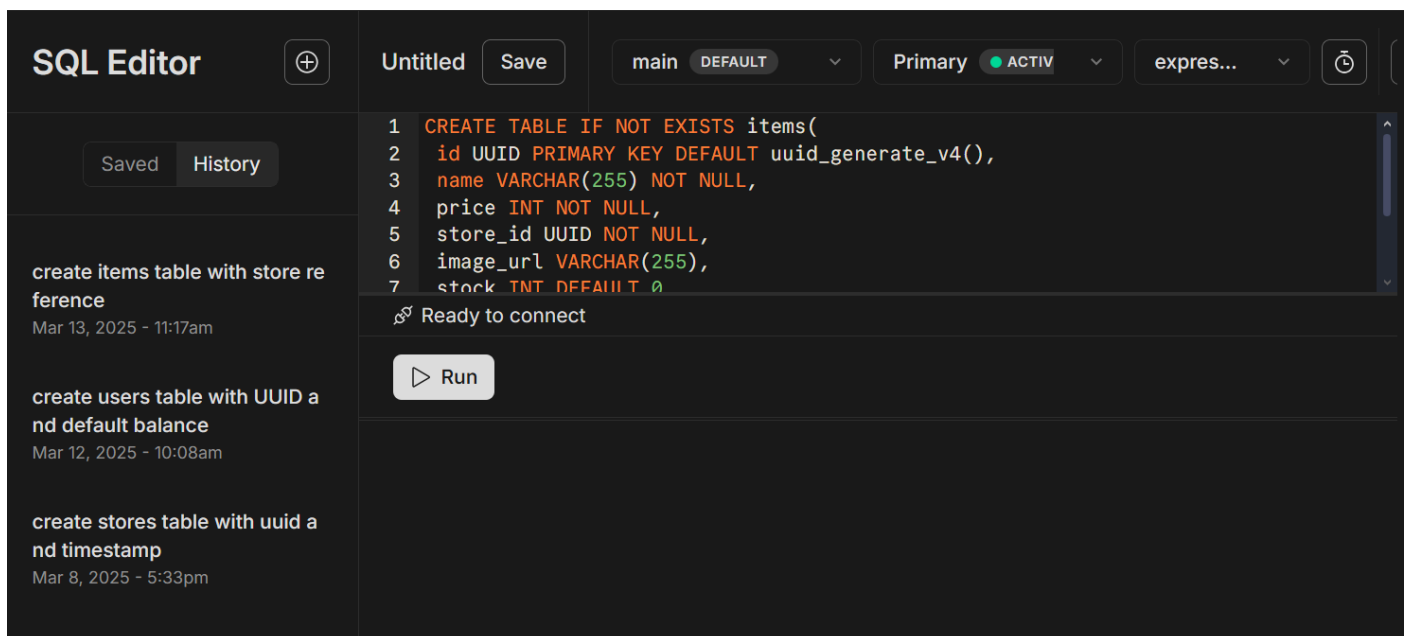


PRAKTIKUM SISTEM BASIS DATA

Nama	Deandro Najwan Ahmad Syahbanna	No. Modul	5
NPM	2306213174	Tipe	TUTAM

1. Jalankan Query di SQL Editor

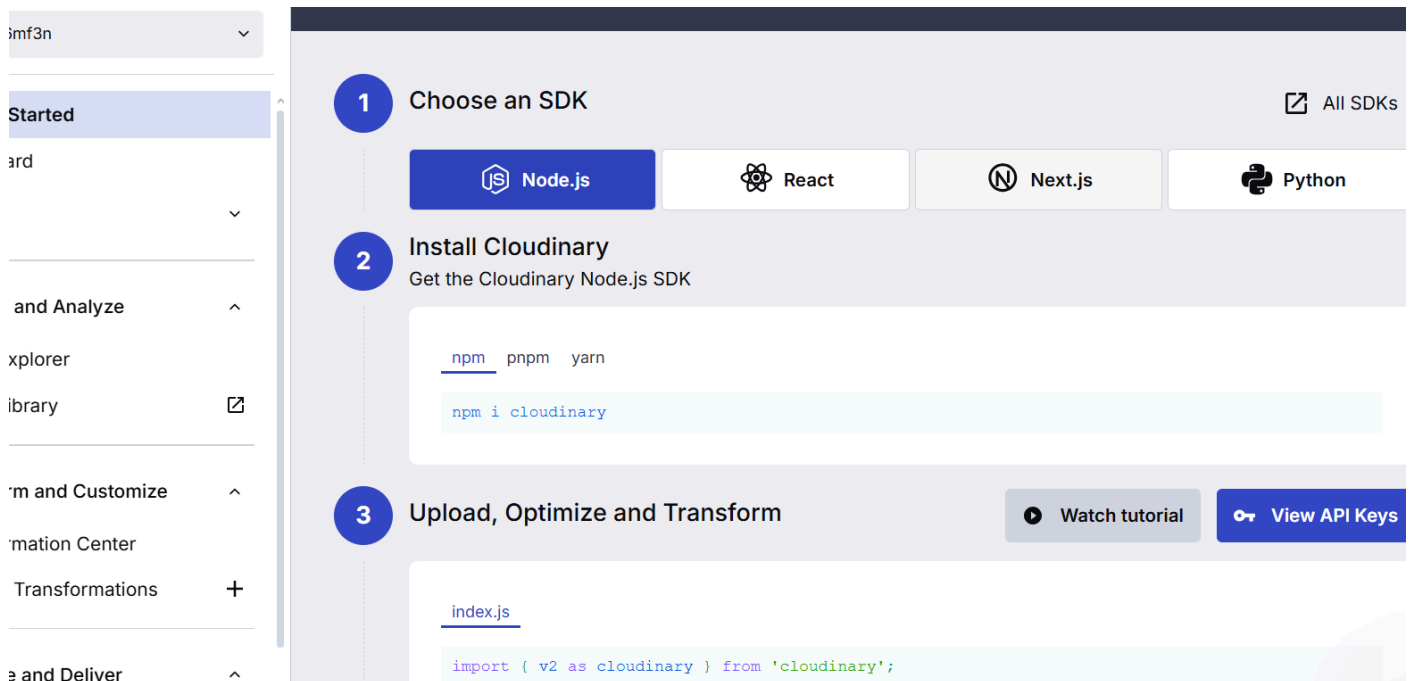


The screenshot shows a SQL Editor interface with a dark theme. On the left, there is a sidebar with a 'History' tab and three saved queries: 'create items table with store reference' (Mar 13, 2025 - 11:17am), 'create users table with UUID and default balance' (Mar 12, 2025 - 10:08am), and 'create stores table with uuid and timestamp' (Mar 8, 2025 - 5:33pm). The main editor area shows a query to create an 'items' table with columns: 'id' (UUID PRIMARY KEY), 'name' (VARCHAR(255) NOT NULL), 'price' (INT NOT NULL), 'store_id' (UUID NOT NULL), 'image_url' (VARCHAR(255)), and 'stock' (INT DEFAULT 0). Below the query, there is a 'Run' button and a status bar indicating 'Ready to connect'.

2. buat form data pada item.route.js

```
> routes > JS item.route.js > ...
1  const express = require('express');
2  const multer = require('multer');
3
