Chapter: 4

Relational Databases and Databases with Python

About the Course

“ This course, "Introduction to Relational Databases and MySQL," covers relational database principles, MySQL design and management, SQL syntax, querying, and database administration. It also teaches integrating databases with Python for dynamic applications. Through practical exercises and examples, students gain skills in database management. By course end, they will effectively use MySQL and Python for data management. ”

Learning Objectives

You will learn in this lesson:

- To understand the concepts of DBMS.
- To understand the concepts of RDBMS.
- To get a practical hands-on experience of the concepts of RDBMS.

Activity

An imaginative scenario:

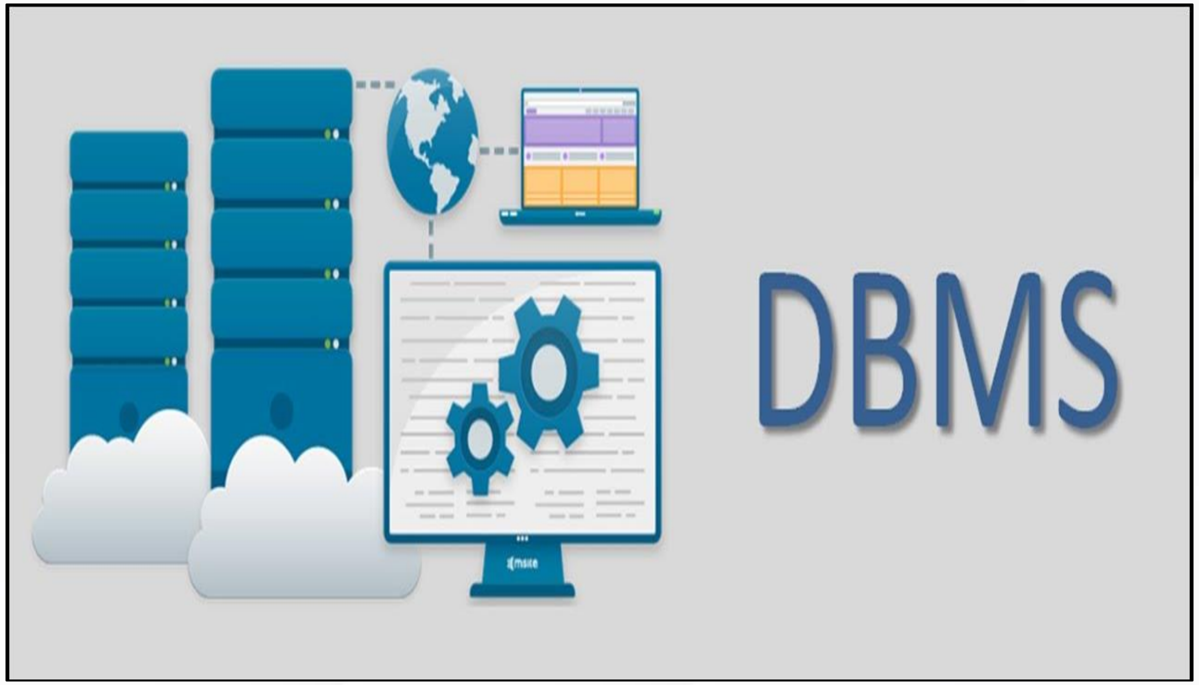
Once upon a time, in a bustling office filled with stacks of paper, a small business named "PaperWorks Co." was struggling to keep up with its growing list of customers, orders, and products. Every time they needed to find an order or update customer information, they had to search through piles of documents and folders. It was a real mess!

A new employee, Sarah, joins the company. She would like to help but don't know how. How can you help?

Source: https://th.bing.com/th/id/OIP.uhqnL_P82sFw9_RVWMGvdAHaEK?w=331

Introduction

- Data is a collection of recorded facts and figures with implicit meaning.
- DBMS (Database Management System) stores and accesses inter-related data effectively.
- Examples of inter-related data include a phone book or shopping list, stored on paper or digitally.



Source: https://i.ytimg.com/vi/lDpB9zF8LBw/maxresdefault.jpg

Why use a Database?

Databases can store very large numbers of records efficiently

It is very quick and easy to find information.

It is easy to add new data and to edit or delete old data.

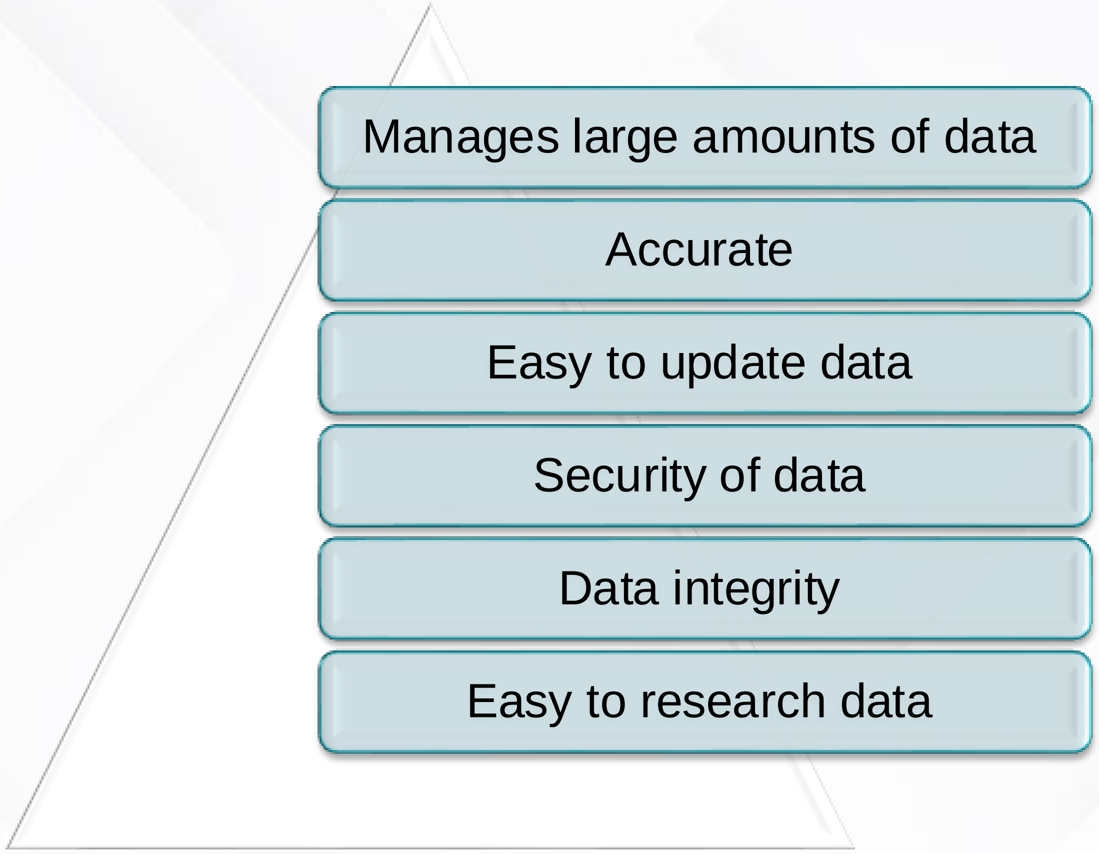
Data can be searched & Sorting easily,

Data can be imported into other applications

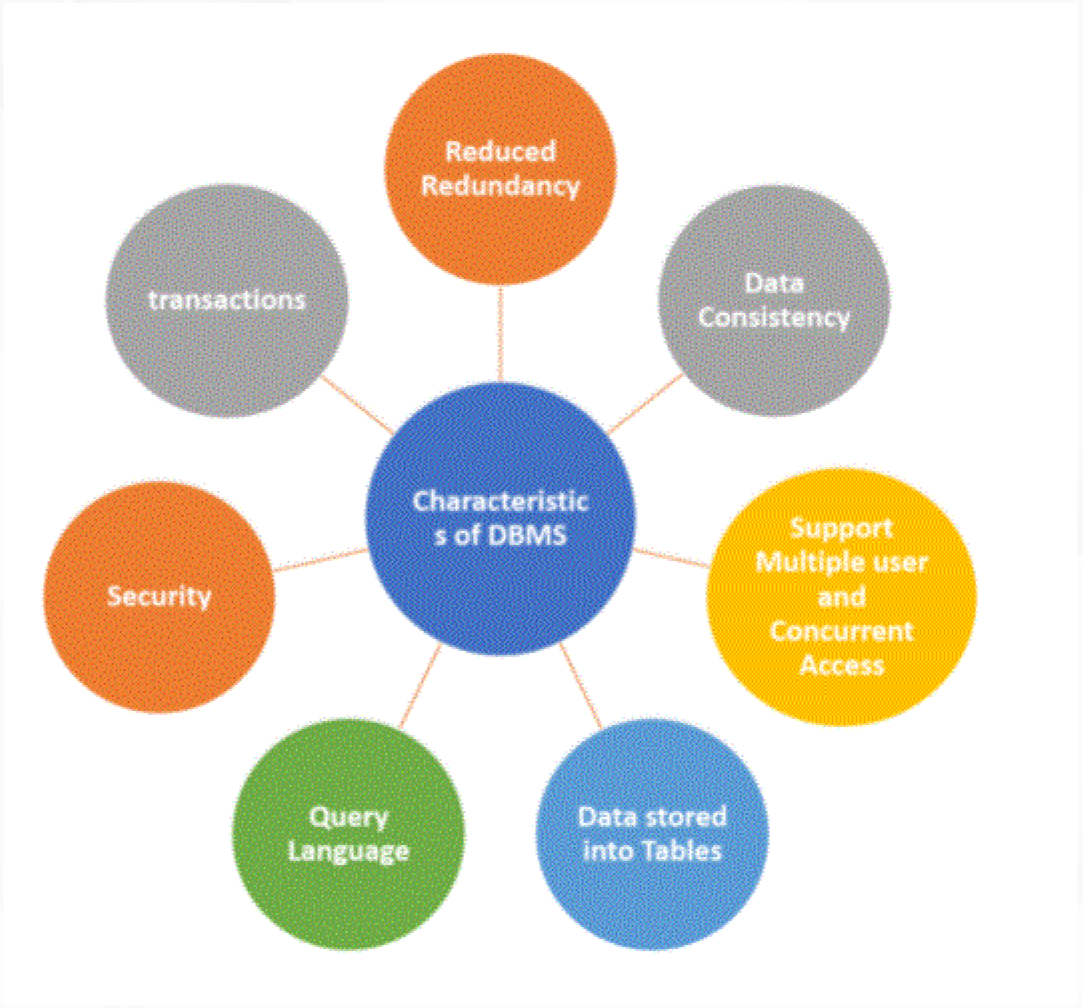
More than one person can access the same database at the same time - multi-access.

Security may be better than in paper files.

Why do we need a Database?



Characteristics of DBMS



Source : https://www.acte.in/wp-content/uploads/2021/12/Characteristics-of-DBMS-ACTE.gif

Advantages of DBMS

- Controls database redundancy
- Data sharing
- Easily Maintenance
- Reduce time
- Backup
- Multiple user interface

