DBMS vs RDBMS

DBMS (e.g. MS Excel, Foxpro, etc)

1. Naming conventions (Nomenclatures) are different
2. Relationship between 2 files is maintained programmatically
3. More programming
4. More time is required for s/w development
5. In DBMS we have high network traffic
6. Processing on Client machine
7. Slow and expensive
8. File level Locking
9. Not suitable for multi-user
10. Distributed databases are not supported
11. Client-server architecture is not supported
12. No security of data

RDBMS (e.g. Oracle, MySQL, etc.)

1. Naming conventions (Nomenclatures) are different
2. Relationship between 2 tables can be specified at the time of table creation
   (e.g. foreign key constraints)
3. Less programming
4. Less time is required for s/w development
5. In DBMS we have low network traffic
6. Processing on Server machine (knows as client server architecture)
7. Faster and Cheaper (in terms of hardware, infrastructure)
8. Row level locking (Internally, table is not a file; internally, every row is a file)
9. Suitable for multi-user
10. Most of the RDBMS support distributed databases
11. Most of the RDBMS's support client-server architecture
12. Multiple levels of security (Security is in-build feature of RDBMS)
    (Does not allow access to the table through the OS)
    a. logging in security (MySQL database username/password)
    b. command level security (Permisiion to give MySQL commands
    create table, create procedure, create user, etc.)
    c. object level security (Permission to access tables and other objects of other users)

Various RDBMS

Informix

- fastest in terms of processing speed
- programming has to be in Assembly

Oracle
- Most Popular - because programming is very easy
- Largest programs in oracle - 9 line
- Product of Oracle corporation (1977)
- #1 largest overall s/w company in the world
- #1 largest database s/w company in the world
- Larry Ellison - Founder of Oracle
- 63% of world commercial database market in the client-server environment
- 86% of world commercial database market in the Internet environment
- works on 130 os

Sybase
- going down
- recently acquired by SAP

MS SQL Server
- good RDBMS from Microsoft
- works only with Windows OS

Open Source Free RDBMS (character/text based) :-
- Ingres
- Postgres
- Unify

DB Server has to be a mainframe (super computer):-
- DB2 (good RDBMS from IBM)
- CICS
- TELON
- IDMS

Single user RDBMS (PC based)(cannont have server-client):-
- MS Access
- Paradox
- Vatcom SQL

MySQL
- was launched by a swedish company in 1995
- its name is a combination of "My", the name of co-founder Michael Widenius' daughter, and "SQL"
- MySQL is an open-source RDBMS
- most widely used open-source RDBMS
- part ofthe widely used LAMP open-source web application software
- free-software open-source projects that require a RDBMS often use MySQL
- Facebook, Twitter, Joomla, WordPress, Google(though for not searches), Flickr, youtube
- MySQL occupies of 42% of world open-source database market
- Sun Microsystems acquired MySQL in 2008
- Oracle acquired Sun Microsystems in 2010

Best s/w development tools from MySQL

tools provided by MySQL to make programming easy

MysQL database

- Store tables data, secure the database, etc

SQL

- Structures Query Language
- Create, Drop, Alter
- Inser, Update, Delete,
- Security - Grankt and revoke
- Select data - Select
- conforms to ANSI standards (e.g. 1 char occupies 1 Byte of storage)
- conforms to ISO standards (for QA)
- common for all RDBMS
- SQL was initially founded by IBM (1975-77)
- created using c, c++(90%) and assembly (10%)
- initially known as RQBE (Relational Query by Example)
- IBM gave RQBE free of cost to ANSI
- ANSI renamed RQBE to SQL
- SQL is now controlled by ANSI (hence SQL is common for all RDBMS)
- in 2005, source code of SQL is rewritten in Java (100%)

MySQL Command Line Client

- MySQL Client s/w
- used to connect to MysQL database, issue the SQL and MySQL commands, run MysQL-PL programs
- character/text based

MySQL Workbench

- MySQL Client s/w
- used to connect to MysQL database, issue the SQL and MySQL commands, run MysQL-PL programs
- GUI based

phpMyAdmin

- not a product of MySQL
- 3rd party product
- MySQL Client s/w
- used to connect to MysQL database, issue the SQL and MySQL commands, run MysQL-PL programs
- GUI based

MySQL-PL

- MySQL programming language
- programming language from MySQL
- used for database programming

e.g. HRA_CALC, TAX_CALC, ATTENDANCE_CALC, etc.

