

STRUCTURED QUERY LANGUAGE

ASSIGNMENT – 1



AAFT, SCHOOL OF DATA SCIENCE

COURSE-BS.C ARTIFICIAL INTELLIGENCE

NAME-MEHAR MARWAH

ROOM NO.-LAB NO.8

TABLE OF CONTENTS

| |
|--|
| 1. INTRODUCTION TO SQL |
| 2. IMPORTANCE OF DATABASE MANAGEMENT SYSTEM |
| 3. DBMS VS TRADITIONAL FILES |
| 4. AN INTRO TO SQL COMMANDS |
| 5. ABSTRACTION VS SCHEMA |
| |

INTRODUCTION TO SQL

WHAT IS SQL ?

There are several numbers of database query languages in use like Cypher, GraphQL, XQuery etc, either commercially or experimentally. SQL is one of them.

Although we refer to SQL as “query language”, it can do much more than just query a database and specify security constraints.

OVERVIEW OF SQL QUERY LANGUAGE :

IBM developed the original version of SQL, originally called Sequel, as part of the System R project in the early 1970's . This language has evolved since then and now it's named as SQL or Structured Query Language .It has established itself as standard relational database language as now many products support SQL language .

In 1986 the American National Standards Institute (ANSI) and the International Organization for Standardization (ISO) published an SQL standard called SQL-89 in 1989.

Later , the next version of standard was published called SQL-92 followed by SQL-1999 , SQL-2003 , SQL-2006 and SQL-2008.

It has several parts such as :-

1.DDL – Data Definition Language

a) Integrity

b) view definition

2.DML-Data Manipulation language

3)Transaction control

4)Embedded SQL or Dynamic SQL

5) Authorization

IMPORTANCE OF DATABASE MANAGEMENT SYSTEM

WHAT IS DBMS ?

A DBMS is a collection of interrelated data and a set of programs to access those data.

The collection of data , usually referred to as database information relevant to an enterprise . The primary goal of DBMS is to provide a way to store and retrieve database that is both convenient and efficient.

IMPORTANCE OF DATABASE:

IT is designed to manage large bodies of information.

- 1. IT is used for storing and retrieving information conveniently and efficiently.**
- 2. It helps in managing large bodies of data**
- 3. IT helps in defining structures for storage and manipulation.**
- 4. Ensuring safety of information against crashers or unauthorized access.**
- 5. It also helps in avoiding anomalous results when data is shared among several users .**

WHAT ARE TRADITIONAL FILES ?

Traditional files were used for data storage and retrieval before databases.

