Databases and Supabase

Database, SQL, Supabase, Connecting JS Apps with Supabase, Supabase MCP

Svetlin Nakov, PhD

Co-founder @ SoftUni

SoftUni AI

https://ai.softuni.bg

Agenda

SoftUni AI

1. **Software Technologies**: Front-End, Back-End, Databases
2. **Databases**: Tables, Rows, Columns, Relationships, DB Schema
3. **Supabase** Intro: Projects, Database, Schema, Tables, Table Editor
4. **Database Design**: DB Tables, Data Types, DB Schemas
5. **SQL Language**: Table Structure: CREATE / ALTER TABLE, CRUD Operations: SELECT, INSERT, UPDATE, DELETE, Joins
6. **Supabase Services**: Database, Auth, Storage, Edge Functions, Logs, Backup, API Keys, Row-Level Security (RLS)
7. **JS Apps with Supabase**: API Connection, CRUD Operations
8. **Supabase MCP**: Connecting GitHub Copilot with Supabase

Sli.do Code

#Soft-Tech-AI

Join at

slido.com

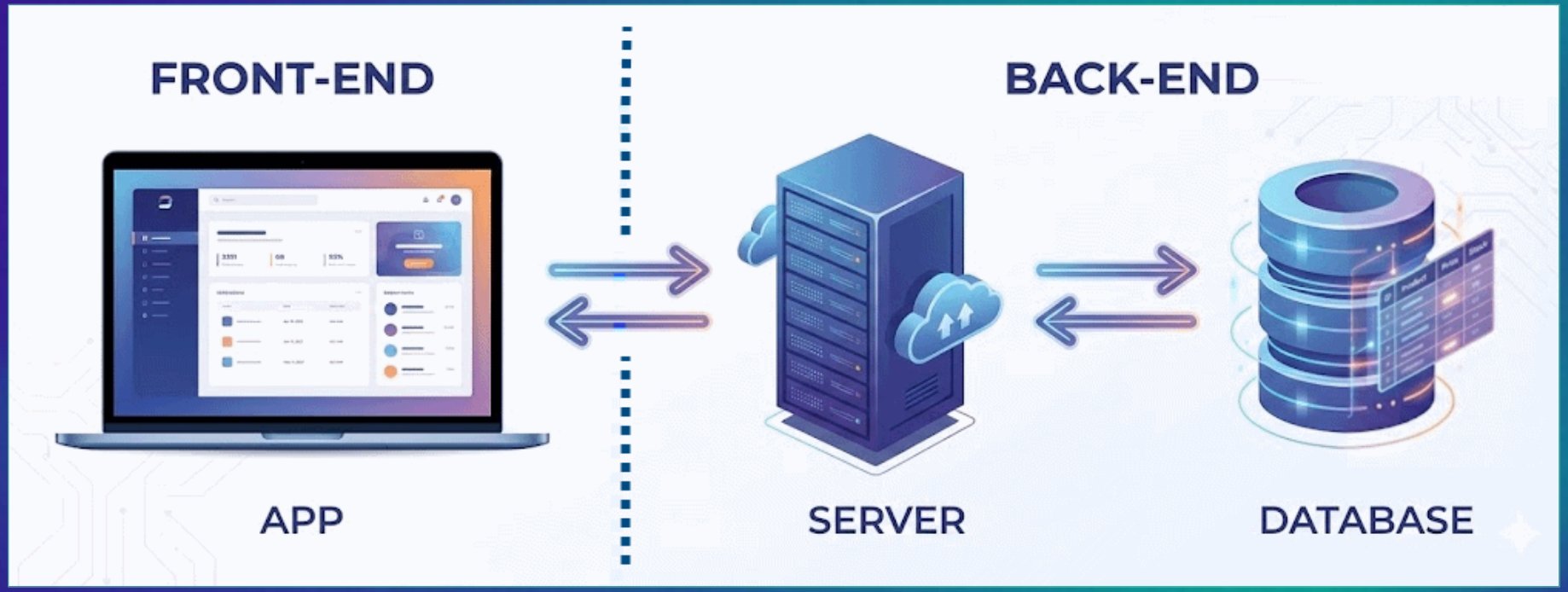
#Soft-Tech-AI

Breaks

20:00 / 21:00

Front-End and Back-End

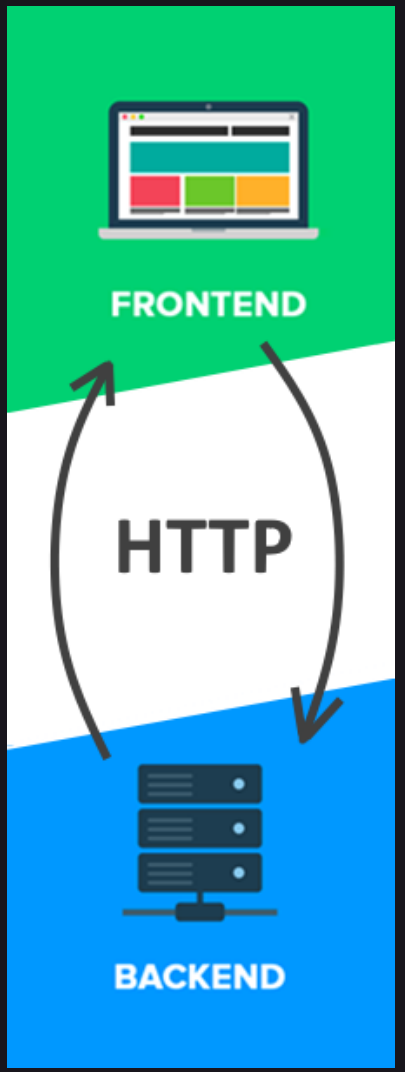
Software Architecture and Technologies, Front-End, Back-End, Databases



Front-End and Back-End

SoftUni AI

- **Front-end** and **back-end** split modern apps into **client-side** components (UI) and **server-side** components (data & logic)
- **Front-end** == client-side components
  - Implement the **user interface** (UI)
- **Back-end** == server-side components (data and business logic + APIs)
  - Implements **data storage** and **processing logic**



Front-End Technologies

SoftUni AI

- **Front-end** technologies
  - Implement app's **user interface** (UI)
  - HTML, CSS, JavaScript, TypeScript
  - React, Tailwind, Vite, React Router

- **Front-end developers** deal with UI, UX and front-end technologies and frameworks
- **AI-assisted dev tools** generate front-end using AI
  - GitHub Copilot, Cursor, Lovable, Bolt, Replit, Firebase Studio
  - Developers focus on the app logic and UI, not on the code

Back-End Technologies

SoftUni AI

- **Back-end** technologies
  - Implement **server-side logic** and **data storage**
  - Technologies: Node.js, Express.js, Next.js
  - Store data in **databases** (like Supabase)
  - Expose **APIs** (access points for the front-end)
  - Server-side components are typically hosted in the **cloud**

- **Backend as a Service (BaaS)**, e. g. **Supabase**
  - Implement app's data storage, processing and authentication in the cloud, as a service

Intro to Databases

Databases, Relational Model, Tables, Rows, Columns, Relationships, DB Schema

What is a Database?

SoftUni AI

- A **database** is a collection of data, organized to be easily accessed, managed and updated
- Modern databases are managed by a Database Management Systems (**DBMS**), like **PostgreSQL**
  - Define database **structure**, e. g. tables, collections, columns, relations, indexes
  - **CRUD** operations: **C**reate / **R**ead / **U**pdate / **D**elete data (table rows / entities)
  - Execute **queries** (filter / search data)

Databases

SoftUni AI

- **Databases** hold and manage data in the back-end systems
- **Relational databases** (RDBMS systems)
  - Hold data in **tables** + **relationships**
  - Use the **SQL** language to query / modify data
  - Examples: **PostgreSQL**, **Supabase**, **MySQL**

- **NoSQL** databases
  - Hold **document collections** or **key-value pairs**
  - Examples: **MongoDB**, **IndexedDB**, **Redis**

Relational Databases (SQL Databases)

SoftUni AI

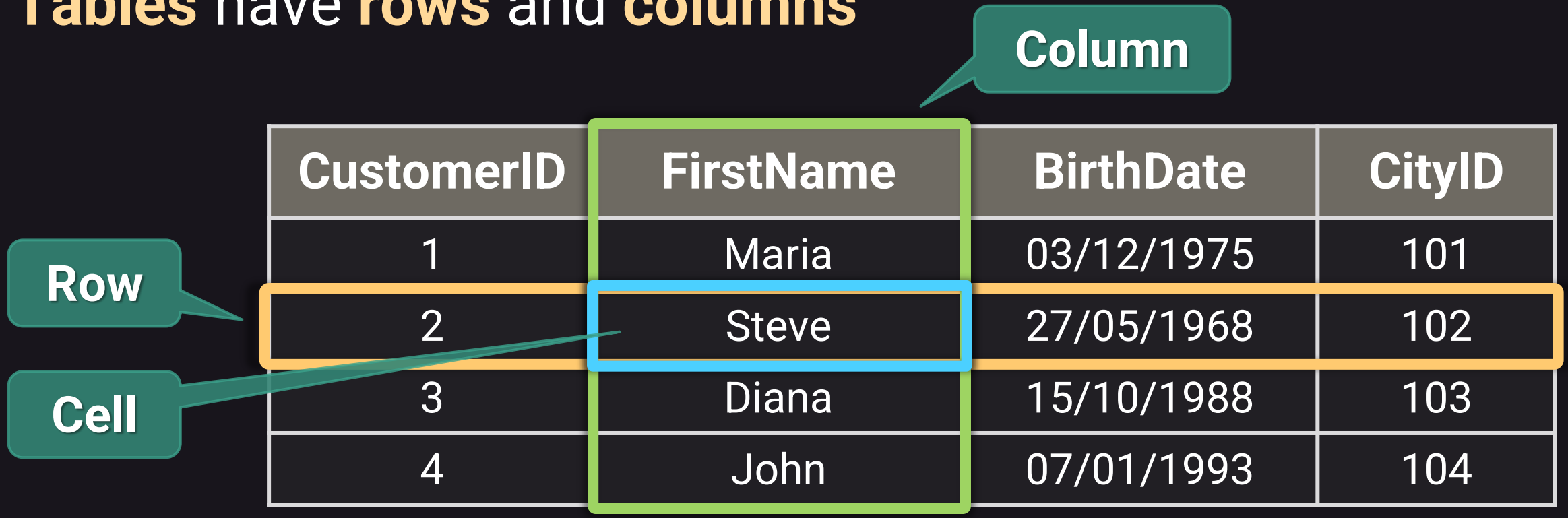
- **Relational (SQL) databases** organize data in **tables**
  - **Tables** have strict **structure** (columns of certain data types)
  - Can have **relationships** (connections) to other tables
- Relational databases use the **S**tructured **Q**uery **L**anguage (**SQL**) to define / manipulate data
  - Extremely **powerful** for complex queries

- **Relational databases** are the **most widespread** data management technology

Tables, Rows, Columns, Cells

SoftUni AI

- **Relational databases** (like Supabase) hold data in **tables**
- **Tables** have **rows** and **columns**



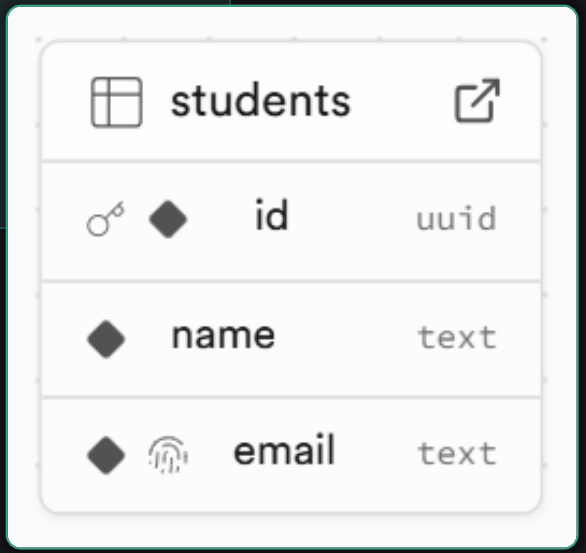
- **Columns** define **data type** (e. g. text, number, date)

SQL Language

SoftUni AI

- **SQL** is a database-level language, used to **define** table structure, **query** and **modify** table data


```
CREATE TABLE students (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  name TEXT NOT NULL,
  email TEXT NOT NULL UNIQUE
)
```




```
INSERT INTO students(name, email) VALUES
('Steve', 'steve@gmail.com')
```

```
SELECT name, email FROM students
```

Relationships between Tables

- Relational data is stored into database **tables** with
  - A **primary key** (unique key identifying each row) and
  - **Foreign keys** defining relationships (connections)

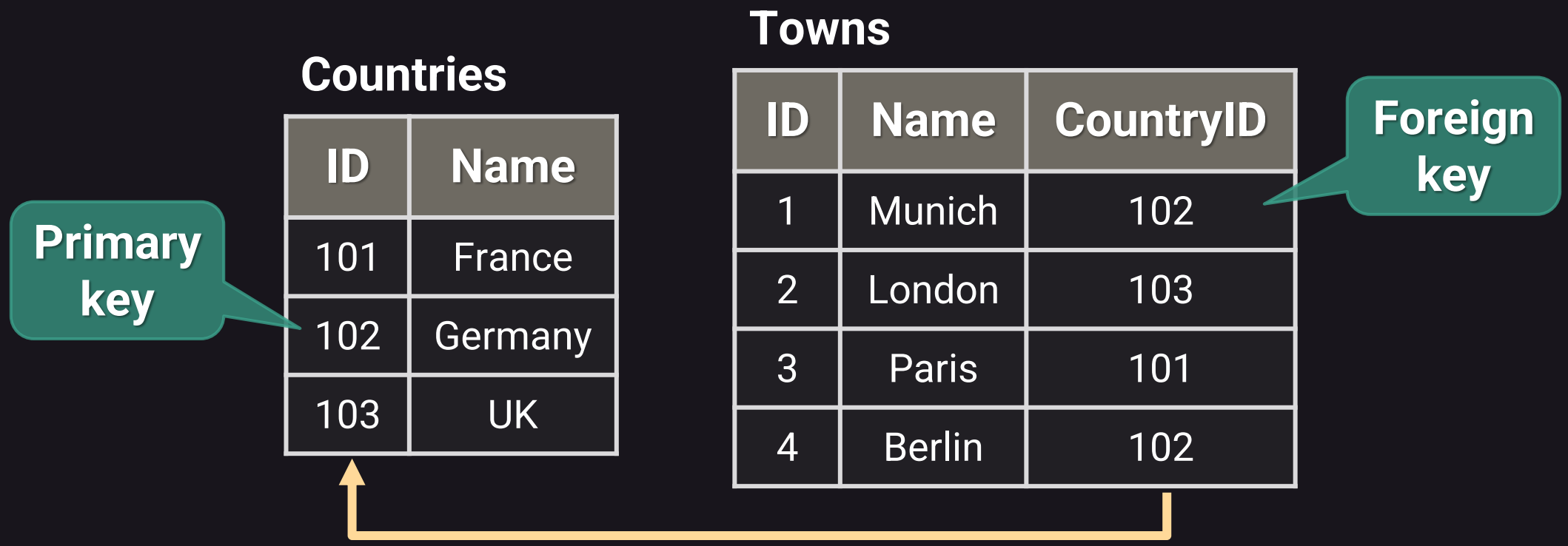


Table Relationships – Example

SoftUni AI

OrderItems

ID	OrderID	Name	Quantity	Price
1	11	Table	1	250.00
2	11	Chair	4	160.00

Customers

ID	Name	Email
101	Peter	peter.j@yahoo.com
102	Jayne	jayne98@gmail.com

Orders

ID	CustomerID	Date	Total Price
11	101	1/07/2026	410.00
12	102	12/07/2026	570.30

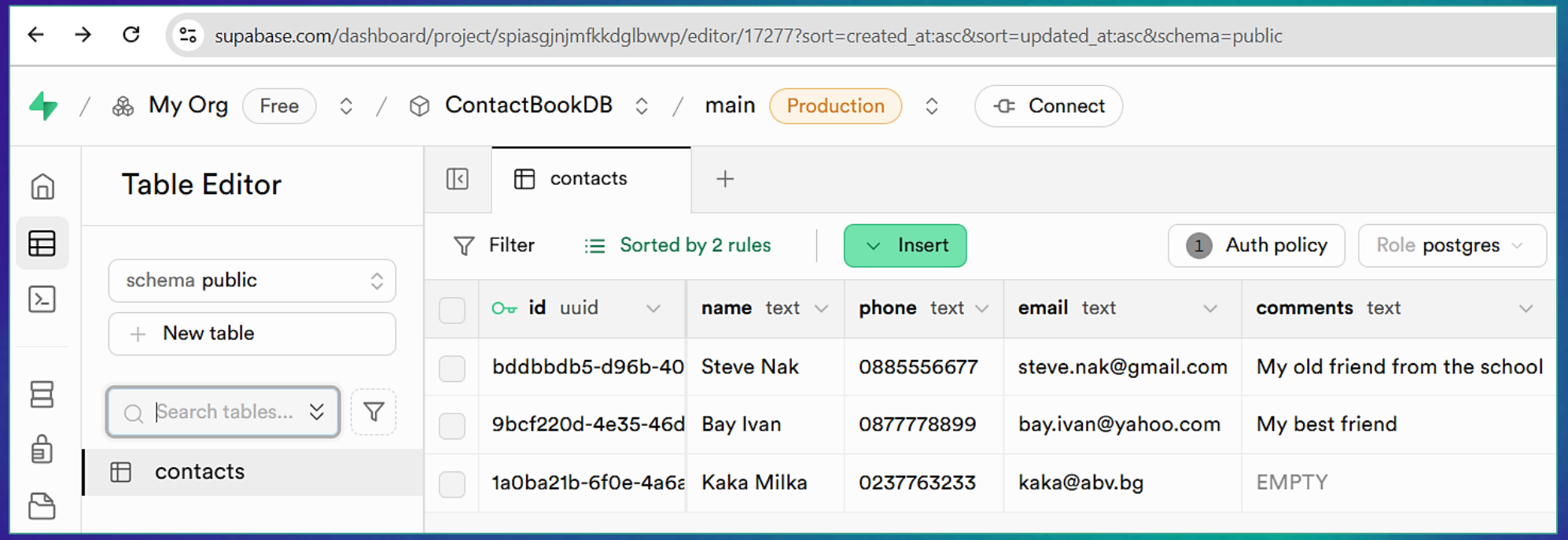
Database Servers

SoftUni AI

- **Database servers** (DBMS systems) **define**, **manipulate**, **retrieve** and **manage** data in a database
  - Local DB servers, cloud-bases DB servers
- Examples of DB servers:
  - **MySQL** / **MariaDB** – very popular, simple, open-source, for Web sites / apps
  - **PostgreSQL** – very powerful, open-source, for complex projects and apps
  - **Supabase** – hosted cloud instances of **PostgreSQL** + services
  - Others: **MS SQL Server**, **Oracle**, **MongoDB**, **Redis**, **SQLite**, …

