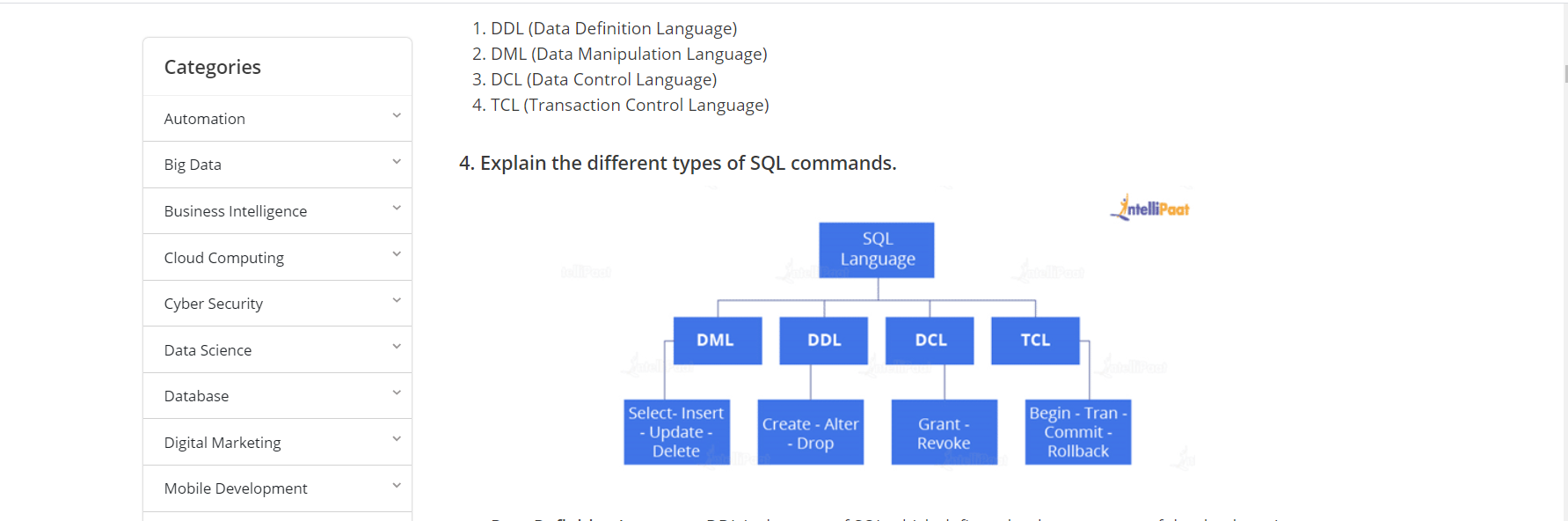
What are the different types of command we have in sql?