Disadvantages of DBMS

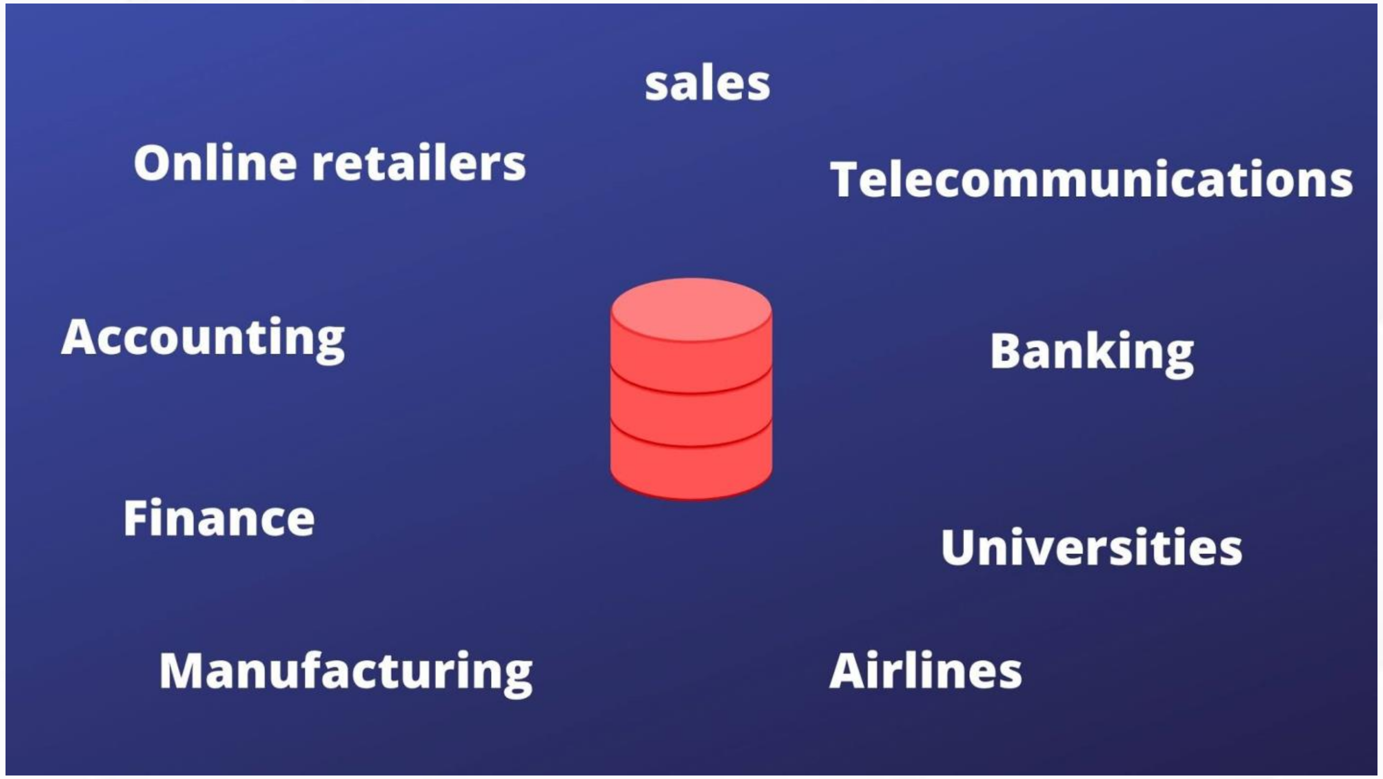
Cost of Hardware and software

Size

Complexity

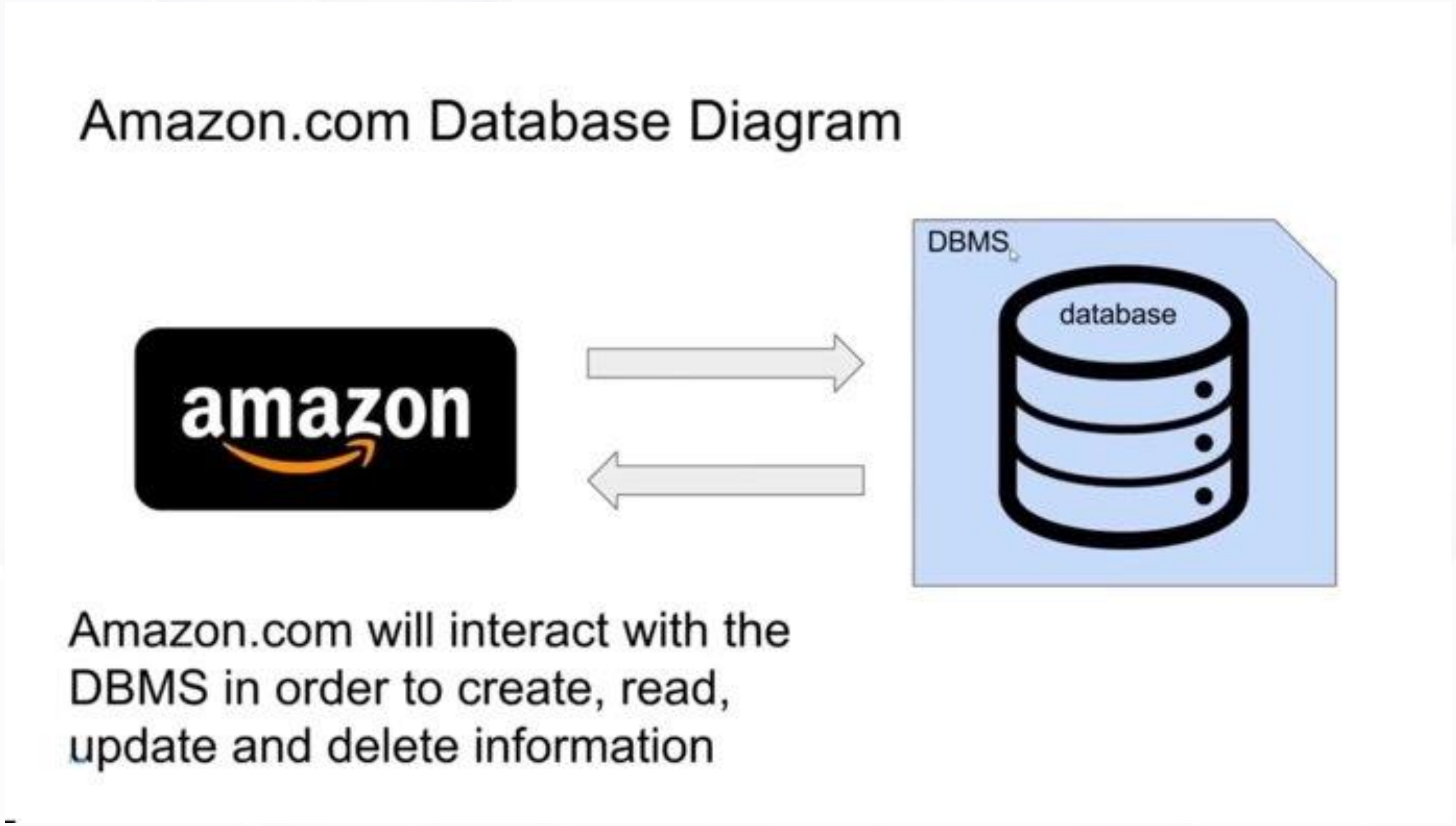
Higher impact of failure

Application of DBMS



Source: https://www.thecscience.com/2021/07/DBMS-applications.html

Example for DBMS



Amazon.com will interact with the DBMS in order to create, read, update and delete information

Source: https://youtu.be/HXV3zeQKqGY?si=ULWXLbHW-OIVDbOj

Database Queries

- Queries are requests made to the database management system for specific information.
- As the database's structure become more and more complex, it becomes more difficult to get the specific pieces of information we want.
- A Google/Bing search is a query.

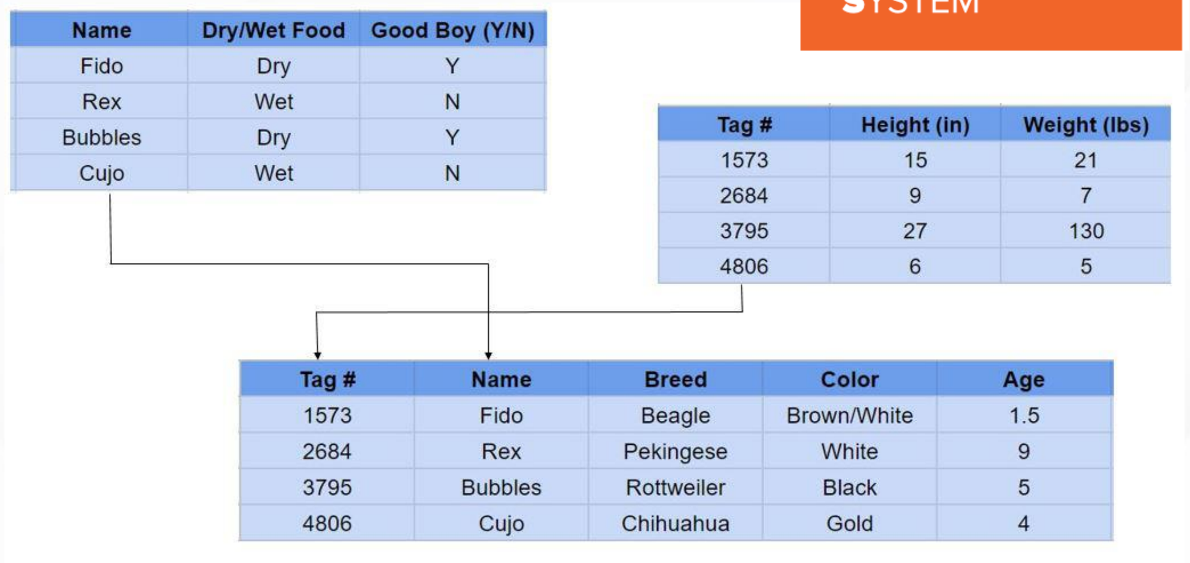
Source: https://www.flaticon.com/search?word=queries

Introducing Relational Database

- A database that follows the relational model and stores data in a tabular format is known as a relational database. The database has rows and columns and a unique key for each data point.

Examples:

Microsoft SQL Server, Oracle, MYSQL



Name	Dry/Wet Food	Good Boy (Y/N)
Fido	Dry	Y
Rex	Wet	N
Bubbles	Dry	Y
Cujo	Wet	N

Tag #	Height (in)	Weight (lbs)
1573	15	21
2684	9	7
3795	27	130
4806	6	5

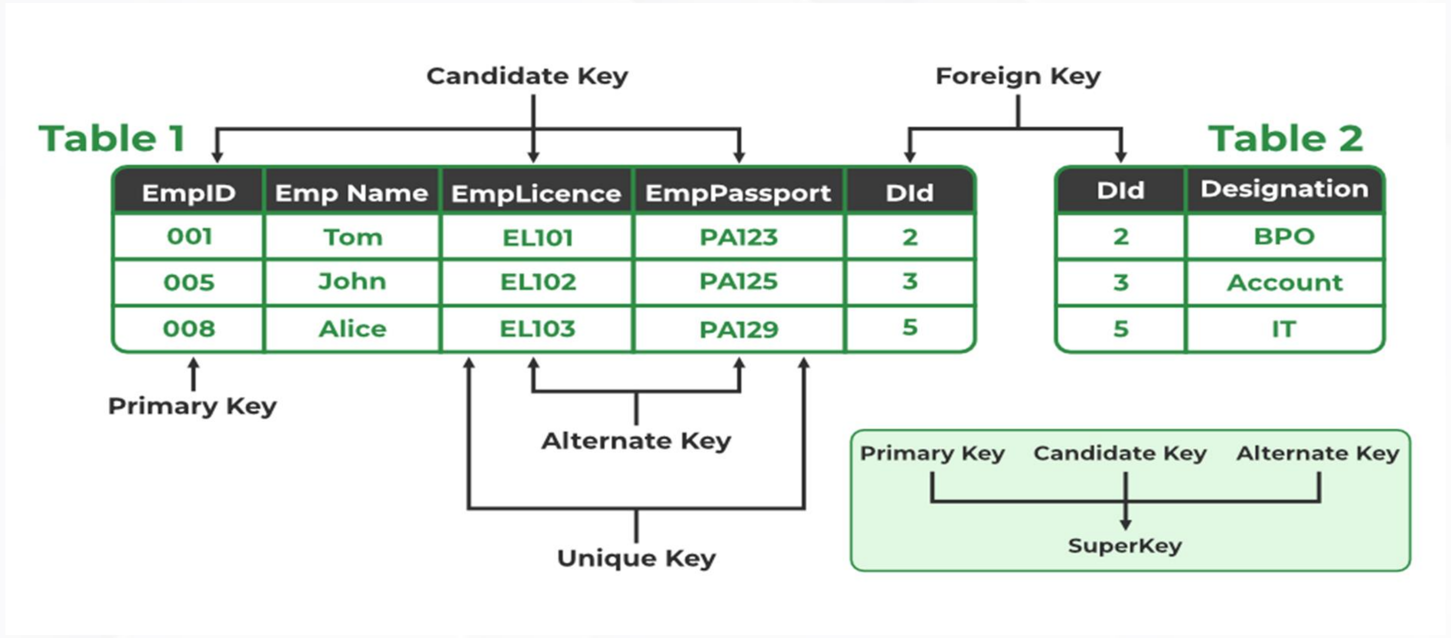
Tag #	Name	Breed	Color	Age
1573	Fido	Beagle	Brown/White	1.5
2684	Rex	Pekingese	White	9
3795	Bubbles	Rottweiler	Black	5
4806	Cujo	Chihuahua	Gold	4

Source: https://media.geeksforgeeks.org/wp-content/uploads/20200427225738/RDBMS.png

Terminology in RDBMS

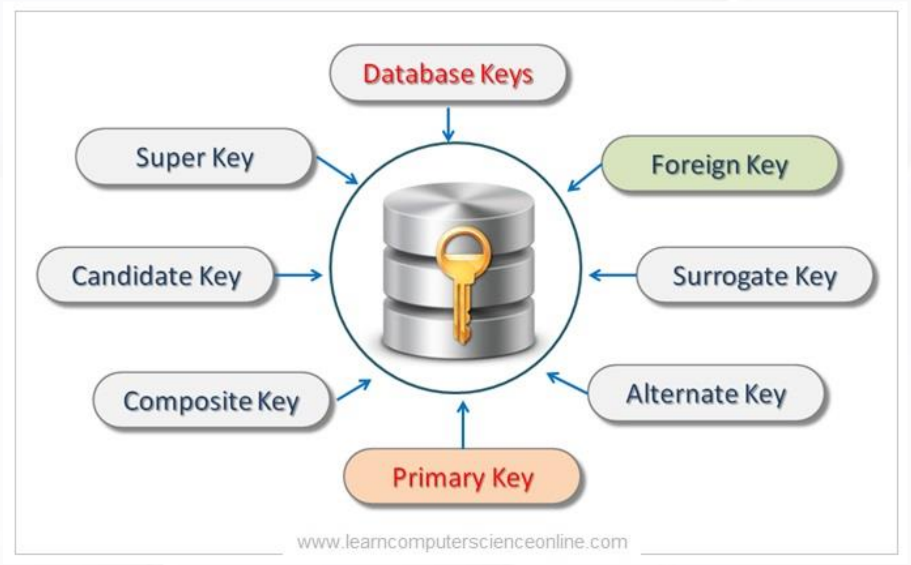
Table	A table is a set of data represented by columns and rows.
Row	It is a combination of column values and is referred to as a record.
Column	It is referred to as a field.
Relation	It defines database relationships in the form of tables.
Tuples	A single row of a table, which contains a single record for that relation
Attributes	An individual piece of data in a record is known as a field, or attribute.
Degree	A degree of relationship represents the number of entity types that associate in a relationship
Cardinality	It refers to the uniqueness of a column in a table.
Domain	It is a unique set of values permitted for an attribute in a table.