MySQL Connectors

- used for connecting other frontend software with MySQL database

- drivers for database connectivity
- JDBC, ODBC, Python, C, C++, etc.

MySQL for Excel
- import, export and edit MySQL data using MS Excel

MySQL Notifier
- startup and shutdown MySQL database
- normally the DBA' (Database Administrator) job

MySQL Enterprise Backup
- used to take backups of table data
- also to restore from backups
- normally the DBA' (Database Administrator) job
- used to transport the tables from one MySQL database into another MySQL database

MySQL Enterprise High Availability
- for Replication
- data has to available at all times
- maintain Primary and Secondary DB servers

MySQL Enterprise Encryption
- used to encrypt and decrypt the MySQL table over the network

MySQL Enterprise Monitor
- for Monitoring MySQL database server
- for performance planning, monitoring, tuning

MySQL Query Analyzer
- used for query training
etc.

SQL

- Structured Query Language
- commonly pronounced as "Sequel"
- conforms to ANSI adn ISO standard
- common for all RDBMS

4 sub-divisions of SQL -

- DDL (Data Definition Language)
- DML (Data Manipulation Language)
- DCL (Data Control Language)
- DQL (Data Quary Language)

DDL :

- Data Definition Language
- DDL or Data Definition Language actually consists of the SQL commands that can be used to define the database schema. It simply deals with descriptions of the database schema and is used to create and modify the structure of database objects in the database.DDL is a set of SQL commands used to create, modify, and delete database structures but not data. These commands are normally not used by a general user, who should be accessing the database via an application.
- CREATE
  This command is used to create the database or its objects (like table, index, function, views, store procedure, and triggers).
- DROP
  This command is used to delete objects from the database.
- ALTER
  This is used to alter the structure of the database.

DML :

- Data Manipulation Language
- The SQL commands that deals with the manipulation of data present in the database belong to DML or Data Manipulation Language and this includes most of the SQL statements. It is the component of the SQL statement that controls access to data and to the database. Basically, DCL statements are grouped with DML statements.
- INSERT
  It is used to insert data into a table.
- UPDATE
  It is used to update existing data within a table.
- DELETE
  It is used to delete records from a database table.

DCL :

- Data Control Language
- DCL includes commands such as GRANT and REVOKE which mainly deal with the rights, permissions, and other controls of the database system.
- GRANT
  This command gives users access privileges to the database.
- REVOKE
  This command withdraws the user's access privileges given by using the GRANT command.

DQL :

- Data Quary Language
- DQL statements are used for performing queries on the data within schema objects. The purpose of the DQL Command is to get some schema relation based on the query passed to it. We can define DQL as follows it is a component of SQL statement that allows getting data from the database and imposing order upon it. It includes the SELECT statement. This command allows getting the data out of the database to perform operations with it. When a SELECT is

fired against a table or tables the result is compiled into a further temporary table, which is displayed or perhaps received by the program i.e. a front-end.

- SELECT
  It is used to retrieve data from the database.

Extra in MySQL and Oracle RDBMS

- Non an ANSI standard
  5th component of SQL -
  DTL/TCL :
- Data Transaction Language / Transaction Control Language
- COMMIT
  Commits a Transaction.
- ROLLBACK
  Rollbacks a transaction in case of any error occurs.
- SAVEPOINT
  Sets a savepoint within a transaction.

DDL :

- RENAME
  This is used to rename an object existing in the database.
- TRUNCATE
  This is used to remove all records from a table, including all spaces allocated for the records are removed.

