**WEEK 1**

1. **Introduce about DB**
2. **Introduce about DBMS**
3. **Introduce about RDBMS**
4. **Introduce about SQL**
5. **Introduce about MySQL**
6. **MySQL data types**
7. **SQL Commands**

**What is DBMS**

A database is an organized collection of data. It is the collection of schemas, tables, queries, reports, views, and other objects. The data are typically organized to model aspects of reality in a way that supports processes requiring information, such as modelling the availability of rooms in hotels in a way that supports finding a hotel with vacancies.

A database management system (DBMS) is a computer software application that interacts with the user, other applications, and the database itself to capture and analyze data. A general-purpose DBMS is designed to allow the definition, creation, querying, update, and administration of databases. Well-known DBMSs include MySQL, PostgreSQL, MongoDB, Microsoft SQL Server, Oracle, Sybase, SAP HANA, and IBM DB2. A database is not generally portable across different DBMSs, but different DBMS can interoperate by using standards such as SQL and ODBC or JDBC to allow a single application to work with more than one DBMS. Database management systems are often classified according to the database model that they support; the most popular database systems since the 1980s have all supported the relational model as represented by the SQL language.[disputed – discuss] Sometimes a DBMS is loosely referred to as a 'database'.

## Terminology and overview

Formally, a "database" refers to a set of related data and the way it is organized. Access to this data is usually provided by a "database management system" (DBMS) consisting of an integrated set of computer software that allows users to interact with one or more databases and provides access to all of the data contained in the database (although restrictions may exist that limit access to particular data). The DBMS provides various functions that allow entry, storage and retrieval of large quantities of information and provides ways to manage how that information is organized.

Because of the close relationship between them, the term "database" is often used casually to refer to both a database and the DBMS used to manipulate it.Outside the world of professional information technology, the term database is often used to refer to any collection of related data (such as a spreadsheet or a card index). This article is concerned only with databases where the size and usage requirements necessitate use of a database management system.Existing DBMSs provide various functions that allow management of a database and its data which can be classified into four main functional groups:

**Data definition** – Creation, modification and removal of definitions that define the organization of the data.

**Update** – Insertion, modification, and deletion of the actual data.

**Retrieval** – Providing information in a form directly usable or for further processing by other applications. The retrieved data may be made available in a form basically the same as it is stored in the database or in a new form obtained by altering or combining existing data from the database.

**Administration** – Registering and monitoring users, enforcing data security, monitoring performance, maintaining data integrity, dealing with concurrency control, and recovering information that has been corrupted by some event such as an unexpected system failure.

Both a database and its DBMS conform to the principles of a particular database model. "Database system" refers collectively to the database model, database management system, and database.Physically, database servers are dedicated computers that hold the actual databases and run only the DBMS and related software. Database servers are usually multiprocessor computers, with generous memory and RAID disk arrays used for stable storage. RAID is used for recovery of data if any of the disks fail. Hardware database accelerators, connected to one or more servers via a high-speed channel, are also used in large volume transaction processing environments. DBMSs are found at the heart of most database applications. DBMSs may be built around a custom multitasking kernel with built-in networking support, but modern DBMSs typically rely on a standard operating system to provide these functions from databases before the inception of Structured Query Language (SQL). The data recovered was disparate, redundant and disorderly, since there was no proper method to fetch it and arrange it in a concrete structure.Since DBMSs comprise a significant economical market, computer and storage vendors often take into account DBMS requirements in their own development plans.Databases and DBMSs can be categorized according to the database model(s) that they support (such as relational or XML), the type(s) of computer they run on (from a server cluster to a mobile phone), the query language(s) used to access the database (such as SQL or XQuery), and their internal engineering, which affects performance, scalability, resilience, and security.

## Applications of DBMS

Databases are used to support internal operations of organizations and to underpin online interactions with customers and suppliers .Databases are used to hold administrative information and more specialized data, such as engineering data or economic models. Examples of database applications include computerized library systems, flight reservation systems, computerized parts inventory systems, and many content management systems that store websites as collections of webpages in a database.