Types of RDBMS Keys



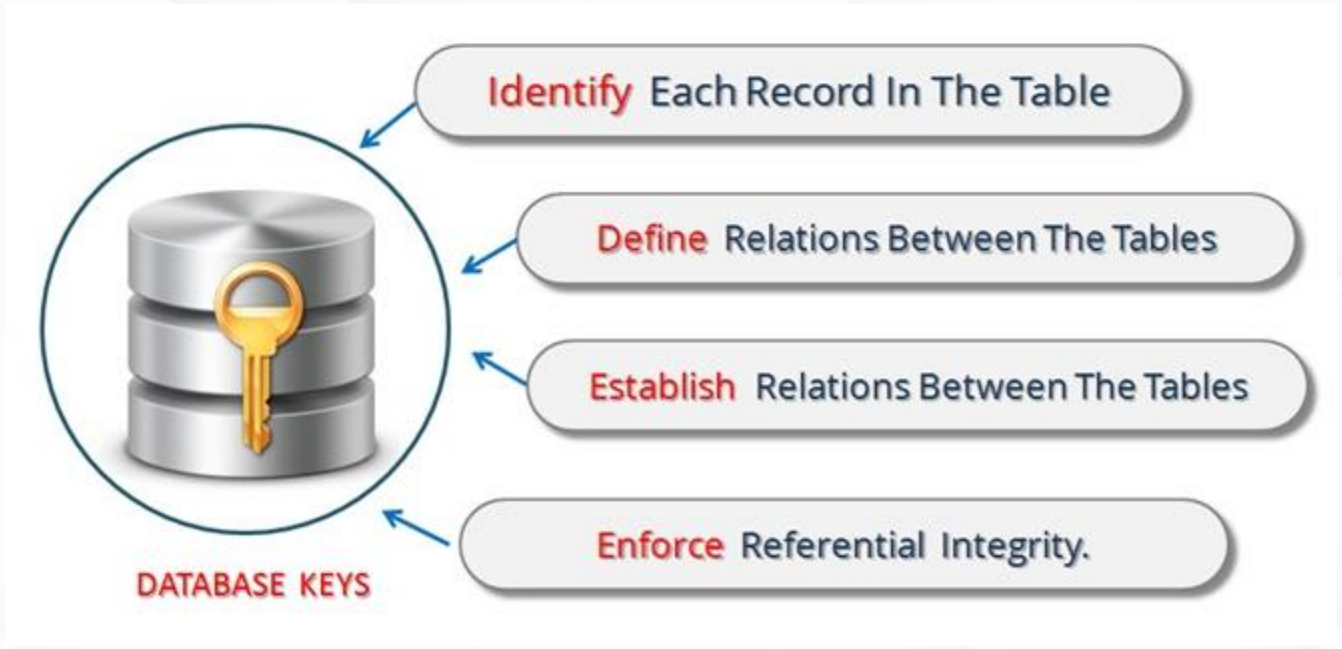
EmpID	Emp Name	EmpLicence	EmpPassport	DId
001	Tom	EL101	PA123	2
005	John	EL102	PA125	3
008	Alice	EL103	PA129	5

DId	Designation
2	BPO
3	Account
5	IT



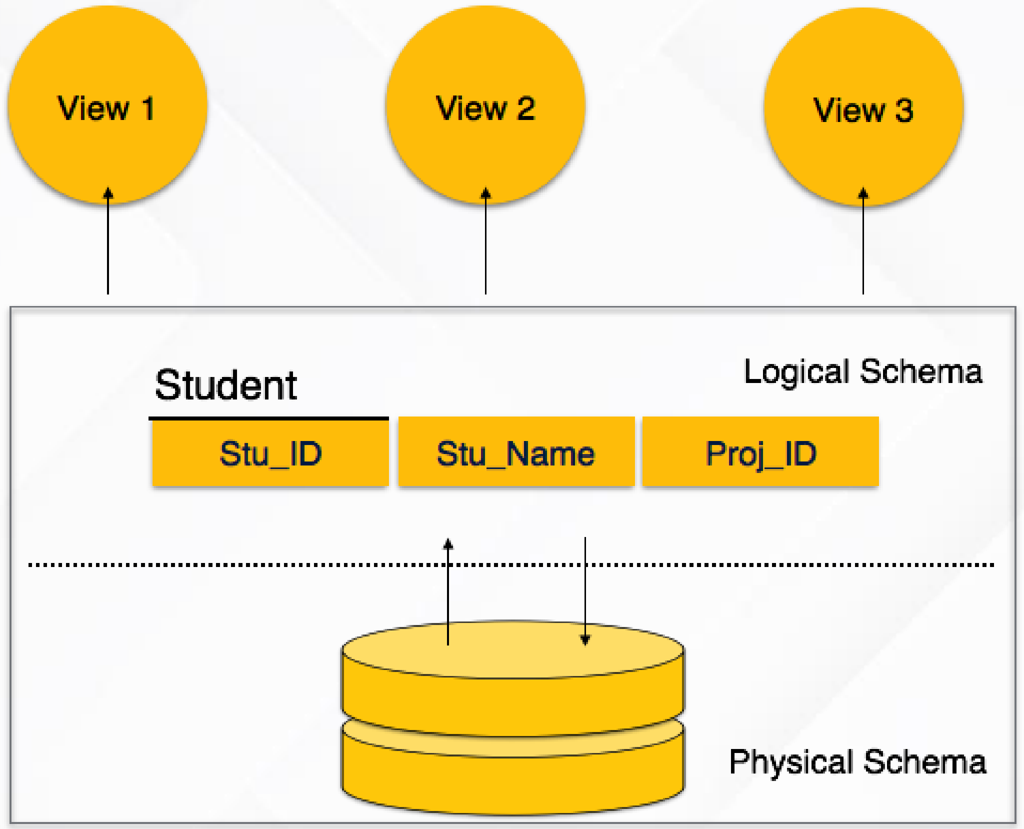
Source: https://www.learncomputerscienceonline.com/database-keys/

What is use of Database Keys ?


Identify Each Record In The Table
Define Relations Between The Tables
Establish Relations Between The Tables
Enforce Referential Integrity.
DATABASE KEYS

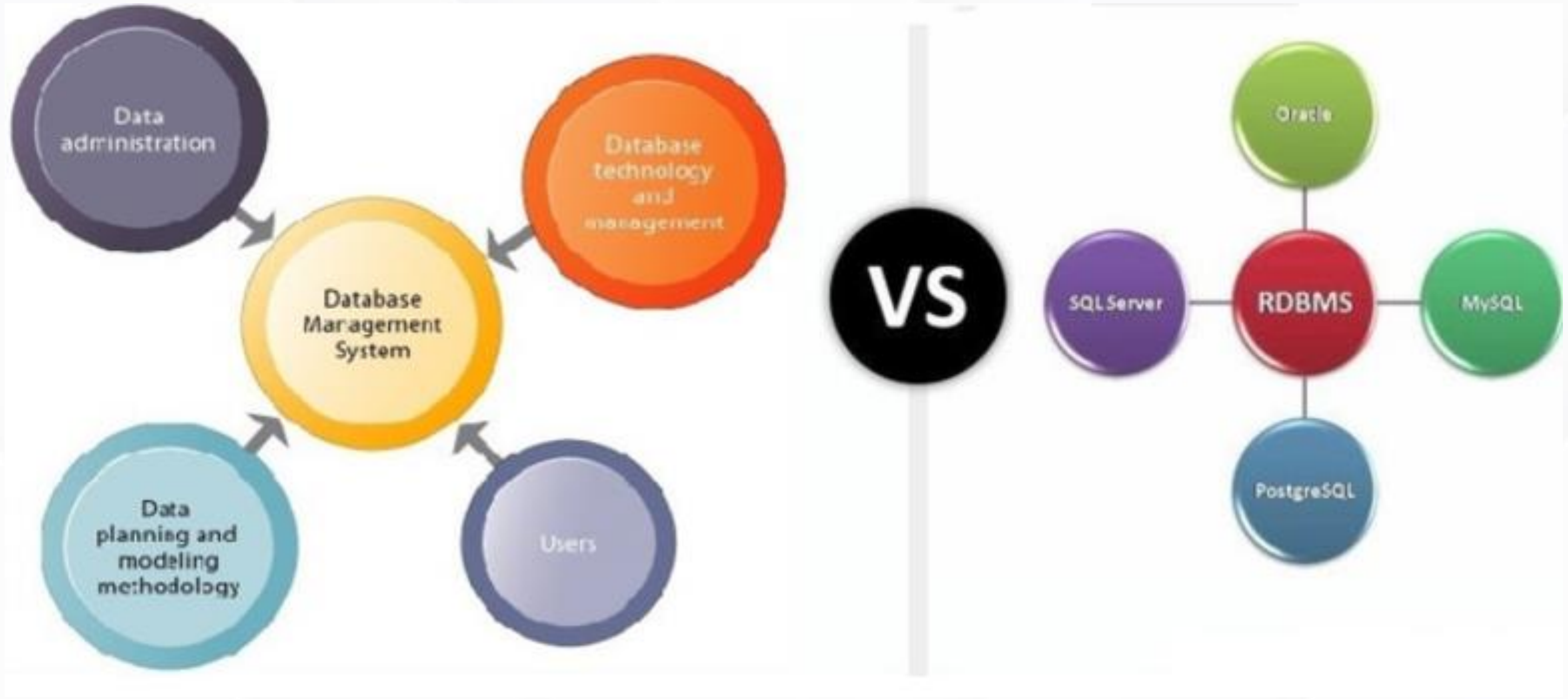
Source: https://www.learncomputerscienceonline.com/database-keys/

Database Schema and Schema Design



Source: https://www.tutorialspoint.com/dbms/dbms_data_schemas.htm

DBMS vs RDBMS


Data administration
Database technology and management
Database Management System
Data planning and modeling methodology
Users
VS
Oracle
SQL Server
RDBMS
MySQL
PostgreSQL

Source: https://www.tutorialsmate.com/2021/02/difference-between-dbms-and-rdbms.html

Class Work

Lab 13: Installation of DBMS Software

Solution: GitHub Link

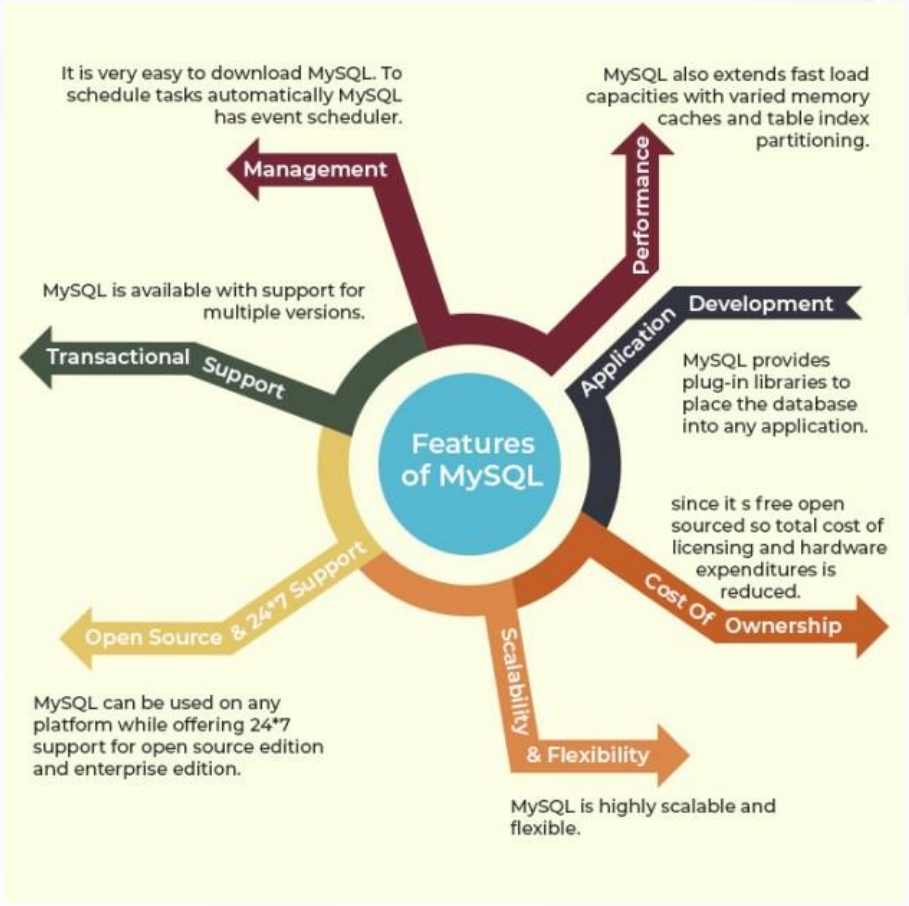
MySQL

What is MySQL?

- MySQL is released under an open-source license.
- MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
- MySQL works very quickly and works well even with large data sets.
- MySQL supports large databases, up to 50 million rows or more in a table.
- MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

Features of MySQL

- Speed
- Ease of use
- Cost
- Query Language Support
- Capability
- Connectivity and security
- Portability



It is very easy to download MySQL. To schedule tasks automatically MySQL has event scheduler.

MySQL also extends fast load capacities with varied memory caches and table index partitioning.

MySQL is available with support for multiple versions.

