

---

# Lecture 1

**Contents:** Introduction; History; Benefits of SQL; Database access; SQL versus high level language; SQL for business requirements; Initiatives by IBM; Initiatives by Microsoft; SQL integration with Java; Exercises

## 1.1 Introduction

SQL stands for *structured query language*. It is a domain-specific language used in programming and designed for managing data held in a relational database management system (RDBMS), or for stream processing in a relational data stream management system (RDSMS). It is useful in handling structured data, i.e. data incorporating relations among entities and variables. There are advantages of SQL over indexed sequential access method (ISAM) or virtual storage access method (VSAM). It introduced the concept of accessing many records with one single command. Also, it eliminates the need to specify how to reach a record with or without the help of an index.

SQL includes data query, data manipulation (insert, update and delete), data definition (schema creation and modification), and data access control. SQL is a declarative language (4GL), but it also includes procedural features.

## 1.2 History

E F Codd published the paper entitled *A Relational Model of Data for Large Shared Data Banks*, in June 1970 in *Communications of the ACM*. The relational model proposed by Codd is now accepted as the definitive model for relational database management systems. The computer language, *Structured English Query Language* (SEQUEL) was developed by IBM to use Codd's model. SEQUEL later became SQL. In 1979, Relational Software (now Oracle) introduced the first commercially available implementation of SQL. In these days, SQL is accepted as the standard RDBMS language.

## 1.3 Benefits of SQL

There are several benefits of SQL as discussed below.

*SQL appears as a standard:* First standard for SQL was made in 1986 by ANSI (American National Standards Institute) and ISO (International Standards Organization) followed in 1987. The latest standard part was made in 2016.

*Portable:* SQL runs on mainframes, PCs, laptops, servers, tablets and smartphones. It runs in local systems, intranet and internet. Databases using SQL can be moved from device to another without problems.

*Used by open source database systems:* There are a number of free RDBMS such as MySQL, MariaDB and PostGres. They all support SQL. A large number of people are engaged in open source software development.

*Used by major DBMS system vendors:* SQL is used by major companies such as IBM, Oracle and Microsoft in their DBMSs.

*Easy to understand:* SQL consists mainly of English statements and making it easy to learn and write SQL queries.

---

*Interactive language:* SQL can be used to talk to databases and get answers to complex questions in seconds.

*Programming language:* SQL can be used by programmers writing applications using databases. Most of the web applications use SQL.

*Complete language for a database:* SQL is used to create databases, manage security of a database. While its main use is for updating, retrieving and sharing data with multiple users.

*Multiple data views:* Different users of the database can view data in different ways.

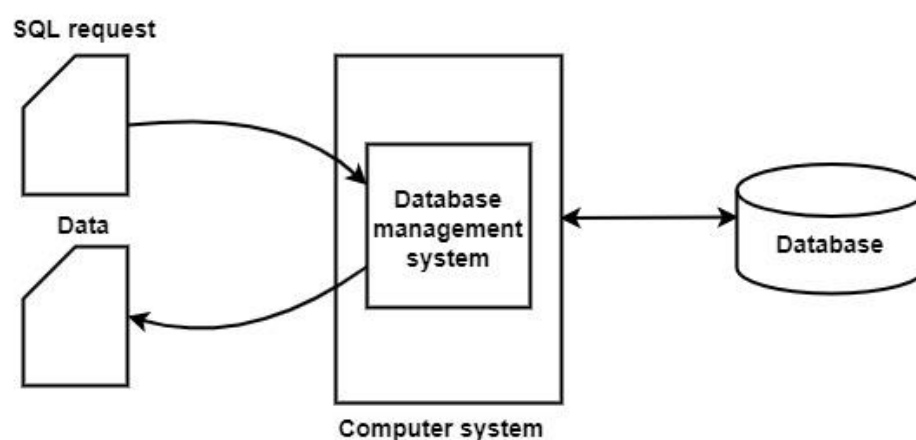
*Client/Server language:* SQL is used to connect front end computers (clients) and back end databases (servers). So, it supports the client-server architecture.

