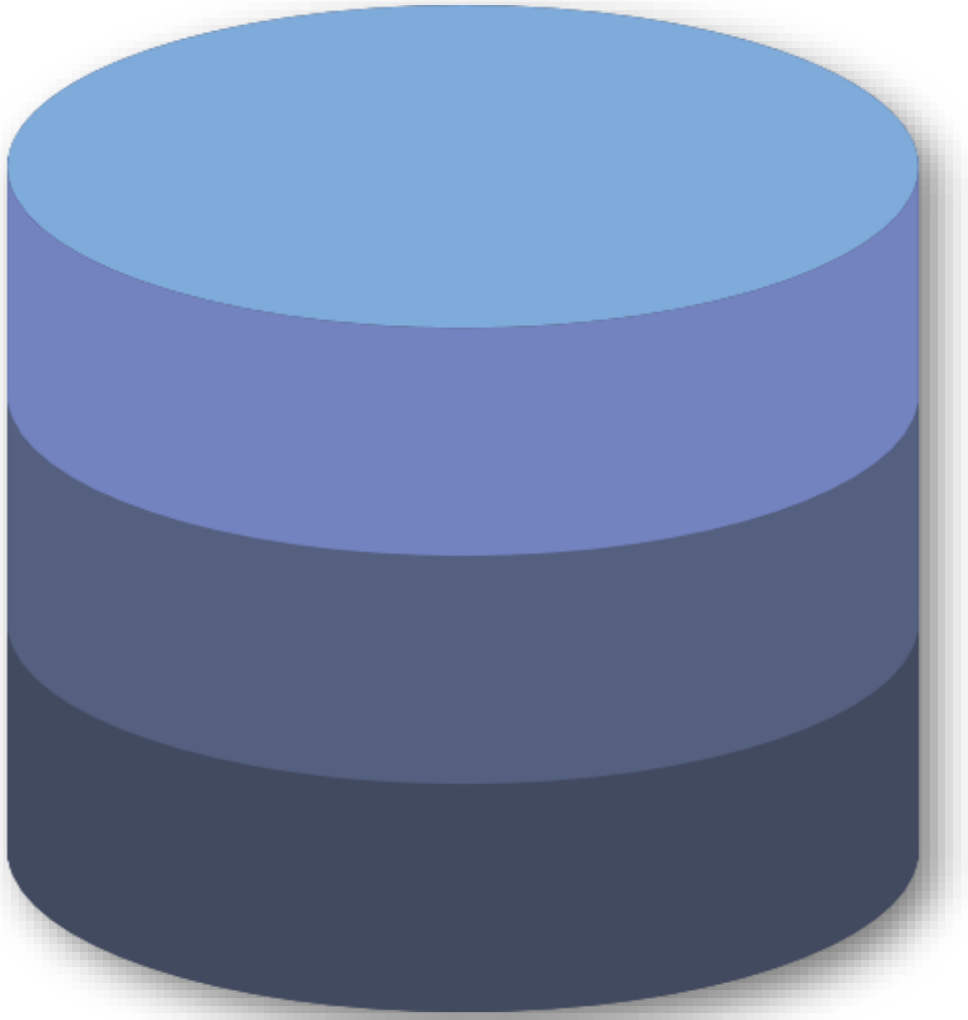
# SQL Tutorial



SQL

# Agenda

- Order By
- Union Operator
- TOP
- Union All Operator
- Group By
- Except Operator
- Having Clause
- Intersect Operator

| e_id | e_name | e_salary | e_age | e_gender | e_dept |
|---|---|---|---|---|---|
| 1 | Sam | 95000 | 45 | Male | Operations |
| 2 | Bob | 80000 | 21 | Male | Support |
| 3 | Anne | 125000 | 25 | Female | Analytics |
| 4 | Julia | 73000 | 30 | Female | Analytics |
| 5 | Matt | 159000 | 33 | Male | Sales |
| 6 | Jeff | 112000 | 27 | Male | Operations |

# Order By

ORDER BY is used to sort the data in ascending or descending order.

