

Q1. What is SQL, and why is it essential in database management?

ANS :-

- SQL stands for Structured Query Language SQL is a standard language for storing, manipulating and retrieving data in databases.
- SQL allows you to access and manipulate the databases.
- To use SQL in: MySQL, SQL Server, MS Access, Oracle, Sybase, Informix, Postgres, and other database systems.
- It is a standard language for Relational Database System. It enables a user to create, read, update and delete relational databases and tables.

Q2. Explain the difference between DBMS and RDBMS.

ANS :-

RDBMS	DBMS
Data stored is in table format	Data stored is in the file format
Data in the form of a table are linked together	No connection between data
Support distributed database	No support for distributed database
Data is stored in a large amount	Data stored is a small quantity
RDBMS supports multiple users	DBMS supports a single user
The software and hardware requirements are higher	The software and hardware requirements are low
Example: Oracle, SQL Server.	Example: XML, Microsoft Access.

Q3. Describe the role of SQL in managing relational databases.

ANS :-A relational database stores information in tabular form, with rows and columns representing different data attributes and the various relationships between the data values.

Its main roles include:-

1. **Data Definition:** Creating, modifying, and deleting database structures like tables, indexes, and schemas using commands like CREATE, ALTER, and DROP.
2. **Data Manipulation:** Inserting, updating, deleting, and retrieving data from tables using commands like INSERT, UPDATE, DELETE, and SELECT.
3. **Data Control:** Managing access and permissions for users to ensure security and privacy with commands like GRANT, REVOKE, and DENY.
4. **Data Querying:** Extracting specific information from the database, filtering and sorting data using SELECT queries with conditions, joins, and aggregations.
5. **Data Integrity:** Enforcing rules (like primary keys, foreign keys, and constraints) to maintain data accuracy and consistency.

Q4. What are the key features of SQL?

ANS :-

1. **DDL (Data Definition Language):** Used to create, modify, or delete database structures (e.g., tables).
2. **DML (Data Manipulation Language):** Used to insert, update, or delete data in the database.
3. **Query Language:** Allows querying data, filtering, sorting, grouping, and joining tables.
4. **Transaction Control:** Enables grouping operations into transactions, which can be rolled back if needed.
5. **Data Integrity:** Ensures data accuracy with constraints and referential integrity.

6. **User Access Control:** Manages user permissions to control who can perform actions in the database.
7. **Portability:** SQL is standardized, making it easy to use across different database systems with minimal changes.

2. SQL Syntax

Q1. What are the basic components of SQL syntax?

ANS :-The primary components include tables , queries , clauses , the select statement , the insert statement , data types and expressions.

- **Keywords:-**

Reserved words that have a specific meaning in SQL.

They are typically used to define actions like SELECT, INSERT, UPDATE, DELETE, WHERE, FROM, JOIN, ORDER BY, etc.

- **Expressions:-**

An expression is a combination of operators, constants, functions, and variables that evaluate to a single value. For example, salary * 1.1 or age > 30.

- **Clauses:-**

SQL queries are composed of different clauses that define specific actions or conditions.

GROUP BY , HAVING , ORDER BY.

- **Aggregate functions:-** COUNT(), SUM(), AVG(), MAX(), MIN().

- **Arithmetic Operators :-**

+ : Addition , - : Subtraction , * : Multiplication , / : Division , % : Modulus (remainder).

- **Comparison Operators :-**

= : Equal to , || or , > : Greater than, < : Less than, >= : Greater than or equal to, <= : Less than or equal to.

- **Logical Operators :-**

AND :- Both conditions must be true

OR :- At least one condition must be true

NOT :-Reverses condition

Q2. Write the general structure of an SQL SELECT statement.

ANS :-

SELECT * FROM Employee;

SELECT Emp_No, Emp_Name from Employee;

SELECT DISTINCT * FROM Employee;

SELECT DISTINCT Emp_No, Emp_Name FROM Employee;

Q3. Explain the role of clauses in SQL statements.

ANS :-

in SQL mainly 4 types of clauses.

