بسم الله الرحمن الرحیم

تکنولوژی کامپیوتر

جلسه‌ی دهم
پایگاه داده

جلسه‌ی گذشته

دیدن بیشتر template ،css در golang

جلسه‌ی جدید

پایگاه داده

چرا پایگاه داده؟

انواع پایگاه‌های داده

انواع پایگاه‌های داده

- بر اساس داده‌ای که توش ذخیره می‌شه.
- براساس روشی که می‌شه ازش داده گرفت

SQL

SQL Databases

- MySQL and MariaDB
- MSSQL
- Postgres (Postgresql)
- Sqlite (Library)

Postgres

- Installation
- Use pgAdmin to view

Database

- Database -> Schema -> (Tables, Views, Functions)

Tables

مثال وبلاگ

Posts

id	title	content	created_at	Author_id
1	سلام	محتوا	2025-03-10	1
2	تست	تستتست س	2025-03-14	2
3	هم	مممم؟	2024-03-14	1

Users

ID	User Name	Encrypted Password	Display Name
1	admin		صاحاب
2	ali		علی

Comments

ID	Post_id	Content
1	1	جالبه
2	2	خوبه
3	1	آه
4	1	خوبه

دیدن این سه جدول در pgadmin

DATA QUERY LANGUAGE (DQL)

SELECT کوئری

```
SELECT [DISTINCT]
       { column_name [ [AS] alias ] | table_name.column_name | expression } [ , ... ]
FROM table_name [ [AS] table_alias ]
     [JOIN/LEFT JOIN/RIGHT JOIN/FULL JOIN other_table
       ON join_condition]  -- more on JOIN syntax below
[WHERE condition]
[GROUP BY expression [ , ... ]]
[HAVING condition]
[ORDER BY expression [ASC | DESC] [ , ... ]]
[LIMIT count]
[OFFSET start]
```

WHERE

- =, >, <, …
- AND, OR
- LIKE

JOIN

- INNER JOIN
- LEFT JOIN
- RIGHT JOIN
- FULL JOIN
- CROSS JOIN

DATA MODIFICATION LANGUAGE (DML)

کوئری INSERT

```
INSERT INTO table_name (column1, column2, ...)
VALUES (value1, value2, ...),
       (value1b, value2b, ...)  -- multi-row insert
[RETURNING * | output_expression [ AS col_alias ] [ , ... ] ];
```

کوئری INSERT

```
INSERT INTO table_name (column1, column2, ...)
VALUES (value1, value2, ...),
       (value1b, value2b, ...)  -- multi-row insert
[RETURNING * | output_expression [ AS col_alias ] [ , ... ] ];
```

کوئری UPDATE

```
UPDATE table_name
SET column1 = expression1,
    column2 = expression2,
    ...
[WHERE condition]
[RETURNING * | output_expression [ , ... ]];
```

کوئری DELETE

```
DELETE FROM table_name
[WHERE condition]
[RETURNING * | output_expression [ , ... ]];
```

DATA DEFINITION LANGUAGE (DDL)

```
CREATE TABLE [IF NOT EXISTS] table_name (
    column_name data_type [DEFAULT default_expr] [column_constraint],
    column_name data_type [DEFAULT default_expr] [column_constraint],
    ...
    [table_constraints]
);
```

- **data_type**: `INTEGER`, `SERIAL`, `VARCHAR(n)`, `TEXT`, `DATE`, `TIMESTAMP`, `BOOLEAN`, etc.
- **column_constraint** examples: `NOT NULL`, `UNIQUE`, `PRIMARY KEY`, `CHECK (expression)`
- **table_constraints** examples:
  - `PRIMARY KEY (column_list)`
  - `FOREIGN KEY (column_list) REFERENCES other_table (column_list)`
  - `UNIQUE (column_list)`

Constraints and Keys

- PRIMARY KEY: Uniquely identifies a row in a table.
- FOREIGN KEY: Ensures referential integrity with another table.
- NOT NULL: Column must have a value.
- UNIQUE: Column values must not repeat.
- CHECK: Custom condition that row values must satisfy.

```
ALTER TABLE table_name
    ADD [COLUMN] column_name data_type [constraint],
    DROP [COLUMN] column_name [CASCADE],
    ALTER [COLUMN] column_name [SET DATA TYPE new_data_type],
    RENAME [COLUMN] old_name TO new_name,
    RENAME TO new_table_name,
    ...
```

```
DROP TABLE [IF EXISTS] table_name [CASCADE | RESTRICT];
```

INDEXES

```
CREATE [UNIQUE] INDEX index_name
ON table_name [USING method] (column_name [ASC|DESC] [ , ... ])
[WITH ( storage_parameter = value [ , ... ] )]
[WHERE predicate];
```

- **UNIQUE**: Enforces uniqueness on the indexed columns.
- **method**: Typically `btree` (default), `hash`, `gin`, or `gist`, etc.
- **WHERE**: A partial index (only index rows that satisfy the condition).

TRANSACTIONS AND CONCURRENCY

TRANSACTIONS

LOCKS

SELECT … FOR UPDATE

گولنگ

لایبری pgx

مایگریشن