



# RELATIONAL RDBMS

# SUMMARY :

- What Is a Database
- What Is a DBMS
- What Is a RDBMS
- relational RDBMS : MySQL , PostgreSQL , SQL Server
- A comparison between MySQL vs PostgreSQL vs SQL

Server

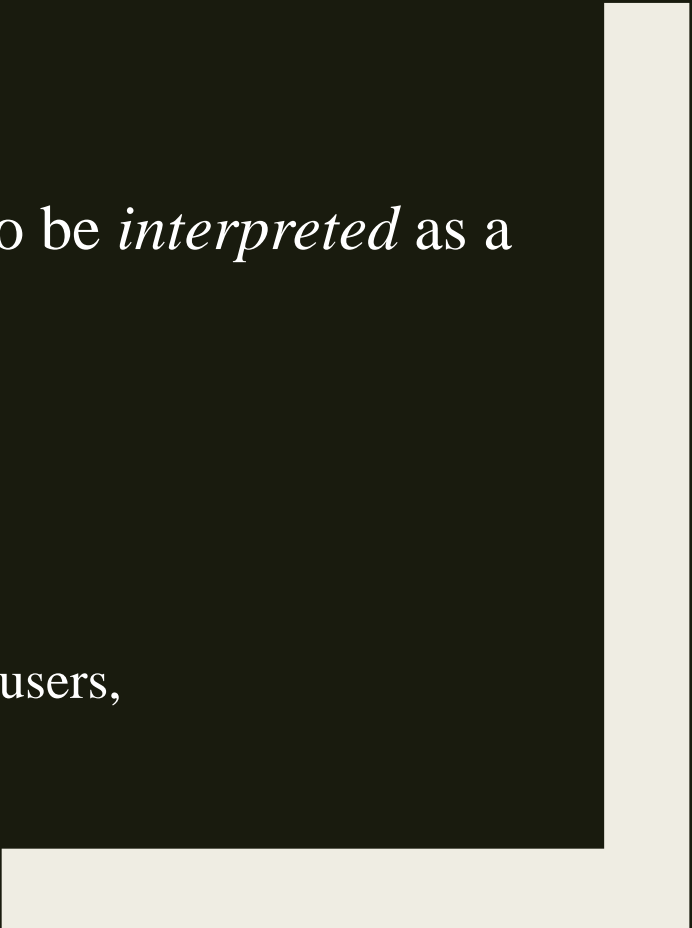
# What Is a Database?



An organised, machine-readable collection of *symbols*, to be *interpreted* as a *true* account of some *enterprise*.

Machine-updatable, too ...

A database is typically available to a community of users,  
with possibly varying requirements.



# What Is a Relational Database?

A database whose symbols are organised into a collection of relations. Here is a relation, shown in tabular form:

StudentId	Name	CourseId
S1	Anne	C1
S1	Anne	C2
S2	Boris	C1
S3	Cindy	C3
S4	Devinder	C1

Might be the value currently assigned to ENROLMENT, a relation variable (“relvar”).

# Anatomy of a Relation

StudentId	Name	CourseId
S1	Anne	C1

*attribute name*

*attribute values*

*Heading* (a set of attributes)  
The *degree* of this heading is 3,  
which is also the degree of the relation.

*n-tuple*, or *tuple*.  
This is a 3-tuple.  
The tuples  
constitute the *body*  
of the relation.  
The number of  
tuples in the body  
is the *cardinality* of  
the relation.

# What Is a DBMS?



A piece of software for managing databases and providing access to them.

A DBMS responds to imperatives (“statements”) given by application programs, custom-written or general-purpose, executing on behalf of users.

Imperatives are written in the database language of the DBMS (e.g., SQL).

Responses include completion codes, messages and results of *queries*.

In response to requests given by application programs:

- creates and destroys variables

- takes note of integrity rules (constraints)

- takes note of authorisations (who is allowed to do what, to what)

- updates variables (honouring constraints and authorisations)

- provides results of queries

- and more

Now, how does a relational DBMS do these things? ...

