Data Modeling Questions

1. What is cardinality?

Cardinality refers to one entity type that could have relationship or how they interact with other entities, one to one, one to many, many to many.

1. Describe the differences in the first through fifth normalization forms

First: The domain of each attribute contains only atomic values, and the value of each attribute contains only a single value from that domain.

Second: No non-prime attribute in the table is functionally dependent on a proper subset of any candidate key.

Third: Every non-prime attribute is non-transitively dependent on every candidate key in the table. The attributes that do not contribute to the description of the primary key are removed from the table. In other words, no transitive dependency is allowed.

BCNF: Any table is said to be in BCNF, if its candidate keys do not have any partial dependency on the other attributes. i.e.; in any table with (x, y, z) columns, if (x, y)->z and z->x then it's a violation of BCNF. If (x, y) are composite keys and (x, y)->z, then there should not be any reverse dependency, directly or partially.

Fourth: Every non-trivial multivalued dependency in the table is a dependency on a superkey.

Fifth: Every non-trivial join dependency in the table is implied by the superkeys of the table.

1. When might someone denormalize their data?

to reduce the number of table joins.

1. What are the elements of an ERD?

Entities for which someone is seeking information, attributes of these attributes and relationships between the entities.

1. What is DBMS ?

The database management system is a collection of programs that enables user to store, retrieve, update and delete information from a database.

1. What is RDBMS ?

Relational Database Management system (RDBMS) is a database management system (DBMS) that is based on the relational model.

1. What is a database transaction?

Database transaction takes database from one consistent state to another. At the end of the transaction the system must be in the prior state if the transaction fails or the status of the system should reflect the successful completion if the transaction goes through.

1. What are properties of a transaction?

Atomicity

A transaction consists of many steps. When all the steps in a transaction get completed, it will get reflected in DB or if any step fails, all the transactions are rolled back.

Consistency

The database will move from one consistent state to another, if the transaction succeeds and remain in the original state, if the transaction fails.

Isolation

Every transaction should operate as if it is the only transaction in the system.

Durability

Once a transaction has completed successfully, the updated rows/records must be available for all other transactions on a permanent basis.

1. What is a database lock?

Database lock tells a transaction, if the data item in questions is currently being used by other transactions.

1. What are the types of locks?

1. Shared Lock

When a shared lock is applied on data item, other transactions can only read the item, but can't write into it.

2. Exclusive Lock

When an exclusive lock is applied on data item, other transactions can't read or write into the data item.

1. What is a unique key?

Unique key is same as primary with the difference being the existence of null. Unique key field allows one value as NULL value.

Advanced SQL Questions

1. What is the difference between an inner and outer join?

An inner join involving two tables where a common key in both, but an outer join involving two tables that may not have a match in the first or the second table.

1. How do you maintain database integrity where deletions from one table will automatically cause deletions in another table?

1.Create a trigger 2. Use foreign key with delete cascade.

1. What is the difference between join and union?

Join allow us to look for more than one table to find records that we need, union allow us to add two or more similar datasets.

1. What is the difference between where and having?

They both filter out records based on one or more conditions. The difference is that where can only be used in the non-aggregated column but having is used for aggregated columns

1. What is the difference among union, except and intersect?

Union combine two or more datasets and eliminates duplicate records from the result set. Except give us the dataset that is present in the first table but not present in the second table. Intersect returns us only the matching or common records in all involved datasets.

1. What is self join and why is it required?
2. How to go about picking a good PARTITION key for a big table, and what are some of the "gotchas" of a badly chosen PARTITION key.
3. Someone complains that their app is running slow because of what they think is a "database problem".  How do you go about investigating the slowness and helping them to improve their queries?
4. What are some differences between MyISAM and InnoDB storage engines?
5. Specific to InnoDB, what are some differences between the storage of a primary key and a secondary index?
6. When would you use one versus the other?
7. What types of tables would you advise using the MEMORY storage engine?
8. WHAT IS THE MAIN DIFFERENCE BETWEEN PRIMARY KEY, UNIQUE KEY, AND FOREIGN KEY?

#### PRIMARY KEY:

1. The primary key cannot have a NULL value.  
2. Every table can have only one primary key.  
3. By default, Primary key supports clustered index. Thus data in the database table are physically organized in the sequence of clustered index.  
4. It can be related to another table as a Foreign Key.  
5. It supports the generation of ID automatically with the help of Auto Increment field.

#### UNIQUE KEY:

1. Unique Constraint may have a NULL value.  
2. Each table can have more than one Unique Constraint.  
3. By default, Unique key is a unique non-clustered index.  
4. It is not related to another table as a Foreign Key.  
5. Unique Constraint doesn’t support Auto Increment value.

#### FOREIGN KEY:

1. A Foreign key is a field in a table whereas, it is the primary key in another table.  
2. It can accept multiple null values.  
3. A Foreign key does not automatically create an index, clustered or non-clustered. You must manually create an index on the foreign key.  
4. We can have more than one foreign key in a table.  
5. There are advantages of having a foreign key supported with a clustered index, but you get only one per table. The advantage using a clustered index is that, on selecting the parent plus all child records, it can bring all child records next to each other.   
6. The Foreign key shouldn’t have a null value. Else, the system will consider it as an orphan record.

14. WHAT IS NORMALIZATION AND HOW DOES IT WORK?

It is the process of designing database tables to minimize the data redundancy is called normalization.

We need to divide a database into two or more tables and define relationships between them.

1. WHAT IS THE DIFFERENCE BETWEEN A SUPERKEY AND THE CANDIDATE KEY?

A superkey is a combination of columns that uniquely identifies any row within a relational database management system (RDBMS) table.

