Introduction to SQL.

In a database environment, the client or user interface should send a request to the server (database) and the server should respond to the client's request. For any communication, a language is necessary. The language used here is the Structured Query Language, commonly called SQL. It is a simple English-like language that is easy to understand and write. SQL is tied very closely with the relational model.

Languages in SQL

Data Retrieval Language (DRL) -- retrieves data from the server.
Key word

SELECT	----	Read records from the tables.

Data Definition Language (DDL) -- defines the structure of the objects in the database.
Key words

CREATE	----	creates a new object in the database.
ALTER	----	changes the structure of the objects
DROP	----	removes the object from the database
TRUNCATE	----	removes all the records and retains the structure
RENAME	----	changes the name of the table

Data Manipulation Language (DML) -- Manipulates the existing records
Key words

INSERT	-----	adds a new record or row to the table
UPDATE	-----	changes the values of the existing records
DELETE	-----	removes one or more records from the table.

Transaction Control Language (TCL) -- Controls the transactions
Key words

COMMIT ----- saves the changes in the database.
ROLLBACK ----- undo the change made.
SAVE POINT ----- intermediate buffer or a point between the transactions

Data Control Language (DCL) -- Controls the database actions and data access.
Key words

GRANT ----- provide privileges on the database objects to the users.
REVOKE ----- removes privileges to the database objects

Data Definition Language

It refers to the set of SQL commands, which are used to define the structure of a database object.
Various commands available under DDL are

- CREATE
- ALTER
- TRUNCATE
- RENAME
- DROP

i) CREATE command: It is used for creating objects in the database. Table is the base object in the database.
Syntax

```
CREATE TABLE table_name (
Column1_name      data    type      [constraint]    DEFAULT
default_value,
Column2_name data type   [constraint],
.....
Column_name  data type, [Constraint(column_name)]);
```

In this syntax:

First, specify the table name.

Second, list all columns of the table within the parentheses. In case a table has multiple columns, then the columns must be separated by a comma (,).

Column definition is a combination of four things 1) column name followed by its 2) data type and size e.g., INT(3), VARCHAR2, and 3) an optional default value 4) column constraint such as NOT NULL, PRIMARY KEY, CHECK etc..

Third, add table constraints if applicable e.g: PRIMARY KEY, FOREIGN KEY, CHECK etc

Note : Keywords given within [ ] are optional.

Example:

```
SQL> CREATE TABLE students (
     Student_id int PRIMARY KEY,
     Student_name VARCHAR2(10),
     Date_of_birth DATE,
     Mobile_number int(10);
```

The above command creates a table named students with columns Student_id, Student_name etc, having data types NUMBER, VARCHAR2 respectively.

PRIMARY KEY is a constraint, which is specified for a column or combination of columns that ensures unique values. A table can have only one primary key.

Data Types in MySQL

MySQL supports several standard SQL data types. Each column can contain only one data type. In MySQL, data types are grouped in different categories:

.

Numeric

- Date and time
- String

Numeric Data Types (Number Formats)

MySQL supports numeric data types such as integers, decimals, and floating-point data types

Integer data types

TINYINT :	Allowable range from 0 to 255 and width max 4 digits.
SMALLINT	Allowable range from 0 to 65535 and width max 5 digits
MEDIUMINT	Allowable range from 0 to 16777215 and width max 9 digits
INT	Allowable range from 0 to 4294967295 and width max 11 digits
BIGINT	Allowable range from 0 to 18446744073709551615 and width max 20 digits.
FLOAT(m,d)	Display length (m) and decimal (d) max 24 decimal places
DOUBLE(m,d)	max. 53 decimal place.

Date and time data types

DATE	Default format 'yyyy-mm-dd'
TIME	Default format 'HH:MI:SS'
DATETIME	Default format 'yyyy-mm-dd hh:mm:ss'
TIMESTAMP	Default format 'YYYY-MM-DD HH:MI:SS'

String data types

CHAR(n)	Fixed length string	max of 255 characters.
VARCHAR(n)	Variable length string	max 255 characters.
TINYTEXT	Variable length string	max 255 characters
TEXT	Variable length string	max 65535 characters
MEDIUM TEXT	Variable length string	max 16,777,215 char
BLOB	Binary large Object	max 65535 bytes
MEDIUMBLOB	Binary large Object	max 16,777,215 bytes
LONGBLOB	Binary large Object	max 4gb
ENUM	Allows only the values in the given list	

Naming conventions for objects/columns etc

1. Name must start with an alphabet.
2. Can contain numbers and special characters (_, $, #).
3. Must not contain space.
4. Must be unique.

ii) ALTER command: The ALTER table command is used to modify the structure of the table. Modifying the structure includes adding / dropping columns/constraints, changing data types and sizes, and renaming columns.

