**JEREMY PETERSON DUTOYA**

**DR. MOHAMMAD AWWAL**

**PMBA-8317-RHB APPLIED DATA MANAGEMENT FOR BUSINESS USERS**

**LAB 2 – EXECUTE BASIC SQL STATEMENTS**

1. Use a SELECT statement to display the SKU\_DESCRIPTION from SKU\_DATA with no duplicates. (Hint: use distinct command)

→ SELECT DISTINCT SKU\_DESCRIPTION FROM SKU\_DATA;

1. Use a SELECT statement to display all rows from the INVENTORY table and all columns in the table using the '\*' notation for the columns list in the SELECT clause and sort by the SKU\_Description column.

→ SELECT \* FROM INVENTORY

ORDER BY SKU\_DESCRIPTION;

OR

SELECT \* FROM INVENTORY

ORDER BY SKU\_DESCRIPTION ASC;

1. Write a SQL statement to display the SKU, SKU\_Description, for all products in the INVENTORY table that have a QuantityOnHand value equal to 0 and a QuantityOnOrder value greater than 0. **Explain what useful information this statement provides for the business.**

**→** SELECT SKU,SKU\_DESCRIPTION FROM INVENTORY

WHERE (QuantityOnHand=0) AND (QuantityOnOrder>0);

This statement is useful as it lists the products that customers want to purchase, that are successful but that are not physically available in the warehouse. This implies that the business must react quickly and reorder the item from its suppliers or increase its production to satisfy the demand or it will lose customers by not satisfying them on time.

1. Write a SQL statement to show sku\_Description for all products having sku\_description starting with ‘S’ in the INVENTORY table.

→ SELECT SKU\_DESCRIPTION FROM INVENTORY

WHERE SKU\_DESCRIPTION LIKE 'S%';

1. Write a SQL statement to show sku\_Description for all products having sku\_ description starting with ‘a’ in the second position from the beginning and ending with ‘r’ in the INVENTORY table.

→ SELECT SKU\_DESCRIPTION FROM INVENTORY

WHERE SKU\_DESCRIPTION LIKE '\_a%r';

1. Write a SQL statement to show the SKU and Description for all products having a description ending with ‘Tent’.

→ SELECT SKU, SKU\_DESCRIPTION FROM INVENTORY

WHERE SKU\_DESCRIPTION LIKE '%Tent';

1. Write a SQL statement to show the total number of products in the INVENTORY table.

→ SELECT COUNT(SKU\_DESCRIPTION) AS "Total Number of Product"

FROM INVENTORY;

OR

→ SELECT COUNT(\*) AS "Total Number of Product"

FROM INVENTORY;

1. Write a SQL statement to show a total number of products by product description (sku\_description) in INVENTORY table. Sort the output by product description in ascending order.

→ SELECT SKU\_DESCRIPTION, COUNT(SKU\_DESCRIPTION) AS "Total Number of Product"

FROM INVENTORY

GROUP BY SKU\_DESCRIPTION

ORDER BY SKU\_DESCRIPTION ASC;

OR

SELECT SKU\_DESCRIPTION, COUNT(\*) AS "Total Number of Product"

FROM INVENTORY

GROUP BY SKU\_DESCRIPTION

ORDER BY SKU\_DESCRIPTION ASC;

1. Write a SQL statement to show the sum of quantityonhand by product description (sku\_description) in INVENTORY table. Sort the output by product quantity in ascending order. What is the business value of this query?

→ SELECT SKU\_DESCRIPTION, SUM(QuantityOnHand) PRODUCT\_QUANTITY

FROM INVENTORY

GROUP BY SKU\_DESCRIPTION

ORDER BY PRODUCT\_QUANTITY ASC;

This query is valuable for a business to know the on-hand total quantity of each product it has in its inventory. If the business notices that it runs out of inventory it can take the appropriate decisions to manufacture/order more or less of a product. It is a way of determining stock and tells you what products are short on stock or which one have a high inventory and see why they are not selling enough of this product.

