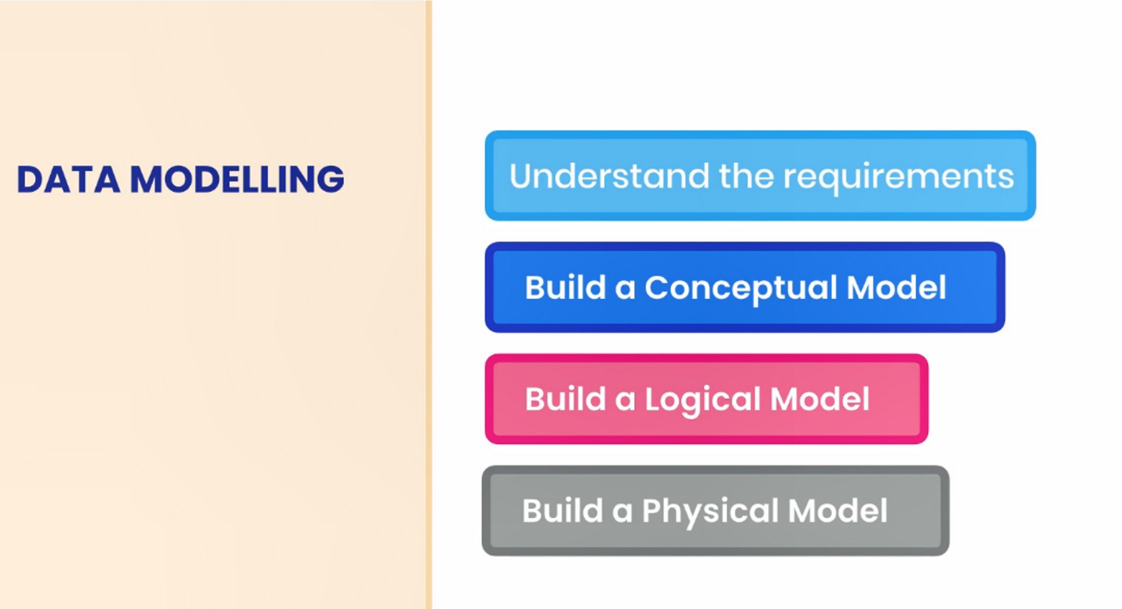
12. Designing Databases

Tuesday, April 16, 2024 8:18 AM



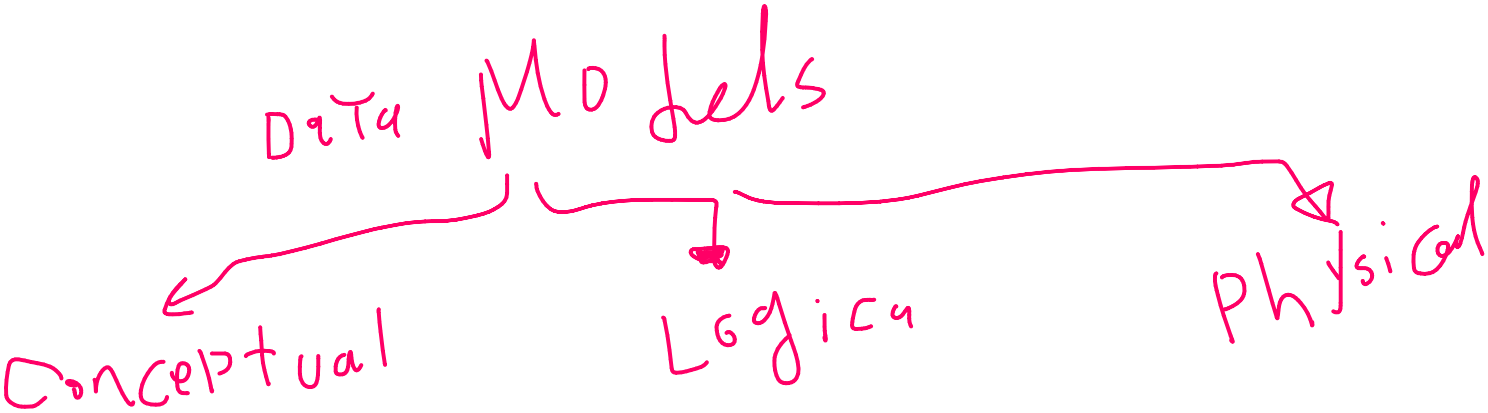
DATA MODELLING

Understand the requirements

Build a Conceptual Model

Build a Logical Model

Build a Physical Model



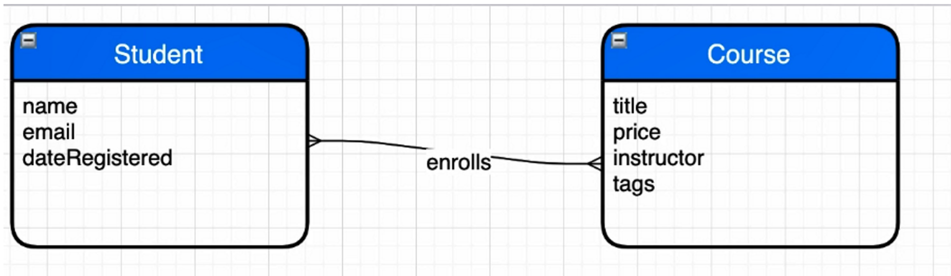
# Conceptual Model

You can see that it's the language that business stack holder and developers use to be in common area so they can talk freely