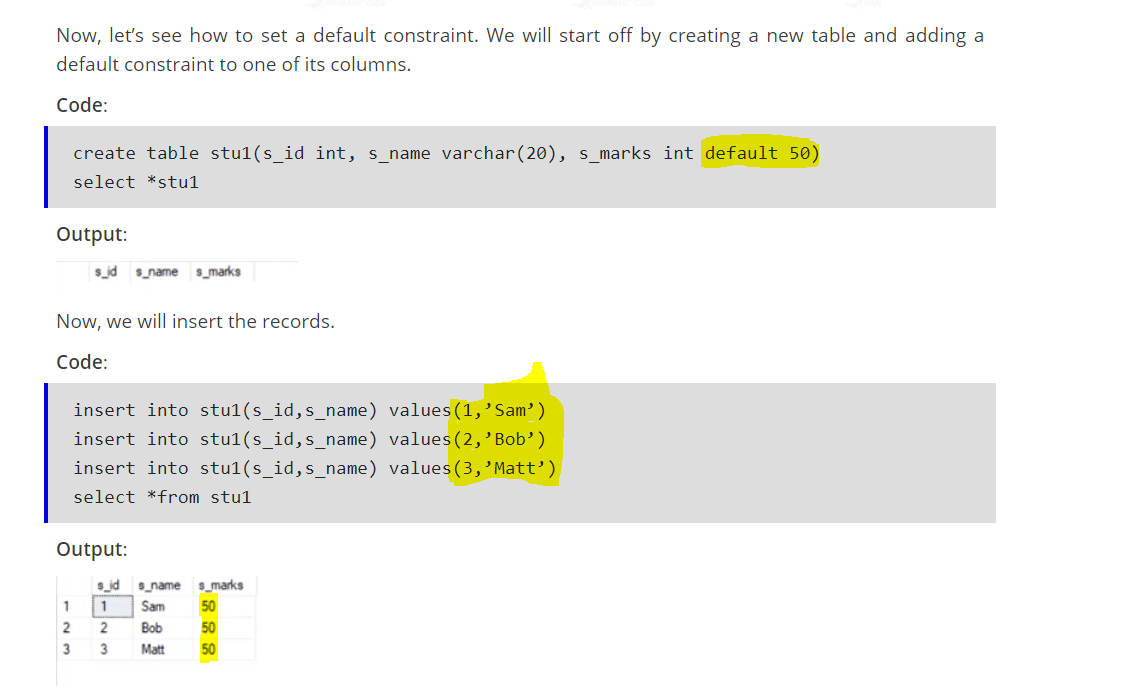


What is default constraints in SQL?

**What is a default constraint?**

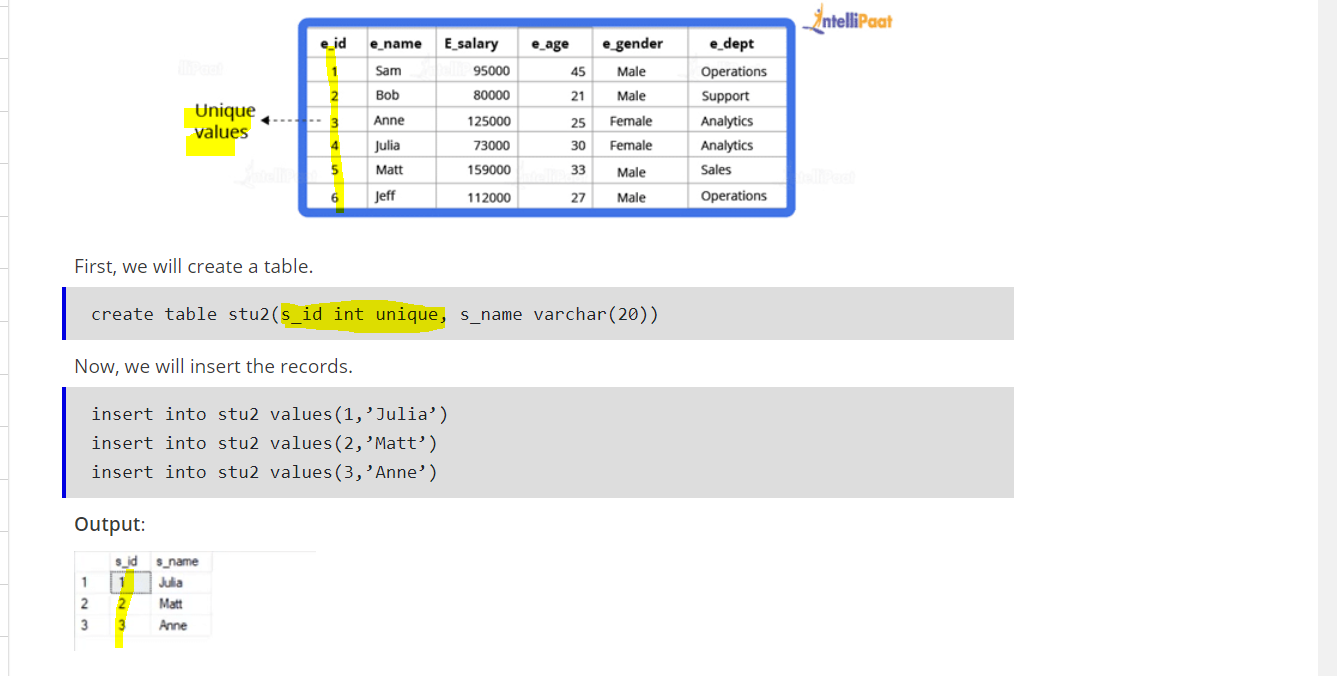
Constraints are used to specify some sort of rules for processing data and limiting the type of data that can go into a table. Now, let’s understand the default constraint.

