Task 6: SQL Queries and Reports

6.1 Choo Zhen Hao

6.1.1 Query/Report 1: Report of Invoice issued on a certain date

Purpose: The purpose of this query is to find out the invoice issued on a certain date

SQL Statement:

```
SET LINESIZE 120
SET pagesize 100

VARIABLE v_date VARCHAR2 (10);

PROMPT 'Enter the invoice date in DD/MM/YYYY format:'

COLUMN customer_name FORMAT A30 HEADING 'Customer Name'
COLUMN invoice_id FORMAT A15 HEADING 'Invoice ID'
COLUMN order_date  FORMAT A15 HEADING 'Order Date'
COLUMN total_amount HEADING 'Total Amount (MYR)'
COLUMN customer_id  FORMAT A15 HEADING 'Customer ID'

TTITLE LEFT 'Invoice Report ' SKIP 1

SELECT i.invoice_id, i.order_date, i.total_amount, c.customer_id
FROM invoice i
INNER JOIN customer c ON i.customer_id = c.customer_id
WHERE i.order_date like '&v_date';
CLEAR COLUMNS
CLEAR BREAKS
TTITLE OFF;
```

Sample output:

```
SQL> start  C:\Users\TARUC\Desktop\query1.txt
'Enter the invoice date in DD/MM/YYYY format:'
Enter value for v_date: 28/10/2021
old   4: WHERE i.order_date like '&v_date'
new   4: WHERE i.order_date like '28/10/2021'

Invoice Report
Invoice ID      Order Date      Total Amount (MYR) Customer ID
--------------- --------------- ------------------ ---------------
V000017         28/10/2021                   1665 C0000017
V000033         28/10/2021                   2394 C0000033
```

6.1.2 Query/Report 2: Inventory level report of warehouses

Purpose : To show the current inventory levels for each product in each warehouse . It can help to identify which products are running low on inventory and need to be restocked

SQL Statement:

```
SET LINESIZE 120
SET PAGESIZE 30

COLUMN product_id FORMAT A20 HEADING 'Product ID'
COLUMN product_name FORMAT A40 HEADING 'Product Name'
COLUMN warehouse_id FORMAT A30 HEADING 'Warehouse ID'
COLUMN stock_quantity FORMAT 9999 HEADING 'Stock Quantity'
```

```
TTITLE CENTER 'Current Inventory Levels by Product and Warehouse'
SKIP 2

SELECT p.product_id, p.product_name, w.warehouse_id, s.stock_quantity
FROM product p
JOIN stock s ON p.product_id = s.product_id
JOIN warehouse w ON s.warehouse_id = w.warehouse_id
ORDER BY p.product_id, w.warehouse_id;
```

Sample output:

```
SQL> start C:\Users\TARUC\Desktop\query2.txt

                              Current Inventory Levels by Product and Warehouse

Product ID          Product Name                           Warehouse ID                Stock Quantity
------------------- -------------------------------------- --------------------------- --------------
P0000002            Kleenex Facial Tissue                  W0000002                            10
P0000003            Colgate Toothpaste                     W0000003                             2
P0000004            Scott Toilet Paper                     W0000004                           219
P0000006            Pantene Shampoo                        W0000006                           112
P0000007            Pears Soap                             W0000007                           232
P0000008            Rexona Deodorant                       W0000005                           212
P0000008            Rexona Deodorant                       W0000008                           423
P0000010            Oral-B Toothbrush                      W0000010                           143
P0000012            Ketchup                                W0000012                           222
P0000014            Pepsi                                  W0000014                            50
P0000015            Sprite                                 W0000013                            90
P0000015            Sprite                                 W0000015                            40
P0000016            Fanta                                  W0000016                           200
P0000017            Doritos                                W0000017                           150
P0000018            Pringles                               W0000018                           750
P0000020            Tostitos                               W0000020                           910

16 rows selected.
```

6.1.3 Query/Report 3: Report of total sales to customers in a year

Purpose: To show the total sales for each customer in a year. This helps to identify the top / most valuable customers to our company.