4  const itemController = require('../controllers/item.controller');
5  const router = express.Router();
6  const storage = multer.memoryStorage();
7  const upload = multer({ storage: storage });
8
9  router.post('/create', upload.single('image'), itemController.createItem);
10 router.put('/', upload.single('image'), itemController.updateItem);
11
12
13 module.exports = router;
14
```

3. Sign UP cloudinary dan Install Cloudinary



The image shows the Cloudinary SDK installation guide. On the left is a sidebar with a search bar and a list of categories: Started, and, and Analyze, xplorer, library, m and Customize, mation Center, Transformations, e and Deliver. The main content area has three steps:

- 1 Choose an SDK** (All SDKs)
 - Node.js (selected)
 - React
 - Next.js
 - Python
- 2 Install Cloudinary** (Get the Cloudinary Node.js SDK)
 - npm pnpm yarn
 - npm i cloudinary
- 3 Upload, Optimize and Transform** (Watch tutorial View API Keys)
 - index.js
 - import { v2 as cloudinary } from 'cloudinary';

4. Nambahin Cloudinary ke .env

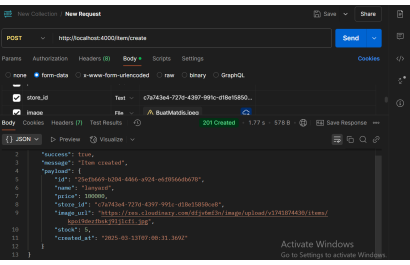
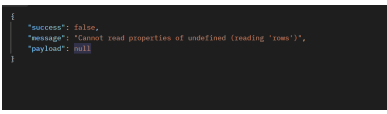