1) Where clauses :-

- The Where clause is used to filter records.
- It is used to extract only those records that fulfill a specified condition.
- SELECT * from table_name where condition;
- SELECT * from employees where id=101;

2) Group By clauses :-

- Groups those rows that have the same values into summary rows.
- It collects data from multiple records and groups the results by one or more columns.
- Generally we use the group by with some aggregate functions.
- The GROUP BY statement is used with the SQL SELECT statement.
- **Syntax** :- SELECT column FROM table_name WHERE conditions GROUP BY column
- SELECT COMPANY, COUNT(*) FROM PRODUCT_MAST GROUP BY COMPANY;

3) Having Clauses :-

- HAVING clause is used to specify a search condition for a group or an aggregate.
- Having is used in a GROUP BY clause. If you are not using GROUP BY clause then you can use HAVING function like a WHERE clause.

4) Order By clauses :-

- The ORDER BY clause sorts the result-set in ascending or descending order.
- It sorts the records in ascending order by default.
- DESC keyword is used to sort the records in descending order.

Q1. What are constraints in SQL? List and explain the different types of constraints.

ANS :-

- SQL constraints are used to specify rules for the data in a table.
- Constraints are used to limit the type of data that can go into a table.
- Constraints can be column level or table level. Column level constraints apply to a column, and table level constraints apply to the whole table.

The following constraints are commonly used in SQL: -

- **NOT NULL** - Ensures that a column cannot have a NULL value
- **UNIQUE** - Ensures that all values in a column are different
- **PRIMARY KEY** - A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table
- **FOREIGN KEY** - Prevents actions that would destroy links between tables
- **CHECK** - Ensures that the values in a column satisfies a specific condition
- **DEFAULT** - Sets a default value for a column if no value is specified
- **CREATE INDEX** - Used to create and retrieve data from the database very quickly

Q2. How do PRIMARY KEY and FOREIGN KEY constraints differ?

ANS :-

PRIMARY KEY	FOREIGN KEY
A primary key is used to ensure data in the specific column is unique and data is not null.	A foreign key is a column or group of column in a RDBMS table that provides a link between data in two tables.
Only one primary key is allowed in data base table.	More than one foreign key is allowed in a table.
It defines a record in the RDBMS.	It refers to the table which is the primary key of another table.
It is combination of UNIQUE and null values.	It can contain duplicate values and a table in relational.
Its value cannot be deleted from the parent table.	Its value can be deleted from the child table.

Q3. What is the role of NOT NULL and UNIQUE constraints?

ANS :-

NOT NULL – The NOT NULL Constraint is a rule that a column cannot have a null value

UNIQUE –The UNIQUE Constraint is a rule that all values in a column are different.

A unique constraints also referred to as a unique key constraint.

4. Main SQL Commands and Sub-commands(DDL)

Q1. Define the SQL Data Definition Language (DDL).

ANS :-

- DDL stands for Data Definition Language.
- DDL Data Definition Language Actually consists of the SQL commands that can be used to defining , altering , and deleting database structures such as table And schemas.
- All the command of DDL are auto-committed that means it permanently save all the changes in the database.

Here are some commands that come under DDL: -

- CREATE
- ALTER
- DROP
- TRUNCAT
- RENAME

IN DDL many common commands :-

1) Create Command :-Create database or its objects , table , view , function , procedure , triggers,

Syntax :-Create table table_name(column 1 data_type , column 2 data_type ..);

Example :- create table student(s_idint , s_name varchar(60) , s_city text , s_stdint);

2) Alter Command :-

It is used to alter the structure of the database.

Alter command use to modify , add , drop table and column.

Syntax:-

ALTER TABLE table_name ADD column_name COLUMN-definition;

ALTER TABLE MODIFY(COLUMN DEFINITION....);

Example :-

ALTER TABLE STU_DETAILS ADD(ADDRESS VARCHAR2(20));

ALTER TABLE STU_DETAILS MODIFY (NAME VARCHAR2(20));

3) Drop Command :- It is used to delete both the structure and record stored in the table.

Syntax :-DROP TABLE ;

Example: DROP TABLE EMPLOYEE;

4) TRUNCATE:-It is used to delete all the rows or records from the table and free the space containing the table.

Syntax: -TRUNCATE TABLE Table_name;

Example:-TRUNCATE TABLE EMPLOYEE;

5) Rename :-Rename an column or table in database.

Syntax:- rename table old_name to new_name;

Q2. Explain the CREATE command and its syntax.

