

Ace Training
Training Databases

1/17/2016

Daniel McHugh

Contents

Chapter one.....	3
Introduction	3
History of Databases	3
Flat File Databases.....	4
Hierarchical Databases.....	6
Network Databases	9
Relational Databases.....	12
Object Oriented Databases	14
Scenario.....	17
Requirements.....	17
Business Rules	25
Business Operation	27
Chapter two.....	31
Conceptual Design.....	31
Conceptual Diagram	31
Entity Relationships.....	33
Diagram script	36
Logical Design.....	39
Logical Design.....	39
Normalization.....	41
Cumulative Design.....	41
Normalized Tables.....	42
Chapter three	58
Implementation.....	58
Field Attributes.....	58
Chapter four	64
SQL Implementation	64
Database.....	64
Tables	65

Users.....	93
Chapter six.....	97
Security and Contingency.....	97
Security.....	97
Contingency.....	98
Chapter seven	104
Testing.....	104
Graphical User Interface Design.....	120
Data Capture	123
Conclusion	134
References.....	134
Appendix	135

Chapter one

Introduction

In this report I will talking about the database that I have created each section will explain what is it about and what effect it has on the database the aim of this report is to inform the reader about databases and help them learn about databases in a friendly manner also I will be trying to inform you about what a database is and how businesses use them to store massive amounts of data.

History of Databases

In this section of the report the history of databases will be explored in this you will find information detailing other databases that have been used in the past each of these database was

very popular at their times of release also each have different methods on how they collect data using these past databases has allowed new and better databases to come up.

Flat File Databases

Key points

Flat file databases is the first generation in databases it was first implemented in the USA by a man called Herman Hollerith he conceived the idea that census data could be represented by holes punched in paper cards he sold this idea to the US census bureau and in 1890 the USE was the first computerized databases consisting in essence of thousands of boxes full of punched cards. As a way to store work times this would allow administrators to check if the workers were getting in on time and if they were working the right amount of hours.

They collect this data using punched card that would be placed in a time box when the card was placed inside it would print on the card what time it was placed in then at the end of the day the workers would give it to the administrators they would enter this data into their databases which would form the flat file databases as this was first generation the data was very simple the cards had around eighty columns every time a new week came they would have to update these databases to keep them as current as possible.

Details of research

In 1968 most databases used flat file model this was one of the only ways to store data in a database with structure before flat file was fully created they used punched card technology which stored data this was the framework to fully completing the flat file model. Later on businesses would use files to store this data this was the only difference between punch card technology and flat files model but having this didn't have any really advantages rather this technology had more than several limitations which would make it harder to use.

The whole database was designed around the idea of a single table or a list with fields to represent the parameters needed many flat files databases contained many fields more often with duplicate data which would corrupt the whole data if not fixed properly as it was simple technology it was hard to make changes to the database this would result in a higher risk that data would be lost. An example of flat file database is shown in figure A as you can see you would need at least this amount of records for one person also you would have to update this by hand every time you wanted to make a change.

Figure A shows an example of flat file database.

Field Name	Data Type
CUSTOMER_ID	AutoNumber
TITLE	Text
SURNAME	Text
STREET	Text
TOWN	Text
COUNTY	Text
PHONE_NUMBER	Text
PET'S_NAME	Text
TYPE	Text
DATE_OF_BIRTH	Date/Time
MEDICAL_CONDITIONS	Text
DATE_OF_APPOINTMENT	Date/Time
TIME_OF_APPOINTMENT	Date/Time
SYMPTOMS	Text
TREATMENT	Text

Advantages

- One of the only advantages was that it could be accessed through a number of methods some of these were sequential, indexed and random at the time this was quite good as it was the first time a user could access the data in different methods for example if you needed to find some you could have the data arranged in indexed this would allow you to find the person faster.

Disadvantages

- To manage this database you would have to know certain programming language this means that not anyone can use this you would need to train people or get people that have know about different programming languages this would push the cost of your staff up if you are getting people with certain skills also if you are training new staff it will cost you time and while this training is happening the new database can't be used.
- As each program is on its own computer it maintains its own set of data so only one computer would be able to hold the original database as this is the case if you wanted to update the database you would have to do it on the same computer because the data would only be on that computer.
- Duplication of data many business would duplicable the data that was on the database in case the databases corrupts this would mean that the same data was being held by different programs this created space that was being wasted because of the databases high risk factor also this creates a waste in resources with could be used for something else.

- Weak security was another disadvantage as it was a simple database it only could use simple security means one of these security means is a password protect were the business would place a password on the database to stop people from accessing their database. But if they found out about the password they would have unlimited control over the database so anyone could change any data that was being held on the database this could cause massive problems for the business as losing data would damage the business.

Requirements

When looking at this with my requirements it won't work as for one this database type is too old and it has many problems one of its main problems is security I would not trust this database to hold all my important data. Also if you wanted to share your work with another computer this won't be possible as it saves work on that one computer so if you made any changes to the database only that computer would be effected this won't be good if one person needed to update current data and couldn't find out where certain data was as someone else has moved it location.

Recommendations

For the amount of limitations this model has it would not be recommended to anyone as another system was still active back then which had a lower risk of error the choice would be better if you were trying to keep a hold on your data but in terms of security even if it was simple it would be a challenge for most people so in the end the punch card technology would be better as it was used in the same manner.

Hierarchical Databases

Key points

Hierarchical databases is the second generation in databases this structure was developed by the IBM (International Business Machines Corporation) in 1960s and 1980s this was IBM's first DBMS (Databases Management System) called IMS (Information Management System) it was designed like a tree to show the connect between each different part of the business a key was used to tell people what each shape means.

These databases were popular in early databases design as it was simple to show the connection between one too many relationships one would be the owner or parent and the linked one would be called the child.

Details of research

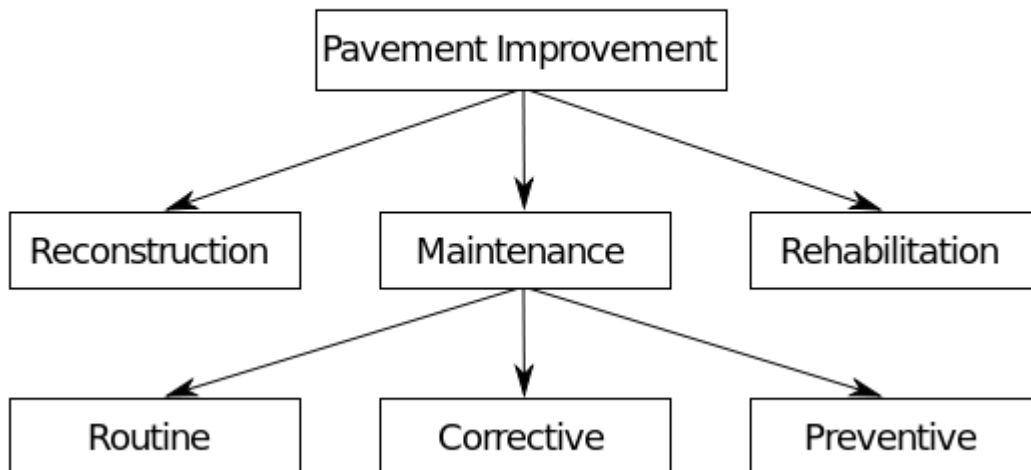
The idea behind the hierarchical database model is to show the connections between one too many relationship as the one would be the parent the many would be the child this model is most useful for storing certain types of data.

For example you could show that a company which would be the parent is linked to all of its departments this would be the child which then is linked to other smaller departments in the business as the business grows you can create massive trees which shows how each department is linked as the tree grows the roles will change so on the first and second tier the company will be the parent and the department will be the child but as we move to the second and third tier the department becomes the parent and the smaller departments become the child this continues until there are no departments left in the business also you can show the flow of data using this model as departments that are linked would share the same data.

Another use would be to build profiles on the people that are working in the business so you would collect data on the worker at your business but you might also ask for other data about their wife/husband and child this will build a tree and show how they are linked together.

Figure B shows an example of a Hierarchical model.

Hierarchical Model



Advantages

- It would be a lot easier to find people in your business as long as you know the department they work on you can search for that department and look at the list of people that work on that department this would give you clear information about that person as they would have a profile about them.
- Unlike the first generation databases this database is a lot more secure as technology as moved on the way we handle data has improved as well so not only would access be restricted from certain users other department wouldn't be allowed see other departments data as it would be not use to them this means the likelihood that data could be leaked is very low as only some user will be able to access data on the database. Other security measures would be enforced as well to prevent people from accessing certain data also this model is a lot more stable than the last model so the risk of data becoming damage is low.
- As each department would be handling different jobs in the business each department would be in control of their own data this means that the amount of control we have on data has increased massively when compared to the last model this means that the data in each department is more independent also with this model we are far less likely to get duplicate data.

Disadvantage

- The implementation of this database is very complex as it can vary depending on the size of your business if you have a big business you are going to have more departments that handle different tasks in the business this means that the tree will be more complex to design which means to create the database is more complex also this means the price to implement it in the business will go up.
- Again another disadvantage that ties with this is the difficulty to manage the database as the database grows the more data you need to store and handle properly this increase the amount of manpower you are going to need to keep it in running order.
- Another disadvantage would be that is can't handle many too many relationships too well if you want to create a tree model that has one too many relationships then it's great but it has problems with many too many the reason why is because child can't have more than one parent this is why many too many doesn't work well on this model.

Requirements

For some of the requirements it would work well for example student and staff it would work well as you could use this to find certain people that are involved in this category and see what they link to this would be a great use for this database. But for other cases this type of database won't be useful for example student progress has many different links to many different requirements in the database and this database can't handle complex designs so it would get really confusing for the client.

Recommendation

This model is great for certain data types if you want to create a simple tree model that shows the links in the business then it is great this would be one to many but if you want to create a complex tree model that's when you start to hit the limits of the model as it can't handle many too many to well it becomes more complex to show the links as you have to have more links than needed. But as a whole database it has a lot of advantages so this would be good if you were a small to medium size business.

Network Databases

Key point

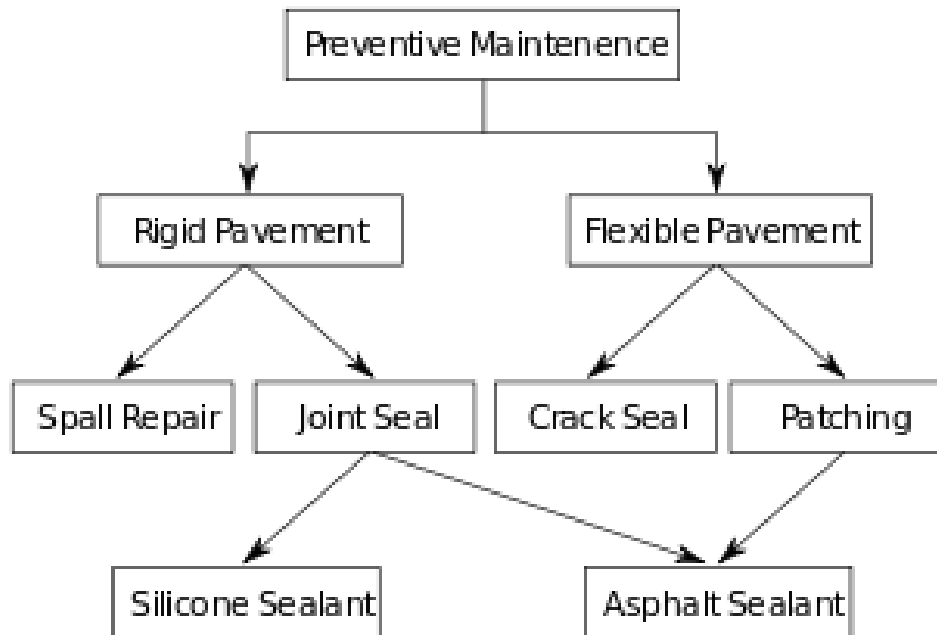
Network databases is the third generation in databases it was invented by Charles Bachman it was first developed in 1969 by the conference on Data System Languages (CODASYL) then it was released again in 1971 by this time the model had been standardized by the same group this became the basis for most implementations.

Details of research

After the results of the Hierarchical model which were great it was really popular for a long time so with the framework from that they created the Network model this had some of the features from the Hierarchical model but with more improvements such as one of the biggest problems with the Hierarchical model was the lack of many to many links with the network model they made it so the model can have many to many links so on this model one child can have many parents and one parent can have many children this allows for many to many links to be made this makes modeling the database far easier.

Figure C shows an example of a Network model.

Network Model



Advantages

- Many too many relationships this is the best advantage this model has it allows for more flexibility in the database this allows information to be accessed a lot faster as more departments have been linked together alongside this the flow of data is more stable as the model is better design.
- Another advantage is that you can find certain data faster so if you were looking for a person in the business all you would need to know is what department they work on and you can find out information about like what title they hold in the business, what task they have been assigned to, who they report to and other information. Also you could find personal information out as well so you could see if they have any children and how many, you could see if they have a wife, where they live.

Disadvantages

- As with the other models the size of your business plays a big part of how complex your database is so and as this model is a better version of the Hierarchical model it still has some of its problems one of them is how complex they become once you start to

implement them . As the business grows they create many different departments all these need to be linked together so they can share data as this starts to happen it become hard to see what links with what and how many different parts of the business link to one part for a person that is new to databases it can be overwhelming to see all these different links.

Requirements

This database is very similar to the last one as it is an improvement it will do the same things as the last one but better so I would say that this type of database is really good at what it can do. So this would be Student and Staff but it can do more complex designs allowing for better links to be created this will not only make the design look better but it will allow the client to understand how they link as well this would be good for student progress as it is linked to many different requirements which would be hard to understand if not for the improved relations.

Recommendations

This is a great model is has a lot of advantages to using it and as stated is follows the hierarchical structure and it has improved the relationships so you are less restricted on what you can link together it creates a smoother database allowing data to travel between links a lot faster. This is great if you are a business that is growing as you will have many different departments doing different tasks it allows everybody to share data to help the business understand if they are any problems within.

Relational Databases

Key points

Relational databases is the fourth generation in databases this database again follows close to the hierarchical structure as it shows data in the same way like with the network model. But differently at the same time unlike the other models the relational model uses grid-like mathematical structures which use columns and rows the person to create this idea was E.F Codd he proposed this idea to the IBM later on this model would become the basis of all relational databases most people have see this type of model and most people have used it at least once before as this model is called a table if you create is physical.

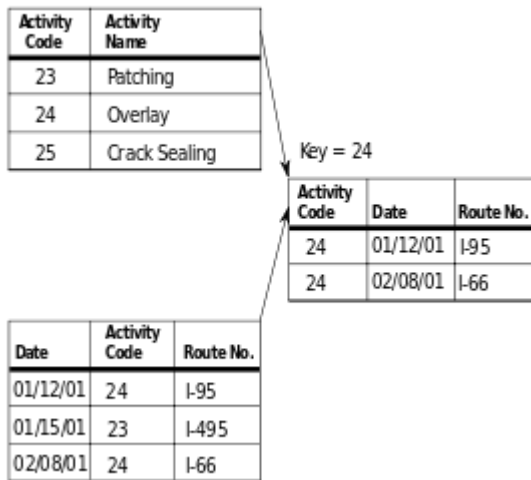
Details of research

When using this model all data must be stored in relations these are the tables that are created and each of the tables have rows and columns also each relation must have a header which is the list of columns in the table and they must have a body this is the data that populates the relation which will go in the rows as you place data in these rows you can create unique values using the columns you have set up doing this is called tuple.

Keys are used in this model as well to order data and to relate data to other table so if you wanted to show that this table relates to another table you would use one of these keys to show that they are related in some way this is very important to remember as these keys play a massive part in how this database works. As we are talking about keys there are two which are use one of them is the Primary key which is used to identify each row of data also it is used to make querying of data easier the second kind of key is the Foreign key this is used to relate data in one relation to the primary key of another relation so this used to link tables together using each tables primary key.

Figure D shows an example of a Network model.

Relational Model



Advantages

- Easy to use and understand as the data is presented in a table it gives clear information to the user about what the table is about as long as the user has to key they should be able to find the relative data they are looking for also as these tables use keys they are linked together so this gives the user more freedom which can help them find information faster.
- As this model uses keys it is a lot more secure than other models this allows you to put hid sensitive information better as to access this information you would have to one of these keys which only a few members of staff would be able to access this also allows to link sensitive information together so you are able to see that they both relate to one another.
- Another advantage is that this model uses SQL language which is good for both complex and simple tasks and SQL can be used to interact with the database so you can type in queries that will help you find the information you need.

Disadvantages

- One of the main problems with relational models is the machine performance if you have a large amount of tables which relationships to be established are larger this effects the performance in responding to the sql queries. So you need to make sure that you have the right hardware to support the database otherwise the whole system will start to run slow this could cause the system to crash.
- Another disadvantage would be a slow extraction of meaning from data if the data is sorted like the hierarchical model it could take longer to understand the meaning of certain data this would slow down makes it hard to see the link between the data in the table.

Requirements

This type of database is really good at security as it uses keys to link everything together so you can only access a certain area if it has been linked with one of these keys this makes access information a lot more secure. This would be one of the best types of databases to use for the requirements that have been outlined for the database as it uses certain features from the last database so it can make complex designs which were great plus now you have the added security that comes with this database so all the data that is stored in the database will be protected from an attackers.

Recommendations

This model is really good it is very secure as it uses keys you can't go accessing information that is not linked so to access certain information you would have to use certain keys which only certain people would have access to and it allows you to set up data in a better fashion this allows people that don't know about databases to understand the link between one table and another table. This would be used in bigger business as it is for business that have a lot of information to handle also because of the performance issue only businesses that have really good hardware would be able to keep the databases running smoothly.

Object Oriented Databases

Key points

Object oriented databases is the fifth generation in databases this database is like none of the other this uses objects which represents the data in the database this model is not as successful or well-known as the other mainstream databases but it still has it uses. It first appeared in 1985 this model was used in a lot of projects as it was not as popular as the other models it didn't really start growing until 1990s and 2004 this is when we see the most growth in object oriented databases this was because open source object databases emerged which was cheap and easy to use so a lot of people that were new to programming and databases would start by using this then as they got better they would learn how to write java or C#.

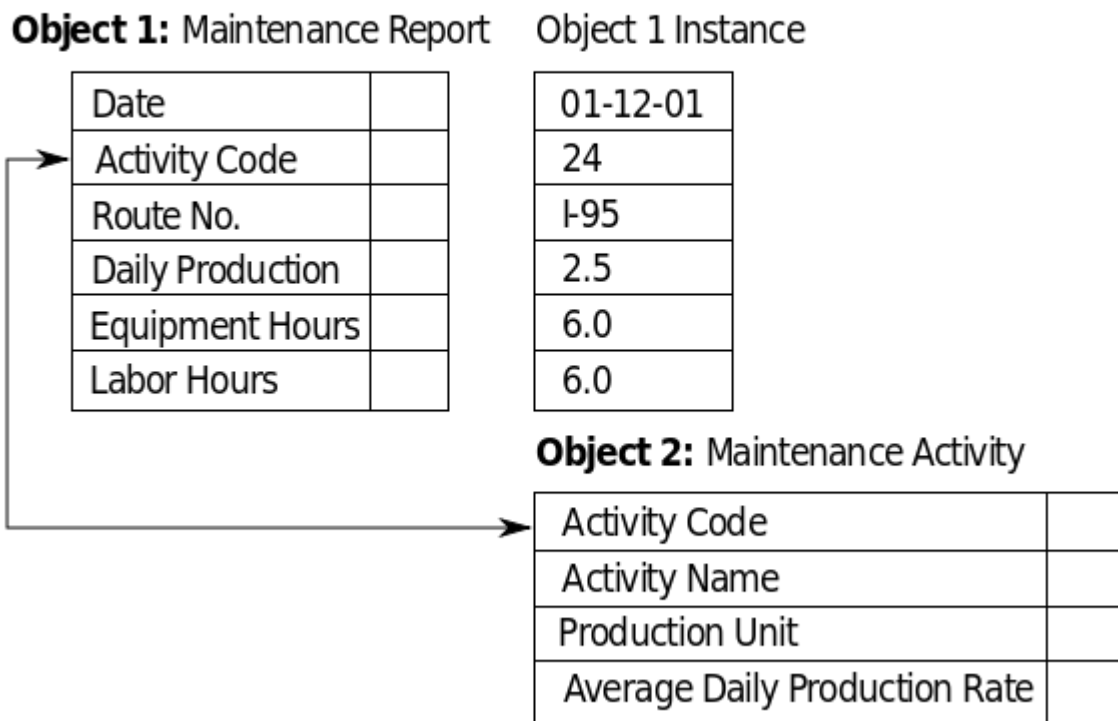
Details of research

As it says in the name object oriented databases are about objects by using this model you can define objects which are different from normal objects in a database in object oriented database these objects are the ability to develop a product then define and name it when this is done the object can then be referenced and called later. Each object can be different as you need different objects for different purposes but what you need to remember is that when we talk about objects we are talking about the whole object for example a car engine has many different parts each of these parts are a standalone component but when they are placed together we get a single object

which we called an engine so basically an object is multiply components put together to create the main finished produced when we use this ability is shortens the time to do certain actions as you are referring to the finished produced.

Figure E shows an example of an Object Oriented model.

Object-Oriented Model



Advantages

- Can manage large number of different data types.
- Objects with complex behaviors are easy to handle using inheritance and polymorphism.
- Reduces the large number of relations by creating objects.

Disadvantages

- Switching an existing database to OODBMS requires an entire change from scratch.
- An OODBMS is typically tied to a specific programming language this reduces its flexibility.
- Creates problem when deleting data in bulk.

Requirements

This the newest out of all the other database types and you can see that it uses a lot of expert knowledge if you know about programming and how to work with C# or C in general then this would be one that you would use. But in terms of using it for the database that will be outlined I don't think this is the best one to use as it don't use the coding that we will be don't so we would need to use C language to implement our requirements which we need to use certain programs that will allow us to use C.

Recommendations

This model is good if you are good at programming as this uses a lot of programming terms like objects and it use's open source, java and C# another good thing about this model is that it is cheap and easy to use both for beginners and experts so you don't have to worry about not being about to use it also as it can handle many data types you don't need to worry about only using text or integers. If you were a small business then this could be good for you but medium to large businesses would not use that as if you want to switch an existing database you would have to redo it from scratch this would cost too much and time a lot of time.

Conclusion

In this section of the report you will a detailed overview of the databases that have been around for many years using these as example we can start to think about what a database is and how it collects data. In the next section of the report we outline the requirements of our database.

Scenario

In this section I will be analyzing what the database is required to do to better understand what the database needs I will be picking out key features that the client has or would like in the database continuing from this I will be setting up business rules and operations this will allow me to restrict or give access to certain users on the database.

Requirements

In this section of the report they will outline what is required in the database this allows us to have an idea about what the database should contain so we list the entities and attributes that will be included in the database. These will allow us to understand the needs of that database so we can help our client also we should outline what each user should be required to do on the database this allows us to think about what rules and operations the database should follow.

