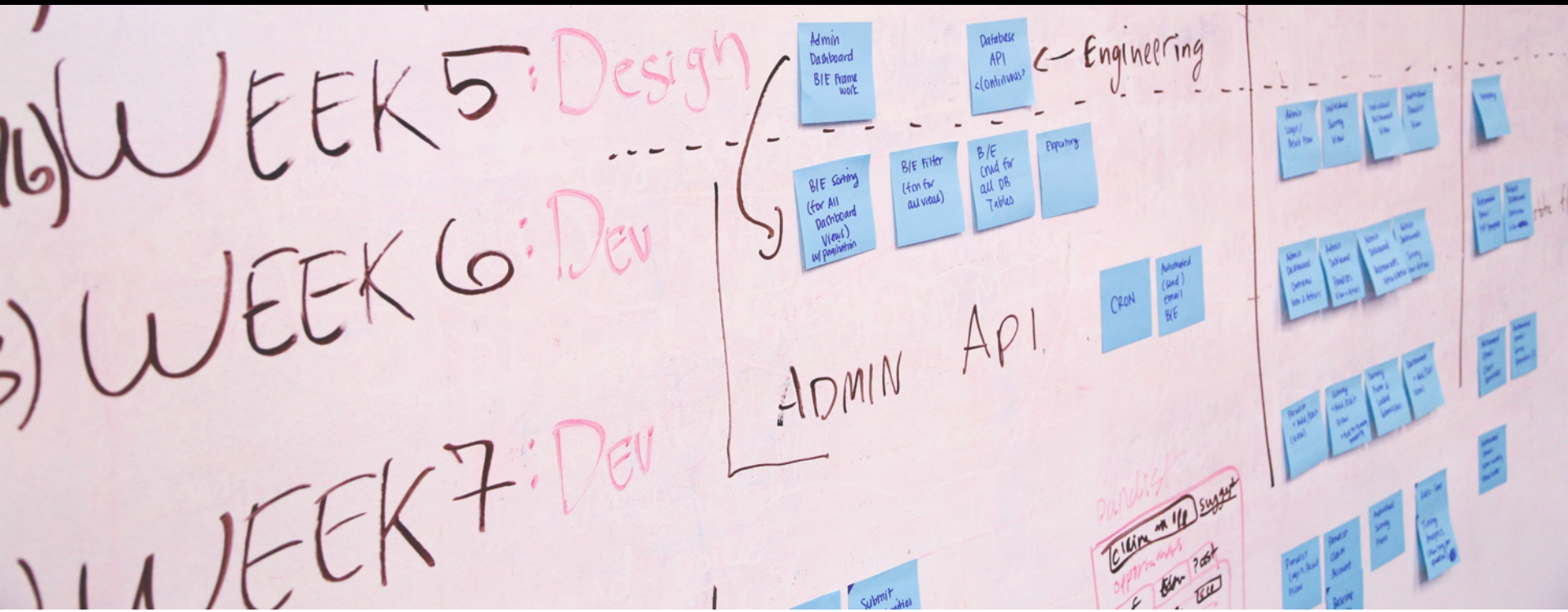


COM5961 DATA DRIVEN PRODUCTS & SERVICES DESIGN:
LESSON 3 - INTRO TO RELATIONAL DATABASE & SQL

Bernard Suen
Center for Entrepreneurship
Chinese University of Hong Kong

Today’s agenda.

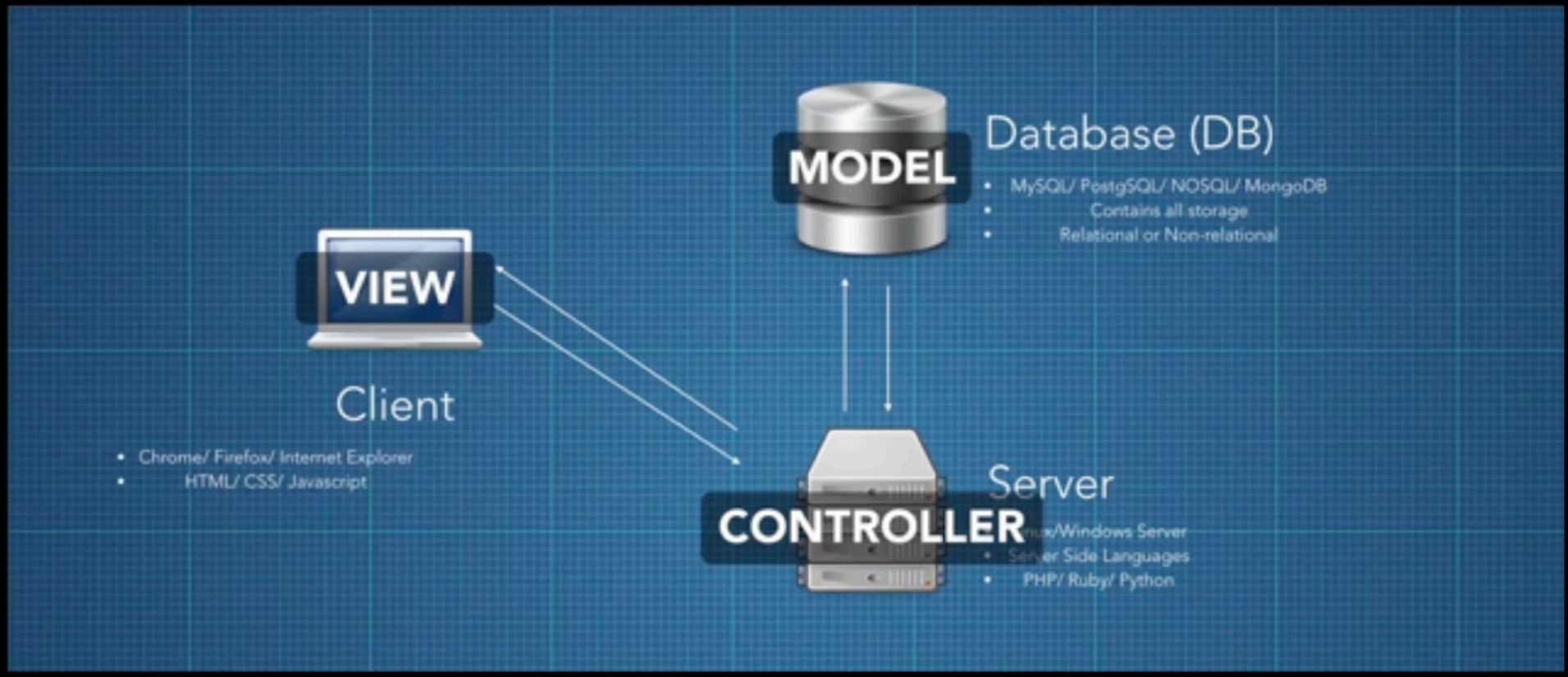
1. The Product Manager's Context for Requirement Definition and Analysis Using the MVC Framework.
2. Storing Data: CSV file vs. Relational (SQL) Database.
3. Representing relational database schema in ERD (Entity Relationship Diagram) visually.
4. What is SQL (Structured Query Language)?
5. Learning SQL database in DB Browser for SQLite3.
6. Performing SQLite3 database operations with Python Jupyter Notebook.

