

1>> How to create empty tables with the same structure as another table?

```
CREATE TABLE new_table
AS (SELECT * FROM old_table WHERE 1=2);
Eg:
```

```
CREATE TABLE student
AS (SELECT * FROM person WHERE 1=2);
```

This would create a new table called student that includes all columns from the person table, but no data from the person table

Other ways .

1. CREATE TABLE new_table SELECT * FROM original_table;
2. CREATE TABLE adminUsers SELECT * FROM users;
3. CREATE TABLE new_table LIKE original_table;

2>> Explain Lead() with example.

At many instances, user would really like to check the worth of the present row with the worth of the subsequent row. to resolve this problem in SQL Server's LEAD() window function are often used.

LEAD() :

Function provides access to a row at a set physical offset following this row. **LEAD()** function will allows to access data of the following row, or the row after the subsequent row, and continue on.

Syntax :

```
LEAD(return_value, offset [, default])
OVER (
    [PARTITION BY partition_expression]
    ORDER BY sort_expression [ASC | DESC]
)
```

3>> How to drop the primary key in mysql?

```
ALTER TABLE table_name
DROP PRIMARY KEY;
```

4>> List down the difference between primary key and candidate key.

Both Primary Key and Candidate key are used to get records from tables. These keys are also used to create relationship between tables. Primary Key and Candidate key both are used to identify records uniquely in a table.

Following are the important differences between Primary Key and Candidate key.

Sr. No.	Key	Primary Key	Candidate key
1	Definition	Primary Key is a unique and non-null key which identify a record uniquely in table. A table can have only one primary key.	Candidate key is also a unique key to identify a record uniquely in a table but a table can have multiple candidate keys.
2	Null	Primary key column value can not be null.	Candidate key column can have null value.
3	Objective	Primary key is most important part of any relation or table.	Candidate key signifies as which key can be used as Primary Key.
4	Use	Primary Key is a candidate key.	Candidate key may or may not be a primary key.

5>> Difference between having and group by clause.

Having Clause is basically like the aggregate function with the GROUP BY clause. The HAVING clause is used instead of WHERE with aggregate functions. While the **GROUP BY** Clause groups rows that have the same values into summary rows. The having clause is used with the where clause in order to find rows with certain conditions. The having clause is always used after the group By clause.

S.No. Having Clause

- It is used for applying some extra condition to the query.

- Having can be used without groupby clause,in aggregate function,in that

GroupBy Clause

The groupby clause is used to group the data according to particular column or row.

groupby can be used without having clause with the select

S.No.	Having Clause	GroupBy Clause
	case it behaves like where clause.	statement.
3.	The having clause can contain aggregate functions.	It cannot contain aggregate functions.
4.	It restricts the query output by using some conditions	It groups the output on basis of some rows or columns.

Extra tip>>> having vs where

The main difference between them is that the WHERE clause is used to specify a condition for filtering records before any groupings are made, while the HAVING clause is used to specify a condition for filtering values from a group.

Comparison Basis	WHERE Clause	HAVING Clause
Definition	It is used to perform filtration on individual rows.	It is used to perform filtration on groups.
Basic	It is implemented in row operations.	It is implemented in column operations.
Data fetching	The WHERE clause fetches the specific data from particular rows based on the specified condition	The HAVING clause first fetches the complete data. It then separates them according to the given condition.
Aggregate Functions	The WHERE clause does not allow to work with aggregate functions.	The HAVING clause can work with aggregate functions.
Act as	The WHERE clause acts as a pre-filter.	The HAVING clause acts as a post-filter.
Used with	We can use the WHERE clause with the SELECT, UPDATE, and DELETE statements.	The HAVING clause can only be used with the SELECT statement.
GROUP BY	The GROUP BY clause comes after the WHERE clause.	The GROUP BY clause comes before the HAVING clause.

6>> What are ' triggers '? What are the types of triggers in MySQL?