*Three tiered Internet architecture:* SQL can be used in three tiered Internet architecture. It includes a client, application server and a database.

*SQL is a desirable skill:* Many jobs including IT support, web development and business data analysis require skills in SQL.

#### 1.4 Database access

SQL serves as a tool for organizing, managing, and retrieving data stored in a computer database. SQL works with a specific type of database, called relational database. Fig.1.1 shows how SQL interacts with a database system. We discuss below some functions of SQL in the realm of data management.



**Figure 1.1:** Accessing database using SQL

*Data definition:* SQL allows a user to define the structure and organization of the stored data and relationships among the data items.

*Data retrieval:* SQL allows a user or an application program to retrieve data from a database and use it.

*Data manipulation:* SQL allows a user or an application program to update the database by adding new record, removing record, and modifying record.

*Access control:* SQL allows to restrict a user's ability to retrieve, add, and modify data, protecting stored data against unauthorized access.

---

*Data sharing:* SQL is used to coordinate data sharing by concurrent users, ensuring that they do not interfere with one another.

*Data integrity:* SQL allows to define integrity constraints in the database, and thus, protects database from corruption due to inconsistent updates or system failures.

### **1.5 SQL versus high level language**

There are differences between SQL and a high level language. Some important points are noted below.

- SQL is a database programming language, but a high level language is a general purpose programming language.
- SQL is not a structured language. Many high level language such as C, Pascal and Java are structured.
- A high level language is also known as third generation language, whereas, SQL is a fourth generation language. Some high level languages such as Java, serve as host languages. This allows SQL statements can be embedded in Java. As a result, it enhances data accessing capabilities even working with Java.

### **1.6 SQL for business requirements**

Incorporation of SQL in programming activities affects a business significantly.

*Faster IT response to requirements:* Programmers can concentrate on delivering business logic and are relieved from the responsibilities of implementing relational data processing. It results in a reduction in the amount of code that developers have to write.

*Reduced development / maintenance costs:* As we move the common business rules and processes into the database engine, it results in a smaller amount of software code to write and maintain. Some business rules and processes can be reused, instead of being recreated as new applications and business functions.

*Improved performance and scalability:* Performance of the applications can be improved overall because the database engine is doing the processing at a lower level in the system than the original application logic. Most SQL databases are vertically scalable, which means that you can increase the load on a single server by increasing components like RAM, SSD, or CPU.

*Sensitive data protection:* Sensitive data are more easily protected with the help of encryption using the SQL *field procedure* support. This technology permits the deployment of column-level encryption solutions with minimal disruption to existing applications.

### **1.7 Initiatives by IBM**

SQL was originally invented by IBM researchers and it becomes a strategic product for IBM based on its flagship DB2 database. SQL support is available on all major IBM product families, from personal computers through midrange systems (AS/400 and RS/6000) to IBM mainframes running both the MVS and VM operating systems. IBM's commitment and broad support significantly helped the market acceptance of SQL.

### **1.8 Initiatives by Microsoft**

---

## Lecture 2

**Contents:** Relation; Relational model; Introduction to MySQL; Advantages of MySQL; Some applications of MySQL; Some remarks; Disadvantages of MySQL; Exercises

### 2.1 Relation

A  $n$ -ary relation over sets  $X_1, \dots, X_n$  is a subset of the Cartesian product  $X_1 \times \dots \times X_n$ . Consider the following example.

$$X_1 = \{Ramu, Geeta, Arshad\}, X_2 = \{likes, dislikes\}, X_3 = \{dog, apple\}$$

Now,  $X_1 \times X_2 \times X_3 = \{(Ramu, likes, dog), (Ramu, likes, apple), (Ramu, dislikes, dog), (Ramu, dislikes, apple), (Geeta, likes, dog), (Geeta, likes, apple), (Geeta, dislikes, dog), (Geeta, dislikes, apple), (Arshad, likes, dog), (Arshad, likes, apple), (Arshad, dislikes, dog), (Arshad, dislikes, apple)\}$

Now, we take a subset of  $X_1 \times X_2 \times X_3$  as *RamuData*, where

$$RamuData = \{(Ramu, likes, dog), (Ramu, likes, apple), (Ramu, dislikes, dog), (Ramu, dislikes, apple)\}$$

*RamuData* is a subset of  $X_1 \times X_2 \times X_3$ . So, *RamuData* is a relation or, table. We present *RamuData* as row-by-column format, i.e., table format as given below.

