

What is MySQL

MySQL is an open-source relational database management system (RDBMS) developed by Oracle Corporation. This database is based on structured query language (SQL).

It is often called the most popular open-source relational database due to its easiness, powerful and secure characteristics. So basically, anyone getting involved in enterprise data or general IT should have minimum knowledge of this database.

Database and Relational database

Do you know, we deal with data every day. ?

When you browse through your phone's contact list to get any specific contact you are basically consulting with a database. In this case, your contact list is the database.

When you are uploading any photo to social media like Facebook, in this case as well you are interacting with a database. In this case, the photo gallery is a database.

So, the database is a collection of data stored in an organized manner for the purpose of easy access and management.

If the data is stored in tabular form i.e. which has rows and columns and the tables are related to each other then the database is called a relational database.

The software that interacts with the database, applications, and end-users is called a database management system (DBMS).

If the software is built to manage a relational database then the software is called a relational database management system (RDBMS).

What is Data?

Data is nothing but facts and statistics stored or free flowing over a network, generally it's raw and unprocessed. For example: When you visit any website, they might store you IP address, that is data, in return they might add a cookie in your browser, marking you that you visited the website, that is data, your name, it's data, your age, it's data.

Data becomes **information** when it is processed, turning it into something meaningful. Like, based on the cookie data saved on user's browser, if a website can analyse that generally men of age 20-25 visit us more, that is information, derived from the data collected.

What is a Database?

A **Database** is a collection of related data organised in a way that data can be easily accessed, managed and updated. Database can be software based or hardware based, with one sole purpose, storing data.

During early computer days, data was collected and stored on tapes, which were mostly write-only, which means once data is stored on it, it can never be read again. They were slow and bulky, and soon computer scientists realised that they needed a better solution to this problem.

Larry Ellison, the co-founder of **Oracle** was amongst the first few, who realised the need for a software based Database Management System.

What is DBMS?

A **DBMS** is a software that allows creation, definition and manipulation of database, allowing users to store, process and analyse data easily. DBMS provides us with an interface or a tool, to perform various operations like creating database, storing data in it, updating data, creating tables in the database and a lot more.

DBMS also provides protection and security to the databases. It also maintains data consistency in case of multiple users.

Here are some examples of popular DBMS used these days:

- MySql
- Oracle

- SQL Server
- IBM DB2
- PostgreSQL
- Amazon SimpleDB (cloud based) etc.

Characteristics of Database Management System

A database management system has following characteristics:

1. **Data stored into Tables:** Data is never directly stored into the database. Data is stored into tables, created inside the database. DBMS also allows to have relationships between tables which makes the data more meaningful and connected. You can easily understand what type of data is stored where by looking at all the tables created in a database.
2. **Reduced Redundancy:** In the modern world hard drives are very cheap, but earlier when hard drives were too expensive, unnecessary repetition of data in database was a big problem. But DBMS follows **Normalisation** which divides the data in such a way that repetition is minimum.
3. **Data Consistency:** On Live data, i.e. data that is being continuously updated and added, maintaining the consistency of data can become a challenge. But DBMS handles it all by itself.
4. **Support Multiple user and Concurrent Access:** DBMS allows multiple users to work on it(update, insert, delete data) at the same time and still manages to maintain the data consistency.
5. **Query Language:** DBMS provides users with a simple Query language, using which data can be easily fetched, inserted, deleted and updated in a database.

6. **Security:** The DBMS also takes care of the security of data, protecting the data from un-authorised access. In a typical DBMS, we can create user accounts with different access permissions, using which we can easily secure our data by restricting user access.
7. DBMS supports **transactions**, which allows us to better handle and manage data integrity in real world applications where multi-threading is extensively used.