Ascending

Descending

# Order By: Syntax


```
SELECT column_list FROM table_name
ORDER BY col1, col2,……. ASC | DSC
```

# TOP Clause

TOP clause is used to fetch the **top N** records.

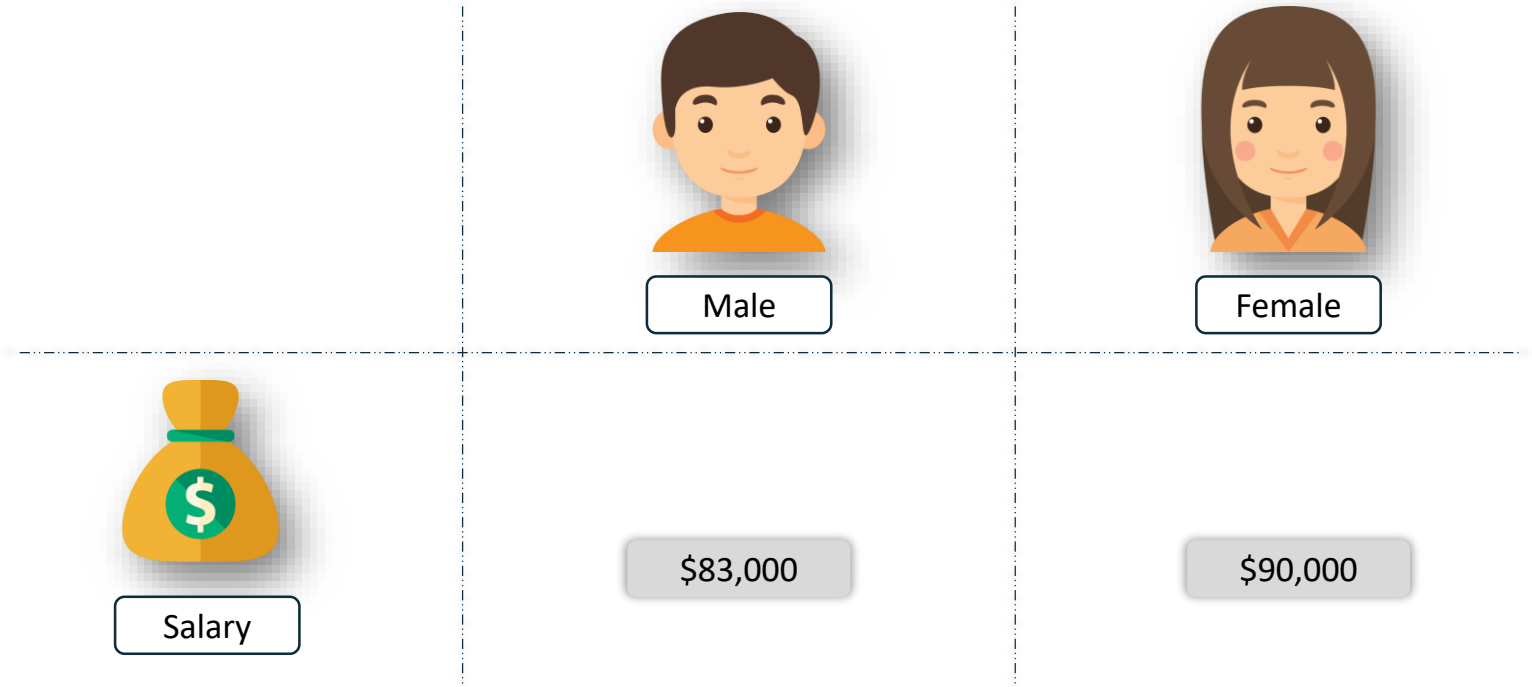
| e_id | e_name | e_salary | e_age | e_gender | e_dept |
|---|---|---|---|---|---|
| 1 | Sam | 95000 | 45 | Male | Operations |
| 2 | Bob | 80000 | 21 | Male | Support |
| 3 | Anne | 125000 | 25 | Female | Analytics |
| 4 | Julia | 73000 | 30 | Female | Analytics |
| 5 | Matt | 159000 | 33 | Male | Sales |
| 6 | Jeff | 112000 | 27 | Male | Operations |

# TOP Clause: Syntax


```
SELECT TOP x column_list FROM
table_name;
```

# Group By

Group By is used to get an aggregate result with respect to a group.

