WEEK 1 : SQL for DATA SCIENCE

Q1.What is database?

A database is a systematic collection of data. It stores and manipulates the data. Databases make data management easy.

Q2.What is SQL and why it is important?

SQL is used to create a database, define its structure, implement it, and perform various functions on the database. SQL is also used for accessing, maintaining, and manipulating already created databases. SQL is a well built language for entering data, modifying data, and extracting data in a database.

Q3.What is the main role of SQL in Data Science?

Data Scientists work on structured data located on database servers. So, it's crucial to connect the databases using Python or R and pull the data in order to build machine learning models.

Q4.Define MySQL ?

MySQL is a database management system.
To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network.

Q5.What do you meant by DBMS?

Database Management System (DBMS) is a software that is used to define, create and maintain a database and provides controlled access to the data.

Q6.Difference between DBMS and RDBMS

DBMS stores data as file.	RDBMS stores data in tabular form.

Data elements need to access individually.	Multiple data elements can be accessed at the same time.
No relationship between data.	Data is stored in the form of tables which are related to each other.
Normalization is not present.	Normalization is present.
DBMS does not support distributed database.	RDBMS supports distributed database.
It stores data in either a navigational or hierarchical form.	It uses a tabular structure where the headers are the column names, and the rows contain corresponding values.
It deals with small quantity of data.	It deals with large amount of data.

Q7. Explain RDBMS.

The software used to store, manage, query, and retrieve data stored in a relational database is called a relational database management system (RDBMS). The RDBMS provides an interface between users and applications and the database, as well as administrative functions for managing data storage, access, and performance.

Q8.ACID property in SQL

The ACID refers to the four key properties of a transaction: atomicity, consistency, isolation, and durability.

Q9.What are the basic commands in SQL?

SELECT - extracts data from a database.

FROM – From where to select.
UPDATE - updates data in a database.
DELETE - deletes data from a database.
INSERT INTO - inserts new data into a database.
CREATE DATABASE - creates a new database.
ALTER DATABASE - modifies a database.
CREATE TABLE - creates a new table.

Q10. Why data constrains are used in SQL?

SQL constraints are used to specify rules for the data in a table. Constraints are used to limit the type of data that can go into a table.

Q11. Brief out some constrains which we mostly use in SQL?

1. NOT NULL - Ensures that a column cannot have a NULL value
2. UNIQUE - Ensures that all values in a column are different
3. PRIMARY KEY - A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table
4. FOREIGN KEY - Prevents actions that would destroy links between tables
5. CHECK - Ensures that the values in a column satisfies a specific condition
6. DEFAULT - Sets a default value for a column if no value is specified
7. CREATE INDEX - Used to create and retrieve data from the database very quickly.

Q12. What are some of famous examples by whom mysql is used ?

- Huge websites like Facebook, Twitter, Airbnb, Booking.com, Uber, GitHub, YouTube, etc.
- Content Management Systems like WordPress, Drupal, Joomla!, Contao, etc.
- A very large number of web developers around the world

Q13. How to create a table that includes columns of person,s id , name and address in sql

```
CREATE TABLE Persons (

    PersonID int,

    Name varchar(50),

    Address varchar(100),

    );
```

Q14… What is the syntax to show a list of multiple tables from any database in sql?

SYNTAX = Show Tables

Q15.. Write a query to extract all the values from one table from database in sql

Select * From Table_name

Q16.What is the difference between Atomicity and Consistency?

Atomicity: A transaction must be an atomic unit of work; either all of its data modifications are performed or none of them is performed.

Consistency: When completed, a transaction must leave all data in a consistent state. In a relational database, all rules must be applied to the transaction's modifications to maintain all data integrity.

Q17.What are the applications of SQL?

1.Writing data integration scripts.
2.Setting and running analytical queries.
3.Retrieving subsets of information within a database for analytics applications and transaction processing.
4.Adding, updating, and deleting rows and columns of data in a database.

Q18.What is meant by table and field in SQL?

An organized data in the form of rows and columns is said to be a table. Simply put, it is a collection of related data in a table format.

Here rows and columns are referred to as tuples and attributes, and the number of columns in a table is referred to as a field. In the record, fields represent the characteristics and attributes and contain specific information about the data.

Q19..What are SQL operators?

The SQL operators can be categorized into the following types:Arithmetic Operators: For mathematical operations on numerical data

- addition (+)
- subtraction (-)
- multiplication (*)
- division (/)
- remainder/modulus (%)

- Logical Operators: For evaluating the expressions and return results in True or False
    - ALL
    - AND
    - ANY
    - ISNULL
    - EXISTS
    - BETWEEN
    - IN
    - LIKE
    - NOT
    - OR
    - UNIQUE
- Comparison Operators: For comparisons of two values and checking whether they are the same or not
    - equal to (=)
    - not equal to (!= or <>)
    - less than (<),
    - greater than (>)
    - less than or equal to (<=)
    - greater than or equal to (>=)
    - not less than (!<)
    - not greater than (!>)
- Bitwise Operators: For bit manipulations between two expressions of integer type. It first performs conversion of integers into binary bits and then applied operators
    - AND (& symbol)
    - OR (|, ^)
    - NOT (~)
- Compound Operators: For operations on a variable before setting the variable's result to the operation's result
    - Add equals (+=)
    - subtract equals (-=)
    - multiply equals (*=)
    - divide equals (/=)
    - modulo equals (%=)
- String Operators: For concatenation and pattern matching of strings
    - + (String concatenation)

- += (String concatenation assignment)
- % (Wildcard)
- [] (Character(s) matches)

Q20..What is a data warehouse?

A data warehouse is a large store of accumulated data, from a wide range of sources, within an organization. The data helps drive business decisions.

Q21.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.

Q22.What is the difference between SQL and MySQL?

SQL is a standard language for retrieving and manipulating structured databases. On the contrary, MySQL is a relational database management system, like SQL Server, Oracle or IBM DB2, that is used to manage SQL databases.

Q23.What is a Query?

A query is a request for data or information from a database table or combination of tables. A database query can be either a select query or an action query.

```
SELECT fname, lname    /* select query */
```

Q24.What is Cursor? How to use a Cursor?

A database cursor is a control structure that allows for the traversal of records in a database. Cursors, in addition, facilitates processing after traversal, such as retrieval, addition, and deletion of database records. They can be viewed as a pointer to one row in a set of rows.

Example: DECLARE a cursor after any variable declaration. The cursor declaration must always be associated with a SELECT Statement.

Q25.What is the purpose of DEALLOCATE statement ?

DEALLOCATE statement is used to delete the cursor definition and release the associated resources.

WEEK 2: SQL Data Types

1)What is Data Type?

A Data Type in SQL Server is defined as the type of data that any column or variable can store. It is a type of data that an object holds like integer, character, string, etc. While creating any table or variable, in addition to specifying the name, you also set the Type of Data it will store.

2)What are the Type of SQL Type.

1. Binary Data type
2. Exact Numeric Data type
3. Approximate Numeric Data type
4. Character String Data type
5. Date and Time Data type
6. Unicode Character String Data type

3)What is syntax of create base in my SQL Server?

```
CREATE DATABASE mydatabase;
```

4)How can we create tables in SQL?

The command to create a table in SQL is extremely simple: We will start off by giving the keywords, CREATE TABLE, then we will give the name of the table. After that in braces, we will list out all the columns along with their data types.

For example, if we want to create a simple employee table:

```
CREATE TABLE employee (
 name varchar(25),
 age int,
 gender varchar(25), ....
);
```

5)How can we insert data in SQL?

It is possible to write the INSERT INTO statement in two ways:

1. Specify both the column names and the values to be inserted:

```
INSERT INTO table_name (column1, column2, column3, ...)
VALUES (value1, value2, value3, ...);
```

2. If you are adding values for all the columns of the table, you do not need to specify the column names in the SQL query. However, make sure the order of the values is in the same order as the columns in the table. Here, the INSERT INTO syntax would be as follows:

INSERT INTO table_name VALUES (value1, value2, value3, ...);

6)How can we change a table name in SQL?

We will start off by giving the keywords ALTER TABLE, then we will follow it up by giving the original name of the table, after that, we will give in the keywords RENAME TO and finally, we will give the new table name.

For example, if we want to change the "employee" table to "employee_information", this will be the command:

```
ALTER TABLE employee
RENAME TO employee_information;
```

7)What are user defined datatypes and when you should go for them?

User defined datatypes let you extend the base SQL Server datatypes by providing a descriptive name, and format to the database. Take for example, in your database, there is a column called Flight_Num which appears in many tables. In all these tables it should be varchar(8). In this case you could create a user defined datatype called Flight_num_type of varchar(8) and use it across all your tables.

8)What are constraints?

Constraint can be used to specify the limit on the data type of table. Constraint can be specified while creating or altering the table statement.

9)What are 5 major constraints are used in SQL?

There are five types of constraints in SQL Server:

- Primary Key Constraint
- Foreign Key Constraint
- Unique Constraint
- Check Constraint
- Default Constraint

10)What is primary key?

A primary key is a combination of fields which uniquely specify a row. This is a special kind of unique key, and it has implicit NOT NULL constraint. This means, Primary key values cannot be NULL.

11)What is a unique key?

A Unique key constraint uniquely identifies each record in a database. This provides uniqueness for the column or set of columns. A Primary key constraint has automatic unique constraint defined on it. There can be many unique constraints defined per table, but only one Primary key constraint defined per table.

12) What is a foreign key?

A foreign key is one table which can be related to the primary key of another table. Relationships need to be created between two tables by referencing the foreign key with the primary key of another table.

13)What the syntax for the INSERT statement when inserting a single record using the VALUES keyword in SQL Server

```
INSERT INTO table
(column1, column2, ... )
VALUES
(expression1, expression2, ... ),
(expression1, expression2, ... ),
...;
```

14)Write to add column to a table, you use the ALTER TABLE ADD syntax:

```
ALTER TABLE table_name
ADD
   new_column_name column_definition
   [FIRST | AFTER column_name]
```

15) Write To remove existing tables, you use the MySQL DROP TABLE statement.

```
DROP TABLE  table_name;
```

16)What is the difference between primary key and unique constraints?

The primary key cannot have NULL values, the unique constraints can have NULL values. There is only one primary key in a table, but there can be multiple unique constraints. The primary key creates the clustered index automatically but the unique key does not.

17) What is the purpose of TRUNCATE command?

TRUNCATE command is used to delete all the rows from the table and free the space containing the table.

```
TRUNCATE TABLE Candidates;
```

18) What is the difference between DROP and TRUNCATE statements?

If a table is dropped, all things associated with the tables are dropped as well. This includes - the relationships defined on the table with other tables, the integrity checks and constraints, access privileges and other grants that the table has. To create and use the table again in its original form, all these relations, checks, constraints, privileges and relationships need to be redefined. However, if a table is truncated, none of the above problems exist and the table retains its original structure.

19) What is the difference between DELETE and TRUNCATE statements?

The TRUNCATE command is used to delete all the rows from the table and free the space containing the table.
The DELETE command deletes only the rows from the table based on the condition given in the where clause or deletes all the rows from the table if no condition is specified. But it does not free the space containing the table

20) How to create empty tables with the same structure as another table?

