

## DBMS VIVA QUESTIONS

### 1. What is DBMS?

A)

A database management system (or DBMS) is essentially nothing more than a computerized data-keeping system. Users of the system are given facilities to perform several kinds of operations on such a system for either manipulation of the data in the database or the management of the database structure itself. Database Management Systems (DBMSs) are categorized according to their data structures or types.

There are several types of databases that can be used on a mainframe to exploit z/OS®: inverted list, hierarchic, network, or relational.

Hierarchical

Hierarchical and relational database systems have common benefits. RDBMS has the additional, significant advantage over the hierarchical DB of being non-navigational. By **navigational**, we mean that in a hierarchical database, the application programmer must know the structure of the database. The program must contain specific logic to navigate from the root segment to the desired child segments containing the desired attributes or elements. The program must still access the intervening segments, even though they are not needed.

A database organized according to ownership of records, allowing records to have multiple owners and thus providing multiple access paths to the data.

### Three schema Architecture

- The three schema architecture is also called ANSI/SPARC architecture or three-level architecture.
- This framework is used to describe the structure of a specific database system.
- The three schema architecture is also used to separate the user applications and physical database.
- The three schema architecture contains three-levels. It breaks the database down into three different categories.

The three-schema architecture is as follows:

The main objective of three level architecture is to enable multiple users to access the same data with a personalized view while storing the underlying data only once. Thus it separates the user's view from the physical structure of the database. This separation is desirable for the following reasons:

- Different users need different views of the same data.
- The approach in which a particular user needs to see the data may change over time.
- The users of the database should not worry about the physical implementation and internal workings of the database such as data compression and encryption techniques, hashing, optimization of the internal structures etc.
- All users should be able to access the same data according to their requirements.
- DBA should be able to change the conceptual structure of the database without affecting the user's
- Internal structure of the database should be unaffected by changes to physical aspects of the storage.

#### 1. Internal Level

- The internal level has an internal schema which describes the physical storage structure of the database.
- The internal schema is also known as a physical schema.
- It uses the physical data model. It is used to define that how the data will be stored in a block.
- The physical level is used to describe complex low-level data structures in detail.

#### 2. Conceptual Level

- The conceptual schema describes the design of a database at the conceptual level. Conceptual level is also known as logical level.
- The conceptual schema describes the structure of the whole database.
- The conceptual level describes what data are to be stored in the database and also describes what relationship exists among those data.
- In the conceptual level, internal details such as an implementation of the data structure are hidden.
- Programmers and database administrators work at this level.

### 3.3. External Level

- At the external level, a database contains several schemas that sometimes called as subschema. The subschema is used to describe the different view of the database.
- An external schema is also known as view schema.
- Each view schema describes the database part that a particular user group is interested and hides the remaining database from that user group.
- The view schema describes the end user interaction with database systems.

### 3.What are the advantages of using DBMS?

#### **Improved Data Integrity and Consistency**

One of the primary advantages of using a DBMS is its improved data integrity and consistency. A DBMS enforces rules and constraints on the data, ensuring that all data is accurate and consistent. For example, a DBMS can be programmed only to allow unique values in a specific field, preventing duplicate entries. Additionally, DBMSs provide various tools for data validation, such as check constraints and triggers, further enhancing the integrity of the data.

#### **Enhanced Data Security**

DBMSs provide various security mechanisms, such as authentication and access control, to protect the data from unauthorized access. Additionally, DBMSs offer encryption options to secure sensitive data, such as credit card numbers or personal information.

*Must read- [Data Collection Methods and Types](#)*

#### **Improved Data Accessibility and Control**

A DBMS also allows for improved data accessibility and control. With a DBMS, authorized users can easily retrieve and manipulate data while ensuring that the data remains consistent and accurate. Additionally, DBMSs provide tools for managing concurrent access to the data, ensuring that multiple users can access the data simultaneously without corrupting it. It uses techniques such as locking and transactions to ensure that the data in the database is accessed and modified in a consistent and controlled manner.