ANS :-

1) Create Command :-Create database or its objects , table , view , function , procedure , triggers,

Syntax :-Create table table_name(column 1 data_type , column 2 data_type ..);

Example :- create table student(s_idint , s_name varchar(60) , s_city text , s_stdint);

Q3. What is the purpose of specifying data types and constraints during table creation?

ANS :-

Data type:-define the type of data that can be stored in a column.

Purpose of specifying data types :-

Datatype is a guideline for SQL to understand what type of data is expected inside of each column , and what data stored in column.

Basically used datatype in SQL:-

- **Int:**declare the positive value.
- **Float:**declare decimal value.
- **Char(size) :-** A fixed length string .can be from 0 to 255.
- **Varchar(size):**declare string value and declare size . can be from 0 to 65535.
- **BLOB(size):-**holds up to 65,535 bytes of data.
- **Text:** declare string value and not declare size.
- **Date:**declare date automatically YYYY-MM-DD.
- **Datetime:** declare date and time YYYY-MM-DD and hh-mm-ss.
- **Timestamp :-**A timestamp values are stored as the number of current time in your system.

- **Time:** declare time hh-mm-ss.

Purpose of CONSTRAINTS :-

- SQL constraints are used to specify rules for the data in a table.
- Constraints are used to limit the type of data that can go into a table.
- Constraints can be column level or table level. Column level constraints apply to a column, and table level constraints apply to the whole table.

5. ALTER Command**Q1. What is the use of the ALTER command in SQL?**

ANS :-

- It is used to alter the structure of the database.
- Alter command use to modify , add , drop table and column.

Q2. How can you add, modify, and drop columns from a table using ALTER?

ANS :-

ADD Syntax:- ALTER TABLE table_name ADD column_name COLUMN-definition;

MODIFY Syntax :-ALTER TABLE MODIFY(COLUMN DEFINITION....);

DROP Syntax :-Alter Table drop column_name;

Example :-ALTER TABLE STU_DETAILS ADD(ADDRESS VARCHAR2(20));

Example :-ALTER TABLE STU_DETAILS MODIFY (NAME VARCHAR2(20));

Example :-Alter table drop column name varchar(20);

6. DROP Command**Q1. What is the function of the DROP command in SQL?**

ANS :-

- It is used to delete both the structure and record stored in the table.
- Drop commands used to permanently deleted from database and they are cannot be rolled back.
- **Syntax :-** DROP TABLE ; , DROP DATABASE , DROP FUNCTION , DROP VIEW , DROP PROCEDURE.

Q2. What are the implications of dropping a table from a database?

ANS :-

DROP TABLE :- name of the table to be deleted .

DROP DATABASE :- name of the database to be deleted.

DROP VIEW :- name of the view to be deleted .

7. Data Manipulation Language (DML)

1. Define the INSERT, UPDATE, and DELETE commands in SQL.

ANS :-

- DML stands for Data manipulatipn language.
- DML commands are used to modify the database.
- It is responsible for all form of CHANGES in the database.
- The command of DML is not auto-committed that means it can't permanently save all the changes in the database. They can be rollback.

Here are some commands that come under DML: -

INSERT :-

- The INSERT statement is a SQL query.
- It is used to insert data into the row of a table.
- **Syntax:-** INSERT INTO TABLE_NAME (col1, col2, col3,... col N) VALUES (value1, value2, value3, valueN); OR
- INSERT INTO TABLE_NAME VALUES (value1, value2, value3, valueN);

- **Example:-** INSERT INTO XYZ (Author, Subject) VALUES ("Sonoo", "DBMS");

UPDATE :-

- This command is used to update or modify the value of a column in the table.
- **Syntax:-** UPDATE table_name SET [column_name1= value1] [WHERE CONDITION]
- **Example:** UPDATE students SET User_Name = 'Sonu' WHERE Student_Id = ';

DELETE :-

- Delete records from a database table .
- **Syntax :-** DELETE from table_name where condition;
- **Example :-** DELETE from employee where e_id=101;

2. What is the importance of the WHERE clause in UPDATE and DELETE operations?**ANS :-**

- The sql WHERE clause is used to filter records.
- It is used to extract only those records that fulfill a specified condition.
- In SQL updating records and deleting records entirely from a database , the where clause plays an importance role in defining which rows will be affected by the query.
- The where clause specifies with records that should be updated.

