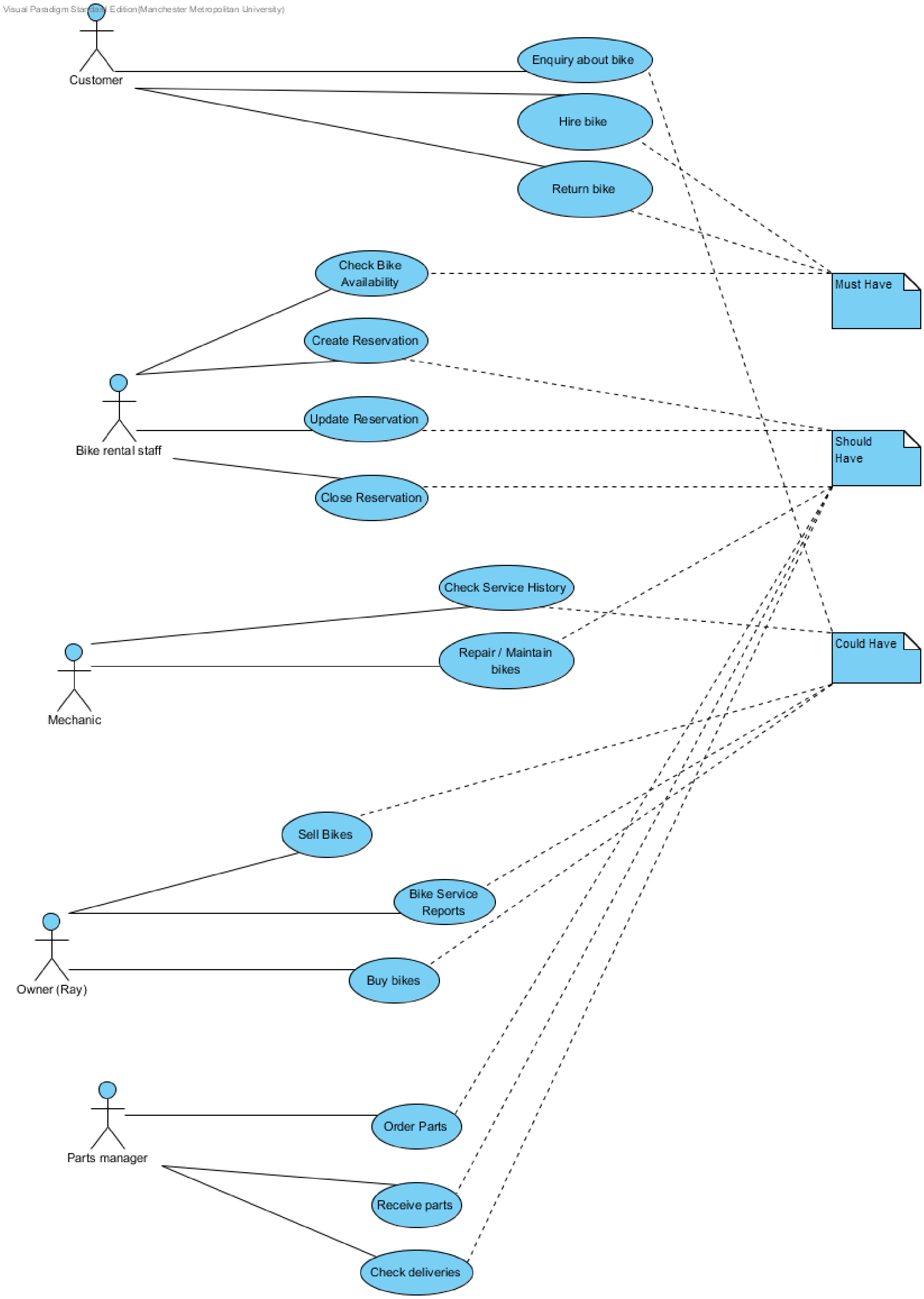
Eccles, M., Julyan, F., Boot, G. and van Belle, J. (2004). *The Principles of Business Computing*. 5th ed. Juta & Co Ltd.

Whiteley, D. (2013). *An introduction to information systems*. 1st ed. Palgrave Macmillan.

Nagpal, D. (2011). *Textbook on management information systems*. New Delhi: S Chand.

Appendices

Extra Use case diagram which is referred to in Part 2 but not included in the project



Extra Use Case Specifications

Use Case: Enquires about bike
Owner: Customer
Pre-Conditions
Customer rings, email or asks reception about a certain bike.
Post-Conditions
Customer receives list of bike prices. Customer record created.
Primary Path
Customer ring, emails or in person attends Check database for availability Create report of available bikes and prices Send list to customer Create customer record
Alternate Path
Notes
Customers details is checked after one week if no reservation is made, follow up enquiry.

Use Case: Hire bike

Owner: customer

Pre-Conditions

Customer chooses a bike and informs a staff

Post-Conditions

Customer leaves with bike with due date and time after it has been checked out in database from a staff.

Primary Path

Customer provides staff with their contact details, chooses type of bike and hire date

Staff records customers information and save it in database and checks availability.

Customer hires bike

Customer makes payment

Alternate Path

Customer makes reservation in advance

Staff updates reservation for bike record.

Notes

Use Case: Return bike

Owner: Customer

Pre-Conditions

Customer returns back bike to a member of staff

Post-Conditions

Rental record is updated with return time

Customer receives invoice

Primary Path

Staff put back bike in place

Staff update bike record with the time it has been returned

Staff creates invoice for customer.

Alternate Path

Notes

Customer makes a complaint about the bike

Staff records complaint and updates bike fault records

Use Case: Pay for hire

Owner: Customer

Pre-Conditions

Customer makes payment cash or cheque

Post-Conditions

Customer receives a confirmation with a receipt that payments has been made

Primary Path

Staff checks total price

Customer makes payment by cash

Receipt printed for customer to keep safe

Alternate Path

Customer sends cheque through the post in advance

Receipt is posted back to the customer

Notes

Use Case: Repair / Maintain Bikes

Owner: Mechanic

Pre-Conditions

Owner Ray sends list of bikes which not been serviced for a month.

Mechanics receive bike faults from hiring department which are pointed out by the customers.

Post-Conditions

When mechanic carried out the work and serviced the bike maintenance history is updated

Primary Path

Owner sends list of bikes for maintenance

The availability list is to be changed for the specific type of bike

Bike details are entered into the bike service/maintenance file

When the work is carried out maintenance history is updated.

Alternate Path

Notes

Use Case: Buy Bikes

Owner: Ray

Pre-Conditions

Need more bikes

Post-Conditions

Have new bikes

Have completed bike record and rental record for each bike

Primary Path

Buy and receive bike from manufacturer

Create bike record, which includes: bike number; model; manufacturer; date of purchase; cost; classification; size; disposal details; maintenance history

Create rental record, which includes: bike number; bike name; bike type; bike size; rent date; time out; time back (due and actual); customer details; amount paid

Alternate Path

None

Notes

Use Case: Bike Service Reports

Owner: Ray

Pre-Conditions

Find out which bikes need servicing

Post-Conditions

Details of which bikes need servicing have been passed to the maintenance department

Primary Path

Check bike records

Create report where last service date is more than one month ago

List of bikes is passed to the maintenance department

Alternate Path

Customer complains about bike fault

Reception updates a list of bikes with faults

List of bikes is passed to the maintenance department

Notes

Use Case: Sell Bikes

Owner: Ray

Pre-Conditions

Bike is more than 2 years old

Post-Conditions

Bike is sold to a local dealer

Primary Path

Check the bike records

Create a report for all bikes with a date-of-purchase which is more than two years old

Sell bikes in the report to a local dealer

Update bike record with the details of who bought the bike

Alternate Path

None

Notes

Use Case: Receive Parts
Owner: Parts Manager
Pre-Conditions
Parts Manager ordered bike parts
Post-Conditions
Parts are received and order file updated
Primary Path
Ordered parts delivery received Check the ordered parts are delivered checking against the order file with the invoice / delivery note received. Copies of the parts ordered and delivery notes are stored in the database
Alternate Path
Notes