SQL Statement:

```
SET LINESIZE 120
SET PAGESIZE 30
SET VERIFY OFF
ACCEPT enter_year PROMPT 'Enter the year to generate sales report: '

COLUMN customer_id FORMAT A20 HEADING 'Customer ID'
COLUMN sales_year FORMAT 9999 HEADING 'Year'
COLUMN total_sales FORMAT 999999.99 HEADING 'Total Sales (RM)'

TTITLE LEFT 'Sales Report for Year &enter_year' SKIP 1

SELECT c.customer_id, TO_CHAR(i.order_date, 'YYYY') AS sales_year,
SUM(i.total_amount) AS total_sales
FROM customer c
JOIN invoice i ON c.customer_id = i.customer_id
```

```
WHERE TO_CHAR(i.order_date, 'YYYY') = '&enter_year'
GROUP BY c.customer_id, TO_CHAR(i.order_date, 'YYYY');
```

Sample output:

```
SQL> start C:\Users\TARUC\Desktop\query3.txt
Enter the year to generate sales report: 2021

Sales Report for Year 2021
Customer ID          Year Total Sales (RM)
-------------------- ---- ----------------
C0000005             2021           929.00
C0000023             2021          2052.00
C0000007             2021           554.00
C0000019             2021          1378.00
C0000042             2021          2654.00
C0000048             2021          1196.00
C0000017             2021          1665.00
C0000021             2021           719.00
C0000029             2021          1825.00
C0000031             2021           764.00
C0000033             2021          2394.00
C0000036             2021           924.00
C0000015             2021          2026.00
C0000027             2021           980.00
C0000004             2021           829.00
C0000008             2021           494.00
C0000026             2021          1934.00
C0000035             2021          2547.00
C0000044             2021           219.00

19 rows selected.
```

6.2 Wong Kai Chen

6.2.1 Query/Report 1: Detail report of the total amount of the products available in one warehouse

Purpose: The purpose of this report is to inform about low quantity of products in the warehouse

SQL statement:

```
PROMPT 'Total amount of products in a single warehouse location'
PROMPT
PROMPT

COLUMN warehouse_id HEADING 'Warehouse'
COLUMN location_name HEADING 'Location Name'
COLUMN total_products HEADING 'Amount'


TTITLE LEFT 'Total Amount of Products in a Warehouse Location'

SELECT w.warehouse_id, l.location_name, COUNT(s.product_id) as
total_products
FROM warehouse w
JOIN location l ON w.location_id = l.location_id
LEFT JOIN stock s ON w.warehouse_id = s.warehouse_id
GROUP BY w.warehouse_id, l.location_name
ORDER BY total_products DESC;


CLEAR COLUMNS
CLEAR BREAKS
TTITLE OFF
```

Sample output:

```
'Total amount of products in a single warehouse location'



Total Amount of Products in a Warehouse Location
Warehouse Location Name                Amount
--------- ------------------------ ----------
W0000005  Alor Setar                        1
W0000012  Petaling Jaya                     1
W0000003  Batu Pahat                        1
W0000017  Jasin                             1
W0000007  Keningau                          1
W0000008  Alor Setar                        1
W0000010  Tumpat                            1
W0000016  Kemaman                           1
W0000018  Kangar                           1
W0000020  Setiawangsa                       1

Total Amount of Products in a Warehouse Location
Warehouse Location Name                Amount
--------- ------------------------ ----------
W0000002  Batu Gajah                        1
W0000004  Sarikei                          1
W0000006  Bagan Serai                       1
W0000015  Bentong                          1
W0000014  Kota Tinggi                       1
W0000013  Bentong                          1
W0000009  Lumut                            0
W0000001  Port Dickson                      0
W0000019  Bangi                            0
W0000011  Wangsa Maju                       0

20 rows selected.
```

6.2.2 Query/Report 2: Detail report of customers that purchased the most

Purpose: The purpose of this report is to inform about who's the most loyal customer and what is the maximum of a person willing to spend on our company

