

# My SQL for Beginners

The background is a deep blue gradient. Overlaid on this are several glowing, concentric circular nodes in shades of blue and white. These nodes are connected by a network of thin, white, circuit-like lines that branch out across the frame. The overall aesthetic is futuristic and technological, suggesting a theme of data, computing, or digital networks.

# Table of contents

- Overview
- Types of data
  - Structured data
  - Unstructured data
  - Semi Structure data
- Difference b/n each data's
- SQL Commands
- Data types:
- Constraints
- Install and Server connect

# Overview

**SQL** stands for **Structured Query Language** which is a computer language for storing, manipulating and retrieving data stored in a relational database. SQL was developed in the 1970s by IBM, it is a language to operate databases. It includes Database Creation, Database Deletion, Fetching Data Rows, Modifying & Deleting Data rows, etc.

RDBMS stands for Relational Database Management System. RDBMS is the basis for SQL, and for all modern database systems like MS SQL Server, IBM DB2, Oracle, MySQL, and Microsoft Access.

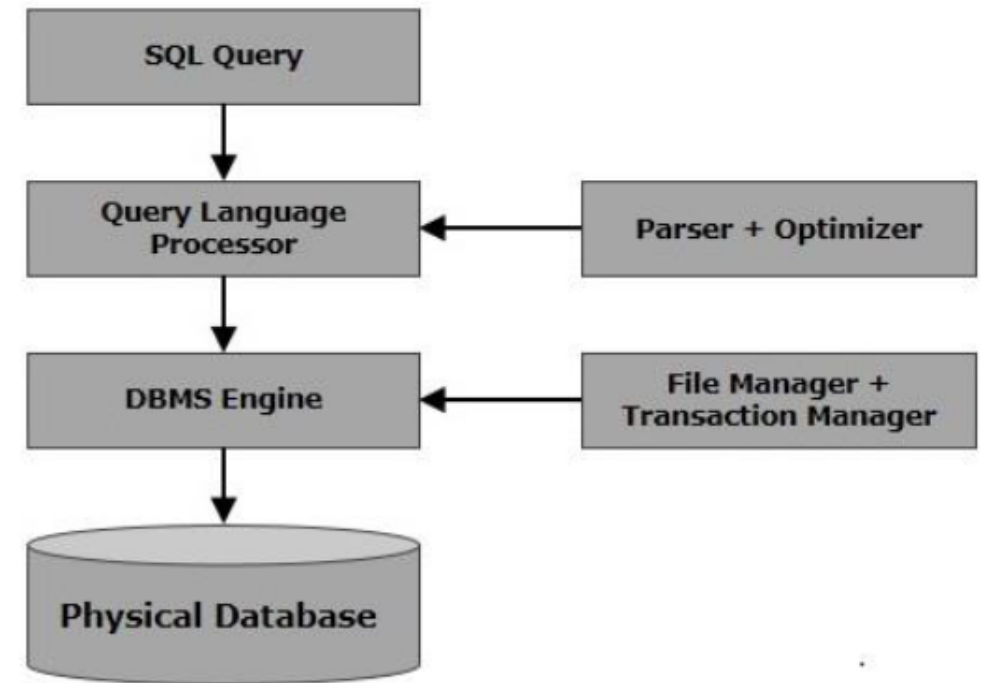
the most popular RDBMS are listed below –

- ❑ MySQL
- ❑ MS SQL Server
- ❑ ORACLE
- ❑ MS ACCESS
- ❑ PostgreSQL
- ❑ SQLite


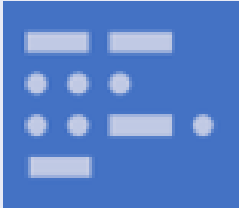
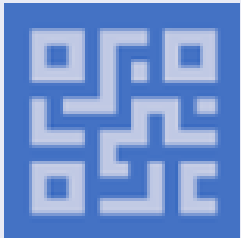
## SQL Applications

SQL is one of the most widely used Query Language over the databases. SQL provides following functionality to the database programmers –

- ❑ Executes different database queries against a database.
- ❑ Defines the data in a database and manipulates that data.
- ❑ Creates data in a relational database management system.
- ❑ Accesses data from the relational database management system.
- ❑ Creates and drops databases and tables.
- ❑ Creates and maintains database users.
- ❑ Creates views, stored procedures, functions in a database.
- ❑ Sets permissions on tables, procedures and views.