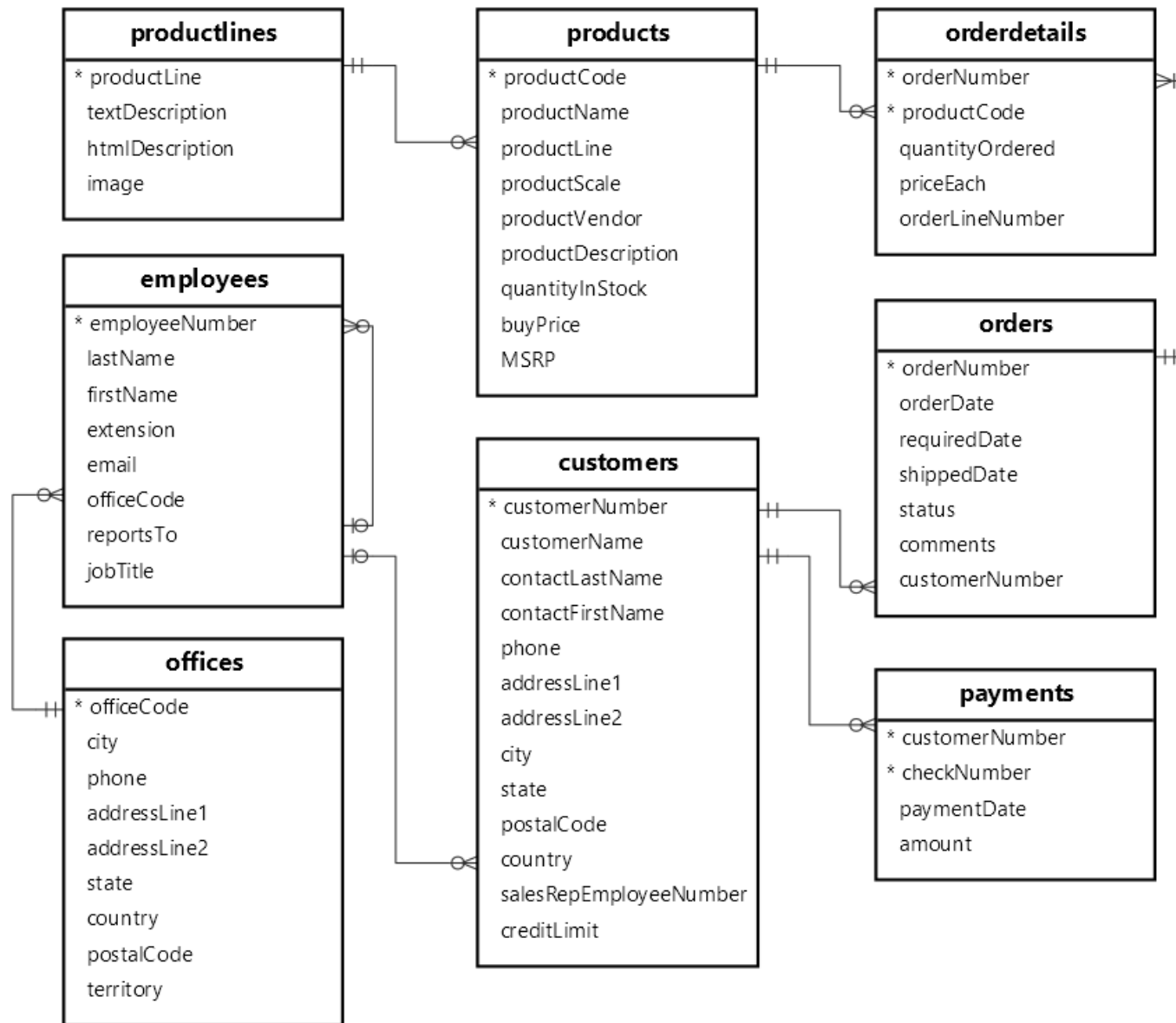
Advantages of DBMS

- Segregation of applicaion program.
- Minimal data duplicacy or data redundancy.
- Easy retrieval of data using the Query Language.
- Reduced development time and maintenance need.
- With Cloud Datacentres, we now have Database Management Systems capable of storing almost infinite data.
- Seamless integration into the application programming languages, which makes it very easier to add a database to almost any application or website.

Disadvantages of DBMS

- It's Complexity
- Except MySQL, which is open source, licensed DBMSs are generally costly.
- They are large in size.

How to Load the Sample Database into MySQL Server



MySQL Sample Database Schema

The MySQL sample database schema consists of the following tables:

- **customers:** stores customer's data.
- **products:** stores a list of scale model cars.
- **productlines:** stores a list of product lines.
- **orders:** stores sales orders placed by customers.
- **orderdetails:** stores sales order line items for every sales order.
- **payments:** stores payments made by customers based on their accounts.
- **employees:** stores employee information and the organization structure such as who reports to whom.
- **offices:** stores sales office data.

MySQL Monitor, Shell, and Workbench are three primary client tools for interacting with MySQL, each serving different developer and administrator needs.

