Data Models :

* relational : related tables } structured
* XML : hierarchical } semi-structured data
* NoSQL : not only SQL
* text documents } unstructured

schema on read vs. schema on write
↓
database design

steps for building and using a relational database

1. Design the schema
2. Create → with DDL (Data Definition Language)

```
CREATE DATABASE university

CREATE TABLE Student (
    Name    Char(30)
    SID     INT NOT NULL
    GPA     float
    Class   INT
)
```

3. Load the database with initial values

```
INSERT INTO Student VALUES
('Smith', 17, 3.8, 1)
('Brown', 100, 3.2, 2),
   :
   :
```

④ * query & updates

DML Data Manipulation Language

DDL } SQL declarative (not procedural)
DML }

```
SELECT name, GPA
FROM   Student
WHERE  GPA > 3.4
```

Name	GPA
Mary	3.5
Sam	3.8
Coly	3.6

* Result of a query is a table - closed

SQL is based on

- set theory
- relational algebra
- relational calculus

Database Example

- The University Database
  - Common example used in many texts and tutorials
- Purpose of the DB:
  - Maintain information regarding students, courses and grades
- Database Structure:
  - Organized as 5 files (or collections) of data records
    - STUDENT — Student data
    - COURSE — Course data
    - SECTION — Data about each section of a course
    - GRADE_REPORT — Grades that students receive in a course
    - PREREQUISITE — Prerequisites for courses

Tables

University Database

STUDENT

STUDENT — Table Name

Name	Student_number	Class	Major
Smith	17	1	CS
Brown	8	2	CS

- Each student record is assigned:
  - Name (string – alphabetical characters)
  - Student_number (integer)
  - Class (integer) – Freshman (1), Sophmore (2)
  - Major (string)

} columns

University Database

- COURSE

COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

- SECTION

Who teaches data structures ?

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	07	King
92	CS1310	Fall	07	Anderson
102	CS3320	Spring	08	Knuth
112	MATH2410	Fall	08	Chang
119	CS1310	Fall	08	Anderson
135	CS3380	Fall	08	Stone

University Database

- GRADE_REPORT

GRADE_REPORT

Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

- PREREQUISITE

PREREQUISITE

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

University Database

Describing the DB:

- DBMS Catalog contains a complete definition of the DB structure and constraints (**meta-data**)

RELATIONS

Relation_name	No_of_columns
STUDENT	4
COURSE	4
SECTION	5
GRADE_REPORT	3
PREREQUISITE	2

COLUMNS

Column_name	Data_type	Belongs_to_relation
Name	Character (30)	STUDENT
Student_number	Character (4)	STUDENT
Class	Integer (1)	STUDENT
Major	Major_type	STUDENT
Course_name	Character (10)	COURSE
Course_number	XXXXNNNN	COURSE
….	….	…..
….	….	…..
….	….	…..
Prerequisite_number	XXXXNNNN	PREREQUISITE

Note: Major_type is defined as an enumerated type with all known majors.
XXXXNNNN is used to define a type with four alpha characters followed by four digits.

University Database

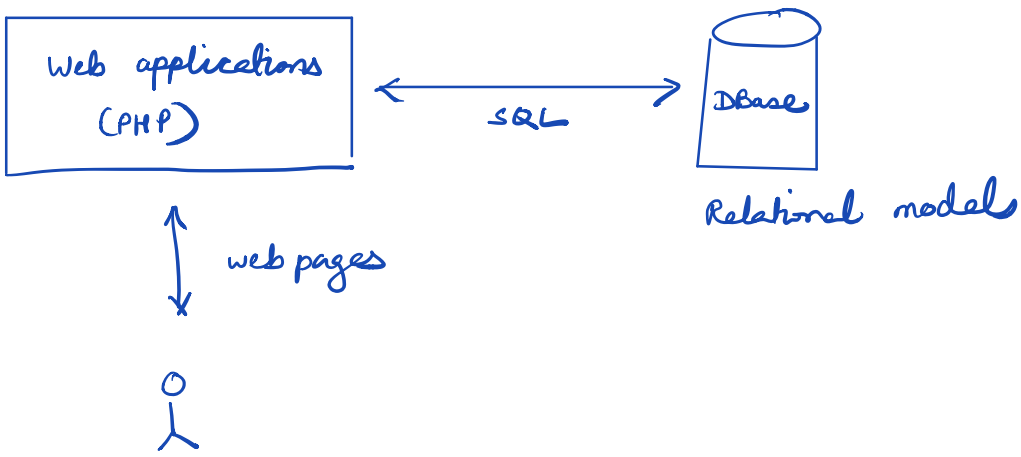
- Viewing the DB:
  - Through the DBMS, a **view** of the database can be created.
  - Subset of the data or combined data from multiple files
  - Different views for different types of users