Course Detail

Course name

Course ID

Credit value

Duration

Start date

End date

Fee

Tutor requirement

The database will store details for each course that is presented by Ace Training while looking through the scenario that has been given to us information which relates has been listed in the above table. The tutor would be required to create the course this means they will also have permission to add new course name, see credit value, duration, start date, end date, Fee and Course ID also the tutor would be able to update these details as appropriate.

Student requirement

The student would be required to view these details so when they are fully enrolled on their course they would be able to see all the information that is related to the course but they will be able to see certain information about the course before they join the course as this will give them more information about the course so they can make see if they would enjoy doing the different modules. So they would be able to see Course name, Credit Value, Duration, Start Date, End Date, Fee and Course ID. But they would only be able to view these details while the tutor can edit these the student won't be allowed.

Student Detail

First name

Middle name

Last name

Date of birth

Address one

Address two

County

Postal code

E-mail address

Date enrolled

Visa expiry date

Passport number

Tutor requirement

The database will store details for each course that is presented by Ace Training while looking through the scenario that has been given to us information which relates has been listed in the above table. The tutor would be able to see these details if the student is on their course so this means he would be able to see the students First name, Middle name, Last name, Date of Birth, Address one, Address two, County, Postal code, E-mail address, Date Enrolled, Visa Expiry Date and Passport Number also he won't be able to update these records as he sees fit.

Student requirement

The student would be required to fill out these details when they enroll on their course this would include First name, Middle name, Last name, Date of Birth, Address one, Address two, County, Postal code, E-mail address, Date Enrolled, Visa Expiry Date and Passport Number they would only fill out Passport Number and Visa Expiry if they are an international student. These details can be changed at any time but they would have to inform administrator about these changes for them to stay as the administrator will be in charge of changing any information they will need to be informed or the changes might not happen.

Tutor Detail

First name

Middle name

Last name

Date of birth

Address one

Address two

County

Postal code

E-mail address

Tutor ID

Next of kin

Office number

Phone number

National insurance number

Tutor requirements

The database will store details for each course that is presented by Ace Training while looking through the scenario that has been given to us information which relates has been listed in the above table. The tutor would be required to view these details this would include First name, Middle name, Last name, Date of Birth, Address one, Address two, County, Postal code, E-mail address, Tutor ID, Next of kin, Office number, Phone number and National insurance number. These details can be changed by the tutor but if they do change anything they need to inform an administrator.

Student requirements

The student would be required to view some of these details these would include First Name, Middle Name, Last Name, E-mail address and Office number. As some of the tutor details are personal the student would not have access to that information this would be for security reason as giving important information like Postal code could be dangerous for some tutors also students would not be able to edit these details only view.

Next of Kin

First name

Middle name

Last name

Date of birth

Address one

Address two

County

Phone number

E-mail address

Relation to Person

Tutor requirements

The database will store details for each course that is presented by Ace Training while looking through the scenario that has been given to us information which relates has been listed in the above table. The tutor would be required to view these details First name, Middle name, Last name, Date of Birth, Address one, Address two, County, Postal code, E-mail address, Date of Birth, Phone number and Relation to person. As the tutor he will be able to see his or her own next of kin details also the tutor will be able to see students as well but only those that are enrolled on their course and they will be able to edit their own details nut not the students.

Student requirements

The student would be required to view these details First name, Middle name, Last name, Date of Birth, Address one, Address two, County, Postal code, E-mail address, Date of Birth, Phone number and Relation to person. As this person is a student they will only be able to see their own details so they can't see their tutor next of kin this is for security reason as this person is not related to the university so the student doesn't need to see these details.

Resource

Type of resource

Date of availability

Available to who

Tutor requirements

The database will store details for each course that is presented by Ace Training while looking through the scenario that has been given to us information which relates has been listed in the above table. The tutor would be required to edit and view these details which include Type of resource, Date of availability and Available to who. Tutors would have full control over what resource is on their course also they will be able to select who can see the resources they put up and they can set a date when the resource is available.

Student requirements

The student would be only be able to see the resource if the tutor made it available for them if we take this into account the only detail they will see is the Type of resource. As they are a student they will not be able to edit any of the documents that are put on the database but they can view documents that are made available.

Quiz

Type of quiz

Name of quiz

Quiz ID

Number of quizzes

Correct answers

Incorrect answers

Date uploaded

Deadline date

Tutor requirements

The database will store details for each course that is presented by Ace Training while looking through the scenario that has been given to us information which relates has been listed in the above table. The tutor would be required to edit and view these details which include Type of Quiz, Quiz ID, Number of Quizzes, Correct answers and Incorrect answers. Tutors would have full control over what type of quizzes there are and the date they need to be done by.

Student requirement

The student would only be able to see the quiz if the tutor made it available for them if we take this into account the only detail they will see is the Type of quiz, quiz ID and the name of the quiz. As they are a student they will not be able to edit any of the quizzes that are put on the database but when downloaded they can edit the quiz.

Student Progress

Student Name

Student ID

Number of completed quizzes

Score in quizzes

Average grade

Highest score in quiz

Lowest score in quiz

Progress bar

Tutor requirement

The database will store details for each course that is presented by Ace Training while looking through the scenario that has been given to us information which relates has been listed in the above table. The tutor would be required to edit and view these details which include Student name, Student ID, Number of completed quizzes, Score in quizzes, Average grade, Highest score in quiz, Lowest score in quiz and Progress bar. The tutor would have to update these details as the course is in progress.

Student requirement

The student would only be required to view these details which include Student name, Student ID, Number of completed quizzes, Score in quizzes, Average grade, Highest score in quiz, Lowest score in quiz and Progress bar. The student would only be able to view these details not edit them as this would be the job of the tutor.

Attendance

Tutor ID

Course ID

Student ID

Number of Classes attended

Number of absences

Number of explained absences

Time

Date

Average attendance

Tutor requirement

The database will store details for each course that is presented by Ace Training while looking through the scenario that has been given to us information which relates has been listed in the above table. The tutor would be required to edit and view these details which include Tutor ID, Student ID, Student ID, Number of classes attended, Number of absences, Number of explained absences, Time, Date and Average attendance. Each tutor would be in charge of taking attendance this will list who is attending class and who isn't they will then send this to the administrators which will take the action needed.

Student requirement

The student would only be required to view these details which include Tutor ID, Student ID, Student ID, Number of classes attended, Number of absences, Number of explained absences, Time, Date and Average attendance. If the student has missed any lessons the administrators will e-mail them telling them which class they missed also what time and date the class was on.

Conclusion

In this section the requirements of the database have been outlined as to the clients request some added features have been put in the database this was also asked from the client with these requirements listed we now move on to business rules and operations.

Business Rules

A business rule is a restriction on the database so it is something that you can't do on the database this could be because that person doesn't have the permission to do a certain action or they might not have any need to do this action we place these rules so we can control who can do what on the database.

Course Details

BR1	Tutors cannot delete a course
BR2	Tutors cannot update Fee
BR3	Students cannot update any of the course details
BR4	Students cannot delete any course details
BR5	Students cannot insert course details

Student Details

BR6	Tutors cannot select students that aren't on their course
BR7	Tutors cannot update students details
BR8	Tutors cannot delete a student's details
BR9	Tutors cannot insert student details
BR10	Student cannot select other students details
BR11	Students cannot delete other student details
BR12	Student cannot delete their own student details
BR13	Student cannot update other student details
BR14	Student cannot insert student details
BR15	Administrators cannot update personal detail unless the student has changed them
BR16	Administrators cannot delete a student's profile while there are enrolled on a course

Tutors Details

BR17	Tutors cannot select other tutors details
BR18	Tutors cannot delete tutors details
BR19	Tutors cannot insert tutor details
BR20	Students cannot update tutors details
BR21	Students cannot select tutors personal details
BR22	Students cannot delete tutors details
BR23	Student cannot insert tutor details
BR24	Administrators cannot update tutors details unless the tutor has changed them
BR25	Administrators cannot delete a tutors profile while there are working at the university

Next of kin

BR26	Tutors cannot update students next of kin details
BR27	Tutors cannot delete students next of kin details
BR28	Tutors cannot select students next of kin details
BR29	Tutors cannot insert students next of kin details
BR30	Tutors cannot select other tutors next of kin details
BR31	Tutors cannot delete tutors next of kin details
BR32	Tutors cannot update other tutor next of kin details
BR33	Tutors cannot insert tutors next of kin details
BR34	Students cannot update tutors next of kin details
BR35	Students cannot delete tutors next of kin details
BR36	Students cannot select tutors next of kin details
BR37	Students cannot select other students next of kin details
BR38	Students cannot insert other students next of kin details
BR39	Students cannot delete other student next of kin details
BR40	Students cannot update other students next of kin details
BR41	Administrators cannot delete a tutor next of kin while they are working at the university
BR42	Administrators cannot delete a student's next of kin while they are enrolled on a course

Resource

BR43	Tutors cannot upload hyperlinks
BR44	Tutors cannot upload the same file while one exist on the database
BR45	Students cannot update resources that have been uploaded
BR46	Students cannot delete resources on the database
BR47	Students cannot insert resources on the database
BR48	Administrators cannot delete any resources in a activity course

Quiz

BR49	Tutors cannot upload a quiz that shares the same ID
BR50	Students cannot view quizzes until the tutor has allowed them to see it
BR51	Students cannot delete quizzes that are uploaded to the database
BR52	Students cannot edit quizzes that are uploaded to the database
BR53	Students cannot insert quizzes on the database
BR54	Administrators cannot delete any quizzes that are on a activity course
BR55	Administrators cannot edit any quizzes that are on a activity course
BR56	Administrators cannot insert quizzes
BR57	Administrators cannot update quizzes

Student Progress	
BR58	Tutors cannot delete a student progress profile
BR59	Tutors cannot view student progress that aren't on their course
BR60	Tutors cannot insert student progress
BR61	Students cannot delete other student progress
BR62	Students cannot delete their own progress
BR63	Students cannot update their own progress
BR64	Students cannot update other student progress
BR65	Students cannot select other student progress
BR66	Students cannot insert their own student progress
BR67	Students cannot insert other student progress
BR68	Administrators cannot delete a student progress while they are enrolled on a course
BR69	Administrators cannot update student progress

Attendance	
BR70	Tutors cannot delete a student's attendance
BR71	Tutors cannot select students attendance that aren't on their course
BR72	Students cannot delete their own attendance
BR73	Students cannot delete other students attendance
BR74	Students cannot select other students attendance
BR75	Students cannot update their own attendance
BR76	Students cannot update other students profiles
BR77	Students cannot insert their own attendance
BR78	Students cannot insert other students attendance
BR79	Administrators cannot delete a student's attendance while they are enrolled on a course
BR80	Administrators cannot update a students attendance

Business Operation

A Business Operation is a database action this means actions that can take place on the database for example one of them could be Tutors can view Students profiles this is an action that tutors can do on the database the amount of actions you can do on the database depends what your position is in the business.

Course Details	
BO1	Tutors can insert new courses
BO2	Tutors can update courses that already exist but only if they are assigned to that course

BO3	Tutors can select the course they are assigned to
BO4	Students can select course details
BO5	Administrators can update course details if they need to be updated
BO6	Administrators can delete a course if a course is not active

Student Details

BO7	Tutors can select student details that are enrolled on their course
BO8	Students can update Student details
BO9	Students can select their own details
BO10	Administrators can select student details
BO11	Administrators can update the students details if the student has told them about any new information that needs updating
BO12	Administrators can delete a student delete that aren't enrolled at the university
BO13	Administrators can insert new Student details

Tutors Details

BO14	Tutors can select their own details
BO15	Tutors can update their own details
BO16	Students can select tutors that are teaching them but they will only have access to certain information so personal information won't be on display
BO17	Administrators can select tutor details
BO18	Administrators can insert new tutor details
BO19	Administrators can assign roles to the tutor
BO20	Administrators can delete tutor details that aren't working at the university

Next of kin

BO21	Tutors can select their own Next of kin
BO22	Tutors can update their own Next of kin
BO23	Tutors can insert new Next of Kin
BO24	Students can select their own Next of kin
BO25	Students can update their own Next of kin
BO26	Administrators can select tutors Next of kin
BO27	Administrators can select students Next of kin
BO28	Administrators can insert tutors Next of kin
BO29	Administrators can insert student Next of kin
BO30	Administrators can delete tutors Next of kin if they are no longer working at the university
BO31	Administrators can delete students Next of kin if they are no longer at the university

Resource	
BO32	Tutors can insert new resources to the database
BO33	Tutors can set dates on when the resource should become available
BO34	Tutors can set who can see certain resources
BO35	Tutors can delete resource uploaded to the database
BO36	Tutors can select resources that are on the database
BO37	Students can select resources that are available to them
BO38	Students can download resources that have been uploaded to the database
BO39	Administrators can take away a tutors permission to upload to the database if needed
BO40	Administrators can delete resource on the database if needed
BO41	Administrators can block certain files from being uploaded
BO42	Administrators can select resources on the database

Quiz	
BO43	Tutors can insert new quizzes
BO44	Tutors can update quizzes
BO45	Tutors can select quizzes
BO46	Students can select quizzes when it becomes available to them
BO47	Students can insert their quizzes to certain spots that the tutor has created
BO48	Students can download quizzes that have been uploaded to the database
BO49	Administrators can delete quizzes that are on course which aren't active
BO50	Administrators can select quizzes on the database

Student Progress	
BO51	Tutors can select their students' progress
BO52	Tutors can update their student's progress
BO53	Students can select their own progress
BO54	Administrators can delete a student's progress if that student is not enrolled at the university
BO55	Administrators can insert student progress when a new student enrolls on a course

Attendance	
BO56	Tutors can select their student's attendance
BO57	Tutors can update their students' attendance
BO58	Tutors can report a student's low attendance to the administrators
BO59	Students can select their own attendance
BO60	Administrators can select a student's attendance when reported by the tutor
BO61	Administrators can delete a student's attendance if they are no longer at the university
BO62	Administrators can contact the student if they get reported a number of times

Conclusion

In this section I have listed the different business rules and operations that will be enforced when it comes to implementing the database in SQL we will use these rules and operations to restrict and grant users actions on the database. In the next section we will be designing the conceptual diagram this will be a basic structure of the database.

Chapter two

Conceptual Design

Conceptual Diagram

This is the first design of the database to show the client how each entity will be connected this also allows us to understand how the database works this is need in order to create a logical design. The reason in creating a conceptual diagram is to show the client how the database will be linked together in a easy and simple way to read so we can clearly see the different relationships we have as well as the different entities that are going to be on the database as well.

There are many different terms in conceptual design that will be used to show the client how the database works below will be an explanation of these terms...

Entities- This is where we will store all our data that relates to that entity for example Student is the entity and we would store information inside it which relates to student this could be first name, last name and other possible data. Entities can be any object in the world some of these are physical objects with real existence for example a car or a person would be included in this as they have a physical form others like gases don't so we put them as metaphysical.

Attributes- This is the data we add to our entities to make them produce values for example an attribute would be eye color and the value for this could be blue as long as it is used to describe the entity it would be an attribute.

Describing attribute- An attribute that is used to describe the entity would follow under this for example Hair color is a describing attribute as it tell us something about the entity but gives us no data that would be used to find one person if you see a line with a white circle on top of it coming out of an entity this represents a describing attribute.

Identifying attribute- This is an attribute that is used to identify someone or something for example if we are talking about a person then we could use an ID number this would allow us to find a certain person or product as it would be give a special code if you a line with a black circle on top of it coming out of an entity this represents a identifying attribute.

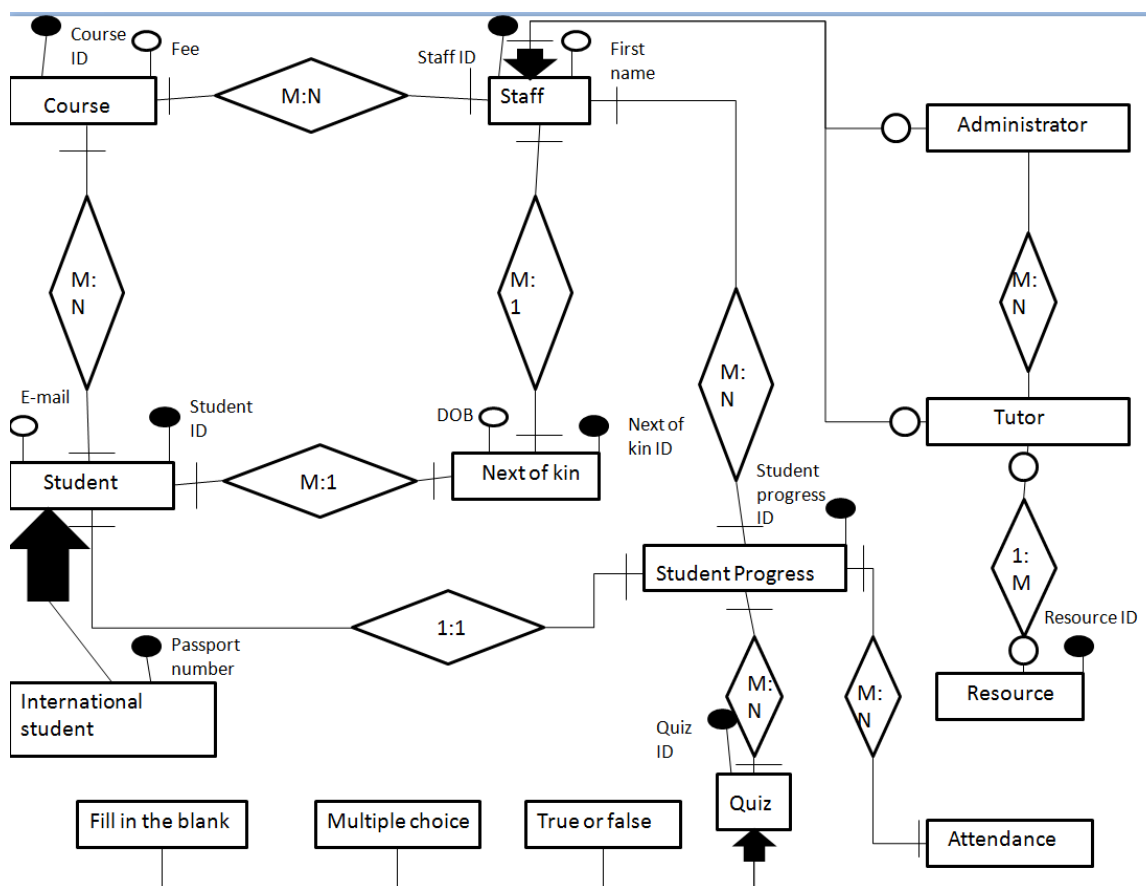
Relationship- There is many different relationships in a conceptual design the first is one to one this is a single link that links two entities together. The next type is called one too many or many to one this is where one entity has many relations with the other entity for example we can say that a single student will have one next of kin while one next of kin will have many students this is a rough example but you should get the idea of this relation. The last one is called many too many this follows the same idea as one too many but instead of one entity having many relations with other one both entities have many relations with each other.

Directionality- This allows us to see which entities are strong and weak by using an arrow we can see which is strong and which is weak this helps us to identify parent and child entities.

Recursive- This is when one entity instance is related to another entity instance in the same entity this means that if you have one entity which we will call staff then you have two instance of that entity and they both relate they you have a recursive relationship an example of this would be staff and staff manager.

Generalization- In this we have two types of entities one of them is the super type and another one is the sub type when we create this type of relation the subtypes will allows have the same attributes as the super type and it will have one extra attribute.

Strong and weak entities- A strong entity is one that is none dependent so it can exist by itself on the other hand a weak entity is one that needs other entities to exist so an example would be a person and a bus ticket the person can exist without a bus ticket but the bus ticket can't exist without the person.

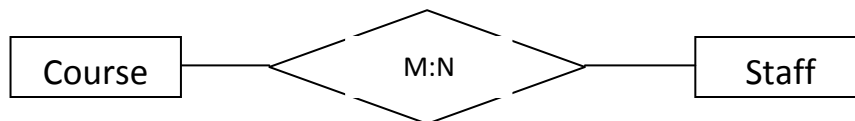


Conclusion

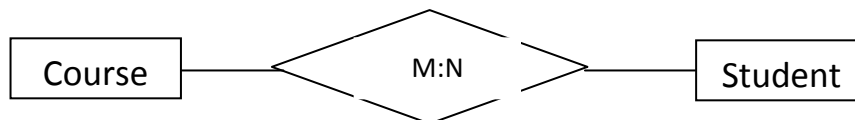
In this section of the report the conceptual diagram was set up to show the user how each entity would link together and their relationship between each entity was outlined also explanations of the terms used in conceptual diagram were outlined. This moves us on to entity relationships.

Entity Relationships

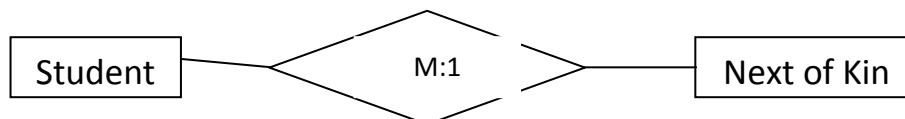
In this section we outline the how each entity is related to one another by doing this we can clearly see what entity links with what. We do this so we can see how they link and what they link to this further break the conceptual design down to a simpler form so the client can understand.



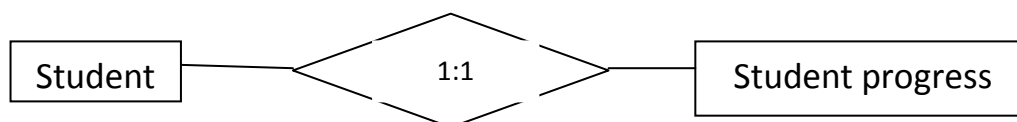
A course may have many staff on it.
A member of staff may teach on many courses.



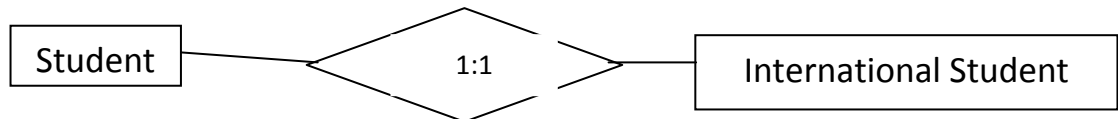
A course may have many students on it.
A student may be enrolled on many courses.



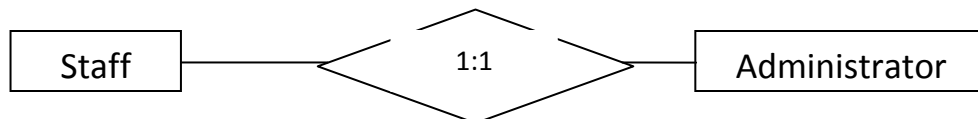
A single student has one next of kin.
A next of kin may have many students.



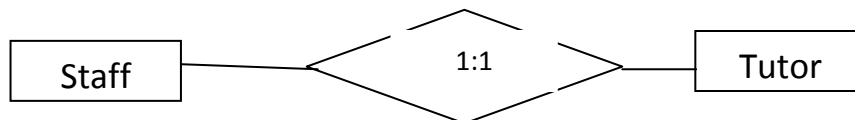
A single student has one student progress.
A single student progress will have one student.



