

# Introduction to Databases, DBMS and Web Frameworks

Aditya Sonpal

Department of Chemical and Biological Engineering  
University at Buffalo, The State University of New York

24 July 2017

# Topics to be covered:



**University at Buffalo**

*The State University of New York*

1. Data
2. Databases & DBMS
  - What is a database & DBMS?
  - Why databases?
  - Components of a database
  - Connecting to a database
  - Conceptual design
3. Structured Query Language
  - Examples
4. Web frameworks
  - Web framework example: django
5. Summary

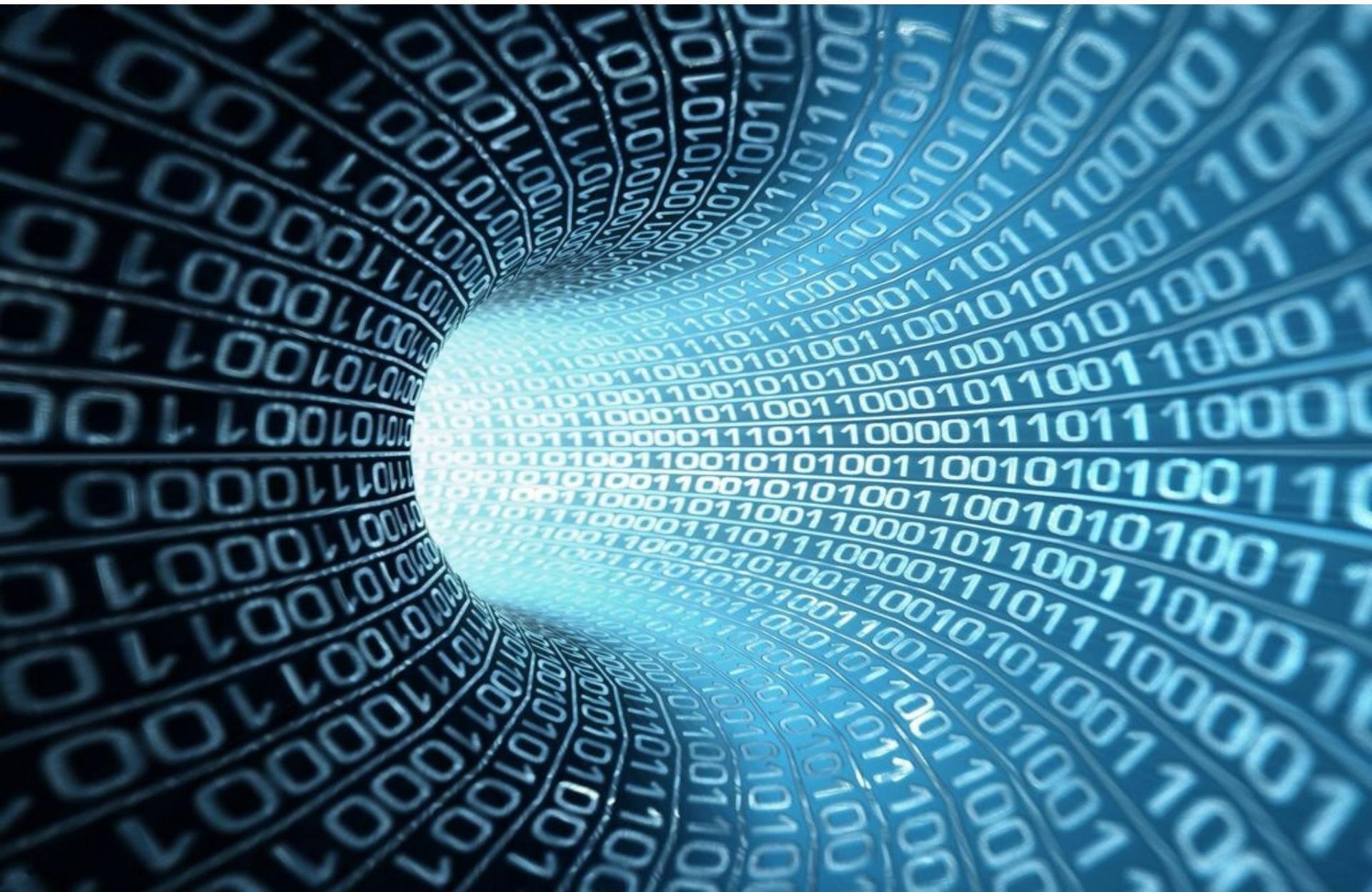


# Data:



**University at Buffalo**

*The State University of New York*



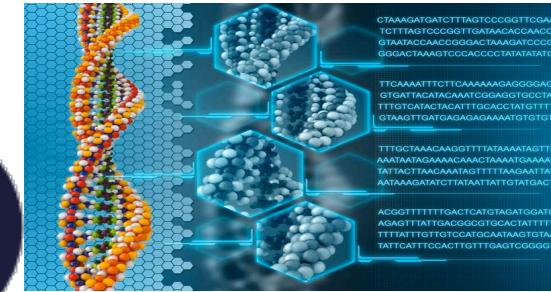
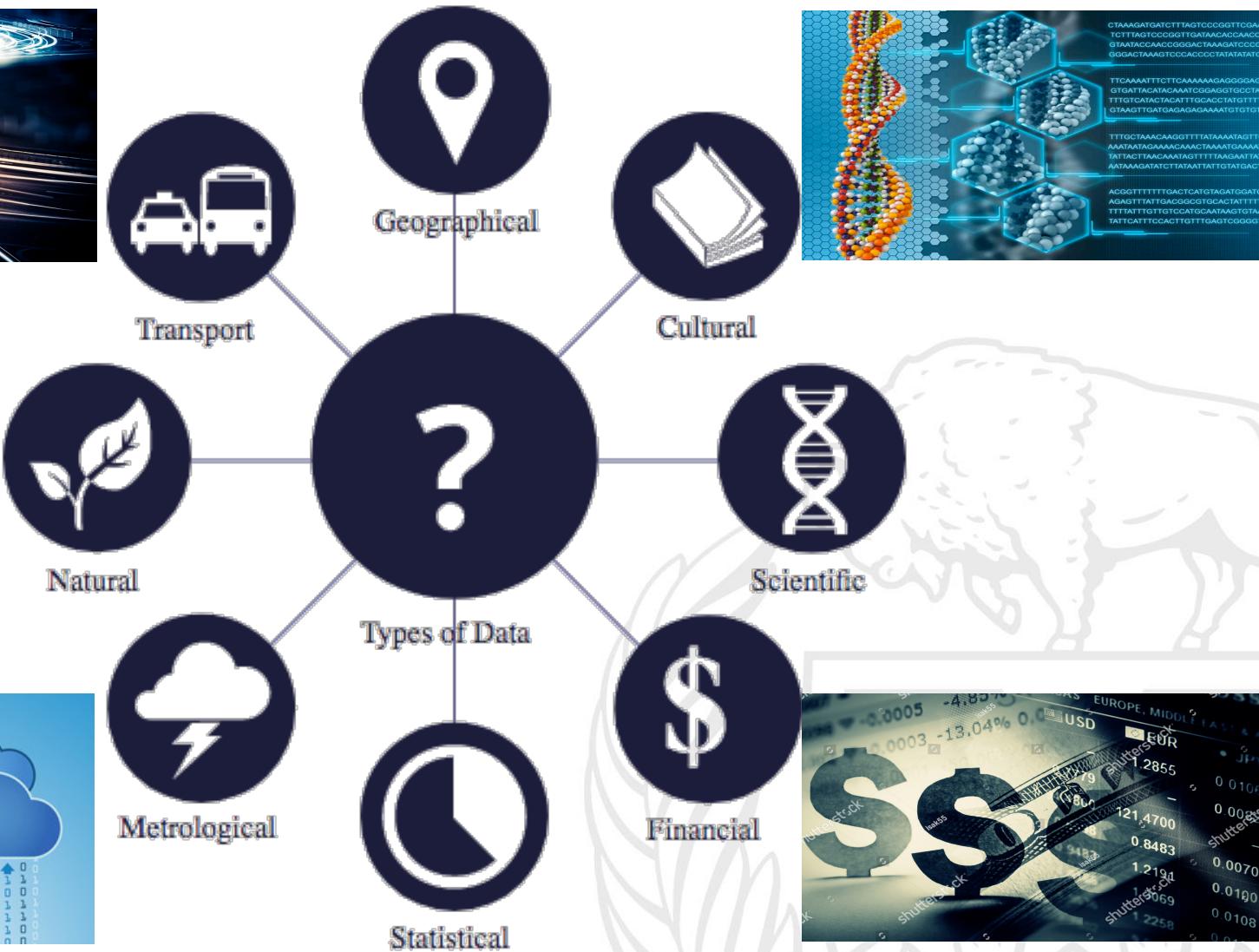
# Data:



University at Buffalo

The State University of New York

- Data is distinct pieces of information, usually **formatted** in a special way.



- **Unstructured Data:** doesn't reside in traditional rows and columns



- **Unstructured Data:** doesn't reside in traditional rows and columns



**DBMS**

- **Structured Data** : resides within fixed field, record or file.



# Databases & DBMS: What is a Database & DBMS?



University at Buffalo

*The State University of New York*

- **Database** is an **organized** collection of **related** data. It can be easily accessed, managed and optimized.
  
- **DBMS** is a system software to define, manipulate, retrieve and manage data within a database.



# Databases & DBMS: Why databases?



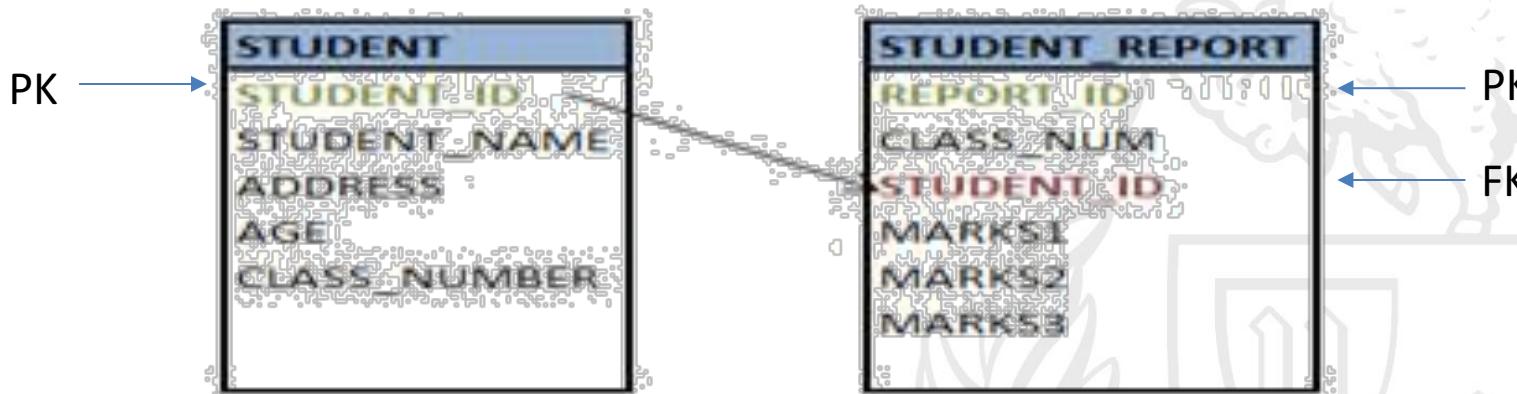
University at Buffalo

The State University of New York

## ➤ File system vs database system

- **Mapping, access and isolation**
- Redundancy
- Dependence & Inconsistency
- Security
- Atomicity
- Concurrent access

File System	DBMS
No connection between related files	Two tables can be connected through primary key-foreign key relation



# Databases & DBMS: Why databases?



University at Buffalo

The State University of New York

- File system vs database system
  - Mapping, access and isolation
  - **Redundancy**
  - Dependence & Inconsistency
  - Security
  - Atomicity
  - Concurrent access

File System	DBMS
Redundant & repetitive data is allowed.	Redundancies not allowed. ( primary key concept )



# Databases & DBMS: Why databases?



University at Buffalo

The State University of New York

- File system vs database system
  - Mapping, access and isolation
  - Redundancy
  - **Dependence & Inconsistency**
  - Security
  - Atomicity
  - Concurrent access

File System	DBMS
Changing the file format will change the type of program required.	Data is independent, and consistency is maintained.