1. Write a SQL statement to show the sum of QuantityOnOrder by product description (sku\_description) in inventory table. Use alias to display sku\_description as “Product Description”, sum(quantityonorder) as “Number of products ordered”

→ SELECT SKU\_DESCRIPTION AS "Product Description", SUM(QuantityOnOrder) AS "Number of products ordered"

FROM INVENTORY

GROUP BY SKU\_DESCRIPTION

ORDER BY SUM(QuantityOnOrder) ASC;

On this particular question, I had an error, I first entered the following query:

SELECT SKU\_DESCRIPTION AS PRODUCT\_DESCRIPTION, SUM(QuantityOnOrder) AS NUMBER\_OF\_PRODUCTS\_ORDERED

FROM INVENTORY

GROUP BY PRODUCT\_DESCRIPTION

ORDER BY NUMBER\_OF\_PRODUCTS\_ORDERED ASC;

My mistake was that I grouped by PRODUCT\_DESCRIPTION which is the alias instead of the name of the actual attribute SKU\_DESCRIPTION. I thought that once you do an alias you have to use the alias name in the rest of the query but it seems that I was wrong, we cannot use the alias in the GROUP BY because of the syntax rule of SQL. Great to learn from my mistakes and on the internet with adequate resources: [click here](https://www.bing.com/ck/a?!&&p=81df112cd49cc0bfJmltdHM9MTY2ODEyNDgwMCZpZ3VpZD0yYzhiOTczMS03YWUyLTZhNWMtMzYxMS04NTE3N2JjYjZiZTQmaW5zaWQ9NTQzMA&ptn=3&hsh=3&fclid=2c8b9731-7ae2-6a5c-3611-85177bcb6be4&psq=can+we+use+alias+in+group+by+clause&u=a1aHR0cHM6Ly9kYmEuc3RhY2tleGNoYW5nZS5jb20vcXVlc3Rpb25zLzEwNDUzOS91c2UtY29sdW1uLWFsaWFzLWluLWdyb3VwLWJ5&ntb=1).

1. Write a SQL statement to display sku\_description and QuantityOnOrder from INVENTORY table for product order quantity is greater than 99 and less than 501. (Hint, you can use > & < operator or use between operator)

→ SELECT SKU\_DESCRIPTION, QUANTITYONORDER

FROM INVENTORY

WHERE (QuantityOnOrder>99) AND (QuantityOnOrder<501)

ORDER BY QUANTITYONORDER ASC;

Or with the BETWEEN Clause

→ SELECT SKU\_DESCRIPTION, QUANTITYONORDER

FROM INVENTORY

WHERE (QuantityOnOrder BETWEEN 99 AND 501)

ORDER BY QUANTITYONORDER ASC;

We can also run this one if we don’t want duplicate in the result

→ SELECT DISTINCT SKU\_DESCRIPTION, QUANTITYONORDER

FROM INVENTORY

WHERE (QuantityOnOrder BETWEEN 99 AND 501)

ORDER BY QUANTITYONORDER ASC;

1. In warehouse table, write two SQL statements to display warehousecity, manager for warehousestate = GA or CA (Hint: write one SQL using OR and another statement using IN)

With OR AND LIKE

→ SELECT WAREHOUSECITY, MANAGER FROM WAREHOUSE

WHERE WarehouseState LIKE 'GA' OR WarehouseState LIKE 'CA';

WITH OR AND =

→ SELECT WAREHOUSECITY, MANAGER FROM WAREHOUSE

WHERE WarehouseState='GA' OR WarehouseState='CA';

With IN

→ SELECT WAREHOUSECITY, MANAGER FROM WAREHOUSE

WHERE WarehouseState IN('GA','CA');

1. Use of ALIAS to format output
   1. Write a SQL statement to display warehouseid as “Warehouse ID”, sum of quantityonhand as “TotalItemsonHand” from INVENTORY table for quantityonhand <5. Sort the output by TotalItemsonHand in descending order.

