Ray's Rentals – Information Systems Project

Huseyin Arpalikli - 13153439 & Mohammed Haroon Bilimoria - 13155867

Table of Contents:

1.0 Project Introduction

This project goes through the process of converting the paper-based system to a computerised system within Ray's Rentals. The report will start by investigating the current system and its flaws, requirements for a new system and then investigation of how management reports can be used within the business.
Following this, the project will go through the process of analysing the current system and use this to carry out the system design, including the use case diagrams, entity relationship diagrams, and normalisation. This will then be used to carry out the final part of the project, the main implementation of the new system for Ray's Rentals. This will consist of the database design, the implementation within Oracle SQL and then creating SQL Queries to create the management reports discussed at the beginning of this report.
During this project, I, Huseyin, put the structure of the report together. The actual content of the report was just about delegated equally between Haroon, and myself. There were some issues with Haroon's understanding, where he would complete some areas incorrectly to the extent that the whole area had to be redone, by myself. This was because he did not fully understand the task. Overall, the report was completed 55% by myself and 45% by Haroon.

2.0 Management Reports Executive Summary

The use of a paper-based system within Ray's Rentals has proved to be very problematic. Ray's Rentals are looking for a way to improve or replace their current system to maximise sales, profit and efficiency. This section of the report will outline how the problems will be resolved with the use of a computerised management information system and management information reports.
This section of the report will start by analysing the current system then identifying main problems with the current system. Following this, the requirements for the new system will be summarised. These requirements will take database requirements, computer requirements, and system functionality into account. In addition to this, management information reports and a few selected types will be investigated and defined and then these report types will then be applied to how they could be used within Ray's Rentals after implementation of a new system.

3.0 Problems with current paper-based system at Ray's Rentals

After examining the current paper based system, which Ray's Rentals uses, there are problems, which will be outlined in this management report.

Lost Records

Paper records within Ray's Rentals may be misfiled or lost. For example once the shop sells a bike, they keep the record for two years for warranty purposes, but if this records were lost, then it would be a problem for the customer returning the bike for repair as Ray's Rentals may reject taking it in mistakenly due to unknowing loss of the bike record.

Insecure Storage (Data Protection)

A big issue with the paper-based system is the likelihood of data protection breach. Ray's Rentals have stated that they keep customer details, (i.e. address, name) on a 'piece of paper', as well as on the bike records. In line with the Data Protection Act 1998,

this data must be kept accurate, secure and up to date, which can be hard with the current system as the data can easily be compromised.
(Gov.uk, 2013)

Storage Issues

With the current system, filing cabinet storage will fill up with bike records, rental records, receipts, orders and delivery notes. This means buying more storage as well as using more physical space. Ray's Rentals have stated that parts orders and delivery notes are kept in piles and are stored in cabinets only if someone remembers to put them away. This means that other space around the workplace will also be taken up with storage of more cluttered paper.

Slow Record Retrieval/Access

When Ray's Rentals need to access records, time is wasted looking for the cluttered paper records. When 'pieces of paper' are left lying around, this can also cause slow retrieval if staff forget where they placed the notes.

Clashing Reservations

Rental records are not an efficient way of booking reservations. Currently the company takes reservations days/weeks in advance, using the 'next available bike' method, but when reservations get forgotten about or accidentally double booked this causes embarrassment to the company.

Incorrect Hire Price List

A major issue with the current system is the hire list getting out of date, and being used regardless. As well as not looking professional it could cause embarrassment if Ray's Rentals provides wrong pricing to customers.

Illegible Records

The system at Ray's Rentals is paper based. It has been noted that at times, some handwritten notes, receipts and records can become illegible. This defeats the purpose of having them if they cannot be read.

Repair Prioritisation

Within Ray's Rentals there is no real way of prioritising servicing and repair work. This means that sometimes bikes, which are in heavy demand, are off the road for longer than they need to be.

Overstocked and Out of Stock Parts

There are currently no records or forms keeping record of parts stock. This would mean that it is hard for Ray's Rentals to keep track of parts stock and could cause issues when it comes to needing parts to repair bikes and also the fact that unused excess stock would be dormant in the stock room and as well as loosing value over time, money would have been wasted buying the stock in the first place.

Crosscheck on Parts Orders and Deliveries

With the use of the current paper based system, there is no easy way to crosscheck the part orders made against the deliveries and what they already have available. The current method would be compare orders and delivery notes to see what has and hasn't been delivered yet. This is inefficient and would waste a lot of time.

4.0 System requirements for proposed computerised system

To create a new system for Ray's Rentals the following are essential requirements:

- A relational database will need to be created.
- There will need to be links between every table using foreign and primary keys; in the case of this database, they will be the IDs specified in tables.
- The relational database will need to contain the following tables:
    - Customers; containing details of customers; examples of attributes include customer id, customer name, address, etc.
    - Bike Rental Reservations; containing records of rental reservations; some attributes include reservation number, bike number, customer id, rental date, rental time, time due back, actual time back, etc.
    - Bikes; containing details of bikes available for rental; examples of attributes include bike number, manufacturer id, class, size, etc.
    - Maintenance; containing details of servicing for bikes; some attribute include maintenance ref, bike number, part number, fault details, fault date, etc.
    - Parts purchased; containing details of purchased bike parts; examples of attributes here include part number, part details, cost, maintenance ref, etc.
    - Employee; containing details of employees; some attributes here include the employee id, employee name, employee position, etc.; the purpose of this will mainly be to track who makes what reservations.
    - Manufacturers; containing details of bike manufacturers will be stored here; examples of attributes here include manufacturer id, name, phone number, etc.
    - Sold bikes; containing details of bikes sold; attributes here include dealer id, bike number, sale price, etc.
    - Dealers; containing records of bike buyers; examples of attributes here include dealer id, dealer name, etc.

As well as the initial database creation requirements, other requirements that are required will need to be considered to create a successful system:

The system will be required to:

- Show/hide sold bikes in the bike database table;
- Automatically assign the next available bike to a reservation; the bike would need to be from the selected class and size;
- Order records by different attributes in tables (i.e. by date, name, id number, etc.);
- Add/edit/delete records in database tables;
- Provide password protection for access; employees would be required to enter their assigned username and password;
- Limit access for some users; i.e. staff on the reservation team should not be able to edit/delete bike records, maintenance records, or any other records not relevant to their role;
- Calculate aggregated results for management analysis reports;

- Generate sample price lists for customers by using pricing by bike class and size.

The following are further requirements:

- A server that will store the database and provide access to clients computers on the network; this server will need sufficient hard disk space, memory, a fast processor, and fast broadband
- Computers in each department which connect to the server;
- Creation of a website, where customer can make and pay for reservations – will require a password protected system for customers and system admin staff.
- Price lists on the website;
- A web domain where the website can be accessed.

5.0 Management Information Reports

5.1 What are Management Information Reports?

Management Information Reports are the output reports produced by processing raw data from a management information system (MIS) or external sources, to facilitate management staff in decision-making and analysis in an organisation. (*Whiteley*, 2013, pp. 145) (*Small Business - Chron.com*, 2013) There are many types of management information reports that can be used. The three covered in this report are analysis reports, exception reports, and key target reports.

5.2 Analysis Reports

Analysis reports are reports where a set of data is aggregated to simplify the data into a view, which can make analysing easier. This type of report works by putting the aggregated data into a single table, chart or graph to show a summary that management staff in the organisation can inspect and make comparisons, etc. (*Info Systems: mgt info & decision making*, 2013, pp. 5)

An example of how analysis reports could be used in an order processing system, could be, for example, to aggregate data into sales by month for the current year, number of sales monthly in the current year, age of credit, stock availability, etc.

Below is another example of an analysis report. This report shows Market Share (%) in the Grocery Industry by Grocer:

	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998[a]
Total market size (£bn)	45.5	49.5	53.4	58.2	62.9	67.2	70.8	74.0	78.1	82.3	86.9	93.3
Tesco	8.4	9.0	9.4	10.0	10.2	10.4	10.7	11.7	13.7	14.6	15.2	15.4
J. Sainsbury	10.0	10.2	10.7	11.3	11.6	12.3	12.4	12.6	12.5	12.6	12.7	12.2
ASDA	4.9	4.7	5.6	7.0	6.7	6.5	6.7	6.9	7.4	8.0	8.5	8.6
Safeway	6.8	6.9	7.0	7.3	7.4	7.5	7.7	7.8	7.5	7.9	7.7	7.6
Somerfield	7.6	7.7	6.5	5.2	4.8	4.4	4.4	4.5	4.3	4.1	3.9	6.9[b]
% share of top 5	37.7	38.5	39.2	40.8	40.7	41.1	41.9	43.5	45.4	47.2	48.0	50.7

Source: Institute for Grocery Distribution Information Unit
[a] Sales for 12 months to December 1998
[b] Sales from Somerfield and Kwik Save combined with effect from 1 July 1998

(*Institute for Grocery Distribution Information Unit*, 1999)

In the example above, data about the total market share and the top five grocers have been aggregated to see how the industry has changed year by year. Data in this form

makes it easier for managerial staff from the various grocers to look at their competition and make well thought out decisions.

5.3 Exception Reports

Exception reports highlight outliers within data. These highlighted outliers, which need attention, are aggregated into a table and then presented separately. Reports should be generated at a regular interval so that it is easier for managers to fix flagged issues that require immediate attention, instead of going through a report from a long period with a long list of old exceptions. Most of the time, exceptions will need to be resolved. Further details are usually needed to resolve an issue. These details are known as diagnosis information. (*Info Systems: mgt info & decision making*, 2013, pp. 6-7) (*Small Business - Chron.com*, 2013)

Two possible examples of where exceptions could be used are:
Exception reports could be used in a large human resources department of an organisation to flag an unusually high level of sick days of staff and take action to investigate.
When stock levels are low for specific items, exception reports could be used to notify a shop manager that levels are low and that they need to be reordered. This report would allow the shop to have notice before stock sells out.

5.4 Key Target Reports

Key target reports are used to measure performance against key targets. They are commonly used in organisations. Key target reports can be outputted in various forms, such as, into tables or various types of graphs (line graph, bar chart, etc.). The output generally shows the target result in comparison to the actual result. The targets are set by management and are well thought out by basing them on previous outcomes as well as what can be realistic. Key target reports are useful for nearly any target as they can be used very broadly. (*Info Systems: mgt info & decision making*, 2013, pp. 7)
An example of its use could be the following:
A car sales company could have a key target of selling at least 10 vehicles every week. Once the key targets have been made, the company will then aggregate data and see whether they employers have met the sales target or not.
Another example would be a coach company. They have individual key targets for journey time, petrol cost and so on.

6.0 Management Information Reports within Ray's Rentals

After exploring different types of management reports in general, a decision has been made on which reports are ideal for Ray's Rentals use. The ideal reports identified are analysis reports, exception reports and key target reports.

6.1 Analysis Reports

As previously identified, analysis reports can be used to aggregate nearly any type of data and present results. Within Ray's Rentals there are many ways that analysis reports will be ideal.
The first way analysis reports could be used would be the most obvious; to calculate how much revenue is being made within the company. This would be useful because if Ray's Rentals were made aware of their revenue on a regular basis, they would be able to compare different periods with each other and also find ways to increase revenue if a decreasing trend is found. As well as this, Ray's Rentals can use different types of monetary based measurements that can be reported (this includes reporting by cost of parts, by bikes bought, bikes sold, profit, etc.).

Another way analysis reporting could be used is to find out the total number of reservations made within a specific period. This would be another way that Ray's Rentals would measure how well the company is doing. They could also break this down even further and calculate the total number of reservations by classification or size of the bikes. This would be used to see which type of bikes are the most popular.

6.2 Exception Reports

Exception reports will also be a useful way of reporting within Ray's Rentals. This is because exception reports will be useful to flag up irregularities within their business. One way exception reporting could be used within Ray's Rentals would be to flag up unpaid rentals after the hire date has already passed.
Exception reporting could also be used for flagging up which bikes are due for servicing or when a fault with specific bikes has been recorded and are awaiting repair.
Another exception report could flag up bikes that have not been hired for a long time, so Ray's Rentals are made aware. This report could help the business order priority for repairs.
Exception could also be used to flag up records with missing details, i.e. customer records with missing address, number, etc., or bike records with missing details. The use of this will be very useful for Ray's Rentals, as it would allow them to ensure the accuracy of records in the system.
It could be possible to use exception reporting to find which bike records are no longer needed (after the warranty has expired) and are ready to be archived or deleted; this could be done by checking which records have exceeded 2 years after the date of sale.

6.3 Key Target Reports

Key Target Reports are a good way to keep up motivation for staff in the business and for management to monitor progress.
Ray's Rentals could use key target reports to monitor total sales over a specific period. The management would set target sales amount and would aim to reach this. At the end of a given period, management would see if targets were met and then produce an output report.
Key Target reports can also be used to measure customer satisfaction over a specified period. Obtaining feedback for customers would be easy for Ray's Rentals and would simply consist of either asking the customer in person if they were satisfied they were with the hire, or just give them a call after the hire to obtain their response. For example, if management set a satisfaction target of 95%, employees would aim to do as well as they can with customers. Ray's Rentals would try to make sure the hiring procedure for every customer is as smooth as possible. After management collates feedback and reports have been produced, figures could be used for advertisement and promotional purposes (i.e. mentioning a 95% customer satisfaction). When targets are constantly being met or are not being met, management may revise the target to increase or decrease.
Key Target reports could also be used for maintenance/fault repair targets. This report would show the percentage of the time of when repairs are done within a fixed number of days (i.e. percentage of repairs/maintenance completed within 3 working days). This target would be of great use, as meeting these targets are essential to ensure the popular bikes are available for hire.

7.0 Management Report Conclusion

The management report in this section has attended to the problems that Ray's Rentals were facing with the current system as well as the flaws in this system. The main problems found with the current system were with lost records, data protection issues,

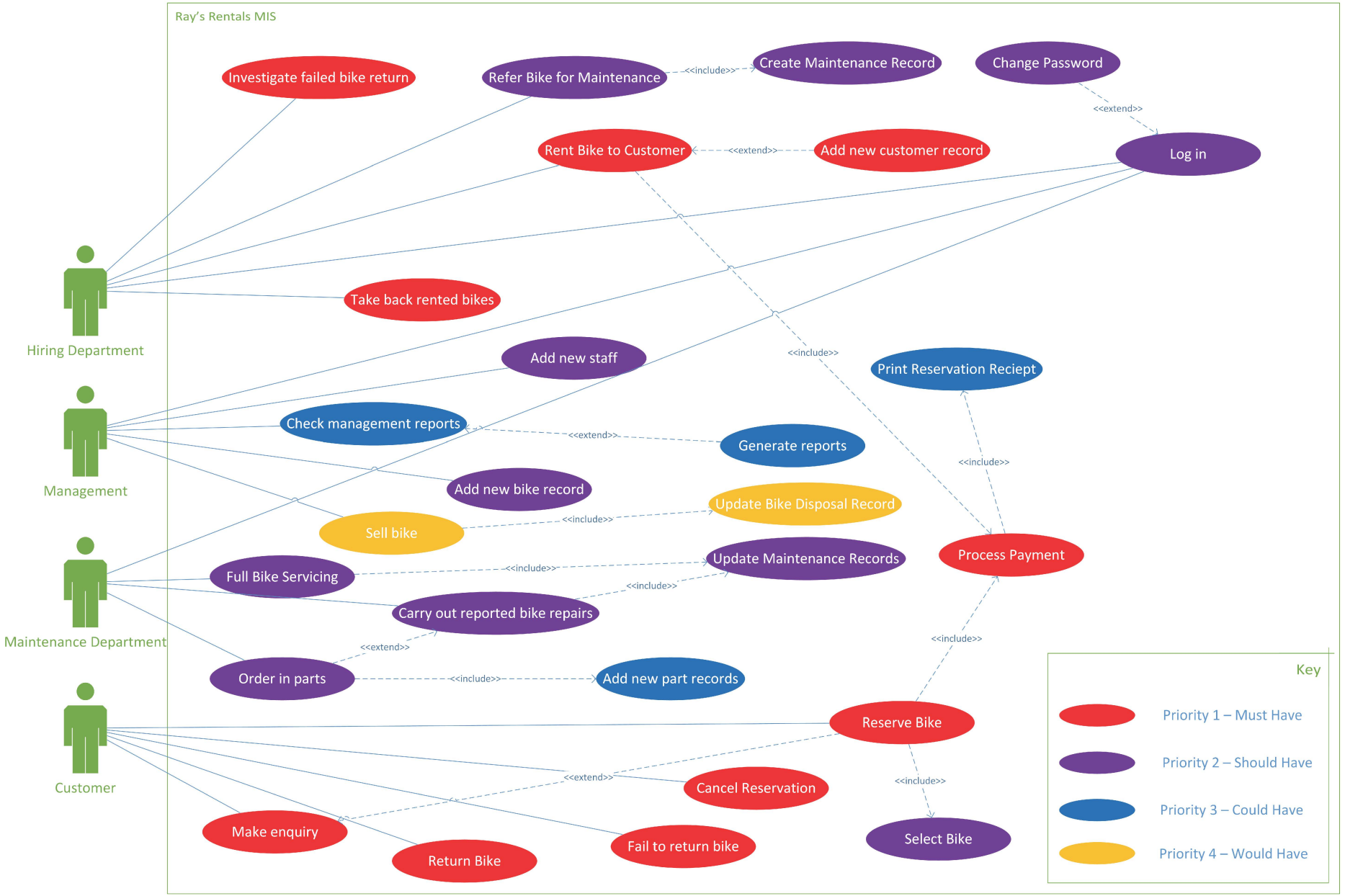
and clashing reservations. System requirements for the creation of a new computerised database system were then outlined. This system would be the resolution to the problems with the current system. Following this management reports were explained and then types of management reports were investigated and explored, as to how they could be used to assist Ray's Rentals to create a successful system. The final recommendation in this report is to replace the system with a computerised database system. The system requirements have explained in detail, as to what would need to be done. The other recommendation is to use the three highlighted management reports (Analysis, Key Target and Exception), alongside the new system within Ray's Rentals.