#### **Better Data Backup and Recovery**

DBMSs offer various backup and recovery options, such as incremental backups and point-in-time recovery, to ensure that the data can be quickly restored in the event of a failure. Additionally, it uses techniques such as backups, replication, and journaling to ensure that the data in the database can be recovered in case of failure.

#### **Data Abstraction**

## Enhance Decision Making

Users can have better-quality information and make better decisions due to DBMS's well-managed data and improved data access. It offers a framework to facilitate data quality initiatives. Better data management techniques produce higher-quality information, enabling better organization decision-making.

## Data Interoperability

A DBMS allows data to be shared and accessed by different applications and platforms, regardless of the underlying hardware and software. This allows the data in the database to be accessed and used by different applications and platforms without the need for complex data conversions or data duplication.

*Check out- [What is the Difference Between Hardware and Software?](#)*

## Scalability

A DBMS allows the data in the database to be scaled and expanded as the organization's needs grow. It allows the data in the database to be partitioned, replicated, and distributed to meet the organization's needs.

## Easy to Understand

In Database management systems (DBMS), we can present data straightforwardly and logically. It is simple to carry out many activities, such as adding, deleting, or creating files or data.

## Cost-Effective

Using a DBMS is cost-effective, reducing the need for manual data entry and management. It also reduces the need for redundant data storage and duplication, reducing the overall cost of data storage and management.

## 4. Which are the different data models?

- What are the Data Models in DBMS?
- Types of Data Models in DBMS
  - Hierarchical Model
  - Network Model
  - Entity-Relationship Model
  - Relational Model
  - Object-Oriented Data Model
  - Object-Relational Data Model
  - Flat Data Model
  - Semi-Structured Data Model
  - Associative Data Model
  - Context Data Model

ER model in DBMS is the high-level data model. It stands for the Entity-relationship model and is used to represent a logical view of the system from a data perspective. Creating an ER Model in DBMS is considered a best practice before implementing your database because it makes it easier for the developers to understand the database system just by looking at the ER model. It develops a conceptual design for a database that provides a very simple and straightforward view of the data. ER model makes use of ER diagrams, which are the diagrams sketched to design a database. ER diagrams are built on three basic concepts: entities, attributes, and relationships between them. An ER diagram defines entities, associated attributes, and relationships between entities. This helps visualize the logical structure of the database.

6. Which are the different types of attributes in ER model?

There are six such types of attributes: Simple, Composite, Single-valued, Multi-valued, and Derived attribute.

7. entity and entity set

1. **Entity** : An entity is a thing in a real-world with independent existence. An entity can exist independently and is distinguishable from other objects.
2. **Entity Set** : An entity set is a collection or set of all entities of a particular entity type at any point in time. The type of all the entities should be the same.

In a table each row is an entity and the complete dataset of all the entities is called a data set

8. What is a relationship and a relationship set

The relationship between various entities that belong to different entity types in a specific manner is known as a relationship type. **A collection of various relationships that belongs to the same relationship type is called a relationship set.**

9. What is a weak entity

An entity is like a unique object and a set of attributes. Talking about the types of entities, we have a strong entity and a weak entity. **A strong entity is independent, whereas a weak entity is reliant on another entity.**

10. What is partial key?

**The set of attributes that are used to uniquely identify a weak entity set** is called the Partial key. Only a bunch of the tuples can be identified using the partial keys.

11. What is RDBMS?

12. What are DDL commands

DDL is an abbreviation of **Data Definition Language**.

The DDL Commands in Structured Query Language are used to create and modify the schema of the database and its objects. The syntax of DDL commands is predefined for describing the data. The commands of Data Definition Language deal with how the data should exist in the database. Following are the five DDL commands in SQL:

1. CREATE Command
2. DROP Command
3. ALTER Command
4. TRUNCATE Command
5. RENAME Command

13. What are DML commands? Give examples.

The SQL commands that deal with the manipulation of data present in the database belong to DML or Data Manipulation Language and this includes most of the SQL statements. It is the component of the SQL statement that controls access to data and to the database. Basically, DCL statements are grouped with DML statements.