Supabase Intro

Supabase BaaS, Projects, Databases, DB Schema, Table Editor, SQL Editor



Supabase

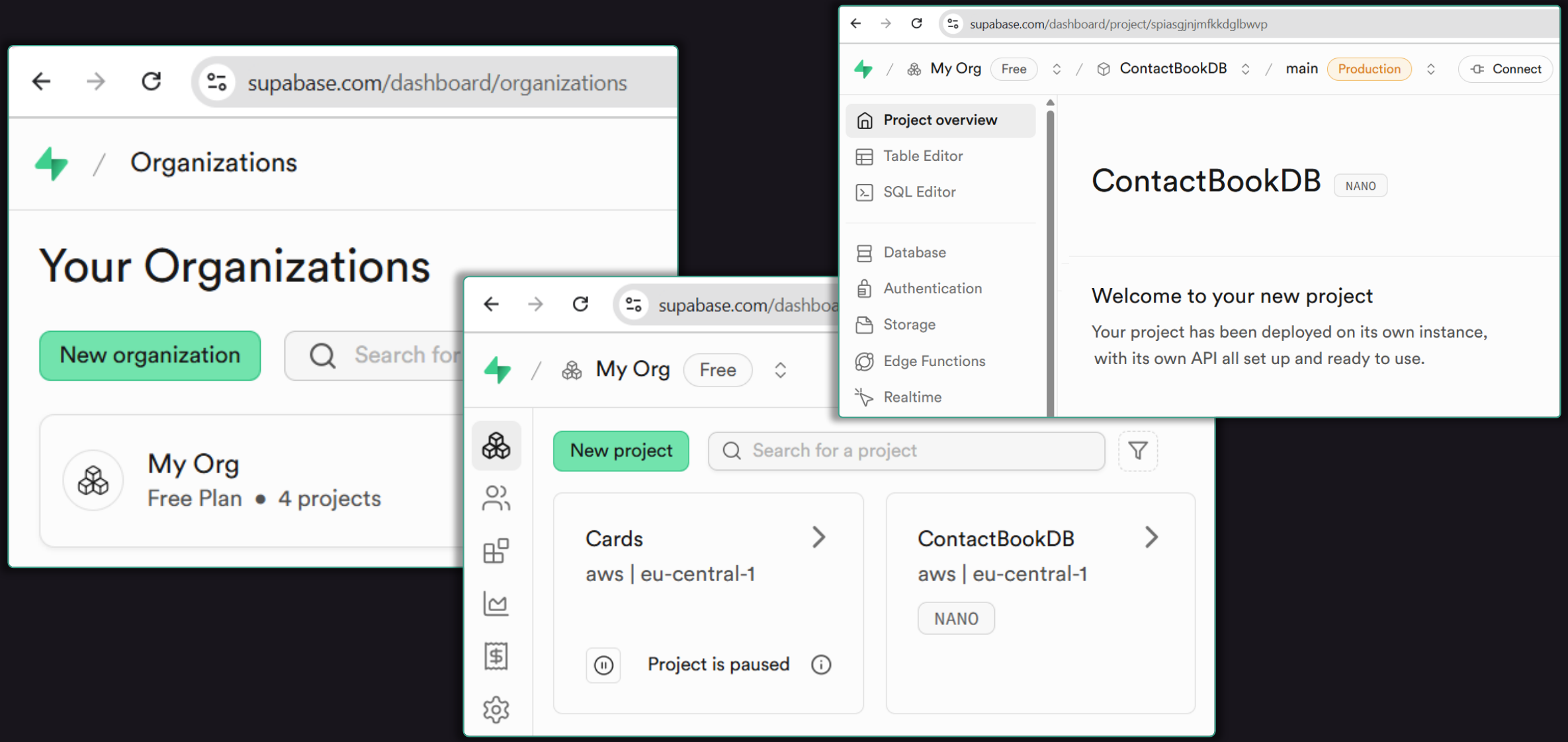
SoftUni AI

- **Supabase** is a popular open-source **BaaS** platform (Backend-as-a-Service)

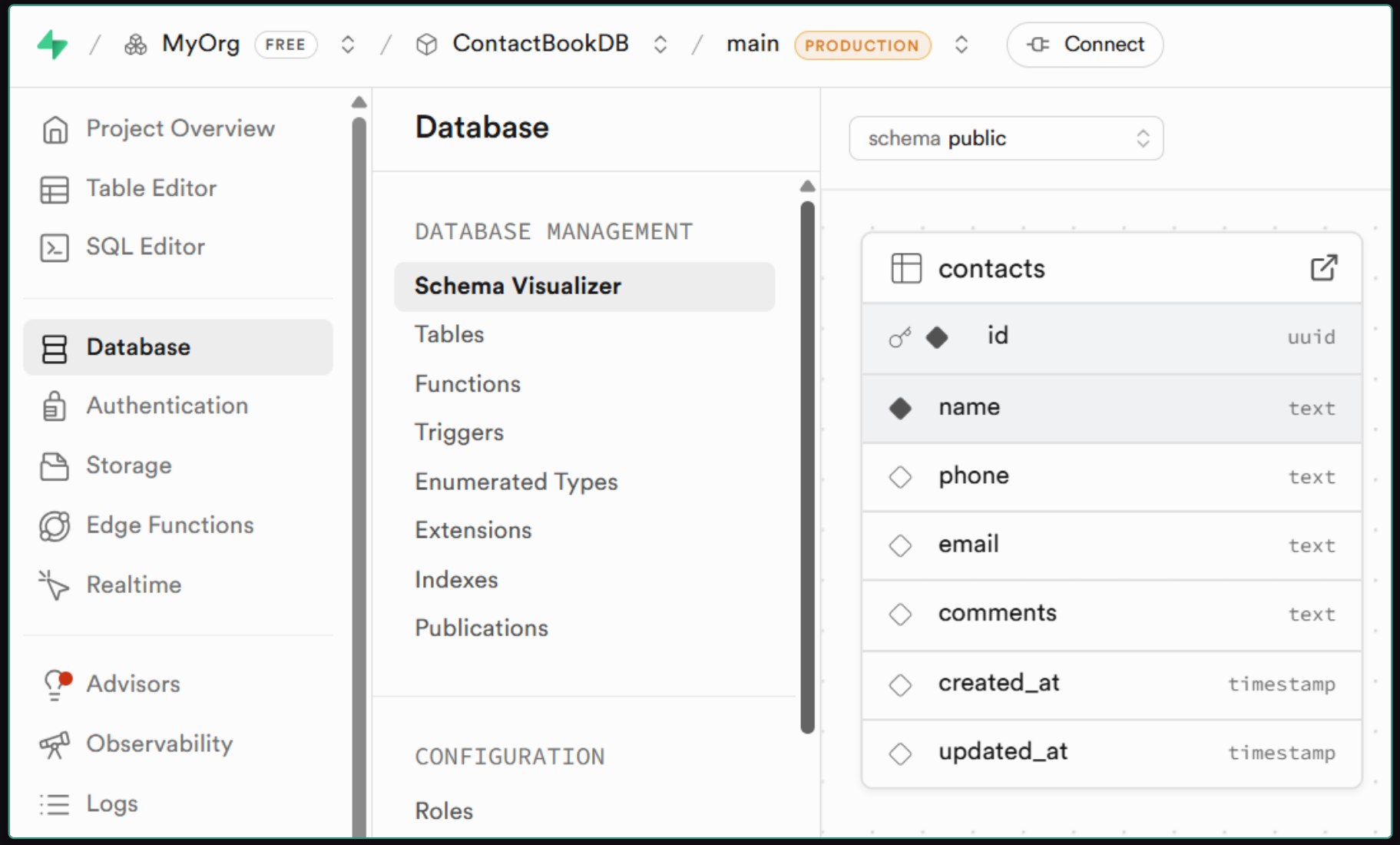
  - **PostgreSQL** database (as a service in the cloud)
  - Built-in **authentication** (user management)
  - Serverless **storage** (images, files, documents, media)
  - **Edge functions** (serverless code in the cloud)
  - Real-time **events**, message **queues**, **vector** DB, **cron** jobs
  - https://supabase.com

Supabase Orgs and Projects

SoftUni AI

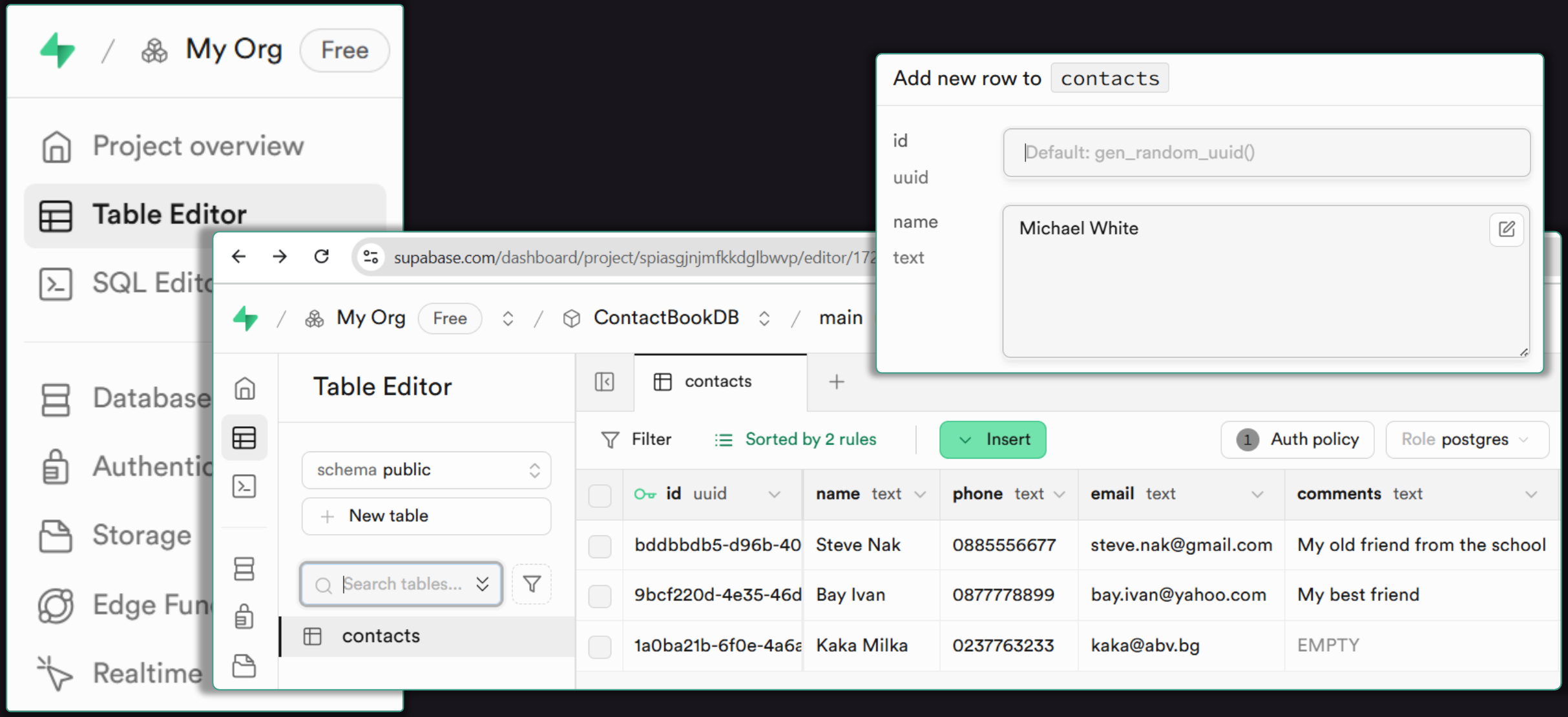


Database Schema View



Supabase Table Editor

SoftUni AI



Database Schema Design

Defining Table Structure, Relationships and Database Schema

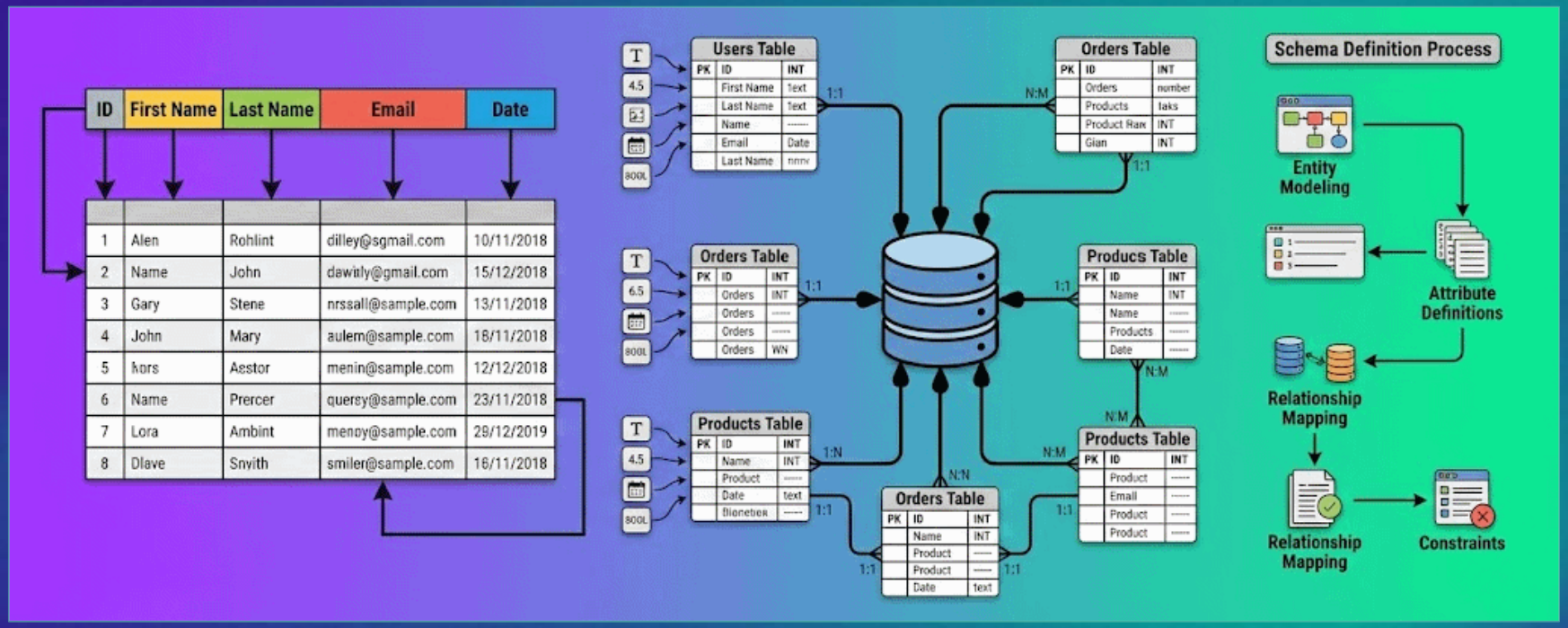
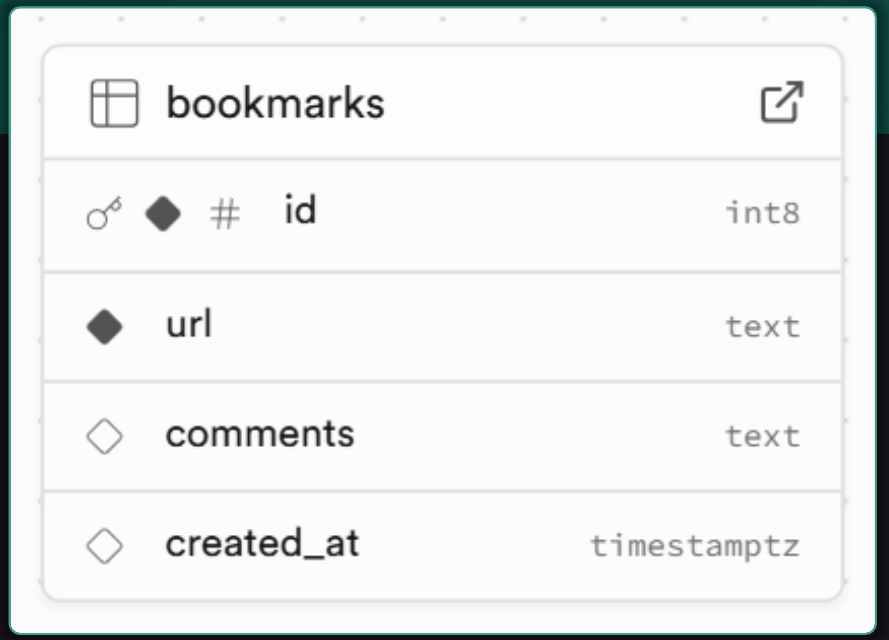
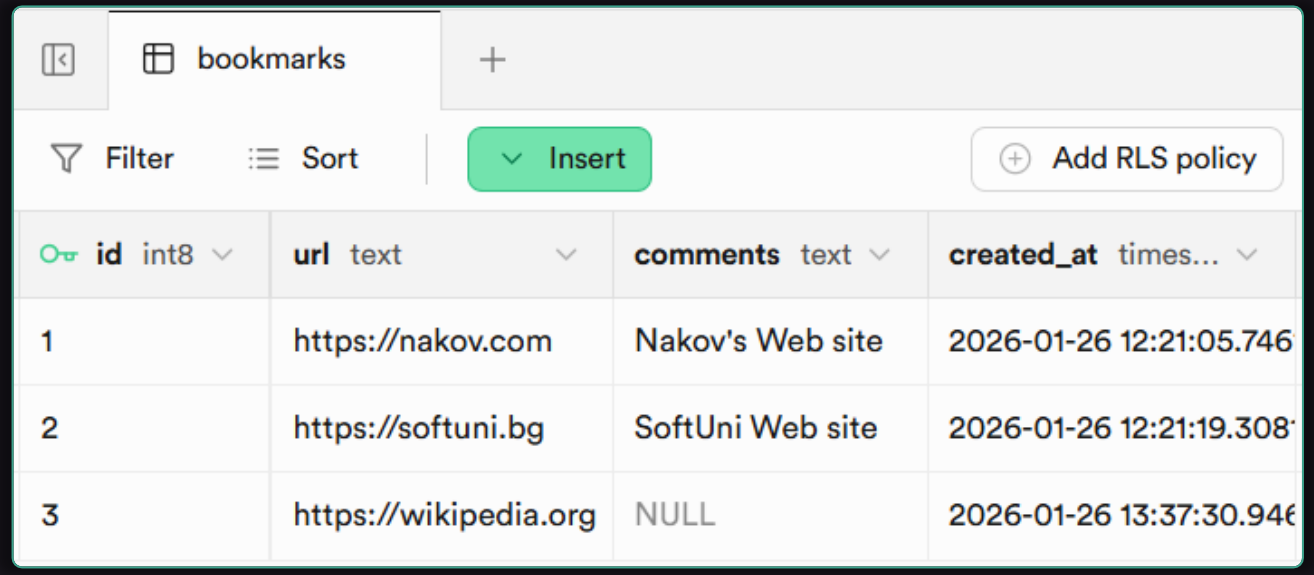


Table Structure

- DB table structure: **columns** of certain **data type**
- Example: table **Bookmarks**
  - id: **int** (primary key)
  - url: **text** (not null)
  - comments: **text**
  - created_at: **date & time**





PostgreSQL Data Types

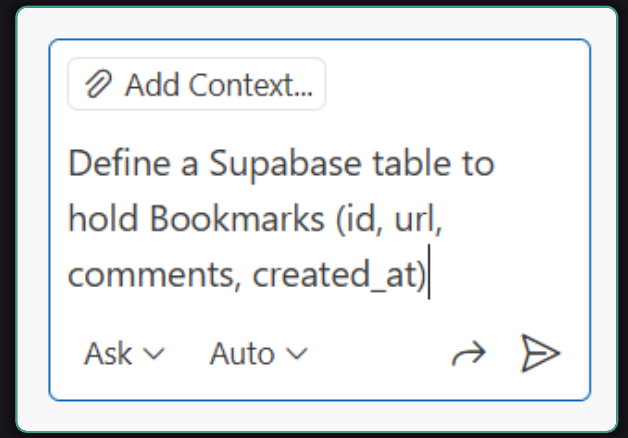
SoftUni AI

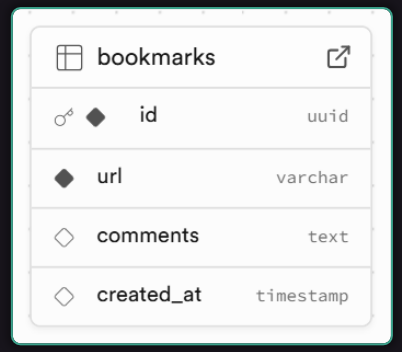
- **Columns** in the database tables have **data type**:
  - **integer** / **bigint**, e. g. 150
  - **uuid** (unique id), e. g. 373cc6ac-cf4e-470d-9ff4-f2f611b23958
  - **text** / **varchar**, e. g. "Steve Smith"
  - **timestamptz** (date + time), e. g. 2026-01-26 12:21:05.746115+00
  - **numeric(10, 2)** (money), e. g. 3.45
  - **jsonb** (JSON object), e.g. {"name":"Steve", "age":28}
  - **text[]** (array of strings), ["news", "front-page", "news"]