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;
```

21) How can we disable a cursor?

To disable the cursor, use the CLOSE command.

22) what is the command for removing the cursor definition?

Use the DEALLOCATE command to remove the cursor definition and free up the resources connected with it.

23) What is a Self-Join?

A self-join is a type of join that can be used to connect two tables. As a result, it is a unary relationship. Each row of the table is attached to itself and all other rows of the same table in a self-join. As a result, a self-join is mostly used to combine and compare rows from the same database table.

24) What is the use of MINUS operator?

The MINUS operator is used to return rows from the first query but not from the second query.

25) In which format is data stored in DBMS and RDBMS?

Data is stored in a hierarchical format in a DBMS, whereas an RDBMS uses a table with headers that serve as column names and rows that hold the associated values.

WEEK 3:SELECT Statements,Arithmetic Operators,Mathematical Functions etc

Q1- Query all columns for all rows in the given CITY table.?

ID	NUMBER
NAME	VARCHAR(20)
COUNTRYCODE	VARCHAR(5)
DISTRICT	VARCHAR(20)
POPULATION	NUMBER

Ans1

SELECT * FROM CITY

Q2- Query all columns for a country with ID = 2345 in the given CITY table?

Ans2 – SELECT * FROM CITY

WHERE ID = 2345

Q3- Query all cities in the America from the CITY Table. The country code for America is USA?

Ans3 – SELECT NAME FROM CITY

WHERE COUNTRYCODE = 'USA'

Q4- Query a list of District and population from the CITY table for JAPAN. The country code for Japan is JPY?

Ans4 – SELECT DISTRICT, POPULATION FROM CITY

WHERE COUNTRYCODE = 'JPY'

Q5 Query all columns in the given table named CITY where population of the people is larger than 100000 and country code for INDIA is IN.

ANS5

SELECT * FROM CITY

WHERE POPULATION > 100000 AND COUNTRYCODE = 'IN'

Q6 Query a city names from the CITY that have an Even ID number?

Ans6 SELECT NAME FROM CITY

WHERE ID % 2 = 0

Q7- Query a list of city names starting with vowels (‘a’, ‘e’, ‘I’, ‘o’, ‘u’) from CITY. Result cannot contain duplicates?

Ans7 SELECT DISTINCT NAME FROM CITY

WHERE NAME LIKE ‘[AEIOU]%’;

Q8. Query a list of city names starting with vowels and ending with vowels. Result cannot contain duplicates?

Ans8 SELECT DISTINCT NAME FROM CITY

WHERE NAME LIKE ‘[AEIOU]%[AEIOU]”

Q9- Query a list of city names not starting with vowels (‘a’, ‘e’, ‘I’, ‘o’, ‘u’) from CITY. Result cannot contain duplicates?

Ans9 SELECT DISTINCT NAME FROM CITY

WHERE NAME LIKE ‘[^AEIOU]%’ ;

Q10- Query a list of city names not ending with vowels (‘a’, ‘e’, ‘I’, ‘o’, ‘u’) from CITY. Result cannot contain duplicates?

Ans10 SELECT DISTINCT NAME FROM CITY

WHERE NAME LIKE ‘%[^AEIOU]’ ;

Q11- Query a list of city names that either do not start with vowels or do not end with vowels. Result cannot contain duplicates?

Ans11- SELECT DISTINCT NAME FROM CITY

WHERE NAME NOT LIKE ‘[AEIOU]%’ OR NAME NOT LIKE ‘%[AEIOU]’;

Q12- What is an Alias?

Ans12 - Aliases are the temporary names given to table or column for the purpose of a particular SQL query. It is used when name of column or table is used other than their original names, but the modified name is only temporary. Aliases are created to make table or column names more readable. The renaming is just a temporary change and table name does not change in the original database. Aliases are useful when table or column names are big or not very readable. These are preferred when there are more than one table involved in a query.

Eg SELECT NAME AS CITY_NAME FROM CITY;

Q13. Write a query to retrieve first 5 rows from CITY table where COUNTRY IS CHINA The country code of china is CN?

ANS13. SELECT * FROM CITY

WHERE COUNTRY = 'CN'

LIIMIT 5;

Q14 Query the table to display city and population of all countries except INDIA. Country Code of INDIA is IN?

Ans14 SELECT NAME, POPULATION FROM CITY

WHERE COUNTRYCODE <> 'IN'

Q15- Query the CITY Table to display the column in following way – DISTRICT CITY-NAME COUNTRY-NAME and ailias this as FULL NAME. Display the corresponding the popoylatiob also?

ANS15 SELECT (DISTRICT+ ' ' + NAME + ' ' + COUNTRY) AS FULLNAME, POPULATION

FROM CITY

Q16 Given the following table. Query the SALES table to display OrderID, ProductID, Unitprice, Quantity and CostPrice. CostPrice is calculated as multiplication of Unitprice and CostPrice.

OrderID	INTEGER
ProductID	INTEGER
Unitprice	REAL
Quantity	INT

Ans16 SELECT OrderID, ProductID, Unitprice, Quantity, Unitprice * Quantity As CostPrice

FROM SALES

Q17. Given the following INVENTORY table. Query the table to show which products need reordering and display the corresponding product Id?

ProductID	INTEGER
ProductName	VARCHAR(256)

UnitsInStock	INT
ReorderLevel	INT

ANS17 SELECT ProductID, ProductName FROM INVENTORY

WHERE UnitsInStock < ReorderLevel

Q18.. How can we select a pattern from a specific column?

The following SQL statement selects all customers with a City containing the pattern "es":

Example

```
SELECT * FROM Customers
WHERE City LIKE '%es%';
```

Q19..How to Select all records where the second letter of elements in the Column named City is an "a"?

Select all records where the second letter of the City is an "a".

```
SELECT * FROM Customers
WHERE City LIKE '_a%';
```

Q20) What is a Self-Join?

A self JOIN is a case of regular join where a table is joined to itself based on some relation between its own column(s). Self-join uses the INNER JOIN or LEFT JOIN clause and a table alias is used to assign different names to the table within the query.

Q21) What are the usecases of Aliases?

Aliases can be useful when:

- There are more than one table involved in a query
- Functions are used in the query
- Column names are big or not very readable
- Two or more columns are combined together

Q22) What is the use of COUNT() function?

The COUNT() function returns the number of rows that matches a specified criterion.

Q23) Which function returns the total sum of a numeric column?

SUM()

Q24) What is the usecase of AVg() function?

The AVG() function returns the average value of a numeric column.

Q25) Give example for Alias Column Syntax?

Alias Column Syntax

SELECT column_name AS alias_name

FROM table_name;

WEEK 4: String and Date Fucntions

1) Differentiate CHAR_LENGTH and LENGTH?

CHAR_LENGTH is character count whereas the LENGTH is byte count. The numbers are same for Latin characters but they are different for Unicode and other encodings.

2) Give string types available for column?

The string types are:

- SET
- BLOB
- ENUM
- CHAR
- TEXT
- **VARCHAR**

3) What does a TIMESTAMP do on UPDATE CURRENT_TIMESTAMP data type?

TIMESTAMP column is updated with Zero when the table is created. UPDATE CURRENT_TIMESTAMP modifier updates the timestamp field to current time whenever there is a change in other fields of the table

4) How to enter Characters as HEX Numbers?

If you want to enter characters as HEX numbers, you can enter HEX numbers withsingle quotes and a prefix of (X), or just prefix HEX numbers with (Ox).

A HEX number string will be automatically converted into a character string, if the expression context is a string.

5) What is the different between NOW() and CURRENT_DATE()?

NOW () command is used to show current year,month,date with hours,minutes and seconds.

SYNTAX: SELECT NOW();

Gives—->2022-09-27 06:44:10

CURRENT_DATE() shows current year,month and date only.

Syntax: SELECT CURRENT_DATE();

Gives—>2022-09-27

6) What are all the Common SQL Function?

CONCAT(A, B) – Concatenates two string values to create a single string output. Often used to combine two or more fields into one single field.

FORMAT(X, D) – Formats the number X to D significant digits.

CURRDATE(), CURRTIME() – Returns the current date or time.

NOW() – Returns the current date and time as one value.

MONTH(), DAY(), YEAR(), WEEK(), WEEKDAY() – Extracts the given data from a date value.

HOUR(), MINUTE(), SECOND() – Extracts the given data from a time value.

DATEDIFF(A, B) – Determines the difference between two dates and it is commonly used to calculate age

SUBTIMES(A, B) – Determines the difference between two times.

FROMDAYS(INT) – Converts an integer number of days into a date value.

7)Write a query to calculate the age in year

```
SELECT YEAR(CURRENT_TIMESTAMP) -
      YEAR("1967-06-08") -
        (RIGHT(CURRENT_TIMESTAMP, 5) <
           RIGHT("1967-06-08", 5)) as age;
```

8)Write a Query to combine the strings , “Skill” and “Lync”

```
SELECT CONCAT('Skill', 'Lync') AS ConcatenatedString;
```

9)What is the difference between LOCATE() & INSTR()

The LOCATE() function returns the position of the first occurrence of a substring in a string. The INSTR() function returns the position of the first occurrence of a string in another string

For example:

```
SELECT LOCATE("H", "PHP") AS MatchPosition;` -- -> returns 2

SELECT INSTR("PHP", "H") AS MatchPosition; -- -> returns 2
```

10)what is convert() function ?

The CONVERT() function converts a value into the specified datatype or character set.

For example: SELECT CONVERT("2022-08-01", DATE); returns 2022-08-01

11)What is the output of the following query

```
SELECT DATE_FORMAT("2001-01-01", "%W %M %e %Y");
```

Monday January 01 2001

12)Write a query to get current date & time including seconds

SELECT CURRENT_TIMESTAMP();

13)Write query to fetch day of 2017-06-15.

2017SELECT DAYNAME("-06-15"); →returns Thursday

14)what is the output of following query

```
SELECT
   TODATETIMEOFFSET (
      '2019-03-06 07:43:58',
      '-08:00'
   );
```

2019-03-06 07:43:58.0000000 -08:00

15)Which function is used to get system's current date and time with timezone

SYSTEMDATETIMEOFFSET()

16) Which function is used to find the length of a word?

CHAR_LENGTH(): function is used to find the length of a word.

Syntax: SELECT char_length('Hello!');
Output: 6

17) What is the use case of CONCAT() function?

This function is used to add two words or strings.
Syntax: SELECT 'Goa' || ' ' || 'is having beaches' FROM dual;
Output: ‘Goais having beaches’

18) State the difference between INSERT() and INSTR() functions?

INSERT(): This function is used to insert the data into a database.
INSTR(): This function is used to find the occurrence of an alphabet.

19) How to make a string to upper case?

UCASE(): This function is used to make the string in upper case.
Syntax: UCASE ("car");
Output:
CAR

20)How to typecast a numeric or date input to character type?

TO_CHAR Function :
TO_CHAR function is used to typecast a numeric or date input to character type with a format model

21) How to Convert a character string to a date format using the TO_DATE function?

TO_DATE(char[, ’format_model’])