SQL statement:

```
PROMPT 'Most order purchased by customers'
PROMPT
PROMPT

COLUMN customer_id HEADING 'Customer ID'
COLUMN cust_name HEADING 'Customer Name'
COLUMN total_purchase_amount HEADING 'Total Purchase Amount'

TTITLE LEFT 'Most order purchased by customers'
```

```
  SELECT c.customer_id, c.cust_name, SUM(sod.total_price) as
total_purchase_amount
  FROM customer c
  JOIN sales_order so ON c.customer_id = so.customer_id
  JOIN sales_order_detail sod ON so.order_id = sod.order_id
  GROUP BY c.customer_id, c.cust_name
  ORDER BY total_purchase_amount DESC;

TTITLE LEFT 'Most order purchased by customers'

SELECT *
FROM (
  SELECT c.customer_id, c.cust_name, SUM(sod.total_price) as
total_purchase_amount
  FROM customer c
  JOIN sales_order so ON c.customer_id = so.customer_id
  JOIN sales_order_detail sod ON so.order_id = sod.order_id
  GROUP BY c.customer_id, c.cust_name
  ORDER BY total_purchase_amount DESC
)
WHERE ROWNUM = 1;


CLEAR COLUMNS
CLEAR BREAKS
TTITLE OFF
```

Sample output:

```
SQL> start C:\query3s.txt
'Most order purchased by customers'



Most order purchased by customers
Customer  Customer Name              Total Purchase Amount
--------- ------------------------- --------------------
C0000002  Ramli Jani                                5155.7
C0000006  Jing Qi                                  4948.06
C0000005  Wei Han                                  4468.47
C0000003  Kingston                                 3872.37
C0000004  Sandy                                    3641.72
C0000007  Tang Jie                                 3119.17
C0000010  Shanti Kaur                              2736.43
C0000001  Mat Sambas                              2007.09
C0000008  Intan Wahida                            1558.14
C0000009  Naidu                                   1501.64

10 rows selected.


Most order purchased by customers
Customer  Customer Name              Total Purchase Amount
--------- ------------------------- --------------------
C0000002  Ramli Jani                                5155.7
```

6.2.3 Query/Report 3: Detail report of the total amount of products sold

Purpose: The purpose of this report is to inform about which product made the most sales and if the decision to work together with the other company is beneficial or not.

SQL statement:

```
PROMPT 'Total amount of products'
ACCEPT v_numberRows NUMBER DEFAULT 10 PROMPT 'Enter Row: '

COLUMN product_name HEADING 'Product Name'
COLUMN total_ordered HEADING 'Amount ordered'
```

```
TTITLE LEFT 'Total Amount of Products'

SELECT p.product_name, SUM(sod.quantity) AS total_ordered
FROM sales_order_detail sod
JOIN product p ON sod.product_id = p.product_id
WHERE ROWNUM <= '&v_numberRows'
GROUP BY p.product_name
ORDER BY total_ordered DESC;



CLEAR COLUMNS
CLEAR BREAKS
TTITLE OFF
```

Sample output:

```
'Total amount of products'
Enter Row: 16
old   4: WHERE ROWNUM <= '&v_numberRows'
new   4: WHERE ROWNUM <= '        16'

Total Amount of Products
Product Name               Amount ordered
-------------------------- --------------
Rexona Deodorant                    97471
Scott Toilet Paper                  92251
Tostitos                            86758
Ketchup                             82377
Kleenex Facial Tissue               77450
Pringles                            75741
Doritos                             74977
Oral-B Toothbrush                   65725
Lifebuoy Hand Sanitizer             62560
Pantene Shampoo                     59385

Total Amount of Products
Product Name               Amount ordered
-------------------------- --------------
Sprite                              41072
Colgate Toothpaste                  31399
Coca-Cola                           29129
Fanta                               14194
Pepsi                               13000
Pears Soap                          12964

16 rows selected.
```

