Question 3 Queries & Results

```
# USE Ass1_Q3;
# 1 - Write a query that displays the following statistics for each country:
    #  Total number of customers
    #  Total number of male customers
    #  Total number of female customers
    #  Percent of all customers that are male (Percent Male).
SELECT Country, Total_Customers, Male, (Total_Customers - Male) as Female, (Male / Total_Customers) * 100 as Percent_Male
FROM
(SELECT male_count.Country, male_count.male, all_count.Total_Customers
FROM
(SELECT Country, Gender, COUNT(*) as Male
FROM customer
GROUP BY  Gender, Country
HAVING Gender = 'M') male_count
INNER JOIN
(SELECT Country, COUNT(*) as Total_Customers FROM customer GROUP BY Country) all_count
ON male_count.Country = all_count.Country
) combined
```

Result Grid

Country	Total_Customers	male	Female	Percent_Male
ZA	4	1	3	25.0000
CA	15	7	8	46.6667
US	28	15	13	53.5714
AU	8	5	3	62.5000
DE	10	7	3	70.0000
IL	5	5	0	100.0000
TR	7	7	0	100.0000

Figure1: Q3/1 Expression & Output

```
# 2 - Create a result by combining two tables.
    #  Include columns Product_ID, Product_Name from product_dim table.
    #  Include a column with the label Total Sold. Use a summary function to create
    #       this column, which displays the quantity sold for each product.
    # Specifies the tables product_dim, with the alias p and order_fact with the alias o
    # Join the tables by matching the values of the appropriate columns in each table.
    # Groups the results by Product_ID from product_dim table and Product_Name.
    # Orders the rows so that products with the highest number sold appear at the
    #       top of the report and then by Product_Name.

SELECT p.Product_ID, p.Product_NAME, SUM(o.Quantity) as Total_Sold
FROM product_dim p
INNER JOIN order_fact o
ON p.Product_ID = o.Product_ID
GROUP BY p.Product_ID, p.Product_Name
ORDER BY Total_Sold DESC;
```

Result Grid

Product_ID	Product_NAME	Total_Sold
230100600022	Expedition10,Medium,Right,Blue Ribbon	9
230100600016	Expedition Zero,Medium,Right,Charcoal	8
230100700011	Hurricane 4	8
230100500056	Knife	8
230100600030	Outback Sleeping Bag, Large,Left,Blue/Black	8
240700100001	Armour L	8
220101400387	N.d.gear Cap	7
230100600031	Outback Sleeping Bag, Large,Right, Blue/Black	7
240500100017	A-team Sweat Round Neck, Small Logo	7
240700200010	Bat - Home Run S	7
240400300035	Smasher Shorts	7
240200200039	Eagle Plain Cap	6

Figure2: Q3/2 Expression & Output – Displays 12 of 481

```
# 3- Display result of all trainees and workers at company. For each trainee or temporary worker,
#       the report should include the employee ID, name and job title, and manager ID and name.
#       The report should be ordered by Employee_ID.

SELECT emp.Employee_ID, emp.Employee_Name, emp.Job_Title, emp.Manager_ID,employee_addresses.Employee_Name as Manager_Name
FROM
(SELECT s.Employee_ID, s.Job_Title, e.Employee_Name, s.Manager_ID
FROM staff s
INNER JOIN employee_addresses e
ON s.Employee_ID = e.Employee_ID
WHERE s.Job_Title in ( 'Trainee', 'Temp. Sales Rep.')) emp
INNER JOIN employee_addresses
ON emp.Manager_ID = employee_addresses.Employee_ID
ORDER BY Employee_ID;
```

Employee_ID	Employee_Name	Job_Title	Manager_ID	Manager_Name
120181	Cantatore, Lorian	Temp. Sales Rep.	120103	Dawes, Wilson
120182	Barreto, Geok-Seng	Temp. Sales Rep.	120103	Dawes, Wilson
120183	Blanton, Brig	Temp. Sales Rep.	120103	Dawes, Wilson
120184	Moore, Ari	Temp. Sales Rep.	120103	Dawes, Wilson
120185	Bahlman, Sharon	Temp. Sales Rep.	120103	Dawes, Wilson
120186	Quinby, Merryn	Temp. Sales Rep.	120103	Dawes, Wilson
120187	Catenacci, Reyne	Temp. Sales Rep.	120103	Dawes, Wilson
120188	Baran, Shanmuganathan	Temp. Sales Rep.	120103	Dawes, Wilson
120189	Lachlan, Mihailo	Temp. Sales Rep.	120103	Dawes, Wilson
120190	Czernezkyi, Ivor	Trainee	120103	Dawes, Wilson
120191	Graham-Rowe, Jannene	Trainee	120103	Dawes, Wilson
120192	Nicholas, Anthony	Trainee	120103	Dawes, Wilson

Figure3: Q3/3 Expression & Output – Displays 12 of 42

```
# 4- 1) Calculate the difference between the salary of the current row and the previous row
SET @quot=NULL;
SELECT Employee_ID, Salary, lag_salary, (Salary - lag_salary) as Lag_diff
FROM
(SELECT Employee_ID,Salary,@quot lag_salary, @quot:=salary curr_salary
FROM employee_payroll
ORDER BY Employee_ID) a;


#    2) Calculate the difference between the salary of current row and the following row.
SELECT t1.Employee_ID, t1.Salary, t2.Salary as Lead_Salary, (t2.Salary - t1.Salary) as Lead_diff
FROM (
    SELECT Employee_ID, Salary, @rownum1 := @rownum1 + 1 AS id
    FROM
     employee_payroll  , (SELECT @rownum1 := 0) r
    ORDER BY
```

Employee_ID	Salary	lag_salary	Lag_diff
120103	87975	108255	-20280
120104	46230	87975	-41745
120105	27110	46230	-19120
120106	26960	27110	-150
120107	30475	26960	3515
120108	27660	30475	-2815
120109	26495	27660	-1165
120110	28615	26495	2120
120111	26895	28615	-1720
120112	26550	26895	-345
120113	26870	26550	320

Figure3: Q3/4-1 Expression & Output – Displays 12 of 424

```
#    2) Calculate the difference between the salary of current row and the following row.
SELECT t1.Employee_ID, t1.Salary, t2.Salary as Lead_Salary, (t2.Salary - t1.Salary) as Lead_diff
FROM (
    SELECT Employee_ID, Salary, @rownum1 := @rownum1 + 1 AS id
    FROM
     employee_payroll  , (SELECT @rownum1 := 0) r
    ORDER BY
     employee_payroll.Employee_ID ASC
) AS t1
LEFT OUTER JOIN (
    SELECT Employee_ID, Salary, @rownum2 := @rownum2 + 1 AS id
    FROM
     employee_payroll, (SELECT @rownum2 := 0) r
    ORDER BY
     employee_payroll.Employee_ID ASC
) AS t2
  ON t1.id +1 = t2.id;
```

Employee_ID	Salary	Lead_Salary	Lead_diff
120101	163040	108255	-54785
120102	108255	87975	-20280
120103	87975	46230	-41745
120104	46230	27110	-19120
120105	27110	26960	-150
120106	26960	30475	3515
120107	30475	27660	-2815
120108	27660	26495	-1165

Result 13

Figure4: Q3/4-2 Expression & Output – Displays 8 of 424