Adding a column to the table

To add a column to an existing table, use the ALTER TABLE...ADD statement.

Syntax:

ALTER TABLE table_name ADD column_name data type [FIRST |AFTER column_name];

Add Multiple columns

ALTER TABLE table_name ADD (column1_name data type [FIRST |AFTER column_name], ADD column2_name data type… [FIRST |AFTER column_name])

Note : FIRST | AFTER column_name: It is optional. It tells MySQL where in the table to create the column. If this parameter is not specified, the new column will be added to the end of the table.

Example :

```
> ALTER TABLE students ADD address VARCHAR(15);

> ALTER TABLE students ADD  regno INT  VARCHAR(15) FIRST;

> ALTER TABLE students ADD (city VARCHAR(10),

  add pincode MEDIUMINT);
```

Modify the column of a table

Use the ALTER TABLE...MODIFY statement to modify an existing column definition. You can modify column data type, default value, column constraint, and increase or decrease the length of the column.

Syntax:

```
> ALTER TABLE  table_name  MODIFY  column column_name  data type(size);
```

Example :

```
> ALTER TABLE students MODIFY column city VARCHAR(20);
```

Drop a column from the table

The columns which are no more needed can be dropped from the table using ALTER TABLE... DROP statement.

Syntax:

```
> ALTER TABLE table_name DROP COLUMN column_name;

> ALTER TABLE table_name DROP column1,  drop column2…);
```

Example :

```
> ALTER TABLE students DROP COLUMN pincode;
```

Note: You cannot drop all columns from a table

Renaming a column

The ALTER TABLE ….. RENAME COLUMN…. statement is used to change the name of the column or change the name of the table.

Syntax:

```
> ALTER TABLE table_name RENAME COLUMN old_name TO new_name
```

Example:

```
> ALTER TABLE students RENAME COLUMN date_of_birth TO dob;
```

Renaming a table

Syntax:

```
> ALTER TABLE table_name RENAME RENAME TO new_name
```

Example:

```
> ALTER TABLE students RENAME TO dob;
```

iii) TRUNCATE command: TRUNCATE is used to delete all the rows in the table, but maintains the structure of the table. It also releases the space allocated. It cannot be rolled back.

Syntax:

```
TRUNCATE TABLE table_name;
```

Example:

```
TRUNCATE TABLE students;
```

iv) RENAME command: The RENAME command will change the name of the existing table

Syntax:

```
RENAME TABLE old_name TO new_name;
```

```
Renaming Multiple tables

RENAME TABLE old_name1 TO new_name1,

             TABLE old_name2 TO new_name2;
```

Data Manipulation language (DML)

DML statements are the elements of SQL language which are used to manipulate the values in the database. It is used for adding ,deleting and modifying data

DML consists of the following set of commands.

- INSERT
- UPDATE
- DELETE

Insert statement:
This statement is used for adding / inserting rows to a table in the database.

Syntax:

```
INSERT INTO table_name (column1_name,column2_name….)

 VALUES (value1,value2…..);
```

Example:

```
INSERT INTO students_profile(regno,name,age,mailid)
VALUES(1001,’Marshal’,18,’marshal@gmail.com’ )
```

Note:

- Column name list is optional if values are given for all columns.
- Character and date values must be enclosed within single quotes.
- Date must be in ‘YYYY-MM-DD’ format.
- Values are case sensitive.

Inserting multiple values:

Syntax:

```
INSERT   INTO   table_name   (c1,c2,...)     VALUES         (v11,v12,...),
(v21,v22,...),    ...  (vnn,vn2,...);
```

Example:

```
INSERT INTO tasks(title, priority) VALUES   ('My first task', 1),
                                            ('It is the second task',2),
                                     ('This is the third task of the week',3);
```

Inserting default values:

Syntax:

```
INSERT INTO table_name (c1,c2,...)  VALUES    (v11,v12,.DEFAULT..);
```

Example:

```
INSERT INTO tasks(title,priority) VALUES('My default tsk',DEFAULT);
```

NULL Value: NULL is the term used to represent a missing value or unknown value. It is not equal to 0 nor to Space.

