# Lab 2 – Displaying Data

## Let's start by creating a new query on the *QAStore* database

* Use the **QAStore** database



* Create a new query
* Click: **File / New / Query with Current Connection**
* Click: **File / Save file As** 02DisplayingDataExercises.sql

## Exercise 2.1 – Simple Query – SELECT

**Exercise 2.1.1 – All Columns**

1. **dept**
2. **salesperson**
3. **sale**
4. **contact**

**Exercise 2.1.2 – Comments**

1. Add a comment line (using the –- characters).
2. An inline comment. (use –- on the same SQL script line)
3. A comment block. (use /\* \*/)

**Exercise 2.1.3 – Specifying Columns**

1. **dept\_no** and **dept\_name** from dept.
2. **emp\_no, last\_name** and **tel** from **salesperson.**
3. **company\_no, order\_no** and **order\_value** from **sale.**
4. **job\_title** and **contact\_name** from **contact.**

**Exercise 2.1.4 – Calculated / Virtual Column and Alias**

1. **dept\_no**, **dept\_name**, **sales\_target** as ‘Old Sales Target’   
   and **sales\_target** increased by 50% as ‘New Sales Target’   
   from **dept.**
2. **order\_no**, **order\_value** as ‘Old Order Value’   
   and **order\_value** reduced by 50% as ‘New Order Value’   
   from **sale.**

**Exercise 2.1.5 – Arithmetic Operators**

1. **first\_name**, **last\_name**, **salary** increased by 50% as ‘New Salary’,   
   **sales\_target** increased by 50% as ‘New Target’ and   
   New Salary \* 10% + New Target \* 20% as ‘Potential Bonus’   
   from **salesperson.**

**Exercise 2.1.6 – Arithmetic Operators using Brackets**

Copy and modify your code from 2.1.5 using brackets to make the calculation clearer if you have not done so already.

**Exercise 2.1.7 – Arithmetic Operator on Text**

1. **company\_no**,   
   **contact\_code, contact\_name, job\_title** with a ‘**/**’ between each field  
   as ‘*Contact Details*’   
   from **contact.**

**Exercise 2.1.8 – DISTINCT**

1. **company\_no** from **contact**.
2. **company\_no** and **contact\_no** from **sale.**

## Exercise 2.2 – Sort Results – ORDER BY

**Exercise 2.2.1 – ORDER BY Ascending**

1. All columns from **dept** sorted by **dept\_no** in ascending order.
2. All columns from **dept** sorted by **manager** in alphabetical order.
3. **first\_name**, **last\_name** and **emp\_no** from **salesperson**   
   sorted by **first\_name** and then **last\_name** in alphabetical order.
4. **order\_date**, **order\_value**, **order\_no** and **emp\_no** from **salesperson** sorted by smallest to biggest **order\_value** within earliest to latest **order\_date.**

**Exercise 2.2.2 – ORDER BY Descending**

1. All columns for **company** sorted by **company\_no** in descending order.
2. All columns for **company** sorted by **company\_name** in reverse alphabetical order.
3. **company\_order\_no**, **order\_no**, **company\_no** and **contact\_code** from **sale** sorted by **company\_order\_no** and then **order\_no** in reverse order.
4. **order\_value**, **order\_date**, **company\_no** and **order\_no** from **sale** sorted by biggest to smallest **order\_value** within latest to earliest **order\_date.**

**Exercise 2.2.3 – ORDER BY Combination**

1. **order\_date**, **order\_value**, **order\_no** and **emp\_no** from **sale**   
   sorted by latest to earliest **order\_date** and then smallest to biggest **order\_value.**
2. **order\_value**, **order\_date**, **company\_no** and **order\_no** from **sale**   
   sorted by biggest to smallest **order\_value** within earliest to latest **order\_date.**

**Exercise 2.2.4 – ORDER BY using Alias**

1. **first\_name** and **last\_name** with a space in between as ‘*Full Name*’,   
   **county** and **post\_code** from **salesperson** sorted by ‘Full Name’.
2. **salary** + 20% \* **sales\_target** as ‘*Potential Pay*’ and **emp\_no**   
   from **salesperson** sorted by biggest to smallest ‘*Potential Pay*’.