8. Data Query Language (DQL)**1. What is the SELECT statement, and how is it used to query data?****ANS :-**

- The SELECT Statement is a commands of DQL (data Query language) are used for retrieve data from the database .
- Select statement is return a result set of row from table.

- This is the same as the projection operation of relational algebra.
- It is used to select the attribute based on the condition described by WHERE clause.
- **Syntax:-** SELECT * FROM TABLES WHERE conditions;
- SELECT Emp_No, Emp_Name from Employee;
- SELECT DISTINCT * FROM Employee;
- **Example:-** SELECT emp_name FROM employee WHERE age > 20

2. Explain the use of the ORDER BY and WHERE clauses in SQL queries.

ANS :-

The ORDER BY clause :-

- The ORDER BY clause sorts the result-set in ascending or descending order.
- It sorts the records in ascending order by default.
- DESC keyword is used to sort the records in descending order.

Syntax :-

- SELECT column1, column2 FROM table_name WHERE condition ORDER BY column1 DESC;
- SELECT column1, column2 FROM table_name WHERE condition ORDER BY column1 DESC;

Example :-

- SELECT E_ID , E_NAME from EMP_1 WHERE E_ID=1 ORDER BY E_NAME;
- SELECT * FROM CUSTOMER ORDER BY NAME DESC;

The WHERE clause :-

- The sql WHERE clause is used to filter records.
- It is used to extract only those records that fulfill a specified condition.
- In SQL updating records and deleting records entiers from a database , the where clause plays an importance role in defining which rows will be affected by the query.

- The where clause specifies with records that should be updated.
- The where clause specifies with records that should be deleted.
- **NOTE :-** The Where clause only used to DML commands like insert , update and delete.

9. Data Control Language (DCL)

1. What is the purpose of GRANT and REVOKE in SQL?

ANS :-The GRANT and REVOKE is type of DCL (DATA CONTROLLING LANGUAGE) helps users to retrieve and modify the database with some specified queries.

GRANT:-

- It is used to give user access privileges to a database.
- It helps to provide any kind of access to any user.
- SQL GRANT command is used to provide privileges to database object for a user.
- **Syntax :-** GRANT SELECT , UPDATE ON table_name to user_name;

REVOKE:-

- It is used to take back permissions from the user.
- SQL REVOKE command is used to removes permission if any granted user to the users on database objects.
- **Syntax :-** REVOKE ALTER TABLE FROM user_name;

2. How do you manage privileges using these commands?

ANS :-

- IN DCL (DATA CONTROLLING LANGUAGE) provide two commands GRANT and REVOKE to privileges database .
- The GRANT command is manage to give the privileges (permission) to table to any user , means any user easily access any table data .

Example :-

- grant insert on accounts to ram; , GRANT UPDATE ON accounts to ram; , GRANT ALL ON accounts to ram;
- The REVOKE command is manage to take back the privileges any users.
- **Example :-** revoke insert on accounts from ram;

10. Transaction Control Language (TCL)

Q1. What is the purpose of the COMMIT and ROLLBACK commands in SQL?

ANS :-

COMMIT:- COMMIT command is used to save all the transactions to the database.

ROLLBACK:- ROLLBACK command is used to undo transactions that have not already been saved to the database.

2. Explain how transactions are managed in SQL databases.

ANS :-

- IN SQL manage the transaction manage used to TCL (TRANSACTION CONTROL LANGUAGE) command .
- TCL commands can only use with DML commands like INSERT, DELETE and UPDATE only.

Here are some commands that come under TCL: -

- COMMIT
- ROLLBACK
- SAVEPOINT

1) COMMIT :- The COMMIT command used to data saved Permenatly in table.

Syntax :- COMMIT ;

Example :- DELETE FROM CUSTOMERS WHERE AGE = 25; COMMIT;

ROLLBACK:-

- ROLLBACK command is used to undo transactions that have not already been saved to the database.
- BY default one time undo affect in table.

Syntax: ROLLBACK;

Example: DELETE FROM CUSTOMERS WHERE AGE = 25; ROLLBACK;

SAVEPOINT: It is used to roll the transaction back to a certain point without rolling back the entire transaction.