Inserting NULL values

Option 1: Exclude the column names for which NULL values are to be inserted.

Syntax

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

Example

```
INSERT INTO students_profile(regno,name,age)
VALUES(1001,’Marshal’,18);
```

Option 2: Specify NULL explicitly

```
INSERT INTO students_profile VALUES(1001,’Marshal’,18,null);
```

UPDATE statement: The UPDATE statement is used to modify or change the values in a table. Either all the rows can be updated or a subset may be chosen using a condition in the WHERE clause.

Syntax:

```
UPDATE table_name SET column1=value1, column2=value2 [WHERE
column=condition];
```

Example:

```
UPDATE students_profile SET NAME=’ADAMS’,
WHERE regno=2;
```

Updating Multiple Fields:

If you are going to update multiple fields, you should separate each field assignment with a comma.

Example:

```
UPDATE students  SET User_Name = 'Walkman',
        First_Name = 'Johnny'  WHERE Student_Id = '3'
```

DELETE statement: The DELETE statement removes one or more records from a table. A subset may be defined for deletion using a condition, otherwise all records are removed.

Syntax

```
DELETE FROM table_name [WHERE column=condition];
```

Example

```
DELETE FROM students_profile
WHERE regno=2;
```

To delete all the records

Syntax

```
DELETE FROM table_name;
```

Example

```
DELETE FROM student_profile;
```

Transaction Control Language (TCL)

What is a Transaction?

Database Transaction is a single unit of logic or operation that is carried out by a user or application. The operations can include retrieval, (Read), insertion, deletion and modification (Write) A transaction must be either completed or aborted. Transaction that changes the contents of the database must alter the database from one consistent state to another. A consistent database state is one in which all data integrity constraints are satisfied.A successful transaction can change the database from one CONSISTENT STATE to another. DBMS transactions must be atomic, consistent, isolated and durable

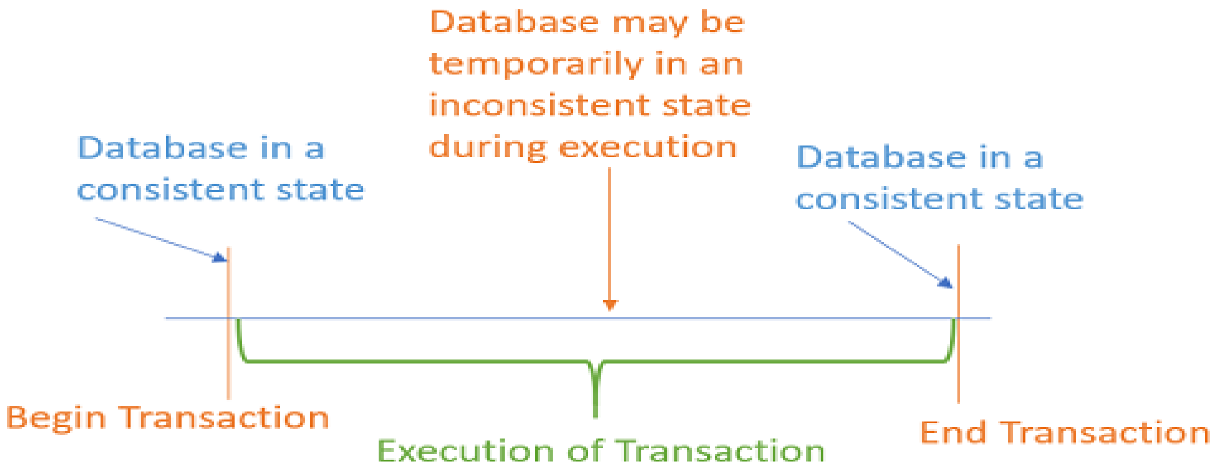

Database may be temporarily in an inconsistent state during execution
Database in a consistent state
Database in a consistent state
Begin Transaction
Execution of Transaction
End Transaction

Figure 2.1

Example: Consider the following example of transaction operations to be performed to withdraw cash from an ATM vestibule.

Steps for ATM Transaction

- Transaction Start.
- Insert your ATM card.
- Select a language for your transaction.
- Select the Savings Account option.
- Enter the amount you want to withdraw.
- Enter your secret pin.
- Wait for some time for processing.
- Collect your Cash.
- Transaction Completed.

