Digital Career Institute

Python Course- Databases overview

Digital Career Institute

Goal of the Module

The goal of this submodule is to help the learners to know the general concepts of databases in Python. By the end of this submodule learners will be able to understand:

- Database basic and uses
- DBMS (Database Management System)
- Relational database and model
- How to store data, problems, solutions
- SQL vs NoSQL (comparison)
- Data clustering basic

Topics

- Introduction to Databases (DB)
- Use cases of Databases
- Introduction to DBMS (Database Management System)
- What is Relational Database and its structure(RDBMS)
  - Table
  - Row
  - Column
- RDBMS examples
- Data problem and solution(storing)
- What is SQL and NoSQL(Comparison)
- What is data clustering(Basic)

Glossary

Term	Definition
DB	Database: an organized structure that store information
DBMS	Database Management System
RDBMS	Relational DBMS
SQL	Structured Query Language for manipulating data
NoSQL	Non-SQL/Non-Relational database

Introduction to Databases

Database(DB)

Definition: Database is an organized collection of information which stored electronically in a computer system/online platform. Shortly, DB

Characteristics:

- Can store Electronic data/information
  - Students information (e.g name, age, address, country, course etc.)
  - Financial data (e.g income statement, balance sheet etc.)
  - Medical record (e.g. patient info, characteristics, records, history etc.)
- Data can easily accessed, managed, modified, updated, controlled, and organized by Database
- Operate large amount of information
- Controlled by a system called DBMS(detail in slide 5)
- Database is also referred to as Database system which is a combination of data and DBMS

Use cases of Databases

DB use cases

- **Scenario 1:**
  You have a company with 100 employees. Every month you have to pay them based on their information like marriage status, children, tax class, bonuses etc. To handle this you need their detail information and also access of these datas. You can do it with spreadsheet but for 100 employees its not easy and also time consuming. So, you need dynamic way to store those data and manipulate them
- **Scenario 2:**
  Amazon.de has millions of products, orders and customers. It's almost impossible to handle all this massive information manually or technically using spreadsheets.
- **Scenario 3:**
  Giant platforms like Facebook and Youtube have large data consisting of videos, pictures, text. All this info is stored physically in their data-center. But also we need information about their location within the data-center and access of this data when needed for specific users

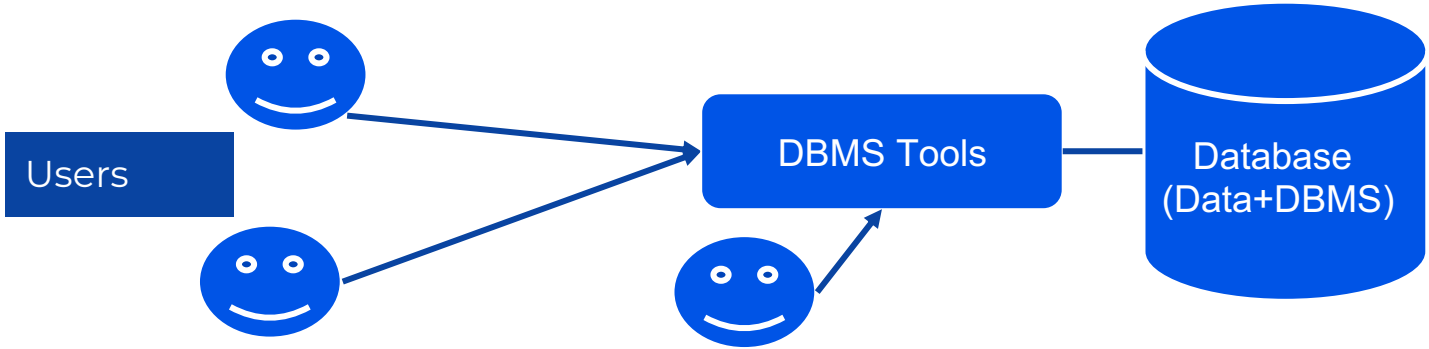
Best solution of these 3 scenarios are database system(DB)