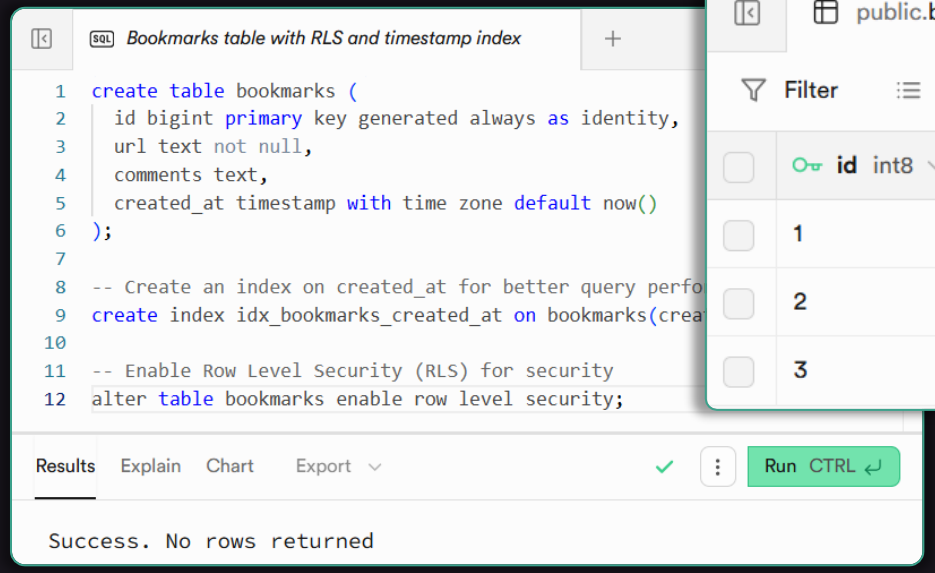
Exercise: Bookmarks DB Table

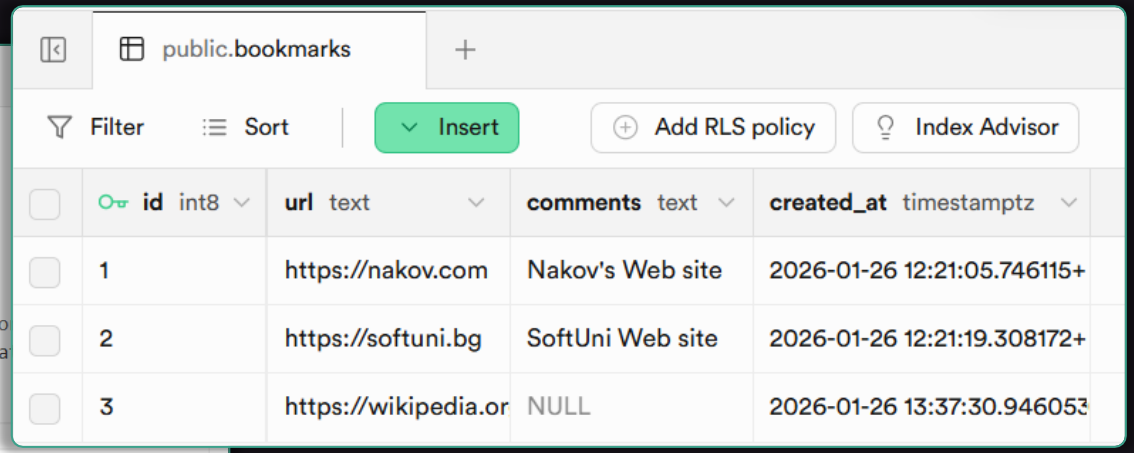
SoftUni AI

- Using an AI chatbot, define a Supabase table to hold **Bookmarks** (**id**, **url**, **comments**, **created_at**)
- **Run the SQL** to create the table in the DB
- Insert a few **data rows**





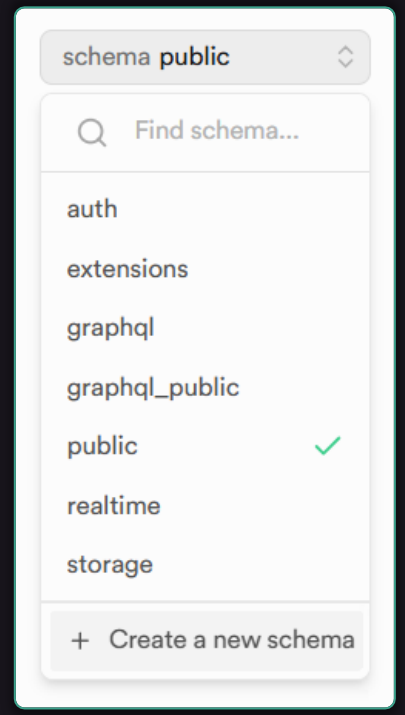




Database Schema

SoftUni AI

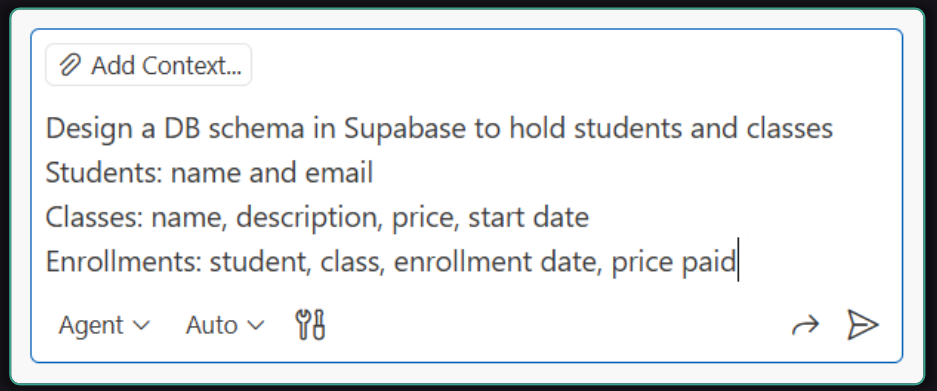
- **Database schema** is the structure of your database
  - The structure of all **tables** (columns, data types)
  - **Relationships** (primary key / foreign key)
  - Unique keys, indices, triggers, functions
- **Supabase** databases hold **multiple schemas**:
  - **public** (default schema) – your app's tables
  - **auth** (Supabase schema) – users and auth
  - **storage** (Supabase schema) – buckets and files



Exercise: Design a Database Schema

SoftUni AI

- Design a **DB schema in Supabase** to hold students and classes
  - **Students**: name and email
  - **Classes**: name, description, price, start date
  - **Enrollments**: student, class, enrollment date, price paid
  - Create **SQL script** to create the DB tables and relationships



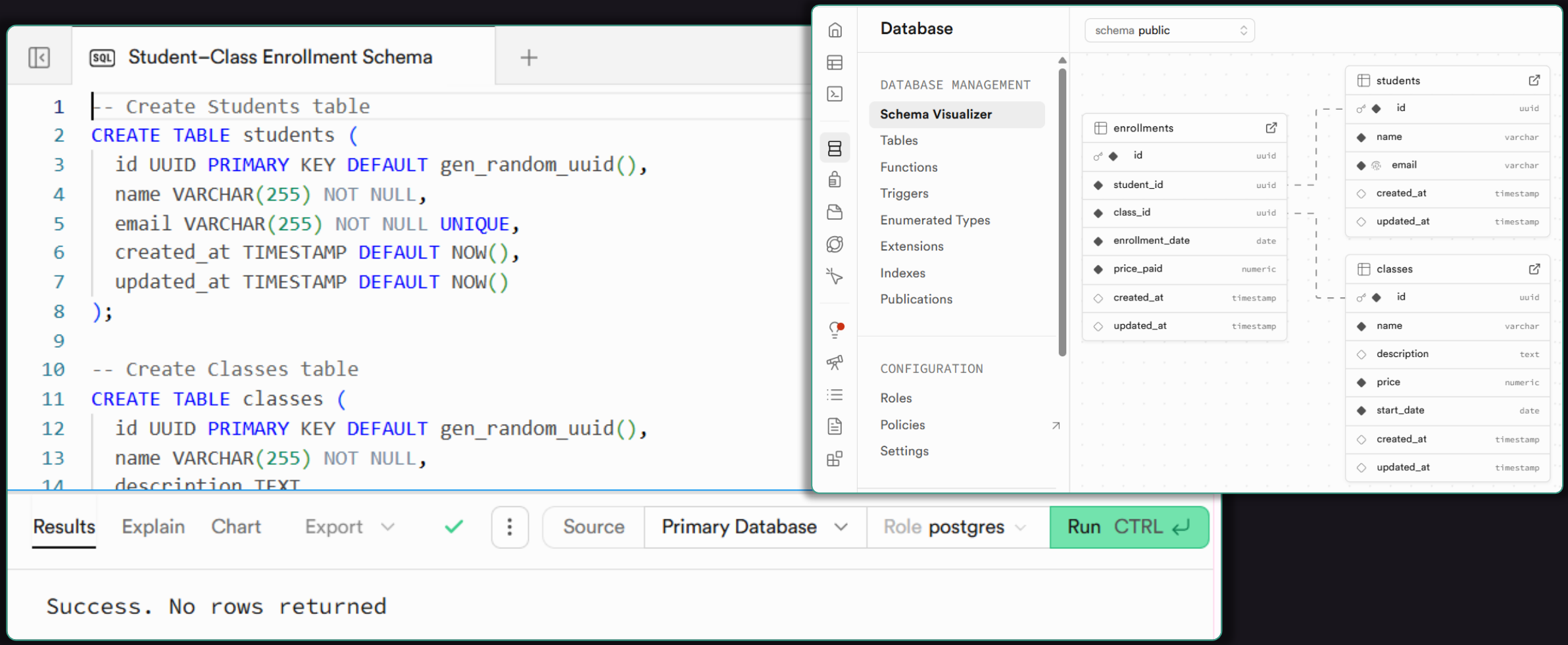
schema.sql

```sql
-- Create Students table
CREATE TABLE students (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    name VARCHAR(255) NOT NULL,
    email VARCHAR(255) NOT NULL UNIQUE,
    created_at TIMESTAMP DEFAULT NOW(),
    updated_at TIMESTAMP DEFAULT NOW()
);

-- Create Classes table
CREATE TABLE classes (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    name VARCHAR(255) NOT NULL,
    description TEXT,
    price DECIMAL(10, 2) NOT NULL,
    start_date DATE NOT NULL,
    created_at TIMESTAMP DEFAULT NOW(),
    updated_at TIMESTAMP DEFAULT NOW()
);
```

Exercise: Create DB Tables in Supabase

SoftUni AI

- **Run the SQL script** to create the tables in Supabase

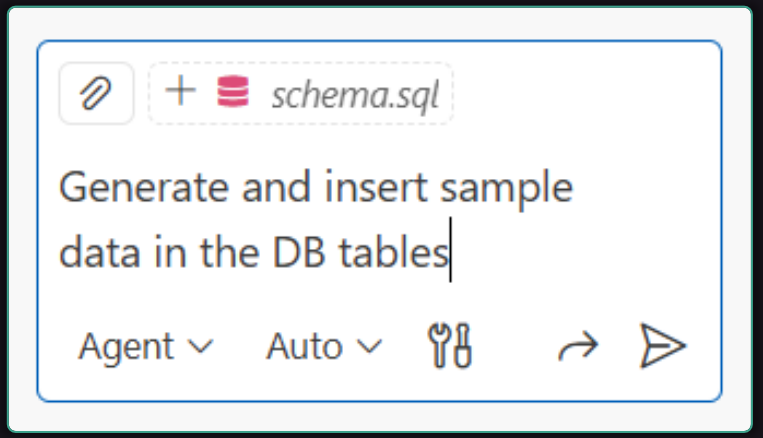


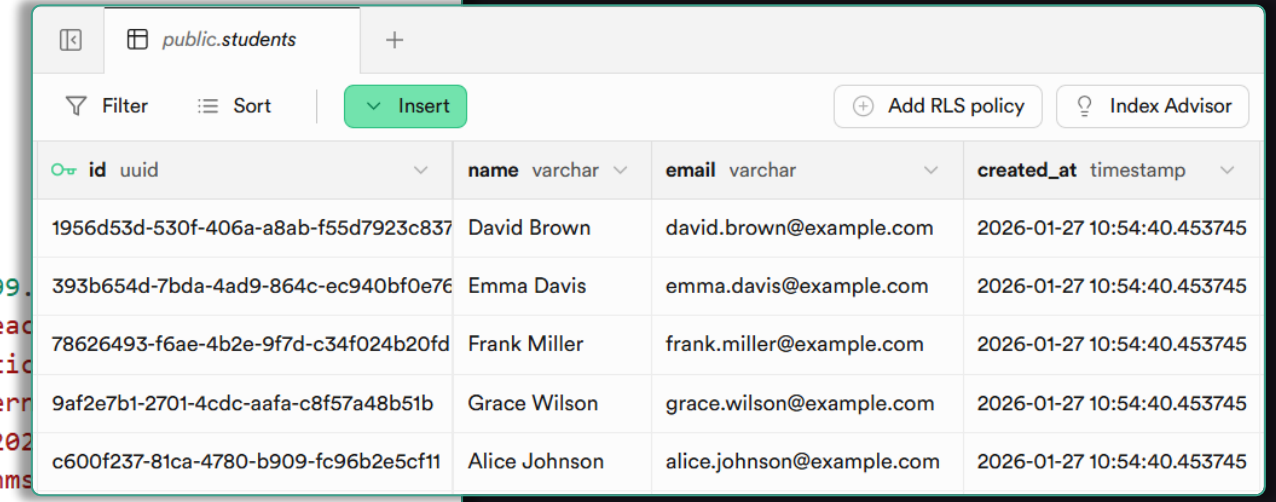
Exercise: Insert Sample Data

SoftUni AI

- Generate and **insert sample data** in the DB tables

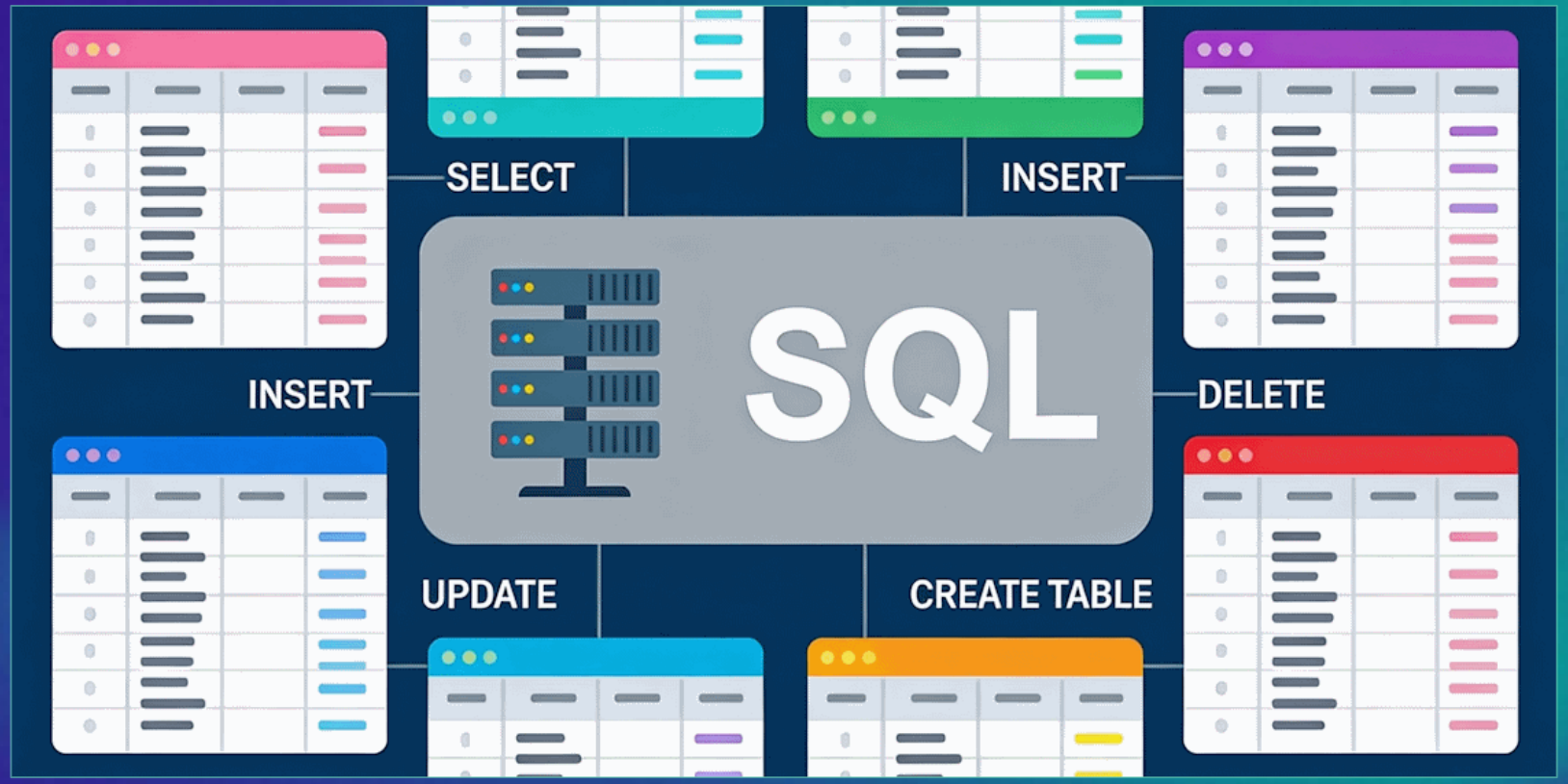
```sql
-- Insert sample students
INSERT INTO students (name, email) VALUES
('Alice Johnson', 'alice.johnson@example.com'),
('Bob Smith', 'bob.smith@example.com'),
('Carol Williams', 'carol.williams@example.com'),
('David Brown', 'david.brown@example.com'),
('Emma Davis', 'emma.davis@example.com'),
('Frank Miller', 'frank.miller@example.com'),
('Grace Wilson', 'grace.wilson@example.com'),
('Henry Moore', 'henry.moore@example.com');

-- Insert sample classes
INSERT INTO classes (name, description, price, start_date) VALUES
('Introduction to Python', 'Learn the basics of Python programming', 99.
('Web Development with React', 'Build modern web applications using Reac
('Data Science Fundamentals', 'Introduction to data science and analytic
('Advanced JavaScript', 'Master advanced JavaScript concepts and patterr
('SQL Database Design', 'Design and optimize SQL databases', 89.99, '202
('Machine Learning Basics', 'Introduction to machine learning algorithms
('Cloud Computing with AWS', 'Deploy applications on AWS', 179.99, '2026-04-01'),
('Mobile App Development', 'Create iOS and Android applications', 199.99, '2026-04-10');
```





SQL Language and DB Schema

Table Structure: CREATE / ALTER / DROP TABLE



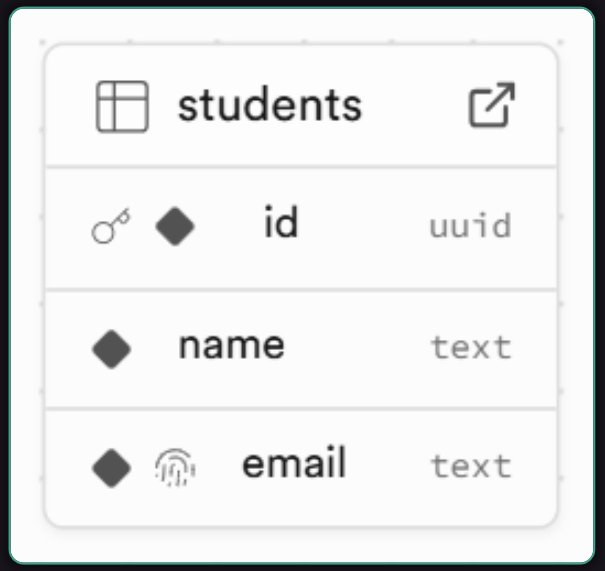
What is SQL?

