


Databases

Friday, May 17, 2024

12:51 PM

- 1. About Databases** 
- 2. Types of Mappings**
- 3. ORM (Object Relational Mapping)**
- 4. SQL Alchemy and SQLite Database**
- 5. Code Demo**

1. What is a Database?

- A database is a means of storing data in an organized manner.
- Helps to store, process and manipulate large volumes of data rather conveniently.
- Allows complex datasets to be stored as separate entities and establish relationships between each other for easy identification.
- Most preferred method for storing application data in the server side in the back-end.

2. Types of Databases ?

Broadly speaking, databases can be categorized as:

- *Relational* databases (ex: MySQL, PostgreSQL, Oracle)
- *Non-relational* databases (ex: MongoDB, Redis, Apache Cassandra)

3. Contents of a Relational Database:

- Tables
- Rows
- Columns
- Primary Key → uniquely identifies a particular row
- Foreign Key → Reference PK of another table

Students

ID	Name	Department	Marks
1	Bruce	CSE	85
2	Clark	ECE	77
3	Steve	IT	91
4	Peter	CSE	64
5	Steve	ECE	56
6	Tony	IT	85

referencing

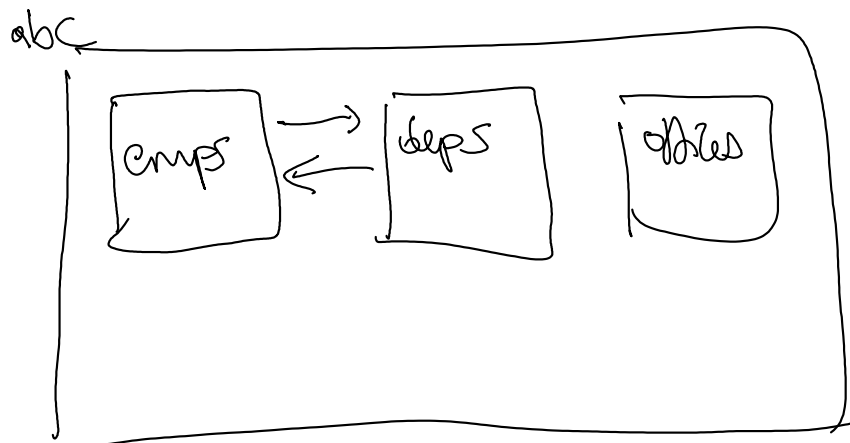
Subjects

ID	Subject	Topper ID
1	Mathematics	1
2	English	4
3	Physical Education	3
4	Physics	1
5	Journalism	2