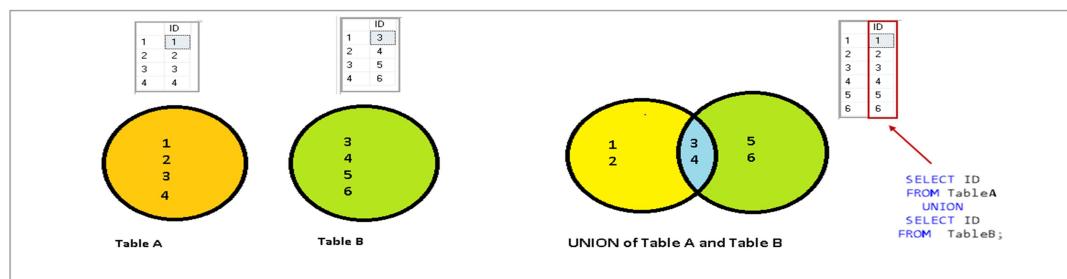
A MySQL **trigger** is a stored program (with queries) which is executed automatically to respond to a specific event such as insertion, updation or deletion occurring in a table.

There are 6 different types of triggers in MySQL:

- Before Update Trigger: As the name implies, it is a trigger which enacts before an update is invoked. ...
- After Update Trigger: ...
- Before Insert Trigger: ...
- After Insert Trigger: ...
- Before Delete Trigger: ...
- After Delete Trigger:

7>> Difference between Union and Union all .

The only difference between Union and Union All is that Union extracts the rows that are being specified in the query while Union All extracts all the rows including the duplicates (repeated values) from both the queries.



8>> What are different types of Normalization Levels or Normalization Forms ?

Normalization is a database design technique that reduces data redundancy and eliminates undesirable characteristics like Insertion, Update and Deletion Anomalies. Normalization rules divides larger tables into smaller tables and links them using relationships. The purpose of Normalisation in SQL is to eliminate redundant (repetitive) data and ensure data is stored logically.

Here is a list of Normal Forms in SQL:(mostly used 1,2,3,BCNF)

- 1NF (First Normal Form)
- 2NF (Second Normal Form)

- 3NF (Third Normal Form)
- BCNF (Boyce-Codd Normal Form)
- 4NF (Fourth Normal Form)
- 5NF (Fifth Normal Form)
- 6NF (Sixth Normal Form)

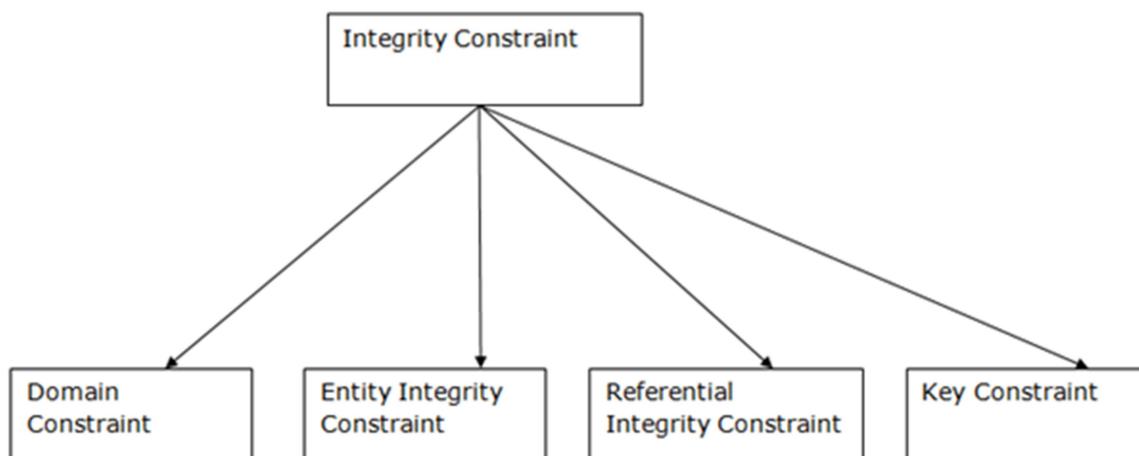
9>> **Can a FOREIGN KEY column contain NULL values?**

Yes, a foreign key can be null. When a foreign key's value is not known at the time of record generation then it's kept null, given that that column is created nullable (default value is null).