SoftUni AI

- **SQL** (**S**imple **Q**uery **L**anguage)
  - Database-level language for managing **DB schema** and **table data** in relational databases (RDBMS)

- Sample SQL code:

```
CREATE TABLE students (
  id UUID PRIMARY KEY
    DEFAULT gen_random_uuid(),
  name TEXT NOT NULL,
  email TEXT NOT NULL UNIQUE
)
```



DB Schema Definition with SQL

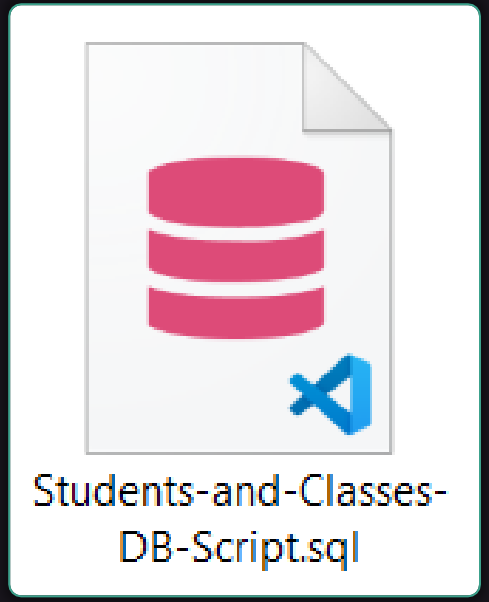
SoftUni AI

- **DDL** (Data Definition Language) in SQL
  - Define the table structure: **CREATE** / **ALTER** / **DROP TABLE**

```
CREATE TABLE enrollments (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  student_id UUID NOT NULL REFERENCES students(id) ON DELETE CASCADE,
  class_id UUID NOT NULL REFERENCES classes(id) ON DELETE CASCADE,
  enrollment_date DATE NOT NULL,
  price_paid DECIMAL(10, 2) NOT NULL,
  created_at TIMESTAMP DEFAULT NOW(),
  updated_at TIMESTAMP DEFAULT NOW(),
  UNIQUE(student_id, class_id) -- Prevent duplicate enrollments
);
```

SQL DB Create Script

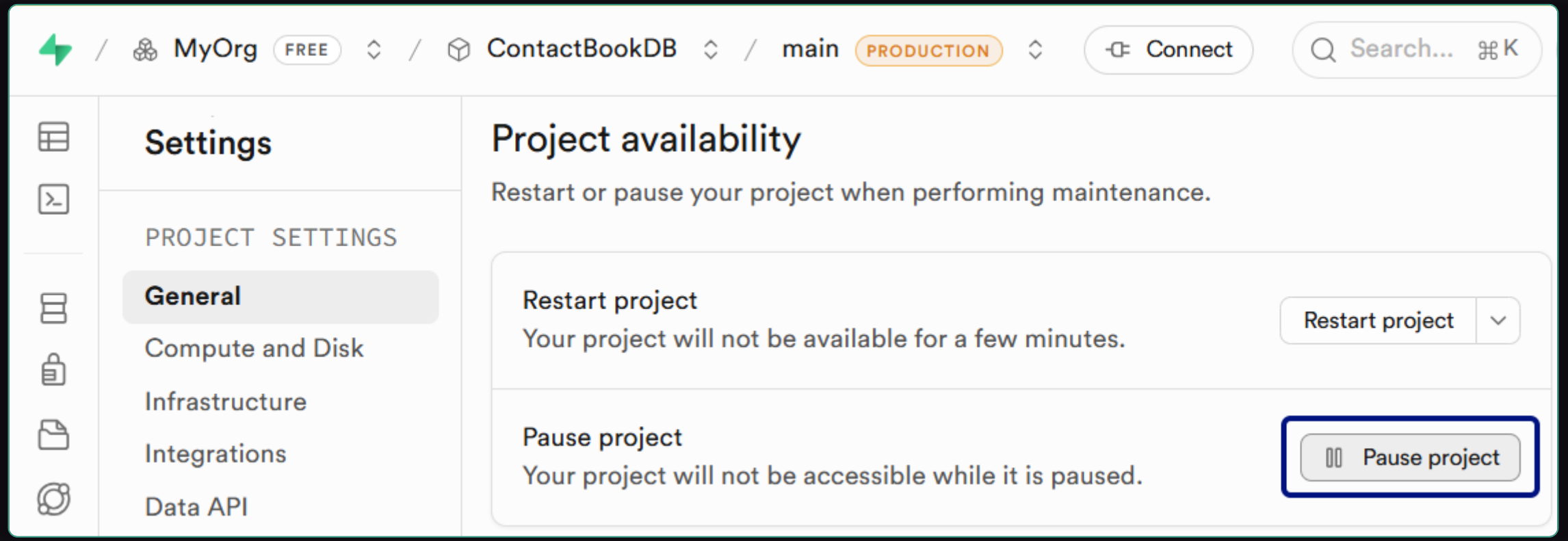
- You can create / backup / restore databases with a **SQL script** (holding schema + data)
- How to **restore a database** from SQL script?
  - Create a **new project** in Supabase
  - **Run the SQL script** (in the SQL Editor)
  - Check for **errors** / fix errors
- Example: run the attached SQL script
  - **Students-and-Classes-DB-Script.sql**

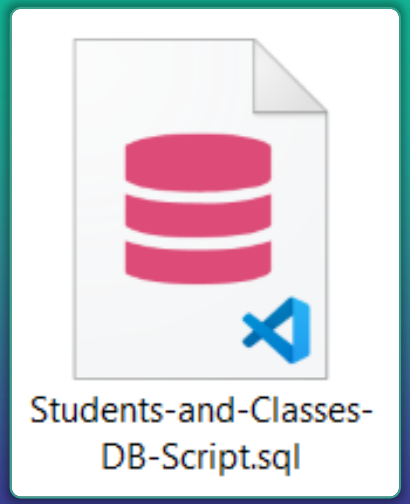


Pause Unused Projects in Supabase

SoftUni AI

- The **free** tier in Supabase allows **up to 2 active projects**
- You can **pause unused projects**, to allow new projects



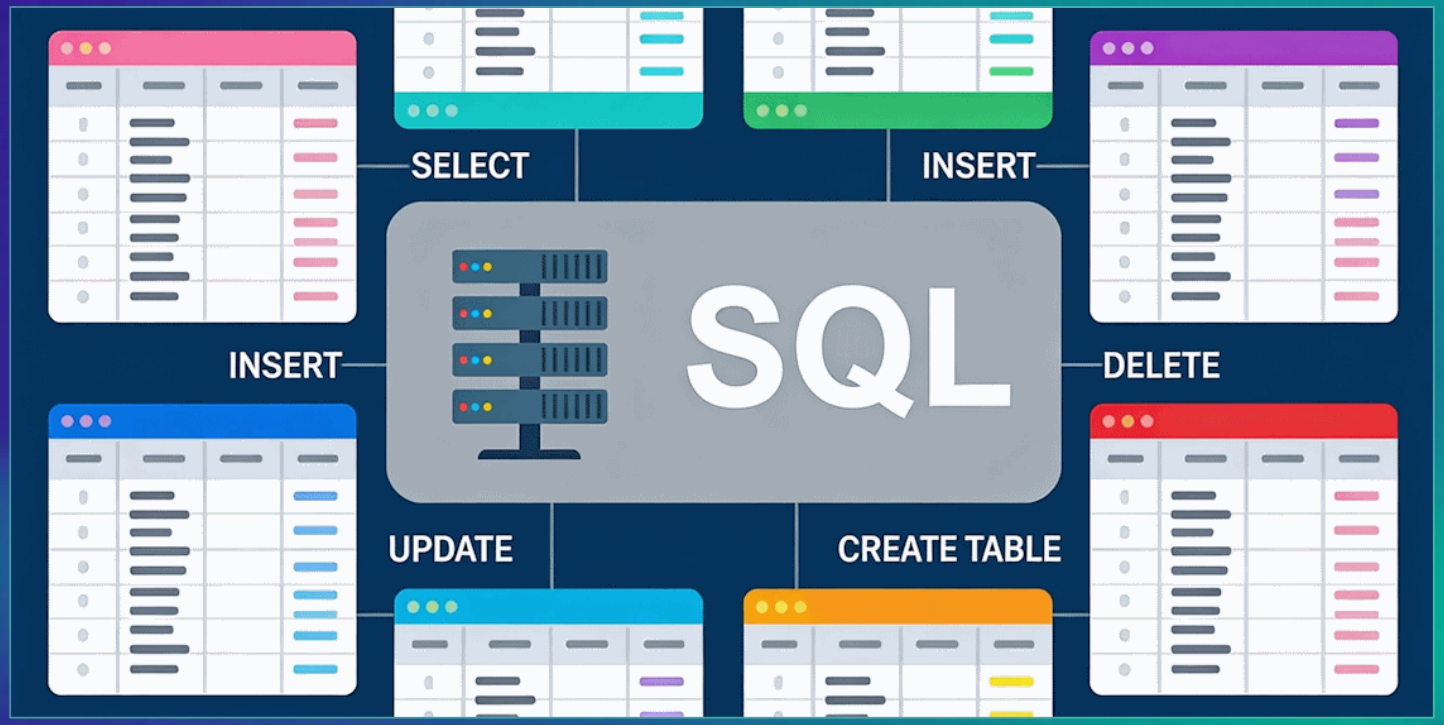


Create a DB from SQL Script

Live Demo

SQL Query and Edit Data

CRUD Operations, SELECT, INSERT, UPDATE, DELETE, Filtering, Joins, Aggregations, Grouping



CRUD Operations with SQL

SoftUni AI

- **DML** (Data Manipulation Language)
  - CRUD Operations: **SELECT**, **INSERT**, **UPDATE**, **DELETE**
- SQL **queries**:
  - **SELECT** with **WHERE**, **ORDER BY**, **JOIN**, functions
  - **Joins**, **Grouping** and **Aggregations**
- Sample SQL commands:

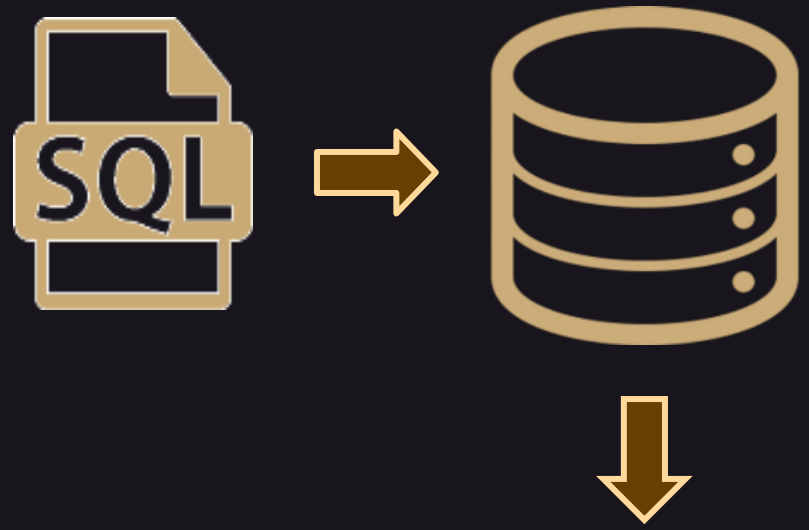
```
SELECT name, price FROM classes
```

```
UPDATE classes SET price = price * 1.2
```

SQL Query Execution

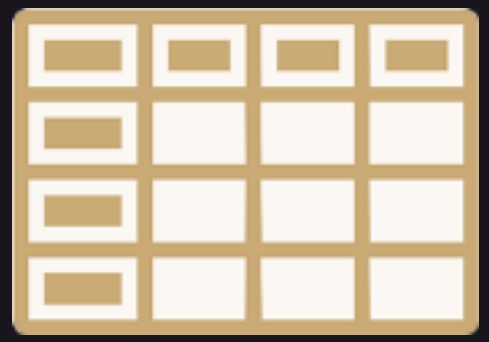
- Sample SQL query:

```
SELECT * FROM students
```



- The query is **executed** in the DB
  - It returns a sequence of **data rows**

id	name	email	created_at
cd9ca4f4-7e24-4c34-8371-63430927d34e	Alice Johnson	alice.johnson@example.com	2026-01-27 11:27:40.746427
15b550ef-0537-4209-9fdb-688a1e007382	Bob Smith	bob.smith@example.com	2026-01-27 11:27:40.746427
89fb4905-876b-4d81-9e69-8af2977cd306	Carol Williams	carol.williams@example.com	2026-01-27 11:27:40.746427
6bb294b4-ac61-4e30-a518-1ffcd13b1558	David Brown	david.brown@example.com	2026-01-27 11:27:40.746427
3cacfd4f-8b03-404a-ba6c-13a914a3e174	Emma Davis	emma.davis@example.com	2026-01-27 11:27:40.746427
b7ad8fcb-1208-4a8d-8957-b917021b59e4	Frank Miller	frank.miller@example.com	2026-01-27 11:27:40.746427



SQL INSERT, UPDATE, DELETE

SoftUni AI

- **Insert** a data row in a DB table:

```
INSERT INTO students(name, email)
VALUES ('Steve New', 'steve.new@gmail.com')
```

- **Update** data rows in a DB table:

```
UPDATE students SET name = name || ' (old)'
WHERE email='steve.new@gmail.com'
```

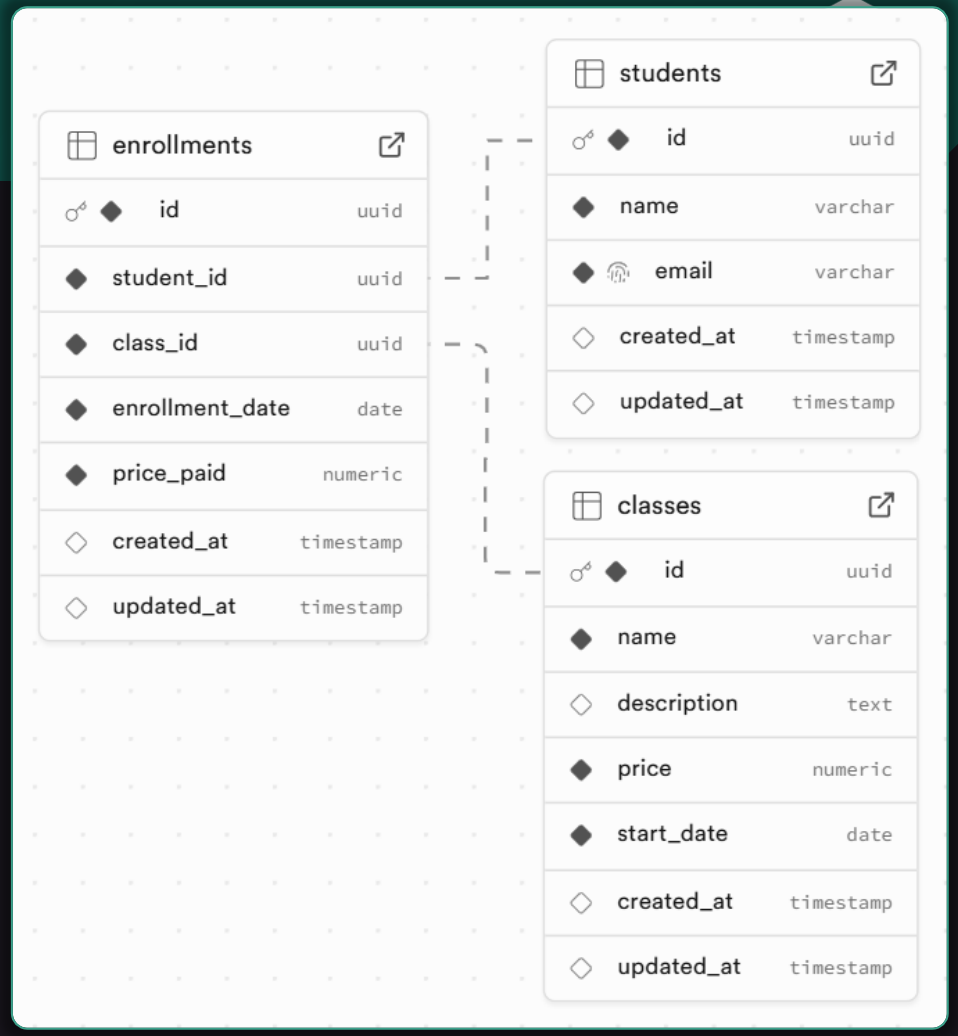
- **Delete** data rows from a DB table:

```
DELETE FROM students
WHERE email='steve.new@gmail.com'
```

SQL Queries with JOIN

- **Joining tables** (**JOIN** operator)

```sql
SELECT
    s.name as student_name,
    s.email,
    c.name as class_name,
    e.enrollment_date,
    c.price,
    e.price_paid,
    (c.price - e.price_paid) as discount
FROM enrollments e
JOIN students s ON e.student_id = s.id
JOIN classes c ON e.class_id = c.id
ORDER BY s.name;
```



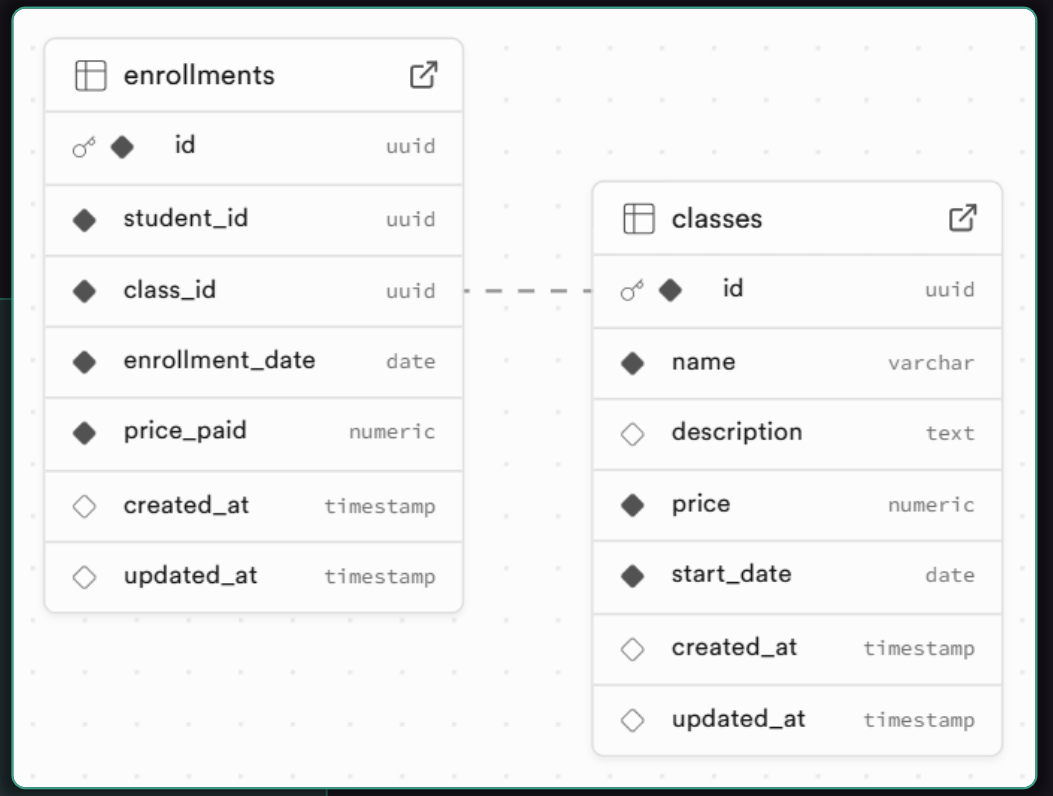
student_name	email	class_name	enrollment_date	price	price_paid	discount
Alice Johnson	alice.johnson@example	Introduction to Python	2026-01-15	119.99	89.99	30.00
Alice Johnson	alice.johnson@example	Web Development with R	2026-01-20	179.99	149.99	30.00
Bob Smith	bob.smith@example.com	Data Science Fundament	2026-01-10	239.99	199.99	40.00
Bob Smith	bob.smith@example.com	SQL Database Design	2026-01-18	107.99	79.99	28.00
Carol Williams	carol.williams@exampl	Advanced JavaScript	2026-01-12	155.99	129.99	26.00