FK is Subject
PK is Student

abc

- employees
- departments
- offices



Relational \mathcal{L}

Employees

↳ name

hagl

↳ email

↳ contact

L. Dep

id	name	age	email	contact

well-structured⁴

non-relat DB { id1: { name: —, end: — }, id2: { } }

Types of Mappings

Friday, May 17, 2024 3:19 PM

1. One to One

2. One to Many

3. Many to Many

Countries

name	continent	population

one
to
one
→

Capitals

city	country

Team

name	owner	state

one
to
many
→

Players

name	nationality	age

Customers

name	email	age

many
to
many
→

Products

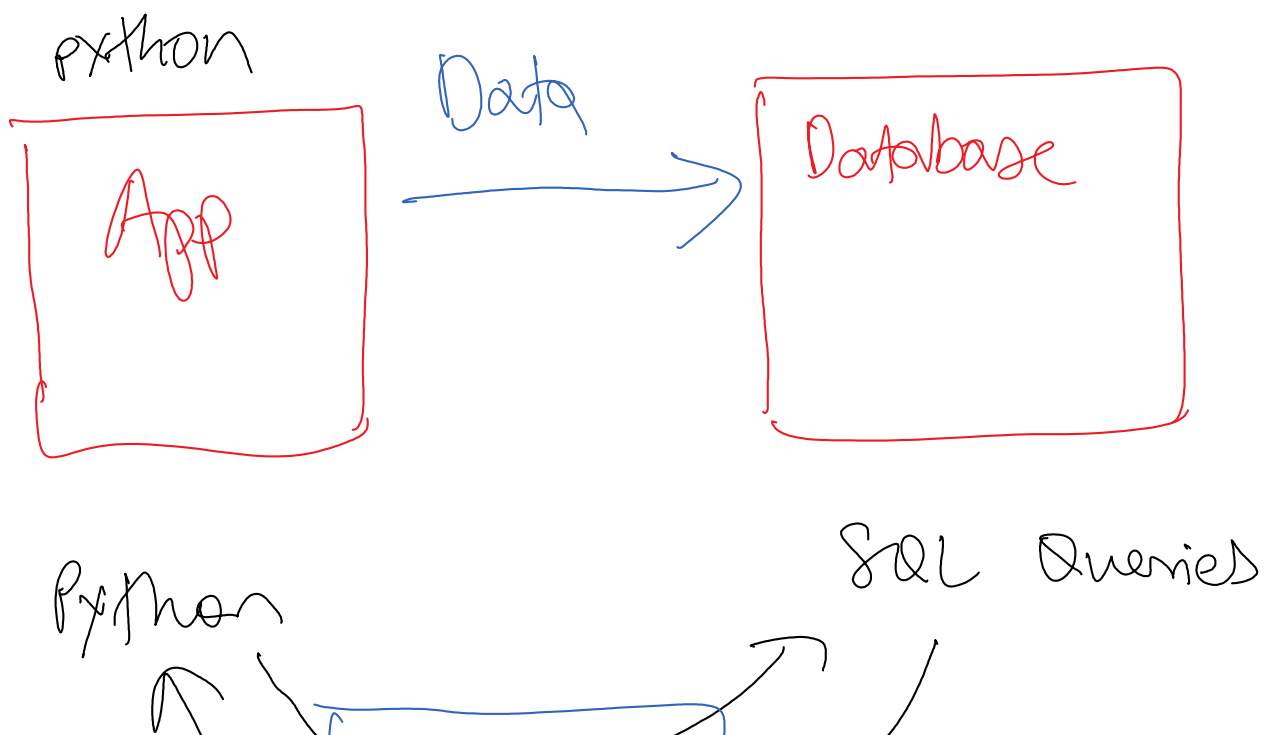
item	price

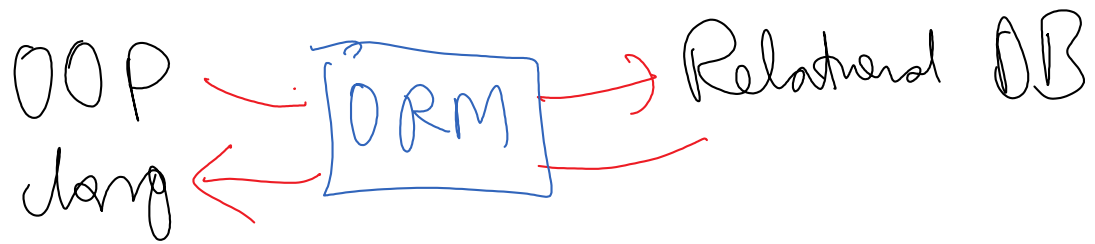
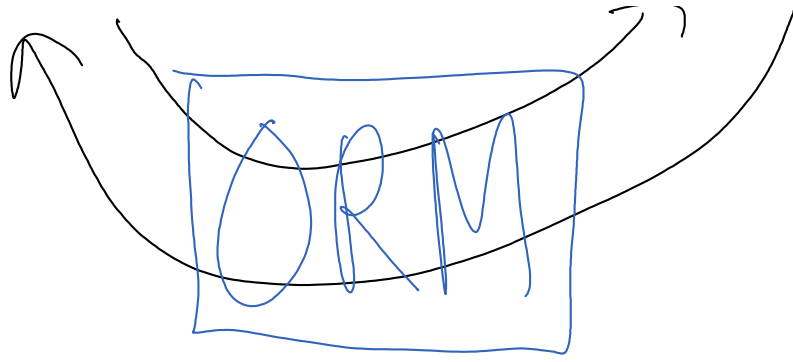
1. What is ORM ?

- It's a tool that acts as an interface between an OOP language and a Relational Database.
- Helps represent database tables in corresponding object-oriented formats in OOP language.
- Allows accessing and manipulation of database tables just as we would objects in OOP language of choice.
- Helps translate code written in OOP languages into SQL queries for data processing and vice versa.

