

SQL Questions

SQL, which stands for Structured Query Language, is a specialized programming language designed for managing and interacting with relational databases.

Relational databases organize data into structured tables with rows and columns, and SQL serves as the means to communicate with these databases.

Functions of SQL

1. SQL can execute queries against a database.
2. SQL can retrieve data from a database.
3. SQL can insert records in a database.
4. SQL can update records in a database.
5. SQL can delete records from a database.
6. SQL can create new databases.
7. SQL can create new tables in a database.
8. SQL can create stored procedures in a database
9. SQL can create views in a database.
10. SQL can set permissions on tables, procedures, and views.

SQL Interview Questions

Ques 1: Is SQL-supported programming?

Ans.

Basically, SQL is a command-based programming language but it doesn't contain any control flow statements.

Ques 2: State the difference between char and varchar.

Ans.

Both char and varchar are datatypes only, used for character strings. The only difference between them is the length. Varchar is used for a variable length of strings and char is used for a fixed length of strings. In char data type, if the length of the

string is less than the fixed length then it is padded with blank spaces to attain the fixed length. While varchar, padding won't be done.

Ques 3: Write the Syntaxes for joins.

Ans.

- **INNER JOIN:** select column_name From table1 INNER JOIN table2 on table1.column_name = table2.column_name;
- **OUTER JOIN:** select column_name From table1 OUTER JOIN table2 on table1.column_name = table2.column_name;
- **LEFT JOIN:** select column_name From table1 LEFT JOIN table2 on table1.column_name = table2.column_name;
- **RIGHT JOIN:** select column_name From table1 RIGHT JOIN table2 on table1.column_name = table2.column_name;
- **FULL JOIN:** select column_name From table1 INNER JOIN table2 on table1.column_name = table2.column_name WHERE condition;

Ques 4: Write the syntaxes for set operators.

Ans. Consider two tables T1 and T2 having attributes like id, name.

- **UNION:** select name from T1 UNION select name from T2;
- **UNION ALL:** select name from T1 UNION ALL select name from T2;
- **INTERSECT:** select name from T1 INTERSECT select name from T2;
- **MINUS:** select name from T1 MINUS select name from T2;

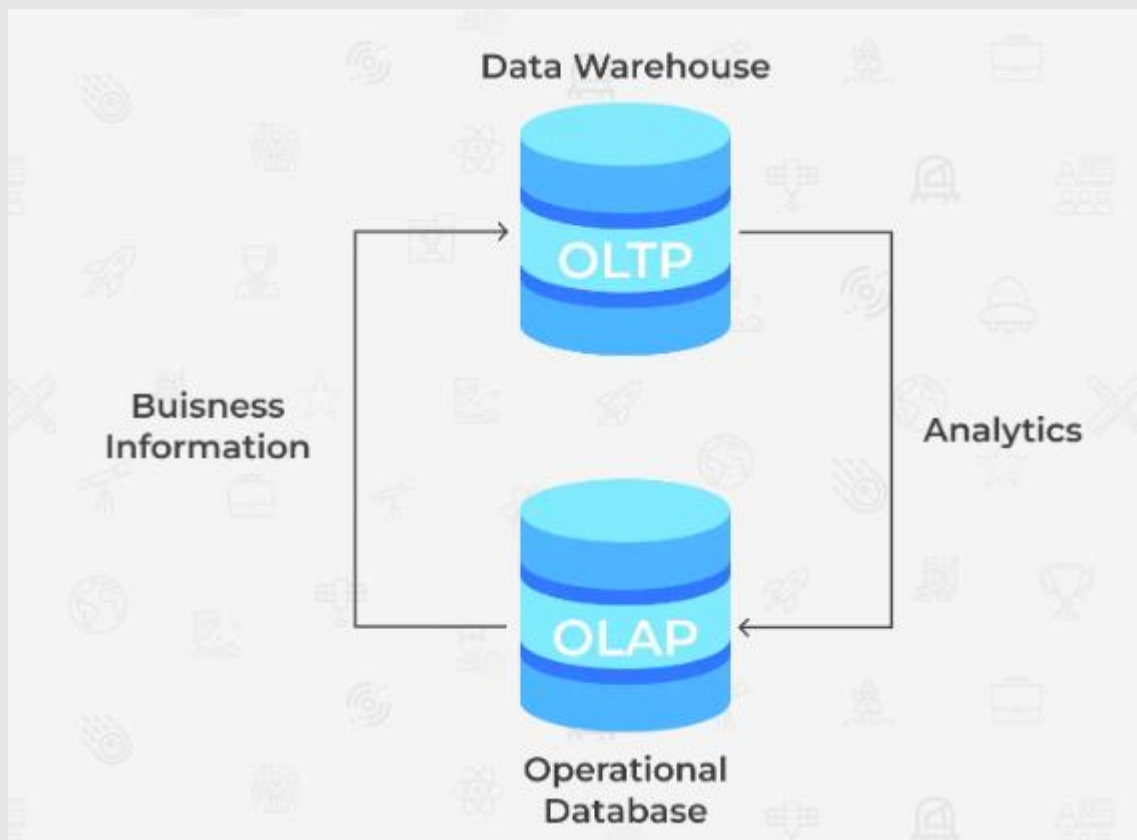
Ques 5: What is DBMS? Explain types of DBMS.