# Databases & DBMS: Why databases?



University at Buffalo

The State University of New York

- File system vs database system
  - Mapping, access and isolation
  - Redundancy
  - Dependence & Inconsistency
  - **Security**
  - Atomicity
  - Concurrent access

File System	DBMS
Cannot secure particular parts of data while make the rest of the set visible	Offers multiple layers of security



# Databases & DBMS: Why databases?



University at Buffalo

The State University of New York

- File system vs database system
  - Mapping, access and isolation
  - Redundancy
  - Dependence & Inconsistency
  - Security
  - **Atomicity**
  - Concurrent access

File System	DBMS
No roll back mechanism	Rollback mechanism



# Databases & DBMS: Why databases?



University at Buffalo

The State University of New York

- File system vs database system
  - Mapping, access and isolation
  - Redundancy
  - Dependence & Inconsistency
  - Security
  - Atomicity
  - **Concurrent access**

File System	DBMS
Multiple users accessing the same file at the same time can lead to incorrect data	Concurrent access without incorrect data

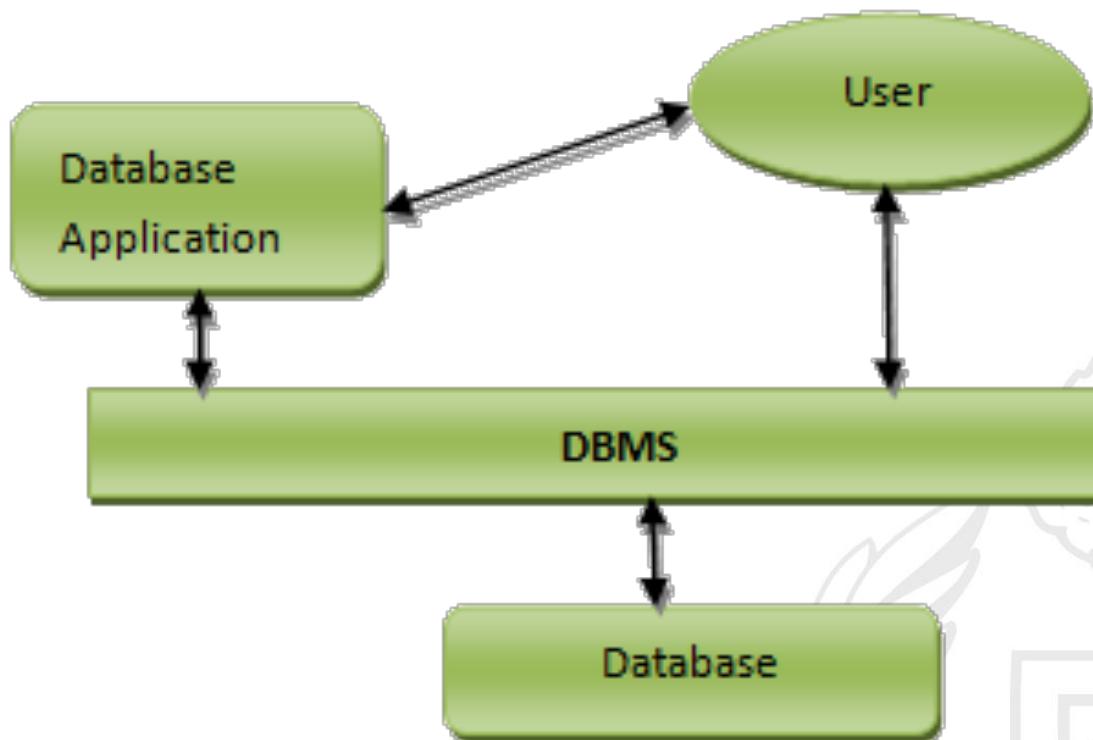