Intro DBMS

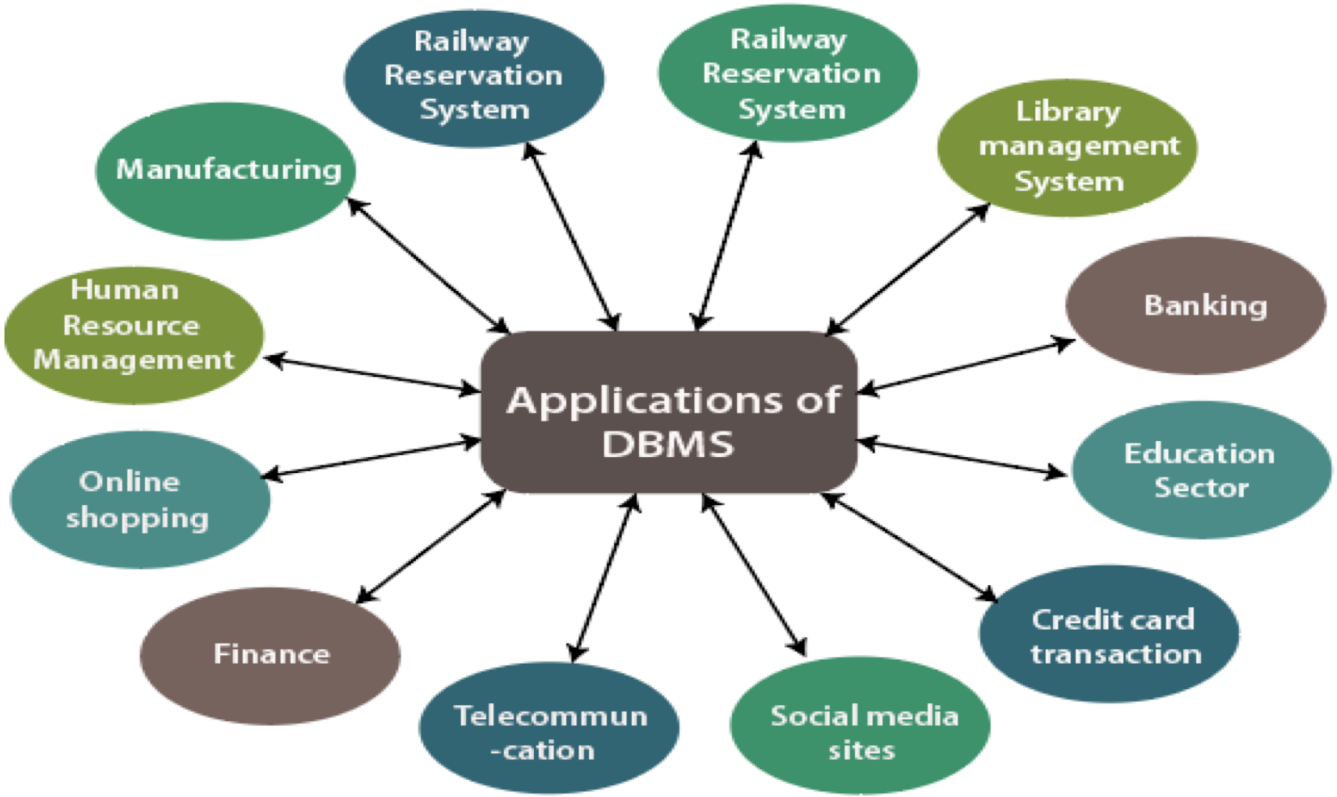
Digital Career Institute

What is DBMS

- DBMS(Database Management System) is a software used to store and retrieve database(DB)
- Provides interface for end users to perform creation, modification, deletion of data
- Contains collection of program which acts according to users instructions
- Users can have specific access control and security for data
- Can control data redundancy, so no extra data
- Using Query languages make it more faster and powerful than traditional file-processing system
- Examples: MySQL, Oracle, Microsoft SQL server etc.