```
EXAMPLE :
SELECT last_name, hire_date
FROM employees
WHERE hire_date = TO_DATE(’May 24, 1999’, ’fxMonth DD, YYYY’);
```

22) Which function can be used to Subtracts a specified time interval from a date?

DATE_SUB()

23) How can we obtain the current date and time?

GETDATE()
Returns the current date and time

24) What is the use case of DATEPART() function?

Returns a single part of a date/time

25) Give the function to find the difference between two dates?

DATEDIFF()

WEEK 5 : Comparison,Arithmetic Operators,Groip by,Order by and Aggregate Functions

Q1.Create a following students_details table using sql querry

ROLL_NO	NAME	ADDRESS	AGE
1	HARSH	BANGALORE	22
2	ARWIND	BANGALORE	23
3	RAHUL	MUMBAI	22
4	KARAN	PUNE	24

```
Create Table students_details( ROLL_NO INTEGER  NOT NULL , NAME VARCHAR ( 30 ) , ADDRESS VARCHAR ( 50 ) , AGE SMALLINT );

INSERT INTO students_details(ROLL_NO, NAME, ADDRESS, AGE)

VALUES ( 1, 'HARSH','BANGALORE' , 22 ),( 2, 'ARWIN','BANGALORE' , 23 ),

( 3, 'RAHUL', 'MUMBAI' , 22 ),

( 4, 'KARAN','PUNE', 24 );

SELECT * FROM students_details
```

Q2.. From given table print the all values where AGE is greater than 22

```
SELECT * FROM student_details

WHERE AGE > 22
```

Q3.. Select NAMES column in students_details whose AGE is 23

```
SELECT NAME FROM students_details
```

WHERE AGE = 23

Q4.. Explain any four logical operators

All --- TRUE if all of the subquery values meet the condition

AND ---- TRUE if all the conditions separated by AND is TRUE

ANY ---- TRUE if any of the subquery values meet the condition

BETWEEN ---- TRUE if the operand is within the range of comparisons

EXISTS ---- TRUE if the subquery returns one or more records

IN ----- TRUE if the operand is equal to one of a list of expressions

LIKE ---- TRUE if the operand matches a pattern

Q5. Write a querry to print all values from students_details table where ADDRESS starts from B.

SELECT * FROM students_details

WHERE ADDRESS LIKE "B%"

Q6.. Write a querry to NAMES from students_details table where ADDRESS starts with B and AGE is greater than 22

SELECT NAME FROM students_details

WHERE ADDRESS LIKE "B%" AND AGE > 22

Q7.. In the given table above which ROLL_NO has AGE more than 23.

SELECT ROLL_NO FROM students_details

WHERE AGE > 23

Q8. In students_details table check the values who have ROLL_NO is 4 and ADDRESS is BANGALORE

SELECT * FROM students_details

WHERE ROLL_NO = 4 AND ADDRESS = BANGALORE

No such kind of value doesn't exists in the table

Q9.. Give the main difference between Group by and Order by

The Group By clause is used to group data based on the same value in a specific column.

The ORDER BY clause, on the other hand, sorts the result and shows it in ascending or descending order.

Q10.. Make the table in alphabetical order with respect to NAME column

SELECT * FROM students_details

ORDER BY NAME

Q11.. Combine the ROLL_NO and NAME column using union

SELECT ROLL_NO FROM students_details

UNION

SELECT NAME FROM students_details

Q12.. Using sub querry delete the rows whose AGE is greater than 23 in students_details table

DELETE FROM students_details

WHERE AGE IN (SELECT AGE FROM students_details WHERE AGE > 23)

Q13 What is Aggregate Function in SQL and list down the type of Aggregate Functions

An aggregate function in SQL returns one value after calculating multiple values of a column. We often use aggregate functions with the GROUP BY and HAVING clauses of the SELECT statement.
Various types of SQL aggregate functions are:

- Count()
- Sum()
- Avg()
- Min()
- Max()

Q14.. Write a query to check count of NAME in table

```
SELECT COUNT( NAME )
FROM students_details
```

Q15. Write a querry to do sum of all values present in AGE column which are equal to 22

```
SELECT SUM( AGE )
FROM students_details
WHERE AGE = 22
```

Q16.. What is the use case of COUNT() function?

The COUNT() function returns the number of rows in a database table.

Syntax: COUNT(*)

17) How can we find the lowest value (minimum) in a set of non-NULL values?

MIN() Function

The MIN() aggregate function returns the lowest value

18) What do you understand by 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.

9) Give some examples for logistic operators in SQL?

ALL	TRUE if all of the subquery values meet the condition
AND	TRUE if all the conditions separated by AND is TRUE
ANY	TRUE if any of the subquery values meet the condition
BETWEEN	TRUE if the operand is within the range of comparisons
EXISTS	TRUE if the subquery returns one or more records
IN	TRUE if the operand is equal to one of a list of expressions
LIKE	TRUE if the operand matches a pattern
NOT	Displays a record if the condition(s) is NOT TRUE

OR	TRUE if any of the conditions separated by OR is TRUE
SOME	TRUE if any of the subquery values meet the condition

Q20. How to sort the result-set in ascending or descending order?

The ORDER BY keyword is used to sort the result-set in ascending or descending order.The ORDER BY keyword sorts the records in ascending order by default. To sort the records in descending order, use the DESC keyword.

Q21.Write a SQL statement selects all customers from the "Customers" table, sorted

DESCENDING by the "Country" column?

```
SELECT * FROM Customers

ORDER BY Country DESC;
```

Q22.What is the use case of Group By multiple columns?

Group by multiple column is say for example, GROUP BY column1, column2. This means to place all the rows with same values of both the columns column1 and column2 in one group.

Consider below query,

```
SELECT SUBJECT, YEAR, Count(*)

FROM Student

GROUP BY SUBJECT, YEAR;
```

Q23.What do you understand by HAVING clause?

We can use HAVING clause to place conditions to decide which group will be the part of final result-set. Also we can not use the aggregate functions like SUM(), COUNT() etc. with WHERE clause. So we have to use HAVING clause if we want to use any of these functions in the conditions.

Q24.What is the use case of SOME operator?

TRUE if some of a set of comparisons are TRUE.

Q25.What do you mean by BETWEEN operator?

TRUE if the operand is within a range.

GROUP BY SUBJECT, YEAR;

WEEK 6: Joins

1)What is a join in SQL? What are the types of joins?

An SQL Join statement is used to combine data or rows from two or more tables based on a common field between them. Different types of Joins are:

• INNER JOIN
• LEFT JOIN
• RIGHT JOIN
• FULL JOIN

2)What is cross join?

Cross join is a Cartesian product where number of rows in the first table multiplied by number of rows in the second table.

3)What is the difference between JOIN and UNION?

JOIN

JOIN in SQL is used to combine data from many tables based on a matched condition between them. The data combined using JOIN statement results into new columns.

UNION

UNION in SQL is used to combine the result-set of two or more SELECT statements. The data combined using UNION statement results into new distinct rows.

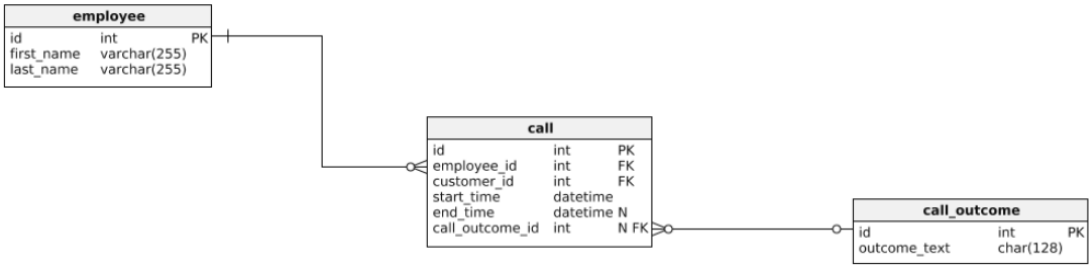
4)How to Join 3 Tables in MYSQL Server ?

```
SELECT *
FROM <table 1 name>
INNER JOIN <table 2 name>
ON <table1.field> = <table2.field>
INNER JOIN <table 3 name>
ON <table2.field> = <table3.field>;
```

5) How to Remove products that have not sold using JOIN.

```
DELETE FROM table-name1
WHERE condition
```

6) How to Join multiple tables using INNER JOIN



The query that does the job is given below:

```
SELECT E.first_name,E.last_name,C.start_time,C.end_time, O.outcome_text
FROM employee AS E
INNER JOIN call AS C ON C.employee_id = E.id
INNER JOIN call_outcome AS O ON C.call_outcome_id = O.id
ORDER BY C.start_time ASC;
```

Output is below:

Results | Messages

	first_name	last_name	start_time	end_time	outcome_text
1	Thomas (Neo)	Anderson	2020-01-11 09:00:15.000	2020-01-11 09:12:22.000	finished - successfully
2	Agent	Smith	2020-01-11 09:02:20.000	2020-01-11 09:18:05.000	finished - unsuccessfully
3	Thomas (Neo)	Anderson	2020-01-11 09:14:50.000	2020-01-11 09:20:01.000	finished - successfully
4	Thomas (Neo)	Anderson	2020-01-11 09:24:15.000	2020-01-11 09:25:05.000	finished - unsuccessfully
5	Thomas (Neo)	Anderson	2020-01-11 09:26:23.000	2020-01-11 09:33:45.000	finished - successfully
6	Thomas (Neo)	Anderson	2020-01-11 09:40:31.000	2020-01-11 09:42:32.000	finished - successfully
7	Agent	Smith	2020-01-11 09:41:17.000	2020-01-11 09:45:21.000	finished - successfully
8	Thomas (Neo)	Anderson	2020-01-11 09:42:32.000	2020-01-11 09:46:53.000	finished - unsuccessfully
9	Agent	Smith	2020-01-11 09:46:00.000	2020-01-11 09:48:02.000	finished - successfully
10	Agent	Smith	2020-01-11 09:50:12.000	2020-01-11 09:55:35.000	finished - successfully

7)What is syntax of LEFT OUTER JOIN syntax ?

```
SELECT column-names
FROM table-name1
LEFT OUTER JOIN table-name2 ON column-name1 = column-name2
WHERE condition
```

8) What is syntax of RIGHT OUTER JOIN syntax ?

```
SELECT column-names
FROM table-name1
RIGHT OUTER JOIN table-name2 ON column-name1 = column-name2
WHERE condition
```

9) What is syntax of SELF JOIN syntax ?

