Information Systems Part 4

RYAN PARKINS, MADALINE JONES, MICHELLE WILLIAMSON, KARAN NIHALANI

Contents

Executive summary 3
Introduction 4
Background 4
Aims 4
Part 1 4
Introduction 4
Background 4
Aims 4
Executive Summary 5
Question 1 – Problems with current system at Ray's Rentals 5
Question 2 – System Requirements for new database system 6
Question 3 – What are data enquiries and management reports 6
Question 4 – Outline several data enquiries and management reports that would be useful for Ray's rentals 7
Conclusion 8
Abstract 8
Figures 8
Log Entries From Piazza 10
Part 2 11
Introduction- 11
Background 11
Aims 11
Executive Summary- 11
Prioritised Use Case Diagram Of New System. 12
Commentary explaining decisions made when creating the UCD. 13
Use case specs and ERD's (1 per student) 13
Top-Down ERD of the system. 20
RDA of two documents provided in case study and a bottom-up ERD of merged RDA's. 21
..... 22
Finalised Group ERD (top-down and bottom-up) covering the complete system. 23
Commentary Explaining decisions made when creating the ERD for the finalised system 24
Conclusion 24
Appendix 25

Log Entries From Piazza 25
Part 3 27
Introduction 27
Background 27
Aims 27
Executive summary 27
Question 1 – Amended ERD 28
Question 2 – Data Dictionaries 28
Ryan 28
Michelle 29
Karan 30
Maddie 32
Question 3 – Data dictionaries implemented into ORACLE 33
Question 4 – SQL Queries 35
Michelle 35
Ryan 36
Madaline 37
Karan 38
Question 5 – Document database design with an ORACLE script file, and outputs of when the script is run 40
Question 6 – Commentary explaining what has been learnt 42
Conclusion 43
Appendix 44
Log entries from Piazza 44
Conclusion 46
References 47
Appendices 48
Appendix A : Log Book 48
Appendix B : Presentation slides 49

Executive summary

This report sets out to analyse a cycle hire store called Ray's rentals, the store currently uses an outdated paper-based system, this system has many disadvantages that will be outlined including loss of records, double reservation bookings, maintenance issues including running out of stock and a general lack of organisation.

The report will then look at ways that an electronic system would help to improve these problems, it was decided that an electronic database system would be suitable for Ray's rentals and this database should be able to take queries to produce management reports. These management reports will include stock queries, reservation queries and maintenance history. The report will also investigate types of data enquiries and management reports.

Furthermore, the management reports were then implemented into a use case diagram, the use case diagram is used to describe a set of actions that can be performed by the system's users. Creating this showed the connections between use cases. Knowing these connections meant that the system could be implemented into an entity relationship diagram.

The entity relationship diagram was created and the connections previously established could then be created. Relationships were added to the diagram, and a many to many relationship was resolved using a weak entity.

Primary and foreign keys were established, this allowed each record to have a unique value such as customerID. After establishing all keys and relationships, data dictionaries were made, and these allowed the basic database structure to be developed.

The data dictionaries were then translated into structured query language (SQL) where they were run and the database was created. Insert statements were used to input data into the tables for later queries to work successfully.

The queries created helped Ray's rentals to become more functional and reliable; the problems previously outlined will be removed. Queries have also been created that search the database and present data to staff in an easy to read and understandable way.

Introduction

Background

Ray's rentals is a cycle hire shop that currently uses an outdated paper-based system, the shop owns around one hundred and fifty bikes, there are many problems with the way this business is run, these include loosing paper records belonging to bikes or maintenance reports, also double bookings that often occur due to the complicated booking process of having to write all reservations for one bike on the same sheet of paper.

Aims

The aims of this report are to analyse the business of Ray's rentals and design a system that will be suitable for them, this system needs to be able to remove the current problems that are faced when using the paper-based system. The system will then be created using ORACLE SQL which will be suitable for implementation.

The new system aims to increase productivity, efficiency and ease of use for both the staff and the customer. All these aims should increase business at Ray's rentals therefore increasing profit and allowing the business to increase as a whole.

Part 1

Introduction

Background

Ray's Rentals is a small bike hire business located in a small town, the business owns around one hundred and fifty bikes, there are multiple problems with the way that the business is run, this report will examine the way that the business operates

Aims

The aims of this report will be to examine problems in the business and determine possible solutions to these problems, the report will then decide what the system requirements will be for the proposed new system. Data enquiries and management reports will be explained and reports and enquiries that would be useful for Ray's Rentals will be determined.

Executive Summary

Ray's rentals is a small business that specialises in the rental of bikes. The business has many weaknesses that all contribute to the poor efficiency of the business, this report outlines these weaknesses and suggests methods of improvement. When analysing the case study, it was clear that the paper-based system at Ray's rentals to blame for this.

The report analyses these problems and suggests ways to modernise and improve these old methods to overcome some of the confusion and frustration that the business currently endures. The report concludes that a computerised system alongside an electronic database system would be able to help improve the functionality and running, thus improving business and custom.

Question 1 – Problems with current system at Ray's Rentals

The current paper-based system at Ray's rentals has many limitations to its functionality, these limitations cause delays and cause uncertainty in the business.

Firstly, A customer must either come into the store or phone the store to rent a bike. If they come into the store, they will give the staff member all the details regarding the bike they wish to rent, the staff member will write this information down and then sort through pieces of paper that have the rental information on for each individual bike. When phoning the store, the process is the same, however, the staff will take details, look to see if a bike is available and then ring the customer back if a bike is free. This is very time consuming and can often lead to double bookings and not finding reservations that have been made.

Secondly, bikes have their own maintenance records that are all written on paper. These records have the bikes last service date on them as well as any maintenance work or repairs that has been carried out on them. This system makes it difficult to track all bikes at one time. When bikes have not been serviced in a month, they are selected and then serviced. This is especially difficult when having to look through 150 sheets of paper and trying to figure out which bikes are due for service.

Moreover, when bikes require maintenance they sometimes need parts, these parts should always be kept in stock at Ray's rentals. When they receive parts that have been ordered, they are not recorded and accounted for so there is no stock control. This can cause problems as things will get ordered when not needed and other things will be missed when out of stock.

Finally, when a customer comes to pay for their rental, their receipt is hand written at the time of purchase, this can be a problem as there will be no real record kept by the business for future reference and this can be time consuming as all information is not always at hand.

Question 2 – System Requirements for new database system

A new computerised system will allow Rays Rentals to run much more efficiently and will be much better for business.

This new computerised system should firstly be able to keep records of all the bikes Ray owns and should also give information about them. It needs to keep maintenance records and service histories of each individual bike. The system should also be able to take queries for reservations, show the customer what is available for them to rent and the rent prices.

Reservations should also be managed by the system to help staff know what bikes are due out for rental each day and what bikes are due back for rental. This would benefit the staff as they can ensure that all bikes are ready for collection at the time they are due out.

Stock at Rays Rentals should be managed by this system, current stock levels will be recorded by adding delivery invoices to the system and recording what stock is then used for repairs on bikes. Altering these figures will alter the stock levels electronically and will automatically generate a recommended delivery and inform staff when it needs ordering. Using this system will also record what parts have been used on specific bikes, this can then be searched and bikes due for service will be flagged up on the system.

When customers pay for their reservations, they should receive a printed receipt. The system will be able to print this out at the time of payment and will include information regarding their purchase, this information will be stored by the company for a limited time for future reference.

Question 3 – What are data enquiries and management reports

Data enquiries are enquiries which “gather information from agents end-users or software components, to be used later on in the process” **(cossac.org,2011)**.

Management Reports are reports that include “general documentation, project management, financial control, email communication and business presentation” **(Carr et al, 2001)**. All of the information in the report can be summarised and used to aid a manager to reach targets that have been set out.

There are numerous management reports which could be used, the first one being an analysis report. An analysis report “delivers a summary of data for a specific period of time”. **(Stephen Gordon; 2017)**. When looking at an analysis report, it will be clear what the strengths and weaknesses are in the business. This will help Rays Rentals as it will provide the staff with areas of improvement and they will be able to track the developments that are being made.

The next management report is an exception report these are a “type of summary report that identifies events that are outside the scope” **(Tatum, 2017)** this is mainly to do with the processing of inventory reconciliation which is “the process of counting and evaluating sock-

in-trade" (erpnext.org, 2014). Exception Reports can help Rays Rentals because it will provide the staff with the keep track of the inventory that is in stock and will help them to know what needs to be ordered for customers, this will enable Rays Rentals to save time and focus it on the maintenance and repair of the bikes so that they can gain more customers and therefore increasing their revenue.

The final management report is key target reports. The reports are "A style of management information report where performance data is calculated and set out against predefined targets". **(Whiteley, 2013).** As the targets "need to be limited in number, wisely selected and attainable" **(Whiteley, 2013)** this will allow the staff at Rays Rentals to monitor their improvements and the areas they have developed strengths and they can pinpoint where they are lacking in skills and improve upon this by developing their knowledge. Stock targets can also be put in place for the whole business when for example over summer, more bikes will be rented.

Question 4 – Outline several data enquiries and management reports that would be useful for Ray's rentals

The first data enquiry that would be of use to Ray's Rentals would be to analyse the rental history of bikes to find the most requested type of bike. By pinpointing the most popular type of bike, Ray's rentals could possibly invest some of their profits in purchasing this type of bike.

The second data enquiry will be able to check if a bike has already been rented out for that day, this would prevent rentals from overlapping, this allows them to have more control over the rental process.

The final data enquiry will be to track stock/parts this will be of use to Rays Rentals as it will allow the staff to know what is needed for repairs to the bikes. Thus, saving time for staff as they will not need to update the maintenance papers.

The first management report we would use will be an analysis report as this will help Rays Rentals to see the profits for a year, whether they had risen, fallen and help them highlight reasons for this. An example of this is shown in the diagram below (**See fig 3**). Rays Rentals could determine through the data the impact of different situations, such as seasonal variations, stock management control and cyclical variations.

Another management report which could be used is exception reports. As exception reports highlight unusual cases, Rays Rentals could possibly use this to their advantage and identify which bikes are always breaking down and needing the most repairs. An example of this is shown below (**see fig 4**). This would be beneficial as they could see where most of their revenue is going and whether it would be cheaper to invest in a new bike rather than paying for numerous repairs.

Finally, the last management report which we would use are key target reports. This would motivate managers and staff more as they would have something which they would be aiming to, allowing them to see whether they are completing tasks effectively. An example of a key target report which could be used is the number of bikes which are being rented over a particular period of time. This may be impacted by advertisements and customer satisfaction, as the higher the reviews then the more custom they may gain. An example of key target report is shown below (**see fig 5**).

Conclusion

This report was set out to analyse how Ray's Rentals operates and to decide on what sort of system would be better suited to the business in order to increase business, the system needs to be able to minimise the problems that have been outlined such as loosing bike records, the data enquiries and management reports should be able to help with these issues in the future by keeping records computerised, this will make the system much more manageable and easier to use to staff.

Abstract

Figures

Figure 1 – (Schofield, 2018)

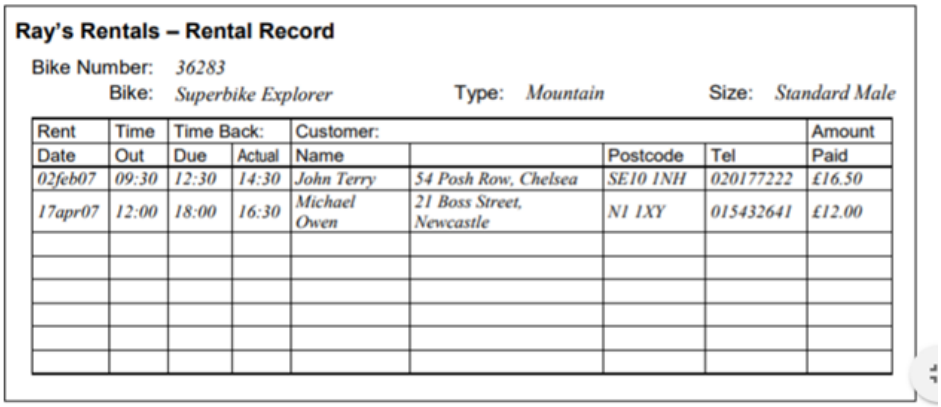
Ray's Rentals – Bike Record

Bike Number:	*36283*	Manufacturer:	*Superbike*
Model:	*Explorer*	address:	*Buxton Road*
			Sheffield
Purchase: date	*30 Jan 07*	postcode	*S10 1AB*
price	*£350.00*	telephone	*0114 369 1476*
Classification:		Disposal Details:	
mountain	☑	sale date:	*15 Mar 09*
road	☐	dealer name:	*Joe's Junk*
tandem	☐	- address	*15 Back Lane*
Size:			*Buxton*
large male	☐	- telephone	*0736 553422*
standard male:	☑	sale price	*£45.00*
small male:	☐		
standard female:	☐		
child:	☐		

Maintenance History

ref no	fault details	fault date	action taken	action date
1	*Puncture - rear*	*15feb07*	*New Inner Tube*	*16feb07*
2	*Chain keeps breaking*	*10jul07*	*Fit new Chain*	*27aug07*
3			*Normal Service*	*15dec07*
4			*Normal Service*	*20jan09*
5	*Split Saddle*	*02feb09*	*Fit new Saddle*	*27feb09*

Figure 2 – (Schofield, 2018)

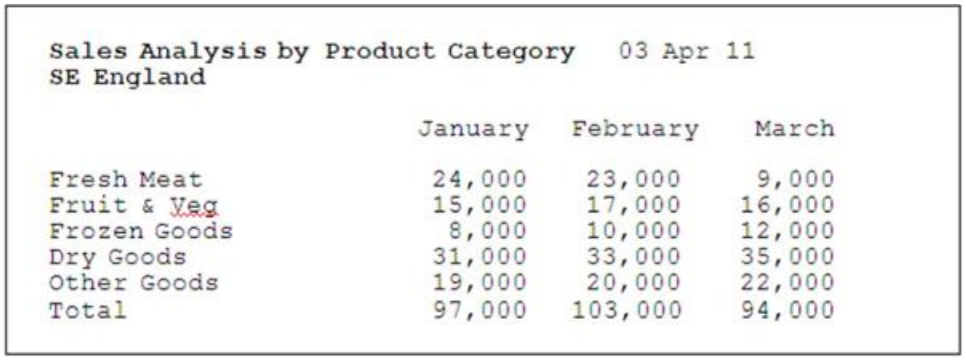


Ray's Rentals – Rental Record

Bike Number: *36283*
Bike: *Superbike Explorer* Type: *Mountain* Size: *Standard Male*

Rent	Time	Time Back:		Customer:				Amount
Date	Out	Due	Actual	Name		Postcode	Tel	Paid
02feb07	*09:30*	*12:30*	*14:30*	*John Terry*	*54 Posh Row, Chelsea*	*SE10 1NH*	*020177222*	*£16.50*
17apr07	*12:00*	*18:00*	*16:30*	*Michael Owen*	*21 Boss Street, Newcastle*	*N1 1XY*	*015432641*	*£12.00*

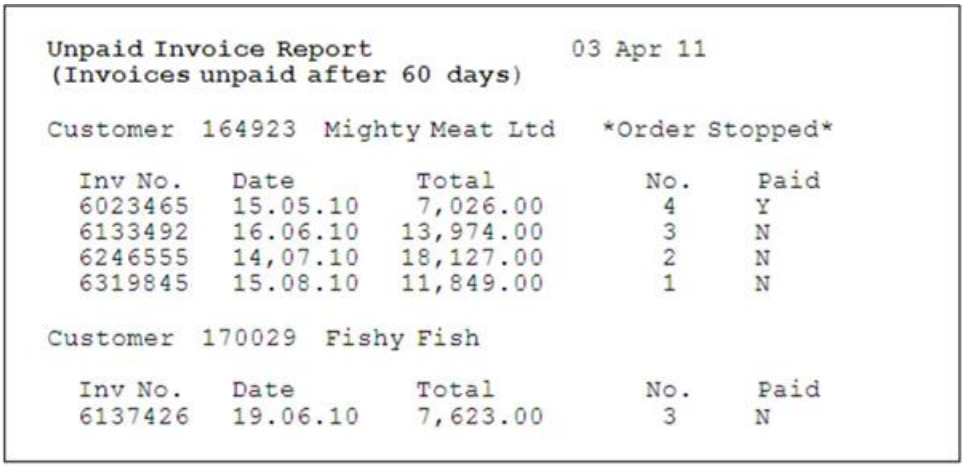
Figure 3 – (Gordon, 2017)



Sales Analysis by Product Category 03 Apr 11
SE England

	January	February	March
Fresh Meat	24,000	23,000	9,000
Fruit & Veg	15,000	17,000	16,000
Frozen Goods	8,000	10,000	12,000
Dry Goods	31,000	33,000	35,000
Other Goods	19,000	20,000	22,000
Total	97,000	103,000	94,000

Figure 4 – (Gordon, 2017



Unpaid Invoice Report 03 Apr 11
(Invoices unpaid after 60 days)

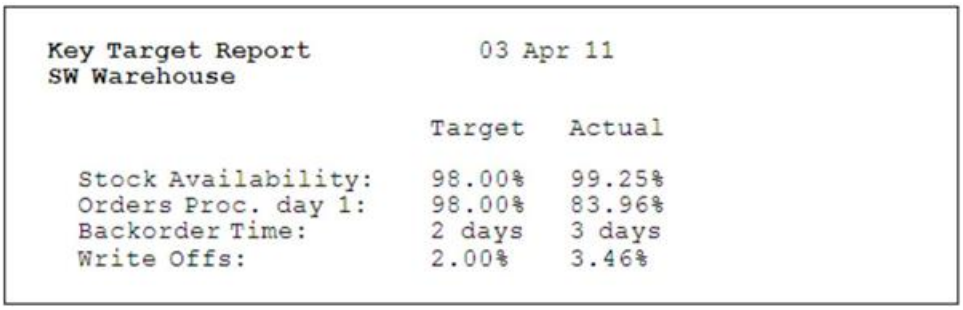
Customer 164923 Mighty Meat Ltd *Order Stopped*

Inv No.	Date	Total	No.	Paid
6023465	15.05.10	7,026.00	4	Y
6133492	16.06.10	13,974.00	3	N
6246555	14,07.10	18,127.00	2	N
6319845	15.08.10	11,849.00	1	N

Customer 170029 Fishy Fish

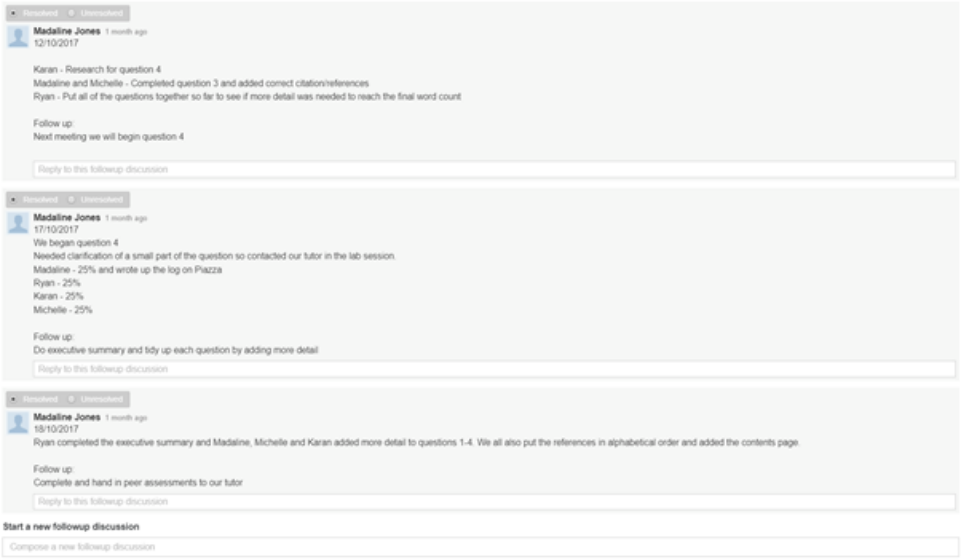
Inv No.	Date	Total	No.	Paid
6137426	19.06.10	7,623.00	3	N

Figure 5 – (Gordon, 2017)



```
Key Target Report              03 Apr 11
SW Warehouse

                            Target   Actual

  Stock Availability:       98.00%   99.25%
  Orders Proc. day 1:       98.00%   83.96%
  Backorder Time:           2 days   3 days
  Write Offs:               2.00%    3.46%
```

Log Entries From Piazza



Resolved Unresolved

Madaline Jones 1 month ago
12/10/2017

Karan - Research for question 4
Madaline and Michelle - Completed question 3 and added correct citation/references
Ryan - Put all of the questions together so far to see if more detail was needed to reach the final word count

Follow up:
Next meeting we will begin question 4

Reply to this followup discussion

Resolved Unresolved

Madaline Jones 1 month ago
17/10/2017
We began question 4
Needed clarification of a small part of the question so contacted our tutor in the lab session.
Madaline - 25% and wrote up the log on Piazza
Ryan - 25%
Karan - 25%
Michelle - 25%

Follow up:
Do executive summary and tidy up each question by adding more detail

Reply to this followup discussion

Resolved Unresolved

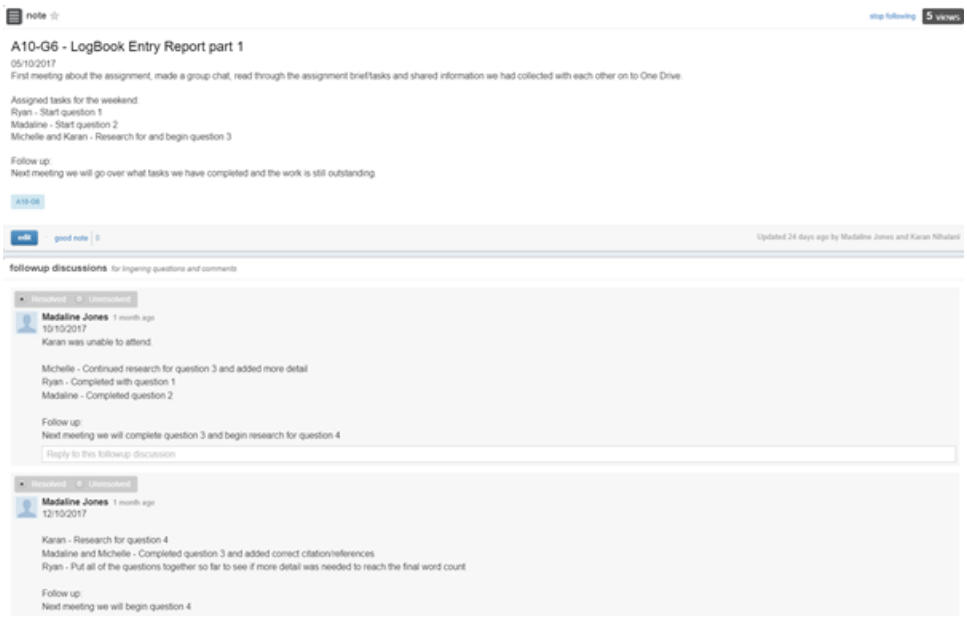
Madaline Jones 1 month ago
18/10/2017
Ryan completed the executive summary and Madaline, Michelle and Karan added more detail to questions 1-4. We all also put the references in alphabetical order and added the contents page.