Example : delete, insert, update

14. What are DCL Commands? Give examples.

DCL includes commands such as GRANT and REVOKE which mainly deal with the rights, permissions, and other controls of the database system.

List of DCL commands:

**GRANT:** This command gives users access privileges to the database.

**REVOKE:** This command withdraws the user's access privileges given by using the GRANT command.

15. What are TCL Commands? Give examples.

Transactions group a set of tasks into a single execution unit. Each transaction begins with a specific task and ends when all the tasks in the group successfully complete. If any of the tasks fail, the transaction fails. Therefore, a transaction has only two results: success or failure. You can explore more about transactions [here](#). Hence, the following TCL commands are used to control the execution of a transaction:

**COMMIT:** Commits a Transaction.

**ROLLBACK:** Rollbacks a transaction in case of any error occurs.

**SAVEPOINT:** Sets a save point within a transaction.

**SET TRANSACTION:** Specifies characteristics for the transaction.

16. Differentiate between primary key, candidate key and super key.

- **Super Key** – A super key is a group of single or multiple keys which identifies rows in a table.
- **Primary Key** – is a column or group of columns in a table that uniquely identify every row in that table.
- **Candidate Key** – is a set of attributes that uniquely identify tuples in a table. Candidate Key is a super key with no repeated attributes.

17. What is referential integrity constraint?

A referential integrity constraint is also known as **foreign key constraint**. A foreign key is a key whose values are derived from the Primary key of another table.

18. Explain check constraint, unique constraint and not null constraint.

Check

The **CHECK** constraint is used to limit the value range that can be placed in a column.

If you define a **CHECK** constraint on a column it will allow only certain values for this column.

If you define a **CHECK** constraint on a table it can limit the values in certain columns based on values in other columns in the row.

- **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

SYNTAX : **CONSTRAINT** UC\_Person **UNIQUE** (ID,LastName)

19. What is a view?

A **VIEW** is a virtual table, through which a selective portion of the data from one or more tables can be seen. A view does not contain data of their own. They are used to restrict access to the database or to hide data complexity. A view is stored as a SELECT statement in the database.

DML operations on a view like INSERT, UPDATE, DELETE affects the data in the original table upon which the view is based.

The Syntax to create a sql view is:

CREATE VIEW view\_name

AS

SELECT column\_list

FROM table\_name [WHERE condition];

- view\_name is the name of the VIEW.
- The SELECT statement is used to define the columns and rows that you want to display in the view.
- 

20.What is PL/SQL?

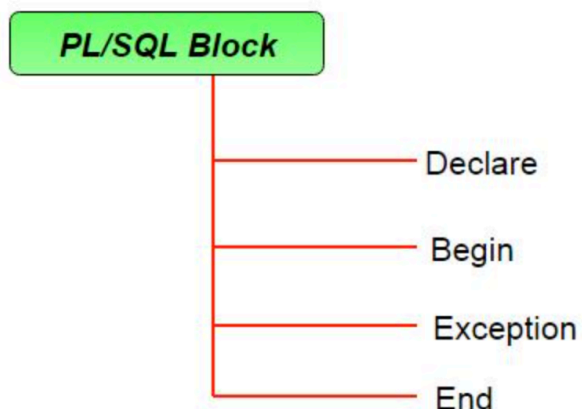
PL/SQL is a block structured language that enables developers to combine the power of SQL with procedural statements. All the statements of a block are passed to oracle engine all at once which increases processing speed and decreases the traffic.

21.

SQL	PL/SQL
SQL is a single query that is used to perform DML and DDL operations.	PL/SQL is a block of codes that used to write the entire program blocks/ procedure/ function, etc.
It is declarative, that defines what needs to be done, rather than how things need to be done.	PL/SQL is procedural that defines how the things needs to be done.
Execute as a single statement.	Execute as a whole block.
Mainly used to manipulate data.	Mainly used to create an application.
Cannot contain PL/SQL code in it.	It is an extension of SQL, so it can contain

#### Structure of PL/SQL Block:

PL/SQL extends SQL by adding constructs found in procedural languages, resulting in a structural language that is more powerful than SQL. The basic unit in PL/SQL is a block. All PL/SQL programs are made up of blocks, which can be nested within each other.





