1. Relational Database Designs.

Difference: sql and plsql
SQL is a non-procedural language that executes a single query at a time
PL/SQL is a procedural language and executes blocks of code at once which helps to increases processing speed.

PL/pgSQL (Procedural Language/PostgreSQL) is a loadable procedural language for the PostgreSQL database system.

● **Features of PL/pgSQL**
1.PL/pgSQL is easy to use.
2. PL/pgSQL adds control structures to the SQL language.
3. PL/pgSQL can be used to create functions and trigger procedures.
4. It can perform complex computations.

● **Advantages of Using pl/pgSQL**

1. Portability
2. SQL Support
3. Better performance

● **Statements**
A statement performs an action within PL/pgSQL code, such as assignment of a value to a variable or the execution of a query.
Types: 1.Assignment statement
2.perform statement 3. Select into statement
4.execute statement

● **Expressions**
Expressions are calculations or operations that return their results as one of postgreSQL's base data types.

● **Declaration**
variables are used within plpgsql code to store data of specified type.

Syntax: Variable_name [CONSTANT] type [NOT NULL] [(DEFAULT):-) expression];

● **Attribute**
PL/pgSQL provides variable attributes to assist in working with database objects.
Syntax: variable_name table_name.column_name%TYPE

1. The %type attribute : The %TYPE is used to declare a variable with the type of a referenced database object
   Syntax: variable_name table_name.column_name%TYPE
2. The %ROWTYPE Attribute : The %ROWTYPE is used to declare a PL/pgSQL row with the same structure as the row specified during declaration.

● **Record**
Record variables are similar to row variables, but they have no predefined structure.
Variable_name   RECORD;

● **Rename**
variable identifiers can be renamed using rename keyword.
RENAME old identifier TO new _identifier;

● **Conditional Statements:**A conditional statement specifies an action or set of actions that should be executed based on the result of logical condition.
1.if then statement
in if then statement a statement or block of statements is executed if given condition evaluates to true.
IF condition THEN statement;
END IF;

2.If then else statement
The if then else statement colours used to execute a block of statements if a condition evaluates to true, otherwise a block of statements in else part is executed.
IF condition THEN statements ELSE statements
END IF;

● **Loops**
Loops conditions are used to perform some task repeatedly for fixed number of time or until some are valid. PL/pgSQL provides three iterative loops Le, the basic loc WHILE loop and the FOR loop.

1.The Basic Loop
This is the basic unconditional loop which starts with keyword LOOP and executes t statements within its body until terminated by an EXIT or RETURN statement.
Syntax:
[<<1abel>>] LOOP statements END LOOP.

2. While Loop
The WHILE statement repeats a sequence of statements till the condition evaluates to true.
[<<label>>]
WHILE condition LOOP Statements END LOOP [label];

3.FOR Loop
This form of FOR creates a loop that iterates over a range of integer values.

● **Views**
view is a subset of a real table, selecting certain columns or certain rows from an ordinary table.

1.Creating views
The PostgreSQL views are created using the create view statement.
Syntax for creating view:
CREATE view_name AS SELECT column1, column2….. FROM table_name WHERE [condition];

2.Dropping Views
To drop a view, simply use the DROP VIEW statement with the view_name.
DROP VIEW [IF EXISTS] view_name;

● **Functions**
Functions can accept and return values called function parameters or arguments.
1. Calling function : select function_name(arguments);
2. Dropping function : DROP FUNCTION deletes the definition of an existing function.
Syntax: DROP FUNCTION function_name (argtype[, ...]) [CASCADE | RESTRICT]

● **Cursors**
A PL/pgSQL cursor allows us to encapsulate a query and process each individual to at a time.
Types:
1.Implicit cursors are declared and managed by PL/pgSQL for all DML an PL/pgSQL SELECT statements.
2. Explicit cursors are declared and managed by the programmer.

Explicit cursor operations are as follows:
Step 1: Declare a cursor. (Declare my_cursor refcursor;)

Step 2: Open the cursor.(open cursor_variable)
Step 3: Fetch rows from the result set into a target.
Step 4: Check if there are more rows left to fetch
Step 5: Close the cursor. (Close cursor_variable;)

● **Triggers**

A trigger is a set of actions that are run automatically when a specified change operation (SQL INSERT, UPDATE, DELETE or TRUNCATE statement) is performed on a specified table or view.