8.0 System Analysis and Design Introduction

In this section we will use the information from the case study as well as outlined system requirements from the management report as a base to analyse and design a new computerised management information system.

The section will commence by outlining all the processes that the new system should include. This is in the form of a use case diagram using MOSCOW prioritisation. A summary explaining decisions made while creating the diagram as well as what will be learnt from creating the diagram. Following this, two use case specifications will be created for two different essential processes within the system. These specifications will include top down entity relationship diagrams showing which entities data will be retrieved from to carry out the specific processes. In addition to this, this section of the report will then look at relational database analysis (RDA) of the two documents in the case study appendix. Bottom up entity relationship diagrams will then show how the normalised data from the documents links as well as including the attributes. Lastly, a group entity relationship diagram will be created. This will be a combination of the top down process as well as the bottom up process as used in previous parts of the report. This will include summary explaining decisions made while creating the diagram as well as what will be learnt from creating the ERD.

9.0 Use Case Diagram



9.1 Use Case Diagram Commentary

To create our use case diagram the first thing we did was to look at the case study and identify all of the potential actors. Following this we looked at both the case study and the appendix and highlighted all possible and potential main business processes we could find. We then listed out the processes and then went through each one and linked them to our potential actors. Following this we went back to the case study and documents and looked for further process we could find. Then we repeated as previously and linked them to the identified actors.
After looking at the case study to identify all of the actors and processes we then looked into our system requirements from our management report and added more processes based on these requirements. We then used Microsoft Visio 2013 to construct the use case diagram from all of the obtained information. Lastly, we looked at all of the processes and identified the priorities of each using the MOSCOW system of prioritisation.
While creating this diagram we have learnt an effective way to identify actors and processes from given information as well as thinking outside the box and adding further processes based on requirements. We learnt that the way we think while creating the diagram should be with external view and that we should not think like developers at this stage. We also learnt how to use Visio to create diagrams using stencils. We decided to use Visio instead of ArgoUML, as we found the functionality better and found it easier to use.

10.1 Haroon's Use Case Specification

By Haroon Bilimoria

Use Case: Take back rented bike
Actors: Hiring Department **Version:** 1.0
Pre-Conditions
The bike must be rented Must have a bike number assigned Reservation number exist beforehand Bike returned must match rented bike
Post-Conditions
Bike returned successfully Return with damages
Primary Path
1. The use case instigates when the customer walks inside the shop 2. Customer gives hiring department reservation number. 3. Hiring department searches the system to see if the reservation matches the bike being returned. 4. The hiring department will then check if the correct bike is being returned. 5. If the correct bike has been returned, then the hiring will fill in date and time on the reservation record. 6. The hiring department will print out a receipt for the customer and hand it to him/her. 7. The receipt will be used to prove that the customer has paid and returned the bike. 8. The use case ends successfully.

Alternate Path

Invalid Reservation Number

If in step 3 of the primary path use case, the reservation number does not match the bike being returned.

1. Hiring department will look at the returned bikes number and check which reservation it was for.
2. Then the hiring department will ask for additional details.
3. The hiring department asks for name/address/date of rental/.
4. If the details match then hiring will print receipt.
5. If the details do not match then customer will be asked to pay a fine.
6. Use case will then end.

Late return

If in step 1 the customer returns the bike late, then

1. The hiring department will ask for reservation number from the customer.
2. The hiring department will then check the bikes number
3. The hiring department will then match the bikes number on the system.
4. The hiring department will then state to the customer that the bike was returned late.
5. The hiring department will then ask the customer to pay a fine for returning the bike late.
6. The customer will then give the money to the hiring department.
7. The hiring department will then process the fine payment.
8. The hiring department will then print out a receipt for the customer.
9. The hiring department will the give the receipt to the customer.
10. The customer will leave the shop.
11. The use case then resumes at step 8.

Return with damages

If in step 1 the customer returns the bike with damages, then

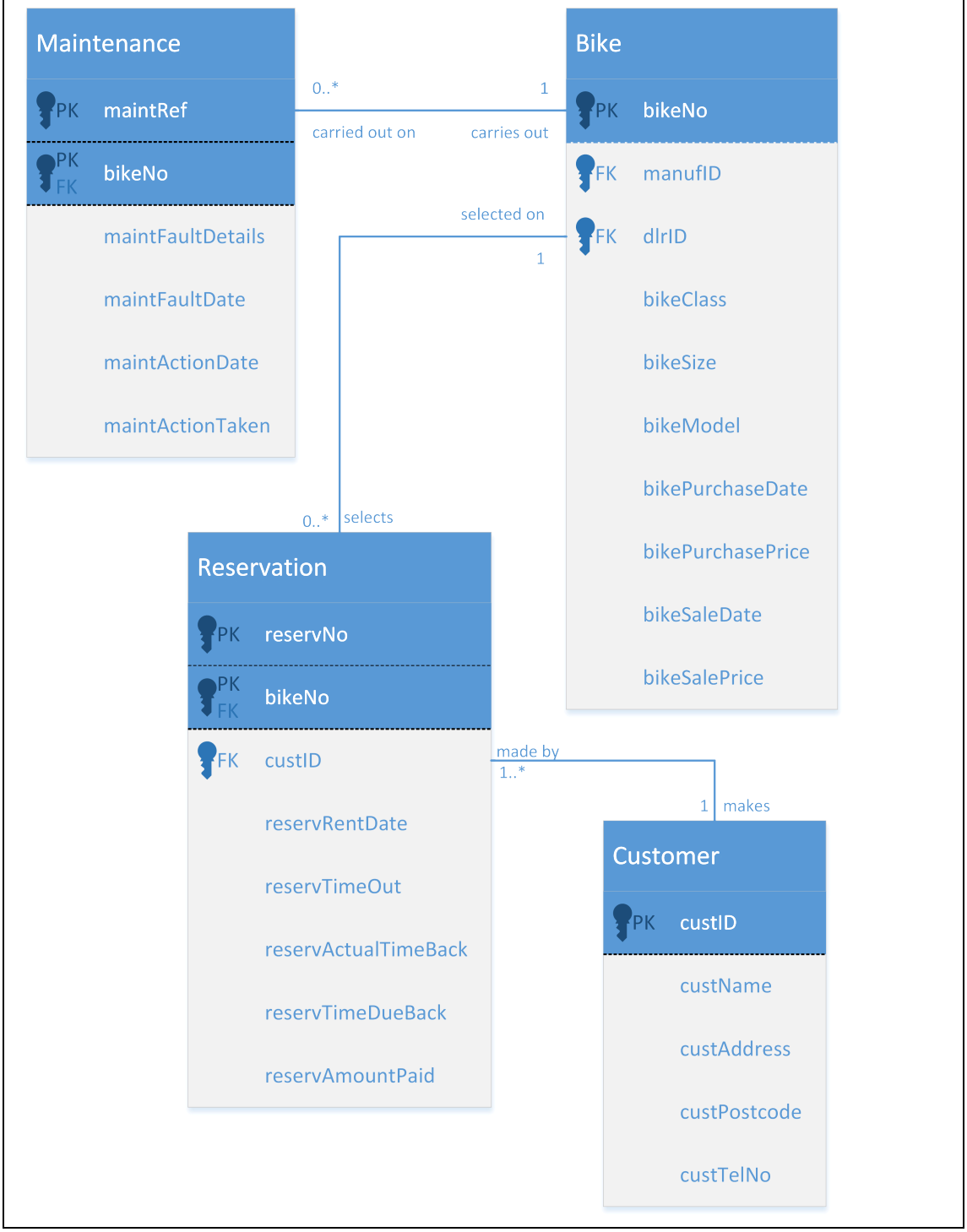
1. The hiring department will inspect the returned bike for any damages.
2. The hiring department will then record the damages on the system.
3. The hiring department will then calculate the fine costs to be made to the customer.
4. The hiring department will then ask the customer to pay a fine for returning the bike with damages.
5. The customer will then give the money to the hiring department.
6. The hiring department will then process the fine payment.
7. The use then resumes from step 4.

Reservation not present with customer

If in step 2 the customer does not have the reservation number with him, then

1. The hiring department will search the system for any additional details of the customer.
2. The hiring department will then match the number on the bike to the systems.
3. The hiring department will then state to the customer that they have to have the reservation number when they return the bikes next time.
4. The use case will then resume from step 4.

Entity Relationship Diagram



10.2 Huseyin's Use Case Specification

By Huseyin Arpalikli

Use Case: Carry Out Reported Bike Repairs including Update Maintenance Records

Actors: Maintenance Department

Version: 1.0

Pre-Conditions

Use Case: Create Maintenance Record has been carried out by the HIRING DEPARTMENT as part of repair referral for applicable bikes

Bike being repaired is has a record present in the system

Post-Conditions

Successful Completion: Bike is successfully repaired and maintenance records are updated accordingly

Failure Condition: Bike is not repaired and is recorded in the SYSTEM

Primary Path

1. The use case begins when MAINTENANCE DEPARTMENT accesses the maintenance records for new repair requests.
2. MAINTENANCE DEPARTMENT inspects the list sorted by popularity and selects the new maintenance record for the most popular bike that has recently been referred.
3. SYSTEM displays the bike details and maintenance details on screen.
4. The MAINTENANCE DEPARTMENT then retrieves the bike from storage.
5. MAINTENANCE DEPARTMENT checks that the reported damages are physically present and that repair is needed and/or possible.
6. MAINTENANCE DEPARTMENT checks parts required from suppliers are available.
7. Use Case: *Order in parts* and Use Case*: Add new part records* are performed.
8. Following receiving parts ordered the MAINTENANCE DEPARTMENT would carry out the repair.
9. When repair is complete the maintenance details and the part number are added to the maintenance record in the SYSTEM by the MAINTENANCE DEPARTMENT.
10. The bike is put back into storage by the MAINTENANCE DEPARTMENT for rental.
9. The use case ends successfully.

Alternate Path

Repair not needed after inspection

If after step 5 in the primary path, repair is not needed, then

1. The MAINTENANCE DEPARTMENT will open the appropriate maintenance record in the SYSTEM.
2. MAINTENANCE DEPARTMENT will enter a comment into action taken stating that the bike does not repair and a reason and then an action date will be recorded.

3. The use case ends with a failure condition.

Bike needing repair is out on rental and needs returning as soon as possible

If at step 3 in the primary path, the bike selected for repair is out on rental and has a serious fault, then

1. MAINTENANCE DEPARTMENT will check fault details in the SYSTEM.
2. MAINTENANCE DEPARTMENT will search for and inspect reservation record in the SYSTEM to find out whom the bike is rented out to.
3. MAINTENANCE DEPARTMENT will call the CUSTOMER and notify them that they need to return the bike as soon as possible due to serious faults and that they will be offered a replacement rental.
4. CUSTOMER returns bike, Use Case: *Return Bike*; HIRING DEPARTMENT records return of bike, Use Case: *Take back rented bikes.*
5. CUSTOMER Use Case: *Reserve Bike*; including *Select Bike*; HIRING STAFF Use Case: Rent bike to customer.
6. The use case resumes at step 4.

Bike needing repair is out on rental and will be repaired upon return

If in step 3 in the primary path, the bike selected for repair is out on rental and has a minor fault, then

1. MAINTENANCE DEPARTMENT will not enter anything into the SYSTEM and will cancel repair.
2. When time has come for customer to return rental CUSTOMER returns bike, Use Case: *Return Bike*; HIRING DEPARTMENT records return of bike, Use Case: *Take back rented bikes.*
3. The use case resumes at step 4.

Bike is too badly damaged to repair

If in step 5 in the primary path, the bike selected for repair is too badly damaged to repair, then

1. MAINTENANCE DEPARTMENT will contact MANAGEMENT to inform them that bike is beyond economical repair.
2. MAINTENANCE DEPARTMENT will take the bike to MANAGEMENT.
3. MANAGEMENT will inspect the bike and decide whether they agree with MAINTENANCE DEPARTMENT.
4. MANAGEMENT will inform MAINTENANCE DEPARTMENT of the decision.
5. MAINTENANCE DEPARTMENT will update maintenance record in the SYSTEM stating the decision and details.
6. MANAGEMENT will sell bike, Use Case: *Sell Bike* including Use Case: *Update Bike Disposal Record.*
7. The use case ends with a failure condition

Parts not needed to repair

If after step 5 in the primary path, the bike selected for repair does not need any new parts, then

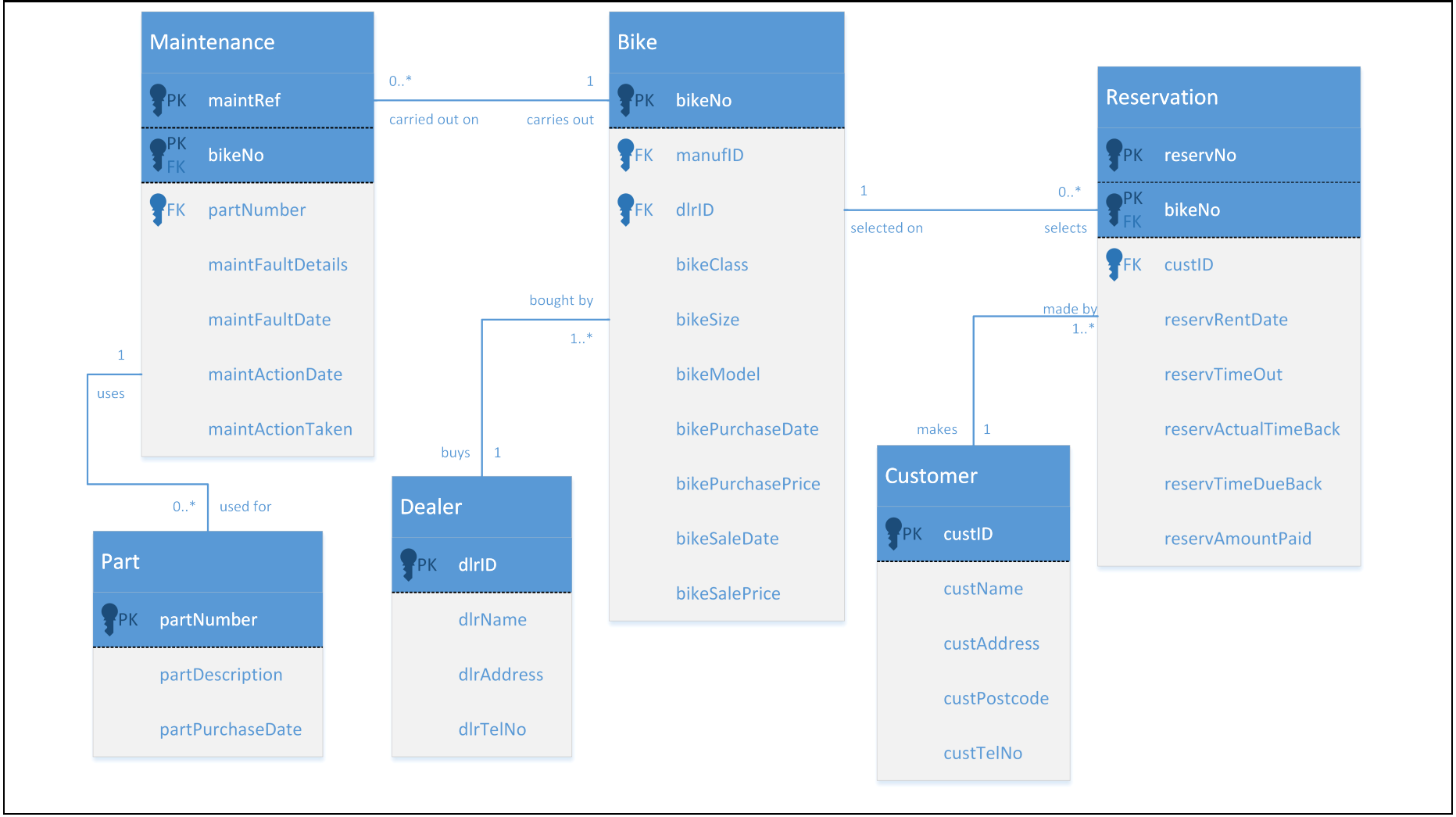
1. MAINTENANCE DEPARTMENT would carry out the repair.
2. MAINTENANCE DEPARTMENT would update maintenance record in the SYSTEM recording action details.
3. The use case resumes at step 10.