# Databases & DBMS: Components of a database



University at Buffalo

The State University of New York



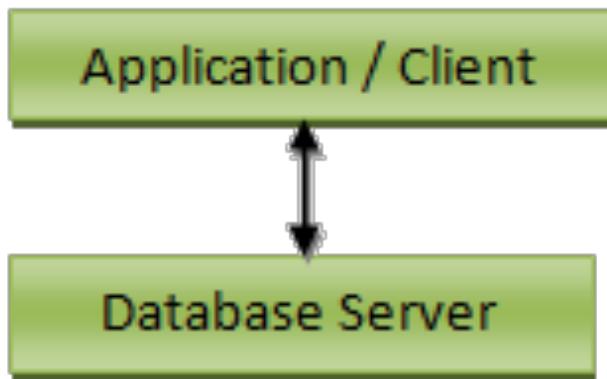
# Databases & DBMS: Connecting to a database



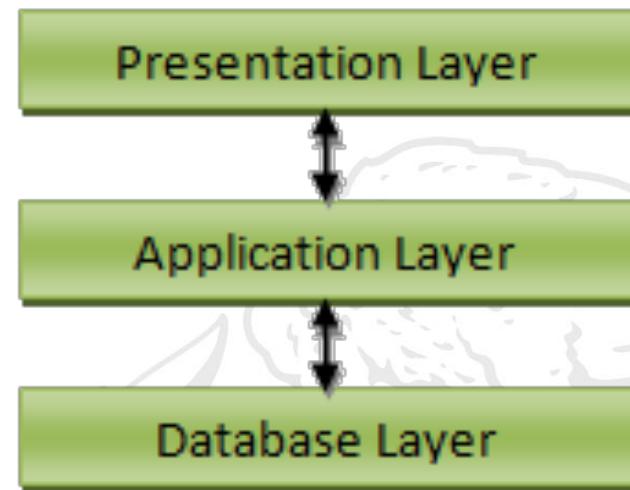
University at Buffalo

The State University of New York

2-tier architecture



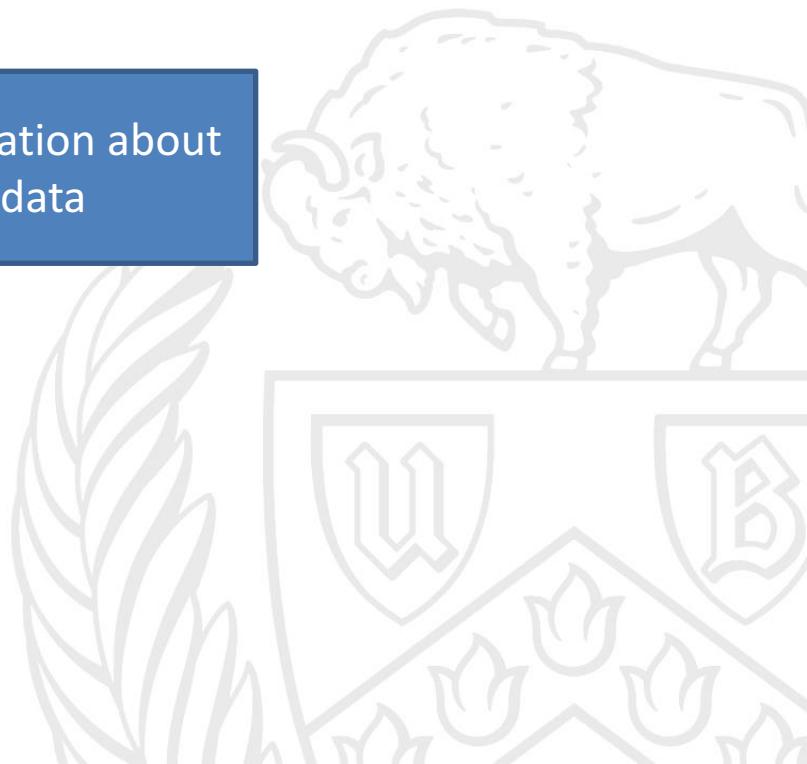
3-tier architecture



- The **logical configuration** of the database is represented by a **schema**.
- Schema may be a **visual** representation or a **set of formulae**.
- **DB Schema = data model + meta data**



Information about  
data



- The **logical configuration** of the database is represented by a **schema**.
- Schema may be a **visual** representation or a **set of formulae**.
- **DB Schema = data model + meta data**



Planning the  
structure of the DB



- The **logical** configuration of the database is represented by a **schema**.
- Schema may be a **visual** representation or a **set of formulae**.
- **DB Schema = data model + meta data**
- Helps to understand and establish relationships between objects
- Allows us to make changes before implementation.