Syntax: SAVEPOINT SAVEPOINT_NAME;

11. SQL Joins

1. Explain the concept of JOIN in SQL. What is the difference between INNER JOIN, LEFT JOIN, RIGHT JOIN, and FULL OUTER JOIN?

ANS :-

- In SQL JOIN means to combine something.
- A JOIN clause is used to combine data from two or more tables , based on a related column between them.

Mainly 4 types of JOIN in SQL :-

1) INNER JOIN :- returns all records that have matching values in both tables.

2) LEFT JOIN OR LEFT OUTER JOIN :-returns all records from the left table , and the matched records from the right table.

3) RIGHT JOIN OR RIGHT OUTER JOIN :- returns all records from the right table , and the matched records from the left table.

4) FULL JOIN :- returns all records when there is a match in either left or right table.

2. How are joins used to combine data from multiple tables?

ANS :-JOIN mainly 4 types used to combine data from multiple tables.

1) INNER JOIN :- SELECT table1.column , table1.column , table2.column FROM table1 INNER JOIN table2 ON table1.matching column = table2.matching column;

2) LEFT JOIN :- SELECT table1.column , table1.column , table2.column FROM table1 LEFT JOIN table2 ON table1.matching column = table2.matching column;

3) RIGHT JOIN :- SELECT table1.column , table1.column , table2.column FROM table1 RIGHT JOIN table2 ON table1.matching column = table2.matching column;

4) FULL JOIN :- SELECT table1.column , table1.column , table2.column FROM table1 FULL JOIN table2 ON table1.matching column = table2.matching column;

Example :-

select product.pro_id, product.Pro_name, category.Cat_Name from product
INNER JOIN category on product.Cat_ID=category.Cat_ID; OR

SELECT * FROM customer INNER JOIN PAYMENT ON customer.c_id =
payment.c_id;

12. SQL Group By

Q1. What is the GROUP BY clause in SQL? How is it used with aggregate functions?

ANS :-

- The GROUP BY clause that groups all the with the same column value.
- SQL GROUP BY statement is used to arrange identical data into groups.
- The GROUP BY statement is used with the SQL SELECT statement.
- The GROUP BY statement follows the WHERE clause in a SELECT statement and precedes the ORDER BY clause.
- The GROUP BY statement is used with aggregation function.

How is used to Aggregate function :-

Simple words, there is any query that we have to count(), max(), min(), avg(), sum() a group of values from a table resolve this kind of issue in SQL we use GROUP BY clause with aggregate function.

Example :- SELECT department , COUNT(*) from employee GROUP BY department;

2. Explain the difference between GROUP BY and ORDER BY.

ANS :-

GROUP BY	ORDER BY
SQL GROUP BY statement is used to arrange identical data into groups.	The ORDER BY clause sorts the result-set in ascending or descending order.
GROUP BY statement is used to group the rows that have the same value.	The ORDER BY clause is used to set records in ascending or descending order.
It may be allowed in CREATE VIEW statement.	It does not allowed in CREATE VIEW statement.
Example :- SELECT COMPANY, COUNT(*) FROM PRODUCT_MAST GROUP BY COMPANY;	Example :- SELECT * FROM CUSTOMER ORDER BY NAME DES
IN SELECT statement, it is always used to before the order by keyword.	IN SELECT statement, it is always used to After the group by keyword.
GROUP BY controls the presentation of records(rows).	ORDER BY clause controls the presentation of columns.

13. SQL Stored Procedure

Q1. What is a stored procedure in SQL, and how does it differ from a standard SQL query?

ANS :-

- Procedure is like a function but it will never return any value. It will always performed by parameter(argument) or without parameter(argument).

- SQL Stored Procedure are a powerful feature in database management systems , is a collection of SQL statements to perform a specific task.
- Procedure is used to execute block of code to perform.

Q2. Explain the advantages of using stored procedures.

ANS :-

- 1) improved performance :-** they execute faster than running multiple individual queries.
- 2) Enhanced Security :-**user can be granted permission to execute stored procedure without tables.
- 3) Code Reusability :-** making it easier to maintain and update code.
- 4) Reduced Network Traffic :-** procedure use to multiple SQL statements in One call , so reduce network load .
- 5) Better Error Handling :-**SQL stored procedures provide a structured way to manage errors using blocks.

