

SQL Exercise 4

1. Display the minimum Status in the Supplier table.

Ans÷

```
mysql> SELECT MIN(Status) AS Min_Status FROM s;
```

```
+-----+
| Min_Status |
+-----+
| 10         |
+-----+
```

1 row in set (0.00 sec)

2. Display the maximum Weight in the Parts table.

Ans÷

```
mysql> SELECT MAX(Weight) AS Max_Weight FROM p;
```

```
+-----+
| Max_Weight |
+-----+
| 8.5        |
+-----+
```

1 row in set (0.00 sec)

3. Display the average Weight of the Parts.

Ans÷

```
mysql> SELECT AVG(Weight) AS Avg_Weight FROM p;
```

```
+-----+
| Avg_Weight |
+-----+
| 11.75      |
+-----+
```

1 row in set (0.00 sec)

4. Display the total Quantity sold for part 'P1'.

Ans÷

```
mysql> SELECT SUM(CAST(QTY AS UNSIGNED)) AS Total_Quantity_P1
-> FROM spj
-> WHERE `P#` = 'P1';
```

```
+-----+
| Total_Quantity_P1 |
+-----+
| NULL              |
+-----+
```

1 row in set (0.02 sec)

5. Display the total Quantity sold for each part.

Ans÷

```
mysql> SELECT `P#` AS Part_Number,
-> SUM(CAST(QTY AS UNSIGNED)) AS Total_Quantity
-> FROM spj
-> GROUP BY `P#`;
```

Part_Number	Total_Quantity
P4	150
P5	250
P6	300
P2	180
P7	220

5 rows in set (0.00 sec)

6. Display the average Quantity sold for each part.

Ans÷

```
mysql> SELECT `P#`, AVG(CAST(QTY AS DECIMAL)) AS Average_Quantity_Sold
-> FROM spj
-> GROUP BY `P#`;
```

P#	Average_Quantity_Sold
P4	150.0000
P5	250.0000
P6	300.0000
P2	180.0000
P7	220.0000

5 rows in set (0.00 sec)

7. Display the maximum Quantity sold for each part, provided the maximum Quantity is greater than 800.

Ans÷

```
mysql> SELECT `P#`, MAX(CAST(`QTY` AS DECIMAL)) AS Max_Quantity_Sold
-> FROM spj
-> GROUP BY `P#`
-> HAVING MAX(CAST(`QTY` AS DECIMAL)) > 800;
```

P#	Max_Quantity_Sold
P2	1500
P1	1000
P3	2000

3 rows in set (0.00 sec)

8. Display the Status and the count of Suppliers with that Status.

Ans÷

```
mysql> SELECT `STATUS`, COUNT(*) AS Supplier_Count
-> FROM s
-> GROUP BY `STATUS`;
```

STATUS	Supplier_Count
--------	----------------

STATUS	Supplier_Count
NULL	3
10	1
20	1
30	2
25	1
15	1

6 rows in set (0.00 sec)

9. Display the count of Projects going on in different cities.

Ans÷

```
mysql> SELECT CITY, COUNT(*) AS Project_Count
-> FROM j
-> GROUP BY CITY;
```

CITY	Project_Count
NULL	3
Berlin	1
Rome	1
Paris	1

4 rows in set (0.00 sec)

10. What is the difference between COUNT(Status) and COUNT(*) ?

Ans÷

ChatGPT said:

Certainly! Here's the difference between COUNT(Status) and COUNT(*) in the context of a table, represented in column format:

Table Example: supplier

S#	SNAME	STATUS	CITY
S1	John	Active	New York
S2	Alice	NULL	Boston
S3	Bob	Inactive	Chicago
S4	Charlie	Active	New York
S5	Eve	NULL	Boston

1. COUNT(Status) :

- **Counts:** Rows where the STATUS column has **non-NULL** values.
- **Does not count:** Rows where the STATUS column contains NULL.

COUNT(Status)

Explanation

3

There are 3 rows where STATUS is non-NULL (Active, Inactive, Active).

2. COUNT(*) :

- **Counts:** All rows in the table, **including rows with NULL values** in any column.
- **Does not ignore:** NULL values in any column.

COUNT(*)

Explanation

5

There are 5 rows in total, regardless of whether STATUS is NULL or not.

11. Display the Status and the Count of Suppliers with that Status in the following format as shown below:-

Status	Count
Ten	1
Twenty	2
Thirty	3

Ans÷

```
mysql> SELECT
-> CASE
->   WHEN STATUS = '10' THEN 'Ten'
->   WHEN STATUS = '20' THEN 'Twenty'
->   WHEN STATUS = '30' THEN 'Thirty'
->   ELSE 'Other'
-> END AS Status,
-> COUNT(*) AS Count
-> FROM s
-> GROUP BY Status;
```

Status	Count
Other	3
Ten	1
Twenty	1
Thirty	2
Other	1
Other	1

6 rows in set, 1 warning (0.00 sec)