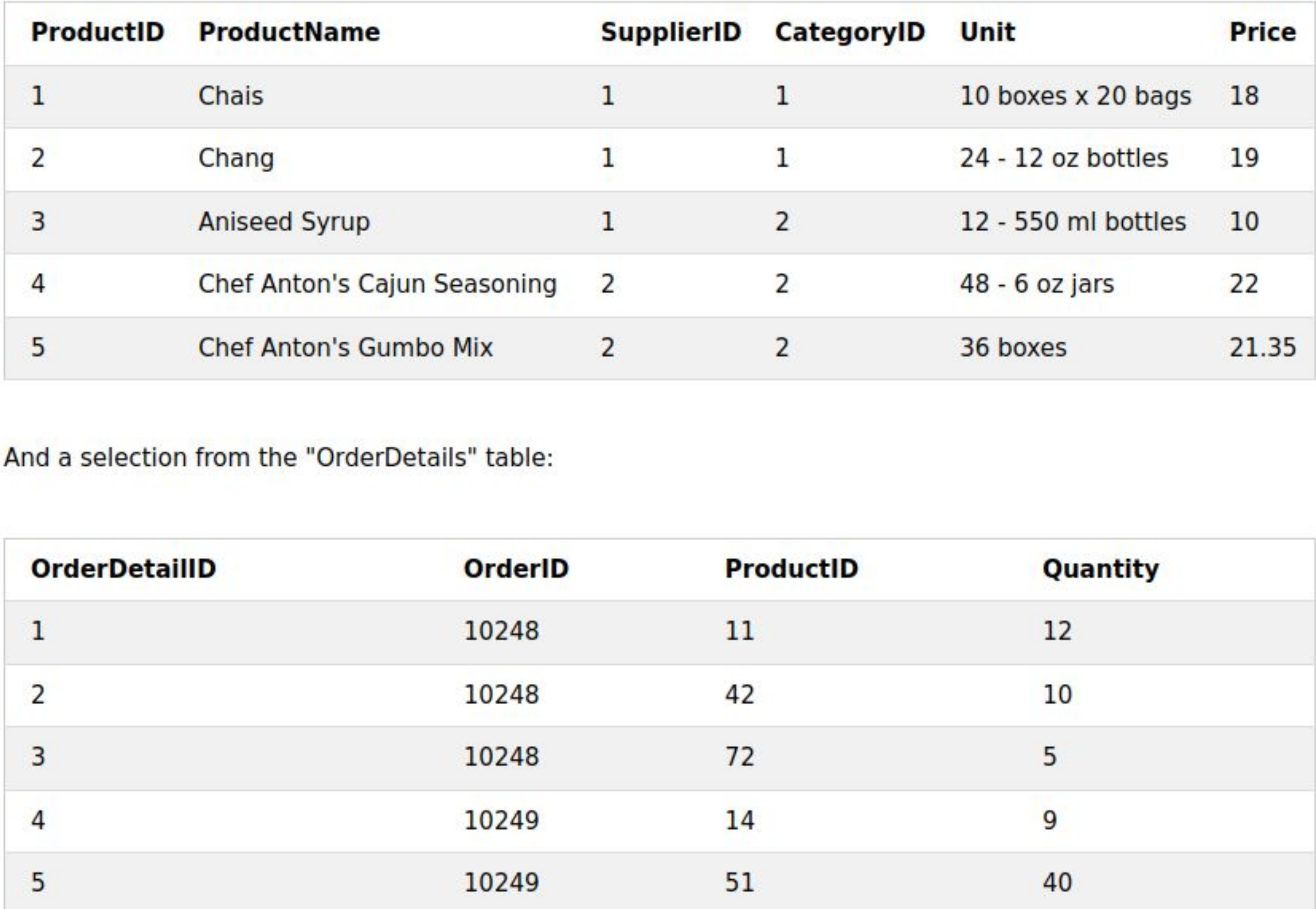
IN,ALL, ANY



ProductID	ProductName	SupplierID	CategoryID	Unit	Price
1	Chais	1	1	10 boxes x 20 bags	18
2	Chang	1	1	24 - 12 oz bottles	19
3	Aniseed Syrup	1	2	12 - 550 ml bottles	10
4	Chef Anton's Cajun Seasoning	2	2	48 - 6 oz jars	22
5	Chef Anton's Gumbo Mix	2	2	36 boxes	21.35