Ans.

DBMS is software that interacts with users, applications, and the database itself to capture and analyze the data. It is a structured collection of data. There are two types of DBMS. They are:

- **Relational DBMS:** Data is stored in tables(relations). This involves the concept of tuples, attributes.
- **Non – Relational DBMS:** In this, there won't be any concept of tables, tuples, attributes.

Ques 6: State the difference between OLTP and OLAP.



- **OLTP**: OLTP means Online Transaction Processing, it is an online database modification system. When data in the database is changed, this database transaction occurs.
- **OLAP**: OLAP means Online Analytical Processing, it is an online query response system.

Ques 7: What are joins in SQL?

Ans. As the name suggests a JOIN clause is used to combine rows from two or more tables, based on a related column between them. It is used to merge two tables or retrieve data from there. There are 4 joins in SQL namely:

- - Inner Join
 - Right Join

- Left Join
- Self Join

Ques 8: What are tables and Fields?

Ans. Data organized in a model with Rows and Columns is called a table. Rows are horizontal and Columns can be categorized as vertical. A table has a specified number of columns called fields but can have an infinite number of rows which is called a record.

Ques 9: What is virtual memory?

Ans.

Virtual memory is a memory management method that helps to execute the process using the primary and secondary memory. Though the program gets executed using the main memory, the resources and pages load from the secondary memory.

Ques 10: What is Foreign Key?

Ans. FOREIGN KEY: This is a key used for linking 2 tables.

1. A FOREIGN KEY is a field or maybe a collection of fields in a table that corresponds to the PRIMARY KEY of another table.
2. The table containing the FOREIGN KEY is called the CHILD TABLE and the table containing the CANDIDATE KEY is called the PARENT TABLE

Ques 11: What is the difference between DELETE and TRUNCATE statements?

Ans: DELETE:

- The DELETE command helps to delete a row of a table.
- The rollback of data is possible after using the delete statement.
- It is a DML command.
- It is slower than a TRUNCATE statement.

TRUNCATE:

- The TRUNCATE command helps to delete a row of a table.
- Rollback of data is not possible.
- It is a DDL command.
- As compared to DELETE the TRUNCATE command is faster.

Ques 12: What is Data Integrity?

Ans: The assurance of consistency and accuracy of the data over the entire life cycle is Data Integrity. It is a critical aspect of the implementation, design, and usage of any system that processes, stores or retrieves data. It also defines integrity constraints to enforce business rules on the data when it is entered into an application or a database.

Ques 13: What is the SELECT statement?

Ans: SELECT is a SQL command that is used for selecting data from a database. The data which is returned is saved in a result table, called the result-set. Example: select * from customers;

Ques 14: What is normalization?

