



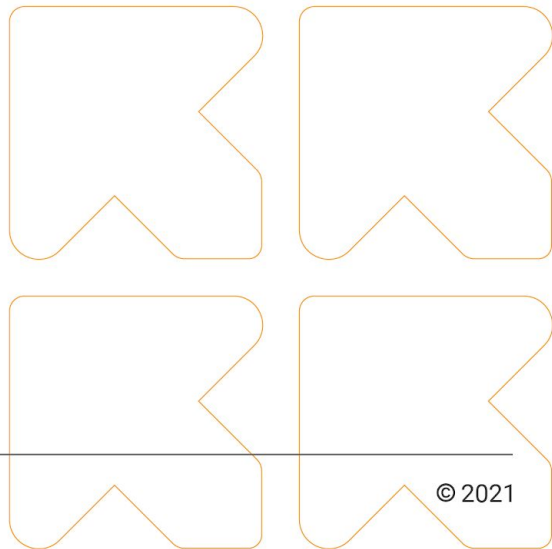
Introduction

Postgresql



Rules

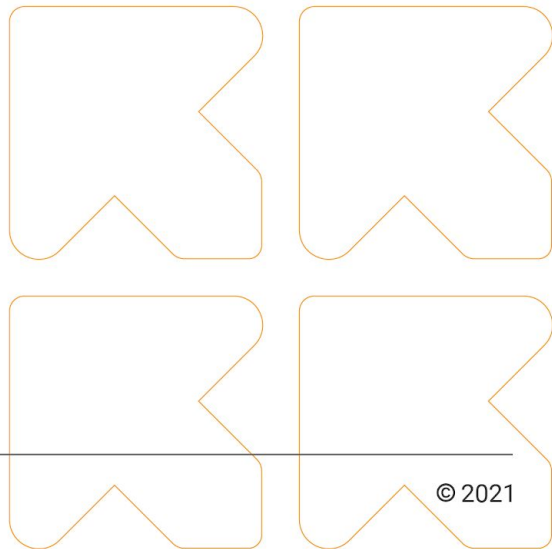
- Selalu hadir di setiap kegiatan bootcamp
- Ikuti peraturan kelas selama bootcamp berlangsung
- Tanyakan apapun ke trainer (Private atau Public)
- Bicaralah untuk kepentinganmu terlebih dahulu
- Trainer ada dari 09.00 s.d. 17.00 WIB
- Bekerja mandiri
- Bekerja keras
- Lakukan yang terbaik
- Selalu ada perkembangan kemampuan individual





Objective

- Postgresql (What, How)
- Managing Tables
- Data types
- Constraints
- Managing Data
- Database Relationship
- Joining Multiple Tables





Postgresql

- PostgreSQL adalah sebuah sistem basis data yang disebarluaskan secara bebas menurut Perjanjian lisensi BSD. postgres merupakan salah satu basis data yang paling banyak digunakan saat ini, selain MySQL dan Oracle.
- PostgreSQL adalah sistem database yang kuat untuk urusan relasi. Memiliki lebih dari 15 tahun pengembangan aktif dan sudah terbukti segala rancangan arsitekturnya telah mendapat reputasi tentang “kuat”, “handal”, “integritas data”, dan “akurasi data”.

Feature	PostgreSQL	MySQL
Open Source	Yes	Yes
ACID	Yes	Partially
SQL Standard	Yes	Partially
High-Performance	Yes	Read Only
Community	Yes	Yes
JSON Support	Yes	Yes
NoSQL Support	Yes	No



Managing Tables

- Tables merupakan kumpulan data yang terorganisir dalam bentuk field dan record. Setiap data yang disimpan dalam table akan memiliki tipe data dan juga ketentuan tertentu (constrain) untuk memudahkan management data. Hal yang dapat dilakukan dengan table :
 - Create table, update table, delete table
 - Add column, update column, delete column