```
SELECT a.col1, b.col2,..., a.coln
FROM table1 a, table1 b
```

```
WHERE a.commonfield = b.commonfield;
```

10) What is syntax of SQL UPDATE JOIN ?

UPDATE tablename
INNER JOIN tablename
ON tablename.columnname = tablename.columnname
SET tablenmae.columnnmae = tablenmae.columnname;

11) What are SQL multiple joins?

Multiple joins can be described as follows; multiple join is a query that contains the same or different join types, which are used more than once. Thus, we gain the ability to combine multiple tables of data in order to overcome relational database issues.

12) Which is a query that retrieves rows from more than one table or view

Join

13)Which join refers to join records from the right table that have no matching key in the left table are include in the result set:

Right outer join

14)Relation R1 has 10 tuples and 5 attributes. Relation R2 has 0 tuples and 7 attributes. When a CROSS JOIN is achieved between R1 and R2, how many tuples would the resultant set have?

Zero

15) What is Normalization in SQL?

Normalization is used to decompose a larger, complex table into simple and smaller ones. This helps us in removing all the redundant data.Generally, in a table, we will have a lot of redundant information which is not required, so it is better to divide this complex table into multiple smaller tables which contain only unique information.
First normal form:A relation schema is in 1NF, if and only if:

- All attributes in the relation are atomic(indivisible value)
- And there are no repeating elements or groups of elements.

Second normal form:A relation is said to be in 2NF, if and only if:

- It is in 1st Normal Form.
- No partial dependency exists between non-key attributes and key attributes.

Third Normal form:A relation R is said to be in 3NF if and only if:

- It is in 2NF.
- No transitive dependency exists between non-key attributes and key attributes through another non-key attribute

16) What do you mean by cartesian join?

The CARTESIAN JOIN is also known as CROSS JOIN. In a CARTESIAN JOIN there is a join for each row of one table to every row of another table. This usually happens when the matching column or WHERE condition is not specified.

17) What do you know about Self Join?

SELF JOIN: As the name signifies, in SELF JOIN a table is joined to itself. That is, each row of the table is joined with itself and all other rows depending on some conditions. In other words we can say that it is a join between two copies of the same table

18) what is the use case of Full Join?

FULL JOIN creates the result – set by combining results of both LEFT JOIN and RIGHT JOIN. The result- set will contain all the rows from both tables.
For the rows for which there is no matching, the result will contain NULL values.

19) What do you know about left and right join?

• LEFT JOIN: This join returns all the rows of the table on the left side of the join and matching rows for the table on the right side of the join. For the rows for which there is no matching row on the right side, the result-set will be null. LEFT JOIN is also known as LEFT OUTER JOIN.
• RIGHT JOIN: RIGHT JOIN is similar to LEFT JOIN. This join returns all the rows of the table on the right side of the join and matching rows for the table on the left side of the join. For the rows for which there is no matching row on the left side, the result-set will contain null. RIGHT JOIN is also known as RIGHT OUTER JOIN.

20) What do you know about FULL JOIN?

FULL JOIN: FULL JOIN creates the result-set by combining results of both LEFT JOIN and RIGHT JOIN. The result-set will contain all the rows from both tables. For the rows for which there is no matching, the result-set will contain NULL values.

21) What is the use case of UPDATE JOIN?

SQL UPDATE JOIN could be used to update one table using another table and join condition.

22. What is the importance of SQL joins in database management?

SQL joins are important in database management for the following reasons:

- A method of stitching a database back together to make it easier to read and use.
- Additionally, they maintain a normalized database. Data normalization helps us keep data redundancy low so that when we delete or update a record, we will have fewer data anomalies in our application.
- Joins have the advantage of being faster, and as a result, are more efficient.

23) Explain merge join in SQL?

Merge join produces a single output stream resulting from the joining of two sorted datasets using an INNER, FULL, or LEFT join. It is the most effective of all the operators for joining data. Specifically, merge join requires that both inputs be sorted as well as matching meta-data in the joined columns. Users can't join columns of different data types together. Users are not permitted to combine a column with a numeric data type with a column with a character data type.

24) What is a hash join in SQL?

Just like any other join, the hash join requires two inputs, which are the probe input (inner table) and the build input (outer table). A hash join involves the use of a hash table to identify rows matching between two tables. The hash join is the option when no other join is preferred (possibly due to the absence of sorting or indexing etc). Hash joins are best when joining large data sets that are unsorted and non-indexed.

Q25) Can you explain nested join in SQL?

A JOIN is one of the mechanisms that we use to combine the data of more than one table in a relational database, and a Nested Join is one of the simplest methods involving the physical joining of two tables. In essence, a Nested Join uses one joining table as an outer input table while the other one serves as an inner input table. With a Nested Loop Join, one row from the outer table is retrieved and then the row is searched for in the inner table; this process is repeated until all the output rows from the outer table have been searched for in the inner table. Nested Loop Join may further be sub-categorized into Indexed Nested, Naive Nested and Temporary Index Nested Loop Join.

WEEK 7 : Data Exploration

Q1- What is Data Exploration?

Data exploration is the process of identifying what is actually stored in the database. Data exploration can be counted as the initial phase of data analysis. Data Analysis is finding the meaningful and required information from the database. Data is obtained from sources like customers, internet resources, web browsing history, telephone calls, purchase record, entertainment platforms etc.

The user explores a large dataset to evaluate the type of data, to get a broader picture of the stored values. Studying metadata (data about data) and documentation helps users to understand the type of data recorded about a particular object. Understanding the need of recorded data in perspective of origin of database and targeted goal (business approach). A few ways we can go about this by asking questions like, how many columns are there? What is stored in each column? Etc.

Q2 What is a Sub-Query?

Subquery is a query inside a query. Oftentimes large databases lead to more complexity, using simple SELECT statements will not work for the complex tasks. Sub-query enables users to perform complex tasks into partitions. Data from multiple tables can be fetched using sub-queries. Also errors related with unwanted row selection can be minimized by dividing complex tasks into multiple select statements. It enables easy data exploration and analysis processes, smart optimizing with sub-queries is easy to understand. The database stores data but data can be retrieved with querying and efficient querying can be achieved through sub-queries.

Q3. Give examples of a few in-built functions?

- Mathematical Functions
  - PI ()
  - RAND ()
  - CEIL ()
- String Functions
  - CONCAT ()
  - LOWER ()

    - UPPER ()
- Date Functions
    - CURRENT_DATE ()
    - DATEDIFF ()
    - DAYOFMONTH ()
- Conversion Functions
    - CONVERT ()
- Aggregate Functions
    - AVG ()
    - COUNT ()

Q4 What is the significance of in-built functions?

Ans4. Functions like COUNT (), AVG (), MIN (), MAX (), LOWER (), UPPER (), CURRENT_DATE (), CONVERT () etc. are used very frequently by data analysts for mining useful information from the stored data. It is not necessary that all studied functions will be used by a data analyst or miner. Usage of inbuilt functions depend on the type of project or database. Unformatted data from web sources may require use of multiple string functions. Finding initial trends in the database can be achieved by aggregate functions. Effect of minor data computation can be realized with mathematical functions. Data can come from multiple sources and these different sources can follow different table structures. There is a different format for storing date, a different format for storing phone numbers. These inbuilt functions are very much useful for making uniformity in the data. Analysis can be performed with the same type of data. Functions like STDDEV () can help to detect absurd values in the stored data. Data recorded from scientific experiments require execution of complex mathematical functions. Computed results can be exported as .csv or in spreadsheets. The result set can be further presented in the form of Graphs, Charts using different tools like excel, calc, python etc. In-built functions make SQL tool more powerful as there is no requirement of additional tools for data computation tasks. Data formatting to data computation can be performed through these sets of instructions.

Q5. What are other useful capabilities of SQL for Data Exploration?

Ans5. Some capabilities of SQL for Data Exploration Are Views, Stored Procedures, Data Analyst or miner can get database from different sources that has been stored in different tool or platform. He/she is free to use any SQL tool or other tool for exploring the database. Statements like CREATE, INSERT, DELETE, UPDATE Comes into picture in such cases. Creation of temporary tables with the relevant attributes is the part of initial phase of data analysis.

Q6 What is the significance of joins?

Ans6. Joins is used in Joining multiple tables in the database. It is the most powerful tool of SQL. Complex algorithms are running behind the SQL interface to generate the result set. While dealing with large dataset, you cannot keep eye on each row. Here are few common mistakes-losing rows in the resultant set, having extra rows in the resultant set. The type of Join matters, condition given in Join definition matters.

Here are the different types of the JOINs in SQL:

(INNER) JOIN: Returns records that have matching values in both tables

LEFT (OUTER) JOIN: Returns all records from the left table, and the matched records from the right table

RIGHT (OUTER) JOIN: Returns all records from the right table, and the matched records from the left table.

FULL (OUTER) JOIN: Returns all records when there is a match in either left or right table.

Q7. Given the following table. Query the PRODUCT table to display OrderID, ProductID, Unitprice, Quantity and CostPrice. CostPrice is calculated as summation of Unitprice and UnitOnOrder and their multiplication with UnitPrice. UnitsOnOrder is an optional column and may contain NULL VALUES?

FIELDS	TYPE
OrderID	INTEGER
ProductID	INTEGER
Unitprice	REAL
Quantity	INT
UnitsonOrder	INT

```
SELECT ProductID, UnitPrice * ( UnitsInStock + IFNULL( UnitsOnOrder , 0))
FROM PRODUCT;
```

Or