Ramu	likes	dog
Ramu	likes	apple
Ramu	dislikes	dog
Ramu	dislikes	apple

With column names, the above table can be shown as follows.

Name	Taste	Item
Ramu	likes	dog
Ramu	likes	apple
Ramu	dislikes	dog
Ramu	dislikes	apple

**Table 2.1:** *RamuData*

Table *RamuData* is a relation. In particular,  $X_1 \times X_2 \times X_3$  (whole dataset) and  $\phi$  (null dataset) are also relations. Each relation is given a name in the database for identification.

### 2.2 Relational model

#### 2.2.1 Characteristics of relations

Some characteristics of a relation are given below. These characteristics can be obtained from Codd's rules as stated in the next section.

- 
- \* The tuples in a relation do not have any particular order. Although tuples can be ordered logically based on certain attribute(s).
  - \* An  $n$ -tuple is an ordered list of  $n$  values. However, at a logical level, the order of attributes and their values is not that important as long as the correspondence between attributes and values is maintained.
  - \* Each value in a tuple is an atomic value. It is not divisible into components within the framework of the basic relational model. Thus, composite and multivalued attributes are not allowed.
  - \* An important concept is that of null values. Null value is used to represent the value of attribute that may be unknown or may not apply to a tuple.
  - \* Some relations may represent facts about entities, whereas other relations may represent facts about relationships.
  - \* The relation schema can be interpreted as a declaration or a type of assertion.

### 2.2.2 Codd's rules

Twelve rules are cited below as part of a test to determine whether a product that is claimed to be fully relational. Vendors in their advertisements and manuals have translated the term "minimally relational" to "fully relational". Codd's rules for relational database systems is actually a collection of thirteen rules as stated below. (Wikipedia)

#### Rule 0: *The foundation rule*

For any system that is advertised as, or claimed to be, a relational database management system, that system must be able to manage databases entirely through its relational capabilities.

#### Rule 1: *The information rule*

All information in a relational data base is represented explicitly at the logical level and in exactly one way – by values in tables.

#### Rule 2: *The guaranteed access rule*

Each and every datum (atomic value) in a relational database is guaranteed to be logically accessible by resorting to a combination of table name, primary key value and column name.

#### Rule 3: *Systematic treatment of null values*

Null values (distinct from the empty character string or a string of blank characters and distinct from zero or any other number) are supported in fully relational DBMS for representing missing information and inapplicable information in a systematic way, independent of data type.

#### Rule 4: *Dynamic online catalog based on the relational model*

The database description is represented at the logical level in the same way as ordinary data, so that authorized users can apply the same relational language to its interrogation as they apply to the regular data.

#### Rule 5: *The comprehensive data sublanguage rule*

A relational system may support several languages and various modes of terminal use (for example, the fill-in-the-blanks mode). However, there must be at least one language whose statements are

---

expressible, per some well-defined syntax, as character strings and that is comprehensive in supporting all of the following items:

1. Data definition
2. View definition
3. Data manipulation (interactive and by program)
4. Integrity constraints
5. Authorization
6. Transaction boundaries (begin, commit and rollback)

Rule 6: *The view updating rule*

All views that are theoretically updatable are also updatable by the system.

Rule 7: *Possible for high-level insert, update, and delete*

The capability of handling a base relation or a derived relation as a single operand applies not only to the retrieval of data but also to the insertion, update and deletion of data.