21. What is the difference between %type, %rowtype and %rowcount?

The %ROWTYPE attribute provides a record type that represents a row in a database table. The record can store an entire row of data selected from the table or fetched from a cursor or cursor variable. Fields in a record and corresponding columns in a row have the same names and datatypes.

The %TYPE attribute, used in PL/SQL variable and parameter declarations, is supported by the data server. Use of this attribute **ensures that type compatibility between table columns and PL/SQL variables is maintained.**

The SQL%ROWCOUNT cursor attribute returns the numbers of rows affected by the most recent INSERT, UPDATE, or DELETE statement executed in your session. Check this value to verify that the action completed properly.

22. What is raise\_application\_error?

The RAISE\_APPLICATION\_ERROR procedure **accepts user-defined error-number values from -20000 to -20999**. The SQLCODE that is returned in the error message is SQL0438N. The SQLSTATE contains class 'UD' plus three characters that correspond to the last three digits of the error-number value.

23. Differentiate between char, varchar and varchar2.

The CHAR datatype **stores fixed-length character strings**. When you create a table with a CHAR column, you must specify a column length (in bytes, not characters) between 1 and 255 for the CHAR column (default is 1).

**The VARCHAR datatype is currently synonymous with the VARCHAR2 datatype.** However, in a future version of Oracle, the VARCHAR datatype might store variable-length character strings compared with different comparison semantics. Therefore, use the VARCHAR2 datatype to store variable-length character strings.

24. What are aggregate functions?

In database management an aggregate function is a function where the values of multiple rows are grouped together as input on certain criteria to form a single value of more significant meaning.

Various Aggregate Functions

- 1) Count()
- 2) Sum()
- 3) Avg()
- 4) Min()
- 5) Max()

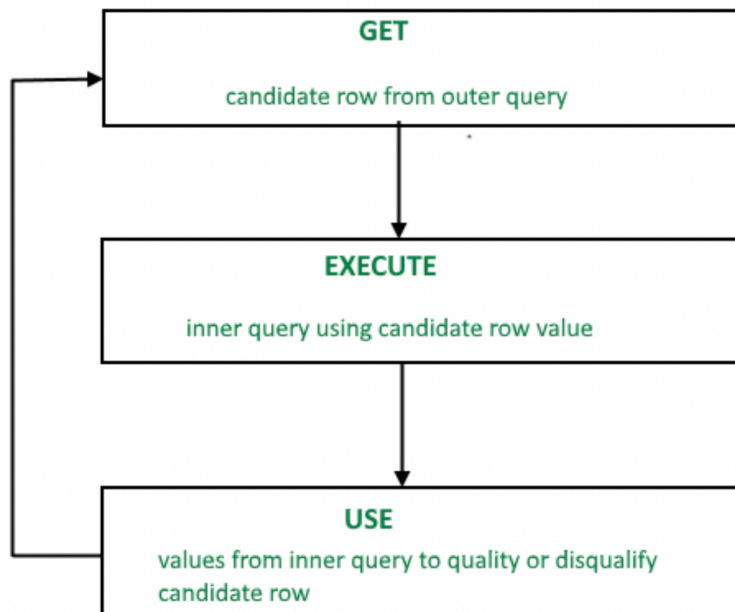
25. What is a subquery?

In SQL a Subquery can be simply defined as a query within another query. In other words we can say that a Subquery is a query that is embedded in WHERE clause of another SQL query.

26. What is a correlated subquery?

Correlated subqueries are used for row-by-row processing. Each subquery is executed once for every row of the outer query.

A correlated subquery is evaluated once for each row processed by the parent statement. The parent statement can be a **SELECT, UPDATE, or DELETE** statement.