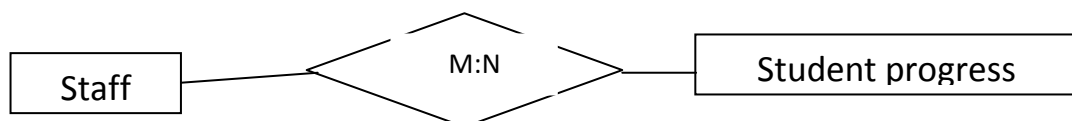
A single student may be an international student.
A single international student will be a student.



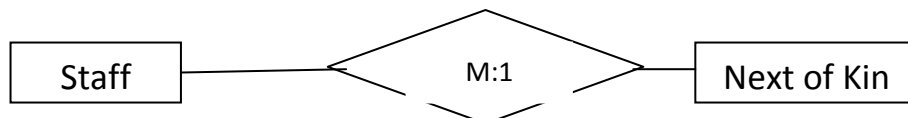
A single staff member may be an administrator.
A single administrator is a staff member.



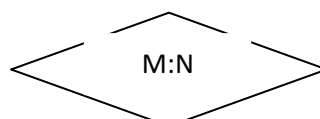
A single staff member may be a tutor.
A single tutor is a staff member.



A staff member may be in charge of many student progresses.
A student progress may be managed by many staff members.

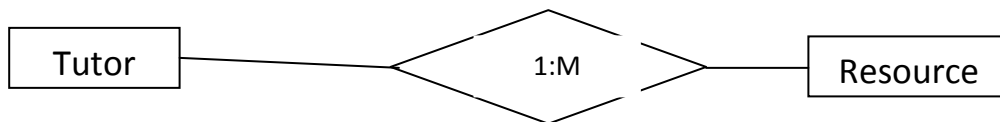


A staff member will have one next of kin.
A single next of kin may belong to many staff members.

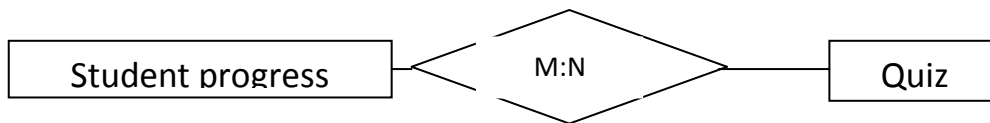




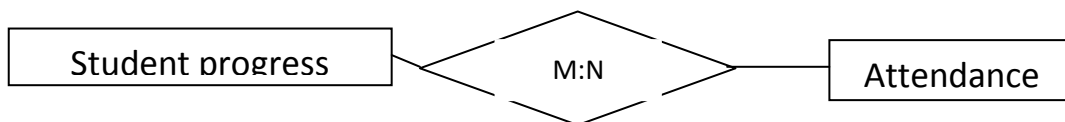
An administrator may create many Tutors.
A tutor is created by many administrators.



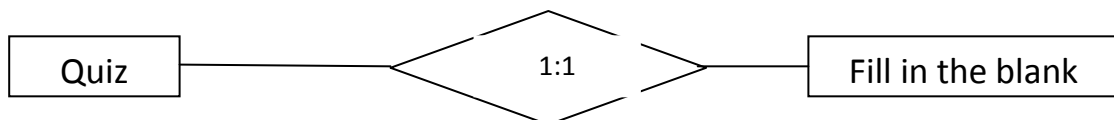
A tutor may own many resources
A resource may be owned by one tutor.



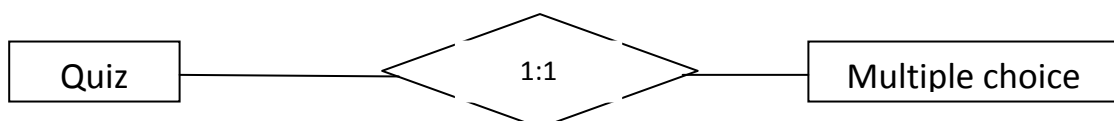
A student progress may have many quizzes.
A quiz may belong to many student progresses



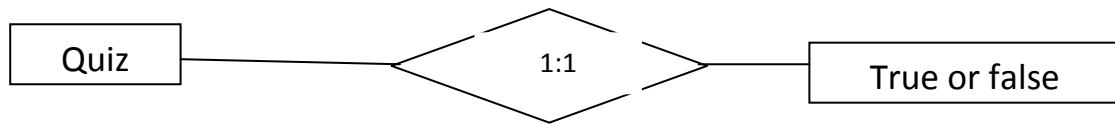
A student progress may have many attendance records
A attendance will be stored on many student progress.



A single quiz may be fill in the blank.
A single fill in the blank will belong to quiz.



A single quiz may be multiple choice.
 A single multiple choice will belong to quiz.



A single quiz may be true of false.
 A single true of false will belong to quiz.

Conclusion

In this section all the relationships in the conceptual design were highlighted and given a short statement on how the user should read them this gives us an idea of they will work in the database. We now move on to the diagram script.

Diagram script

In this section we create a series of tables which allows us to start making the logical design we would normally start this first then half way through this process we would start the logical design this would then allow us to finish the script. The reason why we do this is to break down the logical design in a simpler form so anyone can understand it.

Entity	Attribute	Synonym	Related to
Course	Course name Course ID Credit value Duration Start date End date Fee	Module	Student, Staff
Student	First name Middle name Last name Date of birth Address one Address two County Postal code E-mail address Date enrolled	Learner	Course, Next of Kin and International student
International student	Visa expiry date Passport number	Learner	Student
Staff	First name	Employee	Course, Course tutor,

	Middle name Last name Date of birth Address one Address two County Postal code E-mail address Staff ID Next of kin Office number Phone number National insurance number		Administrators, Next of Kin, Tutors and Student progress
Administrator	Administrators ID	Employee	Staff, Tutors
Tutor	Staff ID	Employee	Staff, Resource and Administrators
Next of kin	First name Middle name Last name Date of birth Address one Address two County Phone number E-mail address Relation to Person	Near relative	Staff and Student
Resource	Type of resource Date of availability Available to who	support	Tutors and Course Tutors
Quiz	Type of quiz Name of quiz Quiz ID Number of quizzes Correct answers Incorrect answers Date uploaded Deadline date	Test of knowledge	Student progress, Fill in the blank, multiple choice and true or false
Fill in the blank			Quiz
Multiple choice			Quiz
True or false			Quiz
Student Progress	Student Name Student ID Number of completed quizzes	Growth	Staff , Attendance, Quiz and Student

	Score in quizzes		
	Average grade		
	Highest score in quiz		
	Lowest score in quiz		
	Progress bar		
Attendance	Room ID	Participation	Student progress
	Number of Classes		
	attended		
	Number of absences		
	Number of explained		
	absences		
	Time		
	Date		
	Average attendance		

Conclusion

In this section of the report the diagram script has been fully designed and ready to show to the client in this we have listed entities that will be used in our database and their attributes by using this we can move on the next stage of the report which is logical design.

Logical Design

In this section of the report the logical design will be set out so the client can see not only the entities that will be in the database but each of the attributes that are in the database this furthers our understanding of how the database will work as we can see what each entity will store.

Logical Design

Before the user looks at the logical design I would recommend that they take a look at this explanation part below as it will explain a few things about the logical design and why it is important that we do this stage before going into further design work.

Primary and foreign keys: In each table you must have a primary key this key is an attribute that is uniquely identify a specific instance of an entity and foreign keys are primary keys that have been used to link with another table when they do this they become foreign keys.

Entities and attributes: Just like with the conceptual diagram we layout our entities and attributes but this time we put them together to show how they will act in the database. For example one of our entities is called course we would show it on the design with a table then just below that will be the attributes that have been listed in the diagram script this shows the client what each entity has in terms of attributes and how they will collect this information.

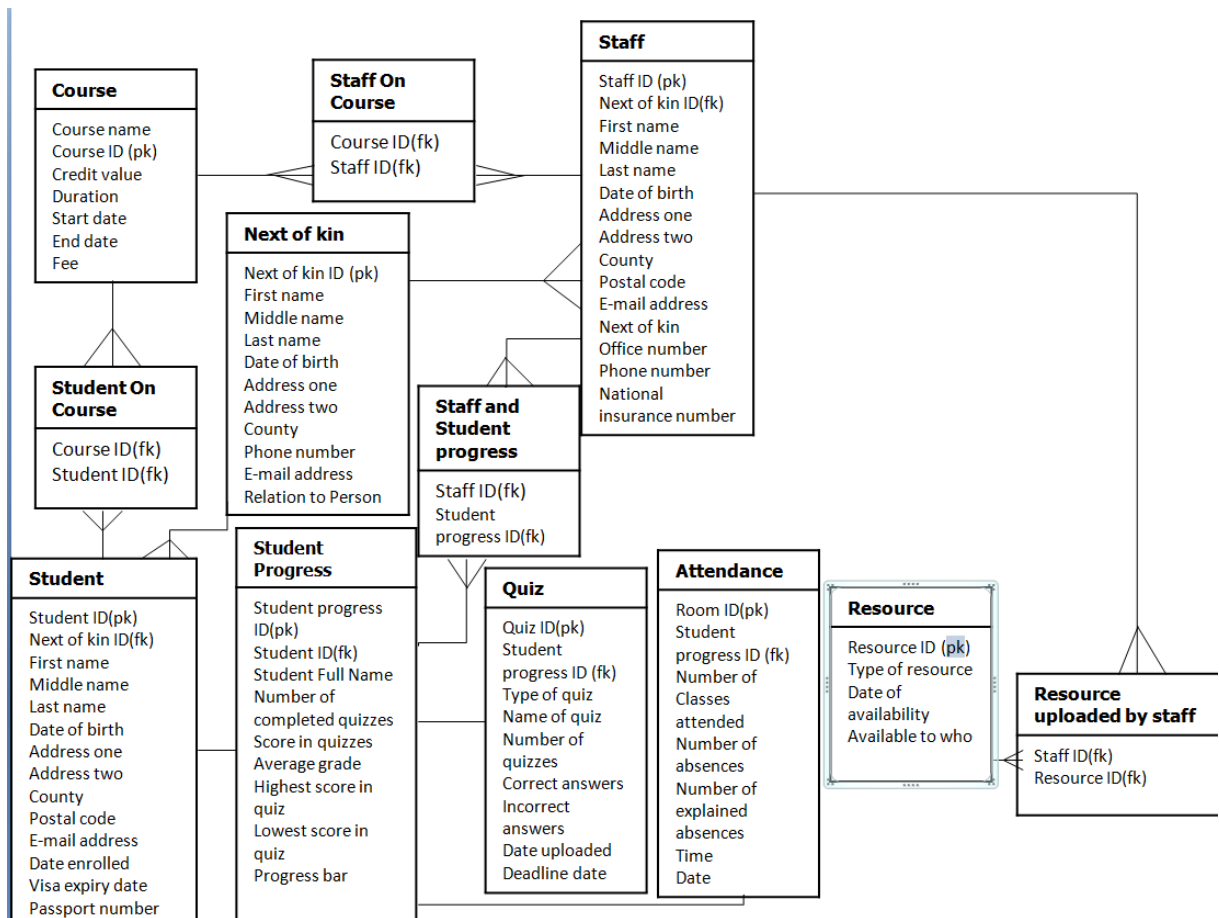
Relationships in logical design: Like with the conceptual diagram relationships play a big role in how we set our tables up but before we set our tables we need to understand how these relationships work as they act differently in logical design. To start off with we can't use many to many relationships and we can't use Generalizations these are not allowed in logical design also we have to relook at one to one relations as they act differently as well first we should look at many to many and Generalization as they are easier to understand.

The way we present many to many in a logical is by having two entities share a table and place arrows that lead from the entities to the table this shows us that this table is connect to these entities it will become more easier when you see one of these tables in the design just try and think how it would look. When it comes to Generalization we can do two methods the first one is collapse it back into the super class so it will become a part of the original entity or create a separate table and link it with a one to one relation.

One to one relations are normally not allowed if you can reduce it to a single entity then you have to do that but before we think about this we have to look at the different options we have when it

comes to a one to one relation the first one is mandatory at both ends and then mandatory at one end and optional at the other and final optional at both ends. When it comes to mandatory on both ends there are a few reasons why you might not put them together one of these could be that both entities represent entities in the real world so by putting them together you would be getting rid of something that exist in the real world which you can't do at it is real.

If the relation is mandatory and optional then you need to combine them and finally if the relation is optional and optional you keep them apart.



Conclusion

In this section of the report the logical design has been outlined to the client this shows how the entities and attributes will work also it highlights the primary and foreign keys in each table. We now move on to Normalization which starts off at cumulative design.

Normalization

Cumulative Design

This is a breakdown of the logical design this makes it easier to read for the user as they can see each entity and the follows attributes. By creating this basic idea our SQL can start to be designed.

This is also called DDL which stands for data definition language which is a standard for commands that define different structures in a database so if you wanted to create anything in a table or edit current data then you would have to use DDL as when you use this the database understands what you are saying and will allow the changes you want to happen.

Course (Course ID, Course name, Credit value, Duration, Start date, End date, Fee)

StudentonCourse (Course ID, Student ID)

FK Course ID → Course, UPDATE CASCADE, DELETE RESTRICT

FK Student ID → Student, UPDATE CASCADE, DELETE RESTRICT

StaffOnCourse (Course ID, Staff ID)

FK Course ID → Course, UPDATE CASCADE, DELETE RESTRICT

FK Staff ID → Staff, UPDATE CASCADE, DELETE RESTRICT

Student (Student ID, Next of kin ID, First name, Middle name, Last name, Date of birth, Address one, Address two, County, Postal code, E-mail address, Phone number, Date enrolled, Visa expiry date, Passport number)

FK Next of Kin ID → Next of Kin, UPDATE CASCADE, DELETE RESTRICT

Staff(Staff ID, Next of kin ID, First name, Middle name, Last name, Date of birth, Address one, Address two, County, Postal code, E-mail address, Office number, Phone number, National insurance number)

FK Next of Kin ID → Next of Kin, UPDATE CASCADE, DELETE RESTRICT

Next of Kin (Next of kin ID, First name, Middle name, Last name, Date of birth, Address one, Address two, County, Postcode, Phone number, E-mail address, Relation to Person)

Student Progress (Student progress ID, Student ID, Student full Name, Number of completed quizzes, Score in quizzes, Average grade, Highest score in quiz, Lowest score in quiz, Progress bar)

StaffandStudentProgress (Staff ID, Student progress ID)

FK Staff ID → Staff, UPDATE CASCADE, DELETE RESTRICT

FK Student progress ID → Student Progress, UPDATE CASCADE, DELETE RESTRICT

Quiz (Quiz ID, Student progress ID, Type of quiz, Name of quiz, Number of quizzes, Correct answers, Incorrect answers, Date uploaded, Deadline date)

FK Student progress ID → Student Progress, UPDATE CASCADE, DELETE RESTRICT

Attendance (Room ID, Student progress ID, Number of Classes attended, Number of absences, Number of explained absences)

Resource (Resource ID, Type of resource, Date of availability, Available to who)

Resourceuploadedbystaff (Staff ID, Resource ID,)

FK Staff ID → Tutor, UPDATE CASCADE, DELETE RESTRICT

FK Resource ID → Resource, UPDATE CASCADE, DELETE RESTRICT

Conclusion

In this section of the report the cumulative design has been created this will be used later to build the SQL database now we move on to normalizing our tables.

Normalized Tables

In this section we create our normalized tables the main objective is to make a logical data model for relational database systems the reason why normalization happens in a design document is because it helps reduce data redundancies, helps eliminate data anomalies and produces controlled redundancies to link tables.

UNF-At this stage we need to create tables in order for us to start the normalization process these tables need to have columns and rows in them.

1NF- At this stage we need to split our table into two different tables the first one is our table that doesn't have any repeating attributes the second one has all our repeating attributes in it. After we split then we move the primary key from the first table to the second one this allows us to show they are linked together.

2NF- At this stage we are checking for key dependency then we spilt then into the right about of tables needed to show what is dependent on what is important here is that all non key attribute is fully functionally dependent on the primary key.

3NF- In this stage we check the non-key attributes and see if they are dependent on the primary key if they are related to the primary key then they stay in the same table but if they are independent then they will be moved to another table.

UNF - Student ID		
Field	Value one	Value two
Course name		
Course ID		
Credit value		
Duration		
Start date		
End date		
Fee		
<u>Student ID</u>		
First name		
Middle name		
Last name		
Date of birth		
Address one		
Address two		
County		
Postal code		
E-mail address		
Date enrolled		
Visa expiry date		
Passport number		
Staff ID		
First name		
Middle name		
Last name		

Date of birth
Address one
Address two
County
Postal code
E-mail address
Next of kin
Office number
Phone number
National insurance number
Next Of Kin ID
First name
Middle name
Last name
Date of birth
Address one
Address two
County
Phone number
E-mail address
Relation to Person
Resource ID
Type of resource
Date of availability
Available to who
Quiz ID
Type of quiz
Name of quiz
Number of quizzes
Correct answers
Incorrect answers
Date uploaded
Deadline date
Student Progress ID
Student Name
Number of completed quizzes
Score in quizzes
Average grade

Highest score in quiz
Lowest score in quiz
Progress bar
Room ID
Number of Classes attended
Number of absences
Time
Date
Number of explained absences
Average attendance

1NF - Student ID			
Field	Value one	Value two	Value three
<u>Student ID</u>			
First name			
Middle name			
Last name			
Date of birth			
Address one			
Address two			
County			
Postal code			
E-mail address			
Date enrolled			
Visa expiry date			
Passport number			
Next Of Kin ID			
First name			
Middle name			
Last name			
Date of birth			
Address one			
Address two			
County			
Phone number			
E-mail address			

Relation to Person
Student Progress ID
Student Name
Number of completed quizzes
Score in quizzes
Average grade
Highest score in quiz
Lowest score in quiz
Progress bar
Average attendance

UNF - Course name			
Field	Value one	Value two	Value three
<u>Student ID</u>			
<u>Course name</u>			
Course ID			
Credit value			
Duration			
Start date			
End date			
Fee			
Staff ID			
First name			
Middle name			
Last name			
Date of birth			
Address one			
Address two			
County			
Postal code			
E-mail address			
Next of kin			
Office number			
Phone number			
National insurance number			

Resource ID
Type of resource
Date of availability
Available to who
Quiz ID
Type of quiz
Name of quiz
Number of quizzes
Correct answers
Incorrect answers
Date uploaded
Deadline date
Room ID
Number of Classes attended
Number of absences
Time
Date
Number of explained absences

1NF - Course name			
Field	Value one	Value two	Value three
<u>Student ID</u>			
<u>Course name</u>			
Course ID			
Credit value			
Duration			
Start date			
End date			
Fee			
Staff ID			
First name			
Middle name			
Last name			
Date of birth			
Address one			
Address two			

County
Postal code
E-mail address
Next of kin
Office number
Phone number
National insurance number

1NF - Resource ID			
Field	Value one	Value two	Value three
<u>Student ID</u>			
<u>Course name</u>			
<u>Resource ID</u>			
Date of availability			
Available to who			

1NF - Quiz ID			
Field	Value one	Value two	Value three
<u>Student ID</u>			
<u>Course name</u>			
<u>Quiz ID</u>			
Name of quiz			
Number of quizzes			
Correct answers			
Incorrect answers			
Date uploaded			
Deadline date			

1NF - Room ID			
Field	Value one	Value two	Value three
<u>Student ID</u>			
<u>Course name</u>			

Room ID

Number of Classes attended

Number of absences

Number of explained absences

1 NF - Time ID

Field	Value one	Value two	Value three
-------	--------------	--------------	----------------

Student ID

Course name

Room ID

Time ID

Date

2NF - Student ID

Field	Value one	Value two	Value three
-------	--------------	--------------	----------------

Student ID

First name

Middle name

Last name

Date of birth

Address one

Address two

County

Postal code

E-mail address

Date enrolled

Visa expiry date

Passport number

Next Of Kin ID

First name

Middle name

Last name

Date of birth

Address one

Address two

County
Phone number
E-mail address
Relation to Person
Student Progress ID
Student Name
Number of completed quizzes
Score in quizzes
Average grade
Highest score in quiz
Lowest score in quiz
Progress bar
Average attendance

2NF - CourseNameonStudentID			
Field	Value one	Value two	Value three
<u>Student ID</u>			
<u>Course name</u>			

2NF - Course name			
Field	Value one	Value two	Value three
<u>Course name</u>			
Course ID			
Credit value			
Duration			
Start date			
End date			
Fee			
Staff ID			
Type of staff			
First name			
Middle name			
Last name			
Date of birth			
Address one			

Address two
County
Postal code
E-mail address
Next of kin
Office number
Phone number
National insurance number

2NF - StudentIDandCourseNameonResourceID			
Field	Value one	Value two	Value three
<u>Student ID</u>			
<u>Course name</u>			
<u>Resource ID</u>			

2NF - ResourceID			
Field	Value one	Value two	Value three
<u>Resource ID</u>			
Date of availability			
Available to who			

2NF- StudentIDandCourseIDonQuizID			
Field	Value one	Value two	Value three
<u>Student ID</u>			
<u>Course name</u>			
<u>Quiz ID</u>			

2NF - Quiz ID			
Field	Value one	Value two	Value three
<u>Quiz ID</u>			
Name of quiz			

Number of quizzes
Correct answers
Incorrect answers
Date uploaded
Deadline date

2NF - CourseNameonStudentIDandRoomID			
Field	Value one	Value two	Value three
<u>Course name</u>			

2NF - StudentIDandRoomID			
Field	Value one	Value two	Value three
<u>Student ID</u>			
<u>Room ID</u>			
Staff ID			
Course ID			
Number of Classes attended			
Number of absences			
Number of explained absences			

2NF - CourseNameonStudentIDadnRoomIDandTimeID			
Field	Value one	Value two	Value three
<u>Course name</u>			

2NF - StudentIDadnRoomIDandTimeID			
Field	Value one	Value two	Value three
<u>Student ID</u>			
<u>Room ID</u>			
<u>Time ID</u>			
Date			

3NF - Student ID

Field	Value one	Value two	Value three
-------	-----------	-----------	-------------

Student ID**First name****Middle name****Last name****Date of birth****Address one****Address two****County****Postal code****E-mail address****Date enrolled****Visa expiry date****Passport number****Next Of Kin ID****Student Progress ID**

3NF - Next of Kin

Field	Value one	Value two	Value three
-------	-----------	-----------	-------------

Next Of Kin ID**First name****Middle name****Last name****Date of birth****Address one****Address two****County****Phone number****E-mail address****Relation to Person**

3NF - Student Progress ID			
Field	Value one	Value two	Value three
<u>Student Progress ID</u>			
Student Name			
Number of completed quizzes			
Score in quizzes			
Average grade			
Highest score in quiz			
Lowest score in quiz			
Progress bar			
Average attendance			

3NF - Course name on Student ID			
Field	Value one	Value two	Value three
<u>Student ID</u>			
<u>Course name</u>			

3NF - Course name			
Field	Value one	Value two	Value three
<u>Course name</u>			
Course ID			
Credit value			
Duration			
Start date			
End date			
Fee			
Staff ID			

3NF - Staff ID			
Field	Value one	Value two	Value three
<u>Staff ID</u>			
Type of staff			
First name			
Middle name			
Last name			
Date of birth			
Address one			
Address two			
County			
Postal code			
E-mail address			
Next of kin			
Office number			
Phone number			
National insurance number			

3NF - StudentIDandCourseNameonResourceID			
Field	Value one	Value two	Value three
<u>Student ID</u>			
<u>Course name</u>			
<u>Resource ID</u>			

3NF - ResourceID			
Field	Value one	Value two	Value three
<u>Resource ID</u>			
Date of availability			
Available to who			

3NF - StudentIDandCourseIDonQuizID

Field	Value one	Value two	Value three
-------	--------------	--------------	----------------

Student ID

Course name

Quiz ID

3NF - Quiz ID

Field	Value one	Value two	Value three
-------	--------------	--------------	----------------

Quiz ID

Name of quiz

3NF - Name of Quiz

Field	Value one	Value two	Value three
-------	--------------	--------------	----------------

Name of quiz

Number of quizzes

Correct answers

Incorrect answers

Date uploaded

Deadline date

3NF - CourseNameonStudentIDandRoomID

Field	Value one	Value two	Value three
-------	--------------	--------------	----------------

Course name

3NF - StudentIDandRoomID

Field	Value one	Value two	Value three
-------	--------------	--------------	----------------

Student ID

Room ID

Staff ID

Number of Classes attended

Number of absences

Number of explained absences

3NF -StaffIDand CourseID

Field	Value one	Value two	Value three
-------	--------------	--------------	----------------

Staff ID

Course ID

3NF - CourseNameonStudentIDadnRoomIDandTimeID
--

Field	Value one	Value two	Value three
-------	--------------	--------------	----------------

<u>Course name</u>

3NF - StudentIDadnRoomIDandTimeID

Field	Value one	Value two	Value three
-------	--------------	--------------	----------------

<u>Student ID</u>

<u>Room ID</u>

<u>Time ID</u>

Date

Conclusion

In this section of the report all the tables have been normalized so that the client can see how data flows though the database in the next section we cover implementation.