2. Why use an ORM ?

- Provides abstraction from complex SQL queries
- Seamless mapping between OOP language and Relational Database.
- Let's switch between different Databases quickly and effortlessly
- Less code to write to perform operations
- Helps overcome SQL Injection Attack






SQLAlchemy & SQLite Database

Friday, May 17, 2024 3:39 PM

SQLAlchemy:

Flask-SQLAlchemy is an extension for [Flask](#) that adds support for [SQLAlchemy](#) to your application. It simplifies using SQLAlchemy with Flask by setting up common objects and patterns for using those objects, such as a session tied to each web request, models, and engines.

- A Python-specific ORM tool
- Can be used to connect with any database
- Helps represent Database tables as Python classes
- Helps work with rows and columns of a Database table as Class instances & attributes respectively in Python

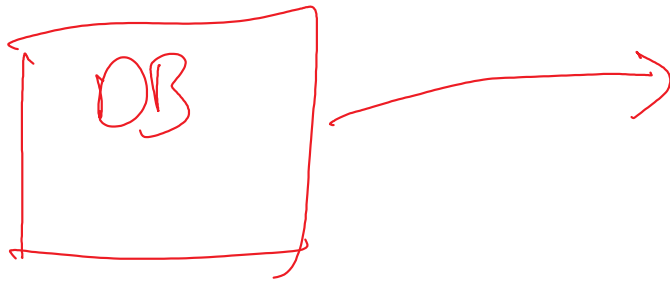


Concept	Database	Python
Data	Table	Class
Feature	Column	Attribute
Observation	Row	Class Instance
Operation	Query	Calling Methods

SQLite Database:

- Server-less, lightweight database engine
- Written in C programming language
- The database is stored as a file in the file system
- Convenient for quick use & small-scale storage purposes





c1	c2	c3	c4

cClass Table:

c1 =

c2 =

c3 =

obj1 = Table()

obj2 = Table()

1. Table

↳ CRUD

p1 = Employee (name, age)