The default constraint is used to define a default value for a column so that the default value will be added to all the new records if no other value is specified. For example, if we assign a default constraint for the E\_salary column in the below table and set the default value as 85000, then all the entries of this column will have a default value of 85000 unless no other value has been assigned during the insertion.

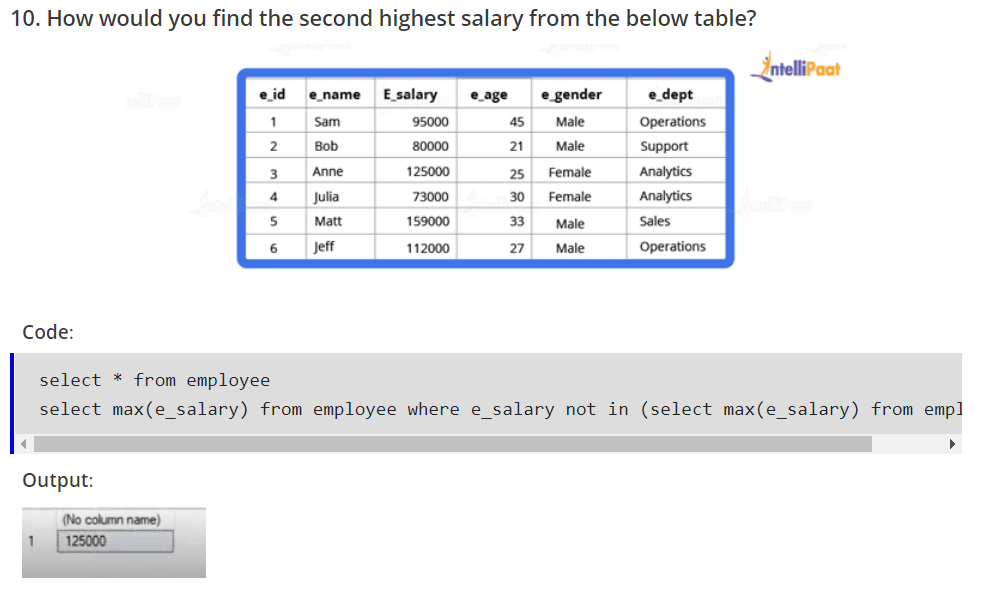


What is the unique constraints in SQL?

**Unique constraints** ensure that all the values in a column are different. For example, if we assign a unique constraint to the e\_name column in the below table, then every entry in this column should have a unique value.



2nd Highest Salary:

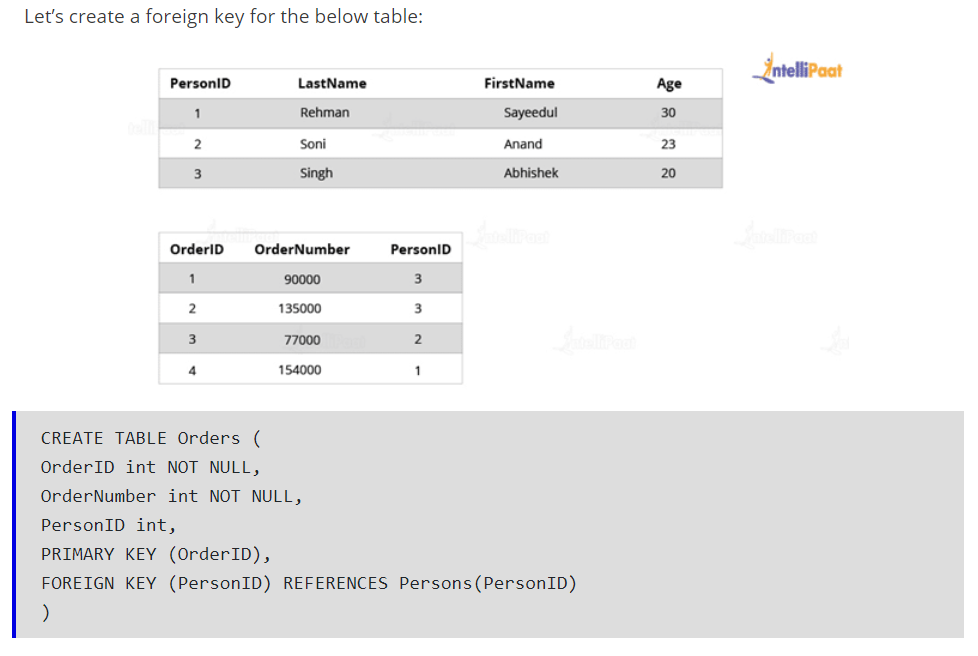


### ****What is the difference between a primary key and a unique key?****

Both primary and unique keys carry unique values, but the primary key cannot have a null value while the unique key can have it. And in a table, there cannot be more than one primary key, but it can have multiple unique keys.

### ****What is a foreign key?****

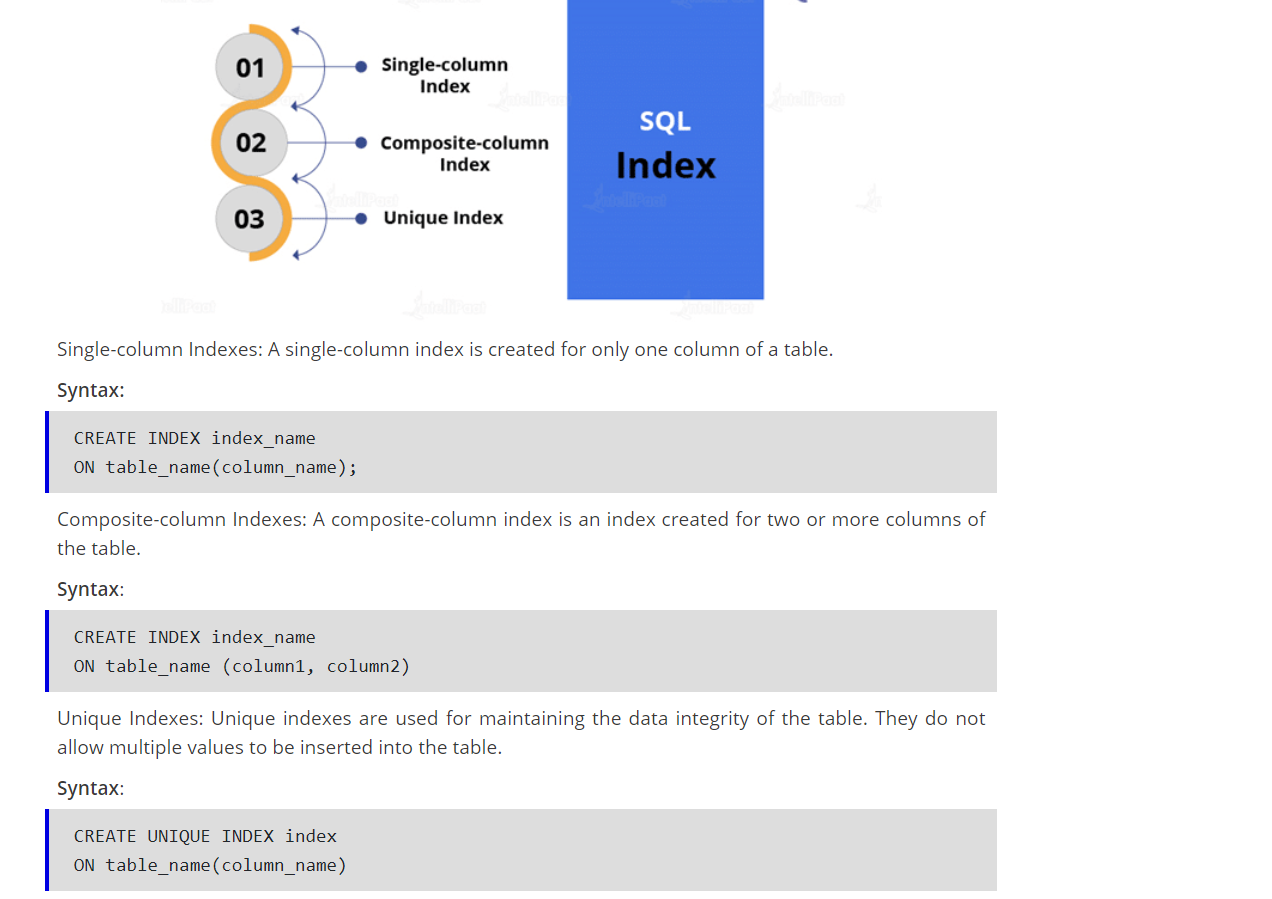
A foreign key is an attribute or a set of attributes that references to the primary key of some other table. Basically, it is used to link together two tables.



