SQL is a standard language for storing, manipulating and retrieving data in databases.

Our SQL tutorial will teach you how to use SQL in: MySQL, SQL Server, MS Access, Oracle, Sybase, Informix, Postgres, and other database systems.

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

Using SQL in Your Web Site

To build a web site that shows data from a database, you will need:

- An RDBMS database program (i.e. MS Access, SQL Server, MySQL)
- To use a server-side scripting language, like PHP or ASP
- To use SQL to get the data you want
- To use HTML / CSS to style the page

Who Uses MySQL?

- Huge websites like Facebook, Twitter, Airbnb, Booking.com, Uber, GitHub, YouTube, etc.
- Content Management Systems like WordPress, Drupal, Joomla!, Contao, etc.
- A very large number of web developers around the world

Show Data On Your Web Site

To build a web site that shows data from a database, you will need:

- An RDBMS database program (like MySQL)
- A server-side scripting language, like PHP
- To use SQL to get the data you want
- To use HTML / CSS to style the page

What is RDBMS?

RDBMS stands for Relational Database Management System.

RDBMS is a program used to maintain a relational database.

RDBMS is the basis for all modern database systems such as MySQL, Microsoft SQL Server, Oracle, and Microsoft Access.

RDBMS uses SQL queries to access the data in the database.

Oracle, SQL Server, MySQL 5 and PostgreSQL are the most commonly used databases and people normally get caught in the comparison between them. The inability to work around all aspects of comparison becomes a limitation and leads to indecisiveness about which database to use.

Here is a comparison of the all the four in a pointwise manner:

Microsoft SQL Server is relational DBMS by Microsoft but both MySQL and PostgreSQL are widely used open source RDBMS.

Primary Database Model: A database model is a series of concepts which are used to describe data, its relationships, constraints, and even the semantics. It thus decides the logical makeup of any database. It also determines the manner in which data can be stored, organized and further used. The most widely used database model is the relational database model which is based on a table-based format. All basically use the Relational Database Management System which supports the relational data model.

Secondary Database Model: It is a vital element of comparison.

- **Document Store:** Document stores are common to all. They are also called database systems which are document-oriented and are known for schema-free data organization. Latter basically means that there is no uniform structure of records and can have a nested form.
- **Key Store:** Another common feature is the key-value store which basically stores keys and values in pairs. This is a simple database system which is not considered appropriate for complex applications. But mostly the simplicity is the primary aspect which makes it more attractive in some special instances.
- **Graph DBMS:** Another notable attribute which is only seen in Oracle and Microsoft SQL Server is the Graph DBMS. Latter is also known as a Graph Database. Herein, the database is represented in graph structures as nodes and edges which represent the relationship between nodes. The processing of data becomes easier with simple calculations of particular graph properties. Indexing on all nodes is not really provided.
- **RDF Store:** In the secondary model, RDF stores is seen in Oracle. It is basically a method for describing the information which was originally developed for detailing the metadata of IT resources.

Ranking:

As per the latest ranking in January 2019, Oracle has the highest score and is ranked number 1. This is followed by MySQL, then Microsoft SQL and ultimately PostgreSQL. While the first three shares close scores the last one has a wide gap in scores.Server Operating System: This has a huge variation when it comes to a comparison of the four databases. Below is a list of the supported OS by all four individually:

- **Microsoft SQL Server:** It supports Linux and Windows.
- **Oracle:** It supports AIX, HP-UX, Linux, OS X, Solaris, Windows, z/OS.
- **PostgreSQL:** It supports FreeBSD, HP-UX, Linux, NetBSD, OpenBSD, OS X, Solaris, Unix and Windows.
- **MySQL:** It supports FreeBSD, Linux, OS X, Solaris, and Windows.

Supported Languages for Programming: A programming language is a structured and organized language which comprises various instructions for producing various kinds of outputs. They are basically used in computer programming for the creation of programs which go to implement certain algorithms. Here again, there is a huge variation seen. Oracle has by far seen to be supporting a maximum number of languages followed by MySQL.