Chapter three

Implementation

In this section of the report we implement our Cumulative Design to SQL but before we can start coding our database. There are a few things that need to be done first the database needs to be designed this allows us to see what will be in each table that will be on the database to do this we use this section called Field Attributes.

This tells us how many tables will be in the database and what they will contain this shows the client what will be stored in that table and what data is needed in order for data to be inputted into the table.

Field Attributes

tblCourse						
Field name	Data Type	Field Size	Format	Key type	Index	Required
Course ID	Smallint	10	Autonumber	Primary	Yes (no duplicates)	Yes
Course name	Varchar	25				Yes
Credit value	Tinyint	3				Yes
Duration	Varchar	10				Yes
Start date	Date					Yes
End date	Date					Yes
Fee	Decimal		3,3			Yes

tblStudentOnCourse						
Field name	Data Type	Field Size	Format	Key type	Index	Required
Course ID	smallint	10		foreign		Yes
Student ID	smallint	10		Foreign		Yes

FK Course ID → Course, UPDATE CASCADE, DELETE RESTRICT

FK Student ID → Student, UPDATE CASCADE, DELETE RESTRICT

tblStaffOnCourse						
Field name	Data Type	Field Size	Format	Key type	Index	Required

Course ID	Smallint	10		foreign		Yes
Staff ID	smallint	10		Foreign		Yes

FK Course ID → Course, UPDATE CASCADE, DELETE RESTRICT

FK Staff ID → Staff, UPDATE CASCADE, DELETE RESTRICT

tblStudent						
Field name	Data Type	Field Size	Format	Key type	Index	Required
Student ID	Smallint	10	Autonumber	Primary	Yes (no duplicates)	Yes
Next of Kin ID	smallint	10		Foreign		Yes
First name	Varchar	35				No
Middle name	Varchar	35				No
Last name	Varchar	35			Yes (no duplicates ok)	Yes
DOB	Date				Yes (duplicates ok)	Yes
Address 1	Varchar	20				Yes
Address 2	Varchar	20				No
County	Varchar	15				Yes
Postcode	Varchar	7	aaddddaa			Yes
E-mail	Varchar	30	aaddddaa			No
Phone	varchar	15				Yes

FK Next of Kin ID → Next of Kin, UPDATE CASCADE, DELETE RESTRICT

tblNextof Kin						
Field name	Data Type	Field Size	Format	Key type	Index	Required
Next of Kin ID	smallint	10		Primary	Yes (no duplicates)	Yes
First name	varchar	35				Yes
Middle name	Varchar	35				No
Last name	Varchar	35			Yes (no duplicates)	Yes
DOB	Date					Yes
Address 1	Varchar	20				Yes
Address 2	Varchar	20				No

County	Varchar	15				Yes
Postcode	Varchar	7				Yes
Phone Number	Varchar	15				Yes
E-mail address	Varchar	30				No
Relation to Person	Varchar	6				Yes

tblStaff						
Field name	Data Type	Field Size	Format	Key type	Index	Required
Staff ID	smallint	10	Autonumber	Primary	Yes (no duplicates)	Yes
Next of Kin ID	Smallint	10		Foreign		Yes
First name	Varchar	35				No
Middle name	Varchar	35				No
Last name	Varchar	35			Yes (no duplicates ok)	Yes
DOB	Date				Yes (duplicates ok)	Yes
Address 1	Varchar	20				Yes
Address 2	Varchar	20				No
County	Varchar	15				Yes
Postcode	Varchar	7	aaddddaa			Yes
E-mail	Varchar	30	aaddddaa			No
Office number	Varchar	10				No
Phone	Varchar	15				Yes
National insurance number	Varchar	9				Yes

FK Next of Kin ID → Next of Kin, UPDATE CASCADE, DELETE RESTRICT

tblStudentProgress						
Field name	Data Type	Field Size	Format	Key type	Index	Required
Student Progress ID	smallint	10	Autonumber	Primary	Yes (no duplicates)	Yes
Student ID	Smallint	10		Foreign		Yes
Student full name	Varchar	105				Yes
Number of completed quizzes	Tinyint	3				Yes
Score in quizzes	Tinyint	3				Yes
Average grade	Varchar	2				Yes
Highest score in quiz	Tinyint	3				Yes
Lowest score in quiz	Tinyint	3				Yes
Progress bar	Tinyint	3				Yes

FK Student ID → Student, UPDATE CASCADE, DELETE RESTRICT

tblStaffandStudentProgress						
Field name	Data Type	Field Size	Format	Key type	Index	Required
Staff ID	Smallint	10		foreign		Yes
Student Progress ID	smallint	10		Foreign		Yes

FK Staff ID → Staff, UPDATE CASCADE, DELETE RESTRICT

FK Student progress ID → Student Progress, UPDATE CASCADE, DELETE RESTRICT

tblQuiz						
Field name	Data Type	Field Size	Format	Key type	Index	Required
Quiz ID	smallint	10	Autonumber	Primary	Yes (no duplicates)	Yes
Student progress ID	Smallint	10		Foreign		Yes
Type of quiz	Varchar	40				Yes
Name of quiz	Varchar	40				Yes
Correct answers	Tinyint	3				Yes
Incorrect answers	Tinyint	3				Yes
Date uploaded	DateTime					Yes
Deadline Date	DateTime					Yes

FK Student progress ID → Student Progress, UPDATE CASCADE, DELETE RESTRICT

tblAttendance						
Field name	Data Type	Field Size	Format	Key type	Index	Required
Room ID	Smallint	10	Autonumber	Primary	Yes (no duplicates)	Yes
Student progress ID	Smallint	10		Foreign		Yes
Number of Classes attended	Tinyint	3				Yes
Number of Absences	Tinyint	3				Yes
Number of explained absences	Tinyint	3				Yes
Time	Time					Yes
Date	Date					Yes

FK Student progress ID → Student Progress, UPDATE CASCADE, DELETE RESTRICT

tblResource						
Field name	Data Type	Field Size	Format	Key type	Index	Required
Resource ID	Smallint	10	Autonumber	Primary	Yes (no duplicates)	Yes
Type of resource	Varchar	20				Yes
Date of availability	Date					Yes
Available to who	Varchar	15				Yes

tblResourceuploadedbyStaff						
Field name	Data Type	Field Size	Format	Key type	Index	Required
Staff ID	Smallint	10		Foreign		Yes
Resource ID	Smallint	10		Foreign		Yes

FK Staff ID → Tutor, UPDATE CASCADE, DELETE RESTRICT

FK Resource ID → Resource, UPDATE CASCADE, DELETE RESTRICT

Conclusion

In this section of the report each table has been outlined as to what it will contain also each table should know how to deal with being updated or deleted. In the next section of this report will be starting SQL Implementation.

Chapter four

SQL Implementation

In this section I will be showing you what SQL is needed in order to create a simply database which will be able to store records, store users and select data that has been inputted into the database following the steps that I've put in you can see how my database was created .

Database

In this part we set up the database so store data that will be needed later as you can see by doing these steps I have created my database.

```
Mysql> CREATE DATABASE Acetraining;
```

```
Mysql> SHOW DATABASES;
```

```
Mysql> USE Acetraining;
```


Tables

In this step I will show you what SQL is needed in order to create tables these are important in any database as it allows us to store data also under each table I will be showing what records have been added to the database. For this you will need to access your database as I've called mine "Acetraining" I will be using that database.

```
Mysql> USE Acetraining;
```

```
Mysql> CREATE TABLE tblCourse (
```

```
CourseID smallint auto_increment,  
Course name varchar(25) not null,  
Credit value varchar(25) not null,  
Duration varchar(15) not null,  
Start date Date not null,  
End date Date not null,  
Fee Decimal(3,3) not null,
```

```
Primary key (CourseID)  
);
```

```
INSERT INTO tblCourse (
```

```
(Course ID, Course name, Credit value, Duration, Start date, End  
date, Fee)
```

```
Values
```

```
('10001245', 'Art', '60', Three years', '2016-08-15', '2019-06-30',  
'9.000');
```

```
INSERT INTO tblCourse (
```

```
(Course ID, Course name, Credit value, Duration, Start date, End  
date, Fee)
```

```
Values
```

```
('10001246', 'Computing', '60', 'Three years', '2016-08-20',  
'2019-06-30', '9.000');
```

```
INSERT INTO tblCourse (
```

```
(Course ID, Course name, Credit value, Duration, Start date, End  
date, Fee)
```

```
Values
```

```
('1001246', 'Human Biology', '60', 'Two years', '2016-08-10',  
'2018-03-21', '6.000');
```

```

INSERT INTO tblCourse (

(Course ID, Course name, Credit value, Duration, Start date, End
date, Fee)
Values
('1001247', 'Car Repair', '60', 'Two years', '2016-08-15', '2018-
05-21', '6.000');

INSERT INTO tblCourse (

(Course ID, Course name, Credit value, Duration, Start date, End
date, Fee)
Values
('1001248', 'Biology', '30', 'One years', '2016-08-10', '2017-06-
05', '3.000');

INSERT INTO tblCourse (

(Course ID, Course name, Credit value, Duration, Start date, End
date, Fee)
Values
('1001249', 'Geography', '60', 'Three years', '2016-08-01', '2019-
06-24', '9.000');

INSERT INTO tblCourse (

(Course ID, Course name, Credit value, Duration, Start date, End
date, Fee)
Values
('1001250', 'Games design', '60', 'Three years', '2016-08-05',
'2019-06-30', '9.000');

INSERT INTO tblCourse (

(Course ID, Course name, Credit value, Duration, Start date, End
date, Fee)
Values
('1001251', 'Music', '30', 'Two years', '2016-07-28', '2018-05-15',
'6.000');

```

```
INSERT INTO tblCourse (  
  
  (Course ID, Course name, Credit value, Duration, Start date, End  
  date, Fee)  
Values  
(\1001252', 'English', '60', 'One years', '2016-08-05', '2017-06-  
10', '3.000');
```

```
INSERT INTO tblCourse (  
  
  (Course ID, Course name, Credit value, Duration, Start date, End  
  date, Fee)  
Values  
(\1001253', 'Law', '60', 'Two years', '2016-08-19', '2018-04-14',  
'6.000');
```

```
INSERT INTO tblCourse (  
  
  (Course ID, Course name, Credit value, Duration, Start date, End  
  date, Fee)  
Values  
(\1001254', 'Software design', '60', 'One years', '2016-08-20',  
'2017-06-17', '3.000');
```

```
Mysql> CREATE TABLE tblStudentOnCourse (  
  
  CourseID smallint not null,  
  StudentID smallint not null,  
  
  Foreign key (CourseID) references tblCourse(CourseID) ON UPDATE  
  CASCADE ON DELETE RESTRICT,  
  Foreign key (StudentID) references tblStudent(StudentID) ON UPDATE  
  CASCADE ON DELETE RESTRICT  
);
```

```
INSERT INTO tblStudentOnCourse (  
  
  Course ID, Student ID)  
VALUES  
(\10001245', '1000001');
```

```
INSERT INTO tblStudentOnCourse (  
  
  Course ID, Student ID)  
VALUES  
(\10001246', '1000002');
```

```
INSERT INTO tblStudentOnCourse (  
Course ID, Student ID)  
VALUES  
( '10001247', '1000003' );  
  
INSERT INTO tblStudentOnCourse (  
Course ID, Student ID)  
VALUES  
( '10001248', '1000004' );  
  
INSERT INTO tblStudentOnCourse (  
Course ID, Student ID)  
VALUES  
( '10001249', '1000005' );  
  
INSERT INTO tblStudentOnCourse (  
Course ID, Student ID)  
VALUES  
( '10001250', '1000006' );  
  
INSERT INTO tblStudentOnCourse (  
Course ID, Student ID)  
VALUES  
( '10001251', '1000007' );  
  
INSERT INTO tblStudentOnCourse (  
Course ID, Student ID)  
VALUES  
( '10001252', '1000008' );  
  
INSERT INTO tblStudentOnCourse (  
Course ID, Student ID)  
VALUES  
( '10001253', '1000009' );  
  
INSERT INTO tblStudentOnCourse (  
Course ID, Student ID)  
VALUES  
( '10001254', '1000010' );
```

```
Mysql> CREATE TABLE tblStaffOnCourse (  
  
CourseID smallint not null,  
StaffID smallint not null,  
  
Foreign key (CourseID) references tblCourse(CourseID) ON UPDATE  
CASCADE ON DELETE RESTRICT,  
Foreign key (StaffID) references tblStaff(StaffID) ON UPDATE  
CASCADE ON DELETE RESTRICT  
);  
  
INSERT INTO tblStudentOnCourse (  
  
Course ID, Staff ID)  
VALUES  
( '10001245', '1200001' );  
  
INSERT INTO tblStudentOnCourse (  
  
Course ID, Staff ID)  
VALUES  
( '10001246', '1200002' );  
  
INSERT INTO tblStudentOnCourse (  
  
Course ID, Staff ID)  
VALUES  
( '10001247', '1200003' );  
  
INSERT INTO tblStudentOnCourse (  
  
Course ID, Staff ID)  
VALUES  
( '10001248', '1200004' );  
  
INSERT INTO tblStudentOnCourse (  
  
Course ID, Staff ID)  
VALUES  
( '10001249', '1200005' );  
  
INSERT INTO tblStudentOnCourse (  
  
Course ID, Staff ID)  
VALUES  
( '10001250', '1200006' );
```

```
INSERT INTO tblStudentOnCourse (  
Course ID, Staff ID)  
VALUES  
( '10001251', '1200007');
```

```
INSERT INTO tblStudentOnCourse (  
Course ID, Staff ID)  
VALUES  
( '10001252', '1200008');
```

```
INSERT INTO tblStudentOnCourse (  
Course ID, Staff ID)  
VALUES  
( '10001253', '1200009');
```

```
INSERT INTO tblStudentOnCourse (  
Course ID, Staff ID)  
VALUES  
( '10001254', '1200010');
```

```
Mysql> CREATE TABLE tblStudent (  
  
StudentID smallint auto_increment,  
NextofKinID smallint not null,  
First name varchar(35) not null,  
Middle name varchar(35) null,  
Last name varchar(35) not null,  
DOB Date not null,  
AddressL1 varchar(20) not null,  
AddressL2 varchar(20) not null,  
County varchar(15) not null,  
Postcode varchar(7) not null,  
Email varchar(30) null,  
Phone number varchar(15) not null,  
Visa expiry Date null,  
Passport number tinyint null unique,  
  
Primary key (StudentID),  
Foreign key (CourseID) references tblCourse(CourseID) ON UPDATE
```

```
CASCADE ON DELETE RESTRICT
);
```

```
INSERT INTO tblStudent (
Student ID, Next of kin ID, First name, Middle name, Last name,
Date of birth, Address one, Address two, County, Postal code, E-
mail address, Phone number, Date enrolled, Visa expiry date,
Passport number)
Values
('1000001', '11010001', 'Peter', 'James', 'Parker', '1981-06-17',
'46 Oak Lane', 'Grimby', 'Lincolnshire', 'DN32 9BT',
'PeterPan@hotmail.com', '01204 836452' '2016-06-15', 'NULL',
'NULL');
```

```
INSERT INTO tblStudent (
Student ID, Next of kin ID, First name, Middle name, Last name,
Date of birth, Address one, Address two, County, Postal code, E-
mail address, Phone number, Date enrolled, Visa expiry date,
Passport number)
Values
('1000002', '11010002', 'Ben', 'NULL', 'Car', '1974-05-23', '3
Dulwich Street', 'Seaford', 'East Sussex', 'BN25 3PH',
'Ben1974@hotmail.com', '01204 273645' , '2016-06-10', 'NULL',
'NULL');
```

```
INSERT INTO tblStudent (
Student ID, Next of kin ID, First name, Middle name, Last name,
Date of birth, Address one, Address two, County, Postal code, E-
mail address, Phone number, Date enrolled, Visa expiry date,
Passport number)
Values
('1000003', '11010003', 'Harry', 'NULL', 'Pot', '1991-01-13', '45
St John Street', 'NULL', 'London', 'EC1M 4LX', 'Potter@gmail.com',
'01204 183746', '2016-06-01', 'NULL', 'NULL');
```

```
INSERT INTO tblStudent (
Student ID, Next of kin ID, First name, Middle name, Last name,
Date of birth, Address one, Address two, County, Postal code, E-
mail address, Phone number, Date enrolled, Visa expiry date,
Passport number)
```

```
Values
('1000004', '11010004', 'Daniel', 'David', 'McHugh', '1994-06-
30', '17 Church Hill', 'Wolverhampton', 'West Midlands', 'WV4 5NP',
'NULL', '01204 847563', '2016-06-05', 'NULL', 'NULL');
```

```
INSERT INTO tblStudent (
```

```
Student ID, Next of kin ID, First name, Middle name, Last name,
Date of birth, Address one, Address two, County, Postal code, E-
mail address, Phone number, Date enrolled, Visa expiry date,
Passport number)
```

```
Values
('1000005', '11010005', 'John', 'NULL', 'Lewis', '1981-06-10', '2
Woburn View', 'Millisle', 'Newtownards', 'BT22 2FH',
'Lewis81@hotmail.com', '01204 937465', '2016-06-10', 'NULL',
'NULL');
```

```
INSERT INTO tblStudent (
```

```
Student ID, Next of kin ID, First name, Middle name, Last name,
Date of birth, Address one, Address two, County, Postal code, E-
mail address, Phone number, Date enrolled, Visa expiry date,
Passport number)
```

```
Values
('1000006', '11010006', 'Derik', 'NULL', 'Whole', '1991-02-
26', '21 Zoo Lane', 'NULL', 'Jordan', '67JN 77F',
'DerikZoo@hotmail.com', '01204 905967', '2016-06-15', '2018-07-12',
'12326751');
```

```
INSERT INTO tblStudent (
```

```
Student ID, Next of kin ID, First name, Middle name, Last name,
Date of birth, Address one, Address two, County, Postal code, E-
mail address, Phone number, Date enrolled, Visa expiry date,
Passport number)
```

```
Values
('1000007', '11010007', 'Ali', 'NULL', 'Zanzibar', '1993-02-
10', '10 River Lane', 'NULL', 'Bambina', 'HU76 1W2', 'NULL',
'01204 012745', '2016-05-25', '2018-02-10', '14792647');
```

```
INSERT INTO tblStudent (
```

```
Student ID, Next of kin ID, First name, Middle name, Last name,
Date of birth, Address one, Address two, County, Postal code, E-
```



```

mail address, Phone number, Date enrolled, Visa expiry date,
Passport number)
Values
('1000008', '11010008', 'Ron', 'Butt', 'Hair', '1994-04-12', '76
London Road', 'Bagshot', 'Surrey', 'GU19 5BT', 'Ron95@gmail.com',
'01204 186700', '2016-06-05', 'NULL', 'NULL');

INSERT INTO tblStudent (

Student ID, Next of kin ID, First name, Middle name, Last name,
Date of birth, Address one, Address two, County, Postal code, E-
mail address, Phone number, Date enrolled, Visa expiry date,
Passport number)
Values
('1000009', '11010009', 'Amy', 'NULL', 'Hot', '1984-06-27', '40
Bynner Street', 'Shrewbury', 'Shropshire', 'SY3 7NZ',
'AmyHot84@gmail.com', '01204 098123', '2016-06-24', 'NULL',
'NULL');

INSERT INTO tblStudent (

Student ID, Next of kin ID, First name, Middle name, Last name,
Date of birth, Address one, Address two, County, Postal code, E-
mail address, Phone number, Date enrolled, Visa expiry date,
Passport number)
Values
('1000010', '11010010', 'Kate', 'ford', 'North', '1994-09-25', '35
Charlton Lane', 'Shepperton', 'Surrey', 'Tw17 8QB',
'KNorth@gmail.com', '01204 295760', '2016-06-29', 'NULL',
'12475967');

Mysql> CREATE TABLE tblNextofKin (

NextofKinID smallint auto_increment,
First name varchar(35) not null,
Middle name varchar(35) not null,
Last name varchar(35) not null,
DOB Date not null,
AddressL1 varchar(20) not null,
AddressL2 varchar(20) null,
County varchar(15) not null,
Postcode varchar(7) not null,
Phone varchar(15) not null,