And a selection from the "OrderDetails" table:

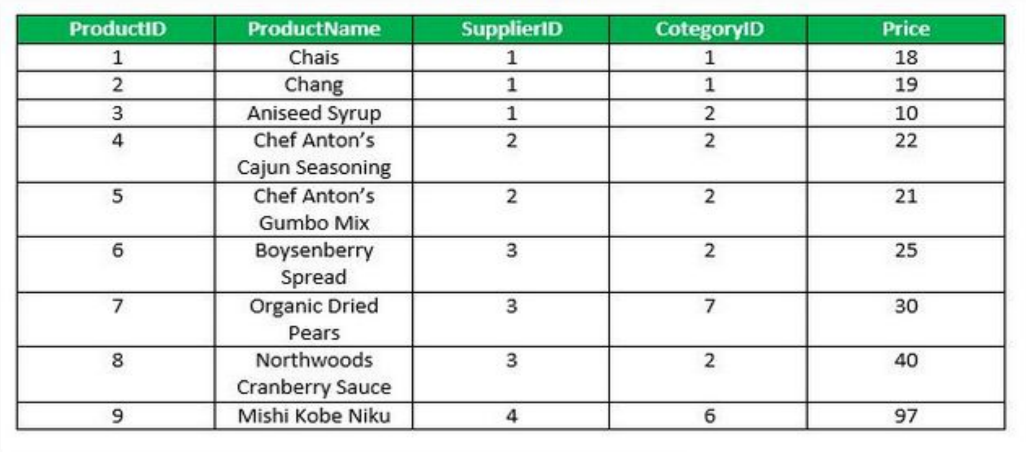
OrderDetailID	OrderID	ProductID	Quantity
1	10248	11	12
2	10248	42	10
3	10248	72	5
4	10249	14	9
5	10249	51	40

ANY

ANY compares a value to each value in a list or results from a query and evaluates to true if the result of an inner query contains at least one row.

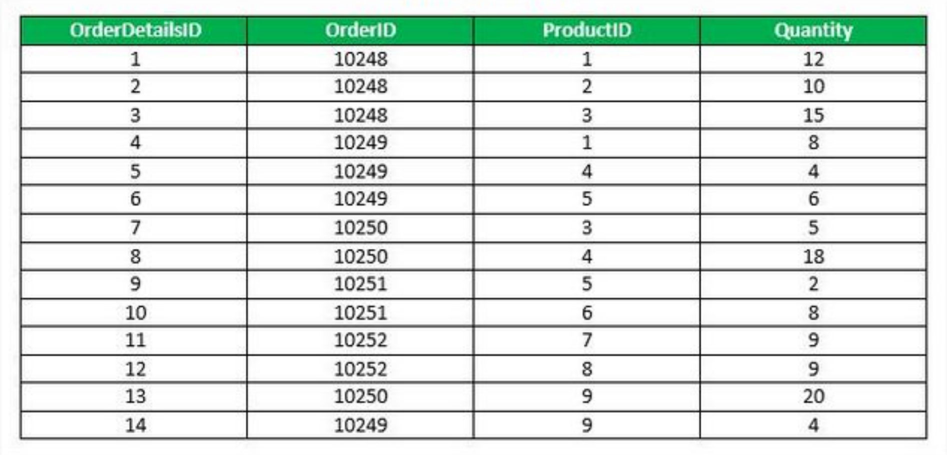
- ANY return true if any of the subqueries values meet the condition.