MySQL provides plug-in libraries to place the database into any application.

since it s free open sourced so total cost of licensing and hardware expenditures is reduced.

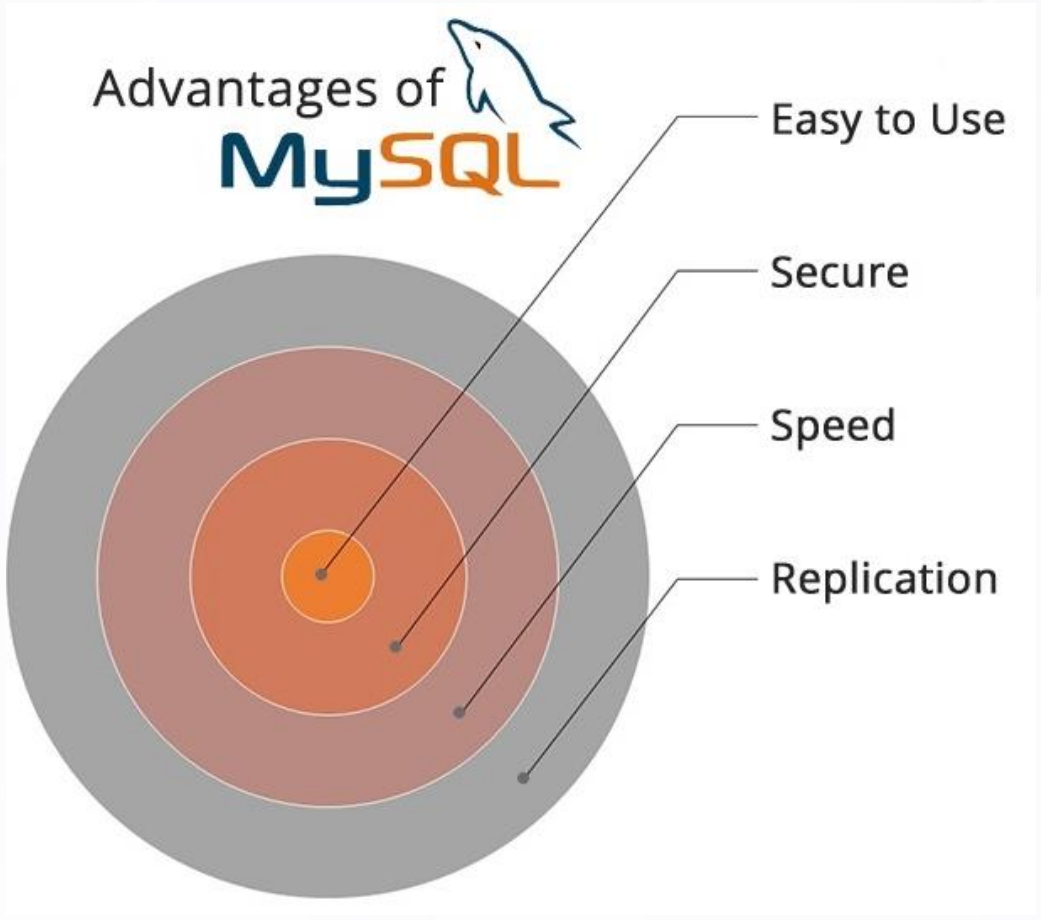
MySQL can be used on any platform while offering 24*7 support for open source edition and enterprise edition.

MySQL is highly scalable and flexible.

Source: https://i2.wp.com/d3d2ir91ztzaym.cloudfront.net/uploads/2020/06/Features-of-MySQL.jpg

Benefits of MySQL

- Flexible and easy to use
- High performance
- A mature DBMS
- Secure database
- Free installation
- Simple syntax



Source: https://www.janbasktraining.com/blog/uploads/images/image_750x_5dcd28ff0dfd4.jpg

Cons of MySQL

- Scalability issues
- Owned by Oracle
- Large databases
- Limited support for SQL standards
- Tools
- Data Corruption

Class Work

Lab 14: Creating the First Database

Solution: GitHub Link

Self Practice

<u>**Lab 14 - Create a database called movie and perform relevant operations**</u>

Solution: <u>GitHub Link</u>

Tables in DBMS

- In a Database Management System (DBMS), tables are the fundamental structures used to organize and store data.
- Tables are often referred to as relations, and they play a crucial role in defining the schema of a database.

Source: https://www.javatpoint.com/dbms-full-form

Class Work

<u>Lab 15: Creating Tables in Database</u>

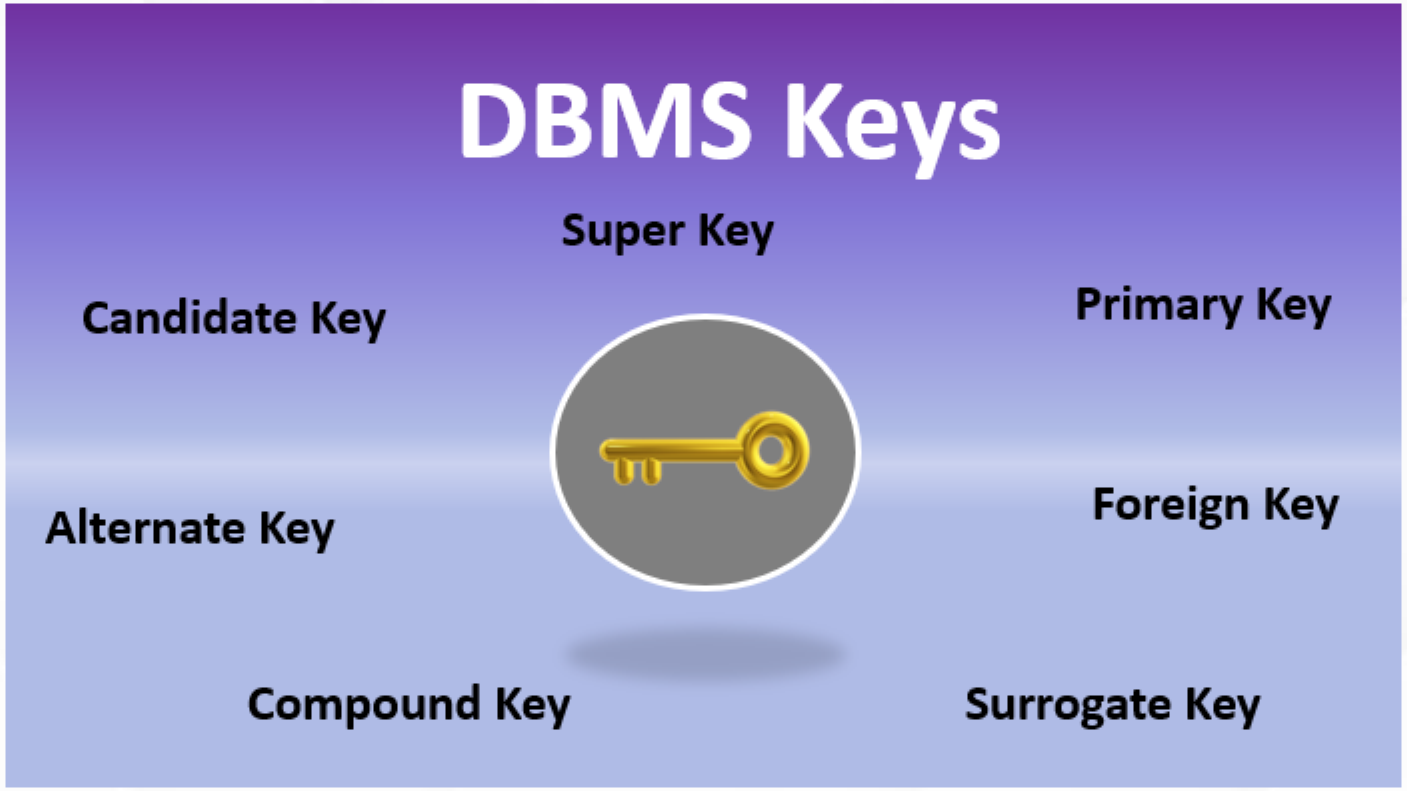
Solution: <u>GitHub Link</u>

Self Practice

Lab 15 - Write a SQL statement to create a table employees including columns employee_id, first_name, last_name, job_id, salary and make sure that, the employee_id column does not contain any duplicate value at the time of insertion, and the foreign key column job_id, referenced by the column job_id of jobs table, can contain only those values which are exists in the jobs table. The InnoDB Engine have been used to create the tables. The specialty of the statement is that, The ON UPDATE CASCADE action allows you to perform cross-table update and ON DELETE RESTRICT action reject the deletion. The default action is ON DELETE RESTRICT.

Solution: GitHub Link

Different Keys in DBMS



Source: https://www.educba.com/dbms-keys/

Keys in DBMS

In DBMS, keys are important concepts that play a crucial role in:

- Defining the relationships between tables
- Ensuring data integrity
- Facilitating efficient data retrieval

Source: https://www.javatpoint.com/dbms-full-form

Class Work

Lab 16 : Use of Different Keys

Solution: GitHub Link

Self Practice

Lab 16 - In a table, you establish relations between different columns. A key helps to unique identify a row. Let's have some practical understanding of different keys in DBMS. So, without losing anymore time, let's start the hands-on session.

Solution: GitHub Link

Structured Query Language (SQL)

SQL is a language used for interacting with Relational Database Management System (RDBMS)

You can use SQL to get the RDBMS to :

- Create, retrieve, update & delete data
- Create & manage databases
- Design & create databases tables
- Perform administration tasks (security, user management, import/export etc.)

Categories of SQL Statements

DDL (Data Definition Language):

DDL statements are used to alter/modify a database or table structure and schema. These statements handle the design and storage of database objects.

- **CREATE** - create a new Table, database, schema
- **ALTER** - alter existing table, column description
- **DROP** - delete existing objects from database

Categories of SQL Statements

DML (Data Manipulation Language):

These are basic operations we perform on data such as selecting a few records from a table, inserting new records, deleting unnecessary records, and updating/modifying existing records.

- **SELECT** - select records from a table
- **INSERT** - insert new records
- **UPDATE** - update/Modify existing records
- **DELETE** - delete existing records

Categories of SQL Statements

DCL (Data Control Language):

DCL statements control the level of access that users have on database objects.

- **GRANT** - allows users to read/write on certain database objects
- **REVOKE** - keeps users from read/write permission on database objects

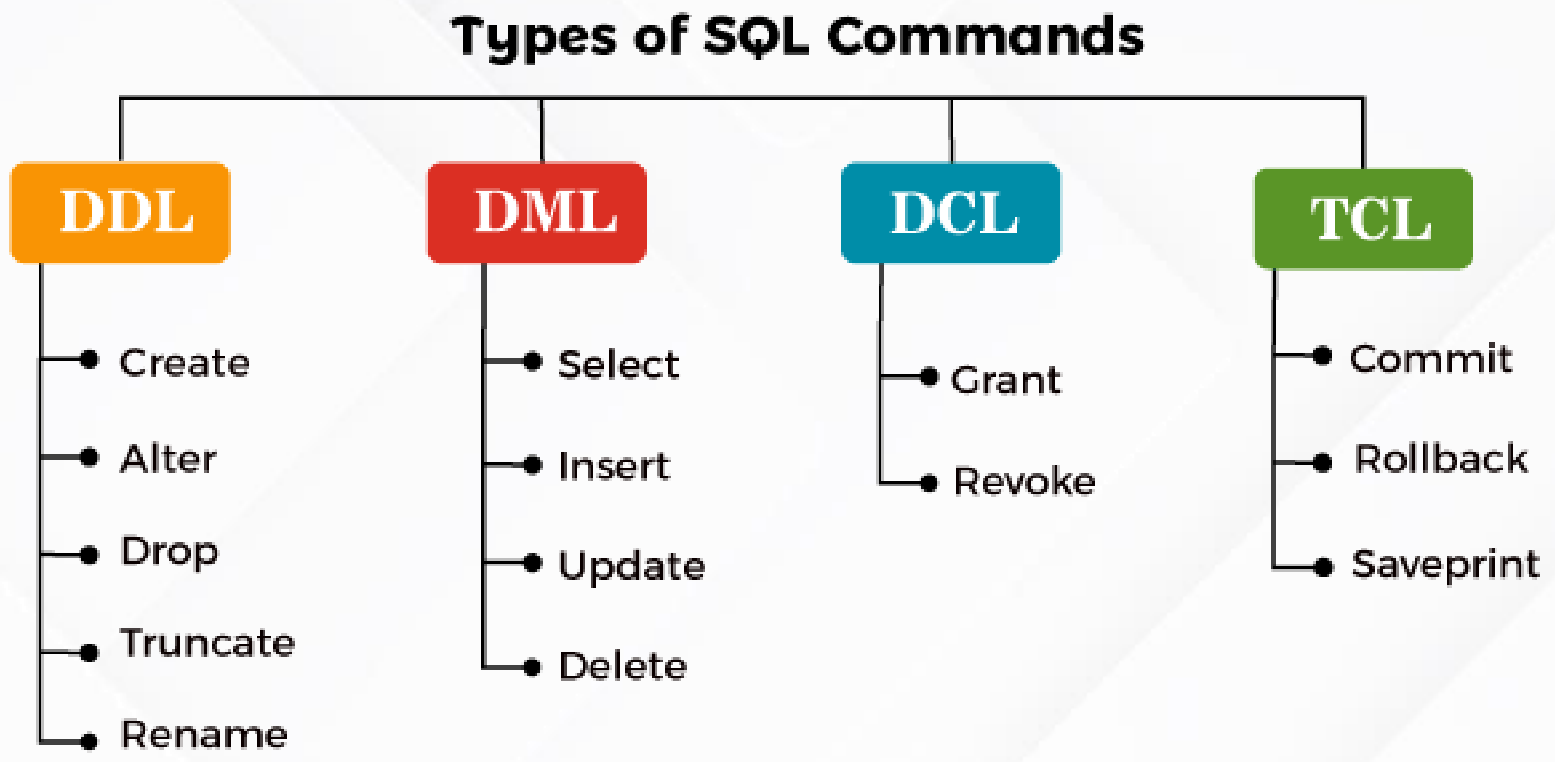
Categories of SQL Statements

TCL (Transaction Control Language)

TCL statements allow you to control and manage transactions to maintain the integrity of data within SQL statements.