You can use both [ERD - UML] To draw the conceptual model

USEFUL WEBSITES

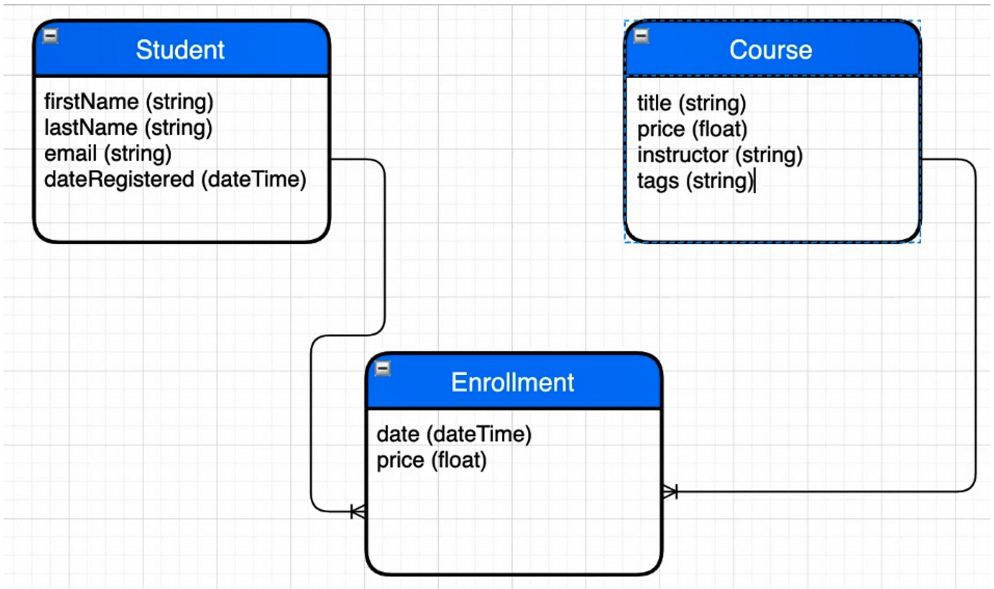
- [draw io]()
- [Microsoft Visio]()



[Note] There is no technical details here it's just concept where business owner and developers can talk and share ideas back and forth