Assignment #1 Review

A PM's Work Context for Using the MVC Framework

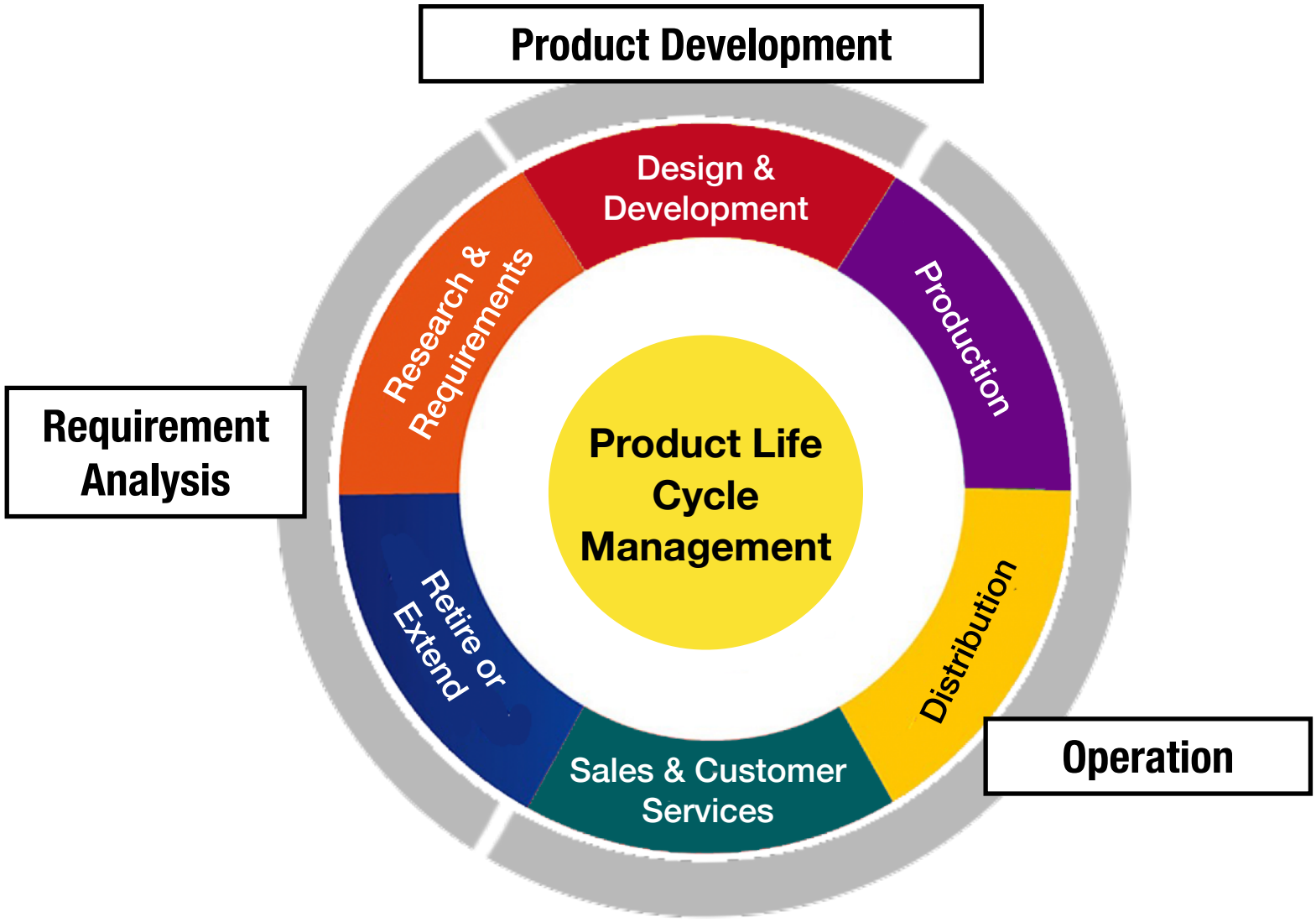
HOW DOES A WEBSITE WORK?