# Types of Data

	Structured data	Semi-Structured data	Unstructured data
Characteristics	<p>Defined is well organized,</p> <p>Organized means – relational database,</p> <p>Matured transaction, multiple concurrency techniques</p> <p>Tuples, rows and tables</p> <p>Schema dependent and less flexible</p> <p>Query performance is the highest, structured query can be performed allowing complex joins</p>	<p>Data is organized to some extent</p> <p>Partially organized, e.g by XML/RDF</p> <p>Transaction is adapted from DBMS, but data concurrency can pose problems</p> <p>Tuples or graphs are possible</p> <p>Data is more flexible than structured</p> <p>Queries over anonymous nodes are possible</p>	<p>Data is fully non organized</p> <p>Based on character and binary data</p> <p>Difficult but achievable transaction management and data concurrency</p> <p>Versioning usually on whole data or chunks</p> <p>The most flexible</p> <p>Schema on-read so query performance is the lowest</p>
Examples	<p>Transactional information, Names, Dates and Addresses</p> <ul style="list-style-type: none"> <li>•Survey</li> <li>•Questionnaires</li> <li>•Tests</li> <li>•Claim Forms</li> </ul> 	<p>XML/JSON Data, HTML, Emails, Web pages</p> <ul style="list-style-type: none"> <li>•Invoices</li> <li>•Purchase Orders</li> <li>•Bills of Lading</li> <li>•Explanation of Benefits</li> </ul> 	<p>Documents - PDFs, Text files</p> <ul style="list-style-type: none"> <li>•ViContracts</li> <li>•Letters</li> <li>•Articles</li> <li>•Memos</li> </ul> <p>deos, Audio, Images files,</p> 

Structured



Semi Structured



Unstructured



# Structure

- Structured data is organized and easily searchable with a predefined format, like data in a table.
- Definition: Data that conforms to a predefined data model, typically organized in rows and columns, like in a relational database.
- Examples: Dates, numbers, product SKUs, customer information stored in a CRM system.
- Characteristics:
  - Well-defined schema and format.
  - Easily searchable using SQL (Structured Query Language).
  - Data integrity is enforced through validation rules.
  - Typically stored in data warehouses.
- Use Cases: Customer relationship management (CRM), inventory management, financial transactions.
- Benefits: Easy to analyze, manage, and retrieve information.
- Drawbacks: Can be inflexible and may require significant effort to adapt to new requirements.

Most popular tools for the management of structured data

**MySQL:** Embedding data in mass-deployed software.

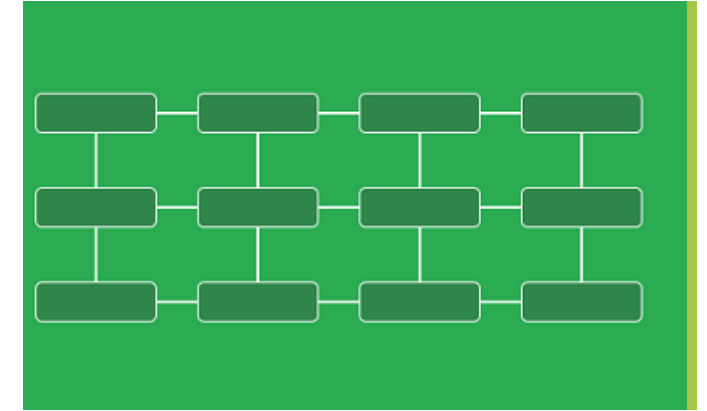
•**OLAP (Online Analytical Processing):** Data analysis.

•**SQLite:** Relational database.

•**Oracle database:** Advanced database management system.