MySQL Monitor

- The MySQL Monitor is the traditional command-line client, often invoked as `mysql`, which allows users to execute SQL queries, manage database objects, and perform administrative actions directly via text commands.
- It is lightweight, supports scripting and batch operations, and is available across all platforms that support MySQL.
- Common commands include connecting to databases, executing queries, and using statements such as `SHOW FULL PROCESSLIST`; to monitor active queries.

MySQL Shell

- The MySQL Shell is a modern, advanced command-line client designed for both interactive and scripting tasks.
- Offers support for SQL, JavaScript, and Python modes, empowering users with scripting capability beyond standard SQL.
- It provides utilities for working with MySQL servers, executing queries, configuring servers, and even managing JSON documents or MySQL InnoDB Cluster deployments.

MySQL Workbench

- MySQL Workbench is a graphical user interface designed for database architects, developers, and DBAs.
- It contains modules for SQL development, data modeling, and server administration, allowing users to visually manage connections, design schemas, run queries, monitor performance, and conduct database migration tasks.
- Available in Community (open source), Standard, and Enterprise Editions, Workbench adds value especially for those who prefer point-and-click interfaces and graphical database diagrams over command-line interactions.

Feature	Monitor (mysql)	Shell	Workbench
Interface	CLI	CLI (multi-language)	GUI
Query Execution	SQL only	SQL, JS, Python	SQL (visual editor)

Feature	Monitor (mysql)	Shell	Workbench
Data Modeling	No	Limited	Yes (visual)
Administration	Yes	Yes	Yes (visual tools)
Monitoring	Basic commands	Advanced scripting	Visual, graphical
Platform Support	All	All	Windows, macOS, Linux
Migration Tools	No	Some	Yes (migration wizard)

Monitor is ideal for quick, scriptable tasks and direct database access.

Shell is intended for advanced automation and hybrid SQL/NoSQL workflows.

Workbench is best for comprehensive database design, graphical administration, and cross-platform ease.

Connect to the MySQL server using the mysql

`mysql -u root -p`

- `mysql`: This is the command to start the MySQL client, which allows you to connect to interact with databases.
- `-u root`: This specifies the user that you want to connect to the MySQL database server. In this case, it is the root user that has full administrative privileges.
- `-p`: This flag will prompt you to enter the password for the root user after you execute the command.

`source c:/temp/mysqlsampledatabase.sql`

`show databases;`

SQL

SQL serves as a fundamental tool for analytics because it facilitates the retrieval and analysis of data, which enables synergies across business units

Databases are dumb – they can't infer what data you need in order to conduct an analysis for your boss. That's why we use SQL SELECT queries to exactly tell the database what data we need to fetch and display.

The most basic **SELECT** Query contains two essential SQL keywords: **SELECT** followed by **FROM**.

SELECT column1, column2, ...