Products Table



ProductID	ProductName	SupplierID	CotegoryID	Price
1	Chais	1	1	18
2	Chang	1	1	19
3	Aniseed Syrup	1	2	10
4	Chef Anton's Cajun Seasoning	2	2	22
5	Chef Anton's Gumbo Mix	2	2	21
6	Boysenberry Spread	3	2	25
7	Organic Dried Pears	3	7	30
8	Northwoods Cranberry Sauce	3	2	40
9	Mishi Kobe Niku	4	6	97

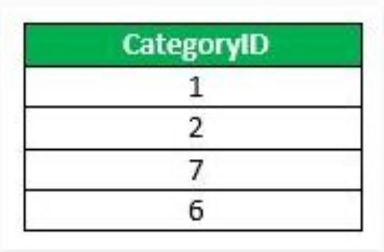
OrderDetails Table



OrderDetailsID	OrderID	ProductID	Quantity
1	10248	1	12
2	10248	2	10
3	10248	3	15
4	10249	1	8
5	10249	4	4
6	10249	5	6
7	10250	3	5
8	10250	4	18
9	10251	5	2
10	10251	6	8
11	10252	7	9
12	10252	8	9
13	10250	9	20
14	10249	9	4

Find the Distinct CategoryID of the products which have any record in OrderDetails Table.

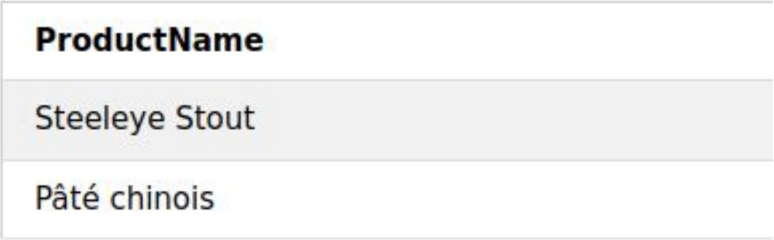
```
SELECT DISTINCT CategoryID

FROM Products

WHERE ProductID = ANY (SELECT ProductID

                    FROM OrderDetails);
```



CategoryID
1
2
7
6

ANY- The following SQL statement returns TRUE and lists the product names if it finds ANY records in the OrderDetails table that quantity > 99:

```sql
SELECT ProductName
FROM Products
WHERE ProductID = ANY (SELECT ProductID FROM OrderDetails WHERE
Quantity > 99);
```



ProductName
Steeleye Stout
Pâté chinois

ALL & ANY are logical operators in SQL. They return boolean value as a result.

ALL

ALL operator is used to select all tuples of SELECT STATEMENT. It is also used to compare a value to every value in another value set or result from a subquery.

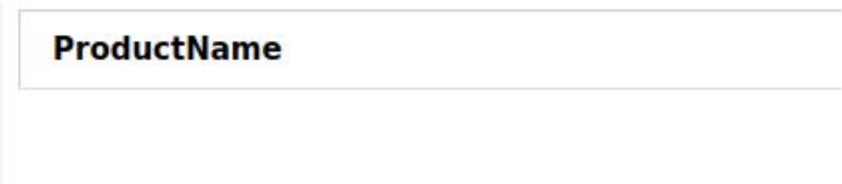
- The ALL operator returns TRUE if all of the subqueries values meet the condition. The ALL must be preceded by comparison operators and evaluates true if all of the subqueries values meet the condition.
- ALL is used with SELECT, WHERE, HAVING statement.

ALL

The ALL operator returns TRUE if all of the subquery values meet the condition.