Structured		
1001	1001	0101
1010	1110	1110
1110	0110	1000
1001	1001	1001
1010	0010	1010
1000	1110	1110
1001	1001	0101
1010	1010	1110
1110	1000	1000



## Pros of using structured data

- Easy for the average user to utilize and understand
- Easy for machine learning algorithms to utilize
- A greater number of analytics tools can use the data
- Requires less storage space

## Cons of using structured data

- Limited to specific uses
- More limited storage options
- Difficult and expensive to make changes

## Use cases for utilizing structured data

- **Ecommerce:** Product IDs, pricing data and customer account data
- **Healthcare:** Patient forms, medical insurance data and medical billing data
- **Banking:** Customer account data and financial transactions
- **Customer relationship management (CRM) software:** Names, phone numbers and addresses
- **Travel industry:** Reservation data, ticket pricing information and dates

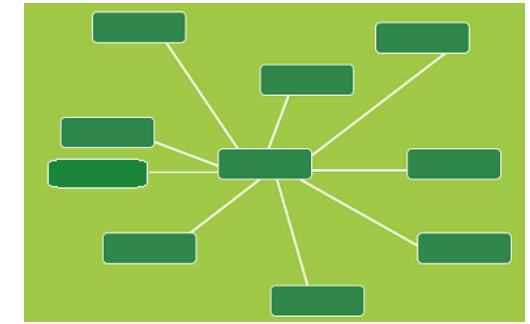
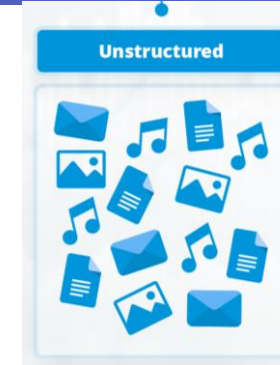
## Examples of structured data





# Unstructure

- Unstructured data lacks this predefined structure and can include various forms like text, images, and audio, making it harder to search and analyze directly. Most of the world's data is unstructured, highlighting the need for specialized tools and techniques to manage and extract insights from it.
  - Definition: Data that doesn't conform to a predefined structure and has no easily searchable format.
  - Examples: Social media posts, images, videos, audio files, emails, text documents.
  - Characteristics:
    - Lacks a predefined format or schema.
    - More difficult to search and analyze directly.
    - Often requires specialized tools and techniques for processing.
    - Can be stored in data lakes.
  - Use Cases: Sentiment analysis, image recognition, fraud detection, personalized recommendations.
  - Benefits: Provides rich, contextual information and can reveal hidden patterns.
  - Drawbacks: Requires significant effort to process and extract meaningful insights.
- 
- **Ecommerce:** Identify spending patterns and customer behavior
  - **Healthcare:** Determine treatment recommendations and forecast changes in a patient
  - **Finance:** Track markets and perform risk analysis



## Pros of using unstructured data

- Easier to store due to being in native format
- Collecting and storing are faster
- Cheaper to store unstructured data using data lakes
- Provides more granular information

## Cons of using unstructured data

- More complicated to work with
- Requires highly specialized tools for organizing
- Expertise needed

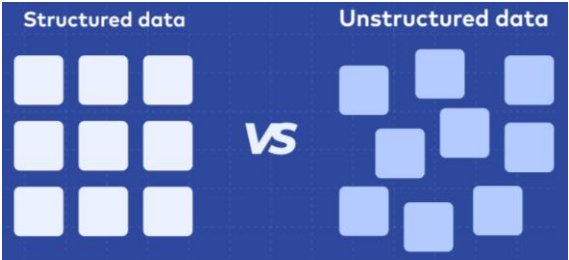
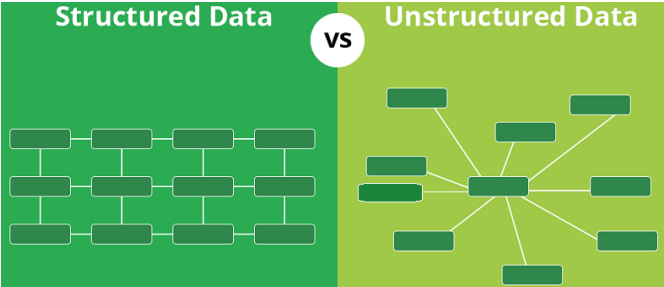
Most popular tools for working with unstructured data include:

- NoSQL (not only Structured Query Language) Database Management System (non-tabular database)
- MongoDB
- Apache Hadoop
- Microsoft Azure
- Amazon DynamoDB
- Video (WMV, MP4 and MOV)
- Images (JPG, PNG and GIF)
- Audio (MP3, WAV and MPEG)

# Semistruktur



# Difference between each structure

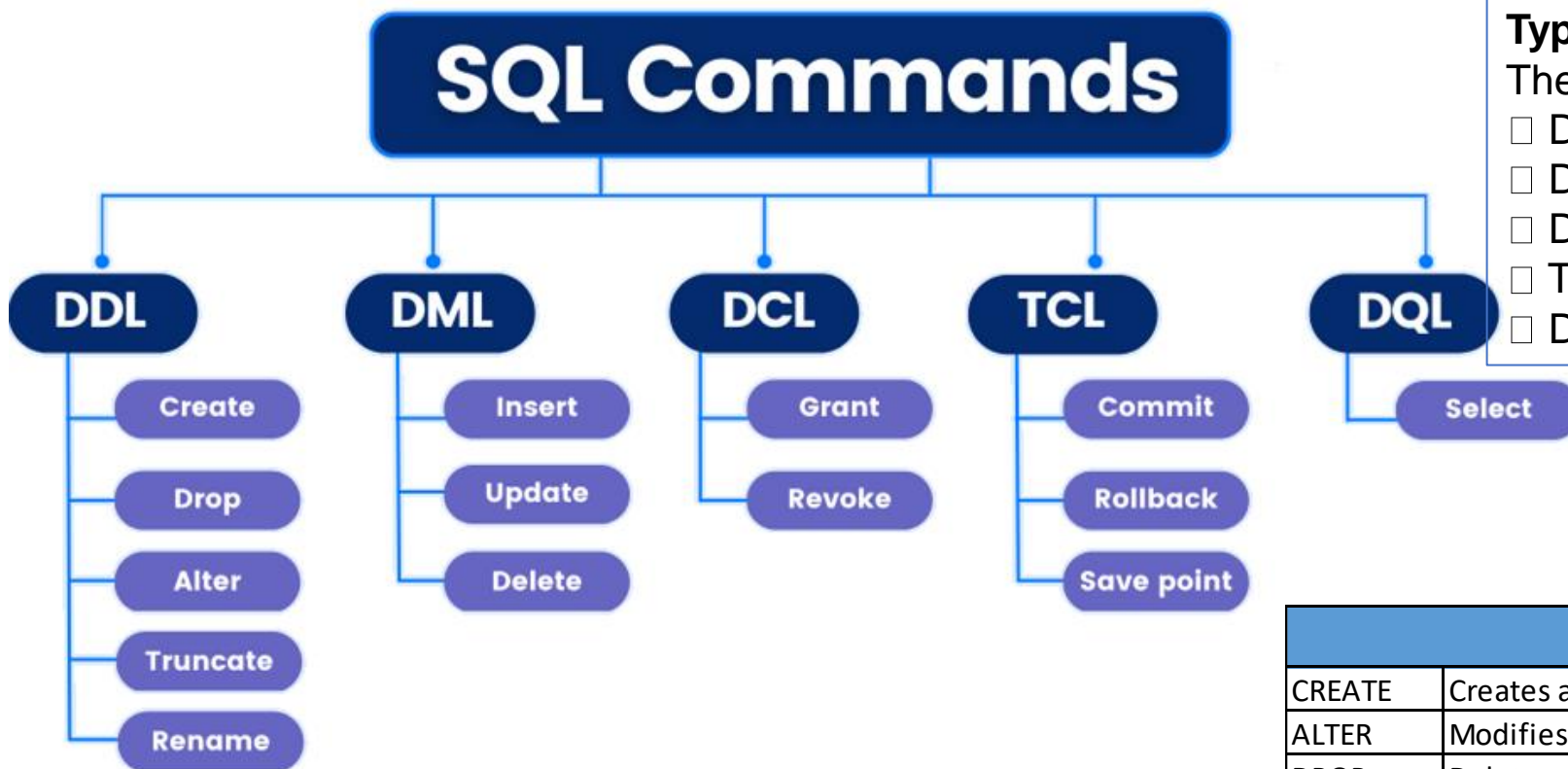


Properties	Structured data	Unstructured data
Data Type	Defined relational	Undefined non-relational
Data model	Pre-defined/not flexible	Not pre-defined/flexible
Databases	SQL Relational databases	NoSQL Non-relational databases
Data nature	Quantitative	Qualitative
	Organized information	Diverse structure for information
Flexibility	Not flexible	flexible
Storages	Data ware houses	Data lakes
Storage size	less storage	more storage
Formats	Several formats	A huge variety of formats
Ease of search	Easy to search	Difficult to search
Source location	Source from online relational and tabular forms	Source from videos, emails, documents, social media, etc

Analysis methods	<ul style="list-style-type: none"><li>• Classification</li><li>• Regression</li><li>• Data clustering</li></ul>	<ul style="list-style-type: none"><li>• Data stacking<ul style="list-style-type: none"><li>• Data mining</li></ul></li></ul>
Tools and technologies	<ul style="list-style-type: none"><li>• RDBMS</li><li>• CRM</li><li>• OLAP</li><li>• OLTP</li></ul>	<ul style="list-style-type: none"><li>• NoSQL DBMS</li><li>• AI-driven tools</li><li>• Data storage architectures</li><li>• Data visualization tools</li></ul>
Specialists to handle data	Business analysts Software engineers Marketing analysts	Data scientists, engineers, and analysts with deep expertise