Parts needed are no longer available from any of the company's part suppliers

If in step 6 in the primary path, the parts required are not available from the suppliers, then

1. MAINTENANCE DEPARTMENT will attempt to repair bike without parts as much as possible.
2. Where parts are inevitably needed for repair, bike will be put back into storage.
3. MAINTENANCE DEPARTMENT will update the maintenance record stating that repair is pending order of new parts.
4. MAINTENANCE DEPARTMENT will notify MANAGEMENT that their part suppliers no longer stock the parts and will gain permission to order parts from an alternative supplier.
5. MAINTENANCE DEPARTMENT will order parts from alternative supplier. Use Case *Order: in parts* and Use Case*: Add new part records,* are performed.
6. The use case resumes at step 7.

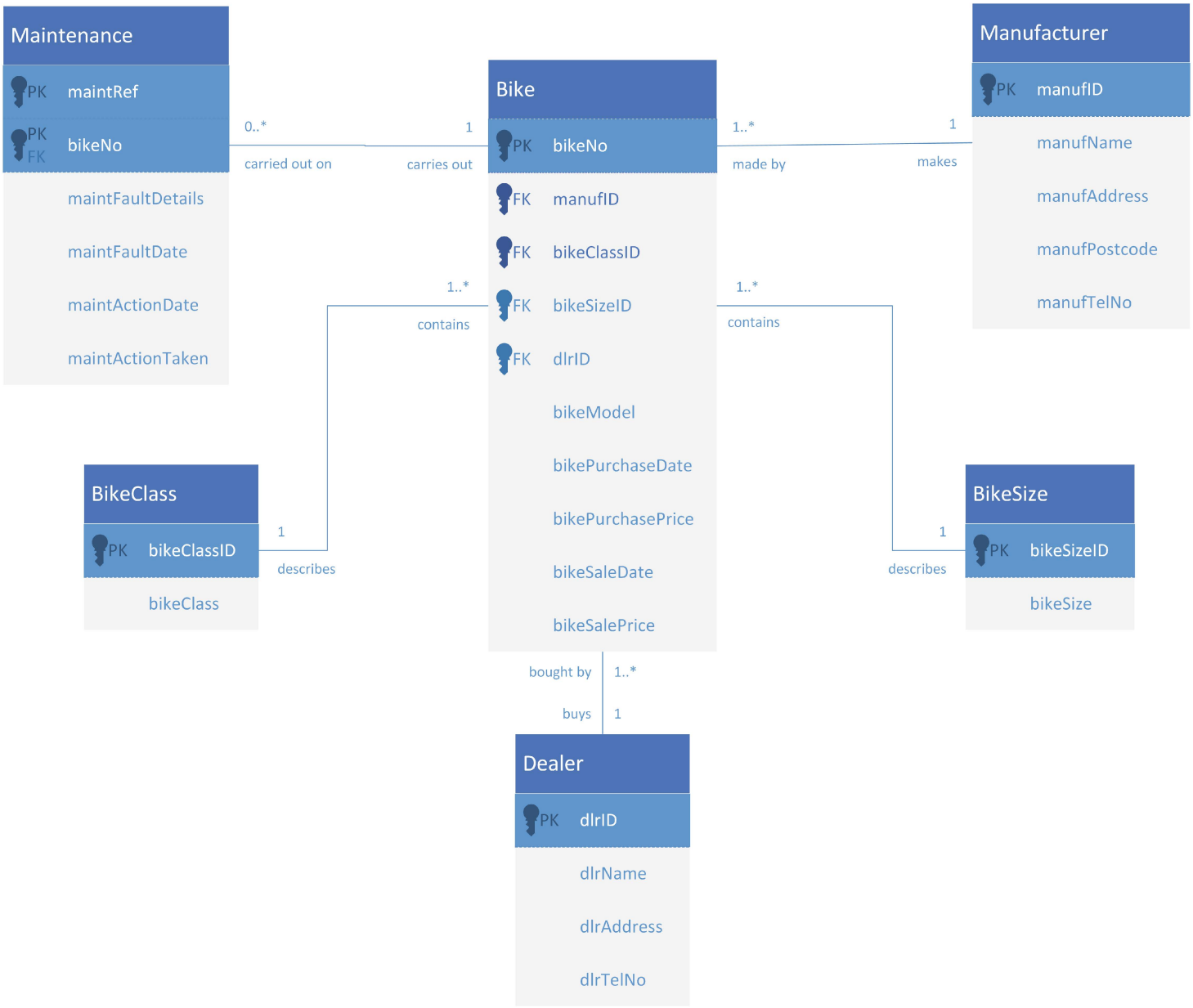
Entity Relationship Diagram



11.1 Relational Data Analysis for Bike Record

UNF	1NF	2NF	3NF
Bike Number	Bike Number	Bike Number	Bike Number
Bike Model	Bike Model	Bike Model	Bike Model
Bike Purchase Date	Bike Purchase Date	Bike Purchase Date	Bike Purchase Date
Bike Purchase Price	Bike Purchase Price	Bike Purchase Price	Bike Purchase Price
(Bike Classification ID)	Bike Classification ID	Bike Classification ID	Bike Classification ID*
Bike Classification	Bike Classification	Bike Classification	Bike Size ID*
(Bike Size ID)	Bike Size ID	Bike Size ID	Manufacturer ID*
Bike Size	Bike Size	Bike Size	Dealer ID*
(Manufacturer ID)	Manufacturer ID	Manufacturer ID	Bike Sale Price
Manufacturer Name	Manufacturer Name	Manufacturer Name	Bike Sale Date
Manufacturer Address	Manufacturer Address	Manufacturer Address	
Manufacturer Postcode	Manufacturer Postcode	Manufacturer Postcode	Bike Classification ID
Manufacturer Telephone	Manufacturer Telephone	Manufacturer Telephone	Bike Classification
(Dealer ID)	Dealer ID	Dealer ID	
Dealer Name	Dealer Name	Dealer Name	Bike Size ID
Dealer Address	Dealer Address	Dealer Address	Bike Size
Dealer Telephone	Dealer Telephone	Dealer Telephone	
Bike Sale Price	Bike Sale Price	Bike Sale Price	Manufacturer ID
Bike Sale Date	Bike Sale Date	Bike Sale Date	Manufacturer Name
Maintenance Ref No			Manufacturer Address
Maintenance Fault Details	Maintenance Ref No	Maintenance Ref No	Manufacturer Postcode
Maintenance Fault Date	Bike Number*	Bike Number*	Manufacturer Telephone
Maintenance Action Taken	Maintenance Fault Details	Maintenance Fault Details	
Maintenance Action Date	Maintenance Fault Date	Maintenance Fault Date	Dealer ID
	Maintenance Action Taken	Maintenance Action Taken	Dealer Name
	Maintenance Action Date	Maintenance Action Date	Dealer Address
			Dealer Telephone
			Maintenance Ref No
			Bike Number*
			Maintenance Fault Details
			Maintenance Fault Date
			Maintenance Action Taken
			Maintenance Action Date

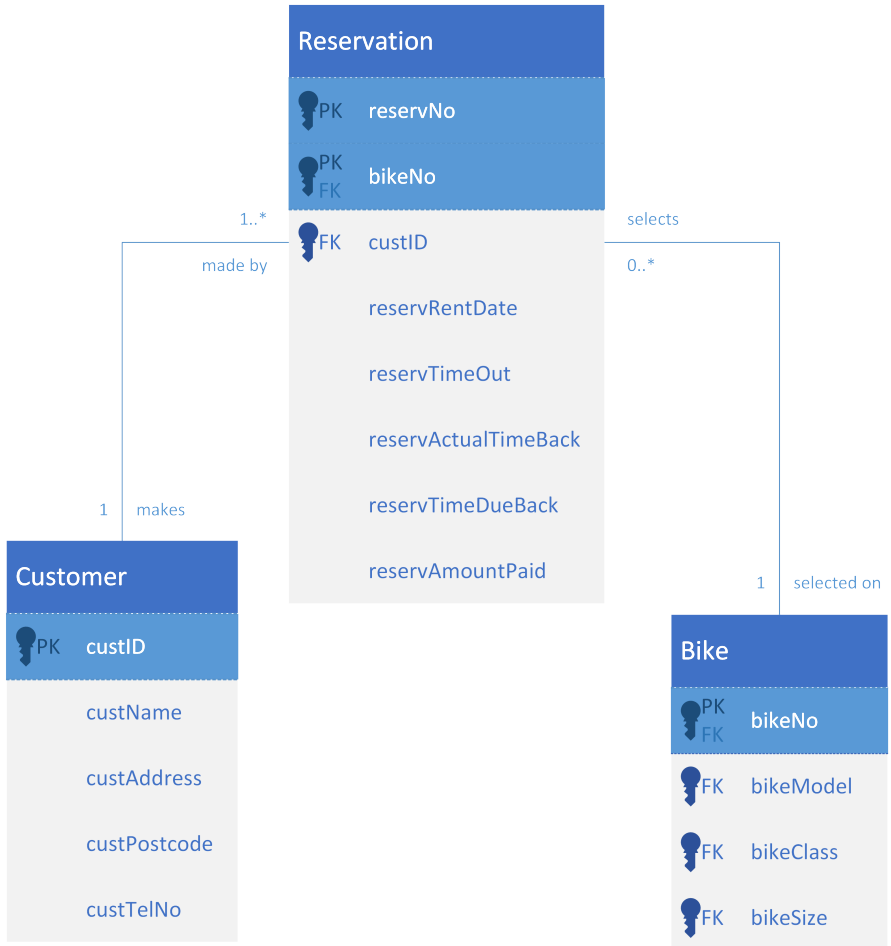
11.2 Bottom-Up Entity Relationship Diagram for Bike Record



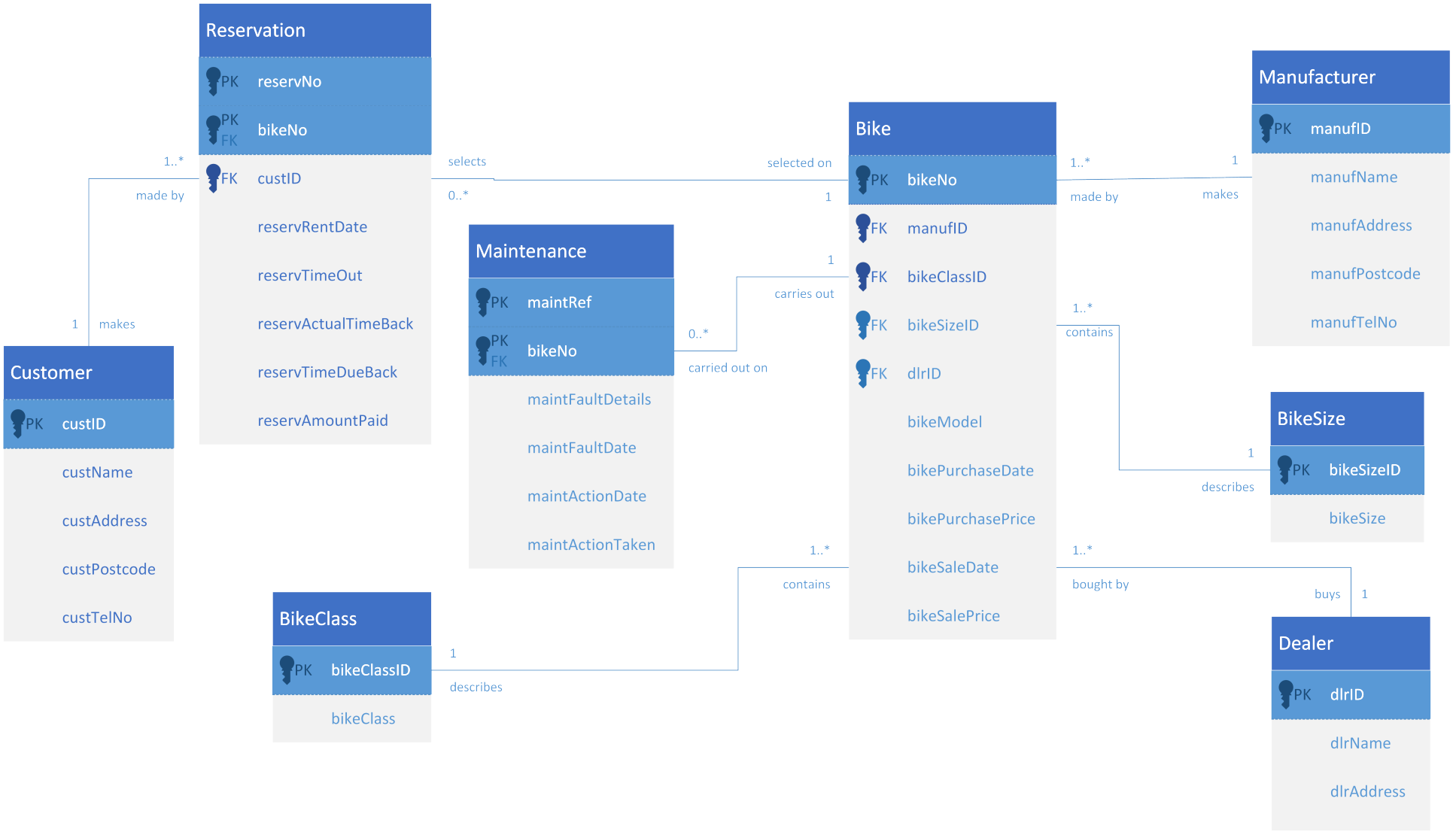
11.3 Relational Data Analysis for Rental Record

UNF	1NF	2NF	3NF
Bike Number*	Bike Number*	Bike Number*	Bike Number*
Bike Model	Bike Model	Bike Model	Bike Model
Classification	Classification	Classification	Classification
Size	Size	Size	Size
(Reservation Number)			
Rent Date	Reservation Number	Reservation Number	Reservation Number
Time Out	Bike Number*	Bike Number*	Bike Number*
Time Due Back	Rent Date	Rent Date	Rent Date
Actual Time Back	Time Out	Time Out	Time Out
(Customer ID)	Time Due Back	Time Due Back	Time Due Back
Customer Name	Actual Time Back	Actual Time Back	Actual Time Back
Customer Address	Customer ID	Customer ID	Customer ID*
Customer Postcode	Customer Name	Customer Name	Amount Paid
Customer Telephone	Customer Address	Customer Address	
Amount Paid	Customer Postcode	Customer Postcode	Customer ID
	Customer Telephone	Customer Telephone	Customer Name
	Amount Paid	Amount Paid	Customer Address
			Customer Postcode
			Customer Telephone