**Short description about RDBMS**

A **relational database management system** (**RDBMS**) is a [database management system](https://en.wikipedia.org/wiki/Database_management_system) (DBMS) that is based on the [relational model](https://en.wikipedia.org/wiki/Relational_model) as invented by [E. F. Codd](https://en.wikipedia.org/wiki/Edgar_F._Codd), of IBM's [San Jose Research Laboratory](https://en.wikipedia.org/wiki/IBM_Almaden_Research_Center). In 2016, many of the databases in widespread use are based on the [relational database](https://en.wikipedia.org/wiki/Relational_database) model.

RDBMSs have been a common choice for the storage of information in new databases used for financial records, manufacturing and logistical information, personnel data, and other applications since the 1980s. Relational databases have often replaced legacy [hierarchical databases](https://en.wikipedia.org/wiki/Hierarchical_database) and [network databases](https://en.wikipedia.org/wiki/Network_database) because they are easier to understand and use. However, relational databases have received unsuccessful challenge attempts by [object database](https://en.wikipedia.org/wiki/Object_database) management systems in the 1980s and 1990s (which were introduced trying to address the so-called [object-relational impedance mismatch](https://en.wikipedia.org/wiki/Object-relational_impedance_mismatch) between relational databases and object-oriented application programs) and also by [XML database](https://en.wikipedia.org/wiki/XML_database) management systems in the 1990s. Despite such attempts, RDBMSs keep most of the market share, which has also grown over the years.

**The Advantages of a Relational Database Management System**

A Relational Database Management System (RDBMS) is a software system that provides access to a relational database. The software system is a collection of software applications that can be used to create, maintain, manage and use the database. A "relational database" is a database structured on the "relational" model. Data are stored and presented in a tabular format, organized in rows and columns with one record per row.

### DATA STRUCTURE

The table format is simple and easy for database users to understand and use. RDBMSs provide data access using a natural structure and organization of the data. Database queries can search any column for matching entries.

### MULTI-USER ACCESS

RDBMSs allow multiple database users to access a database simultaneously. Built-in locking and transactions management functionality allow users to access data as it is being changed, prevents collisions between two users updating the data, and keeps users from accessing partially updated records.

### PRIVILEGES

Authorization and privilege control features in an RDBMS allow the database administrator to restrict access to authorized users, and grant privileges to individual users based on the types of database tasks they need to perform. Authorization can be defined based on the remote client IP address in combination with user authorization, restricting access to specific external computer systems.

### NETWORK ACCESS

RDBMSs provide access to the database through a server daemon, a specialized software program that listens for requests on a network, and allows database clients to connect to and use the database. Users do not need to be able to log in to the physical computer system to use the database, providing convenience for the users and a layer of security for the database. Network access allows developers to build desktop tools and Web applications to interact with databases.

### SPEED

The relational database model is not the fastest data structure. RDBMS advantages, such as simplicity, make the slower speed a fair trade-off. Optimizations built into an RDBMS, and the design of the databases, enhance performance, allowing RDBMSs to perform more than fast enough for most applications and data sets. Improvements in technology, increasing processor speeds and decreasing memory and storage costs allow systems administrators to build incredibly fast systems that can overcome any database performance shortcomings.

### MAINTENANCE

RDBMSs feature maintenance utilities that provide database administrators with tools to easily maintain, test, repair and back up the databases housed in the system. Many of the functions can be automated using built-in automation in the RDBMS, or automation tools available on the operating system.

### LANGUAGE

RDBMSs support a generic language called "Structured Query Language" (SQL). The SQL syntax is simple, and the language uses standard English language keywords and phrasing, making it fairly intuitive and easy to learn. Many RDBMSs add non-SQL, database-specific keywords, functions and features to the SQL language.

**List of**[**relational database management systems**](https://en.wikipedia.org/wiki/Relational_database_management_system):

According to DB-Engines, the most popular systems are Oracle, MySQL, Microsoft SQL Server, PostgreSQL and IBM DB2.

Some more examples

* [4th Dimension](https://en.wikipedia.org/wiki/4th_Dimension_(Software))
* [Adabas D](https://en.wikipedia.org/wiki/Adabas_D)
* [Alpha Five](https://en.wikipedia.org/wiki/Alpha_Five_(database))
* [Apache Derby](https://en.wikipedia.org/wiki/Apache_Derby)
* [Aster Data](https://en.wikipedia.org/wiki/Aster_Data_Systems)
* [Amazon Aurora](https://en.wikipedia.org/wiki/Amazon_Relational_Database_Service)

**Applications of RDBMS**

Inventory, Supply chain management, Hospitals, Schools, Library, Data Mining and Data Warehouses there are scores of places where data and historical data are important.