- **Microsoft SQL Server:** It supports C#, C++, Delphi, Go, Java, JavaScript, PHP, Python, R, Ruby, Visual Basic
- **Oracle:** It supports C, C#, C++, Clojure, Cobol, Delphi, Eiffel, Erlang, Fortran, Groovy, Haskell, Java, JavaScript, Lisp, Objective C, OCaml, Perl, PHP, Python, R, Ruby, Scala, Tcl, Visual Basic.
- **PostgreSQL:** It supports.Net, C, C++, Delphi, Java, JavaScript, Perl, PHP, Python, and Tcl.
- **MySQL:** It supports Ada, C, C#, C++, D, Delphi, Eiffel, Erlang, Haskell, Java, JavaScript, Objective-C, OCaml, Perl, PHP, Python, Ruby, Scheme, and Tcl.

Licensing: This forms an important criterion of comparison. The Microsoft SQL Server uses Commercial licensing which is closed source and includes features in various levels resting on version and Free Crippleware. Licensing for MySQL is Open Source, Commercial while for PostgreSQL it is also BSD Open Source. In Oracle, people can opt from a number of database services which rest on Oracle Cloud.

Availability of ODBC, JDBC, ADO.NET drivers: These are vital for an aspect of Data Connectivity but are generally overlooked. They can significantly enhance the performance of application, reliability, and portability. All these are powerful, economical and easy. These are supported by all the databases in comparison.

Installation and Maintenance: There are a lot of variations seen. While Microsoft SQL Server is the hardest to install and consumes a lot of time. It is easiest in case of MySQL and of medium level of difficulty for PostgreSQL. In case of Oracle as well, it is hard.**Can columns be added or Change names or data type views without dropping**: This can be easily handled in both Microsoft SQL and MySQL but can be highly bothersome in case of PostgreSQL especially in case of views which depend on other views.

Drop Tables: This is possible in both Microsoft SQL and MySQL but is not possible in PostgreSQL. This is also available for Oracle.

Computed Columns: This is possible in Microsoft SQL Server. This is less often used. Only in the case when you need the computed columns indexed you use this. Computed Columns are actually limited in use. This is, however, not possible in MySQL and PostgreSQL. In latter, however, there are functional indexes. In MySQL, it might feature in some future release. In Oracle, there was a complete absence of this feature but in 11g, a new feature lets you create a 'virtual column' which is basically an empty column and has a function over other columns of the table.

Functional Indexes: These indexes let you make an index based on a particular expression or function. These can have many columns, arithmetic expressions or even a PL/SQL function. These are absent in both Microsoft SQL and MySQL but are present in PostgreSQL. In Oracle, they were introduced in Oracle 8i.

Partial Indexes: These can be created by adding the Indexing Partial clause. The feature was completely absent in Microsoft SQL Server and MySQL but is present in PostgreSQL and Oracle. Oracle emulates partial indexes in a unique way.

Foreign Key: A foreign key which has a cascade delete actually means that if any record in the parent table gets deleted then the corresponding records in all the child tables will get deleted by default. This is present in Microsoft SQL, PostgreSQL and Oracle as well. In MySQL, this is present as InnoDB and not MyISAM.

Multi-Row Value Insert: This is not present in Microsoft SQL but is seen in all other three. The INSERT ALL statement in Oracle is used for adding many rows with only one statement.

UPSERT Logic: This is not possible in Microsoft SQL Server and PostgreSQL but is present in MySQL and Oracle. This is basically a feature where you can both insert and update together, hence the name UPSERT.

Programming of procs or functions in many languages: This is present in all but in MySQL although it is being developed in the same. While in Microsoft SQL Server, it is possible for any language which complies with CLR but it has to be compiled into a dll first. In PostgreSQL, the language environment is hosted by the downside server.

Dynamic and Action SQL in functions: This feature is not there in both Microsoft SQL Server and MySQL. In both you can however, it can be done in Stored Procedures. In PostgreSQL and Oracle, the feature is present.**Date Time Support**: Although the feature is present in all four it is best in PostgreSQL. It comprises Date, TimeStamp and TimeStamp with Timezone.

Support Creation of Functions: This is present in all four databases. Authentication: This stands for verification of the identity of someone who needs to access data or resources.