27. What is a cursor?

**Cursor** is a Temporary Memory or Temporary Work Station. It is Allocated by Database Server at the Time of Performing DML(Data Manipulation Language) operations on Table by User. Cursors are used to store Database Tables. There are 2 types of Cursors: Implicit Cursors, and Explicit Cursors. These are explained as following below.

**Implicit Cursors:**

Implicit Cursors are also known as Default Cursors of SQL SERVER. These Cursors are allocated by SQL SERVER when the user performs DML operations.

**Explicit Cursors :**

Explicit Cursors are Created by Users whenever the user requires them. Explicit Cursors are used for Fetching data from Table in Row-By-Row Manner.

How to create Explicit Cursor:

**Declare Cursor Object.**

**Syntax :** DECLARE cursor\_name CURSOR FOR SELECT \* FROM table\_name  
**DECLARE**  
**s1 CURSOR FOR SELECT \* FROM studDetails**

Open Cursor Connection.

**Syntax :** OPEN cursor\_connection

OPEN s1

**Fetch Data from cursor.**

There are total 6 methods to access data from cursor. They are as follows :

**FIRST** is used to fetch only the first row from cursor table.

**LAST** is used to fetch only last row from cursor table.

**NEXT** is used to fetch data in forward direction from cursor table.

**PRIOR** is used to fetch data in backward direction from cursor table.

**ABSOLUTE n** is used to fetch the exact nth row from cursor table.

**RELATIVE n** is used to fetch the data in incremental way as well as decremental way.

**Syntax :** FETCH NEXT/FIRST/LAST/PRIOR/ABSOLUTE n/RELATIVE n FROM

cursor\_name **FETCH FIRST FROM s1**

**FETCH LAST FROM s1**

**FETCH NEXT FROM s1**



```
FETCH PRIOR FROM s1  
FETCH ABSOLUTE 7 FROM s1  
FETCH RELATIVE -2 FROM s1
```

Close cursor connection.

Syntax : CLOSE cursor\_name **CLOSE** s1

Deallocate cursor memory.

**Syntax** : DEALLOCATE cursor\_name

28.What are the cursor attributes ?

Each cursor has a set of attributes that enables an application program to test the state of the cursor. These attributes are **%ISOPEN**, **%FOUND**, **%NOTFOUND**, and **%ROWCOUNT**. This attribute is used to determine whether a cursor is in the open state.

30.Explain the steps in executing an explicit cursor.

There are four steps in using an Explicit Cursor.

- DECLARE the cursor in the declaration section.
- OPEN the cursor in the Execution Section.
- FETCH the data from cursor into PL/SQL variables or records in the Execution Section.
- CLOSE the cursor in the Execution Section before you end the PL/SQL Block.

31.What is the difference between procedure and function?

Parameters	Function	Procedure
Basics	Functions calculate the results of a program on the basis of the given input.	Procedures perform certain tasks in a particular order on the basis of the given inputs.
Try-Catch Blocks	Functions do not provide support for the try-catch Blocks.	Procedures provide support for the try-catch Blocks.
SQL Query	We can call a function in a SQL Query.	We cannot call a procedure in a SQL Query.
SELECT	The SELECT statements can have function calls.	The SELECT statements can never have procedure calls.
Return	A function would return the returning value/control to the code or calling function.	A procedure, on the other hand, would return the control, but would not return any value to the calling function or the code.
DML Statements	We cannot use the DML statements in a function, (functions such as Update, Delete, and Insert).	We can always use the DML statements in the case of a procedure.
Call	A function can be called using a procedure.	A procedure cannot be called using any function.
Compilation	The compilation of a function occurs when we call them in a program.	The compilation of the procedures needs to occur once, and in case it is necessary, these can be called repeatedly, and we don't have to compile them every single time.
Expression	A function must deal with expressions.	A procedure need not deal with expressions.
Explicit Transaction Handling	Functions cannot have explicit transaction handling.	Explicit transaction handling exists in the case of a procedure.

32.What is the difference between IN, OUT and IN OUT variable?

**IN :** A variable passed in this mode is of read only nature. This is to say, the value cannot be changed and its scope is restricted within the procedure. The procedure receives a value from this argument when the procedure is called.