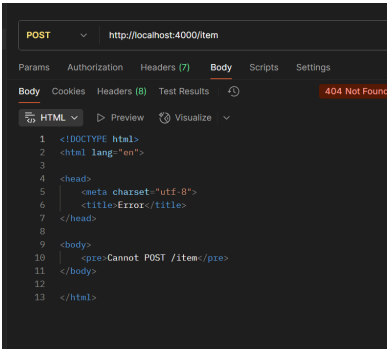
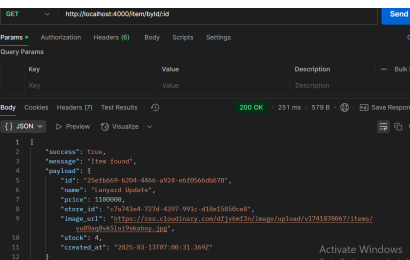
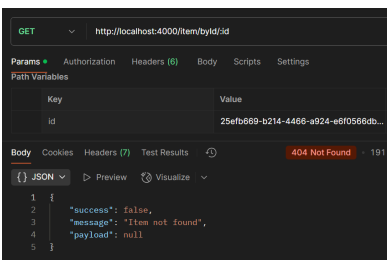
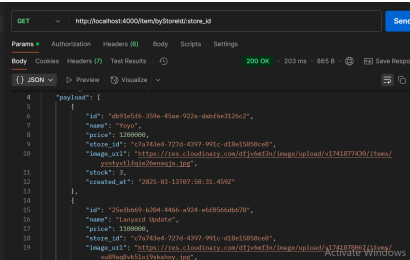
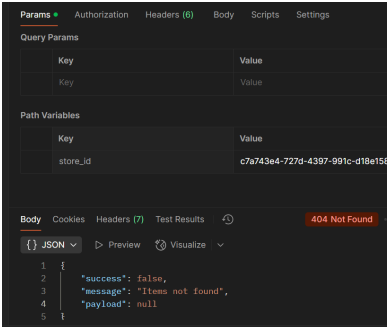
```
.env
Open in Postman | Sync to Postman | You, 1 second ago | 1 author (You)
Untracked
1 DATABASE_URL=postgresql://express_deandro_owner:npg_0CL3AsNYMVTf@ep-twili
2 PORT=4000
3 CLOUDINARY_URL=cloudinary://829175862462816:XepEeX3wnK10KtUVsd5Fuypf-jc@df
```

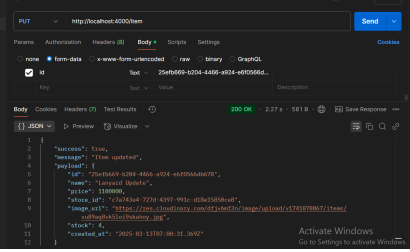
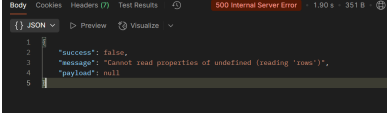
5. Hubungin Cloudinary Ke Node Modules

```
src > utils > JS cloudinary.js > uploadFile > <function>
1 require("dotenv").config();
2 const cloudinary = require("cloudinary").v2;
3
4 cloudinary.config({
5   cloud_name: process.env.CLOUDINARY_CLOUD_NAME,
6   api_key: process.env.CLOUDINARY_API_KEY,
7   api_secret: process.env.CLOUDINARY_API_SECRET,
8 });
9
10 const uploadFile = (file, folder) => {
11   return new Promise((resolve, reject) => {
12     if (!file) return resolve(null); // No file to upload
13
14     const b64 = Buffer.from(file.buffer).toString("base64");
15     const dataURI = `data:${file.mimetype};base64,${b64}`;
16
17     cloudinary.uploader.upload(dataURI, {
18       folder: folder,
19     }, (error, result) => {
20       if (error) reject(error);
21       else resolve(result.secure_url);
22     });
23   });
24 };
25
26 module.exports = {
27   uploadFile,
28   ...cloudinary
29 };
```

install beberapa library yang dibutuhkan seperti cloudinary dsb.

6. Untuk kode perubahannya mungkin bisa dilihat lebih jelas dalam kode yang dikirim, tapi secara umum setelah saya menambahkan cloudinary dan menghubungkannya ke dalam kode express, saya membuat file khusus untuk mengatur tabel items, sama halnya dengan tabel stores dan users.
7. Response Format

method	endpoint	success	false
POST	/item/create		
GET	/item		
GET	/item/byId/:id		
GET	/item/byStoreId/:store_id		

PUT	/item		
DELETE	/item/:id	