Uses	Machine learning	Natural language processing and text mining
Examples	SQL	JPEG, DOC, PDF's, MOV, etc
	ID codes for databases	Videos, Images, text messages, HTML



# SQL Commands



## Types of SQL commands

There are 5 main types of commands Listed below:

- DDL (Data Definition Language) commands
- DML (Data Manipulation Language) commands
- DCL (Data Control Language) commands
- Transaction Control Language(TCL) commands
- Data Query Language(DQL) commands

## Transaction Control Language(TCL) commands

START TRANSACTION / BEGIN:	All subsequent DML (Data Manipulation Language) statements
COMMIT:	committed, the changes are irreversible and visible to other transactions.
ROLLBACK	changes made within the current transaction that have not yet been committed
SAVEPOINT	This allows for partial rollbacks, where you can undo changes only up to a specific
ROLLBACK TO SAVEPOINT	This command rolls back the transaction to the specified savepoint

## Data Definition Language (DDL)

CREATE	Creates a new table, a view of a table, or other object in the database.
ALTER	Modifies an existing database object, such as a table.
DROP	Deletes an entire table, a view of a table or other objects in the database.
TRUNCATE	Truncates the entire table in a go.

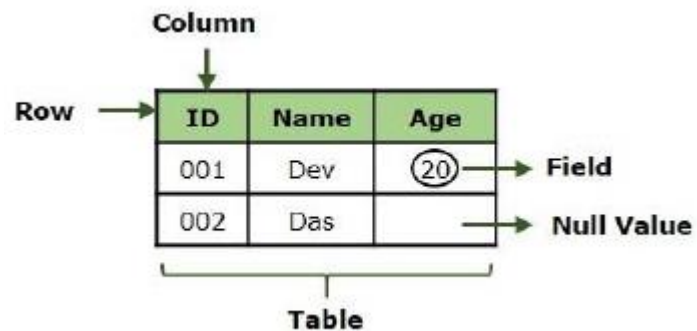
## Data Manipulation Language (DML)

SELECT	Retrieves certain records from one or more tables.
INSERT	Creates a record.
UPDATE	Modifies records.
DELETE	Deletes records.

## Data Control Language (DCL)

GRANT	Gives a privilege to user
REVOKE	Takes back privileges granted from user.

