Database Management System

Unit 6: Retrieving Data using SQL

Retrieving Data using SQL

6.1 Operators

6.1.1 Arithmetic Operators

6.1.2 Relational Operators (Comparison Operators)

6.1.3 Logical Operators

6.1.4 Other Operators

6.1.5 Set Operators

CE: 6.1.1
Arithmetic Operators

Arithmetic Operators:

Operator	Use
+	Addition
-	Subtraction
/	Division
*	Multiplication
**	Exponentiation
()	Enclosed operation

CE: 6.1.2
Relational Operators

Relational Operators:

Operator	Use
=	Equal to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
<>, !=	Not equal to

CE: 6.1.3
Logical Operators

Logical Operators:

Operator	Use
AND	Returns TRUE, if both conditions are TRUE.
OR	Returns TRUE, if either one condition is TRUE.
NOT	Negate a condition to its opposite. NOT TRUE => FALSE NOT FALSE => TRUE NOT UNKNOWN => UNKNOWN

CE: 6.1.4
Other Operators

Other Operators:

- **BETWEEN…AND**
- **LIKE**
- **IN and NOT IN**
- **IS NULL and IS NOT NULL**

6.1.4.1 Condition Based on a Range Searching:

BETWEEN operator:

- It is used to select data that is within a ***range*** of user specified values.
- The ***range*** includes both lower value and the upper value.
- The values in between the range must be linked with the logical operator **AND.**

6.1.4.2 Condition Based on a Pattern Matching:

LIKE predicate:

- The SQL includes string-matching operator using **LIKE**.
- LIKE predicate allows comparisons of one string value with another string value using patterns.
- Patterns are described using two wildcard characters:
  1. Percentage (%)
  2. Underscore (_)

6.1.4.2 Condition Based on a Pattern Matching:

LIKE predicate: (Conti…)

1. **“%”** – allows to find a match for any string of any length, including zero length.
2. **“_”** – allows to find a match for any single character.

- The patterns are case-sensitive.
- Thus, LIKE is used to select rows containing columns that match a wildcard pattern.

6.1.4.3 Condition Based on a List:

IN and NOT IN Predicates:

➢ IN predicate:

- It is used to compare to a list of values.
- It selects the values that ***match*** any value in a given list of values.
- It helps to reduce the need of multiple OR conditions.

➢ NOT IN predicate:

- It is opposite of IN predicate.
- It select the values that ***do not match*** any value in a given list of values.

6.1.4.4 Searching for NULL:

- The NULL value in a column can be searched in a table, using **IS NULL** in the where clause.

- And the Non-NULL values in a table can also be listed using **IS NOT NULL**.

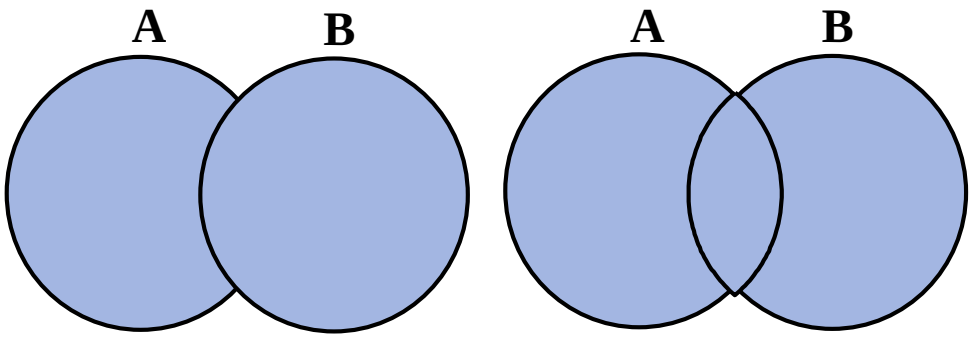
Note: Relational operators like =, >, <, <>, etc. can't be used with NULL.

CE: 6.1.5
Set Operators

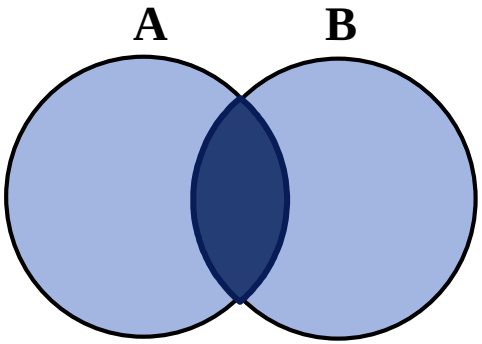
6.1.5 Set Operators:

- Set operators are used to combine the multiple queries into a single query.