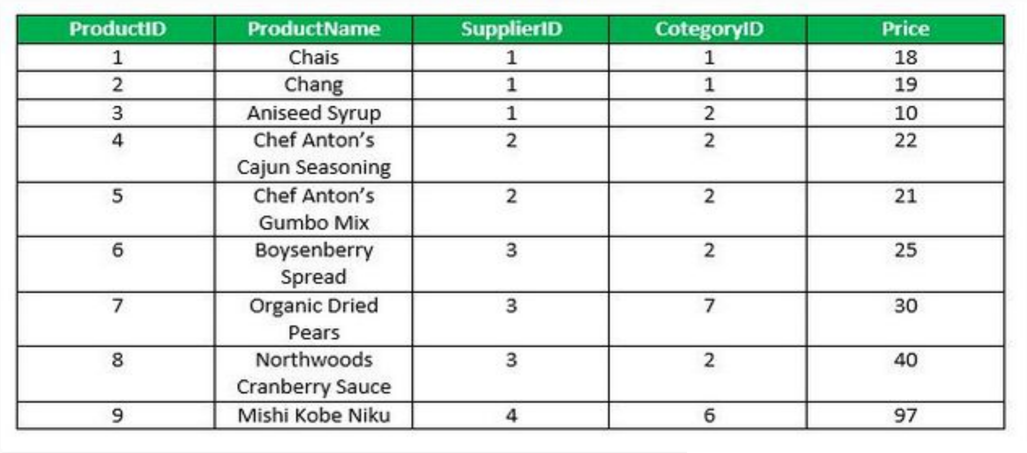
The following SQL statement returns TRUE and lists the product names if ALL the records in the OrderDetails table has quantity = 10 (so, this example will return FALSE, because not ALL records in the OrderDetails table has quantity = 10):

```
SELECT ProductName

FROM Products

WHERE ProductID = ALL (SELECT ProductID FROM OrderDetails WHERE Quantity = 10);
```



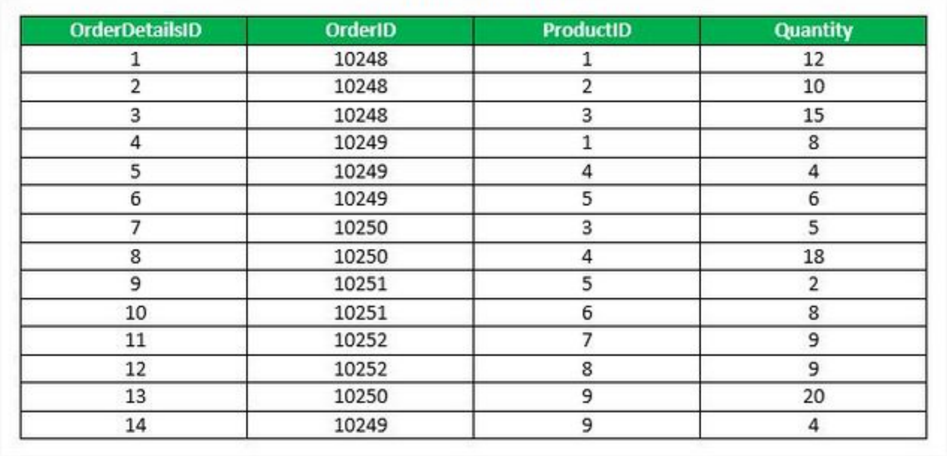
ProductName

Products Table



ProductID	ProductName	SupplierID	CotegoryID	Price
1	Chais	1	1	18
2	Chang	1	1	19
3	Aniseed Syrup	1	2	10
4	Chef Anton’s Cajun Seasoning	2	2	22
5	Chef Anton’s Gumbo Mix	2	2	21
6	Boysenberry Spread	3	2	25
7	Organic Dried Pears	3	7	30
8	Northwoods Cranberry Sauce	3	2	40
9	Mishi Kobe Niku	4	6	97

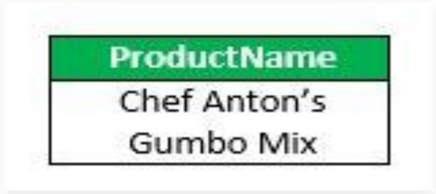
OrderDetails Table



OrderDetailsID	OrderID	ProductID	Quantity
1	10248	1	12
2	10248	2	10
3	10248	3	15
4	10249	1	8
5	10249	4	4
6	10249	5	6
7	10250	3	5
8	10250	4	18
9	10251	5	2
10	10251	6	8
11	10252	7	9
12	10252	8	9
13	10250	9	20
14	10249	9	4

Find the name of the product if all the records in the OrderDetails has Quantity either equal to 6 or 2.