### ****What is an index?****

Indexes help speed up searching in the database. If there is no index on any column in the WHERE clause, then SQL Server has to skim through the entire table and check each and every row to find matches, which might result in slow operation on large data.

Indexes are used to find all rows matching with some columns and then to skim through only those subsets of the data to find the matches.



### ****What is the ACID property in a database?****

The full form of ACID is atomicity, consistency, isolation, and durability. To check the reliability of transactions, ACID properties are used.

* Atomicity refers to completed or failed transactions, where the transaction refers to a single logical operation on data. This implies that if any aspect of a transaction fails, the whole transaction fails, and the database state remains unchanged.
* Consistency means that the data meets all of the validity guidelines. The transaction never leaves the database without finishing its state.
* Concurrency management is the primary objective of isolation.
* Durability ensures that once a transaction is committed, it will occur regardless of what happens in between, such as a power outage, a fire, or some other kinds of issues.

### ****What is the need for group functions in SQL?****

Group functions operate on a series of rows and return a single result for each group. COUNT(), MAX(), MIN(), SUM(), AVG(), and VARIANCE() are some of the most widely used group functions.

**What is the Difference between Delete, Truncate and Drop in SQL?**

**Delete comes in Data Manipulation Language (like Update)**

**Drop comes in Data Definition Language (Like Create)**

**Truncate comes in Data Definition Language.**

**Delete:**

**Delete from table\_name:**

**It will delete all the rows one by one and before deleting it copy the to a log file. Hence delete statement is slow and we can recover our data using rollback. (if we don’t commit it)**

**As delete command only delete the data available on the table hence after delete command if we query the below command, we can see the table definition in output.**

**Desc table\_name;**

**We can delete specific rows from a table using the where clause.**

**Delete from table\_name where id =123.**

**Drop:**

**Drop table tableName;**

**It will delete table and table schema from the table.**

**Desc tableName; --🡪 error as no object present in the database with tablename.**

**Truncate:**

**Truncate is fast compare to delete as it deletes all the rows at a same time.**

**We cannot use where clause in truncate which we can do it delete.**

**As it doesn’t store rows values in log file so it is fast, and we cannot rollback the data once it is truncated.**

**Truncate table\_name;**

**Desc table\_name 🡪get the schema details of the table**

### ****What is wrong with the below-given SQL query?****

SELECT gender, AVG(age) FROM employee WHERE AVG(age)>30 GROUP BY gender

Will get error if we execute the code.