# What Is a RDBMS?



A relational database management system (RDBMS) is a collection of programs and capabilities that enable IT teams and others to create, update, administer and otherwise interact with a relational database. RDBMSes store data in the form of tables, with most commercial relational database management systems using Structured Query Language (SQL) to access the database. However, since SQL was invented after the initial development of the relational model, it is not necessary for RDBMS use.

The RDBMS is the most popular database system among organizations across the world. It provides a dependable method of storing and retrieving large amounts of data while offering a combination of system performance and ease of implementation.



## Advantages of MySQL



- Other advantages of the RDBMS include:
- Flexibility -- updating data is more efficient since the changes only need to be made in one place.
- Maintenance -- database administrators can easily maintain, control and update data in the database. Backups also become easier since automation tools included in the RDBMS automate these tasks.
- Data structure -- the table format used in RDBMSes is easy to understand and provides an organized and structural manner through which entries are matched by firing queries

## Uses of RDBMS



- Relational database management systems are frequently used in disciplines such as manufacturing, human resources and banking. The system is also useful for airlines that need to store ticket service and passenger documentation information as well as universities maintaining student databases.
- Some examples of specific systems that use RDBMS include IBM, Oracle, MySQL, Microsoft SQLServer and PostgreSQL.

## RDBMS vs. DBMS

In general, databases store sets of data that can be queried for use in other applications. A database management system supports the development, administration and use of database platforms.

An RDBMS is a type of database management system (DBMS) that stores data in a row-based table structure which connects related data elements. An RDBMS includes functions that maintain the security, accuracy, integrity and consistency of the data. This is different than the file storage used in a DBMS.