THE FLOW

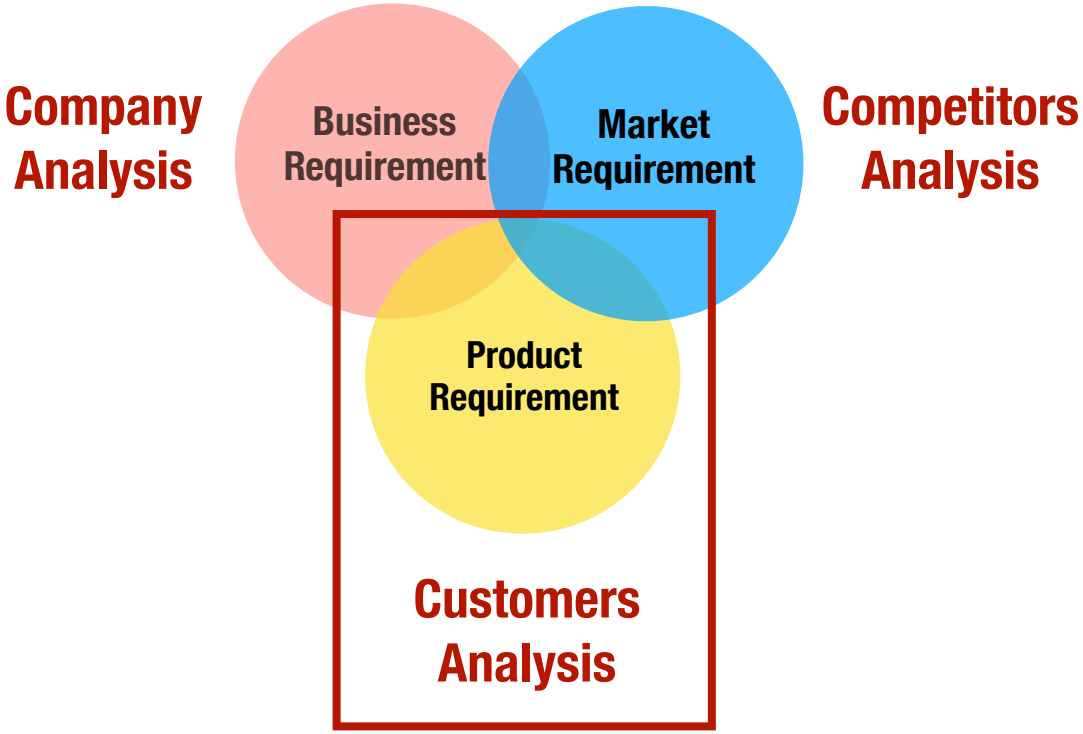


Essential Roles of a Product Manager

Product Life Cycle Management

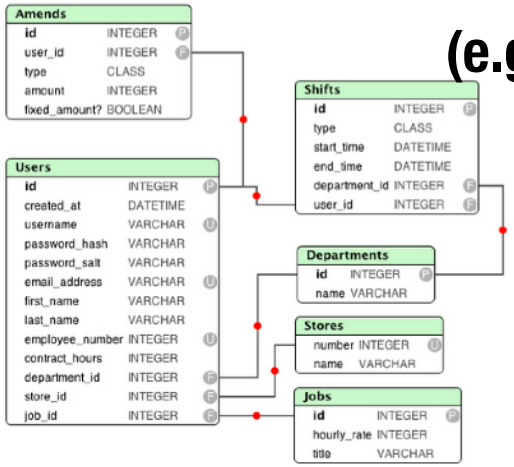

Product Development
Design & Development
Production
Distribution
Sales & Customer Services
Retire or Extend
Research & Requirements
Requirement Analysis
Product Life Cycle Management
Operation

Requirement Analysis



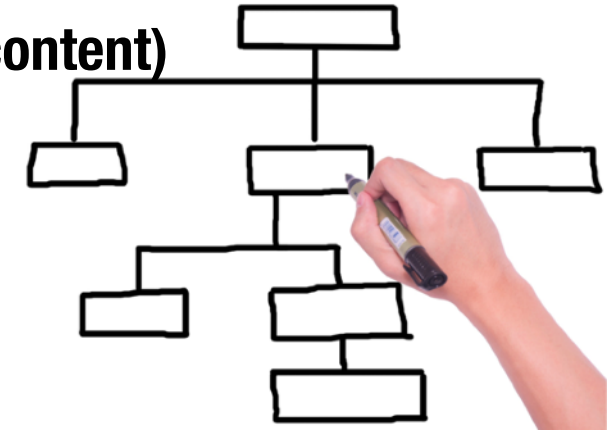
Source: commons.wikimedia.org

Source: Caius Durling (Flickr)



V(iew)
(e.g. user interface, media content)

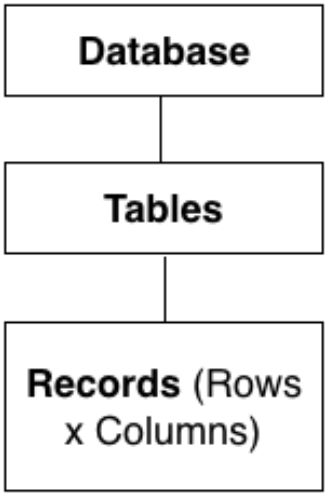
Source: pexels.com



M(odel)
(e.g. database/tables, relations, and queries)

C(ontroller)
(e.g. sitemap, navigation, routes)

What is a “relational database”?