```
SELECT ProductID, UnitPrice * ( UnitsInStock + COALESCE( UnitsOnOrder , 0))
FROM PRODUCT;
```

Q8. What is an Entity-Relationship Model?

Ans8 .ER-Diagram is a pictorial representation of data that describes how data is communicated and related to each other. Any object, such as entities, attributes of an entity, sets of relationship, and other attributes of relationship, can be characterized with the help of the ER diagram. ER modeling is a top-down structure to database design that begins with identifying the important data called entities and relationships in combination with the data that must be characterized in the model. Then database model designers can add more details such as the information they want to hold about the entities and relationships, which are the attributes and any constraints on the entities, relationships, and attributes. ER modeling is an important technique for any database designer to master and forms the basis of the methodology.

Q9 What Are Entities?

Entities are represented using the rectangle-shaped box. These rectangles are named with the entity set they represent.

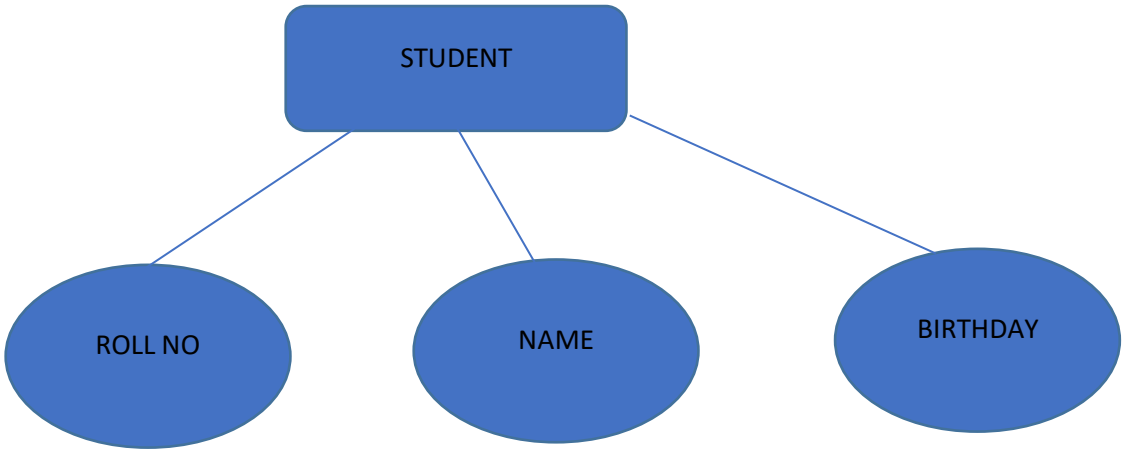
Entity type: It is a group of objects with the same properties that are identified by the enterprise as having an independent existence. The basic concept of the ER model is the entity type that is used to represent a group of 'objects' in the 'real world' with the same properties. An entity type has an independent existence within a database.

Entity occurrence: A uniquely identifiable object of an entity type.

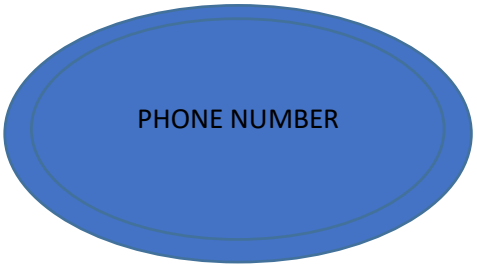
Each entity type is shown as a rectangle labeled with the name of the entity, which is usually a singular noun.

Q10 What Are Attributes?

Attributes are the properties of entities that are represented using ellipse-shaped figures. Every elliptical figure represents one attribute and is directly connected to its entity (which is represented as a rectangle).



Multi-valued attributes are represented using double ellipse like this:



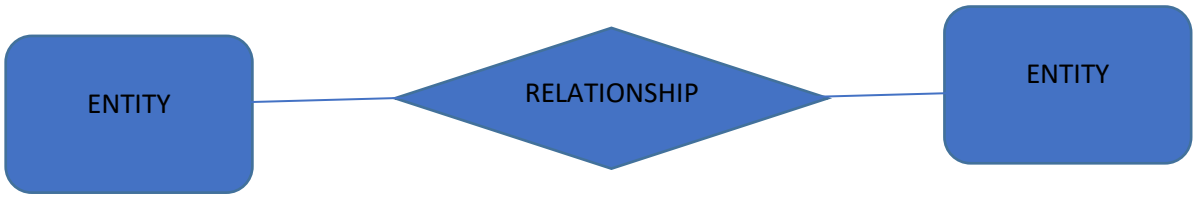
Q11. What are Relationships?

Ans11. A relationship is a set of associations between one or more participating entity types. Each relationship type is given a name that describes its function. A diamond-shaped box represents relationships. All the entities (rectangle-shaped) participating in a relationship get connected using a line.

There are four types of relationships. These are:

- One-to-one: When only a single instance of an entity is associated with the relationship, it is termed as '1:1'.

- One-to-many: When more than one instance of an entity is related and linked with a relationship, it is termed as '1:N'.
- Many-to-one: When more than one instance of an entity is linked with the relationship, it is termed as 'N:1'.
- Many-to-many: When more than one instance of an entity on the left and more than one instance of an entity on the right can be linked with the relationship, then it is termed as N:N relationship.



Q12 . From the following tables, write a SQL query to find all the orders issued by the salesman 'Ronnie Pickering'. Return ord_no, purch_amt, ord_date, customer_id and salesman_id.

ORDER TABLE

Ord_no	Purch_amt	Ord_date	Customer_id	Salesman_id
23400	1500.23	2011-10-01	2399	1056
23401	3000.33	2011-10-02	2266	1097
23402	6000.34	2011-10-03	2199	1145
23403	4523.23	2011-10-04	2456	1243

SALESMAN

NAME	CITY	ID	COMMISSION	AGE
JAMES DOE	NEW YORK	2199	0.15	34

RONNIE PICKERING	LONDON	2456	0.12	40
PAUL ADAM	SAN JOSE	2399	0.17	32

SELECT * FROM ORDERS

WHERE Salesman_id = (SELECT ID FROM SALESMAN

WHERE NAME = 'RONNIE PICKERING');

Q13 Given Salesman table find details of the salesman who is youngest?

Sol13 SELECT * FROM SALESMAN

WHERE AGE = (SELECT MIN(AGE)

FROM SALESMAN)

Q14. Given a field containing name of the salesman. Extract the first 5 letters of the name.

Sol14 SELECT LEFT(NAME, 5) AS FIRST_NAME

FROM SALESMAN

Q15 Given a field containing name of the salesman. Extract the first and last letters of the name for the nickname.

SELECT LEFT(NAME, 1) + RIGHT(NAME, 1) AS nickname

From SALESMAN

Q16.What is the disadvantage of GROUP BY statement?

It will not provide us with he the all records in the table . it is used mainly when we want to perform aggregate operations based on specific group and it returns a single value for multiple rows

Example: 1) Counting number of employees by department.
2) Number of sales done by each sales man etc.

Ex query:

1. SELECT department, COUNT(*) from employees
   GROUP BY department;

2. SELECT deptname, MAX(salary) from employees
   GROUP BY deptname ;

	dept_name	max(salary)
▶	Admin	5000
	HR	8000
	IT	11000
	Finance	6500

Above we are just seeing four distinct rows based on which we grouped the data.It will provide us with only those limited row/records, if we want all the records with specific aggregation operation then we go for Window functions/ Analytic functions.

Q17.What are these window functions?

In SQL, a window function or analytic function is a function which uses values from one or multiple rows to return a value for each row. Here, with this we can be able display all the records with the specific aggregated value (it can be count/sum/max/min/stddev/variance)

Q18.What are the different types of Window functions ?

1. ROW_NUMBER()
2. RANK()
3. DENSE_RANK()
4. LEAD()
5. LAG() etc.

To make a window in the table first we use over( ) function along with the aggregate function.

Ex:
```
SELECT *,                          # * selects all the rows
MAX(salary)  OVER( ) as max_sal    # it creates a new column max_sal
FROM employee  ;
```

	emp_ID	emp_NAME	DEPT_NAME	SALARY	max_salary
▶	101	Mohan	Admin	4000	11000
	102	Rajkumar	HR	3000	11000
	103	Akbar	IT	4000	11000
	104	Dorvin	Finance	6500	11000
	105	Rohit	HR	3000	11000
	106	Rajesh	Finance	5000	11000
	107	Preet	HR	7000	11000
	108	Maryam	Admin	4000	11000
	109	Sanjay	IT	6500	11000
	109	Sanjay	IT	6500	11000
	110	Vasudha	IT	7000	11000
	111	Melinda	IT	8000	11000

Above we got all the records(only few are shown) along with aggregated value maximum salary. And no grouping is done .if we want we can do that by using partition by command inside over( ) method which acts just like the group by command.

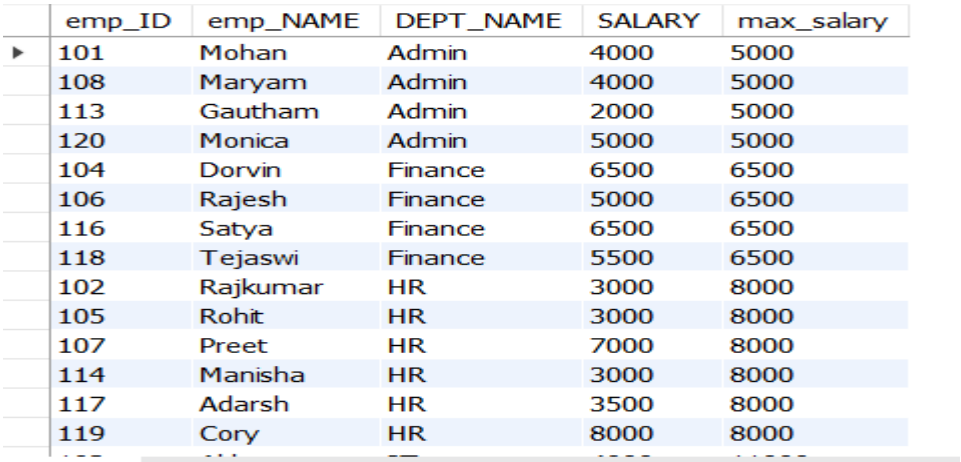
Q19.How to group the data and keep it in the order using window functions?

We use PARTITION BY command inside the over( ) method ,now it will create many windows based on partition.

Ex: select *,

max(salary) over(partition by dept_name) as max_sal
from employee;

;output:



emp_ID	emp_NAME	DEPT_NAME	SALARY	max_salary
101	Mohan	Admin	4000	5000
108	Maryam	Admin	4000	5000
113	Gautham	Admin	2000	5000
120	Monica	Admin	5000	5000
104	Dorvin	Finance	6500	6500
106	Rajesh	Finance	5000	6500
116	Satya	Finance	6500	6500
118	Tejaswi	Finance	5500	6500
102	Rajkumar	HR	3000	8000
105	Rohit	HR	3000	8000
107	Preet	HR	7000	8000
114	Manisha	HR	3000	8000
117	Adarsh	HR	3500	8000
119	Cory	HR	8000	8000

Above we can observe , the grouped data based on department name.

We can keep it in the order of salaries they get. Use ORDER BY inside over( ) method.

Ex: select *,

max(salary) over(partition by dept_name) as max_sal

from employee ;

Q20.How to use ROW_NUMBER( ) method works ?

It gives numbers to the each record.

Assigning row number to each record without partition by.

Ex: select *,

row_number( ) over( ) as rowNumber

from employee ;

emp_ID	emp_NAME	DEPT_NAME	SALARY	rowNumber
101	Mohan	Admin	4000	1
102	Rajkumar	HR	3000	2
103	Akbar	IT	4000	3
104	Dorvin	Finance	6500	4
105	Rohit	HR	3000	5
106	Rajesh	Finance	5000	6
107	Preet	HR	7000	7
108	Maryam	Admin	4000	8
109	Sanjay	IT	6500	9
109	Sanjay	IT	6500	10
110	Vasudha	IT	7000	11
111	Melinda	IT	8000	12
112	Komal	IT	10000	13
113	Gautham	Admin	2000	14
114	Manisha	HR	3000	15

Above we can see numbering to each row without any partion by department.

Now we will see the row_number with partition by

Ex: select *,

row_number( ) over(partition by dept_name order by salary) as rowNumber

from employee ;

emp_ID	emp_NAME	DEPT_NAME	SALARY	rowNumber
120	Monica	Admin	5000	1
101	Mohan	Admin	4000	2
108	Maryam	Admin	4000	3
113	Gautham	Admin	2000	4
104	Dorvin	Finance	6500	1
116	Satya	Finance	6500	2
118	Tejaswi	Finance	5500	3
106	Rajesh	Finance	5000	4
119	Cory	HR	8000	1
107	Preet	HR	7000	2
117	Adarsh	HR	3500	3
102	Rajkumar	HR	3000	4
105	Rohit	HR	3000	5
114	Manisha	HR	3000	6

Above we can numbering is based on departments , for each department count starts from number 1. I hope you got the difference between the row_number with partition by and without partition by.

Q21.Application of row_number is , if you want to get the details top 2 or 3 employees to join the company we can use it along with subquery(Query inside a query)

Ex: select * from (

subquery

```
select e.*,
 row_number( ) over(partition by dept_name order by emp_id) as rn
 from employee e)  x              # x is alias to subquery.
 where x.rn < 3;
```

emp_ID	emp_NAME	DEPT_NAME	SALARY	rn
101	Mohan	Admin	4000	1
108	Maryam	Admin	4000	2
104	Dorvin	Finance	6500	1
106	Rajesh	Finance	5000	2
102	Rajkumar	HR	3000	1
105	Rohit	HR	3000	2
103	Akbar	IT	4000	1
109	Sanjay	IT	6500	2

Above , WHERE x.rn < 3 means it will fetch first 2 employees who joined in the company.

Q22.Difference between RANK( ) and DENSE_RANK()?

Ex:

```
select *,
 rank() over(partition by dept_name order by salary desc) as rnk
 from employee ;
```

emp_ID	emp_NAME	DEPT_NAME	SALARY	rnk
120	Monica	Admin	5000	1
101	Mohan	Admin	4000	2
108	Maryam	Admin	4000	2
104	Dorvin	Finance	6500	1
116	Satya	Finance	6500	1
118	Tejaswi	Finance	5500	3
119	Cory	HR	8000	1
107	Preet	HR	7000	2
117	Adarsh	HR	3500	3
124	Dheeraj	IT	11000	1
112	Komal	IT	10000	2
111	Melinda	IT	8000	3
122	Ibrahim	IT	8000	3
123	Vikram	IT	8000	3

Above we can see ranking based on salary, two persons with equal salary got same rank, just like how two persons with the same marks get same rank in college.

Now we will see how DENSE_RANK works.

Ex:

```
select *,
dense_rank() over(partition by dept_name order by salary desc) as dense_rnk
from employee ;
```

emp_ID	emp_NAME	DEPT_NAME	SALARY	dense_rnk
120	Monica	Admin	5000	1
101	Mohan	Admin	4000	2
108	Maryam	Admin	4000	2
113	Gautham	Admin	2000	3
104	Dorvin	Finance	6500	1
116	Satya	Finance	6500	1
118	Tejaswi	Finance	5500	2
106	Rajesh	Finance	5000	3
119	Cory	HR	8000	1
107	Preet	HR	7000	2
117	Adarsh	HR	3500	3
102	Rajkumar	HR	3000	4
105	Rohit	HR	3000	4
114	Manisha	HR	3000	4
124	Dheeraj	IT	11000	1

Above if we observe it is a bit different from RANK( ) result. May be it is little bit confusing , but if we observe clearly we can get the difference. It (DESNSE_RANK()) is used to answer the questions like what is 1st highest salary, 2nd highest salary or 3rd highest salary etc.

You can observe the difference with the following examples.

Ex: display second highest salaried persons from all departments?

