Implementation

Queries

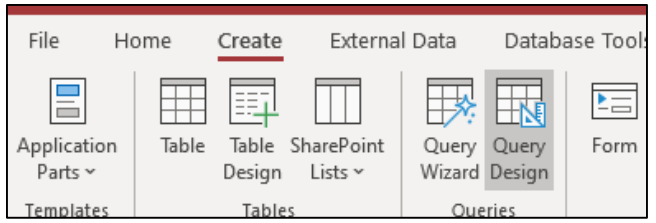
To add, display, change or delete data from a database, we use a special programming language called SQL (sometimes pronounced SEQUEL).

There are four queries that you need to know for National 5:

- SELECT
- INSERT
- UPDATE
- DELETE

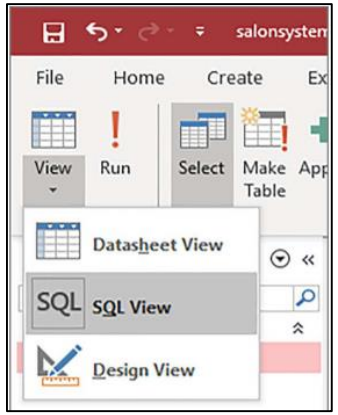
Creating Queries in Microsoft Access

1. Select the Create ribbon and then click Query Design



2. Close the table selection pop up that appears

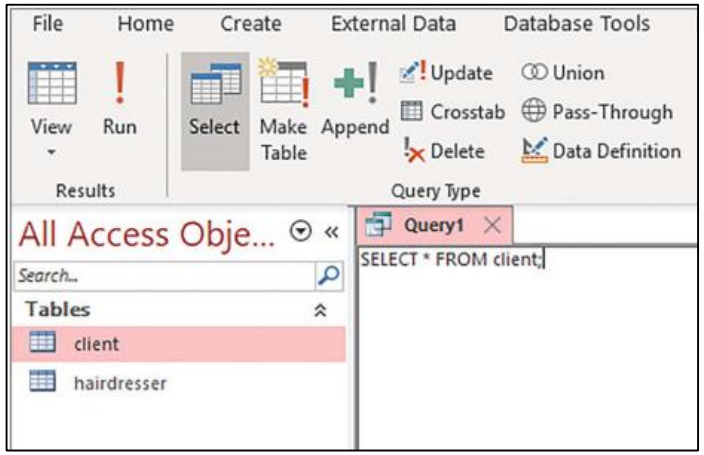
3. Select SQL View



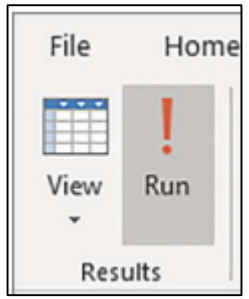
Creating Queries in Microsoft Access

1. H
2. H
3. H

4. Write your SQL in the SQL Editor



5. Once written, execute the query by clicking the Run button



SELECT Query Structure

The SELECT query is used to display information that is stored in the database. The SELECT query has up to four clauses:

SELECT	Mandatory	Defines which attributes are going to be displayed in the output.
FROM	Mandatory	Specifies which entity/entities to get the data from.
WHERE	Optional	Identifies the data required by specifying a condition that the outputted data must meet.
ORDER BY	Optional	Defines the attribute(s) that should be ordered, and how they should be ordered (ascending or descending).

SELECT Query Example

```
SELECT clientfirstname, phonenumber
FROM client
WHERE hairdresserid = 2210
ORDER BY clientfirstname desc;
```

Output:

clientfirstname	phonenumber
Phillip	0131 496 0734
Emily	0207 946 0126

SELECT Query Example

We can also easily create a SELECT query from a design.

Field(s)	clientfirstname, phonenumber
Table(s)	client
Search criteria	hairdresserid = 2210
Sort order	clientfirstname descending

```
SELECT clientfirstname, phonenumber
FROM client
WHERE hairdresserid = 2210
ORDER BY clientfirstname desc;
```

SELECT Query Structure

The asterisk (*) represents all attributes, so SELECT * would display every attribute.

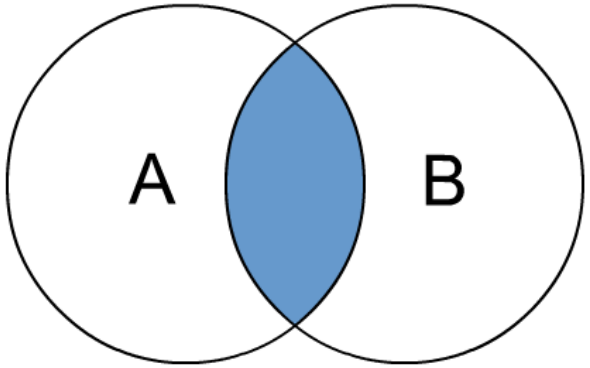
Ascending and descending are written shorthand in SQL queries as ‘asc’ and ‘desc’ respectively.