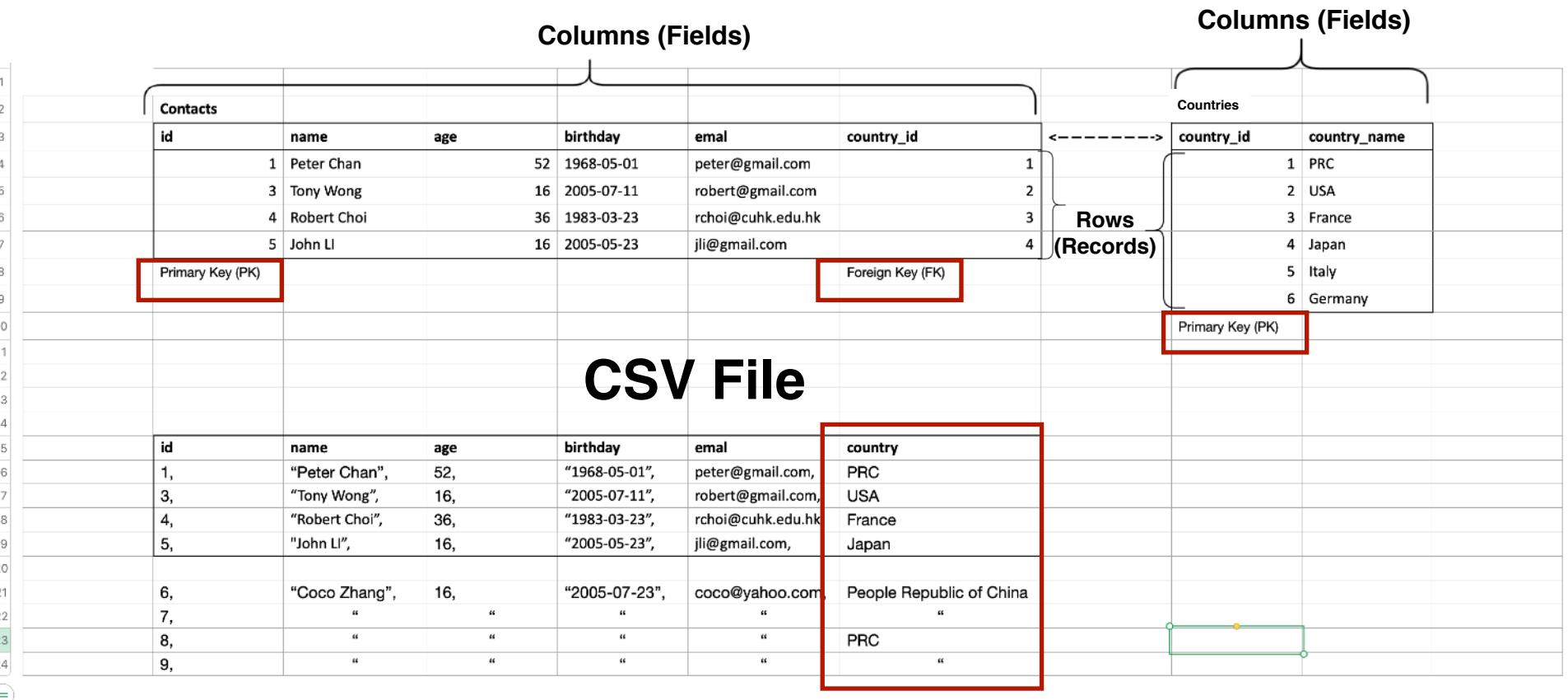
“A **relational database** is made up of a collection of **tables** which relate to each other for storing and managing data entries (**records**), organised by rows and columns.

What is SQL (Structured Query Language)?

- SQL is a set of commands for interacting with relational database to control creation of the database through DDL (Data Definition Language) and related database operations through DML (Data Manipulation Language)
- SQL can further be used to query the database for data aggregation and analysis through DQL (Data Query Language)
- SQL can also be used to grant access and rights to control user privileges over data through DCL (Data Control Language)

SQL Database vs. Delimited CSV (Comma Separated Values) File

Relational Database



id	name	age	birthday	emal	country_id
1	Peter Chan	52	1968-05-01	peter@gmail.com	1
3	Tony Wong	16	2005-07-11	robert@gmail.com	2
4	Robert Choi	36	1983-03-23	rchoi@cuhk.edu.hk	3
5	John LI	16	2005-05-23	jli@gmail.com	4

country_id	country_name
1	PRC
2	USA
3	France
4	Japan
5	Italy
6	Germany

CSV File

id	name	age	birthday	emal	country
1,	“Peter Chan”,	52,	“1968-05-01”,	peter@gmail.com,	PRC
3,	“Tony Wong”,	16,	“2005-07-11”,	robert@gmail.com,	USA
4,	“Robert Choi”,	36,	“1983-03-23”,	rchoi@cuhk.edu.hk	France
5,	"John LI”,	16,	“2005-05-23”,	jli@gmail.com,	Japan
6,	“Coco Zhang”,	16,	“2005-07-23”,	coco@yahoo.com,	People Republic of China
7,	“	“	“	“	“
8,	“	“	“	“	PRC
9,	“	“	“	“	“

Advantages and Disadvantages in Using CSV File

- Simple to use and build.
- Hard to update (when field value changes, all related entries have to be changed manually).
- Lend itself to inconsistent entries when different people enter different values representing the same attribute (e.g. country name).
- Lead to enormous number of fields (columns) per table when dealing with complex data models.
- Make data modelling overly complicated.
- CSV file is not as secured as relational database.

Structured Query Language (SQL)

"Language (commands) for instructing the database to perform data creation, manipulation, queries, and controls."