Ans. Normalization is a way of organizing fields and tables of the database in a way that reduces redundancy and dependency. The main objective is to create a single table where operations like add, delete or modify can be performed on the field.

Ques 15: What is the ACID property in a database?

Ans.

To ensure that the data transactions are processed reliably in a database system we use the ACID property.

- Atomicity: It states that each transaction is all or nothing. It states that the entire transaction fails if one part of the transaction fails and the database state is left unchanged.
- Consistency: It ensures that the data must follow all validation rules. According to this a transaction never leaves your database without its state being completed.
- Isolation: The main goal of providing isolation is concurrency control. This property ensures that the concurrent property of execution should not be met.
- Durability: this property states that once a transaction has been committed, it remains committed, whatever the situation be, even power loss, crashes, or errors.

Questions for SQL Interview

Ques 16: Explain different types of indexes.

Ans.

There are three major types of index:

- **Unique Index**: It does not allow the field to possess redundant values if the column is indexed uniquely. A unique index can be applied automatically if a primary key is defined.
- **Clustered Index**: It rearranges the order of the table and searches based on key values. Every table has only one clustered index.
- **Non-Clustered Index**: It causes no change in the order of the table and keeps a logical order of the data intact. There can be many non-clustered indexes for a table.

Ques 17: What are the types of joins and explain each?

Ans. There are 4 types of Join:-

- **Inner Join**: Inner join returns rows when there is at least one match of rows between the tables.
- **Right Join**: Right join return rows which are common between the tables and all rows of the Right-hand side table. Simply, it returns all the rows from the right-hand side table even though there are no matches in the left-hand side table.
- **Left Join**: Left join return rows which are common between the tables and all rows of the Left-hand side table. Simply, it returns all the rows from the Left-hand side table even though there are no matches in the Right-hand side table.
- **Full Join**: Full join return rows when there are matching rows in any one of the tables. This means it returns all the rows from the left-hand side table and all the rows from the right-hand side table.

Ques 18: What do you mean by “Trigger” in SQL?

Ans. A trigger is a special type of stored procedure that executes automatically in place or after data changes. It allows executing a batch of code when an insert, update, or any other query is executed against a specific table.

Ques 19 :What is a constraint?

Ans.

Constraints are the limitations on the data type of a given table. A constraint can be specified while creating or altering the table statement.

- NOT NULL.
- CHECK.
- DEFAULT.
- UNIQUE.
- PRIMARY KEY.
- FOREIGN KEY.

Ques 20: What are the set operators in SQL?

Ans.

SQL supports a set of operations that can be performed on the table data. These are used to get meaningful results from data stored in the table, under different special conditions. The set operators in SQL are :

- UNION.
- UNION ALL
- INTERSECT
- MINUS

Ques 21: What is the usage of the DISTINCT keyword?

Ans. The SQL DISTINCT keyword is used in conjunction with the SELECT statement to eliminate all the duplicate records and fetch unique records.

Ques 22: What is a primary key?

Ans. A primary key also called a primary keyword, is a key in a relational database that is unique for each record. This is a special kind of unique key, and it has an implicit, NOT NULL constraint. It means Primary key values cannot be NULL. A relational database must always have one and only one primary key.

Ques 23: What is a unique key?

Ans. A Unique key constraint uniquely identified each record in the database. This provides uniqueness for the column or set of columns. A unique key is a set of one or more than one field/column of a table that uniquely identifies a record in a database table. You can say that it is a little like a primary key but it can accept only one null value and it cannot have duplicate values.

Ques 24:What are all the different normalizations?

Ans.

The normal forms can be divided into 4 forms, and they are explained below -.

- **First Normal Form (1NF)**: According to this, remove all the redundant columns from a table. Creation of tables for the related data and identification of unique columns.
- **Second Normal Form (2NF)**: Should follow all requirements of the first normal form. Arranging the data subsets in separate tables and Creation of relationships between the tables using a primary key.
- **Third Normal Form (3NF)**: Should follow all requirements of the 2NF. Removing the columns which are not dependent on primary key constraints.
- **BCNF (Boyce-Codd Normal Form)**: Meeting all the requirements of the third normal form and it should not have multi-valued dependencies.