Types:

1) A row level trigger: is fired for each affected row.
2) A statement level trigger: is fired only once for a statement.

To create a new trigger, we use the CREATE TRIGGER statement.

Syntax of Creating Trigger:

CREATETRIGGER trigger_name {BEFORE |AFTER} {event_ name}
ON table_name
FOR EACH {ROW | STATEMENT} EXECUTE PROCEDURE
function_name (arguments);

A trigger procedure is created with the CREATE FUNCTION command, declaring it as a function with no arguments and a return type of trigger.

Dropping trigger : To delete or destroy or drop a trigger, use a DROP TRIGGER statement.
DROP TRIGGER trigger_name ON table_name;

2. Transaction Concepts

● **Transaction**: collection of operations that forms a single logical unit of work is called transaction.

● **Transaction concepts:** a logical unit of database processing that includes one or more database access operations.

● **properties of transaction**
1. atomicity : no changes have occurred to the database or the database has been changed in a consistent manner.
2. consistency : data is in a consistent state in a transaction starts and When It Ends.
3.Isolation : action performed by a transaction will be hidden from outside the transaction until the transaction terminates.
4. Durability: Once a transaction completes successfully the changes it has made to the database persist even if there are system failures.

● **Transaction states**
Transaction goes through many different states throughout its life cycle. These states are called transaction States.
1.Active state : this is the initial state of transaction. Atransaction is active when it is executing.
2. Partially committed: transaction completes its last statement it enters in partially committed state.
3. Failed: If the system decides that the normal execution of the transaction can no longer proceed, then transaction is termed as failed.
4.Committed: When the transaction completes its execution successfully it enters committed state from partially committed state.

● **Schedule**: a sequence of operations by a set of concurrent transaction that preserves the order of the operations in each of the individual transactions.

●**Types of Schedules:**
Serial schedule
Non serial schedule : 1.serializable(conflict serializable & view serializable)
2.Non serializable schedule : Recoverable(Cascading sche.,Cascadless , strict schedule)
Non recoverable

■ **Serial Schedule :** transaction are executed one by one from start to finish the schedule is called as a serial schedule.
■**Concurrent schedule(Non serial) :** when several transactions are executed concurrently the corresponding schedule is called concurrent schedule.
■ **serializable schedule :** concurrent schedule results in a consistent state if its a result is equivalent to a serial schedule of that transaction.
Types of serializability
1 conflict serializable : Every conflict serializable schedule is view si realizable, but there are a few serializable schedule that are not conflict serializable.
2 **View :** a view serializable schedule in which blind right appear is not a conflict serializable.
■ **Non serializable schedule**
1.Recoverable : only reads are allowed before write operation on same data.
2. Non recoverable
■ **Strict Schedule:** If schedule contains no read or write before commit then it is known as a strict schedule.
■ **Cascadless schedule:** when No read or write occurs before execution of transaction.

- Cascading abord can also be rollback.

3. Concurrency Control

Concurrency Control : CC in transaction management preserves the consistency of database.
Concurrency control schemes:
1. Lock-based protocol : if 1 transaction is accessing a data item, no other transaction can modify that data item.
2. Time-stamp based protocol
3. Validation based protocol
4. Multiple granularity
5. Multiversion schemes

● **Two-Phase Locking (2PL) Protocol**
A locking protocol is a set of rules followed by all transactions while requesting and releasing locks.
1. **Growing phase**: transaction may obtain locks, but may not release any lock.
2. **Shrinking phase :** A transaction may release locks, but may not obtain any new locks.

●**Strict two phase locking protocol**
Cascading rollbacks can be avoided by a modification of two phase locking called strict 2pl protocol.

Thomas write rule : is the modification to the basic timestamp ordering

● **Modes of lock**
1. Shared mode lock (S).
2. Exclusive mode lock (X).
3. Intension-shared mode lock (IS).
4. Intension-exclusive mode lock (IX).
5. Shared and Intension-exclusive mode lock (SIX).

