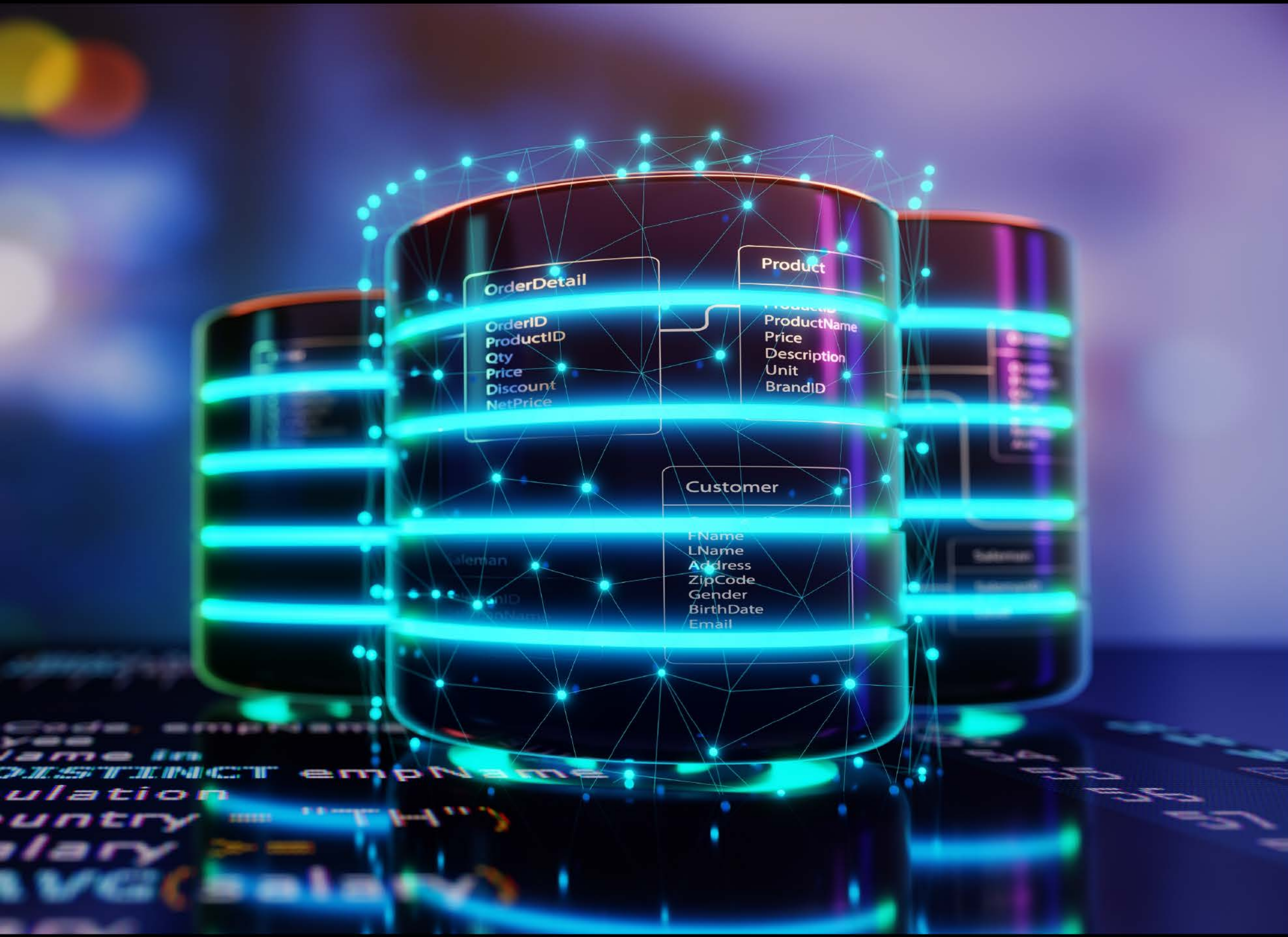
MySQL Essentials 9



Neil Smyth

Payload publishing

MySQL 9 Essentials

MySQL 9 Essentials

ISBN-13: 978-1-965764-10-7

Rev: 1.0

Payload
publishing

https://www.payloadbooks.com

Chapter 2

2. The Basics of Databases

The chances are that if you have ever logged into a website or purchased an item on the internet you have interacted with a database in some way. Anything that involves the retrieval or storage of information on a computer system is most likely to involve a database. In fact, databases are the core of almost every application that relies on data of some form to complete a task. In this chapter, we will introduce the basic concepts of databases.

