FACTFILE:
GCE DIGITAL TECHNOLOGY

UNIT A2 1: INFORMATION SYSTEMS

Rewarding Learning

Databases 3

Learning Outcomes

Students should be able to:

- Use Structured Query Language (SQL) to create tables and to retrieve, update, insert, and delete data in a relational database.
- Evaluate Query By Example (QBE) as an alternative SQL

Content in Databases 3

- The Use Structured Query Language (SQL) to create tables and to retrieve, update, insert, and delete data in a relational database.
- Evaluate Query By Example (QBE) as an alternative SQL

The characteristics of data in un-normalised form

Structured Query language(SQL) and Query by Example(QBE) are both used to access and manipulate data in a database. SQL is used widely by programmers to interrogate databases efficiently. QBE is a simple query language where the user creates a query using a form. Criteria are entered into the form and the query is executed to return a set of data which fulfils the query.

Structured Query Language

A Relational Database Management System is the basis for SQL. Consider the Student table below.

Student Table

StudentID	StudentName	DateOfBirth	Email	Address	City	PostCode
1001	Anne Browne	12/12/98	AB@mymail.com	1 Long Road	Coleraine	BT77 9PP
1002	John Gormley	12/5/97	jgormley@hitmail.com	37 Palm Ave	Limavady	BT02 7NN
1003	Ella Greer	14/6/98	egreer12@mymail.com	14 Beech Street	Omagh	BT67 8UU
1004	Martin McIntyre	01/01/97	marty@mcintyre.com	34 Palm Ave	Limavady	BT77 9PP
1005	Eoin Smyth	03/04/98	es123@hitmail.com	20 Oak Mews	Cookstown	BT67 2YY

The fields in the Student table are StudentID, StudentName, DateOfBirth, Email, Address, City, and PostCode. A field is equivalent to a column in the table. A record, also called a row, is each individual entry that exists in a table. For example, there are 5 records in the Student table. Each record has 7 fields. You have seen this already in the factfiles Databases 1 and Databases 2.

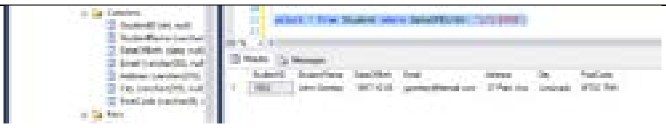
In order to create the database structure above, it is necessary to make use of SQL commands. The most common commands and their functions are listed below.

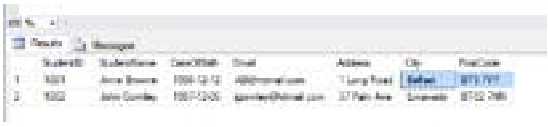
Task and SQL command	Example

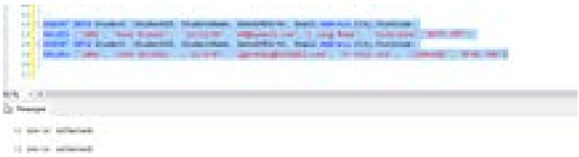
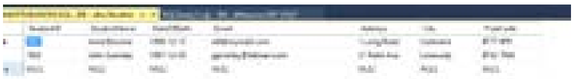
Create a new database
- **CREATE DATABASE**
CREATE DATABASE
databasename;
	CREATE DATABASE collegeDB – creates a database called collegeDB on the SQL server on the computer.

Create a table - **CREATE TABLE**
CREATE TABLE *table_name* (
column1 datatype,
column2 datatype,
column3 datatype,
etc.
);
	`CREATE TABLE Student (`
	`StudentID int not null,`
	`StudentName varchar(255),`
	`DateOfBirth Date,`
	`Email varchar(50),`
	`Address varchar(255),`
	`City varchar(255),`
	`PostCode varchar(8)`
	`Primary Key (StudentID) );`
	
	The code above will create a new table dbo.Student on the server. Note that there is a primary key, StudentID. Later in this document the table will be modified to provide for linking to another table.