Follow up:
Complete and hand in peer assessments to our tutor

Reply to this followup discussion

Start a new followup discussion

Compose a new followup discussion



note

stop following 5 views

A10-G6 - LogBook Entry Report part 1
05/10/2017
First meeting about the assignment, made a group chat, read through the assignment brief/tasks and shared information we had collected with each other on to One Drive.

Assigned tasks for the weekend:
Ryan - Start question 1
Madaline - Start question 2
Michelle and Karan - Research for and begin question 3

Follow up:
Next meeting we will go over what tasks we have completed and the work is still outstanding.

A10-G6

edit good note 0

Updated 24 days ago by Madaline Jones and Karan Nihalani

followup discussions for lingering questions and comments

Resolved Unresolved

Madaline Jones 1 month ago
10/10/2017
Karan was unable to attend.

Michelle - Continued research for question 3 and added more detail
Ryan - Completed with question 1
Madaline - Completed question 2

Follow up:
Next meeting we will complete question 3 and begin research for question 4

Reply to this followup discussion

Resolved Unresolved

Madaline Jones 1 month ago
12/10/2017

Karan - Research for question 4
Madaline and Michelle - Completed question 3 and added correct citation/references
Ryan - Put all of the questions together so far to see if more detail was needed to reach the final word count

Follow up:
Next meeting we will begin question 4

Part 2

Introduction-

Background

Ray's Rentals is a bike rental shop that currently uses an outdated paper-based recording system. There are many problems to the running of this business that have been highlighted before this report as well as possible solutions to this.

Aims

This report aims to design the new system that will simplify and modernise the old system but with an aim to increase the organisation of the business and more importantly improved efficiency, this in turn will increase productivity and revenue.

Executive Summary-

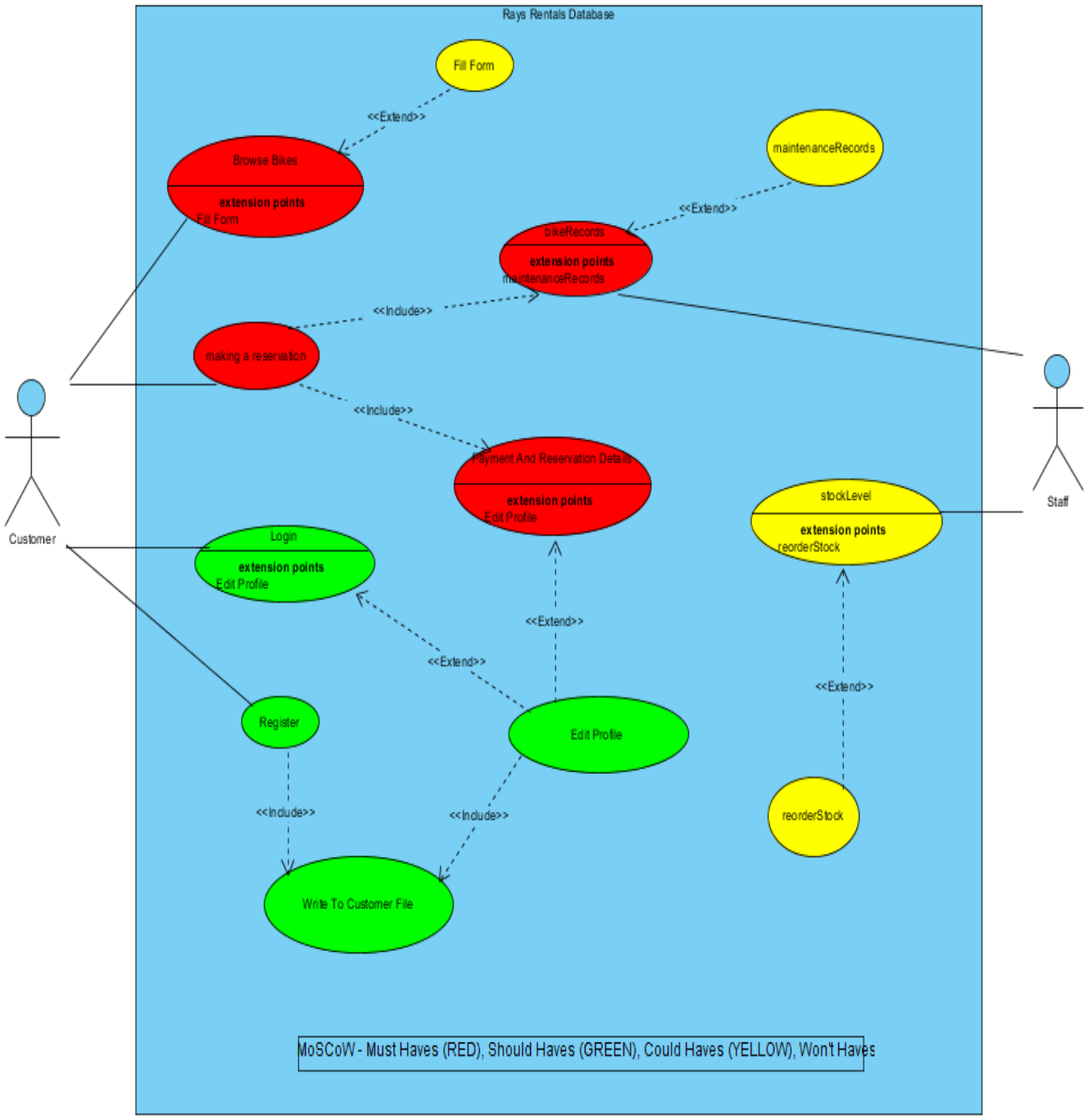
Contained within this report is a recommendation for a new system to be used at Ray's Rentals, a bike rental shop. As mentioned previously, there are many problems that have been outlined with the business and possible solutions were made. This report outlines all of the steps taken to design the new system for this business.

The system that is recommended for Ray's Rentals was designed using many data diagrams such as use case diagrams, entity relationship diagrams and relational data analysis diagrams. The final system that has been created is displayed in an entity relationship diagram, this diagram shows off the system and the data that the system will store.

This system has been proposed for the business as implementing this system will remove many problems that Ray's Rentals currently face, this system will also make the whole process of renting a bike much easier for both the staff and the customer, this in turn will drive sales and increase staff productivity and reliability.

The next step in this process is the technical design, this process involves designing the electronic system, and using the diagrams created in this report will make ease of the process, thus reducing time and cost.

Prioritised Use Case Diagram Of New System.



Commentary explaining decisions made when creating the UCD.

The use case diagram for the new system at Ray's rentals was created with two actors, staff and customer. The decisions for using these two actors was made as "staff" represents the staff at Ray's rentals and "customer" represents the customers.

The use cases created for this diagram included register, login, edit profile, browse bikes, payment and reservation. We decided to create these use cases as they were best suited to the system at Ray's rentals and covered all parts of the system. It was decided that the customer would have to interact with login, make a reservation and browsing bikes as these are all use cases that the customer must initiate before anything will happen with the system. It was also decided that staff would only need to interact with bike records and stock level, this decision was made as anything that the staff would need to access would be available in one of those two use cases. Also included were dependencies within the diagram to show the flow of data between certain use cases.

The MoSCoW system was used in the diagram to display priorities with the use cases and which ones were necessary for the system to work, which should be included, and which use cases could be included but the system could still be used without.

After completing this diagram, it was learnt that use case diagrams are an important part of the system analysis and design stage of the waterfall model. Finally it was learnt that using the MoSCoW system allows us to prioritise each of the use cases, helping us to understand which tasks are most necessary to be able to carry out the rental thoroughly.

Use case specs and ERD's (1 per student)

Use Case: BIKE RENTAL RECORDS (Karan Nihalani)
Owner: STAFF MEMBER
Pre-Conditions
Firstly, the bike information and details are made sure to be accurate and up-to-date, stored automatically in a database within the company. These details should include the date and times of hire, the details of the customer that is willing to rent the bike and the fee that is to be paid for hiring the bikes, normally depending on the amount of time it is hired for (half day or full day).
Post-Conditions
Once, the hiring and/or the payment is actually done, all the data is entered into the database or even when the bike is returned, the data is entered.

Finally, it is made sure that as many details as possible are entered by the staff member, the data being, Rent Date, Customer Name, Customer Address, Post Code, Telephone Number, Bike Class and Size.

Primary Path

1. The Bike Rental details will be entered, once a bike has been ordered for hire.
2. The customer would pick the bike they want to rent.
3. This rental will be added to the database.
4. These records are updated automatically as each new rental is carried out.

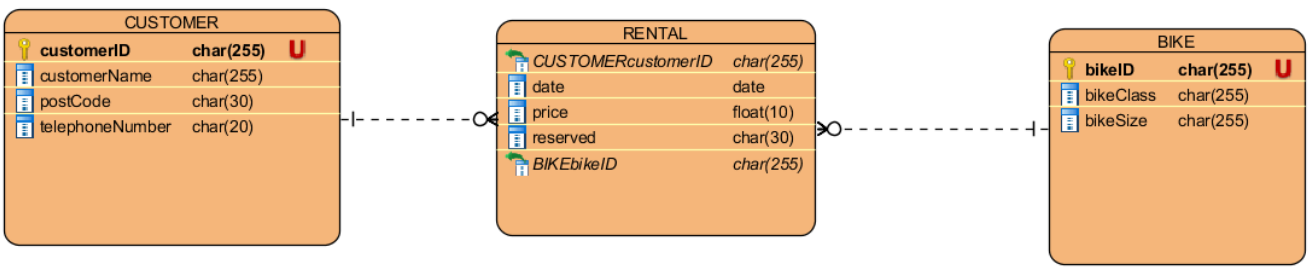
Alternate Path

1. **"Lost" Reservations, due to reservations taken weeks/days in advance:**

   1.1 There is an automatic search for an available bike that is carried out.

   1.2 Once this is done, the bike rental is immediately shown as reserved for the customer that has ordered it for hire, so that it doesn't get lost due to the record being added on the next line.

Notes



Use Case: Payment (Madaline Jones)

Owner: Customer

Pre-Conditions

When customers make payment for hiring a bike they are given and hand-written receipt. These receipts can also be posted out to customers who have sent a cheque in advance.

Post-Conditions

When customers pay for the reservations they have made they should receive a printed receipt. The system will print this receipt at the time the of payment and will include details about their purchase which will also be stored on the system for a limited time for future reference.

Primary Path

1) Customers will pick the bike they want to rent.
2) Customers will pay a deposit for the bike before collecting it.
3) Customer will collect bike and pay in full for the rental.
4) Staff will process this payment and give the customer the receipt.

Alternate Path

2a) Customers unable to make deposit on this bike as card was declined

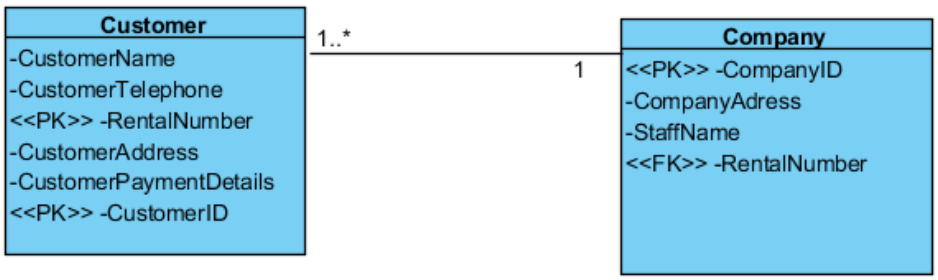
2a1) Customer uses different payment method

3a) Customers unable to pay in full for the bike as their card is declined/payment not cleared

3a1) Explain to customer that their payment has been declined and ask for another payment method.

Notes

None



Use Case: Making a reservation (Michelle Williamson)

Owner: Customer

Pre-Conditions

Details of the requested reservation are noted down on a piece of paper, along with the name and phone number of the customer who is making an enquiry. The staff member then checks to see whether the right bike is available when the customer wants it. After this has been checked, the hiring department then phone the customer back to inform them whether the bike is available or not. If the bike is available then the staff member takes the customers details, these are entered on the Rental Record. The rest of the data is entered when the customers payment is received.

Another path which could be taken is if the customer has sent a letter in, enquiring making a reservation of a bike. In this case, the letter is sent in and the Hiring Department puts a copy of the sample pricing of bike hires. If these are out of date or inaccurate then Sheila or Megan, make hand written corrections on the original and then posts it off to the customer. The above path is then followed on.

Post-Conditions

An online system will be put in place where the customers can enter the date and time when they want to hire the bike, they can also get the option to chose which type of bike they want, if theyre after a specific one. The system will tell customers which bikes they can hire at which times and whether the one they require is available, this is so they don't have to ring up – this will be saving both the customers time and staffs time. Also, the customer wont have to wait for the hiring department to ring them back to confirm whether the bike is available.

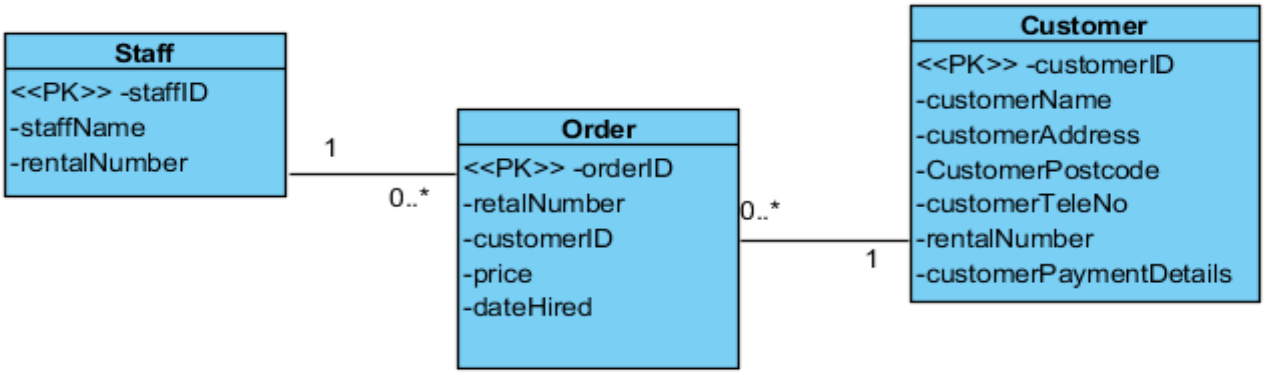
Primary Path

1) Customer enters their details online – when they want their bike/which bike they would like to hire.
2) System states whether the bike is available at that date/time.
3) Rent prices displayed for that specific reservation.
4) If customer is satisfied they make a deposit on the bike.
5) This reservation is automatically put in the database, this stops double booking for the same bike.
6) Customer collects the bike on the required date/time.
7) The customer pays for the bike before they leave with it.
8) The customer is given a receipt for the deposit and the total cost of the bike hiring.
9) Customer returns with the bike, if it is damaged then they don't get their deposit money back.

Alternate Path

1) The bike which the customer enquired about isn't available when they want it.

2) The website offers alternatives. 3) If they aren't suitable for the customers then they can leave their name and phone number on a form. 4) The customer will be informed if there is a cancellation on the bike they wanted. 5) Customers can continue browsing the website, if they find another suitable date and time then follow the primary path from step 3.
Notes



Use Case: Maintenance (Ryan Parkins)

Owner: Staff

Pre-Conditions

Bikes are maintained by staff at Ray's rentals, if a bike incurs a fault, it will be taken to the workshop and fixed. Parts that are used to repair bikes are sometimes not ordered and so delays would occur. All maintenance carried out on the bikes will be recorded on each individual bike record. Bikes are also serviced at a one-month interval, looking at the bike maintenance history papers allows staff to identify when bikes are due for a service.

Post-Conditions

Bike details are all contained within a database, bikes that are due to be serviced will be flagged up on the system for the staff. This allows for bikes to never be missed out, bikes that occur a fault that require immediate maintenance will be logged on the system with their problem, staff will then see all bikes that need repairing on the system and can mark them as complete when they have been repaired.

Primary Path

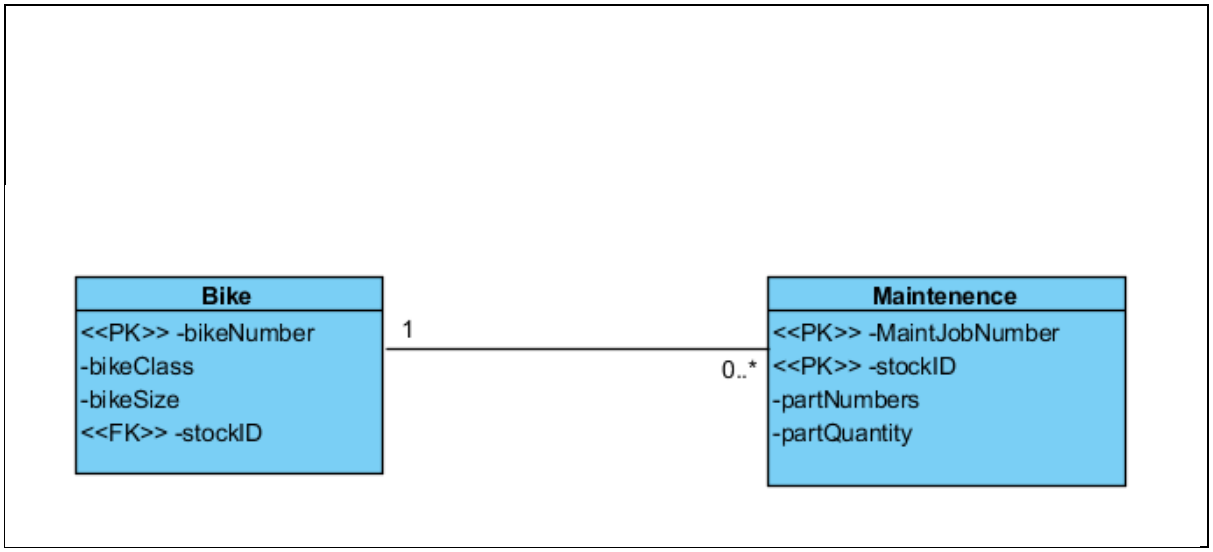
1) Bike occurs a fault / is due to be serviced
2) Bike will be allocated a maintenance job and flagged up on the system and marked as unavailable for rental.
3) Staff will repair and service bikes that have been flagged up on the system
4) Once a bike has been repaired / serviced, it will be marked as complete on the system and open for rentals.