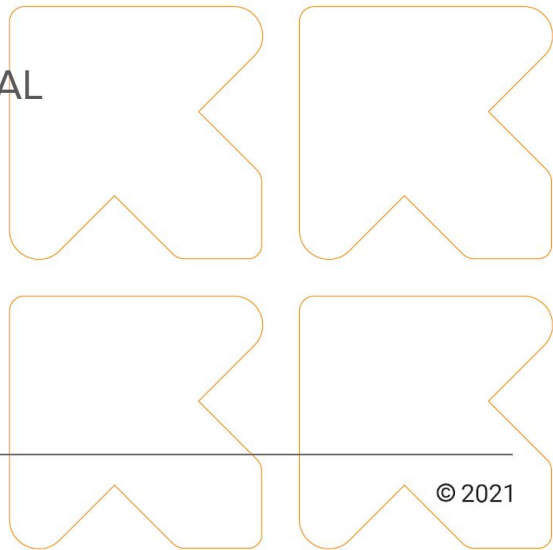


```
CREATE TABLE [IF NOT EXISTS] table_name (  
    column1 datatype(length) column_constraint,  
    column2 datatype(length) column_constraint,  
    column3 datatype(length) column_constraint,  
    table_constraints  
);  
  
CREATE TABLE users (  
    user_id serial PRIMARY KEY,  
    username VARCHAR ( 50 ) UNIQUE NOT NULL,  
    password VARCHAR NOT NULL,  
    email VARCHAR ( 255 ) UNIQUE NOT NULL,  
    created_at TIMESTAMP NOT NULL,  
    updated_at TIMESTAMP  
);
```



Data types

- **Character:** CHAR(n), VARCHAR(n), and TEXT.
- **Integer:** SMALLINT, INT, SERIAL
- **Float:** FLOAT(n), REAL/FLOAT8, NUMERIC(p, s)
- **Temporal:** DATE, TIME, TIMESTAMP, TIMESTAMPTZ, INTERVAL
- **Boolean:** TRUE OR FALSE
- **UUID:** Universal Unique Identifiers defined by RFC 4122
- **Array:** can store an array of strings, an array of integers, etc.,
- **Json:** JSON, JSONB



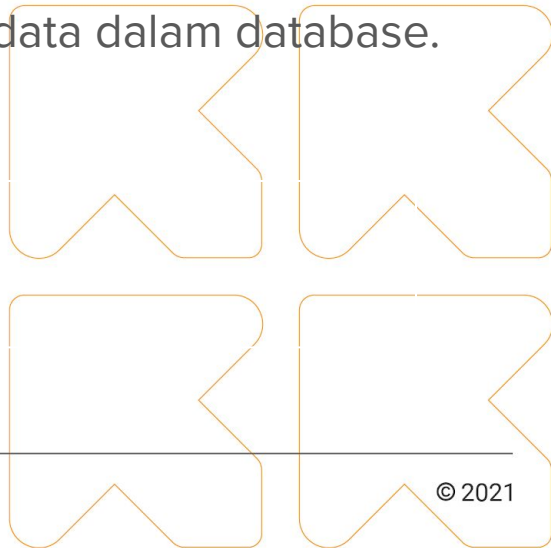


Constraints

- Constraint adalah aturan yang diberlakukan pada data dalam table. Ini digunakan untuk mencegah data yang tidak valid dimasukkan ke dalam database. Ini memastikan keakuratan dan keandalan data dalam database.

Constrain dalam postgres :

- | | |
|-----------|------------|
| ● PRIMARY | ● UNIQUE |
| ● FOREIGN | ● NOT NULL |





Constraints

- **PRIMARY Key:** digunakan untuk mengidentifikasi setiap record dalam tabel database.
Hanya ada satu kunci utama dalam sebuah tabel.
- **FOREIGN Key:** menentukan bahwa nilai dalam kolom (atau sekelompok kolom) harus cocok dengan nilai yang ada di tabel lain.
- **UNIQUE:** mencegah dua data memiliki nilai identik dalam kolom tertentu.
- **NOT NULL:** mencegah sebuah nilai dalam table tidak diisi (wajib diisi).