**(Introduction to SQL)What is SQL?**

SQL stands for Structured Query Language

SQL lets you access and manipulate databases

SQL is an ANSI (American National Standards Institute) standard

**What Can SQL do?**

* SQL can execute queries against a database
* SQL can retrieve data from a database
* SQL can insert records in a database
* SQL can update records in a database
* SQL can delete records from a database
* SQL can create new databases
* SQL can create new tables in a database
* SQL can create stored procedures in a database
* SQL can create views in a database
* SQL can set permissions on tables, procedures, and views

Although SQL is an ANSI (American National Standards Institute) standard, there are many different versions of the SQL language.However, to be compliant with the ANSI standard, they all support at least the major commands (such as SELECT, UPDATE, DELETE, INSERT, WHERE) in a similar manner.

# MySQL

MySQL (officially pronounced as /maɪ ˌɛskjuːˈɛl/ "My S-Q-L") is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius' daughter, and "SQL", the abbreviation for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation. For proprietary use, several paid editions are available, and offer additional functionality.

MySQL is a central component of the LAMP open-source web application software stack (and other "AMP" stacks). LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python". Applications that use the MySQL database include: TYPO3, MODx, Joomla, WordPress, phpBB, MyBB, and Drupal. MySQL is also used in many high-profile, large-scale websites, including Google (though not for searches), Facebook, Twitter, Flickr,and YouTube.

MySQL is written in C and C++. Its SQL parser is written in yacc, but it uses a home-brewed lexical analyzer. MySQL works on many system platforms, including AIX, BSDi, FreeBSD, HP-UX, eComStation, i5/OS, IRIX, Linux, OS X, Microsoft Windows, NetBSD, Novell NetWare, OpenBSD, OpenSolaris, OS/2 Warp, QNX, Oracle Solaris, Symbian, SunOS, SCO OpenServer, SCO UnixWare, Sanos and Tru64. A port of MySQL to OpenVMS also exists.

**What is MySQL?**

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation.

**MySQL is a database management system.**

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

**MySQL databases are relational.**

A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one-to-one, one-to-many, unique, required or optional, and “pointers” between different tables. The database enforces these rules, so that with a well-designed database, your application never sees inconsistent, duplicate, orphan, out-of-date, or missing data.

The SQL part of “MySQL” stands for “Structured Query Language”. SQL is the most common standardized language used to access databases. Depending on your programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax.

SQL is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. In this manual, “SQL-92” refers to the standard released in 1992, “SQL:1999” refers to the standard released in 1999, and “SQL:2003” refers to the current version of the standard. We use the phrase “the SQL standard” to mean the current version of the SQL Standard at any time.

**MySQL software is Open Source.**

Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL (GNU General Public License), http://www.fsf.org/licenses/, to define what you may and may not do with the software in different situations. If you feel uncomfortable with the GPL or need to embed MySQL code into a commercial application, you can buy a commercially licensed version from us. See the MySQL Licensing Overview for more information (http://www.mysql.com/company/legal/licensing/).

**The MySQL Database Server is very fast, reliable, scalable, and easy to use.**

If that is what you are looking for, you should give it a try. MySQL Server can run comfortably on a desktop or laptop, alongside your other applications, web servers, and so on, requiring little or no attention. If you dedicate an entire machine to MySQL, you can adjust the settings to take advantage of all the memory, CPU power, and I/O capacity available. MySQL can also scale up to clusters of machines, networked together.

MySQL Server was originally developed to handle large databases much faster than existing solutions and has been successfully used in highly demanding production environments for several years. Although under constant development, MySQL Server today offers a rich and useful set of functions. Its connectivity, speed, and security make MySQL Server highly suited for accessing databases on the Internet.

**MySQL Server works in client/server or embedded systems.**

The MySQL Database Software is a client/server system that consists of a multi-threaded SQL server that supports different backends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs).

We also provide MySQL Server as an embedded multi-threaded library that you can link into your application to get a smaller, faster, easier-to-manage standalone product.