2.1 Database vs. DBMS

The first step in learning MySQL is understanding the difference between a database and a database management system (DBMS). The term database refers to the entity that stores the actual data (such as ID numbers, names, and addresses, for example) in a structured way. A database management system (DBMS), on the other hand, refers to the software used to store, access, and manipulate the data stored in the database. All interactions with the database are performed via the DBMS.

Modern databases and database management systems are not restricted to storing just text. Today, databases store such items as images, videos, and software objects.

2.2 Client-server databases

MySQL is classified as a client-server database management system (DBMS). This type of DBMS consists of two main components. The server, which usually resides on the same physical computer as the database files, is responsible for all interactions with the database. The second component is the client, which sends database requests to the server. The server processes these requests and returns the results to the client.

There are several key advantages to using a client-server architecture for a database management system (DBMS). First, the client does not need to run on the same computer as the server. Instead, clients can send requests over a network or internet connection to a server located on a remote host. This setup makes the database accessible to a large number of clients. In large-scale enterprise environments, it also allows for fault tolerance, high performance, and load balancing to be implemented effectively.

Second, separating the client from the server allows a wider range of client types to be used to access the database. Typical clients include MySQL tools, desktop and mobile apps, web-based applications, web servers, and even other database servers:

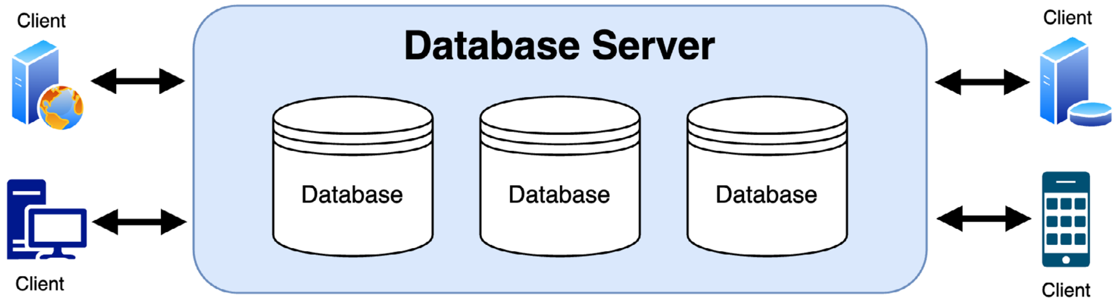


Figure 2-1

2.3 What is a database server?

The term "database server" can be somewhat misleading as it can refer to different concepts. One definition relates to the computer system hosting a Database Management System (DBMS) and other applications and services. However, in this book, we will specifically refer to the software component of a DBMS responsible for executing database operations on behalf of clients and returning the results. In the context of MySQL, this role is fulfilled by MySQL Server.

A database server can contain multiple databases, each containing one or more tables.

2.4 Understanding database tables

Database tables provide the most basic level of data structure in a database. Each database can contain multiple tables, each designed to hold information of a specific type. For example, a database may contain a customer table containing the name, address, and telephone number for all the customers of a particular business. The same database may also include a product table that stores the product descriptions with associated product codes and pricing for the items the business sells.

Each table in a database is assigned a name that must be unique to that particular database. A table name, once assigned to a table in one database, may only be re-used within the context of a different database:

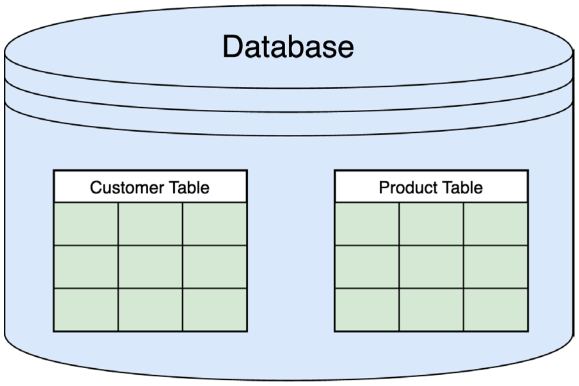


Figure 2-2

2.5 Introducing database schema

Database schemas define the characteristics of the data stored in a database table. For example, the schema for a customer database table might define that the customer name is a string of no more than 20 characters in length and that the customer phone number is a numerical data field of a specific format.

Schemas are also used to define the structure of entire databases and the relationship between the various tables in each database.

2.6 Columns and data types

At this stage, it is helpful to begin viewing a database table similar to a spreadsheet, where data is stored in rows and columns.

Each column represents a data field in the corresponding table. For example, a table's name, address, and telephone data fields are all columns.

Each column, in turn, is defined to contain a specific data type, which dictates the type of data it can store. Therefore, a column designed to store numbers would be defined as a numerical data type.