Ques 25: What is a query?

Ans. A query is a request for data. A DB query is a code written to get the information back from the database. You ask the database for something and it answers in the best way it knows with data as a result of a query.

SQL Interview Ques

Ques 26: What is a stored procedure? State the Advantages and Disadvantages of Stored Procedure?

Ans. A stored procedure is a prepared SQL code that you can save, so the code can be reused over and over again. Stored Procedure is a function that consists of many SQL statements to access the database system. Stored procedure supports faster execution instead of executing multiple queries. This reduces network traffic and

provides better security to the data. The disadvantage is that it can be executed only in the Database and uses more storage memory.

Ques 27: Write a SQL query to find the names of employees that begin with 'A'?

Ans. To display the name of the employees that begin with 'A', type in the below command:

```
SELECT * FROM Table_name WHERE EmpName like 'A%'
```

Ques 28: What is the need for group functions in SQL?

Ans. An alias is a SQL function that most, if not all, RDBMSs support. It's a fictitious name given to a table or table column for the purposes of a SQL query. Furthermore, aliasing can be used as an obfuscation strategy to protect the true names of database fields. A correlation name is another name for a table alias. The "AS" keyword represents an alias.

Ques 29: What is a User-defined function? What are its various types?

Ans. The user-defined functions in SQL are like functions in any other programming language that accept parameters, perform complex calculations, and return a value. They are written to use the logic repetitively whenever required. There are two types of SQL user-defined functions:

- **Scalar Function**: As explained earlier, user-defined scalar functions return a single scalar value.
- **Table-Valued Functions**: User-defined table-valued functions return a table as output.
 - Inline: returns a table data type based on a single SELECT statement.
 - Multi-statement: returns a tabular result-set but, unlike inline, multiple SELECT statements can be used inside the function body.

Ques 30: What is CLAUSE in SQL?

Ans. SQL clause helps to limit the result set by providing a condition to the query. A clause helps to filter the rows from the entire set of records.

For example – WHERE, HAVING clause.

SQL Interview Questions and Answers

Ques 31: What is the difference between 'HAVING' CLAUSE and a 'WHERE' CLAUSE?

Ans. HAVING clause can be used only with SELECT statement. It is usually used in a GROUP BY clause and whenever GROUP BY is not used, HAVING behaves like a WHERE clause. Having Clause is only used with the GROUP BY function in a query whereas WHERE Clause is applied to each row before they are a part of the GROUP BY function in a query.

Ques 32: What is AUTO INCREMENT?

Ans. AUTO_INCREMENT is used in SQL to automatically generate a unique number whenever a new record is inserted into a table. Since the primary key is unique for each record, we add this primary field as the AUTO_INCREMENT field so that it is incremented when a new record is inserted. The AUTO-INCREMENT value is by default starts from 1 and is incremented by 1 whenever a new record is inserted.

Ques 33: How to create empty tables with the same structure as another table?

Ans. Creating empty tables with the same structure can be done smartly by fetching the records of one table into a new table using the INTO operator while fixing a WHERE clause to be false for all records. Hence, SQL prepares the new table with a duplicate structure to accept the fetched records but since no records get fetched due to the WHERE clause in action, nothing is inserted into the new table.

```
SELECT * INTO Students_copy FROM Students WHERE 1 = 2;
```

Ques 34: How to create empty tables with the same structure as another table?

Ans. Creating empty tables with the same structure can be done smartly by fetching the records of one table into a new table using the INTO operator while fixing a WHERE clause to be false for all records. Hence, SQL prepares the new table with a

duplicate structure to accept the fetched records but since no records get fetched due to the WHERE clause in action, nothing is inserted into the new table.

Ques 35: What is the main difference between SQL and PL/SQL?