DBMS applications



Relational Data

Relational Data & RDBMS

- **Relational Data** means pre-defined relationship among datas in Database(DB)
- **Relational Database Management system(RDBMS)** is the most popular database model in DBMS
- RDBMS use structured data/information
- Stores data in a table(relation) contains rows(tuples) and column(attributes)
- Uses **SQL(Structured Query languages)** for data manipulate

	Column 1	Column 2	Column	Column n
Row 1	Attribute 1	Attribute 2	…..	Attribute n
Row 2	Value 1	Value 1	…...	Value 1
...	Value 2	Value 2	…...	Value 2
..				
Row m	Value m	Value m	…...	Value m

Fig: RDBMS table structure

Table, Row, Column: (RDBMS)

Tables in RDBMS:

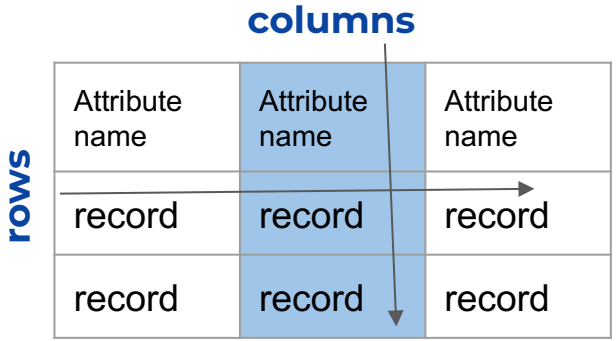
- Table is a relation of data represented by attributes and tuples
- Most simplest form of data storage

Rows in RDBMS:

- A single entry/value of table is called row
- It also called **record/tuple**
- Row could be duplicate, developers task to reduce it
- One table could have many rows

Column in RDBMS:

- Set of similar type data values or records called column
- It also called **attribute or field**
- Table rows can be broken down into smaller parts of column
- Each column holds certain type of data and have a attribute name



Table(relation) of 2x3 (rowsxcolumns)

RDBMS examples

Example 1(DCI employees)

Here we can show a RDBMS table for DCI Employees.

- Number of columns/fields 4
- Number of rows/records 5
- Attributes name: ID, Name, Designation, Campus
- 5x4 relational table

ID	Name	Designation	Campus
1	Arif	Senior Teacher	Düsseldorf
2	Mariam	Program Manager	Berlin
3	Leandro	Head of Teacher	Berlin
4	Alper	Web dev Teacher	Düsseldorf
5	Hiba	Assistant teacher	Berlin

Example 2(DCI Students)

Here we can show a RDBMS table for DCI Students.

- Number of columns/fields 6
- Number of rows/records 5
- 6X5(Columns x rows) relational table

ID	Last Name	Country	Campus	Course type	Class name
1	Adel	Syria	Düsseldorf	Web dev	FBW19
2	Jessica	Argentina	Düsseldorf	Marketing	FBW23
3	Cagri	Turkey	Berlin	Web dev	FBW44
4	Lara	Palestine	Düsseldorf	AWS	FBW66
5	Sujatha	India	Hamburg	Web dev	FBW41-9

Data Storing problem & Solution

Data Storing problem

We can save and store our data in many different ways:

- Spreadsheets like excel
- Plain text files like csv
- Using file-processing system and store in JSON
- SQL database like MySQL, PostgreSQL
- NoSQL database like MongoDB

Problems with Spreadsheets:

- It's not easy to handle a lot of data in spreadsheet
- It has access limit for users and slower than database
- Adding new data can be tedious from outside

Benefit of Database:

- Data volume: Can handle a lot of data
- Data accessibility and speed: Faster and have many options for data access and retrieve
- Editing: Data entry easier
- Data redundancy: avoid duplicate data
- Error detection and solving is easier than spreadsheet
- Security: centralized data storage provides better security

So, Database is the better solution for all these problems

At the core of the lesson

Digital Career Institute

Lessons Learned:

- We learned what is database(DB)
- We know about DBMS and how user use it
- We know RDBMS and real life examples of RDBMS
- We why we need to use Database rather than spreadsheets