Retrieve data from a database – **SELECT**
SELECT *column1, column2, etc.*
FROM *table_name*;
(column1, column2 are fieldnames)
	`To select the StudentName and DateOfBirth`
	`SELECT StudentName, DateOfBirth  FROM Student;`
	
	Criteria can be added to the SELECT statement
	SELECT * FROM Student; displays all fields and records in the table Student.
	`SELECT * FROM Student WHERE DateOfBirth< '1/1/1998'; displays all fields for pupils who were born before 1/1/1998 (see results below)`

Update data in a table – **UPDATE**
 	
UPDATE *table_name*
SET *column1* = *value1*,
 column2 = *value2*, ...
WHERE *condition*;
	UPDATE Student
	SET City= 'Belfast', PostCode= 'BT9 7YY'
	WHERE StudentID = 1001;
	
	The above statement will change the value of the City and Postcode to those shown for the Student whose StudentID is 1001 (see below)
	
	

Insert data into a table - **INSERT INTO**
 	
INSERT INTO
table_name (*column1*,
column2,
column3,etc)
VALUES (*value1*,
value2,
value3,etc)
	To insert two records into the table use the following code:
	
	INSERT INTO Student (StudentID, StudentName, DateOfBirth, Email,Address,City,PostCode)
	VALUES ('1001', 'Anne Browne', '12/12/98', 'AB@mymail.com','1 Long Road', 'Coleraine','BT77 9PP')
	INSERT INTO Student (StudentID, StudentName, DateOfBirth, Email,Address,City,PostCode)
	VALUES ('1002', 'John Gormley' ,'12/5/97', 'jgormley@hitmail.com','37 Palm Ave', 'Limavady', 'BT02 7NN')
	
	
	
	The content of the table is shown below
	
	

Delete data from a table – **DELETE**
 	
DELETE FROM
table_name
WHERE *condition*;
	Delete from Student
	where StudentName='John Gormley';
	The above statement will delete the entry for John Gormley as shown below.

```
CREATE TABLE Student (
   StudentID int not null,
    StudentName varchar(255),
    DateOfBirth Date,
    Email varchar(50),
    Address varchar(255),
    City varchar(255),
    PostCode varchar(8)
       Primary Key (StudentID)
);

--
INSERT INTO Student (StudentID, StudentName, DateOfBirth, Email,Address,City,PostCode)
VALUES ('1001',       'Anne Browne',       '12/12/98',    'AB@mymail.com','1 Long Road',
       'Coleraine','BT77 9PP')
INSERT INTO Student (StudentID, StudentName, DateOfBirth, Email,Address,City,PostCode)
VALUES ('1002',       'John Gormley'       ,'12/5/97',    'jgormley@hitmail.com',     '37 Palm
Ave',  'Limavady',    'BT02 7NN')

--

SELECT StudentName, DateOfBirth  FROM Student;

select * from Student where DateOfBirth< '1/1/1998';


UPDATE Student
SET City= 'Belfast', PostCode= 'BT9 7YY'
WHERE StudentID = 1001;

Delete from Student
where StudentName='John Gormley';
```

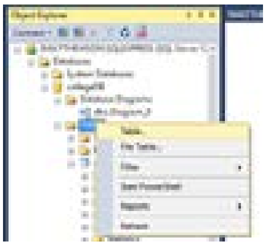
SQL Code for the database created

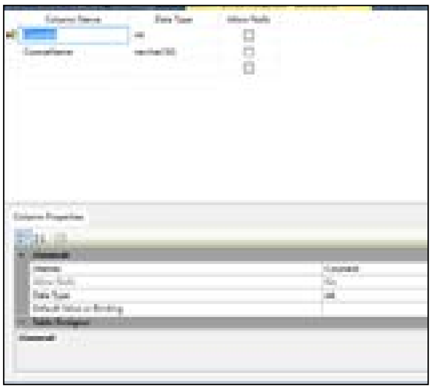
Much of this SQL code can be generated automatically through the Object Explorer in SQL Server Manager (or your chosen SQL tool). This code generated can look a little more complex but the outcome is the same.

Query By Example (QBE)

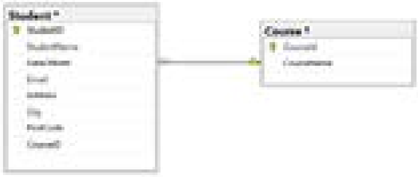
When you ask SQL Server or any other DBMS (including Access) a question about the data in a database, the question is called a query. With QBE users create queries by entering column names and other criteria using an on-screen interface, and data appears on the screen in tabular form. It is easier to create queries in this way, especially for beginners. The SQL for the query is automatically generated. SQL server provides a graphical user interface for creating and modifying tables and relationships and also for generating queries.

For the purposes of this example an additional table has been added (Course) which contains courses studied by the student. This can be done by right clicking on the Table and selecting the Table option from the drop down menu.

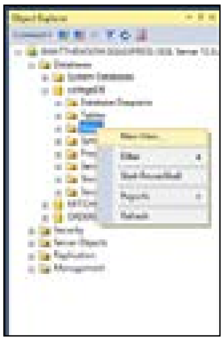




The table design window for creating the new table is very much like that presented by Access. Data types can be selected from the drop dow list and Primary Keys can be selected by right clicking the field.