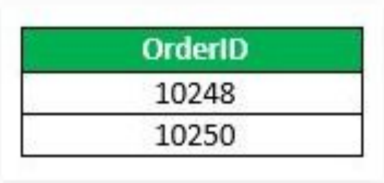
```
SELECT ProductName

FROM Products

WHERE ProductID = ALL (SELECT ProductId

                 FROM OrderDetails

                 WHERE Quantity = 6 OR Quantity = 2);
```



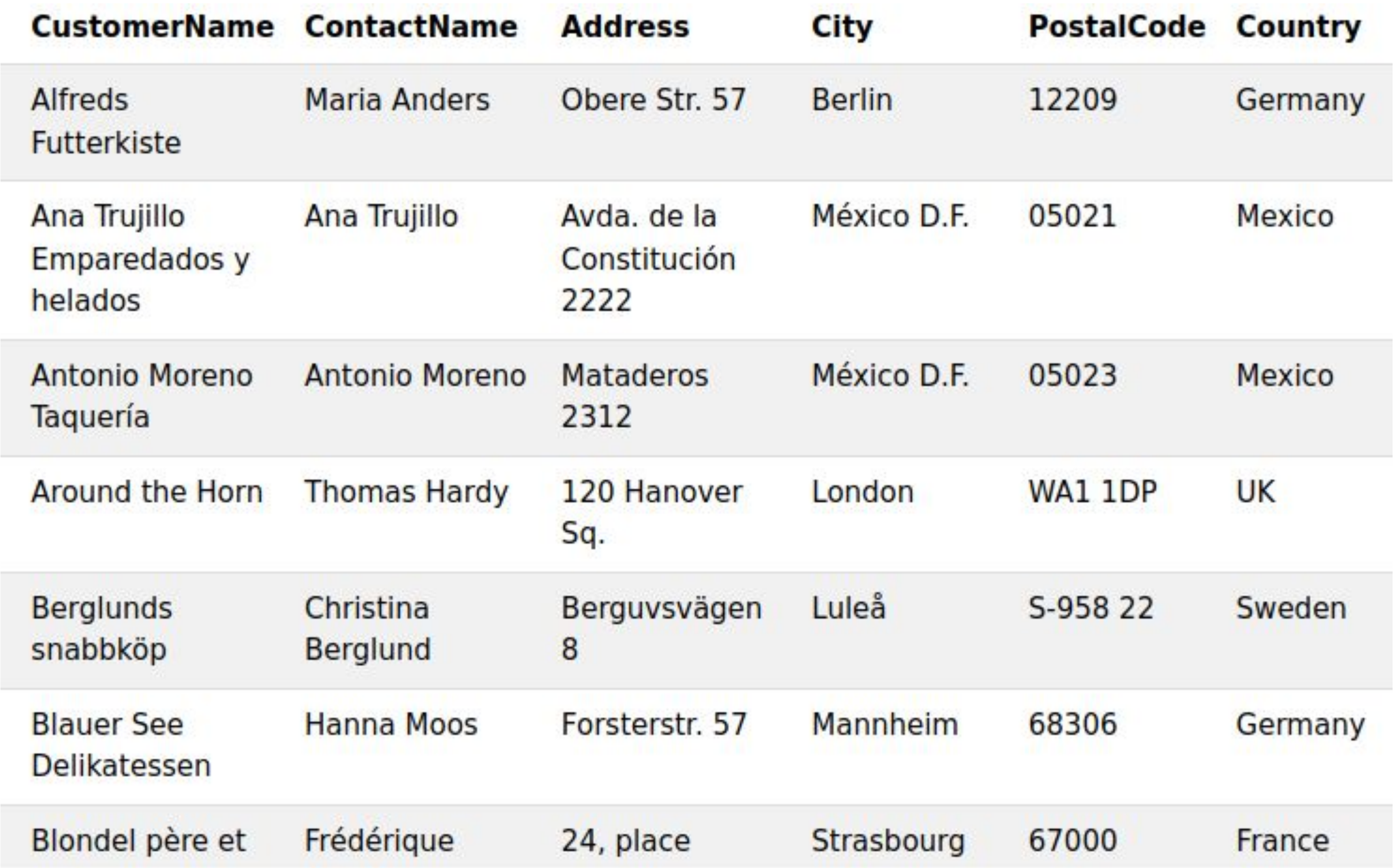
ProductName
Chef Anton's Gumbo Mix

Find the OrderID whose maximum Quantity among all product of that OrderID is greater than average quantity of all OrderID.

```
SELECT OrderID

FROM OrderDetails

GROUP BY OrderID

HAVING max(Quantity) > ALL (SELECT avg(Quantity)

              FROM OrderDetails

              GROUP BY OrderID);
```