- **Microsoft SQL:** It has standard Db security and NT/Active Directory Authentication
- **MySQL:** It also has standard Db security with table-driven security.
- **PostgreSQL:** It has extensive security with standard, LDAP, SSPI, PAM, trust etc.
- **Oracle:** Standard Db security

Case-insensitivity: This basically means the databases are sensitive to cases. Apart from PostgreSQL none of the other databases are case-insensitive. Only PostgreSQL is case-sensitive by default

. **Conclusion:**

All the four databases offer unique features which have great usability for different audience. Microsoft SQL Server is the most preferred choice by many and thus will have great potential in 2019. A planned and systematic training course can open many new opportunities for you.

Certainly, here's a brief comparison of MySQL, Oracle, SQL Server, and PostgreSQL, four popular relational database management systems:

MySQL:

- Open-source RDBMS.

- Known for its speed, reliability, and ease of use.
- Good for small to medium-sized applications, especially web-based ones.
- Widely used in the startup and web development community.
- May have limitations for handling very large-scale or complex enterprise-level data.

Oracle:

- Commercial RDBMS.
- Designed for large-scale, enterprise-level applications.
- Offers high performance, scalability, and advanced features.
- Suitable for data-intensive industries, financial institutions, and complex systems.
- Can have a steeper learning curve and higher costs due to licensing.

SQL Server:

- Developed by Microsoft.
- Offers seamless integration with other Microsoft products and services.
- Known for its ease of use and administrative tools.
- Supports Windows-based environments.
- Suitable for businesses heavily reliant on Microsoft technologies.

PostgreSQL:

- Open-source RDBMS.
- Known for its extensibility, advanced features, and strong compliance with SQL standards.
- Suitable for a wide range of applications, from small projects to large enterprises.
- Offers powerful customization and is particularly favored by developers.
- Can handle complex queries and large datasets effectively.

Choosing the right database system depends on factors like the size of your application, the complexity of your data, performance requirements, budget, and your team's familiarity with the system. Each of these databases has its own strengths and weaknesses, so it's essential to evaluate your specific needs before making a decision.

DBMS vs RDBMS:

DBMS (Database Management System): A Database Management System (DBMS) is software that allows users to store, manage, and retrieve data in a structured way. It provides an interface for interacting with databases, allowing users to perform tasks like creating, updating, and querying data. A DBMS can handle various types of data, including structured, semi-structured, and unstructured data. It may or may not adhere strictly to the relational model.

RDBMS (Relational Database Management System): A Relational Database Management System (RDBMS) is a type of DBMS that specifically follows the relational model of data management. In an RDBMS, data is organized into tables (relations), which consist of rows (tuples) and columns (attributes). The data in these tables is structured and interconnected using keys and relationships. RDBMS systems ensure data integrity, enforce constraints, and support SQL (Structured Query Language) for querying and managing data.

Key Differences:

- **Data Structure:** In an RDBMS, data is structured into tables with rows and columns, and relationships are established between these tables using keys. A general DBMS might not necessarily follow the relational model and could handle various data structures.
- **Data Integrity:** RDBMS systems offer better data integrity through constraints like primary keys, foreign keys, and referential integrity, ensuring accurate and consistent data.
- **Query Language:** Both DBMS and RDBMS systems may support query languages, but RDBMS systems specifically support SQL, which is tailored for querying and manipulating structured relational data.
- **Relation Model:** RDBMS adheres to the strict relational model principles, whereas a DBMS could support other data models as well.
- **Use Cases:** RDBMS is suitable for applications where structured and relational data management is essential, like financial systems, CRM systems, and content management systems. A general DBMS might be used for managing less structured or diverse types of data.

In summary, an RDBMS is a specific type of DBMS that follows the relational model, focusing on structured data organized in tables with predefined relationships. On the other hand, a DBMS is a broader term that encompasses various types of data management systems, including those that might not adhere to the relational model.