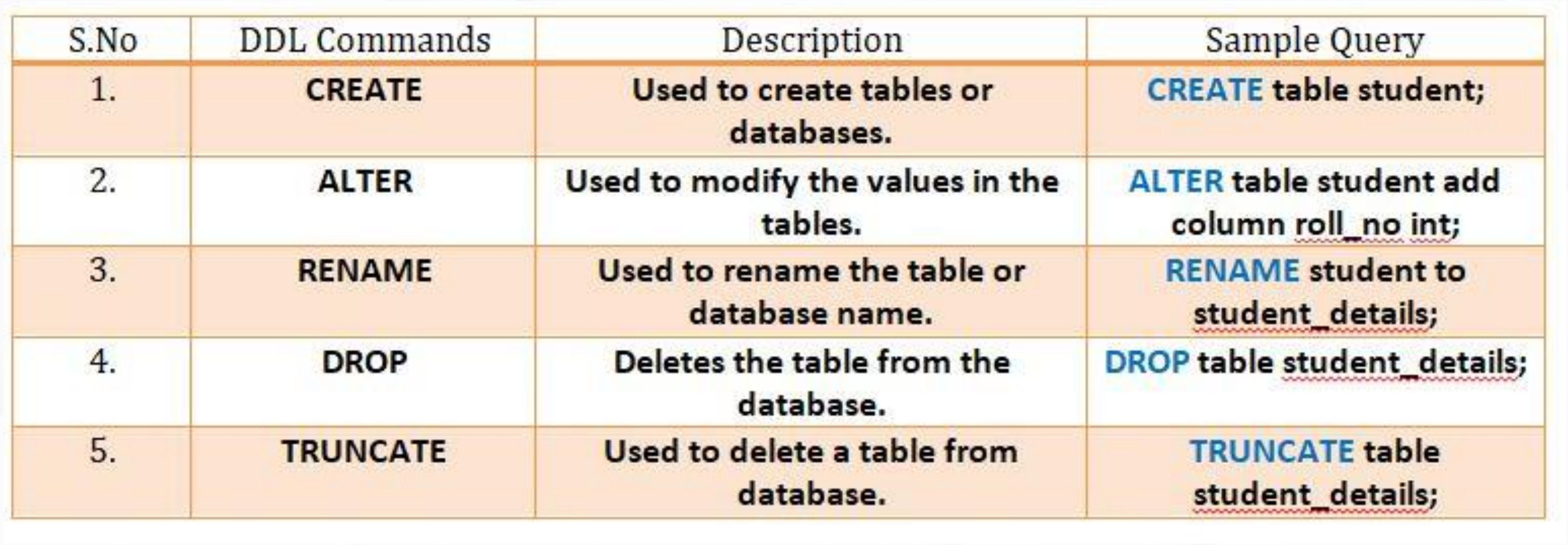
- **BEGIN Transaction** - opens a transaction
- **COMMIT Transaction** - commits a transaction
- **ROLLBACK Transaction** - ROLLBACK a transaction in case of any error

SQL Commands



Source: https://static.javatpoint.com/sqlpages/images/types-of-sql-commands.png

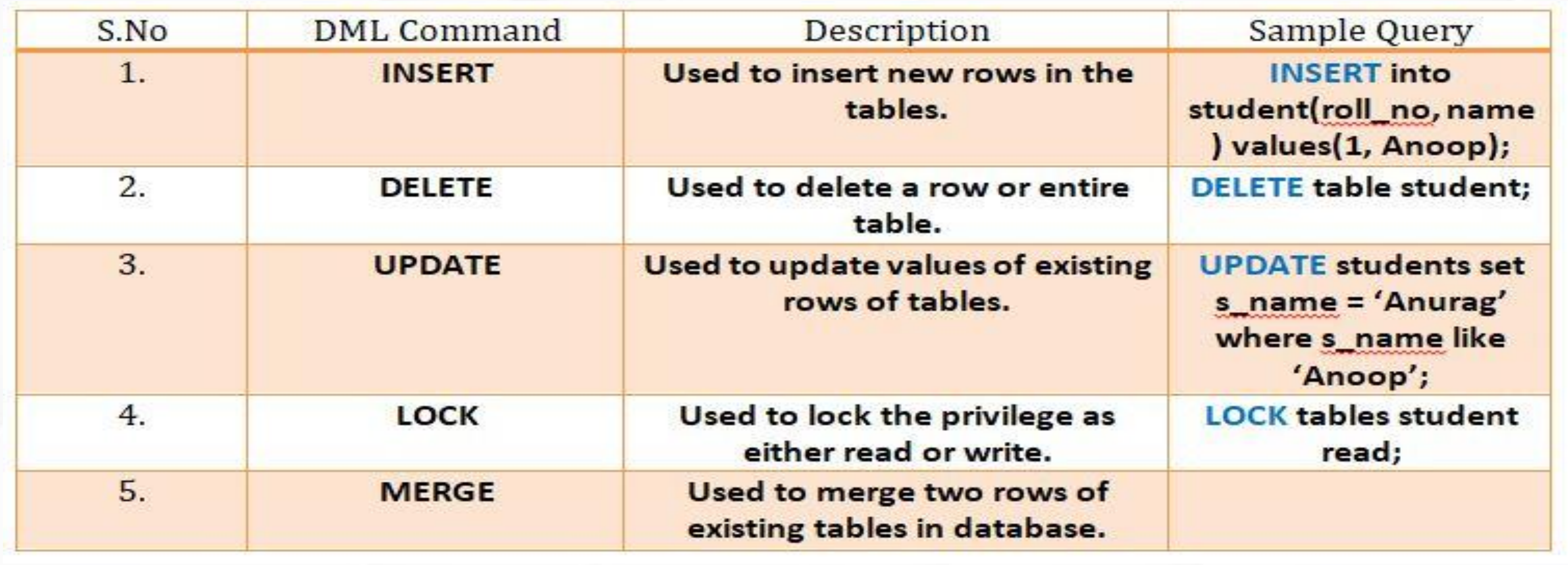
SQL Commands



S.No	DDL Commands	Description	Sample Query
1.	CREATE	Used to create tables or databases.	CREATE table student;
2.	ALTER	Used to modify the values in the tables.	ALTER table student add column roll_no int;
3.	RENAME	Used to rename the table or database name.	RENAME student to student_details;
4.	DROP	Deletes the table from the database.	DROP table student_details;
5.	TRUNCATE	Used to delete a table from database.	TRUNCATE table student_details;

Source: https://minigranth.in/sql-tutorial/images/tutorials/SQL%20IMAGES/DDL%20Commands.jpg

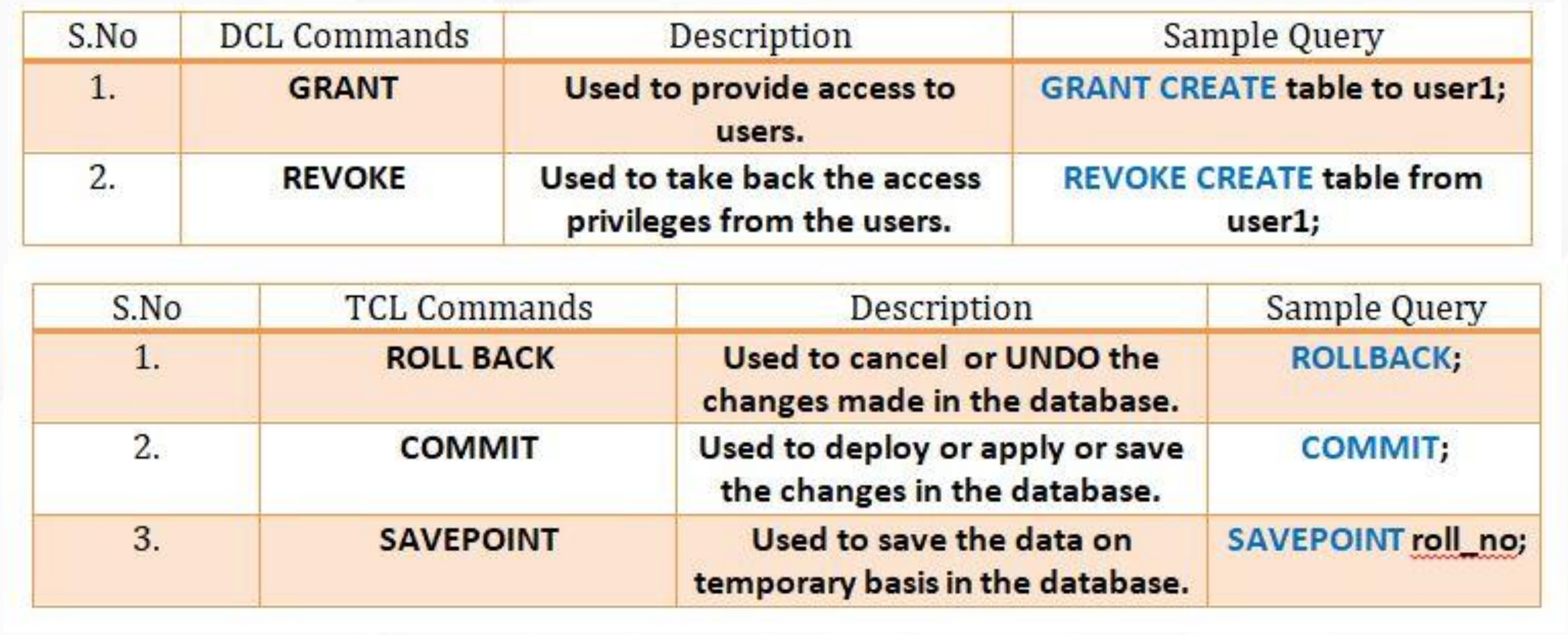
SQL Commands



S.No	DML Command	Description	Sample Query
1.	INSERT	Used to insert new rows in the tables.	INSERT into student(roll_no, name) values(1, Anoop);
2.	DELETE	Used to delete a row or entire table.	DELETE table student;
3.	UPDATE	Used to update values of existing rows of tables.	UPDATE students set s_name = 'Anurag' where s_name like 'Anoop';
4.	LOCK	Used to lock the privilege as either read or write.	LOCK tables student read;
5.	MERGE	Used to merge two rows of existing tables in database.	

Source: https://minigranth.in/sql-tutorial/images/tutorials/SQL%20IMAGES/DML%20Commands.jpg

SQL Commands



S.No	DCL Commands	Description	Sample Query
1.	**GRANT**	**Used to provide access to users.**	**GRANT CREATE table to user1;**
2.	**REVOKE**	**Used to take back the access privileges from the users.**	**REVOKE CREATE table from user1;**

S.No	TCL Commands	Description	Sample Query
1.	**ROLL BACK**	**Used to cancel or UNDO the changes made in the database.**	**ROLLBACK;**
2.	**COMMIT**	**Used to deploy or apply or save the changes in the database.**	**COMMIT;**
3.	**SAVEPOINT**	**Used to save the data on temporary basis in the database.**	**SAVEPOINT roll_no;**

Source: https://minigranth.in/sql-tutorial/images/tutorials/SQL%20IMAGES/DCL%20Commands.jpg

Class Work

- **Lab 17:** CRUD operations

Solution: GitHub Link

Self-Practice

Lab 17 – Write a mysql query to create a table called student and perform crud operations on it and display results in each steps

Solution: GitHub Link

Integration of Database Technologies with Python

Python Modules for Database Connection

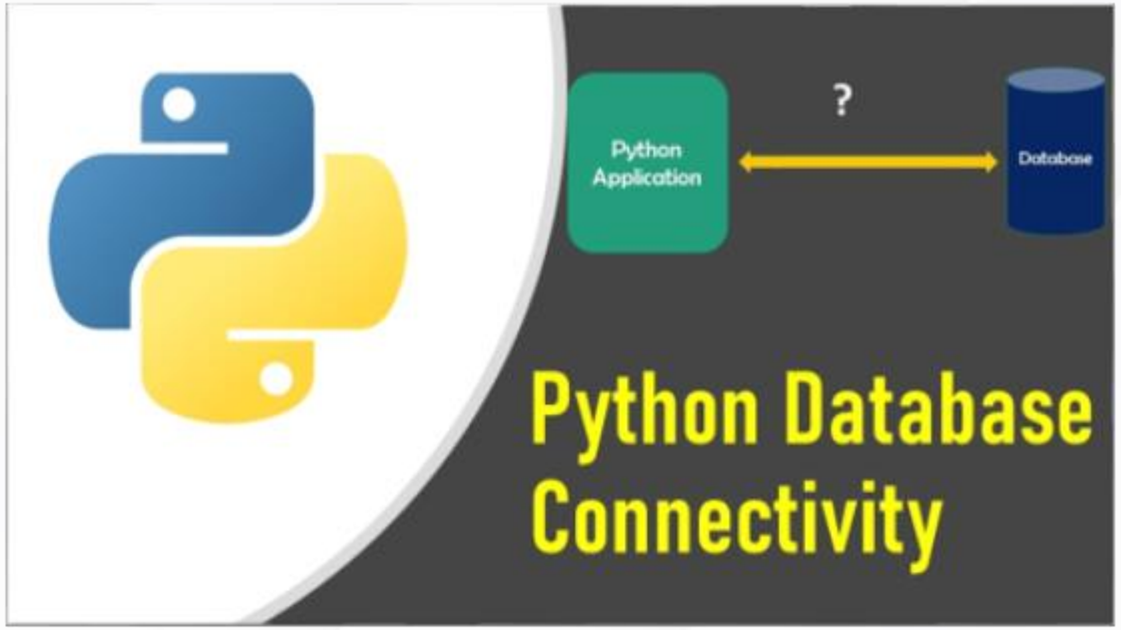
- In Python, we can use the following modules to communicate with MySQL

  - MySQL Connector Python
  - PyMySQL
  - MySQLDB
  - MySqlClient
  - OurSQL

- Above all interfaces or modules adhere to Python Database API Specification v2.0 (PEP 249) i.e., the **syntax, method, and way of accessing the database are the same in all**.