11.4 Bottom-Up Entity Relationship Diagram for Rental Record



11.5 Bottom-Up Entity Relationship Diagram for Merged Documents

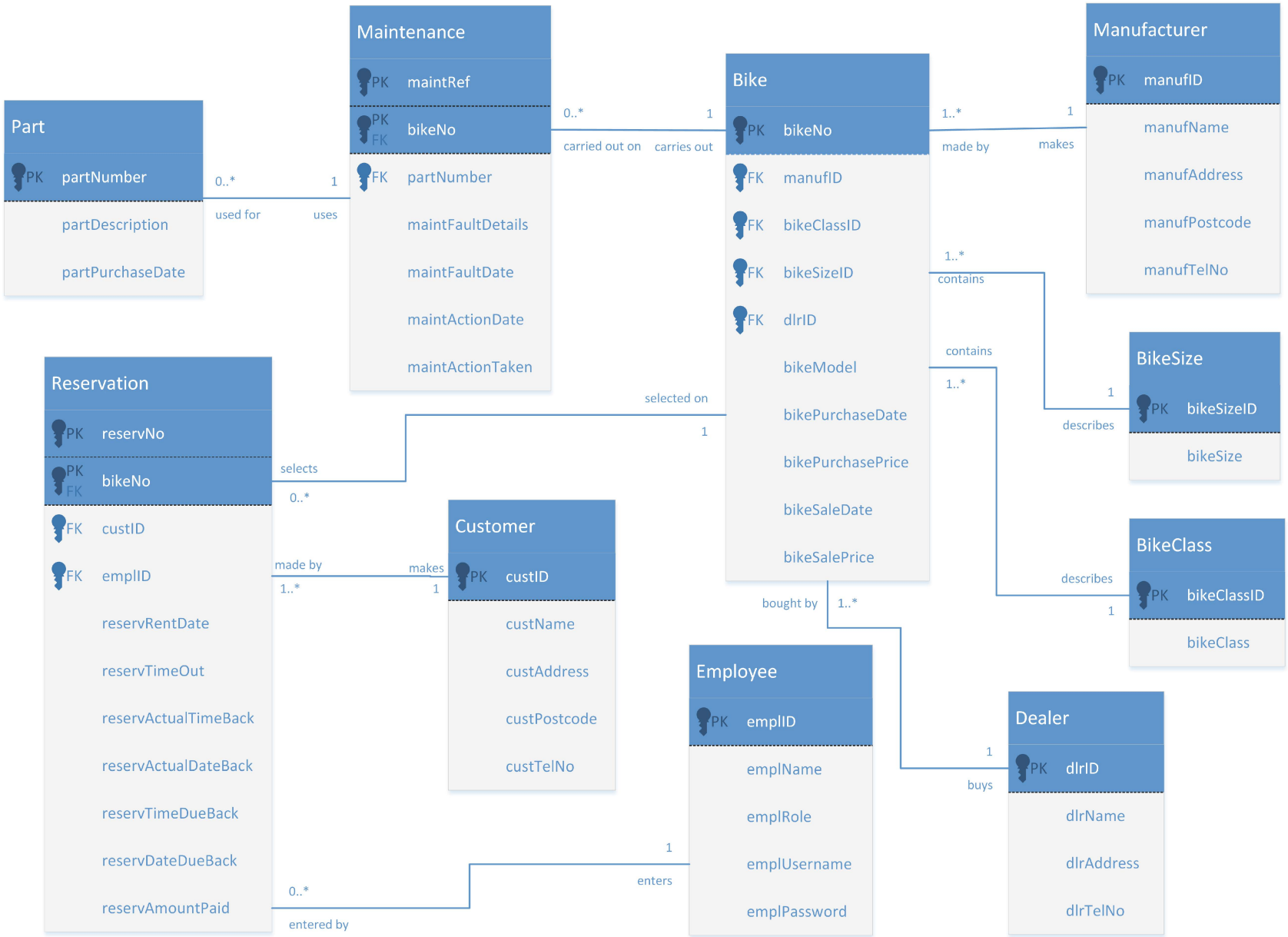


11.6 RDA Commentary and Footnotes

In this section the documents have been normalised to third normal form and two separate bottom-up ERD's have been created, then merged together to form into the third diagram.

In the relational database analysis above:
* - An asterisks notes a foreign key
_ - Underlining notes a primary key
() - Brackets note a new ID attribute added to identify some other attributes

12.0 Group ERD of Complete System



12.1 Group ERD Commentary

To create the above entity relationship diagram we have merged our merged bottom-up diagram and top-down diagrams. The bottom-up parts of the diagram (seen in section 4) are based solely on the documents in the case study. The top-down approach looked at the information in the case study and the documents. As you can see in our group entity relationship diagram, we have added some attributes that seem essential in the system in our opinion as well as further entities, which we judged were essential from information in the case study as well as our system requirements from our management report.

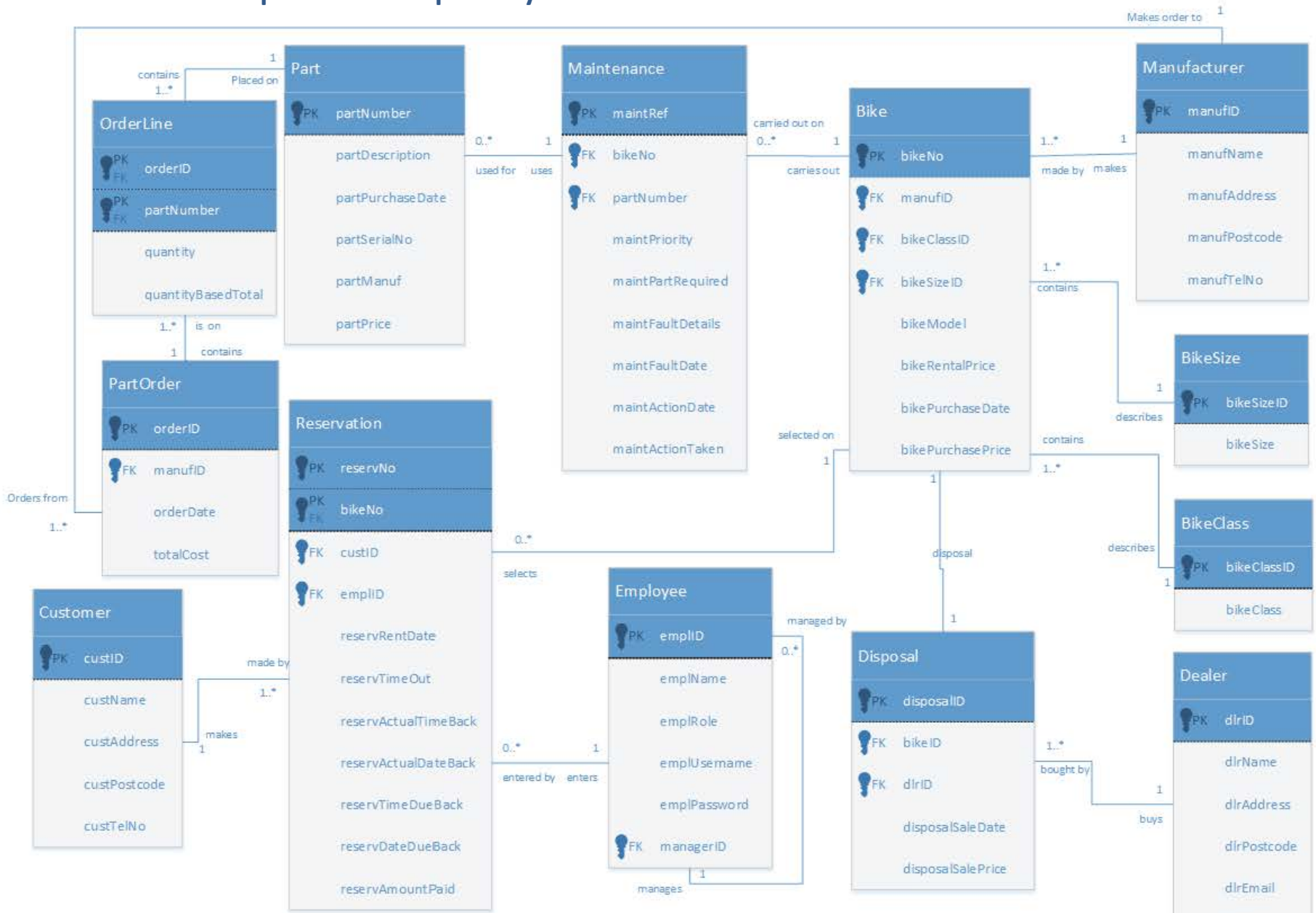
On top of the bottom-up entities and attributes we also added an employee entity as we plan to allow users to login to the system and also a part entity so record parts purchases. A notable attribute we added to reservation was date due back and actual date back. We found that these were missing from the reservation document in the case study and without it Ray's Rentals would not be able to identify the return date.

While creating this group entity relationship diagram we learnt how to analyse information and identify entities and attributes, as well as how to identify multiplicities between identified entities. As we used Visio, we also learnt how to use the stencil to create our ERD's.

13.0 Database Design & Oracle Implementation Introduction

In this part of the report, we will be looking at the database design for Ray's Rentals and will be implementing the database in Oracle DB. We will start by showing the amended group ERD along with some notes about changes, followed by a data dictionary and then the SQL queries inputted and the outputs produced in SQLDeveloper from these queries. We will then show our implemented select reports based on our chosen management reports discussed in part one of this project. Lastly, we will be reflecting what we learnt in this section of the project.

14.0 Amended Group ERD of Complete System



14.1 Amended Group ERD Commentary

Following the completion of part 2, feedback has assisted us in making some minor changes to our group entity relationship diagram.
The charges we made include the following:

- Added minor attributes: managerID in employee, dlrEmail in dealer
- Name change with disposal attributes to match the table name
- Moved sale details out of bike and into its own entity joining dealer and bike together. Even though the new disposal entity has a one to one relationship with bike, we still went ahead as we felt it would be pointless to have the sale details all in the bike record while the bike is not sold so we felt that with a separate entity, a new disposal record can then be created for the appropriate bike. I (Huseyin) have shown using my select query below that this entity can still be used successfully within the database.
- A new pig's ear relationship has been added within employee to show that the employee ID and manager ID have a relationship.
- Two new entities, Part Order and Order Line have been added; these are showing the order of parts as well as the specific parts on the order. Order Line is a weak entity linking Part Order and Part records.

15.0 Data Dictionary

Entity (Table) Name	***Attribute Name***	***Key Type***	***Data Type***	***FK Table***	***FK Column***	***Constraint***
Bike	bikeNo	PK	NUMBER(6)			Primary Key
	manufID	FK	NUMBER(6)	Manufacturer	manufID	Foreign Key; Not Null
	bikeClassID	FK	NUMBER(6)	BikeClass	bikeClassID	Foreign Key; Not Null
	bikeSizeID	FK	NUMBER(6)	BikeSize	bikeSizeID	Foreign Key; Not Null
	bikeModel		VARCHAR2(40)			Not Null
	bikeRentalPrice		NUMBER(9,2)			Not Null; Check>0
	bikePurchaseDate		DATE In format DD/MON/YY			Not Null;
	bikePurchasePrice		NUMBER(6)			Not Null; Check>0
Disposal	disposalID	PK	NUMBER(6)			Primary Key
	bikeNo	FK	NUMBER(6)	Bike	bikeNo	Foreign Key; Not Null; Unique

	dlrID	FK	NUMBER(6)	Dealer	dlrID	Foreign Key
	disposalSaleDate		DATE In format DD/MON/YY			Not Null
	disposalSalePrice		NUMBER(9, 2)			Not Null; Check>0
Dealer	dlrID	PK	NUMBER(6)			Primary Key
	dlrName		VARCHAR2(40)			Not Null
	dlrAddress		VARCHAR2(60			Not Null
	dlrPostcode		VARCHAR2(8)			Not Null
	dlrEmail		VARCHAR2(100)			
	dlrTelNo		NUMBER(12) Country code followed by number			Not Null; Unique
BikeClass	bikeClassID	PK	NUMBER(6)			Primary Key
	bikeClass		VARCHAR2(30)			Not Null; Check ('Mountain', 'Road', 'Tandem')
BikeSize	bikeSizeID	PK	NUMBER(6)			Primary Key
	bikeSize		VARCHAR2(30)			Not Null; Check('Large Male', 'Standard Male', 'Small Male', 'Standard Female', 'Child')
Manufacturer	manufID	PK	NUMBER(6)			Primary Key
	manufName		VARCHAR2(40)			Not Null
	manufAddress		VARCHAR2(60)			Not Null
	manufPostcode		VARCHAR2(8)			Not Null
	manufTelNo		NUMBER(12)			Not Null; Unique

			Country code followed by number			
Maintenance	maintRef	PK	NUMBER(6)			Primary Key
	bikeNo	FK	NUMBER(6)	Bike	bikeNo	Foreign Key; Not Null
	partNumber	FK	NUMBER(6)	Part	partNumber	Foreign Key
	maintPriority		VARCHAR2(10)			Not Null
	maintPartRequired		VARCHAR2(100)			
	maintFaultDetails		VARCHAR2(100)			Not Null
	maintFaultDate		DATE In format DD/MON/YY			Not Null
	maintActionDate		DATE In format DD/MON/YY			
	maintActionTaken		VARCHAR2(20)			
Part	partNumber	PK	NUMBER(6)			Primary Key
	partDescription		VARCHAR2(100)			Not Null
	partPurchaseDate		DATE In format DD/MON/YY			Not Null
	partSerialNo		NUMBER(15)			
	partManuf		VARCHAR2(36)			Not Null
	partPrice		NUMBER(9,2)			Not Null; Check >0
OrderLine	orderID	PK FK	NUMBER(6)	PartOrder	orderID	Primary Key; Foreign Key
	partNumber	PK FK	NUMBER(6)	Part	partNumber	Primary Key; Foreign Key
	quantity		NUMBER(2)			Not Null; Check >0
	quantityBasedTotal		NUMBER(9,2)			Not Null; Check >0
PartOrder	orderID	PK	NUMBER(6)			Primary Key

	manufID	FK	NUMBER(6)	Manufacturer	manufID	Foreign Key; Not Null
	orderDate		DATE In format DD/MON/YY			Not Null
	totalCost		NUMBER(9, 2)			Not Null; Check >0
Reservation	reservNo	PK	NUMBER(6)	Reservation	reservNo	Primary Key
	bikeNo	PK FK	NUMBER(6)	Bike	bikeNo	Primary Key; Foreign Key; Not Null
	custID	FK	NUMBER(6)	Customer	custID	Foreign Key; Not Null
	empID	FK	NUMBER(6)	Employee	empID	Foreign Key; Not Null
	reservRentDate		DATE In format DD/MON/YY			Not Null
	reservTimeOut		VARCHAR2(5) In format HH:MM			Not Null
	reservActualTimeBack		VARCHAR2(5) In format HH:MM			
	reservActualDateBack		DATE In format DD/MON/YY			
	reservTimeDueBack		VARCHAR2(5) In format HH:MM			Not Null
	reservDateDueBack		DATE In format DD/MON/YY			Not Null
	reservAmountPaid		NUMBER(9, 2)			
Customer	custID	PK	NUMBER(6)			Primary Key
	custName		VARCHAR2(60)			Not Null
	custAddress		VARCHAR2(70)			Not Null
	custPostcode		VARCHAR2(8)			Not Null

	custTelNo		NUMBER(12) Country code followed by number			Not Null; Unique
Employee	emplID	PK	NUMBER(6)			Primary Key
	emplName		VARCHAR2(40)			Not Null
	emplRole		VARCHAR2(30)			Not Null
	emplUsername		VARCHAR2(20)			Not Null; Unique
	emplPassword		VARCHAR2(20)			Not Null
	managerID	FK	NUMBER(6)	Employee	emplID	Foreign Key

16.0 SQL Scripts

All SQL create, insert and describe statement can be found in Appendices, Section 20.1.