Grouping and Aggregations

SoftUni AI

- **Group functions** and **aggregations** with SQL

```
SELECT
   c.name,
   c.price,
   COUNT(e.id) as enrollments,
   SUM(e.price_paid) as revenue,
   AVG(e.price_paid) as avg_price
FROM classes c
LEFT JOIN enrollments e
   ON c.id = e.class_id
GROUP BY c.name, c.price;
```



name	price	enrollments	revenue	avg_price
Web Development with React	179.99	2	289.98	144.99
Cloud Computing with AWS	215.99	2	349.98	174.99
Machine Learning Basics	299.99	1	249.99	249.99

Access Control with SQL

SoftUni AI

- **DCL** (Data Control Language)
  - Database users, roles and permissions: **GRANT** / **REVOKE**
- **RLS** (Row-level security)
  - **CREATE POLICY**, **ALTER POLICY**, **DROP POLICY**

```
CREATE POLICY "Users can view their own exams"
ON exams FOR SELECT
USING (
  (SELECT auth.uid()) = user_id
);
```

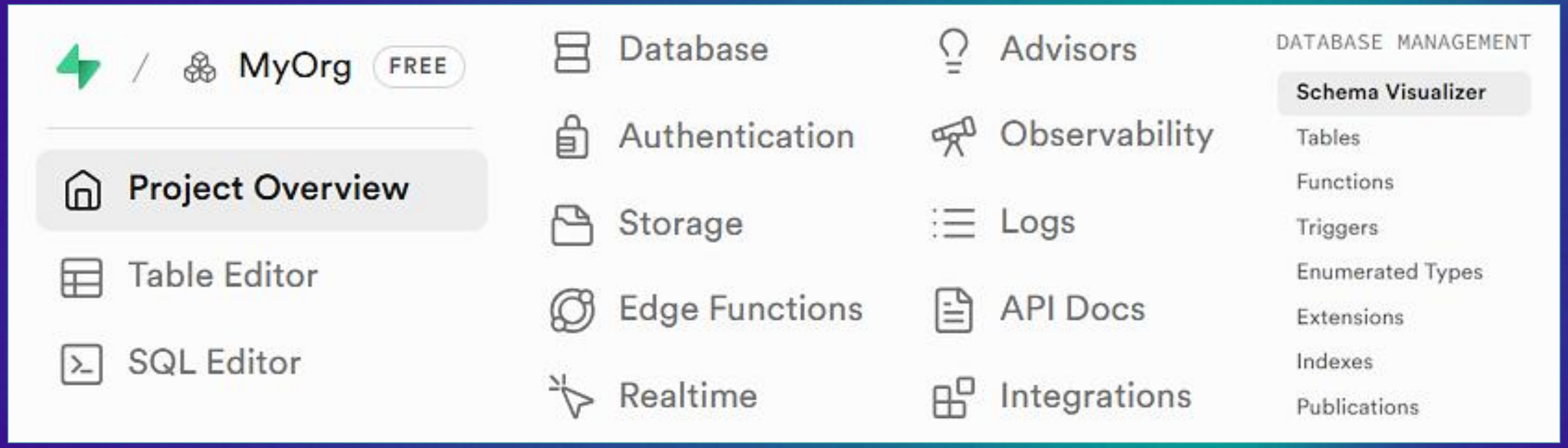
Exercise: SQL Commands with AI

SoftUni AI

- Given the database "**Students, Classes and Enrollments**", perform the following tasks using **AI assistance**:
  - **Add** a new class (SQL **INSERT**)
  - **Enroll** several students for it (SQL **INSERT**)
  - **Increase the price** for existing class (SQL **UPDATE**)
  - **Delete** a student with its enrollments (SQL **DELETE**)
  - Print all **students**, their # of **enrollments** and **total fees paid**
  - Find a list of **all enrollments** in the **latest 2 courses**
  - Find all **enrollments** with **revenue by weeks** (W1-2026, W2-2026, …)

Supabase Services

Supabase Services (DB, Auth, Storage, Edge), Logs, Backup, API Keys, Row-Level Security



Supabase Services

SoftUni AI

- **Supabase** is a **backend** platform (**BaaS**)
  - Provides many **services** for devs, not just a database

- **Supabase DB** (tables and SQL)
  - **PostgreSQL** (cloud-based relational database instance)
  - Tables, relationships, views, enums, indexes, SQL, stored procedures / functions, triggers, row-level security (RLS)

- **REST & GraphQL APIs** for developers
  - JS apps connect to Supabase and DB via these APIs
  - Auth, CRUD operations, query, filter, sort, paginate, etc.

Supabase Services (2)

SoftUni AI

- **Supabase Auth** (users and access control)
  - **Authentication** and **authorization** system (access control), integrated with the database with **RLS** at database level
  - Built-in **user management**: sign-up, sign-in, sessions (with JWT tokens), OAuth (external login, e. g. Google login)
  - Auth methods: email + password, OAuth, magic links
- **Supabase Storage** (images, files, documents, videos)

  - **Buckets** (root folders), **files** (with file paths, holding **folders**)
  - File **upload**, **download**, **share**, access control (with RLS)

Supabase Services (3)

SoftUni AI

- **Supabase Edge Functions** (serverless)
  - **Serverless functions**, acting as a lightweight back-end
  - Implemented in **JavaScript**, accessed through the API
  - Implement **payments**, **emails**, **webhooks** (Web callbacks)
  - Have **unrestricted DB access**, skipping the RLS policies

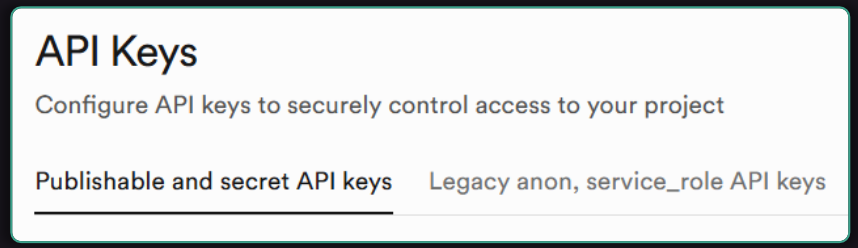
- **Supabase Realtime** (live stream DB changes)
  - DB emits **events** (change notifications) to **channels**, clients subscribe to channels, events are filtered by RLS

- **Supabase Logs** (track DB access, auth, storage, etc.)

API Keys in Supabase

SoftUni AI

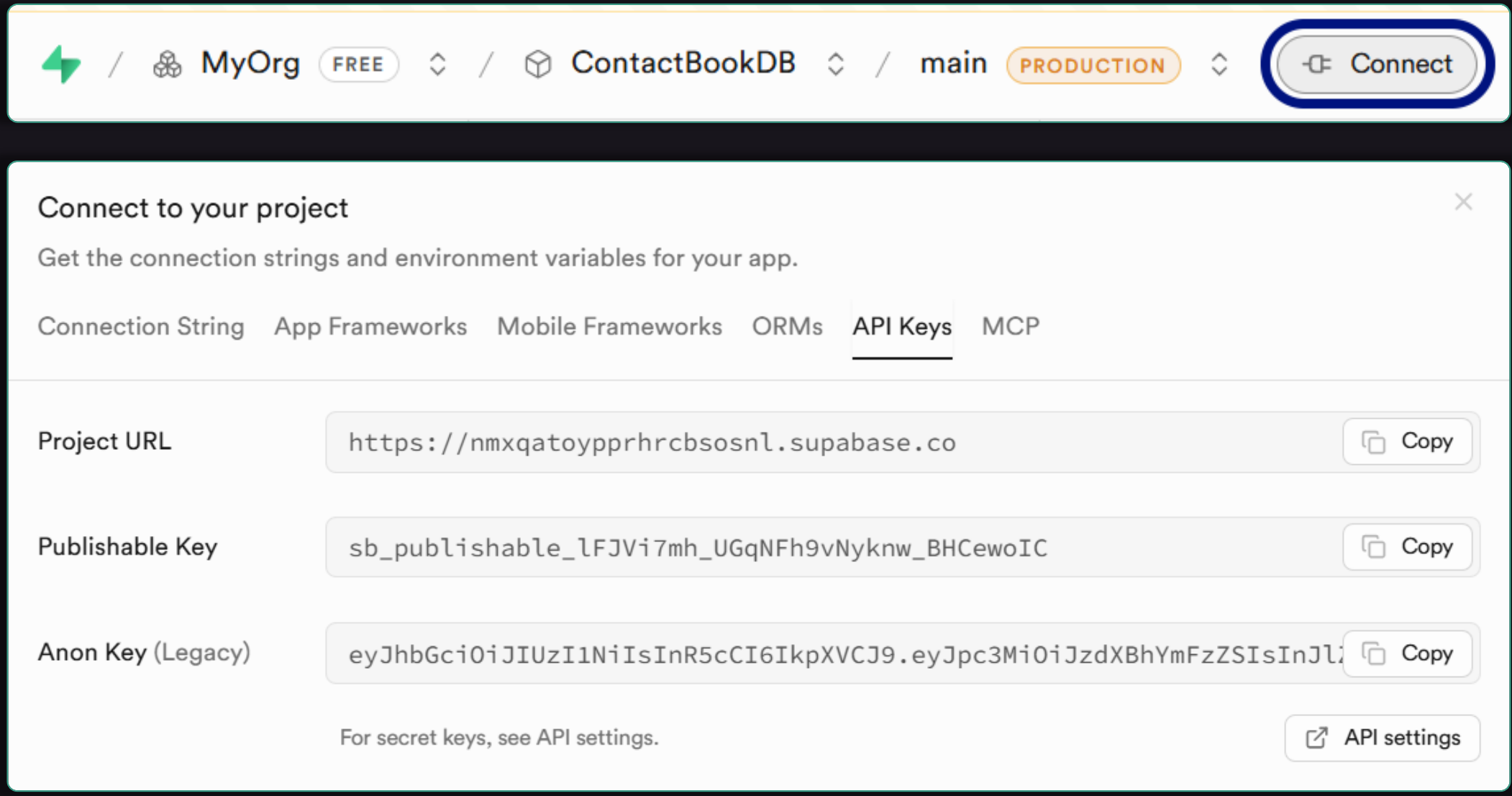
- **API keys** in Supabase connect your app with your Supabase project



- ✅ **Publishable (anon) key**
  - Restricted access key
  - For browsers / mobile apps
  - Safe to **expose publicly**
  - **Access control** based on RLS (Row-Level Security)
- ⚠️ **Secret (service) key**
  - Full access key
  - Used in back-ends
  - **Never expose** publicly
  - Bypassed all access control policies (RLS)

API Keys in Supabase – Example

SoftUni AI



Row-Level Security

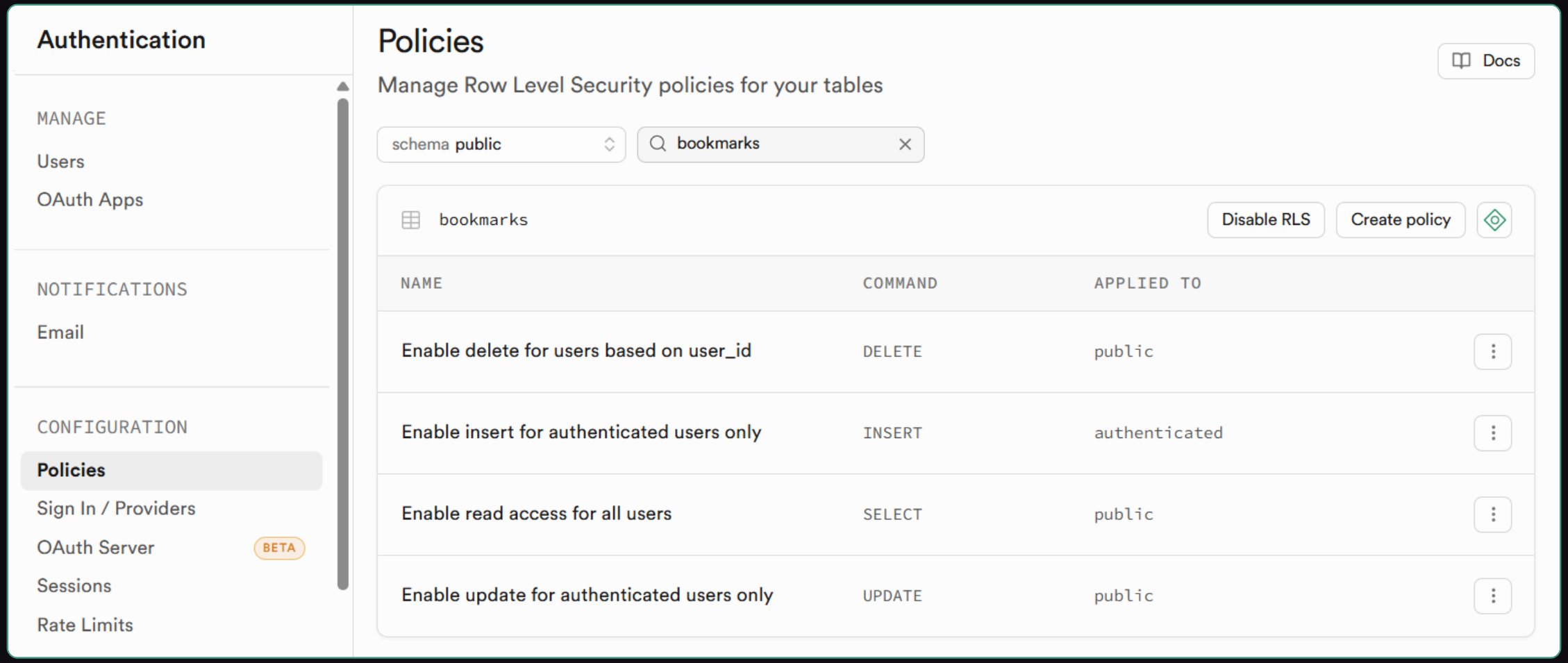
SoftUni AI

- **Row-Level Security** (RLS) define which rows a user can read or write (access control)
  - Defined by **security policies** in the database (SQL code)
  - Based on the **logged-in user**: anonymous / authenticated
- Typical RLS policy:
  - **Anonymous** users: can **read all public data**, e. g. products
  - **Authenticated users**: can read / write their **own data**, e. g. own products, own messages, own user profile
  - **Admin users**: can read / write **any data** in the DB

Sample RLS Policies

SoftUni AI

- Sample **RLS policies** in Supabase:



Supabase Pricing and Free Tier

- Supabase has **free** and **paid** subscription plans
- Supabase **free plan** (for hobby projects):
  - Max **2 active projects**, with minimal computing resources
  - Projects automatically **pause** after ~1 week of inactivity
  - Manual backup (not easy)
- Supabase **paid plans** (for real projects):
  - $25 / month for 1 project (more projects: $10+ / month)
  - Supabase pricing: https://supabase.com/pricing

Supabase Self-Hosting

SoftUni AI

- Supabase can be **self-hosted** on a VPS server
  - It's an **open-source project** → you can deploy your own Supabase instance for free
  - **Technical skills** are needed (Docker and Docker Compose)
  - **Highly reduced costs**, but manual setup + maintenance
  - You can find a VPS provider here: https://www.vpsbenchmarks.com/best_vps/2026
  - Or use template-based deployment, e.g. https://hostinger.com/vps/supabase-hosting

JS Apps with Supabase

Connecting JS Apps with Supabase: API Keys, Supabase API, CRUD Operations

Building an App with Database

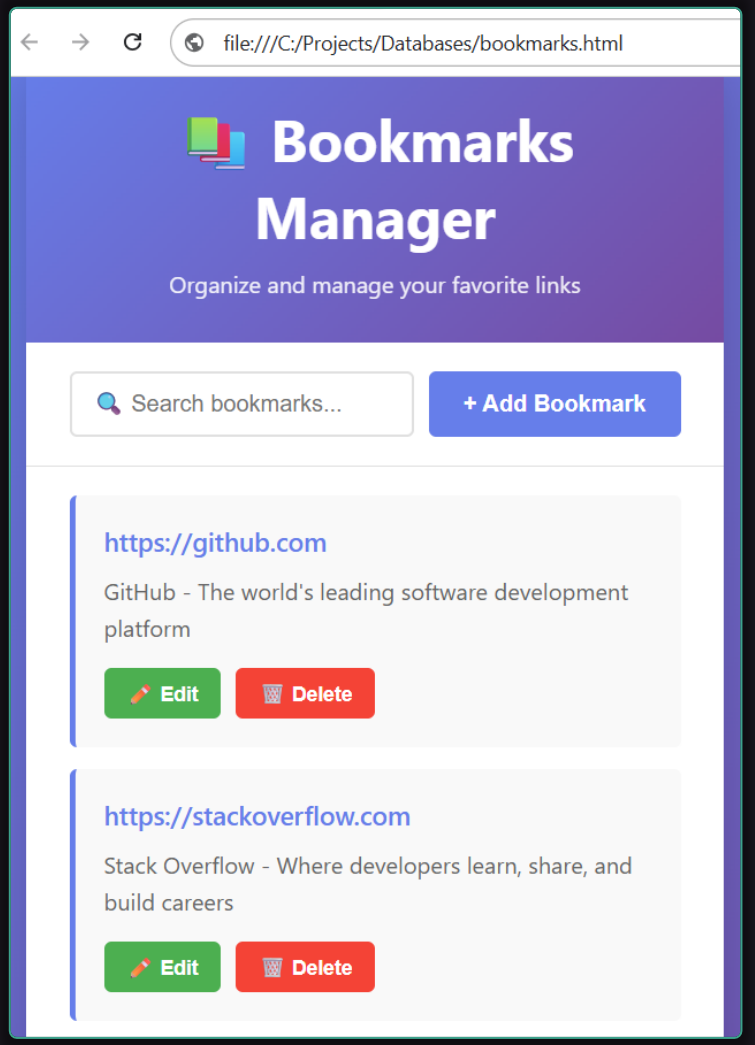
SoftUni AI

- Steps to build an **app with a database** (DB):
  1. Build the **app functionality** without a database
     - Initially keep all app data in the memory
  2. Connect a database & **configure a DB connection**
     - Choose an existing DB / create a new DB
  3. Update the app code to **store app data in the DB**
     - Sample prompt: "*Update the app to store its data in DB*"
  4. **Run** and **test** the app DB operations
     - Check if app data is stored correctly in the DB tables

Example: Bookmarks without a DB

SoftUni AI

- Step 1: Create the **app without DB**
  - Create a **Bookmarks app**: view list of bookmarks, search, create, edit, delete
    - Each bookmark has: **URL**, **comments**
  - Use **popup dialogs** to add / edit / confirm delete bookmarks
  - Keep app data in the **local storage**
    - Initially add a few sample bookmarks
  - Use simple **HTML** + **CSS** + **JS**