Structured Query Language (SQL) is divided into:

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

Jupyter Notebook and DB Browser for SQLite Demo

Data Definition Language (DDL)

Set Up Database (Schema)

CREATE DATABASE database name

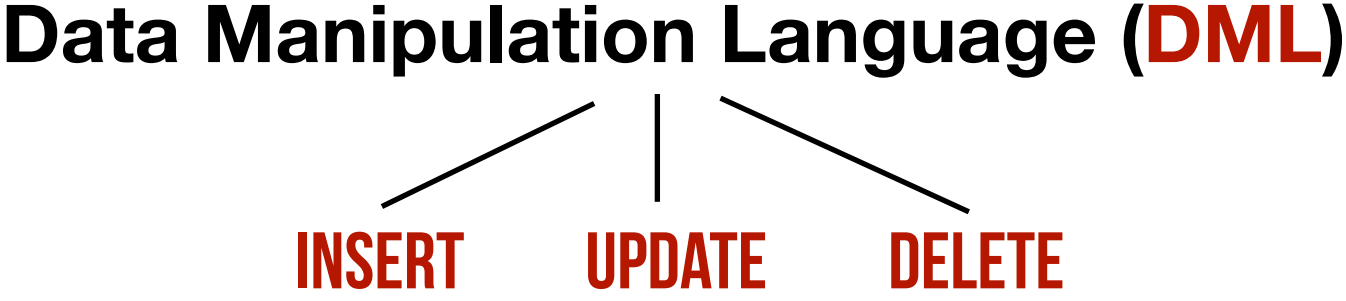
DROP DATABASE database name

SHOW DATABASES

Data Definition Language (DDL)

Set Up Table (Schema)

CREATE TABLE IF NOT EXISTS table name
SHOW TABLES


Data Manipulation Language (DML)
INSERT
UPDATE
DELETE

Insert a Record

INSERT INTO table name **VALUES** (first field value, ..., last field value)

Update a Record

UPDATE table name **SET** field name = {value|NULL|DEFAULT}, ...
WHERE condition

Delete a Record

DELETE FROM table name **WHERE** condition

Data Control Language (DCL)

Grant and Revoke Rights

GRANT privileges
ON table name
TO user

REVOKE privileges
ON table name
FROM user

privileges

SELECT	Ability to perform SELECT statements on the table.
INSERT	Ability to perform INSERT statements on the table.
UPDATE	Ability to perform UPDATE statements on the table.
DELETE	Ability to perform DELETE statements on the table.
REFERENCE	Ability to create a constraint that refers to the table.
ALTER	Ability to perform ALTER TABLE statements to change the table definition.
ALL	ALL does not revoke all permissions for the table. Rather, it revokes the ANSI-92 permissions which are SELECT, INSERT, UPDATE, DELETE, and REFERENCES.

