

Relational Algebra:

Relational Algebra is a procedural query language which takes a relation as an input and generates a relation as an output.

Basic Operator	Semantic
σ (Selection)	Select rows based on given condition
π (Projection)	Project some columns
X (Cross Product)	Cross product of relations, returns m*n rows where m and n are number of rows in R1 and R2 respectively.
U (Union)	Return those tuples which are either in R1 or in R2. Max no. of rows returned = m+n and Min no. of rows returned = max(m,n)
-(Minus)	R1-R2 returns those tuples which are in R1 but not in R2. Max no. of rows returned = m and Min no. of rows returned = m-n
ρ (Rename)	Renaming a relation to another relation.
Extended Operator	Semantic
\cap (Intersection)	Returns those tuples which are in both R1 and R2. Max no. of rows returned = min(m,n) and Min no. of rows returned = 0
\bowtie (Conditional Join)	Selection from two or more tables based on some condition (Cross product followed by selection)
\bowtie (Equi Join)	It is a special case of conditional join when only equality conditions are applied between attributes.

⋈(Natural Join)	<p>In natural join, equality conditions on common attributes hold and duplicate attributes are removed by default.</p> <p>Note: Natural Join is equivalent to cross product if two relations have no attribute in common and natural join of a relation R with itself will return R only.</p>
⋈(Left Outer Join)	<p>When applying join on two relations R and S, some tuples of R or S do not appear in the result set which does not satisfy the join conditions. But Left Outer Joins gives all tuples of R in the result set. The tuples of R which do not satisfy the join condition will have values as NULL for attributes of S.</p>
⋈(Right Outer Join)	<p>When applying join on two relations R and S, some tuples of R or S do not appear in the result set which does not satisfy the join conditions. But Right Outer Joins gives all tuples of S in the result set. The tuples of S which do not satisfy the join condition will have values as NULL for attributes of R.</p>
⋈(Full Outer Join) /(Division Operator)	<p>When applying join on two relations R and S, some tuples of R or S do not appear in the result set which does not satisfy the join conditions. But Full Outer Joins gives all tuples of S and all tuples of R in the result set. The tuples of S which do not satisfy the join condition will have values as NULL for attributes of R and vice <u>versa</u>.</p> <p>Division operator A/B will return those tuples in A which are associated with every tuple of B. Note: Attributes of B should be a proper subset of attributes of A. The attributes in A/B will be Attributes of A- Attribute of B.</p>

File Structures:

- **Primary Index:** A primary index is an ordered file, records of fixed length with two fields. First field is the same as the primary key as a data file and the second field is a pointer to the data block, where the key is available. The average number of block accesses using index = $\log_2 B_i + 1$, where B_i = number of index blocks.
- **Clustering Index:** Clustering index is created on data file whose records are physically ordered on a non-key field (called Clustering field).
- **Secondary Index:** Secondary index provides secondary means of accessing a file for which primary access already exists.

SQL

DDL:

DDL is short name of **Data Definition Language**, which deals with database schemas and descriptions, of how the data should reside in the database.

- CREATE - to create a database and its objects like (table, index, views, store procedure, function, and triggers)
- ALTER - alters the structure of the existing database
- DROP - delete objects from the database
- TRUNCATE - remove all records from a table, including all spaces allocated for the records are removed
- RENAME - rename an object

DML:

DML is short name of **Data Manipulation Language** which deals with data manipulation and includes most common SQL statements such SELECT, INSERT, UPDATE, DELETE, etc., and it is used to store, modify, retrieve, delete and update data in a database.

- SELECT - retrieve data from a database
- INSERT - insert data into a table
- UPDATE - updates existing data within a table
- DELETE - Delete all records from a database table
- MERGE - UPSERT operation (insert or update)

DCL:

DCL is short name of **Data Control Language** which includes commands such as GRANT and mostly concerned with rights, permissions and other controls of the database system.

- GRANT - allow users access privileges to the database
- REVOKE - withdraw users access privileges given by using the GRANT command

TCL:

TCL is short name of Transaction Control Language which deals with a transaction within a database.

- COMMIT - commits a Transaction
- ROLLBACK - rollback a transaction in case of any error occurs

SAVEPOINT - to roll back the transaction making points within groups

SQL is a standard language for storing, manipulating and retrieving data in databases.

SELECT:

The SELECT statement is used to select data from a database.

Syntax -

- SELECT *column1, column2, ...*
FROM *table_name*;
- Here, column1, column2, ... are the field names of the table you want to select data from. If you want to select all the fields available in the table, use the following syntax: ● SELECT * FROM *table_name*;

Ex –

- SELECT CustomerName, City FROM Customers;

SELECT DISTINCT:

The SELECT DISTINCT statement is used to return only distinct (different) values.