Oracle Script File

```
--Rays Rental database desgin


DROP SEQUENCE ENQUIRY_SEQUENCE;
DROP SEQUENCE PARTSORDERED_SEQUENCE;
DROP SEQUENCE Bike_Sequence;
DROP SEQUENCE Cust_Sequence;
DROP SEQUENCE Main_Sequence;
DROP SEQUENCE RESERVATION_SEQUENCE;


DROP TABLE RR_MAINTENANCE;
DROP TABLE RR_ORDERDETAILS;
DROP TABLE RR_PARTSORDERED;
DROP TABLE RR_PARTS;
DROP TABLE RR_RESERVATION;
DROP TABLE RR_ENQUIRY;
DROP TABLE RR_SELL;
DROP TABLE RR_BIKES;
DROP TABLE RR_STAFF;
DROP TABLE RR_LOCALDEALER;
DROP TABLE RR_SUPPLIERS;
DROP TABLE RR_MANUFACTURER;
DROP TABLE RR_DEPARTMENT;
DROP TABLE RR_CUSTOMER;




create table RR_CUSTOMER(
CustId NUMBER(10) Constraint CustIDPK PRIMARY KEY,
CustName Varchar2(25) Constraint CustNmNN NOT NULL,
CustAddress Varchar2(60) Constraint CustAdrNN NOT NULL,
CustPostCode Varchar2(15) Constraint CustPCNN NOT NULL,

CustPhone Varchar2 (14) Constraint CustPhNN NOT NULL,

CustEmail VARCHAR2(40)

);


create table RR_DEPARTMENT(

DeptName VARCHAR2(30) constraint DeptNmPK PRIMARY KEY,

DeptManager VARCHAR2(30)

);


Create table RR_MANUFACTURER(

ManId VARCHAR2(3) Constraint ManIdPK PRIMARY KEY,

ManName VARCHAR2(25) Constraint ManNameNN NOT NULL,

ManAddress VARCHAR2(80) Constraint ManAddNN NOT NULL,

ManPostCode VARCHAR2(15) Constraint ManPCNN NOT NULL,

ManPhone VARCHAR2(20),

ManEmail VARCHAR2(40),

ManWebsite VARCHAR2(30)

);


Create table RR_SUPPLIERS(

SuppId NUMBER(3) Constraint SuppIdPK PRIMARY KEY,

SuppName VARCHAR2(25) Constraint SuppNmNN NOT NULL,

SuppAddress VARCHAR2(80) Constraint SuppAdrNN NOT NULL,

SuppPostcode VARCHAR2 (15) Constraint SuppPCNN NOT NULL,

SuppPhone VARCHAR2(14) Constraint SuppPhNNUQ NOT NULL UNIQUE

);


Create table RR_LOCALDEALER(

LocalDealerId VARCHAR2(10) Constraint LocalDealerIdPK PRIMARY KEY,

LDName VARCHAR2(25) Constraint LDNmNN NOT NULL,
LDAddress VARCHAR2(60) Constraint LDAdrNN NOT NULL,
LDPostCode VARCHAR2(15) Constraint LDPCNN NOT NULL,
LDPhone VARCHAR2(14) Constraint LDPhone NOT NULL
);

create table RR_BIKES(
BikeId NUMBER(6) Constraint BikeIdPK PRIMARY KEY,
BikeModel VARCHAR2(20) Constraint BikeModelNNChk NOT NULL,
BikeClassification VARCHAR2(10) Constraint BikeClassChk CHECK(BikeClassification in('Mountain','Road','Tandem')),
BikeSize VARCHAR2(15) CONSTRAINT BikeSizeChk CHECK(BikeSize IN ('large male','standard male','small male','standard female','child')),
BikeDOP DATE Constraint BikeDOPNN NOT NULL,
BikeCost NUMBER(6,2) Constraint BikeCstChk check(BikeCost>0),
BikeRentCost NUMBER(4,2) NOT NULL,
LocalDealerId VARCHAR2(10), foreign key (LocalDealerId) references RR_LOCALDEALER (LocalDealerId),
ManId VARCHAR2(3),  foreign key (ManId) references RR_MANUFACTURER(ManId)
);

Create table RR_SELL(
BikeId NUMBER(6),
LocalDealerId VARCHAR2(10),
SaleAmount NUMBER(6,2) Constraint SellSAmtChk CHECK (SaleAmount>=0),
SaleDate DATE,
PRIMARY KEY(BikeId,LocalDealerId)
);

create table RR_STAFF(
StaffId NUMBER(5) constraint StfIdPK PRIMARY KEY,
StaffName VARCHAR2(30) constraint StfNmNN NOT NULL,
StaffAddress VARCHAR2(60) constraint StfAdrsNN NOT NULL,
StaffPostCode VARCHAR2(15) constraint StfPCNN NOT NULL,
StaffPhone VARCHAR2(20) constraint StfPhNN NOT NULL,
HireDate DATE constraint StfHDtNN NOT NULL,
DeptName Varchar(30), FOREIGN KEY(DeptName) REFERENCES RR_DEPARTMENT(DeptName)
);

create table RR_ENQUIRY(
EnquiryId NUMBER(10) Constraint EnqIDPK PRIMARY KEY,
EnquiryDateNTime DATE Constraint EnqDtNN NOT NULL ,
EnquiryPeriod NUMBER(1) Constraint EnqPrdNN NOT NULL,
StaffId Number(5), FOREIGN KEY(StaffId) References RR_Staff(StaffId),
CustId Number (10),  FOREIGN KEY(CustId) References RR_Customer(CustId)
);

create table RR_RESERVATION(
BikeId NUMBER(6),FOREIGN KEY (BikeId)REFERENCES RR_BIKES(BikeId),
RentDateAndTime DATE,
TimeBackDue DATE,
TimeBackActual DATE,
RentPeriod NUMBER(2)  DEFAULT '1' constraint ResrvRntPrdNN NOT NULL,
RentPaid CHAR (1) DEFAULT 'N' CONSTRAINT ResrvRntPdCk CHECK (RentPaid IN('Y','N')),
PaymentType VARCHAR2 (15) CONSTRAINT InvPyTypChk Check (paymentType in('Cash','card', 'cheque')),
PaymentRefNo NUMBER(10),
StaffId NUMBER(5 ),FOREIGN KEY (StaffId)REFERENCES RR_STAFF(StaffId),
CustId NUMBER(10), FOREIGN KEY(CustId) REFERENCES RR_CUSTOMER(CustId),
PRIMARY KEY(BikeId, RentDateAndTime)
);

create table RR_PARTS(
PartId NUMBER(4) Constraint PartIdPK PRIMARY KEY,
PartName VARCHAR2(25) Constraint PartNmNN NOT NULL,
PartDescription VARCHAR2(60),
PartCost NUMBER(6,2) Constraint PartCostChk check(PartCost>0),
StockLevel NUMBER(3),
ReOrderLevel NUMBER(2),
UnitsOnOrder NUMBER(3),
SuppId NUMBER(3), FOREIGN KEY (SuppId) REFERENCES RR_SUPPLIERS (SuppId),
ManId Varchar2(3), FOREIGN KEY(ManId) REFERENCES RR_MANUFACTURER (ManId)
);

create table RR_PARTSORDERED(
OrderNo NUMBER(10) Constraint PartsOrderNoPK PRIMARY KEY,
OrderDate DATE,
DeliveryExpectedDate DATE,
SuppId NUMBER(3), FOREIGN KEY(SuppId) REFERENCES RR_SUPPLIERS (SuppId),
ManId Varchar2(3),FOREIGN KEY(ManId) REFERENCES RR_MANUFACTURER (ManId)
);

create table RR_ORDERDETAILS(
OrderNo NUMBER(10), FOREIGN KEY(OrderNo) REFERENCES RR_PARTSORDERED (OrderNo),
PartId NUMBER(3), FOREIGN KEY(PartId) REFERENCES RR_PARTS (PartId),
OrderQuantity NUMBER(3) Constraint OrderDetQtyChk CHECK(OrderQuantity > 0),
ReceivedQuantity NUMBER(3),
OrderReceivedDate DATE,
DeliveryNo VARCHAR2(15),
Comments VARCHAR(50),
PRIMARY KEY(OrderNo, PartId)
);

create table RR_MAINTENANCE(
MaintenanceId NUMBER(10) PRIMARY KEY,
BikeFault VARCHAR2(30),
DateReported DATE,
DateActioned DATE,
ActionTaken VARCHAR2(30),
PartQuantity NUMBER(2),
PartId NUMBER(4),foreign key (PartId) references RR_Parts(PartId),
BikeId NUMBER(6),Foreign key(BikeId) references RR_Bikes(BikeId)
);



CREATE SEQUENCE Cust_Sequence START WITH 1
INCREMENT BY 1
MINVALUE 1
MAXVALUE 1000000;

INSERT INTO RR_Customer VALUES (
Cust_Sequence.nextval, 'Omar Alobaidi', '3 Harrow Avenue, Hollinwood, Oldham','OL8 4HZ','0161 6285698', 'oalobaidi@yahoo.com');
INSERT INTO RR_Customer VALUES (
Cust_Sequence.nextval, 'Martyn Amos', '116 Oxford Road, Werneth, Oldham','OL9 7SJ','0161 624 9700', 'm.amos@mmu.ac.uk');
INSERT INTO RR_Customer VALUES (
Cust_Sequence.nextval, 'Andrew Attwood', '70 Delamere Road, Levenshulme, Manchester','M19 3WR','0161 2256748', 'a.attwood@mmu.ac.uk');
INSERT INTO RR_Customer VALUES (
Cust_Sequence.nextval, 'Marie Carroll', '92 Harrow Avenue, Hollinwood, Oldham','OL8 4HZ','0161 6285698', 'm.carroll@mmu.ac.uk');
INSERT INTO RR_Customer VALUES (
Cust_Sequence.nextval, 'Nicholas Costen', '3 Ford Lane, Wythenshawe, Manchester, UK','M22 4WE','0161 6276436', 'n.costen@mmu.ac.uk');
INSERT INTO RR_Customer VALUES (
Cust_Sequence.nextval, 'Alan Crispin', '19 Stamford Road, Manchester, Manchester, UK','M13 0SE','0161 2256749', 'a.crispin@mmu.ac.uk');
INSERT INTO RR_Customer VALUES (
Cust_Sequence.nextval, 'Matthew Crossley', '12-14 Lodge Street, Middleton, Manchester, UK','M24 6AL','0161
2242770', 'm.crossley@mmu.ac.uk');

INSERT INTO RR_Customer VALUES (

Cust_Sequence.nextval, 'Slivester Czanner', '4 Harling Road, Wythenshawe, Manchester, UK','M22 4UZ','0161
2252785', 's.czanner@mmu.ac.uk');

INSERT INTO RR_Customer VALUES (

Cust_Sequence.nextval, 'John Darby', '19 Lansdowne Avenue, Audenshaw, Manchester, UK','M34 5SZ','0161
2265834', 'j.darby@mmu.ac.uk');

INSERT INTO RR_Customer VALUES (

Cust_Sequence.nextval, 'Luciano Gerber', '4 Victory Grove, Audenshaw, Manchester, UK','M34 5XA','0161
2269432', 'l.gerber@mmu.ac.uk');

INSERT INTO RR_Customer VALUES (

Cust_Sequence.nextval, 'Huw Lloyd', '19 Granada Road, Denton, Manchester, UK','M34 2LL','0161 226723',
'huw.lloyd@mmu.ac.uk');

INSERT INTO RR_Customer VALUES (

Cust_Sequence.nextval, 'David McLean', '22A Stoneyside Avenue, Worsley, Manchester, UK','M28 3PE','0161
2274572', 'd.mclean@mmu.ac.uk');

INSERT INTO RR_Customer VALUES (

Cust_Sequence.nextval, 'Ruth Meyer', '72 Market Street, Manchester, Manchester, UK','M25 9TE','0161
2287872', 'r.meyer@mmu.ac.uk');

INSERT INTO RR_Customer VALUES (

Cust_Sequence.nextval, 'Maybin Muyeba', '212 River View Close, Prestwich, Manchester, UK','M20 1PL','0161
22965872', 'm.muyeba@mmu.ac.uk');

INSERT INTO RR_Customer VALUES (

Cust_Sequence.nextval, 'Kevin Tan', '62 Rudheath Avenue, Manchester, Manchester, UK','M21 7NE','0161
22454872', 'k.tan@mmu.ac.uk');

INSERT INTO RR_Customer VALUES (

Cust_Sequence.nextval, 'Leigh Travis', '15 Grindley Avenue, Manchester, Manchester, UK','M23 9DW','0161
2269872', 'l.travis@mmu.ac.uk');

INSERT INTO RR_Customer VALUES (

Cust_Sequence.nextval, 'Brian Wendl', '16 Stansfield Road, Failsworth, Manchester, UK','M35 9EA','0161
4685564', 'b.wendl@mmu.ac.uk');

INSERT INTO RR_Customer VALUES (

Cust_Sequence.nextval, 'Moi-Hoon Yap', '36 Ascot Road, Manchester, Manchester, UK','M40 2TZ','0161
9723412', 'm.yap@mmu.ac.uk');
```

```
INSERT INTO RR_Customer VALUES (

Cust_Sequence.nextval, 'Keith Yates', '9 Melton Street, Radcliffe, Manchester, UK','M26 4BJ','0161 5678123',
'k.yates@mmu.ac.uk');
```

```
INSERT INTO RR_Department

VALUES ('Boss', 'Ray');

INSERT INTO RR_Department

VALUES ('Hirings', 'Pete');

INSERT INTO RR_Department

VALUES ('Maintenence', 'Alf');

INSERT INTO RR_Department

VALUES ('Parts', 'Paul');
```

```
INSERT INTO RR_Manufacturer

VALUES ('101', 'Merida', 'Unit 13 Nott`m Sth. Wilford Ind. Est. Ruddigton Lane Wilford Nottingham', 'NG11 7EP',
'+44(0)1159817788', 'merida@meridauk.com', 'www.merida-bikes.com');

INSERT INTO RR_Manufacturer

VALUES ('102', 'Raleigh', ' ', ' ', '01773 532600', 'info@raleigh.co.uk', 'http://www.raleigh.co.uk/');

INSERT INTO RR_Manufacturer

VALUES ('103', 'Giant', 'Charnwood Edge, Syston Road, Cossington, Leicester', 'LE7 4UZ', '0844 245 9030',
'info@giant-bicycles.com/', 'http://www.giant-bicycles.com/');

INSERT INTO RR_Manufacturer

VALUES ('104', 'Tandem Group PLC', '35 Tameside Drive, Castle Bromwich, Birmingham', 'B35 7AG', '+44 (0)121
748 8075', 'info@tandemgroup.co.uk', 'http://tandemgroupplc.co.uk/');

INSERT INTO RR_Manufacturer

VALUES ('105', 'Pashley Cycles', 'Stratford-Upon-Avon, Warwickshire', 'CV37 9NL', '+44 (0)1789 292 263',
'hello@pashley.co.uk', 'http://www.pashley.co.uk/');

INSERT INTO RR_Manufacturer

VALUES ('106', 'Genesis', ' ', ' ', ' ', 'http://www.genesisbikes.co.uk/contact', 'http://www.genesisbikes.co.uk');
```

```
INSERT INTO RR_SUPPLIERS

VALUES (101, 'M + J DISTRIBUTORS LTD', 'UNIT A, HANIX BUILDINGS , WINDMILL LANE, DENTON ,
MANCHESTER', 'M34 3SP',  '0161 337 9600');

INSERT INTO RR_SUPPLIERS

VALUES (102, 'Reece Cycles', '100 Alcester Street Birmingham', 'B12 0QB', '0121 622 0180');

INSERT INTO RR_SUPPLIERS

VALUES (103, 'Cycle Division Ltd', 'Unit 27 Gatehouse Enterprise Centre, Albert Street, Lockwood Huddersfield',
'HD1 3QD', '0845 0508 500');

INSERT INTO RR_SUPPLIERS

VALUES (104, 'Ison Distribution Ltd', '201 Lancaster Way, Business Park, Ely, Cambridgeshire.', 'CB6 3NX', '0845
0507 500');

INSERT INTO RR_SUPPLIERS

VALUES (105, 'Hykeham Wholesale Ltd', 'Unit 7, Earlsfield Close, Sadler Road, Lincoln', 'LN6 3RT', '01522
801550');

INSERT INTO RR_SUPPLIERS

VALUES (106, 'Haven Distribution ', '2 Red Kiln Close, Horsham, West Sussex Manchester ', 'RH13 5Q', '07827
797044');



INSERT INTO RR_LocalDealer

VALUES ('BCH1001M4', 'Bishopthorpe Cycle Hire', 'Unit 3 Manchester Road, Newton Health, Manchester', 'M40
2EP',  '0161 912 8300');

INSERT INTO RR_LocalDealer

VALUES ('GCH1001M21', 'Grimsby Cycle Hub', '68-70 Dickenson Road, Manchester ', 'M21 7LA', '0161 224
1303');

INSERT INTO RR_LocalDealer

VALUES ('SC1001M16', 'Snowdonia Cycles', '26 Burton Road, Manchester', 'M16 5LW', '0161 4453492');

INSERT INTO RR_LocalDealer

VALUES ('CBH1001M20', 'Cheltenham Bike Hire', '5 Lane End Road, Manchester', 'M20 1AL', '0161 431 0777');

INSERT INTO RR_LocalDealer

VALUES ('BC1001SK5', 'Bourton Cycles', '201 Houldsworth Mill Waterhouse Way, Reddish, Stockport', 'SK5 9NL',
'07940 859672');

INSERT INTO RR_LocalDealer

VALUES ('KCH100M14', 'Kool Cycle Hire ', '7 Wilmslow Road Manchester ', 'M14 5LW ', '0161 2573897');
```

```
CREATE SEQUENCE Bike_Sequence START WITH 1
INCREMENT BY 1
MINVALUE 1
MAXVALUE 1000000;


INSERT INTO RR_Bikes
VALUES (Bike_Sequence.nextval, 'Big.Nine 300', 'Mountain', 'large male', '12-Apr-11', 250.00, 5,'BCH1001M4',
'101');
INSERT INTO RR_Bikes
VALUES (Bike_Sequence.nextval, 'Big.Nine 300', 'Mountain', 'large male', '12-Apr-11', 250.00, 5,'GCH1001M21',
'101');
INSERT INTO RR_Bikes
VALUES (Bike_Sequence.nextval, 'Big.Nine 300', 'Mountain', 'large male', '12-Apr-11', 250.00, 5,'BC1001SK5',
'101');
INSERT INTO RR_Bikes
VALUES (Bike_Sequence.nextval, 'Big.Nine 300', 'Mountain', 'large male', '12-Apr-11', 250.00, 5,'SC1001M16',
'101');
INSERT INTO RR_Bikes
VALUES (Bike_Sequence.nextval, 'Big.Nine 300', 'Mountain', 'large male', '12-Apr-11', 250.00, 5,'CBH1001M20',
'101');
INSERT INTO RR_Bikes
VALUES (Bike_Sequence.nextval, 'Reacto 400', 'Road', 'large male', '15-May-11', 300.00, 5,'GCH1001M21', '101');
INSERT INTO RR_Bikes
VALUES (Bike_Sequence.nextval, 'Reacto 400', 'Road', 'large male', '15-May-11', 300.00, 5,'KCH100M14', '101');
INSERT INTO RR_Bikes
VALUES (Bike_Sequence.nextval, 'Reacto 400', 'Road', 'large male', '15-May-11', 300.00, 5,'GCH1001M21', '101');
INSERT INTO RR_Bikes
VALUES (Bike_Sequence.nextval, 'Misceao 2.0', 'Mountain', 'standard male', '26-Mar-12', 200.00, 5,
'SC1001M16', '102');
INSERT INTO RR_Bikes
```

```
VALUES (Bike_Sequence.nextval, 'Misceao 2.0', 'Mountain', 'standard male', '26-Mar-12', 200.00, 5,
'CBH1001M20', '102');

INSERT INTO RR_Bikes

VALUES (Bike_Sequence.nextval, 'Misceao 2.0', 'Mountain', 'standard male', '26-Mar-12', 200.00, 5,
'KCH100M14', '102');

INSERT INTO RR_Bikes

VALUES (Bike_Sequence.nextval, 'Misceao 2.0', 'Mountain', 'standard male', '26-Mar-12', 200.00, 5,
'CBH1001M20', '102');

INSERT INTO RR_Bikes

VALUES (Bike_Sequence.nextval, 'Misceao 2.0', 'Mountain', 'standard male', '26-Mar-12', 200.00, 5,
'GCH1001M21', '102');

INSERT INTO RR_Bikes

VALUES (Bike_Sequence.nextval, 'Big.Nine 300', 'Mountain', 'large male', '12-Apr-13', 250.00, 5,'', '101');

INSERT INTO RR_Bikes

VALUES (Bike_Sequence.nextval, 'Big.Nine 300', 'Mountain', 'large male', '12-Apr-13', 250.00, 5,'', '101');

INSERT INTO RR_Bikes

VALUES (Bike_Sequence.nextval, 'Big.Nine 300', 'Mountain', 'large male', '12-Apr-13', 250.00, 5,'', '101');

INSERT INTO RR_Bikes

VALUES (Bike_Sequence.nextval, 'Big.Nine 300', 'Mountain', 'standard male', '12-Apr-13', 250.00, 5,'', '101');

INSERT INTO RR_Bikes

VALUES (Bike_Sequence.nextval, 'Big.Nine 300', 'Mountain', 'large male', '12-Apr-13', 250.00, 5,'', '101');

INSERT INTO RR_Bikes

VALUES (Bike_Sequence.nextval, 'Dawes Duet Twin', 'Tandem', '', '10-May-13', 800.00, 8, '', '104');

INSERT INTO RR_Bikes

VALUES (Bike_Sequence.nextval, 'Dawes Duet Twin', 'Tandem', '', '10-May-13', 800.00, 8, '', '104');

INSERT INTO RR_Bikes

VALUES (Bike_Sequence.nextval, 'Dawes Duet Twin', 'Tandem', '', '10-May-13', 800.00, 8, '', '104');

INSERT INTO RR_Bikes

VALUES (Bike_Sequence.nextval, 'Dawes Duet Twin', 'Tandem', '', '10-May-13', 800.00, 8, '', '104');

INSERT INTO RR_Bikes

VALUES (Bike_Sequence.nextval, 'Dawes Duet Twin', 'Tandem', '', '10-May-13', 800.00, 8, '', '104');

INSERT INTO RR_Bikes
```

```
VALUES (Bike_Sequence.nextval, 'Reacto 400', 'Road', 'large male', '15-May-13', 300.00, 5,'', '101');
INSERT INTO RR_Bikes
VALUES (Bike_Sequence.nextval, 'Reacto 400', 'Road', 'large male', '15-May-13', 300.00, 5,'', '101');
INSERT INTO RR_Bikes
VALUES (Bike_Sequence.nextval, 'Reacto 400', 'Road', 'large male', '15-May-13', 300.00, 5,'', '101');
INSERT INTO RR_Bikes
VALUES (Bike_Sequence.nextval, 'Royal', 'Road', 'standard male', '10-Jun-13', 400.00, 6, '', '102');
INSERT INTO RR_Bikes
VALUES (Bike_Sequence.nextval, 'Royal', 'Road', 'standard male', '10-Jun-13', 400.00, 6, '', '102');