16.1 Select Statements

Huseyin Arpalikli - QUERY 1: Display total reservation revenue by bikes in the last 12 months excluding disposed bikes as well as displaying the number of reservations, the age of bike, and bike details, and hence providing class, size and manufacturer names from their respective tables.

```
SELECT bikeNo "Bike Number", manufName "Manufacturer Name",  bikeModel "Model", bikeClass
"Classification", bikeSize "Size", bikeRentalPrice "Rental Price", bikePurchasePrice "Purchase Price",
ROUND(MONTHS_BETWEEN(SYSDATE,bikePurchaseDate),0) "Bike Age in Months",
SUM(reservAmountPaid) "Total Reservation Revenue",
COUNT(reservNo) "Total Reservations"

FROM bike NATURAL JOIN reservation NATURAL JOIN bikeClass NATURAL JOIN bikeSize
NATURAL JOIN manufacturer

WHERE MONTHS_BETWEEN (SYSDATE, reservRentDate) < 12
AND bikeNo NOT IN (SELECT disposal.bikeNo FROM disposal)

GROUP BY bikeNo, manufName, bikeModel, bikeClass, bikeSize, bikeRentalPrice, bikePurchasePrice,
ROUND(MONTHS_BETWEEN(SYSDATE,bikePurchaseDate),0)

ORDER BY SUM(reservAmountPaid) DESC;
```

Result:

SQL | All Rows Fetched: 4 in 0.016 seconds

	Bike Number	Manufacturer Name	Model	Classification	Size	Rental Price	Purchase Price	Bike Age in Months	Total Reservation Revenue	Total Reservations
1	3	Franklin Bicycles Productions Ltd	Partner Rider	Tandem	Large Male	10.5	120	5	31.5	2
2	5	Rowley Bikes	RoadRanger	Road	Standard Male	9.8	111.25	11	19.6	1
3	2	Mohammed Bell Manufacturing plc	Roadster	Road	Small Male	15	250	8	15	1
4	7	Rowley Bikes	LadyPower	Mountain	Standard Female	14	145.67	9	14	1

Haroon Bilimoria – Query 2: Displays maintenance details with bike details for maintenance records which are pending resolution, ordered by fault date.

```
SELECT bikeNo, manufName "Manufacturer", bikeModel, bikeClass, bikeSize, bikeRentalPrice,
maintref "Repair Ref", maintpriority, maintPartRequired, partnumber,
maintfaultdate,maintfaultdetails, maintactiondate, maintactiontaken

FROM bike NATURAL JOIN maintenance NATURAL JOIN bikeClass NATURAL JOIN bikeSize NATURAL
JOIN manufacturer

WHERE maintactionDate IS NULL

ORDER BY maintFaultDate ASC;
```

Result:

	BIKENO	Manufacturer	BIKEMODEL	BIKEC...	BIKESIZE	BIKERENTALPRICE	Repair Ref	MAINTPRIORITY	MAINTPARTR...	PARTNU...	MAINTFAUL...	MAINTFAULTDETAILS	MAINTACTIONDATE	MAINTA
1	6	Lees Cycles	KidRider	Road	Child	7.89	4	Low	Bike Saddle	4	11-NOV-13	Saddle worn, needs replacement	(null)	(null)
2	5	Rowley Bikes	RoadRanger	Road	Standard Male	9.8	6	Low	(null)	(null)	27-FEB-14	Full Servicing	(null)	(null)

16.2 Proof of data entry

Part Table

	PARTNUMBER	PARTDESCRIPTION	PARTPURCHASEDATE	PARTSERIALNO	PARTMANUF	PARTPRICE
1	1	3 Metre Superbike Chain	20-OCT-12	654235	SuperBike	35
2	2	18" Tyre	10-JAN-14	4857485094	Raleigh	35.2
3	3	Handlebars, complete	11-NOV-13	4885432	Lees Cycles	55.5
4	4	Medium Saddle	11-NOV-13	68366	Rowley Bikes	80
5	5	Brake wires 2 piece set	16-OCT-13	6373534	Currys	18.8

Bike Table

	BIKENO	MANUFID	BIKECLASSID	BIKESIZEID	BIKEMODEL	BIKERENTALPRICE	BIKEPURCHASEDATE	BIKEPURCHASEPRICE
1	1	3	2	4	SuperRider	12	23-MAR-12	265
2	2	3	2	3	Roadster	15	18-JUN-13	250
3	3	4	3	1	Partner Rider	10.5	29-SEP-13	120
4	4	2	1	4	MountRider	13.2	14-AUG-12	162.48
5	5	1	2	2	RoadRanger	9.8	20-MAR-13	111.25
6	6	2	2	5	KidRider	7.89	02-JAN-13	85.25
7	7	1	1	4	LadyPower	14	06-JUN-13	145.67

Bike Class Table

	BIKECLASSID	BIKECLASS
1	1	Mountain
2	2	Road
3	3	Tandem

Bike Size Table

	BIKESIZEID	BIKESIZE
1	1	Large Male
2	2	Standard Male
3	3	Small Male
4	4	Standard Female
5	5	Child

Customer Table

	CUSTID	CUSTNAME	CUSTADDRESS	CUSTPOSTCODE	CUSTTELNO
1	1	Liam Finch	13 Bassano Street, London	SE22 8RY	447988660192
2	2	Millie Giles	2 Kings Square, York	YO1 8BH	441484859678
3	3	Skye Wallace	111A Perrymans Farm Road, Ilford, Greater London	IG2 7LU	442085748575
4	4	Jake Miah	1A Shaw Street, Rochdale, Greater Manchester	OL12 9SN	441615847364
5	5	Lara Howarth	25 Old Park Hill, Dover, Kent	CT16 2AW	441584958675
6	6	Riley Pratt	24 Woodview Road, Norman Hill, Dursley, Gloucestershire	GL11 5RW	441948576858
7	7	Hayden Gibson	30-32 Gribble Road, Liverpool, Merseyside	L10 7NF	441475847395
8	8	Rosie Herbert	1 Reservoir Cottages, Catcleugh, Newcastle upon Tyne, Northumberland	NE19 1TX	441574837475
9	9	George Norris	London Road, South Downs National Park, Liss, Hampshire	GU33 7QJ	441847596876

Dealer Table

	DLRID	DLRNAME	DLRADDRESS	DLRPOSTCODE	DLRTELNO
1	1	Simons Scrap Ltd	83 Kingsway North HOLMSGARTH	ZE1 7EB	447001022952
2	2	Benjamin Brennan	55 Canterbury Road VIEWPARK	G71 0JN	447983057034
3	3	Jude Edwards	94 Vicar Lane SAUNDAIG	PA77 2JN	447800932079
4	4	Robert Allen	52 Felix Lane SHORNE	DA12 6TU	447967368497
5	5	Millie Ali	98 Walden Road GREATHAM	GU33 5ND	447079278709
6	6	Jordan Newman	88 Bridge Street GORS	SY23 9NT	447711007430
7	7	Stimtone	26 Highland Park, Feltham, Greater London	TW13 4QW	447930107694
8	8	Charlotte Joyce	84 Far Lane, Hepworth, Holmfirth, West Yorkshire	HD9 1TL	447930187694
9	9	UnaHow	80A Albert Drive, Deganwy, Conwy	LL31 9RH	447744869327
10	10	Zendexon Ltd	1A Brandon Road, Leeds, West Yorkshire	LS3 1AH	447815893773

Disposal Table

	DISPOSALID	BIKENO	DLRID	DISPOSALSALEDATE	DISPOSALSALEPRICE
1	1	1	1	11-JAN-14	120
2	2	4	3	21-FEB-14	86.5
3	3	6	7	18-DEC-13	45.2

Employee Table

	EMPLID	EMPLNAME	EMPLROLE	EMPLUSERNAME	EMPLPASSWORD	MANAGERID
1	1	Raymond Smith	Owner	RaySmith	admin	(null)
2	2	Lisa McDonald	Hire Manager	LisaMcD	lisa	1
3	3	Precious Igbinosun	Customer Services	PreciousIgbinosun	precious	2
4	4	Sanjit Khan	Parts Manager	SanKhan	sanjit	1
5	5	Simon Read	Maintenance	SimonRead	simon	4

Maintenance Table

	MAINTREF	BIKENO	PARTNUMBER	MAINTPRIORITY	MAINTPARTREQUIRED	MAINTFAULTDETAILS	MAINTFAULTDATE	MAINTACTIONDATE	MAINTACTIONTA
1	1	1	1	High	Bike Chain	Chain has snapped, replacement needed	19-OCT-12	22-OCT-12	Chain replaced
2	2	4	(null)	Low	(null)	Full Servicing	14-AUG-13	21-AUG-13	Servicing carr
3	3	2	5	Normal	Break Wires	Wires have worn out, replacement needed	16-OCT-13	19-OCT-13	Wires replaced
4	4	6	4	Low	Bike Saddle	Saddle worn, needs replacement	11-NOV-13	(null)	(null)
5	5	3	2	High	Bike Tyre	Bike Tyres need replacing x2	09-JAN-14	10-JAN-14	Tyres replaced
6	6	5	(null)	Low	(null)	Full Servicing	27-FEB-14	(null)	(null)

Manufacturer Table

	MANUFID	MANUFNAME	MANUFADDRESS	MANUFPOSTCODE	MANUFTELNO
1	1	Rowley Bikes	19 Mawfa Avenue, Sheffield, South Yorkshire	S14 1AJ	441574837475
2	2	Lees Cycles	66 Wansbeck Road, Jarrow, Tyne and Wear	NE32 5SS	447827373694
3	3	Mohammed Bell Manufacturing plc	26 Holland Road, Marlow, Buckinghamshire	SL7 1UJ	441485869584
4	4	Franklin Bicycles Productions Ltd	2 Boonwood Agric Cottages, Gosforth, Seascale, Cumbria	CA20 1EA	441586473744

OrderLine Table

	ORDERID	PARTNUMBER	QUANTITY	QUANTITYBASEDTOTAL
1	1	1	2	70
2	2	3	1	55.5
3	2	4	1	80
4	3	5	1	18.8
5	4	2	2	70.4

Part Order Table

	ORDERID	MANUFID	ORDERDATE	TOTALCOST
1	1	2	22-FEB-14	70
2	2	3	11-NOV-13	135.5
3	3	4	16-OCT-13	18.8
4	4	1	10-JAN-14	70.4

Reservation Table

	RESERVNO	BIKENO	CUSTID	EMPLID	RESERVRENTDATE	RESERVTIMEOUT	RESERVTIMEDUEBACK	RESERVDATEDUEBACK	RESERVACTUALTIMEBACK	RESERVACTUALDATEBACK	RESERVAMOUNTPAID
1	1	1	7	1	13-APR-12	09:50	17:00	13-APR-12	17:00	13-APR-12	12
2	2	2	3	2	18-OCT-13	07:30	15;00	18-OCT-13	14:20	18-OCT-13	15
3	3	4	1	3	03-NOV-13	11:00	21:00	03-NOV-13	21:05	03-NOV-13	13.3
4	4	3	4	2	21-NOV-13	15;30	15:30	22-NOV-13	15:30	22-NOV-13	21
5	5	5	2	1	11-DEC-13	10:30	16;00	12-DEC-13	15:30	12-DEC-13	19.6
6	6	1	6	2	22-JAN-14	12:20	18:20	23-DEC-13	18:30	23-JAN-14	18
7	7	6	9	3	03-JAN-14	13:30	18;00	04-JAN-14	17:25	04-JAN-14	15.78
8	8	3	7	1	06-FEB-14	09:45	19:45	06-FEB-14	19:45	06-FEB-14	10.5
9	9	7	5	3	12-FEB-14	09:00	19:00	12-FEB-14	19:00	12-FEB-14	14
10	10	6	7	1	18-FEB-14	13:30	18;00	18-FEB-14	18:00	18-FEB-14	15.78

17.0 Reflections on what we learnt during this project

17.1 Huseyin's Reflections

During the process of creating the database using Oracle as well as implementing queries such as create, insert, drop, select, describe and sequences, I feel I have now learnt how to create a fully functioning database that can run queries to display reports, which would help the business, Ray's Rentals, to have better control over the various areas of the business. For example, maintenance is fully linked with the other tables and therefore faulty bikes can be seen clearly and fixed more promptly. Before creating the tables we noticed some issued with our original ERD and we quickly resolved these by modifying the ERD to fix the issues. I have also been able to learn how to create a concise data dictionary.

Using the insert commands, I have learnt that I can enter data in bulk or in the alternative method we have learnt using the '&' symbol for manual entry for each table. As well as all this, I have independently learnt about using sequences to increment values in the database and how to create and drop sequences and insert sequence values into records are various tables.

Personally, the part where I felt I learnt the most was creating my own select statement based on the analysis management report. Through this I felt that I learnt about many different functions and ways to display data. I experimented with various different combinations to get my result and I looked for the best way to implement (such as using 'NOT IN' instead of 'NOT EXISTS'). Before implementing my select query, I wrote out on a piece of paper exactly what I wanted my report to display and I successfully completed this using all of the functions required and also used functions which I learnt independently. I have learnt much about the natural join function, too and feel that

although, I may have provided a lot of detail on the report I feel that this showed a good understanding of using natural joins.
Overall I have learnt a lot about complex SQL functions and the many possibilities in using these functions for select statements to implement various management reports. After doing this project, I now feel confident that I could work with SQL to manage databases at a business level.

17.2 Haroon's Reflections

Whilst learning how to create databases using Oracle, I have also learnt how to make database reports for Ray's Rentals. Also, I learnt how to make a data dictionary. I have learnt that I do not need to add data manually, but I can do it all together at the same time in Oracle SQL. Also I have learned that sequences can also be used to add values within fields in the database tables. In conclusion, I have learnt so much more about SQL statements and how they could benefit a company like Ray's Rentals and how they can make the business more manageable.

18.0 Project Conclusion

Following the completion of this report it is recommended for Ray's Rentals that the business should now move over from using the paper based system to the new database system created. If they plan to become further successful, this would be ideal, as it would mean that expanding the business would be easy as in terms of managing the data and the day to day running within the business. Along with this, we have been able to create some important management reports, which could be used within Ray's Rentals and would surely provide support to management to improve the business in terms of revenue and efficiency and quality of customer service and repairs.

18.1 Huseyin's Conclusion

During this project, I have learnt a lot about systems development. I have learnt about analysing existing systems and how these can be replaced with database systems. To find out how this could be implemented, I have drawn up information on the issues with the current system and the requirements for a new system. Following this, an analysis on management reports, which could be integrated into Ray's Rentals' system, was carried out.
I then learnt about different ways to represent data and information essential for designing a successful system. These include use case diagrams and entity relationship diagrams. One of the most important skills I have learnt, as part of this process is normalising data given from flat files or paper records to create databases.
After designing the actual system, I then learnt a lot about how to use Oracle SQL, as well as the design of the database with the use of data dictionaries and the SQL create queries. After actually creating the database, I have learnt about using different functions to create management reports based on our chosen reports in the first part of this project.
Overall, I have found that the project has provided me with some essential skills and has made me more confident in designing and using databases within real businesses.

18.2 Haroon's Conclusion

From taking part in this project there was many things that I learnt. One of the first things that I learnt was that so businesses use a paper-based system to log down their events. Mainly for keeping track of what goes on within the business. From this I learnt that now you can keep track of everything through electronic databases. By doing this it keeps the company more manageable. Also let's you track down any events that have occurred in the past very quickly.

Also another thing that I learnt from doing this project was the use of management information reports. These reports would split into 3 types. By using these three types of reward you are able to keep you company regulated and make targets for you. Because one of the main priorities of opening up you own company would be to make profit. To do this you would have to make reports for yourself so you can look back at them as a reference,
Also another thing that I learned was the Use Case Diagrams. I learned how to make these diagrams and what they would be used for around the business.
Another key aspect that I learnt from this project was the purpose of making SQL scripts. I learned what they are used and how they would keep your company clean, organised and manageable. Also they would be used to keep track of events around the company. These are the main things that I learned from doing the project, and it has boosted my knowledge the way information is used in everyday companies.

19.0 Bibliography

Gov.uk. 2013. *Data protection and your business - GOV.UK*. [online] Available at: https://www.gov.uk/data-protection-your-business/overview [Accessed: 21 Oct 2013].

Info Systems: mgt info & decision making. 2013. [PDF] Manchester: MMU. pp. 5-7. Available at: http://moodle.mmu.ac.uk/pluginfile.php/918647/mod_resource/content/4/Mgt%20Info%20%20Decision%20Making_sn.pdf [Accessed: 22 Oct 2013].