Deadlock detection : this is dealing with deadlock.It allows the system enter in a deadlock state and then try to recover using deadlock detection.
Deadlock Recovery : to recover from deadlock is to rollback 1 or more transactions to break the deadlock.

Deadlock prevention : DP means that if we design such a system where there is no chance of having deadlock.

■ **Timestamp :** any conflicting read and write operations are executed in timestap order.

● **W-Timestamp (Q):** It denotes the largest timestamp of any transaction that executed write (Q) successfully.
● **R-Timestamp (Q)**: It denotes the largest timestamp of any transaction that executed read (Q) successfully.

● **Wait for graph :** Deadlocks can be detected using a directed graph called Wait-for graph. The Wait for graph consists of a pair G (V. E) where V is a set of vertices and E is a set of edges.
● **wait die :** Scheme is based on non-preemptive technique. When a transaction T requests a lock on a data item currently held by T, T, is allowed to wait only if it has a timestamp smaller than that of T. Otherwise T, is rolled back (die).
● **Wound wait :** This schemes is based on premptive technique.

Difference between Timestamping and Locking
1.It is used to decide whether transaction should wait or rollback.
It is used for conference a control.
2.It is used for deadlock prevention.
It is used to improve performance.

● **Multiversion two-phase locking scheme** : combines the advantages of multiversion concurrency control with two phase locking. It allows the system to enter a deadlock state, and then try to recover using deadlock detection and dead-lock recovery scheme.

4. Crash Recovery

Crash Recovery : is the process by which the database is moved back to a consistent state.

Various types of transaction failures.
1.Transaction failure
2.System Crash
3.Disk failure

Types of errors:
system error : transaction cannot be continue with its normal execution.
Logical error

● **Recovery**: Recovery means restoring data into its previous Stote before the incident.

● **Database Recovery**: Database recovery is the process of restoring the database to a connect State in the event of a failure.

● **Crash Recovery**: Crosh recovery is the process by which the database is moved back to a consistent & unable state.

● **Commit**: Commit command is used to permanently save any transaction into the database.
Syntax: Commit;

●**Log:** Log is a structure used to store the database modification.

● **Log-based recovery**: It maintain transaction logs to keep track of all update Operations of the transaction.
1)Defered update
2) Immediate update.

Difference between Deferred update & Immediate update.
Undo is not needed / undo my be necessary
Redo may be necessary/Redo is not needed

●**Buffer Management**: The assignment and management of the operating system that performs this task is called BM.

● **ARIES (Algorithms for recovery and Isolation Exploiting Somantics)**
uses a Steal /no force approach for writing.
It is based on three concepts:
Write ahead logging
Repeating History during Redo
Logging changes during Undo.

Shadow paging: Is an alternative to log-based crash recovery technique.

5.Database Security

Database Security : Protecting a database from unauthorized access malicious destruction & even any accidental loss.

*● **Threats to Database Security**:
Loss of Confidentiality.

Loss of privacy.
Loss of tegrity
Loss of Availability suolla

- **Levels of security measures**

Physica
l Human
Operating System

- **Methods of for database Security.**

Authorization
Access Control
Statastical Database security

- **sql grant command :** is used to provide access on the database objects.
- **sql revoke command :** is included for the purpose of canceling privileges.

Queries

ALTER table :ALTER TABLE table_name ADD column_name datatype;   It is used to add columns to a table in a database.

CREATE table: CREATE TABLE table_name (column_1 datatype, column_2 datatype,   column_3 datatype);   It is used to create a new table in a database and specify the name of the table and columns inside

DELETE : Drop table table name;It is used to remove the rows from a table

INSERT : INSERT INTO table_name (column_1, column_2, column_3) VALUES (value_1, ‘value_2’, value_3);   It is used to add new rows to a table

SELECT : SELECT column_name FROM table_name; It is a statement that is used to fetch data from a database.

Error and Exception Handling
Any error occurring in a PL/pgSQL function aborts execution of the function. Errors can be trapped and recovered by using a Begin block with an Exception clause.

```
Syntax:Declare
Declarations
Begin
Statements
Exception
When condition then
Handler statements
End;
```