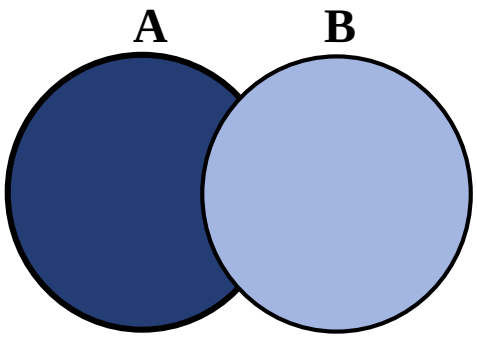
6.1.5 Set Operators: (Conti…)



UNION/
UNION ALL

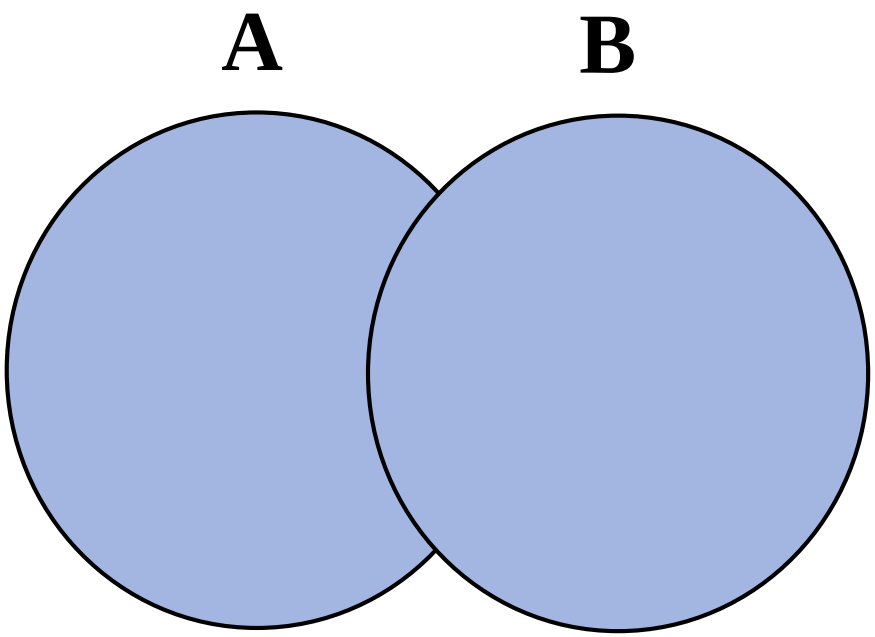


INTERSECT



MINUS

UNION Operator:



- The UNION operator returns results from both the queries by sorting and eliminating the duplications.

Consider the tblInstitute and tblDepartment tables for further examples

```
mysql> select * from tblInstitute;
+----+--------+
| id | name   |
+----+--------+
|  1 | BMIIT  |
|  2 | SRIMCA |
|  3 | DCST   |
|  4 | CGPIT  |
|  5 | MPC    |
+----+--------+
5 rows in set (0.00 sec)

mysql> select * from tblDepartment;
+----+-------------+-------------+
| id | name        | instituteid |
+----+-------------+-------------+
|  1 | MSC IT      |           1 |
|  2 | Int. MSC IT |           1 |
|  3 | BCA         |           2 |
|  4 | MCA         |           2 |
|  5 | B. TECH     |           4 |
+----+-------------+-------------+
5 rows in set (0.00 sec)
```

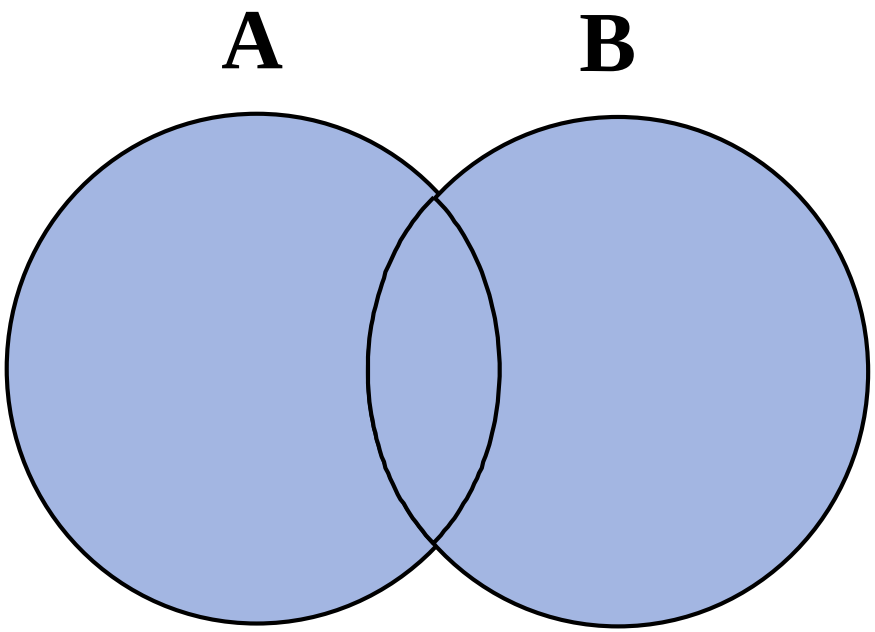
UNION Operator: (Conti…)