# MySQL , PostgreSQL , SQL Server :



MY SQL

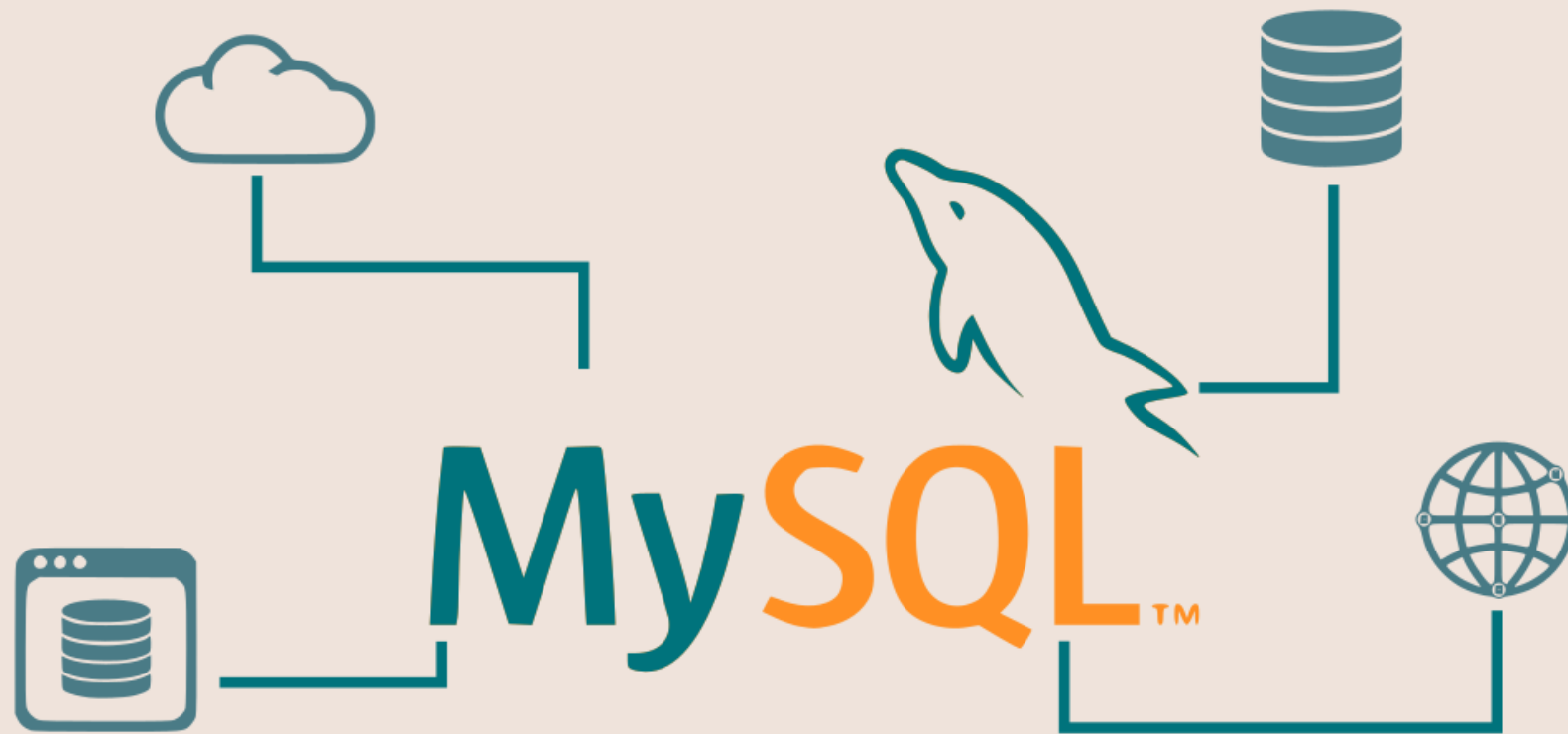


POSTGRE SQL



SQL SERVER

# MySQL



## Key features of MySQL

- As an RDBMS, MySQL is based on SQL. The key features of MySQL are as follows:
- The creators of this RDBMS used C and C++ to create MySQL.
- MySQL uses a client-server architecture.
- This RDBMS works with a wide range of operating systems like Windows, Linux, Unix, OS/2, FreeBSD, etc.
- You can install and run MySQL on all key platforms.

## Advantages of MySQL



Apart from being an open-source and free RDBMS, MySQL offers the following advantages:

- The ease of installation
- Lightweight.
- Tooling support
- Compatibility:
- The ease of use
- Security
- Performance
- Scalability
- Availability
- High productivity
- Efficiency
- Support

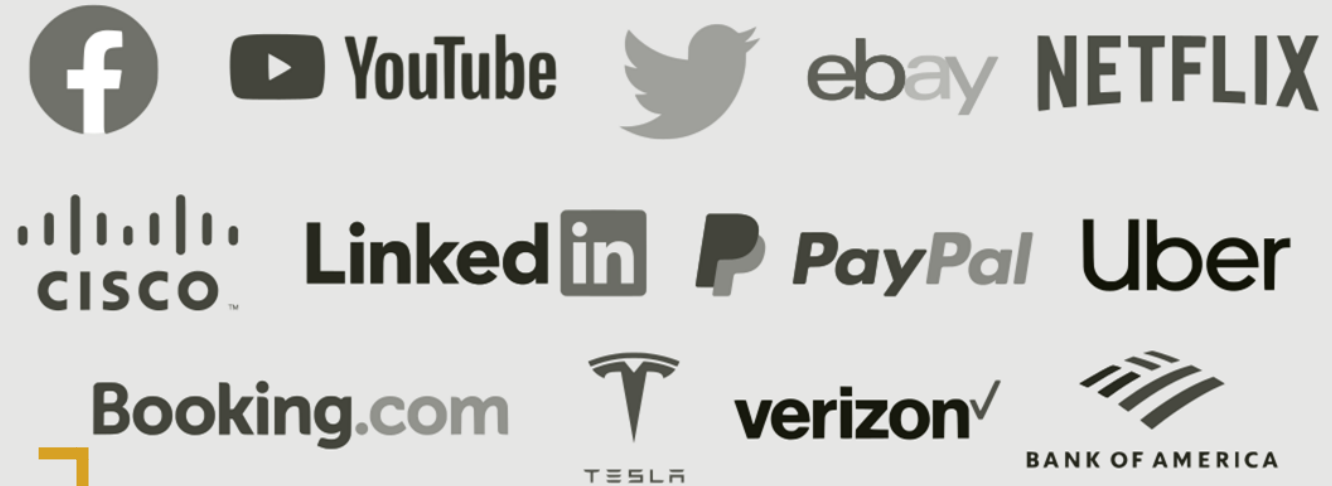
## Disadvantages of MySQL



MySQL has a few disadvantages too, which are as follows:

- Limitations vis-a-vis embedding in an application
- Business logic-related limitations
- Stability issues
- The dependency on add-ons
- Limitations concerning tools when compared to paid databases
- MySQL isn't very efficient when handling large databases.
- MySQL versions prior to 5.0 don't support commits and stored procedures.

## Companies that use MySQL™

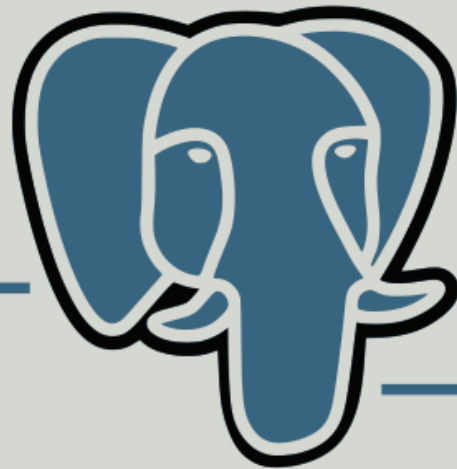


The following are a few examples of prominent companies/apps that use MySQL:

Developers use MySQL widely for [web application development](#). Popular “Content Management Systems” (CMSs) like WordPress, Joomla, Drupal, etc. use MySQL.

# Postgre SQL

**PostgreSQL**





## Key features of PostgreSQL

The following are the noticeable features of PostgreSQL:

- This open-source RDBMS is free.
- Its creators used the C language to develop PostgreSQL.
- PostgreSQL supports all the key platforms like Windows, Linux, Mac, etc.
- This RDBMS uses SQL and offers features similar to traditional RDBMSs like Oracle and DB2.
- PostgreSQL offers noticeable extensibility. You can use it along with other popular RDBMSs like Oracle and MySQL. Furthermore, you can use it along with popular NoSQL databases like MongoDB.
- PostgreSQL supports all popular languages like Java, Python, C, C++, C#, JavaScript, Ruby, Go, etc.

## Advantages of PostgreSQL



PostgreSQL offers the following advantages:

- Powerful features
- The ease of handling transactions
- Code comments
- Parameters
- Extensibility
- Security
- “Rich” SQL
- Flexibility
- Performance
- Popularity
- Support

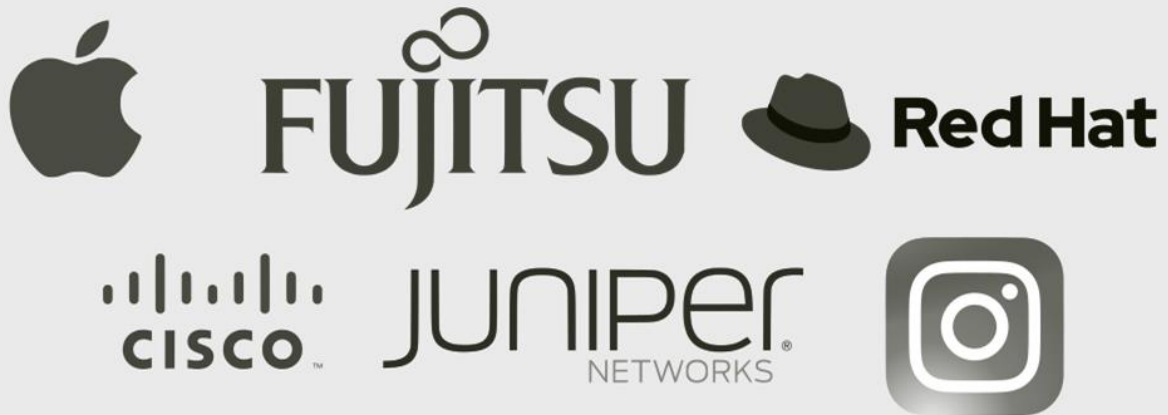
## Disadvantages of PostgreSQL



PostgreSQL has a few disadvantages, which are as follows:

- The lack of data compression
- Analytics-related limitations
- Performance-related limitations
- Administration-related challenges

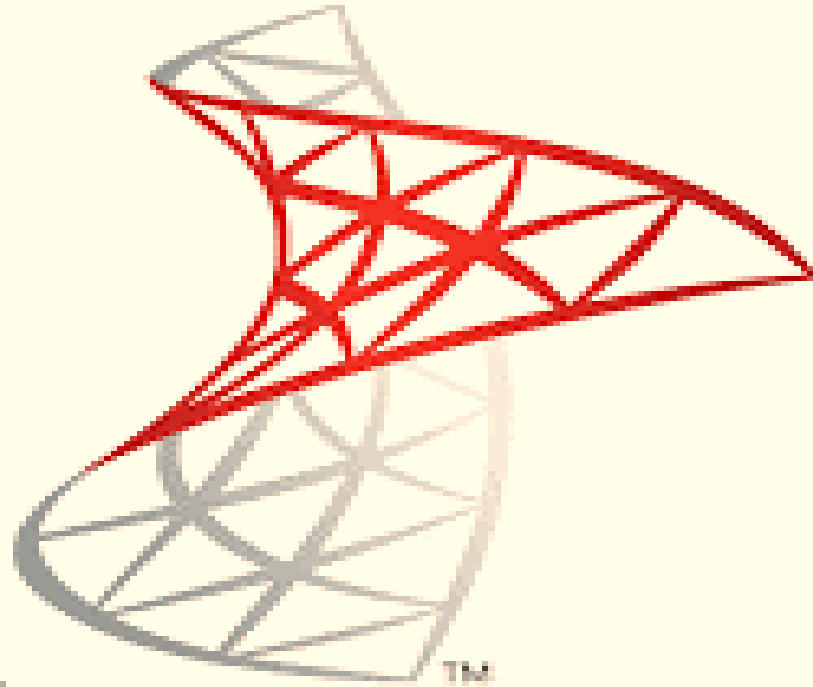
## Companies that use PostgreSQL



The following are examples of well-known companies/apps that use PostgreSQL:

Many organizations and developers use PostgreSQL to develop applications, websites, and tools. In addition to using PostgreSQL as a general-purpose transaction database, you can use it for geospatial use cases too.

# SQL Server



Microsoft®  
**SQL Server®**

**SQL Server** is a relational database management system, or RDBMS, developed and marketed by Microsoft. Similar to other RDBMS software, SQL Server is built on top of SQL, a standard programming language for interacting with relational databases. SQL Server is tied to Transact-SQL, or T-SQL, the Microsoft's implementation of SQL that adds a set of proprietary programming constructs. SQL Server works exclusively on the Windows environment for more than 20 years. In 2016, Microsoft made it available on Linux. SQL Server 2017 became generally available in October 2016 that ran on both Windows and Linux.

## Key features of SQL Server

### The following are the noticeable features of SQL Server

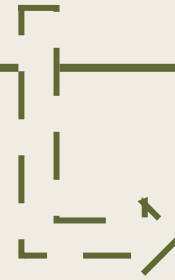
1. Intelligence on all your data with Big Data clusters: being able to query your entire data estate from SQL Serve to Oracle without replication.
2. Choice of Language and Platform: From Windows or Linux, to Kubernetes deployments.
3. Intelligent database capabilities: in-memory, persistent memory support, in-memory optimized tempdb
4. Data encryption and compliance: Its data protection, monitoring and classification system has made it one of the top most secure platforms according to the National Institute of Standards and Technology database for 9 years.
5. Mobile BI and scalability: Allowing you to easily integrate your database management systems with any device and Azure services for better performance and analytical capabilities on data.