Extra in oracle RDBMS only:-
DML :

- MERGE
- UPSERT

Rules for table names, column names, variable names -
- Maximum 30 characters
- A-Z, a-z, 0-9 allowed
- $ # _ allowrd
- case-insensitive
- has to begin with an alphabet
EMP2022 ← allowed
2022EMP ← error
- in MySQL, to use reserved character such as # in tablename and columnname, enclose it in backquotes
e.g., `EMP#` → tablename using #
- 134 Reserved words not allowed in tablename

MySQL Datatypes -

String data types

The `CHAR` and `VARCHAR` types are declared with a length that indicates the maximum number of characters you want to store.

The CHAR and VARCHAR Types

CHAR

The length of a `CHAR` column is fixed to the length that you declare when you create the table. The length can be any value from 0 to 255. When `CHAR` values are stored, they are right-padded with spaces to the specified length. When `CHAR` values are retrieved, trailing spaces are removed unless the `PAD_CHAR_TO_FULL_LENGTH` SQL mode is enabled.
- allows any chracter
- max upto 255 chracters
- default width 1
- fixed length
- wastage of harddisk space
- searching and retrival will be very fast
- e.g., PAN_NO, ROLL_NO

VARCHAR

Values in `VARCHAR` columns are variable-length strings. The length can be specified as a value from 0 to 65,535. The effective maximum length of a `VARCHAR` is subject to the maximum row size (65,535 bytes, which is shared among all columns) and the character set used.
- allows any character
- max upto 65,535 characters (64kb - 1)
- width has to be specified
- variable length
- conserve on Harddisk space
- searching and retrival will be slow
- e.g., NAME, ADDRESS, CITY

The following table illustrates the differences between `CHAR` and `VARCHAR` by showing the result of storing various string values into `CHAR(4)` and `VARCHAR(4)` columns (assuming that the column uses a single-byte character set such as `latin1`).

Value	CHAR(4)	Storage Required	VARCHAR(4)	Storage Required
''	' '	4 bytes	''	1 byte
'ab'	'ab '	4 bytes	'ab'	3 bytes
'abcd'	'abcd'	4 bytes	'abcd'	5 bytes
'abcdefgh'	'abcd'	4 bytes	'abcd'	5 bytes

The BLOB and TEXT Types

A `BLOB` is a binary large object that can hold a variable amount of data.
The four `BLOB` types are `TINYBLOB`, `BLOB`, `MEDIUMBLOB`, and `LONGBLOB`.
`BLOB` values are treated as binary strings (byte strings). They have the `binary` character set and collation, and comparison and sorting are based on the numeric values of the bytes in column values.
TINYBLOB
- max 255 Bytes of binary data

BLOB
- max 65,535 Bytes of binary data

MEDIUMBLOB
- max 16,777,215 characters (16 Mb - 1)

LONGBLOB
- max 4,294,967,295 characters (6 Gb - 1)

- e.g., BARCODE, PICTURE_CODE, QR_CODE, SIGNATURES, FINGERPRINTS, BIOMETRICS, PICTURES, PHOTOGRA{HS, SOUND, MUSIC, VIDEOS, GRAPHS, CHARTS, MAPS, etc
- Blob is the multimedia datatypes of MySQL
- MySQL maintains a LOCATOR (harddisk pointer) from the table row to the text data

The four `TEXT` types are `TINYTEXT`, `TEXT`, `MEDIUMTEXT`, and `LONGTEXT`
Tinytext
- allows any characters
- variable length
- max 255 characters

Text
- allows any character
- variable legth
- max 65,535 characters (64 kb - 1)

Mediumtext
- allows any characters
- variable length
- max 16,777,215 characters (16 Mb - 1)

Longtest
- allows any characters
- variable length
- max 4,294,967,295 characters (4 Gb - 1)

- Width does not have to specified for all of the above 4 datatypes
- above 4 are stored outside the table
- above 4 are stored away from the table row
- MySQL maintains a LOCATOR (harddisk pointer) from the table row to the text data
- used for columns that will be not used for searching

- used for columns that will be used only for storage and display
- e.g., REMARKS, COMMENTS, RESUME, EXPERIENCE, FEEDBACK, etc.