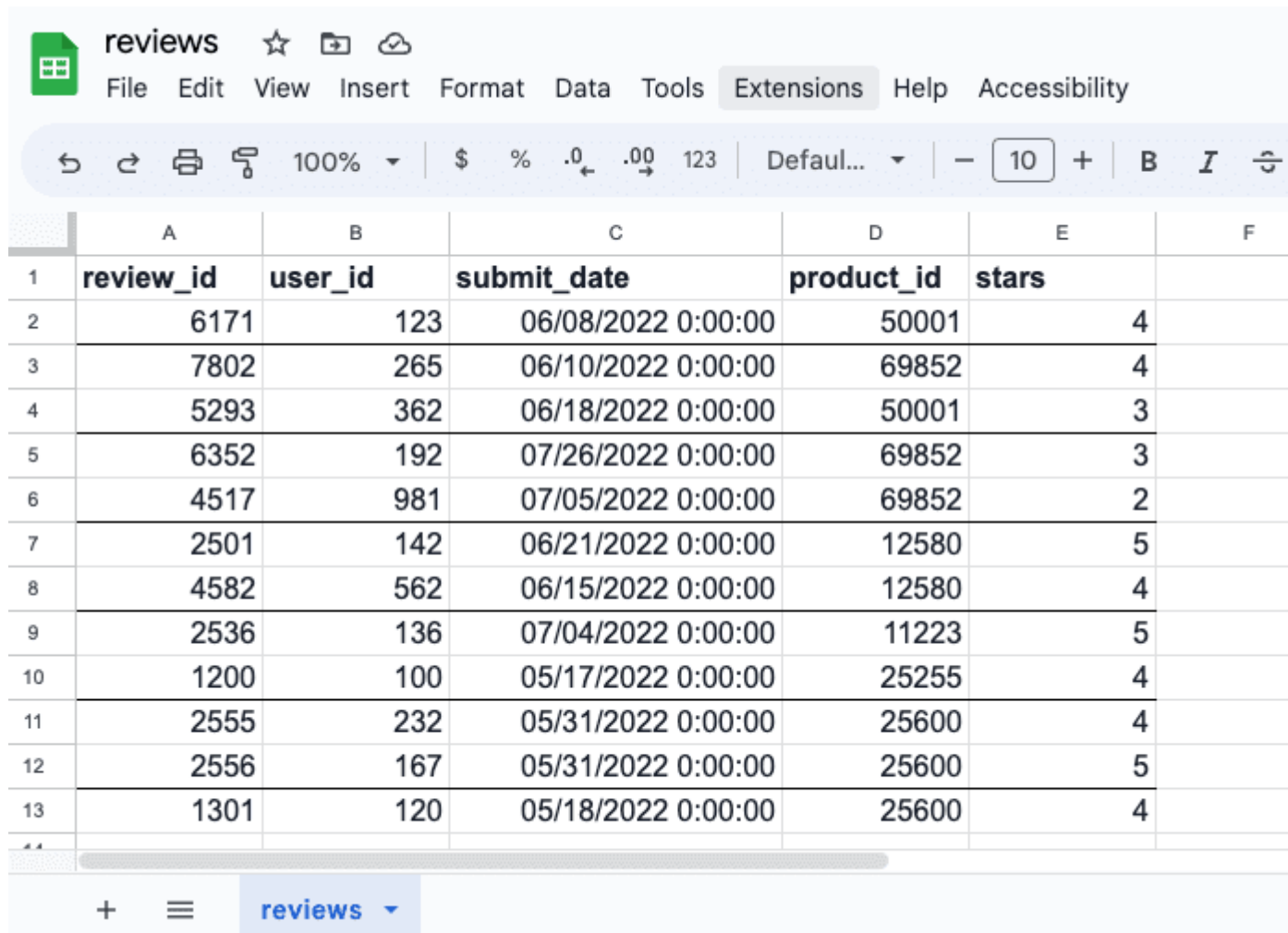
FROM table_name;

SELECT tells the database that you want to output data, and is followed by the specific names of the columns you want to have the database print out.

FROM tells the database which specific table contains that data we want to output.

Amazon SQL SELECT Example

Suppose you had access to Amazon's database, and there was a table called *reviews* which has data related to Amazon product reviews. Let's start by selecting a couple of columns from the reviews table!



The screenshot shows a Google Sheets interface with a spreadsheet titled "reviews". The spreadsheet contains a table with 6 columns: review_id, user_id, submit_date, product_id, stars, and an empty column. The data is as follows:

	A	B	C	D	E	F
1	review_id	user_id	submit_date	product_id	stars	
2	6171	123	06/08/2022 0:00:00	50001	4	
3	7802	265	06/10/2022 0:00:00	69852	4	
4	5293	362	06/18/2022 0:00:00	50001	3	
5	6352	192	07/26/2022 0:00:00	69852	3	
6	4517	981	07/05/2022 0:00:00	69852	2	
7	2501	142	06/21/2022 0:00:00	12580	5	
8	4582	562	06/15/2022 0:00:00	12580	4	
9	2536	136	07/04/2022 0:00:00	11223	5	
10	1200	100	05/17/2022 0:00:00	25255	4	
11	2555	232	05/31/2022 0:00:00	25600	4	
12	2556	167	05/31/2022 0:00:00	25600	5	
13	1301	120	05/18/2022 0:00:00	25600	4	

We're trying to have the database show us something similar, using the below SQL SELECT query:

SELECT review_id, submit_date, stars

FROM reviews;

The result of this query will be a two-dimensional set of rows and columns, effectively a copy of the reviews table, but only with the columns that we requested.

SELECT * Example

In SQL, to select each and every column in a table, you can use * instead of manually typing out all the column names:

SELECT *

FROM table_name;

The * comes in handy because in the real world, there might be hundreds of columns in the reviews table, used to represent the hundreds of attributes associated with a single Amazon review. To output all the columns in one go, you'd run this SELECT * query:

SELECT *

FROM reviews;