Certainly, here are some key points differentiating DBMS (Database Management System) from RDBMS (Relational Database Management System):

DBMS (Database Management System):

1. **Data Structure:** DBMS can manage various data structures, including hierarchical and network models, in addition to relational data.
2. **Flexibility:** It offers more flexibility in terms of data organization and relationships. It may not enforce a strict relational model.
3. **Schema:** DBMS systems may not require a predefined schema or fixed structure for data.
4. **Data Integrity:** Data integrity features might be limited compared to RDBMS. Constraints and referential integrity are often optional.
5. **Query Language:** It might support query languages tailored for specific data models other than SQL.
6. **Use Cases:** DBMS systems are used for managing diverse data types and are often employed in scenarios where data structures are less predictable or where the relationship between data elements is not well-defined.

RDBMS (Relational Database Management System):

1. **Data Structure:** RDBMS strictly follows the relational model, organizing data into tables with rows and columns. Relationships are established using keys.
2. **Structured Data:** It excels at managing structured data and enforces data integrity through features like primary keys, foreign keys, and constraints.
3. **Schema:** RDBMS requires a predefined schema with a fixed structure for data. Changes to the schema often involve altering table structures.
4. **Data Integrity:** RDBMS systems have robust data integrity mechanisms, ensuring accurate and consistent data through referential integrity and constraints.
5. **Query Language:** RDBMS systems use SQL (Structured Query Language) as the primary query language, which is specifically designed for managing and querying structured relational data.
6. **Use Cases:** RDBMS systems are suitable for applications where structured and relational data management is essential, such as financial systems, CRM (Customer Relationship Management) software, and content management systems.

In summary, the main distinction between DBMS and RDBMS lies in their approach to data management and the strict adherence to the relational model in RDBMS. RDBMS is favored when data integrity, structured data, and well-defined relationships are critical, while DBMS provides more flexibility for managing diverse data structures. The choice between them depends on the specific requirements of your application and the nature of your data.

Main difference between DBMS and RDBMS

The main difference between a Database Management System (DBMS) and a Relational Database Management System (RDBMS) lies in how they handle data and the structure they enforce:

DBMS (Database Management System):

1. **Data Structure:** DBMS can manage various data structures, including hierarchical and network models, in addition to relational data.
2. **Flexibility:** It offers more flexibility in terms of data organization and relationships. It may not enforce a strict relational model.
3. **Schema:** DBMS systems may not require a predefined schema or fixed structure for data.
4. **Data Integrity:** Data integrity features might be limited compared to RDBMS. Constraints and referential integrity are often optional.
5. **Query Language:** It might support query languages tailored for specific data models other than SQL.
6. **Use Cases:** DBMS systems are used for managing diverse data types and are often employed in scenarios where data structures are less predictable or where the relationship between data elements is not well-defined.

RDBMS (Relational Database Management System):

1. **Data Structure:** RDBMS strictly follows the relational model, organizing data into tables with rows and columns. Relationships are established using keys.
2. **Structured Data:** It excels at managing structured data and enforces data integrity through features like primary keys, foreign keys, and constraints.

3. **Schema:** RDBMS requires a predefined schema with a fixed structure for data. Changes to the schema often involve altering table structures.
4. **Data Integrity:** RDBMS systems have robust data integrity mechanisms, ensuring accurate and consistent data through referential integrity and constraints.
5. **Query Language:** RDBMS systems use SQL (Structured Query Language) as the primary query language, which is specifically designed for managing and querying structured relational data.
6. **Use Cases:** RDBMS systems are suitable for applications where structured and relational data management is essential, such as financial systems, CRM (Customer Relationship Management) software, and content management systems.

In summary, the key difference is that DBMS provides more flexibility in data management, while RDBMS enforces a strict relational model, ensuring structured, well-defined data with strong data integrity mechanisms. The choice between them depends on the specific requirements of your application and the nature of your data.

DBMS (Database Management System):

- Think of DBMS as a data organizer that can handle all sorts of data in different ways.
- It's like a versatile toolbox for storing and managing data, where data can be structured in various formats.
- DBMS doesn't enforce strict rules about how data is related or structured.

RDBMS (Relational Database Management System):

- RDBMS, on the other hand, is like a specialized, structured filing system.
- It organizes data in neat tables with rows and columns, like a spreadsheet.
- RDBMS enforces strict rules for how data is related and structured, making it highly organized and suitable for structured data like names, addresses, and numbers.