The following rules apply to foreign key definitions:

- A table can have many foreign keys
- **A foreign key is nullable if any part is nullable**
- A foreign key value is null if any part is null.

10>> **What is Entity integrity?**



An entity is any person, place, or thing to be recorded in a database. Each table represents an entity, and each row of a table represents an instance of that entity. For example, if *order* is an entity, the **orders** table represents the idea of an order and each row in the table represents a specific order.

To identify each row in a table, the table must have a primary key. The primary key is a unique value that identifies each row. This requirement is called the *entity integrity constraint*.

For example, the **orders** table primary key is **order_num**. The **order_num** column holds a unique system-generated order number for each row in the table.

11>> **What is De normalization?**

Denormalization is a database optimization technique in which we add redundant data to one or more tables. This can help us avoid costly joins in a relational database. Note that *denormalization* does not mean ‘reversing normalization’ or ‘not to normalize’. It is an optimization technique that is applied after normalization.

Basically, The process of taking a normalized schema and making it non-normalized is called denormalization, and designers use it to tune the performance of systems to support time-critical operations.

In a traditional normalized database, we store data in separate logical tables and attempt to minimize redundant data. We may strive to have only one copy of each piece of data in a database.

For example, in a normalized database, we might have a Courses table and a Teachers table. Each entry in Courses would store the teacherID for a Course but not the teacherName. When we need to retrieve a list of all Courses with the Teacher’s name, we would do a join between these two tables.

In some ways, this is great; if a teacher changes his or her name, we only have to update the name in one place.

The drawback is that if tables are large, we may spend an unnecessarily long time doing joins on tables.

Denormalization, then, strikes a different compromise. Under denormalization, we decide that we’re okay with some redundancy and some extra effort to update the database in order to get the efficiency advantages of fewer joins.

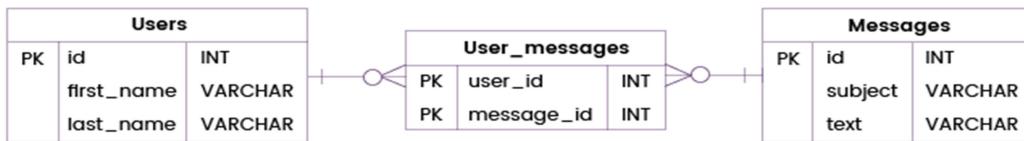
Pros of Denormalization:

1. Retrieving data is faster since we do fewer joins
2. Queries to retrieve can be simpler (and therefore less likely to have bugs), since we need to look at fewer tables.

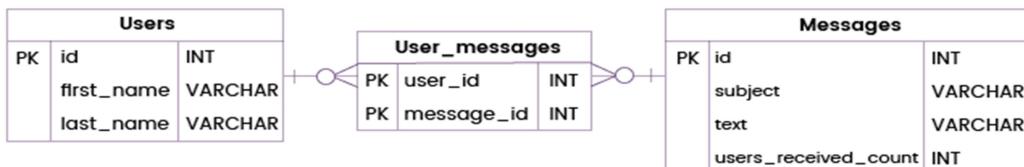
Cons of Denormalization:

1. Updates and inserts are more expensive.
2. Denormalization can make *update* and *insert* code harder to write.
3. Data may be inconsistent.
4. Data redundancy necessitates more storage.