```

```
Email varchar(30) null,  
Relation to Person varchar(6) not null,
```

```
Primary key (NextofKinID)  
);
```

```
INSERT INTO tblNextofKin (
```

```
Next of kin ID, First name, Middle name, Last name, Date of birth,  
Address one, Address two, County, Postcode, Phone number, E-mail  
address, Relation to Person)
```

```
VALUES
```

```
('11010001', 'John', 'Jam', 'Parker', '1941-06-17', '46 Oak Lane',  
'Grimby', 'Lincolnshire', 'DN32 9BT', '01204 836452',  
'Jam@hotmail.com', 'Father');
```

```
INSERT INTO tblNextofKin (
```

```
Next of kin ID, First name, Middle name, Last name, Date of birth,  
Address one, Address two, County, Postcode, Phone number, E-mail  
address, Relation to Person)
```

```
VALUES
```

```
(11010002', 'Sue', 'NULL', 'Car', '1954-05-23', '3 Dulwich Street',  
'Seaford', 'East Sussex', 'BN25 3PH', '01204 273645', 'NULL',  
'Mother');
```

```
INSERT INTO tblNextofKin (
```

```
Next of kin ID, First name, Middle name, Last name, Date of birth,  
Address one, Address two, County, Postcode, Phone number, E-mail  
address, Relation to Person)
```

```
VALUES
```

```
('11010003', 'James', 'Harry', 'Pot', '1971-03-12', '45 St John  
Street', 'NULL', 'London', 'EC1M 4LX', '01204 183746',  
'JamesPot@gmail.com', 'Father');
```

```
INSERT INTO tblNextofKin (
```

```
Next of kin ID, First name, Middle name, Last name, Date of birth,  
Address one, Address two, County, Postcode, Phone number, E-mail  
address, Relation to Person)
```

```
VALUES
```

```
('11010004', 'David', 'NULL', 'McHugh', '1971-01-12', '17 Church  
Hill', 'Wolverhampton', 'West Midlands', 'WV4 5NP', '01204 847563',  
'DmcHugh71@hotmail.com', 'Father');
```

```

INSERT INTO tblNextofKin (
Next of kin ID, First name, Middle name, Last name, Date of birth,
Address one, Address two, County, Postcode, Phone number, E-mail
address, Relation to Person)
VALUES
('11010005', 'Jim', 'Fart', 'Lewis', '1951-05-18', '2 Woburn View',
'Millisle', 'Newtownards', 'BT22 2FH', '01204 937465', 'NULL',
'Father');

```

```

INSERT INTO tblNextofKin (
Next of kin ID, First name, Middle name, Last name, Date of birth,
Address one, Address two, County, Postcode, Phone number, E-mail
address, Relation to Person)
VALUES
('11010006', 'Fanny', 'NULL', 'Whole', '1961-02-11', '21 Zoo Lane',
'NULL', 'Jordan', '67JN 77F', '01204 905967', 'Fwhole@hotmail.com',
'Mother');

```

```

INSERT INTO tblNextofKin (
Next of kin ID, First name, Middle name, Last name, Date of birth,
Address one, Address two, County, Postcode, Phone number, E-mail
address, Relation to Person)
VALUES
('11010007', 'Charlotte', 'Light', 'Zanzibar', '1961-09-17', '10
River Lane', 'NULL', 'Bambina', 'HU76 1W2', '01204 012745',
'LightPotato@gmail.com', 'Mother');

```

```

INSERT INTO tblNextofKin (
Next of kin ID, First name, Middle name, Last name, Date of birth,
Address one, Address two, County, Postcode, Phone number, E-mail
address, Relation to Person)
VALUES
('11010008', 'Fred', 'Daniel', 'Hair', '1961-12-22', '76 London
Road', 'Bagshot', 'Surrey', 'GU19 5BT', '01204 186700',
'Fred61@hotmailmail.com', 'Father');

```

```

INSERT INTO tblNextofKin (
Next of kin ID, First name, Middle name, Last name, Date of birth,
Address one, Address two, County, Postcode, Phone number, E-mail
address, Relation to Person)

```

```

VALUES
('11010009', 'Luke', 'NULL', 'Hot', '1960-02-30', '40 Bynner
Street', 'Shrewbury', 'Shropshire', 'SY3 7NZ', '01204 098123',
'NULL', 'Father');

INSERT INTO tblNextofKin (

Next of kin ID, First name, Middle name, Last name, Date of birth,
Address one, Address two, County, Postcode, Phone number, E-mail
address, Relation to Person)
VALUES
('11010010', 'South', 'East', 'North', '1977-06-23', '35 Charlton
Lane', 'Shepperton', 'Surrey', 'Tw17 8QB', '01204 295760',
'SWE77@gmail.com', 'Mother');

INSERT INTO tblNextofKin (

Next of kin ID, First name, Middle name, Last name, Date of birth,
Address one, Address two, County, Postcode, Phone number, E-mail
address, Relation to Person)
VALUES
('11010011', 'Marvin', 'Chernobog', 'O Fionnagain', '1961-04-27',
'9 Osier Grove', 'Birmingham', 'West Midlands', 'B23 7XU', '01204
716254', 'NULL', 'Partner');

INSERT INTO tblNextofKin (

Next of kin ID, First name, Middle name, Last name, Date of birth,
Address one, Address two, County, Postcode, Phone number, E-mail
address, Relation to Person)
VALUES
('11010012', 'Mark', 'NULL', 'Milovan', '1971-05-23', '48
Greatfields Road', 'Barking', 'Greater London', 'IG11 7TZ', '01204
857694', 'NULL', 'Partner');

INSERT INTO tblNextofKin (

Next of kin ID, First name, Middle name, Last name, Date of birth,
Address one, Address two, County, Postcode, Phone number, E-mail
address, Relation to Person)
VALUES
('11010013', 'Darma', 'Jimmy', 'Romero', '1971-10-01', '4 Sanworth
Street', 'Todmorden', 'West Yorkshire', 'OL14 5BU', '01204 294857',
'Jimmy71@gmail.com', 'Partner');

```

```

INSERT INTO tblNextofKin (
Next of kin ID, First name, Middle name, Last name, Date of birth,
Address one, Address two, County, Postcode, Phone number, E-mail
address, Relation to Person)
VALUES
('11010014', 'Khaqa', 'Mess', 'Rask', '1978-02-30', '16 Aberdeen
Walk', 'Scarborough', 'North Yorkshire', 'YO11 1XP', '01204
347135', 'NULL', 'Partner');

INSERT INTO tblNextofKin (
Next of kin ID, First name, Middle name, Last name, Date of birth,
Address one, Address two, County, Postcode, Phone number, E-mail
address, Relation to Person)
VALUES
('11010015', 'Neer', 'Lilija', 'Sanford', '1969-04-13', '10
Broadway', 'Grays', 'Essex', 'RM17 6EW', '01204 294856',
'Neer69@hotmail.com', 'Partner');

INSERT INTO tblNextofKin (
Next of kin ID, First name, Middle name, Last name, Date of birth,
Address one, Address two, County, Postcode, Phone number, E-mail
address, Relation to Person)
VALUES
('11010016', 'Emil', 'Raguel', 'Venalainen', '1978-01-29', '110
Chesterfield Road', 'Bristol', 'City of Bristol', 'BS6 5DR',
'01204 194624', 'Raguel78@gmail.com', 'Partner');

INSERT INTO tblNextofKin (
Next of kin ID, First name, Middle name, Last name, Date of birth,
Address one, Address two, County, Postcode, Phone number, E-mail
address, Relation to Person)
VALUES
('11010017', 'Aldo', 'NULL', 'Alberici', '1975-08-22', '59 Knole
Lane', 'Bristol', 'City of Bristol', 'BS10 6SE', '01204 385987',
'NULL', 'Partner');

INSERT INTO tblNextofKin (
Next of kin ID, First name, Middle name, Last name, Date of birth,
Address one, Address two, County, Postcode, Phone number, E-mail
address, Relation to Person)

```

```

VALUES
('11010018', 'Leon', 'NULL', 'Amin', '1977-01-30', '12 Kellys
Lane', 'Dungannon', 'Dungannon and South Tyrone', 'BT70 3SU',
'01204 374659', 'KLeon77@hotmail.com', 'Partner');

INSERT INTO tblNextofKin (

Next of kin ID, First name, Middle name, Last name, Date of birth,
Address one, Address two, County, Postcode, Phone number, E-mail
address, Relation to Person)
VALUES
('11010019', 'Anna', 'Heru', 'Davies', '1978-12-25', '20 Monroe
Street', 'Warrington', 'Warrington', 'WA1 4LW', '01204 283746',
'ADavies@gmail.com', 'Partner');

INSERT INTO tblNextofKin (

Next of kin ID, First name, Middle name, Last name, Date of birth,
Address one, Address two, County, Postcode, Phone number, E-mail
address, Relation to Person)
VALUES
('11010020', 'Jesus', 'Lord', 'Christ', '1970-11-01', '48 Woodside,
Castleford, 'West Yorkshire', 'WF10 3RT', '01204 293164',
'Christlord@gmail.com', 'Partner');

Mysql> CREATE TABLE tblStaff (

StaffID smallint auto_increment,
NextofKinID smallint not null,
First name varchar(35) not null,
Middle name varchar(35) not null,
Last name varchar(35) not null,
DOB Date not null,
AddressL1 varchar(20) not null,
AddressL2 varchar(20) null,
County varchar(15) not null,
Postcode varchar(7) not null,
Email varchar(30) null,
Office number varchar(10) not null
Phone varchar(15) not null,
National insurance number varchar(9) not null,

Primary key (StaffID),
Foreign key (NextofKinID) references tblNextofKin(NextofKinID) ON

```

```
UPDATE CASCADE ON DELETE RESTRICT
);
```

```
INSERT INTO tblStaff (
StaffID, Next of kin ID, First name, Middle name, Last name, Date
of birth, Address one, Address two, County, Postcode, E-mail
address, Office number, Phone number, National insurance number)
VALUES
('1200001', '11010011', 'Devraj', 'Maddison', 'O Fionnagain',
'1965-02-10', '9 Osier Grove', 'Birmingham', 'West Midlands', 'B23
7XU', 'Devraj@hotmail.com', 'XMQ 134', '01204 716254',
'U8269532G');
```

```
INSERT INTO tblStaff (
StaffID, Next of kin ID, First name, Middle name, Last name, Date
of birth, Address one, Address two, County, Postcode, E-mail
address, Office number, Phone number, National insurance number)
VALUES
('1200002', '11010012', 'Khazhak', 'Karen', 'Milovan', '1975-07-
29', '48 Greatfields Road', 'Barking', 'Greater London', 'IG11
7TZ', 'Khazhak@gmail.com', 'RTX 220', '01204 857694',
'K10485620');
```

```
INSERT INTO tblStaff (
StaffID, Next of kin ID, First name, Middle name, Last name, Date
of birth, Address one, Address two, County, Postcode, E-mail
address, Office number, Phone number, National insurance number)
VALUES
('1200003', '11010013', 'Evangelos', 'NULL', 'Romero', '1973-08-
15', '4 Sanworth Street', 'Todmorden', 'West Yorkshire', 'OL14
5BU', 'NULL', 'RTX 404', '01204 294857', 'P0948576Q');
```

```
INSERT INTO tblStaff (
StaffID, Next of kin ID, First name, Middle name, Last name, Date
of birth, Address one, Address two, County, Postcode, E-mail
address, Office number, Phone number, National insurance number)
VALUES
('1200004', '11010014', 'Chananyah', 'Albina', 'Rask', '1980-01-
01', '16 Aberdeen Walk', 'Scarborough', 'North Yorkshire', 'YO11
```

```
1XP', 'Chan80@hotmail.com', 'XMQ 222', '01204 347135',  
'A1048567T');
```

```
INSERT INTO tblStaff (
```

```
StaffID, Next of kin ID, First name, Middle name, Last name, Date  
of birth, Address one, Address two, County, Postcode, E-mail  
address, Office number, Phone number, National insurance number)  
VALUES
```

```
('1200005', '11010015', 'Viola', 'Artur', 'Sanford', '1969-02-22',  
'10 Broadway', 'Grays', 'Essex', 'RM17 6EW', 'Viola69@gmail.com',  
'AWU 110', '01204 294856', 'N0031846T');
```

```
INSERT INTO tblStaff (
```

```
StaffID, Next of kin ID, First name, Middle name, Last name, Date  
of birth, Address one, Address two, County, Postcode, E-mail  
address, Office number, Phone number, National insurance number)  
VALUES
```

```
('1200006', '11010016', 'Nazih', 'Phoebe', 'Venalainen', '1975-07-  
11', '110 Chesterfield Road', 'Bristol', 'City of Bristol', 'BS6  
5DR', 'PhoebeNaz@hotmail.com', 'QOE 110', '01204 194624',  
'P0173547J');
```

```
INSERT INTO tblStaff (
```

```
StaffID, Next of kin ID, First name, Middle name, Last name, Date  
of birth, Address one, Address two, County, Postcode, E-mail  
address, Office number, Phone number, National insurance number)  
VALUES
```

```
('1200007', '11010017', 'Regina', 'NULL', 'Alberici', '1980-11-  
21', '59 Knole Lane', 'Bristol', 'City of Bristol', 'BS10 6SE',  
'NULL', 'XMQ 134', '01204 385987', 'L1284652Z');
```

```
INSERT INTO tblStaff (
```

```
StaffID, Next of kin ID, First name, Middle name, Last name, Date  
of birth, Address one, Address two, County, Postcode, E-mail  
address, Office number, Phone number, National insurance number)  
VALUES
```

```
('1200008', '11010018', 'Gordan', 'Diodotos', 'Amin', '1972-12-19',  
'12 Kellys Lane', 'Dungannon', 'Dungannon and South Tyrone', 'BT70  
3SU', 'Gordan72@gmail.com', 'XMQ 404', '01204 374659',  
'X1678302S');
```



```

INSERT INTO tblStaff (
StaffID, Next of kin ID, First name, Middle name, Last name, Date
of birth, Address one, Address two, County, Postcode, E-mail
address, Office number, Phone number, National insurance number)
VALUES
('1200009', '11010019', 'Aibek', 'Itri', 'Davies', '1980-07-12',
'20 Monroe Street', 'Warrington', 'Warrington', 'WA1 4LW',
'AibekDav@gmail.com', 'RTX 009', '01204 283746', 'V2782597X');

```

```

INSERT INTO tblStaff (
StaffID, Next of kin ID, First name, Middle name, Last name, Date
of birth, Address one, Address two, County, Postcode, E-mail
address, Office number, Phone number, National insurance number)
VALUES
('1200010', '11010020', 'Mary', 'NULL', 'Christ', '1960-12-30',
'48 Woodside', 'Castleford', 'West Yorkshire', 'WF10 3RT', 'NULL',
'QSS 001', '01204 293164', 'A2349680P');

```

```

Mysql> CREATE TABLE tblStudentProgress (
StudentProgressID smallint auto_increment,
Student ID smallint(10) not null,
Student full name varchar(105) not null,
Number of completed quizzes tinyint(3) not null,
Score in quizzes tinyint(3) not null,
Average grade varchar(2) not null,
Highest score in quiz tinyint(3) not null,
Lowest score in quiz tinyint(3) not null,
Progress bar tinyint(3) not null,
Primary key (StudentProgressID)
Foreign key (StudentID) references tblStudent(StudentID) ON UPDATE
CASCADE ON DELETE RESTRICT
);

```

```

INSERT INTO tblStudentProgress (

StudentProgressID, StudentID, Student full Name, Number of
completed quizzes, Score in quizzes, Average grade, Highest score
in quiz, Lowest score in quiz, Progress bar)
VALUES
('1011111', '1000001', 'Peter James Parker', '05', '106', 'D-',
'75', '11', '015');

INSERT INTO tblStudentProgress (

StudentProgressID, StudentID, Student full Name, Number of
completed quizzes, Score in quizzes, Average grade, Highest score
in quiz, Lowest score in quiz, Progress bar)
VALUES
('1011112', '1000002', 'Ben NULL Car', '09', '682', 'C+', '100',
'36', '066');

INSERT INTO tblStudentProgress (

StudentProgressID, StudentID, Student full Name, Number of
completed quizzes, Score in quizzes, Average grade, Highest score
in quiz, Lowest score in quiz, Progress bar)
VALUES
('1011113', '1000003', 'Harry NULL Pot', '04', '290', 'D+', '100',
'0', '025');

INSERT INTO tblStudentProgress (

StudentProgressID, StudentID, Student full Name, Number of
completed quizzes, Score in quizzes, Average grade, Highest score
in quiz, Lowest score in quiz, Progress bar)
VALUES
('1011114', '1000004', 'Daniel David McHugh', '06', '560', 'B+',
'100', '59', '056');

INSERT INTO tblStudentProgress (

StudentProgressID, StudentID, Student full Name, Number of
completed quizzes, Score in quizzes, Average grade, Highest score
in quiz, Lowest score in quiz, Progress bar)
VALUES
('1011115', '1000005', 'John NULL Lewis', '07', '100', 'E-', '30',
'10', '010');

```

```

INSERT INTO tblStudentProgress (

StudentProgressID, StudentID, Student full Name, Number of
completed quizzes, Score in quizzes, Average grade, Highest score
in quiz, Lowest score in quiz, Progress bar)
VALUES
('1011116', '1000006', 'Derik NULL Whole', '02', '150', 'E-',
'100', '50', '005');

INSERT INTO tblStudentProgress (

StudentProgressID, StudentID, Student full Name, Number of
completed quizzes, Score in quizzes, Average grade, Highest score
in quiz, Lowest score in quiz, Progress bar)
VALUES
('1011117', '1000007', 'Ali NULL Zanzibar', '04', '550', 'B+',
'100', '25', '045');

INSERT INTO tblStudentProgress (

StudentProgressID, StudentID, Student full Name, Number of
completed quizzes, Score in quizzes, Average grade, Highest score
in quiz, Lowest score in quiz, Progress bar)
VALUES
('1011118', '1000008', 'Ron Butt Hair', '01', '50', 'F-', '50',
'0', '001');

INSERT INTO tblStudentProgress (

StudentProgressID, StudentID, Student full Name, Number of
completed quizzes, Score in quizzes, Average grade, Highest score
in quiz, Lowest score in quiz, Progress bar)
VALUES
('1011119', '1000009', 'Amy NULL Hot', '08', '770', 'A-', '100',
'70', '069');

INSERT INTO tblStudentProgress (

StudentProgressID, StudentID, Student full Name, Number of
completed quizzes, Score in quizzes, Average grade, Highest score
in quiz, Lowest score in quiz, Progress bar)
VALUES
('1011120', '1000010', 'Kate ford North', '05', '499', 'D-', '57',
'10', '034')

```

```

Mysql> CREATE TABLE tblStaffandStudentProgress (

StaffID smallint not null,
StudentProgressID smallint not null,
Foreign key (StaffID) references tblStaff(StaffID) ON UPDATE
CASCADE ON DELETE RESTRICT,
Foreign key (StudentProgressID) references
tblStudentProgress(StudentProgressID) ON UPDATE CASCADE ON DELETE
RESTRICT
);

INSERT INTO tblStaffandStudentProgress (

StaffID, StudentProgressID)
VALUES
('1200001', '1011111');

INSERT INTO tblStaffandStudentProgress (

StaffID, StudentProgressID)
VALUES
('1200002', '1011112');

INSERT INTO tblStaffandStudentProgress (

StaffID, StudentProgressID)
VALUES
('1200003', '1011113');

INSERT INTO tblStaffandStudentProgress (

StaffID, StudentProgressID)
VALUES
('1200004', '1011114');

INSERT INTO tblStaffandStudentProgress (

StaffID, StudentProgressID)
VALUES
('1200005', '1011115');

INSERT INTO tblStaffandStudentProgress (

StaffID, StudentProgressID)
VALUES
('1200006', '1011116');

```

```
INSERT INTO tblStaffandStudentProgress (  
StaffID, StudentProgressID)  
VALUES  
(\1200007', \1011117');
```

```
INSERT INTO tblStaffandStudentProgress (  
StaffID, StudentProgressID)  
VALUES  
(\1200008', \1011118');
```

```
INSERT INTO tblStaffandStudentProgress (  
StaffID, StudentProgressID)  
VALUES  
(\1200009', \1011119');
```

```
INSERT INTO tblStaffandStudentProgress (  
StaffID, StudentProgressID)  
VALUES  
(\1200010', \1011120');
```

```
Mysql> CREATE TABLE tblQuiz (  
  
QuizID smallint auto_increment,  
StudentProgressID smallint not null,  
Type of quiz varchar(25) not null,  
Name of quiz varchar(40) not null,  
Correct answers tinyint(3) not null,  
Incorrect answers tinyint(3) not null,  
Date uploaded DateTime not null,  
Deadline Date DateTime not null,  
  
Primary key (QuizID),  
Foreign key (StudentProgressID) references  
tblStudentProgress(StudentProgressID) ON UPDATE CASCADE ON DELETE  
RESTRICT  
);
```

```
INSERT INTO tblQuiz (  
  
Quiz ID, StudentProgressID, Type of quiz, Name of quiz, Number of  
quizzes, Correct answers, Incorrect answers, Date uploaded,  
Deadline date)  
VALUES  
(\1300001', \1011111' 'True or False', 'Can you read', \01', \100',  
\0' 2016-09-22', \2016-09-28');
```

```
INSERT INTO tblQuiz (  
  
Quiz ID, StudentProgressID, Type of quiz, Name of quiz, Number of  
quizzes, Correct answers, Incorrect answers, Date uploaded,  
Deadline date)  
VALUES  
(\1300002', \1011112' 'True or False', 'ABC for adults', \01',  
\65', \35' 2016-08-01', \2016-08-07');
```

```
INSERT INTO tblQuiz (  
  
Quiz ID, StudentProgressID, Type of quiz, Name of quiz, Number of  
quizzes, Correct answers, Incorrect answers, Date uploaded,  
Deadline date)  
VALUES  
(\1300003', \1011113' 'Fill in the Blank', 'Know your stuff', \01',  
\50', \50' 2016-07-11', \2016-07-15');
```

```
INSERT INTO tblQuiz (  
  
Quiz ID, StudentProgressID, Type of quiz, Name of quiz, Number of  
quizzes, Correct answers, Incorrect answers, Date uploaded,  
Deadline date)  
VALUES  
(\1300004', \1011114' 'Multiple choice', 'Choice your path', \01',  
\75', \25' 2016-10-10', \2016-10-19');
```

```
INSERT INTO tblQuiz (  
  
Quiz ID, StudentProgressID, Type of quiz, Name of quiz, Number of  
quizzes, Correct answers, Incorrect answers, Date uploaded,  
Deadline date)  
VALUES  
(\1300005', \1011115' 'Fill in the blank', 'Guess the word', \01',  
\25', \75' 2016-11-01', \2016-11-05');
```

```

INSERT INTO tblQuiz (

Quiz ID, StudentProgressID, Type of quiz, Name of quiz, Number of
quizzes, Correct answers, Incorrect answers, Date uploaded,
Deadline date)
VALUES
('1300006', '1011116' 'True or False', 'Read your books', '01',
'63', '37' 2016-07-01', '2016-07-07');

INSERT INTO tblQuiz (

Quiz ID, StudentProgressID, Type of quiz, Name of quiz, Number of
quizzes, Correct answers, Incorrect answers, Date uploaded,
Deadline date)
VALUES
('1300007', '1011117' 'multiple choice', 'Are Databases fun', '01',
'85', '15' 2016-12-25', '2016-12-29');

INSERT INTO tblQuiz (

Quiz ID, StudentProgressID, Type of quiz, Name of quiz, Number of
quizzes, Correct answers, Incorrect answers, Date uploaded,
Deadline date)
VALUES
('1300008', '1011118' 'True or False', 'Jobs you want', '01', '76',
'24' 2016-10-15', '2016-10-24');

INSERT INTO tblQuiz (

Quiz ID, StudentProgressID, Type of quiz, Name of quiz, Number of
quizzes, Correct answers, Incorrect answers, Date uploaded,
Deadline date)
VALUES
('1300009', '1011119' 'multiple choice', 'Best courses', '01', '0',
'100' 2016-08-09', '2016-08-10');

INSERT INTO tblQuiz (

Quiz ID, StudentProgressID, Type of quiz, Name of quiz, Number of
quizzes, Correct answers, Incorrect answers, Date uploaded,
Deadline date)
VALUES
('1300010', '1011120' 'Fill in the blank', 'Cheese or wine', '01',
'56', '44' 2017-03-12', '2017-03-15');

