# Chapter 1-

## Objectives

The primary objective of this lab is to gain familiarisation with the aggregation capabilities of SQL.

## Reference Material

This practical is based on material in the chapter.

## Overview

In this exercise you will use the QAStore database and write SQL that use SQL’s aggregate functions (SUM, MAX, MIN, AVG, COUNT) and the GROUP BY clause.

## Estimated duration

The estimated duration for this lab is 30 minutes.

## Completed solution

A Visual Studio 2010 solution containing the fully completed code for this lab is located in *CoursewareFolder*\1 Summarised Queries \Solutions.

## Step by Step

1. Start SQL Server Management studio from the Windows Start button.
2. Enter .\SQLEXPRESS as the Server name in the Connect to Server dialog box.
3. Choose QAStore from the drop down list of available databases located on the standard toolbar to ensure that it is selected as the current database.
4. Click the New Query button on the standard toolbar.

### Practical 1

### Exercise 1

Using the salesperson table display the sum and average of the sales\_targets as well as a count of the number of salespeople.

If you do it right then the results should look like the following (1 row 3 columns)

Total Average No of sales people

------------ ---------------- ------------------

66.00 11.000000 6

### Exercise 2

Copy your solution to exercise 1 and amend it to display the same information (sales\_target total(sum), sales\_target average and salespeople count) but this time we want the totals to be for each department. Ensure that you also include dept\_no at the start of the SELECT list.

If you do it right then the results should look like the following (3 rows, 4 columns):

dept\_no Total Average No of sales people

-------- ------ ----------- ------------------

1 9.00 9.000000 1

2 21.00 10.500000 2

3 36.00 12.000000 3

### Exercise 3

Users of the output from exercise 2 tell us "it’s a great report but I am not very good on dept numbers can you include dept names AS WELL, and I would like the dept name to appear just to the right of the dept\_no".

So copy/paste the code from exercise 2 and add dept\_name (now it is broken), then repair it.

If you do it right then the results should look like the following (3 rows, 4 columns):

dept dept Total Avg No of sales

no name people

----- ----------------- ------- ---------- ----------

1 Animal Products 9.00 9.000000 1

2 Business Systems 21.00 10.500000 2

3 Credit Control 36.00 12.000000 3

### Note: All you have done is added a single column to the display but you’ve had to make quite a change to the SQL! Nobody said it would be easy!

### Practical 2

### Exercise 4

Look at your output for exercise 3 and answer this question:

“Which depts (names) have more than 1 person?”

Copy and paste your code exercise 3 and amend it to show the answer to the question. (You do need to use your DELETE key for this one!)

If you do it right then the results should look like the following (2 rows, 1 column).

dept\_name

----------------

Business Systems

Credit Control

### Exercise 5

### Write a new query that displays a COUNT of the number of contacts per company name (4 rows).

### You might like to base it on the code that you wrote for an earlier exercise using JOINS:

SELECT CONT.name 'Contact', COMP.name 'Company'

FROM contact CONT INNER JOIN company COMP

ON CONT.company\_no = COMP.company\_no

Contact Company

-------------------- --------------------

Munching Mike Happy Heaters PLC

Naughty Nick Icicle Igloos PLC

Ollie Octopus Icicle Igloos PLC

Purposeful Peter Judo Jeans PLC

Quentin Quail Judo Jeans PLC

Robber Red Judo Jeans PLC

Moody Mo Kipper Kickers Inc

Sammy Snake Kipper Kickers Inc

Terrible Tim Kipper Kickers Inc

Uppy Umbrella Kipper Kickers Inc

If you do it right then the results should look like the following (4 rows).

name TotalContacts

-------------------- -------------

Happy Heaters PLC 1

Icicle Igloos PLC 2

Judo Jeans PLC 3

Kipper Kickers Inc 4

**Exercise 6**

You’re lucky that at the moment all companies have contacts, but in the future some companies may exist (in 'company') to which we have not yet allocated any 'contacts'.

Copy and Paste your code from exercise 5 and ensure that those 'Companies without contacts' also get included in the report by doing an OUTER JOIN (you decide whether it is LEFT or RIGHT).

To ensure that your output now changes, run this INSERT first.

INSERT INTO company

VALUES( 5000, 'ABC Ltd(no contacts)', '(01456)346782', 'Dorset' , 'ST8 3RG' )

If you do it right then the results should look like the following (5 rows).

name TotalContacts

-------------------- -------------

ABC Ltd(no contacts) 0

Happy Heaters PLC 1

Icicle Igloos Inc 2

Judo Jeans PLC 3

Kipper Kickers Inc 4

Did the ‘TotalContacts’ correctly display as '0', or did it say '1'?

If '1' then solve the problem (think about what you are counting)

**Exercise 7**

CREATE a VIEW called 'NoOfContacts' that contains the SELECT statement of exercise 6.

**Exercise 8**

Be the 'user' who uses the VIEW created in exercise 7 by writing a SELECT statement that uses it ordering the result by descending number of contacts.

name TotalContacts

-------------------- -------------

Kipper Kickers Inc 4

Judo Jeans PLC 3

Icicle Igloos Inc 2

Happy Heaters PLC 1

ABC Ltd(no contacts) 0

Recognise how little the 'user' needs to know of SQL, compared to the person who wrote the underlying SQL.

**Clean Up:**

Run the following delete statement:

DELETE FROM company WHERE company\_no = 5000

**Exercise 9**

Write a SELECT statement that produces a report that specifies the total sales (sum of all order values) by employee per company.

If you do it right then the results should look like the following (7 rows).

company\_no emp\_no Total sales

----------- ----------- -----------

1000 10 5

1000 60 7

2000 10 3

2000 60 12

3000 50 27

3000 60 21

4000 60 2