Alternate Path

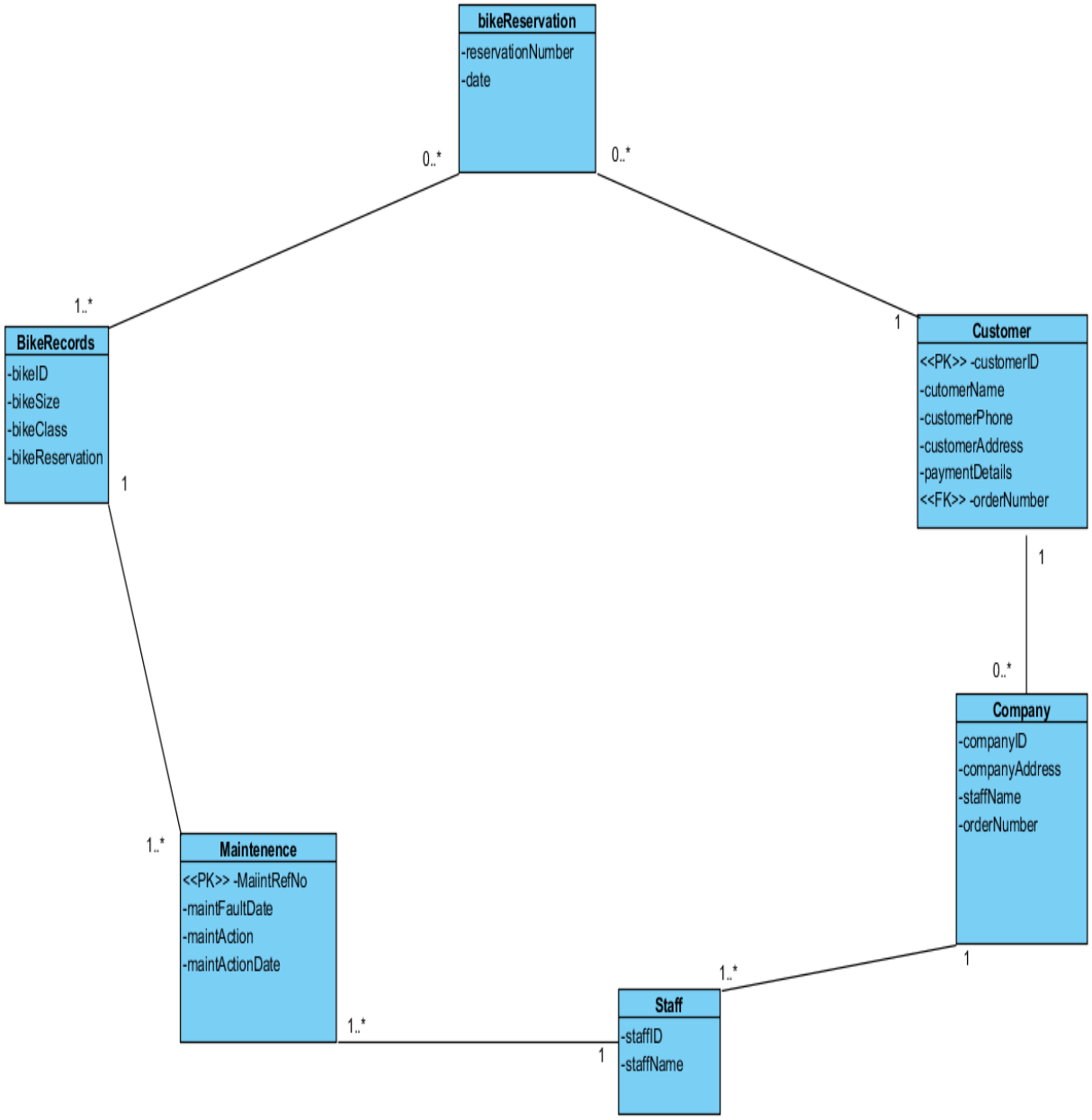
1a) Staff attempt to make a repair but repair is not completed, the main reason for uncompleted repairs can be due to parts being out of stock.

1b) Bike will be marked on the system as not repaired and staff will be able to see why the repair was not completed.

1c) Once parts are in stock again or the other reason has been resolved, the bike will be re allocated a maintenance job.


Bike
<<PK>> -bikeNumber
-bikeClass
-bikeSize
<<FK>> -stockID
1
0..*
Maintenence
<<PK>> -MaintJobNumber
<<PK>> -stockID
-partNumbers
-partQuantity

Top-Down ERD of the system.

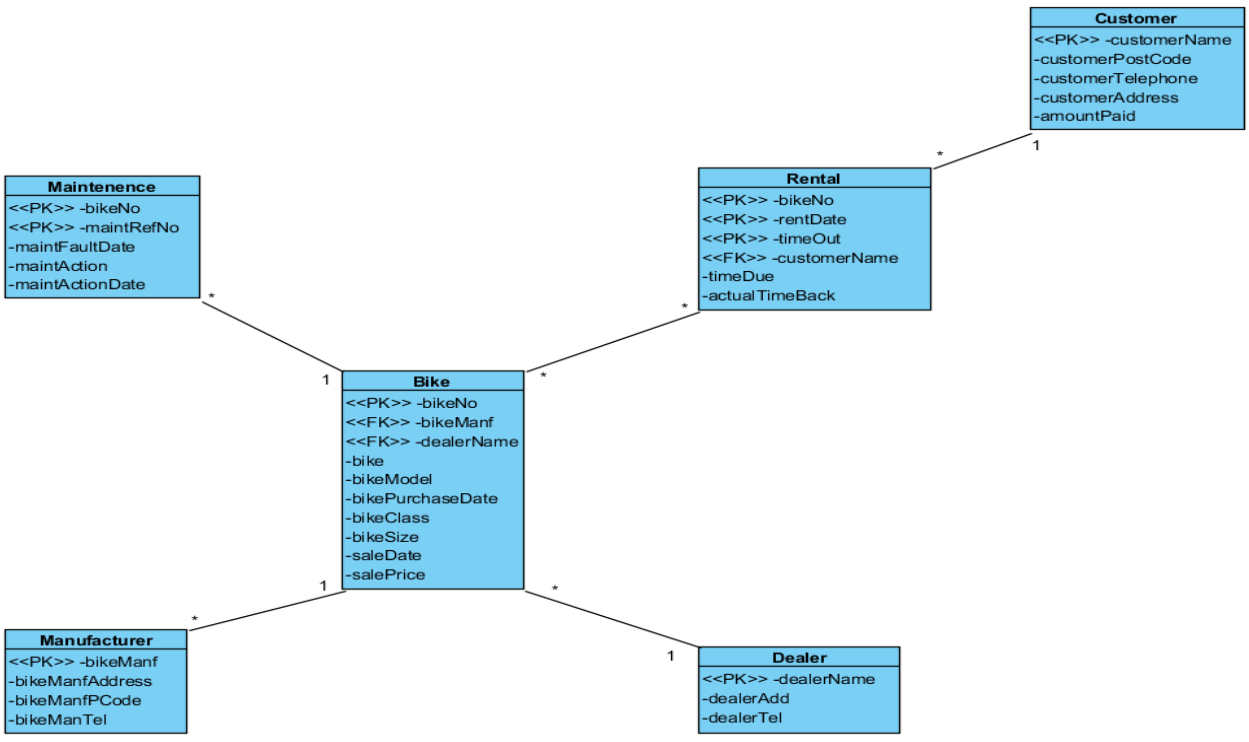


RDA of two documents provided in case study and a bottom-up ERD of merged RDA's.

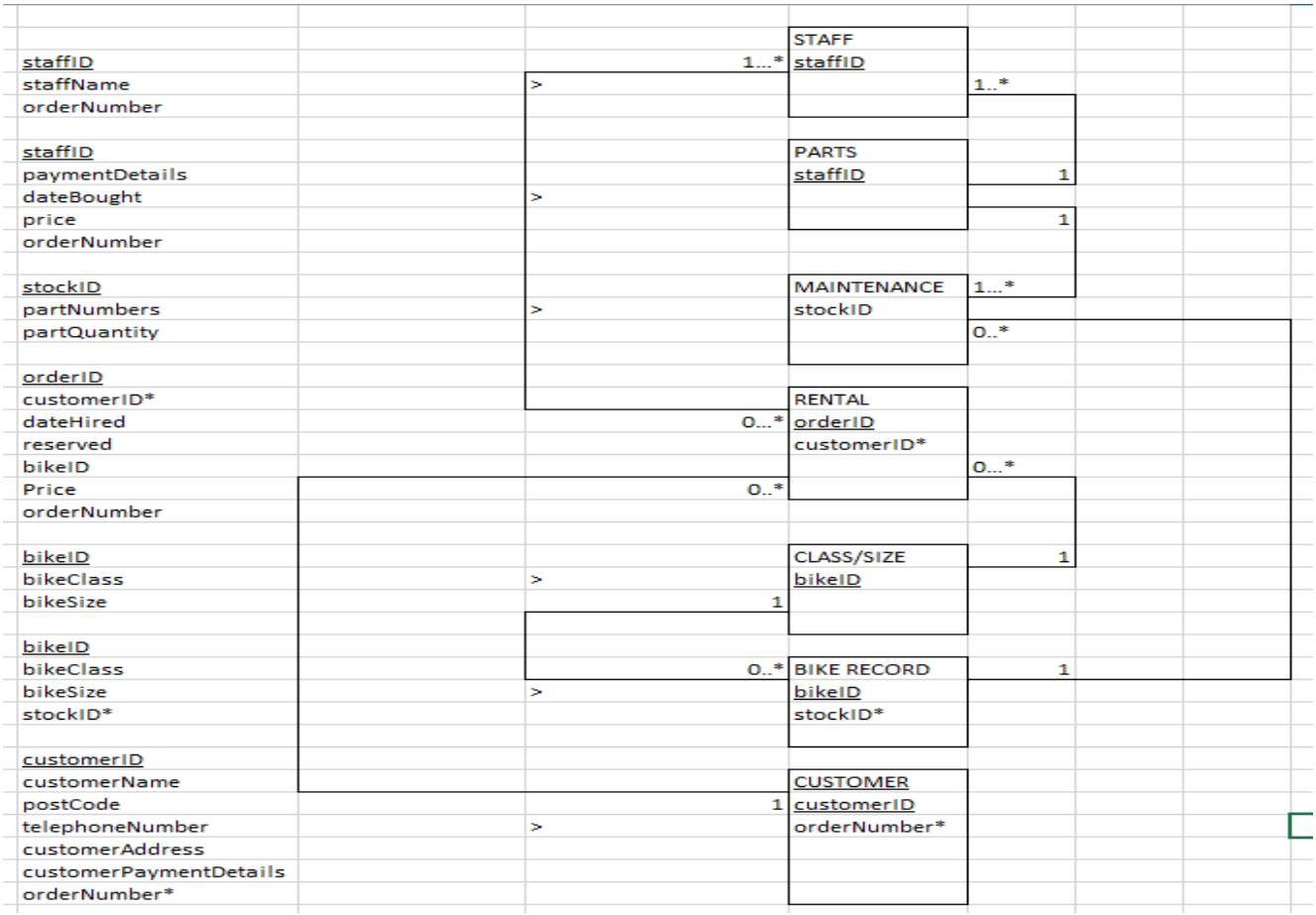
UNF	1NF	2NF	3NF			
bikeNo	bikeNo	bikeNo	bikeNo			
bikeModel	bikeModel	bikeModel	bikeModel			
bikePurchasePrice	bikePurchasePrice	bikePurchasePrice	bikePurchasePrice			
bikePurchaseDate	bikePurchaseDate	bikePurchaseDate	bikePurchaseDate	>	BIKE	
bikeClass	bikeClass	bikeClass	bikeClass	*	bikeNo	
bikeSize	bikeSize	bikeSize	bikeSize		bikeManf*	1
bikeManf	bikeManf	bikeManf	bikeManf*		dealerName*	
bikeManfAddress	bikeManfAddress	bikeManfAddress	saleDate			*
bikeManfpCode	bikeManfpCode	bikeManfpCode	salePrice			
bikeManTel	bikeManTel	bikeManTel	dealerName*			
saleDate	saleDate	saleDate				
salePrice	salePrice	salePrice	bikeManf			
dealerName	dealerName	dealerName	bikeManfAddress			
dealerAdd	dealerAdd	dealerAdd	bikeManfpCode	>	MANUFACTURER	
dealerTel	dealerTel	dealerTel	bikeManTel	1	bikeManf	
maintRefNo						
maintFaultDet	bikeNo	bikeNo	dealerName			
maintFailtDate	maintRefNo	maintRefNo	dealerAdd	>	DEALER	
maintAction	maintFaultDet	maintFaultDet	dealerTel		dealerName	
maintActionDate	maintFailtDate	maintFailtDate				1
	maintAction	maintAction	bikeNo			
	maintActionDate	maintActionDate	maintRefNo			
			maintFaultDet	>	MAINTENANCE	
			maintFailtDate		bikeNo	*
			maintAction		maintRefNo	
			maintActionDate			

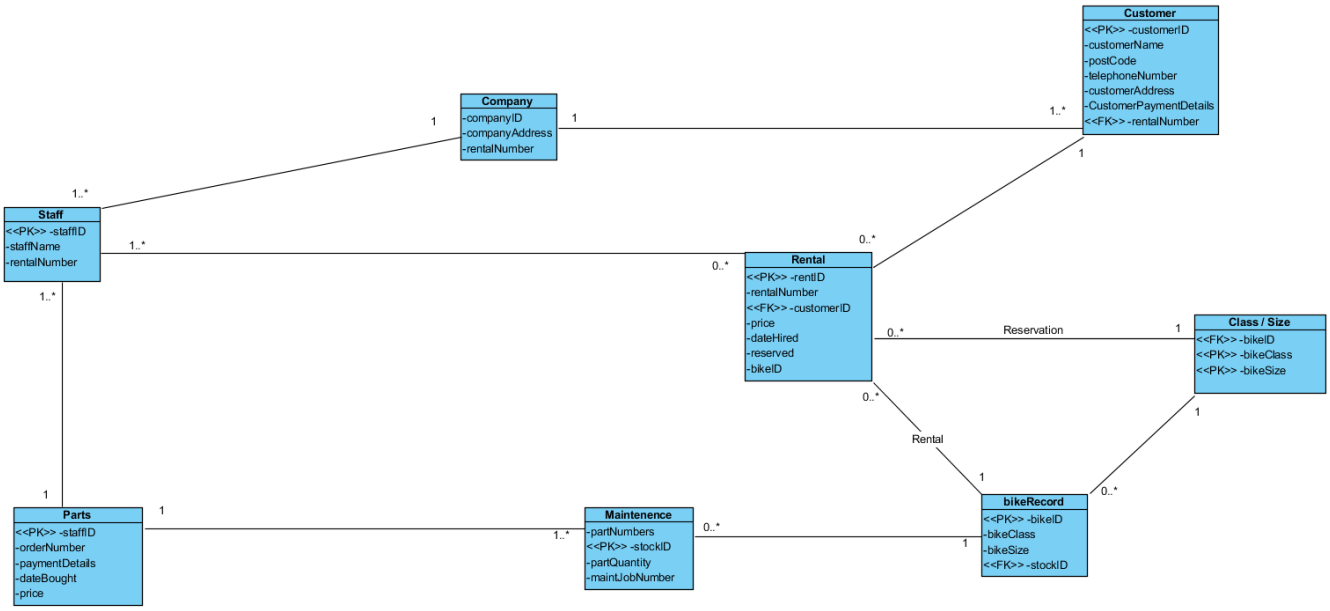
UNF	1NF	2NF	3NF			
bikeNumber	bikeNumber	bikeNumber	bikeNumber		BIKE	
bike	bike	bike	bike	>	bikeNumber	
bikeType	bikeType	bikeType	bikeType			
bikeSize	bikeSize	bikeSize	bikeSize			
rentDate						
timeOut	bikeNumber	bikeNumber	bikeNumber		RENT	
timeDue	rentDate	rentDate	rentDate		bikeNumber	*
actualTimeBack	timeOut	timeOut	timeOut	>	rentDate	
customerName	timeDue	timeDue	timeDue		timeOut	*
customerPostCode	actualTimeBack	actualTimeBack	actualTimeBack			
customerTelephone	customerName	customerName	customerName*			
customerAddress	customerPostCode	customerPostCode				
amountPaid	customerTelephone	customerTelephone				
	customerAddress	customerAddress	customerName			
	amountPaid	amountPaid	customerPostCode	>	CUSTOMER	1
			customerTelephone		customerName	
			customerAddress			
			amountPaid			

UNF	1NF	2NF	3NF		
actualTimeBack	bikeNo	bikeNo	bikeNo		
amountPaid	bike	bike	rentDate	>	RENTAL
bike	bikeModel	bikeModel	timeOut		bikeNo *
bikeClass	bikePurchasePrice	bikePurchasePrice	timeDue		rentDate
bikeManf	bikePurchaseDate	bikePurchaseDate	actualTimeBack		timeOut *
bikeManfAddress	bikeClass	bikeClass	customerName*		customerName*
bikeManfpCode	bikeSize	bikeSize			
bikeManTel	bikeManf	bikeManf	customerName		
bikeModel	bikeManfAddress	bikeManfAddress	customerPostCode		
bikeNo	bikeManfpCode	bikeManfpCode	customerTelephone	>	CUSTOMER INFO 1
bikePurchaseDate	bikeManTel	bikeManTel	customerAddress		customerName
bikePurchasePrice	saleDate	saleDate	amountPaid		
bikeSize	salePrice	salePrice			
customerAddress	dealerName	dealerName	bikeNo		
customerName	dealerAdd	dealerAdd	bike		
customerPostCode	dealerTel	dealerTel	bikeModel		
customerTelephone			bikePurchasePrice		
dealerAdd			bikePurchaseDate	>	BIKE *
dealerName	bikeNo	bikeNo	bikeClass		bikeNo
dealerTel	rentDate	rentDate	bikeSize		bikeManf* 1
maintAction	timeOut	timeOut	bikeManf*		dealerName* *
maintActionDate	timeDue	timeDue	saleDate		
maintFaultDate	actualTimeBack	actualTimeBack	salePrice		1
maintFaultDet	customerName	customerName	dealerName*		
maintRefNo	customerPostCode	customerPostCode			
rentDate	customerTelephone	customerTelephone	bikeManf		
saleDate	customerAddress	customerAddress	bikeManfAddress	>	MANAFACTURER *
salePrice	amountPaid	amountPaid	bikeManfpCode		bikeManf
timeDue			bikeManTel		
timeOut	bikeNo	bikeNo			
	maintRefNo	maintRefNo	dealerName		
	maintFaultDate	maintFaultDate	dealerAdd	>	DEALER
	maintAction	maintAction	dealerTel		dealerName 1
	maintActionDate	maintActionDate			
			bikeNo		
			maintRefNo		
			maintFaultDate	>	MAINTENANCE *
			maintAction		bikeNo
			maintActionDate		maintRefNo



Finalised Group ERD (top-down and bottom-up) covering the complete system.





Commentary Explaining decisions made when creating the ERD for the finalised system.

When making the finalised entity relationship diagram, it was decided that one of the most important aspects would be the entities and therefore these became the prime focus. The entities that were used included staff, customer, parts, rental, maintenance, bike record and class/size. It was then also decided which entities were the primary keys and would therefore be the main component of a specific table, with this the foreign keys were also decided. By creating the bottom up entity relationship diagram, using normalisation, the fields that would relate to each entity were created.