- **For example:**

  Display each institute of all departments only once.

```
mysql> select id from tblInstitute union select instituteid from tblDepartment;
+------+
| id   |
+------+
|    1 |
|    2 |
|    3 |
|    4 |
|    5 |
+------+
5 rows in set (0.00 sec)
```

UNION ALL Operator:



- The UNION ALL operator returns results from both the queries without sorting or removing duplications.

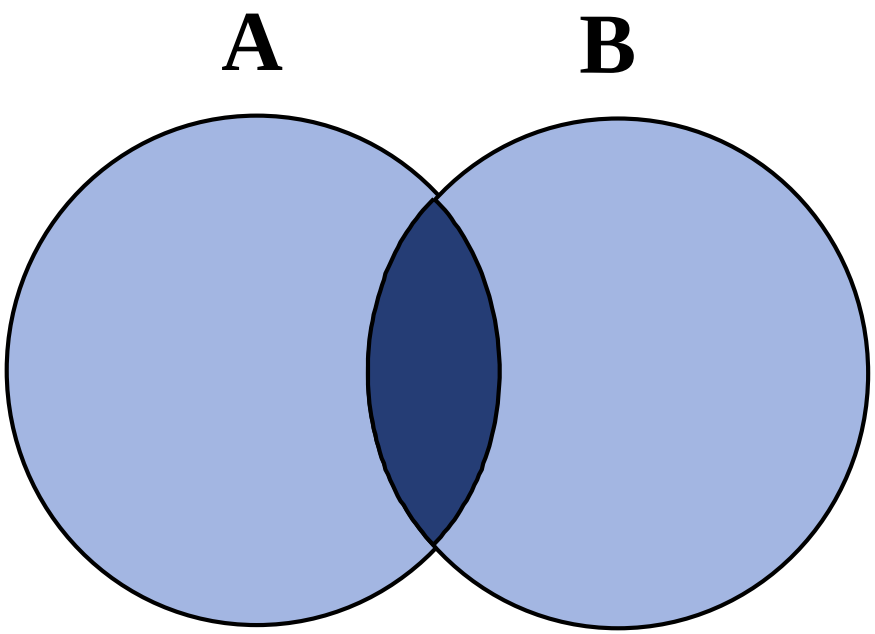
UNION ALL Operator: (Conti…)

- **For example:**

  Display the institute of all departments.

```
mysql> select id from tblInstitute union all select instituteid from tblDepartment;
+------+
| id   |
+------+
|    1 |
|    2 |
|    3 |
|    4 |
|    5 |
|    1 |
|    1 |
|    2 |
|    2 |
|    4 |
+------+
10 rows in set (0.00 sec)
```

INTERSECT Operator:



- The INTERSECT operator returns rows that are common in both queries by sorting and removing the duplications.

INTERSECT Operator: (Conti…)