# Databases & DBMS: Conceptual design



University at Buffalo

The State University of New York



## Person

- char\_Id INT (2)
- name CHAR (30)
- is\_Ruler INT (1)
- quote CHAR (50)
- house\_Id INT(2)

## House

- house\_Id INT (2)
- name CHAR (30)
- words CHAR (50)
- sigil CHAR (50)

## Region

- region\_Id INT (2)
- name CHAR (30)
- house\_Id INT (2)

# Structured Query Language



University at Buffalo

*The State University of New York*

- To interact with a database, 4 types of languages are required :
  - Data definition language (DDL)
  - Data modification langauge (DML)
  - Data control language (DCL)
  - Transaction control language (DCL)
- **SQL provides for all of the above.**



# Structured Query Language



University at Buffalo

The State University of New York

## ➤ CREATE :

- CREATE DATABASE GOT;
- CREATE TABLE house(house\_Id INT NOT NULL AUTO\_INCREMENT, name CHAR (30), words CHAR(100), sigil CHAR(100), PRIMARY KEY(house\_Id));

```
mysql> describe house;
+-----+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key  | Default | Extra       |
+-----+-----+-----+-----+-----+-----+
| house_Id | int(11) | NO   | PRI   | NULL    | auto_increment |
| name      | char(30) | YES  |        | NULL    |              |
| words     | char(100) | YES  |        | NULL    |              |
| sigil     | char(100) | YES  |        | NULL    |              |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

- CREATE TABLE person(char\_Id INT NOT NULL AUTO\_INCREMENT, name CHAR (30),is\_Ruler INT, quote CHAR(100), house\_Id INT, PRIMARY KEY(char\_Id),FOREIGN KEY(house\_Id) REFERENCES house(house\_Id));

```
+-----+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key  | Default | Extra       |
+-----+-----+-----+-----+-----+-----+
| char_Id | int(11) | NO   | PRI   | NULL    | auto_increment |
| name      | char(30) | YES  |        | NULL    |              |
| is_Ruler  | int(11) | YES  |        | NULL    |              |
| quote     | char(100) | YES  |        | NULL    |              |
| house_Id | int(11) | YES  | MUL   | NULL    |              |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

# Structured Query Language



University at Buffalo

The State University of New York

## ➤ Select

- `SELECT * FROM person`

## ➤ Insert

- `insert into person(name,is_Ruler,quote,house_Id) values('Ellaria Sand',1,'Weak men will never rule Dorne',8);`

## ➤ Join

- `select person.name,house.house_name from person inner join house on person.house_Id = house.house_Id;`
- `select person.name,house.house_name,region.region_name from ((person inner join house on house.house_Id = person.house_Id ) inner join region on region.house_Id = house.house_Id) where person.is_Ruler=1;`