In this section, all members learnt how to normalise the data that had been created, normalisation allowed organisation of information and would prevent the system from having redundant data. Members also learnt that having many to many relationships meant that another entity is needed as a hidden entity would be concealed. Therefore, identifying these entities was vital in order for the system to be manageable.

Conclusion

This report was set out to design a system that would simplify and modernise the current system at Ray's Rentals, not just for the company but also for its customers.

The report concludes that using specific techniques such as creating entity relationship diagrams and relational data analysis diagrams will allow an increase in efficiency for the company "Ray's Rentals" at the developmental stage. Creating these data diagrams improves the company's total revenue once the system has been implemented and reduces their costs both for developing the system and also long term. Furthermore, the process of normalisation has been critical, as this results in a prevention of unnecessary and duplicate data, saving staff time.

Overall, it is evident that Ray's Rentals will benefit greatly from the full implementation of the system from its ease of use to the increase in business that should occur due to the relative easiness of renting a bike.

Appendix

Log Entries From Piazza

followup discussions *for lingering questions and comments*

Resolved Unresolved

Madaline Jones 24 days ago
07/11/2017

Ryan and Madaline worked on the third task and began our UCD specification's. We also messaged Karan and Michelle to begin theirs as they were unable to attend the meeting today, we were not given a reason for this.

Follow up:
Once question one has been completed, we will begin working on question two.

Reply to this followup discussion

Resolved Unresolved

Madaline Jones 16 days ago
14/11/2017

Today we all met up and gave out tasks for question 3:
Ryan - Stock management
Michelle - Making a reservation
Madaline - Customer payment
Karan - Bike rental records

Ryan and Madaline worked on the first question (putting it into visualparadige) and Michelle began writing up a commentary for it.
Michelle didn't have her laptop with her and this slowed down the process of us working together on the first question.
Michelle and Karan began their UCD specifications.
We visited a tutor to clear up a query we had about a question.

Follow up: Have UCD specifications completed and complete the Top Down ERD for the new system.

Reply to this followup discussion

Resolved Unresolved

Madaline Jones 9 days ago
21/11/2017
Ryan - Completed UCD specification and created ERD for stock management.
Madaline - Completed UCD specification and created ERD for customer payment
Michelle - Created ERD for making a reservation
Karan - Completed the ERD and started the Top Down System.

We began the ERD top down for the system:
Ryan - 30%
Karan - 30%

Resolved Unresolved

Madaline Jones 24 days ago
07/11/2017

Ryan and Madaline worked on the third task and began our UCD specification's. We also messaged Karan and Michelle to begin theirs as they were unable to attend the meeting today, we were not given a reason for this.

Follow up:
Once question one has been completed, we will begin working on question two.

Reply to this followup discussion

Resolved Unresolved

Madaline Jones 16 days ago
14/11/2017

Today we all met up and gave out tasks for question 3:
Ryan - Stock management
Michelle - Making a reservation
Madaline - Customer payment
Karan - Bike rental records

Ryan and Madaline worked on the first question (putting it into visualparadige) and Michelle began writing up a commentary for it.
Michelle didn't have her laptop with her and this slowed down the process of us working together on the first question.
Michelle and Karan began their UCD specifications.
We visited a tutor to clear up a query we had about a question.

Follow up: Have UCD specifications completed and complete the Top Down ERD for the new system.

Reply to this followup discussion

Resolved Unresolved

Madaline Jones 9 days ago
21/11/2017
Ryan - Completed UCD specification and created ERD for stock management.
Madaline - Completed UCD specification and created ERD for customer payment
Michelle - Created ERD for making a reservation
Karan - Completed the ERD and started the Top Down System.

We began the ERD top down for the system:
Ryan - 30%
Karan - 30%

Resolved Unresolved

Madaline Jones 1 day ago
28/11/2017
After seeing a tutor we had been told that our RDA was in the incorrect format for question 5. We were told the wrong instructions by our tutor for doing this RDA last week, because of this it took us more time as we had to redo the question.

follow up:
Complete question 6 and 7

Reply to this followup discussion

Resolved Unresolved

Madaline Jones 1 day ago
29/11/2017
Ryan, Madaline and Karan completed question 6 with a little help from Michelle.

Follow up:
Complete question 7, introduction, executive summary and conclusion

Reply to this followup discussion

Resolved Unresolved

Madaline Jones 6 hours ago
30/11/2017
We reviewed question 6 after we noticed a many to many relationship within the ERD top down diagram
We completed question 7 worked on question 2 to complete it in full. We also began the introduction, conclusion and executive summary..
Karan worked closely on the conclusion.

Ryan put the final report together by adding in all the diagrams and completing the layout. He also added more detail to the introduction, executive summary and conclusion.

Follow up:
Look over the final report to check over everything
Hand in by Ryan tomorrow

Reply to this followup discussion

Madaline Jones 9 days ago
21/11/2017
Ryan - Completed UCD specification and created ERD for stock management.
Madaline - Completed UCD specification and created ERD for customer payment
Michelle - Created ERD for making a reservation
Karan - Completed the ERD and started the Top Down System.

We began the ERD top down for the system:
Ryan - 30%
Karan - 30%
Madaline - 30%
Michelle - 10% (Due to her having leave half way through the session.)

Follow up:
Complete question 5 and 6, go back to the first question to look over the MoSCoW system and add this to the first question.

Reply to this followup discussion

Resolved Unresolved

Madaline Jones 7 days ago
23/11/2017
After looking over the work we found that question 1 needed more detail, Ryan and Madaline completed the UCD and added the MoSCoW system,
Karan and Michelle looked over how to complete RDA's and shared the video they found with Ryan and Madaline to watch before starting question 5. We need to contact a tutor before completing this question to.

We started question 6 and completed the top down ERD of our system

follow up:
Visit a tutor to look over question 5 and make sure we are along the right lines, then complete it.
Finish question 6 by completing the bottom up ERD of the system
Complete question 7
Look over each of the questions we have completed, see if anything can be added/changed.

Reply to this followup discussion

Resolved Unresolved

Madaline Jones 1 day ago
28/11/2017
After seeing a tutor we had been told that our RDA was in the incorrect format for question 5. We were told the wrong instructions by our tutor for doing this RDA last week, because of this it took us more time as we had to redo the question.

follow up:
Complete question 6 and 7

Reply to this followup discussion

verage Response Time: Special Mentions: Online Now | This

N/A There are no special mentions at this time. 7 |

Part 3

Introduction

Background

Rays Rentals is a small bike rental shop that used an outdated paper records system. Previously we discussed how implementing an up to date database system would increase the efficiency and how this would in turn lead to higher productivity and revenue. The database system has been designed to suit the needs of Ray's rentals and to generally modernize the business.

Aims

This report aims to create the database that has been designed previously. The database will be created in

ORACLE SQL, database tables will be created, data will be inserted into the table using INSERT statements, SELECT statements will also be used to make queries, these queries will be designed for practical use in Ray's rentals, queries that would often need to be used that would make tasks much easier for staff.

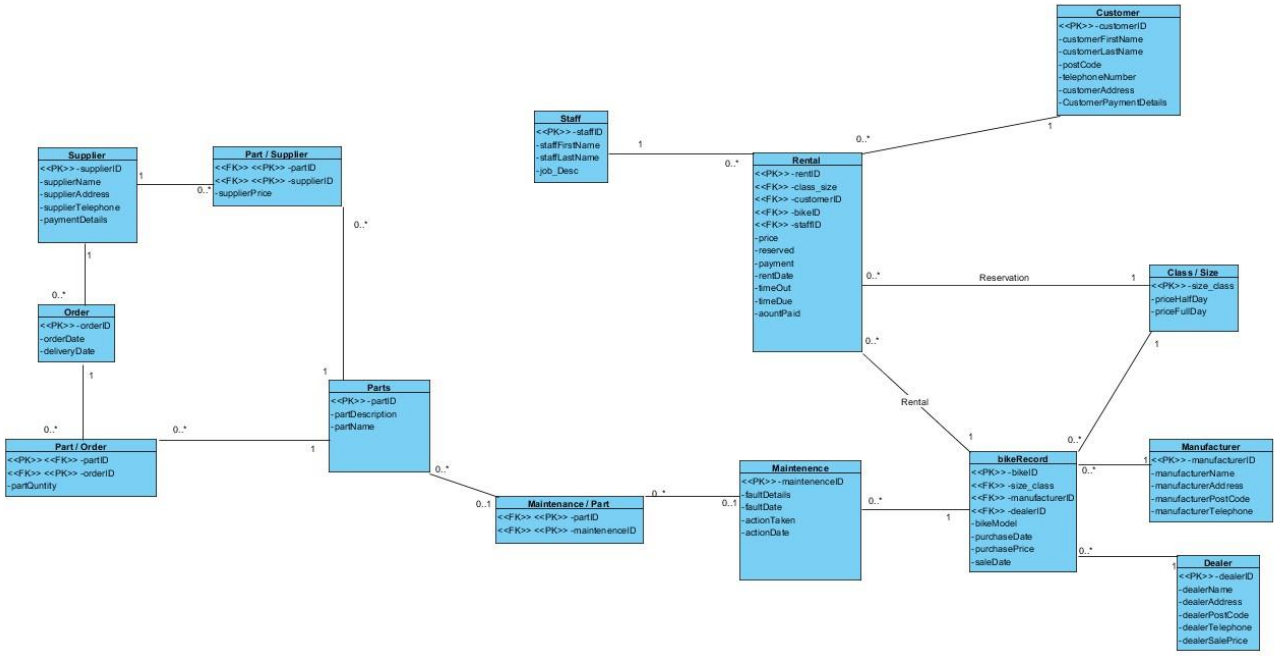
Executive summary

Contained within this report is the creation and implementation of the previously designed new database system for use at Ray's rentals. As mentioned in previous reports, Ray's rentals is a small cycle hire shop. The shop owns roughly 150 bikes and are available to rent, a database design has been previously created that will be implemented in this report.

For the first part of this report, some changes were made to the design of the database to make it physically possible for the design to function correctly when implemented into ORACLE SQL.

The created database will make the business much more functional and reliable; the problems previously outlined will be removed once implemented. Queries have also been created that search the database and present data to staff in an easy to read and understandable way.

Question 1 – Amended ERD



Question 2 – Data Dictionaries

Ryan

Part / Supplier -

Attribute Name	Key Type	FK Table	FK Column	Data Type	Length	Constraint
partID	PK FK	PARTS	PartID	VARCHAR2	10	NOT NULL
supplierID	PK FK	SUPPLIER	supplierID	VARCHAR2	10	NOT NULL
supplierPrice				NUMBER	5	NOT NULL

Maintenance -

Attribute Name	Key Type	FK Table	FK Column	Data Type	Length	Constraint
MaintenanceID	PK	Maintenance	MaintenanceID	VARCHAR2	10	NOT NULL

		/Part				
faultDetails				VARCHAR2	40	NOT NULL
faultDate				DATE	8	
actionTaken				VARCHAR2	40	NOT NULL
actionDate				DATE	8	NOT NULL

Customer -

Attribute Name	Key Type	FK Table	FK Column	Data Type	Length	Constraint
customerID	PK	Rental	customerID	VARCHAR2	10	NOT NULL
customerFirstName				VARCHAR2	15	NOT NULL
customerLastName				VARCHAR2	20	NOT NULL
postcode				VARCHAR2	8	
telephoneNumber				VARCHAR2	12	NOT NULL
customerAddress				VARCHAR2	20	
customerPaymentDetails				VARCHAR2	30	NOT NULL

Michelle

Order -

Attribute Name	Key Type	FK Table	FK Column	Data Type	Length	Constraint
orderID	PK	Part / Order	orderID	VARCHAR2	10	NOT NULL
orderDate				DATE	8	NOT NULL
deliveryDate				DATE	8	

Maintenance / Part -

Attribute Name	Key Type	FK Table	FK Column	Data Type	Length	Constraint
partID	PK FK	Parts	partID	VARCHAR2	10	NOT NULL
maintenenceID	PK FK	Maintenance	MaintenanceID	VARCHAR2	10	NOT NULL

Supplier -

Attribute Name	Key Type	FK Table	FK Column	Data Type	Length	Constraint
supplierID	PK	Part / Supplier	supplierID	VARCHAR2	10	NOT NULL
supplierName				VARCHAR2	15	NOT NULL
supplierAddress				VARCHAR2	20	
supplierNumber				VARCHAR2	12	NOT NULL
paymentDetails				VARCHAR2	30	NOT NULL

Karan

Bike Record -

ATTRIBUTE NAME	KEY TYPE	FK TABLE	FK COLUMN	DATA TYPE	LENGTH	CONSTRAINT
bikeID	PK	Rental	bikeID	VARCHAR2	10	
bikeModel				VARCHAR2	10	
classSizeID	FK	Class / Size	classSizeID	VARCHAR2	10	
manuID	FK	Manufacturer	manuID	VARCHAR2	10	
dealerID	FK	Dealer	dealerID	VARCHAR2	10	
purchaseDate				DATE	8 (DD/MMM/YY)	
purchasePrice				NUMBER	(6,2)	NOT NULL
saleDate				DATE	8 (DD/MMM/YY)	

Class / Size -

ATTRIBUTE NAME	KEY TYPE	FK TABLE	FK COLUMN	DATA TYPE	LENGTH	CONSTRAINT
Class_Size	PK	Rental	Class_Size	VARCHAR2	10	
priceHalfDay				NUMBER	(5,2)	NOT NULL
priceFullDay				NUMBER	(5,2)	NOT NULL

Manufacturer -

ATTRIBUTE NAME	KEY TYPE	FK TABLE	FK COLUMN	DATA TYPE	LENGTH	CONSTRAINT
manufacturerID	PK	bikeRecord	manufacturerID	VARCHAR2	10	
manufacturerName				VARCHAR2	30	NOT NULL
manufacturerAddress				VARCHAR2	100	
manufacturerPostcode				VARCHAR2	10	NOT NULL
manufacturerTelephone				VARCHAR2	15	

Dealer -

ATTRIBUTE NAME	KEY TYPE	FK TABLE	FK COLUMN	DATA TYPE	LENGTH	CONSTRAINT
dealerID	PK	Dealer	dealerID	VARCHAR2	10	
dealerName				VARCHAR2	30	NOT NULL
dealerAddress				VARCHAR2	100	
dealerPostCode				VARCHAR2	10	NOT NULL
dealerTelephone				VARCHAR2	15	
dealerSalePrice				NUMBER	(6,2)	NOT NULL

Parts -

ATTRIBUTE NAME	KEY TYPE	FK TABLE	FK COLUMN	DATA TYPE	LENGTH	CONSTRAINT
partID	PK			VARCHAR2	10	
partDescription				VARCHAR2	100	
partName				VARCHAR2	30	NOT NULL

Maddie

Part / Order -

Attribute Name	Key Type	FK Table	FK Column	Data Type	Length	Constraint
partID	PK FK	Parts	partID	VARCHAR2	10	
orderID	PK FK	Order	orderID	VARCHAR2	10	
partQuantity				NUMBER	(10,0)	

Staff -

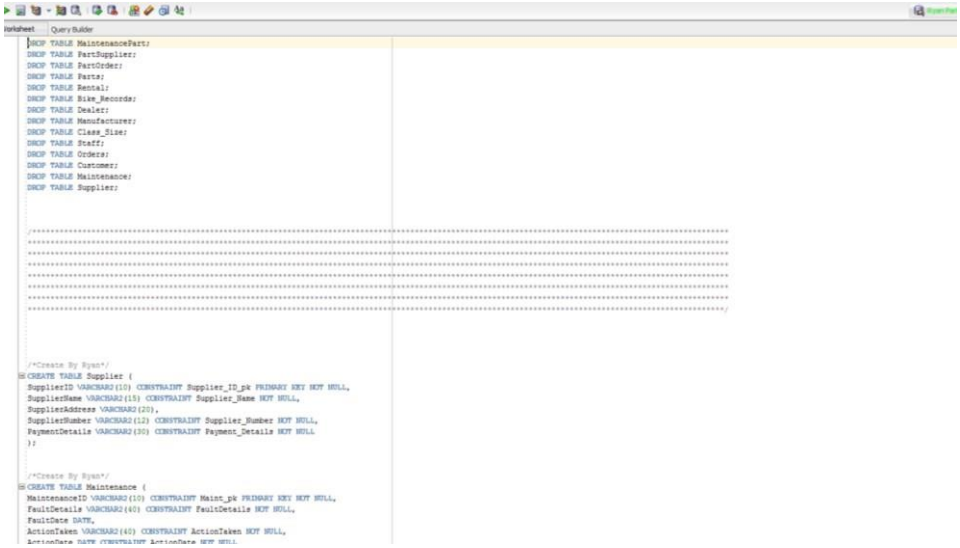
Attribute Name	Key Type	FK Table	FK Column	Data Type	Length	Constraint
staffID	PK	Rental	staffID	VARCHAR2	10	NOT NULL
StaffFirstName				VARCHAR2	30	NOT NULL
StaffLastName				VARCHAR2	30	NOT NULL
Staff_Desc				VARCHAR2	20	

Rental -

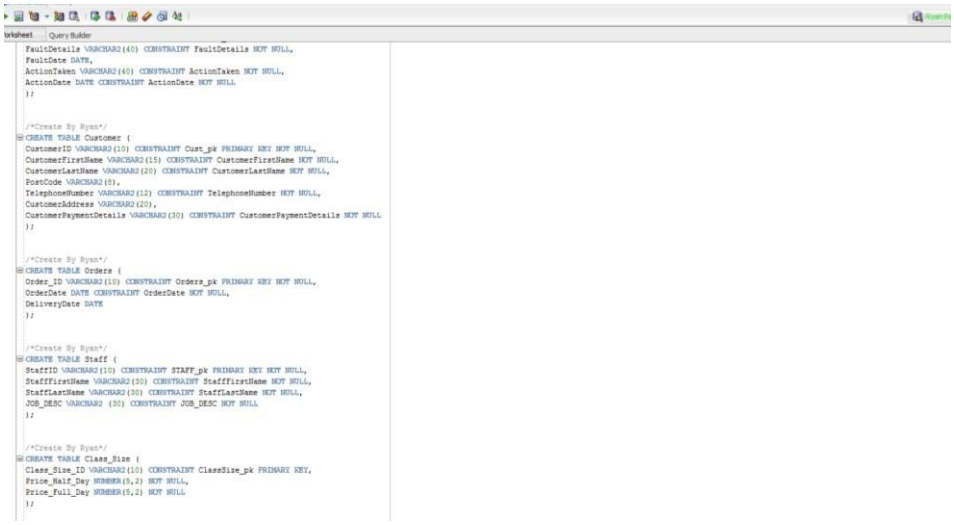
Attribute Name	Key Type	FK Table	FK Column	Data Type	Length	Constraint
rentalID	PK			VARCHAR2	10	NOT NULL
Class_size	FK	Class/size	Class_Size	VARCHAR2	50	NOT NULL
customerID	FK	Customer	customerID	VARCHAR2	30	NOT NULL
bikeID	FK	bikeRecord	bikeID	VARCHAR2	10	NOT NULL
staffID	FK	staff	staffID	VARCHAR2	10	NOT NULL
price				NUMBER	10	NOT NULL
Reserved				VARCHAR2	3	
Payment				VARCHAR2	3	NOT NULL
rentDate				DATE	8 (DD/MON/YY)	
timeOut				TIME	Time/pm/am	
timeDue				TIME	Time/pm/am	
customerName				VARCHAR2	30	NOT NULL
customerAddress				VARCHAR2	30	
CustomerPostCode				VARCHAR2	7	

customerTelephone				VARCHAR2	11	NOT NULL
amountPaid				NUMBER	(5,2)	