Rule 8: *Physical data independence*

Application programs and terminal activities remain logically unimpaired whenever any changes are made in either storage representations or access methods.

Rule 9: *Logical data independence*

Application programs and terminal activities remain logically unimpaired when information-preserving changes of any kind that theoretically permit unimpairment are made to the base tables.

Rule 10: *Integrity independence*

Integrity constraints specific to a particular relational data base must be definable in the relational data sublanguage and storable in the catalog, not in the application programs.

Rule 11: *Distribution independence*

The end-user must not be able to see that the data is distributed over various locations. Users should always get the impression that the data is located at one site only.

Rule 12: *The nonsubversion rule*

If a relational system has a low-level (single-record-at-a-time) language, that low level cannot be used to subvert or bypass the integrity rules and constraints expressed in the higher level relational language (multiple-records-at-a-time).

## **2.3 Introduction to MySQL**

MySQL is a relational database management system (RDBMS). It is free, open-source and suitable for both small and large applications.

MySQL is written in C, C++ that runs on a variety of operating systems including Linux and Windows. MySQL was owned and sponsored by the Swedish company MySQL AB, which was bought by Sun Microsystems (now Oracle Corporation). Current version is 8.0.28 released in January, 2022.

---

## Lecture 3

**Contents:** Downloading MySQL Installer; Connecting to MySQL using shell; Connecting to MySQL server; Exercises

### 3.1 Downloading MySQL Installer

We can install MySQL on the Windows environment using MySQL installer. MySQL installer provides an easy-to-use wizard that helps us to install MySQL with the following components:

- MySQL Server
- All Available Connectors
- MySQL Workbench with Sample Data Models
- MySQL Notifier
- Tools for Excel and Microsoft Visual Studio
- MySQL Sample Databases
- MySQL Documentation

To install MySQL on Windows 11, follow the link: <https://allthings.how/how-to-install-mysql-on-windows-11/>

For other versions of an operating system, search using Google to find appropriate link providing guidelines of installation.

The current version of MySQL is 8.0 as of preparing the lecture material. We have found three types of content in MySQL:

- MySQL server, connectors, workbench, sample databases, documentation and other MySQL tools
- Tools for excel and Microsoft visual studio
- Python

### 3.2 Connecting to MySQL using shell

mysql has a command-line client program that allows you to interact with MySQL in the interactive and non-interactive mode. One needs to go the appropriate path to execute the shell program.

At the top of Fig.3.1, see the path

```
C:\Program Files\MySQL\MySQL Shell 8.0\bin\mysqlsh.exe
```

MySQL Shell open without connecting to a server. By default, Shell remains in JavaScript mode. One can change the mode using the `\sql`, `\py`, and `\js` commands.