→ SELECT WAREHOUSEID AS "Warehouse ID", SUM(QuantityOnHand) AS TotalItemsOnHand

FROM INVENTORY

WHERE QuantityOnHand<5

GROUP BY WAREHOUSEID

ORDER BY TotalItemsOnHand DESC;

Or we can also do this:

SELECT WAREHOUSEID AS "Warehouse ID", SUM(QuantityOnHand) AS "TotalItemsOnHand"

FROM INVENTORY

WHERE QuantityOnHand<5

GROUP BY WAREHOUSEID

ORDER BY SUM(QuantityOnHand) DESC

Learning: If you use “” to define the alias, you can take the name of the alias in the order by.

SQL HISTORY:

**SQL Connection TimeStamp Type Exec Duration**

| SELECT SKU\_DESCRIPTION AS "Product Description", SUM(QuantityOnOrder) AS "Number of products ordered" FROM INVENTORY GROUP BY SKU\_DESCRIPTION ORDER BY SUM(QuantityOnOrder) ASC; | Petersonduj | 1667624913991 | SQL | 1 | 0.111 |
| --- | --- | --- | --- | --- | --- |
| SELECT WAREHOUSEID AS "Warehouse ID", SUM(QuantityOnHand) AS TotalItemsOnHand FROM INVENTORY WHERE QuantityOnHand<5 GROUP BY WAREHOUSEID ORDER BY TotalItemsOnHand; | Petersonduj | 1667624762838 | SQL | 1 | 0.015 |
| SELECT WAREHOUSECITY AS "warehouse City", MANAGER FROM WAREHOUSE WHERE WarehouseState IN('GA','CA'); | Petersonduj | 1667624384365 | SQL | 1 | 0.017 |
| SELECT WAREHOUSECITY, MANAGER FROM WAREHOUSE WHERE WarehouseState IN('GA','CA'); | Petersonduj | 1667624078183 | SQL | 1 | 0.021 |
| SELECT WAREHOUSECITY, MANAGER FROM WAREHOUSE WHERE WarehouseState LIKE 'GA' OR WarehouseState LIKE 'CA'; | Petersonduj | 1667624050452 | SQL | 1 | 0.027 |
| SELECT WAREHOUSECITY, MANAGER FROM WAREHOUSE WHERE WAREHOUSESTATE='GA' OR WAREHOUSESTATE='CA'; | Petersonduj | 1667623986135 | SQL | 1 | 0.03 |
| SELECT WAREHOUSECITY, MANAGER FROM WAREHOUSE WHERE WAREHOUSESTATE LIKE 'GA' OR WAREHOUSESTATE LIKE 'CA'; | Petersonduj | 1667623933299 | SQL | 1 | 0.016 |
| DESC warehouse; | Petersonduj | 1667623728887 | SQL | 1 | 1.461 |
| SELECT DISTINCT SKU\_DESCRIPTION, QUANTITYONORDER FROM INVENTORY WHERE (QuantityOnOrder BETWEEN 99 AND 501) ORDER BY QUANTITYONORDER ASC; | Petersonduj | 1667623587170 | SQL | 1 | 0.015 |
| SELECT SKU\_DESCRIPTION, QUANTITYONORDER FROM INVENTORY WHERE (QuantityOnOrder BETWEEN 99 AND 501) ORDER BY QUANTITYONORDER ASC; | Petersonduj | 1667623536351 | SQL | 1 | 0.044 |
| SELECT SKU\_DESCRIPTION, QUANTITYONORDER FROM INVENTORY WHERE (QuantityOnOrder>99) AND (QuantityOnOrder<501) ORDER BY QUANTITYONORDER ASC; | Petersonduj | 1667623466977 | SQL | 1 | 0.032 |