Ans. SQL is a query language that allows you to issue a single query or execute a single insert/update/delete whereas PL/SQL is Oracle's "Procedural Language" SQL, which allows you to write a full program (loops, variables, etc.) to accomplish multiple operations such as selects/inserts/updates/deletes.

Ques 36: What are Local and Global variables?

Ans.

- **Local variables**: These variables can be used or exist only inside the function. These variables are not used or referred to by any other function.
- **Global variables**: These variables are the variables that can be accessed throughout the program. Global variables cannot be created whenever that function is called.

Ques 37: How can you fetch the first 5 characters of the string?

Ans. There are a lot of ways to fetch characters from a string.

For example: `Select SUBSTRING(StudentName,1,5) as studentname from student;`

Ques 38: What is a View?

Ans.

A view is a virtual table that consists of a subset of data contained in a table. Since views are not present, it takes less space to store. A view can have data of one or more tables combined and it depends on the relationship.

Ques 39: How to change the column data type in SQL?

Ans. We can change the data type of the column using the alter table. This will be the command: `ALTER TABLE table_name MODIFY COLUMN column_name datatype;` We start off by giving the keywords ALTER TABLE, then we will give in the name of the table. After that, we will give in the keywords MODIFY COLUMN. Going

ahead, we will give in the name of the column for which we would want to change the datatype and finally we will give in the data type to which we would want to change.

Ques 40: What is the difference between SQL and NoSQL databases?

Ans. SQL stands for structured query language and is majorly used to query data from relational databases. When we talk about a SQL database, it will be a relational database. But when it comes to the NoSQL database, we will be working with non-relational databases.

SQL Interview Questions with Answers

Ques 41: What is RDBMS? How is it different from DBMS?

Ans.

RDBMS (Relational Database Management System) is a type of database system that organizes and manages data in a tabular format with rows and columns. It differs from a DBMS (Database Management System) by its ability to establish relationships between tables and enforce data integrity through features like foreign keys and normalization. RDBMS systems like MySQL, Oracle, and PostgreSQL are widely used in applications where structured data and relational associations are essential.

Ques 42: What is the difference between SQL and MySQL?

Definition	SQL is a standardized query language used to manage relational databases.	MySQL is a specific relational database management system (RDBMS) that uses SQL as its query language.
Type	Language	Database Management System (DBMS)
Standardization	SQL is a standardized language defined by ANSI and ISO.	MySQL follows the SQL standard and adds its own extensions.
Portability	SQL can be used with various RDBMS, making it transferable between systems.	MySQL is a specific RDBMS, not portable to other DBMS.
Purpose	Used to interact with relational databases regardless of the DBMS.	A specific RDBMS used for data storage and retrieval.
Features	SQL provides a standard set of commands for tasks like querying, modifying data, and managing database structures.	MySQL implements SQL commands and offers additional features specific to its database engine.
Examples of Usage	SQL queries are used with MySQL, PostgreSQL, Oracle, SQL Server, etc.	MySQL is used for applications that require a dedicated database system.

Ques 43: What are Aggregate and Scalar functions?

Ans.

- Aggregate functions perform calculations on sets of data and return a single result, often used with the **GROUP BY** clause.
- Scalar functions operate on individual data points within a row and return a result for each row, allowing data transformation and manipulation.

Ques 44: How can you optimize subqueries?

Ans.

To optimize subqueries:

1. Use EXISTS or IN.
2. Avoid correlated subqueries.
3. Limit results with LIMIT.
4. Ensure proper indexing.
5. Simplify subquery logic.
6. Consider using CTEs.
7. Evaluate execution plans.
8. Test and benchmark for performance.

Ques 45: What is OLTP?

Ans. OLTP, or Online Transaction Processing, is a type of database system designed for handling a large volume of real-time, short, and frequent transactions. It's commonly used in applications like e-commerce, banking, and inventory management to process day-to-day operations efficiently. OLTP systems focus on maintaining data integrity and fast transaction processing.

Ques 46: What is composite key?