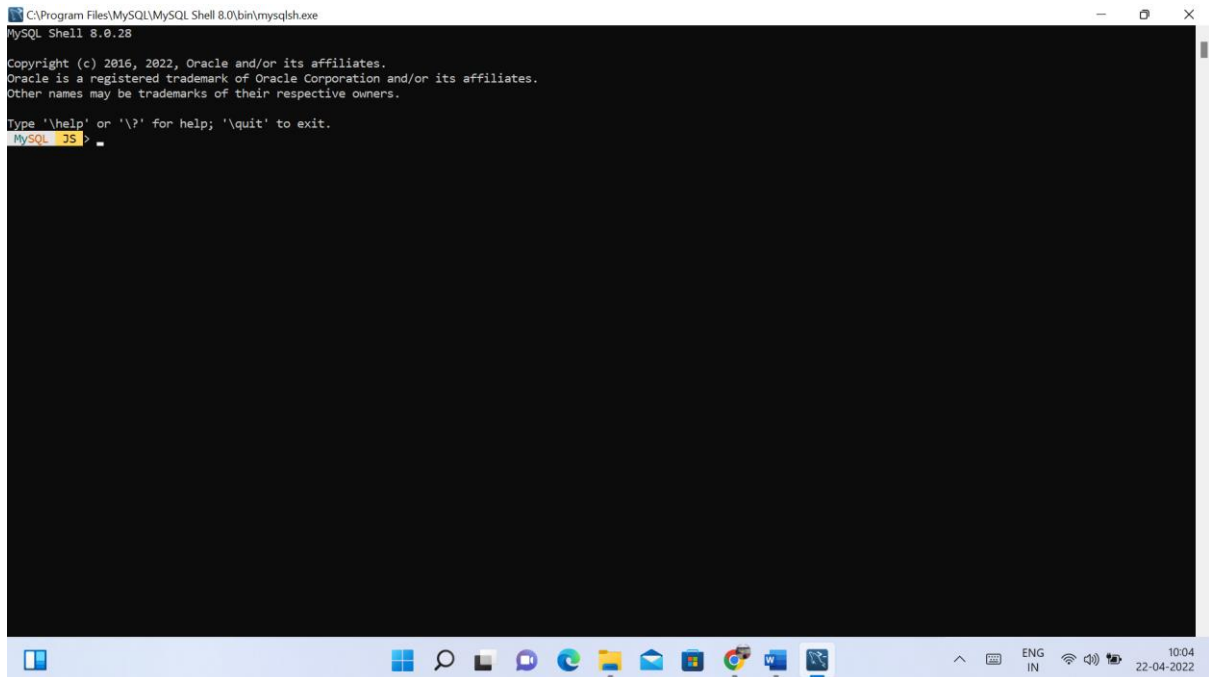


Figure 3.1: First screen of shell

One can see the status of Shell using the following command: `\s`

Shell status can also be obtained by executing the command `shell.status()`

See Fig. 3.2.

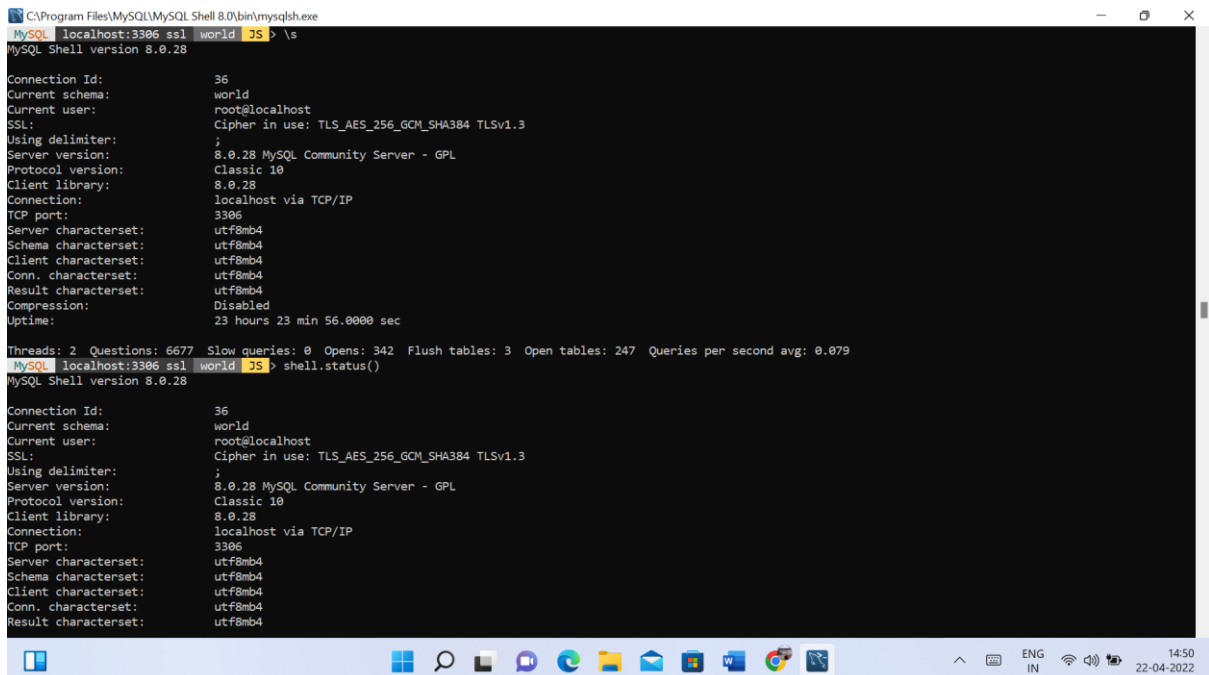


Figure 3.2: Shell status



An interactive session using shell is given in Fig. 3.3.