14. SQL View

Q1. What is a view in SQL, and how is it different from a table?

ANS :-

- In SQL , a VIEW is a virtual table based on the result-set of an SQL.
- A VIEW is created with the CREATE VIEW statement.
- Views in SQL are virtual tables created by querying data from one or more real tables in a database.
- They provide a powerful way to simplify complex queries, enhance data security, and display data presentation to specific user needs.

- You can add SQL statement and functions to a view and present data as if the data were coming from one single unit.

How is differ from table :-

- A view is virtual table used to manipulate some parts of the table OR result of a SQL query .
- A table is a database entity that stores data in the form of rows and columns.

Syntax :-

```
CREATE VIEW view_name AS  
SELECT column_name1, column_name2...column_nameN  
FROM table_name1  
WHERE condition;
```

Example :-

```
CREATE VIEW v_11 AS SELECT s_id ,s_name FROM student Where s_name LIKE  
'A%';  
SELECT * FROM v_11;
```

Q2. Explain the advantages of using views in SQL databases.**ANS :-**

- **Consistency** :-Seamless to make changes to any underlying table structure.
- Using a view in SQL to return data from the tables allow you to hide WHERE clause or columns
- YOU many write simplified select statements against views , there by handling complicated joins and queries.
- **Security** :- each user can be given permission to access the database only through a small set of views that contain.

15. SQL Triggers**Q1. What is a trigger in SQL? Describe its types and when they are used.**

ANS :-

- A trigger is a special type of stored procedure that automatically runs when an event occurs in the database server.
- DML triggers run when a user tries to modify data through a data manipulation language (DML) event.
- DML events are INSERT, UPDATE, or DELETE statements on a table or view.

They are 3 types of triggers :-

- 1) Data manipulation Language trigger
- 2) Data Definition Language trigger
- 3) Logon Trigger

1) AfterTriggers :- activated after data is inserted / updated / deleted.

2)Before Triggers :- activated Before data is inserted / updated / deleted.

After Triggers :- 1) After insert, 2) After Update , 3) After Delete.

Before Triggers :- 1)Before insert , 2) Before Update , 3) Before delete.

2. Explain the difference between INSERT, UPDATE, and DELETE triggers.

ANS :-

1) INSERT Trigger :- Insert trigger is used to inserted holds the new data of the affected rows when an insert statement has been executed.

Syntax :-

DELIMITER \$\$

create TRIGGER tri_candidate AFTER/BEFORE INSERT on candidate for EACH ROW

BEGIN

```
insert into test(id, name, action_performed)VALUES(new.id,new.cname, 'Record inserted');  
end
```

2) UPDATE Trigger :- update trigger is used to updated holds the modify data of the affected rows when an update statement has been executed.

Syntax :-

DELIMITER \$\$

```
create TRIGGER tri_candidate AFTER/BEFORE UPDATE on candidate FOR EACH  
ROW
```

BEGIN

```
insert into test(id, name, action_performed)VALUES(new.id,new.cname, 'Record inserted');  
end
```

3) DELETE Trigger :- DELETE trigger is used to holds the old data of the affected rows when an DELETE statement has been executed.

Syntax :-

DELIMITER \$\$

```
create TRIGGER tri_candidate AFTER/BEFORE UPDATE on candidate FOR EACH  
ROW
```

BEGIN

```
insert into test(id, name, action_performed)VALUES(old .id, old.cname, 'Record inserted');
```

end

16. Introduction to PL/SQL

Q1. What is PL/SQL, and how does it extend SQL's capabilities?

ANS :-

- PL/SQL stands for (procedural Language / Structured Query Language) is a block-structured language developed by oracle.
- PL/SQL is oracle's procedural language extension to SQL.
- PL/SQL is a combination of SQL along with the procedural features and programming languages.
- PL/SQL mainly used to create an application.

Q2. List and explain the benefits of using PL/SQL.