INSERT INTO RR_Bikes
VALUES (Bike_Sequence.nextval, 'Royal', 'Road', 'standard male', '10-Jun-13', 400.00, 6, '', '102');
INSERT INTO RR_Bikes
VALUES (Bike_Sequence.nextval, 'Sonnet Bliss', 'Road', 'standard female', '02-Aug-13', 500.00, 5, '', '105');
INSERT INTO RR_Bikes
VALUES (Bike_Sequence.nextval, 'Sonnet Bliss', 'Road', 'standard female', '02-Aug-13', 500.00, 5, '', '105');
INSERT INTO RR_Bikes
VALUES (Bike_Sequence.nextval, 'Sonnet Bliss', 'Road', 'standard female', '02-Aug-13', 500.00, 5, '', '105');
INSERT INTO RR_Bikes
VALUES (Bike_Sequence.nextval, 'Dawes Lightning Boys', 'Mountain', 'child', '08-Mar-14', 170.00, 3, '', '104');
INSERT INTO RR_Bikes
VALUES (Bike_Sequence.nextval, 'Dawes Lightning Boys', 'Mountain', 'child', '08-Mar-14', 170.00, 3, '', '104');
INSERT INTO RR_Bikes
VALUES (Bike_Sequence.nextval, 'Dawes Lightning Boys', 'Mountain', 'child', '08-Mar-14', 170.00, 3, '', '104');
INSERT INTO RR_Bikes
VALUES (Bike_Sequence.nextval, 'Dawes Venus Girls', 'Mountain', 'child', '08-Mar-14', 170.00, 3, '', '104');
INSERT INTO RR_Bikes
VALUES (Bike_Sequence.nextval, 'Dawes Venus Girls', 'Mountain', 'child', '08-Mar-14', 170.00, 3, '', '104');
INSERT INTO RR_Bikes
VALUES (Bike_Sequence.nextval, 'Dawes Venus Girls', 'Mountain', 'child', '08-Mar-14', 170.00, 3, '', '104');
INSERT INTO RR_Bikes
VALUES (Bike_Sequence.nextval, 'Misceao 2.0', 'Mountain', 'standard male', '26-Mar-14', 200.00, 5, '', '102');
```

```
INSERT INTO RR_Bikes
VALUES (Bike_Sequence.nextval, 'Misceao 2.0', 'Mountain', 'standard male', '26-Mar-14', 200.00, 5, '', '102');
INSERT INTO RR_Bikes
VALUES (Bike_Sequence.nextval, 'Misceao 2.0', 'Mountain', 'standard male', '26-Mar-14', 200.00, 5, '', '102');
INSERT INTO RR_Bikes
VALUES (Bike_Sequence.nextval, 'Misceao 2.0', 'Mountain', 'standard male', '26-Mar-14', 200.00, 5, '', '102');
INSERT INTO RR_Bikes
VALUES (Bike_Sequence.nextval, 'Misceao 2.0', 'Mountain', 'standard male', '26-Mar-14', 200.00, 5, '', '102');
INSERT INTO RR_Bikes
VALUES (Bike_Sequence.nextval, 'Core 10', 'Mountain', 'small male', '17-Apr-14', 550.00, 5, '', '106');
INSERT INTO RR_Bikes
VALUES (Bike_Sequence.nextval, 'Core 10', 'Mountain', 'small male', '17-Apr-14', 550.00, 5, '', '106');
INSERT INTO RR_Bikes
VALUES (Bike_Sequence.nextval, 'Core 10', 'Mountain', 'small male', '17-Apr-14', 550.00, 5, '', '106');
INSERT INTO RR_Bikes
VALUES (Bike_Sequence.nextval, 'Core 10', 'Mountain', 'small male', '17-Apr-14', 550.00, 5, '', '106');
INSERT INTO RR_Bikes
VALUES (Bike_Sequence.nextval, 'Core 10', 'Mountain', 'small male', '17-Apr-14', 550.00, 5, '', '106');
INSERT INTO RR_Bikes
VALUES (Bike_Sequence.nextval, 'Cypress W', 'Road', 'standard female', '05-May-14', 370.00, 5, '', '103');
INSERT INTO RR_Bikes
VALUES (Bike_Sequence.nextval, 'Cypress W', 'Road', 'standard female', '05-May-14', 370.00, 5, '', '103');
INSERT INTO RR_Bikes
VALUES (Bike_Sequence.nextval, 'Cypress W', 'Road', 'standard female', '05-May-14', 370.00, 5, '', '103');
INSERT INTO RR_Bikes
VALUES (Bike_Sequence.nextval, 'Cypress W', 'Road', 'standard female', '07-May-14', 370.00, 5, '', '103');
INSERT INTO RR_Bikes
VALUES (Bike_Sequence.nextval, 'Cypress', 'Road', 'small male', '05-Jul-14', 370.00, 5, '', '103');
INSERT INTO RR_Bikes
VALUES (Bike_Sequence.nextval, 'Cypress', 'Road', 'small male', '05-Jul-14', 370.00, 5, '', '103');
INSERT INTO RR_Bikes
```

```
VALUES (Bike_Sequence.nextval, 'Cypress', 'Road', 'small male', '05-Jul-14', 370.00, 5, '', '103');
INSERT INTO RR_Bikes
VALUES (Bike_Sequence.nextval, 'Cypress', 'Road', 'small male', '05-Jul-14', 370.00, 5, '', '103');
INSERT INTO RR_Bikes
VALUES (Bike_Sequence.nextval, 'Cypress', 'Road', 'small male', '05-Jul-14', 370.00, 5, '', '103');
```

```
INSERT INTO RR_SELL
VALUES (1,'BCH1001M4',78.50,'30-APR-13');
INSERT INTO RR_SELL
VALUES (2,'BCH1001M4',63.50,'30-APR-13');
INSERT INTO RR_SELL
VALUES (3,'GCH1001M21',48.50,'03-APR-13');
INSERT INTO RR_SELL
VALUES (4,'SC1001M16',38.50,'30-APR-13');
INSERT INTO RR_SELL
VALUES (5,'CBH1001M20',63.50,'12-APR-13');
INSERT INTO RR_SELL
VALUES (6,'BC1001SK5',48.50,'03-MAY-13');
INSERT INTO RR_SELL
VALUES (7,'KCH100M14',28.50,'30-MAY-13');
INSERT INTO RR_SELL
VALUES (8,'BCH1001M4',53.50,'13-JUN-13');
INSERT INTO RR_SELL
VALUES (9,'GCH1001M21',48.50,'21-JUN-14');
INSERT INTO RR_SELL
VALUES (10,'SC1001M16',36.00,'30-JUN-14');
INSERT INTO RR_SELL
```

```
VALUES (11,'CBH1001M20',43.50,'30-JUN-14');



INSERT INTO RR_Staff

VALUES (10001, 'Ray', '12 Broadfield Road, Manchester, Manchester', 'M14 4WF', '0161 5564679', '12-Jun-99', 'Boss');

INSERT INTO RR_Staff

VALUES (10002, 'Pete', '22 Bransby Avenue, Manchester, Manchester', 'M9 6JN', '0161 469733', '17-Jun-03', 'Hirings');

INSERT INTO RR_Staff

VALUES (10003, 'Sheila', '12 Wharfside Avenue, Eccles, Manchester', 'M30 0BW', '0161 118524','24-May-05', 'Hirings');

INSERT INTO RR_Staff

VALUES (10004, 'Megan', '16 Caldy Road, Salford, Salford', 'M6 7FU', '0161 876412', '18-Jul-13', 'Hirings');

INSERT INTO RR_Staff

VALUES (10005, 'Alf', '89 Oscar Street, Manchester, Manchester', 'M40 9EG', '0161 895623', '15-Aug-01', 'Maintenence');

INSERT INTO RR_Staff

VALUES (10006, 'Bert', '56 Egerton Street, Prestwich, Manchester', 'M25 1FH', '0161 784512', '21-Apr-05', 'Maintenence');

INSERT INTO RR_Staff

VALUES (10007, 'Paul', '214 Wilbraham Road, Manchester, Manchester', 'M16, UK', '0161 794613', '15-Mar-05', 'Parts');



INSERT INTO RR_PARTS(PartId,PartName,PartDescription,PartCost,StockLevel,ReOrderLevel,UnitsOnOrder,SuppId)

VALUES (101,'Inner Tubes','FWE 700c Presta Valve Inner Tube',4.99,20,5,10,101);

INSERT INTO RR_PARTS(PartId,PartName,PartDescription,PartCost,StockLevel,ReOrderLevel,UnitsOnOrder,SuppId)

VALUES (102,'Tyres','Continental Gatorskin 700C Duraskin Wired Road Tyre',29.95,10,4,8,101);

INSERT INTO RR_PARTS(PartId,PartName,PartDescription,PartCost,StockLevel,ReOrderLevel,UnitsOnOrder,ManId)
```

```
VALUES (103,'Wheels','Shimano Ultegra 6800 Tubeless Ready Wheelset',263.20,20,5,10,'101');

INSERT INTO
RR_PARTS(PartId,PartName,PartDescription,PartCost,StockLevel,ReOrderLevel,UnitsOnOrder,SuppId)

VALUES (104,'Wheels','Mavic Aksium One Disc 700C Front Road Wheel',67.50,10,5,10,101);

INSERT INTO
RR_PARTS(PartId,PartName,PartDescription,PartCost,StockLevel,ReOrderLevel,UnitsOnOrder,SuppId)

VALUES (105,'Inner Tubes','FWE 700c Presta Valve Inner Tube',4.99,20,5,10,101);

INSERT INTO
RR_PARTS(PartId,PartName,PartDescription,PartCost,StockLevel,ReOrderLevel,UnitsOnOrder,SuppId)

VALUES (106,'Pedals ','Shimano R540 SPD SL Road Pedals (OE)',21.99,20,5,10,101);

INSERT INTO
RR_PARTS(PartId,PartName,PartDescription,PartCost,StockLevel,ReOrderLevel,UnitsOnOrder,SuppId)

VALUES (107,'Pedals','Look Keo Grip Cleats OE',11.99,20,5,10,101);

INSERT INTO
RR_PARTS(PartId,PartName,PartDescription,PartCost,StockLevel,ReOrderLevel,UnitsOnOrder,ManId)

VALUES (108,'Cassettes + Freewheels','Shimano Ultegra 6700 10-speed Cassette',38.39,30,5,10,'101');

INSERT INTO
RR_PARTS(PartId,PartName,PartDescription,PartCost,StockLevel,ReOrderLevel,UnitsOnOrder,ManId)

VALUES (109,'Cassettes + Freewheels','Campagnolo Veloce 10spd Cassette',33.5,3,1,10,'101');

INSERT INTO
RR_PARTS(PartId,PartName,PartDescription,PartCost,StockLevel,ReOrderLevel,UnitsOnOrder,ManId)

VALUES (110,'Chains','Shimano Dura-Ace 7900 10 Speed Chain',34.99,5,3,1,'102');

INSERT INTO
RR_PARTS(PartId,PartName,PartDescription,PartCost,StockLevel,ReOrderLevel,UnitsOnOrder,ManId)

VALUES (111,'Chains','KMC X11L 11-speed Gold Chain',35.99,20,5,10,'101');

INSERT INTO
RR_PARTS(PartId,PartName,PartDescription,PartCost,StockLevel,ReOrderLevel,UnitsOnOrder,ManId)

VALUES (112,'Chainrings','Shimano FC-M590 Deore 4-Arm Chainring',11.49,10,4,3,'103');

INSERT INTO
RR_PARTS(PartId,PartName,PartDescription,PartCost,StockLevel,ReOrderLevel,UnitsOnOrder,SuppId)

VALUES (113,'Saddles','Fizik Vesta Kium Women''s Saddle',62.99,20,5,10,104);

INSERT INTO
RR_PARTS(PartId,PartName,PartDescription,PartCost,StockLevel,ReOrderLevel,UnitsOnOrder,SuppId)

VALUES (114,'Saddles','Adamo by ISM Prologue Saddle',103.99,20,5,10,103);
```

```
INSERT INTO
RR_PARTS(PartId,PartName,PartDescription,PartCost,StockLevel,ReOrderLevel,UnitsOnOrder,SuppId)

VALUES (115,'Brakes','SRAM Force Brake Set',84.90,5,2,1,105);

INSERT INTO
RR_PARTS(PartId,PartName,PartDescription,PartCost,StockLevel,ReOrderLevel,UnitsOnOrder,ManId)

VALUES (116,'Brakes','Jagwire V-Brake Blocks',2.15,30,5,10,'101');

INSERT INTO
RR_PARTS(PartId,PartName,PartDescription,PartCost,StockLevel,ReOrderLevel,UnitsOnOrder,ManId)

VALUES (117,'Cables','Clarks Pre-Lube Universal Brake Inner Cable - 2100mm',6.49,3,1,10,'101');

INSERT INTO
RR_PARTS(PartId,PartName,PartDescription,PartCost,StockLevel,ReOrderLevel,UnitsOnOrder,ManId)

VALUES (118,'Cables','FWE Campagnolo Inner Brake Cable',4.99,5,3,1,'102');

INSERT INTO
RR_PARTS(PartId,PartName,PartDescription,PartCost,StockLevel,ReOrderLevel,UnitsOnOrder,SuppId)

VALUES (119,'Rims','Stans No Tubes ZTR Alpha 400 Disc Rim',81.00,5,2,1,'102');

INSERT INTO
RR_PARTS(PartId,PartName,PartDescription,PartCost,StockLevel,ReOrderLevel,UnitsOnOrder,ManId)

VALUES (120,'Brakes','DT Swiss TK 540 700c Touring Rim',38.39,30,5,10,'101');

INSERT INTO
RR_PARTS(PartId,PartName,PartDescription,PartCost,StockLevel,ReOrderLevel,UnitsOnOrder,ManId)

VALUES (121,'Groupsets ','SRAM Force 22 Groupset',299.49,3,1,10,'101');

INSERT INTO
RR_PARTS(PartId,PartName,PartDescription,PartCost,StockLevel,ReOrderLevel,UnitsOnOrder,ManId)

VALUES (122,'Groupsets ','SRAM Red 22 Groupset',678.99,5,3,1,'102');

INSERT INTO
RR_PARTS(PartId,PartName,PartDescription,PartCost,StockLevel,ReOrderLevel,UnitsOnOrder,ManId)

VALUES (123,'Headsets ','FSA Orbit CE Headset',23.99,5,3,1,'102');

INSERT INTO
RR_PARTS(PartId,PartName,PartDescription,PartCost,StockLevel,ReOrderLevel,UnitsOnOrder,ManId)

VALUES (124,'Gear Cable','Jagwire gear cable',1.25,5,3,1,'102');



CREATE SEQUENCE PARTSORDERED_SEQUENCE

START WITH 1 MINVALUE 1 MAXVALUE 10000000 INCREMENT BY 1;
```

```
INSERT INTO RR_PARTSORDERED
VALUES(PARTSORDERED_SEQUENCE.NEXTVAL,'06-JUN-14', '10-JUN-14',103,'104');
INSERT INTO RR_PARTSORDERED
VALUES(PARTSORDERED_SEQUENCE.NEXTVAL,'01-JUL-14', '10-JUL-14',102,'103');
INSERT INTO RR_PARTSORDERED
VALUES(PARTSORDERED_SEQUENCE.NEXTVAL,'03-AUG-14', '05-AUG-14',101,'101');
INSERT INTO RR_PARTSORDERED
VALUES(PARTSORDERED_SEQUENCE.NEXTVAL,'07-SEP-14', '09-SEP-14',105,'105');
INSERT INTO RR_PARTSORDERED
VALUES(PARTSORDERED_SEQUENCE.NEXTVAL,'11-SEP-14', '16-SEP-14',106,'102');
INSERT INTO RR_PARTSORDERED
VALUES(PARTSORDERED_SEQUENCE.NEXTVAL,'12-OCT-14', '15-OCT-14',105,'104');
INSERT INTO RR_PARTSORDERED
VALUES(PARTSORDERED_SEQUENCE.NEXTVAL,'13-DEC-14', '20-DEC-14',102,'106');
INSERT INTO RR_PARTSORDERED
VALUES(PARTSORDERED_SEQUENCE.NEXTVAL,'30-DEC-14', '04-JAN-15',106,'105');
INSERT INTO RR_PARTSORDERED
VALUES(PARTSORDERED_SEQUENCE.NEXTVAL,'06-JAN-15', '11-JAN-15',103,'103');
INSERT INTO RR_PARTSORDERED
VALUES(PARTSORDERED_SEQUENCE.NEXTVAL,'25-JAN-15', '30-JAN-15',102,'106');



INSERT INTO RR_ORDERDETAILS
VALUES(1, 101 , 2, 2,  '9-JUN-14' ,'D687' , 'ALL PARTS RECIEVED');
INSERT INTO RR_ORDERDETAILS
VALUES(2, 103, 7, 6,  '11-JUL-14' ,'D234' , 'ONE WHEEL HAS VALVE MISSING');
INSERT INTO RR_ORDERDETAILS
VALUES(3, 104, 1, 1,  '05-AUG-14' ,'D543' , 'ALL PARTS RECIEVED');
INSERT INTO RR_ORDERDETAILS
```

```
VALUES(4, 102, 19, 19,  '09-SEP-14' ,'D111' , 'ALL PARTS RECIEVED');
INSERT INTO RR_ORDERDETAILS
VALUES(5, 105, 10, 10,  '18-SEP-14' ,'D233' , 'SENT 9 60MM VALVES AND 1 48MM VALVES');
INSERT INTO RR_ORDERDETAILS
VALUES(6, 106, 12, 11,  '14-OCT-14' ,'D656' , 'ONE SET OF PEDALS SHORT');
INSERT INTO RR_ORDERDETAILS
VALUES(7, 109, 3, 3,  '21-DEC-14' ,'D645' , 'ALL PARTS RECIEVED');
INSERT INTO RR_ORDERDETAILS
VALUES(8, 107, 9, 9,  '3-JAN-15' ,'D666' , 'ALL PARTS RECIEVED');
INSERT INTO RR_ORDERDETAILS
VALUES(9, 110, 4, 4,  '12-JAN-15' ,'D976' , 'ALL PARTS RECIEVED');
INSERT INTO RR_ORDERDETAILS
VALUES(10, 111, 12, 12,  '31-JAN-15' ,'D956' , 'ALL PARTS RECIEVED');
/*INSERT INTO RR_ORDERDETAILS
VALUES(11, 112, 5, 5,  '12-FEB-13' ,'D999' , 'ALL PARTS RECIEVED');
INSERT INTO RR_ORDERDETAILS
VALUES(12, 114, 7, 7,  '21-FEB-13' ,'D456' , 'ONE ITEM BROKEN');
INSERT INTO RR_ORDERDETAILS
VALUES(13, 115, 6, 6,  '26-FEB-13' ,'D343' , 'ALL PARTS RECIEVED');
INSERT INTO RR_ORDERDETAILS
VALUES(14, 108, 4, 5,  '12-OCT-14' ,'D933' , 'RECIEVED EXTRA ITEM');
INSERT INTO RR_ORDERDETAILS
VALUES(15, 116, 4, 4,  '12-DEC-14' ,'D923' , 'ALL PARTS RECIEVED');
INSERT INTO RR_ORDERDETAILS
VALUES(16, 118, 1, 1,  '26-DEC-14' ,'D936' , 'ALL PARTS RECIEVED');
INSERT INTO RR_ORDERDETAILS
VALUES(17, 119, 8, 8,  '30-DEC-14' ,'D977' , 'ALL PARTS RECIEVED');
INSERT INTO RR_ORDERDETAILS
VALUES(18, 120, 4, 4,  '12-JAN-15','D944' , 'ALL PARTS RECIEVED');
INSERT INTO RR_ORDERDETAILS
VALUES(19, 121, 4, 4, '22-JAN-15' ,'D346' , 'ALL PARTS RECIEVED');
```

```
INSERT INTO RR_ORDERDETAILS
VALUES(20, 122, 1, 1,  '26-JAN-15','D976' , 'ALL PARTS RECIEVED');*/



CREATE SEQUENCE Main_Sequence START WITH 1
INCREMENT BY 1
MINVALUE 1
MAXVALUE 1000000;

INSERT INTO RR_Maintenance
VALUES (Main_Sequence.nextval, 'Puncture repair', '15-Feb-14', '16-Feb-14', 'New inner tube', 1, 101, 10);
INSERT INTO RR_Maintenance(maintenanceid, bikefault, datereported, dateactioned, actiontaken, partquantity,
bikeid)
VALUES (Main_Sequence.nextval, 'Tyre repair', '18-Feb-14', '19-Feb-14', 'Pumped tyre', 2, 16);
INSERT INTO RR_Maintenance
VALUES (Main_Sequence.nextval, 'brake repair', '25-Feb-14', '26-Feb-14', 'New brakes', 1, 116, 14);
INSERT INTO RR_Maintenance
VALUES (Main_Sequence.nextval, 'Saddle repair', '01-Mar-14', '02-Mar-14', 'Fitted new saddle', 1, 113, 12);
INSERT INTO RR_Maintenance(maintenanceid, bikefault, datereported, dateactioned, actiontaken, partquantity,
bikeid)
VALUES (Main_Sequence.nextval, 'Chain repair', '15-Mar-14', '16-Mar-14', 'Corrected chain', 1, 21);
INSERT INTO RR_Maintenance
VALUES (Main_Sequence.nextval, 'Headset repair', '10-May-14', '12-May-14', 'New Headset', 1, 123, 15);
INSERT INTO RR_Maintenance
VALUES (Main_Sequence.nextval, 'Pedals repair', '15-May-14', '16-May-14', 'Fixed pedals',2, 106, 17);
INSERT INTO RR_Maintenance
VALUES (Main_Sequence.nextval, 'Framewheel repair', '20-June-14', '21-June-14', 'New framewheel', 1, 103,
35);
INSERT INTO RR_Maintenance
VALUES (Main_Sequence.nextval, 'Gear cable repair', '15-July-14', '16-July-14', 'New Gear cable', 1, 124, 11);
INSERT INTO RR_Maintenance
```

```
VALUES (Main_Sequence.nextval, 'Seat tube repair', '21-August-14', '22-August-14', 'New seat tube', 1, 114, 40);

INSERT INTO RR_Maintenance

VALUES (Main_Sequence.nextval, 'Headset repair', '30-August-14', '31-August-14', 'New Headset', 1, 123, 22);

INSERT INTO RR_Maintenance

VALUES (Main_Sequence.nextval, 'Framewheel repair', '31-August-14', '01-September-14', 'New framewheel', 1,
103, 9);



CREATE SEQUENCE RESERVATION_SEQUENCE

start with 1 minvalue 1 maxvalue 9999999999 nocycle increment by 1;


INSERT INTO RR_RESERVATION

VALUES(23,to_date('15-AUG-14 9:00:00','dd-mon-yy HH24:MI:SS'),to_date('15-AUG-14 13:00:00','dd-mon-yy
HH24:MI:SS'), to_date('15-AUG-14 12:45:30','dd-mon-yy
HH24:MI:SS'),1,'Y','Cash',reservation_sequence.nextval,10002,10); -- customer 1 to 19  staff 10002,3 and 4

INSERT INTO RR_RESERVATION

VALUES(24,to_date('15-AUG-14 9:00:00','dd-mon-yy HH24:MI:SS'),to_date('15-AUG-14 13:00:00','dd-mon-yy
HH24:MI:SS'), to_date('15-AUG-14 12:55:30','dd-mon-yy
HH24:MI:SS'),1,'Y','Cash',reservation_sequence.nextval,10002,1);

INSERT INTO RR_RESERVATION

VALUES(25,to_date('15-AUG-14 13:00:00','dd-mon-yy HH24:MI:SS'),to_date('15-AUG-14 17:00:00','dd-mon-yy
HH24:MI:SS'), to_date('15-AUG-14 16:45:30','dd-mon-yy