e.g. GRANT SELECT, UPDATE, INSERT **ON** products **TO** johnchan

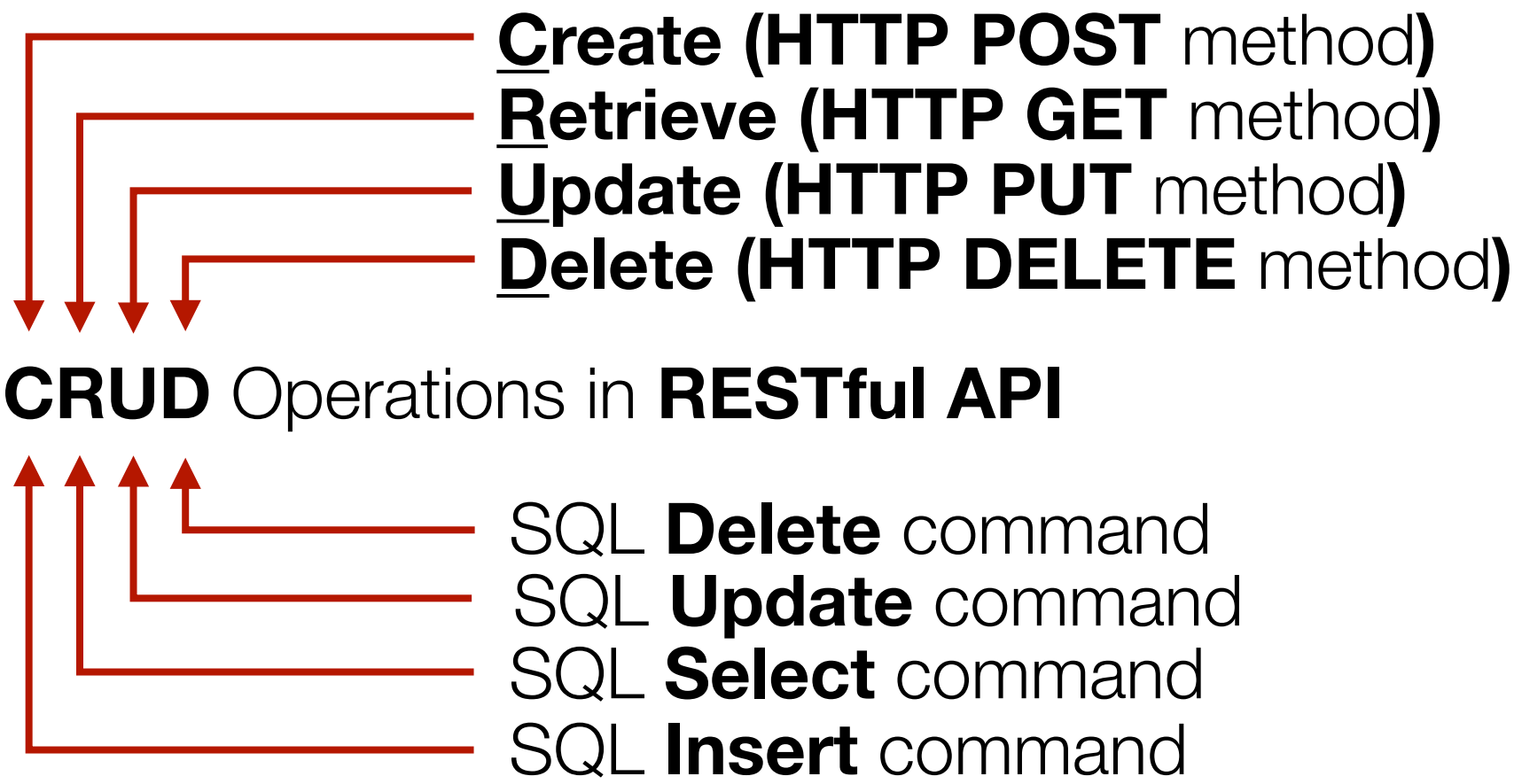
Data Query Language (DQL)***

*** DQL used most frequently in analytics.

Retrieve Record

SELECT FROM table name **WHERE** condition
GROUP BY field name
ORDER BY field name **ASC | DESC**

CRUD Operations in SQL (DML + DQL)


Create (HTTP POST method)
Retrieve (HTTP GET method)
Update (HTTP PUT method)
Delete (HTTP DELETE method)
CRUD Operations in RESTful API
SQL Delete command
SQL Update command
SQL Select command
SQL Insert command

1. **Sort** single or multiple fields as sorting criteria.
2. **Group** entries by a single field or multiple fields to create data segments.
3. **Filter** entries by single or multiple fields as searching conditions.

Use sort, group, and filter to search for information in a table.

Retrieve Records by Aggregate/Statistic Functions

SELECT COUNT (field name**)**
FROM table name **GROUP BY** field name **WHERE** condition

SELECT SUM (field name**)**
FROM table name **GROUP BY** field name **WHERE** condition

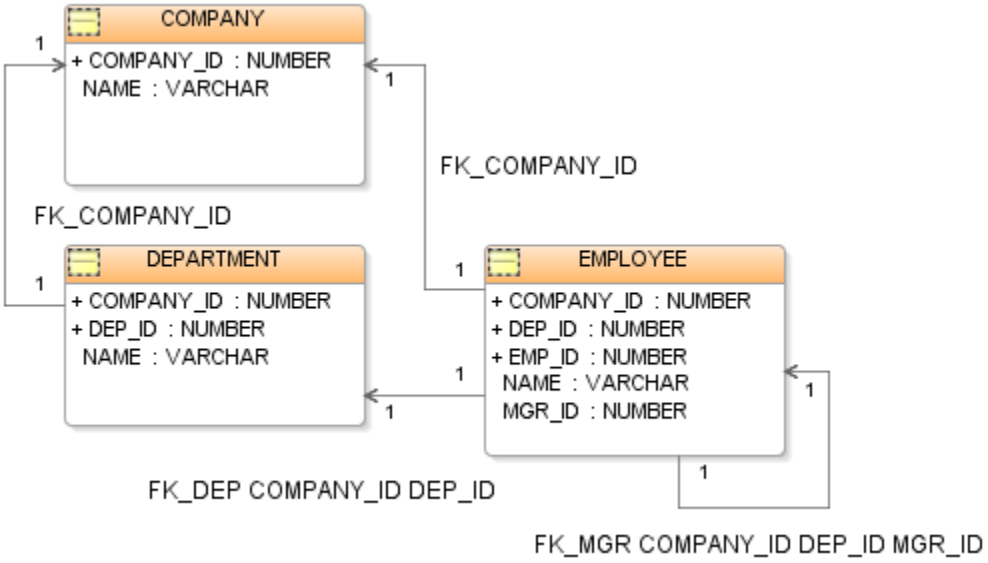
SELECT AVG (field name**)**
FROM table name **GROUP BY** field name **WHERE** condition

SELECT MAX I MIN (field name**)**
FROM table name **GROUP BY** field name **WHERE** condition

Primary Key and Foreign Key

Building relationships between tables by using foreign keys.

- Every entry in a table should be uniquely defined by a **primary key**.
- Build relationships between tables by making one of the fields in the table as a **foreign key** linked to the **primary key** of another table.



Source: commons.wikimedia.org

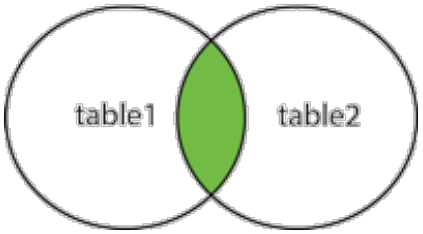
Retrieve Records Through SQL Join (e.g. inner, outer, left, right)

SELECT tableName1.fieldName1,tableName2.fieldName2, tableName1.fieldName3, tableName2,fieldName4
FROM tableName1
INNER JOIN tableName2 ON tableName1.fieldName = tableName2.fieldName;

Example:

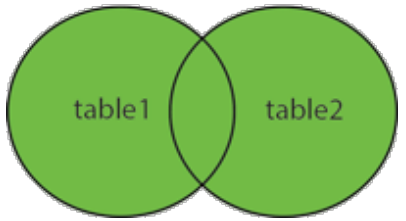
```
SELECT Orders.OrderID, Customers.CustomerName, Orders.OrderDate
FROM Orders
INNER JOIN Customers ON Orders.CustomerID=Customers.CustomerID;
```