**A large amount of contributed MySQL software is available.**

MySQL Server has a practical set of features developed in close cooperation with our users. It is very likely that your favorite application or language supports the MySQL Database Server.

The official way to pronounce “MySQL” is “My Ess Que Ell” (not “my sequel”), but we do not mind if you pronounce it as “my sequel” or in some other localized way.

# Top Reasons to Use MySQL

### 1. Scalability and Flexibility

The MySQL database server provides the ultimate in scalability, sporting the capacity to handle deeply embedded applications with a footprint of only 1MB to running massive data warehouses holding terabytes of information. Platform flexibility is a stalwart feature of MySQL with all flavors of Linux, UNIX, and Windows being supported. And, of course, the open source nature of MySQL allows complete customization for those wanting to add unique requirements to the database server.

### 2. High Performance

A unique storage-engine architecture allows database professionals to configure the MySQL database server specifically for particular applications, with the end result being amazing performance results. Whether the intended application is a high-speed transactional processing system or a high-volume web site that services a billion queries a day, MySQL can meet the most demanding performance expectations of any system. With high-speed load utilities, distinctive memory caches, full text indexes, and other performance-enhancing mechanisms, MySQL offers all the right ammunition for today's critical business systems.

### 3. High Availability

Rock-solid reliability and constant availability are hallmarks of MySQL, with customers relying on MySQL to guarantee around-the-clock uptime. MySQL offers a variety of high-availability options from high-speed master/slave replication configurations, to specialized Cluster servers offering instant failover, to third party vendors offering unique high-availability solutions for the MySQL database server.

### 4. Robust Transactional Support

MySQL offers one of the most powerful transactional database engines on the market. Features include complete ACID (atomic, consistent, isolated, durable) transaction support, unlimited row-level locking, distributed transaction capability, and multi-version transaction support where readers never block writers and vice-versa. Full data integrity is also assured through server-enforced referential integrity, specialized transaction isolation levels, and instant deadlock detection.

### 5. Web and Data Warehouse Strengths

MySQL is the de-facto standard for high-traffic web sites because of its high-performance query engine, tremendously fast data insert capability, and strong support for specialized web functions like fast full text searches. These same strengths also apply to data warehousing environments where MySQL scales up into the terabyte range for either single servers or scale-out architectures. Other features like main memory tables, B-tree and hash indexes, and compressed archive tables that reduce storage requirements by up to eighty-percent make MySQL a strong standout for both web and business intelligence applications.

### 6. Strong Data Protection

Because guarding the data assets of corporations is the number one job of database professionals, MySQL offers exceptional security features that ensure absolute data protection. In terms of database authentication, MySQL provides powerful mechanisms for ensuring only authorized users have entry to the database server, with the ability to block users down to the client machine level being possible. SSH and SSL support are also provided to ensure safe and secure connections. A granular object privilege framework is present so that users only see the data they should, and powerful data encryption and decryption functions ensure that sensitive data is protected from unauthorized viewing. Finally, backup and recovery utilities provided through MySQL and third party software vendors allow for complete logical and physical backup as well as full and point-in-time recovery.

### 7. Comprehensive Application Development

One of the reasons MySQL is the world's most popular open source database is that it provides comprehensive support for every application development need. Within the database, support can be found for stored procedures, triggers, functions, views, cursors, ANSI-standard SQL, and more. For embedded applications, plug-in libraries are available to embed MySQL database support into nearly any application. MySQL also provides connectors and drivers (ODBC, JDBC, etc.) that allow all forms of applications to make use of MySQL as a preferred data management server. It doesn't matter if it's PHP, Perl, Java, Visual Basic, or .NET, MySQL offers application developers everything they need to be successful in building database-driven information systems.

### 8. Management Ease

MySQL offers exceptional quick-start capability with the average time from software download to installation completion being less than fifteen minutes. This rule holds true whether the platform is Microsoft Windows, Linux, Macintosh, or UNIX. Once installed, self-management features like automatic space expansion, auto-restart, and dynamic configuration changes take much of the burden off already overworked database administrators. MySQL also provides a complete suite of graphical management and migration tools that allow a DBA to manage, troubleshoot, and control the operation of many MySQL servers from a single workstation. Many third party software vendor tools are also available for MySQL that handle tasks ranging from data design and ETL, to complete database administration, job management, and performance monitoring.