```
select * from(
select e.*,
dense_rank() over(partition by dept_name order by salary desc) as dense_rnk
 from employee e) as x
 where x.dense_rnk = 2;    # it selects only second salaried persons.
```

emp_ID	emp_NAME	DEPT_NAME	SALARY	dense_rnk
101	Mohan	Admin	4000	2
108	Maryam	Admin	4000	2
118	Tejaswi	Finance	5500	2
107	Preet	HR	7000	2
112	Komal	IT	10000	2

Similarly RANK( ) function answers the questions like top 2 or 3 employees in each department earning the max salary.

Ex: display top 3 employees in each department earning the max salary.

```
select * from (
 select e.*,
 rank() over(partition by dept_name order by salary desc) as rnk
 from employee e) x
 where rnk < 4;
```

emp_ID	emp_NAME	DEPT_NAME	SALARY	rnk
120	Monica	Admin	5000	1
101	Mohan	Admin	4000	2
108	Maryam	Admin	4000	2
104	Dorvin	Finance	6500	1
116	Satya	Finance	6500	1
118	Tejaswi	Finance	5500	3
119	Cory	HR	8000	1
107	Preet	HR	7000	2
117	Adarsh	HR	3500	3
124	Dheeraj	IT	11000	1
112	Komal	IT	10000	2
111	Melinda	IT	8000	3
122	Ibrahim	IT	8000	3
123	Vikram	IT	8000	3

Result 4

Q23.How LAG( ) function is used ?

It is used to fetch previous employee's details like salary , emp_id anything that we mention in the LAG( ) function.

Ex:
```
select e.* ,
lag(salary) over(partition by dept_name order by emp_id) as prev_sal
from employee e;
```

emp_ID	emp_NAME	DEPT_NAME	SALARY	prev_sal
101	Mohan	Admin	4000	NULL
108	Maryam	Admin	4000	4000
113	Gautham	Admin	2000	4000
120	Monica	Admin	5000	2000
104	Dorvin	Finance	6500	NULL
106	Rajesh	Finance	5000	6500
116	Satya	Finance	6500	5000
118	Tejaswi	Finance	5500	6500
102	Rajkumar	HR	3000	NULL
105	Rohit	HR	3000	3000
107	Preet	HR	7000	3000
114	Manisha	HR	3000	7000
117	Adarsh	HR	3500	3000

Above we can see previous employee's salary and NULL represents no employee previous to him/her. If we want employee name place emp_NAME inside LAG().

Q24.How LEAD( ) function is used ?

It is used to fetch next employee's details like salary , emp_id anything that we mention in the LEAD( ) function. It is just similar to the LAG().

Ex: select e.* ,
lead(salary) over(partition by dept_name order by emp_id) as next_sal
from employee e;

emp_ID	emp_NAME	DEPT_NAME	SALARY	next_sal
101	Mohan	Admin	4000	4000
108	Maryam	Admin	4000	2000
113	Gautham	Admin	2000	5000
120	Monica	Admin	5000	NULL
104	Dorvin	Finance	6500	5000
106	Rajesh	Finance	5000	6500
116	Satya	Finance	6500	5500
118	Tejaswi	Finance	5500	NULL
102	Rajkumar	HR	3000	3000
105	Rohit	HR	3000	7000
107	Preet	HR	7000	3000
114	Manisha	HR	3000	3500
117	Adarsh	HR	3500	8000

NULL represents there is no employee after him in the department.

Practice Problems:

25 .Wrtie a query to display running sum of salaries by each department?

select e.*,

sum(salary) over(partition by dept_name order by emp_id) running_sum

from employee e;

	emp_ID	emp_NAME	DEPT_NAME	SALARY	running_sum
▶	101	Mohan	Admin	4000	4000
	108	Maryam	Admin	4000	8000
	113	Gautham	Admin	2000	10000
	120	Monica	Admin	5000	15000
	104	Dorvin	Finance	6500	6500
	106	Rajesh	Finance	5000	11500
	116	Satya	Finance	6500	18000
	118	Tejaswi	Finance	5500	23500
	102	Rajkumar	HR	3000	3000
	105	Rohit	HR	3000	6000
	107	Preet	HR	7000	13000
	114	Manisha	HR	3000	16000
	117	Adarsh	HR	3500	19500

Above if we observe total amount paid to Admin Department is 15000.

All functions together.

```
select e.*,
rank() over(partition by dept_name order by salary desc) as rnk,
dense_rank() over(partition by dept_name order by salary desc) as dense_rnk,
row_number() over(partition by dept_name order by salary desc) as rn
from employee e;
```

	emp_ID	emp_NAME	DEPT_NAME	SALARY	rnk	dense_rnk	rn
▶	120	Monica	Admin	5000	1	1	1
	101	Mohan	Admin	4000	2	2	2
	108	Maryam	Admin	4000	2	2	3
	113	Gautham	Admin	2000	4	3	4
	104	Dorvin	Finance	6500	1	1	1
	116	Satya	Finance	6500	1	1	2
	118	Tejaswi	Finance	5500	3	2	3
	106	Rajesh	Finance	5000	4	3	4
	119	Cory	HR	8000	1	1	1
	107	Preet	HR	7000	2	2	2
	117	Adarsh	HR	3500	3	3	3
	102	Rajkumar	HR	3000	4	4	4
	105	Rohit	HR	3000	4	4	5

LEAD() and LAG() together.

Query:

```
select e.*,
lag(salary) over(partition by dept_name order by emp_id) as prev_empl_sal,
```

lead(salary) over(partition by dept_name order by emp_id) as next_empl_sal

from employee e;

emp_ID	emp_NAME	DEPT_NAME	SALARY	prev_empl_sal	next_empl_sal
101	Mohan	Admin	4000	NULL	4000
108	Maryam	Admin	4000	4000	2000
113	Gautham	Admin	2000	4000	5000
120	Monica	Admin	5000	2000	NULL
104	Dorvin	Finance	6500	NULL	5000
106	Rajesh	Finance	5000	6500	6500
116	Satya	Finance	6500	5000	5500
118	Tejaswi	Finance	5500	6500	NULL
102	Rajkumar	HR	3000	NULL	3000
105	Rohit	HR	3000	3000	7000
107	Preet	HR	7000	3000	3000
114	Manisha	HR	3000	7000	3500
117	Adarsh	HR	3500	3000	8000

similary we can work on :

growth rate,

Distribution of salary slabs in a company,

data summary using UNION command.

WEEK 8: Index, Views, Transactions

1.What is indexing in MySQL?

Indexes are used to find rows with specific column values quickly.

2.What are the features of index?

Increases the execution speed.

Can be imposed on one or two columns.

Most frequently used columns are used for index creation.

Used by database search engine to locate records in the table.

SELECT statement becomes faster with index.

3.How many types of indexes ?

MySQL has three types of indexes: INDEX, UNIQUE (which requires each row to have a unique value), and PRIMARY KEY (which is just a particular UNIQUE index)

4.What is the syntax for drop index inMySQL?

ALTER TABLE *table_name*

DROP INDEX *index_name*;

5.What is the VIEW in Mysql?

The View is a virtual table created by a query by joining one or more tables.

6.What is syntax for creating view

CREATE VIEW view_name AS

SELECT column1,column2…..

FROM table_name

WHERE condition;

view_name:Name for the view

Table_name:Name for the table

condition:Condition to select rows

7.Differentiate between Single user DBMS & Multi user DBMS

Single user DBMS:

- ✓ At a particular instant,only single user can access the database
- ✓ Personal Usage

Multi user DBMS

- ✓ More than one user can access the database
- ✓ Client server architecture
- ✓ Distributed approach

8. What is transaction & write it's syntax ?

A transaction in MySQL is a sequential group of statements, queries, or operations such as select, insert, update or delete to perform as a one single work unit that can be committed or rolled back.

Syntax:

```
START TRANSACTION

SAVEPOINT savepoint_name

ROLLBACK TO savepoint_name

RELEASE SAVEPOINT savepoint_name

COMMIT
```

9. What is SAVEPOINT?

- It creates points within the groups of transactions in which to ROLLBACK. A SAVEPOINT is a point in a transaction in which you can roll the transaction back to a certain point without rolling back the entire transaction.
- This command is used only in the creation of SAVEPOINT among all the transactions. In general ROLLBACK is used to undo a group of transactions.

10. How RELEASE SAVEPOINT works ?

This command is used to remove a SAVEPOINT that you have created. Syntax:

```
RELEASE SAVEPOINT savepoint_name
```

Once a SAVEPOINT has been released, you can no longer use the ROLLBACK command to undo transactions performed since the last SAVEPOINT.

It is used to initiate a database transaction and used to specify characteristics of the transaction that follows.

11. Which command is used to save changes permanently in transaction?

COMMIT is used. If everything is in order with all statements within a single transaction, all changes are recorded together in the database is called committed. The COMMIT command saves all the transactions to the database since the last COMMIT or ROLLBACK command.

12. Difference between COMMIT and ROLLBACK

A COMMIT statement is used to save the changes on the current transaction is permanent. A Rollback statement is used to undo all the changes made on the current transaction. Rollback can be done before commit only and Once SQL Server commits a transaction, you cannot run the ROLLBACK statement.

13 Consider following Student Details table

S_ID	NAME	ADDRESS
1	RAM	HYDERABAD
2	SAI	BANGALORE
3	DEEPA	CHENNAI
4	LATHA	DELHI

CREATE A View named DetailsView ?