A transaction can include the following basic database access operation.

- Read/Access data (R): Accessing the database item from disk (where the database stored data) to memory variable.
- Write/Change data (W): Write the data item from the memory variable to the disk.
- Commit: Commit is a transaction control language that is used to permanently save the changes done in a transaction

Desirable properties of transaction

For a transaction to be performed in DBMS, it must possess several properties often called **ACID properties**.

ACID Properties are used for maintaining the integrity of the database during transaction processing. ACID in DBMS stands for Atomicity, Consistency, Isolation, and Durability.

- **A – Atomicity**
- **C – Consistency**
- **I – Isolation**
- **D – Durability**

Atomicity: A transaction is a single unit of operation. You either execute it entirely or do not execute it at all. There cannot be partial execution.

Consistency: Once the transaction is executed, it should move from one consistent state to another.

Isolation: Transactions should be executed in isolation from other transactions (no Locks). During concurrent transaction execution, intermediate transaction results from simultaneously executed transactions should not be made available to each other.

Durability:· After successful completion of a transaction, the changes in the database should persist. Even in the case of system failures.

Transaction Control language.

Transaction Control Language(TCL) commands are used to manage transactions in databases. These are used to manage the changes made by DML statements. It also allows statements to be grouped together into logical transactions.

TCL Commands are as follows

a) **COMMIT:** COMMIT command is used to permanently save any transaction into the database. It ends the current transaction.

syntax

```
COMMIT;
```

b) ROLLBACK: ROLLBACK command restores database to original since the last COMMIT. It is also used with the savepoint command to jump to a savepoint in a transaction.

syntax

```
ROLLBACK;
```

c) **SAVEPOINT :** SAVEPOINT command is used to temporarily save a transaction so that you can rollback to that point whenever necessary

syntax:

```
Statement 1;
SAVEPOINT savepoint_name1;
Statement 2;
SAVEPOINT savepoint_name2;
ROLLBACK to SAVEPOINT savepoint_name;
```

Example:

```
INSERT INTO students_profile VALUES (10,’Jeni’,18,’jeni@outlook.com’;
SAVEPOINT a;
INSERT INTO students_profile VALUES
(11,’Mithun’,19,’mithun@outlook.com’;
SAVEPOINT b;
UPDATE students_profile SET age=20 WHERE regno=10;
SAVEPOINT c;
DELETE FROM  students WHERE name=’Mithun’;
ROLLBACK to SAVEPOINT b;
```

Exercises

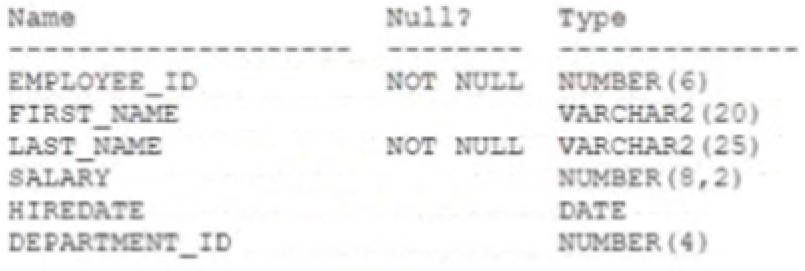
Answer the following

1. Which of the following is/are the DDL statements?
   a. Create
   b. Drop.
   c. Alter
   d. All of the above

2. Alter table command is used to perform which of the following operations?
   a. Removing one or more records
   b. Modify the values of the records.
   c. Add one or more columns.
   d. Remove the table.

3. Which of the following commands is used to remove a database
   a. Drop database.
   b. Delete database.
   c. Remove database.
   d. None of the above.

4. Examine the structure of the EMPLOYEES table



```
Name                  Null?     Type
--------------------  --------  --------------
EMPLOYEE_ID           NOT NULL  NUMBER(6)
FIRST_NAME                      VARCHAR2(20)
LAST_NAME             NOT NULL  VARCHAR2(25)
SALARY                          NUMBER(8,2)
HIREDATE                        DATE
DEPARTMENT_ID                   NUMBER(4)
```

Which two statements will insert a row into the EMPLOYEES table? (Choose two.)