### 9. Open Source Freedom and 24 x 7 Support

Many corporations are hesitant to fully commit to open source software because they believe they can't get the type of support or professional service safety nets they currently rely on with proprietary software to ensure the overall success of their key applications. The questions of indemnification come up often as well. These worries can be put to rest with MySQL as complete around-the-clock support as well as indemnification is available through MySQL Enterprise. MySQL is not a typical open source project as all the software is owned and supported by Oracle, and because of this, a unique cost and support model are available that provides a unique combination of open source freedom and trusted software with support.

### 10. Lowest Total Cost of Ownership

By migrating current database-drive applications to MySQL, or using MySQL for new development projects, corporations are realizing cost savings that many times stretch into seven figures. Accomplished through the use of the MySQL database server and scale-out architectures that utilize low-cost commodity hardware, corporations are finding that they can achieve amazing levels of scalability and performance, all at a cost that is far less than those offered by proprietary and scale-up software vendors. In addition, the reliability and easy maintainability of MySQL means that database administrators don't waste time troubleshooting performance or downtime issues, but instead can concentrate on making a positive impact on higher level tasks that involve the business side of data.

**Difference between DBMS and RDBMS**

**DBMS**

1)A DBMS has to be persistent (it should be accessible when the program created the data donot exist or even the application that created the data restarted).

2) DBMS has to provide some uniform methods independent of a specific application for accessing the information that is stored.

3)DBMS does not impose any constraints or security with regard to data manipulation. It is user or the programmer responsibility to ensure the ACID PROPERTY of the database

4)In DBMS Normalization process will not be present

5)In dbms no relationship concept

6)It supports Single User only

7)It treats Data as Files internally

8)It supports 3 rules of E.F.CODD out off 12 rules

9)It requires low Software and Hardware Requirements.

FoxPro, IMS are Examples

**RDBMS:**

1)RDBMS is based on relational model, in which data is represented in the form of relations, with enforced relationships between the tables.

2)RDBMS defines the integrity constraint for the purpose of holding ACID PROPERTY.

3)In RDBMS, normalization process will be present to check the database table consistency