## ➤ Update

- `update person set house_Id=10 where name='Jon Snow';`

## ➤ Delete

- `delete from house where house_Id=12;`
- `delete from person where char_Id=9;`

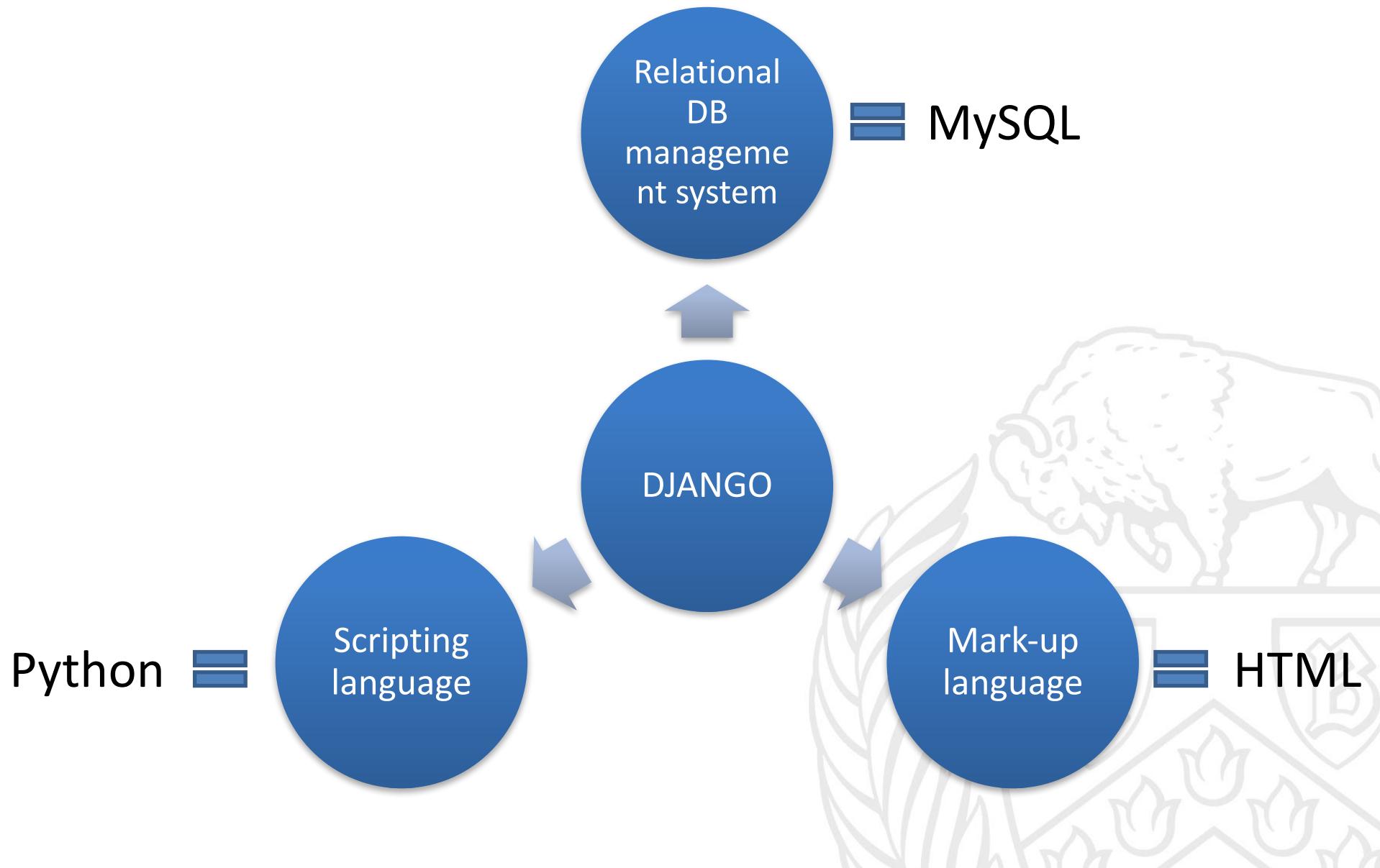
- A web framework is a software framework that is designed to support development of web applications.
- In order to make a dynamic website or a web based application two main types of coding/scripting/programming are required:
  - Client side scripting – HTML, CSS, AJAX, etc.
  - Server side scripting – PHP, ASP, Python, etc.
- Web frameworks provide a simple and effective way to integrate client side scripting and server side scripting with a database by :
  - Automating a majority of processes
  - Providing libraries for database access
  - Templating frameworks
  - Session management
  - Reusability of code

# Web Framework example : Django



University at Buffalo

The State University of New York



- Data = important
- When dealing with large amounts of data, file systems, spreadsheets are slow and inefficient.
- Therefore Databases should be used.
- Conceptual design of database :DB Schema= DB model + meta data
- SQL is the language used to communicate with the DBMS using queries for creating the db structure, inserting data, deleting, updating and modifying data.
- DB and DBMS are essential for any web based application.
- Web frameworks simplify the process of building a web app by templating and automating various processes and forming a medium of communication with the DB