Institute for Grocery Distribution Information Unit. 1999. *Market Share of Grocery Industry*. [image online] Available at: http://thebritishgeographer.weebly.com/uploads/1/1/8/1/11812015/3920429.png?655 [Accessed: 23 Oct 2013].

Micropalsystems.com. 2013. *About document imaging & management*. [online] Available at: http://www.micropalsystems.com/paperproblems.asp [Accessed: 19 Oct 2013].

Searchsqlserver.techtarget.com. 2013. *What is relational database? - Definition from WhatIs.com*. [online] Available at: http://searchsqlserver.techtarget.com/definition/relational-database [Accessed: 24 Oct 2013].

Small Business - Chron.com. 2013. *What Are the Basic Kinds of Reports Produced by a Management Information System*. [online] Available at: http://smallbusiness.chron.com/basic-kinds-reports-produced-management-information-system-30845.html [Accessed: 23 Oct 2013].

Whiteley, D. 2013. Data and Information. *An introduction to information systems*. Basingstoke: Palgrave Macmillan.

20.0 Appendices

20.1 SQL Scripts

18.1.1 Drop Statements

```
DROP TABLE maintenance;
DROP TABLE disposal;
DROP TABLE reservation;
DROP TABLE employee;
DROP TABLE orderLine;
DROP TABLE partOrder;
DROP TABLE bike;
DROP TABLE customer;
DROP TABLE part;
DROP TABLE manufacturer;
DROP TABLE dealer;
DROP TABLE bikeClass;
DROP TABLE bikeSize;

DROP SEQUENCE BIKECLASSID_SEQ;
DROP SEQUENCE BIKENO_SEQ;
DROP SEQUENCE BIKESIZEID_SEQ;
DROP SEQUENCE CUSTID_SEQ;
DROP SEQUENCE DISPOSALID_SEQ;
DROP SEQUENCE DLRID_SEQ;
DROP SEQUENCE EMPLID_SEQ;
DROP SEQUENCE MAINTREF_SEQ;
DROP SEQUENCE MANUFID_SEQ;
DROP SEQUENCE ORDERID_SEQ;
DROP SEQUENCE PARTNUMBER_SEQ;
DROP SEQUENCE RESERVNO_SEQ;
```

20.1.2 Create Statements

```
CREATE TABLE dealer (
dlrID NUMBER(6) CONSTRAINT dlr_dlrID_pk PRIMARY KEY,
dlrName VARCHAR2(40) CONSTRAINT dlr_dlrName_nn NOT NULL,
dlrAddress VARCHAR2(60) CONSTRAINT dlr_dlrAddress_nn NOT NULL,
dlrPostcode VARCHAR2(8) CONSTRAINT dlr_dlrPostcode_nn NOT NULL,
dlrTelNo NUMBER(12) CONSTRAINT dlr_dlrTelNo_nn NOT NULL
                                        CONSTRAINT dlr_dlrTelNo_uk UNIQUE);

CREATE TABLE bikeClass (
bikeClassID NUMBER(6) CONSTRAINT biCl_bikeClassID_pk PRIMARY KEY,
bikeClass VARCHAR2(30) CONSTRAINT biCl_bikeClass_nn NOT NULL
CONSTRAINT biCl_bikeClass_ch CHECK(bikeClass IN ('Mountain', 'Road', 'Tandem')) );

CREATE TABLE bikeSize (
bikeSizeID NUMBER(6) CONSTRAINT biSi_bikeSizeID_pk PRIMARY KEY,
bikeSize VARCHAR2(30) CONSTRAINT biSi_bikeSize_nn NOT NULL
CONSTRAINT biSi_bikeSize_ch CHECK(bikeSize IN ('Large Male', 'Standard Male', 'Small
Male', 'Standard Female', 'Child')) );

CREATE TABLE manufacturer (
```

```
manufID NUMBER(6) CONSTRAINT man_manufID_pk PRIMARY KEY,
manufName VARCHAR2(40) CONSTRAINT man_manufName_nn NOT NULL,
manufAddress VARCHAR2(60) CONSTRAINT man_manufAddress_nn NOT NULL,
manufPostcode VARCHAR2(8) CONSTRAINT man_manufPostcode_nn NOT NULL,
manufTelNo NUMBER(12) CONSTRAINT man_manufTelNo_nn NOT NULL
                        CONSTRAINT man_manufTelNo_uk UNIQUE);

CREATE TABLE part (
partNumber NUMBER(6) CONSTRAINT part_partNumber_pk PRIMARY KEY,
partDescription VARCHAR2(100) CONSTRAINT part_partDesc_nn NOT NULL,
partPurchaseDate DATE CONSTRAINT part_partPurchDate_nn NOT NULL,
partSerialNo NUMBER(15),
partManuf VARCHAR2(36) CONSTRAINT part_partManuf_nn NOT NULL,
partPrice NUMBER(9,2) CONSTRAINT part_partPrice_nn NOT NULL
CONSTRAINT part_partPrice_ch CHECK(partPrice>0));

CREATE TABLE customer (
custID NUMBER(6) CONSTRAINT cust_custID_pk PRIMARY KEY,
custName VARCHAR2(60) CONSTRAINT cust_custName_nn NOT NULL,
custAddress VARCHAR2(70) CONSTRAINT cust_custAddress_nn NOT NULL,
custPostcode VARCHAR2(8) CONSTRAINT cust_custPostcode_nn NOT NULL,
custTelNo NUMBER(12) CONSTRAINT cust_custTelNo_nn NOT NULL
                        CONSTRAINT cust_custTelNo_uk UNIQUE);

CREATE TABLE bike (
bikeNo NUMBER(6) CONSTRAINT bi_bikeNo_pk PRIMARY KEY,
manufID NUMBER(6) CONSTRAINT bi_manufID_fk REFERENCES manufacturer(manufID)
CONSTRAINT bi_manufID_nn NOT NULL,
bikeClassID NUMBER (6) CONSTRAINT bi_biClassID_fk REFERENCES
bikeClass(bikeClassID)
CONSTRAINT bi_biClassID_nn NOT NULL,
bikeSizeID NUMBER (6) CONSTRAINT bi_biSizeID_fk REFERENCES bikeSize(bikeSizeID)
CONSTRAINT bi_biSizeID_nn NOT NULL,
bikeModel VARCHAR2(40) CONSTRAINT bi_biModel_nn NOT NULL,
bikeRentalPrice NUMBER(9,2) CONSTRAINT bi_biRentalPrice_nn NOT NULL
CONSTRAINT bi_bikeRentalPrice_ch CHECK(bikeRentalPrice>0),
bikePurchaseDate DATE CONSTRAINT bi_biPurchaseDate_nn NOT NULL,
bikePurchasePrice NUMBER(9,2) CONSTRAINT bi_biPurchasePrice_nn NOT NULL
CONSTRAINT bi_bikePurchasePrice_ch CHECK(bikePurchasePrice>0) );

CREATE TABLE partOrder (
orderID NUMBER(6) CONSTRAINT partOrder_orderID_pk PRIMARY KEY,
manufID NUMBER(6) CONSTRAINT partOrder_manufID_fk REFERENCES
manufacturer(manufID)
                        CONSTRAINT partOrder_manufID_nn NOT NULL,
orderDate DATE CONSTRAINT partOrder_orderDate_nn NOT NULL,
totalCost NUMBER(9,2) CONSTRAINT partOrder_totalCost_nn NOT NULL
CONSTRAINT partOrder_totalCost_ch CHECK(totalCost>0) );

CREATE TABLE orderLine (
orderID NUMBER(6) CONSTRAINT ordLi_orderID_fk REFERENCES partOrder(orderID),
partNumber NUMBER(6) CONSTRAINT ordLi_partID_fk REFERENCES part(partNumber),
```

```
quantity NUMBER(2) CONSTRAINT ordLi_quantity_nn NOT NULL
CONSTRAINT ordLi_quantity_ch CHECK(quantity>0),
quantityBasedTotal NUMBER(9,2) CONSTRAINT ordLI_quantTotal_nn NOT NULL
                        CONSTRAINT ordLi_quantityBasedTotal_ch
CHECK(quantityBasedTotal>0),
CONSTRAINT ordLi_comp_pk PRIMARY KEY (orderID, partNumber) );

CREATE TABLE employee (
emplID NUMBER(6) CONSTRAINT empl_emplID_pk PRIMARY KEY,
emplName VARCHAR2(40) CONSTRAINT empl_emplName_nn NOT NULL,
emplRole VARCHAR2(30) CONSTRAINT empl_emplRole_nn NOT NULL,
emplUsername VARCHAR2(20) CONSTRAINT empl_emplUsername_nn NOT NULL
                                    CONSTRAINT empl_emplUsername_uk UNIQUE,
emplPassword VARCHAR2(20) CONSTRAINT empl_emplPassword_nn NOT NULL,
managerID NUMBER(6) CONSTRAINT empl_managerID_fk REFERENCES
employee(emplID));

CREATE TABLE reservation (
reservNo NUMBER(6),
bikeNo NUMBER(6) CONSTRAINT reserv_bikeNo_fk REFERENCES bike(bikeNo),
custID NUMBER(6) CONSTRAINT reserv_custID_fk REFERENCES customer(custID)
                                    CONSTRAINT reserv_custID_nn NOT NULL,
emplID NUMBER(6) CONSTRAINT reserv_emplID_fk REFERENCES employee(emplID)
                        CONSTRAINT reserv_emplID_nn NOT NULL,
reservRentDate DATE CONSTRAINT reserv_reservRentDate_nn NOT NULL,
reservTimeOut VARCHAR2(5) CONSTRAINT reserv_reservTimeOut_nn NOT NULL,
reservTimeDueBack VARCHAR2(5) CONSTRAINT reserv_reservTimeDueBack_nn NOT
NULL,
reservDateDueBack DATE CONSTRAINT reserv_reservDateDueBack_nn NOT NULL,
reservActualTimeBack VARCHAR2(5),
reservActualDateBack DATE,
reservAmountPaid NUMBER(9,2),
CONSTRAINT reserv_comp_pk PRIMARY KEY (reservNo, bikeNo) );

CREATE TABLE disposal (
disposalID NUMBER(6) CONSTRAINT dis_disposalID_pk PRIMARY KEY,
bikeNo NUMBER(6) CONSTRAINT dis_bikeNo_fk REFERENCES bike(bikeNo)
                                    CONSTRAINT dis_bikeNo_nn NOT NULL
                                    CONSTRAINT dis_bikeNo_uk UNIQUE,
dlrID NUMBER(6) CONSTRAINT dis_dlrID_fk REFERENCES dealer(dlrID),
disposalSaleDate DATE CONSTRAINT dis_disposalSaleDate_nn NOT NULL,
disposalSalePrice NUMBER(9,2) CONSTRAINT dis_disposalSalePrice_nn NOT NULL
CONSTRAINT dis_disposalSalePrice_ch CHECK (disposalSalePrice>0) );

CREATE TABLE maintenance (
maintRef NUMBER(6) CONSTRAINT maint_maintRef_pk PRIMARY KEY,
bikeNo NUMBER(6) CONSTRAINT maint_bikeNo_fk REFERENCES bike(bikeNo)
                                    CONSTRAINT maint_bikeNo_nn NOT NULL,
partNumber NUMBER(6) CONSTRAINT maint_partNumber_fk REFERENCES
part(partNumber),
maintPriority VARCHAR2(10) CONSTRAINT maint_maintPriority_nn NOT NULL,
maintPartRequired VARCHAR2(100),
```

```
maintFaultDetails VARCHAR2(100) CONSTRAINT maint_maintFaultDetails_nn NOT NULL,
maintFaultDate DATE CONSTRAINT maint_maintFaultDate_nn NOT NULL,
maintActionDate DATE,
maintActionTaken VARCHAR2(100) );
```