4)RDBMS helps in recovery of the database in case of loss of data

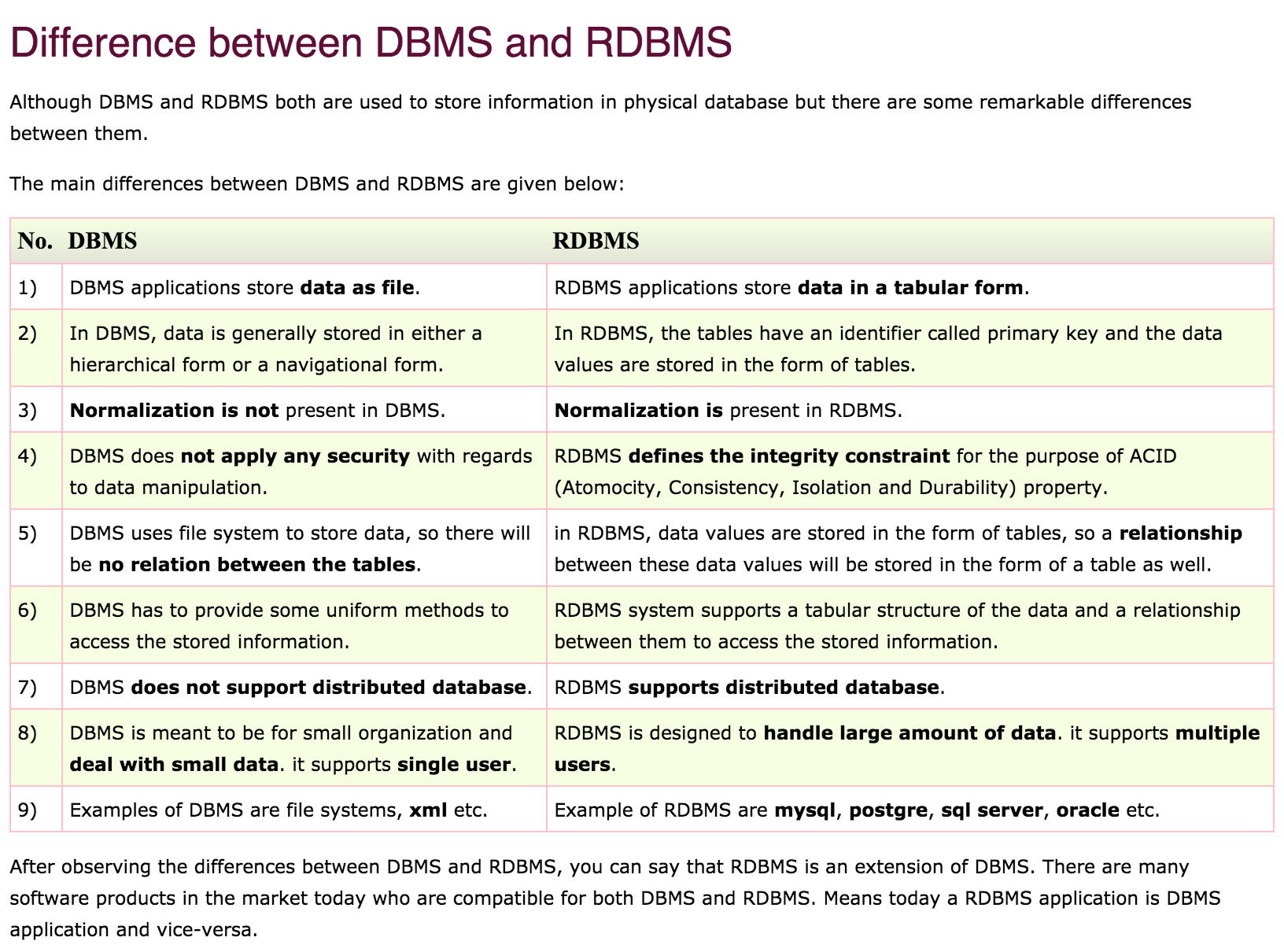
5)It is used to establish the relationship concept between two database objects, i.e, tables

6)It supports multiple users

7)It treats data as Tables internally

8)It supports minimum 6 rules of E.F.CODD

9)It requires High software and hardware



MySQL Data Types

In MySQL there are three main types: text, number, and Date/Time types.

**Text types:**

|  |  |
| --- | --- |
| **Data type** | **Description** |
| CHAR(size) | Holds a fixed length string (can contain letters, numbers, and special characters). The fixed size is specified in parenthesis. Can store up to 255 characters |
| VARCHAR(size) | Holds a variable length string (can contain letters, numbers, and special characters). The maximum size is specified in parenthesis. Can store up to 255 characters. **Note:** If you put a greater value than 255 it will be converted to a TEXT type |
| TINYTEXT | Holds a string with a maximum length of 255 characters |
| TEXT | Holds a string with a maximum length of 65,535 characters |
| BLOB | For BLOBs (Binary Large OBjects). Holds up to 65,535 bytes of data |
| MEDIUMTEXT | Holds a string with a maximum length of 16,777,215 characters |
| MEDIUMBLOB | For BLOBs (Binary Large OBjects). Holds up to 16,777,215 bytes of data |
| LONGTEXT | Holds a string with a maximum length of 4,294,967,295 characters |
| LONGBLOB | For BLOBs (Binary Large OBjects). Holds up to 4,294,967,295 bytes of data |
| ENUM(x,y,z,etc.) | Let you enter a list of possible values. You can list up to 65535 values in an ENUM list. If a value is inserted that is not in the list, a blank value will be inserted.**Note:** The values are sorted in the order you enter them.You enter the possible values in this format: ENUM('X','Y','Z') |
| SET | Similar to ENUM except that SET may contain up to 64 list items and can store more than one choice |