p2

2. One to Many

3. Many to Many

2. One to Many

IPL

Teams

id	team	state

Players

id	name	nationality	teamid

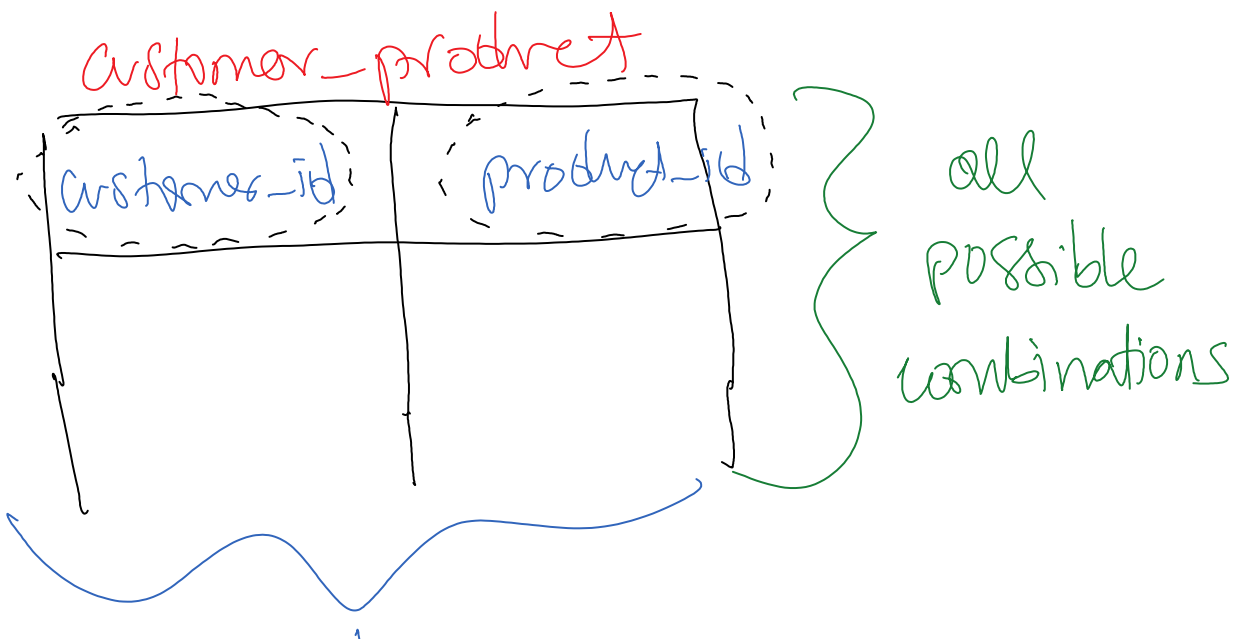
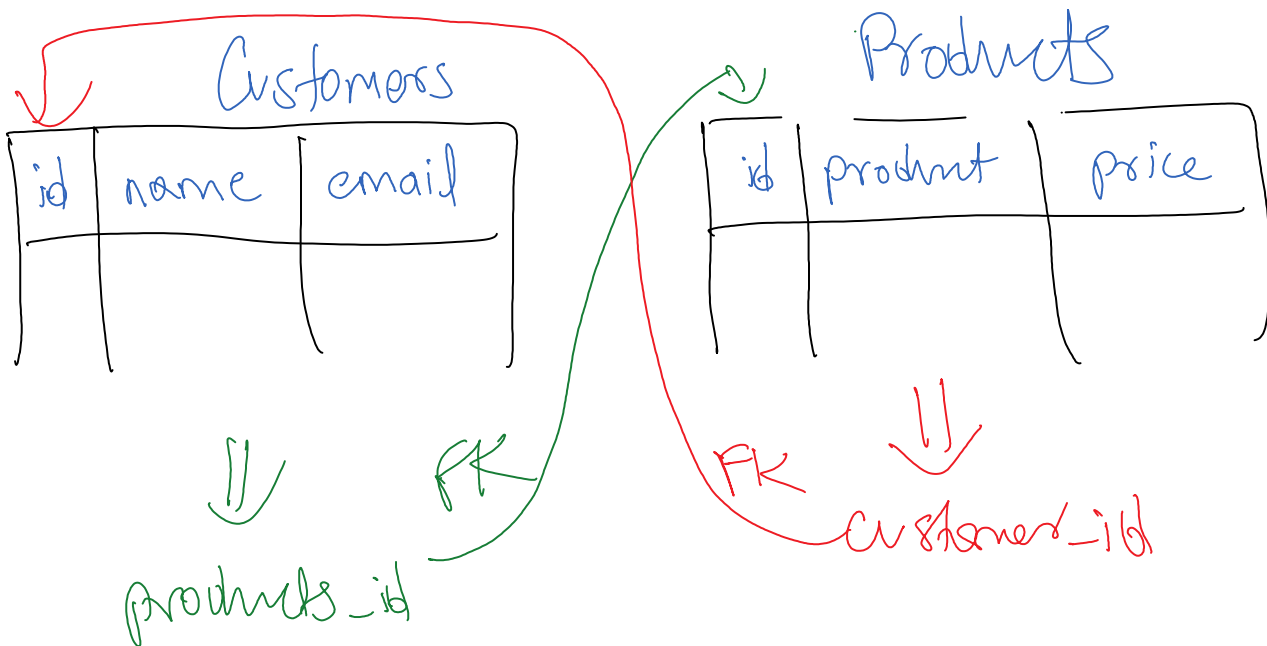
FK

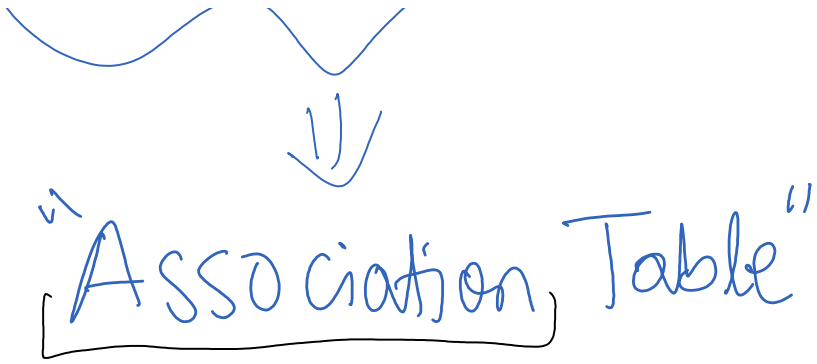
players

id	name	nationality	team_id	team

--	--	--	--	--

3. Many to Many





Association Table