Normalized database



Denormalized database

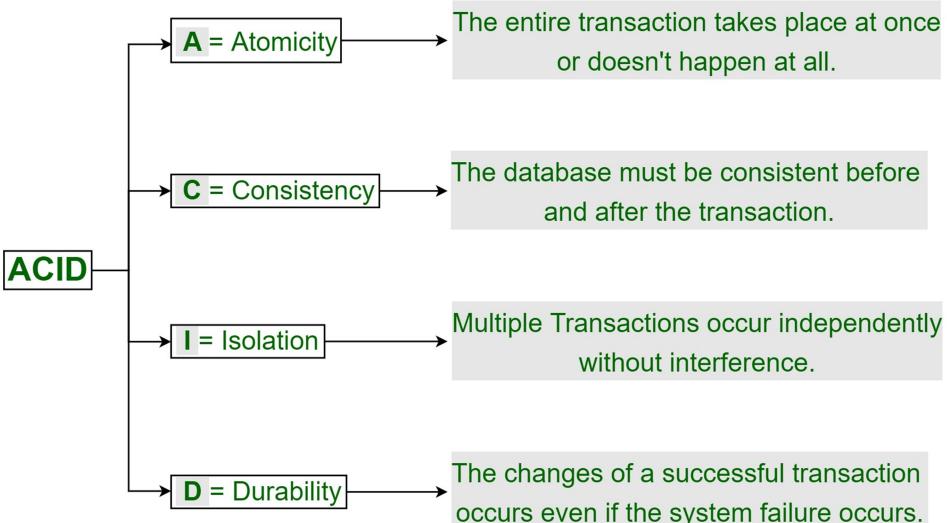


12>> What is a Transaction? 13>> acid properties

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.

ACID Properties in DBMS



14>> How to delete duplicate records in a table?

To delete the duplicate rows from the table in SQL Server, you follow these steps:

- Find duplicate rows using `GROUP BY` clause or `ROW_NUMBER()` function.
- `rank()` also can be used

- Use **DELETE** statement to remove the duplicate rows.

contact_id	first_name	last_name	email
1	Syed	Abbas	syed.abbas@example.com
2	Catherine	Abel	catherine.abel@example.com
3	Kim	Abercrombie	kim.abercrombie@example.com
4	Kim	Abercrombie	kim.abercrombie@example.com
5	Kim	Abercrombie	kim.abercrombie@example.com
6	Hazem	Abolrous	hazem.abolrous@example.com
7	Hazem	Abolrous	hazem.abolrous@example.com
8	Humberto	Acevedo	humberto.acevedo@example.com
9	Humberto	Acevedo	humberto.acevedo@example.com
10	Pilar	Ackeman	pilar.ackeman@example.com

There are many duplicate rows (3,4,5), (6,7), and (8,9) for the contacts that have the same first name, last name, and email.

1)

```
WITH cte AS (
    SELECT
        contact_id,
        first_name,
        last_name,
        email,
        ROW_NUMBER() OVER (
            PARTITION BY
                first_name,
                last_name,
                email
            ORDER BY
                first_name,
                last_name,
                email
        ) row_num
    FROM
        sales.contacts
)
DELETE FROM cte
WHERE row_num > 1;
```

2)

```
SELECT contact_id,
       first_name,
       last_name,
       email
  FROM sales.contacts
 ORDER BY first_name,
          last_name,
          email;
```

contact_id	first_name	last_name	email
2	Catherine	Abel	catherine.abel@example.com
6	Hazem	Abolrous	hazem.abolrous@example.com
8	Humberto	Acevedo	humberto.acevedo@example.com
3	Kim	Abercrombie	kim.abercrombie@example.com
10	Pilar	Ackeman	pilar.ackeman@example.com
1	Syed	Abbas	syed.abbas@example.com

Output after performing delete option

15>> What is SELF REFERENTIAL INTEGRITY?

Self referential integrity constraint refers to two columns related by referential integrity which belong to the same table.

For example ManagerId column in employees table is a foreign key which refers to EmployeeId in the same table.

Self-Referential Integrity

- A foreign key can reference an attribute(s) in the *same* relation as itself.

Example - A manager must also be an employee :-

SUPERVISE	
Mger	Emp
E1	E1
E1	E2
E1	E3
E2	E4
E2	E5
E2	E6
E3	E7



16>> Define deadlock with an example. Write down the necessary conditions for the occurrence of deadlock.(not a dbms topic>>related to OS)

A deadlock in OS is a situation in which more than one process is blocked because it is holding a resource and also requires some resource that is acquired by some other process. The four necessary conditions for a deadlock situation to occur are mutual exclusion, hold and wait, no preemption and circular set.