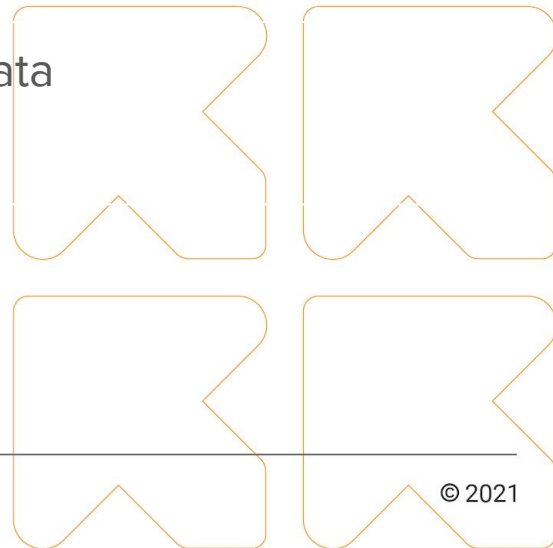


Managing Data

- Beberapa hal yang bisa kita lakukan untuk manage data dalam postgresql antara lain :

- Modifying Data
- Querying Data

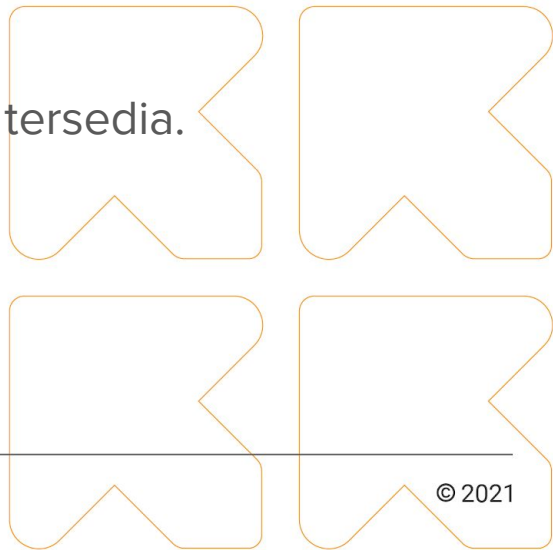
- Filtering Data





Modifying Data

- **Insert:** bagaimana memasukan data dalam database.
- **Update:** merubah data yang sudah ada dengan data baru.
- **Delete:** menghapus data dalam table.
- **Upsert:** merubah atau menambahkan data jika belum tersedia.





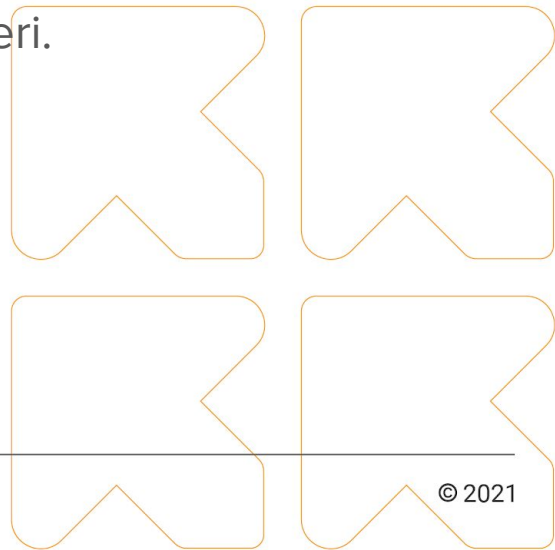
Querying Data

- **Select:** mengambil satu atau lebih data dalam table
- **Column Aliases:** memberikan nama alias untuk column.
- **Order By:** mengurutkan data yang akan diambil dalam table.
- **Select Distinct:** mengambil data dan menghilangkan data yang sama.