This basically means that whenever we are working with aggregate functions and we are using GROUP BY, we cannot use the WHERE clause. Therefore, instead of the WHERE clause, we should use the HAVING clause.

select e\_gender, avg(e\_age) from employee group by e\_gender having avg(e\_age)>30

### ****What are Views? Give an example.****

Views are virtual tables used to limit the tables that we want to display, and these are nothing but the result of a SQL statement that has a name associated with it. Since views are not physically present, they take less space to store.

Let’s consider an example. In the below employee table, say, we want to perform multiple operations on the records with gender ‘Female’. We can create a view-only table for the female employees from the entire employee table.

**Syntax**:

create view female\_employee as select \* from employee where e\_gender=’Female’

select \* from female\_employee

### ****What do you understand by Self Join?****

Self-Join in SQL is used for joining a table with itself. Here, depending upon some conditions, each row of the table is joined with itself and with other rows of the table.

**To delete the duplicate row from a table we required self join.**

### ****What is the difference between Union and Union All operators?****

The **Union** operator is used to combine the result set of two or more select statements. For example, the first select statement returns the fish shown in Image A, and the second returns the fish shown in Image B. Then, the Union operator will return the result of the two select statements as shown in Image A U B. Also, if there is a record present in both tables, then we will get only one of them in the final result.

**Union All** gives all the records from both tables including the duplicates.

### ****What is the use of the Intersect operator?****

The **Intersect** operator helps combine two select statements and returns only those records that are common to both the select statements. So, after we get Table A and Table B over here and if we apply the Intersect operator on these two tables, then we will get only those records that are common to the result of the select statements of these two.

This is same like inner join.

### ****What is the difference between BETWEEN and IN operators in SQL?****

To represent rows based on a set of values, we use the BETWEEN operator. The values may be numbers, text, or dates. The BETWEEN operator returns the total number of values that exist between two specified ranges.

Between we used to select a date range.

To search for values within a given range of values, the IN condition operator is used. If we have more than one value to choose from, we can use the IN operator.

### ****Describe how to delete duplicate rows using a single statement but without any table creation.****

DELETE e1 FROM EMPLOYEE e1, EMPLOYEE e2 WHERE e1.name = e2.name AND e1.id > e2.id

The SQL query above will delete the rows, where the name fields are duplicated, and it will retain only those unique rows in which the names are unique and the ID fields are the lowest. That is, rows with IDs 5 and 6 are deleted, whereas rows with IDs 1 and 2 are retained.

### ****Can you identify the employee who is having the third-highest salary from the given Employee table (with salary-related data)?****

### Below is a simple query to find out the employee who has the third-highest salary. The functions RANK, DENSE RANK, and ROW NUMBER are used to obtain the increasing integer value by imposing the ORDER BY clause in the SELECT statement, based on the ordering of rows. The ORDER BY clause is necessary when we use RANK, DENSE RANK, or ROW NUMBER functions. On the other hand, the PARTITION BY clause is optional.

WITH CTE AS

(

SELECT Name, Salary, RN = ROW\_NUMBER() OVER (ORDER BY Salary DESC) FROM EMPLOYEE

)

SELECT Name, Salary FROM CTE WHERE RN =3

### ****What is the difference between HAVING and WHERE clauses?****

The distinction between HAVING and WHERE clauses in SQL is that while the WHERE clause cannot be used with aggregates, we use the HAVING clause with the aggregated data. The WHERE clause works on the data from a row and not with the aggregated data.

SELECT Name, Salary FROM Employee WHERE Salary >=50000

SELECT Department, SUM(Salary) AS total FROM Employee GROUP BY Department HAVING SUM(Salary)>70000

### Difference between rank, dense rank and row number?

### If we don’t have any duplicate value in salary column then all 3 return same result.

### 

**But when we have the duplicate data then return type is different for everyone.**

**Row number () – return the same sequence as it was earlier.**

**Rank 🡪 rank will return same value for all the duplicate element but after the duplicate the rank will be the same if there had been no duplicate.**

**Dense rank: for duplicate it assign the same value and after that it would be a sequence number.**