```

```

Mysql> CREATE TABLE tblAttendance (

RoomID smallint auto_increment,
StudentProgressID smallint not null,
Number of classes attended tinyint(3) not null,
Number of Absences tinyint(3) not null,
Number of explained absences tinyint(3) not null,

Primary key (RoomID),
Foreign key (StudentProgressID) references tblStudent
Progress(StudentProgressID) ON UPDATE CASCADE ON DELETE RESTRICT
);

INSERT INTO tblAttendance (

RoomID, StudentProgressID, Number of Classes attended, Number of
absences, Number of explained absences)
VALUES
('1400001', '1011111', '40', '40', '20');

INSERT INTO tblAttendance (

RoomID, StudentProgressID, Number of Classes attended, Number of
absences, Number of explained absences)
VALUES
('1400002', '1011112', '60', '30', '10');

INSERT INTO tblAttendance (

RoomID, StudentProgressID, Number of Classes attended, Number of
absences, Number of explained absences)
VALUES
('1400003', '1011113', '5', '95', '5');

INSERT INTO tblAttendance (

RoomID, StudentProgressID, Number of Classes attended, Number of
absences, Number of explained absences)
VALUES
('1400004', '1011114', '100', '0', '0');

INSERT INTO tblAttendance (

RoomID, StudentProgressID, Number of Classes attended, Number of
absences, Number of explained absences)

```



```

VALUES
('1400005', '1011115', '75', '0', '25');

INSERT INTO tblAttendance (

RoomID, StudentProgressID, Number of Classes attended, Number of
absences, Number of explained absences)
VALUES
('1400006', '1011116', '0', '100', '0');

INSERT INTO tblAttendance (

RoomID, StudentProgressID, Number of Classes attended, Number of
absences, Number of explained absences)
VALUES
('1400007', '1011117', '40', '60', '0');

INSERT INTO tblAttendance (

RoomID, StudentProgressID, Number of Classes attended, Number of
absences, Number of explained absences)
VALUES
('1400008', '1011118', '90', '0', '10');

INSERT INTO tblAttendance (

RoomID, StudentProgressID, Number of Classes attended, Number of
absences, Number of explained absences)
VALUES
('1400009', '1011119', '65', '10', '25');

INSERT INTO tblAttendance (

RoomID, StudentProgressID, Number of Classes attended, Number of
absences, Number of explained absences)
VALUES
('1400010', '1011120', '100', '0', '0');

Mysql> CREATE TABLE tblResource (

ResourceID smallint auto_increment,
Type of resource varchar(20) not null,
Date of availability Date not null,
Available to who varchar(15) not null,

```

```

Primary key (ResourceID)
);

INSERT INTO tblResource (
ResourceID, Type of resource, Date of availability, Available to
who)
VALUES
('1500001', 'PowerPoint', '2016-07-22', 'Everyone');

INSERT INTO tblResource (
ResourceID, Type of resource, Date of availability, Available to
who)
VALUES
('1500002', 'Text file', '2017-01-05', 'Tutors');

INSERT INTO tblResource (
ResourceID, Type of resource, Date of availability, Available to
who)
VALUES
('1500003', 'PowerPoint', '2016-09-05', 'Everyone');

INSERT INTO tblResource (
ResourceID, Type of resource, Date of availability, Available to
who)
VALUES
('1500004', 'Word', '2016-08-10', 'Tutors');

INSERT INTO tblResource (
ResourceID, Type of resource, Date of availability, Available to
who)
VALUES
('1500005', 'PDF', '2016-07-01', 'Everyone');

INSERT INTO tblResource (
ResourceID, Type of resource, Date of availability, Available to
who)
VALUES
('1500006', 'PDF', '2017-02-05', 'Everyone');

```

```

INSERT INTO tblResource (
ResourceID, Type of resource, Date of availability, Available to
who)
VALUES
('1500007', 'Excel', '2017-01-01', 'Tutors');

```

```

INSERT INTO tblResource (
ResourceID, Type of resource, Date of availability, Available to
who)
VALUES
('1500008', 'PowerPoint', '2016-09-15', 'Everyone');

```

```

INSERT INTO tblResource (
ResourceID, Type of resource, Date of availability, Available to
who)
VALUES
('1500009', 'Word', '2017-01-07', 'Everyone');

```

```

INSERT INTO tblResource (
ResourceID, Type of resource, Date of availability, Available to
who)
VALUES
('1500010', 'Excel', '2017-03-01', 'Tutors');

```

```

Mysql> CREATE TABLE tblResourceuploadedbyStaff (
StaffID smallint not null,
ResourceID smallint not null,
Foreign key (StaffID) references tblStaff(StaffID) ON UPDATE
CASCADE ON DELETE RESTRICT,
Foreign key (ResourceID) references tblResource(ResourceID) ON
UPDATE CASCADE ON DELETE RESTRICT
);

```

```

INSERT INTO tblResourceuploadedbyStaff (
StaffID, ResourceID)
VALUES
('1200001', '1500001');

```

```
INSERT INTO tblResourceuploadedbyStaff (  
StaffID, ResourceID)  
VALUES  
(\1200002', \1500002');  
  
INSERT INTO tblResourceuploadedbyStaff (  
StaffID, ResourceID)  
VALUES  
(\1200003', \1500003');  
  
INSERT INTO tblResourceuploadedbyStaff (  
StaffID, ResourceID)  
VALUES  
(\1200004', \1500004');  
  
INSERT INTO tblResourceuploadedbyStaff (  
StaffID, ResourceID)  
VALUES  
(\1200005', \1500005');  
  
INSERT INTO tblResourceuploadedbyStaff (  
StaffID, ResourceID)  
VALUES  
(\1200006', \1500006');  
  
INSERT INTO tblResourceuploadedbyStaff (  
StaffID, ResourceID)  
VALUES  
(\1200007', \1500007');  
  
INSERT INTO tblResourceuploadedbyStaff (  
StaffID, ResourceID)  
VALUES  
(\1200008', \1500008');  
  
INSERT INTO tblResourceuploadedbyStaff (  
StaffID, ResourceID)  
VALUES  
(\1200009', \1500009');
```

```
INSERT INTO tblResourceuploadedbyStaff (
StaffID, ResourceID)
VALUES
('1200010', '1500010');
```

```
Msql> SHOW TABLES;
```

Conclusion

In this section of the report all the tables have been inputted on the database and all data has been added to each tables as well. At the end of this you can see all the tables that have been created in the database “Acetraining” to see all these tables we do the command that is list above .This will allow us to see what tables are in each database this helps us to see what each database holds. In the next section we will start with Users.

Users

In this section I will show you how to set up users and grant them certain options on the database we use this to enforce who can do what on the database this helps us in many ways as certain members of staff will need more options than others.

```
mysql> CREATE USER 'Admin'@'localhost' IDENTIFIED BY 'Adminpass';
```

```
mysql> CREATE USER 'Student'@'localhost' IDENTIFIED BY
'Studentpass';
```

```
Mysql> CREATE USER 'Tutor'@'localhost' IDENTIFIED BY 'Tutorpass';
```

```
GRANT SELECT, INSERT (CourseID, Course name, Credit value,
Duration, Start date, End date, Fee), UPDATE (CourseID, Course
name, Credit value, Duration, Start date, End date) ON
AceTraining.tblCourse TO 'Tutor'@'localhost';
```

```
GRANT SELECT (StudentID, NextofkinID, First name, Middle name,
Last name, Date of birth, Address one, Address two, County, Postal
code, E-mail address, Phone number, Date enrolled, Visa expiry
date, Passport number) ON AceTraining.tblStudent TO
'Tutor'@'localhost';
```

```
GRANT SELECT, UPDATE (StaffID, NextofkinID, First name, Middle name, Last name, Date of birth, Address one, Address two, County, Postal code, E-mail address, Office number, Phone number, National insurance number) ON AceTraining.tblStaff TO 'Tutor'@'localhost';
```

```
GRANT SELECT, UPDATE (NextofkinID, First name, Middle name, Last name, Date of birth, Address one, Address two, County, Postcode, Phone number, E-mail address, Relation to Person) ON AceTraining.tblNextofKin TO 'Tutor'@'localhost';
```

```
GRANT INSERT, SELECT, DELETE (ResourceID, Type of resource, Date of availability, Available to who) ON AceTraining.tblResource TO 'Tutor'@'localhost';
```

```
GRANT SELECT, INSERT, UPDATE (Quiz ID, Student progress ID, Type of quiz, Name of quiz, Number of quizzes, Correct answers, Incorrect answers, Date uploaded, Deadline date), UPDATE (Student progress ID, Type of quiz, Name of quiz, Number of quizzes, Correct answers, Incorrect answers, Date uploaded, Deadline date) ON AceTraining.tblQuiz TO 'Tutor'@'localhost';
```

```
GRANT SELECT, UPDATE (Student progress ID, Student full Name, Number of completed quizzes, Score in quizzes, Average grade, Highest score in quiz, Lowest score in quiz, Progress bar) ON AceTraining.tblStudentProgress TO 'Tutor'@'localhost';
```

```
GRANT SELECT, UPDATE (RoomID, StudentID, Number of Classes attended, Number of absences, Number of explained absences) ON tblAttendance TO 'Tutor'@'localhost';
```

```
GRANT SELECT (CourseID, Course name, Credit value, Duration, Start date, End date, Fee) ON AceTraining.tblCourse TO 'Student'@'localhost';
```

```
GRANT SELECT, UPDATE (StudentID, NextofkinID, First name, Middle name, Last name, Date of birth, Address one, Address two, County, Postal code, E-mail address, Phone number, Date enrolled, Visa expiry date, Passport number) ON AceTraining.tblStudent TO 'Student'@'localhost';
```

```
GRANT SELECT (First name, Last name, Office number) ON AceTraining.tblStaff TO 'Student'@'localhost';
```

```
GRANT SELECT, UPDATE (NextofkinID, First name, Middle name, Last name, Date of birth, Address one, Address two, County, Postcode,
```

Phone number, E-mail address, Relation to Person) ON
AceTraining.tblNextofKin TO 'Student'@'localhost';

GRANT SELECT (Resource ID, Type of resource) ON
AceTraining.tblResource TO 'Student'@'localhost';

GRANT SELECT, INSERT (QuizID, StudentProgressID, Type of quiz,
Name of quiz, Date uploaded, Deadline date) ON AceTraining.tblQuiz
TO 'Student'@'localhost';

GRANT SELECT (StudentprogressID, Student full Name, Number of
completed quizzes, Score in quizzes, Average grade, Highest score
in quiz, Lowest score in quiz, Progress bar) ON
AceTraining.tblStudentProgress TO 'Student'@'localhost';

GRANT SELECT (Room ID, StudentID, Number of Classes attended,
Number of absences, Number of explained absences) ON
AceTraining.tblAttendance TO 'Student'@'localhost';

GRANT UPDATE, DELETE (Course ID, Course name, Credit value,
Duration, Start date, End date, Fee) ON AceTraining.tblCourse TO
'Admin'@'localhost';

GRANT INSERT, SELECT, UPDATE, DELETE (Student ID, Next of kin ID,
First name, Middle name, Last name, Date of birth, Address one,
Address two, County, Postal code, E-mail address, Phone number,
Date enrolled, Visa expiry date, Passport number) ON
AceTraining.tblStudent TO 'Admin'@'localhost';

GRANT INSERT, SELECT, DELETE (Staff ID, Next of kin ID, First name,
Middle name, Last name, Date of birth, Address one, Address two,
County, Postal code, E-mail address, Office number, Phone number,
National insurance number) ON AceTraining.tblStaff TO
'Admin'@'localhost';

GRANT INSERT, SELECT, DELETE (Next of kin ID, First name, Middle
name, Last name, Date of birth, Address one, Address two, County,
Postcode, Phone number, E-mail address, Relation to Person) ON
AceTraining.tblNextofKin TO 'Admin'@'localhost';

GRANT SELECT, DELETE (Resource ID, Type of resource, Date of
availability, Available to who) ON AceTraining.tblResource TO
'Admin'@'localhost';

```
GRANT SELECT, DELETE (QuizID, StudentProgressID, Type of quiz,  
Name of quiz, Number of quizzes, Correct answers, Incorrect  
answers, Date uploaded, Deadline date) ON AceTraining.tblQuiz TO  
'Admin'@'localhost';
```

```
GRANT DELETE (Student progress ID, Student full Name, Number of  
completed quizzes, Score in quizzes, Average grade, Highest score  
in quiz, Lowest score in quiz, Progress bar) ON  
AceTraining.tblStudentProgress TO 'Admin'@'localhost';
```

```
GRANT DELETE (RoomID, StudentID, Number of Classes attended,  
Number of absences, Number of explained absences) ON  
AceTraining.tblAttendance TO 'Admin'@'localhost';
```

Conclusion

In this section of the report all user have been set up and they have all the permissions they need in order to do work on the database. In the next section we start with Security and Contingency.

Chapter six

Security and Contingency

Security

The security in any database is very important as a lot of important data is sorted within these database so one of the main factors we have to look at when creating a database is how we plan on protecting it there are many ways you can do this with will be unlined in this section .

```
mysql> CREATE USER 'Admin'@'localhost' IDENTIFIED BY 'Adminpass';
```

```
mysql> CREATE USER 'Student'@'localhost' IDENTIFIED BY  
'Studentpass';
```

```
Mysql> CREATE USER 'Tutor'@'localhost' IDENTIFIED BY 'Tutorpass';
```

To give added protection I have added passwords to all the users in the database this allows them to only access what they are in the business. For example if the person that is logging on the database is a student they will only be able to access the user type student as they only have the password to that user type another reason why we set user types with passwords is so that we can set permissions to each person on the database so you know how much access they have on the database this stops users that could do damage to the database and also helps the business to keeps certain data secure.

Another added security level that I have added is the GRANT function that SQL uses this allows you to set up permissions in the database which allows user to only do certain things in the database for example if you wanted to have your admin user to be able to delete a record then you would have to create a GRANT statement with the word DELETE in it followed by field headings and you need to say what table the user can delete from this would allow the admin to delete a record. If you do not set GRANT statements to your users then by default they would not be able to do anything on the database so we set these statements up so users can interact with the database.

```
GRANT SELECT, INSERT (CourseID, Course name, Credit value,  
Duration, Start date, End date, Fee), UPDATE (CourseID, Course  
name, Credit value, Duration, Start date, End date) ON  
AceTraining.tblCourse TO 'Tutor'@'localhost';
```

This is an example from my database as you can see I have listed that the user which is call Tutor can SELECT and INSERT data in this table but nothing else as I have not granted this user other permissions that would allow him or her to do more with this table for example this user cannot delete any data from this table as it has not been listed in grant.

Conclusion

In this section of the report the security of the database was talked about and what methods have been in place to add different levels to make sure data that is on the database is safe. In the next section of the report we will be talking about contingency.

Contingency

In this section we will talk about what could happen to the database and what steps we are to take in case these threats become reality. Also we will be listing different methods that we are going to use to backup our data this is very important as any database will collect a lot of data so we will need to decide how we are going to backup.

In every business you must make plans for errors in your business this would include errors in your database below this is my contingency plan for my database just in case any of these happen to my database we can maintain some control over the situation...

In the case of natural disaster depending on the disaster for this I will be using an earthquake as an example we will follow this plan

1. All staff will exit the building as quickly as possible through one of the emergency exit.
2. When all staff have gathered at emergency points the building will become closed so no one is allowed back in the building once they have left the building.
3. When the disaster is over send a team to the site to make sure the building is ok for work purposes.
4. When the building is ok for work send a team down to see the damage done to the equipment.
5. Find out what and how much data has been lost due to the disaster.
6. Check to see if any staff members were injured during the disaster.
7. Repair or replace any equipment needed to continue work.

In the case of Vandalism to the database we will follow this plan

1. Report damage to the police.
2. Inform staff about vandalism (depending on what and how much was vandalized).
3. Create a report on what was damaged (this would include cost of damage and a list of what was damaged).
4. Repair or replace any equipment that needs it.

In the case the database gets hacked we will follow this plan

1. Set security to max to prevent any data loss.

2. Start moving data to a safe location.
3. If any data has been lost try and track where the data is going to see where the hacker is.
4. When data has finished moving close access to that database to force the hacker to lose connection to the database.
5. When the attack is over see if any data has been lost if any data has been lost create a report informing the business as to what has been lost.
6. Inform the police about the hacker.

In the case any components have failed or will need to be replaced this is the plan we will follow

1. Transfer data over to a new database for when parts need to be replaced (if part has already failed then try and access the database to see if we can start pulling data off that database).
2. If the database can't be accessed as the damaged parts stop access report what data was on that database and look for backups.
3. When new parts have arrived turn database off to allow the part to be replaced (replace failed part with new parts.)
4. When new parts have been fitted start the database again and run a diagnosed test to make sure all parts are running fine.
5. If backups can be found when the database is set up follow step six if no backups can be found then redo all the missing data.
6. Transfer data back to database.

In the case the server load becomes too high this is the plan we will follow

1. See what server is being overloaded.
2. See which server has the least traffic on it and transfer another copy onto that database.
3. Send new users to the new server IP this will allow the other database to lose some of its traffic making it faster.
4. When the traffic has died down on both servers send copy to backup database and delete copy on second database.

Backup methods

Off-Site (Private)

One of the most used backup methods would be off-site private this is when the business makes an off-site building that is used to house the databases it handles this would be the only purpose of that building a few staff members would be stationed at that building in order to maintain the database in case any errors happen. This method has been used many times before and it has

proven that it works well if you use the right tools as many big businesses use this method as they have a lot of databases that need to be stored.

There are a few advantages when using this method one of these methods is once it has been set up everything else is straight forward all you need to do then is start copying your data over to the right database and you then have a building that stores all your backups so you can restore any data that could be lost in the future. Another advantage is when you create this off site location and you send data over to it you can still maintain ownership of that data this means that you own that data so no one can use it without your permission this is one of the main reasons why big businesses use this methods as you can still prove that you own the data as it is on your database.

Another advantage is that it is very secure as long as you are using a secure connection this means that most data that will pass through this connection will always reach the database that you are sending it to. This provided a sense of security as you know that all your data will be safe while it is being transferred to the right place also it will be harder for hackers to steal the data while it is being transferred as the connection you are using is more secure than normal.

Another important advantage is that this works well with a type of backup so if your business uses incremental backups this would work best for you as this type of backup, backups all changes that you have made since the last backup you will need this if you are planning to restore any data that has been lost with is perfect as you can store all this on a remote database in a off-site location.

Now that we have talked about the advantages we need to talk about the disadvantages when using this system as it is important to view both sides before making a choice. One of the main disadvantages to using this is the cost as you can guess having to create a building in which to house these database isn't cheap you will need to spend a lot of money to create the building. Then you need to spend money on creating the secure network needed to transfer data between the business and the remote building all these add up in creating a very costly way to backup data.

Another disadvantage is the levels of expertise your staff will need to be to first set up the database then to maintain the database itself as you will need more advance staff they will cost a lot more money than box standard staff you will have to take this into account if you are thinking about making more off-site locations.

Another disadvantage is limited space as you are creating a building to house these databases you can only have it so big this means that if you wanted to have more databases you will need more space and in order to do that you will need to create bigger buildings or create more buildings

to house these databases this all costs a lot of money as you will need to create more space in order to hold more data this is one of the main problems with this type of method.

Off-site (Cloud)

This is one of the newest types of backup methods that are used by businesses everywhere as you can see it uses cloud technology to save data and store it in a virtual database when this type of technology first appeared it wasn't the most stable and you couldn't have enough space to save massive amounts of data but as cloud technology has improved so has the use of it. Nowadays you can save as much as you like this means you can have as much space as you like.

An advantage of this is when you have installed all the hardware and software needed you are ready to go this is a easy way to store data and it can be done by anyone as the programs do all of the hard work so all you would need to do is set up what you are storing were and it will send it to the right database. So you won't need expert staff to perform tasks you can have normal staff that will do the job fine this will save you money in costs.

Another advantage is security as you are sending your data to an unknown database many people would have issues with how secure your data really is but as cloud technology has gotten better so has the amount of security making it harder for hackers to get your data. This would include many different levels of encryption so even if your data did get stolen there would be no way the hacker would be able to decrypt the data.

Another advantage would be the amount you can store online as it isn't fixed to a physical location you have unlimited space to work with so you can have as much data stored as you wish the only thing you need to think about is the cost of the space as you will need to pay for space to store your data you need to make sure that you are using every space that you have bought as it would be a waste of money if there is space that is unused.

Now that we have talked about the advantages we need to talk about the disadvantages when using this system as it is important to view both sides before making a choice. One of the main disadvantages with this type of backup is ownership issue as anyone can upload anything online how can you prove that it belongs to you this has become a major issue with saving data online as more and more people use the cloud as a backup method.

Another disadvantage is where your data is located as you are saving to the cloud and you don't know where the physical database is how do you know where it is the simple answer is you don't this could mean that one part of your data is in England and the other part is in china. This also brings me to another disadvantage when dealing with the cloud what data laws do you follow when you transfer and store data as you know now when you upload data you don't know where it is so if your business is in the UK and you store your data to a Chinese server then you transfer

that data to an India server which data laws do you follow this can be confusing as many places have different data laws which you will have to follow if your data ends up in their country.

Justification

Between the two that I have listed I will be going with the private off-site backup there are many different reason why I will be choosing this type of backup one of the advantages that I feel is very important is data ownership even time I backup my data I what to know that data can't be used by other people without my permission. This also means that any data I collect and use is the businesses own data this gives the business more credibility as it proves the business can collect data and present their findings. Another reason why I think this is the best option out of the two is that it provides a more secure transfer line so data won't be interrupted this gives the business a lot of benefits as the business won't need to worry about data being loss or hackers trying to steal data as the line is more secure than normal.

As I've listed above there are a few disadvantages to using this method of backup but I feel like the advantage weight out the disadvantages as one of the problems is space you need a lot of space to house the databases you are using but instead of building more buildings all you need to do is get bigger databases this would solve all the problems that involve space. Also you wouldn't then need to hire a lot of expert members you would only need a small team that would be stationed in that one building. As you get bigger sized databases it will increase the total cost because you are paying for more space that the business can use but as this is a medium to large size business I feel that this is the best method as the long term benefits make this backup type better for this size of business.

Operation of backup

The backup that I will be working with will be automatic it will be scheduled to back up at the end of the week so this would happen on a Sunday it will start at six o'clock in the morning and run on until it ends this will be a weekly backup so all the work done in that work will be stored in the right places. Each time the system does this it will record the date that the backup happened on this will allow us to see what is the latest point we can go back to in case the system does crash and data becomes loss. When the system is backing up it will affect the performance of the system some options in the business will become slow as a lot of resources will be taken by the backup system that will be in place but not all features will be loss to the system for example you will still be able to pull resources from the database but it will be slower than normal.