**OUT :** In this mode, a variable is write only and can be passed back to

the calling program. It cannot be read inside the procedure and needs to be assigned a value.

**INOUT :** This procedure has features of both IN and OUT mode. The procedure can also read the variables value and can also change it to pass it to the calling function.

Explain IN, OUT and INOUT in procedures.

IN, OUT and INOUT are the arguments that are passed to the procedures.

- IN is a 'read only' argument and must be initialised.
- OUT is an uninitialised argument which must be initialized by a function.
- INOUT - A combination of the two above. That is, an initialised argument which can be written to

33.What is an exception?

An exception is **a runtime error or warning condition, which can be predefined or user-defined**. Predefined exceptions are raised implicitly (automatically) by the runtime system. User-defined exceptions must be raised explicitly by RAISE statements.

## Syntax for Exception Handling

The general syntax for exception handling is as follows. Here you can list down as many exceptions as you can handle. The default exception will be handled using **WHEN others THEN**

–

```
DECLARE
    <declarations section>
BEGIN
    <executable command(s)>
EXCEPTION
    <exception handling goes here >
    WHEN exception1 THEN
        exception1-handling-statements
    WHEN exception2 THEN
        exception2-handling-statements
    WHEN exception3 THEN
        exception3-handling-statements
    .....
    WHEN others THEN
        exception3-handling-statements
END;
```

34. Which are the different types of exceptions?

There are three types of exceptions:

- Predefined exceptions are error conditions that are defined by PL/SQL. example NO\_DATA\_FOUND, NOT\_LOGGED\_ON, PROGRAM\_ERROR, ROWTYPE\_MISMATCH, invalid\_cursor, invalid\_number
- 
- Non-predefined exceptions include any standard TimesTen errors.
- 
- User-defined exceptions are exceptions specific to your application.

36. What is a trigger

A trigger **defines a set of actions that are performed in response to an insert, update, or delete operation on a specified table**. When such an SQL operation is executed, the trigger is said to have been activated. Triggers are optional and are defined using the CREATE TRIGGER statement.

37. Which are the different types of trigger?

### Types of Triggers –

We can define 6 types of triggers for each table:

**AFTER INSERT** activated after data is inserted into the table.

**AFTER UPDATE:** activated after data in the table is modified.

**AFTER DELETE:** activated after data is deleted/removed from the table.

**BEFORE INSERT:** activated before data is inserted into the table.

**BEFORE UPDATE:** activated before data in the table is modified.

**BEFORE DELETE:** activated before data is deleted/removed from the table.

38. Differentiate between row level and statement level trigger.

Row Level Triggers	Statement Level Triggers
Row level triggers execute once for each and every row in the transaction.	Statement level triggers execute only once for each single transaction.
Specifically used for data auditing purpose.	Used for enforcing all additional security on the transactions performed on the table.
"FOR EACH ROW" clause is present in CREATE TRIGGER command.	"FOR EACH ROW" clause is omitted in CREATE TRIGGER command.
Example: If 1500 rows are to be inserted into a table, the row level trigger would execute 1500 times.	Example: If 1500 rows are to be inserted into a table, the statement level trigger would execute only once.

39. What is a mutating trigger?

A mutating trigger is **trigger that is currently being modified by DML operation**. For eg., You created a trigger trigger1 on table1, it should fire after update for each row. And you wrote some update statement on the same table (table1) inside the trigger.

The Oracle mutating trigger error **occurs when a trigger references the table that owns the trigger**, resulting in the "ORA-04091: table name is mutating, trigger/function may not see it." Message.

40. What is normalization?

**Normalization** is a database design technique that reduces data redundancy and eliminates undesirable characteristics like Insertion, Update and Deletion Anomalies.

42. What is a transaction?

A transaction is **the logical work unit that performs a single activity or multiple activities in a database**. Transactions may consist of a single read, write, delete, or update operations or a combination of these.

43. What are ACID properties of transaction?

Atomicity

All changes to data are performed as if they are a single operation. That is, all the changes are performed, or none of them are.