# Logical Model

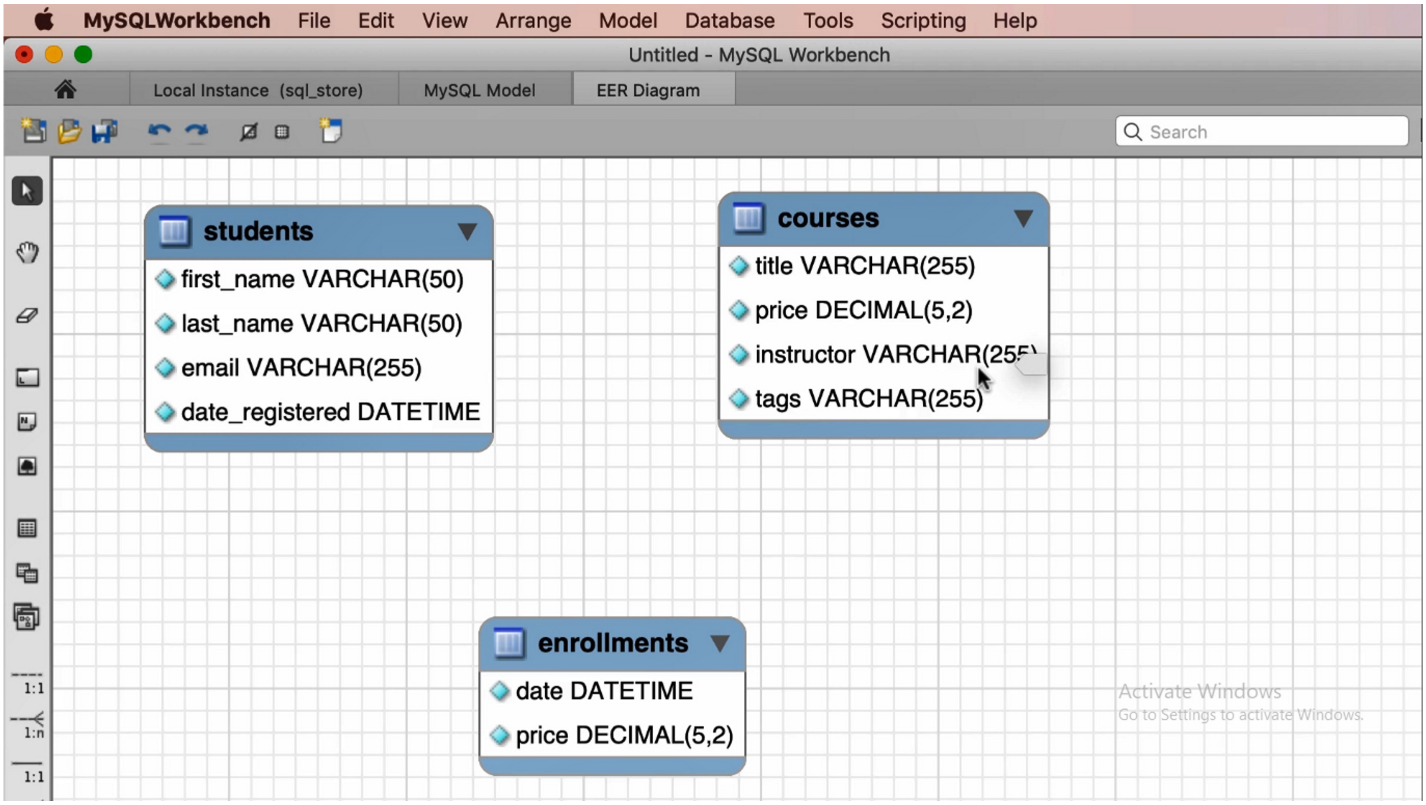
Here we talk about relations and tables we have in more details like data types and so on look below



Look Here we talk about our tables and relations in more details and write the data types of them not specific for particular DBMS
Just general concepts like **[String , float , …………… etc]**
Next Step is PHYSICAL MODEL WHICH IS IN MOST CASES MAPPING OR IMPLEMENTATION OF LOGICAL MODEL IN OUT DBMS

# Physical Model

Here you become so clearly about your DBMS that you use



students
- first_name VARCHAR(50)
- last_name VARCHAR(50)
- email VARCHAR(255)
- date_registered DATETIME

courses
- title VARCHAR(255)
- price DECIMAL(5,2)
- instructor VARCHAR(255)
- tags VARCHAR(255)

enrollments
- date DATETIME
- price DECIMAL(5,2)

Screen Shot From MYSQL workbench

# Normalization

Database normalization is a process in database design that involves organizing data in a way that minimizes redundancy, dependencies, and anomalies. It is a series of guidelines or rules that help create an efficient and well-structured relational database.
The primary objectives of database normalization are:

1. Eliminating data redundancy: Redundant data refers to the repetition of the same data in multiple places within a database. Normalization helps eliminate this redundancy by breaking up the data into smaller, logically organized tables.
2. Reducing data dependencies: Data dependencies occur when changes made to one piece of data require modifications in multiple places. By normalizing the database, data dependencies are minimized, making it easier to update and maintain the database.
3. Preventing anomalies: Anomalies are errors or inconsistencies that can occur in a database. Normalization helps prevent anomalies such as insertion, deletion, and update anomalies, which can lead to data inconsistencies and inaccuracies.

# First Normal Form (1NF)

- **Each column contains atomic (indivisible) values.**
- **Each column has a unique name.**
- **Each row in the table is unique; there are no duplicate rows.**
- **The order of rows and columns doesn't matter.**

# Second Normal Form (2NF)

With Super easy definition :
Every table should describe one entity , and every column in that table should describe that entity

# Third Normal Form

With super easy definition :
Every Column Exist in the table it must NOT depend on other columns , No Derived Columns

Total	**Payment**	~~**Balance**~~

Balance = Total - Payment

First Name	Last Name	~~Full Name~~

# SQL CODE FOR DESIGNING DATABASE

```
CREATE DATABASE IF EXIST MY_STORE ;
DROP DATABASE IF NOT EXIST MY_STORE ;


CREATE TABLE IF NOT EXIST MY_TABLE (

    Column_name data_type constrains

)


ALTER TABLE MY_TABLE
    ADD       column_name  anything    AFTER col
    MODIFY    column_name  anything
    DROP      column_name ;
```

```
ALTER TABLE customer
      ADD          last_name  VARCHAR(50) NOT NULL AFTER first_name ,
      MODIFY    first_name  VARCHAR(50) NOT NULL ,
      DROP first_name ;
```