- **For example:**

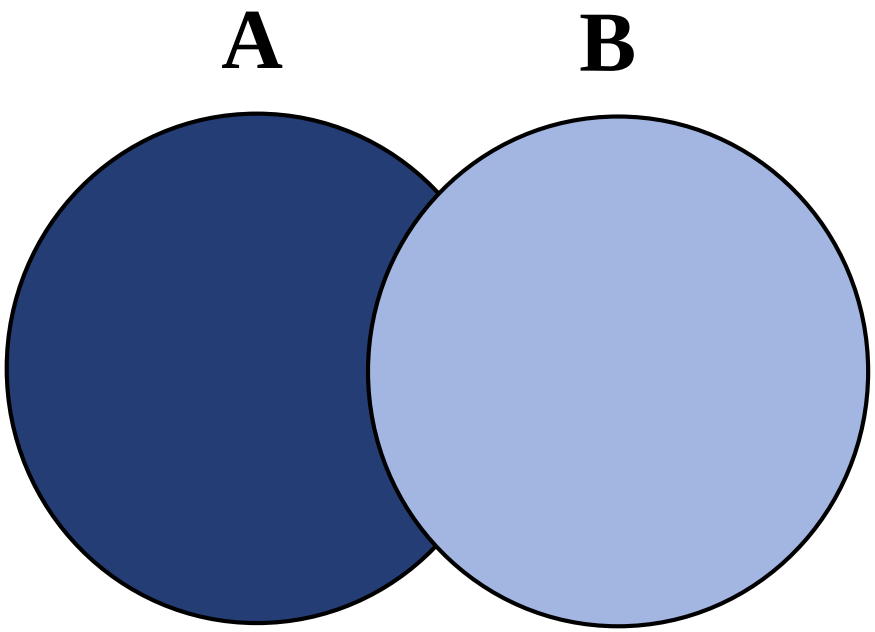
  Display only those institutes who has departments.

```
SQL> select id from tblInstitute intersect select instituteid from tblDepartment;

        ID
----------
         1
         2
         4
```

MySQL doesn't support INTERSECT operator.

MINUS Operator:



- The MINUS operator returns rows in the first query that are not present in the second query by sorting and removing the duplications.

MINUS Operator: (Conti…)

- **For example:**

  Display only those institutes who does not have departments.

```
SQL> select id from tblInstitute minus select instituteid from tblDepartment;

        ID
----------
         3
         5
```

MySQL doesn't support MINUS operator.

6.1.5 Set Operators: (Conti…)

- Set Operator Guidelines:

✓ The expressions in the SELECT lists must match in number and data type.

✓ Parentheses can be used for the sequence of execution.

✓ The ORDER BY clause:

- Can appear only at the very end of the statement.
- Will accept the column name, aliases from the first SELECT statement, or the positional notation.

6.1.5 Set Operators: (Conti…)

- ✓ Duplicate rows are automatically eliminated except in UNION ALL.
- ✓ The output is sorted in ascending order by default except in UNION ALL.
- ✓ Column names from the first query appear in the result.

Summary of Set Operators

- Use of UNION - to return all distinct rows.
- Use of UNION ALL - to return all rows, including duplicates.
- Use of INTERSECT - to return all rows that are shared by both the queries.
- Use of MINUS - to return all distinct rows, that are selected by the first query but not by the second.
- Use of ORDER BY - only at the very end of the statement.

International Certification Question

Q1.	**Which of these set operators will not sort the rows?**
A.	**UNION**
B.	**UNION ALL**
C.	**INTERSECT**
D.	**MINUS**

Ans: B

International Certification Question

Q2.	**Which of these operators will remove duplicate rows from the final result?**
A.	**UNION**
B.	**UNION ALL**
C.	**INTERSECT**
D.	**MINUS**

Ans: A, C and D

International Certification Question

Q3.	**What does UNION operator do in a SQL Server statement?**
A.	**Bring common data from the listed tables.**
B.	**Bring data which is not common from the listed tables.**
C.	**Bring all data from the listed tables.**
D.	**Bring all distinct from the listed tables.**

Ans: D

International Certification Question

Q4.	**How can we get all records from union operator?**
A.	**Using ‘ALL’ operator with UNION.**
B.	**Using ‘DISTINCT’ operator with UNION.**
C.	**We get all records with UNION operator by default.**
D.	**None of the above.**

Ans: A

International Certification Question

Q5.	**Is UNION or UNION ALL operator valid for long datatype column?**
A.	**True**
B.	**False**

Ans: B

Industry Interview Questions

1. What are use of set operators?
2. Which operators will remove duplicate rows from the final result?
3. What is the difference between UNION and UNION ALL?
4. What is the use of INTERSECT operator?
5. What is the use of MINUS operator?

Home Work

1. What are set operators? [1 Mark]
2. State the use of INTERSECT operator. [1 Mark]
3. What is the usage MINUS operator? [1 Mark]
4. List the set operations in SQL. [2 Marks]
5. How many types of set operators are there in SQL? Which are they? [2 Marks]
6. What is the difference between UNION and UNION ALL? [2 Marks]
7. What are the different set operators in SQL? Demonstrate how they are used in query giving examples. [5 Marks]

Thank You