## Advantages of SQL Server



1. Increases data security
2. Ease of configuration
3. Optimized data storage
4. Data recovery support

## Disadvantages of SQL Server



1. Cost
2. Restricted compatibility
3. Hardware restrictions

# **A comparison between MySQL vs PostgreSQL vs SQL Server**

## The focus on advanced features in MySQL vs PostgreSQL vs SQL Server

Are you looking for powerful features when you choose an open-source RDBMS? You should use PostgreSQL. It supports concurrency well. Furthermore, it offers other powerful features like nested transactions.

PostgreSQL complies completely with the ACID standards. This makes it a better choice for use cases where data integrity is very important. PostgreSQL offers the best compatibility with SQL standards, and it's ahead of MySQL and SQL Server in this regard.

## The comparison of performance between MySQL vs PostgreSQL vs SQL Server

Are you looking for speed and performance in an open-source RDBMS? Choose MySQL. We talked about a few limitations in MySQL concerning capabilities, however, that provides one advantage too! Since the developers behind MySQL didn't implement certain complex features, this RDBMS can offer speed. It takes a lead over PostgreSQL and SQL Server in this regard



## The comparison of security between MySQL vs PostgreSQL vs SQL Server

Does security have a high priority in your project requirements? You should choose MySQL over PostgreSQL and SQL Server. MySQL allows you to set passwords at the level of the root user. You can set the security level of the password at the installation-level. You can remove unwanted test databases, furthermore, you can grant access privileges at the level of a user.

## The comparison of support between MySQL vs PostgreSQL vs SQL Server

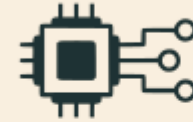
Do you want excellent community support for the open-source RDBMS you choose? After all, you aren't using a commercial product. Robust community support has importance for you. You should choose PostgreSQL in this case since a large and vibrant community supports it. MySQL and SQLite have their developer communities too. However, PostgreSQL maintains a lead in this regard.

## The comparison of popularity between MySQL vs PostgreSQL vs SQL Server

Do you want to find developers easily, therefore, you want to use the most popular open-source RDBMS? Most developers will likely have skills in such an RDBMS. This makes finding developers easier. In this case, you should choose MySQL. While PostgreSQL and SQLite are popular too, MySQL enjoys higher popularity

# WHEN TO USE MYSQL VS POSTGRESQL ?

Developing Embedded Applications



Launching Websites & Web Apps  
that grow in the future



Creating Distributed Applications



Developint Complex Applications



System Properties Comparison Microsoft SQL Server vs. MySQL vs. PostgreSQL

Name	Microsoft SQL Server	MySQL	PostgreSQL
Description	<u>Relational DBMS</u>	<u>Relational DBMS</u>	<u>Relational DBMS</u>
Secondary database models	<u>Document store</u> <u>Graph DBMS</u> <u>Spatial DBMS</u>	<u>Document store</u> <u>Spatial DBMS</u>	<u>Document store</u> <u>Spatial DBMS</u>
Developer	Microsoft	Oracle	PostgreSQL Global Development Group
Initial release	1989	1995	1989
License	commercial	Open Source	Open Source
Cloud-based only	no	no	no

Implementation language	C++	C and C++	C
Server operating systems	Linux Windows	FreeBSD Linux OS X Solaris Windows	FreeBSD HP-UX Linux NetBSD OpenBSD OS X Solaris Unix Windows
Supported programming languages	C# C++ Delphi Go Java JavaScript (Node.js) PHP Python R Ruby Visual Basic	Ada C C# C++ Eiffel Haskell Java JavaScript (Node.js) PHP Python Ruby	.Net C C++ Delphi Java JavaScript (Node.js) Perl PHP Python Tcl



**THANK YOU FOR YOUR ATTENTION**