HH24:MI:SS'),1,'Y','Cash',reservation_sequence.nextval,10003,12);

INSERT INTO RR_RESERVATION

VALUES(33,to_date('16-AUG-14 9:00:00','dd-mon-yy HH24:MI:SS'),to_date('16-AUG-14 13:00:00','dd-mon-yy
HH24:MI:SS'), to_date('16-AUG-14 12:50:30','dd-mon-yy
HH24:MI:SS'),1,'Y','Cash',reservation_sequence.nextval,10003,4);

INSERT INTO RR_RESERVATION

VALUES(40,to_date('16-AUG-14 9:00:00','dd-mon-yy HH24:MI:SS'),to_date('16-AUG-14 17:00:00','dd-mon-yy
HH24:MI:SS'), to_date('16-AUG-14 16:45:30','dd-mon-yy
HH24:MI:SS'),2,'Y','Cash',reservation_sequence.nextval,10002,19);

INSERT INTO RR_RESERVATION

VALUES(37,to_date('17-AUG-14 9:00:00','dd-mon-yy HH24:MI:SS'),to_date('17-AUG-14 17:00:00','dd-mon-yy
HH24:MI:SS'), to_date('17-AUG-14 16:55:30','dd-mon-yy
HH24:MI:SS'),2,'Y','Cash',reservation_sequence.nextval,10004,15);

INSERT INTO RR_RESERVATION
```

```
VALUES(25,to_date('17-AUG-14 13:00:00','dd-mon-yy HH24:MI:SS'),to_date('17-AUG-14 17:00:00','dd-mon-yy
HH24:MI:SS'), to_date('17-AUG-14 16:50:30','dd-mon-yy
HH24:MI:SS'),1,'Y','Cash',reservation_sequence.nextval,10003,4);

INSERT INTO RR_RESERVATION

VALUES(27,to_date('17-AUG-14 13:00:00','dd-mon-yy HH24:MI:SS'),to_date('17-AUG-14 17:00:00','dd-mon-yy
HH24:MI:SS'), to_date('17-AUG-14 16:45:30','dd-mon-yy
HH24:MI:SS'),1,'Y','Cash',reservation_sequence.nextval,10003,7);

INSERT INTO RR_RESERVATION

VALUES(28,to_date('18-AUG-14 9:00:00','dd-mon-yy HH24:MI:SS'),to_date('18-AUG-14 13:00:00','dd-mon-yy
HH24:MI:SS'), to_date('18-AUG-14 12:45:30','dd-mon-yy
HH24:MI:SS'),1,'Y','Cash',reservation_sequence.nextval,10002,15);

INSERT INTO RR_RESERVATION

VALUES(38,to_date('18-AUG-14 9:00:00','dd-mon-yy HH24:MI:SS'),to_date('19-AUG-14 17:00:00','dd-mon-yy
HH24:MI:SS'), to_date('19-AUG-14 16:55:30','dd-mon-yy
HH24:MI:SS'),4,'Y','Cash',reservation_sequence.nextval,10004,18);

INSERT INTO RR_RESERVATION

VALUES(39,to_date('19-AUG-14 9:00:00','dd-mon-yy HH24:MI:SS'),to_date('19-AUG-14 13:00:00','dd-mon-yy
HH24:MI:SS'), to_date('19-AUG-14 12:45:30','dd-mon-yy
HH24:MI:SS'),1,'Y','Cash',reservation_sequence.nextval,10002,19);

INSERT INTO RR_RESERVATION

VALUES(32,to_date('19-AUG-14 9:00:00','dd-mon-yy HH24:MI:SS'),to_date('19-AUG-14 17:00:00','dd-mon-yy
HH24:MI:SS'), to_date('19-AUG-14 16:45:30','dd-mon-yy
HH24:MI:SS'),2,'Y','Cash',reservation_sequence.nextval,10002,16);

INSERT INTO RR_RESERVATION

VALUES(34,to_date('19-AUG-14 13:00:00','dd-mon-yy HH24:MI:SS'),to_date('20-AUG-14 17:00:00','dd-mon-yy
HH24:MI:SS'), to_date('20-AUG-14 16:55:30','dd-mon-yy
HH24:MI:SS'),3,'Y','Cash',reservation_sequence.nextval,10003,6);

INSERT INTO RR_RESERVATION

VALUES(37,to_date('20-AUG-14 9:00:00','dd-mon-yy HH24:MI:SS'),to_date('20-AUG-14 13:00:00','dd-mon-yy
HH24:MI:SS'), to_date('20-AUG-14 12:45:30','dd-mon-yy
HH24:MI:SS'),1,'Y','Cash',reservation_sequence.nextval,10003,8);

INSERT INTO RR_RESERVATION

VALUES(41,to_date('20-AUG-14 9:00:00','dd-mon-yy HH24:MI:SS'),to_date('20-AUG-14 17:00:00','dd-mon-yy
HH24:MI:SS'), to_date('20-AUG-14 16:50:30','dd-mon-yy
HH24:MI:SS'),2,'Y','Cash',reservation_sequence.nextval,10003,12);

INSERT INTO RR_RESERVATION
```

```
VALUES(42,to_date('20-AUG-14 13:00:00','dd-mon-yy HH24:MI:SS'),to_date('20-AUG-14 17:00:00','dd-mon-yy
HH24:MI:SS'), to_date('20-AUG-14 16:50:30','dd-mon-yy
HH24:MI:SS'),1,'Y','Cash',reservation_sequence.nextval,10004,11);

INSERT INTO RR_RESERVATION

VALUES(33,to_date('21-AUG-14 9:00:00','dd-mon-yy HH24:MI:SS'),to_date('21-AUG-14 13:00:00','dd-mon-yy
HH24:MI:SS'), to_date('21-AUG-14 12:50:30','dd-mon-yy
HH24:MI:SS'),1,'Y','Cash',reservation_sequence.nextval,10002,10);

INSERT INTO RR_RESERVATION

VALUES(36,to_date('21-AUG-14 9:00:00','dd-mon-yy HH24:MI:SS'),to_date('21-AUG-14 13:00:00','dd-mon-yy
HH24:MI:SS'), to_date('21-AUG-14 12:45:30','dd-mon-yy
HH24:MI:SS'),1,'Y','Cash',reservation_sequence.nextval,10002,9);

INSERT INTO RR_RESERVATION

VALUES(37,to_date('21-AUG-14 9:00:00','dd-mon-yy HH24:MI:SS'),to_date('21-AUG-14 13:00:00','dd-mon-yy
HH24:MI:SS'), to_date('21-AUG-14 12:55:30','dd-mon-yy
HH24:MI:SS'),1,'Y','Cash',reservation_sequence.nextval,10003,2);

INSERT INTO RR_RESERVATION

VALUES(40,to_date('21-AUG-14 9:00:00','dd-mon-yy HH24:MI:SS'),to_date('22-AUG-14 17:00:00','dd-mon-yy
HH24:MI:SS'), to_date('22-AUG-14 16:45:30','dd-mon-yy
HH24:MI:SS'),3,'Y','Cash',reservation_sequence.nextval,10003,3);

INSERT INTO RR_RESERVATION

VALUES(31,to_date('22-AUG-14 9:00:00','dd-mon-yy HH24:MI:SS'),to_date('22-AUG-14 17:00:00','dd-mon-yy
HH24:MI:SS'), to_date('22-AUG-14 16:45:30','dd-mon-yy
HH24:MI:SS'),2,'Y','Cash',reservation_sequence.nextval,10003,14);

INSERT INTO RR_RESERVATION

VALUES(24,to_date('22-AUG-14 13:00:00','dd-mon-yy HH24:MI:SS'),to_date('22-AUG-14 17:00:00','dd-mon-yy
HH24:MI:SS'), to_date('22-AUG-14 16:45:30','dd-mon-yy
HH24:MI:SS'),1,'Y','Cash',reservation_sequence.nextval,10003,3);

INSERT INTO RR_RESERVATION

VALUES(23,to_date('23-AUG-14 9:00:00','dd-mon-yy HH24:MI:SS'),to_date('23-AUG-14 13:00:00','dd-mon-yy
HH24:MI:SS'), to_date('23-AUG-14 12:55:30','dd-mon-yy
HH24:MI:SS'),1,'Y','Cash',reservation_sequence.nextval,10002,1);

INSERT INTO RR_RESERVATION

VALUES(36,to_date('24-AUG-14 9:00:00','dd-mon-yy HH24:MI:SS'),to_date('24-AUG-14 13:00:00','dd-mon-yy
HH24:MI:SS'), to_date('24-AUG-14 12:45:30','dd-mon-yy
HH24:MI:SS'),1,'Y','Cash',reservation_sequence.nextval,10002,9);

INSERT INTO RR_RESERVATION
```

```
VALUES(37,to_date('24-AUG-14 9:00:00','dd-mon-yy HH24:MI:SS'),to_date('24-AUG-14 13:00:00','dd-mon-yy
HH24:MI:SS'), to_date('24-AUG-14 12:55:30','dd-mon-yy
HH24:MI:SS'),1,'Y','Cash',reservation_sequence.nextval,10003,2);

INSERT INTO RR_RESERVATION

VALUES(40,to_date('24-AUG-14 9:00:00','dd-mon-yy HH24:MI:SS'),to_date('24-AUG-14 17:00:00','dd-mon-yy
HH24:MI:SS'), to_date('24-AUG-14 16:45:30','dd-mon-yy
HH24:MI:SS'),2,'Y','Cash',reservation_sequence.nextval,10003,3);

INSERT INTO RR_RESERVATION

VALUES(31,to_date('25-AUG-14 9:00:00','dd-mon-yy HH24:MI:SS'),to_date('25-AUG-14 17:00:00','dd-mon-yy
HH24:MI:SS'), to_date('25-AUG-14 16:45:30','dd-mon-yy
HH24:MI:SS'),2,'Y','Cash',reservation_sequence.nextval,10003,14);

INSERT INTO RR_RESERVATION

VALUES(24,to_date('25-AUG-14 13:00:00','dd-mon-yy HH24:MI:SS'),to_date('25-AUG-14 17:00:00','dd-mon-yy
HH24:MI:SS'), to_date('25-AUG-14 16:45:30','dd-mon-yy
HH24:MI:SS'),1,'Y','Cash',reservation_sequence.nextval,10003,3);

INSERT INTO RR_RESERVATION

VALUES(28,to_date('26-AUG-14 9:00:00','dd-mon-yy HH24:MI:SS'),to_date('26-AUG-14 13:00:00','dd-mon-yy
HH24:MI:SS'), to_date('26-AUG-14 12:55:30','dd-mon-yy
HH24:MI:SS'),1,'Y','Cash',reservation_sequence.nextval,10002,1);

INSERT INTO RR_RESERVATION

VALUES(36,to_date('26-AUG-14 9:00:00','dd-mon-yy HH24:MI:SS'),to_date('27-AUG-14 17:00:00','dd-mon-yy
HH24:MI:SS'), to_date('26-AUG-14 16:45:30','dd-mon-yy
HH24:MI:SS'),4,'Y','Cash',reservation_sequence.nextval,10002,9);

INSERT INTO RR_RESERVATION

VALUES(37,to_date('27-AUG-14 9:00:00','dd-mon-yy HH24:MI:SS'),to_date('27-AUG-14 13:00:00','dd-mon-yy
HH24:MI:SS'), to_date('27-AUG-14 12:55:30','dd-mon-yy
HH24:MI:SS'),1,'Y','Cash',reservation_sequence.nextval,10003,2);

INSERT INTO RR_RESERVATION

VALUES(40,to_date('27-AUG-14 9:00:00','dd-mon-yy HH24:MI:SS'),to_date('27-AUG-14 17:00:00','dd-mon-yy
HH24:MI:SS'), to_date('27-AUG-14 16:45:30','dd-mon-yy
HH24:MI:SS'),2,'Y','Cash',reservation_sequence.nextval,10003,3);

INSERT INTO RR_RESERVATION

VALUES(31,to_date('28-AUG-14 9:00:00','dd-mon-yy HH24:MI:SS'),to_date('28-AUG-14 17:00:00','dd-mon-yy
HH24:MI:SS'), to_date('28-AUG-14 16:45:30','dd-mon-yy
HH24:MI:SS'),2,'Y','Cash',reservation_sequence.nextval,10003,14);

INSERT INTO RR_RESERVATION
```

```
VALUES(22,to_date('28-AUG-14 13:00:00','dd-mon-yy HH24:MI:SS'),to_date('28-AUG-14 17:00:00','dd-mon-yy
HH24:MI:SS'), to_date('28-AUG-14 16:45:30','dd-mon-yy
HH24:MI:SS'),1,'Y','Cash',reservation_sequence.nextval,10003,3);

INSERT INTO RR_RESERVATION

VALUES(23,to_date('29-AUG-14 9:00:00','dd-mon-yy HH24:MI:SS'),to_date('29-AUG-14 13:00:00','dd-mon-yy
HH24:MI:SS'), to_date('29-AUG-14 12:55:30','dd-mon-yy
HH24:MI:SS'),1,'Y','Cash',reservation_sequence.nextval,10002,1);

INSERT INTO RR_RESERVATION

VALUES(15,to_date('30-AUG-14 9:00:00','dd-mon-yy HH24:MI:SS'),to_date('30-AUG-14 13:00:00','dd-mon-yy
HH24:MI:SS'), to_date('30-AUG-14 12:45:30','dd-mon-yy
HH24:MI:SS'),1,'Y','Cash',reservation_sequence.nextval,10002,9);

INSERT INTO RR_RESERVATION

VALUES(37,to_date('30-AUG-14 9:00:00','dd-mon-yy HH24:MI:SS'),to_date('31-AUG-14 17:00:00','dd-mon-yy
HH24:MI:SS'), to_date('31-AUG-14 16:55:30','dd-mon-yy
HH24:MI:SS'),4,'Y','Cash',reservation_sequence.nextval,10003,2);

INSERT INTO RR_RESERVATION

VALUES(40,to_date('31-AUG-14 9:00:00','dd-mon-yy HH24:MI:SS'),to_date('31-AUG-14 17:00:00','dd-mon-yy
HH24:MI:SS'), to_date('31-AUG-14 16:45:30','dd-mon-yy
HH24:MI:SS'),2,'Y','Cash',reservation_sequence.nextval,10003,3);

INSERT INTO RR_RESERVATION

VALUES(31,to_date('31-AUG-14 9:00:00','dd-mon-yy HH24:MI:SS'),to_date('31-AUG-14 17:00:00','dd-mon-yy
HH24:MI:SS'), to_date('31-AUG-14 16:45:30','dd-mon-yy
HH24:MI:SS'),2,'Y','Cash',reservation_sequence.nextval,10003,14);

INSERT INTO RR_RESERVATION

VALUES(24,to_date('31-AUG-14 13:00:00','dd-mon-yy HH24:MI:SS'),to_date('31-AUG-14 17:00:00','dd-mon-yy
HH24:MI:SS'), to_date('31-AUG-14 16:45:30','dd-mon-yy
HH24:MI:SS'),1,'Y','Cash',reservation_sequence.nextval,10003,3);

INSERT INTO RR_RESERVATION

VALUES(23,to_date('01-SEP-14 9:00:00','dd-mon-yy HH24:MI:SS'),to_date('01-SEP-14 13:00:00','dd-mon-yy
HH24:MI:SS'), to_date('01-SEP-14 12:55:30','dd-mon-yy
HH24:MI:SS'),1,'Y','Cash',reservation_sequence.nextval,10002,1);

INSERT INTO RR_RESERVATION

VALUES(36,to_date('02-SEP-14 9:00:00','dd-mon-yy HH24:MI:SS'),to_date('02-SEP-14 13:00:00','dd-mon-yy
HH24:MI:SS'), to_date('02-SEP-14 12:45:30','dd-mon-yy
HH24:MI:SS'),1,'Y','Cash',reservation_sequence.nextval,10002,9);

INSERT INTO RR_RESERVATION
```

```
VALUES(39,to_date('02-SEP-14 9:00:00','dd-mon-yy HH24:MI:SS'),to_date('03-SEP-14 17:00:00','dd-mon-yy
HH24:MI:SS'), to_date('02-SEP-14 12:55:30','dd-mon-yy
HH24:MI:SS'),4,'Y','Cash',reservation_sequence.nextval,10003,2);

INSERT INTO RR_RESERVATION

VALUES(40,to_date('03-SEP-14 9:00:00','dd-mon-yy HH24:MI:SS'),to_date('04-SEP-14 9:00:00','dd-mon-yy
HH24:MI:SS'), to_date('04-SEP-14 16:45:30','dd-mon-yy
HH24:MI:SS'),2,'Y','Cash',reservation_sequence.nextval,10003,3);

INSERT INTO RR_RESERVATION

VALUES(31,to_date('04-SEP-14 9:00:00','dd-mon-yy HH24:MI:SS'),to_date('03-SEP-14 17:00:00','dd-mon-yy
HH24:MI:SS'), to_date('03-SEP-14 16:45:30','dd-mon-yy
HH24:MI:SS'),2,'Y','Cash',reservation_sequence.nextval,10003,14);

INSERT INTO RR_RESERVATION

VALUES(25,to_date('04-SEP-14 13:00:00','dd-mon-yy HH24:MI:SS'),to_date('04-SEP-14 17:00:00','dd-mon-yy
HH24:MI:SS'), to_date('04-SEP-14 16:45:30','dd-mon-yy
HH24:MI:SS'),1,'Y','Cash',reservation_sequence.nextval,10003,3);

INSERT INTO RR_RESERVATION

VALUES(23,to_date('04-SEP-14 9:00:00','dd-mon-yy HH24:MI:SS'),to_date('04-SEP-14 13:00:00','dd-mon-yy
HH24:MI:SS'), to_date('04-SEP-14 12:55:30','dd-mon-yy
HH24:MI:SS'),1,'Y','Cash',reservation_sequence.nextval,10002,1);

INSERT INTO RR_RESERVATION

VALUES(18,to_date('04-SEP-14 9:00:00','dd-mon-yy HH24:MI:SS'),to_date('05-SEP-14 13:00:00','dd-mon-yy
HH24:MI:SS'), to_date('05-SEP-14 12:45:30','dd-mon-yy
HH24:MI:SS'),3,'Y','Cash',reservation_sequence.nextval,10002,9);

INSERT INTO RR_RESERVATION

VALUES(38,to_date('05-SEP-14 12:00:00','dd-mon-yy HH24:MI:SS'),to_date('05-SEP-14 14:00:00','dd-mon-yy
HH24:MI:SS'), to_date('05-SEP-14 13:55:30','dd-mon-yy
HH24:MI:SS'),1,'Y','Cash',reservation_sequence.nextval,10003,2);

INSERT INTO RR_RESERVATION

VALUES(40,to_date('05-SEP-14 15:00:00','dd-mon-yy HH24:MI:SS'),to_date('05-SEP-14 17:00:00','dd-mon-yy
HH24:MI:SS'), to_date('05-SEP-14 16:45:30','dd-mon-yy
HH24:MI:SS'),1,'Y','Cash',reservation_sequence.nextval,10003,3);

INSERT INTO RR_RESERVATION

VALUES(31,to_date('06-SEP-14 9:00:00','dd-mon-yy HH24:MI:SS'),to_date('06-SEP-14 17:00:00','dd-mon-yy
HH24:MI:SS'), to_date('06-SEP-14 16:45:30','dd-mon-yy
HH24:MI:SS'),2,'Y','Cash',reservation_sequence.nextval,10003,14);

INSERT INTO RR_RESERVATION
```

```
VALUES(24,to_date('06-SEP-14 13:00:00','dd-mon-yy HH24:MI:SS'),to_date('07-SEP-14 17:00:00','dd-mon-yy
HH24:MI:SS'), to_date('07-SEP-14 16:45:30','dd-mon-yy
HH24:MI:SS'),3,'N',null,reservation_sequence.nextval,10003,3);

INSERT INTO RR_RESERVATION

VALUES(40,to_date('07-SEP-14 9:00:00','dd-mon-yy HH24:MI:SS'),to_date('07-SEP-14 17:00:00','dd-mon-yy
HH24:MI:SS'), to_date('07-SEP-14 16:55:30','dd-mon-yy
HH24:MI:SS'),2,'N',null,reservation_sequence.nextval,10002,1);



CREATE SEQUENCE ENQUIRY_SEQUENCE

START WITH 1 MINVALUE 1 MAXVALUE 9999999999

NOCYCLE INCREMENT BY 1;



INSERT INTO RR_ENQUIRY VALUES(

Enquiry_sequence.nextval,To_date('02-FEB-15 9:10:00','DD-MON-YY HH24:MI:SS'),1,10002,1);  --customer from
1 to 19 staff 1002-4

INSERT INTO RR_ENQUIRY VALUES(

Enquiry_sequence.nextval,To_date('02-FEB-15 9:10:00','DD-MON-YY HH24:MI:SS'),1,10002,1);

INSERT INTO RR_ENQUIRY VALUES(

Enquiry_sequence.nextval,To_date('02-FEB-15 9:30:20','DD-MON-YY HH24:MI:SS'),2,10003,4);

INSERT INTO RR_ENQUIRY VALUES(

Enquiry_sequence.nextval,To_date('02-FEB-15 9:40:05','DD-MON-YY HH24:MI:SS'),2,10002,6);

INSERT INTO RR_ENQUIRY VALUES(

Enquiry_sequence.nextval,To_date('02-FEB-15 10:10:03','DD-MON-YY HH24:MI:SS'),1,10004,7);

INSERT INTO RR_ENQUIRY VALUES(

Enquiry_sequence.nextval,To_date('02-FEB-15 10:12:00','DD-MON-YY HH24:MI:SS'),2,10003,11);

INSERT INTO RR_ENQUIRY VALUES(

Enquiry_sequence.nextval,To_date('02-FEB-15 10:50:06','DD-MON-YY HH24:MI:SS'),2,10002,13);

INSERT INTO RR_ENQUIRY VALUES(

Enquiry_sequence.nextval,To_date('02-FEB-15 11:11:00','DD-MON-YY HH24:MI:SS'),2,10002,12);

INSERT INTO RR_ENQUIRY VALUES(
Enquiry_sequence.nextval,To_date('02-FEB-15 12:00:00','DD-MON-YY HH24:MI:SS'),2,10004,3);

INSERT INTO RR_ENQUIRY VALUES(

Enquiry_sequence.nextval,To_date('02-FEB-15 12:40:00','DD-MON-YY HH24:MI:SS'),1,10002,5);

INSERT INTO RR_ENQUIRY VALUES(

Enquiry_sequence.nextval,To_date('02-FEB-15 12:50:50','DD-MON-YY HH24:MI:SS'),1,10003,8);

INSERT INTO RR_ENQUIRY VALUES(

Enquiry_sequence.nextval,To_date('02-FEB-15 13:10:00','DD-MON-YY HH24:MI:SS'),1,10002,9);

INSERT INTO RR_ENQUIRY VALUES(

Enquiry_sequence.nextval,To_date('02-FEB-15 14:20:40','DD-MON-YY HH24:MI:SS'),2,10002,10);

INSERT INTO RR_ENQUIRY VALUES(

Enquiry_sequence.nextval,To_date('02-FEB-15 15:07:03','DD-MON-YY HH24:MI:SS'),2,10004,11);

INSERT INTO RR_ENQUIRY VALUES(

Enquiry_sequence.nextval,To_date('03-FEB-15 9:10:00','DD-MON-YY HH24:MI:SS'),2,10002,1);

INSERT INTO RR_ENQUIRY VALUES(

Enquiry_sequence.nextval,To_date('03-FEB-15 9:20:05','DD-MON-YY HH24:MI:SS'),1,10003,11);

INSERT INTO RR_ENQUIRY VALUES(

Enquiry_sequence.nextval,To_date('03-FEB-15 10:30:10','DD-MON-YY HH24:MI:SS'),1,10002,16);

INSERT INTO RR_ENQUIRY VALUES(

Enquiry_sequence.nextval,To_date('03-FEB-15 10:10:20','DD-MON-YY HH24:MI:SS'),2,10004,17);

INSERT INTO RR_ENQUIRY VALUES(

Enquiry_sequence.nextval,To_date('03-FEB-15 11:10:00','DD-MON-YY HH24:MI:SS'),2,10004,18);

INSERT INTO RR_ENQUIRY VALUES(

Enquiry_sequence.nextval,To_date('04-FEB-15 11:10:00','DD-MON-YY HH24:MI:SS'),2,10002,18);

INSERT INTO RR_ENQUIRY VALUES(

Enquiry_sequence.nextval,To_date('04-FEB-15 11:10:00','DD-MON-YY HH24:MI:SS'),1,10004,19);

INSERT INTO RR_ENQUIRY VALUES(

Enquiry_sequence.nextval,To_date('04-FEB-15 11:10:00','DD-MON-YY HH24:MI:SS'),2,10004,1);

INSERT INTO RR_ENQUIRY VALUES(

Enquiry_sequence.nextval,To_date('05-FEB-15 11:10:00','DD-MON-YY HH24:MI:SS'),1,10004,2);

INSERT INTO RR_ENQUIRY VALUES(

Enquiry_sequence.nextval,To_date('05-FEB-15 10:10:00','DD-MON-YY HH24:MI:SS'),2,10002,8);
```