20.1.3 Insert Statements

```
INSERT INTO bikeclass (bikeclassid,bikeclass)
VALUES (bikeclassid_seq.nextval,'Mountain');
INSERT INTO bikeclass (bikeclassid,bikeclass)
VALUES (bikeclassid_seq.nextval,'Road');
INSERT INTO bikeclass (bikeclassid,bikeclass)
VALUES (bikeclassid_seq.nextval,'Tandem');

INSERT INTO bikesize (bikesizeid,bikesize)
VALUES (bikesizeid_seq.nextval,'Large Male');
INSERT INTO bikesize (bikesizeid,bikesize)
VALUES (bikesizeid_seq.nextval,'Standard Male');
INSERT INTO bikesize (bikesizeid,bikesize)
VALUES (bikesizeid_seq.nextval,'Small Male');
INSERT INTO bikesize (bikesizeid,bikesize)
VALUES (bikesizeid_seq.nextval,'Standard Female');
INSERT INTO bikesize (bikesizeid,bikesize)
VALUES (bikesizeid_seq.nextval,'Child');

INSERT INTO dealer (dlrid,dlrname,dlraddress,dlrpostcode,dlrtelno)
VALUES (dlrid_seq.nextval,'Simons Scrap Ltd','83 Kingsway North HOLMSGARTH','ZE1
7EB',447001022952);
INSERT INTO dealer (dlrid,dlrname,dlraddress,dlrpostcode,dlrtelno)
VALUES (dlrid_seq.nextval,'Benjamin Brennan','55 Canterbury Road VIEWPARK','G71
0JN',447983057034);
INSERT INTO dealer (dlrid,dlrname,dlraddress,dlrpostcode,dlrtelno)
VALUES (dlrid_seq.nextval,'Jude Edwards','94 Vicar Lane SAUNDAIG','PA77
2JN',447800932079);
INSERT INTO dealer (dlrid,dlrname,dlraddress,dlrpostcode,dlrtelno)
VALUES (dlrid_seq.nextval,'Robert Allen','52 Felix Lane SHORNE','DA12
6TU',447967368497);
INSERT INTO dealer (dlrid,dlrname,dlraddress,dlrpostcode,dlrtelno)
VALUES (dlrid_seq.nextval,'Millie Ali','98 Walden Road GREATHAM','GU33
5ND',447079278709);
INSERT INTO dealer (dlrid,dlrname,dlraddress,dlrpostcode,dlrtelno)
VALUES (dlrid_seq.nextval,'Jordan Newman','88 Bridge Street GORS','SY23
9NT',447711007430);
INSERT INTO dealer (dlrid,dlrname,dlraddress,dlrpostcode,dlrtelno)
VALUES (dlrid_seq.nextval,'Stimtone','26 Highland Park, Feltham, Greater
London','TW13 4QW',447930107694);
INSERT INTO dealer (dlrid,dlrname,dlraddress,dlrpostcode,dlrtelno)
VALUES (dlrid_seq.nextval,'Charlotte Joyce','84 Far Lane, Hepworth, Holmfirth, West
Yorkshire ','HD9 1TL',447930187694);
INSERT INTO dealer (dlrid,dlrname,dlraddress,dlrpostcode,dlrtelno)
VALUES (dlrid_seq.nextval,'UnaHow','80A Albert Drive, Deganwy, Conwy','LL31
9RH',447744869327);
INSERT INTO dealer (dlrid,dlrname,dlraddress,dlrpostcode,dlrtelno)
```

```
VALUES (dlrid_seq.nextval,'Zendexon Ltd','1A Brandon Road, Leeds, West Yorkshire','LS3
1AH',447815893773);

INSERT INTO manufacturer
(manufid,manufname,manufaddress,manufpostcode,manuftelno)
VALUES (manufid_seq.nextval,'Rowley Bikes','19 Mawfa Avenue, Sheffield, South
Yorkshire','S14 1AJ',441574837475);
INSERT INTO manufacturer
(manufid,manufname,manufaddress,manufpostcode,manuftelno)
VALUES (manufid_seq.nextval,'Lees Cycles','66 Wansbeck Road, Jarrow, Tyne and
Wear','NE32 5SS',447827373694);
INSERT INTO manufacturer
(manufid,manufname,manufaddress,manufpostcode,manuftelno)
VALUES (manufid_seq.nextval,'Mohammed Bell Manufacturing plc','26 Holland Road,
Marlow, Buckinghamshire','SL7 1UJ',441485869584);
INSERT INTO manufacturer
(manufid,manufname,manufaddress,manufpostcode,manuftelno)
VALUES (manufid_seq.nextval,'Franklin Bicycles Productions Ltd','2 Boonwood Agric
Cottages, Gosforth, Seascale, Cumbria','CA20 1EA',441586473744);

INSERT INTO part (partnumber, partdescription, partpurchasedate, partserialno,
partmanuf, partprice)
VALUES (partnumber_seq.nextval, '3 Metre Superbike Chain', '20/OCT/12', 654235,
'SuperBike', 35);
INSERT INTO part (partnumber, partdescription, partpurchasedate, partserialno,
partmanuf, partprice)
VALUES (partnumber_seq.nextval, '18" Tyre', '10/JAN/14', 4857485094, 'Raleigh', 35.20);
INSERT INTO part (partnumber, partdescription, partpurchasedate, partserialno,
partmanuf, partprice)
VALUES (partnumber_seq.nextval, 'Handlebars, complete', '11/NOV/13', 4885432, 'Lees
Cycles', 55.50);
INSERT INTO part (partnumber, partdescription, partpurchasedate, partserialno,
partmanuf, partprice)
VALUES (partnumber_seq.nextval, 'Medium Saddle', '11/NOV/13', 68366, 'Rowley Bikes',
80);
INSERT INTO part (partnumber, partdescription, partpurchasedate, partserialno,
partmanuf, partprice)
VALUES (partnumber_seq.nextval, 'Brake wires 2 piece set', '16/OCT/13', 6373534,
'Currys', 18.80);

INSERT INTO customer (custid,custname,custaddress,custpostcode,custtelno)
VALUES (custid_seq.nextval,'Liam Finch','13 Bassano Street, London','SE22
8RY',447988660192);
INSERT INTO customer (custid,custname,custaddress,custpostcode,custtelno)
VALUES (custid_seq.nextval,'Millie Giles','2 Kings Square, York ','YO1
8BH',441484859678);
INSERT INTO customer (custid,custname,custaddress,custpostcode,custtelno)
VALUES (custid_seq.nextval,'Skye Wallace','111A Perrymans Farm Road, Ilford, Greater
London','IG2 7LU',442085748575);
INSERT INTO customer (custid,custname,custaddress,custpostcode,custtelno)
VALUES (custid_seq.nextval,'Jake Miah','1A Shaw Street, Rochdale, Greater
Manchester','OL12 9SN',441615847364);
```

```
INSERT INTO customer (custid,custname,custaddress,custpostcode,custtelno)
VALUES (custid_seq.nextval,'Lara Howarth','25 Old Park Hill, Dover, Kent ','CT16
2AW',441584958675);
INSERT INTO customer (custid,custname,custaddress,custpostcode,custtelno)
VALUES (custid_seq.nextval,'Riley Pratt','24 Woodview Road, Norman Hill, Dursley,
Gloucestershire ','GL11 5RW',441948576858);
INSERT INTO customer (custid,custname,custaddress,custpostcode,custtelno)
VALUES (custid_seq.nextval,'Hayden Gibson','30-32 Gribble Road, Liverpool,
Merseyside','L10 7NF',441475847395);
INSERT INTO customer (custid,custname,custaddress,custpostcode,custtelno)
VALUES (custid_seq.nextval,'Rosie Herbert','1 Reservoir Cottages, Catcleugh, Newcastle
upon Tyne, Northumberland ','NE19 1TX',441574837475);
INSERT INTO customer (custid,custname,custaddress,custpostcode,custtelno)
VALUES (custid_seq.nextval,'George Norris','London Road, South Downs National Park,
Liss, Hampshire','GU33 7QJ',441847596876);

INSERT INTO bike (bikeno, manufid, bikeclassid, bikesizeid, bikemodel, bikerentalprice,
bikepurchasedate, bikepurchaseprice)
VALUES (bikeno_seq.nextval, 3, 2, 4, 'SuperRider', 12, '23/MAR/12',265);
INSERT INTO bike (bikeno, manufid, bikeclassid, bikesizeid, bikemodel, bikerentalprice,
bikepurchasedate, bikepurchaseprice)
VALUES (bikeno_seq.nextval, 3, 2, 3, 'Roadster',15, '18/JUN/13', 250);
INSERT INTO bike (bikeno, manufid, bikeclassid, bikesizeid, bikemodel, bikerentalprice,
bikepurchasedate, bikepurchaseprice)
VALUES (bikeno_seq.nextval, 4, 3, 1, 'Partner Rider',10.50, '29/SEP/13', 120);
INSERT INTO bike (bikeno, manufid, bikeclassid, bikesizeid, bikemodel, bikerentalprice,
bikepurchasedate, bikepurchaseprice)
VALUES (bikeno_seq.nextval, 2, 1, 4, 'MountRider',13.20, '14/AUG/12', 162.48);
INSERT INTO bike (bikeno, manufid, bikeclassid, bikesizeid, bikemodel, bikerentalprice,
bikepurchasedate, bikepurchaseprice)
VALUES (bikeno_seq.nextval, 1, 2, 2, 'RoadRanger',9.80, '20/MAR/13', 111.25);
INSERT INTO bike (bikeno, manufid, bikeclassid, bikesizeid, bikemodel, bikerentalprice,
bikepurchasedate, bikepurchaseprice)
VALUES (bikeno_seq.nextval, 2, 2, 5, 'KidRider',7.89, '02/JAN/13', 85.25);
INSERT INTO bike (bikeno, manufid, bikeclassid, bikesizeid, bikemodel, bikerentalprice,
bikepurchasedate, bikepurchaseprice)
VALUES (bikeno_seq.nextval, 1, 1, 4, 'LadyPower', 14, '06/JUN/13', 145.67);

INSERT INTO partOrder (orderid, manufid, orderdate, totalcost)
VALUES (orderid_seq.nextval, 2, '22/FEB/14', 70);
INSERT INTO partOrder (orderid, manufid, orderdate, totalcost)
VALUES (orderid_seq.nextval, 3, '11/NOV/13', 135.5);
INSERT INTO partOrder (orderid, manufid, orderdate, totalcost)
VALUES (orderid_seq.nextval, 4, '16/OCT/13', 18.8);
INSERT INTO partOrder (orderid, manufid, orderdate, totalcost)
VALUES (orderid_seq.nextval, 1, '10/JAN/14', 70.40);

INSERT INTO orderLine (orderid, partnumber, quantity, quantitybasedtotal)
VALUES (1, 1, 2, 70);
INSERT INTO orderLine (orderid, partnumber, quantity, quantitybasedtotal)
VALUES (2, 3, 1, 55.5);
INSERT INTO orderLine (orderid, partnumber, quantity, quantitybasedtotal)
```

```
VALUES (2, 4, 1, 80);
INSERT INTO orderLine (orderid, partnumber, quantity, quantitybasedtotal)
VALUES (3, 5, 1, 18.8);
INSERT INTO orderLine (orderid, partnumber, quantity, quantitybasedtotal)
VALUES (4, 2, 2, 70.40);

INSERT INTO employee
(emplid,emplname,emplrole,emplusername,emplpassword,managerid)
VALUES (emplid_seq.nextval,'Raymond Smith','Owner','RaySmith','admin',NULL);
INSERT INTO employee
(emplid,emplname,emplrole,emplusername,emplpassword,managerid)
VALUES (emplid_seq.nextval,'Lisa McDonald','Hire Manager','LisaMcD','lisa', 1);
INSERT INTO employee
(emplid,emplname,emplrole,emplusername,emplpassword,managerid)
VALUES (emplid_seq.nextval,'Precious Igbinosun','Customer
Services','PreciousIgbinosun','precious',2);
INSERT INTO employee
(emplid,emplname,emplrole,emplusername,emplpassword,managerid)
VALUES (emplid_seq.nextval,'Sanjit Khan','Parts Manager','SanKhan','sanjit',1);
INSERT INTO employee
(emplid,emplname,emplrole,emplusername,emplpassword,managerid)
VALUES (emplid_seq.nextval,'Simon Read','Maintenance','SimonRead','simon', 4);

INSERT INTO reservation
(reservno,bikeno,custid,emplid,reservrentdate,reservtimeout,reservtimedueback,reserv
datedueback,reservactualtimeback,reservactualdateback,reservamountpaid)
VALUES (reservno_seq.nextval,1,7,1,'13/APR/12','09:50',
'17:00','13/APR/12','17:00','13/APR/12',12);
INSERT INTO reservation
(reservno,bikeno,custid,emplid,reservrentdate,reservtimeout,reservtimedueback,reserv
datedueback,reservactualtimeback,reservactualdateback,reservamountpaid)
VALUES (reservno_seq.nextval,2,3,2,'18/OCT/13','07:30',
'15;00','18/OCT/13','14:20','18/OCT/13',15);
INSERT INTO reservation
(reservno,bikeno,custid,emplid,reservrentdate,reservtimeout,reservtimedueback,reserv
datedueback,reservactualtimeback,reservactualdateback,reservamountpaid)
VALUES (reservno_seq.nextval,4,1,3,'03/NOV/13','11:00',
'21:00','03/NOV/13','21:05','03/NOV/13',13.30);
INSERT INTO reservation
(reservno,bikeno,custid,emplid,reservrentdate,reservtimeout,reservtimedueback,reserv
datedueback,reservactualtimeback,reservactualdateback,reservamountpaid)
VALUES (reservno_seq.nextval,3,4,2,'21/NOV/13','15;30',
'15:30','22/NOV/13','15:30','22/NOV/13',21);
INSERT INTO reservation
(reservno,bikeno,custid,emplid,reservrentdate,reservtimeout,reservtimedueback,reserv
datedueback,reservactualtimeback,reservactualdateback,reservamountpaid)
VALUES (reservno_seq.nextval,5,2,1,'11/DEC/13','10:30',
'16;00','12/DEC/13','15:30','12/DEC/13',19.60);
INSERT INTO reservation
(reservno,bikeno,custid,emplid,reservrentdate,reservtimeout,reservtimedueback,reserv
datedueback,reservactualtimeback,reservactualdateback,reservamountpaid)
```

```
VALUES (reservno_seq.nextval,1,6,2,'22/JAN/14','12:20',
'18:20','23/DEC/13','18:30','23/JAN/14',18);
INSERT INTO reservation
(reservno,bikeno,custid,emplid,reservrentdate,reservtimeout,reservtimedueback,reserv
datedueback,reservactualtimeback,reservactualdateback,reservamountpaid)
VALUES (reservno_seq.nextval,6,9,3,'03/JAN/14','13:30',
'18;00','04/JAN/14','17:25','04/JAN/14',15.78);
INSERT INTO reservation
(reservno,bikeno,custid,emplid,reservrentdate,reservtimeout,reservtimedueback,reserv
datedueback,reservactualtimeback,reservactualdateback,reservamountpaid)
VALUES (reservno_seq.nextval,3,7,1,'06/FEB/14','09:45',
'19:45','06/FEB/14','19:45','06/FEB/14',10.50);
INSERT INTO reservation
(reservno,bikeno,custid,emplid,reservrentdate,reservtimeout,reservtimedueback,reserv
datedueback,reservactualtimeback,reservactualdateback,reservamountpaid)
VALUES (reservno_seq.nextval,7,5,3,'12/FEB/14','09:00',
'19:00','12/FEB/14','19:00','12/FEB/14',14);
INSERT INTO reservation
(reservno,bikeno,custid,emplid,reservrentdate,reservtimeout,reservtimedueback,reserv
datedueback,reservactualtimeback,reservactualdateback,reservamountpaid)
VALUES (reservno_seq.nextval,6,7,1,'18/FEB/14','13:30',
'18;00','18/FEB/14','18:00','18/FEB/14',15.78);

INSERT INTO disposal (disposalid, bikeno, dlrid, disposalsaledate, disposalsaleprice)
VALUES (disposalid_seq.nextval, 1, 1, '11/JAN/14', 120);
INSERT INTO disposal (disposalid, bikeno, dlrid, disposalsaledate, disposalsaleprice)
VALUES (disposalid_seq.nextval, 4, 3, '21/FEB/14', 86.50);
INSERT INTO disposal (disposalid, bikeno, dlrid, disposalsaledate, disposalsaleprice)
VALUES (disposalid_seq.nextval, 6, 7, '18/DEC/13', 45.20);

INSERT INTO maintenance (maintref, bikeno, partnumber, maintpriority,
maintpartrequired, maintfaultdetails, maintfaultdate, maintactiondate,
maintactiontaken)
VALUES (maintref_seq.nextval, 1, 1, 'High', 'Bike Chain', 'Chain has snapped,
replacement needed', '19/OCT/12', '22/OCT/12', 'Chain replaced');
INSERT INTO maintenance (maintref, bikeno, partnumber, maintpriority,
maintpartrequired, maintfaultdetails, maintfaultdate, maintactiondate,
maintactiontaken)
VALUES (maintref_seq.nextval, 4, NULL, 'Low', NULL, 'Full Servicing', '14/AUG/13',
'21/AUG/13', 'Servicing carried out, No part replacement');
INSERT INTO maintenance (maintref, bikeno, partnumber, maintpriority,
maintpartrequired, maintfaultdetails, maintfaultdate, maintactiondate,
maintactiontaken)
VALUES (maintref_seq.nextval, 2, 5, 'Normal', 'Break Wires', 'Wires have worn out,
replacement needed', '16/OCT/13', '19/OCT/13', 'Wires replaced');
INSERT INTO maintenance (maintref, bikeno, partnumber, maintpriority,
maintpartrequired, maintfaultdetails, maintfaultdate, maintactiondate,
maintactiontaken)
VALUES (maintref_seq.nextval, 6, 4, 'Low', 'Bike Saddle', 'Saddle worn, needs
replacement', '11/NOV/13', NULL, NULL);
```

```
INSERT INTO maintenance (maintref, bikeno, partnumber, maintpriority,
maintpartrequired, maintfaultdetails, maintfaultdate, maintactiondate,
maintactiontaken)
VALUES (maintref_seq.nextval, 3, 2, 'High', 'Bike Tyre', 'Bike Tyres need replacing x2',
'09/JAN/14', '10/JAN/14', 'Tyres replaced');
INSERT INTO maintenance (maintref, bikeno, partnumber, maintpriority,
maintpartrequired, maintfaultdetails, maintfaultdate, maintactiondate,
maintactiontaken)
VALUES (maintref_seq.nextval, 5, NULL, 'Low', NULL, 'Full Servicing', '27/FEB/14', NULL,
NULL);
```

20.1.4 Sequence Statements

These are the sequence statements I have used to create sequences to automatically increment various primary keys across my database tables.

```
CREATE SEQUENCE dlrID_seq
START WITH 000001
INCREMENT BY 1
MINVALUE 000001
MAXVALUE 999999;

CREATE SEQUENCE bikeClassID_seq
START WITH 000001
INCREMENT BY 1
MINVALUE 000001
MAXVALUE 999999;

CREATE SEQUENCE bikeSizeID_seq
START WITH 000001
INCREMENT BY 1
MINVALUE 000001
MAXVALUE 999999;

CREATE SEQUENCE manufID_seq
START WITH 000001
INCREMENT BY 1
MINVALUE 000001
MAXVALUE 999999;

CREATE SEQUENCE partNumber_seq
START WITH 000001
INCREMENT BY 1
MINVALUE 000001
MAXVALUE 999999;

CREATE SEQUENCE custID_seq
START WITH 000001
INCREMENT BY 1
MINVALUE 000001
MAXVALUE 999999;

CREATE SEQUENCE bikeNo_seq
START WITH 000001
```

```
INCREMENT BY 1
MINVALUE 000001
MAXVALUE 999999;

CREATE SEQUENCE orderID_seq
START WITH 000001
INCREMENT BY 1
MINVALUE 000001
MAXVALUE 999999;

CREATE SEQUENCE emplID_seq
START WITH 000001
INCREMENT BY 1
MINVALUE 000001
MAXVALUE 999999;

CREATE SEQUENCE reservNo_seq
START WITH 000001
INCREMENT BY 1
MINVALUE 000001
MAXVALUE 999999;

CREATE SEQUENCE disposalID_seq
START WITH 000001
INCREMENT BY 1
MINVALUE 000001
MAXVALUE 999999;

CREATE SEQUENCE maintRef_seq
START WITH 000001
INCREMENT BY 1
MINVALUE 000001
MAXVALUE 999999;
```