TRANSCRIPT

Student_name	Student_transcript				
	Course_number	Grade	Semester	Year	Section_id
Smith	CS1310	C	Fall	08	119
	MATH2410	B	Fall	08	112
Brown	MATH2410	A	Fall	07	85
	CS1310	A	Fall	07	92
	CS3320	B	Spring	08	102
	CS3380	A	Fall	08	135

People

* database users
* application programmers
* database administrators
* database design



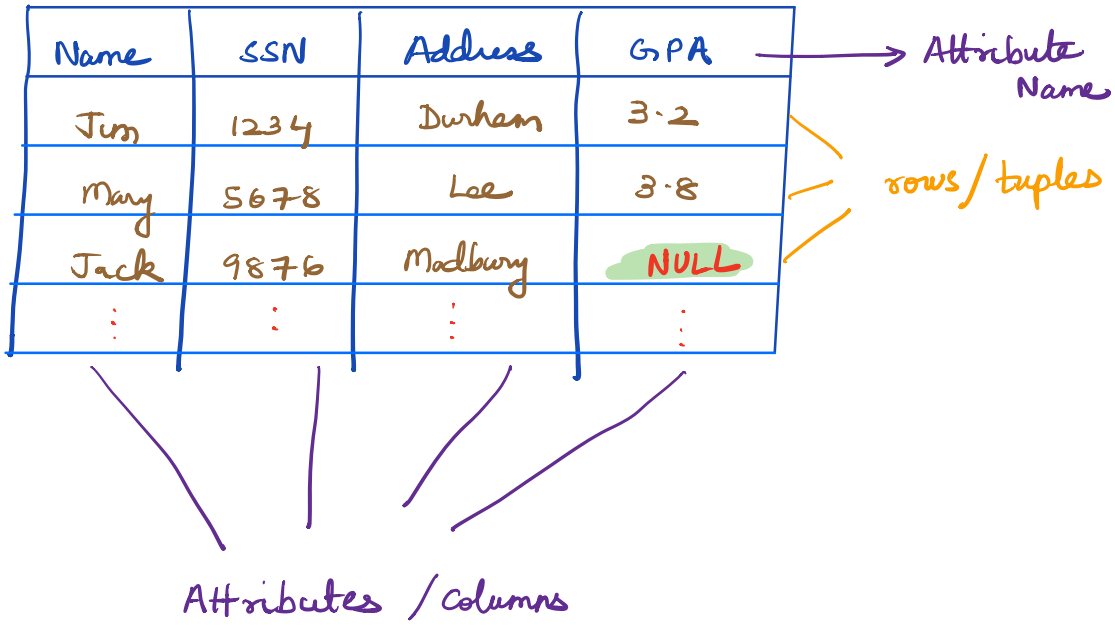
Chapter 5.1 new edition

3.1 old edition

Relational model

* Ted Codd, IBM 1970
* started a several billion dollar industry
* unit = table/relation (interconnected)
* theoretical basis: relational algebra & calculus

Student – Table Name



* Database consists of a set of tables
* Schema: structured description of the tables
* Instance: contents of tables

* no tuple (row) ordering
* column ordering matters

* each tuple (row) is distinct.
  ↳ key: minimal set of attributes whose value is unique in each tuple.

e.g. SSN – key SSN + address ✗

Lname + mi + fname - key

John A Smith

John B Smith

- table may have multiple keys
  ↓
  candidate keys

- one of the candidate keys is selected as a primary key.