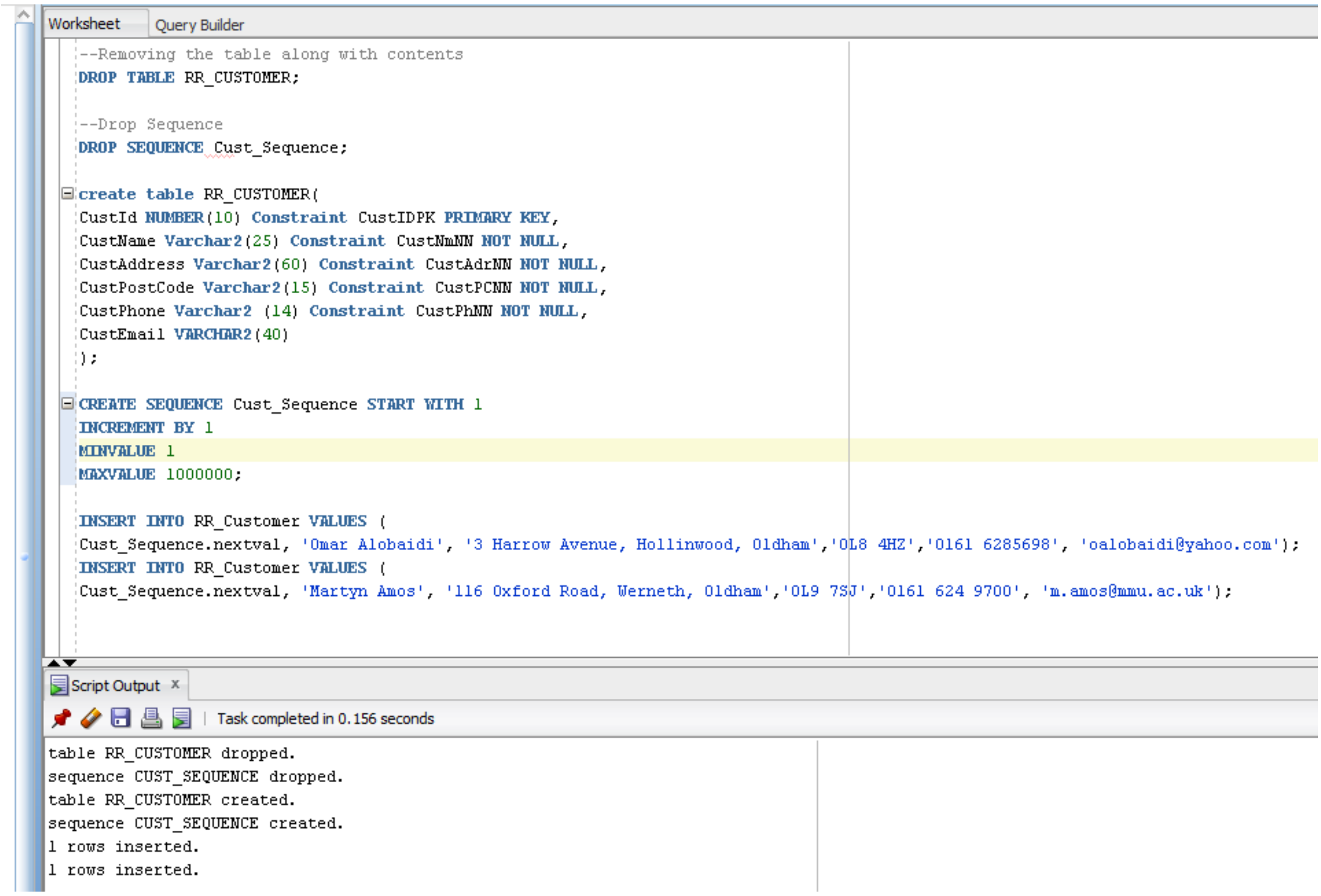
```
INSERT INTO RR_ENQUIRY VALUES(
Enquiry_sequence.nextval,To_date('06-FEB-15 11:10:00','DD-MON-YY HH24:MI:SS'),1,10004,7);
INSERT INTO RR_ENQUIRY VALUES(
Enquiry_sequence.nextval,To_date('06-FEB-15 12:10:00','DD-MON-YY HH24:MI:SS'),1,10004,8);
INSERT INTO RR_ENQUIRY VALUES(
Enquiry_sequence.nextval,To_date('06-FEB-15 13:10:00','DD-MON-YY HH24:MI:SS'),2,10002,9);
INSERT INTO RR_ENQUIRY VALUES(
Enquiry_sequence.nextval,To_date('06-FEB-15 14:10:00','DD-MON-YY HH24:MI:SS'),2,10004,10);
INSERT INTO RR_ENQUIRY VALUES(
Enquiry_sequence.nextval,To_date('07-FEB-15 09:10:00','DD-MON-YY HH24:MI:SS'),2,10004,11);
INSERT INTO RR_ENQUIRY VALUES(
Enquiry_sequence.nextval,To_date('07-FEB-15 09:10:00','DD-MON-YY HH24:MI:SS'),2,10002,12);
INSERT INTO RR_ENQUIRY VALUES(
Enquiry_sequence.nextval,To_date('08-FEB-15 10:10:00','DD-MON-YY HH24:MI:SS'),2,10004,13);
INSERT INTO RR_ENQUIRY VALUES(
Enquiry_sequence.nextval,To_date('09-FEB-15 10:10:00','DD-MON-YY HH24:MI:SS'),2,10002,14);
INSERT INTO RR_ENQUIRY VALUES(
Enquiry_sequence.nextval,To_date('10-FEB-15 10:10:00','DD-MON-YY HH24:MI:SS'),2,10004,15);
INSERT INTO RR_ENQUIRY VALUES(
Enquiry_sequence.nextval,To_date('11-FEB-15 10:10:00','DD-MON-YY HH24:MI:SS'),2,10002,16);
INSERT INTO RR_ENQUIRY VALUES(
Enquiry_sequence.nextval,To_date('12-FEB-15 11:10:00','DD-MON-YY HH24:MI:SS'),2,10004,19);

--Queries and Management Reports

--Maryam Elgahmi 12009935--
--Shows bike faults by manufacturer (natural join 2 tables, order by, column alias)
select bikeid, bikemodel"Model", bikedop"Date of Purchase", bikefault"Fault", manname"Manufacturer"
from rr_bikes natural join rr_maintenance natural join rr_manufacturer
order by manid, bikefault;
```

```
--Shows all reservations by particular customer (natural join 3 tables, column alias, arithmetic expression, comparison operator)

select custname"Customer Name", bikemodel"Bike Model", bikesize"Size", bikeclassification"Classification",

bikerentcost"Rental cost per half day", rentdateandtime"Rental date", rentperiod"# of half days", bikerentcost*rentperiod "Total Rent Cost"

from rr_bikes natural join rr_customer natural join rr_reservation

where custid = 14;


--Shows Total rental income for the selected week (functions, column alias, natural join 2 tables, logical operator)

select sum(rentperiod*bikerentcost) as "Income for w/e 22 August 2014"

from rr_reservation natural join rr_bikes

where rentdateandtime between '15-Aug-14' and '22-Aug-14';


--Shows popularity of each bike model (functions, natural join 2 tables, group by, order by, column alias)

select distinct(bikemodel)"Bike Model", count(bikemodel)"# of reservations"

from rr_bikes natural join rr_reservation

group by bikemodel

order by count(bikemodel) desc;


--Total spent on each classification in 2013 (Column alias, logical operator, function, group by)

SELECT bikeclassification"Classification", sum(bikecost)"Total spent in 2013"

FROM rr_bikes

WHERE bikedop between '1-jan-13' and '31-dec-13'

group by  bikeclassification;


--Display all the departments name and the count of departments if the department names count is more than Hiring Department

--(Group by, function, comparison operator,sub query, Having clause)

select deptname,count(deptname) from rr_staff

group by deptname

having count(deptname)>=
(select count(deptname) from rr_staff where
deptname='Hirings');

--Display all the age of Bikes in years if they have purchased on or before 31 december 2012
select bikeid,bikeclassification,bikemodel,bikedop,trunc(months_between(sysdate,bikedop)/12,0)
"Bike Age"
from RR_bikes
where bikedop<= '01-Jan-13';

--Mark Bellingham--14032098
--Check orders and order details (Column alias, equi join 3 tables, comparison operator, order by)
select orderdate"Order Date", deliveryexpecteddate"Expected Delivery Date", orderreceiveddate"Date Received",
ORDERQUANTITY"Quantity", receivedquantity"Received Quantity", partname"Part Name", partdescription"Description", comments"Comments"
from rr_partsordered join rr_orderdetails on rr_partsordered.orderno = rr_orderdetails.orderno
join rr_parts on rr_orderdetails.partid = rr_parts.partid
order by orderdate;

--Bikes which have not been sold after 2 years (Column alias, Comparison operator, left outer join, months between, sysdate)
select rr_bikes.bikeid"Bike ID", bikemodel"Bike Model", manid"Manufacturer ID", bikedop"Bike Date of Purchase"
from rr_bikes left outer join rr_sell on rr_bikes.bikeid = rr_sell.bikeid
where months_between(sysdate,bikedop)>24 and rr_sell.bikeid is null;

--Shows income by customer for the second half of 2014 (equi join to join 3 tables, column alias, function, arithmetic operator, table alias, logical operator, group by, order by)
select custname" Customer Name",custaddress" Customer Address",custpostcode" Customer PostCode",custphone" Customer Telephone",custemail" Customer Email Id",
rentpaid" Paid Y/N " ,sum(bikerentcost*rentperiod) "Total Invoice Amount",paymenttype
from RR_customer c,RR_reservation r,RR_bikes b
where c.custid=r.custid and b.bikeid=r.bikeid and rentdateandtime between '1-jul-14' and '31-dec-14'
```

```
group by custname, custaddress, custpostcode, custphone, custemail, rentpaid, paymenttype
order by custname;


--Display staff details who works in the same department where Bert and Pete works(Subquery, order by, logical operator, comparison operator)
select staffname,deptname from RR_staff where deptname=
(select deptname from RR_staff where staffname='Pete')
or deptname like
(select deptname from RR_staff where staffname='Bert')
order by staffname,deptname;


--Find bike rentals created by a particular member of staff (Join four tables, column alias, order by, comparison operator, arithmetic operator)
SELECT bikemodel"Bike Model", bikeclassification"Bike Classification", bikesize"Bike Size", custname"Customer Name",
rentdateandtime"Rental Date", bikerentcost*rentperiod"Rental Cost", staffname"Staff Name"
FROM rr_bikes b, rr_reservation r, rr_customer c, rr_staff s
WHERE b.bikeid = r.bikeid and c.custid = r.custid and s.staffid = r.staffid and s.staffid = 10002
ORDER BY rentdateandtime;




--Janet D'Souza--14059185
--Displays customers who have rented bikes where the cost of the bike is more than average (comparison operator, function, logical operator, subquery, table alias, equi join 3 tables)
select custname, bikerentcost, bikecost, b.bikeid, bikeclassification, bikemodel
from rr_customer c, rr_bikes b, rr_reservation r
where b.bikeid = r.bikeid and c.custid = r.custid and bikecost>(select avg(bikecost)
from rr_bikes );
```

```
--Display how much total money been spent to purchase all Mountain bikes in the year 2014 (Column alias, functions (sum and count),  logical operator, group by)

select bikeclassification, count(bikeclassification)"Quantity Purchased",sum(bikecost)"Total Amount Spent "

from RR_bikes

where bikeclassification like 'Mountain' and bikedop between '1-jan-14' and '31-dec-14'

group by bikeclassification;


--Shows the most expensive bike by type where cost is greater then Â£500 (Column alias, group by, comparison operator, function, having clause)

select bikemodel"Bikes costing more than £500", bikeclassification"Bike Classification", max(bikecost)"Highest Price"

from rr_bikes

group by bikeclassification, bikemodel

having max(bikecost)>500;


--Display the exact age of the bike in years months and days(concatenation operator, arithmatic operator, functions, sysdate)

select 'Bike bought on '||to_char(bikedop,'FMMonth DD YYYY') || ' and the bike is '|| trunc(months_between(sysdate,bikedop)/12) || 'year(s) '

||trunc(mod((months_between(sysdate,bikedop)),12))||' month(s) ' ||(trunc(sysdate)-add_months(bikedop,

(months_between(sysdate,bikedop))))||' day(s) old.'"Age of the Bike" from RR_bikes;


--Invoice details of all the customers who hired and not paid for the bikes (Equi Join to join 3 tables, Column alias, table alias, to_char, decode, logical operator, order by)

select custname" Customer Name",custaddress" Customer Address",custpostcode" Customer PostCode",custphone" Customer Telephone",custemail" Customer Email Id",

to_char(Rentdateandtime,'DD/MM/YYYY')"Rent Date", to_char(rentdateandtime,'hh:mi:ss AM')"Rent Time",

decode(rentperiod, 1,'Half Day', 2,'Full Day', 3,'Day and half', 4,'Two days')"Rent Period",

rentpaid" Paid Y/N " ,bikerentcost*rentperiod "Invoice Amount",paymenttype

from RR_customer c,RR_reservation r,RR_bikes b

where c.custid=r.custid and b.bikeid=r.bikeid and rentpaid like 'N'

order by custname, rentdateandtime;

--End of Queries
```