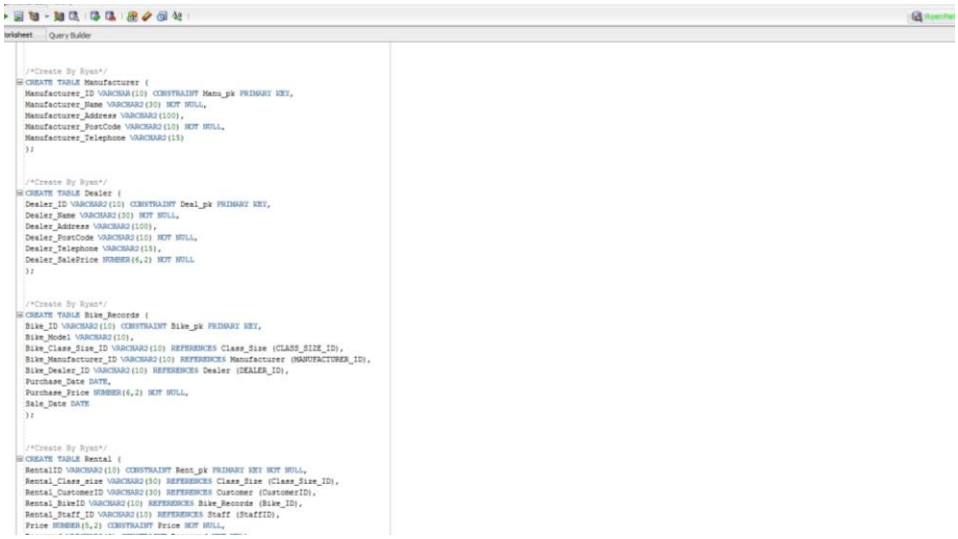
Question 3 – Data dictionaries implemented into ORACLE



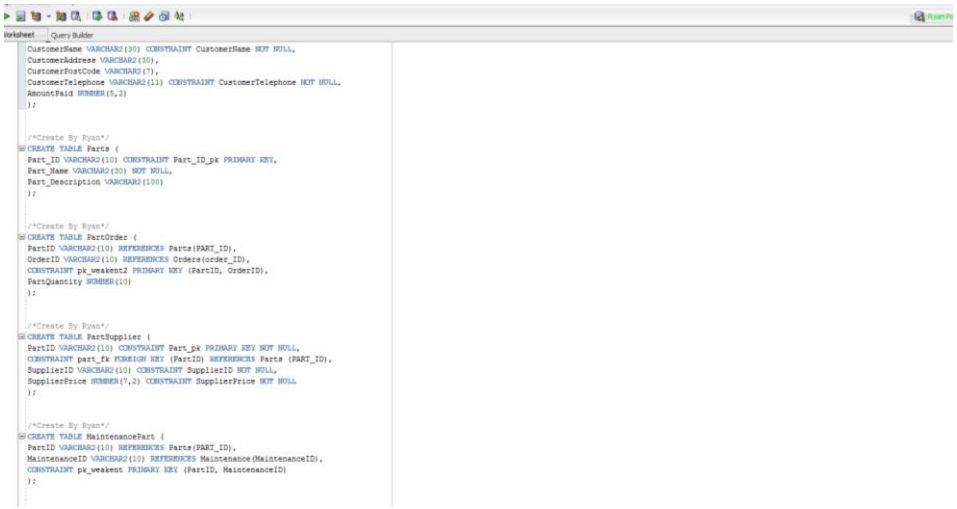
```
DROP TABLE MaintenancePart;
DROP TABLE PartSupplier;
DROP TABLE PartOrder;
DROP TABLE Parts;
DROP TABLE Rental;
DROP TABLE Bike_Records;
DROP TABLE Dealer;
DROP TABLE Manufacturer;
DROP TABLE Class_Size;
DROP TABLE Staff;
DROP TABLE Orders;
DROP TABLE Customer;
DROP TABLE Maintenance;
DROP TABLE Supplier;

/*Create By Ryan*/
CREATE TABLE Supplier (
SupplierID VARCHAR2(10) CONSTRAINT Supplier_ID_pk PRIMARY KEY NOT NULL,
SupplierName VARCHAR2(15) CONSTRAINT Supplier_Name NOT NULL,
SupplierAddress VARCHAR2(20),
SupplierNumber VARCHAR2(12) CONSTRAINT Supplier_Number NOT NULL,
PaymentDetails VARCHAR2(30) CONSTRAINT Payment_Details NOT NULL
);

/*Create By Ryan*/
CREATE TABLE Maintenance (
MaintenanceID VARCHAR2(10) CONSTRAINT Maint_pk PRIMARY KEY NOT NULL,
FaultDetails VARCHAR2(40) CONSTRAINT FaultDetails NOT NULL,
FaultDate DATE,
ActionTaken VARCHAR2(40) CONSTRAINT ActionTaken NOT NULL,
```



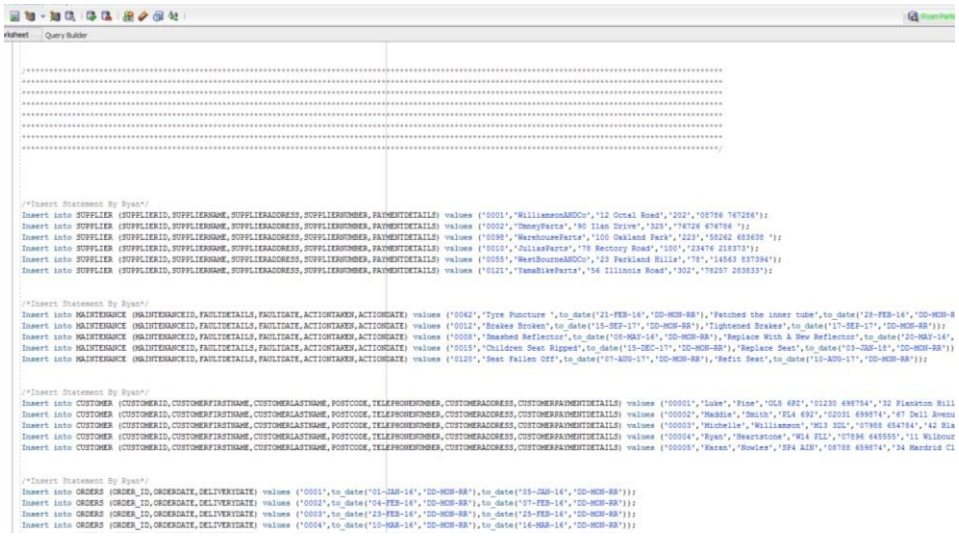
```
FaultDetails VARCHAR2(40) CONSTRAINT FaultDetails NOT NULL,
FaultDate DATE,
ActionTaken VARCHAR2(40) CONSTRAINT ActionTaken NOT NULL,
ActionDate DATE CONSTRAINT ActionDate NOT NULL
);

/*Create By Ryan*/
CREATE TABLE Customer (
CustomerID VARCHAR2(10) CONSTRAINT Cust_pk PRIMARY KEY NOT NULL,
CustomerFirstName VARCHAR2(15) CONSTRAINT CustomerFirstName NOT NULL,
CustomerLastName VARCHAR2(20) CONSTRAINT CustomerLastName NOT NULL,
PostCode VARCHAR2(8),
TelephoneNumber VARCHAR2(12) CONSTRAINT TelephoneNumber NOT NULL,
CustomerAddress VARCHAR2(20),
CustomerPaymentDetails VARCHAR2(30) CONSTRAINT CustomerPaymentDetails NOT NULL
);

/*Create By Ryan*/
CREATE TABLE Orders (
Order_ID VARCHAR2(10) CONSTRAINT Orders_pk PRIMARY KEY NOT NULL,
OrderDate DATE CONSTRAINT OrderDate NOT NULL,
DeliveryDate DATE
);

/*Create By Ryan*/
CREATE TABLE Staff (
StaffID VARCHAR2(10) CONSTRAINT STAFF_pk PRIMARY KEY NOT NULL,
StaffFirstName VARCHAR2(30) CONSTRAINT StaffFirstName NOT NULL,
StaffLastName VARCHAR2(30) CONSTRAINT StaffLastName NOT NULL,
JOB_DESC VARCHAR2 (30) CONSTRAINT JOB_DESC NOT NULL
);

/*Create By Ryan*/
CREATE TABLE Class_Size (
Class_Size_ID VARCHAR2(10) CONSTRAINT ClassSize_pk PRIMARY KEY,
Price_Half_Day NUMBER(5,2) NOT NULL,
Price_Full_Day NUMBER(5,2) NOT NULL
);
```



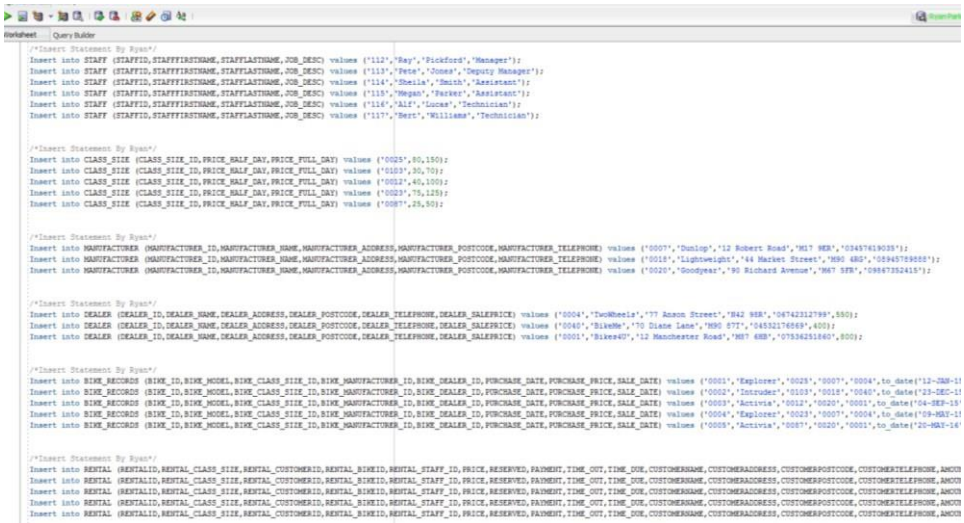
```
/*Create By Ryan*/
CREATE TABLE Manufacturer (
Manufacturer_ID VARCHAR(10) CONSTRAINT Manu_pk PRIMARY KEY,
Manufacturer_Name VARCHAR2(30) NOT NULL,
Manufacturer_Address VARCHAR2(100),
Manufacturer_PostCode VARCHAR2(10) NOT NULL,
Manufacturer_Telephone VARCHAR2(15)
);

/*Create By Ryan*/
CREATE TABLE Dealer (
Dealer_ID VARCHAR2(10) CONSTRAINT Deal_pk PRIMARY KEY,
Dealer_Name VARCHAR2(30) NOT NULL,
Dealer_Address VARCHAR2(100),
Dealer_PostCode VARCHAR2(10) NOT NULL,
Dealer_Telephone VARCHAR2(15),
Dealer_SalePrice NUMBER(6,2) NOT NULL
);

/*Create By Ryan*/
CREATE TABLE Bike_Records (
Bike_ID VARCHAR2(10) CONSTRAINT Bike_pk PRIMARY KEY,
Bike_Model VARCHAR2(10),
Bike_Class_Size_ID VARCHAR2(10) REFERENCES Class_Size (CLASS_SIZE_ID),
Bike_Manufacturer_ID VARCHAR2(10) REFERENCES Manufacturer (MANUFACTURER_ID),
Bike_Dealer_ID VARCHAR2(10) REFERENCES Dealer (DEALER_ID),
Purchase_Date DATE,
Purchase_Price NUMBER(6,2) NOT NULL,
Sale_Date DATE
);

/*Create By Ryan*/
CREATE TABLE Rental (
RentalID VARCHAR2(10) CONSTRAINT Rent_pk PRIMARY KEY NOT NULL,
Rental_Class_size VARCHAR2(50) REFERENCES Class_Size (Class_Size_ID),
Rental_CustomerID VARCHAR2(30) REFERENCES Customer (CustomerID),
Rental_BikeID VARCHAR2(10) REFERENCES Bike_Records (Bike_ID),
Rental_Staff_ID VARCHAR2(10) REFERENCES Staff (StaffID),
Price NUMBER(5,2) CONSTRAINT Price NOT NULL,
```



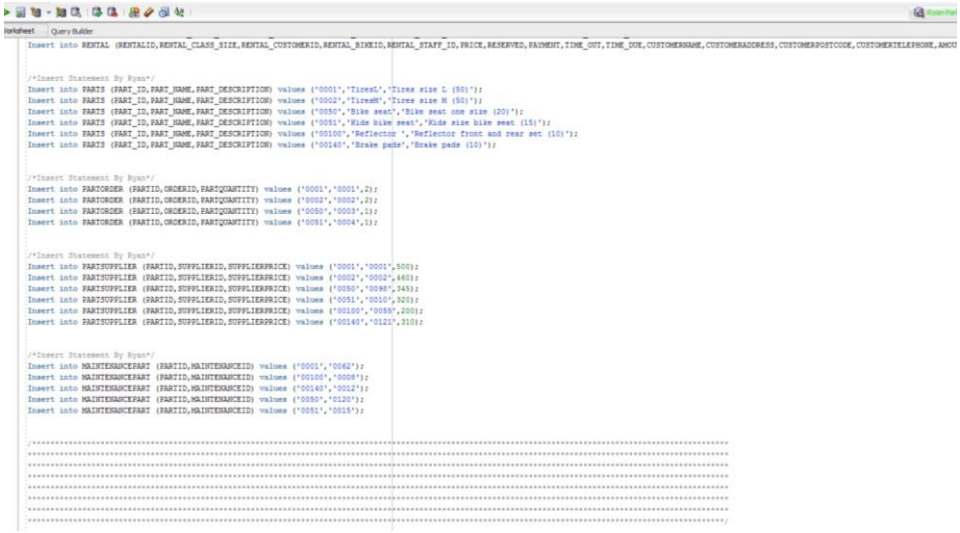
```
CustomerName VARCHAR2(30) CONSTRAINT CustomerName NOT NULL,
CustomerAddress VARCHAR2(30),
CustomerPostCode VARCHAR2(7),
CustomerTelephone VARCHAR2(11) CONSTRAINT CustomerTelephone NOT NULL,
AmountPaid NUMBER(5,2)
);


/*Create By Ryan*/
CREATE TABLE Parts (
Part_ID VARCHAR2(10) CONSTRAINT Part_ID_pk PRIMARY KEY,
Part_Name VARCHAR2(30) NOT NULL,
Part_Description VARCHAR2(100)
);


/*Create By Ryan*/
CREATE TABLE PartOrder (
PartID VARCHAR2(10) REFERENCES Parts(PART_ID),
OrderID VARCHAR2(10) REFERENCES Orders(order_ID),
CONSTRAINT pk_weakent2 PRIMARY KEY (PartID, OrderID),
PartQuantity NUMBER(10)
);


/*Create By Ryan*/
CREATE TABLE PartSupplier (
PartID VARCHAR2(10) CONSTRAINT Part_pk PRIMARY KEY NOT NULL,
CONSTRAINT part_fk FOREIGN KEY (PartID) REFERENCES Parts (PART_ID),
SupplierID VARCHAR2(10) CONSTRAINT SupplierID NOT NULL,
SupplierPrice NUMBER(7,2) CONSTRAINT SupplierPrice NOT NULL
);


/*Create By Ryan*/
CREATE TABLE MaintenancePart (
PartID VARCHAR2(10) REFERENCES Parts(PART_ID),
MaintenanceID VARCHAR2(10) REFERENCES Maintenance(MaintenanceID),
CONSTRAINT pk_weakent PRIMARY KEY (PartID, MaintenanceID)
);
```



```
/**********************************************************************************************************************************
**********************************************************************************************************************************
**********************************************************************************************************************************
**********************************************************************************************************************************
**********************************************************************************************************************************
**********************************************************************************************************************************
**********************************************************************************************************************************
**********************************************************************************************************************************/



/*Insert Statement By Ryan*/
Insert into SUPPLIER (SUPPLIERID,SUPPLIERNAME,SUPPLIERADDRESS,SUPPLIERNUMBER,PAYMENTDETAILS) values ('0001','WilliamsonAMDCo','12 Octal Road','202','08786 767256');
Insert into SUPPLIER (SUPPLIERID,SUPPLIERNAME,SUPPLIERADDRESS,SUPPLIERNUMBER,PAYMENTDETAILS) values ('0002','UmneyParts','90 Ilan Drive','325','76726 676786 ');
Insert into SUPPLIER (SUPPLIERID,SUPPLIERNAME,SUPPLIERADDRESS,SUPPLIERNUMBER,PAYMENTDETAILS) values ('0098','WarehouseParts','100 Oakland Park','223','58262 683638 ');
Insert into SUPPLIER (SUPPLIERID,SUPPLIERNAME,SUPPLIERADDRESS,SUPPLIERNUMBER,PAYMENTDETAILS) values ('0010','JuliasParts','78 Rectory Road','100','23476 218373');
Insert into SUPPLIER (SUPPLIERID,SUPPLIERNAME,SUPPLIERADDRESS,SUPPLIERNUMBER,PAYMENTDETAILS) values ('0055','WestBourneAMDCo','23 Parkland Hills','78','14563 837394');
Insert into SUPPLIER (SUPPLIERID,SUPPLIERNAME,SUPPLIERADDRESS,SUPPLIERNUMBER,PAYMENTDETAILS) values ('0121','YamaBikeParts','56 Illinois Road','302','78257 283833');


/*Insert Statement By Ryan*/
Insert into MAINTENANCE (MAINTENANCEID,FAULTDETAILS,FAULTDATE,ACTIONTAKEN,ACTIONDATE) values ('0062','Tyre Puncture ',to_date('21-FEB-16','DD-MON-RR'),'Patched the inner tube',to_date('28-FEB-16','DD-MON-R
Insert into MAINTENANCE (MAINTENANCEID,FAULTDETAILS,FAULTDATE,ACTIONTAKEN,ACTIONDATE) values ('0012','Brakes Broken',to_date('15-SEP-17','DD-MON-RR'),'Tightened Brakes',to_date('17-SEP-17','DD-MON-RR'));
Insert into MAINTENANCE (MAINTENANCEID,FAULTDETAILS,FAULTDATE,ACTIONTAKEN,ACTIONDATE) values ('0008','Smashed Reflector',to_date('08-MAY-16','DD-MON-RR'),'Replace With A New Reflector',to_date('20-MAY-16',
Insert into MAINTENANCE (MAINTENANCEID,FAULTDETAILS,FAULTDATE,ACTIONTAKEN,ACTIONDATE) values ('0015','Children Seat Ripped',to_date('15-DEC-17','DD-MON-RR'),'Replace Seat',to_date('03-JAN-18','DD-MON-RR'))
Insert into MAINTENANCE (MAINTENANCEID,FAULTDETAILS,FAULTDATE,ACTIONTAKEN,ACTIONDATE) values ('0120','Seat Fallen Off',to_date('07-AUG-17','DD-MON-RR'),'Refit Seat',to_date('10-AUG-17','DD-MON-RR'));


/*Insert Statement By Ryan*/
Insert into CUSTOMER (CUSTOMERID,CUSTOMERFIRSTNAME,CUSTOMERLASTNAME,POSTCODE,TELEPHONENUMBER,CUSTOMERADDRESS,CUSTOMERPAYMENTDETAILS) values ('00001','Luke','Finn','GL5 4PZ','01230 698754','32 Plankton Hill
Insert into CUSTOMER (CUSTOMERID,CUSTOMERFIRSTNAME,CUSTOMERLASTNAME,POSTCODE,TELEPHONENUMBER,CUSTOMERADDRESS,CUSTOMERPAYMENTDETAILS) values ('00002','Maddie','Smith','EL4 4PQ','02031 698754','67 Dell Avenu
Insert into CUSTOMER (CUSTOMERID,CUSTOMERFIRSTNAME,CUSTOMERLASTNAME,POSTCODE,TELEPHONENUMBER,CUSTOMERADDRESS,CUSTOMERPAYMENTDETAILS) values ('00003','Michelle','Williamson','M13 3DL','07988 654784','42 Bla
Insert into CUSTOMER (CUSTOMERID,CUSTOMERFIRSTNAME,CUSTOMERLASTNAME,POSTCODE,TELEPHONENUMBER,CUSTOMERADDRESS,CUSTOMERPAYMENTDETAILS) values ('00004','Ryan','Heartstone','W14 9LL','07896 645555','11 Wilbour
Insert into CUSTOMER (CUSTOMERID,CUSTOMERFIRSTNAME,CUSTOMERLASTNAME,POSTCODE,TELEPHONENUMBER,CUSTOMERADDRESS,CUSTOMERPAYMENTDETAILS) values ('00005','Karan','Bowles','SP4 4JH','08788 659874','34 Mardrid Cl


/*Insert Statement By Ryan*/
Insert into ORDERS (ORDER_ID,ORDERDATE,DELIVERYDATE) values ('0001',to_date('01-JAN-16','DD-MON-RR'),to_date('05-JAN-16','DD-MON-RR'));
Insert into ORDERS (ORDER_ID,ORDERDATE,DELIVERYDATE) values ('0002',to_date('04-FEB-16','DD-MON-RR'),to_date('07-FEB-16','DD-MON-RR'));
Insert into ORDERS (ORDER_ID,ORDERDATE,DELIVERYDATE) values ('0003',to_date('23-FEB-16','DD-MON-RR'),to_date('25-FEB-16','DD-MON-RR'));
Insert into ORDERS (ORDER_ID,ORDERDATE,DELIVERYDATE) values ('0004',to_date('10-MAR-16','DD-MON-RR'),to_date('14-MAR-16','DD-MON-RR'));
```



```
/*Insert Statement By Ryan*/
Insert into STAFF (STAFFID,STAFFFIRSTNAME,STAFFLASTNAME,JOB_DESC) values ('112','Ray','Pickford','Manager');
Insert into STAFF (STAFFID,STAFFFIRSTNAME,STAFFLASTNAME,JOB_DESC) values ('113','Pete','Jones','Deputy Manager');
Insert into STAFF (STAFFID,STAFFFIRSTNAME,STAFFLASTNAME,JOB_DESC) values ('114','Sheila','Smith','Assistant');
Insert into STAFF (STAFFID,STAFFFIRSTNAME,STAFFLASTNAME,JOB_DESC) values ('115','Megan','Parker','Assistant');
Insert into STAFF (STAFFID,STAFFFIRSTNAME,STAFFLASTNAME,JOB_DESC) values ('116','Alf','Lucas','Technician');
Insert into STAFF (STAFFID,STAFFFIRSTNAME,STAFFLASTNAME,JOB_DESC) values ('117','Bert','Williams','Technician');


/*Insert Statement By Ryan*/
Insert into CLASS_SIZE (CLASS_SIZE_ID,PRICE_HALF_DAY,PRICE_FULL_DAY) values ('0025',80,150);
Insert into CLASS_SIZE (CLASS_SIZE_ID,PRICE_HALF_DAY,PRICE_FULL_DAY) values ('0103',30,70);
Insert into CLASS_SIZE (CLASS_SIZE_ID,PRICE_HALF_DAY,PRICE_FULL_DAY) values ('0012',40,100);
Insert into CLASS_SIZE (CLASS_SIZE_ID,PRICE_HALF_DAY,PRICE_FULL_DAY) values ('0023',75,125);
Insert into CLASS_SIZE (CLASS_SIZE_ID,PRICE_HALF_DAY,PRICE_FULL_DAY) values ('0087',25,50);


/*Insert Statement By Ryan*/
Insert into MANUFACTURER (MANUFACTURER_ID,MANUFACTURER_NAME,MANUFACTURER_ADDRESS,MANUFACTURER_POSTCODE,MANUFACTURER_TELEPHONE) values ('0007','Dunlop','12 Robert Road','M17 9ER','03457419035');
Insert into MANUFACTURER (MANUFACTURER_ID,MANUFACTURER_NAME,MANUFACTURER_ADDRESS,MANUFACTURER_POSTCODE,MANUFACTURER_TELEPHONE) values ('0018','Lightweight','44 Market Street','M90 4RS','08945789658');
Insert into MANUFACTURER (MANUFACTURER_ID,MANUFACTURER_NAME,MANUFACTURER_ADDRESS,MANUFACTURER_POSTCODE,MANUFACTURER_TELEPHONE) values ('0020','Goodyear','90 Richard Avenue','M67 5FR','09567352415');


/*Insert Statement By Ryan*/
Insert into DEALER (DEALER_ID,DEALER_NAME,DEALER_ADDRESS,DEALER_POSTCODE,DEALER_TELEPHONE,DEALER_SALEPRICE) values ('0004','TwoWheels','77 Anson Street','B42 9ER','06742312799',550);
Insert into DEALER (DEALER_ID,DEALER_NAME,DEALER_ADDRESS,DEALER_POSTCODE,DEALER_TELEPHONE,DEALER_SALEPRICE) values ('0040','BikeMe','70 Diane Lane','M90 6TT','04532176569',400);
Insert into DEALER (DEALER_ID,DEALER_NAME,DEALER_ADDRESS,DEALER_POSTCODE,DEALER_TELEPHONE,DEALER_SALEPRICE) values ('0001','Bikes4U','12 Manchester Road','M67 4HB','07534251840',800);


/*Insert Statement By Ryan*/
Insert into BIKE_RECORDS (BIKE_ID,BIKE_MODEL,BIKE_CLASS_SIZE_ID,BIKE_MANUFACTURER_ID,BIKE_DEALER_ID,PURCHASE_DATE,PURCHASE_PRICE,SALE_DATE) values ('0001','Explorer','0025','0007','0004',to_date('12-JAN-1
Insert into BIKE_RECORDS (BIKE_ID,BIKE_MODEL,BIKE_CLASS_SIZE_ID,BIKE_MANUFACTURER_ID,BIKE_DEALER_ID,PURCHASE_DATE,PURCHASE_PRICE,SALE_DATE) values ('0002','Intruder','0103','0018','0040',to_date('23-DEC-1
Insert into BIKE_RECORDS (BIKE_ID,BIKE_MODEL,BIKE_CLASS_SIZE_ID,BIKE_MANUFACTURER_ID,BIKE_DEALER_ID,PURCHASE_DATE,PURCHASE_PRICE,SALE_DATE) values ('0003','Activa','0012','0020','0001',to_date('04-SEP-16'
Insert into BIKE_RECORDS (BIKE_ID,BIKE_MODEL,BIKE_CLASS_SIZE_ID,BIKE_MANUFACTURER_ID,BIKE_DEALER_ID,PURCHASE_DATE,PURCHASE_PRICE,SALE_DATE) values ('0004','Explorer','0023','0007','0004',to_date('09-MAY-1
Insert into BIKE_RECORDS (BIKE_ID,BIKE_MODEL,BIKE_CLASS_SIZE_ID,BIKE_MANUFACTURER_ID,BIKE_DEALER_ID,PURCHASE_DATE,PURCHASE_PRICE,SALE_DATE) values ('0005','Activa','0087','0020','0001',to_date('20-MAY-16'


/*Insert Statement By Ryan*/
Insert into RENTAL (RENTALID,RENTAL_CLASS_SIZE,RENTAL_CUSTOMERID,RENTAL_BIKEID,RENTAL_STAFF_ID,PRICE,RESERVED,PAYMENT,TIME_OUT,TIME_DUE,CUSTOMERNAME,CUSTOMERADDRESS,CUSTOMERPOSTCODE,CUSTOMERTELEPHONE,AMOUN
Insert into RENTAL (RENTALID,RENTAL_CLASS_SIZE,RENTAL_CUSTOMERID,RENTAL_BIKEID,RENTAL_STAFF_ID,PRICE,RESERVED,PAYMENT,TIME_OUT,TIME_DUE,CUSTOMERNAME,CUSTOMERADDRESS,CUSTOMERPOSTCODE,CUSTOMERTELEPHONE,AMOUN
Insert into RENTAL (RENTALID,RENTAL_CLASS_SIZE,RENTAL_CUSTOMERID,RENTAL_BIKEID,RENTAL_STAFF_ID,PRICE,RESERVED,PAYMENT,TIME_OUT,TIME_DUE,CUSTOMERNAME,CUSTOMERADDRESS,CUSTOMERPOSTCODE,CUSTOMERTELEPHONE,AMOUN
Insert into RENTAL (RENTALID,RENTAL_CLASS_SIZE,RENTAL_CUSTOMERID,RENTAL_BIKEID,RENTAL_STAFF_ID,PRICE,RESERVED,PAYMENT,TIME_OUT,TIME_DUE,CUSTOMERNAME,CUSTOMERADDRESS,CUSTOMERPOSTCODE,CUSTOMERTELEPHONE,AMOUN
Insert into RENTAL (RENTALID,RENTAL_CLASS_SIZE,RENTAL_CUSTOMERID,RENTAL_BIKEID,RENTAL_STAFF_ID,PRICE,RESERVED,PAYMENT,TIME_OUT,TIME_DUE,CUSTOMERNAME,CUSTOMERADDRESS,CUSTOMERPOSTCODE,CUSTOMERTELEPHONE,AMOUN
```