```
CREATE  VIEW DetailsView AS
SELECT NAME,ADDRESS
FROM StudentDetails
WHERE S_ID<5;
#check
SELECT * FROM DetailsView ;
```

14.How to delete views ?

We can delete or drop a View using DROP statement

15.Can we change the view after creating? If yes ,which statement is used ?

YES,we can change or replace a view by using CREATE OR REPLACE VIEW statement

16.Why it is not recommended to create indexes on small tables?

It takes the SQL Server Engine less time scanning the underlying table than traversing the index when searching for specific data. In this case, the index will not be used but it will still

negatively affect the performance of data modification operations, as it will be always adjusted when modifying the underlying table's data.

17.What is difference between INDEX & VIEW?

A view is just a way of abbreviating a subquery. An index is used to optimize matching column data

WEEK 9: Transforming Table Results

1.What are case statements?

The CASE expression goes through conditions and returns a value when the first condition is met (like an if-then-else statement). So, once a condition is true, it will stop reading and return the result. If no conditions are true, it returns the value in the ELSE clause. If there is no ELSE part and no conditions are true, it returns NULL.

Syntax: SELECT CASE

WHEN *condition1* THEN *result1*

WHEN *condition2* THEN *result2*

WHEN *conditionN* THEN *resultN*

ELSE *result*

END AS
Column_
name
FROM
table_na
me ;

2.Write a query to get the details of all employees who were hired between 1986 and 1989?

```
SELECT * FROM employees
WHERE YEAR(hire_date) BETWEEN 1986 AND 1989 ;
```

emp_no	birth_date	first_name	last_name	gender	hire_date
10001	1953-09-02	Georgi	Facello	M	1986-06-26
10003	1959-12-03	Parto	Bamford	M	1986-08-28
10004	1954-05-01	Chirstian	Koblick	M	1986-12-01
10005	1955-01-21	Kyoichi	Maliniak	M	1989-09-12
10006	1953-04-20	Anneke	Preusig	F	1989-06-02
10007	1957-05-23	Tzvetan	Zielinski	F	1989-02-10
10010	1963-06-01	Duangkaew	Piveteau	F	1989-08-24
10014	1956-02-12	Berni	Genin	M	1987-03-11
10015	1959-08-19	Guoxiang	Nooteboom	M	1987-07-02
10018	1954-06-19	Kazuhide	Peha	F	1987-04-03
10021	1960-02-20	Ramzi	Erde	M	1988-02-10

3. **Write a query to get the number of candidates hired in a year 1986 using CASE Expression.**

QUERY:

```
SELECT
SUM(CASE WHEN YEAR(hire_date)= 1986 THEN 1 ELSE 0 END AS)
FROM employees;
```

4. **Write a query to get year wise number of candidates hired using CASE Expression.**

```
SELECT
SUM(CASE WHEN YEAR(hire_date)= 1986 THEN 1 ELSE 0 END ) Total_hired_in_1986,
SUM(CASE WHEN YEAR(hire_date)= 1987 THEN 1 ELSE 0 END ) Total_hired_in_1987,
SUM(CASE WHEN YEAR(hire_date)= 1988 THEN 1 ELSE 0 END ) Total_hired_in_1988,
SUM(CASE WHEN YEAR(hire_date)= 1989 THEN 1 ELSE 0 END ) Total_hired_in_1989,
SUM(CASE WHEN YEAR(hire_date)= 1990 THEN 1 ELSE 0 END ) Total_hired_in_1990 FROM
employees;
```

	Total_hired_in_1986	Total_hired_in_1987	Total_hired_in_1988	Total_hired_in_1989	Total_hired_in_1990
▸	36150	33501	31436	28394	25610

5. **Write a query to Check the existence of a particular value in the columns ? For this we use EXISTS command**

QUERY:

```
SELECT EXISTS
(SELECT 1 from titles
WHERE title = "Senior Engineer") AS Result;
```

	Result
▸	1

It gives results = 1 if the title exists in the title column.

6. **Update the salaries of the employees based on their existed salary. If employees salary is below 40000 then add 10000 else add 5000. Query:**

```
UPDATE      salaries
SET salary = CASE
                    WHEN salary <40000 THEN salary +10000 ELSE
                    salary + 5000
END ;
```

Similar to above we can update the salaries of the employees based on their hired date.

7. How to handle with null values with the help of CASE expression.

Here IS NULL ,IS NOT NULL are used. If

Query:

```
SELECT CASE
            WHEN city IS NULL THEN 'unknown'
            WHEN city IS NOT NULL THEN city
    END as city;
```

Above query update null valued cities with 'unknown' as value

8. Difference between Simple and Searched Case expression?

The simple CASE statement evaluates a single expression and compares it to several potential values.

The searched CASE statement evaluates multiple Boolean expressions and chooses the first one whose value is TRUE .

9. Consider below table named student_details.

ROLL_NO	NAME	ADDRESS	AGE
1	HARSH	BANGALORE	22
2	ARWIND	BANGALORE	23
3	RAHUL	MUMBAI	22
4	KARAN	PUNE	24
5	PAWAN	PUNE	21

10.Write a sql querry using above table to print :

if the AGE is greater than 22 THEN PRINT 'The AGE is greater than 22'

if AGE = 22 THEN PRINT 'The AGE is 22'

otherwise print 'The AGE is under 22' using CASE

Query:

```
SELECT AGE, CASE
   WHEN AGE > 22 THEN 'The AGE is greater than 22' WHEN AGE = 22 THEN
   'The AGE is 22'
   ELSE 'The AGE is under 22' END AS AGEText
FROM students_details;
```

11.Using case expression , in the given above table if person's ADDRESS is Bangalore then prints " person is from Karnataka " , otherwise prints "person is from other state"

Query:

```
SELECT
ADDRESS, CASE
   WHEN ADDRESS == 'BANGALORE' THEN 'PERSON IS FROM KARNATKA'
   ELSE 'person is from other state' END
AS ADDRESSText
FROM students_details;
```

12.What are the main applications of Case Expresssion in SQL? These are mainly used to display values, order sort results, or filter records 12.What do you mean by error codes in sql?

SQL Error Code is used on a day to day basis for the diagnosis of programming failures as a result of SQL calls by DB2 computer programs.What are the some common error codes of sql ?

1146 – Table/ record does not exist 1049 – Database does not eixst 1064 - Syntax error.

1406 – value too long.

1630– Function does not exist.

13. What do you understand by json ?

JSON stands for JavaScript Object Notation JSON is a lightweight data-interchange format

JSON is plain text written in JavaScript object notation JSON is used to send data between computers

JSON is language independent

` **14. What ae the Syntax rules of json**

- Data is in name/value pairs
- Data is separated by commas
- Curly braces hold objects
- Square brackets hold arrays

15. What are json datatypes?

- a string
- a number
- an object (JSON object)
- an array
- a boolean
- null

16. Create a string in json

‘{"name":"John"}’

17. Create an array in json

‘{

"employees":["John", "Anna", "Peter"]

}’

18. How to create null value in json

‘{“name":null}’

WEEK 10: Stored Procedures And Cursors

___1)What are the benefits of using a Stored Procedure in SQL ?

Stored procedures provide some crucial benefits, which are:

- Reusable: As mentioned, multiple users and applications can easily use and reuse stored procedures by merely calling it.
- Easy to modify: You can quickly change the statements in a stored procedure as and when you want to, with the help of the ALTER TABLE command.
- Security: Stored procedures allow you to enhance the security of an application or a database by restricting the users from direct access to the table.
- Low network traffic: The server only passes the procedure name instead of the whole query, reducing network traffic.
- Increases performance: Upon the first use, a plan for the stored procedure is created and stored in the buffer pool for quick execution for the next time.

2. How many types of stored procedures are there in SQL?

There are two types of stored procedures available in SQL Server: User defined stored procedures. System stored procedures.

3.What are system stored procedures?

These stored procedures are just the mechanism the provider or driver uses to communicate user requests to an instance of SQL Server. They are intended only for the internal use of the provider or the driver. Calling them explicitly from a SQL Server-based application is not supported.

4. How to Create a Simple Stored Procedure in SQL without parameters?

Syntax:
```
DELIMITER //
CREATE PROCEDURE procedurename();
BEGIN
Statement
```

```
END //
DELIMITER ;
CALL procedurename();
```

Ex .Query:

```
DELIMITER //
CREATE PROCEDURE emp()
 BEGIN
SELECT * FROM employees;
END //
DELIMITER ;
CALL emp();
```

5. What is the use of delimiter?

In computer programming, a delimiter is a character that identifies the beginning or the end of a character string (a contiguous sequence of characters) and to pass complete procedure as single statement

6. What are the different types of parameters in Stored Procedures?

- Parameters make any stored procedure more flexible

Mode	Significance
IN	• Default mode • Parameter's value is protected as stored procedure works on the copy.
OUT	• Value of parameter can be changed • Updated value is passed to the calling statement
INOUT	• Argument value can be given to stored procedure while calling • Updated value is returned to the calling statement

7. Syntax to create Stored procedures with parameters?

Syntax:

CREATE PROCEDURE procedurename (IN parametername datatype(size))

CREATE PROCEDURE procedurename (OUT parametername datatype(size))

CREATE PROCEDURE procedurename (INOUT parametername datatype(size))

8. What is the difference between IN, OUT and INOUT parameters in stored procedure?

Main difference between IN and OUT parameters is that IN is used to pass data to the database via SQL query, while OUT parameter is used to read results from the database, especially in the case of stored procedure. INOUT parameters are to both send inputs as well as retrieve results .

9. Create a procedure with IN parameter?

Query :

```
DELIMITER $$
    CREATE PROCEDURE empGender (IN arg1 char(2))
    BEGIN
    SELECT * FROM employees WHERE gender = arg1;
END$$
DELIMITER ;
CALL empGender('M');    # it displays only Male employees
CALL empGender('F');    # it displays only Female employees
```

We can also pass two or more IN parameters based query requirement.

10. Create a procedure with IN, OUT parameter?

Query:

```
CREATE PROCEDURE empCount
    (IN arg1 CHAR(2), OUT empCount INT)
    BEGIN
```

```
SELECT COUNT(*) INTO empCount

FROM employees WHERE gender = arg1;

END

CALL empCount('F',@count1);

SELECT @count1;

CALL empCount('M',@count1);
```

Above @count1 is session variable.

11. What are session variables and how to defined sessiona variable?

Session variable begins with '@'

A session variable is a named memory variable that you access through SQL statements. Session variables let you share data between SQL statements without the need for application logic to support this data transfer. Session variables are a way to store data about a user in a database and retrieve it later. Cookies are a way to store data about a user on the user's computer. Session variables are typically used in applications that need to keep track of a user's activity.

They are valid till a particular sesion. And session variables are passed to procedure while calling to the output.

And we can check the value in output by simply selecting the variable