INNER JOIN



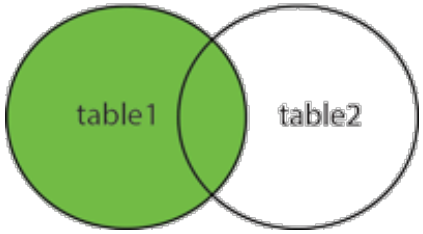

```
SELECT column_name(s)
FROM table1
INNER JOIN table2
ON table1.column_name = table2.column_name;
```

FULL OUTER JOIN



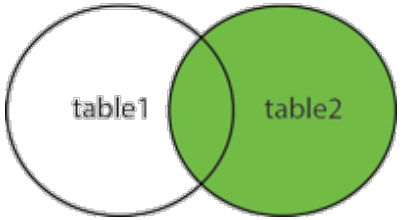

```
SELECT column_name(s)
FROM table1
FULL OUTER JOIN table2
ON table1.column_name = table2.column_name
WHERE condition;
```

LEFT JOIN




```
SELECT column_name(s)
FROM table1
LEFT JOIN table2
ON table1.column_name = table2.column_name;
```

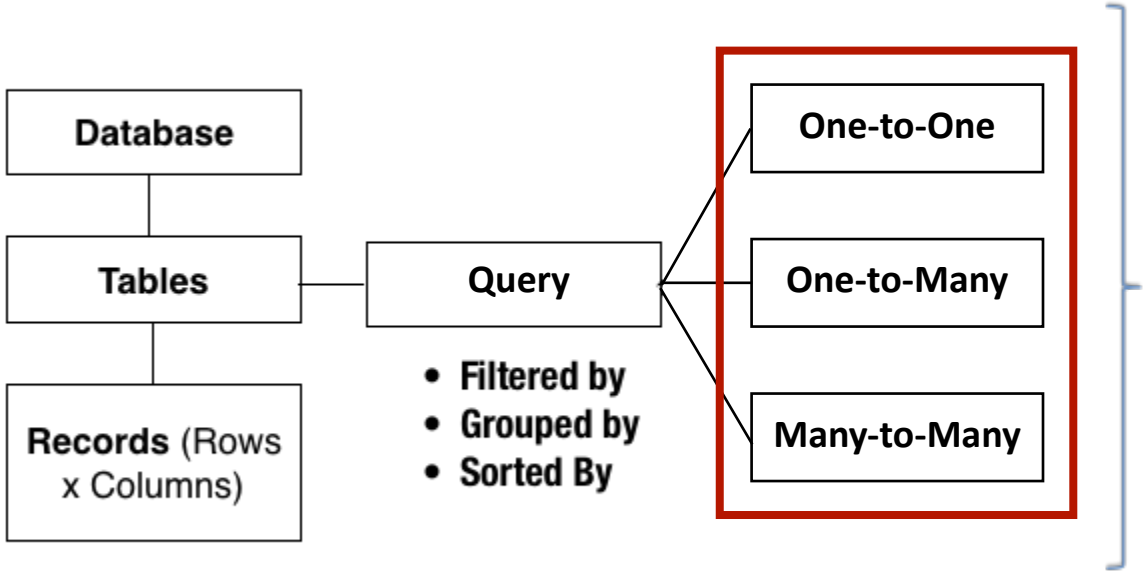
RIGHT JOIN



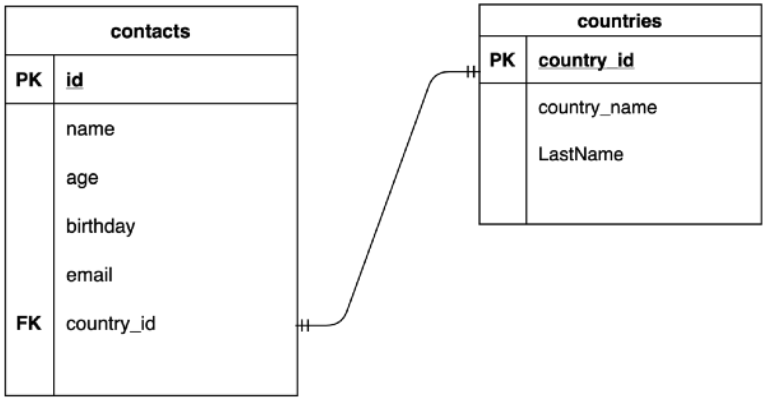

```
SELECT column_name(s)
FROM table1
RIGHT JOIN table2
ON table1.column_name = table2.column_name;
```

Source: https://www.w3schools.com/sql/

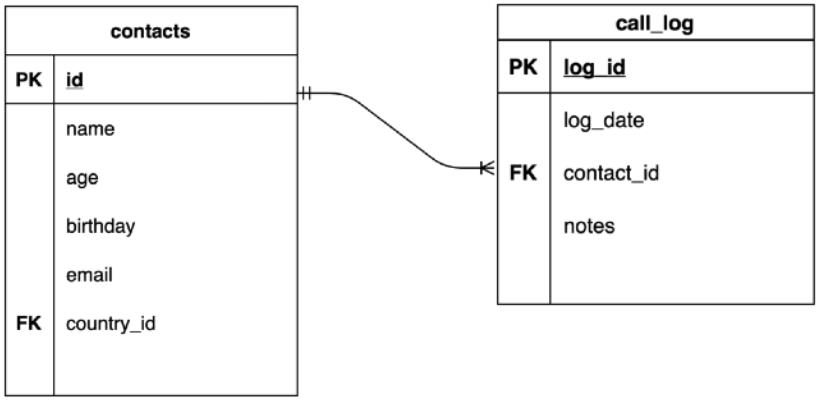
The Three Most Common Types of Relationship in Relational Database