Numeric Data Types

Integer Types (Exact Value) -

INTEGER, INT, SMALLINT, TINYINT, MEDIUMINT, BIGINT

MySQL supports the SQL standard integer types `INTEGER` (or `INT`) and `SMALLINT`. As an extension to the standard, MySQL also supports the integer types `TINYINT`, `MEDIUMINT`, and `BIGINT`. The following table shows the required storage and range for each integer type.

Type	Storage (Bytes)	Minimum Value Signed	Minimum Value Unsigned	Maximum Value Signed	Maximum Value Unsigned
TINYINT	1	-128	0	127	255
SMALLINT	2	-32768	0	32767	65535
MEDIUMINT	3	-8388608	0	8388607	16777215
INT	4	-2147483648	0	2147483647	4294967295
BIGINT	8	-2^{63}	0	$2^{63}-1$	$2^{64}-1$

Fixed-Point Types (Exact Value) - DECIMAL, NUMERIC

The `DECIMAL` and `NUMERIC` types store exact numeric data values. These types are used when it is important to preserve exact precision, for example with monetary data. In MySQL, `NUMERIC` is implemented as `DECIMAL`, so the following remarks about `DECIMAL` apply equally to `NUMERIC`.
In a `DECIMAL` column declaration, the precision and scale can be (and usually is) specified. For example:

```
salary DECIMAL(5,2)
```

In this example, `5` is the precision and `2` is the scale. The precision represents the number of significant digits that are stored for values, and the scale represents the number of digits that can be stored following the decimal point.

- stores double as a string
    - eg. "536.7"
- max number of digits is 65
- used when it is important to preserve exact precision,
    - for example with monetary data

Floating-Point Types (Approximate Value) - FLOAT, DOUBLE

The `FLOAT` and `DOUBLE` types represent approximate numeric data values. MySQL uses four bytes for single-precision values and eight bytes for double-precision values.

Float
- upto 7 decimals

Double
- upto 15 decimals

Boolean

- logical datatypes
- True and False evaluate to 1 and 0 respectively
    - e.g. EXAM_RESULT, MARITAL_STATUS, etc
- can insert true, false, 1 or 0
- output will display 1 or 0

Date and Time Data Types

Date

- Supported Dates: 1st Jan 1000 AD to 31st Dec 9999 AD
- no problem of Y2K
- 'YYYY-MM-DD' is the default date format in MysQL
  e.g., '2022-05-07'
- 'YY-MM-DD' is also allowed
  e.g., '22-05-07'
- year values in the range 70-99 are converted to 1970-1999
- year values in the range 00-69 are converted to 2000-2069
- date arithmetic is allowed
  can subtract one date from another date
  date1 - date2 = return number days between date1 and date2

'2022=05-07' - '2020-04-19'
1st Jan 1000AD - 1
2nd Jan 1000AD - 1
3rd Jan 1000AD - 1
...
7th May 2022 AD - 1456852

- internally date is stored as a fixed-length number and it occupies 7 Bytes of storage
- internally date is stored as the number of days since 1st Jan 1000 AD
- 'DD-MON-YY' is the default date format in Oracle
  e.g., '10-JAN-70'
- 1970 is known as the year of the Epoch (the year when Unix was launched)

Time

- 'hh:mm:ss' or 'HHH:MM:SS'
- time values may range from '-838:59:59' to '838:59:59'

Datetime

- date and time are stored together
- 'YYYY-MM-DD hh:mm:ss'
- '1000-01-01 00:00:00' to '9999-12-31 23:59:59'
- datetime1 - datetime2 = returns number of days, hours, minutes and seconds between the two

Year

- YYYY
- 1901 to 2155
- in MysQL, we can have maximum 4096 columns per table provided the row size <= 65,535 Bytes
- in MySQL, no limit on number of rows per table provided that table size does not exceed 64 TB
- in Oracle, no limit on table size