6.3 Leong Hon Yan

6.3.1 Query/Report 1: Report of customer information

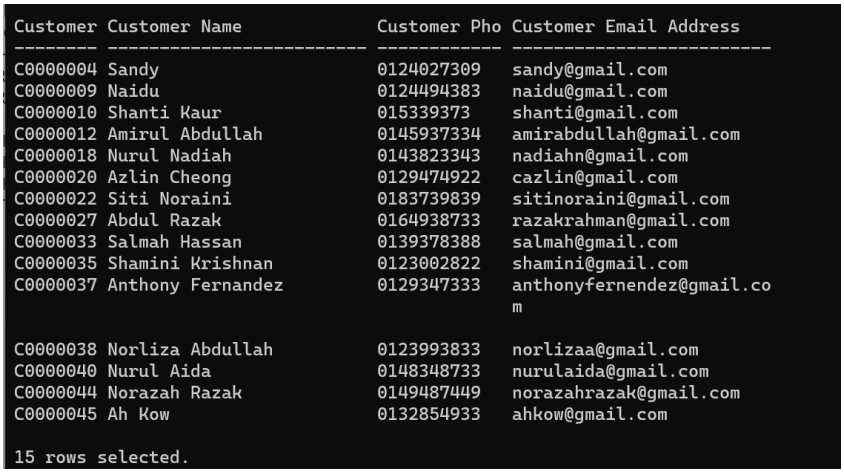
Purpose: Find the information of the customer based on the alphabet that had been selected.

SQL statement:

```
SET LINESIZE 120
SET PAGESIZE 30

TTITLE CENTRE '----Sort customer name by A, S and N—- '


COLUMN customer_id FORMAT A8 HEADING "Customer ID";
COLUMN cust_name FORMAT A25 HEADING "Customer Name";
COLUMN cust_phone FORMAT A12 HEADING "Customer Phone Number";
```

```
COLUMN cust_email FORMAT A25 HEADING "Customer Email Address";

SELECT  c.customer_id, c.cust_name, c.cust_phone, c.cust_email
FROM customer c
WHERE cust_name LIKE 'A%'
OR cust_name LIKE 'S%'
OR cust_name LIKE 'N%';

CLEAR COLUMNS
CLEAR BREAKS
CLEAR COMPUTES
TTITLE OFF
```

Sample output:



```
Customer Customer Name            Customer Pho Customer Email Address
-------- ------------------------ ------------ ------------------------
C0000004 Sandy                    0124027309   sandy@gmail.com
C0000009 Naidu                    0124494383   naidu@gmail.com
C0000010 Shanti Kaur              015339373    shanti@gmail.com
C0000012 Amirul Abdullah          0145937334   amirabdullah@gmail.com
C0000018 Nurul Nadiah             0143823343   nadiahn@gmail.com
C0000020 Azlin Cheong             0129474922   cazlin@gmail.com
C0000022 Siti Noraini             0183739839   sitinoraini@gmail.com
C0000027 Abdul Razak              0164938733   razakrahman@gmail.com
C0000033 Salmah Hassan            0139378388   salmah@gmail.com
C0000035 Shamini Krishnan         0123002822   shamini@gmail.com
C0000037 Anthony Fernandez        0129347333   anthonyfernendez@gmail.co
                                               m

C0000038 Norliza Abdullah         0123993833   norlizaa@gmail.com
C0000040 Nurul Aida               0148348733   nurulaida@gmail.com
C0000044 Norazah Razak            0149487449   norazahrazak@gmail.com
C0000045 Ah Kow                   0132854933   ahkow@gmail.com

15 rows selected.
```

6.3.2 Query/Report 2: Location report

Purpose: Find the location id in the Klang Valley (KL, Selangor and Putrajaya) based on the selected postcode .

SQL statement:

```
SET LINESIZE 120
SET PAGESIZE 30

TTITLE CENTRE '----Location in Klang Valley—-'

ACCEPT l_location_id CHAR FORMAT  A8 PROMPT 'Enter location ID: '
ACCEPT l_location_name CHAR FORMAT  A25 PROMPT 'Enter location name:
'
```

```
ACCEPT l_address_post_code CHAR FORMAT  A30 PROMPT 'Enter post code:
'


COLUMN location_id FORMAT A8 HEADING "Location ID";
COLUMN location_name FORMAT A25 HEADING "Location Name";
COLUMN address_post_code FORMAT A30 HEADING "Post Code";


SELECT l.location_id, l.location_name, l.address_post_code
FROM location l
WHERE l.address_post_code BETWEEN '40000' AND '69999'
ORDER BY l.address_post_code DESC ;

CLEAR COLUMNS
CLEAR BREAKS
CLEAR COMPUTES
TTITLE OFF
```

Sample output:

```
Enter location ID: L000001
Enter location name: Port Dickson
Enter post code: 71000

Location Location Name             Post Code
-------- ------------------------ ------------------------------
L000020  Setiawangsa              54200
L000011  Wangsa Maju              53300
L000012  Petaling Jaya            47800
L000019  Bangi                    43650
```

6.3.3 Query/Report 3: Warehouse and location report

Purpose: Find the location id based on the range of the warehouse id that had been selected.