## Exercise 2.3 – Filter Results – WHERE

**Exercise 2.3.1 – WHERE using Relational Operator**

1. **dept**, **dept\_name** and **sales\_target** from **dept** with **sales target** bigger than 15 sorted by smallest to biggest **sales\_target.**
2. **dept**, **dept\_name** and **sales\_target** from **dept** with a **sales target** less than or equal to 10 sorted by biggest to smallest **sales\_target.**
3. **order\_no**, **order\_value** and **order\_date** from **sale** with an **order\_date** less than or equal to ’23-July-2000’.
4. **company\_no**, **company\_name** and **county** from **company** where **county** not equal to ‘Devon’.

**Exercise 2.3.2 – WHERE using Logical Operator**

1. **dept**, **dept\_name** and **sales\_target** from **dept** with a **sales\_target** bigger than or equal to 15 and less than or equal to 25 sorted by smallest to biggest **sales\_target.**
2. **dept**, **dept\_name** and **sales\_target** from **dept** with a **sales target** less than or equal to 10 or greater than or equal to 40 sorted by biggest to smallest **sales\_target.**
3. **order\_no**, **order\_value** and **order\_date** from **sale** with an **order\_date** between ’01-July-2000’ and ‘31-Aug-2000’ inclusive.
4. **company\_no**, **company\_name** and **county** from **company** where **company\_name** starts with an ‘H’ through to ‘J’.

**Exercise 2.3.3 – WHERE using BETWEEN**

1. Redo 2.3.2 a, b, c, d using the BETWEEN syntax.

**Exercise 2.3.4 – WHERE using IN**

1. **first\_name**, **last\_name** and **county** from **salesperson**   
   where **county** in list ‘Hampshire, Kent or ‘Surrey’.
2. **first\_name**, **last\_name** and **dept\_no** from **salesperson**   
   where **dept\_no** in list ‘2,3 or 4’.

**Exercise 2.3.5 – WHERE using LIKE**

1. All data from **dept** where **dept\_name** starts with a ‘C’.
2. All data from **dept** where **dept\_name** ends with an ‘S’.
3. All data from **dept** where **dept\_name** has an ‘O’ somewhere in it.
4. All data from **dept** where **dept\_name** has an ‘M’ as the second last letter.

**Exercise 2.3.7 – WHERE using Multiple AND / OR**

1. All data from **sale**   
   where **company\_no** equals 3000 and **order value** is less than 20 or  
   **order\_date** later than or equal to ’01-May-2000’   
   – Note you should just see data for **company\_no** equals 3000.

## Exercise 2.4 – Complex Query

**Exercise 2.4.1 – TOP**

Display the specified data for the given tables:

1. All the data from **dept** for the two smallest **sales\_target.**
2. Display all the data from **dept** for the two biggest **sales\_target.**

**Exercise 2.4.2 – CASE using Value**

All the data from **dept** with an extra column called ‘**Market**’ that is   
determined by the **dept\_no** as follows:

* 1 or 2 = ‘Europe’
* 3 = ‘UK’
* 4 = ‘USA’
* Other = ‘Market Unknown’

Sort the data by increasing **dept\_no.**

**Exercise 2.4.3 – WHERE using NULL**

1. **emp\_no, first\_name, last\_name** and **post\_code**from **salesperson**   
   if **post\_code** is NULL.
2. **emp\_no, first\_name, last\_name** and **post\_code**   
   from **salesperson**   
   if **post\_code** is NOT NULL.
3. **emp\_no, first\_name, last\_name** and **post\_code**from **salesperson**   
   if **post\_code** is not equal to ‘RT8 8LP’.