**Number types:**

|  |  |
| --- | --- |
| **Data type** | **Description** |
| TINYINT(size) | -128 to 127 normal. 0 to 255 UNSIGNED\*. The maximum number of digits may be specified in parenthesis |
| SMALLINT(size) | -32768 to 32767 normal. 0 to 65535 UNSIGNED\*. The maximum number of digits may be specified in parenthesis |
| MEDIUMINT(size) | -8388608 to 8388607 normal. 0 to 16777215 UNSIGNED\*. The maximum number of digits may be specified in parenthesis |
| INT(size) | -2147483648 to 2147483647 normal. 0 to 4294967295 UNSIGNED\*. The maximum number of digits may be specified in parenthesis |
| BIGINT(size) | -9223372036854775808 to 9223372036854775807 normal. 0 to 18446744073709551615 UNSIGNED\*. The maximum number of digits may be specified in parenthesis |
| FLOAT(size,d) | A small number with a floating decimal point. The maximum number of digits may be specified in the size parameter. The maximum number of digits to the right of the decimal point is specified in the d parameter |
| DOUBLE(size,d) | A large number with a floating decimal point. The maximum number of digits may be specified in the size parameter. The maximum number of digits to the right of the decimal point is specified in the d parameter |
| DECIMAL(size,d) | A DOUBLE stored as a string , allowing for a fixed decimal point. The maximum number of digits may be specified in the size parameter. The maximum number of digits to the right of the decimal point is specified in the d parameter |

\*The integer types have an extra option called UNSIGNED. Normally, the integer goes from an negative to positive value. Adding the UNSIGNED attribute will move that range up so it starts at zero instead of a negative number.

**Date types:**

|  |  |
| --- | --- |
| **Data type** | **Description** |
| DATE() | A date. Format: YYYY-MM-DD  **Note:** The supported range is from '1000-01-01' to '9999-12-31' |
| DATETIME() | \*A date and time combination. Format: YYYY-MM-DD HH:MM:SS  **Note:** The supported range is from '1000-01-01 00:00:00' to '9999-12-31 23:59:59' |
| TIMESTAMP() | \*A timestamp. TIMESTAMP values are stored as the number of seconds since the Unix epoch ('1970-01-01 00:00:00' UTC). Format: YYYY-MM-DD HH:MM:SS  **Note:** The supported range is from '1970-01-01 00:00:01' UTC to '2038-01-09 03:14:07' UTC |
| TIME() | A time. Format: HH:MM:SS  **Note:** The supported range is from '-838:59:59' to '838:59:59' |
| YEAR() | A year in two-digit or four-digit format.  **Note:** Values allowed in four-digit format: 1901 to 2155. Values allowed in two-digit format: 70 to 69, representing years from 1970 to 2069 |

\*Even if DATETIME and TIMESTAMP return the same format, they work very differently. In an INSERT or UPDATE query, the TIMESTAMP automatically set itself to the current date and time. TIMESTAMP also accepts various formats, like YYYYMMDDHHMMSS, YYMMDDHHMMSS, YYYYMMDD, or YYMMDD.

**SQL Commands:**

**SQL commands** are instructions, coded into SQL statements, which are used to communicate with the database to perform specific tasks, work, functions and queries with data.

SQL commands can be used not only for searching the database but also to perform various other functions like, for example, you can [*create tables*](http://beginner-sql-tutorial.com/sql-create-statement.htm), add data to tables, or modify data, drop the table, set permissions for users. SQL commands are grouped into four major categories depending on their functionality:

* **Data Definition Language (DDL)** - These SQL commands are used for creating, modifying, and dropping the structure of database objects.

The commands are

CREATE DATABASE - creates a new database

ALTER DATABASE - modifies a database

CREATE TABLE - creates a new table

ALTER TABLE - modifies a table

DROP TABLE - deletes a table

CREATE INDEX - creates an index (search key)

DROP INDEX - deletes an index

* **Data Manipulation Language (DML)** - These SQL commands are used for storing, retrieving, modifying, and deleting data.

These Data Manipulation Language commands are:

* SELECT - extracts data from a database
* UPDATE - updates data in a database
* DELETE - deletes data from a database
* INSERT INTO - inserts new data into a database
* **Transaction Control Language (TCL)** - These SQL commands are used for managing changes affecting the data.

These commands are

* COMMIT-to save the changes.
* ROLLBACK-to roll back the changes.
* SAVEPOINT-creates points within groups of transactions in which to ROLLBACK
* **Data Control Language (DCL)** - These SQL commands are used for providing security to database objects.

These commands are

* GRANT-used to provide access or privileges on the database objects to the users.
* REVOKE-removes user access rights or privileges to the database objects.