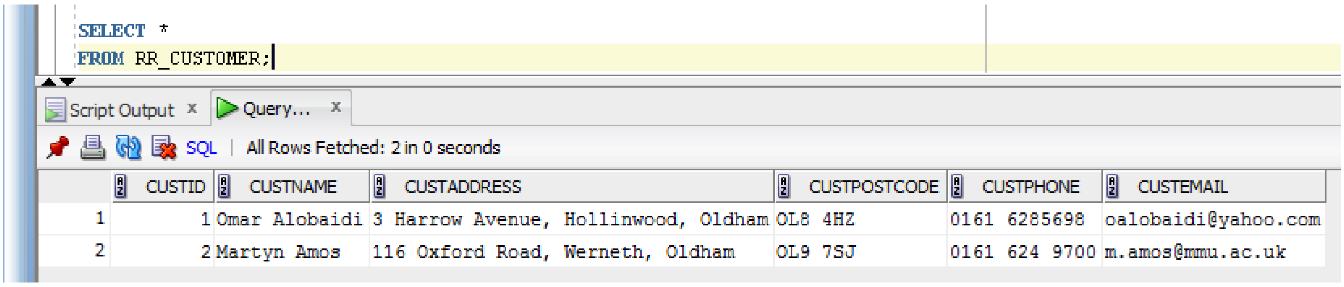
Output produced when script is run



```
--Removing the table along with contents
DROP TABLE RR_CUSTOMER;

--Drop Sequence
DROP SEQUENCE Cust_Sequence;

create table RR_CUSTOMER(
CustId NUMBER(10) Constraint CustIDPK PRIMARY KEY,
CustName Varchar2(25) Constraint CustNmNN NOT NULL,
CustAddress Varchar2(60) Constraint CustAdrNN NOT NULL,
CustPostCode Varchar2(15) Constraint CustPCNN NOT NULL,
CustPhone Varchar2 (14) Constraint CustPhNN NOT NULL,
CustEmail VARCHAR2(40)
);

CREATE SEQUENCE Cust_Sequence START WITH 1
INCREMENT BY 1
MINVALUE 1
MAXVALUE 1000000;

INSERT INTO RR_Customer VALUES (
Cust_Sequence.nextval, 'Omar Alobaidi', '3 Harrow Avenue, Hollinwood, Oldham','OL8 4HZ','0161 6285698', 'oalobaidi@yahoo.com');
INSERT INTO RR_Customer VALUES (
Cust_Sequence.nextval, 'Martyn Amos', '116 Oxford Road, Werneth, Oldham','OL9 7SJ','0161 624 9700', 'm.amos@mmu.ac.uk');
```