In simple terms, DBMS is like a general-purpose data organizer, while RDBMS is a structured, rule-following system for well-organized data.

CONSTRAINTS :
It is a rule given to a column for validation .

Types of Constraints :

1. UNIQUE
2. NOT NULL
3. CHECK
4. PRIMARY KEY
5. FOREIGN KEY .

1. **UNIQUE** : "*It is used to avoid duplicate values into the column* ".

2. **NOT NULL** : *"It is used to avoid Null ".*

3. **CHECK** : "*It is an extra validation with a condition If the condition is satisfied then the value is* accepted else Rejected ".

EXAMPLE: CHECK(LENGTH(PH_NO))=10
CHECK (PERCENTAGE>60)

4. **PRIMARY KEY** : "*It is a constraint which is used to identify a record Uniquely from the table*" .
Characteristics of Primary key :

- We can have only 1 PK in a table
- PK cannot accept duplicate / repeated values .
- PK cannot accept Null
- PK is always a combination of Unique and Not Null Constraint.
- PK is not mandatory but highly recommended.

5. **FOREIGN KEY** : "*It is used to establish a connection between the The tables* "
Characteristics of Foreign key :

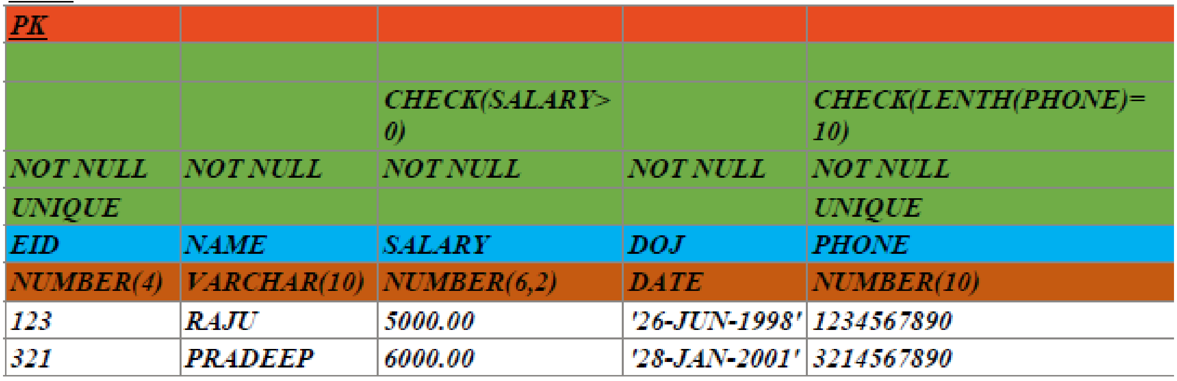
- We can have Multiple FK in a table
- FK can accept duplicate / repeated values .
- FK can accept Null
- FK is not a combination of Unique and Not Null Constraint.
- For an Attribute (column) to become a FK ,it is mandatory

that it must be a PK in its own table .

- FK are present in child table but it actually belongs to parent table
- FK is also known as "***REFRENTIAL INTEGRITY CONSTRAINT***"