Ans. A composite key in a relational database consists of multiple columns used together to uniquely identify rows in a table. It's used when a single column cannot provide unique identification on its own.

```
CREATE TABLE SalesOrderItems (  
    OrderNumber INT,  
    LineItemNumber INT,  
    ProductID INT,  
    Quantity INT,  
    PRIMARY KEY (OrderNumber, LineItemNumber)  
);
```

Ques 47: Discuss the optimization techniques for OLAP system?

Ans.

Optimization techniques for OLAP (Online Analytical Processing) systems include aggregating data, partitioning, indexing, using materialized views, caching, parallel processing, compression, and hardware acceleration. These methods aim to speed up query execution and enhance data analysis performance in OLAP environments.

Ques48: How does data indexing work?

Ans.

Indexing in a database creates a separate structure (like a B-tree) that stores a sorted version of certain columns, allowing for faster data retrieval. It speeds up queries but requires maintenance and careful selection of indexed columns for optimal performance.

Ques 49: What is an Alias in SQL?

In SQL, an alias is a user-defined alternative name assigned to a table or a column in a query. Aliases serve several purposes:

1. **Readability:** Aliases make SQL queries more readable by providing shorter and more descriptive names for tables and columns. This is especially useful in complex queries involving multiple tables or when column names are lengthy or cryptic.

- Example (Table Alias):
- `SELECT e.employee_id, e.first_name, d.department_name`
- `FROM employees AS e`
- `JOIN departments AS d ON e.department_id = d.department_id;`
- Example (Column Alias):
- `SELECT first_name AS "First Name", last_name AS "Last Name"`
- `FROM employees;`

2. **Clarity in Results:** Column aliases help clarify the meaning of columns in the result set, providing more context to users or applications consuming the data.
3. **Handling Duplicate Column Names:** In cases where multiple tables in a query have columns with the same name, aliases are essential to distinguish between them.
4. **Simplifying Expressions:** Aliases can simplify the use of complex expressions or calculations by assigning a name to the result, making it easier to reference.

Ques 50: What is Cursor? How to use a Cursor?

Ans.

A cursor in databases is a pointer used to iterate through the rows of a result set. To use a cursor:

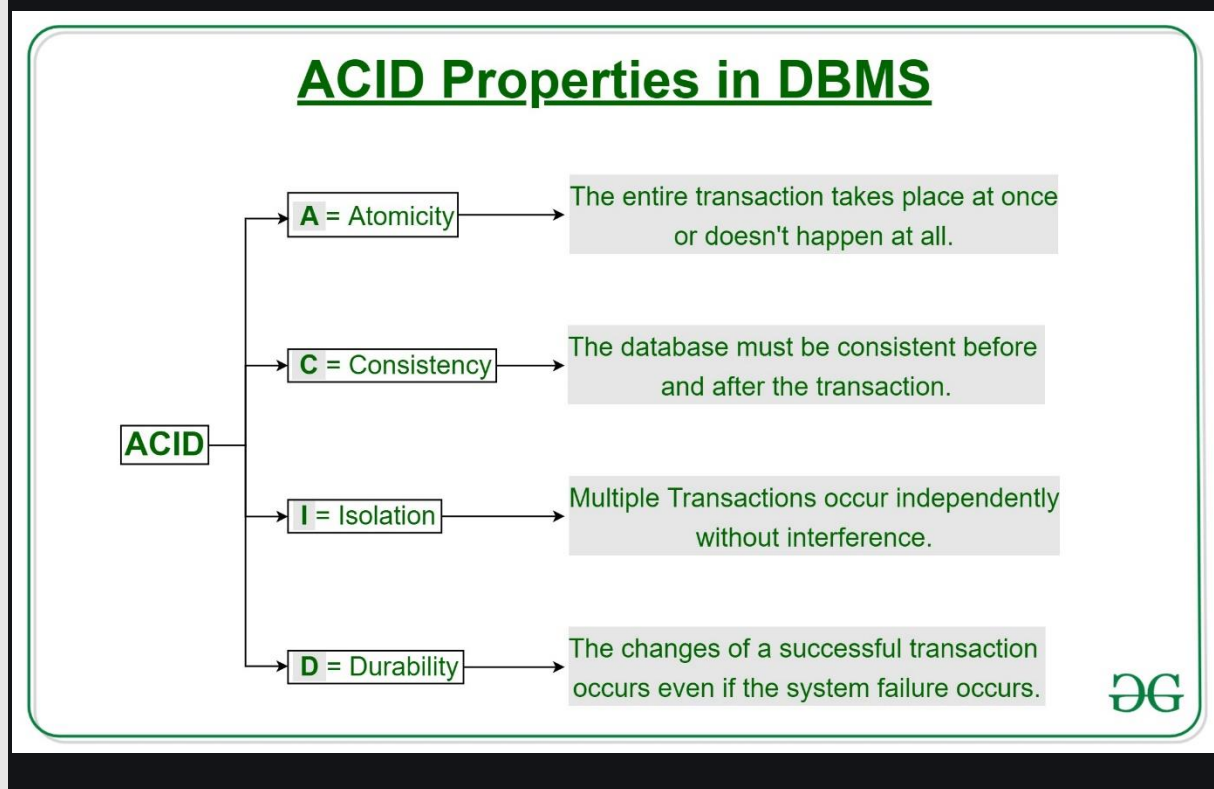
1. **Declare:** Declare a cursor, specifying which result set to iterate through.
2. **Open:** Open the cursor to establish the result set.
3. **Fetch:** Retrieve rows one by one from the result set using FETCH.
4. **Process:** Process the fetched data or perform actions.
5. **Close:** Close the cursor when done.
6. **Deallocate:** Optionally, deallocate the cursor to free resources.