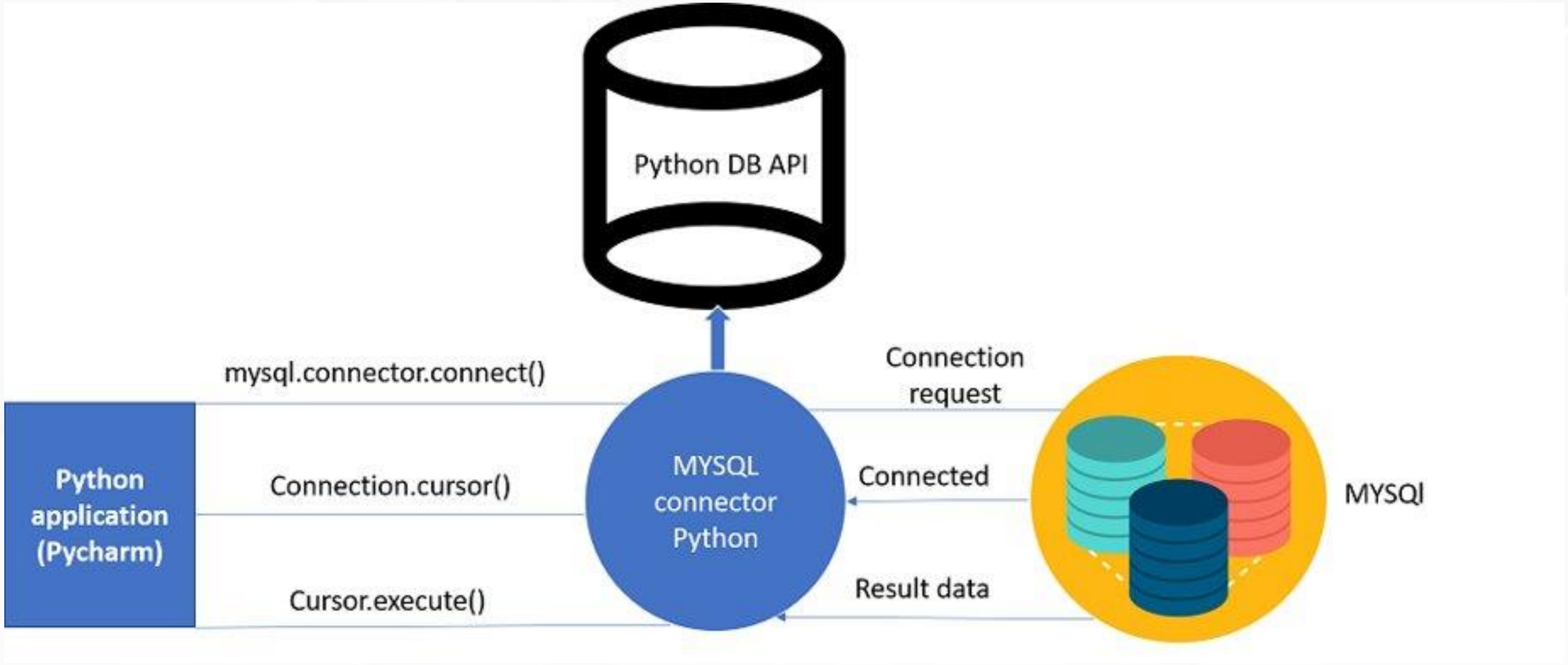
Python with Databases

- Python supports relational database systems.
- It is very easy to migrate and port database application interfaces (compatible database APIs).
- Python also supports Data Definition Language (DDL), Data Manipulation Language (DML) and Data Query Statements.



Source: [maxresdefault.jpg (1280×720) (ytimg.com)]

Procedure



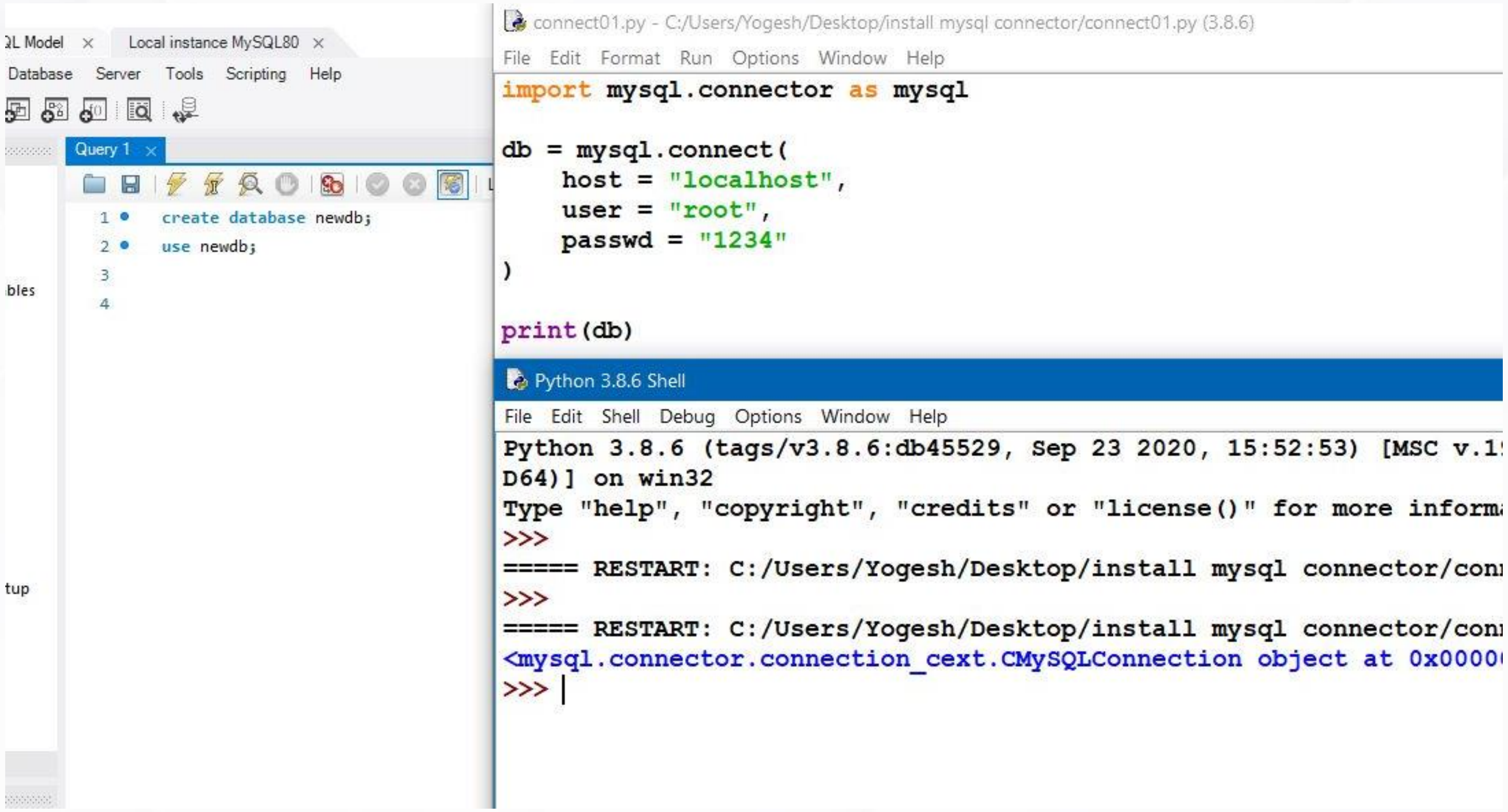
Source: https://yuiltripathee.medium.com/connect-your-database-in-python-code-works-almost-everywhere-21637b311bb4

Establishing a Database Connection

connect() method

Takes 4 parameters -

1. host
2. user
3. password
4. database


QL Model
Local instance MySQL80
Database Server Tools Scripting Help
Query 1
bles
tup
connect01.py - C:/Users/Yogesh/Desktop/install mysql connector/connect01.py (3.8.6)
File Edit Format Run Options Window Help
Python 3.8.6 Shell
File Edit Shell Debug Options Window Help

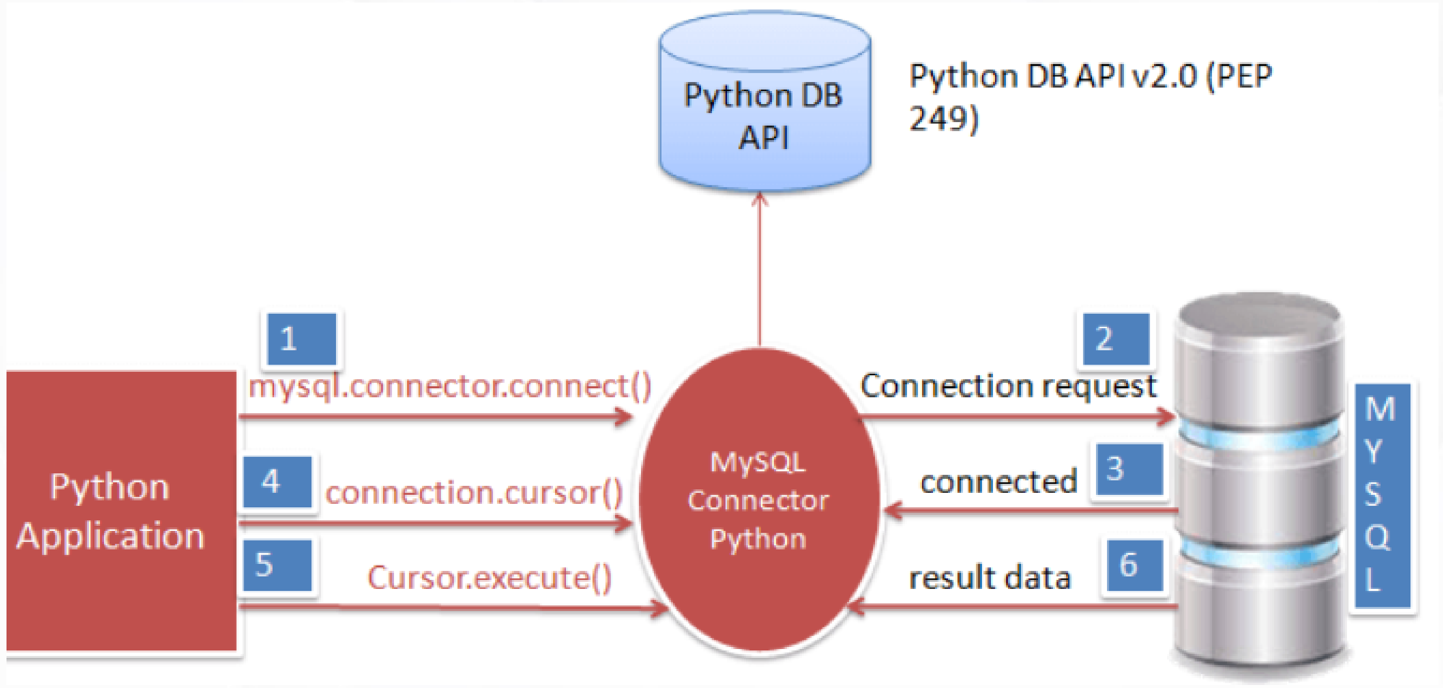
```sql
create database newdb;
use newdb;
```

```python
import mysql.connector as mysql

db = mysql.connect(
    host = "localhost",
    user = "root",
    passwd = "1234"
)

print(db)
```

```
Python 3.8.6 (tags/v3.8.6:db45529, Sep 23 2020, 15:52:53) [MSC v.1
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more inform
>>>
===== RESTART: C:/Users/Yogesh/Desktop/install mysql connector/con
>>>
===== RESTART: C:/Users/Yogesh/Desktop/install mysql connector/con
<mysql.connector.connection_cext.CMySQLConnection object at 0x0000
>>> |
```