Example: Bookmarks with DB

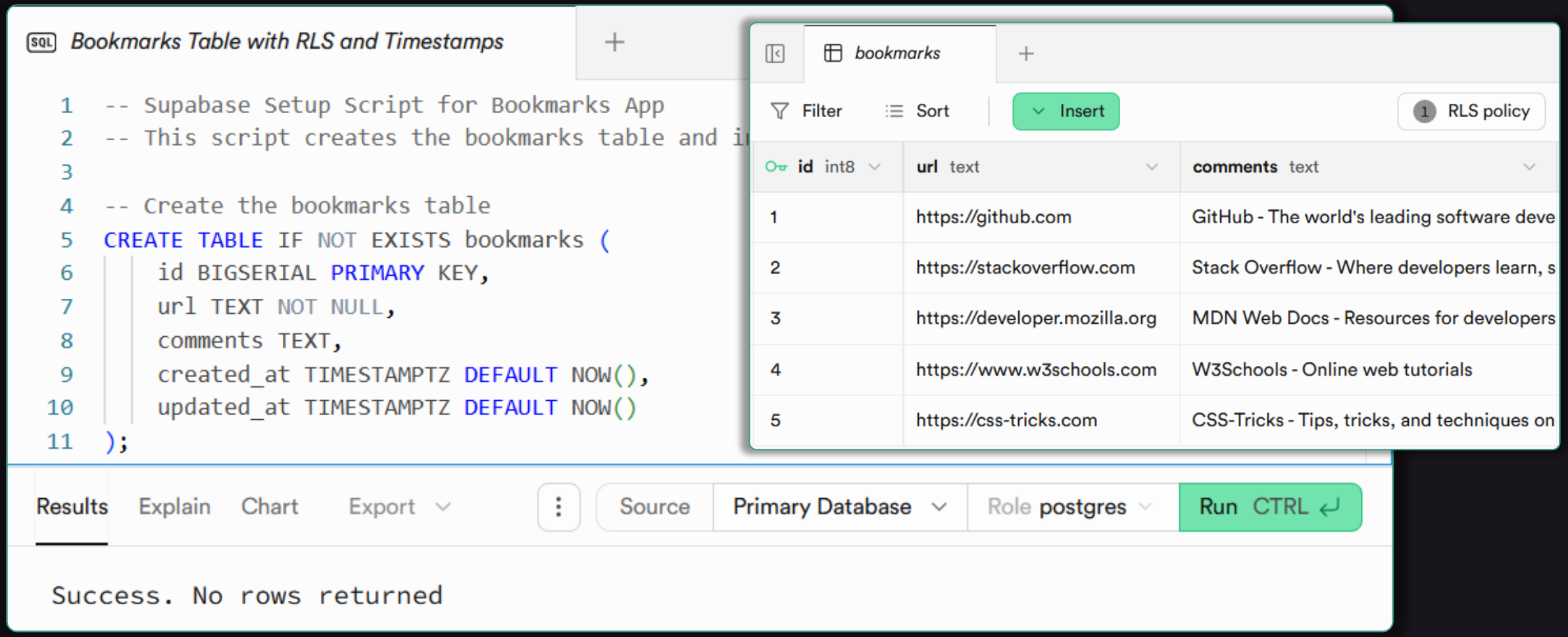
SoftUni AI

- Step 2: **Connect to Supabase**
  - Update the app to **store its data in Supabase**
  - Create Supabase **SQL script** to define the DB tables
    - Initially, insert a few **sample data** in the DB
    - Allow everyone to freely read / modify data **without auth**
    - **Note**: generally, this is **unsafe** (we'll add auth later)
  - Connect the **Supabase client library** with **API key**
  - Implement a **"loading" spinner** at the top of the screen, displayed for DB waits + **toast notifications**

Example: Bookmarks SQL Script

SoftUni AI

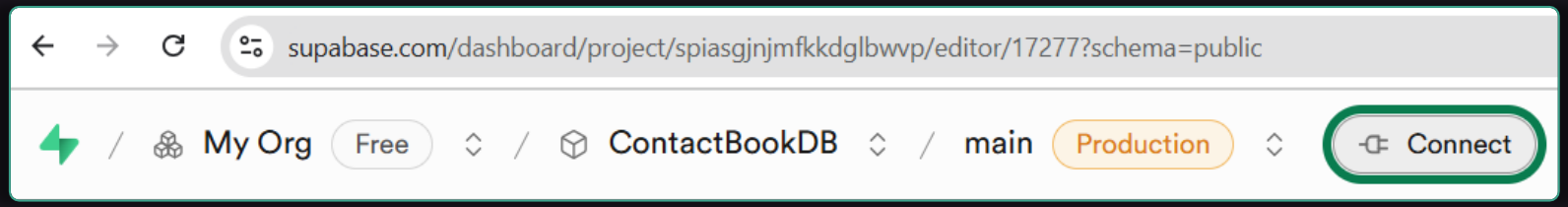
- Step 3. **Run the SQL script** to create the tables in Supabase

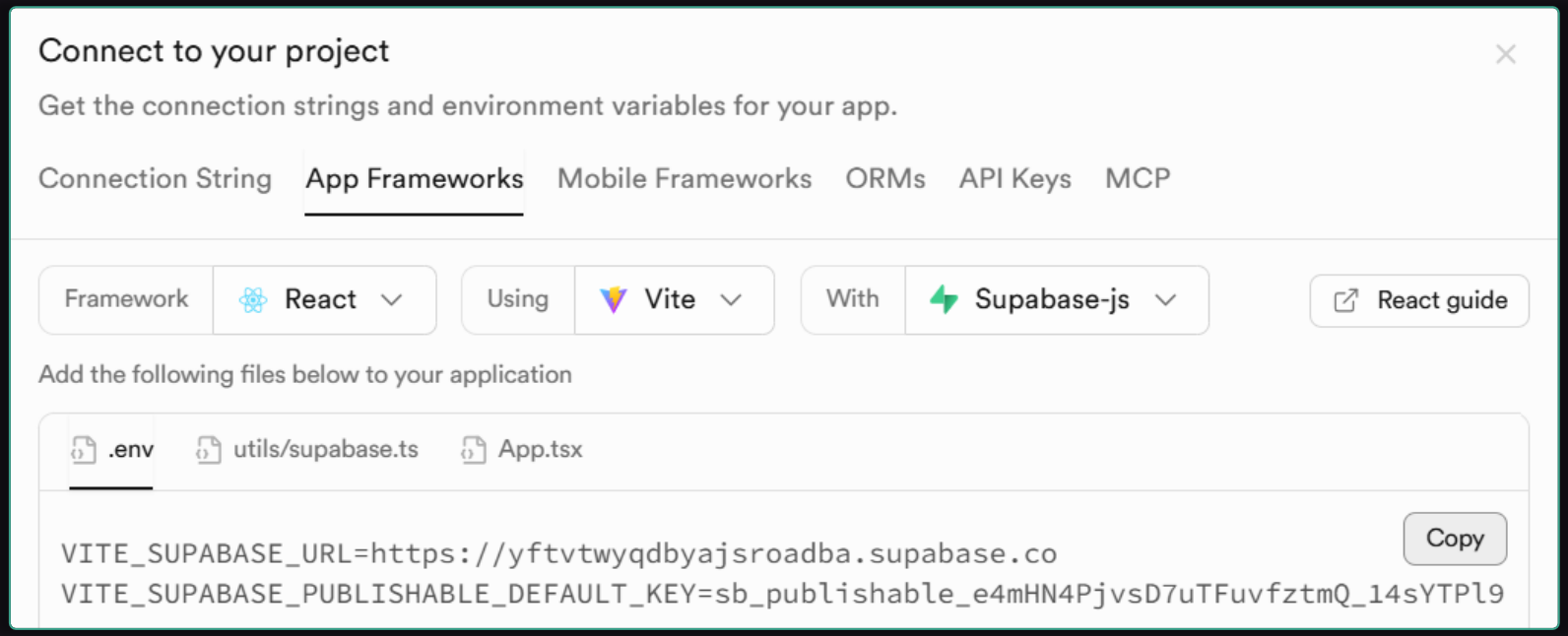


Connecting Apps to Supabase

SoftUni AI

- To connect to Supabase BaaS project, you need an **API key**:
  - **Server URL**
  - **Publishable key**
    (anonymous API key)
- Dashboard
  → Connect
  → App Frameworks
  → React
  → Vite





Example: Supabase API Key

SoftUni AI

- Step 4: **Update Supabase connection settings**
  - Update the Supabase **URL** + **anon key** in the JS code

JS bookmarks.js > SUPABASE_URL

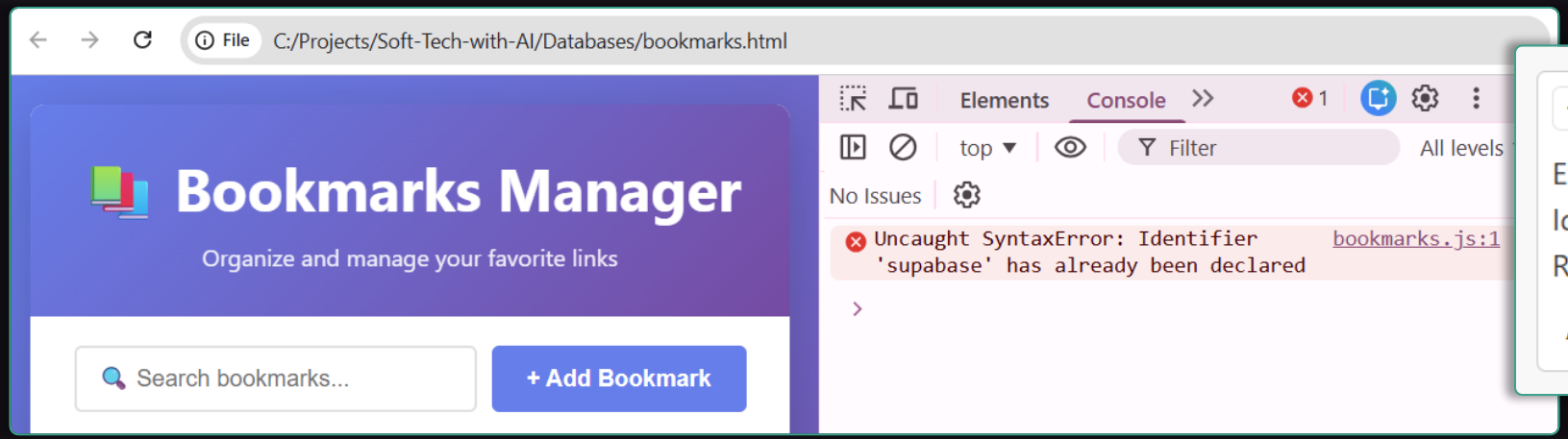
```
1   // ============================================
2   // SUPABASE CONFIGURATION
3   // ============================================
4   // TODO: Replace these with your actual Supabase project credentials
5   // Find these in your Supabase project settings -> API
6   const SUPABASE_URL = 'YOUR_SUPABASE_URL_HERE';
7   const SUPABASE_ANON_KEY = 'YOUR_SUPABASE_ANON_KEY_HERE';
```

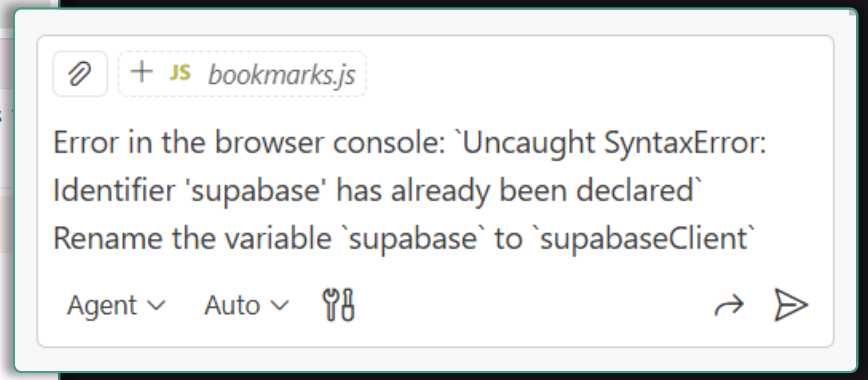
```
6   const SUPABASE_URL = 'https://yftvtwyqdbyajsroadba.supabase.co';
7   const SUPABASE_ANON_KEY = 'sb_publishable_e4mHN4PjvsD7uTFuvfztmQ_14sYTPl9';
```

Example: Testing and Bug Fixing

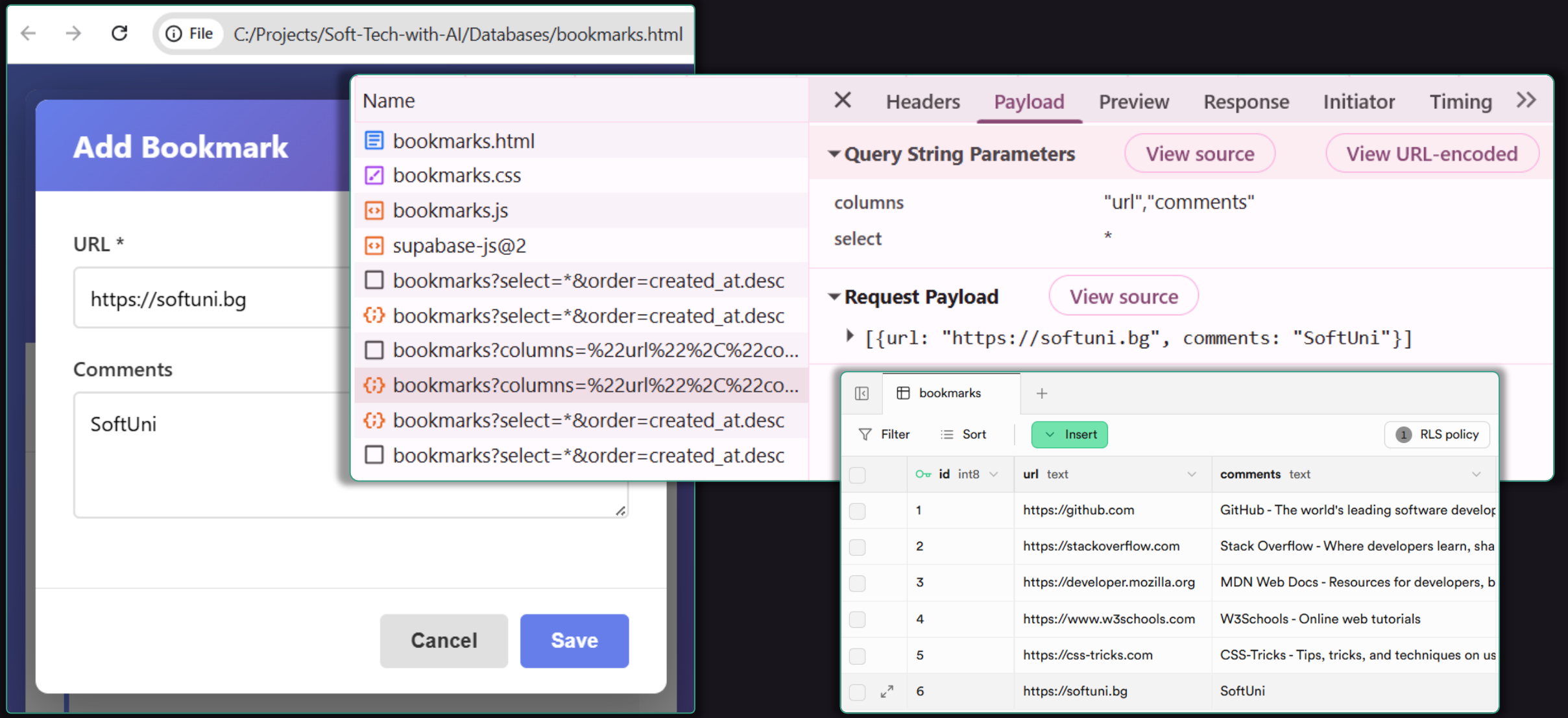
SoftUni AI

- Step 5. **Testing and bug fixing** the Supabase integration
  - Sometimes AI writes wrong JS code to connect to Supabase
  - Error in the browser console: `*Uncaught SyntaxError: Identifier 'supabase' has already been declared*`
  - Fix: rename the variable `*supabase*` to `*supabaseClient*`





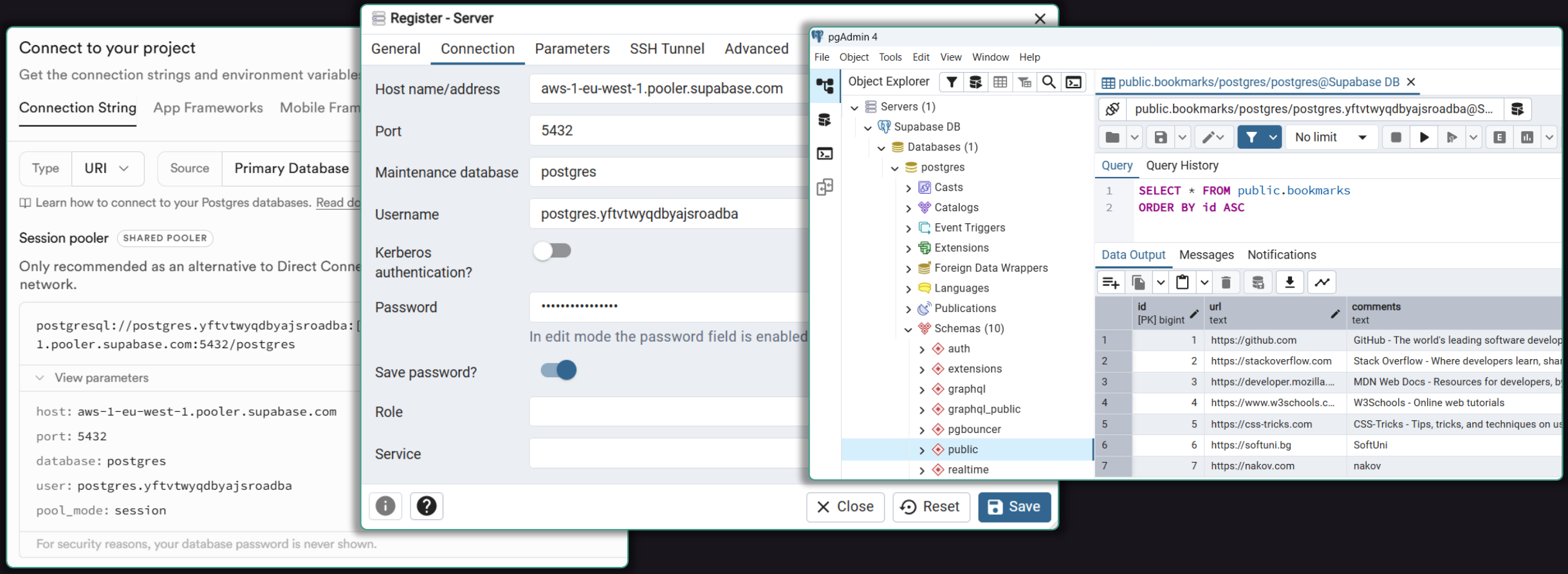
Example: Bookmarks App in Action



Direct Connection to Supabase DB

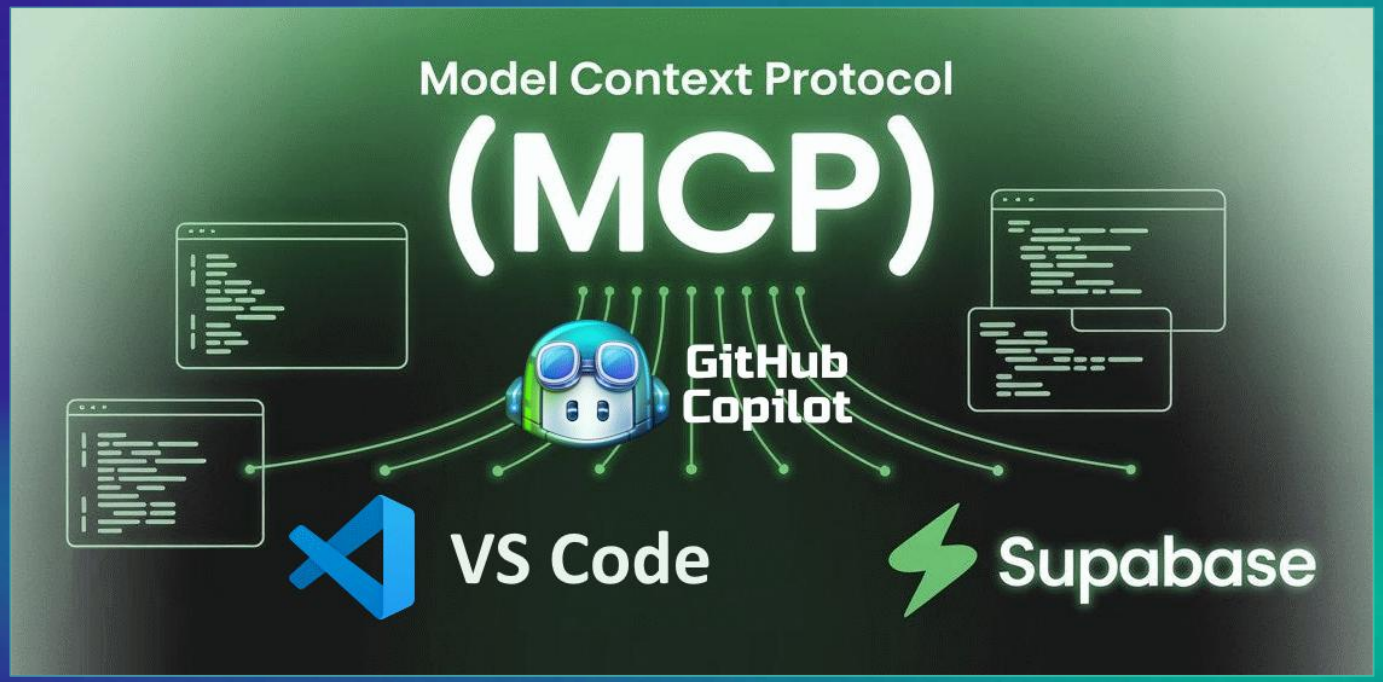
SoftUni AI

- You can connect to your Supabase DB with a standard **PostgreSQL admin tools** like **pgAdmin**, **HeidiSQL**, etc.