Cursors are typically used for row-by-row data manipulation and are supported in SQL-based database systems like SQL Server, Oracle, and PostgreSQL.

ACID Properties in DBMS

A [transaction](#) is a single logical unit of work that accesses and possibly modifies the contents of a database. Transactions access data using read and write operations.

In order to maintain consistency in a database, before and after the transaction, certain properties are followed. These are called **ACID** properties.



Atomicity:

By this, we mean that either the entire transaction takes place at once or doesn't happen at all. There is no midway i.e. transactions do not occur partially. Each transaction is considered as one unit and either runs to completion or is not executed at all. It involves the following two operations.

—**Abort**: If a transaction aborts, changes made to the database are not visible.

—**Commit**: If a transaction commits, changes made are visible.

Atomicity is also known as the 'All or nothing rule'.

Consider the following transaction **T** consisting of **T1** and **T2**: Transfer of 100 from account **X** to account **Y**.

Before: X : 500	Y: 200
Transaction T	
T1	T2
Read (X) X: = X - 100 Write (X)	Read (Y) Y: = Y + 100 Write (Y)
After: X : 400	Y : 300

If the transaction fails after completion of **T1** but before completion of **T2**.(say, after **write(X)** but before **write(Y)**), then the amount has been deducted from **X** but not added to **Y**. This results in an inconsistent database state.

Therefore, the transaction must be executed in its entirety in order to ensure the correctness of the database state.

Consistency:

This means that integrity constraints must be maintained so that the database is consistent before and after the transaction. It refers to the correctness of a database. Referring to the example above,

The total amount before and after the transaction must be maintained.

Total **before T** occurs = **500 + 200 = 700**.

Total **after T** occurs = **400 + 300 = 700**.

Therefore, the database is **consistent**. Inconsistency occurs in case **T1** completes but **T2** fails. As a result, T is incomplete.

Isolation:

This property ensures that multiple transactions can occur concurrently without leading to the inconsistency of the database state. Transactions occur independently without interference. Changes occurring in a particular transaction will not be visible to any other transaction until that particular change in that transaction is written to memory or has been committed. This property ensures that the execution of transactions concurrently will result in a state that is equivalent to a state achieved these were executed serially in some order.

Let $X = 500$, $Y = 500$.

Consider two transactions **T** and **T''**.