Filtering Data

- **Where:** filter data berdasarkan kondisi tertentu.
- **Limit:** dapatkan subset dari baris yang dihasilkan oleh kueri.
- **Fatch:** batasi jumlah baris yang dikembalikan oleh kueri.
- **Like:** memfilter data berdasarkan pencocokan pola



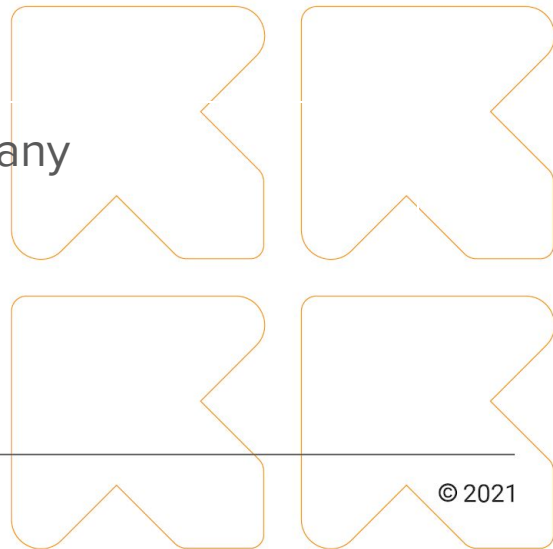


Database Relationship

- Hubungan dibuat antara dua tabel database ketika satu tabel menggunakan kunci asing (Foreign Key) yang mereferensikan kunci utama (Primary Key) dari tabel lain. Database relasional antara lain :