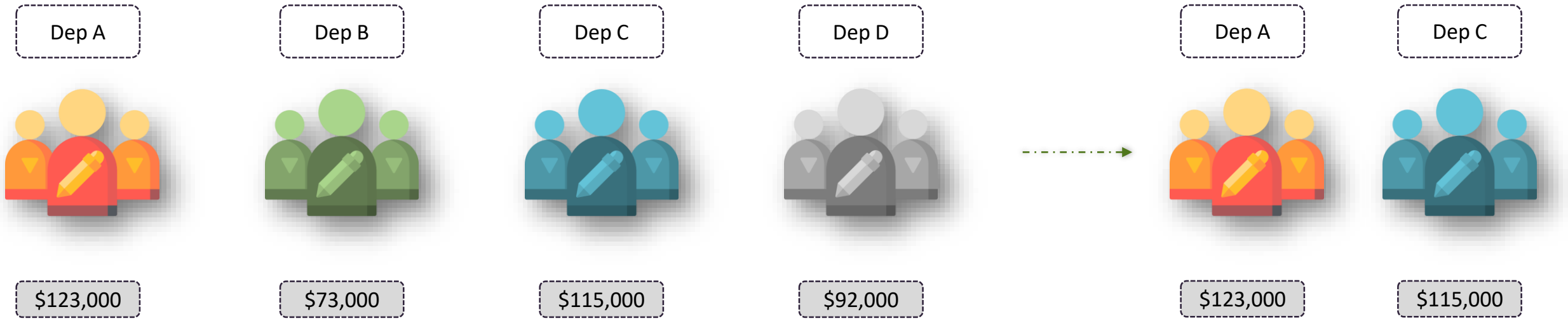


# Group By: Syntax


```
SELECT column_list
FROM table_name
WHERE condition
GROUP BY colname(s)
ORDER BY colname(s)
```

# Having Clause

Having clause is used in combination with Group By to impose conditions on groups.