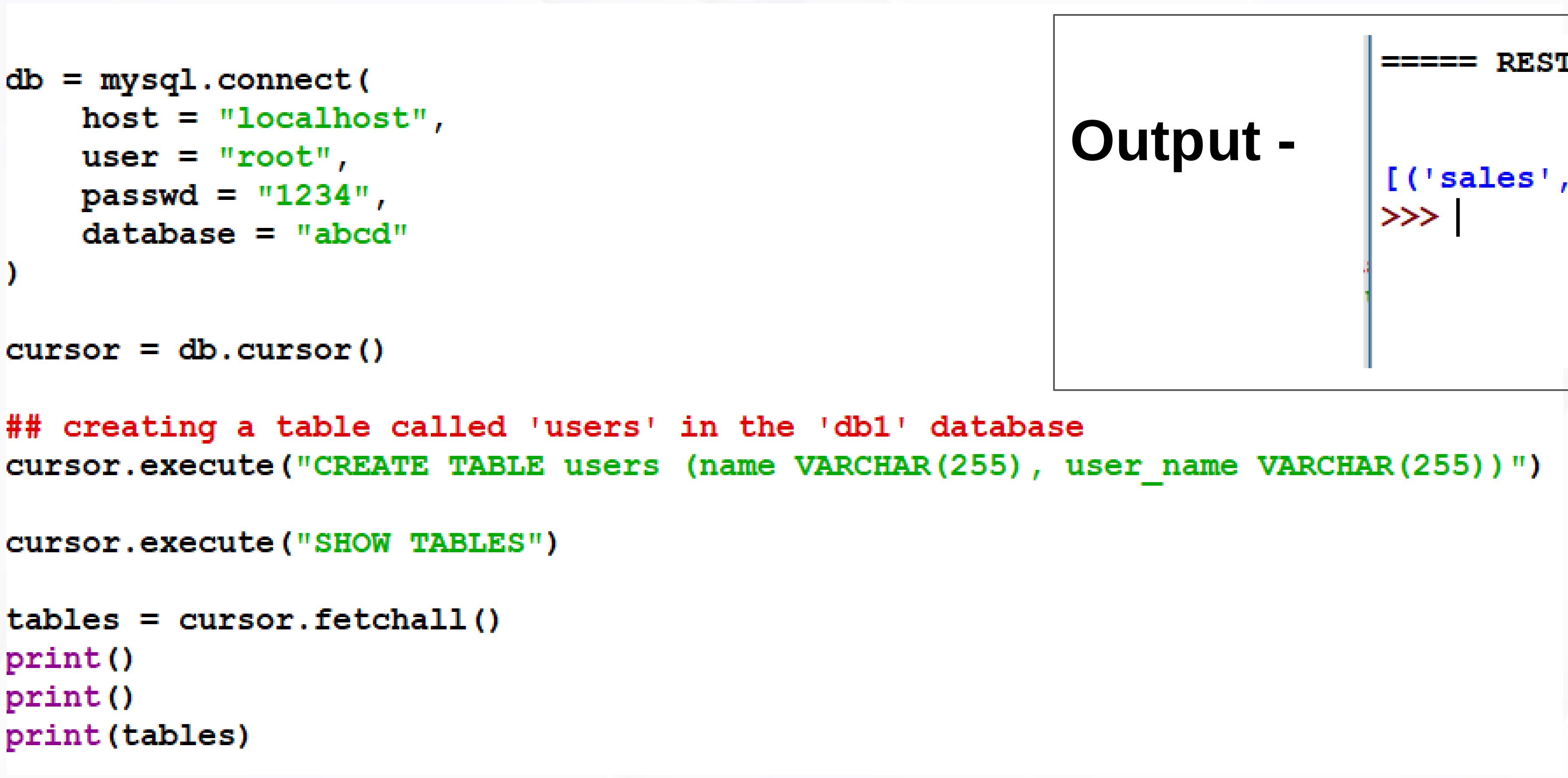
Creating Database Tables in Python

- cursor() - used to execute SQL statements
- execute() - used to compile SQL statements
- fetchall() - fetches all the rows from the last executed statement



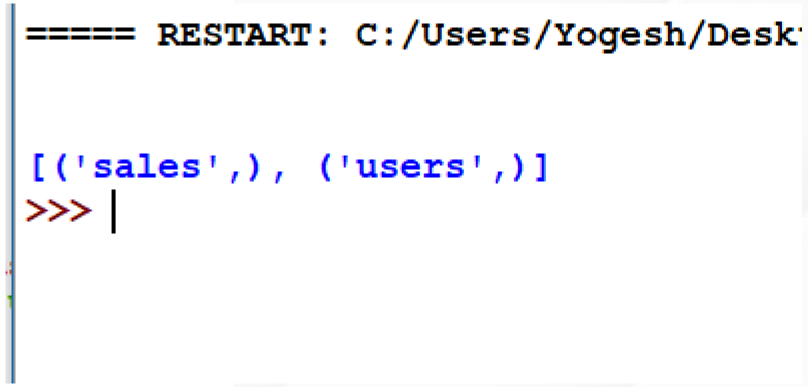
Source: py-db-connection-edureka.png (979×411)

Creating Database Tables in Python



```
db = mysql.connect(
    host = "localhost",
    user = "root",
    passwd = "1234",
    database = "abcd"
)

cursor = db.cursor()

## creating a table called 'users' in the 'db1' database
cursor.execute("CREATE TABLE users (name VARCHAR(255), user_name VARCHAR(255))")

cursor.execute("SHOW TABLES")

tables = cursor.fetchall()
print()
print()
print(tables)
```

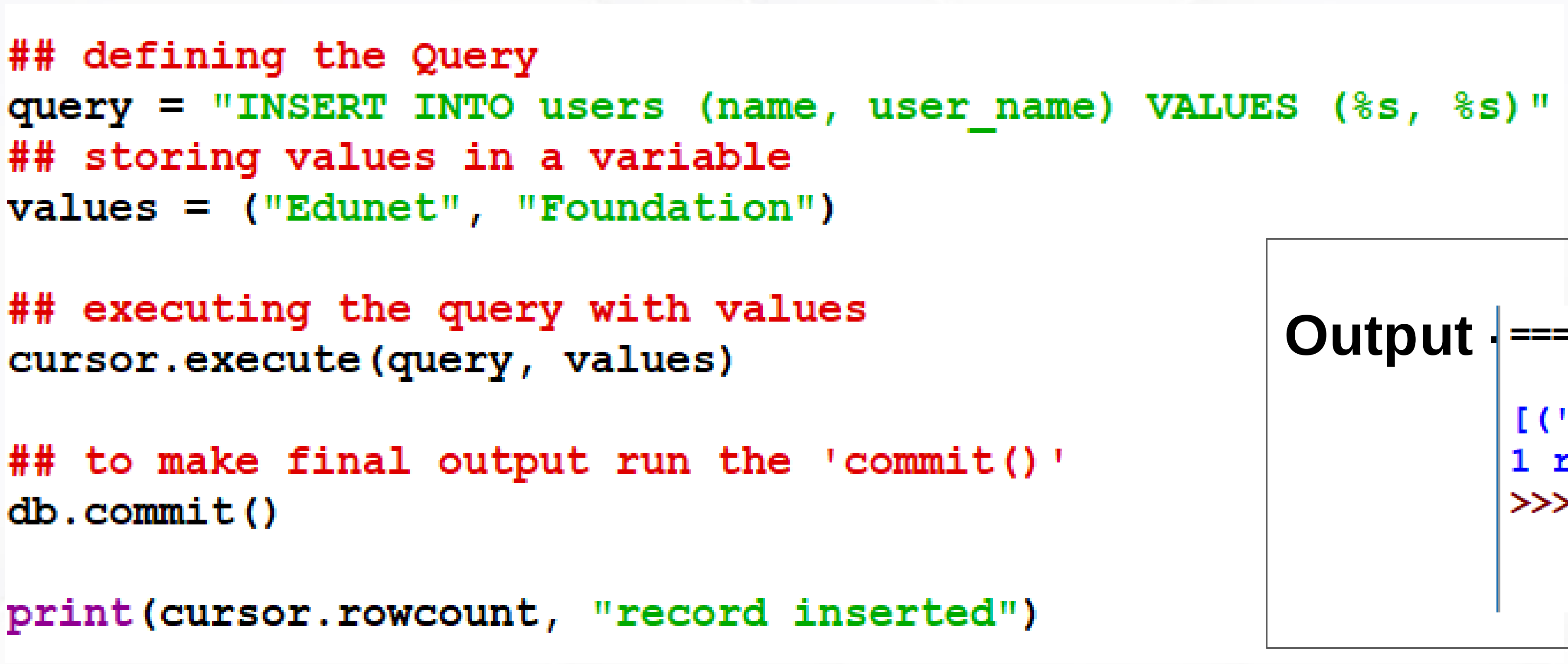
Output -



```
===== RESTART: C:/Users/Yogesh/Desk


[('sales',), ('users',)]
>>> |
```

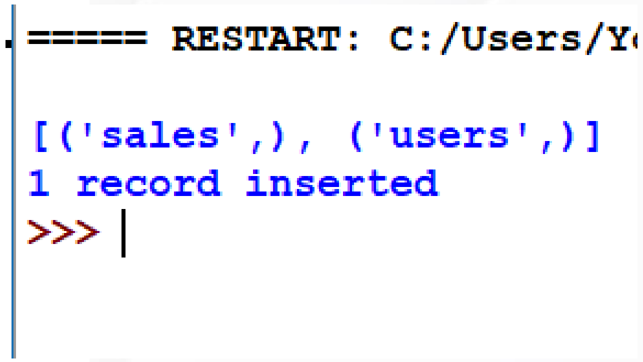
Database INSERT in Python

INSERT INTO table_name (column_names) VALUES (data)



```python
## defining the Query
query = "INSERT INTO users (name, user_name) VALUES (%s, %s)"
## storing values in a variable
values = ("Edunet", "Foundation")

## executing the query with values
cursor.execute(query, values)

## to make final output run the 'commit()'
db.commit()

print(cursor.rowcount, "record inserted")
```

Output

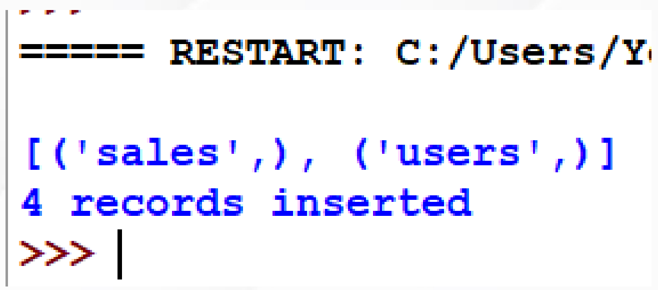


```
===== RESTART: C:/Users/Y

[('sales',), ('users',)]
1 record inserted
>>> |
```

Inserting Multiple rows in Python

```python
## defining the Query
query = "INSERT INTO users (name, user_name) VALUES (%s, %s)"
## storing values in a variable
values = [
    ("Edunet", "Foundation"),
    ("Amy", "Watson"),
    ("Michael", "Diwit"),
    ("Hennah", "Aziz")
]

## executing the query with values
cursor.executemany(query, values)

db.commit()

print(cursor.rowcount, "records inserted")
```

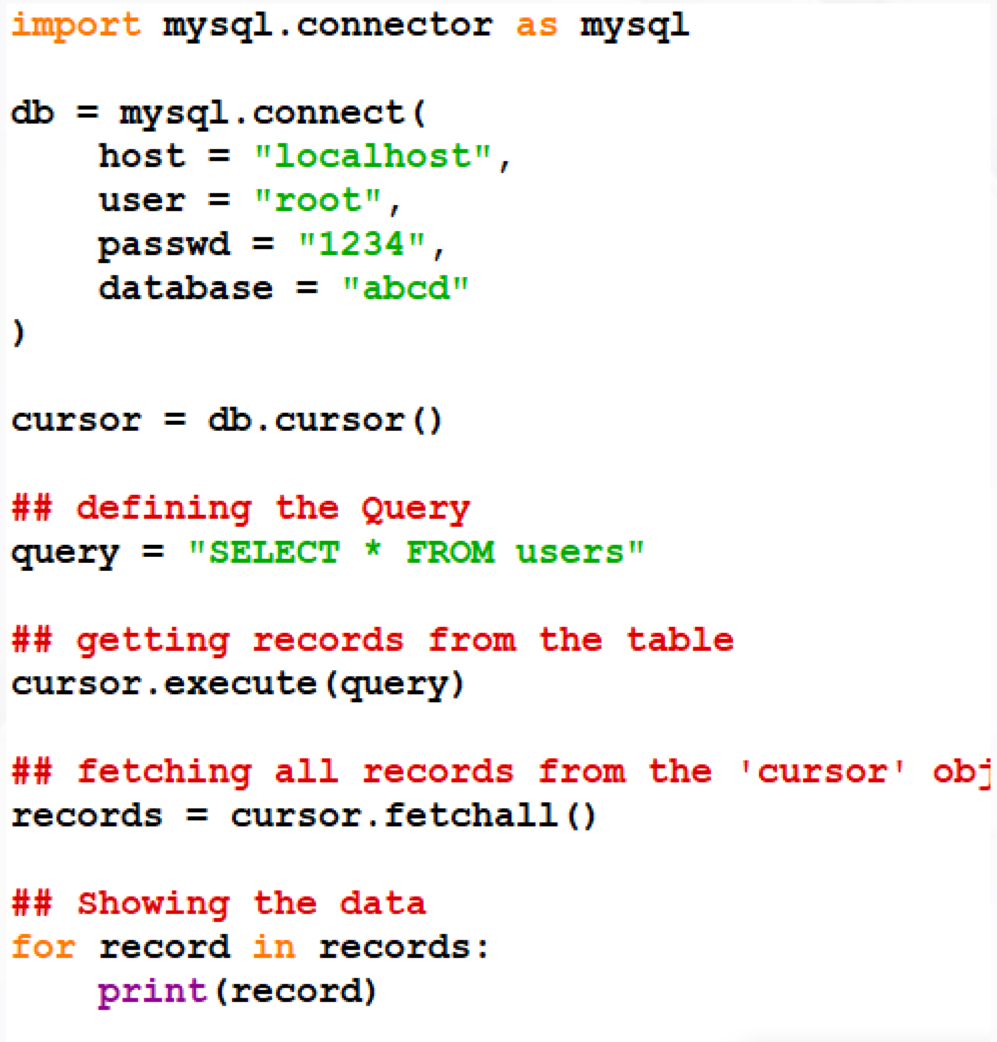
Output -



```
===== RESTART: C:/Users/Y

[('sales',), ('users',)]
4 records inserted
>>> |
```