SQL statement:

```
SET LINESIZE 120
SET PAGESIZE 30
```

```
TTITLE CENTRE '----List of Location based on location id--'

ACCEPT w_warehouse_id CHAR FORMAT  A8 PROMPT 'Enter warehouse ID: '
ACCEPT l_location_id CHAR FORMAT  A8 PROMPT 'Enter location ID: '

COLUMN warehouse_id FORMAT A8 HEADING "Warehouse ID";
COLUMN location_id FORMAT A8 HEADING "Location ID";

SELECT l.location_id, w.warehouse_id
FROM warehouse w
JOIN location l ON w.location_id = l.location_id
WHERE w.warehouse_id BETWEEN 'W0000006' AND 'W0000013'
ORDER BY l.location_id DESC ;

CLEAR COLUMNS
CLEAR BREAKS
CLEAR COMPUTES
TTITLE OFF
```

Sample output:

```
Location Warehous
-------- --------
L000015  W0000013
L000012  W0000012
L000011  W0000011
L000010  W0000010
L000009  W0000009
L000008  W0000008
L000007  W0000007
L000006  W0000006

8 rows selected.
```

6.4 Cheah Zhen Hui

6.4.1 Query/Report 1: Report of customers who have made orders

Purpose: The purpose of this report is to find out customers who have made orders with shipment and invoice information, sorted by date and total amount.

SQL statement:

```
SET linesize 120
SET pagesize 30

PROMPT 'Customer order sorted by shipment date and amount'''
PROMPT
PROMPT

COLUMN customer_id FORMAT A8 HEADING "Customer ID"
COLUMN customer_name FORMAT A25 HEADING "Customer Name"
COLUMN shipment_date FORMAT A11 HEADING "Shipment Date"
COLUMN total_amount FORMAT 99999.99 HEADING "Amount"

TTITLE RIGHT 'Page No: ' FORMAT 999 SQL.PNO SKIP 2

SELECT c.customer_id, c.cust_name, s.shipment_date, i.total_amount
FROM customer c
JOIN sales_order so ON c.customer_id = so.customer_id
JOIN shipment s ON so.shipment_id = s.shipment_id
JOIN invoice i ON c.customer_id = i.customer_id AND so.order_id =
i.order_id
ORDER BY s.shipment_date DESC, i.total_amount DESC;


CLEAR COLUMNS
CLEAR BREAKS
TTITLE OFF
```

Sample output:

```
Customer CUST_NAME                    Shipment Da    Amount
-------- ---------------------------- ---------- ---------
C0000006 Jing Qi                      27/02/2023     977.00
C0000007 Tang Jie                     27/02/2023     554.00
C0000008 Intan Wahida                 27/02/2023     494.00
C0000002 Ramli Jani                   14/02/2023     667.00
C0000005 Wei Han                      07/12/2021     929.00
C0000004 Sandy                        07/12/2021     829.00
C0000003 Kingston                     07/12/2021     429.00
C0000010 Shanti Kaur                  11/11/2021    2979.00
C0000009 Naidu                        11/11/2021     283.00

9 rows selected.
```

6.4.2 Query/Report 2: Report of the products that are out of stock

Purpose: Retrieve a list of products that are stocked in multiple warehouses, along with the total stock quantity in each warehouse, sorted by the total stock quantity in descending order.In this way, it is easier for us to know which warehouse is to be out of stock for a certain product, and we can restock it in time.

