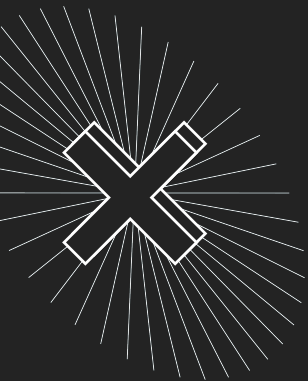


# Database

Postgres



# Table of contents

**01**

**Database  
Design**

**02**

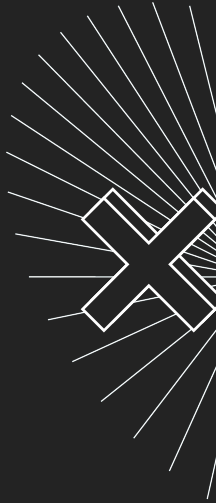
**Primary Key &  
Foreign Key**

**03**

**ALTER Table**

**04**

**Relational  
System + ALTER**



# Database Design Basics

- Database design is about organizing data efficiently. Think of it like planning a library:
  - **Table:** type of object. (Different shelves)
  - **Row:** one record. (Individual items on each shelf)
  - **Column:** one attribute (Details about each item like title, author)
- **Bad** design example:  
CREATE TABLE messy\_data (name TEXT, course TEXT, instructor TEXT, grade TEXT);
- **Problem:**
  - Repeats course info for every student.
  - Hard to update instructor names.
  - Wastes spaces.

# Database Design Basics

- **Good** design example:
  - CREATE TABLE students (student\_id SERIAL PRIMARY KEY, name VARCHAR(20));
  - CREATE TABLE course (course\_id SERIAL PRIMARY KEY, title VARCHAR(100), instructor VARCHAR(50));
  - CREATE TABLE enrollment (student\_id INT REFERENCES students(student\_id), course\_id INT REFERENCES courses(course\_id), grade VARCHAR(2));
- **Relationship Types:**
  - **One to One:** User - Profile.
  - **One to Many :** Instructor - Courses.
  - **Many to many:** Students - Courses.



# PRIMARY & FOREIGN KEYS

- **Primary Key:**

- **Unique ID for each row.**
- **Never Null.**

```
CREATE TABLE instructors (instructor_id INT PRIMARY KEY, name VARCHAR(20));
```

- **Foreign Key:**

- **Links to a primary key in another table**
- **Maintains relationships**

```
CREATE TABLE courser (course_id SERIAL PRIMARY KEY, title VARCHAR(20),  
instructor_id INT REFERENCES instructors(instructor_id));
```



# Alter tables in Postgres:

- Why to use:
  - Evolving database.
  - Fixing mistakes.
  - Add new requirements.
- **ALTER TABLE**
  - **Add column:** ALTER TABLE students ADD email VARCHAR(20);
  - **Drop column:** ALTER TABLE students DROP COLUMN email;
  - **Rename column:** ALTER TABLE students RENAME COLUMN name TO full\_name;
  - **Alter type:** ALTER TABLE students ALTER COLUMN gpa TYPE REAL;
  - **Add constraints:** ALTER TABLE students ADD CONSTRAINT unique\_email UNIQUE (email);



# Foreign Key with ALTER

```
ALTER TABLE enrollments  
ADD CONSTRAINT fk_student  
FOREIGN KEY (student_id) REFERENCES students(student_id);
```



# LAB

## 1- Create companies table

company\_id (SERIAL PRIMARY KEY)  
name (VARCHAR(100)) NOT NULL  
industry VARCHAR(50)  
Founded\_date DATE  
Is\_hiring BOOLEAN true by default

## 2- Create a courses job with:

job\_id (SERIAL PRIMARY KEY)  
title (VARCHAR(100)) NOT NULL  
salary NUMERIC(10,2)  
posted\_date DATE  
company\_id INT FK with company

## 3- Create an applications table:

application\_id (SERIAL PRIMARY KEY)  
job\_seeker\_name (VARCHAR(100)) NOT NULL  
application\_date DATE  
job\_id INT FK with jobs





# LAB

- 4- Add a column location to companies.
- 5- Rename salary to monthly\_salary in jobs
- 6- Add a UNIQUE constraint on title in jobs.
- 7- Link a new applicant\_email field to a future users table