```
SELECT @count1;
```

12. How to define variable in store procedure?

Variables in Stored Procedure

- Variable – Object to store value and its value can be changed during execution of stored procedure
- Declaration of Variable
  Syntax: DECLARE variablename datatype (size) [DEFAULT defaultvalue]

EX: DECLARE a, b INT DEFAULT 0;

How to assign value to variable

syntax: SET variablename = value;

SET a = 100;

SET b = 200;

SELECT INTO

DECLARE empCount INT DEFAULT 0;

SELECT COUNT(*) INTO empCount FROM employees;

13. What is the life of a defined variable?

Defined at	Scope
Inside at Stored Procedure	Till END statement of Procedure
Inside BEGIN and END	Till END statement

14. How to create a stored procedure with INOUT, OUT parameter?

Query: CREATE PROCEDURE empCount

(INOUT arg1 CHAR(2), OUT empCount INT)

BEGIN

SELECT COUNT(*) INTO empCount

FROM employees WHERE gender = arg1;

END

SET @count2 = 'F'; # defining a session variable

CALL empCount (@count2,@count1);

SELECT @count1;

for INOUT parameter in the procedure we pass session variable as an argument ,above it is @count2.

15. How to drop a procedure?

Syntax: DROP PROCEDURE IF EXISTS procedurename;

Ex: DROP PROCEDURE IF EXISTS emp;

If you don't use IF EXISTS command ,it gives an error if that particular procedure is not present.

Syntax: DROP PROCEDURE procedurename;

Ex: DROP PROCEDURE emp;

Error Code:1305.PROCEDURE employees.emp does not exist

16. What are the advantages of MySQL stored procedures ?

1. Reduce network traffic because we only sent the procedure name to SQL server by a call.
2. Make database more secure
3. No need to write set of of SQL statement repetitively it makes things easy for us. So it is faster in execution.

17. MySQL stored procedures dis-advantages?

1. Resource usages: If you use many stored procedures, the memory usage of every connection will increase substantially.
2. Troubleshooting:MySQL does not provide any facilities to debug stored procedures like other enterprise database products such as Oracle and SQL Server
3. Maintenances Developing and maintaining stored procedures often requires a specialized skill set that not all application developers possess.

18. Does SQL support programming language features?

It is true that SQL is a language, but it does not support programming as it is not a programming language, it is a command language. We do not have conditional statements in

SQL like for loops or if..else, we only have commands which we can use to query, update, delete, etc. data in the database. SQL allows us to manipulate data in a database.

19. Write a SQL WHILE loop syntax and example?

```
WHILE condition
BEGIN
  {...statements...}
END
#       Example for While loop in SQL
DECLARE @Counter INT
SET @Counter=1
WHILE ( @Counter <= 10)
BEGIN
   PRINT 'The counter value is = ' + CONVERT(VARCHAR,@Counter)
   SET @Counter  = @Counter  + 1
END
```

20. What is a cursor in SQL?

A database Cursor is a control which enables traversal over the rows or records in the table. This can be viewed as a pointer to one row in a set of rows. Cursor is very much useful for traversing such as for retrieval, addition and removal of database records.

21. Why Use a Cursor in SQL Server ?

Although using an INSERT, UPDATE or DELETE statement to modify all of the applicable data in one transaction is generally the best way to work with data in SQL Server, a cursor may be needed for:

- Iterating over data one row at a time
- Completing a process in a serial manner such as SQL Server database backups
- Updating data across numerous tables for a specific account
- Correcting data with a predefined set of data as the input to the cursor

22. SQL Server Cursor Types ?

Simon Liew has written a detailed technical tip on five Different Ways to Write a Cursor in SQL Server which includes the following:

- Most Common SQL Server Cursor Syntax
- SQL Server Cursor as a Variable
- SQL Server Cursor as Output of a Stored Procedure
- SQL Server Cursor Current Of Example
- SQL Server Cursor Alternative with a WHILE Loop

23. Which type of cursor is automatically declared by Oracle every time an SQL statement is executed?

a. An Implicit

b. An Explicit

c. Both A & B

d. None of the above

Explanation:

The implicit cursor are automatically created. We cannot control this cursor. This cursor is associated with the DML statements.

24. An Explicit cursor is defined by the program for any query that returns more than one row of data. ?

✓ True

Explanation:

An explicit cursor is a user defined cursor and is used for gaining control over the context area. It is mostly used on the SELECT statement which returns more than one row.

25. Which statements execute a sequence of statements multiple times?

- a. EXIT b). LOOP c. Both A & B d. None of the above

Explanation:

- A loop helps us in executing a statement or a group of statement multiple times depending on the block of code.

26. What is SQL for Data Analysis? and Types of Data Analysis ?

SQL for data analysis refers to the database querying language's ability to interact with multiple databases at once, as well as its use of relational databases.

- Statistical Analysis
- Diagnostic Analysis
- Predictive Analysis
- Prescriptive Analysis

27. Explain the various SQL languages.?

SQL queries can be classified into five parts as they perform specific roles to execute queries on any RDBMS system, and they are:

a) Data Definition Language (DDL)

DDL commands include create, alter, drop, rename and truncate, dealing with the structure of the databases. It operates on database objects like views, tables, indexes, and triggers.

b) Data Manipulation Language (DML)

DML commands include insert, update, and delete operations to modify data in existing databases.

c) Data Query Language (DQL)

This command includes a select operation to retrieve data matching criteria specified by the user. To condense data efficiently, DQL commands also involve nested queries.

d) Data Control Language (DCL)

This command is used by data administrators to grant and revoke permission to access data in the organization's database.

e) Transaction Control Language (TCL)

TCL commands help in managing transactions in databases to commit or roll back a current transaction. TCL command is used to commit a DML operation, and it has the ability to club multiple commands in a single operation.

WEEK11: MYSQL for EXCEL

1.What is MySQL for EXCEL?

Oracle's MySQL for Excel is an add-in for Windows-based Excel. It provides a wizard-like interface for browsing MySQL schemas, tables, views, and procedures, and performing data operations against them in Excel.

2.What are the system requirements for the add-in MYSQL for EXCEL.?

Requirement:

a. Microsoft Windows (Not Compatible with MACOS and LINUX)
b. .NET framework
c. Excel 2007 or later
d. Visual Studio Tools for Office 4.0

3. What can be accessed?

- Table
- View
- Schema
- Procedures

But we can't access the Binary Data

4. What is the difference between Excel sheet and MySQL?

Excel built-in functions and features are vast and can perform a multitude of tasks such as statistical functions, financial functions, data visualization and also custom functions where as in MySQL which is database management system there are a lot of functions that are available but minimized to handling the data

5.Can we create a new connection?

Yes , we can create a new connection with the help of option available in excel which will reflect in MySQL workbench .
As both are integrated, changes in one tool will reflect in another tool and vice verca.

6. How do I export data from MySQL to Excel?

Within MySQL for Excel, Open a MySQL Connection, click the employee schema, Next, select the location table, click Edit MySQL Data, then choose Import to import the data into a new Microsoft Excel worksheet for editing.

7.How to import the MySQL table into EXCEL?

Steps :

1.select the schema and the table

2. click on import MySQL table into Excel

3. now we can choose number of row from the selected table and we can choose add summary option OR pivot table option to work on further.

8. What is Excel pivot table used for?

A PivotTable is an interactive way to quickly summarize large amounts of data. You can use a PivotTable to analyze numerical data in detail, and answer unanticipated questions about your data. A PivotTable is especially designed for: Querying large amounts of data in many user-friendly ways.

9.What is the difference between a regular Excel table and a pivot table?

Some of the key difference are:
Straight tables allow interactive sorting, sorting is fixed by the sort order property in pivot tables. Pivot tables allow you to have dimensions displayed on both rows and columns.

10.How to append the data into MySQL table?

Steps:

1.write data

2. select table

3. choose Append excel data to Table

4.choose column mapping method.

While appending the data, the data being appended should satisfy the constraints of table of MySQL

11. What are the different types of mapping methods available in append data?

1. Automatic mapping: Automatic mapping won't allow if we more column values than the actual number columns present in the original table while appending data from excel to MySQL table.

2. manual mapping : here we can append more number of rows at a time , here we have to drag the created to into MySQL table.It takes values if datatypes matches.

WEEK 12: MYSQL data with PYTHON

1)How to connect python with mysql

```
import mysql.connector
testdb = mysql.connector.connect(host="localhost",
user="root",
password="*******")
```

2)CREATE Database USING PYTHON

```
import mysql.connector
testdb = mysql.connector.connect( host="localhost",
user="root",
password="****" )
testcursor = testdb.cursor()
testcursor.execute("CREATE DATABASE test")
```

3) Write an example for table creation in mysql using python

```
import mysql.connector
testdb = mysql.connector.connect(
  host="localhost",
  user="root",
  password="****",
  database="test")
testcursor = testdb.cursor()
        testcursor.execute("CREATE TABLE TableName(roll_no INT, first_name
VARCHAR(30),last_name VARCHAR(30), status VARCHAR(50), location
VARCHAR(50))")
```

4)What is desc command in mysql

The DESC is the short form of DESCRIBE command and used to dipslay the information about a table like column names and constraints on column name.

5)Give An Example For Insertinfg Data To Table In Mysql Using Python

```
import mysql.connector
```

```
testdb =
mysql.connector.connect(host="localhost",user="root",password="****",database="t
est")
testcursor = testdb.cursor()
a = "INSERT INTO TableName (roll_no, first_name, last_name, status, location)
VALUES (%s, %s, %s, %s, %s)"
b = ("121", "John", "Anthony", "Self Employed", "Chandigarh")
testcursor.execute(a,b)
testdb.commit()
print(testcursor.rowcount, "record inserted.")
```

6)Write an example foe deleting a row in mysql using python

```
import mysql.connector

testdb =
mysql.connector.connect(host="localhost",user="root",password="****",database="test")
testcursor = testdb.cursor()
a1 = "DELETE FROM TableName WHERE test_id =%s"
b1 = ("1",)
testcursor.execute(a1,b1)
```

7)Write a python program for drop command in mysql

```
import mysql.connector
        testdb =
mysql.connector.connect(host="localhost",user="root",password="****",database="t
est")
testcursor = testdb.cursor()
testcursor.execute("DROP TABLE TableName)
a = testcursor.fetchall()
```

8)What is the difference between drop and delete command in mysql

DELETE is a Data Manipulation Language command, DML command and is used to remove tuples/records from a relation/table. Whereas DROP is a Data Definition Language, DDL command and is used to remove named elements of schema like relations/table, constraints or entire schema

9)What is the default username in mysql

The default username for the MySQL database is a root.

10)Write simple steps to Connect to mysql Database in Python

STEPS TO CONNECT MYSQL DATABASE IN PYTHON

a.Install MySQL connector module

b.Import MySQL connector module

c.Use the connect() method
d.Use the cursor() method
e.Use the execute() method
f.Extract result using fetchall()
g.Close cursor and connection objects

11)What is fetchall()?

cursor.fetchall(): fetches all the rows of a query result. It returns all the rows as a list of tuples. An empty list is returned if there is no record to fetch.

12)What is fetchmany(size)?

cursor.fetchmany(size): returns the number of rows specified by size argument. When called repeatedly, this method fetches the next set of rows of a query result and returns a list of tuples. If no more rows are available, it returns an empty list.

13)What is fetchone()?

cursor.fetchone(): returns a single record or None if no more rows are available.

14)Write python program to check whether database exist or not

```
import mysql.connector
testdb = mysql.connector.connect(
host="localhost",
user="root",
password="****")
testcursor = testdb.cursor()
testcursor.execute("SHOW DATABASES")
for x in testcursor:
        print(x)
```

15)Write a python program for table alteration

```
import mysql.connector
testdb = mysql.connector.connect(
  host="localhost",
  user="root",
  password="****",
  database="test")
testcursor = testdb.cursor()
testcursor.execute("ALTER TABLE TableName ADD COLUMN test_id
int    auto_increment primary key")
```