The main responsible of this falls on the experts that work in the off-site private building as they have most contact with the database they will be in charge of the backup system also they will be in charge of setting up time schedules were the backup can happen. If there are any changes to the date then they will need to inform everyone in the business that the backup system will take

place on this day after that it will return to the normal date. If a problems does happen then they will inform the manager about what has happen and if any data was lost this report then will be passed along to the technicians in the business where they will solve the issue and report back about what will be done about it.

Cost

As this is one of the more expensive methods it will cost more money to first build and second to maintain we will have to take into account the maintains cost for the databases and the building itself also we need to think about the cost of hiring expert staff to keep these databases running so in total we would be talking about a lot of money. For example to rent a 20GB database the cost is £900 a month so if we have one of these for a single year it will cost us £10,800 as you're not going to have just one database you could five or ten more like this so to save the cost you would either have a few large databases or you would have a lot of smaller ones. The biggest expanse would be the price of the building as you need to pay for materials, work force, land, safety measures, permissions to make it a workable building, water, gas, power and so on all these need to be set out in order to form a cost document.

Conclusion

In this section of the report the contingency plan has been set up it gives a detailed plan as what the business should do it something happens also the backup system has been set up in this section it details when the business should backup and the method to backup. The next section of the report is testing.

Chapter seven

Testing

In this part of the report I will be testing my database to see if I encounter any problems this allows me to see if all the commands I have inputted work if they do then the correct response should come up. To fully test the database I will be going through each command in the database and recording what happens some of these test will be shown with screenshots to allow you to see step by step what has happened.

Test Number	One
User	Tutor
Table	tblCourse
What is being tested	GRANT SELECT (CourseID, Course name, Credit value, Duration, Start date, End date, Fee) ON AceTraining.tblCourse TO 'Tutor'@'localhost';
Expected outcome	When this command runs the user should be able to select all data from the table under these fields
Outcome of test	The command was accepted

```
mysql> GRANT SELECT ON tblCourse TO 'Tutor'@'localhost';  
Query OK, 0 rows affected (0.03 sec)
```

Test Number	Two
User	Tutor
Table	tblCourse
What is being tested	GRANT INSERT (CourseID, Course name, Credit value, Duration, Start date, End date, Fee) ON AceTraining.tblCourse TO 'Tutor'@'localhost';
Expected outcome	When this command runs the user should be able to insert all data from the table under these fields
Outcome of test	The command was accepted

```
mysql> GRANT INSERT ON tblCourse TO 'Tutor'@'localhost';  
Query OK, 0 rows affected (0.00 sec)
```


Test Number	Three
User	Tutor
Table	tblCourse
What is being tested	GRANT UPDATE (CourseID, Course name, Credit value, Duration, Start date, End date) ON AceTraining.tblCourse TO 'Tutor'@'localhost';
Expected outcome	When this command runs the user should be able to update all data from the table under these fields
Outcome of test	The command was accepted

```
mysql> GRANT UPDATE (CourseID, CourseName, CreditValue, Duration, StartDate, EndDate) ON tblCourse TO 'Tutor'@'localhost';
Query OK, 0 rows affected (0.08 sec)
```

Test Number	Four
User	Tutor
Table	tblStudent
What is being tested	GRANT SELECT (StudentID, NextofkinID, First name, Middle name, Last name, Date of birth, Address one, Address two, County, Postal code, E-mail address, Phone number, Date enrolled, Visa expiry date, Passport number) ON AceTraining.tblStudent TO 'Tutor'@'localhost';
Expected outcome	When this command runs the user should be able to select all data from the table under these fields
Outcome of test	The command was accepted

```
mysql> GRANT SELECT ON tblStudent TO 'Tutor'@'localhost';
Query OK, 0 rows affected (0.00 sec)
```

Test Number	Five
User	Tutor
Table	tblStaff
What is being tested	GRANT SELECT (StaffID, NextofkinID, First name, Middle name, Last name, Date of birth, Address one, Address two, County, Postal code, E-mail address, Office number, Phone number, National insurance number) ON AceTraining.tblStaff TO 'Tutor'@'localhost';
Expected outcome	When this command runs the user should be able to select all data from the table under these fields
Outcome of test	The command was accepted

```
mysql> GRANT SELECT ON tblStaff TO 'Tutor'@'localhost';
Query OK, 0 rows affected (0.00 sec)
```

Test Number	Six
User	Tutor
Table	tblStaff
What is being tested	GRANT UPDATE (StaffID, NextofkinID, First name, Middle name, Last name, Date of birth, Address one, Address two, County, Postal code, E-mail address, Office number, Phone number, National insurance number) ON AceTraining.tblStaff TO 'Tutor'@'localhost';
Expected outcome	When this command runs the user should be able to update all data from the table under these fields
Outcome of test	The command was accepted

```
mysql> GRANT UPDATE ON tblStaff TO 'Tutor'@'localhost';
Query OK, 0 rows affected (0.00 sec)
```

Test Number	Seven
User	Tutor
Table	tblNextofKin
What is being tested	GRANT SELECT (NextofkinID, First name, Middle name, Last name, Date of birth, Address one, Address two, County, Postcode, Phone number, E-mail address, Relation to Person) ON AceTraining.tblNextofKin TO 'Tutor'@'localhost';
Expected outcome	When this command runs the user should be able to select all data from the table under these fields
Outcome of test	The command was accepted

```
mysql> GRANT SELECT ON tblNextofKin TO 'Tutor'@'localhost';
Query OK, 0 rows affected (0.00 sec)
```

Test Number	Eight
User	Tutor
Table	tblNextofKin
What is being tested	GRANT UPDATE (NextofkinID, First name, Middle name, Last name, Date of birth, Address one, Address two, County, Postcode, Phone number, E-mail address, Relation to Person) ON AceTraining.tblNextofKin TO 'Tutor'@'localhost';
Expected outcome	When this command runs the user should be able to update all data from the table under these fields
Outcome of test	The command was accepted

```
mysql> GRANT UPDATE ON tblNextofKin TO 'Tutor'@'localhost';
Query OK, 0 rows affected (0.00 sec)
```

Test Number	Nine
User	Tutor
Table	tblResource
What is being tested	GRANT SELECT (ResourceID, Type of resource, Date of availability, Available to who) ON AceTraining.tblResource TO 'Tutor'@'localhost';
Expected outcome	When this command runs the user should be able to select all data from the table under these fields
Outcome of test	The command was accepted

```
mysql> GRANT SELECT ON tblResource TO 'Tutor'@'localhost';
Query OK, 0 rows affected (0.00 sec)
```

Test Number	Ten
User	Tutor
Table	tblResource
What is being tested	GRANT INSERT (ResourceID, Type of resource, Date of availability, Available to who) ON AceTraining.tblResource TO 'Tutor'@'localhost';
Expected outcome	When this command runs the user should be able to insert all data from the table under these fields
Outcome of test	The command was accepted

```
mysql> GRANT INSERT ON tblResource TO 'Tutor'@'localhost';
Query OK, 0 rows affected (0.00 sec)
```

Test Number	Eleven
User	Tutor
Table	tblResource
What is being tested	GRANT DELETE (ResourceID, Type of resource, Date of availability, Available to who) ON AceTraining.tblResource TO 'Tutor'@'localhost';
Expected outcome	When this command runs the user should be able to delete all data from the table under these fields
Outcome of test	The command was accepted

```
mysql> GRANT DELETE ON tblResource TO 'Tutor'@'localhost';
Query OK, 0 rows affected (0.00 sec)
```

Test Number	Twelve
User	Tutor
Table	tblQuiz
What is being tested	GRANT SELECT (Quiz ID, Student progress ID, Type of quiz, Name of quiz, Number of quizzes, Correct answers, Incorrect answers, Date uploaded, Deadline date) ON AceTraining.tblQuiz TO 'Tutor'@'localhost';
Expected outcome	When this command runs the user should be able to select all data from the table under these fields
Outcome of test	The command was accepted

```
mysql> GRANT SELECT ON tblQuiz TO 'Tutor'@'localhost';
Query OK, 0 rows affected (0.01 sec)
```

Test Number	Twelve
User	Tutor
Table	tblQuiz
What is being tested	GRANT INSERT (Quiz ID, Student progress ID, Type of quiz, Name of quiz, Number of quizzes, Correct answers, Incorrect answers, Date uploaded, Deadline date) ON AceTraining.tblQuiz TO 'Tutor'@'localhost';
Expected outcome	When this command runs the user should be able to insert all data from the table under these fields
Outcome of test	The command was accepted

```
mysql> GRANT INSERT ON tblQuiz TO 'Tutor'@'localhost';
Query OK, 0 rows affected (0.00 sec)
```

Test Number	Thirteen
User	Tutor
Table	tblQuiz
What is being tested	GRANT UPDATE (Quiz ID, Student progress ID, Type of quiz, Name of quiz, Number of quizzes, Correct answers, Incorrect answers, Date uploaded, Deadline date) ON AceTraining.tblQuiz TO 'Tutor'@'localhost';
Expected outcome	When this command runs the user should be able to update all data from the table under these fields
Outcome of test	The command was accepted

```
mysql> GRANT UPDATE ON tblQuiz TO 'Tutor'@'localhost';
Query OK, 0 rows affected (0.00 sec)
```

Table	tblStudentProgress
What is being tested	GRANT SELECT (Student progress ID, Student full Name, Number of completed quizzes, Score in quizzes, Average grade, Highest score in quiz, Lowest score in quiz, Progress bar) ON AceTraining.tblStudentProgress TO 'Tutor'@'localhost';
Expected outcome	When this command runs the user should be able to select all data from the table under these fields
Outcome of test	The command was accepted

```
mysql> GRANT SELECT ON tblStudentProgress TO 'Tutor'@'localhost';
Query OK, 0 rows affected (0.00 sec)
```

Test Number	Fifteen
User	Tutor
Table	tblStudentProgress
What is being tested	GRANT UPDATE (Student progress ID, Student full Name, Number of completed quizzes, Score in quizzes, Average grade, Highest score in quiz, Lowest score in quiz, Progress bar) ON AceTraining.tblStudentProgress TO 'Tutor'@'localhost';
Expected outcome	When this command runs the user should be able to update all data from the table under these fields
Outcome of test	The command was accepted

```
mysql> GRANT UPDATE ON tblStudentProgress TO 'Tutor'@'localhost';
Query OK, 0 rows affected (0.00 sec)
```

Test Number	Sixteen
User	Tutor
Table	tblAttendance
What is being tested	GRANT SELECT (RoomID, StudentID, Number of Classes attended, Number of absences, Number of explained absences) ON AceTraining.tblAttendance TO 'Tutor'@'localhost';
Expected outcome	When this command runs the user should be able to select all data from the table under these fields
Outcome of test	The command was accepted

```
mysql> Grant SELECT ON tblAttendance TO 'Tutor'@'localhost';
Query OK, 0 rows affected (0.00 sec)
```

Table	tblAttendance
What is being tested	GRANT UPDATE(RoomID, StudentID, Number of Classes attended, Number of absences, Number of explained absences) ON AceTraining.tblAttendance TO 'Tutor'@'localhost';
Expected outcome	When this command runs the user should be able to update all data from the table under these fields
Outcome of test	The command was accepted

```
mysql> GRANT UPDATE ON tblAttendance TO 'Tutor'@'localhost';
Query OK, 0 rows affected (0.00 sec)
```

Test Number	Eighteen
User	Student
Table	tblCourse
What is being tested	GRANT SELECT (CourseID, Course name, Credit value, Duration, Start date, End date, Fee) ON AceTraining.tblCourse TO 'Student'@'localhost';
Expected outcome	When this command runs the user should be able to select all data from the table under these fields
Outcome of test	The command was accepted

Test Number	Nineteen
User	Student
Table	tblStudent
What is being tested	GRANT SELECT (StudentID, NextofkinID, First name, Middle name, Last name, Date of birth, Address one, Address two, County, Postal code, E-mail address, Phone number, Date enrolled, Visa expiry date, Passport number) ON AceTraining.tblStudent TO 'Student'@'localhost';
Expected outcome	When this command runs the user should be able to select all data from the table under these fields
Outcome of test	The command was accepted

Test Number	Nineteen
User	Student
Table	tblStudent
What is being tested	GRANT UPDATE (StudentID, NextofkinID, First name, Middle name, Last name, Date of birth, Address one, Address two, County, Postal code, E-mail address, Phone number, Date enrolled, Visa expiry date, Passport number) ON AceTraining.tblStudent TO 'Student'@'localhost';
Expected outcome	When this command runs the user should be able to update all data from the table under these fields
Outcome of test	The command was accepted

```
mysql> GRANT UPDATE ON tblStudent TO 'Student'@'localhost';
Query OK, 0 rows affected (0.00 sec)
```

Test Number	Twenty
User	Student
Table	tblStaff
What is being tested	GRANT SELECT (First name, Last name) ON AceTraining.tblStaff TO 'Student'@'localhost';
What is not being tested	Office number
Expected outcome	When this command runs the user should be able to select all data from the table under these fields
Outcome of test	The command was accepted

```
mysql> GRANT SELECT (FirstName, LastName) ON tblStaff TO 'Student'@'localhost';
Query OK, 0 rows affected (0.00 sec)
```

Test Number	Twenty one
User	Student
Table	tblNextofKin
What is being tested	GRANT SELECT (NextofkinID, First name, Middle name, Last name, Date of birth, Address one, Address two, County, Postcode, Phone number, E-mail address, Relation to Person) ON AceTraining.tblNextofKin TO 'Student'@'localhost';
Expected outcome	When this command runs the user should be able to select all data from the table under these fields
Outcome of test	The command was accepted

Test Number	Twenty two
User	Student
Table	tblNextofKin
What is being tested	GRANT UPDATE (NextofkinID, First name, Middle name, Last name, Date of birth, Address one, Address two, County, Postcode, Phone number, E-mail address, Relation to Person) ON AceTraining.tblNextofKin TO 'Student'@'localhost';
Expected outcome	When this command runs the user should be able to update all data from the table under these fields
Outcome of test	The command was accepted

Test Number	Twenty three
User	Student
Table	tblResource
What is being tested	GRANT SELECT (Resource ID, Type of resource) ON AceTraining.tblResource TO 'Student'@'localhost';
Expected outcome	When this command runs the user should be able to select all data from the table under these fields
Outcome of test	The command was accepted

```
mysql> GRANT SELECT (ResourceID, TypeofResource) ON tblResource TO 'Student'@'localhost';
Query OK, 0 rows affected (0.05 sec)
```

Test Number	Twenty four
User	Student
Table	tblQuiz
What is being tested	GRANT SELECT (QuizID, Type of quiz, Date uploaded, Deadline date) ON AceTraining.tblQuiz TO 'Student'@'localhost';
Not being tested	StudentProgressID, name of quiz
Expected outcome	When this command runs the user should be able to select all data from the table under these fields
Outcome of test	The command was accepted

```
mysql> GRANT SELECT (QuizID, TypeofQuiz, DateUploaded, DeadlineDate) ON tblQuiz TO 'Student'@'localhost';
Query OK, 0 rows affected (0.00 sec)
```


Test Number	Twenty five
User	Student
Table	tblQuiz
What is being tested	GRANT INSERT (QuizID, Type of quiz, Date uploaded, Deadline date) ON AceTraining.tblQuiz TO 'Student'@'localhost';
Not being tested	StudentProgressID, name of quiz
Expected outcome	When this command runs the user should be able to insert all data from the table under these fields
Outcome of test	The command was accepted

```
mysql> GRANT INSERT (QuizID, TypeofQuiz, DateUploaded, DeadlineDate) ON tblQuiz TO 'Student'@'localhost';
Query OK, 0 rows affected (0.01 sec)
```

Test Number	Twenty six
User	Student
Table	tblStudentProgress
What is being tested	GRANT SELECT (Student progress ID, Student full Name, Number of completed quizzes, Score in quizzes, Average grade, Highest score in quiz, Lowest score in quiz, Progress bar) ON AceTraining.tblStudentProgress TO 'Student'@'localhost';
Expected outcome	When this command runs the user should be able to select all data from the table under these fields
Outcome of test	The command was accepted

Test Number	Twenty seven
User	Student
Table	tblAttendance
What is being tested	GRANT SELECT (RoomID, StudentID, Number of Classes attended, Number of absences, Number of explained absences) ON AceTraining.tblAttendance TO 'Student'@'localhost';
Expected outcome	When this command runs the user should be able to select all data from the table under these fields
Outcome of test	The command was accepted

Test Number	Twenty eight
User	Admin
Table	tblCourse
What is being tested	GRANT UPDATE (CourseID, Course name, Credit value, Duration, Start date, End date, Fee) ON AceTraining.tblCourse TO 'Admin'@'localhost';
Expected outcome	When this command runs the user should be able to select all data from the table under these fields
Outcome of test	The command was accepted

Test Number	Twenty eight
User	Admin
Table	tblCourse
What is being tested	GRANT DELETE (CourseID, Course name, Credit value, Duration, Start date, End date, Fee) ON AceTraining.tblCourse TO 'Admin'@'localhost';
Expected outcome	When this command runs the user should be able to delete all data from the table under these fields
Outcome of test	The command was accepted

```
mysql> GRANT DELETE ON tblCourse TO 'Admin'@'localhost';
Query OK, 0 rows affected (0.00 sec)
```

Test Number	Twenty nine
User	Admin
Table	tblStudent
What is being tested	GRANT SELECT (StudentID, NextofkinID, First name, Middle name, Last name, Date of birth, Address one, Address two, County, Postal code, E-mail address, Phone number, Date enrolled, Visa expiry date, Passport number) ON AceTraining.tblStudent TO 'Admin'@'localhost';
Expected outcome	When this command runs the user should be able to select all data from the table under these fields
Outcome of test	The command was accepted

Test Number	Thirty
User	Admin
Table	tblStudent
What is being tested	GRANT INSERT (StudentID, NextofkinID, First name, Middle name, Last name, Date of birth, Address one, Address two, County, Postal code, E-mail address, Phone number, Date enrolled, Visa expiry date, Passport number) ON AceTraining.tblStudent TO 'Admin'@'localhost';
Expected outcome	When this command runs the user should be able to insert all data from the table under these fields
Outcome of test	The command was accepted

```
mysql> GRANT INSERT ON tblStudent TO 'Admin'@'localhost';
Query OK, 0 rows affected (0.00 sec)
```

Test Number	Thirty one
User	Admin
Table	tblStudent
What is being tested	GRANT UPDATE (StudentID, NextofkinID, First name, Middle name, Last name, Date of birth, Address one, Address two, County, Postal code, E-mail address, Phone number, Date enrolled, Visa expiry date, Passport number) ON AceTraining.tblStudent TO 'Admin'@'localhost';
Expected outcome	When this command runs the user should be able to update all data from the table under these fields
Outcome of test	The command was accepted

Test Number	Thirty two
User	Admin
Table	tblStudent
What is being tested	GRANT DELETE (StudentID, NextofkinID, First name, Middle name, Last name, Date of birth, Address one, Address two, County, Postal code, E-mail address, Phone number, Date enrolled, Visa expiry date, Passport number) ON AceTraining.tblStudent TO 'Admin'@'localhost';
Expected outcome	When this command runs the user should be able to delete all data from the table under these fields
Outcome of test	The command was accepted

```
mysql> GRANT DELETE ON tblStudent TO 'Admin'@'localhost';
Query OK, 0 rows affected (0.00 sec)
```

Test Number	Thirty three
User	Admin
Table	tblStaff
What is being tested	GRANT SELECT (Staff ID, Next of kin ID, First name, Middle name, Last name, Date of birth, Address one, Address two, County, Postal code, E-mail address, Office number, Phone number, National insurance number) ON AceTraining.tblStaff TO 'Admin'@'localhost';
Expected outcome	When this command runs the user should be able to select all data from the table under these fields
Outcome of test	The command was accepted

Test Number	Thirty four
User	Admin
Table	tblStaff
What is being tested	GRANT INSERT (Staff ID, Next of kin ID, First name, Middle name, Last name, Date of birth, Address one, Address two, County, Postal code, E-mail address, Office number, Phone number, National insurance number) ON AceTraining.tblStaff TO 'Admin'@'localhost';
Expected outcome	When this command runs the user should be able to insert all data from the table under these fields
Outcome of test	The command was accepted

```
mysql> GRANT INSERT ON tblStaff TO 'Admin'@'localhost';
Query OK, 0 rows affected (0.00 sec)
```

Test Number	Thirty five
User	Admin
Table	tblStaff
What is being tested	GRANT DELETE (Staff ID, Next of kin ID, First name, Middle name, Last name, Date of birth, Address one, Address two, County, Postal code, E-mail address, Office number, Phone number, National insurance number) ON AceTraining.tblStaff TO 'Admin'@'localhost';
Expected outcome	When this command runs the user should be able to delete all data from the table under these fields
Outcome of test	The command was accepted

```
mysql> GRANT DELETE ON tblStaff TO 'Admin'@'localhost';
Query OK, 0 rows affected (0.00 sec)
```

Test Number	Thirty six
User	Admin
Table	tblNextofKin
What is being tested	GRANT SELECT (Next of kin ID, First name, Middle name, Last name, Date of birth, Address one, Address two, County, Postcode, Phone number, E-mail address, Relation to Person) ON AceTraining.tblNextofKin TO 'Admin'@'localhost';
Expected outcome	When this command runs the user should be able to select all data from the table under these fields
Outcome of test	The command was accepted

Test Number	Thirty seven
User	Admin
Table	tblNextofKin
What is being tested	GRANT INSERT (Next of kin ID, First name, Middle name, Last name, Date of birth, Address one, Address two, County, Postcode, Phone number, E-mail address, Relation to Person) ON AceTraining.tblNextofKin TO 'Admin'@'localhost';
Expected outcome	When this command runs the user should be able to insert all data from the table under these fields
Outcome of test	The command was accepted

```
mysql> GRANT INSERT ON tblNextofKin TO 'Admin'@'localhost';
Query OK, 0 rows affected (0.00 sec)
```

Test Number	Thirty eight
User	Admin
Table	tblNextofKin
What is being tested	GRANT DELETE (Next of kin ID, First name, Middle name, Last name, Date of birth, Address one, Address two, County, Postcode, Phone number, E-mail address, Relation to Person) ON AceTraining.tblNextofKin TO 'Admin'@'localhost';
Expected outcome	When this command runs the user should be able to delete all data from the table under these fields
Outcome of test	The command was accepted

```
mysql> GRANT DELETE ON tblNextofKin TO 'Admin'@'localhost';
Query OK, 0 rows affected (0.00 sec)
```

Test Number	Thirty nine
User	Admin
Table	tblResource
What is being tested	GRANT SELECT (Resource ID, Type of resource, Date of availability, Available to who) ON AceTraining.tblResource TO 'Admin'@'localhost';
Expected outcome	When this command runs the user should be able to delete all data from the table under these fields
Outcome of test	The command was accepted