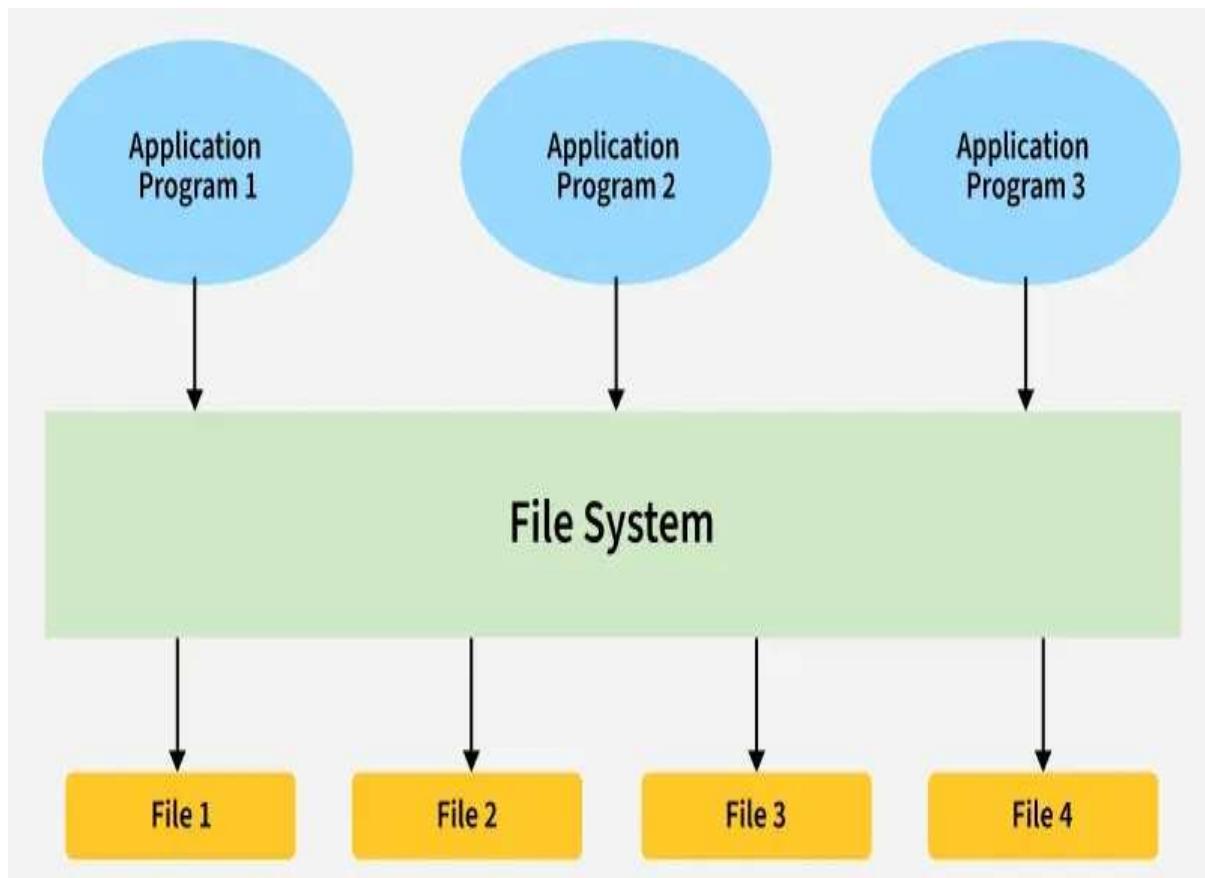
It is a method of storing and managing data in individual files. These files are organized in directories or folders on a computer's storage device .In this system each file is designed to store data related to specific purpose example bank transactions and customer records but not inherently connected to specific files.

The data here is accessed, updated, or deleted manually by custom records.

Before the use of computer, a manual file system was used to maintain files and records . It was stored and processed using a traditional file system and it makes easy access to find any information.

In traditional file system , each file is independent of other files, and data in the file can be integrated only by writing an individual program for each application.

WORKING OF A TRADITIONAL FILE :



DBMS VS TRADITIONAL FILES :

| DBMS | TRADITIONAL FILES |
|--|---|
| <p>1. It is a collection of data. In DBMS the user is not required to write the procedures such as Data Consistency, Security and access control and etc.</p> <p>2. In DBMS users don't see all the raw details of how data is actually stored in database.</p> <p>3. It provides a good protection mechanism .</p> <p>4. It has a wide variety of techniques to store and retrieve the data.</p> <p>5. Redundancy is controlled in DBMS.</p> <p>6. Unauthorized access is restricted in data.</p> <p>7. It is made to allow flexible access to data.</p> <p>8. It helps to coordinate i.e accessing the same data at the same time.</p> | <p>1. In Traditional file system data and program are inter-dependent.</p> <p>2. It can cause data redundancy i.e The data may be duplicated in different files.</p> <p>3. It also causes data inconsistency i.e data files may be different that cause data inconsistency.</p> <p>4. It cannot be shared as it is stored in different files.</p> <p>5. It does not provide consistency constraints.</p> <p>6. It is a less complex system.</p> <p>7. It takes more space and more memory is wasted.</p> <p>8. To generate different reports to take a crucial decision .</p> |

AN INTRODUCTION TO SQL COMMANDS :

THERE ARE 5 TYPES TO SQL COMMANDS :

1.DDL

Data definition language commands are employed for creating, modifying and deleting database objects.

It contains 4 things:

1. **CREATE** :IT builds new database objects , CREATE TABLE EMPLOYEE(...)
2. **ALTER**: Modifies existing objects ALTER TABLE Employees ADD Phone;
3. **DROP** : Drop table employees
4. **TRUNCATE** : Removes all rows from a table, retaining structure TRUNCATE Table Logs ;

2.DML

Data manipulating Language commands are used for managing data with database objects.

