# Practical: SQL – Advanced SELECT statement, Subqueries

## Instructions

Use the same connection as Practical 11.

Refer to the **ORDER** database schema (given in Practical 11). Write the SQL statements for the following queries:

**Aggregate Functions**

1. (i) List the full details of all the Customers. (28 rows selected)

**Select \* from Customer**

(ii) Amend the SQL to output the total number of Customers. (1 row -> 28)

**Select count(\*) from Customer**

1. (i) List the full details of Customers who live in the state ‘California’. (18 rows selected)

**SELECT \* FROM customer where STATE\_CODE ="CA"**

(ii) Amend the SQL to output the total number of Customers who live in the state ‘California’. (1 row -> 18)

**SELECT count(\*) FROM customer where STATE\_CODE ="CA"**

1. (i) List the order number and the shipping charge of those Orders placed by ‘Anthony Higgins’. (4 rows selected)

**select order\_num , ship\_charge from orders inner join customer on customer.CUSTOMER\_NUM = orders.customer\_num where CONCAT(customer.fname, ' ', customer.lname) = "Anthony Higgins";**

(ii) Amend the SQL to output the total number of Orders placed by ‘Anthony Higgins’ and the total shipping charges he has to pay. (1 row -> 4, $38)

**select count(order\_num) , sum(ship\_charge) from orders inner join customer on customer.CUSTOMER\_NUM = orders.customer\_num where CONCAT(customer.fname, ' ', customer.lname) = "Anthony Higgins";**

1. (i) List the unit price of all the Products. (74 rows selected)

**select unit\_price from product**

(ii) Amend the SQL to output the cost of the most expensive Product. (1 row -> $960)

**select max(unit\_price) from product**

1. (i) List the product number and the unit price of all the Products with a description of ‘running shoes’. (6 rows selected).

**SELECT p.prod\_num , p.UNIT\_PRICE FROM product p inner join product\_desc pd on pd.PROD\_NUM = p.PROD\_NUM where pd.PROD\_DESC="running shoes";**

(ii) Amend the SQL to output the count of the different kinds, the most expensive, the least expensive and the average price (round to 2 dec pl) of all the ‘running shoes’ Products. (1 row -> 6, $102, $42.50, $83.08).

**select count(distinct unit\_price) , min(unit\_price) , max(UNIT\_PRICE) , round(avg(UNIT\_PRICE) , 2) from product**

**inner join product\_desc on product.PROD\_NUM = product\_desc.PROD\_NUM**

**where PROD\_DESC = "running shoes";**

**Aggregate Functions, GROUP BY & HAVING clause**

1. (i) As one customer can place one or more orders, list for each customer, the customer’s number and the total number of orders he/she has placed. (17 rows)

**SELECT CUSTOMER\_NUM, COUNT(ORDER\_NUM) AS Total\_Orders**

**FROM ORDERS**

**GROUP BY CUSTOMER\_NUM;**

(ii) Modify your answer, to display also the *customer’s first name and last name*. (17 rows)

**SELECT CUSTOMER.CUSTOMER\_NUM, CUSTOMER.FNAME, CUSTOMER.LNAME, COUNT(ORDERS.ORDER\_NUM) AS Total\_Orders**

**FROM ORDERS**

**JOIN CUSTOMER ON ORDERS.CUSTOMER\_NUM = CUSTOMER.CUSTOMER\_NUM**

**GROUP BY CUSTOMER.CUSTOMER\_NUM , lname fname;**

(iii) Modify your answer, to display only for *those customers who have placed more than 1 orders*. (4 rows -> customer\_num : 104, 106, 110, 117)

**SELECT CUSTOMER.CUSTOMER\_NUM, CUSTOMER.FNAME, CUSTOMER.LNAME, COUNT(ORDERS.ORDER\_NUM) AS Total\_Orders**

**FROM ORDERS**

**JOIN CUSTOMER ON ORDERS.CUSTOMER\_NUM = CUSTOMER.CUSTOMER\_NUM**

**GROUP BY CUSTOMER.CUSTOMER\_NUM , lname fname**

**HAVING COUNT(ORDERS.ORDER\_NUM) > 1;**

1. (i) As one order can consists of one or more products, list for each order, the order’s number and the total number of items/products bought in the order. (23 rows) Hint : Query Order\_detail table.

**SELECT ORDER\_NUM, COUNT(PROD\_NUM) AS Total\_Items**

**FROM ORDER\_DETAIL**

**GROUP BY ORDER\_NUM;**

(ii) Modify your answer, to display also the *total price of all products in each order*. (23 rows)

SELECT ORDER\_NUM, COUNT(PROD\_NUM) AS Total\_Items , sum(total\_price)

FROM ORDER\_DETAIL

GROUP BY ORDER\_NUM;

(iii) Modify your answer, to display only for *those orders that have more than 3 items.* (9 rows)

SELECT ORDER\_NUM, COUNT(PROD\_NUM) AS Total\_Items , sum(total\_price)

FROM ORDER\_DETAIL

GROUP BY ORDER\_NUM

having count(prod\_num) > 3 ;

**Subqueries**

1. By using the answer to question 4 (ii) in a subquery or otherwise, list the product details (i.e. product number, supplier code, unit price and product description) *for the most expensive product*. (1 row -> 4, HSK, 960, football)

**SELECT PRODUCT.PROD\_NUM, PRODUCT.SUPPL\_CODE, PRODUCT.UNIT\_PRICE, PRODUCT\_DESC.PROD\_DESC**

**FROM PRODUCT**

**JOIN PRODUCT\_DESC ON PRODUCT.PROD\_NUM = PRODUCT\_DESC.PROD\_NUM**

**WHERE PRODUCT.UNIT\_PRICE = (SELECT MAX(UNIT\_PRICE) FROM PRODUCT);**

1. By using the answer to question 6 (i) in a subquery (to exclude customers who have placed at least an order) or otherwise, list the customer details (i.e. first name, last name) of those *customers who have not placed any orders*. (11 rows, as follows :)

FNAME LNAME

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

Carole Sadler

Philip Currie

Raymond Vector

Charles Ream

Donald Quinn

Jane Miller

Lana Beatty

Frank Albertson

Dick Baxter

James Henry

Frank Lessor

**SELECT FNAME, LNAME**

**FROM CUSTOMER**

**WHERE CUSTOMER\_NUM NOT IN (SELECT CUSTOMER\_NUM FROM ORDERS);**

1. By using the answer to question 7 (iii) in a subquery or otherwise, list the order details (i.e. order number, item number, product number, quantity) *for those orders that have more than 3 items*. (41 rows -> order num : 1004, 1005, 1006, 1007, 1013, 1016, 1018, 1021, 1023)

SELECT ORDER\_NUM, ITEM\_NUM, PROD\_NUM, QUANTITY

FROM ORDER\_DETAIL

WHERE ORDER\_NUM IN (

SELECT ORDER\_NUM

FROM ORDER\_DETAIL

GROUP BY ORDER\_NUM

HAVING COUNT(PROD\_NUM) > 3

);