Whereas, a candidate key is a superkey containing a minimum number of columns that can uniquely identify each row.

#### TCL (TRANSACTION CONTROL LANGUAGE):

It allows you to control and manage transactions to maintain the integrity of data within SQL statements. Following are different TCL statements:

1. <COMMIT> – to save the work  
   ii. <SAVEPOINT> – identify a point in a transaction to which you can rollback at a later point in time when required  
   iii. <ROLLBACK> – restore the database to original since the last COMMIT   
   iv. <SET TRANSACTION> – Change transaction options like isolation level and what rollback segment to use.
2. 1. Atomicity: A transaction may contain two or more discrete pieces of information. Atomicity means either commit all the data or nothing.  
   2. Consistency: A transaction creates a new and valid state of data. However, if any failure occurs, it reverts the data to its original state before the start of the transaction.  
   3. Isolation: A transaction under execution and not yet committed must remain isolated from any other transaction.  
   4. Durability: System stores the committed data so that the data is available in its correct state, in case a failure or system restart happens.

#### DELETE:

1. It is a DML statement.  
2. It applies a filter based on an optional WHERE clause to identify the rows that will get deleted.  
3. It is possible to roll back a transaction that got deleted.  
4. It does not reset the identity of the table.  
6. Triggers will get fired.  
7. On initiating a DELETE operation, all the data first gets copied into Rollback Tablespace and then delete operation gets performed. Thus we can get back the data by ROLLBACK command.  
8. Use this, only when you want to delete specific records. For example, <DELETE FROM table\_name WHERE username = ‘Aditya’;>

#### TRUNCATE:

1. It is a DDL Statement.  
2. Removes all rows from a table and it becomes empty. But, the table structures, its columns, constraints, and indexes remain intact.  
3. It is not possible to roll back the TRUNCATE transaction.  
4. It resets the identity of table i.e. the auto-incrementing keys are reset to 1. It’s just like having a brand new table.  
5. It is faster than DELETE and uses a lesser amount of system and transaction logs.  
6. TRUNCATE cannot be used on a table referenced by a FOREIGN KEY constraint.  
7. No Triggers will get fired.  
8. Cannot use WHERE conditions.  
9. Use this when you just want an empty table.

1. WHAT IS AN INDEX? EXPLAIN THE DIFFERENT TYPES OF INDEX.

#### 1. CLUSTERED INDEX:

It sorts and stores the rows of data in the table or view, based on its keys. These are the columns included in the index definition. There can be only one clustered index per table because sorting of data rows can be done only in one order.

#### 2. NONCLUSTERED INDEX:

It contains the nonclustered index key value and each key value entry, in turn, has a pointer to the data row. Thus a nonclustered index contains a pointer to the physical location of the record. Each table can have 999 nonclustered indexes.

#### 3. UNIQUE INDEX:

This indexing does not allow the field to have duplicate values if the column is unique indexed. It can be applied automatically when a primary key is defined.

### WHAT IS THE PURPOSE OF ISOLATION LEVELS IN SQL?

**Ans.**

Transactions use an isolation level that specifies the extent to which a transaction must be isolated from any data modifications caused by other transactions. These also help in identifying which concurrency side-effects are permissible.

Please refer the below list for more clarity on the different type of levels.

#### I. READ COMMITTED.

It ensures that SELECT query will use committed values of the table only. If there is any active transaction on the table in some other session, then the SELECT query will wait for any such transactions to complete. Read Committed is the default transaction isolation level.

#### II READ UNCOMMITTED.

There is a transaction to update a table. But, it is not able to reach to any of these states like complete, commit or rollback. Then these values get displayed (as Dirty Read) in SELECT query of “Read Uncommitted” isolation transaction.

#### III. REPEATABLE READ.

This level doesn’t guarantee that reads are repeatable. But it does ensure that data won’t change for the life of the transaction once.

#### IV. SERIALIZABLE.

It is similar to Repeatable Read level. The only difference is that it stops Phantom Read and utilizes the range lock. If the table has an index, then it secures the records based on the range defined in the WHERE clause (like where ID between 1 and 3). If a table does not have an index, then it locks complete table.

#### V. SNAPSHOT.

It is similar to Serializable isolation. The difference is that Snapshot does not hold a lock on a table during the transaction. Thus allowing the table to get modified in other sessions. Snapshot isolation maintains versioning in Tempdb for old data. In case any data modification happens in other sessions then existing transaction displays the old data from Tempdb.

1. How can we swap the gender column in a employee table ie: male should replace by female and vice versa.
2. How to delete duplicate rows in a table.
3. Difference between index seek and index scan.
4. How to write your DOB using SQL date function.
5. Difference between cluster and non cluster index.
6. How to find 2nd or 3rd max and min salary of an employee.
7. How view is important for the security purpose.
8. Different type of joins and datatypes.
9. Difference between row number , rank and dense rank.
10. They can also give you any case scenario.
11. Difference between commit, rollback and save point.
12. Difference between with(nolock ) and readpast.
13. Explain the scenario of blocking and deadlock.
14. What is the use of nocount command.
15. Different type of triggers.

SQL Query Questions

1. SQL Query to find second highest salary of Employee
2. SQL Query to find Max Salary from each department.
3. How do you find all employees which are also manager?
4. Get position of 'o' in name 'John' from employee table?(locate)

**select** *locate*('o',**Name**) **from** Employee **where Name**='joe'

1. Get First\_Name from employee table after replacing 'o' with '$'

**select** *replace*(**name**,'o','$') **from** Employee

1. Get names of employees from employee table who has '%' in Last\_Name. Tip : Escape character for special characters in a query.

**Select** FIRST\_NAME **from** employee **where** Last\_Name **like** '%\%%'