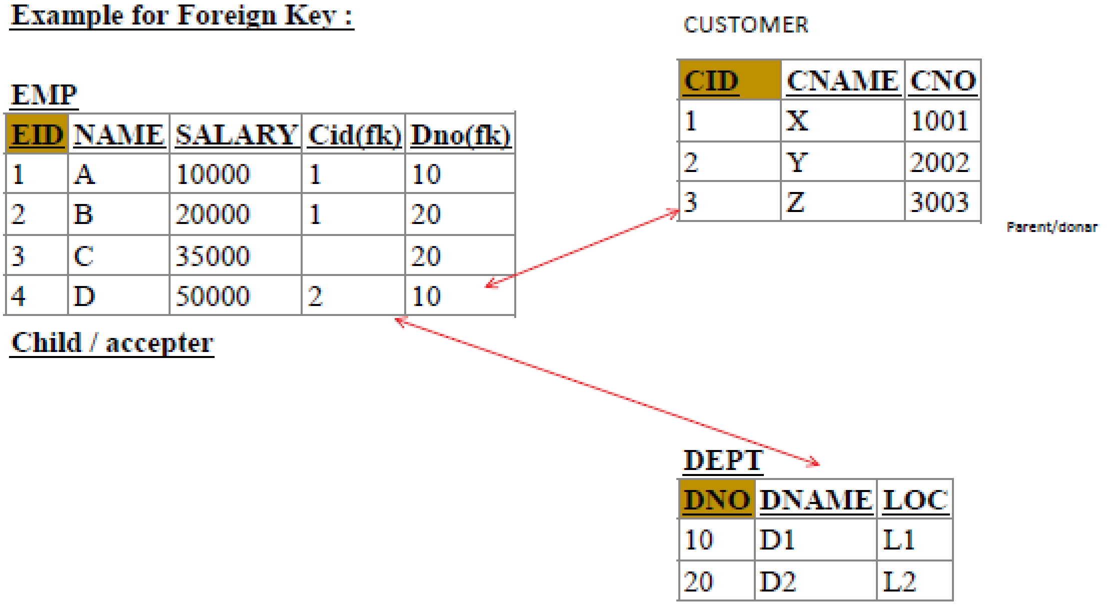
Example :

EMP



PK				
		CHECK(SALARY> 0)		*CHECK(LENTH(PHONE)= 10)*
NOT NULL	*NOT NULL*	*NOT NULL*	*NOT NULL*	*NOT NULL*
UNIQUE				*UNIQUE*
EID	*NAME*	*SALARY*	*DOJ*	*PHONE*
NUMBER(4)	*VARCHAR(10)*	*NUMBER(6,2)*	*DATE*	*NUMBER(10)*
123	*RAJU*	*5000.00*	*'26-JUN-1998'*	*1234567890*
321	*PRADEEP*	*6000.00*	*'28-JAN-2001'*	*3214567890*

Example for Foreign Key :



CUSTOMER

EMP

EID	NAME	SALARY	Cid(fk)	Dno(fk)
1	A	10000	1	10
2	B	20000	1	20
3	C	35000		20
4	D	50000	2	10

Child / accepter

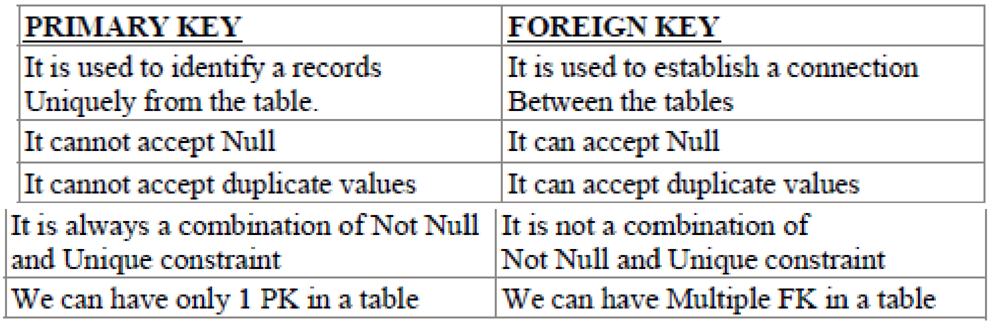
CID	CNAME	CNO
1	X	1001
2	Y	2002
3	Z	3003

Parent/donar

DEPT

DNO	DNAME	LOC
10	D1	L1
20	D2	L2

1. Differentiate between Primary key and Foreign key .



PRIMARY KEY	FOREIGN KEY
It is used to identify a records Uniquely from the table.	It is used to establish a connection Between the tables
It cannot accept Null	It can accept Null
It cannot accept duplicate values	It can accept duplicate values
It is always a combination of Not Null and Unique constraint	It is not a combination of Not Null and Unique constraint
We can have only 1 PK in a table	We can have Multiple FK in a table

NULL :

Null Is a keyword which is used to represent Nothing / Empty Cell.

Characteristics of Null :

- Null doesn’t represent 0 or Space .
- Any operations performed on a Null will result in Null itself
- Null doesn’t Occupy any Memory .
- We cannot Equate Null .