

SELECT Statements



The SELECT statement

- Used to query the database
- Retrieves information from the database
- Does not change anything in the database
- Made up of two parts (clauses) the SELECT and the FROM
SELECT information to retrieve
FROM tablename;



SELECT what... FROM where

- We can SELECT column values... FROM a tablename

Show the product name and lines from the products table

```
SELECT productName, productLine  
FROM products;
```

- We can SELECT all values...FROM a tablename

Show all the information in the products table.

```
SELECT *  
FROM products;
```

Both the SELECT and FROM are mandatory in a basic select statement.



Filtering the information

- WHERE – specifies a condition
- The WHERE clause limits the change to the areas that we specifically wanted.
- It acts as a filter
- It is optional (without it all values will be selected or update or deleted)
- It is all about true conditions

Show the product name and line for all products that are motorcycles

```
SELECT productName, productLine  
FROM products  
WHERE productLine = 'Motorcycles';
```



Comparison Operators

- = means equal to
- <> means not equal to
- < means less than
- > means greater than
- <= means less than or equal to
- >= means greater than or equal to

Select all the information for products that are not motorcycles

```
SELECT *  
FROM products  
WHERE productLine <> 'Motorcycles';
```

Select everything that costs less than 40 dollars

```
SELECT *  
FROM products  
WHERE MSRP < 40;
```



What about strings?

Select all the customer information where the last name starts with the letter a and b

```
SELECT *  
FROM customers  
WHERE contactLastName < 'C';
```

- For numbers it is the standard numeric sequence, for strings it is alphabetical.



LIKE operator

- The LIKE operator implements pattern matching

WILDCARDS

- The % looks for one or more characters before
- The _ looks for one character

Select all the customer information for customers who's last name starts with the word Ash

```
SELECT *  
FROM customers  
WHERE contactLastName LIKE 'Ash%';
```



BETWEEN

- Tests a range

Get all the information from the products table for items that can be bought for 100 – 200 dollars

```
SELECT *  
FROM products  
WHERE buyPrice BETWEEN 100 AND 200;
```



COMPOUND CONDITIONS

- The AND ... OR.. Conditions

Select all information where the vendor is Second Gear Diecast for Vintage Cars

```
SELECT *  
FROM products  
WHERE productVendor = 'Second Gear Diecast' AND productLine =  
    'Vintage Cars';
```

returns results only where both conditions are met in the same record

Select all the information for products that are classic cars and vintage cars

```
SELECT *  
FROM products  
WHERE productLine = 'Classic Cars' OR productLine = 'Vintage Cars';
```

Returns results where ***either*** of the conditions are met



Compounding things a little further...

- Combining AND and OR... AND takes precedence over OR

```
SELECT *  
FROM products  
WHERE productLine = 'Classic Cars' OR productLine = 'Planes' AND MSRP < 100;
```

This statement is evaluated like:

```
SELECT *  
FROM products  
WHERE productLine = 'Classic Cars' OR (productLine = 'Planes' AND MSRP < 100);
```

- Use parenthesis to get the results that you want

```
SELECT *  
FROM products  
WHERE (productLine = 'Classic Cars' OR productLine = 'Planes') AND MSRP < 100;
```



IN condition

- Used to check a list of values

```
SELECT *  
FROM products  
WHERE productLine IN ('Planes', 'Trains', 'Ships');
```

- Or you can see what is NOT in a list of values

```
SELECT *  
FROM products  
WHERE productLine NOT IN ('Planes', 'Trains', 'Ships');
```

