202. WAQTD THE TOTAL REVENUE GENERATED FROM EVERY STATE.

```
SELECT SUM(P1.PRICE) REVENUE,L1.STATE
FROM PRODUCT P1 INNER JOIN ORDERS O1
ON P1.PRODUCT_ID=O1.PRODUCT_ID INNER JOIN CUSTOMER C1
ON O1.CID=C1.CID INNER JOIN LOCATION L1
ON C1.LID=L1.LID
GROUP BY L1.STATE;
```

203. WAQTD THE TOTAL REVENUE GENERATED FROM EVERY STATE, EVERY MONTH.

```
SELECT
SUM(P1.PRICE),L1.STATE,DATE_FORMAT(O1.ORDER_DATE,'%M')
MONTH
FROM PRODUCT P1 INNER JOIN ORDERS O1
ON P1.PRODUCT_ID=O1.PRODUCT_ID INNER JOIN CUSTOMER C1
ON O1.CID=C1.CID INNER JOIN LOCATION L1
ON C1.LID=L1.LID
```

```
GROUP BY L1.STATE,MONTH
ORDER BY L1.STATE ASC;
```

204. WAQTD THE STATES WHICH HOLD THE TOP 3 BASED ON THE NUMBER OF ORDERS DONE SO FAR.

```
SELECT L1.STATE,COUNT(*) TOTAL_ORDERS
FROM LOCATION L1 INNER JOIN CUSTOMER C1
ON L1.LID=C1.LID INNER JOIN ORDERS O1
ON C1.CID=O1.CID
GROUP BY L1.STATE
ORDER BY TOTAL_ORDERS DESC
LIMIT 3;
```

═══

OUTER JOINS :

═══════════

Outer Joins are used to obtain matched records along with the unmatched records.

→ LEFT OUTER JOIN
→ RIGHT OUTER JOIN
→ FULL OUTER JOIN

LEFT OUTER JOIN

Left Outer joins are used to obtain the matched and the unmatched records from the LEFT table.

Syntax:

```
SELECT column_name
FROM table_name T1 LEFT OUTER JOIN table_name T2
ON T1.column_name=T2.column_name;
```

RIGHT OUTER JOINS

Right Outer joins are used to obtain the matched and the unmatched records from the RIGHT table.

Syntax:

```
SELECT column_name
FROM table_name T1 RIGHT OUTER JOIN table_name T2
ON T1.column_name=T2.column_name;
```

FULL OUTER JOIN

> Full Outer joins are used to obtain the matched and the unmatched records from the both the tables.
> It is the combination of both left outer joins and right outer joins.

Syntax:

```
SELECT column_name
FROM table_name T1 LEFT OUTER JOIN table_name T2
ON T1.column_name=T2.column_name
UNION
SELECT column_name
FROM table_name T1 RIGHT OUTER JOIN table_name T2
ON T1.column_name=T2.column_name;
```

205. WAQTD THE CUSTOMERS WHO HAVE NOT ORDERED SO FAR.

```
SELECT C1.FIRST_NAME
FROM CUSTOMER C1 LEFT OUTER JOIN ORDERS O1
ON C1.CID=O1.CID
WHERE O1.CID IS NULL;
```

206. WAQTD THE CUSTOMER WHO HAS SPENT THE MOST OVERALL.

```
SELECT C1.FIRST_NAME,SUM(P1.PRICE) TOTAL
FROM CUSTOMER C1 INNER JOIN ORDERS O1
ON C1.CID=O1.CID INNER JOIN PRODUCT P1
ON O1.PRODUCT_ID=P1.PRODUCT_ID
GROUP BY C1.FIRST_NAME
ORDER BY TOTAL DESC
LIMIT 1;
```

207. WAQTD THE LOCATIONS WHERE NO DEPT ARE LOCATED.

```
SELECT L1.LOCATION
FROM DEPT D1 RIGHT OUTER JOIN LOCATION L1
ON D1.LID=L1.LID
WHERE D1.LID IS NULL;
```

208. WAQTD THE PRODUCTS WHICH REMAINS UNSOLD TILL NOW.

```
SELECT P1.PNAME
```

```
FROM PRODUCT P1 LEFT OUTER JOIN ORDERS O1
ON P1.PRODUCT_ID=O1.PRODUCT_ID
WHERE O1.PRODUCT_ID IS NULL;
```

══

══

NATURAL JOINS

═════════════

> Natural Joins are used to obtain the matched records from two tables by using common column names.

> Here, no join condition will be written.

> The common column which is present in both the table is displayed only once as a first column.

Syntax:

───────

```
SELECT column_name
FROM table_name_1 NATURAL JOIN table_name_2;
```

```
SELECT *
FROM EMP NATURAL JOIN DEPT;
```