All SQL queries, including SELECT queries, end with a semi-colon.

SELECT Using EQUI-JOIN

Sometimes you need to combine data from two entities to create the desired output. This is achieved using an **EQUI-JOIN**.

An EQUI-JOIN only displays data from records if there are matching values in both tables, i.e. if the value of the primary key in one table equals the value of the foreign key in the linked table.



EQUI-JOIN Example

The structure of a SELECT query using an EQUI-JOIN is the same as a basic SELECT statement.

To implement an EQUI-JOIN, you need an extra condition in the WHERE clause to check that the **value** of the primary key in one entity equals the **value** of the foreign key in the linked entity, e.g.

```
WHERE hairdresser.hairdresserid = client.hairdresserid;
```

EQUI-JOIN Example

```
SELECT clientfirstname, clientlastname, salon
FROM client, hairdresser
WHERE hairdresser.hairdresserid = client.hairdresserid;
```

Output:

clientfirstname	clientlastname	salon
Wen	Qiu	Cuts & Co
Michael	Waters	Cuts & Co
Faseeha	al-Allam	Cuts & Co
Peta	Mulisdottir	On The Corner
Egisto	Mario	On The Corner
Phillip	Roach	West Style
Emily	Sieff	West Style

INSERT Query Structure

The INSERT query is used to add new records into the database. The INSERT query has two clauses:

INSERT INTO	Mandatory	Defines which entity the record will be entered into, and the attributes that will have data entered.
VALUES	Mandatory	Specifies the data which will be added to the database.

The order of values in the VALUES clause **must match** the order of the attributes in the INSERT INTO clause or the wrong data will be added to the wrong attributes.

INSERT Query Example

```
INSERT INTO client (hairdresserid, clientid,
clientfirstname, clientlastname, phonenumber)
VALUES (2019, 38921, “Rebecca”, “Stewart”,
“01317649127”);
```

Result:

client

hairdresserid	clientid	clientfirstname	clientlastname	phonenumber
1928	10290	Wen	Qiu	0141 496 0536
1928	11766	Michael	Waters	07700 900556
1928	12654	Faseeha	al-Allam	07700 900569
2019	10291	Peta	Mulisdottir	0151 496 0838
2019	20533	Egisto	Mario	0131 496 0294
2019	38921	Rebecca	Stewart	01317649127
2210	36172	Phillip	Roach	0131 496 0734
2210	32100	Emily	Sieff	0207 946 0126

UPDATE Query Structure

The UPDATE query is used to modify information that is stored in the database. The UPDATE query has three clauses:

UPDATE	Mandatory	Defines which entity is going to be updated.
SET	Mandatory	Specifies the attribute(s) that will be modified, and the new value. **Separate multiple values with a comma.**
WHERE	Optional	Identifies the records to be updated by specifying a condition that the data must meet to be changed.

The WHERE clause is **very important** – if you miss it out then you will update every record in the entity. Save a copy of your database before running an UPDATE query just in case!

UPDATE Query Example

```
UPDATE client
SET phonenumber = “01318726418”
WHERE clientid = 11766;
```

Result:

client

hairdresserid	clientid	clientfirstname	clientlastname	phonenumber
1928	10290	Wen	Qiu	0141 496 0536
1928	11766	Michael	Waters	01318726418
1928	12654	Faseeha	al-Allam	07700 900569
2019	10291	Peta	Mulisdottir	0151 496 0838
2019	20533	Egisto	Mario	0131 496 0294
2210	36172	Phillip	Roach	0131 496 0734
2210	32100	Emily	Sieff	0207 946 0126

DELETE Query Structure

The DELETE query is used to remove information from the database. The DELETE query has three clauses:

DELETE	Mandatory	States that a record(s) will be deleted.
FROM	Mandatory	Specifies which entity/entities to remove the data from.
WHERE	Optional	Identifies the data to be deleted by specifying a condition that the data must meet.

The WHERE clause is **very important** – if you miss it out then you will delete every record in the entity. Save a copy of your database before running a DELETE query just in case!

DELETE Query Example

```
DELETE
FROM client
WHERE hairdresserid = 2210;
```

Result:

client

hairdresserid	clientid	clientfirstname	clientlastname	phonenumber
1928	10290	Wen	Qiu	0141 496 0536
1928	11766	Michael	Waters	07700 900556
1928	12654	Faseeha	al-Allam	07700 900569
2019	10291	Peta	Mulisdottir	0151 496 0838
2019	20533	Egisto	Mario	0131 496 0294