Self Study

Types of Databases

- **What is SQL? Find examples of SQL**
- **What is NoSQL? When we need to use it?**
- **Differences between SQL and NoSQL**

[UNDRAW]

Relational Model

Relational Data Modelling 1/2

What is RM:

- Relational Model(RM) represent database as a collection of tables of value or relations
- RM has rows which represent real world data values followed by column name
- Table name and column name interpret each row data

Features:

- **Attribute Domain**: define relation name as column, e.g Student_name, Class_name
- **Tuple**: Single record of data value from a row, e.g Student_name = “Arif”. Arif is tuple value here
- **Table**: The RM model data saved in table format with table name

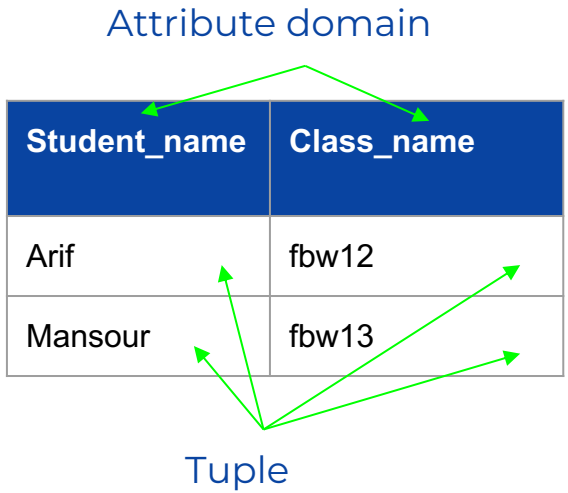


Fig: Student table

Relational Data Modelling 2/2

Features:

- **Relational Schema:** Represent the name of the relation with its attribute, e.g student and department table is a two relational table schema
- **Degree of Relation**: Total number of attributes in the relations, e.g student table has 3 degree
- **Cardinality**: Total number of rows in the relation, student table has 2 cardinality
- **Relation key**: Each row has single or many attributes which is called relation key, student table has relational key student_id. Department table has class_id and student key

Student_id	Student_name	Class_name
111	Arif	fbw12
112	Mansour	fbw13

Student Table

Class_ID	student	campus
1	111	berlin
2	112	leipzig

Department table

Relational Integrity constraints 1/3

Relational Integrity Constraints means the conditions or requirements that must be met for a valid relation.

Here is an example of integrity constraints:

- Key Constraints
- Domain Constraints
- Referential Integrity Constraints

Student_id	Student_name	Class_name
111	Arif	fbw12
112	Mansour	fbw13

Key Constraints/primary key:

- Unique Identification of a tuple which also called primary key
- Should never be null or same for different row of data
- In RDBMS, we underline the key name

For example, from this table we can use the attribute STUDENT_ID=111 to get the value of STUDENT_NAME=”Arif”. STUDENT_ID should be unique or not null.

Relational Integrity constraints 2/3

Domain Constraints:

- Refers to the rules or conditions for the row values that will be stored in certain attribute
- Domain constraints specify key constraints unique identity
- Specify data types of a value as integers, numbers, string, characters, boolean etc.

For example, We can not store student_name in Student_id or Started, Student_id must be unique and integer value

Query could be like that:

```
CREATE Domain STUDENT_ID
CHECK (Value not null)
```