Question 4 – SQL Queries

Michelle

SELECT *

FROM BIKE_RECORDS

ORDER BY PURCHASE_PRICE;

This will order the bikes by the cheapest price; it will allow the users of the database to see the order of the prices and may be easier to choose from if they have a budget.

SQL | All Rows Fetched: 5 in 0 seconds

	BIKE_ID	BIKE_MODEL	BIKE_CLASS_SIZE_ID	BIKE_MANUFACTURER_ID	BIKE_DEALER_ID	PURCHASE_DATE	PURCHASE_PRICE	SALE_DATE
1	0005	Activia	0087	0020	0001	20-MAY-16	250	22-MAY-16
2	0002	Intruder	0103	0018	0040	23-DEC-15	300	08-JAN-16
3	0001	Explorer	0025	0007	0004	12-JAN-15	500	20-FEB-15
4	0003	Activia	0012	0020	0001	04-SEP-15	755	30-AUG-17
5	0004	Explorer	0023	0007	0004	09-MAY-15	800	04-NOV-15

SELECT PARTS.PART_NAME, PARTS.PART_DESCRIPTION, PARTSUPPLIER.SUPPLIERID, PARTSUPPLIER.SUPPLIERPRICE

FROM PARTS

INNER JOIN PARTSUPPLIER

ON PARTS.PART_ID = PARTSUPPLIER.PARTID;

	PART_NAME	PART_DESCRIPTION	SUPPLIERID	SUPPLIERPRICE
1	TiresL	Tires size L (50)	0001	500
2	TiresM	Tires size M (50)	0002	460
3	Reflector	Reflector front and rear set (10)	0055	200
4	Brake pads	Brake pads (10)	0121	310
5	Bike seat	Bike seat one size (20)	0098	345
6	Kids bike seat	Kids size bike seat (15)	0010	320

This query uses the inner join function to join two tables which have partID as the common field.

Ryan

SELECT RENTAL_CUSTOMERID, CUSTOMERNAME,CUSTOMERADDRESS,CUSTOMERPOSTCODE, CUSTOMERTELEPHONE,(PRICE-AMOUNTPAID) "Amount Due"

FROM RENTAL

WHERE LOWER (PAYMENT) = 'no';

SQL | All Rows Fetched: 3 in 0 seconds

	RENTAL_CUSTOMERID	CUSTOMERNAME	CUSTOMERADDRESS	CUSTOMERPOSTCODE	CUSTOMERTELEPHONE	Amount Due
1	00001	Karan	31 Madrid Lane	SP4 AIN	08788659874	20
2	00003	Michelle	42 Blakely Road	M13 3DL	07988654784	30
3	00005	Luke	32 Plankton Hills	O15 6PZ	01230698754	72

This query will search all customers that have not yet made a full payment for their rental; it shows their details and the amount that still needs to be paid. This was done by subtracting the price by the amount paid.

SELECT PARTORDER. ORDERID , PARTORDER.PARTQUANTITY, PARTSUPPLIER.SUPPLIERID,

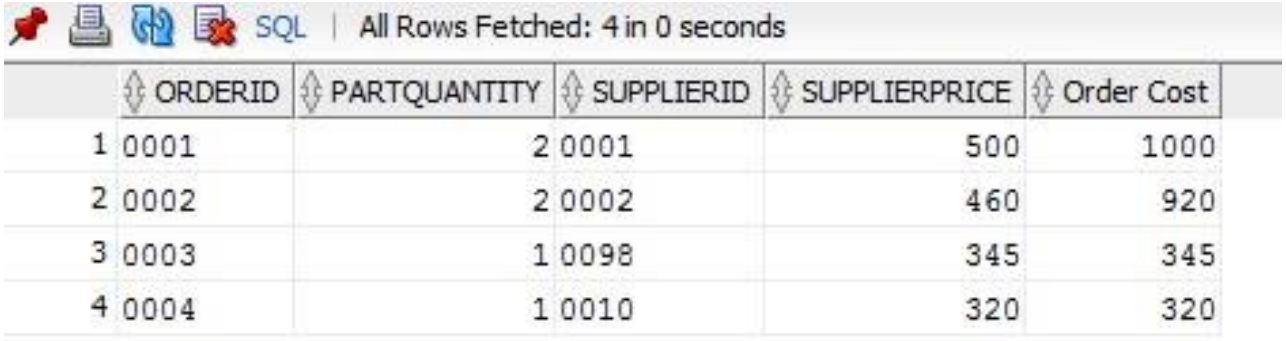
PARTSUPPLIER.SUPPLIERPRICE ,(PARTQUANTITY * SUPPLIERPRICE)AS "Order Cost"

FROM PARTORDER

INNER JOIN PARTSUPPLIER

ON PARTORDER.PARTID = PARTSUPPLIER.PARTID;

This query merges PARTORDER and PARTSUPPLIER, this produces an invoice that shows information about what has been ordered, quantity and the total cost each type of item.



SQL | All Rows Fetched: 4 in 0 seconds

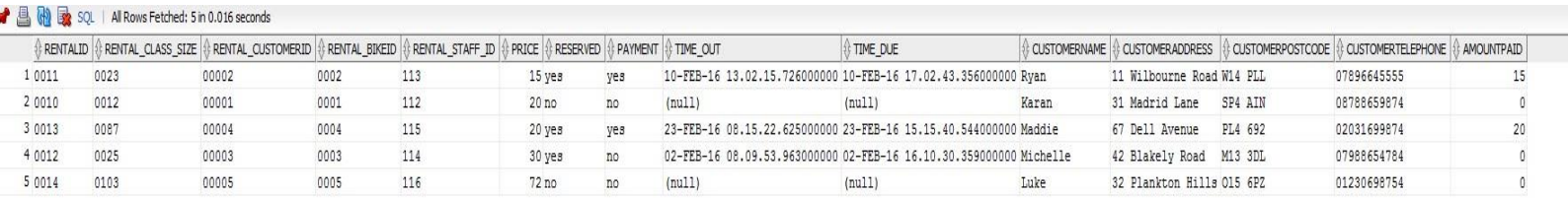
	ORDERID	PARTQUANTITY	SUPPLIERID	SUPPLIERPRICE	Order Cost
1	0001	2	0001	500	1000
2	0002	2	0002	460	920
3	0003	1	0098	345	345
4	0004	1	0010	320	320

Madaline

SELECT *

FROM RENTAL

ORDER BY PRICE ASC;



SQL | All Rows Fetched: 5 in 0.016 seconds

	RENTALID	RENTAL_CLASS_SIZE	RENTAL_CUSTOMERID	RENTAL_BIKEID	RENTAL_STAFF_ID	PRICE	RESERVED	PAYMENT	TIME_OUT	TIME_DUE	CUSTOMERNAME	CUSTOMERADDRESS	CUSTOMERPOSTCODE	CUSTOMERTELEPHONE	AMOUNTPAID
1	0011	0023	00002	0002	113	15	yes	yes	10-FEB-16 13.02.15.726000000	10-FEB-16 17.02.43.356000000	Ryan	11 Wilbourne Road	W14 PLL	07896645555	15
2	0010	0012	00001	0001	112	20	no	no	(null)	(null)	Karan	31 Madrid Lane	SP4 AIN	08788659874	0
3	0013	0087	00004	0004	115	20	yes	yes	23-FEB-16 08.15.22.625000000	23-FEB-16 15.15.40.544000000	Maddie	67 Dell Avenue	PL4 692	02031699874	20
4	0012	0025	00003	0003	114	30	yes	no	02-FEB-16 08.09.53.963000000	02-FEB-16 16.10.30.359000000	Michelle	42 Blakely Road	M13 3DL	07988654784	0
5	0014	0103	00005	0005	116	72	no	no	(null)	(null)	Luke	32 Plankton Hills	O15 6PZ	01230698754	0

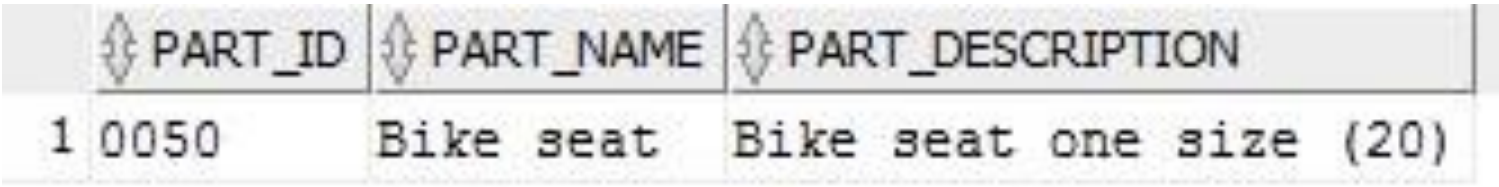
This query checks the price of bike rentals in ascending order; this will be helpful when trying to search for rentals by the lowest price.

SELECT *

FROM PARTS

WHERE part_name = 'Bike seat';

This query checks for parts that are called bike seat, this will be helpful for when a bike seat is needed



	PART_ID	PART_NAME	PART_DESCRIPTION
1	0050	Bike seat	Bike seat one size (20)

SELECT COUNT (*) "Supplier Count"

FROM SUPPLIER;

This query gives the amount of suppliers that Rays Rentals has.

	Supplier Count
1	6

Karan

SELECT BIKE_RECORDS.BIKE_ID, BIKE_RECORDS.BIKE_CLASS_SIZE_ID, DEALER.DEALER_NAME, BIKE_RECORDS.PURCHASE_PRICE

FROM BIKE_RECORDS

INNER JOIN DEALER

ON BIKE_RECORDS.BIKE_DEALER_ID = DEALER.DEALER_ID;

This query uses the INNER JOIN function to be able to "join" two tables (BIKE_RECORDS and DEALER), using the common DEALER_ID field from both the tables. It then displays the records we have selected to show from both the tables with DEALER_ID.

	BIKE_ID	BIKE_CLASS_SIZE_ID	DEALER_NAME	PURCHASE_PRICE
1	0003	0012	Bikes4U	755
2	0005	0087	Bikes4U	250
3	0004	0023	TwoWheels	800
4	0001	0025	TwoWheels	500
5	0002	0103	BikeMe	300

SELECT AVG (PURCHASE_PRICE) As PURCHASE_PRICE_AVERAGE

FROM BIKE_RECORDS;

This query calculates the average from the average purchase price and then, it names it as "PURCHASE_PRICE_AVERAGE", selecting the records from the BIKE_RECORDS table.

PURCHASE_PRICE_AVERAGE
521

SELECT (STAFFFIRSTNAME ||' ' || STAFFLASTNAME) AS "FULL_STAFF_NAME"

FROM STAFF;

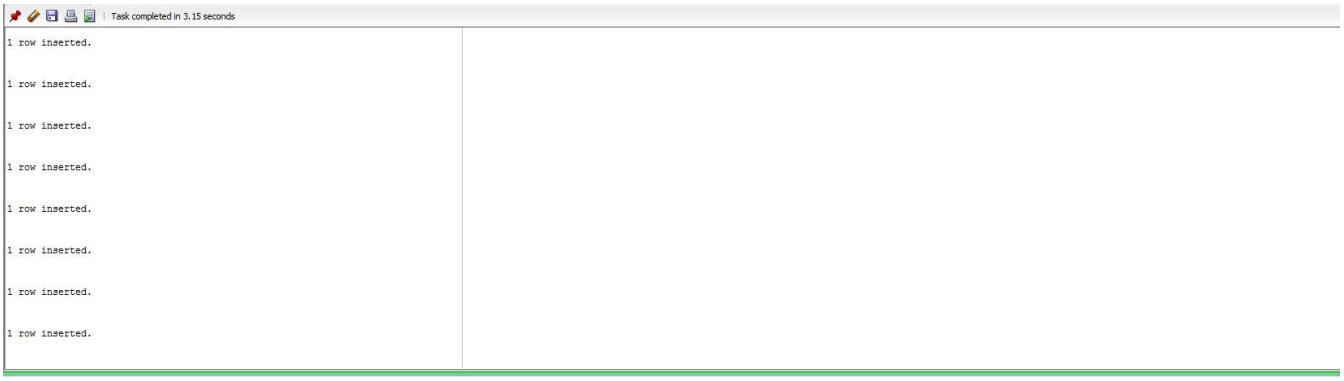
This query uses the Staff's first and last name, concatenating both of the records in these fields. The "|| ' '||" is used to insert a space between both of them and it is named as "FULL_STAFF_NAME."