| SELECT SKU\_DESCRIPTION,QUANTITYONORDER FROM INVENTORY WHERE (QuantityOnOrder>99) AND (QuantityOnOrder<501) ORDER BY QUANTITYONORDER ASC; | Petersonduj | 1667623344933 | SQL | 1 | 0.006 |
| SELECT SKU\_DESCRIPTION AS PRODUCT\_DESCRIPTION, SUM(QuantityOnOrder) AS NUMBER\_OF\_PRODUCTS\_ORDERED FROM INVENTORY GROUP BY SKU\_DESCRIPTION ORDER BY NUMBER\_OF\_PRODUCTS\_ORDERED ASC; | Petersonduj | 1667620383032 | SQL | 1 | 0.069 |
| SELECT SKU\_DESCRIPTION, SUM(QuantityOnHand) PRODUCT\_QUANTITY FROM INVENTORY GROUP BY SKU\_DESCRIPTION ORDER BY PRODUCT\_QUANTITY ASC; | Petersonduj | 1667618484954 | SQL | 1 | 0.11 |
| SELECT SUM(QuantityOnHand) TOTAL\_INVENTORY FROM INVENTORY; | Petersonduj | 1667618166308 | SQL | 1 | 0.02 |
| SELECT \* FROM INVENTORY; | Petersonduj | 1667618075932 | SQL | 1 | 0.017 |
| SELECT SKU, SKU\_DESCRIPTION FROM INVENTORY; -- WHERE SKU\_DESCRIPTION LIKE '%Tent'; | Petersonduj | 1667617686207 | Script | 1 | 0.029 |
| SELECT SKU, SKU\_DESCRIPTION FROM INVENTORY WHERE SKU\_DESCRIPTION LIKE '%Tent'; | Petersonduj | 1667617734367 | SQL | 2 | 0.005 |
| SELECT SKU\_DESCRIPTION FROM INVENTORY WHERE SKU\_DESCRIPTION LIKE '\_a%r'; | Petersonduj | 1667617568135 | SQL | 3 | 0.014 |
| SELECT SKU\_DESCRIPTION FROM INVENTORY WHERE SKU\_DESCRIPTION LIKE 'S%'; | Petersonduj | 1667617220481 | SQL | 1 | 0.015 |
| SELECT SKU,SKU\_DESCRIPTION FROM INVENTORY WHERE (QuantityOnHand=0)AND(QuantityOnOrder>0); | Petersonduj | 1667616372749 | SQL | 1 | 0.014 |
| SELECT \* FROM INVENTORY ORDER BY SKU\_DESCRIPTION; | Petersonduj | 1667616167066 | SQL | 1 | 0.063 |
| SELECT DISTINCT SKU\_DESCRIPTION FROM SKU\_DATA; | Petersonduj | 1667616007840 | SQL | 2 | 0.069 |
| DESC SKU\_DATA; | Petersonduj | 1667615864488 | SQL | 1 | 1.64 |

| SELECT COUNT(SKU\_DESCRIPTION) AS "Total Number of Product" FROM INVENTORY; | Petersonduj | 1668123721744 | SQL | 1 | 0.011 |
| --- | --- | --- | --- | --- | --- |
| SELECT SKU\_DESCRIPTION, COUNT(SKU\_DESCRIPTION) AS "Total Number of Product" FROM INVENTORY GROUP BY SKU\_DESCRIPTION; | Petersonduj | 1668123568843 | SQL | 1 | 0.054 |

| SELECT WAREHOUSEID AS "Warehouse ID", SUM(QuantityOnHand) AS TotalItemsOnHand FROM INVENTORY WHERE QuantityOnHand<5 GROUP BY WAREHOUSEID ORDER BY TotalItemsOnHand DESC; | Petersonduj | 1668145805634 | SQL | 1 | 0.004 |
| --- | --- | --- | --- | --- | --- |