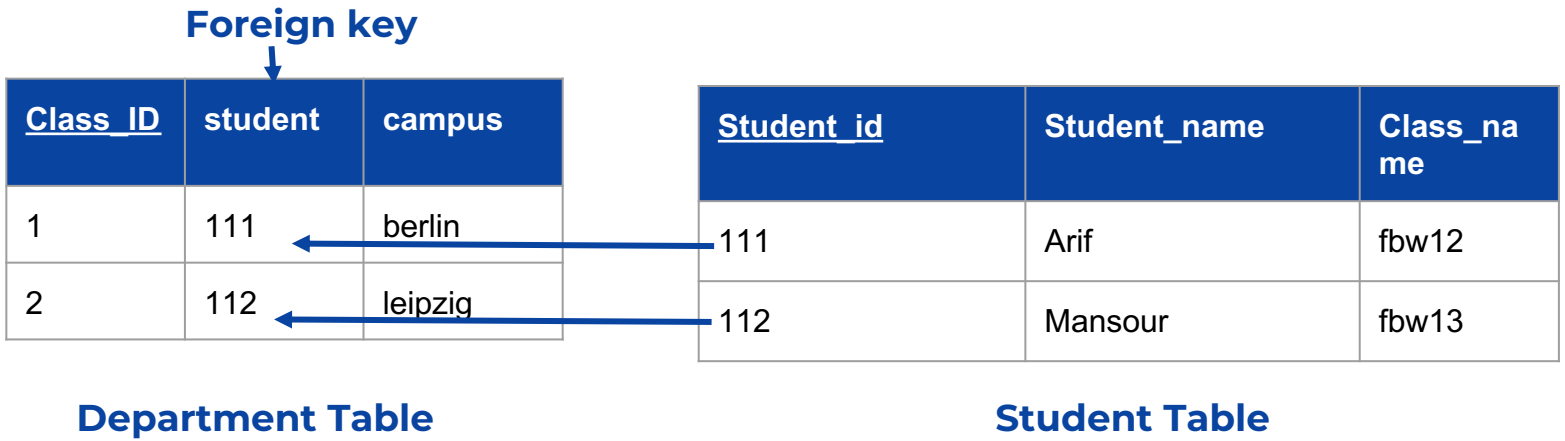
Student_id (data type: Integer)	**Student_name (data type: String)**	**Started (data type: Date)**
111	Arif	19/02/21
112	Mansour	23/04/21

Relational Integrity constraints 3/3

Referential Integrity constraints:

- Give reference of data from another table and makes relation among tables
- A foreign key is used as reference of another table key or attribute
- Most of the time the foreign key or reference key is primary key of another table

For examples, the table department has a foreign key called student and a primary key class_ID by which they can be connected and have a relationship

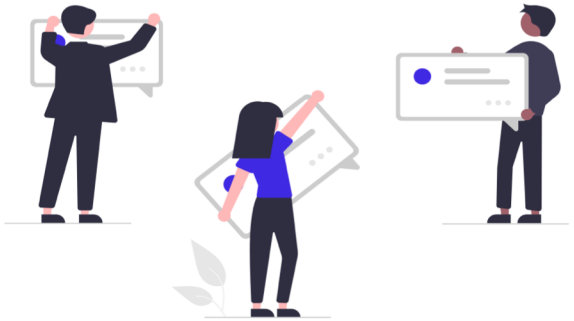


At the core of the lesson

We learned here:

- Information and key features of relational model
- Different kinds of relational integrity constraints with example cases

Digital Career Institute



Expert Round

Digital Career Institute

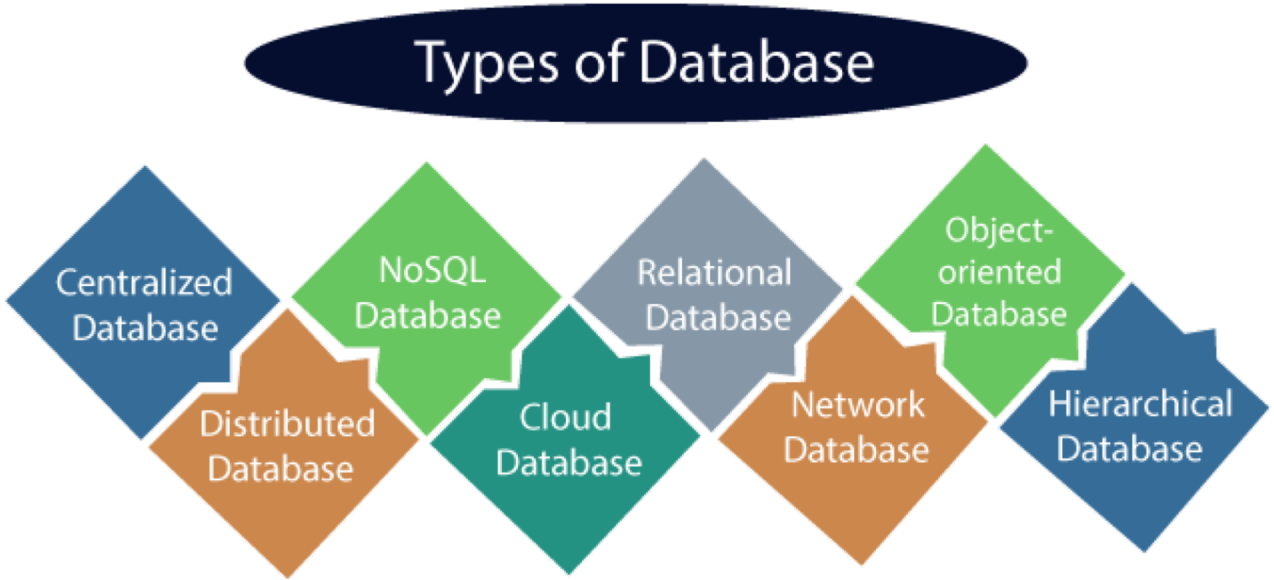
[UNDRAW]