SQL statement:

```
ACCEPT product_id PROMPT 'Enter product ID: '

COLUMN product_id FORMAT A12 HEADING "Product ID"
COLUMN product_name FORMAT A25 HEADING "Product Name"
COLUMN location_id FORMAT A12 HEADING "Location ID"
COLUMN stock_quantity FORMAT 99999 HEADING "Stock Quantity"

SELECT p.product_id, w.location_id, p.product_name, SUM(s.stock_quantity)
AS total_quantity
FROM product p
JOIN stock s ON p.product_id = s.product_id
JOIN warehouse w ON s.warehouse_id = w.warehouse_id
WHERE p.product_id = '&product_id'
GROUP BY p.product_id, w.location_id, p.product_name
HAVING COUNT(DISTINCT w.warehouse_id) > 1
ORDER BY total_quantity DESC;

CLEAR COLUMNS
CLEAR BREAKS
TTITLE OFF
```

Sample output:

```
SQL> start "C:\HAOGUNDAM_PRIVATE\SCHOOL\DDB_ASS\query12.txt"
Enter product ID: P0000008
old   5: WHERE p.product_id = '&product_id'
new   5: WHERE p.product_id = 'P0000008'

Product ID   Location ID  Product Name              TOTAL_QUANTITY
------------ ------------ ------------------------- --------------
P0000008     L000008      Rexona Deodorant                     635
```

6.4.3 Query/Report 3: Customer Sales record

Purpose: To retrieve information about a specific customer's past purchases based on their customer ID..

SQL statement:

```
COLUMN customer_id FORMAT A12 HEADING "Customer ID"
COLUMN order_date FORMAT A15 HEADING "Order Date"
COLUMN product_name FORMAT A25 HEADING "Product Name"
COLUMN quantity FORMAT 99999 HEADING "Quantity"
COLUMN total_amount FORMAT 99999.99 HEADING "Total Amount"
BREAK ON Customer_ID ON Order_date

ACCEPT customer_id PROMPT 'ENTER CUSTOMER ID : '

SELECT c.customer_id as Customer_ID, i.order_date as Order_date,
p.product_name
FROM customer c
JOIN sales_order so ON c.customer_id = so.customer_id
JOIN invoice i ON so.order_id = i.order_id
JOIN shipment s ON so.shipment_id = s.shipment_id
JOIN sales_order_detail sod ON so.order_id = sod.order_id
JOIN product p ON sod.product_id = p.product_id
WHERE c.customer_id = '&customer_id'
ORDER BY i.order_date;

CLEAR COLUMNS
CLEAR BREAKS
TTITLE OFF
```

Sample output:

```
Enter value for customer_id: C0000006
old   8: WHERE c.customer_id = '&customer_id'
new   8: WHERE c.customer_id = 'C0000006'

Customer ID  Order Date      Product Name
------------ --------------- -------------------------
C0000006     24-AUG-22       Pantene Shampoo
                             Fanta
                             Swiffer WetJet Starter Ki
                             t

                             MM's
                             Nivea Men Deodorant
                             Kotex Maxi Pads
                             Milo Chocolate Malt Drink

7 rows selected.
```

6.5 Chin Wen Yee

6.5.1 Query/Report 1: Detail report of total amount of product supplied by supplier

Purpose: The purpose of this report is to generate a report that shows the total amount of product supplied by each supplier, it is useful for analyzing supplier performance and making informed decisions about inventory management and purchasing.

SQL statement:

```
BREAK ON SUPPLIER
SET linesize 120
SET pagesize 30

ACCEPT supplier_name PROMPT 'Enter supplier name: '

PROMPT
PROMPT 'Total amount of product supplied by &supplier_name '
PROMPT


COLUMN supplier_name FORMAT A25 HEADING "Supplier Name"
COLUMN product_name FORMAT A25 HEADING "Product Name"
COLUMN total_quantity_supplied FORMAT 99999 HEADING "Total Quantity Supplied"

TTITLE RIGHT 'Page No: ' SQL.PNO SKIP 2

SELECT s.supplier_name AS Supplier, p.product_name AS Product_Name,
SUM(poi.POIquantity) AS total_quantity_supplied
```

```
FROM supplier s
JOIN purchase_order po ON s.supplier_id = po.supplier_id
JOIN purchase_order_item poi ON po.purchase_order_id = poi.purchase_order_id
JOIN product p ON poi.product_id = p.product_id
WHERE s.supplier_name = '&supplier_name'
GROUP BY s.supplier_name, p.product_name
ORDER BY s.supplier_name;

CLEAR COLUMNS
CLEAR BREAKS
TTITLE OFF
```