17>> What is the purpose of ROW_NUMBER()?

ROW_NUMBER function is a SQL ranking function that **assigns a sequential rank number to each new record in a partition**. When the SQL Server ROW NUMBER function detects two identical values in the same partition, it assigns different rank numbers to both.

The syntax for ROW_NUMBER function in SQL is as follows-

```
ROW_NUMBER() OVER (
    [PARTITION BY expr1, expr2,...]
    ORDER BY expr1 [ASC | DESC], expr2, ...
)
```

18>> What is the purpose of the IN keyword? Give an example.

The **IN** operator allows you to specify multiple values in a **WHERE** clause.

The **IN** operator is a shorthand for multiple **OR** conditions.

Syntax

```
SELECT column_name(s)
FROM table_name
WHERE column_name IN (value1, value2, ...);
```

19>> What is a sub-query? Explain the property of sub-query.

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.

Important rules for Subqueries:

- You can place the Subquery in a number of SQL clauses: **WHERE** clause, **HAVING** clause, **FROM** clause. Subqueries can be used with **SELECT**, **UPDATE**, **INSERT**, **DELETE** statements along with expression operator. It could be equality operator or comparison operator such as **=, >, =, <=** and Like operator.
- A subquery is a query within another query. The outer query is called as **main query** and inner query is called as **subquery**.
- The subquery generally executes first when the subquery doesn't have any **co-relation** with the **main query**, when there is a co-relation the

- parser takes the decision **on the fly** on which query to execute on **precedence** and uses the output of the subquery accordingly.
- Subquery must be enclosed in parentheses.
 - Subqueries are on the right side of the comparison operator.
 - [ORDER BY](#) command **cannot** be used in a Subquery. [GROUPBY](#) command can be used to perform same function as ORDER BY command.

20>> **How to add new columns , and change column names in MySQL?**

```
ALTER TABLE table_name
    ADD new_column_name
    column_definition;
```

```
ALTER TABLE contacts
    ADD last_name varchar(40) ;
```

21>> **Difference between delete and truncate.**

Both the TRUNCATE statement and the DELETE statement are included in the category of SQL queries for deleting the data stored in a table. They carry out deletion operations on records or rows of a table that are no longer needed. A condition is applied before each entry in the table that is being deleted when using the DELETE command. To put it another way, it is possible to delete one or more rows all at once. However, with the TRUNCATE command, each row is removed from the table simultaneously.

When we delete something using the DELETE query, a log entry is produced for each row in the transaction log. Because of this, we are able to retrieve the records by using ROLLBACK before committing the changes. In addition, if we use the TRUNCATE operation, we can retrieve the records by using the ROLLBACK procedure. TRUNCATE, on the other hand, only tracks the deallocation of the page where the data is kept. This is the key difference.

Conclusion

Both the SQL DELETE and SQL TRUNCATE commands can be used to remove records from a table. However, the DELETE command employs the WHERE clause to specify rows in a table for deletion action, whereas the TRUNCATE command does not use any clause and deletes rows all at once. This is the primary distinction between the two commands.

Keep in mind that the TRUNCATE command can be undone in the same way that the DELETE command can.

<u>DELETE</u>	<u>TRUNCATE</u>
It is a DML command.	It is a DDL command
Used to remove specific rows or all the rows depending upon our requirement.	Can only be used to remove all the rows from a table.
It can use the WHERE clause.	It can't use the WHERE clause.
It is comparatively slower.	It is faster.
It records the deleted rows in the transactions log.	Since it removes the page where our table's data is stored, it stores the page in the transactional log.

>>>>>>>>>

Delete vs drop**D R O P I N S Q L****V E R S U S****D E L E T E I N S Q L**

DROP IN SQL	DELETE IN SQL
A command in SQL to delete databases or other objects (such as tables) from an RDBMS	A command in SQL to delete rows from a table in a database of an RDBMS
A data definition language	A data manipulation language
Helps to delete databases or related objects	Helps to remove the records in a table