Basic Operators in Oracle SQL

1. Arithmetic Operators

These operators perform mathematical operations on numerical values.

Operator	Description	Example
+	Addition	SELECT 10 + 5 FROM DUAL; -> 15
-	Subtraction	SELECT 10 - 5 FROM DUAL; -> 5
*	Multiplication	SELECT 10 * 5 FROM DUAL; -> 50
/	Division	SELECT 10 / 5 FROM DUAL; -> 2
MOD	Modulus (Remainder)	SELECT MOD(10, 3) FROM DUAL; -> 1

---

2. Comparison (Relational) Operators

These operators compare two values and return TRUE or FALSE.

Operator	Description	Example
=	Equal to	SELECT * FROM EMPLOYEES WHERE SALARY = 5000;
!= or <>	Not equal to	SELECT * FROM EMPLOYEES WHERE DEPT_ID != 10;
>	Greater than	SELECT * FROM EMPLOYEES WHERE SALARY > 5000;
<	Less than	SELECT * FROM EMPLOYEES WHERE SALARY < 5000;
>=	Greater than or equal to	SELECT * FROM EMPLOYEES WHERE SALARY >= 5000;
<=	Less than or equal to	SELECT * FROM EMPLOYEES WHERE SALARY <= 5000;
BETWEEN	Within a range	SELECT * FROM EMPLOYEES WHERE SALARY BETWEEN 3000 AND 7000;
IN	Matches any value in a list	SELECT * FROM EMPLOYEES WHERE DEPT_ID IN (10, 20, 30);

LIKE | Pattern matching | SELECT * FROM EMPLOYEES WHERE NAME LIKE 'J%';
IS NULL | Check for NULL values | SELECT * FROM EMPLOYEES WHERE MANAGER_ID IS NULL;

---

3. Logical Operators

Operator | Description | Example
---------|------------|--------
AND | Returns TRUE if both conditions are true | SELECT * FROM EMPLOYEES WHERE DEPT_ID = 10 AND SALARY > 3000;
OR | Returns TRUE if at least one condition is true | SELECT * FROM EMPLOYEES WHERE DEPT_ID = 10 OR SALARY > 5000;
NOT | Reverses the logical condition | SELECT * FROM EMPLOYEES WHERE NOT DEPT_ID = 10;

---

4. Set Operators

Operator | Description | Example
---------|------------|--------
UNION | Combines result sets (removes duplicates) | SELECT NAME FROM EMPLOYEES UNION SELECT NAME FROM CUSTOMERS;
UNION ALL | Combines result sets (includes duplicates) | SELECT NAME FROM EMPLOYEES UNION ALL SELECT NAME FROM CUSTOMERS;
INTERSECT | Returns common records | SELECT NAME FROM EMPLOYEES INTERSECT SELECT NAME FROM CUSTOMERS;
MINUS | Returns records from first query that are not in second | SELECT NAME FROM EMPLOYEES MINUS SELECT NAME FROM CUSTOMERS;

---

5. Concatenation Operator

Operator | Description | Example
---------|------------|--------
|| | Concatenates two strings | SELECT 'Hello' || ' World' FROM DUAL; -> Hello World

---

6. Miscellaneous Operators

Operator | Description | Example
---------|------------|--------
EXISTS | Checks if subquery returns any rows | SELECT * FROM EMPLOYEES WHERE EXISTS (SELECT 1 FROM DEPARTMENTS WHERE DEPARTMENTS.ID = EMPLOYEES.DEPT_ID);
ANY | Compares value with any returned by subquery | SELECT * FROM EMPLOYEES WHERE SALARY > ANY (SELECT SALARY FROM EMPLOYEES WHERE DEPT_ID = 10);
ALL | Compares value with all returned by subquery | SELECT * FROM EMPLOYEES WHERE SALARY > ALL (SELECT SALARY FROM EMPLOYEES WHERE DEPT_ID = 10);

Summary:
- Arithmetic, Comparison, Logical, Set, and Concatenation operators provide powerful data manipulation.
- Use EXISTS, ANY, and ALL for advanced subqueries.
- UNION, INTERSECT, and MINUS allow result set combinations.