IT Has 3 Objects

1. INSERT

IT adds new rows of data into a table INSERT INTO Products Values (...)

2. DELETE

Removes rows from a table

DELETE FROM Customers WHERE Region='North';

3. UPDATE

Modifies existing data in a table

UPDATE Orders SET

Status='Shipped' WHERE

ID=1;

3.DQL

Data Query Language is primarily focused on the ‘SELECT Command for fetching data from databases.

IT HAS 3 THINGS:

1. SELECT COLUMN FROM TABLE

Basic data retrieval

2. SELECT...WHERE CONDITION

Filtering data based on criteria

3. SELECT...JOIN...ON

Combining data from multiple tables.

3.DCL AND TCL

| DCL DATA CONTROL LANGUAGE: MANAGES DATABASE permission and access. GRANT: Provides user across privileges . REVOKE : Removes user access privileges. | TCL TRANSACTION CONTROL LANGUAGE: Manages statements by DML statements COMMIT:Saves changes permanently ROLLBACK : Undoes changes since last commit SAVE POINT: Sets a point to roll back to. |
|--|--|
|--|--|

CONCEPT OF DATA ABSTRACTION AND INDEPENDENCE :

DATA ABSTRACTION

Data Abstraction is the process of hiding complex implementation details from the user. Its purpose is to provide users with a simplified , relevant view of the data , focusing on “what” data is needed rather than “how” it’s stored or manipulated .

DATA INDEPENDENCE

Data Independence is the ability to modify the scheme at one level of the database system without affecting the schema at the next higher level. Its core purpose is to protect applications from structural changes in the database system.

Q2) COMMANDS IN SQL:

Screenshot 1: MySQL Workbench - SQL File 3*

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the database schema with tables like art, bookid, customers, employee_and_their_salary, employeeid_and_their_salary, orders, studentdb (Tables, Views, Stored Procedures, Functions), sys, and title.
- SQL Editor:** Contains the following SQL code:


```

1 • CREATE DATABASE IF NOT EXISTS studentDB;
2 • USE studentDB;
3
4
5 • DROP TABLE IF EXISTS Students;
      
```
- Result Grid:** Displays the data from the Students table:

| RollNo | Name | Age |
|--------|-----------------|-----|
| 1 | Rahul Singh | 26 |
| 2 | Aman Sharma | 21 |
| 3 | Archana Singh | 22 |
| 4 | Manjot Bajwa | 23 |
| 5 | Simran Kaur | 23 |
| 7 | Mayank Kapoor | 24 |
| 9 | Harshita Kasana | 22 |
| 10 | Mehar Marwah | 21 |
- Action Output:** Shows the execution history with the following entries:

| # | Time | Action | Message | Duration / Fetch |
|---|----------|---|--|-----------------------|
| 4 | 17:08:54 | USE studentDB | 0 row(s) affected | 0.000 sec |
| 5 | 17:08:54 | DROP TABLE IF EXISTS Students | 0 row(s) affected, 1 warning(s): 1051 Unknown table 'studentdb.students' | 0.015 sec |
| 6 | 17:08:54 | CREATE TABLE Students (RollNo INT PRIMARY KEY, Name VARCHAR(60) NOT NULL, Age INT CHECK (Age >= 10), Gender VARCHAR(10) NOT NULL) | 0 row(s) affected | 0.157 sec |
| 7 | 17:08:54 | INSERT INTO Students (RollNo, Name, Age, Gender) VALUES (1, 'Rahul Singh', 26, 'Male') | 10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0 | 0.031 sec |
| 8 | 17:08:54 | SELECT RollNo, Name, Age FROM Students WHERE Age > 20 LIMIT 0, 1000 | 8 row(s) returned | 0.000 sec / 0.000 sec |
| 9 | 17:08:54 | DROP TABLE Students | 0 row(s) affected | 0.079 sec |

Screenshot 2: MySQL Workbench - SQL File 3*

The second screenshot shows the same environment but with different SQL code:

- SQL Editor:** Contains the following SQL code:


```

8 • CREATE TABLE Students (
9   RollNo INT PRIMARY KEY,
10  Name VARCHAR(60) NOT NULL,
11  Age INT CHECK (Age >= 10),
12  Gender VARCHAR(10) NOT NULL
      
```
- Result Grid:** Displays the data from the Students table (same as Screenshot 1).
- Action Output:** Shows the execution history with the following entries:

| # | Time | Action | Message | Duration / Fetch |
|---|----------|---|--|-----------------------|
| 4 | 17:08:54 | USE studentDB | 0 row(s) affected | 0.000 sec |
| 5 | 17:08:54 | DROP TABLE IF EXISTS Students | 0 row(s) affected, 1 warning(s): 1051 Unknown table 'studentdb.students' | 0.015 sec |
| 6 | 17:08:54 | CREATE TABLE Students (RollNo INT PRIMARY KEY, Name VARCHAR(60) NOT NULL, Age INT CHECK (Age >= 10), Gender VARCHAR(10) NOT NULL) | 0 row(s) affected | 0.157 sec |
| 7 | 17:08:54 | INSERT INTO Students (RollNo, Name, Age, Gender) VALUES (1, 'Rahul Singh', 26, 'Male') | 10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0 | 0.031 sec |
| 8 | 17:08:54 | SELECT RollNo, Name, Age FROM Students WHERE Age > 20 LIMIT 0, 1000 | 8 row(s) returned | 0.000 sec / 0.000 sec |
| 9 | 17:08:54 | DROP TABLE Students | 0 row(s) affected | 0.079 sec |

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: SCHEMAS

Filter objects: studentdb

Result Grid | Filter Rows: | Edit: | Export/Import: | Wraps: | Result Grid | Form Editor

Output: Action Output

| # | Time | Action | Message | Duration / Fetch |
|---|----------|---|--|-----------------------|
| 4 | 17:08:54 | USE studentDB | 0 row(s) affected | 0.000 sec |
| 5 | 17:08:54 | DROP TABLE IF EXISTS Students | 0 row(s) affected, 1 warning(s): 1051 Unknown table 'studentdb.students' | 0.015 sec |
| 6 | 17:08:54 | CREATE TABLE Students (RollNo INT PRIMARY KEY, Name VARCHAR(60) NOT NULL, Age INT, Gender CHAR(1)) | 10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0 | 0.157 sec |
| 7 | 17:08:54 | INSERT INTO Students (RollNo, Name, Age, Gender) VALUES (1, 'Rahul Singh', 26, 'Male') | 10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0 | 0.031 sec |
| 8 | 17:08:54 | SELECT RollNo, Name, Age FROM Students WHERE Age > 20 LIMIT 0, 1000 | 8 row(s) returned | 0.000 sec / 0.000 sec |
| 9 | 17:08:54 | DROP TABLE Students | 0 row(s) affected | 0.079 sec |

Query Completed

32°C Partly sunny

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: SCHEMAS

Filter objects: studentdb

Result Grid | Filter Rows: | Edit: | Export/Import: | Wraps: | Result Grid | Form Editor

Output: Action Output

| # | Time | Action | Message | Duration / Fetch |
|---|----------|---|--|-----------------------|
| 4 | 17:08:54 | USE studentDB | 0 row(s) affected | 0.000 sec |
| 5 | 17:08:54 | DROP TABLE IF EXISTS Students | 0 row(s) affected, 1 warning(s): 1051 Unknown table 'studentdb.students' | 0.015 sec |
| 6 | 17:08:54 | CREATE TABLE Students (RollNo INT PRIMARY KEY, Name VARCHAR(60) NOT NULL, Age INT, Gender CHAR(1)) | 10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0 | 0.157 sec |
| 7 | 17:08:54 | INSERT INTO Students (RollNo, Name, Age, Gender) VALUES (1, 'Rahul Singh', 26, 'Male') | 10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0 | 0.031 sec |
| 8 | 17:08:54 | SELECT RollNo, Name, Age FROM Students WHERE Age > 20 LIMIT 0, 1000 | 8 row(s) returned | 0.000 sec / 0.000 sec |
| 9 | 17:08:54 | DROP TABLE Students | 0 row(s) affected | 0.079 sec |

Query Completed

32°C Partly sunny

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: SCHEMAS

Filter objects

- art
- bookid
- customers
- employee_and_their_salary
- employeeid_and_their_salary
- orders
- studentdb**
 - Tables
 - Views
 - Stored Procedures
 - Functions
- sys
- title