List the benefits of Using PL/SQL :-

- Object-oriented programming
 - Scalability
 - Productivity
 - Portability
 - High performance
 - Manageability
- **High performance:** PL/SQL can send large blocks of statements to a database at once, which reduces network traffic and improves performance.
 - **Portability:** PL/SQL applications can be used on multiple systems.
 - **Security:** PL/SQL has built-in security features.
 - **Object-oriented programming:** PL/SQL supports object-oriented programming.
 - **Scalability:** PL/SQL can scale to meet the needs of growing applications.
 - **Manageability:** PL/SQL offers features that make it easy to manage.

17. PL/SQL Control Structures

Q1. What are control structures in PL/SQL? Explain the IF-THEN and LOOP control structures.

ANS :-

- Control structures in programming are used to control the flow of execution in a program.
- They determine the order in which statements are executed based on specified conditions.
- Control structures allow us to make decisions, repeat blocks of code, and handle different scenarios.

1) IF – THEN Conditional statement :-

The sequence of statements is executed only if the condition is TRUE.

Syntax:-

```
If condition then  
-- do something  
End if;
```

Example :-

```
Declare  
Num1 number=10;  
Num2 number=20;  
BEGIN  
If num1<num2 then  
Dbms_output.put_line('num1 is small');  
End if;
```

2) LOOP statement in PL/SQL :-

- The loop statement is a feature of PL/SQL that allows you to repeatedly execute a block of code until a specified condition is satisfied.
- In SQL two types of LOOP :-
 - 1) For Loop
 - 2) While Loop

Syntax :-

```
LOOP
```



```
--code block  
IF condition THEN  
EXIT;  
END IF;  
END LOOP;
```

Q2. How do control structures in PL/SQL help in writing complex queries?

ANS :-In SQL , write Complex queries to Control Structure in PL/SQL use to three Statement .

- 1) Conditional Statements
- 2) Iteration Loop statements
- 3) Sequential statement

1) Conditional statements :-

It includes various conditional statements that allow developers to execute different blocks of code based on specific conditions.

1. **IF THEN**
2. **IF THEN ELSE**
3. **NESTED-IF-THEN**
4. **IF THEN ELSIF-THEN-ELSE Ladder**

2) Iteration Loop statements:-

The loop statement is a feature of PL/SQL that allows you to repeatedly execute a block of code until a specified condition is satisfied.

5. In SQL two types of LOOP :-

- 1) For Loop
- 2) While Loop

Syntax :-

LOOP

```
--code block  
IF condition THEN  
EXIT;  
END IF;
```

END LOOP;

3) Sequential statement :-

1) GO TO statement :- The GOTO statement performs unconditional branching to another executable statement in the same execution section of a PL/SQL block.

Syntax :- GOTO label_name;

2) NULL statement:- Usually when you write a statement in a program, you want it to do something.

Syntax :- NULL;

18. SQL Cursors

1. What is a cursor in PL/SQL? Explain the difference between implicit and explicit cursors.

ANS :-

- Cursor is a pointer to the query. (points to query)

There are two types of Cursors.

1) Implicit Cursor

2) Explicit Cursor

Implicit Cursor is created & used when it executes SELECT INTO, INSERT, UPDATE ..and all tasks on the cursor is performed transparently by Oracle. (Open, Close, Fetch etc.). It also throws NO_DATA_FOUND and TOO_MANY_ROWS as Oracle handles implicit cursor in the standard way.

Explicit cursor is the one which is declared by us in PL/SQL block's declaration section.

We need to control the cycle of the cursor Open, close, fetch from the cursor. Explicit Cursor need to be declared.

Q2. When would you use an explicit cursor over an implicit one?

ANS :-

Use an explicit cursor when you need more control over how you handle data. It is helpful for the complex tasks where you want to move through the data in a specific way or do the special operations on the each item.

Q1. Explain the concept of SAVEPOINT in transaction management. How do ROLLBACK and COMMIT interact with savepoints?

ANS :-

SAVEPOINT:- It is used to roll the transaction back to a certain point without rolling back the entire transaction.

Syntax:- SAVEPOINT SAVEPOINT_NAME;

- Savepoints name released when the transaction is committed or rolled back.
- The commit and rollback statement releases all savepoint name established within the transactions.

Syntax :- ROLLBACK to SAVEPOINT_name ;

Q2. When is it useful to use savepoints in a database transaction?

ANS :-

- savepoint is used to stored in large transaction to manages transactions in nesting processes.
- Savepoints are useful for complex transaction that require undoing only part of the transaction.