Sample output:

```
Enter supplier name: Zoe

'Total amount of product supplied by Zoe '

old   6: WHERE s.supplier_name = '&supplier_name'
new   6: WHERE s.supplier_name = 'Zoe'

                                                                                Page No:          1

SUPPLIER                 Product Name              Total Quantity Supplied
------------------------ ------------------------- -----------------------
Zoe                      Pringles                                        1
                         Downy Fabric Softener                           1
                         Pepsi                                           5
                         Ketchup                                         3
                         Pears Soap                                      4
                         Pantene Shampoo                                 1
                         Coca-Cola                                      10

7 rows selected.
```

6.5.2 Query/Report 2: Detail report of list of products by category and warehouse.

Purpose: The purpose of this report is to categorize products by their respective categories and warehouses where they are stocked, useful for inventory management and tracking the movement of the products within the warehouse.

SQL statement:

```
BREAK ON CATEGORY
SET linesize 120
SET pagesize 30

PROMPT 'List of products by category and warehouse'
PROMPT
PROMPT

COLUMN description FORMAT A25 HEADING "Category";
COLUMN product_name FORMAT A25 HEADING "Product";
COLUMN warehouse_id FORMAT A8 HEADING "Warehouse";
```

```
TTITLE RIGHT 'Page No: ' FORMAT 999 SQL.PNO SKIP 2

SELECT c.description AS category, p.product_name AS product,
s.warehouse_id AS warehouse
FROM category c
JOIN product p ON c.category_id = p.category_id
JOIN stock s ON p.product_id = s.product_id
GROUP BY c.description, p.product_name , s.warehouse_id
ORDER BY c.description;

CLEAR COLUMNS
CLEAR BREAKS
TTITLE OFF
```

Sample output:

```
'List of products by category and warehouse'

                                                                    Page No:    1

CATEGORY             PRODUCT                  WAREHOUSE
-------------------- ------------------------ ---------
Almeta               Pepsi                    W0000014
                     Scott Toilet Paper       W0000004
Andria               Pringles                 W0000018
                     Rexona Deodorant         W0000005
                     Rexona Deodorant         W0000008
Clayson              Doritos                  W0000017
                     Pears Soap               W0000007
Guido                Fanta                    W0000016
                     Pantene Shampoo          W0000006
Jude                 Sprite                   W0000013
                     Sprite                   W0000015
Kassey               Ketchup                  W0000012
                     Kleenex Facial Tissue    W0000002
Pattie               Oral-B Toothbrush        W0000010
                     Tostitos                 W0000020
Piotr                Colgate Toothpaste       W0000003

16 rows selected.
```

6.5.3 Query/Report 3: Detail report of number of customers assigned to each employee

Purpose: The purpose of this report is used to generate a report of the number of customers assigned to each employee, useful to determine the workload of each employee.

SQL statement:

```
SET linesize 120
SET pagesize 30

PROMPT 'Numbers of customers assigned to each employee'
PROMPT
PROMPT

COLUMN employee_name FORMAT A25 HEADING "Employee Name"
COLUMN num_customers FORMAT 99999 HEADING "No. of customer"

RIGHT 'Page No: ' FORMAT 999 SQL.PNO SKIP 2

SELECT e.employee_name, COUNT(c.customer_id) as num_customers
FROM employee e
JOIN customer c ON e.employee_id = c.employee_id
GROUP BY e.employee_name
ORDER BY num_customers DESC;

CLEAR COLUMNS
CLEAR BREAKS
TTITLE OFF
```

Sample output:

```
Employee Name             No. of customer
------------------------- ---------------
Alex Brown                              7
Olivia Taylor                           6
Tom Wilson                              6
Luke Perez                              6
Grace Lee                               5
Emily Davis                             5
Bob Johnson                             5
Jane Smith                              4
John Doe                                4
Sophia Nguyen                           1

10 rows selected.
```