20.1.5 Describe Statements

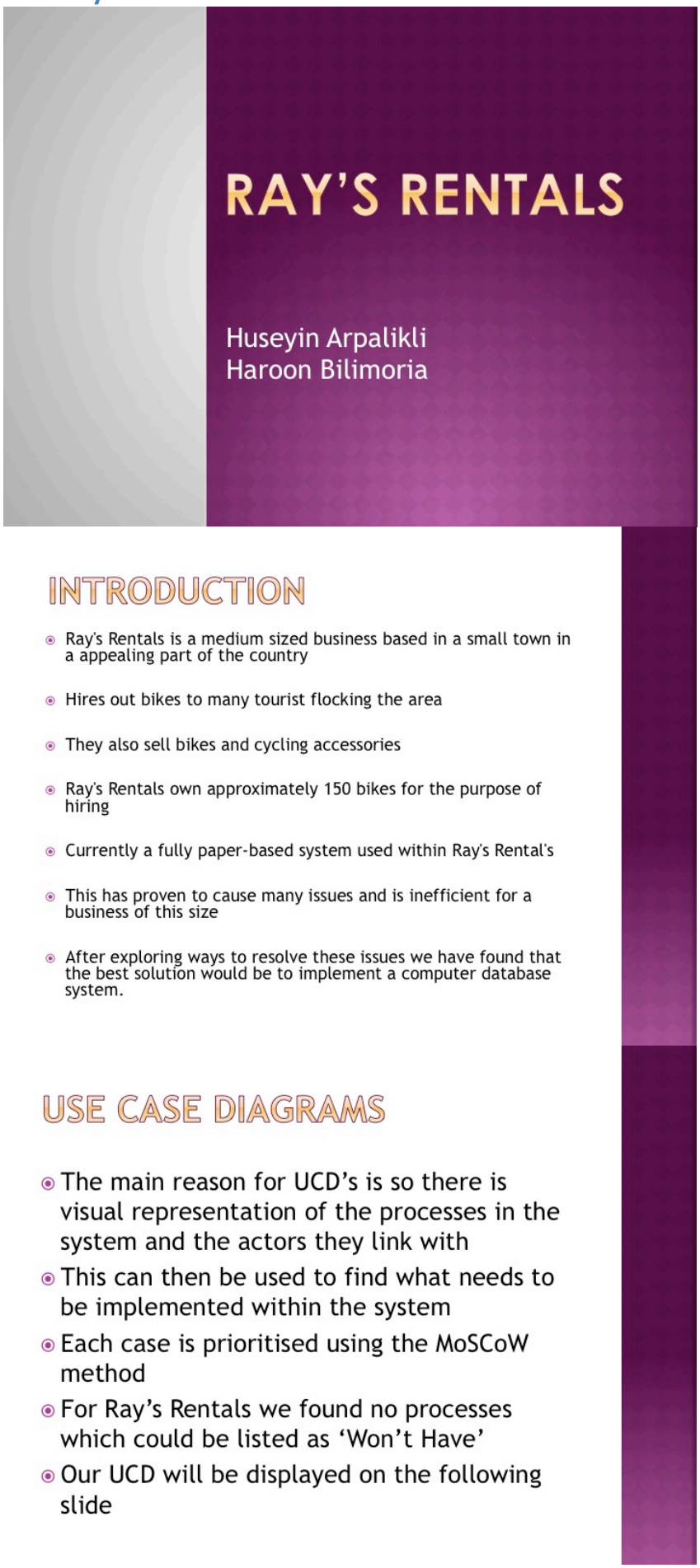
```
DESCRIBE bike; DESCRIBE disposal; DESCRIBE dealer; DESCRIBE bikeClass; DESCRIBE bikeSize;
DESCRIBE manufacturer; DESCRIBE maintenance; DESCRIBE part; DESCRIBE orderLine;
DESCRIBE partOrder; DESCRIBE reservation; DESCRIBE customer; DESCRIBE employee;
```

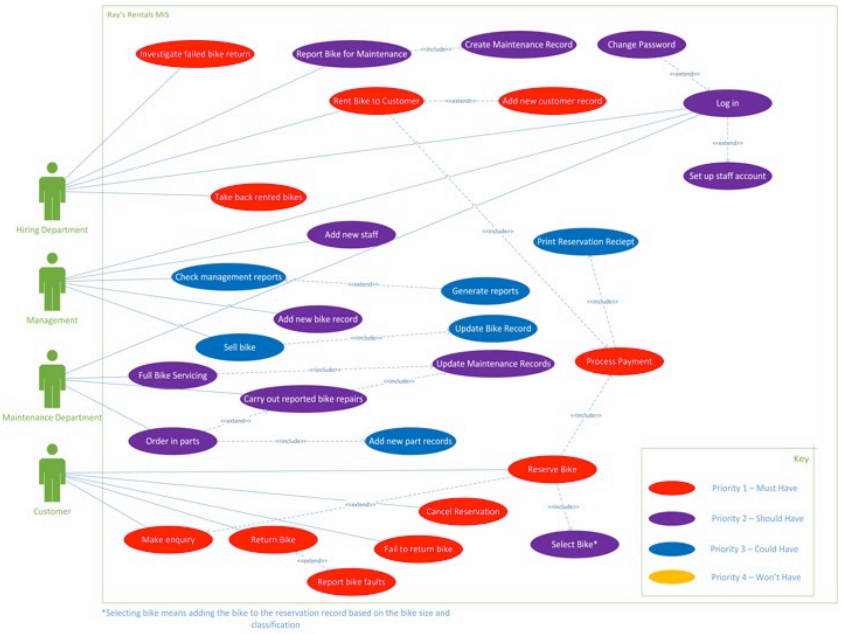
```
DESCRIBE bike
Name              Null     Type
----------------- -------- ------------
BIKENO            NOT NULL NUMBER(6)
MANUFID           NOT NULL NUMBER(6)
BIKECLASSID       NOT NULL NUMBER(6)
BIKESIZEID        NOT NULL NUMBER(6)
BIKEMODEL         NOT NULL VARCHAR2(40)
BIKERENTALPRICE   NOT NULL NUMBER(9,2)
BIKEPURCHASEDATE  NOT NULL DATE
BIKEPURCHASEPRICE NOT NULL NUMBER(9,2)

DESCRIBE disposal
Name              Null     Type
----------------- -------- -----------
DISPOSALID        NOT NULL NUMBER(6)
BIKENO            NOT NULL NUMBER(6)
DLRID                      NUMBER(6)
DISPOSALSALEDATE  NOT NULL DATE
DISPOSALSALEPRICE NOT NULL NUMBER(9,2)

DESCRIBE dealer
Name        Null     Type
----------- -------- ------------
DLRID       NOT NULL NUMBER(6)
DLRNAME     NOT NULL VARCHAR2(40)
DLRADDRESS  NOT NULL VARCHAR2(60)
DLRPOSTCODE NOT NULL VARCHAR2(8)
DLRTELNO    NOT NULL NUMBER(12)
```

```
DESCRIBE bikeClass
Name        Null     Type
----------- -------- ------------
BIKECLASSID NOT NULL NUMBER(6)
BIKECLASS   NOT NULL VARCHAR2(12)

DESCRIBE bikeSize
Name       Null     Type
---------- -------- ------------
BIKESIZEID NOT NULL NUMBER(6)
BIKESIZE   NOT NULL VARCHAR2(12)

DESCRIBE manufacturer
Name          Null     Type
------------- -------- ------------
MANUFID       NOT NULL NUMBER(6)
MANUFNAME     NOT NULL VARCHAR2(40)
MANUFADDRESS  NOT NULL VARCHAR2(60)
MANUFPOSTCODE NOT NULL VARCHAR2(8)
MANUFTELNO    NOT NULL NUMBER(12)

DESCRIBE maintenance
Name               Null     Type
------------------ -------- -------------
MAINTREF           NOT NULL NUMBER(6)
BIKENO             NOT NULL NUMBER(6)
PARTNUMBER                  NUMBER(6)
MAINTPRIORITY      NOT NULL VARCHAR2(10)
MAINTPARTREQUIRED           VARCHAR2(100)
MAINTFAULTDETAILS NOT NULL VARCHAR2(100)
MAINTFAULTDATE     NOT NULL DATE
MAINTACTIONDATE             DATE
MAINTACTIONTAKEN            VARCHAR2(20)
```

```
DESCRIBE part
Name             Null     Type
---------------- -------- -------------
PARTNUMBER       NOT NULL NUMBER(6)
PARTDESCRIPTION  NOT NULL VARCHAR2(100)
PARTPURCHASEDATE NOT NULL DATE
PARTSERIALNO              NUMBER(15)
PARTMANUF        NOT NULL VARCHAR2(36)
PARTPRICE        NOT NULL NUMBER(9,2)

DESCRIBE orderLine
Name               Null     Type
------------------ -------- -----------
ORDERID            NOT NULL NUMBER(6)
PARTNUMBER         NOT NULL NUMBER(6)
QUANTITY           NOT NULL NUMBER(2)
QUANTITYBASEDTOTAL NOT NULL NUMBER(9,2)

DESCRIBE partOrder
Name      Null     Type
--------- -------- -----------
ORDERID   NOT NULL NUMBER(6)
MANUFID   NOT NULL NUMBER(6)
ORDERDATE NOT NULL DATE
TOTALCOST NOT NULL NUMBER(9,2)

DESCRIBE reservation
Name                  Null     Type
--------------------- -------- -----------
RESERVNO              NOT NULL NUMBER(6)
BIKENO                NOT NULL NUMBER(6)
CUSTID                NOT NULL NUMBER(6)
EMPLID                NOT NULL NUMBER(6)
RESERVRENTDATE        NOT NULL DATE
RESERVTIMEOUT         NOT NULL VARCHAR2(5)
RESERVACTUALTIMEBACK           VARCHAR2(5)
RESERVACTUALDATEBACK           DATE
RESERVTIMEDUEBACK     NOT NULL VARCHAR2(5)
RESERVDATEDUEBACK     NOT NULL VARCHAR2(5)
RESERVAMOUNTPAID               NUMBER(9,2)

DESCRIBE customer
Name         Null     Type
------------ -------- ------------
CUSTID       NOT NULL NUMBER(6)
CUSTNAME     NOT NULL VARCHAR2(60)
CUSTADDRESS  NOT NULL VARCHAR2(70)
CUSTPOSTCODE NOT NULL VARCHAR2(8)
CUSTTELNO    NOT NULL NUMBER(12)

DESCRIBE employee
Name         Null     Type
------------ -------- ------------
EMPLID       NOT NULL NUMBER(6)
EMPLNAME     NOT NULL VARCHAR2(40)
EMPLROLE     NOT NULL VARCHAR2(30)
EMPLUSERAME  NOT NULL VARCHAR2(20)
EMPLPASSWORD NOT NULL VARCHAR2(20)
```

20.2 Ray's Rentals Presentation





USE CASE SPECIFICATION

- Normally each use case process is explained using a use case specification but we will just just be showing an example of one process
- The purpose is to describe how the the system will be used by the actors
- The ERD attached with the use case specification shows which data within the database will be needed to carry out the process.

Use Case: Carry Out Reported Bike Repairs including Update Maintenance Records
Actors: Maintenance Department **Version:** 1.0
Pre-Conditions
Use Case: Create Maintenance Record has been carried out by the HIRING DEPARTMENT as part of repair referral for applicable bikes Bike being repaired is has a record present in the system
Post-Conditions
Successful Completion: Bike is successfully repaired and maintenance records are updated accordingly Failure Condition: Bike is not repaired and is recorded in the SYSTEM
Primary Path
1. The use case begins when MAINTENANCE DEPARTMENT accesses the maintenance records for new repair requests. 2. MAINTENANCE DEPARTMENT inspects the list sorted by popularity and selects the new maintenance record for the most popular bike that has recently been referred. 3. SYSTEM displays the bike details and maintenance details on screen. 4. The MAINTENANCE DEPARTMENT then retrieves the bike from storage. 5. MAINTENANCE DEPARTMENT checks that the reported damages are physically present and that repair is needed and/or possible. 6. MAINTENANCE DEPARTMENT checks parts required from suppliers are available. 7. Use Case: *Order in parts* and Use Case: *Add new part records* are performed. 8. Following receiving parts ordered the MAINTENANCE DEPARTMENT would carry out the repair. 9. When repair is complete the maintenance details and the part number are added to the maintenance record in the SYSTEM by the MAINTENANCE DEPARTMENT. 10. The bike is put back into storage by the MAINTENANCE DEPARTMENT for rental. 9. The use case ends successfully.

Alternate Path

Repair not needed after inspection

If after step 5 in the primary path, repair is not needed, then

1. The MAINTENANCE DEPARTMENT will open the appropriate maintenance record in the SYSTEM.
2. MAINTENANCE DEPARTMENT will enter a comment into action taken stating that the bike does not repair and a reason and then an action date will be recorded.
3. The use case ends with a failure condition.

Bike needing repair is out on rental and needs returning as soon as possible

If at step 3 in the primary path, the bike selected for repair is out on rental and has a serious fault, then

1. MAINTENANCE DEPARTMENT will check fault details in the SYSTEM.
2. MAINTENANCE DEPARTMENT will search for and inspect reservation record in the SYSTEM to find out whom the bike is rented out to.
3. MAINTENANCE DEPARTMENT will call the CUSTOMER and notify them that they need to return the bike as soon as possible due to serious faults and that they will be offered a replacement rental.
4. CUSTOMER returns bike, Use Case: *Return Bike*; HIRING DEPARTMENT records return of bike, Use Case: *Take back rented bikes.*
5. CUSTOMER Use Case: *Reserve Bike*; including *Select Bike*; HIRING STAFF Use Case: Rent bike to customer.
6. The use case resumes at step 4.

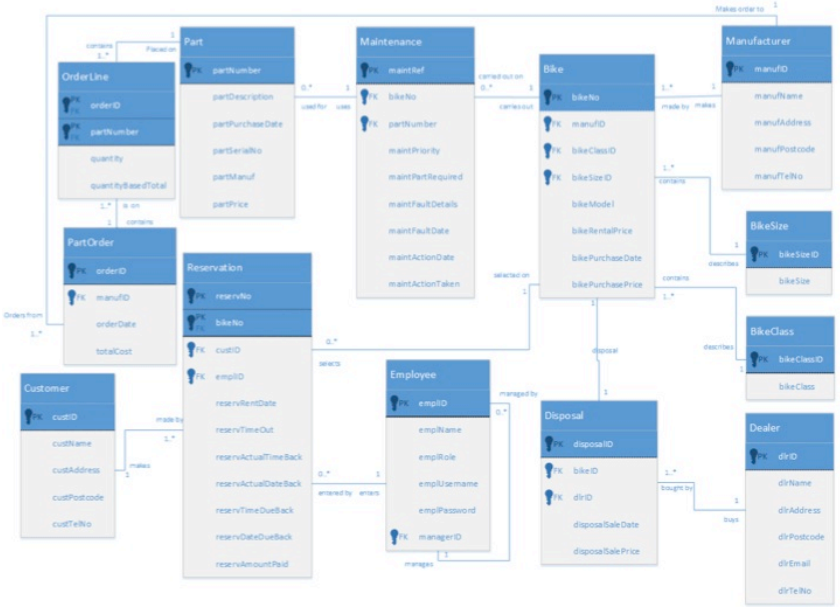
Bike needing repair is out on rental and will be repaired upon return

If in step 3 in the primary path, the bike selected for repair is out on rental and has a minor fault, then

1. MAINTENANCE DEPARTMENT will not enter anything into the SYSTEM and will cancel repair.
2. When time has come for customer to return rental CUSTOMER returns bike, Use Case: *Return Bike*; HIRING DEPARTMENT records return of bike, Use Case: *Take back rented bikes.*
3. The use case resumes at step 4.

ENTITY RELATIONSHIP DIAGRAM

- Following the normalisation of the data provided we come up with a final ERD used for the implementation of the database
- The ERD displays how all data is linked together within the database
- Following slide will show our group ERD…



ORACLE SQL REPORTS DEMO…

CONCLUSION

- The best way to resolve issues with the paper based system is with the new computersied database system
- Using the UCD, ERD and Data Dictionary we have been able to implement a successful database system
- Have also been able to create examples of some necessary reports relevant to Ray's Rentals using SQL