# Contd...



Command	Description
SELECT	Retrieves data from one or more tables.
INSERT	Adds new rows (records) to a table.
UPDATE	Modifies existing data in a table.
DELETE	Removes specific rows from a table.
CREATE TABLE	Creates a new table in the database.
ALTER TABLE	Modifies the structure of an existing table (e.g., add or remove columns).
DROP TABLE	Permanently deletes a table and its data.
TRUNCATE	Removes all rows from a table but keeps its structure intact.
WHERE	Filters records based on a condition.
ORDER BY	Sorts the result set in ascending or descending order.
GROUP BY	Groups rows that have the same values in specified columns.
HAVING	Filters grouped data (used with GROUP BY).
JOIN	Combines rows from two or more tables based on a related column.
DISTINCT	Removes duplicate values from the result set.
IN / BETWEEN / LIKE	Used for advanced filtering conditions.
UNION	Combines the result of two or more SELECT queries.
GRANT	Gives user privileges or permissions.
REVOKE	Removes user privileges.
COMMIT	Saves all changes made in the current transaction.
ROLLBACK	Undoes changes if something goes wrong in a transaction.
SAVEPOINT	Sets a point in a transaction to roll back to if needed.

# Data types:

CHAR(size): Stores fixed-length strings. If the string is shorter than size, it's padded with spaces.

VARCHAR(size) / NVARCHAR(size): Stores variable-length strings up to size. NVARCHAR is for Unicode characters.

TEXT / NTEXT: Stores very large strings. NTEXT is for Unicode.

CAST() or CONVERT() - You can convert numeric or date/time data to strings using functions like

CONCAT() function (widely supported): This function handles NULL values by treating them as empty strings, which can be advantageous over operators.

|| operator (ANSI SQL standard, used in PostgreSQL, Oracle, SQLite):

+ operator (SQL Server).

INTEGER (INT, SMALLINT, BIGINT, TINYINT):

Used for storing whole numbers (integers) without fractional parts. Different variations (SMALLINT, BIGINT, TINYINT) offer varying ranges and storage sizes to optimize space based on the expected magnitude of the numbers.

DECIMAL (DEC, NUMERIC):

Used for storing exact fixed-point numbers with a specified precision (total number of digits) and scale (number of digits after the decimal point). This is crucial for maintaining accuracy in calculations involving decimal values.

Approximate Numeric Data Types: These types store numbers with approximate values, often used for scientific or engineering calculations where a slight loss of precision is acceptable in exchange for a wider range or more efficient storage.

FLOAT (REAL, DOUBLE PRECISION): Used for storing floating-point numbers, which are represented in scientific notation. These types offer a wider range of values but may introduce minor inaccuracies due to the nature of floating-point representation.

REAL typically refers to single-precision floating-point numbers, while DOUBLE PRECISION (or DOUBLE) refers to double-precision.

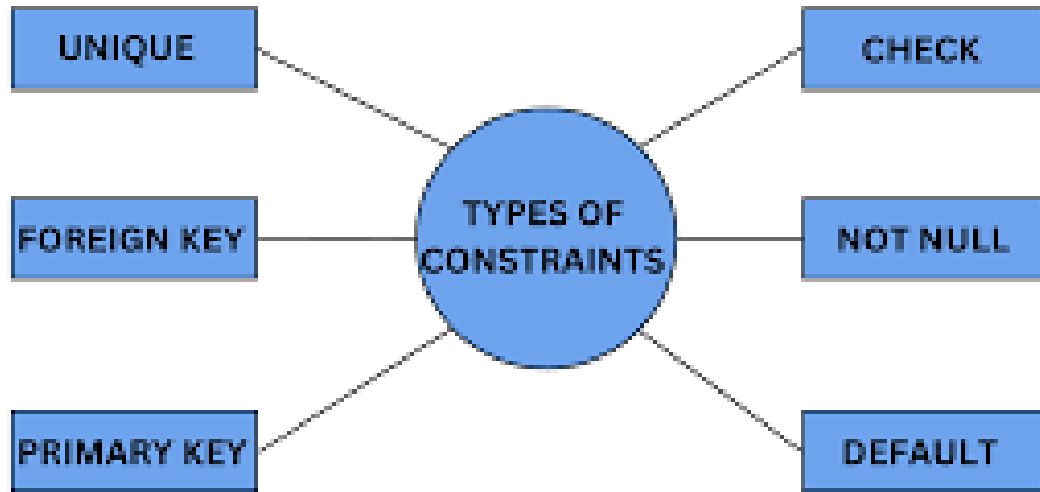
Three SQL data types.