Task completed in 0.156 seconds

```
table RR_CUSTOMER dropped.
sequence CUST_SEQUENCE dropped.
table RR_CUSTOMER created.
sequence CUST_SEQUENCE created.
1 rows inserted.
1 rows inserted.
```



```
SELECT *
FROM RR_CUSTOMER;
```

All Rows Fetched: 2 in 0 seconds

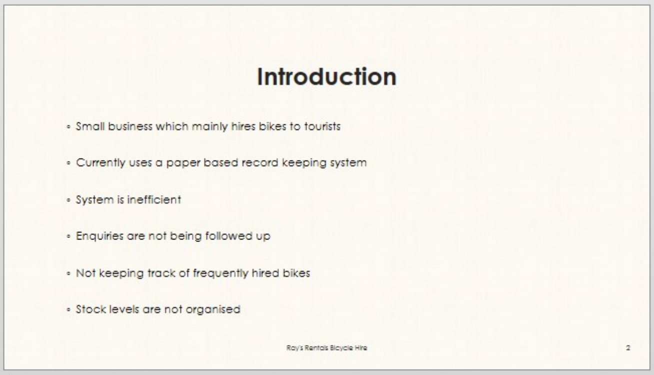
	CUSTID	CUSTNAME	CUSTADDRESS	CUSTPOSTCODE	CUSTPHONE	CUSTEMAIL
1	1	Omar Alobaidi	3 Harrow Avenue, Hollinwood, Oldham	OL8 4HZ	0161 6285698	oalobaidi@yahoo.com
2	2	Martyn Amos	116 Oxford Road, Werneth, Oldham	OL9 7SJ	0161 624 9700	m.amos@mmu.ac.uk

Presentation

RAY'S RENTALS BICYCLE HIRE

Mark Bellingham, Maryam El-gahmi and Janet D'Souza

1

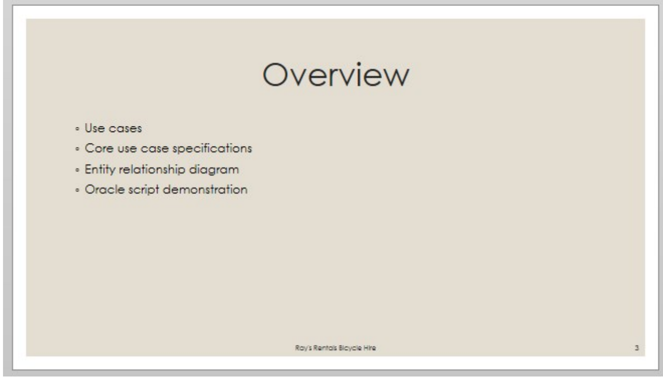


Introduction

- Small business which mainly hires bikes to tourists
- Currently uses a paper based record keeping system
- System is inefficient
- Enquiries are not being followed up
- Not keeping track of frequently hired bikes
- Stock levels are not organised

Rays Rentals Bicycle Hire 2

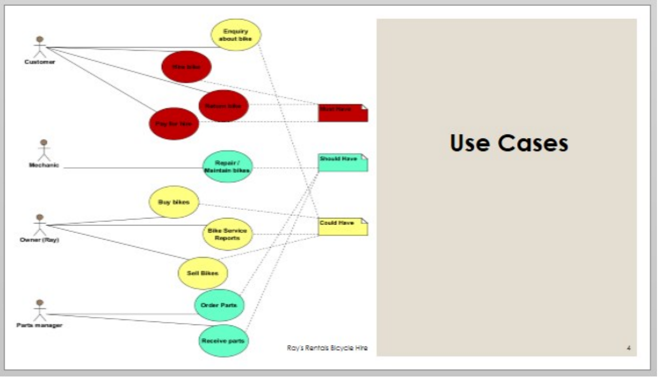
2



Overview

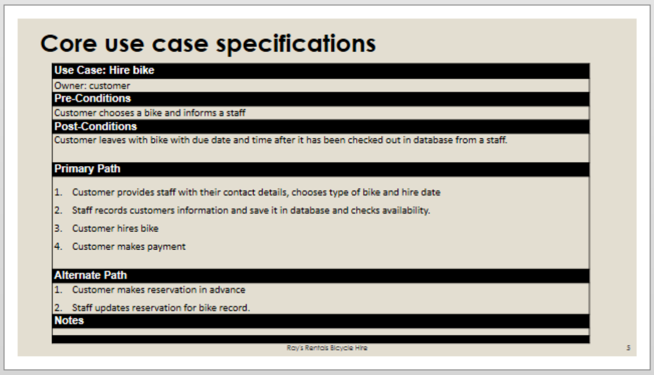
- Use cases
- Core use case specifications
- Entity relationship diagram
- Oracle script demonstration

Rays Rentals Bicycle Hire 3



Use Cases

Rays Rentals Bicycle Hire 4



Core use case specifications

Use Case: Hire bike

Owner: customer

Pre-Conditions

Customer chooses a bike and informs a staff

Post-Conditions

Customer leaves with bike with due date and time after it has been checked out in database from a staff.

Primary Path

1. Customer provides staff with their contact details, chooses type of bike and hire date
2. Staff records customers information and save it in database and checks availability.
3. Customer hires bike
4. Customer makes payment

Alternate Path

1. Customer makes reservation in advance
2. Staff updates reservation for bike record.

Notes

Rays Rentals Bicycle Hire 5

5

Use Case: Bike maintenance

Owner: Ray

Pre-Conditions

Find out which bikes need servicing

Post-Conditions

Details of which bikes need servicing have been passed to the maintenance department

Primary Path

1.Check bike records

2.Create report where last service date is more than one month ago

3.List of bikes is passed to the maintenance department

Alternate Path

1.Customer complains about bike fault

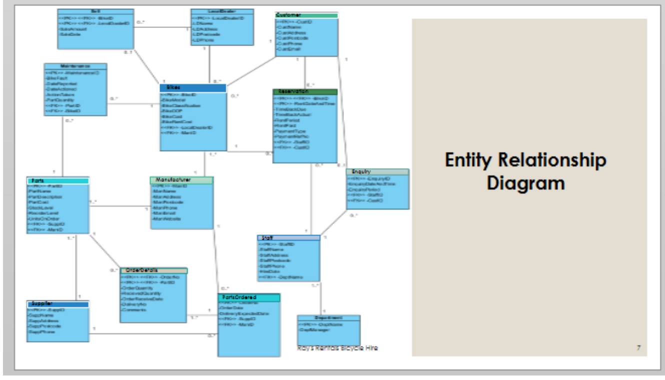
2.Reception updates a list of bikes with faults

3.List of bikes is passed to the maintenance department

Notes

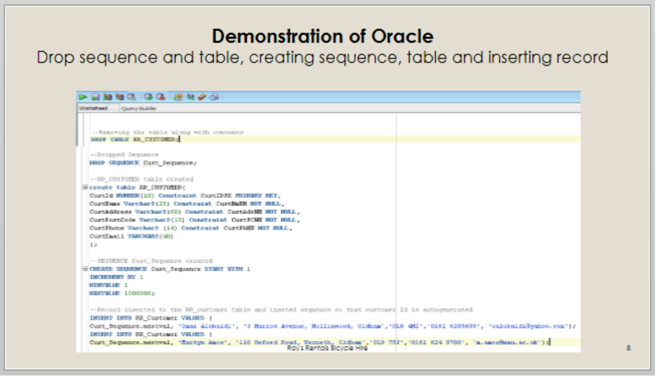
Rays Rentals Bicycle Hire 6

6



Entity Relationship Diagram

Rays Rentals Bicycle Hire 7

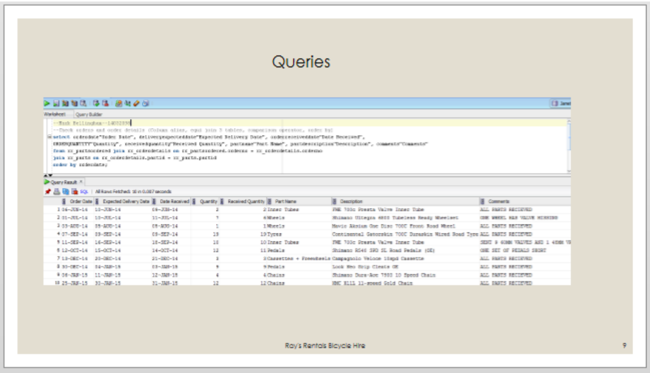


Demonstration of Oracle

Drop sequence and table, creating sequence, table and inserting record

```
--Removing the table along with contents
DROP TABLE RR_CUSTOMER;

--Dropped Sequence
DROP SEQUENCE Cust_Sequence;

--RR_CUSTOMER table created
create table RR_CUSTOMER(
CustId NUMBER(10) Constraint CustIDPK PRIMARY KEY,
CustName Varchar2(25) Constraint CustNaNN NOT NULL,
CustAddress Varchar2(60) Constraint CustAdrNN NOT NULL,
CustPostCode Varchar2(15) Constraint CustPCNN NOT NULL,
CustPhone Varchar2 (14) Constraint CustPhNN NOT NULL,
CustEmail VARCHAR2(40)
);

--SEQUENCE Cust_Sequence created
CREATE SEQUENCE Cust_Sequence START WITH 1
INCREMENT BY 1
MINVALUE 1
MAXVALUE 1000000;

--Record inserted to the RR_customer table and inseted sequence so that customer Id is autogenerated
INSERT INTO RR_Customer VALUES (
Cust_Sequence.nextval, 'Omar Alobaidi', '3 Harrow Avenue, Hollinwood, Oldham','OL8 4HZ','0161 6285698', 'oalobaidi@yahoo.com');
INSERT INTO RR_Customer VALUES (
Cust_Sequence.nextval, 'Martyn Amos', '116 Oxford Road, Werneth, Oldham','OL9 7SJ','0161 624 9700', 'm.amos@mmu.ac.uk');
```