Supabase MCP

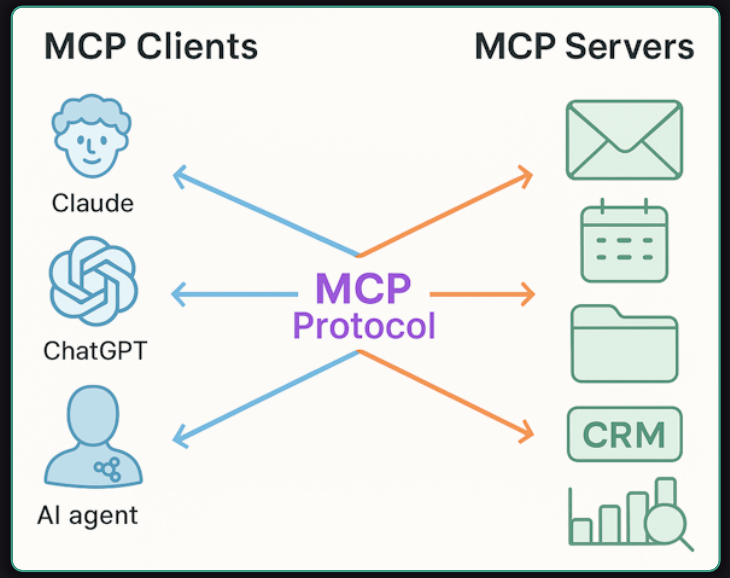
MCP (Model Context Protocol), Install and Configure Supabase MCP for Certain Project



Intro to MCP

SoftUni AI

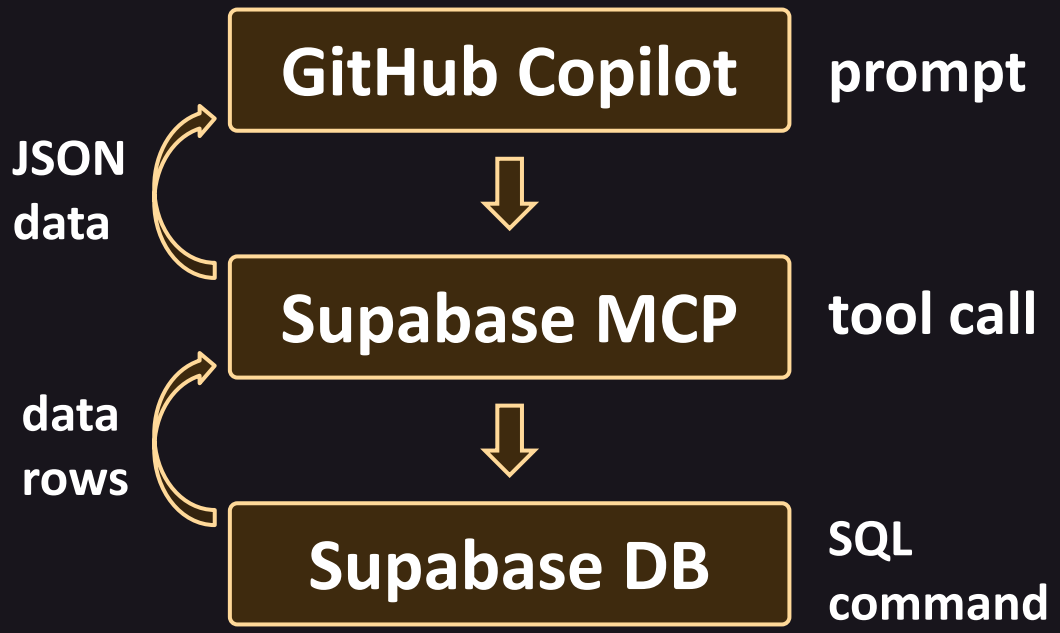
- What is **MCP** (Model Context Protocol)?
  - Open standard for connecting AI models to **external tools**, **data sources** and **APIs**
- Examples of **MCP servers**:
  - Connect Claude Desktop with **Notion**
  - Connect GitHub Copilot with **Supabase** → browse the DB
  - Connect ChatGPT with **Booking.com** → find and book hotels
  - Connect ChatGPT with a **GMail** account → find / send emails
  - Connect an AI agent with your **internal CRM system**

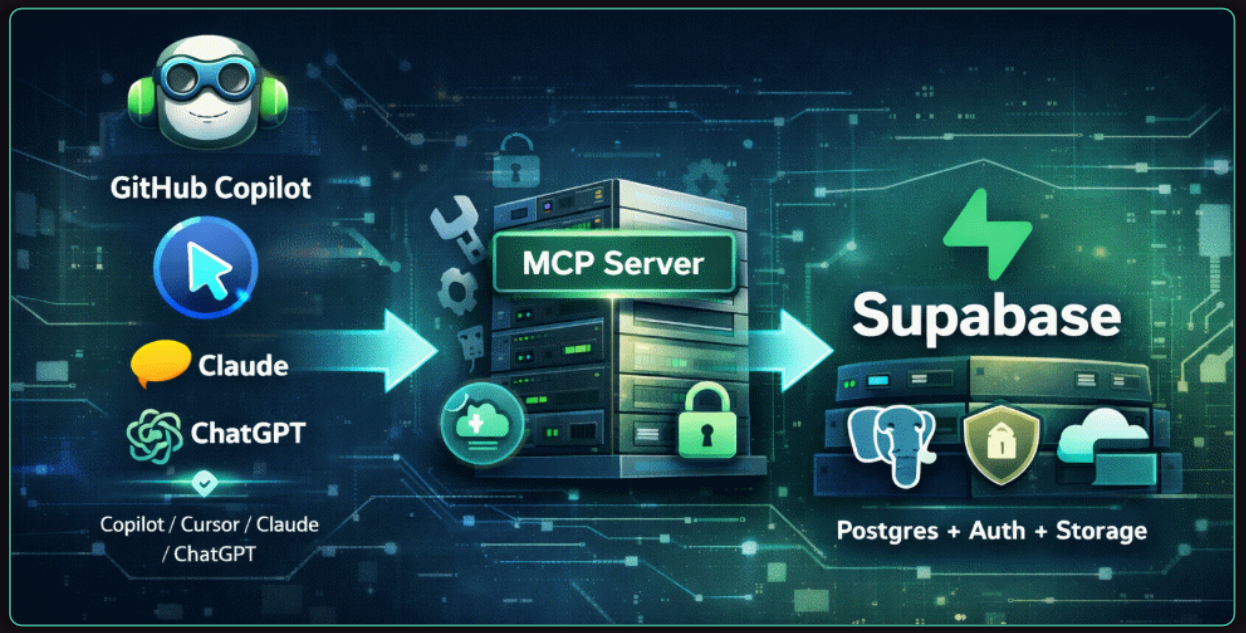


Supabase MCP

SoftUni AI

- **Supabase MCP** connects your **AI Dev Agent** (GitHub Copilot) to your **Supabase** instance via an MCP server



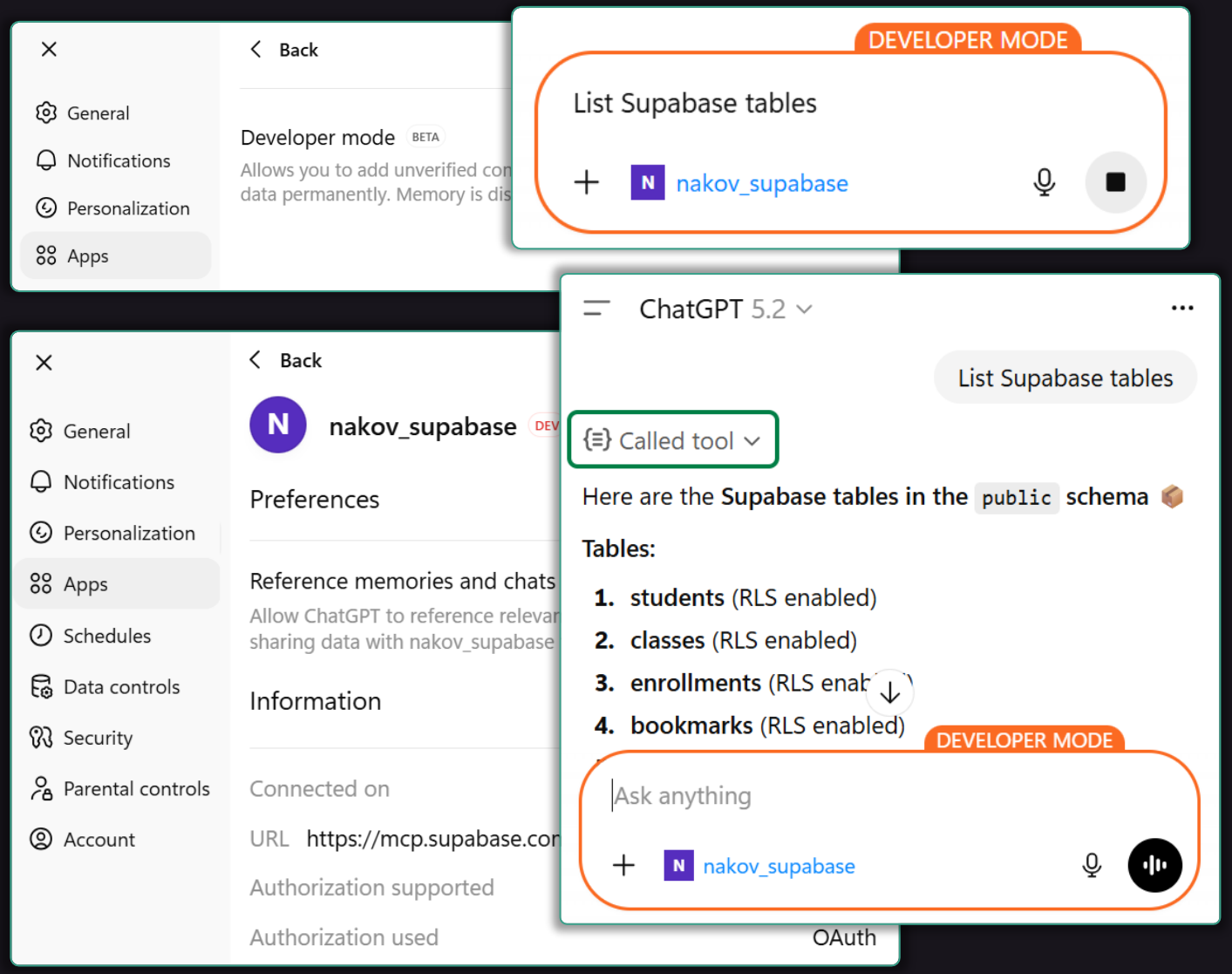


- Supabase provides an **MCP server** for your projects

Supabase MCP in ChatGPT

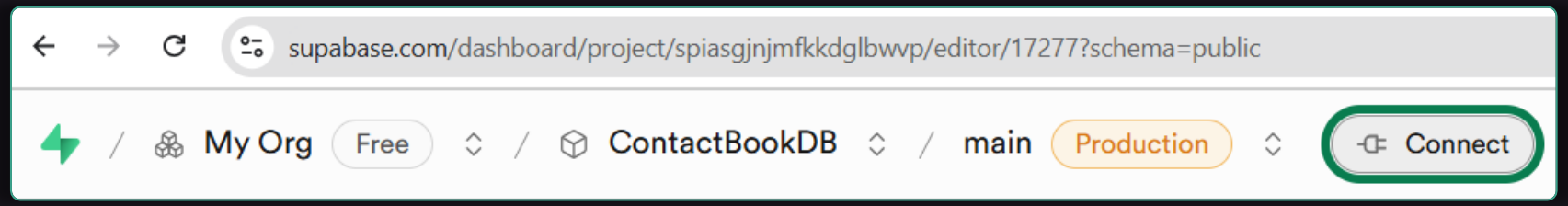
SoftUni AI

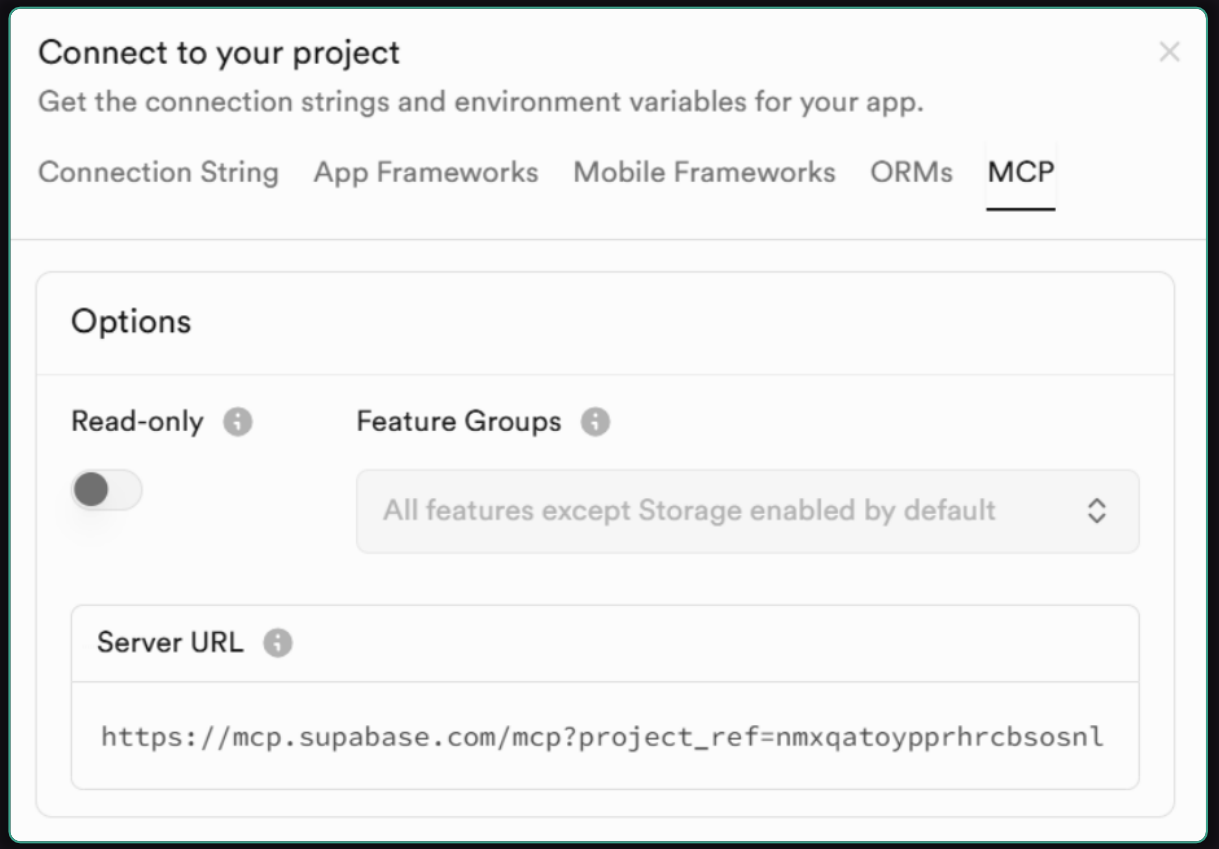
- **ChatGPT** supports the **MCP protocol**
  - Can connect to your Supabase DB
- Step 1: Enable **dev mode**
- Step 2: **Create an app** (MCP configuration)
- Step 3: **AI prompt** with your app