	FULL_STAFF_NAME
1	Ray Pickford
2	Pete Jones
3	Sheila Smith
4	Megan Parker
5	Alf Lucas
6	Bert Williams

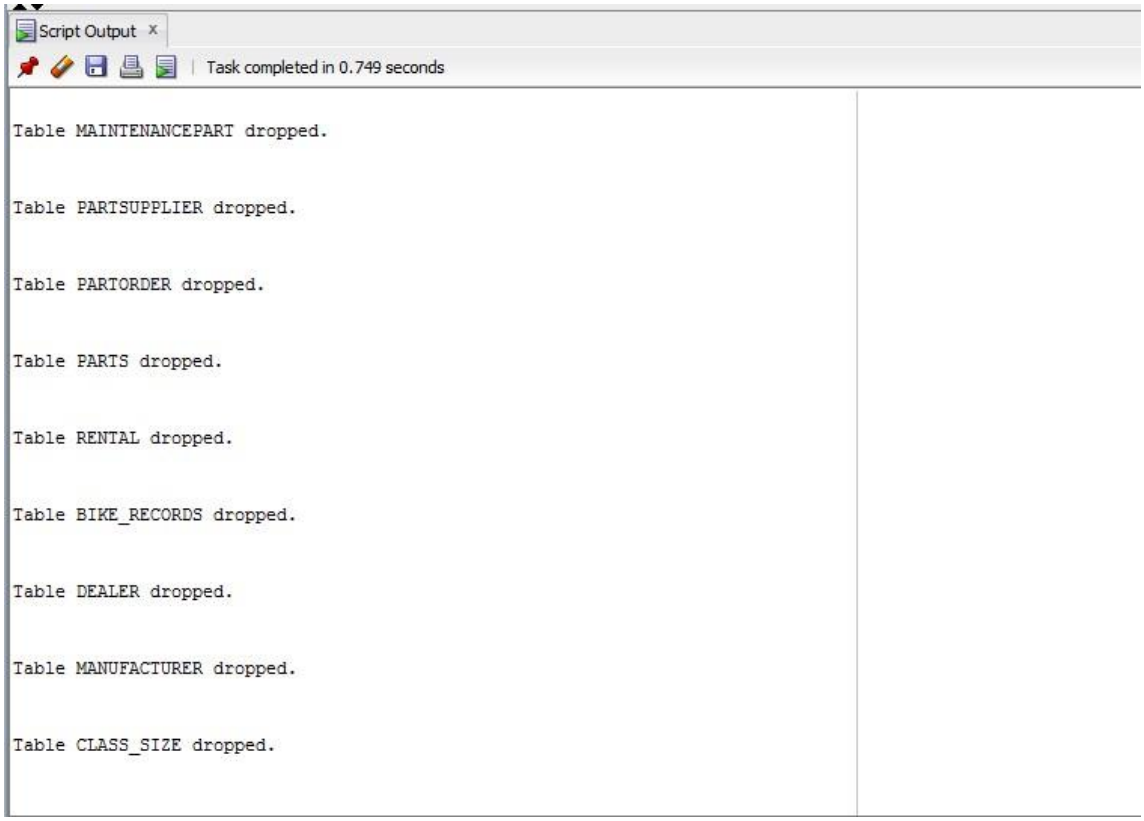
Question 5 – Document database design with an ORACLE script file, and outputs of when the script is run

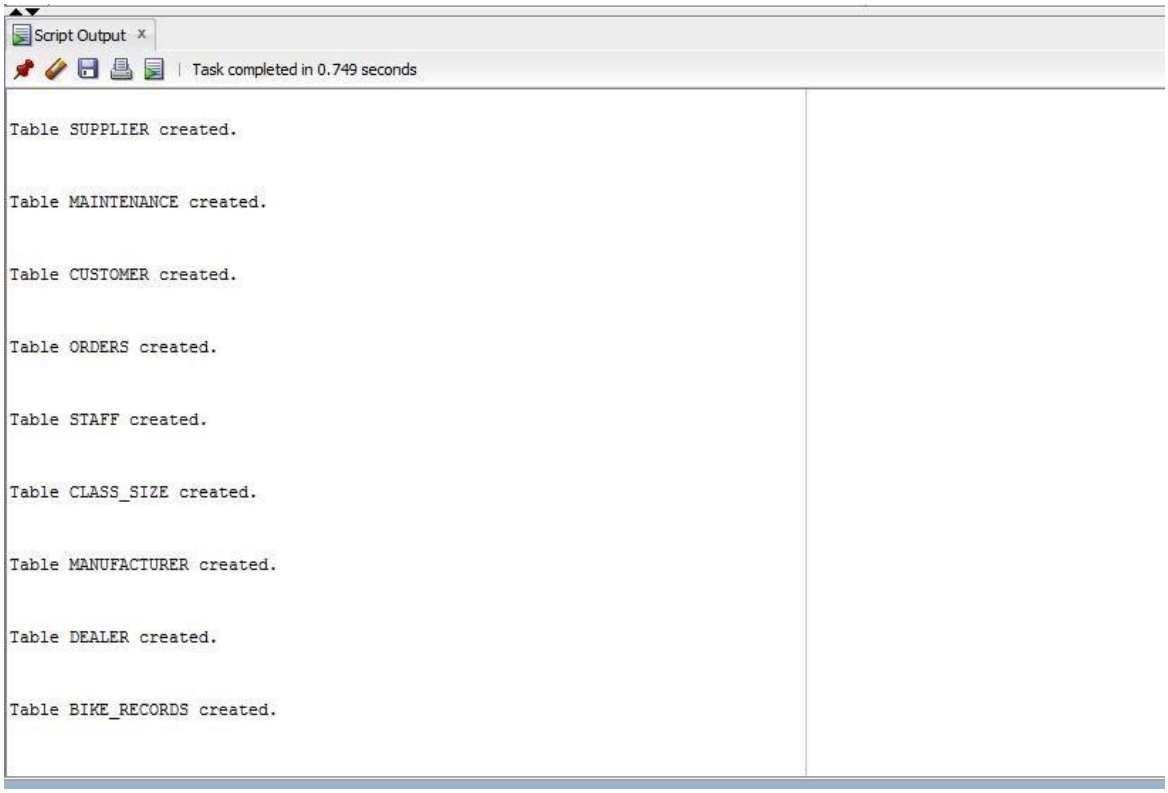
```
/*Query by Michelle*/
SELECT *
FROM BIKE_RECORDS
ORDER BY PURCHASE_PRICE;

/*Query by Michelle*/
SELECT PARTS.PART_NAME, PARTS.PART_DESCRIPTION, PARTSUPPLIER.SUPPLIERID, PARTSUPPLIER.SUPPLIERPRICE
FROM PARTS
INNER JOIN PARTSUPPLIER
ON PARTS.PART_ID = PARTSUPPLIER.PARTID;

/*Query By Ryan*/

SELECT RENTAL_CUSTOMERID, CUSTOMERNAME,CUSTOMERADDRESS,CUSTOMERPOSTCODE, CUSTOMERTELEPHONE,(PRICE-AMOUNTPAID) "Amount Due"
FROM RENTAL
WHERE LOWER (PAYMENT) = 'no';

/*Query By Ryan*/
SELECT PARTORDER. ORDERID , PARTORDER.PARTQUANTITY, PARTSUPPLIER.SUPPLIERID,
PARTSUPPLIER.SUPPLIERPRICE ,(PARTQUANTITY * SUPPLIERPRICE)AS "Order Cost"
FROM PARTORDER
INNER JOIN PARTSUPPLIER
ON PARTORDER.PARTID = PARTSUPPLIER.PARTID;

/*Query By Madaline*/
SELECT *
FROM RENTAL
ORDER BY PRICE ASC;
```

```
INNER JOIN PARTSUPPLIER
ON PARTORDER.PARTID = PARTSUPPLIER.PARTID;

/*Query By Madaline*/
SELECT *
FROM RENTAL
ORDER BY PRICE ASC;

/*Query By Madaline*/
SELECT *
FROM PARTS
WHERE part_name = 'Bike seat';

/*Query By Madaline*/
SELECT COUNT (*) "Supplier Count"
FROM SUPPLIER;

/*Query By Karan*/
SELECT BIKE_RECORDS.BIKE_ID, BIKE_RECORDS.BIKE_CLASS_SIZE_ID, DEALER.DEALER_NAME, BIKE_RECORDS.PURCHASE_PRICE
FROM BIKE_RECORDS
INNER JOIN DEALER
ON BIKE_RECORDS.BIKE_DEALER_ID = DEALER.DEALER_ID;

/*Query By Karan*/
SELECT AVG (PURCHASE_PRICE) As PURCHASE_PRICE_AVERAGE
FROM BIKE_RECORDS;

/*Query By Karan*/
SELECT (STAFFFIRSTNAME || ' ' || STAFFLASTNAME) AS "FULL_STAFF_NAME"
FROM STAFF;
```



```
Task completed in 3.15 seconds

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.
```


Script Output
Task completed in 0.749 seconds
Table MAINTENANCEPART dropped.
Table PARTSUPPLIER dropped.
Table PARTORDER dropped.
Table PARTS dropped.
Table RENTAL dropped.
Table BIKE_RECORDS dropped.
Table DEALER dropped.
Table MANUFACTURER dropped.
Table CLASS_SIZE dropped.


Script Output
Task completed in 0.749 seconds
Table SUPPLIER created.
Table MAINTENANCE created.
Table CUSTOMER created.
Table ORDERS created.
Table STAFF created.
Table CLASS_SIZE created.
Table MANUFACTURER created.
Table DEALER created.
Table BIKE_RECORDS created.

Question 6 – Commentary explaining what has been learnt

Ryan

When creating this database, lessons were learnt from the offset concerning our previous design, we understood how and why we had to rethink some parts of our original plan and this increased our understanding of the process.

When implementing our design into oracle, it was first necessary to create data dictionaries that would plan all the fields for our tables as well as the datatypes and constraints that these fields would accept. Creating a data dictionary also made it easier to display and plan for the relationships between different tables.

Once we had created our data dictionaries, it was relatively easy to write out our SQL to create these tables. We came across several problems when attempting to create these tables, one problem we faced was a misunderstanding of the order we had to create and drop our tables, however with perseverance and some help from a tutor, we were able to correctly drop and create all of our tables.

Further to creating the tables, we had to insert data into these tables, this seemed straight forward at the time using the GUI interface to simply input data, after learning that using this technique is not a suitable way of implementing data as this data could then not be transferred easily when running the script file on a different account.

Furthermore, we attempted to use insert statements to insert our data into our tables, we encountered many difficulties during this stage, as we could not seem to delete our previous data, after other members also could not figure out how to correctly implement insert statements and I decided to persevere and eventually got all statements working. Queries were also created that taught me how to use inner join commands as well as mathematical statements inside of queries; other commands were also learnt that allowed me to create queries.

This task has taught me how to implement our design into ORACLE SQL, taking an idea, designing the system and creating it. I have developed skills such as how to use structured query language (SQL), this project has also helped to develop my problem solving skills, developing solutions to problems. Also developing my time management skills, as I was unable to access my account for a week during this task, so had to manage my time carefully to complete the assignment on time.

Michelle

During the creation of this database, I have learnt a range of things. Primarily, I have learnt how to use ORACLE and create a working and effective database for the use of a small business such as Rays Rentals. Creating tables in ORACLE was one of the first tasks, after dropping the tables, this allowed us to have a starting base for the database and build up from this. After the tables were

created, I then learnt how to insert data into them; this was useful for the queries, which would follow this task. I learnt what queries were and that queries allow the user of the database to separate specific pieces of data, depending on what they wanted to search for. Overall, ORACLE has been an essential part of this project, as I now know what it takes to make a suitable database and the types of coding which must be implemented in order for it to suit the business of Rays Rentals.

Madaline

This part of the project has given me an insight into what it would be like to create a database for a small company. I have learnt some of the key skills needed when creating databases, these are

- Structured Query Language and how to implement the code for databases. Having learnt some of the programming language of SQL, it allowed me to create tables and insert data using the create and insert commands. This skill has also allowed me to write queries using the select command for the tables that were created.
- Problem solving is a skill I have improved upon during this part of the project. As we came across a few of errors when creating the database tables and inputting the data, it was a very important skill to have for this part of the project.

Karan

During this part of the Ray's Rentals Project, I have learned many new skills that we have implemented as a group in our document and database. What has been learnt, is the use of the basic instructions like, SELECT, allowing us to be able to pick out data to make queries with certain criteria; DROP, giving us the ability to remove unwanted tables and CREATE statements, which gave us access to start to build our very own database, after dropping previous tables. Also, while we were creating our queries, we were able to dive into more complex statements, such as, using mathematical operators, INNER JOINS (linking two tables together with the same primary key) and even concatenate functions. Overall, we have learned to try and find more elaborate ways to solve the problems we had with our database at the beginning by revising our syntax and how our code was written.

Conclusion

To conclude on this report, we were set out to create a database suitable for use in Ray's rentals and that would improve the business, we were able to complete this and created a database that would make the lives of the staff and customers easier.

Queries have been created that staff can use to get answers to a question they may need to ask such as "What is your cheapest bike to rent?", "What customers still owe money on their rentals and how much do they owe?" These queries are useful in terms of business; these records will not be lost as they previously could be. These queries are also useful in terms of the business at Ray's Rentals, having a reliable system that can accurately keep track of all bikes, orders,

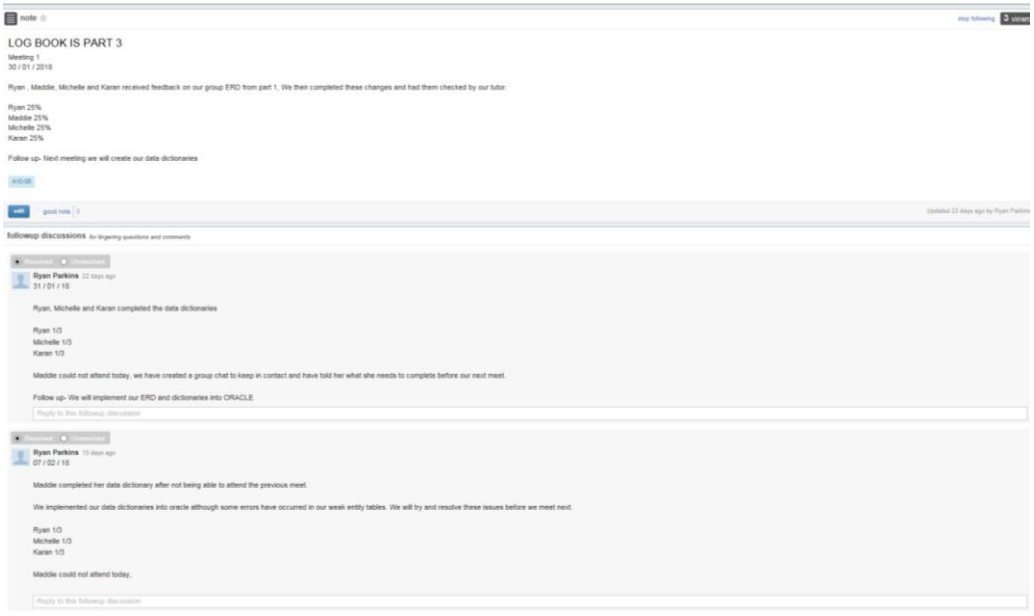
customers, maintenance tasks and staff will ensure a well organized, smooth running, productive and profitable business.

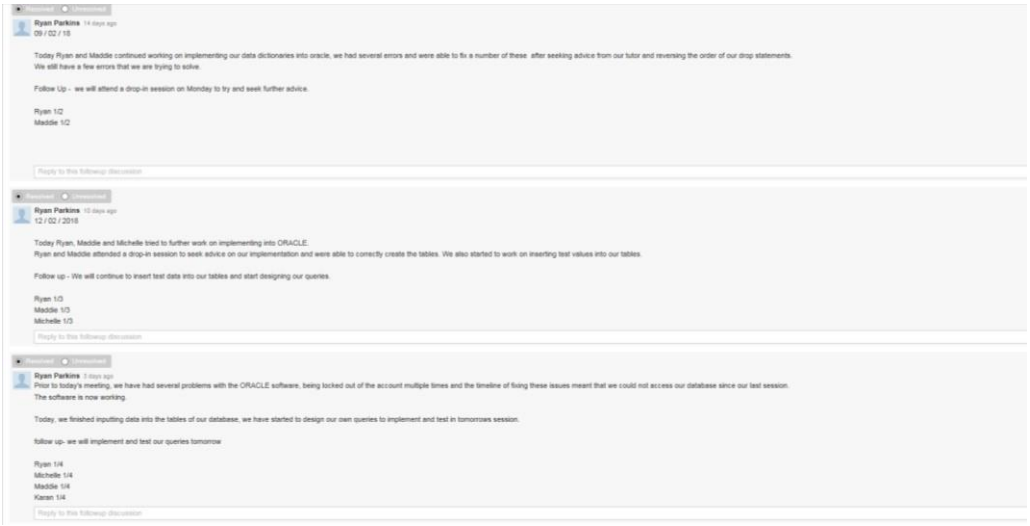
During this project, we were all able to understand the tasks and completed tasks. Even when we faced problems, some of us were able to attend support sessions to get advice on how to solve the problems we were having.

Problems that occurred in this project included losing access to the oracle account that had our tables on; this was a problem for us as the member of the group whose account was locked was unable to get into university until the end of the week. To avoid this problem, we would have shared the SQL script file with each other; another problem that we faced during this task was all members were rarely all able to meet at the same time. This was an issue because all members need to be together for all work to be fairly distributed when doing this sort of task, so during this assignment, there were some parts that were not contributed on my missing members, to avoid this problem, we should have given out work to be completed and deadlines to be completed by.

Appendix

Log entries from Piazza





Ryan Parkins 14 days ago
09 / 02 / 16

Today Ryan and Maddie continued working on implementing our data dictionaries into oracle, we had several errors and were able to fix a number of these after seeking advice from our tutor and reversing the order of our drop statements.
We still have a few errors that we are trying to solve.

Follow Up - we will attend a drop-in session on Monday to try and seek further advice.

Ryan 1/2
Maddie 1/2

Reply to this followup discussion

Ryan Parkins 10 days ago
12 / 02 / 2016

Today Ryan, Maddie and Michelle tried to further work on implementing into ORACLE.
Ryan and Maddie attended a drop-in session to seek advice on our implementation and were able to correctly create the tables. We also started to work on inserting test values into our tables.

Follow up - We will continue to insert test data into our tables and start designing our queries.

Ryan 1/3
Maddie 1/3
Michelle 1/3

Reply to this followup discussion

Ryan Parkins 3 days ago
Prior to today's meeting, we have had several problems with the ORACLE software, being locked out of the account multiple times and the timeline of fixing these issues meant that we could not access our database since our last session.
The software is now working.

Today, we finished inputting data into the tables of our database, we have started to design our own queries to implement and test in tomorrows session.

follow up- we will implement and test our queries tomorrow

Ryan 1/4
Michelle 1/4
Maddie 1/4
Karan 1/4

Reply to this followup discussion

Ryan Parkins 3 days ago
Prior to today's meeting, we have had several problems with the ORACLE software, being locked out of the account multiple times and the timeline of fixing these issues meant that we could not access our database since our last session.
The software is now working.

Today, we finished inputting data into the tables of our database, we have started to design our own queries to implement and test in tomorrows session.

follow up- we will implement and test our queries tomorrow

Ryan 1/4
Michelle 1/4
Maddie 1/4
Karan 1/4

Reply to this followup discussion

Ryan Parkins Just now
22 / 02 / 16

Prior to todays meeting, we all designed our queries and wrote them into SQL.

Today, we implemented and tested our queries in ORACLE.

Ryan 1/4
Michelle 1/4
Maddie 1/4
Karan 1/4

Reply to this followup discussion

Ryan Parkins Just now
23 / 02 / 16

After completing our queries in out last meet, we realised that we did not complete any insert statements as we instead used the GUI to add data.
Ryan created the insert statements and added them to the script.
After solving a few errors, everything works correctly and all queries work.

Today Ryan also added the ERD, data dictionaries and other parts to the final report.
Everybody added their commentaries. Maddie and Ryan completed the introduction, executive summary and the conclusion.