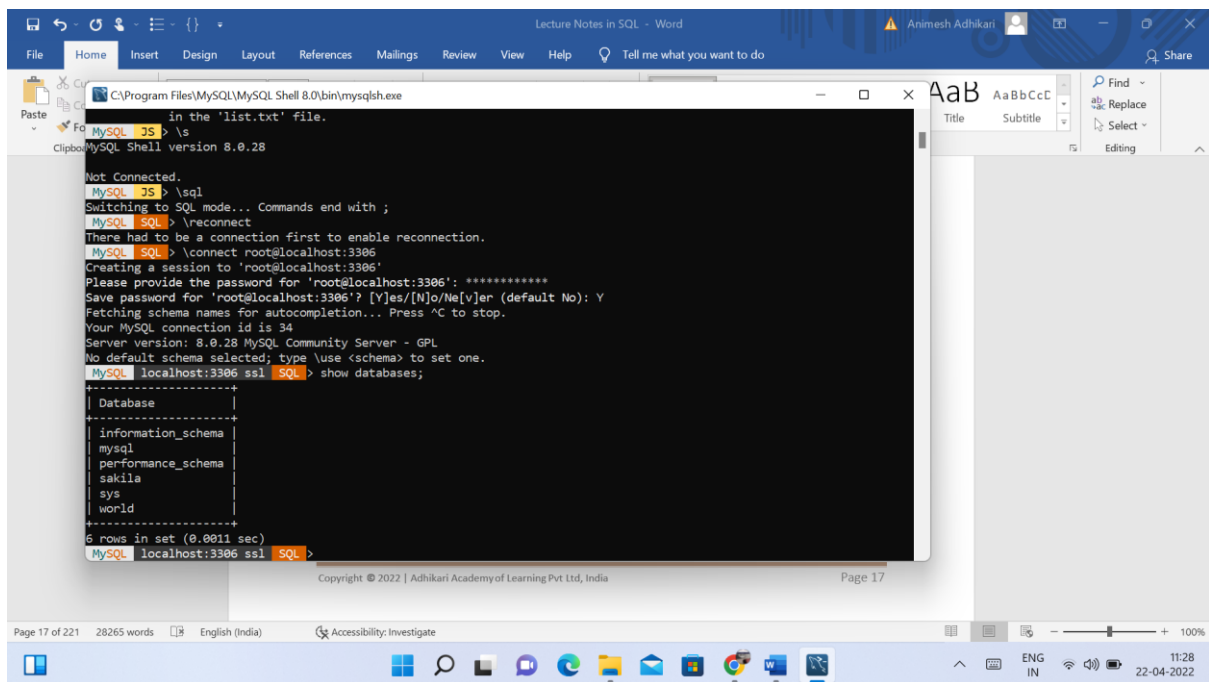


Figure 3.3: An interactive shell session

For more help, follow the link: <https://dev.mysql.com/doc/mysql-shell/8.0/en/mysql-shell-commands.html>

### 3.3 Connecting to MySQL server

Use command `\connect root@localhost:3306` to connect to MySQL Server.  
MySQL default port is 3306.

The password used here is the root password, given during installation of MySQL.

To see the available databases, execute command `show databases;`

To use database `world`, execute `use world;`

To display tables in `world` database, execute `show tables;`

To show full tables, use the command: `show full tables;`

The system shows column `Table_type` along with table name, for each table.

To display the structure of table `city`, use the following command: `describe city;`

An interactive session is shown in Fig. 3.4.