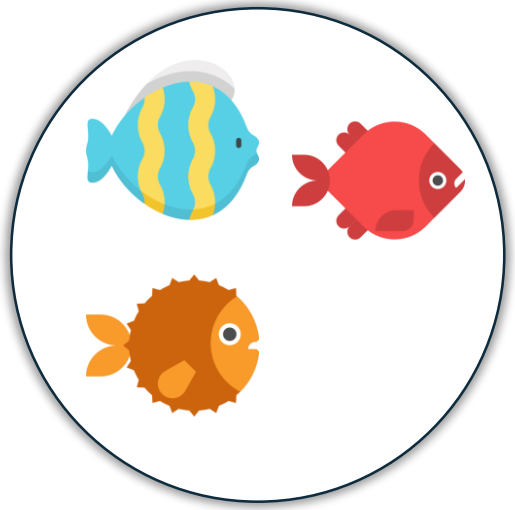


# Having Clause: Syntax

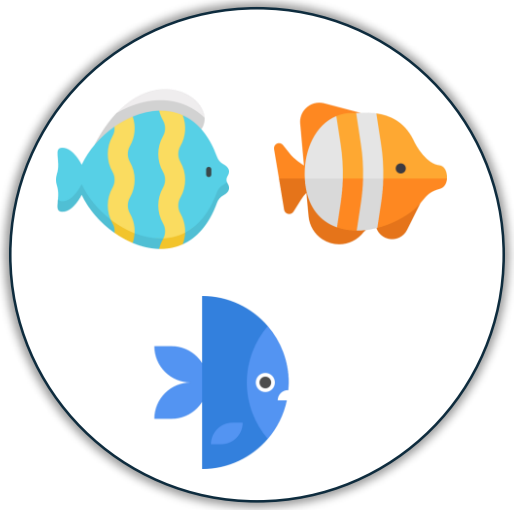

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
HAVING condition
ORDER BY column_name(s);
```

# Union Operator

Union operator is used to combine the result set of two or more SELECT statements.



A



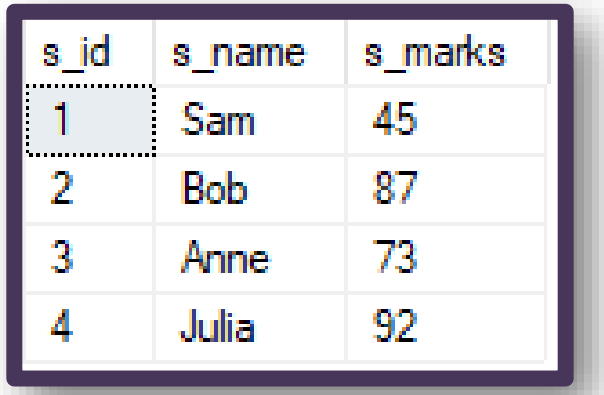
B

A U B

# Union Operator: Syntax

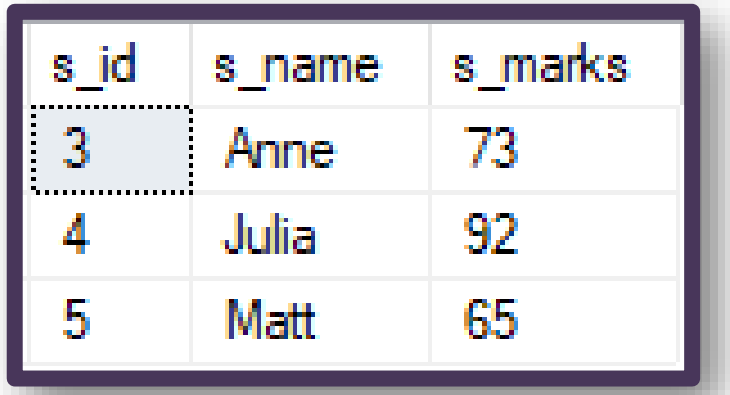