Reply to this followup discussion

Conclusion

This report was set out to examine the business of Ray's rentals and find weaknesses within it. A new system was designed that has removed many of the current problems within the business.

The new database system was designed and implemented into ORACLE SQL and ran successfully. Further on, the queries made, were developed with the purpose of improving Ray's company and therefore, providing better prospects and efficiency in the output of selected data.

Working as a team, helped us improve our leadership skills and qualities and simultaneously, allowed us to progress at a steady pace, during this project. Tasks were evenly distributed between all of us, using the log book on Piazza, justifying that work would be carried out in a productive manner, with the intention of the entire group dedicating their maximum effort.

References

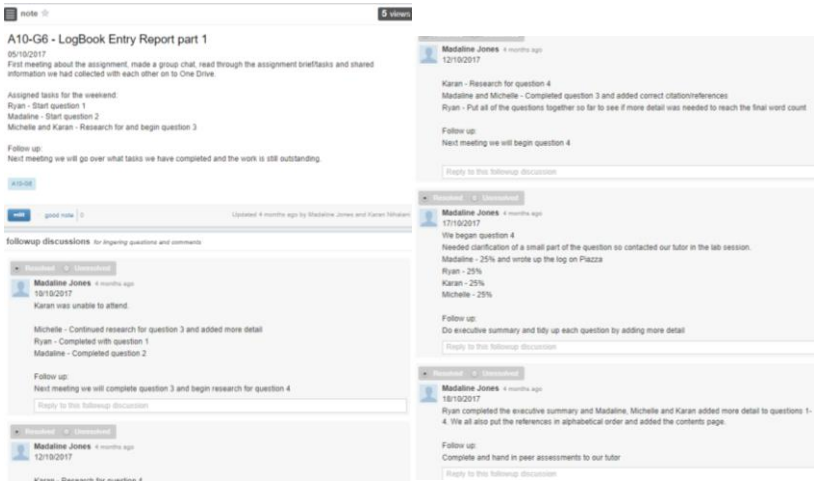
- archive.cossac.org, (2001) http://archive.cossac.org/tallis/Basic_proforma_concepts08.htm (Accessed on 12/10/2017)
- D.Whitely (2013) 'An introduction to Information Systems' page 151 (Accessed on 10/10/2017)
- ecs.soton.ac.uk, (2001) L.car, W.Hall [http://www.ecs.soton.ac.uk/research/projects/212] (Accessed on 10/10/2017)

- **Fig 1** Schofield, A. (2017) 'School of Computing , Mathematics & Digital Technology', (Accessed 15/10/2017).
- **Fig 2** Schofield, A. (2017) 'School of Computing , Mathematics & Digital Technology', (Accessed 15/10/2017).
- **Fig 3 -** Stephen Gordon, Management Information and Decision Making (2017) (Accessed on 12/10/2017)
- **Fig 4 -** Stephen Gordon, Management Information and Decision Making (2017) (Accessed on 12/10/2017)
- **Fig 5 -** Stephen Gordon, Management Information and Decision Making (2017) (Accessed on 12/10/2017)
- Stephen Gordon, Management Information and Decision Making (2017) (Accessed on 09/10/2017)
- wisegeek.com (2017) M.Tatum http://www.wisegeek.com/what-is-an-exception-report.htm (Accessed on 11/10/2017)

Appendices

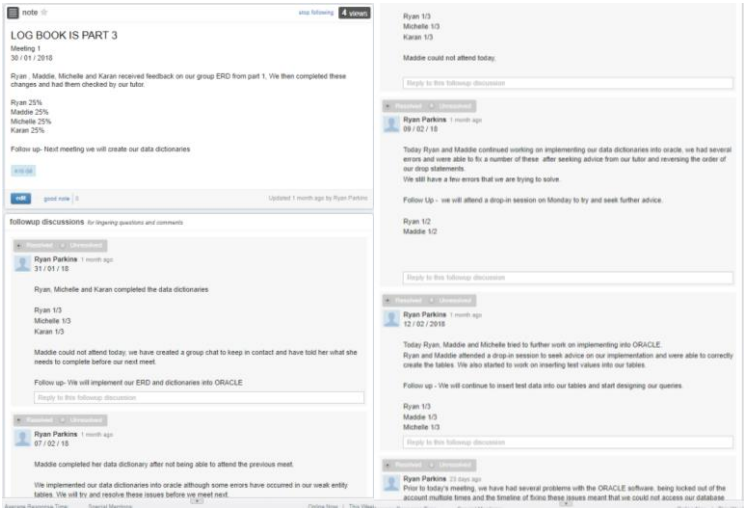
Appendix A : Log Book

Part 1-

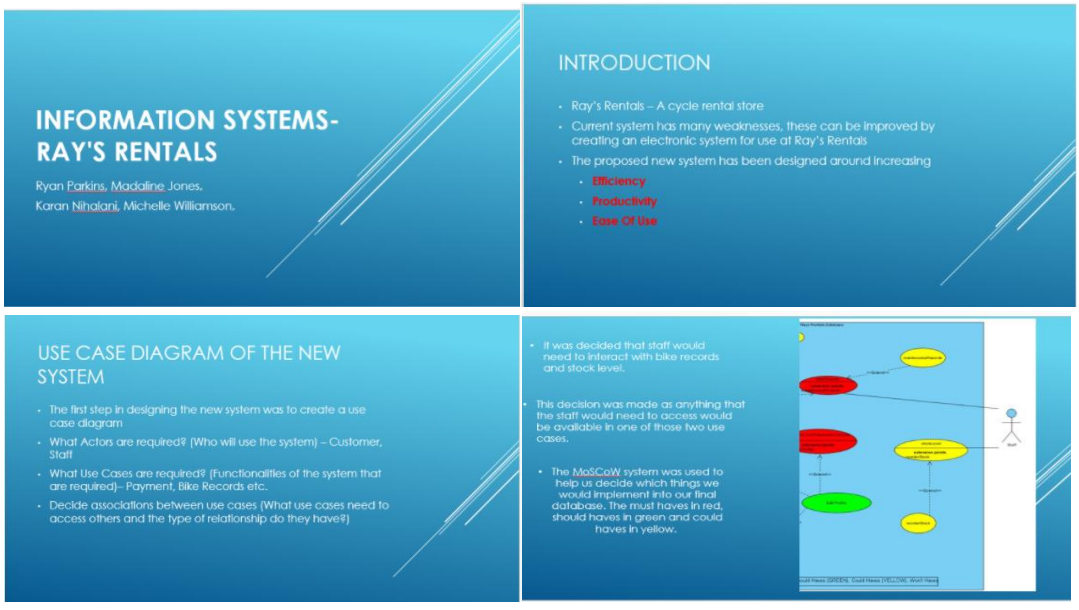


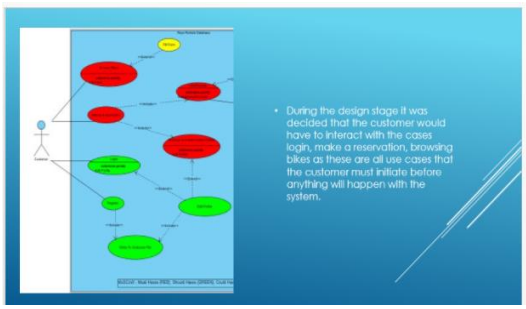
Part 2-

Part 3-



Appendix B : Presentation slides





- During the design stage it was decided that the customer would have to interact with the cases login, make a reservation, browsing bikes as these are all use cases that the customer must initiate before anything will happen with the system.

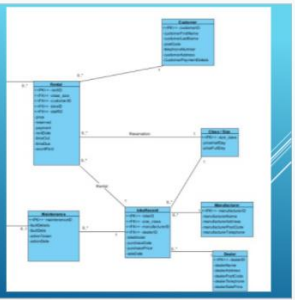
ENTITY RELATIONSHIP DIAGRAM

- The next step in designing the new system was to create an Entity Relationship Diagram.
- This was a way to establish relationships between use cases and therefore between entities.
- These entities and their relationships were useful when implementing the system into ORACLE and helps us to establish constraints.

ERD

Overview

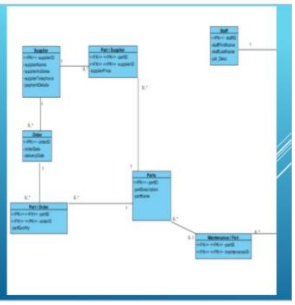
- Established the entities and attributes within them.
- Defined the Primary Keys.
- Provide main structure and allow there to be no duplicate data.
- Foreign Keys created depending on the table they have a relationship with.
- Reservation process.
- Rental process.



ERD

Overview

- Most of relationships are one to many relationships.
- Allows there to be zero child records, one child records or more than one child record.
- Weak entity was created.
- Depends on its owner entity (Maintenance and Parts).
- Composite key used: PK of "owner" entity and partial key of weak entity.
- Prevent further compilcations with the entities in our database system.



ORACLE

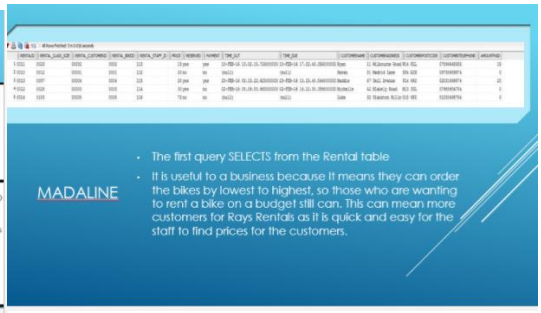
- Implementation into ORACLE SQL

```
DROP TABLE MaintenancePart;
DROP TABLE PartSupplier;
DROP TABLE PartOrder;
DROP TABLE Parts;
DROP TABLE Rental;
DROP TABLE Bike_Records;
DROP TABLE Dealer;
DROP TABLE Manufacturer;
DROP TABLE Class_Size;
DROP TABLE Staff;
DROP TABLE Orders;
DROP TABLE Customer;
DROP TABLE Maintenance;
DROP TABLE Supplier;
```

```
CREATE TABLE Dealer (
Dealer_ID VARCHAR2(10) CONSTRAINT Deal_pk PRIMARY KEY,
Dealer_Name VARCHAR2(30) NOT NULL,
Dealer_Address VARCHAR2(100),
Dealer_PostCode VARCHAR2(10) NOT NULL,
Dealer_Telephone VARCHAR2(15),
Dealer_SalePrice NUMBER(6,2) NOT NULL
);
```

```
Insert into STAFF (STAFFID,STAFFFIRSTNAME,STAFFLASTNAME,JOB_DESC) values ('112','Ray','Pickford','Manager');
Insert into STAFF (STAFFID,STAFFFIRSTNAME,STAFFLASTNAME,JOB_DESC) values ('113','Pete','Jones','Deputy Manager');
Insert into STAFF (STAFFID,STAFFFIRSTNAME,STAFFLASTNAME,JOB_DESC) values ('114','Sheila','Smith','Assistant');
Insert into STAFF (STAFFID,STAFFFIRSTNAME,STAFFLASTNAME,JOB_DESC) values ('115','Megan','Parker','Assistant');
Insert into STAFF (STAFFID,STAFFFIRSTNAME,STAFFLASTNAME,JOB_DESC) values ('116','Alf','Lucas','Technician');
Insert into STAFF (STAFFID,STAFFFIRSTNAME,STAFFLASTNAME,JOB_DESC) values ('117','Bert','Williams','Technician');
```

```
Insert into CLASS_SIZE (CLASS_SIZE_ID,PRICE_HALF_DAY,PRICE_FULL_DAY) values ('0025',80,150);
Insert into CLASS_SIZE (CLASS_SIZE_ID,PRICE_HALF_DAY,PRICE_FULL_DAY) values ('0585',90,70);
Insert into CLASS_SIZE (CLASS_SIZE_ID,PRICE_HALF_DAY,PRICE_FULL_DAY) values ('0012',80,100);
Insert into CLASS_SIZE (CLASS_SIZE_ID,PRICE_HALF_DAY,PRICE_FULL_DAY) values ('0023',75,125);
Insert into CLASS_SIZE (CLASS_SIZE_ID,PRICE_HALF_DAY,PRICE_FULL_DAY) values ('0087',25,50);
```



MADALINE

- The first query SELECTS from the Rental table
- It is useful to a business because it means they can order the bikes by lowest to highest, so those who are wanting to rent a bike on a budget still can. This can mean more customers for Rays Rentals as it is quick and easy for the staff to find prices for the customers.

PART_ID	PART_NAME	PART_DESCRIPTION
1 0050	Bike seat	Bike seat one size (20)

MADALINE

- This second query SELECTS from the parts table
- It is useful to a business because it allows a staff member to check where to buy bike seat from.

Supplier Count
1 6

MADALINE

- This query SELECTS from the supplier table
- This query gives the supplier count, this is useful so that the staff can see how many suppliers the company currently has.

SQL | All Rows Fetched: 3 in 0 seconds

RENTAL_CUSTOMERID	CUSTOMERNAME	CUSTOMERADDRESS	CUSTOMERPOSTCODE	CUSTOMERTELEPHONE	Amount Due
1 00001	Karan	31 Madrid Lane	SP4 AIN	08788659874	20
2 00003	Michelle	42 Blakely Road	M13 3DL	07988654784	30
3 00005	Luke	32 Plankton Hills	O15 6PZ	01230698754	72

RYAN

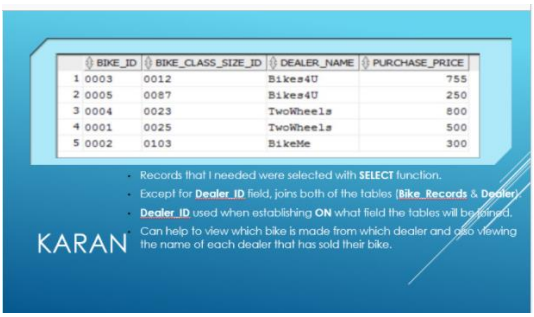
- Selects the rental table
- Where Payment is "No" – Customer has not yet made full payment
- (Price – Amount Paid) = "Amount Due" – total cost remaining for the customer.
- Easier for staff to see all customers that owe money, increases Productivity and Efficiency

SQL | All Rows Fetched: 4 in 0 seconds

ORDERID	PARTQUANTITY	SUPPLIERID	SUPPLIERPRICE	Order Cost
1 0001	2 0001	500	1000	
2 0002	2 0002	460	920	
3 0003	1 0098	345	345	
4 0004	1 0010	320	320	

RYAN

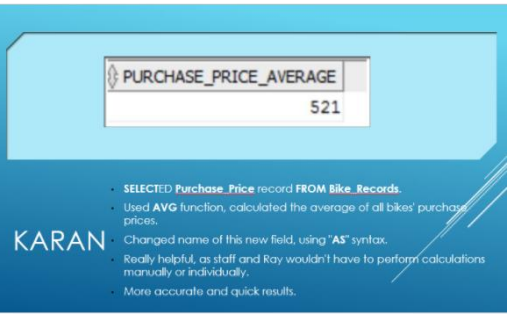
- Merges PARTORDER and PARTSUPPLIER
- Purpose of displaying an invoice of items including buying cost and total cost.
- (SUPPLIERPRICE * PARTQUANTITY) = Order Cost
- Useful for managing stock
- Increases Turnover by ensuring bikes can always be repaired.



	BIKE_ID	BIKE_CLASS_SIZE_ID	DEALER_NAME	PURCHASE_PRICE
1	0003	0012	Bikes4U	755
2	0005	0087	Bikes4U	250
3	0004	0023	TwoWheels	800
4	0001	0025	TwoWheels	500
5	0002	0103	BikeMe	300

- Records that I needed were selected with **SELECT** function.
- Except for **Dealer_ID** field, joins both of the tables (**Bike_Records** & **Dealer**)
- **Dealer_ID** used when establishing **ON** what field the tables will be joined.
- Can help to view which bike is made from which dealer and also viewing the name of each dealer that has sold their bike.

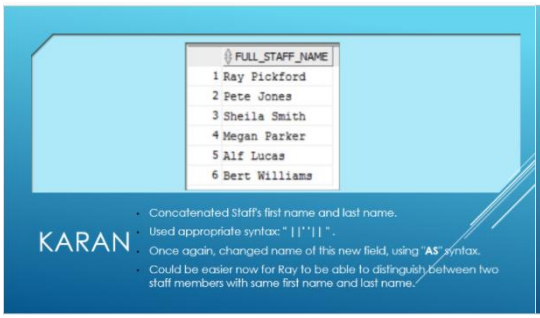
KARAN



	PURCHASE_PRICE_AVERAGE
	521

- **SELECT**ED **Purchase_Price** record **FROM Bike_Records**.
- Used **AVG** function, calculated the average of all bikes' purchase prices.
- Changed name of this new field, using "**AS**" syntax.
- Really helpful, as staff and Ray wouldn't have to perform calculations manually or individually.
- More accurate and quick results.

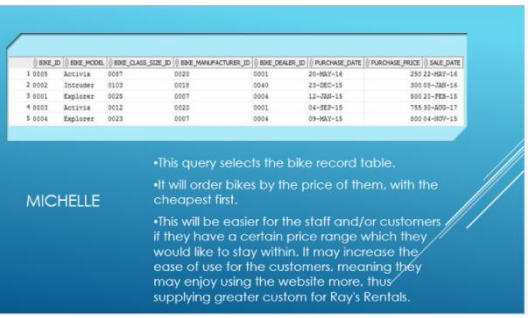
KARAN



	FULL_STAFF_NAME
1	Ray Pickford
2	Pete Jones
3	Sheila Smith
4	Megan Parker
5	Alf Lucas
6	Bert Williams

- Concatenated Staff's first name and last name.
- Used appropriate syntax: " | |'' | | ".
- Once again, changed name of this new field, using "**AS**" syntax.
- Could be easier now for Ray to be able to distinguish between two staff members with same first name and last name.

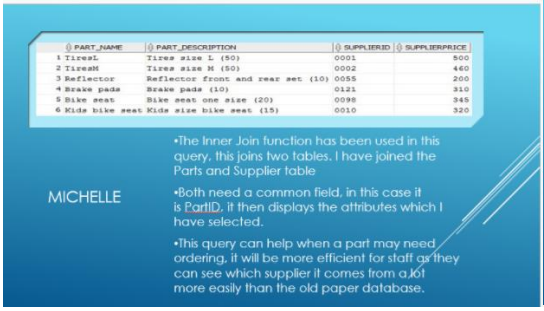
KARAN



	BIKE_ID	BIKE_MODEL	BIKE_CLASS_SIZE_ID	BIKE_MANUFACTURER_ID	BIKE_DEALER_ID	PURCHASE_DATE	PURCHASE_PRICE	SALE_DATE
1	0005	Activia	0087	0020	0001	20-MAY-16	250	22-MAY-16
2	0002	Intruder	0103	0018	0040	23-DEC-15	300	08-JAN-16
3	0001	Explorer	0025	0007	0004	12-JAN-15	500	20-FEB-15
4	0003	Activia	0012	0020	0001	04-SEP-15	755	30-NOV-17
5	0004	Explorer	0023	0007	0004	09-MAY-15	800	04-NOV-15

MICHELLE

- This query selects the bike record table.
- It will order bikes by the price of them, with the cheapest first.
- This will be easier for the staff and/or customers if they have a certain price range which they would like to stay within. It may increase the ease of use for the customers, meaning they may enjoy using the website more, thus supplying greater custom for Ray's Rentals.



	PART_NAME	PART_DESCRIPTION	SUPPLIERID	SUPPLIERPRICE
1	TiresL	Tires size L (50)	0001	500
2	TiresM	Tires size M (50)	0002	460
3	Reflector	Reflector front and rear set (10)	0055	200
4	Brake pads	Brake pads (10)	0121	310
5	Bike seat	Bike seat one size (20)	0098	345
6	Kids bike seat	Kids size bike seat (15)	0010	320

MICHELLE

- The Inner Join function has been used in this query, this joins two tables. I have joined the Parts and Supplier table
- Both need a common field, in this case it is PartID, it then displays the attributes which I have selected.
- This query can help when a part may need ordering, it will be more efficient for staff as they can see which supplier it comes from a lot more easily than the old paper database.

SUMMARY

- The new system developed helped to improve Ray's Rentals' company as a whole.
- This in turn would increase **productivity**, **profits**, **functionality**, **reliability**, and **service quality**.
- Developing Use Case Diagram's and Entity Relationship Diagram's was extremely useful in aiding to create final database and the queries within them.