A. INSERT INTO employees VALUES (101, 'John', 'Smith', 12000, SYSDATE);

B. INSERT INTO employees VALUES (101, 'John', 'Smith', 10, 12000, SYSDATE);

C. INSERT INTO employees (employee_id, salary, first_name,

hiredate, last_name) VALUES (101, 12100, 'John', SYSDATE, 'Smith');

D. INSERT INTO employees (employee_id, first_name, last_name, salary, hiresate)
VALUES ((SELECT 101, 'John', 'Smith'. 12000, SYSDATE FROM dual));

E. INSERT INTO employees SELECT 101, 'John', 'Smith', 12000, (SELECT SYSDATE FROM dual), 10 FROM dual;

F. INSERT INTO employees VALUES (101, 'John', ' ', 12000, SYSDATE, 10);

5. Which three statements are true about the ALTER TABLE....DROP COLUMN.... command? Choose all that apply.

   A. A column can be dropped only if it does not contain any data.
   B. A column can be dropped only if another column exists in the table.
   C. A dropped column can be rolled back.
   D. The column in a composite PRIMARY KEY with the CASCADE option can be dropped.
   E. A parent key column in the table cannot be dropped.

6. Examine this command:

TRUNCATE TABLE test;

Table truncated.

Which two are true? (Choose two.)

A. The structure of the TEST table is removed.

B. All the indexes on the TEST table are dropped.

C. All the constraints on the TEST table are dropped.

D. Removed rows can not be recovered using the ROLLBACK command.

E. All the rows in the TEST table are removed.

7. Which three actions can you perform on an existing table containing data? (Choose all that apply.)

A. Increase the width of a numeric column.

B. Add a new column as the table's first column.

C. Define a default value that is automatically inserted into a column containing nulls.

D. Change a DATE column containing data to a NUMBER data type.

E. Change the default value of a column.

F. Add a new NOT NULL column with a DEFAULT value.

8. ___________________ is used to modify the existing values

a. Modify
b. Alter
c. Change
d. Update

9. Which of the following is not true about the DELETE command.

e. Can remove specific records
f. Can be rolled back
g. Will release the space
h. None of the above.

10. Examine this sequence of statements issued in a new session:

```
INSERT INTO books VALUES
            ('ADV112', 'Adventures of Tom Sawyer', NULL, NULL);
      SAVEPOINT a;
      DELETE FROM books;
      ROLLBACK TO SAVEPOINT a;
      ROLLBACK;
```

11. Which two statements are true? (Choose two.)

a. The second ROLLBACK command replays the delete.
b. The first ROLLBACK command restores the 101 rows that were deleted and commits the inserted row.
c. The first ROLLBACK command restores the 101 rows that were deleted, leaving the inserted row still to be committed.
d. The second ROLLBACK command undoes the insert.
e. The second ROLLBACK command does nothing.

Task 1:

1. Create the following tables for a hotel management system with the attributes given. Choose appropriate data types .
    a. Hotel (hotel_d, hotel_name, location)
    b. Rooms (roomid, hotelid, room_type)
    c. Guest (guestid, guestname, city)
    d. Reservations (roomid, guestid , date_checkin, date_checkout)
2. Store a minimum of 5 records to each of the above tables.
3. Display the records from all the above tables.
4. The mobile number of each guest must be stored. Modify the guest table to store the same.
5. Increase the size of location to 50.

6. Change the name of the column date_ckeckin to checkinDate.
7. Add a column price to the rooms table.
8. Store the price of each room type. Ex: Single room price is 1500. Double room price is 3000 and family suit price is 10000.
9. Remove the details of reservations done on 25-09-2023.
10. Show the structure of all tables.

Task 2:

1. Create the following table for animals with the attributes given. Choose appropriate data types .
   Animals(id, name, species,Life_span).
2. Insert the following records into the above table.

ID	Name	Species	Life_span
1	Cat	Animal	20
2	Dog	Animal	25
3	Monkey	Animal	26
4	Elephant	Animal	65
5	Canary	Bird	20
6	Swift	Bird	5
7	Dove	Bird	12
8	Rabbit	Animal	10
9	Cow	Animal	25

3. Change the length of life_expectancy column in such a way that it can hold a maximum of 3 digits.
4. Add a new column Habitat with a suitable data type.

5. Populate the Habitat column for each animal. The Values can be ‘Domestic’ or ‘Wild’ .
6. Add a new Domestic animal Horse with a life expectancy of 25 years.
7. Increase the life expectancy for all domestic animals by 5 .
8. Delete all the wild animals with life expectancy less than 8 years.
9. Display the details of all animals.
10. Remove all the data from the animals table.