Rays Rentals Bicycle Hire 8



Queries

Worksheet | Query Builder

```
--Mark Bellingham--14032898
--Check orders and order details (Column alias, equi join 3 tables, comparison operator, order by)
select orderdate"Order Date", deliveryexpecteddate"Expected Delivery Date", orderreceiveddate"Date Received",
ORDERQUANTITY"Quantity", receivedquantity"Received Quantity", partname"Part Name", partdescription"Description", comments"Comments"
from rr_partsordered join rr_orderdetails on rr_partsordered.orderno = rr_orderdetails.orderno
join rr_parts on rr_orderdetails.partid = rr_parts.partid
order by orderdate;
```

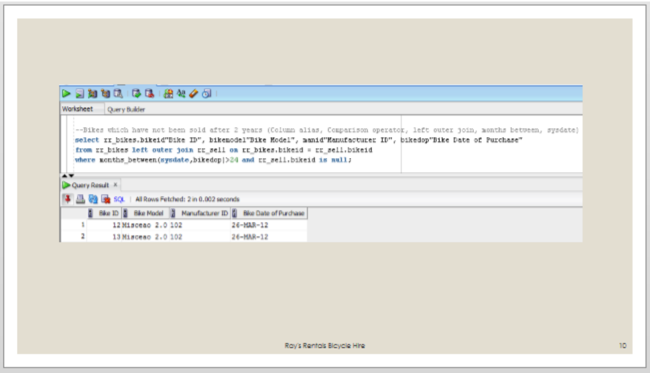
Query Result

All Rows Fetched: 10 in 0.007 seconds

	Order Date	Expected Delivery Date	Date Received	Quantity	Received Quantity	Part Name	Description	Comments
1	06-JUN-14	10-JUN-14	09-JUN-14	2	2	Inner Tubes	FME 700c Presta Valve Inner Tube	ALL PARTS RECIEVED
2	01-JUL-14	10-JUL-14	11-JUL-14	7	6	Wheels	Shimano Ultegra 6800 Tubeless Ready Wheelset	ONE WHEEL HAS VALVE MISSING
3	03-AUG-14	05-AUG-14	08-AUG-14	1	1	Wheels	Mavic Aksium One Disc 700C Front Road Wheel	ALL PARTS RECIEVED
4	07-SEP-14	09-SEP-14	09-SEP-14	19	19	Tyres	Continental Gatorskin 700C Duraskin Wired Road Tyre	ALL PARTS RECIEVED
5	11-SEP-14	16-SEP-14	18-SEP-14	10	10	Inner Tubes	FME 700c Presta Valve Inner Tube	SENT 9 60MM VALVES AND 1 48MM V
6	12-OCT-14	15-OCT-14	14-OCT-14	12	11	Pedals	Shimano R540 SPD SL Road Pedals (SE)	ONE SET OF PEDALS SHORT
7	13-DEC-14	20-DEC-14	21-DEC-14	3	3	Cassettes + Freewheels	Campagnolo Veloce 10spd Cassette	ALL PARTS RECIEVED
8	30-DEC-14	04-JAN-15	03-JAN-15	9	9	Pedals	Look Keo Grip Cleats OK	ALL PARTS RECIEVED
9	06-JAN-15	11-JAN-15	12-JAN-15	4	4	Chains	Shimano Dura-Ace 7900 10 Speed Chain	ALL PARTS RECIEVED
10	25-JAN-15	30-JAN-15	31-JAN-15	12	12	Chains	KMC X11L 11-speed Gold Chain	ALL PARTS RECIEVED

Roy's Rentals Bicycle Hire 9

9



Worksheet | Query Builder

```
--Bikes which have not been sold after 2 years (Column alias, Comparison operator, left outer join, months between, sysdate)
select rr_bikes.bikeid"Bike ID", bikemodel"Bike Model", manid"Manufacturer ID", bikedop"Bike Date of Purchase"
from rr_bikes left outer join rr_sell on rr_bikes.bikeid = rr_sell.bikeid
where months_between(sysdate,bikedop)>24 and rr_sell.bikeid is null;
```

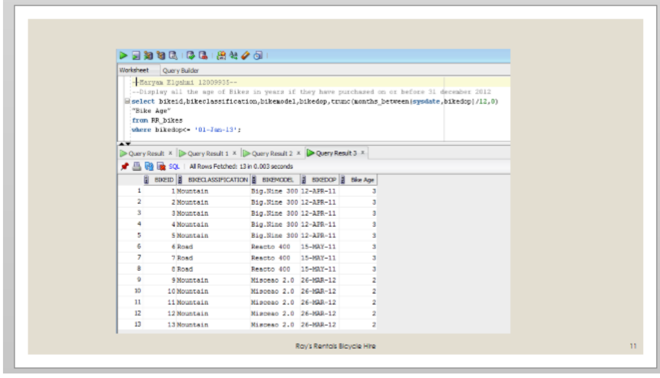
Query Result

All Rows Fetched: 2 in 0.002 seconds

	Bike ID	Bike Model	Manufacturer ID	Bike Date of Purchase
1	12	Misceao 2.0	102	26-MAR-12
2	13	Misceao 2.0	102	26-MAR-12

Roy's Rentals Bicycle Hire 10

10



Worksheet | Query Builder

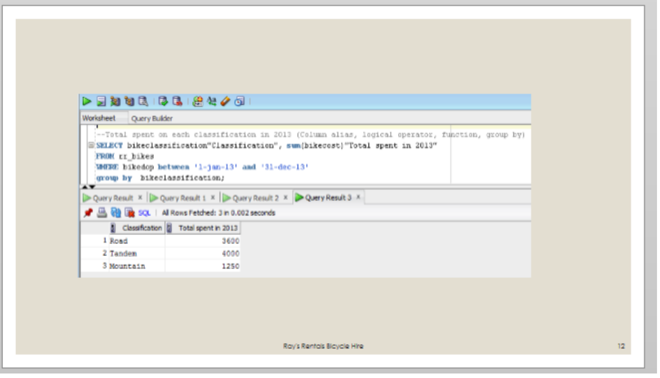
```
--Maryam Elgahmi 12009935--
--Display all the age of Bikes in years if they have purchased on or before 31 december 2012
select bikeid,bikeclassification,bikemodel,bikedop,trunc(months_between(sysdate,bikedop)/12,0)
"Bike Age"
from RR_bikes
where bikedop<= '01-Jan-13';
```

Query Result | Query Result 1 | Query Result 2 | Query Result 3

All Rows Fetched: 13 in 0.003 seconds

	BIKEID	BIKECLASSIFICATION	BIKEMODEL	BIKEDOP	Bike Age
1	1	Mountain	Big.Nine 300	12-APR-11	3
2	2	Mountain	Big.Nine 300	12-APR-11	3
3	3	Mountain	Big.Nine 300	12-APR-11	3
4	4	Mountain	Big.Nine 300	12-APR-11	3
5	5	Mountain	Big.Nine 300	12-APR-11	3
6	6	Road	Reacto 400	15-MAY-11	3
7	7	Road	Reacto 400	15-MAY-11	3
8	8	Road	Reacto 400	15-MAY-11	3
9	9	Mountain	Misceao 2.0	26-MAR-12	2
10	10	Mountain	Misceao 2.0	26-MAR-12	2
11	11	Mountain	Misceao 2.0	26-MAR-12	2
12	12	Mountain	Misceao 2.0	26-MAR-12	2
13	13	Mountain	Misceao 2.0	26-MAR-12	2

Roy's Rentals Bicycle Hire 11



Worksheet | Query Builder

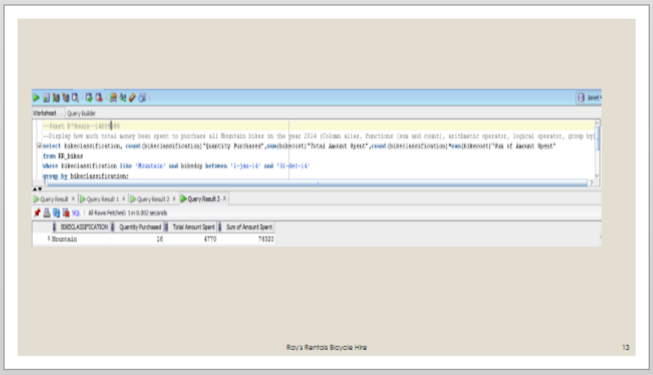
```
--Total spent on each classification in 2013 (Column alias, logical operator, function, group by)
SELECT bikeclassification"Classification", sum(bikecost)"Total spent in 2013"
FROM rr_bikes
WHERE bikedop between '1-jan-13' and '31-dec-13'
group by  bikeclassification;
```

Query Result | Query Result 1 | Query Result 2 | Query Result 3

All Rows Fetched: 3 in 0.002 seconds

	Classification	Total spent in 2013
1	Road	3600
2	Tandem	4000
3	Mountain	1250

Roy's Rentals Bicycle Hire 12



Worksheet | Query Builder

```
--Janet D'Souza--14059085
--Display how much total money been spent to purchase all Mountain bikes in the year 2014 (Column alias, functions (sum and count), arithmetic operator, logical operator, group by)
select bikeclassification, count(bikeclassification)"Quantity Purchased",sum(bikecost)"Total Amount Spent",count(bikeclassification)*sum(bikecost)"Sum of Amount Spent"
from RR_bikes
where bikeclassification like 'Mountain' and bikedop between '1-jan-14' and '31-dec-14'
group by bikeclassification;
```

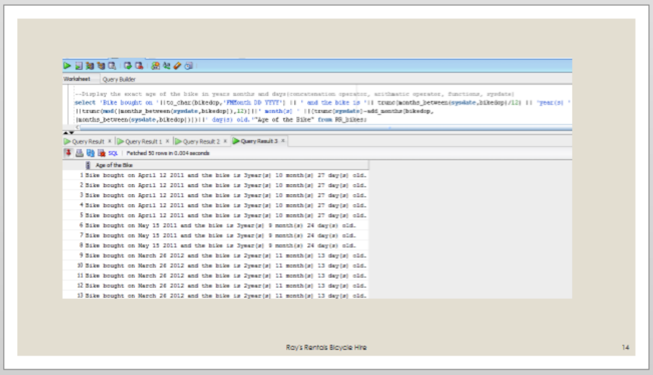
Query Result | Query Result 1 | Query Result 2 | Query Result 3

All Rows Fetched: 1 in 0.002 seconds

	BIKECLASSIFICATION	Quantity Purchased	Total Amount Spent	Sum of Amount Spent
1	Mountain	16	4770	76320

Roy's Rentals Bicycle Hire 13

13



Worksheet | Query Builder

```
--Display the exact age of the bike in years months and days(concatenation operator, arithmetic operator, functions, sysdate)
select 'Bike bought on '||to_char(bikedop,'FMMonth DD YYYY') || ' and the bike is '|| trunc(months_between(sysdate,bikedop)/12) || 'year(s) '
||trunc(mod((months_between(sysdate,bikedop)),12))||' month(s) ' ||(trunc(sysdate)-add_months(bikedop,
(months_between(sysdate,bikedop))))||' day(s) old.'"Age of the Bike" from RR_bikes;
```

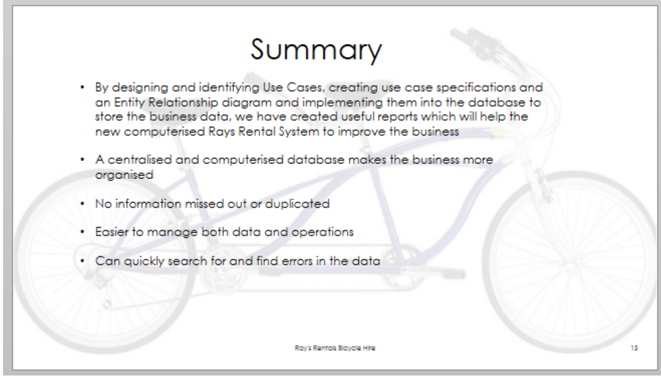
Query Result | Query Result 1 | Query Result 2 | Query Result 3

Fetched 50 rows in 0.004 seconds

	Age of the Bike
1	Bike bought on April 12 2011 and the bike is 3year(s) 10 month(s) 27 day(s) old.
2	Bike bought on April 12 2011 and the bike is 3year(s) 10 month(s) 27 day(s) old.
3	Bike bought on April 12 2011 and the bike is 3year(s) 10 month(s) 27 day(s) old.
4	Bike bought on April 12 2011 and the bike is 3year(s) 10 month(s) 27 day(s) old.
5	Bike bought on April 12 2011 and the bike is 3year(s) 10 month(s) 27 day(s) old.
6	Bike bought on May 15 2011 and the bike is 3year(s) 9 month(s) 24 day(s) old.
7	Bike bought on May 15 2011 and the bike is 3year(s) 9 month(s) 24 day(s) old.
8	Bike bought on May 15 2011 and the bike is 3year(s) 9 month(s) 24 day(s) old.
9	Bike bought on March 26 2012 and the bike is 2year(s) 11 month(s) 13 day(s) old.
10	Bike bought on March 26 2012 and the bike is 2year(s) 11 month(s) 13 day(s) old.
11	Bike bought on March 26 2012 and the bike is 2year(s) 11 month(s) 13 day(s) old.
12	Bike bought on March 26 2012 and the bike is 2year(s) 11 month(s) 13 day(s) old.
13	Bike bought on March 26 2012 and the bike is 2year(s) 11 month(s) 13 day(s) old.

Roy's Rentals Bicycle Hire 14

14



Summary

- By designing and identifying Use Cases, creating use case specifications and an Entity Relationship diagram and implementing them into the database to store the business data, we have created useful reports which will help the new computerised Rays Rental System to improve the business
- A centralised and computerised database makes the business more organised
- No information missed out or duplicated
- Easier to manage both data and operations
- Can quickly search for and find errors in the data

Roy's Rentals Bicycle Hire 15

Roy's Rentals Bicycle Hire 16