```
SELECT column_list FROM table1
Union
SELECT column_list FROM table2
```

# Union Operator



| s_id | s_name | s_marks |
|---|---|---|
| 1 | Sam | 45 |
| 2 | Bob | 87 |
| 3 | Anne | 73 |
| 4 | Julia | 92 |

Student_Details1

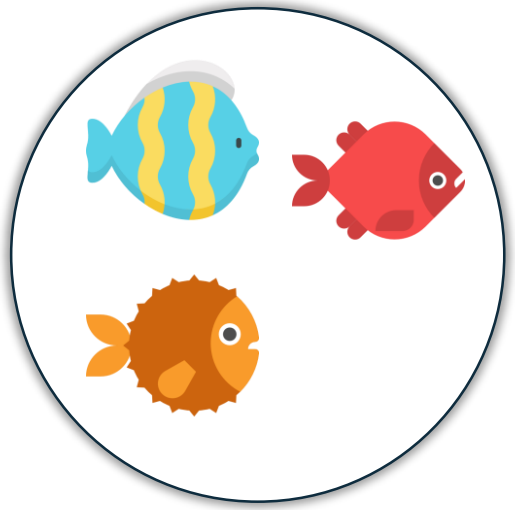


| s_id | s_name | s_marks |
|---|---|---|
| 3 | Anne | 73 |
| 4 | Julia | 92 |
| 5 | Matt | 65 |

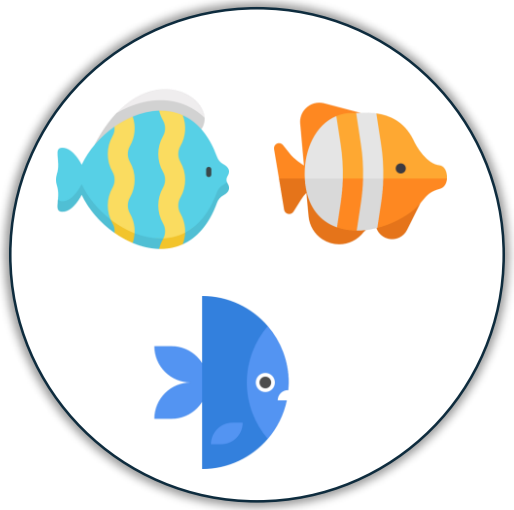
Student_Details2

# Union All Operator

Union All operator gives all rows from both tables including the duplicates.



A

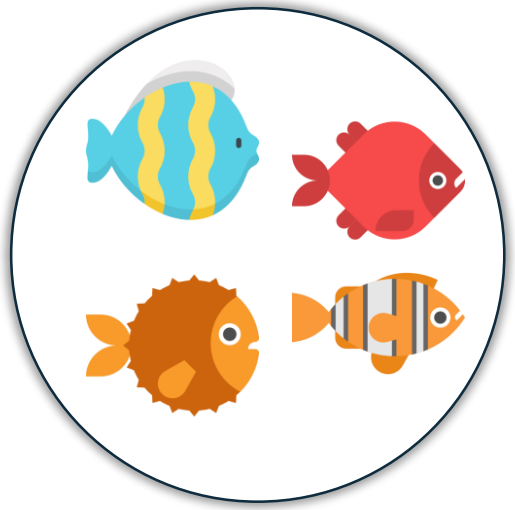


B

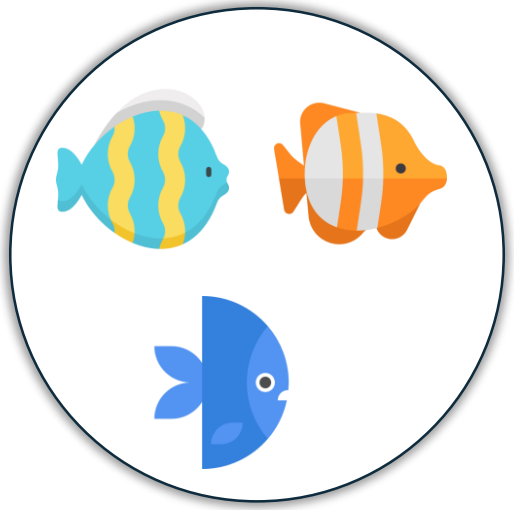
A union all B

# Except Operator

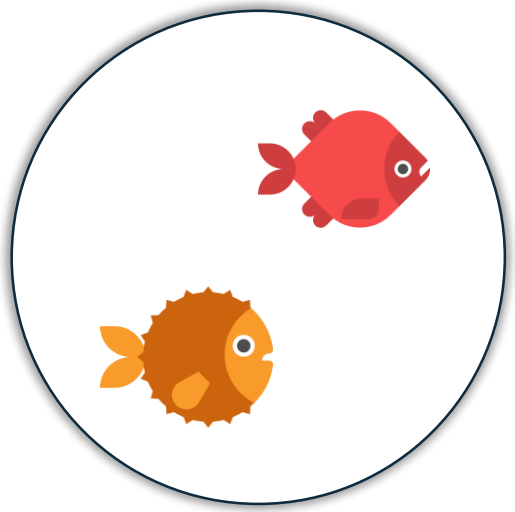
Except Operator combines two select statements and returns unique records from the left query which are not part of the right query.



A



B



A - B

# Except Operator: Syntax