- ☐ String Data types.
- ☐ Numeric Data types.
- ☐ Date and time Data types.

```
CREATE TABLE Products (  
    ProductName VARCHAR(255),  
    ProductCode CHAR(10),  
    Description TEXT  
);
```

# Constraints

## CONSTRAINTS IN SQL



## Constraints

<b>NOT NULL</b>	Ensures that a column cannot have a NULL value.
<b>DEFAULT</b>	Provides a default value for a column when none is specified.
<b>UNIQUE Key</b>	Ensures that all the values in a column are different.
<b>PRIMARY Key</b>	Uniquely identifies each row/record in a database table.
<b>FOREIGN Key</b>	Uniquely identifies a row/record in any other database table.
<b>CHECK</b>	Ensures that all values in a column satisfy certain conditions.
<b>INDEX</b>	Used to create and retrieve data from the database very quickly.

## Data Integrity

The following categories of data integrity exist with each RDBMS –

- ❑ Entity Integrity – This ensures that there are no duplicate rows in a table.
- ❑ Domain Integrity – Enforces valid entries for a given column by restricting the type, the format, or the range of values.
- ❑ Referential integrity – Rows cannot be deleted, which are used by other records.
- ❑ User-Defined Integrity – Enforces some specific business rules that do not fall into entity, domain or referential integrity.

## Database Normalization

Database normalization is the process of efficiently organizing data in a database. There are two reasons of this normalization process –


- ❑ Eliminating redundant data, for example, storing the same data in more than one table.
- ❑ Ensuring data dependencies make sense.

Third Normal Form is more than enough for a normal Database Application.


- ❑ First Normal Form (1NF)
- ❑ Second Normal Form (2NF)
- ❑ Third Normal Form (3NF)

# Install and Server connect


https://pypi.org/project/mysql-connector-python/




**mysql-connector-python 9.4.0**

`pip install mysql-connector-python` 

← ↻ https://pypi.org/project/mysqlclient/



**mysqlclient 2.2.7**

`pip install mysqlclient` 

## Raw Connection to Database

```
import mysql.connector

connection = mysql.connector.connect(
    user='root',
    password='college',
    host='localhost',
    database='employees',
    ssl_disabled=True
)

cursor = connection.cursor()

connection.close()
cursor.close()
```



The background is a deep blue gradient with intricate, glowing white and light blue circuit-like patterns. These patterns consist of various geometric shapes, including circles, squares, and lines, some of which are interconnected. A prominent feature is a large, glowing circular node in the upper right quadrant, with several lines radiating from it. Another smaller glowing node is visible in the lower right. The overall aesthetic is high-tech and digital.

# Thank you



Properties	Structured data	Unstructured data
Format examples	<ul style="list-style-type: none"><li>• CSV</li><li>• Excel</li></ul>	<ul style="list-style-type: none"><li>• audio files (WAV, MP3, OGG)</li><li>• PDF documents</li><li>• images (JPEG, PNG, etc.)</li></ul>
Sources examples	<ul style="list-style-type: none"><li>• online forms</li><li>• point-of-sale (POS) systems</li><li>• online transaction processing (OLTP) systems</li></ul>	<ul style="list-style-type: none"><li>• emails</li><li>• social media posts</li><li>• multimedia files</li><li>• IoT outputs</li></ul>
Nature of data	Quantitative	Qualitative
Databases	Relational (SQL)	Non-relational (NoSQL)
Storage for analytics use	Warehouses and data lakehouses	Data lakes and data lakehouses
Specialists to handle data	Business analysts, software engineers, data analysts	Data scientists, data engineers, data analysts
Main benefits	Easy to search and analyze, doesn't require much space	Easy to collect and store
Main challenges	All data must fit predefined schema	Difficult to search and analyze