SQL File 3* | Limit to 1000 rows | Jump to

```

28
29 •  SELECT RollNo, Name, Age
30   FROM Students
31   WHERE Age > 20;
32
33
34
  
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Writ | Result Grid | Form Editor

| RollNo | Name | Age |
|--------|-----------------|-----|
| 2 | Aman Sharma | 21 |
| 3 | Archana Singh | 22 |
| 4 | Manjot Bajwa | 23 |
| 5 | Simran Kaur | 23 |
| 7 | Mayank Kapoor | 24 |
| 9 | Harshita Kasana | 22 |
| 10 | Mehar Marwah | 21 |

Students 1 | Apply | Revert | Context Help | Snippets

No object selected

Action Output

| # | Time | Action | Message | Duration / Fetch |
|---|----------|--|--|-----------------------|
| 4 | 17:08:54 | USE studentDB | 0 row(s) affected | 0.000 sec |
| 5 | 17:08:54 | DROP TABLE IF EXISTS Students | 0 row(s) affected, 1 warning(s): 1051 Unknown table 'studentdb.students' | 0.015 sec |
| 6 | 17:08:54 | CREATE TABLE Students (RollNo INT PRIMARY KEY, Name VARCHAR(60) NOT NULL, Age INT, Gender CHAR(1)) | 10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0 | 0.157 sec |
| 7 | 17:08:54 | INSERT INTO Students (RollNo, Name, Age, Gender) VALUES (1, 'Rahul Singh', 26, 'Male') | 10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0 | 0.031 sec |
| 8 | 17:08:54 | SELECT RollNo, Name, Age FROM Students WHERE Age > 20 LIMIT 0, 1000 | 8 row(s) returned | 0.000 sec / 0.000 sec |
| 9 | 17:08:54 | DROP TABLE Students | 0 row(s) affected | 0.079 sec |

Query Completed

32°C Partly sunny | Search | 5:14 PM 9/18/2025

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: SCHEMAS

Filter objects

- art
- bookid
- customers
- employee_and_their_salary
- employeeid_and_their_salary
- orders
- studentdb**
 - Tables
 - Views
 - Stored Procedures
 - Functions
- sys
- title

SQL File 3* | Limit to 1000 rows | Jump to

```

30
31   WHERE Age > 20;
32
33
34 •  DROP TABLE Students;
  
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Writ | Result Grid | Form Editor

| RollNo | Name | Age |
|--------|-----------------|-----|
| 2 | Aman Sharma | 21 |
| 3 | Archana Singh | 22 |
| 4 | Manjot Bajwa | 23 |
| 5 | Simran Kaur | 23 |
| 7 | Mayank Kapoor | 24 |
| 9 | Harshita Kasana | 22 |
| 10 | Mehar Marwah | 21 |

Students 1 | Apply | Revert | Context Help | Snippets

No object selected

Action Output

| # | Time | Action | Message | Duration / Fetch |
|---|----------|--|--|-----------------------|
| 4 | 17:08:54 | USE studentDB | 0 row(s) affected | 0.000 sec |
| 5 | 17:08:54 | DROP TABLE IF EXISTS Students | 0 row(s) affected, 1 warning(s): 1051 Unknown table 'studentdb.students' | 0.015 sec |
| 6 | 17:08:54 | CREATE TABLE Students (RollNo INT PRIMARY KEY, Name VARCHAR(60) NOT NULL, Age INT, Gender CHAR(1)) | 10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0 | 0.157 sec |
| 7 | 17:08:54 | INSERT INTO Students (RollNo, Name, Age, Gender) VALUES (1, 'Rahul Singh', 26, 'Male') | 10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0 | 0.031 sec |
| 8 | 17:08:54 | SELECT RollNo, Name, Age FROM Students WHERE Age > 20 LIMIT 0, 1000 | 8 row(s) returned | 0.000 sec / 0.000 sec |
| 9 | 17:08:54 | DROP TABLE Students | 0 row(s) affected | 0.079 sec |

Query Completed

32°C Partly sunny | Search | 5:14 PM 9/18/2025

THANK YOU