```
SELECT column_list FROM table1
EXCEPT
SELECT column_list FROM table2
```

# Except Operator

| s_id | s_name | s_marks |
|---|---|---|
| 1 | Sam | 45 |
| 2 | Bob | 87 |
| 3 | Anne | 73 |
| 4 | Julia | 92 |

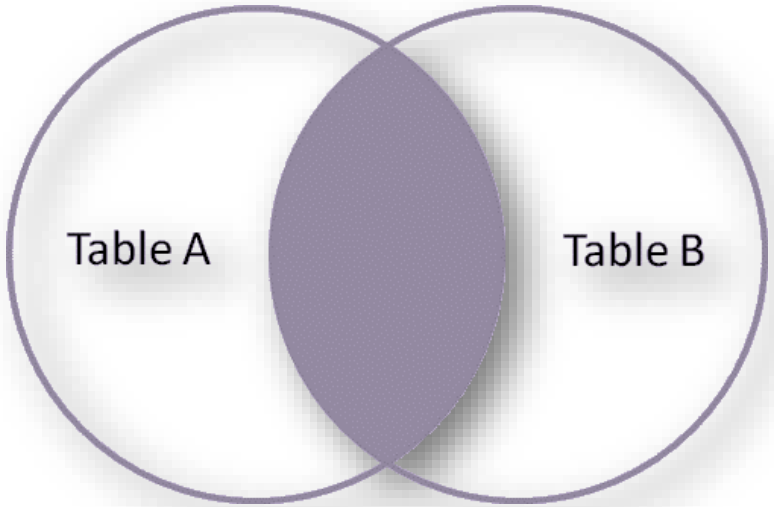
Student_Details1

| s_id | s_name | s_marks |
|---|---|---|
| 3 | Anne | 73 |
| 4 | Julia | 92 |
| 5 | Matt | 65 |

Student_Details2

# Intersect Operator

Intersect Operator helps to combine two select statements and returns the records which are common to both the select statements.




A ∩ B

# Intersect Operator: Syntax


```
SELECT column_list FROM table1
INTERSECT
SELECT column_list FROM table2
```

# Intersect Operator

| s_id | s_name | s_marks |
|---|---|---|
| 1 | Sam | 45 |
| 2 | Bob | 87 |
| 3 | Anne | 73 |
| 4 | Julia | 92 |

Student_Details1

| s_id | s_name | s_marks |
|---|---|---|
| 3 | Anne | 73 |
| 4 | Julia | 92 |
| 5 | Matt | 65 |

Student_Details2

# Quiz

# Quiz

**What is the purpose of ‘Order By’ clause?**

| | |
|---|---|
| A | Sorting your result set using column data |
| B | Aggregation of fields |
| C | Sorting your result set using row data |
| D | None of these |

# Solution

**What is the purpose of ‘Order By’ clause?**

| | |
|---|---|
| **A** | Sorting your result set using column data |
| **B** | Aggregation of fields |
| **C** | Sorting your result set using row data |
| **D** | None of these |

# Quiz

**What is the purpose of ‘Group By’ clause?**

| | |
|---|---|
| A | Group data by column names |
| B | Group data by row values |
| C | Group data by column & row values |
| D | None of these |

# Solution

**What is the purpose of ‘Group By’ clause?**

| | |
|---|---|
| **A** | Group data by column names |
| **B** | Group data by row values |
| **C** | Group data by column & row values |
| **D** | None of these |

# Quiz

**Which of these is used with aggregate functions?**

| | |
|---|---|
| A | Select |
| B | Where |
| C | Group By |
| D | None of these |

# Solution

**Which of these is used with aggregate functions?**

| | |
|---|---|
| A | Select |
| B | Where |
| C | Group By |
| D | None of these |

# Quiz

**Which of these is the correct syntax for union operator?**

| | |
|---|---|
| A | SELECT column_name(s) FROM table_name1 UNION table_name2 |
| B | SELECT column_name(s) FROM table_name1<br>UNION<br>SELECT column_name(s) FROM table_name2 |
| C | UNION SELECT column_name(s) FROM table_name1<br>SELECT column_name(s) FROM table_name2 |
| D | SELECT FROM table_name1 AND table_name2 |

# Solution

**Which of these is the correct syntax for union operator?**

| | |
|---|---|
| A | SELECT column_name(s) FROM table_name1 UNION table_name2 |
| B | SELECT column_name(s) FROM table_name1<br>UNION<br>SELECT column_name(s) FROM table_name2 |
| C | UNION SELECT column_name(s) FROM table_name1<br>SELECT column_name(s) FROM table_name2 |
| D | SELECT FROM table_name1 AND table_name2 |


# Thank You