T	T''
Read (X)	Read (X)
$X := X * 100$	Read (Y)
Write (X)	$Z := X + Y$
Read (Y)	Write (Z)
$Y := Y - 50$	
Write (Y)	

Suppose **T** has been executed till **Read (Y)** and then **T''** starts. As a result, interleaving of operations takes place due to which **T''** reads the correct value of **X** but the incorrect value of **Y** and sum computed by

T'': $(X + Y = 50,000 + 500 = 50,500)$

is thus not consistent with the sum at end of the transaction:

T: $(X + Y = 50,000 + 450 = 50,450)$.

This results in database inconsistency, due to a loss of 50 units. Hence, transactions must take place in isolation and changes should be visible only after they have been made to the main memory.

Durability:

This property ensures that once the transaction has completed execution, the updates and modifications to the database are stored in and written to disk and they persist even if a system failure occurs. These updates now become permanent and are stored in non-volatile memory. The effects of the transaction, thus, are never lost.

Some important points:

Property	Responsibility for maintaining properties
Atomicity	Transaction Manager
Consistency	Application programmer
Isolation	Concurrency Control Manager
Durability	Recovery Manager

The **ACID** properties, in totality, provide a mechanism to ensure the correctness and consistency of a database in a way such that each transaction is a group of

operations that acts as a single unit, produces consistent results, acts in isolation from other operations, and updates that it makes are durably stored.

ACID properties are the four key characteristics that define the reliability and consistency of a transaction in a Database Management System (DBMS). The acronym ACID stands for Atomicity, Consistency, Isolation, and Durability. Here is a brief description of each of these properties:

1. **Atomicity:** Atomicity ensures that a transaction is treated as a single, indivisible unit of work. Either all the operations within the transaction are completed successfully, or none of them are. If any part of the transaction fails, the entire transaction is rolled back to its original state, ensuring data consistency and integrity.
2. **Consistency:** Consistency ensures that a transaction takes the database from one consistent state to another consistent state. The database is in a consistent state both before and after the transaction is executed. Constraints, such as unique keys and foreign keys, must be maintained to ensure data consistency.
3. **Isolation:** Isolation ensures that multiple transactions can execute concurrently without interfering with each other. Each transaction must be isolated from other transactions until it is completed. This isolation prevents dirty reads, non-repeatable reads, and phantom reads.
4. **Durability:** Durability ensures that once a transaction is committed, its changes are permanent and will survive any subsequent system failures. The transaction's changes are saved to the database permanently, and even if the system crashes, the changes remain intact and can be recovered.

Overall, ACID properties provide a framework for ensuring data consistency, integrity, and reliability in DBMS. They ensure that transactions are executed in a reliable and consistent manner, even in the presence of system failures, network issues, or other problems. These properties make DBMS a reliable and efficient tool for managing data in modern organizations.

Advantages of ACID Properties in DBMS:

1. **Data Consistency:** ACID properties ensure that the data remains consistent and accurate after any transaction execution.
2. **Data Integrity:** ACID properties maintain the integrity of the data by ensuring that any changes to the database are permanent and cannot be lost.
3. **Concurrency Control:** ACID properties help to manage multiple transactions occurring concurrently by preventing interference between them.

4. Recovery: ACID properties ensure that in case of any failure or crash, the system can recover the data up to the point of failure or crash.

Disadvantages of ACID Properties in DBMS:

1. Performance: The ACID properties can cause a performance overhead in the system, as they require additional processing to ensure data consistency and integrity.
2. Scalability: The ACID properties may cause scalability issues in large distributed systems where multiple transactions occur concurrently.
3. Complexity: Implementing the ACID properties can increase the complexity of the system and require significant expertise and resources.

Overall, the advantages of ACID properties in DBMS outweigh the disadvantages. They provide a reliable and consistent approach to data

4. management, ensuring data integrity, accuracy, and reliability. However, in some cases, the overhead of implementing ACID properties can cause performance and scalability issues. Therefore, it's important to balance the benefits of ACID properties against the specific needs and requirements of the system.