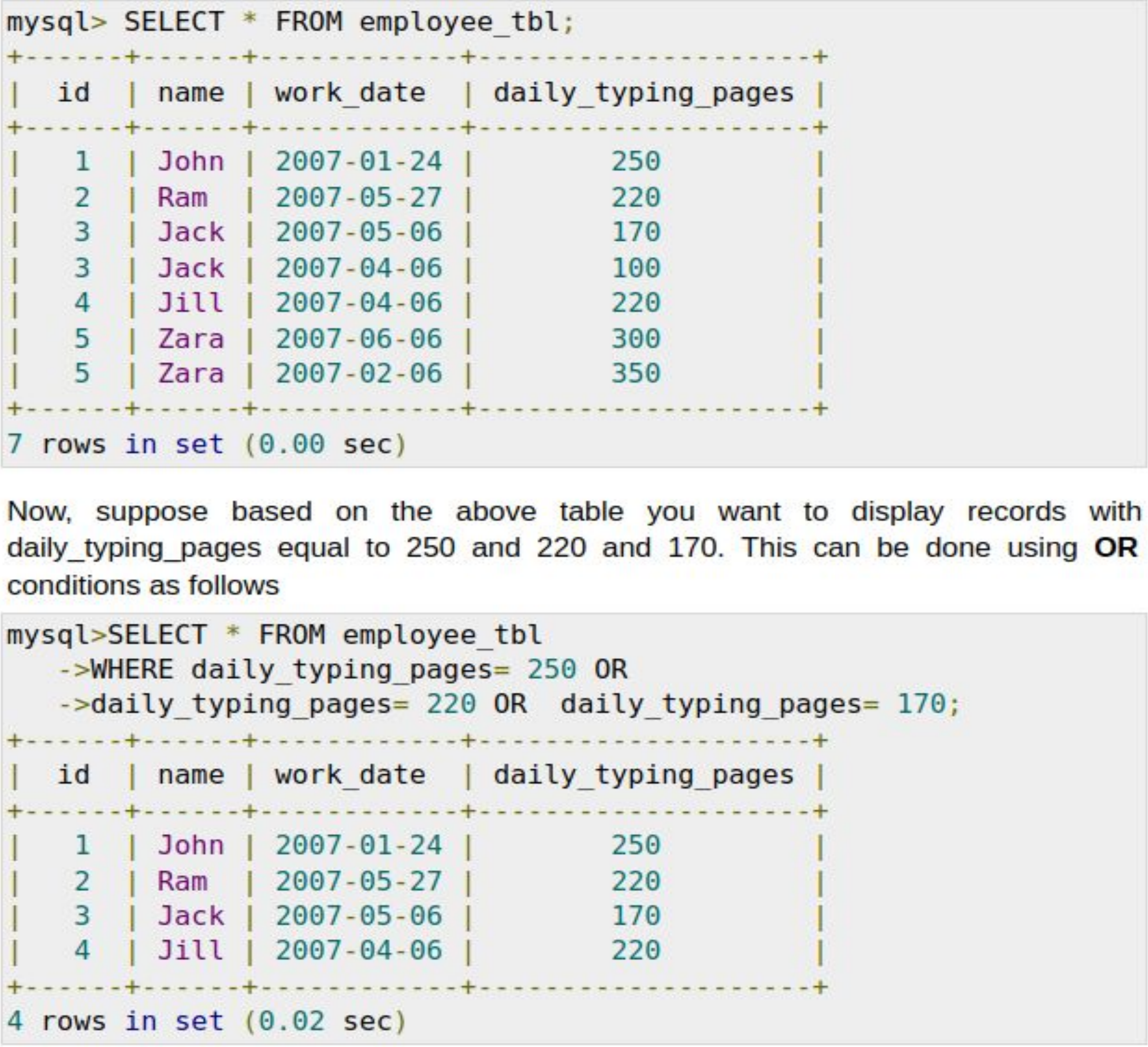


OrderID
10248
10250



CustomerName	ContactName	Address	City	PostalCode	Country
Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden
Blauer See Delikatessen	Hanna Moos	Forsterstr. 57	Mannheim	68306	Germany
Blondel père et	Frédérique	24, place	Strasbourg	67000	France

IN-use **IN** clause to replace many **OR** conditions

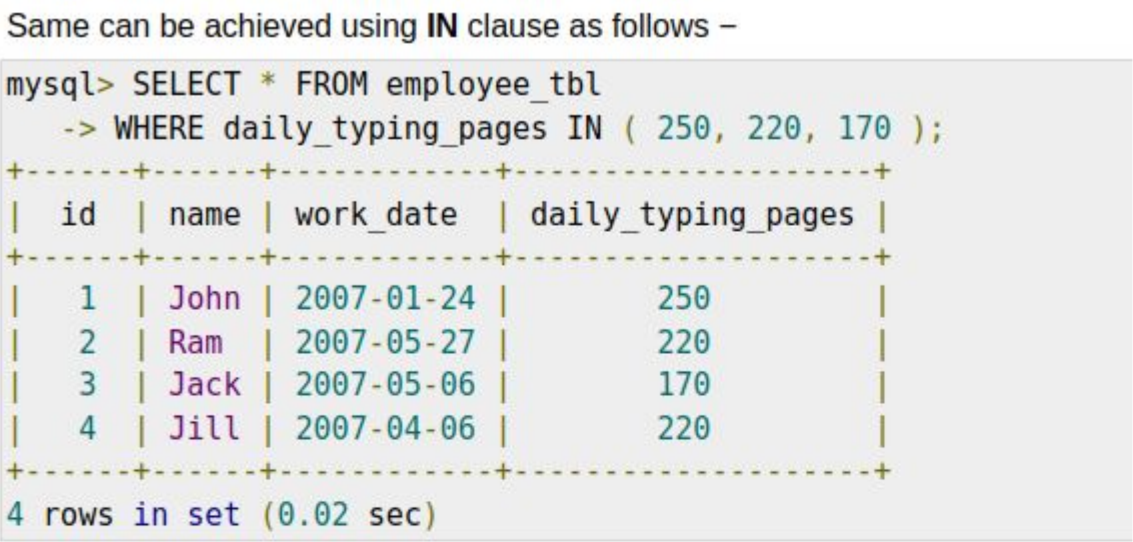


```
mysql> SELECT * FROM employee_tbl;
+------+------+------------+--------------------+
| id   | name | work_date  | daily_typing_pages |
+------+------+------------+--------------------+
|   1  | John | 2007-01-24 |        250         |
|   2  | Ram  | 2007-05-27 |        220         |
|   3  | Jack | 2007-05-06 |        170         |
|   3  | Jack | 2007-04-06 |        100         |
|   4  | Jill | 2007-04-06 |        220         |
|   5  | Zara | 2007-06-06 |        300         |
|   5  | Zara | 2007-02-06 |        350         |
+------+------+------------+--------------------+
7 rows in set (0.00 sec)
```