Syntax –

- SELECT DISTINCT *column1, column2, ...* FROM *table_name*;

Ex –

- SELECT DISTINCT Country FROM Customers;

WHERE:

The WHERE clause is used to filter records.

Syntax –

- SELECT *column1, column2, ...* FROM *table_name*
WHERE *condition*;

Ex –

- SELECT * FROM Customers
WHERE Country='Mexico';

Operator	Description
----------	-------------

=	Equal
>	Greater than
<	Less than
>=	Greater than or equal
<=	Less than or equal
<>	Not equal. Note: In some versions of SQL this operator may be written as !=

AND, OR and NOT:

The WHERE clause can be combined with AND, OR, and NOT operators.

The AND and OR operators are used to filter records based on more than one condition:

- The AND operator displays a record if all the conditions separated by AND are TRUE.
- The OR operator displays a record if any of the conditions separated by OR is TRUE.

The NOT operator displays a record if the condition(s) is NOT TRUE.

Syntax –

- *SELECT column1, column2, ... FROM table_name
WHERE condition1 AND condition2 AND condition3 ...;*
- *SELECT column1, column2, ... FROM table_name
WHERE condition1 OR condition2 OR condition3 ...;*
- *SELECT column1, column2, ... FROM table_name
WHERE NOT condition;*

Ex –

- *SELECT * FROM Customers
WHERE Country='Germany' AND City='Berlin';*
- *SELECT * FROM Customers
WHERE Country='Germany' AND (City='Berlin' OR City='München');*

ORDER BY:

The ORDER BY keyword is used to sort the result-set in ascending or descending order.

The ORDER BY keyword sorts the records in ascending order by default. To sort the records in descending order, use the DESC keyword.

Syntax –

- SELECT *column1, column2, ...* FROM *table_name*
ORDER BY *column1, column2, ...* ASC|DESC;

Ex –

- SELECT * FROM Customers ORDER BY Country;
- SELECT * FROM Customers
ORDER BY Country ASC, CustomerName DESC;

INSERT INTO:

The INSERT INTO statement is used to insert new records in a table.

Syntax –

- INSERT INTO *table_name* (*column1, column2, column3, ...*)
VALUES (*value1, value2, value3, ...*);
- INSERT INTO *table_name*
VALUES (*value1, value2, value3, ...*);

*In the second syntax, make sure the order of the values is in the same order as the columns in the table. **Ex –**

- INSERT INTO Customers (CustomerName, ContactName, Address, City, PostalCode, Country)
VALUES ('Cardinal', 'Tom B. Erichsen', 'Skagen 21', 'Stavanger', '4006', 'Norway');

NULL Value:

It is not possible to test for NULL values with comparison operators, such as =, <, or <>. We will have to use the IS NULL and IS NOT NULL operators instead.

Syntax –

- SELECT *column_names* FROM *table_name*
WHERE *column_name* IS NULL;
- SELECT *column_names* FROM *table_name*

WHERE *column_name* IS NOT NULL;

Ex –

- SELECT CustomerName, ContactName, Address
FROM Customers
WHERE Address IS NULL;

UPDATE:

The UPDATE statement is used to modify the existing records in a table.

Syntax –

- UPDATE *table_name*
SET *column1* = *value1*, *column2* = *value2*, ...
WHERE *condition*;

Ex –

- UPDATE Customers
SET ContactName = 'Alfred Schmidt', City= 'Frankfurt'
WHERE CustomerID = 1;

DELETE:

The DELETE statement is used to delete existing records in a table.

Syntax –

- DELETE FROM *table_name* WHERE *condition*;
- DELETE FROM *table_name*;

In 2nd syntax, all rows are deleted. The table structure, attributes, and indexes will be intact

Ex –

- DELETE FROM Customers WHERE
CustomerName='Alfreds Futterkiste';

SELECT TOP:

The SELECT TOP clause is used to specify the number of records to return.

Syntax –

- SELECT TOP *number*|*percent* *column_name(s)*
FROM *table_name*
WHERE *condition*;

- `SELECT column_name(s) FROM table_name
WHERE condition
LIMIT number;`
- `SELECT column_name(s) FROM table_name
ORDER BY column_name(s)
FETCH FIRST number ROWS ONLY;`
- `SELECT column_name(s) FROM table_name
WHERE ROWNUM <= number;`

*In case the interviewer asks other than the TOP, rest are also correct. (Diff. DB

Systems) **Ex –**

- `SELECT TOP 3 * FROM Customers;`
- `SELECT * FROM Customers LIMIT 3;`
- `SELECT * FROM Customers
FETCH FIRST 3 ROWS ONLY;`