Test Number	Forty
User	Admin
Table	tblResource
What is being tested	GRANT DELETE (Resource ID, Type of resource, Date of availability, Available to who) ON AceTraining.tblResource TO 'Admin'@'localhost';
Expected outcome	When this command runs the user should be able to delete all data from the table under these fields
Outcome of test	The command was accepted

Test Number	Forty one
User	Admin
Table	tblQuiz
What is being tested	GRANT SELECT (QuizID, StudentProgressID, Type of quiz, Name of quiz, Number of quizzes, Correct answers, Incorrect answers, Date uploaded, Deadline date) ON AceTraining.tblQuiz TO 'Admin'@'localhost';
Expected outcome	When this command runs the user should be able to select all data from the table under these fields
Outcome of test	The command was accepted

Test Number	Forty two
User	Admin
Table	tblQuiz
What is being tested	GRANT DELETE (QuizID, StudentProgressID, Type of quiz, Name of quiz, Number of quizzes, Correct answers, Incorrect answers, Date uploaded, Deadline date) ON AceTraining.tblQuiz TO 'Admin'@'localhost';
Expected outcome	When this command runs the user should be able to delete all data from the table under these fields
Outcome of test	The command was accepted

```
mysql> GRANT DELETE ON tblQuiz TO 'Admin'@'localhost';
Query OK, 0 rows affected (0.00 sec)
```

Test Number	Forty three
User	Admin
Table	tblStudentProgress
What is being tested	GRANT DELETE (Student progress ID, Student full Name, Number of completed quizzes, Score in quizzes, Average grade, Highest score in quiz, Lowest score in quiz, Progress bar) ON AceTraining.tblStudentProgress TO 'Admin'@'localhost';
Expected outcome	When this command runs the user should be able to delete all data from the table under these fields
Outcome of test	The command was accepted

```
mysql> GRANT DELETE ON tblStudentProgress TO 'Admin'@'localhost';
Query OK, 0 rows affected (0.00 sec)
```

Test Number	Forty four
User	Admin
Table	tblStudentProgress
What is being tested	GRANT DELETE (RoomID, StudentID, Number of Classes attended, Number of absences, Number of explained absences) ON AceTraining.tblAttendance TO 'Admin'@'localhost';
Expected outcome	When this command runs the user should be able to delete all data from the table under these fields
Outcome of test	The command was accepted

```
mysql> GRANT DELETE ON tblAttendance TO 'Admin'@'localhost';
Query OK, 0 rows affected (0.00 sec)
```

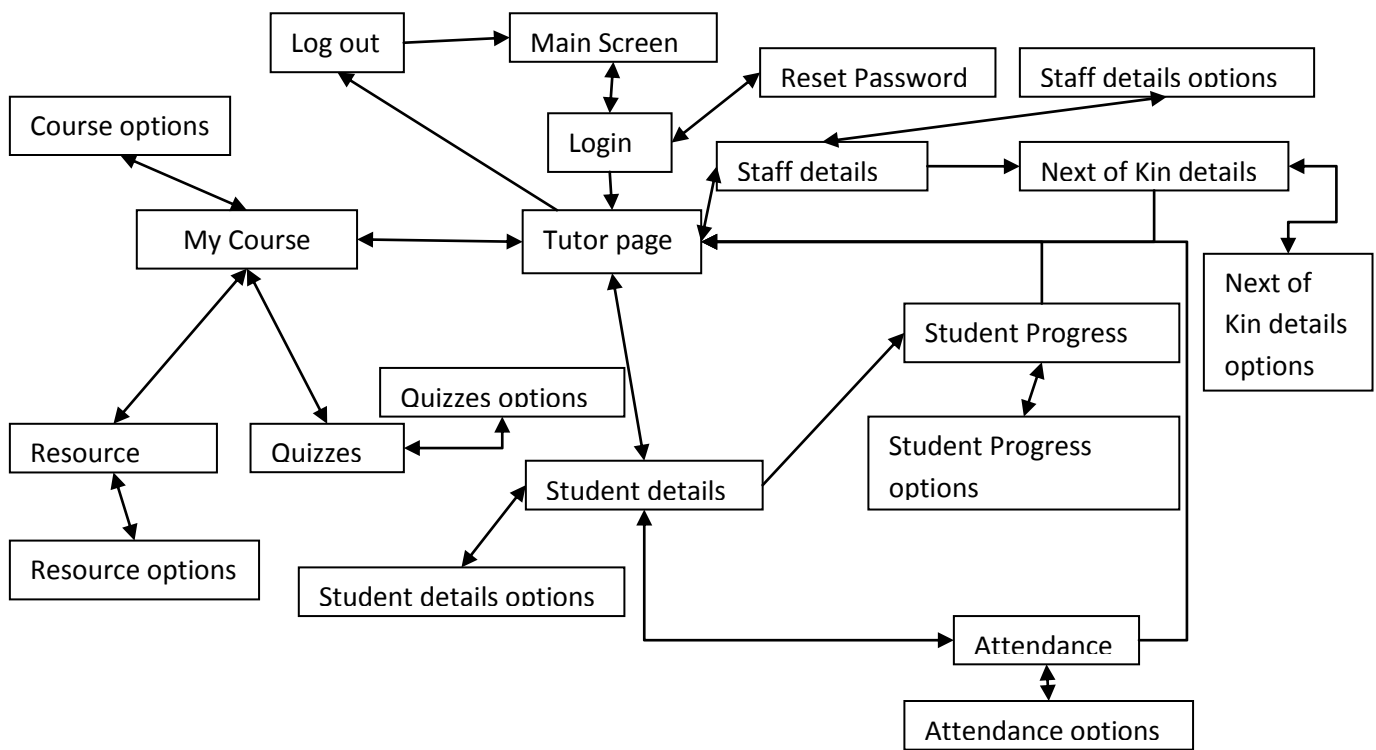
Conclusion

In this section of the report the database has been tested to see if the grant permissions are working this is to show the client that the database is in full working order. The next section of the report is Graphical user interface design.

Graphical User Interface Design

In this section I will show you the design for the database this will allow the user to navigate around the database so they can reach their destination. In this you will see how each page links together we do this to create an idea for the client so they can see how it will look when it is finished also it allows for any changes that the client wishes. Another reason why we create a user interface design is because it allows you to see how you can from point A to point B.

Navigation - Tutor

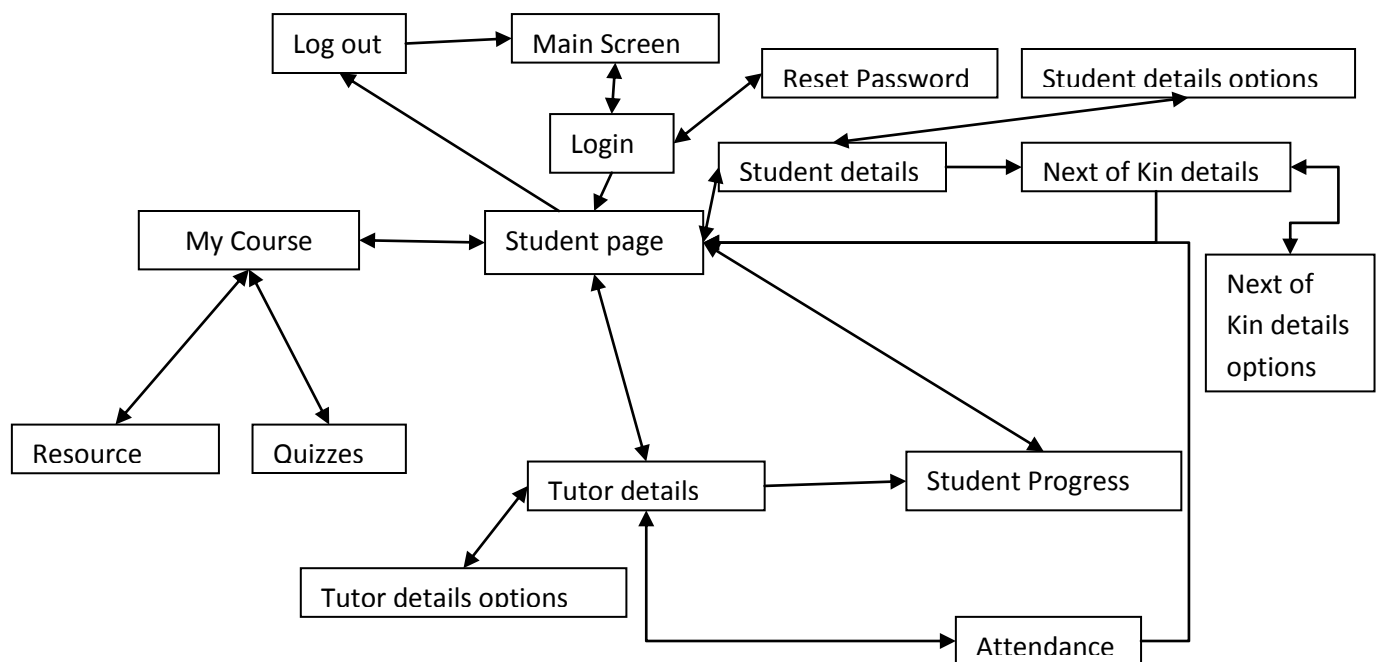


This is the navigation that the tutor will be able to do when they access the database first they will start by logging into the database at this stage they also can reset their passwords if they have forgotten it when they have done that they will reach the tutor page this will act as the hub as most of the pages are linked to it. From here they will be able to travel to Tutor details, Student details, My Course and log out if they wish to access areas that aren't one click away they will have

to access areas that are just before that for example if you wanted to access Attendance you would first have to travel to Student options from their you would be able to access Attendance same with Next of Kin if you wished to access that part you first would have to go through Profile then you could access Next of Kin.

As you can see most of the pages are linked to tutor page so the most you have to travel from that page is two clicks as it is the acting hub I thought it would be important to make sure you can go back to the hub at any time.

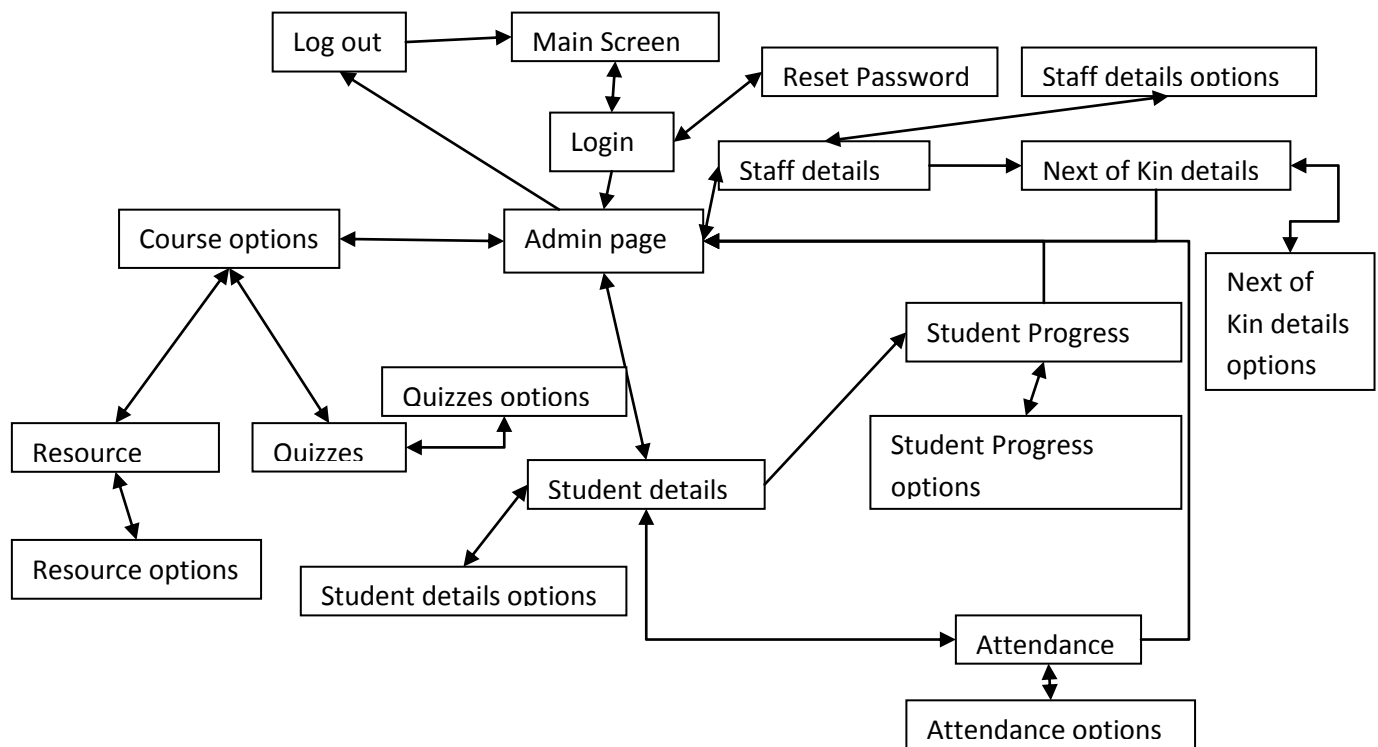
Navigation -Student



This is the navigation that the student will be able to do when they access the database first they will start by logging into the database at this stage they also can reset their passwords if they have forgotten it when they have done that they will reach the student page this will act as the hub as most of the pages are linked to it. From here they will be able to travel to Student details, My Course, Tutor details, Student Progress and log out if they wish to access areas that aren't one click away they will have to access areas that are just before that for example if you wanted to access Attendance you would first have to travel to Student Progress from their you would be able to access Attendance same with Next of Kin if you wished to access that part you first would have to go through Profile then you could access Next of Kin.

As you can see most of the pages are linked to tutor page so the most you have to travel from that page is two clicks as it is the acting hub I thought it would be important to make sure you can go back to the hub at any time.

Navigation -Admin



This is the navigation that the admin will be able to do when they access the database first they will start by logging into the database at this stage they also can reset their passwords if they have forgotten it and they have a register option this will be for any new members at the university it is the admin that will set up peoples accounts. When they have done that they will reach the admin page this will act as the hub as most of the pages are linked to it from here they will be able to travel to Course options, Staff details, Student details and log out unlike with the other two users the admin has a lot more access to more areas within the database as they have more advance permissions this allows them to do more actions within the database. If they wish to access areas that aren't one click away they will have to access areas that are just before that for example if you wanted to access Attendance you would first have to travel to Student Progress from their you would be able to access Attendance same with Next of Kin if you wished to access that part you first would have to go through Profile then you could access Next of Kin.

As you can see most of the pages are linked to tutor page so the most you have to travel from that page is two clicks as it is the acting hub I thought it would be important to make sure you can go back to the hub at any time.

Data Capture

Log in screen

Acetraining


Username:

Password:

Log in

-
- Log in will be a button when the user has inputted the correct values above they will be taken to the home page.
 - If the user has entered anything that is incorrect in the text boxes an error message will appear telling them they made a mistake also the system will tell them where the mistake was and how they should correct it.

Student details options

Title	
Student ID	
Next of Kin ID	
Forename	
Middle name	
Surname	
DOB	
Address one	
Address two	
County	
Postcode	
E-mail	
Phone number	

Add Student

Update Student


Search Student

Delete Student

- If the user has entered anything that is incorrect in the text boxes an error message will appear telling them they made a mistake also the system will tell them where the mistake was and how they should correct it.
- Add, Search, Update and Delete student will be a button when the user has inputted the correct values above they will be taken to the Student details page.
- All names will be 35 characters long.
- DOB will only allow dates to be entered they can either it by text or they can use the calendar that will be in place.

- Students can access this but they can only update their own information so they won't be able to see add and delete buttons and they can't search for other students.
- Tutors can't update this information they can only see it and search for students.

Staff details options

Title	
Staff ID	
Next of Kin ID	
Forename	
Middle name	
Surname	
DOB	
Address one	
Address two	
County	
Postcode	
E-mail	
Office number	
Phone number	
National insurance number	

Add Tutor

Update Tutor

Search Tutor

Delete Tutor

- If the user has entered anything that is incorrect in the text boxes an error message will appear telling them they made a mistake also the system will tell them where the mistake was and how they should correct it.
- Add Tutor, Update Tutor and Delete Tutor will be a button when the user has inputted the correct values above they will be taken to the Tutor details page.
- All names will be 35 characters long.
- DOB will only allow dates to be entered they can either it by text or they can use the calendar that will be in place.
- Tutors can update their own information but can't add or delete records.
- Student can access this but they will only be able to see certain information as most of it is personal information they can search for tutors but only certain information will appear.

Reset Password

Title

Current Username:

Security question:

1) Option one

2) Option two

3) Option three

New Password:

Reset Password

-
- If the user has entered anything that is incorrect in the text boxes an error message will appear telling them they made a mistake also the system will tell them where the mistake was and how they should correct it.
 - Reset Password will be a button when the user has inputted the correct values above they will be taken to the home page.

- Security question will be a radio button when the user clicks on one of the given answers it will check to see if they have chosen the right one if so it will allow the new password to be used.
- When the user is done a message box will tell them that their password has been changed.

My Course—Student and Tutor

Title	
	Hyperlink menu
	Hyperlink menu
	Hyperlink menu
	Search for course:
	Find course

- Both will be a hyper link menu that will take them to the course they have picked this will be the same for both staff and students but staff will have more courses to pick from.
- If a tutor has been assigned many different courses they can search for the course they want only tutor can see this as they will have more courses than students.
- Find course will be a button when it has been clicked it will find the course that was entered in search for course.

Course options-Staff

Title
Search for course:
Edit course:
New information:
Add course:
Delete course:

-
- Delete won't be seen for Tutors as they don't have permission only admin will see the option delete the same will be done for add course admin won't be able to add course.
 - Confirm, clear and cancel will all be buttons also when anyone changes anything to do with a course a message will come up saying do you really want this action to happen the same will appear if they wish to delete a course.

Resource options-Staff

Title
Name of Resource:
Date available:
Available to who:
File:
Delete file:
Block Tutor ID:
<div>Upload to resource</div> <div>Delete</div> <div>Block</div>

- If the user has entered anything that is incorrect in the text boxes an error message will appear telling them they made a mistake also the system will tell them where the mistake was and how they should correct it.
- Upload, Delete and Block are all buttons that will be used when the above boxes have been filled but the buttons will be disabled when certain boxes are filled in for example if you fill in name of resource, Date available, Available to who and File then Delete and block won't work as the system will think you are uploaded a file. This is the same with the other two if only Delete file is filled in then the other two will be disabled and if you only fill in Block Tutor ID then Delete and Upload will be disabled.
- File will be where the user places files they wish to upload.
- When the Resource has been uploaded it will be given an ID number when you view it.

Quizzes options – Staff

Title		
Name of Quiz:		
Type of Quiz:		
Student Progress ID:		
Number of Correct answers:		
Number of Incorrect answers:		
Date uploaded:		
Deadline date:		
Add Quiz	Delete Quiz	Send to Student Progress

- If the user has entered anything that is incorrect in the text boxes an error message will appear telling them they made a mistake also the system will tell them where the mistake was and how they should correct it.
- Add quiz, Delete quiz and send to student progress are all buttons like with the last options some will be disabled if you fill in certain boxes for example if you want to delete a quiz then you have to fill all the information apart from Student progress ID.
- When the Quiz has been uploaded it will be given an ID number when you view it.
- Type of Quiz will be used to place the quiz file in.
- A warning will come up when you try and delete a quiz.
- Tutors won't be able to delete a quiz so they can't see delete quiz button also admin can't add a new quiz so they won't see add quiz.

Student Progress option - Staff

Title		
Student Progress ID:		
Student full name:		
Number of completed quizzes:		
Score in quizzes:		
Average grade:		
Highest score in quiz:		
Lowest score in quiz:		
Progress bar:		
Add new Student Progress	Delete Student Progress	Update Student Progress

- If the user has entered anything that is incorrect in the text boxes an error message will appear telling them they made a mistake also the system will tell them where the mistake was and how they should correct it.
- Add new student progress, Delete student progress and Update student progress are all buttons like with the last options some will be disabled if you fill in certain boxes for example if you want to delete a student progress record then you only need to fill in student progress ID and that students name all other boxes have to be empty in order for the button to work.
- A warning will come up when you try and delete a student progress record.
- Tutors will not be able to delete a student progress record so this option will not be there for tutors and adding will not be there for tutors as well.

Next of Kin options

Title	
Next of Kin ID	
First name	
Middle name	
Last name	
DOB	
Address 1	
Address 2	
County	
Postcode	
Phone Number	
E-mail address	
Relation to Person	

Add Next of Kin

Update Next of Kin

Delete Next of Kin

- If the user has entered anything that is incorrect in the text boxes an error message will appear telling them they made a mistake also the system will tell them where the mistake was and how they should correct it.
- Add, Update and Delete Next of Kin will be a button when the user has inputted the correct values above they will be taken to the Next of Kin details.
- All names will be 35 characters long.
- DOB will only allow dates to be entered they can either it by text or they can use the calendar that will be in place.
- Students can access this but they can only update their own information so they won't be able to see add and delete buttons this will be the same for tutors as well.

Attendance options

Title

Room ID	
Number of Classes attended	
Number of Absences	
Number of explained absences	
Time	
Date	

Add Attendance

Update Attendance

Delete Attendance

- If the user has entered anything that is incorrect in the text boxes an error message will appear telling them they made a mistake also the system will tell them where the mistake was and how they should correct it.
- Add, Update and Delete Attendance will be a button when the user has inputted the correct values above they will be taken to the Attendance.
- Tutors can only updates this page add and delete will not be enabled to them also the student cannot access this page.

Conclusion

While writing this report I have learned a lot about what it takes to build and implement a database I don't say this lightly when I tell you it takes a lot of work by doing these different sections I been given the chance to explore the world of databases to find out what is a database and how they work in real life as I've been going through each section I was faced with new challenges that I would never have come across unless I was doing this subject. It has been very eye opening to understand the world of databases and what it can offer to any person.

References

Muhammad Haadi. (October 18, 2010). *The Evolution of Database*. Available: <https://mhaadi.wordpress.com/2010/10/18/the-evolution-of-database/>. Last accessed 13th Oct 2015.

Gavin Powell (2006). *Beginning Database Design Programmer to programmer Wrox beginning guides*. London: John Wiley & Sons. p0-12.

O'Brien, J. A., & Marakas, G. M. (2009). *Management Information Systems* (9th ed.). New York, NY: McGraw-Hill/Irwin

Clare Churcher (01 Feb 2007). *Beginning Database Design : From Novice to Professional*. New York, United States: aPress. p0-50.

Allen G. Taylor (20 Sep 2013). *SQL For Dummies*. 8th ed. New York, United States: John Wiley & Sons Inc. p0-21

John Garmany, Jeff Walker (12 May 2005). *Logical Database Design Principles*. London: Taylor & Francis Ltd. p0-15.

Margaret Rouse. (September 2005). *Normalization*. Available: <http://searchsqlserver.techtarget.com/definition/normalization>. Last accessed 11/11/2015.

Margaret Rouse. (September 2015). *Contingency plan*. Available: <http://whatis.techtarget.com/definition/contingency-plan>. Last accessed 9/11/2015.

Appendix