For example, in an application that transfers funds from one account to another, the atomicity property ensures that, if a debit is made successfully from one account, the corresponding credit is made to the other account.

Consistency

Data is in a consistent state when a transaction starts and when it ends.

For example, in an application that transfers funds from one account to another, the consistency property ensures that the total value of funds in both the accounts is the same at the start and end of each transaction.

Isolation

The intermediate state of a transaction is invisible to other transactions. As a result, transactions that run concurrently appear to be serialized.

For example, in an application that transfers funds from one account to another, the isolation property ensures that another transaction sees the transferred funds in one account or the other, but not in both, nor in neither.

Durability

After a transaction successfully completes, changes to data persist and are not undone, even in the event of a system failure.

For example, in an application that transfers funds from one account to another, the durability property ensures that the changes made to each account will not be reversed.

44.What are serial schedules?

Serial Schedule: The serial schedule is **a type of Schedule where one transaction is executed completely before starting another transaction**. Non-serial Schedule: In a Non-serial schedule, multiple transactions execute concurrently/simultaneously.

45.What are recoverable schedules?

A recoverable schedule is basically a schedule in which the commit operation of a particular transaction that performs read operation is delayed until the uncommitted transaction either commits or roll backs.

46.What are cascadeless schedules?

**Schedules in which transactions read values only after all transactions whose changes they are going to read commit** are called cascadeless schedules. Avoids that a single transaction abort leads to a series of transaction rollbacks.

47.What is deadlock?

In a database, a deadlock is **a situation in which two or more transactions are waiting for one another to give up locks**. For example, Transaction A might hold a lock on some rows in the Accounts table and needs to update some rows in the Orders table to finish.

48.What is timestamp ordering?

The Timestamp Ordering Protocol is **used to order the transactions based on their Timestamps**. The order of transaction is nothing but the ascending order of the transaction creation. The priority of the older transaction is higher that's why it executes first.

49.What is the use of hoststring in oracle database?

A host string is the name you assign for TNS connectivity in tnsnames. ora. These entries typically identify the machine and port number and SID ( or service name) of the database you are trying to connect to. If the client and database are on same machine, you don't need a host string.

50.What is two phase locking?

In databases and transaction processing, two-phase locking (2PL) is **a concurrency control method that guarantees serializability**. It is also the name of the resulting set of database transaction schedules (histories).

51.What is conflict serializability?

Conflict serializability **orders any conflicting operations in the same way as some serial execution**. A pair of operations is said to conflict if they operate on the same data item and one of them is a write operation.

52. Which are the different types of failure in DBMS?

A DBMS is vulnerable to several kinds of failures and each of these failures needs to be managed differently. There are many reasons that can cause database failures such as **network failure, system crash, natural disasters, carelessness, sabotage (corrupting the data intentionally), software errors**, etc.

53. Give 5 examples for built-in date functions.

Date functions in SQL:

- NOW()
- CURDATE()
- CURTIME()
- DATE()
- EXTRACT()
- DATE\_ADD()
- DATE\_SUB()
- DATEDIFF()

54. Give 5 examples for built-in string functions.

Examples of String Functions in C

- strlen() Syntax. size\_t strlen(const char \*str) ...
- strlen() Syntax. size\_t strlen(const char \*str, size\_t maxlen) ...
- strcmp() Syntax. int strcmp(const char \*str1, const char \*str2) ...
- strcat() Syntax. ...
- strncat() Syntax. ...
- strcpy() Syntax. ...
- strncpy() Syntax. ...

55. Give 5 examples for built-in numeric functions.

ABS(): It returns the absolute value of a number.

- Syntax: SELECT ABS(-243.5);
- Syntax: SELECT ACOS(0.25);
- Syntax: SELECT ASIN(0.25);
- Syntax: SELECT ATAN(2.5);
- Syntax: SELECT CEIL(25.75);
- Syntax: SELECT CEILING(25.75);
- Syntax: SELECT COS(30);
- Syntax: SELECT COT(6);