# Bootcamp 134 | Python

## Course 22 | DataBase

Amir Hossein Chegouniyan

Head of the Technical Team at Dariche Tejarat

Lecturer of Python – Django at Maktab Sharif

Amirhossein-chegounian

# Content

# Introduction to ERD

- What is an ERD?:
  - Overview of ER diagrams and their purpose in database design.
- Key Components:
  - Entities: Real-world objects or concepts represented as tables in a database.
  - Attributes: Properties or details about an entity (e.g., name, age, ID).
  - Relationships: Connections between entities (e.g., one-to-many, many-to-many).
- Notations in ERD: (HW)
  - Crow’s Foot, Chen, UML, and their variations.

# Identifying and Defining Relationships

- Types of Relationships:
  - One-to-One (1:1): Examples and use cases.
  - One-to-Many (1:N): Common in relational databases, like customers and orders.
  - Many-to-Many (M:N): How to handle these relationships using junction tables.
- Primary and Foreign Keys:
  - Importance of keys in defining relationships.
  - Examples of primary and foreign key connections.

# Designing an ERD

- Steps to Create an ERD:
  - Identify entities.
  - Define attributes.
  - Determine relationships.
  - Assign keys.
- Normalization Basics:
  - Eliminating redundancy and ensuring data consistency.
  - Connection to ERD design.

# Introduction to Databases

- What is a database? Overview of relational databases.
- Introduction to SQL as a language for interacting with databases.
- Difference between popular database systems (e.g., PostgreSQL, MySQL, SQLite). (HW)

# SQL Basics

- SQL Syntax Rules: Understanding the structure and conventions of SQL statements.
- SELECT Statements: Retrieving data from tables.
- Selecting Multiple Columns: How to retrieve multiple fields in a query.
- DISTINCT and LIMIT: Removing duplicates and limiting the number of results.
- Sorting Results: Using ORDER BY for ascending and descending order.

# Filtering Data

- The WHERE Statement: Applying conditions to filter data.
- Combining Filters: Using AND and OR for complex conditions.
- Advanced Filters: Using IN, NOT IN, and LIKE for pattern matching and set membership.

# Customizing Results

- Custom Columns: Creating calculated fields or aliasing columns.
- Functions: Using built-in functions like MIN, MAX, AVG, and SUM.
- Subqueries: Writing queries within queries for complex data retrieval.

# Grouping and Aggregation

- GROUP BY Clause: Aggregating data by specific fields.
- Filtering Groups: Using the HAVING clause to refine grouped results.

# Any question?

# Next course

- HTTP Server in Python
- Software Testing in Python
- Introduction to Pytest
- Backend Development Practices