**Advanced Databae Services**

# Assignment 1

## Submission 1

**Your submission will be a single text-based SQL file (.sql) with appropriate header and commenting. Please ensure your file runs when the entire file is executed in SQL Developer. A submission format file(.sql) is provided, you can fill your code in this format file.** Name your submission file as **A1\_Group#.sql**.  
Make sure only one submission per group.

## Style Guide

Your SQL should be written using the standard coding style:

* all keywords are to be upper case,
* all user-defined names are to be lower case, (example: table and field names)
* there should be a carriage return before each major part of the SQL statements (i.e. before SELECT, FROM, WHERE and ORDER BY)

See the following sample:

**SELECT** columns

**FROM** tables

**WHERE** conditions

**ORDER** **BY** column1,column2;

## Submission Checklist Use the following checklist, to make sure you have completed the assignment successfully.

|  |  |  |
| --- | --- | --- |
| **Tasks to be completed** | **Yes** | **No** |
| You have read the assignment group submission and completion policies and all instructions provided in the assignment document and have not missed a word. |  |  |
| Student information and the assignment information have been added to the header of the submission. (Same as the template provided in the assignment documents) |  |  |
| All questions are answered in a text file (a SQL worksheet) and are saved as a ***.sql*** file. |  |  |
| Comments are included. (questions definition or any additional explanation) |  |  |
| All SQL statements are executed successfully without errors. (Use "Run Script" to execute all statements together.) |  |  |

## Group Work

This assignment is to be completed in groups of 3. Please only one submission per group. The comment header MUST have all students’ name and student number.

It is suggested that you **ALL do it individually** and then meet to compare answers. Those not doing the work may be barred from your group resulting in a zero and incomplete on the assignment.

## Assignment Marking Scheme

|  |  |  |  |
| --- | --- | --- | --- |
| **Question** | **Weight** | **Question** | **Weight** |
| 1 | 10 | 6 | 10 |
| 2 | 10 | 7 | 10 |
| 3 | 10 | 8 | 10 |
| 4 | 10 | 9 | 10 |
| 5 | 10 | 10 | 10 |

## VERY IMPORTANT:

Being part of a group is the same as being a part of a team for these assignments. When you submitted your work as part of a group, you are saying that:

* You understood what was submitted and that you fully participated with ALL the group members.
* It does not mean letting others do your work for you.
* It does not mean watching the others do the work.
* For your full participation, you get a mark equal to all the others in the group.
* If on the test, which is very much like the assignment, you cannot answer it strongly indicates that you did not participate and understand the assignment but depended on others for the mark you received. That is very much like submitting their work and claiming it is your work.

## Example Submission (a submission format file is provided)

|  |
| --- |
| -- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* -- Student1 Name: Your Name Student1 ID: ######### -- Student2 Name: Your Name Student2 ID: ######### -- Student3 Name: Your Name Student3 ID: ######### -- Date: The date of assignment completion -- Purpose: Assignment 1 - DBS311 -- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  -- Question 1 – write a brief note about what the question is asking -- Q1 SOLUTION --  SELECT order\_id AS "order id", COUNT(item\_id) AS "number of items" FROM order\_items WHERE order\_id < 5 GROUP BY order\_id ORDER BY order\_id;    order id  number of items ------------   ---------------------          1              13          2               9          3               8          4               8 |

## Tasks

For each question, the columns’ title and the format of the output result **must** match the sample output columns given in that question.

**In your .sql file, include the SQL query and the output result for each question. Put your output result in a comment block.**

1. Hired

Write a query to display employee ID, first name, last name, and hire date for employees who have been hired after the last employee hired in April 2016 but two months before the first employee hired in August 2016.

Sort the result by hire date and employee ID.

**The query returns 6 rows.**

See the output columns:

EMPLOYEE\_ID    FIRST\_NAME    LAST\_NAME   HIRE\_DATE  
----------- ----------    --------- ----------

1. Direct Reports

Display manager ID for managers with more than one direct employee. Answer this question **without** using the COUNT()function.  
Sort the result by manager ID.  
The query returns 15 rows. See the output columns:

Manager ID  
----------

1. Rest

Use your query in Q2 and SET Operator(s) to display manager ID for managers who have only one direct employee. You are still not allowed to use COUNT().  
Sort the result by manager ID.   
The query returns 3 rows. See the output columns:

Manager ID  
----------

1. Frequent Ordered Products

Write a SQL query to display products that have been ordered multiple times (in different orders) on the same day in 2015.  
Display product ID, order date, and the number of times the product has been ordered on that day.  
Sort the result by order date and product ID.  
The query returns 2 rows. See the following output columns:

Product ID   Order Date     Number of orders  
----------   ----------     ----------------

1. Purchased

Write a query to display customer ID and customer name for customers who have purchased all these three products: Products with ID 31, 205, 275.  
Sort the result by customer ID.  
The query returns 1 row. See the following output columns:

CUSTOMER ID     NAME                                        
-----------     ----

1. Salesman

Write a query to display employee ID and the number of orders for employee(s) with the maximum number of orders (sales). **Hint**: A salesman is an employee.  
Sort the result by employee ID.   
The query returns 1 row. See the following output:

Employee ID     Number of Orders  
-----------     ----------------

1. Order Amount

Write a query to display the month number, month name, year, total number of orders, and total sales amount for each month in 2016.  
Sort the result according to month number.  
The query returns 9 rows. See the output result as follows.

Month Number Month           Year Total Number of Orders Sales

           2 February        2016                      3    996895.71

           5 May             2016                      2    1264918.9

           6 June            2016                      7   3334935.14

           7 July            2016                      1    616763.19

           8 August          2016                      5   3665979.49

           9 September       2016                     10   3776557.12

          10 October         2016                      9   2700781.78

          11 November        2016                      5   2148981.02

          12 December        2016                      8   2983793.75

1. Monthly Sales

Write a query to display month number, month name, and average sales amount (per order) for each month in 2016 where the average sales amount is greater than the average sales amount (per order) for the entire year.  
Round the average amount to two decimal places.  
Sort the result by the month number.

HINT: In this query, you will calculate the average sales amount for each month in 2016 and compare it to the overall average sales amount for the entire year. Using a **WITH** clause will simplify your code greatly.

The query returns 5 rows. See the output result as follows:

Month Number Month     Average Sales Amount

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

           5 May                  632459.45

           6 June                 476419.31

           7 July                 616763.19

           8 August                733195.9

          11 November              429796.2

1. Employees

Write a query to display first names in EMPLOYEES that start with letter ‘A’ but do not exist in CONTACTS.

Sort the result by first name.

The query returns 2 rows. See the output column as follows.

First Name            
-----------

1. Calculation

Write a query to generate the following output with the calculated values filled in.

OUTPUT

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

The number of employees with total order amount over average order amount: x

The number of employees with total number of orders greater than 10: x

The number of employees with no order: xx

The number of employees with orders: x

Average order amount is the average amount during salesman. While calculating the average order amount, you should exclude the orders that without a salesman.

Hint: Using a **WITH** clause will simplify your code.

GOOD LUCK!