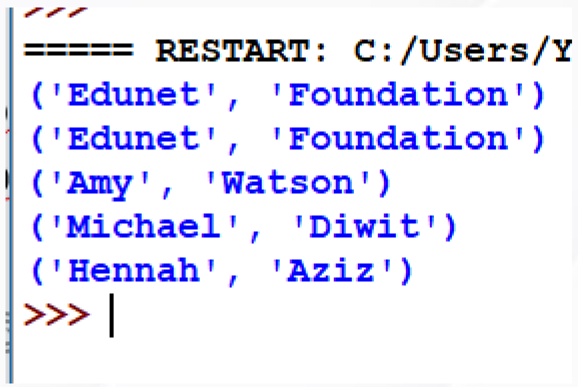
SELECT statement in Python

SELECT column_names FROM table_name



```python
import mysql.connector as mysql

db = mysql.connect(
    host = "localhost",
    user = "root",
    passwd = "1234",
    database = "abcd"
)

cursor = db.cursor()

## defining the Query
query = "SELECT * FROM users"

## getting records from the table
cursor.execute(query)

## fetching all records from the 'cursor' obj
records = cursor.fetchall()

## Showing the data
for record in records:
    print(record)
```

Output -



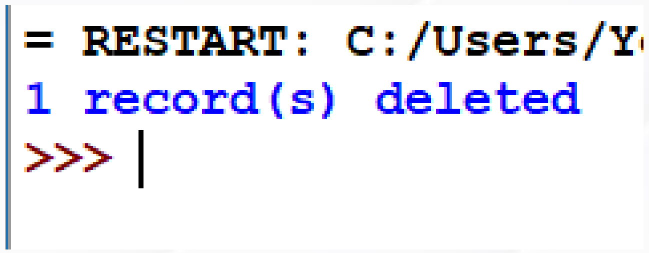
```
===== RESTART: C:/Users/Y
('Edunet', 'Foundation')
('Edunet', 'Foundation')
('Amy', 'Watson')
('Michael', 'Diwit')
('Hennah', 'Aziz')
>>> |
```

DELETE statement in Python

DELETE FROM table_name WHERE condition

```python
import mysql.connector

mydb = mysql.connector.connect(
  host="localhost",
  user="root",
  password="1234",
  database="abcd"
)

mycursor = mydb.cursor()

sql = "DELETE FROM customers WHERE address = 'Mountain 21'"

mycursor.execute(sql)

mydb.commit()

print(mycursor.rowcount, "record(s) deleted")
```

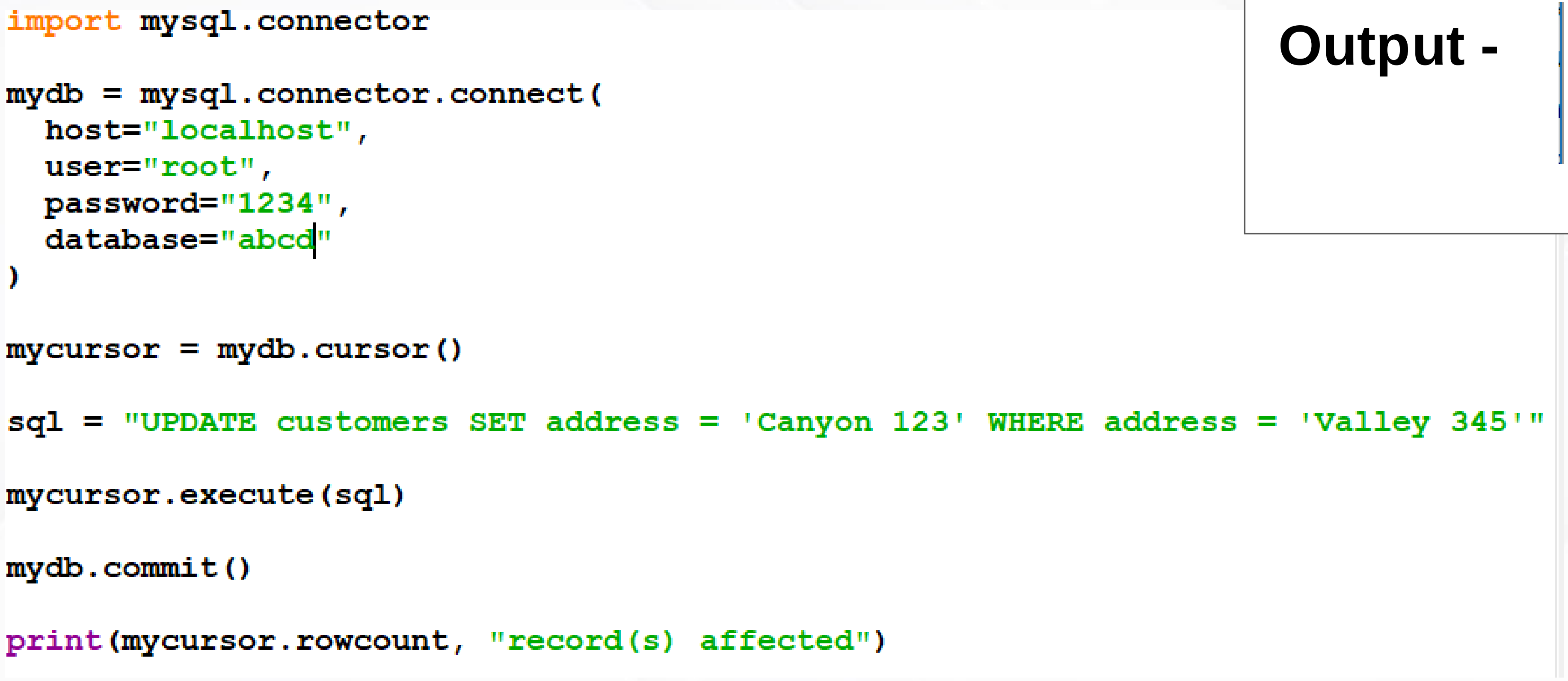
Output -



```
= RESTART: C:/Users/Y
1 record(s) deleted
>>> |
```

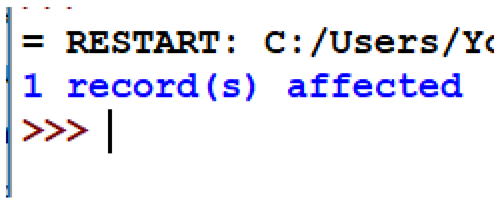
UPDATE Statement in Python

UPDATE table_name SET column_name = new_value WHERE condition



```
import mysql.connector

mydb = mysql.connector.connect(
  host="localhost",
  user="root",
  password="1234",
  database="abcd"
)

mycursor = mydb.cursor()

sql = "UPDATE customers SET address = 'Canyon 123' WHERE address = 'Valley 345'"

mycursor.execute(sql)

mydb.commit()

print(mycursor.rowcount, "record(s) affected")
```

Output -

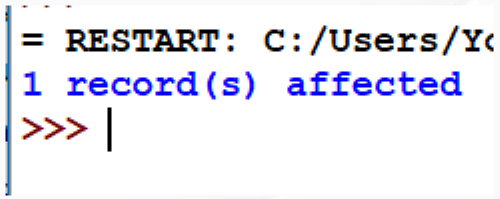


```
= RESTART: C:/Users/Yo
1 record(s) affected
>>> 
```

DROP Statement in Python

```
import mysql.connector

mydb = mysql.connector.connect(
  host="localhost",
  user="root",
  password="1234",
  database="abcd"
)


mycursor = mydb.cursor()

sql = "DROP TABLE customers"

mycursor.execute(sql)
```

Output -



```
= RESTART: C:/Users/Y
1 record(s) affected
>>> 
```

Class Work

Lab 18: Database Connectivity in Python

Solution: GitHub Link

Self Practice

Lab 18

- Create a table named EMPLOYEE in MySQL with five columns namely, FIRST_NAME,AGE, SEX and, INCOME.
- Display created table using SELECT
- Write a MySQL statement which retrieves the records of the employees whose income is greater than 4000.

Solution: GitHub Link

Introduction to Python Popular Frameworks

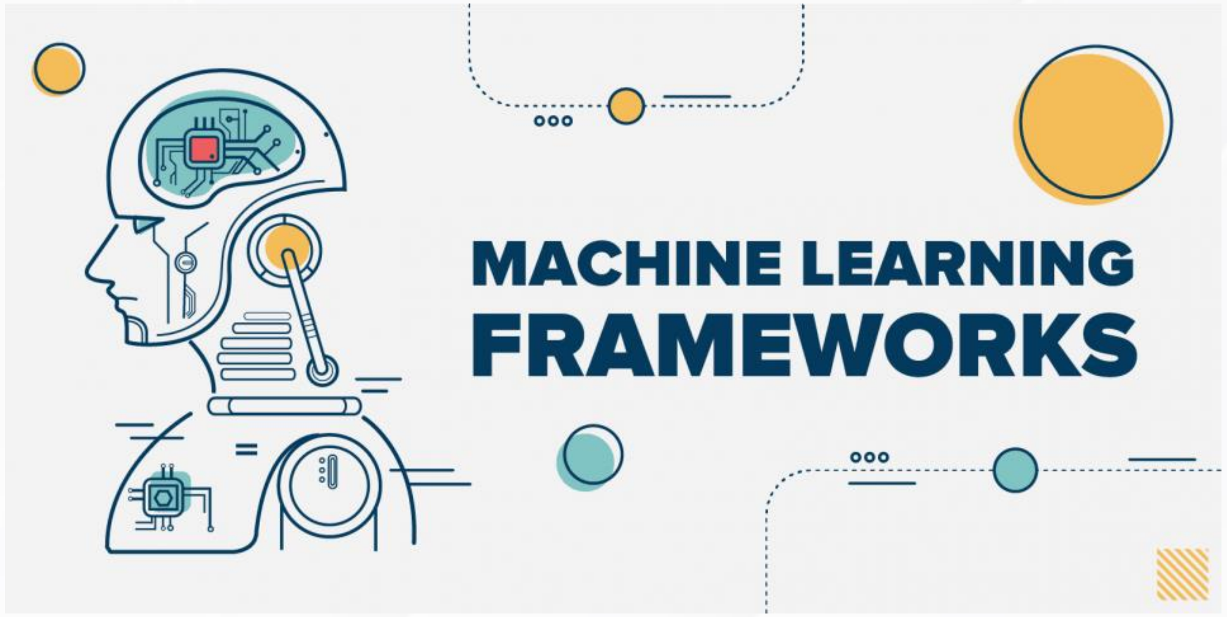
Python is one of the most popular and effective programming languages that contain vast libraries and frameworks for almost every technical domain. Python frameworks automate the implementation of several tasks and give developers a structure for application development. Each framework comes with its own collection of modules or packages that significantly reduce development time.

Python Frameworks for ML and DL

scikit learn

Keras

PyTorch



Source: [1588602415d0xfhuoXiu.png (960×480) (hackr.io)]

Python Frameworks Web Development

Python frameworks available in the market for web development. Depending on the functionality and key features they provide to the user, below are the frameworks available for web development.

Django

Flask

Source: Python-Frameworks-for-Web-Development.jpg (1068×601) (tekkiwebsolutions.com)

Python App using Database (Sqlite)

Personal Contacts Management System

Let's create a small project for managing a simple database using Python. We'll use SQLite, a lightweight and disk-based database, and the sqlite3 module in Python, which provides an easy-to-use interface to SQLite databases.

Summary

- DBMS helps to achieve efficient data management, data integrity and reliability.
- Relational Database Management System (RDBMS) is a software that organizes and manages data in a structured way using tables with predefined relationships.
- MySQL is an open-source Relational Database Management System (RDBMS) known for its speed and reliability in managing structured data.

Quiz

1. In a relational database, what is a primary key used for?

a) Storing large binary data
b) Establishing relationships between tables
c) Sorting data within a table
d) Uniquely identifying each row in a table

Answer: d
Uniquely identifying each row in a table

Quiz

2. Which of the following is NOT a primary role of a Database Management System (DBMS)?

a) Data storage

b) Data retrieval

c) User interface design

d) Data manipulation

Answer: c
User interface design

Quiz

3. What is the purpose of SQL (Structured Query Language)?

a) To design user interfaces

b) To manage hardware components

c) To manage and manipulate databases

d) To create graphical elements

Answer: c

To manage and manipulate databases

Reference

- DBMS Tutorial | What is a Database Management System? - javatpoint
- Introduction to Database (w3schools.in)
- https://www.mysql.com/
- Introduction to SQL Fundamentals (thoughtco.com)

Thank you!