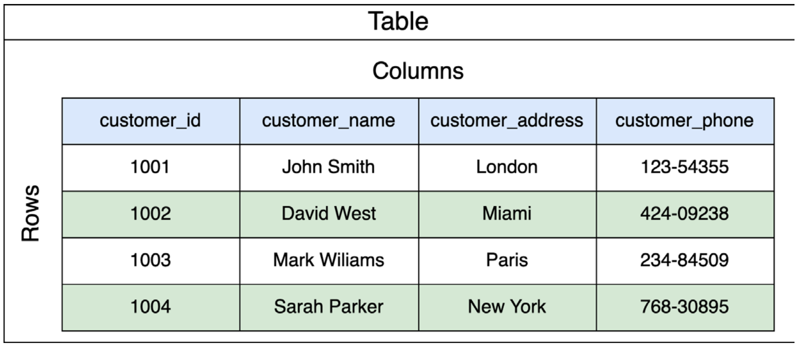
2.7 Database rows

Each new record saved to a table is stored in a row, which consists of the columns of data associated with the saved record.

Once again, consider the spreadsheet analogy described earlier. Each entry in a customer table is equivalent to a row in a spreadsheet, and each column contains the data for each customer (name, address, telephone number, etc.). The individual columns within a specific row are referred to as *fields*.

When a new customer is added to the table, a new row is created, and the data for that customer is stored in the corresponding columns of the new row.

Rows are also sometimes referred to as *records*, and these terms can generally be used interchangeably:



customer_id	customer_name	customer_address	customer_phone
1001	John Smith	London	123-54355
1002	David West	Miami	424-09238
1003	Mark Wiliams	Paris	234-84509
1004	Sarah Parker	New York	768-30895

Figure 2-3

2.8 Primary keys

Each database table must contain one or more columns that uniquely identify each row. This is known in database terminology as the *primary key*. For example, a table may use a bank account number column as the primary key. Alternatively, a customer table may assign unique identifiers to each customer as the primary key.

Primary keys allow the database management system to uniquely identify a specific row in a table. Without a primary key, retrieving or deleting a specific row in a table would be impossible because there is no certainty that the correct row has been selected. For example, suppose a table existed where the customer's last name had been defined as the primary key. Imagine the problem if more than one customer called "Smith" was recorded in the database. Without some guaranteed way to uniquely identify a specific row, it would be impossible to ensure the correct data was being accessed at any given time.

Primary keys can comprise a single column or multiple columns in a table. To qualify as a single column primary key, no two rows can contain matching primary key values. When using multiple columns to construct a primary key, individual column values do not need to be unique, but all the columns combined must be unique.

Finally, while primary keys are not mandatory in database tables, their use is strongly recommended.

2.9 What is SQL?

As discussed previously, a database management system (DBMS) provides the means to access the data stored in a database. One key method for achieving this is via a language called Structured Query Language (SQL), which is abbreviated to SQL and pronounced "sequel".

SQL is a straightforward and easy-to-use language developed at IBM in the 1970s specifically to enable the reading and writing of database data. Because SQL contains a small set of keywords, it can be learned quickly. In addition, SQL syntax is identical in most DBMS implementations, so having learned SQL for one system, your skills will likely transfer to other database management systems.

Throughout this book, particular attention will be paid to explaining the key SQL commands so that you will be proficient in using SQL to read, write, and manage database data.

2.10 Knowledge test

Click the link below or scan the QR code to test your knowledge and understanding of database architecture:

https://www.answertopia.com/vwh2

2.11 Reference points

The main points covered in this chapter are as follows:

- **Databases vs. DBMS**
  - A database stores structured data (e.g., names, addresses).
  - A DBMS (Database Management System) is software to access, store, and manipulate the database (e.g., MySQL).
- **Database Client-Server Architecture**
  - Databases operate on a server, and clients (apps, web servers) access them locally or remotely via a network.
  - A database server may host multiple databases, each with one or more tables.
- **Database Tables**
  - Tables organize data into rows (records) and columns (fields).
  - Each column has a specific data type (e.g., numerical, text).
- **Database Schema**
  - Defines the structure of a database and its tables, including data types and relationships.
- **Primary Keys**
  - Unique identifier for rows in a table (e.g., customer ID).
  - Can be a single column or a combination of columns.
  - Strongly recommended to ensure reliable row identification.
  - Proper table structure and primary keys ensure efficient data management and retrieval.
- **SQL (Structured Query Language)**
  - The primary language for interacting with databases.
  - Used for querying, updating, and managing database data.
  - SQL is simple, widely adopted, and transferable across DBMS platforms.