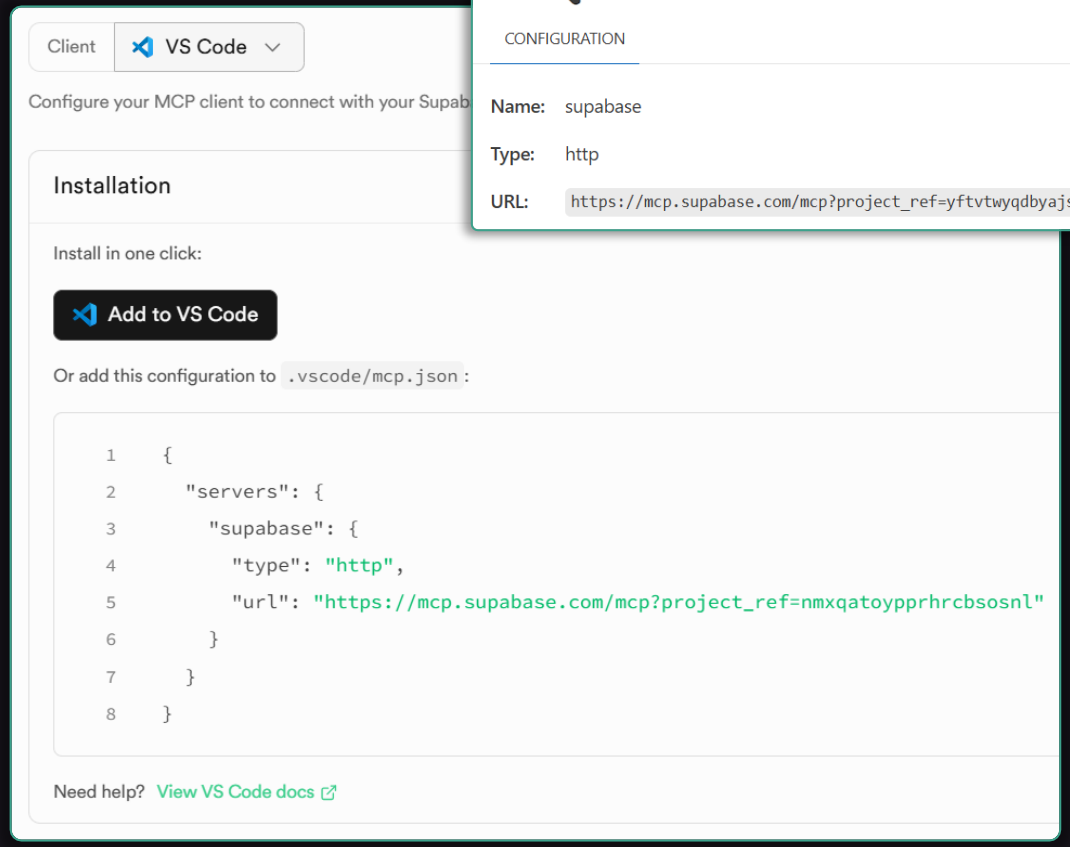


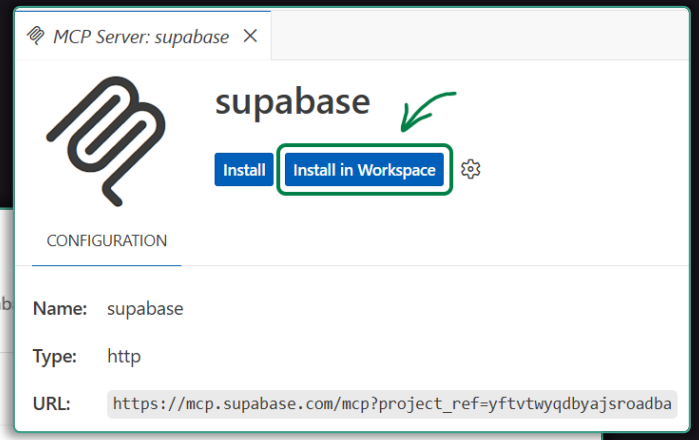
Installing Supabase MCP in VS Code

SoftUni AI





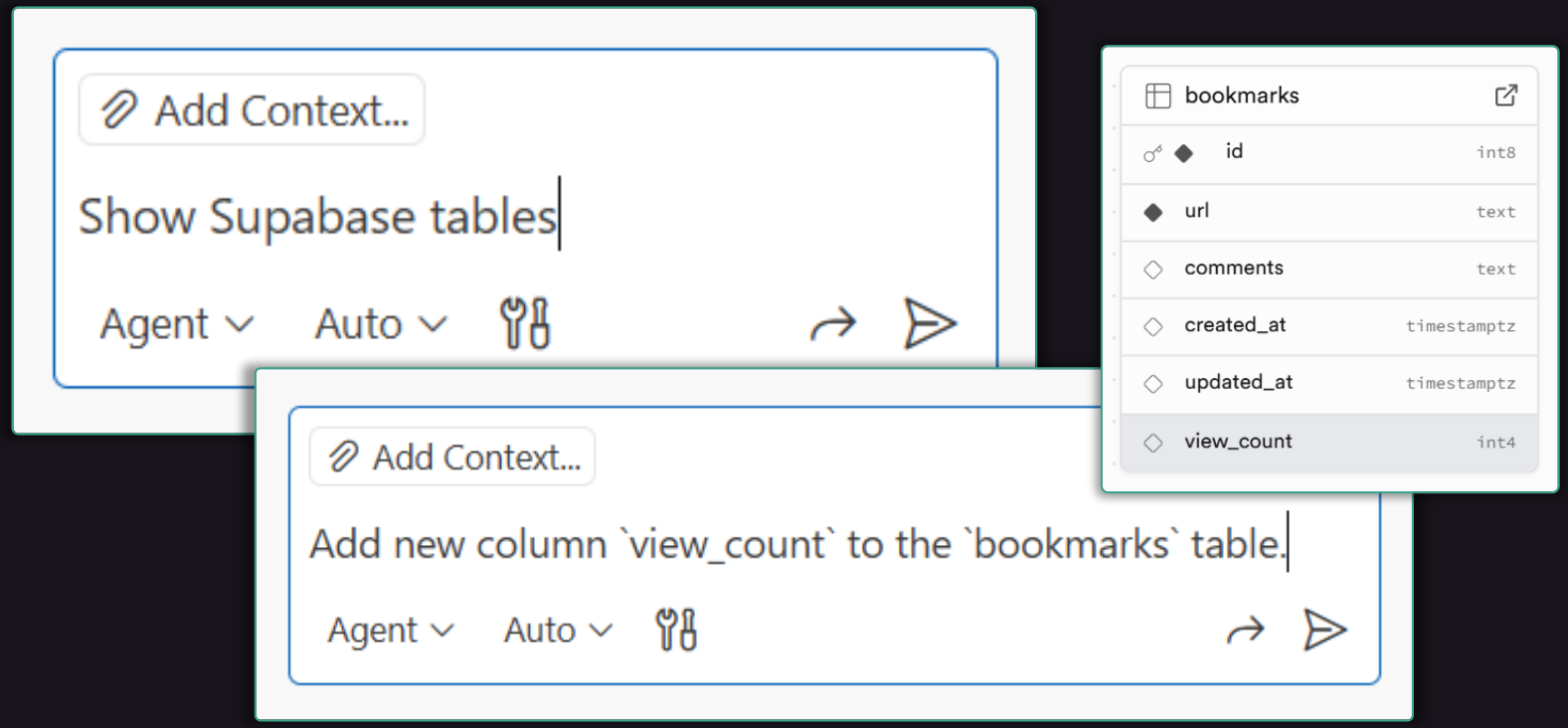


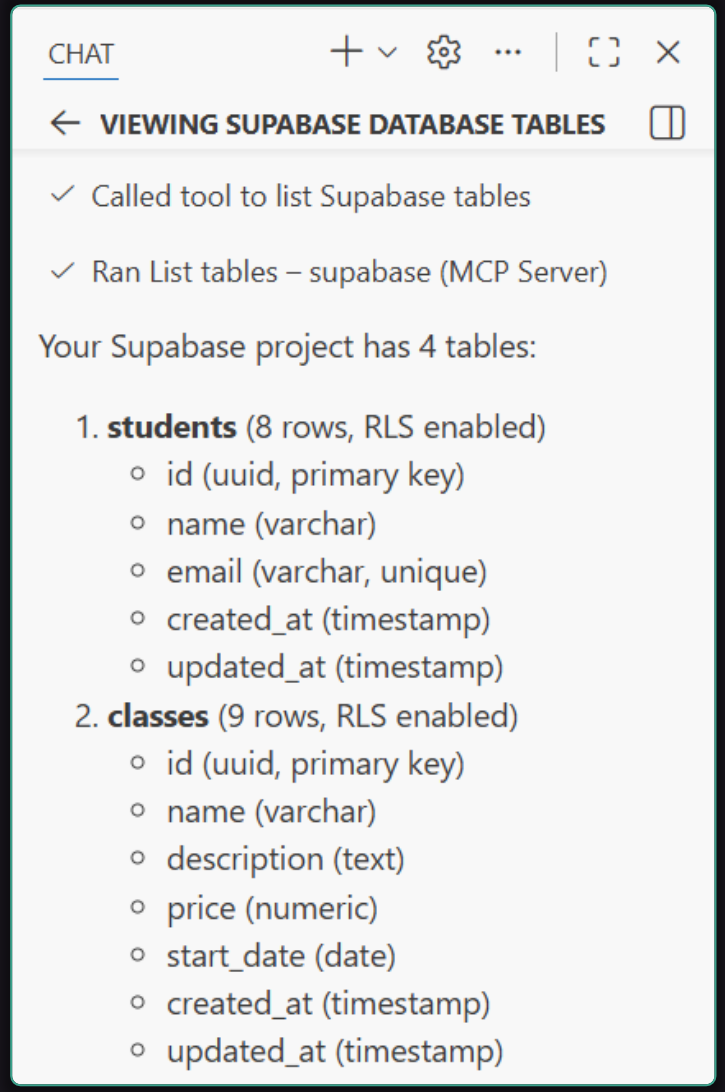


Using the Supabase MCP

SoftUni AI

- Once the **Supabase MCP** is installed and connected, you can **ask the AI agent** to **interact with Supabase**





Exercise: Polls App with MCP

SoftUni AI

- We want to create a **Polls app**: visitors view polls and vote
  - All app data stored in **Supabase** database, public access
- Step 1: Create a **Supabase project** and configure the **Supabase MCP**
  - Create a **new project** in Supabase
  - Install and configure the **Supabase MCP** server:
    - Supabase Dashboard → Connect → MCP → VS Code → Add to VS Code → Install in Workspace
    - Authorize API access for Visual Studio Code

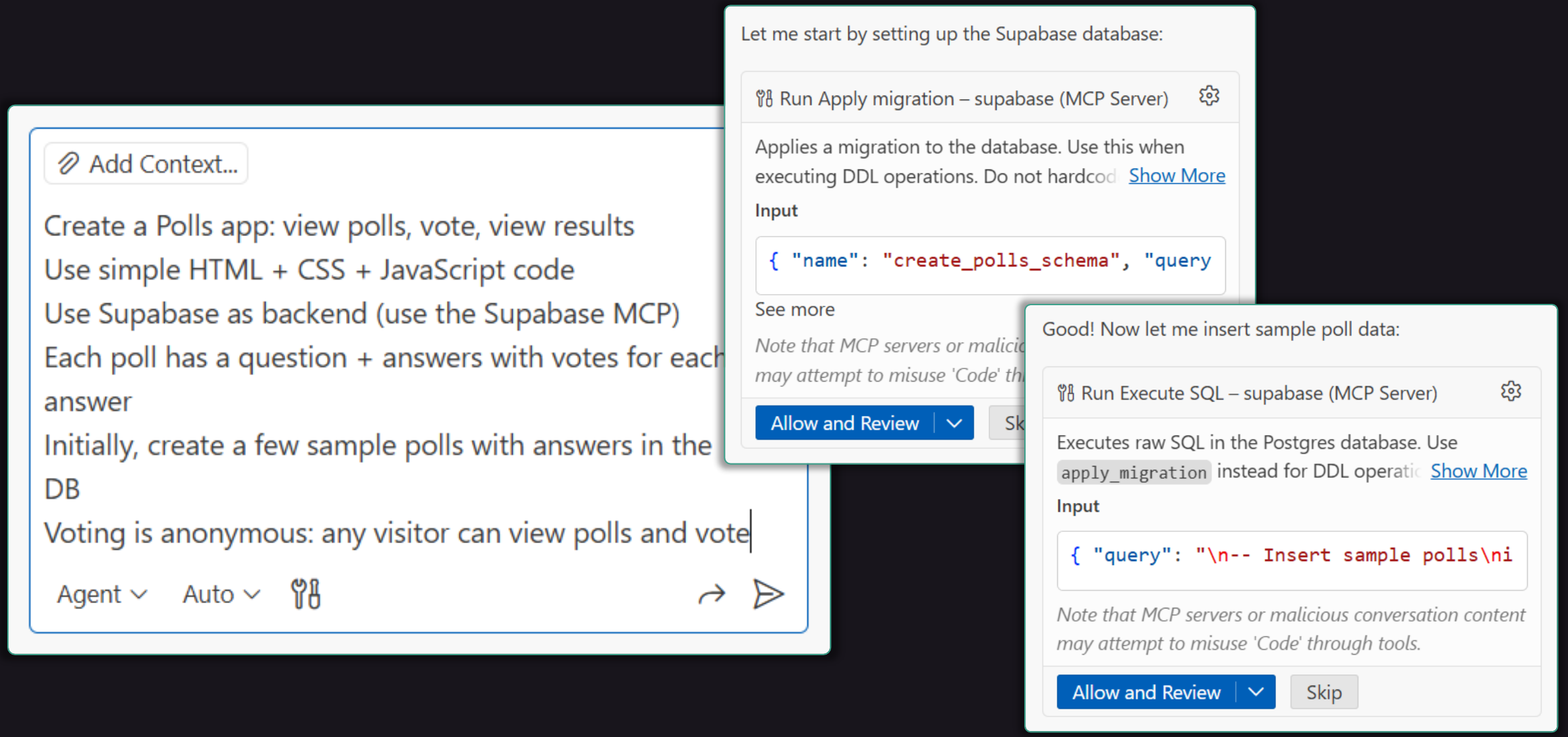
Exercise: Polls App with MCP (2)

SoftUni AI

- Step 2: **Create the app**, using Supabase through MCP
  - Create a **Polls app**: view polls, vote, view results
  - Use simple **HTML** + **CSS** + **JavaScript** code
  - Use **Supabase** as backend (use the Supabase MCP)
  - Each **poll** has a **question** + **answers** with **votes** for each answer
  - Initially, create a few **sample polls** with answers in the DB
  - **Voting is anonymous**: any visitor can view polls and vote

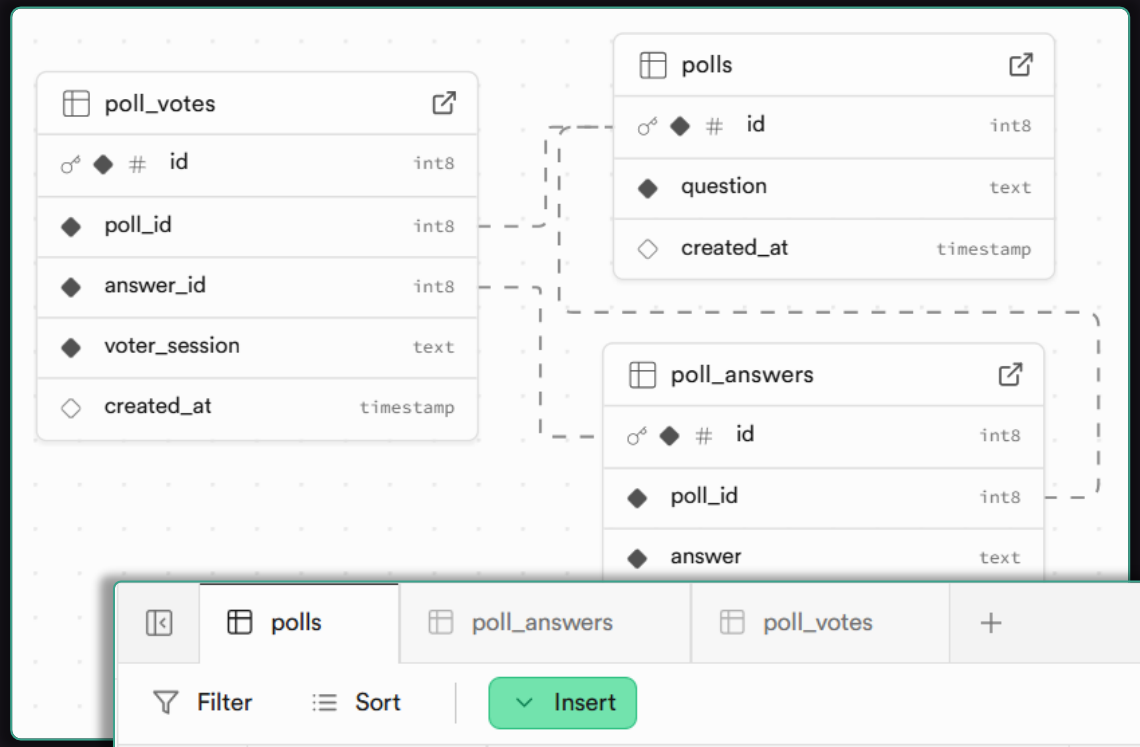
Exercise: Polls App with MCP (3)

SoftUni AI



Exercise: Polls App with MCP (4)

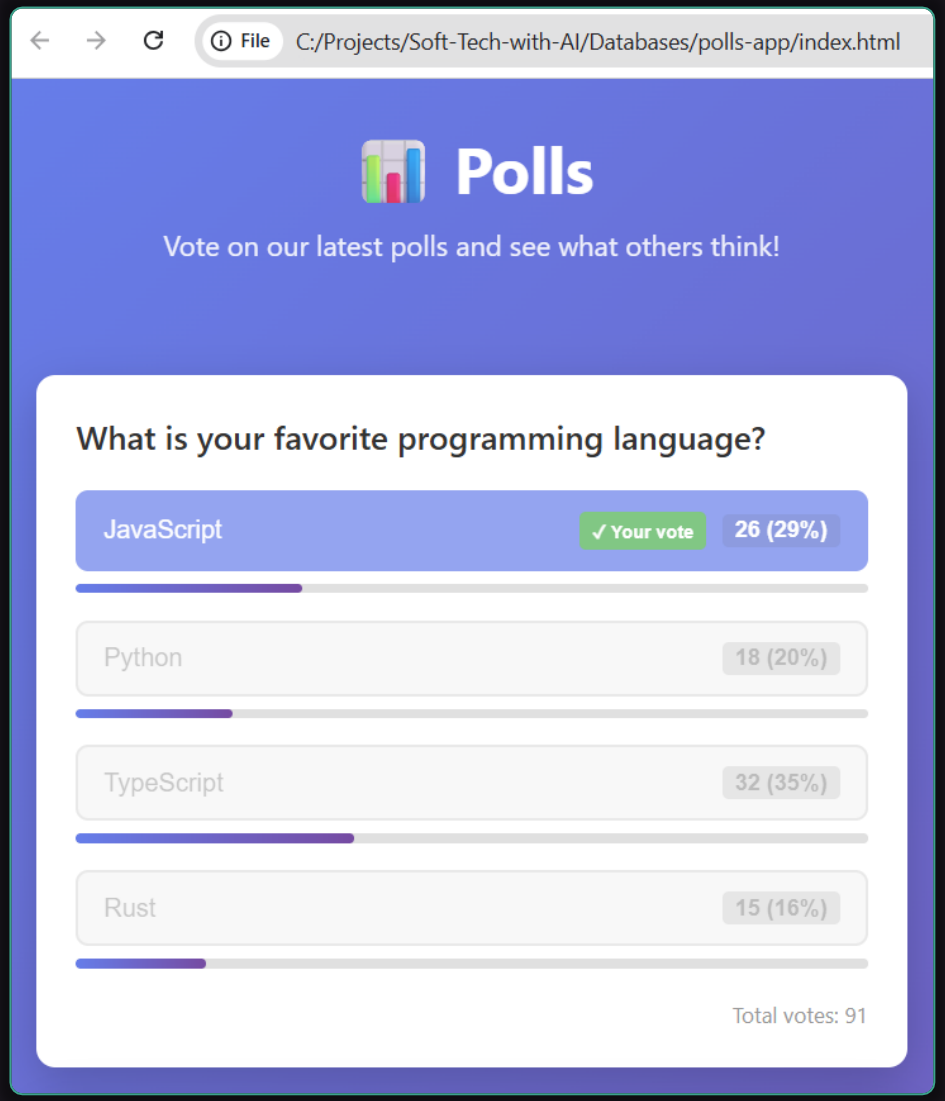
SoftUni AI



polls | poll_answers | poll_votes

Filter | Sort | Insert | RLS disabled

id int8	question text	created_at timestamp
1	What is your favorite programming language?	2026-01-29 07:57:18.882831
2	Which frontend framework do you prefer?	2026-01-29 07:57:18.882831
3	Do you prefer light or dark mode?	2026-01-29 07:57:18.882831
4	What is your preferred work schedule?	2026-01-29 07:57:18.882831



Lesson Summary

SoftUni AI

- **Relational databases** (like PostgreSQL) hold **tables** (with rows and columns) and **relationships** (DB schema)
- **Supabase BaaS**: Projects, Database, Schemas, Tables, Table Editor, SQL Editor, Users and Auth, Storage, Edge Functions
- **SQL** language: table **structure** (CREATE / ALTER TABLE), **CRUD** operations (SELECT, INSERT, UPDATE, DELETE, Joins)
- **JS Apps with Supabase**: create SQL script → create tables in Supabase → API keys → JS code to interact with the DB
- **Supabase MCP**: connect VS Code with Supabase with MCP → create DB and JS app with an AI prompt

Questions?

Postbank – Exclusive Partner for SoftUni AI

- One of the leading **banking institutions** in Bulgaria
- Member of the Eurobank Group with € 103 billion assets
- Innovative trendsetter with next generation **beyond banking**, transforming today, empowering tomorrow
- Certified **Top Employer 2025** by the international Top Employers Institute
- Proven people care and **wellbeing initiatives**
- Benefits and unlimited access to professional, **personal and leadership trainings and programs**
- www.postbank.bg / careers.postbank.bg

Diamond Partners of Software University

Software University

Coca-Cola HBC Bulgaria

Postbank

top EMPLOYER България Bulgaria 2025

FOR A BETTER WORLD OF WORK

SUPER HOSTING .BG

encorp.ai

createX

DRAFT KINGS

THE CROWN IS YOURS

INDEAVR

Serving the high achievers

Diamond Partners of SoftUni Digital

SoftUni Digital

Coca-Cola HBC Bulgaria

Postbank

top EMPLOYER България Bulgaria 2025

FOR A BETTER WORLD OF WORK

SUPER HOSTING .BG

encorp.ai

createX

INDEAVR

Serving the high achievers

DRAFT KINGS

THE CROWN IS YOURS

Diamond Partners of SoftUni Digital

SoftUni Digital

HUMAN

NETPEAK
DIGITAL GROWTH PARTNER

MarmaLAB | E-комерс агенция

ETIENYANEV
Break Your *Limits*. Live Your *Brand*.

1 FOR FIT

Aleikov Studio

Diamond Partners of SoftUni Creative

SoftUni Creative

Coca-Cola HBC Bulgaria

Postbank

top EMPLOYER България Bulgaria 2025

FOR A BETTER WORLD OF WORK

SUPER HOSTING .BG

encorp.ai

createX

INDEAVR

Serving the high achievers

DRAFT KINGS

THE CROWN IS YOURS

Organization Partners of SoftUni Creative

SoftUni Creative

creative corner

Studio π

Think Big • Design Smart

THE BUCKS TOWN'S WORK

DEFFECTO