Topics:

- **DB Overview**
- **Find and explain Similar systems like Database**

Databases types

There are many kinds of databases(DB). Among them most popular ones are Relational, cloud and NoSQL databases for web applications.

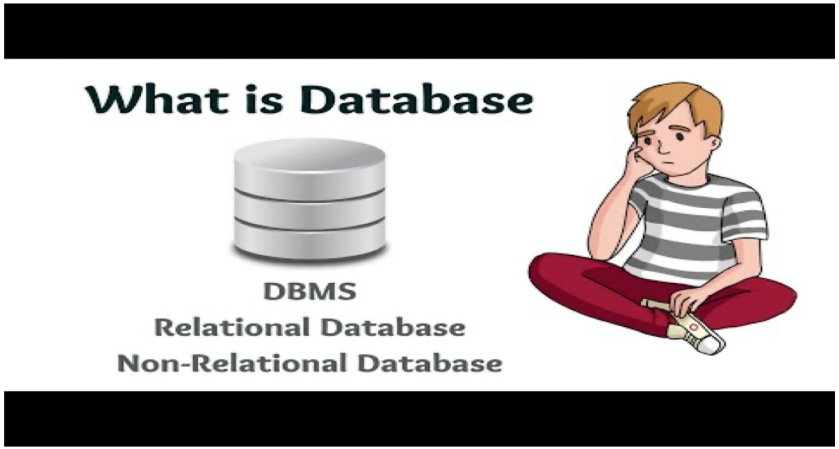


Databases types Resources

Sources for students:

- Types of Database theory:
  - Explanation_01
  - Explanation 02

DB & Types Video explanation(watch this)



Databases List

Relational (RDBMS)/SQL DB	NoSQL DB	Cloud DB	Object Oriented DB	Hierarchical DB	Centralized DB	Distributed DB	Network DB
MySQL, PostgreSQL, Amazon Aurora, Oracle DB	MongoDB, Cassandra, HBase	MongoDB Atlas, Microsoft Azure, OpenStack	Wakanda, ObjectStore	IBM IMS, Windows Registry	Office network with LAN, Central library of school/university	Apache Cassandra, Ignite	Neo4j

Similar system like Databases(DB)

- **Spreadsheets**
  - Microsoft excel
  - Google Sheets

- **Open-Source softwares**
  - LibreOffice spreadsheets
  - LibreOffice base
  - Kexi
  - Axisbase
  - Portabase

At the core of the lesson

Digital Career Institute

We learned in this expert round:

- What is database and why we need this
- Which kinds of database we already have and which one we need
- Similar kinds of application or software like databases

Documentation

Resources

Digital Career Institute

1. Python.org documentation
2. W3Schools
3. Tutorialspoint
4. Studytonight
5. JavaPoint
6. Types of Database

Contact Details
DCI Digital Career Institute gGmbH

Digital Career Institute