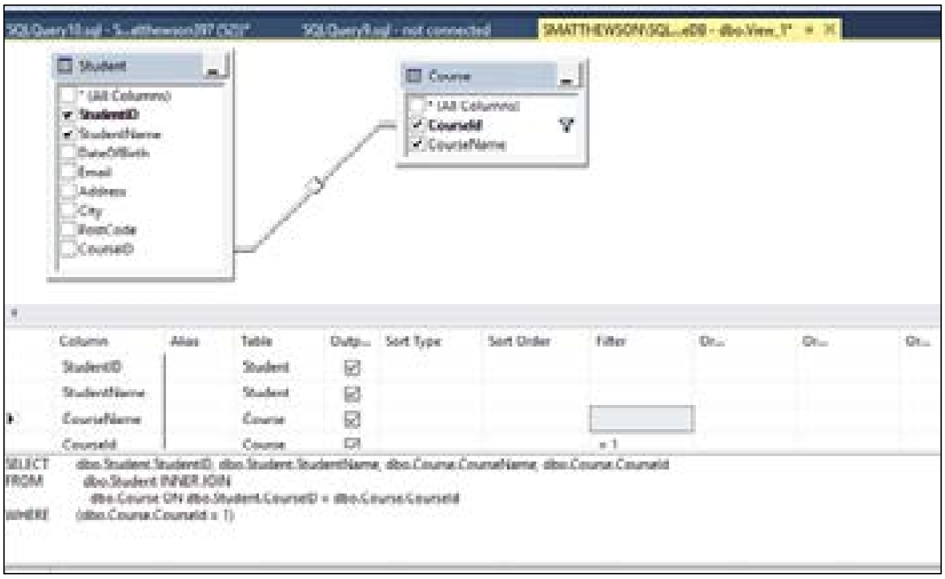


As indicated previously (P3), an additional field (CourseID) has been added to the Student table to facilitate linking the tables together. Relationships can be set up using drag and drop in the Database Diagram object.
CREATE TABLE SQL for the COURSE Table



To create a query using QBE in SQL Server create a new View and add the relevant tables

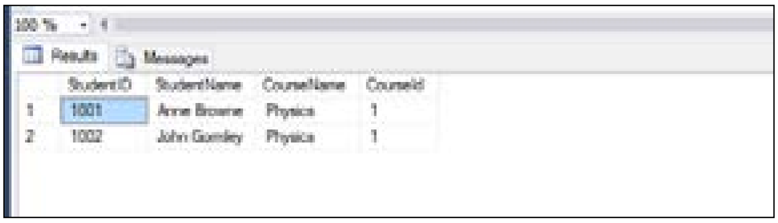
In the example below a new View has been created and the two tables have been added. The query is designed to select all students who are studying the course with CourseID=1. Note that the SQL for the query has been generated automatically.



Column	Alias	Table	Outp...	Sort Type	Sort Order	Filter	Or...	Or...	Or...
StudentID		Student	☑						
StudentName		Student	☑						
CourseName		Course	☑						
CourseId		Course	☑			= 1			

```
SELECT    dbo.Student.StudentID, dbo.Student.StudentName, dbo.Course.CourseName, dbo.Course.CourseId
FROM        dbo.Student INNER JOIN
              dbo.Course ON dbo.Student.CourseID = dbo.Course.CourseId
WHERE     (dbo.Course.CourseId = 1)
```

The results of executing the query can be seen below.



	StudentID	StudentName	CourseName	CourseId
1	1001	Anne Browne	Physics	1
2	1002	John Gormley	Physics	1

Advantages and Disadvantages of QBE

QBE allows inexperienced users to make use of complex database features. Query by Example (QBE) searches through databases filtering data based on data content graphically. This means that the user does not have to know or learn SQL. QBE includes the language necessary to perform the searches in order to achieve results quickly using built-in logical conditions. Microsoft Access uses QBE and can obtain results from complex search queries. QBE can be used for to create and modify data also. Creating a query in Access using QBE and graphical tools to set the query parameters gives a graphical representation of the query. The SQL is automatically generated accurately and will perform an identical query. However, it is not portable in the way that SQL is.

The advantages and disadvantages of each may depend on the technical skills of the user or the time available to create the query. As SQL has industry standards well documented and is used in the relational database industry, this ensures that any query, if properly written, will give the same results across products and therefore is more reliable.