Now, suppose based on the above table you want to display records with daily_typing_pages equal to 250 and 220 and 170. This can be done using **OR** conditions as follows

```
mysql>SELECT * FROM employee_tbl
   ->WHERE daily_typing_pages= 250 OR
   ->daily_typing_pages= 220 OR  daily_typing_pages= 170;
+------+------+------------+--------------------+
| id   | name | work_date  | daily_typing_pages |
+------+------+------------+--------------------+
|   1  | John | 2007-01-24 |        250         |
|   2  | Ram  | 2007-05-27 |        220         |
|   3  | Jack | 2007-05-06 |        170         |
|   4  | Jill | 2007-04-06 |        220         |
+------+------+------------+--------------------+
4 rows in set (0.02 sec)
```

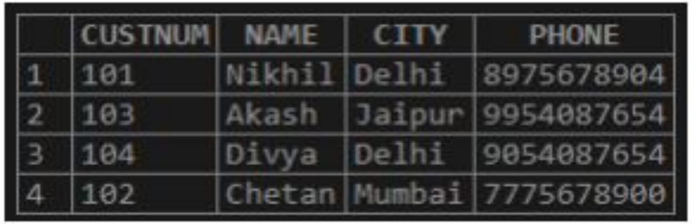
Using IN



Same can be achieved using **IN** clause as follows –

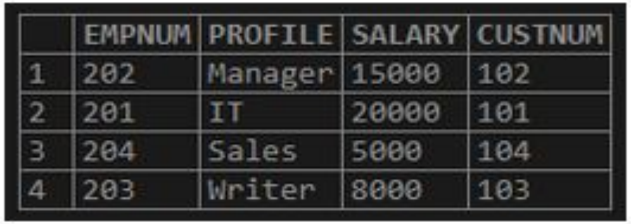
```
mysql> SELECT * FROM employee_tbl
   -> WHERE daily_typing_pages IN ( 250, 220, 170 );
+------+------+------------+--------------------+
|  id  | name | work_date  | daily_typing_pages |
+------+------+------------+--------------------+
|   1  | John | 2007-01-24 |        250         |
|   2  | Ram  | 2007-05-27 |        220         |
|   3  | Jack | 2007-05-06 |        170         |
|   4  | Jill | 2007-04-06 |        220         |
+------+------+------------+--------------------+
4 rows in set (0.02 sec)
```

Customers Table:



	CUSTNUM	NAME	CITY	PHONE
1	101	Nikhil	Delhi	8975678904
2	103	Akash	Jaipur	9954087654
3	104	Divya	Delhi	9054087654
4	102	Chetan	Mumbai	7775678900

Employees Table:

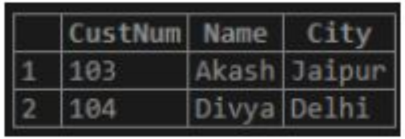


	EMPNUM	PROFILE	SALARY	CUSTNUM
1	202	Manager	15000	102
2	201	IT	20000	101
3	204	Sales	5000	104
4	203	Writer	8000	103

Query:

```
SELECT CustNum,Name, City FROM Customers WHERE CustNum IN
(SELECT CustNum FROM Employees where EmpNum > 202 ) ORDER BY CustNum;
```

Output:



```
   CustNum  Name   City
1  103      Akash  Jaipur
2  104      Divya  Delhi
```