Database
Tables
Records (Rows x Columns)
Query
Filtered by
Grouped by
Sorted By
One-to-One
One-to-Many
Many-to-Many

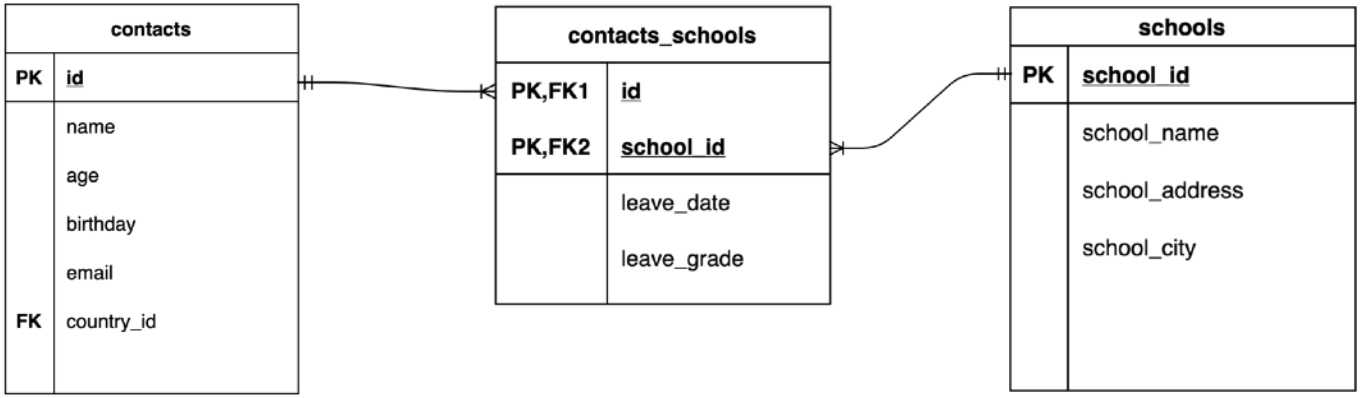
- One-to-One



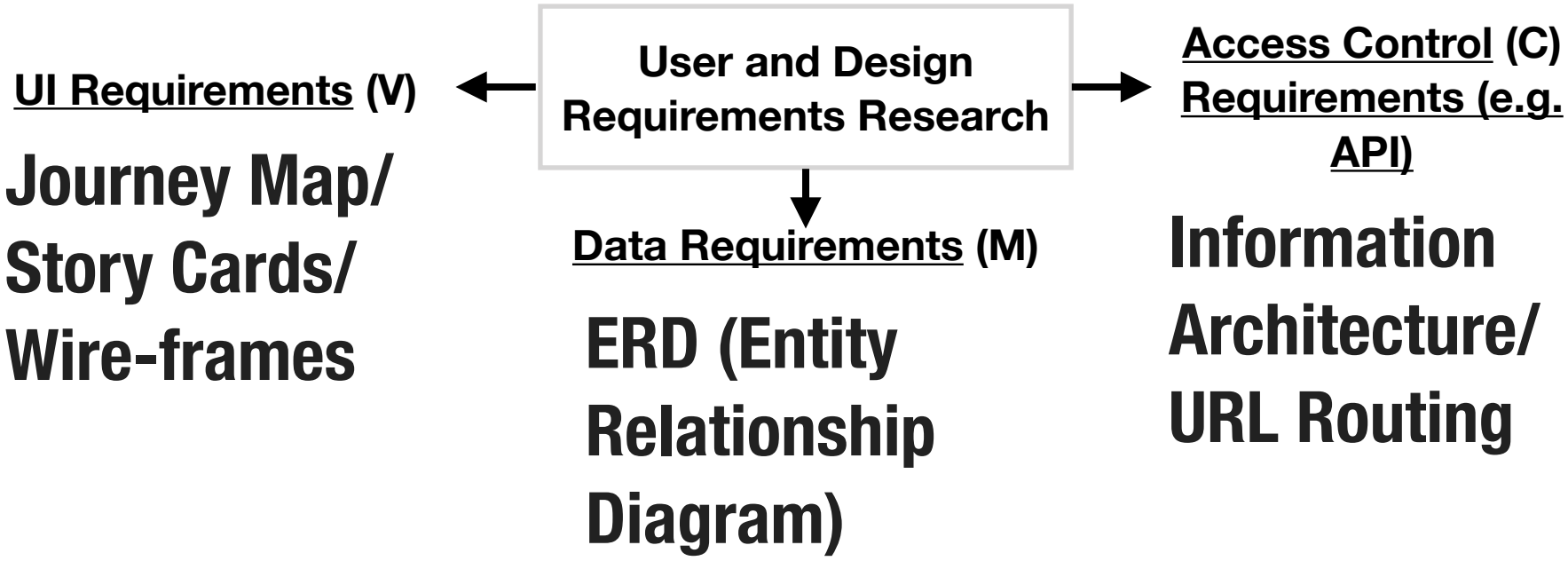
- One-to-Many



- Many-to-Many



Design Research and Requirement Specification


User and Design Requirements Research
UI Requirements (V)
Journey Map/ Story Cards/ Wire-frames
Data Requirements (M)
ERD (Entity Relationship Diagram)
Access Control (C) Requirements (e.g. API)
Information Architecture/ URL Routing

Thank you for your time!