- One-to-one
- One-to-many

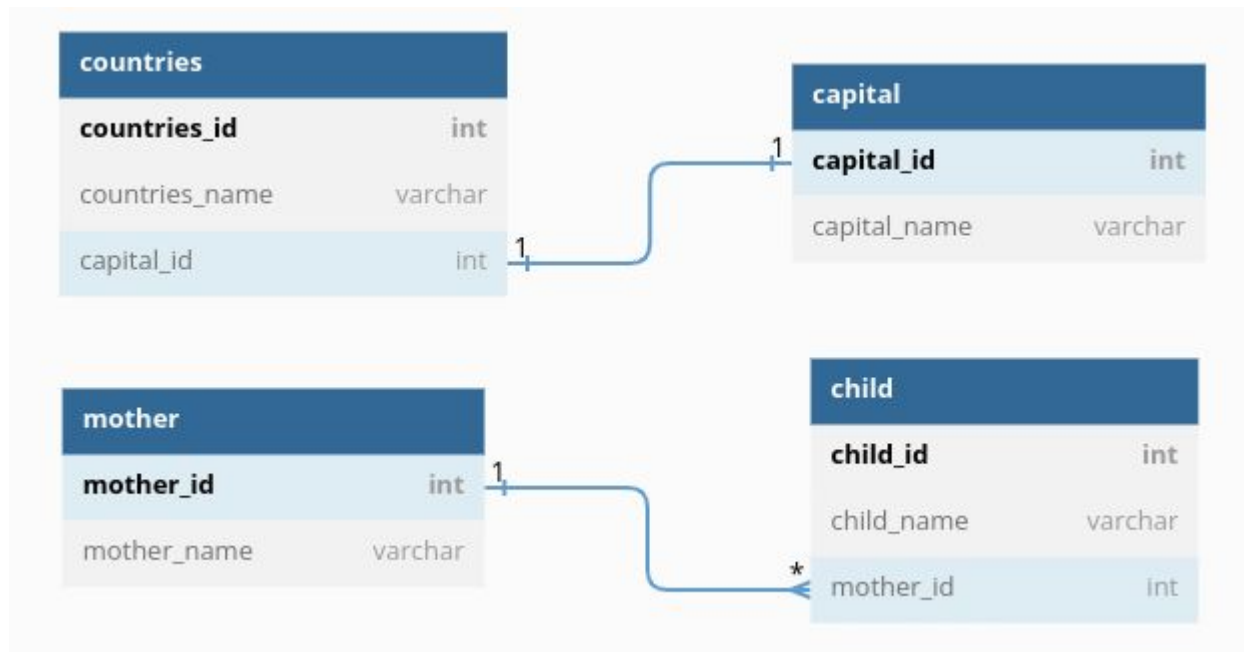
- Many-to-many

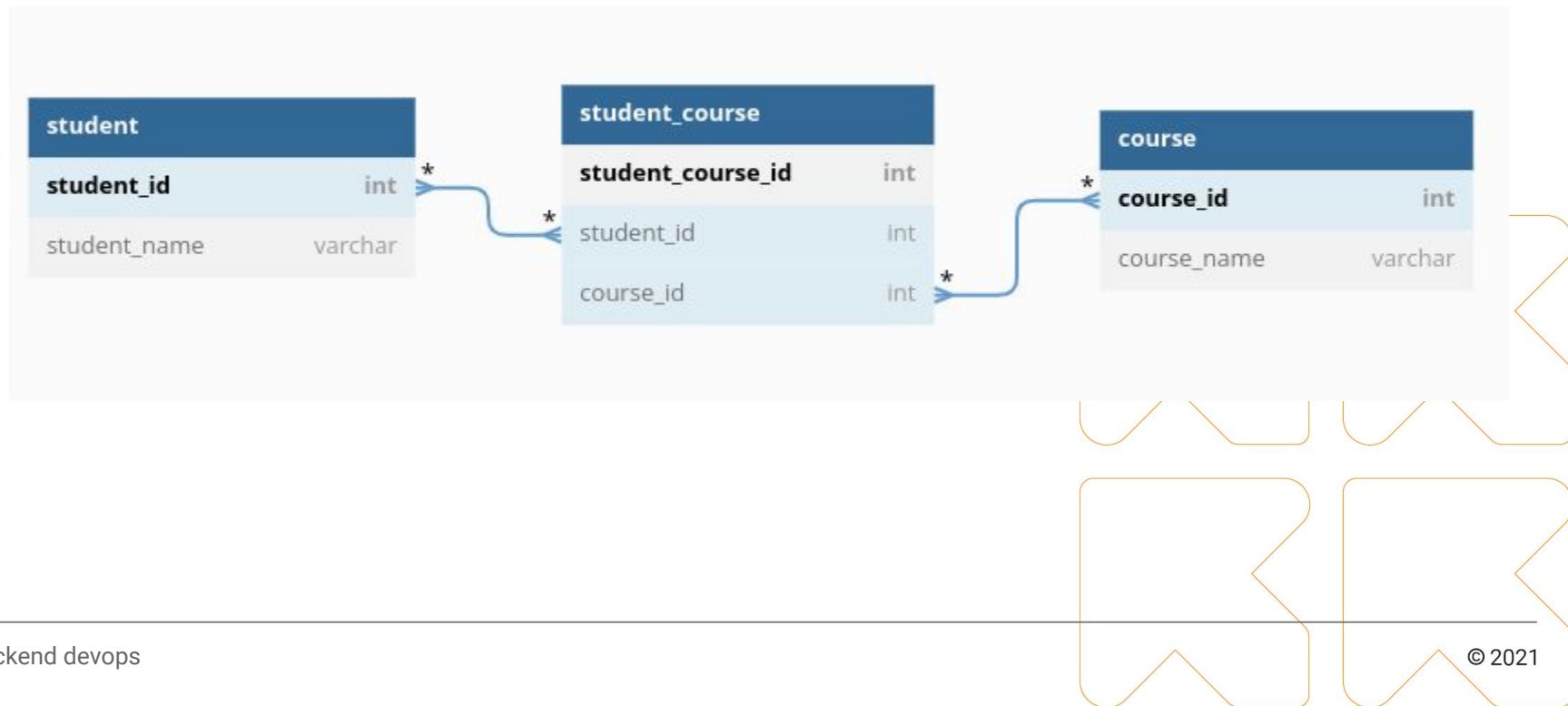




Database Relationship

- **One-to-one:** Kedua tabel hanya dapat memiliki satu record yang terhubung.
- **One-to-many:** Column primary key hanya berisi satu record yang berhubungan dengan tidak ada, satu, atau banyak record dalam tabel terkait.
- **Many-to-many:** Setiap catatan di kedua tabel dapat berhubungan dengan tidak ada atau sejumlah catatan di tabel lainnya. Hubungan ini memerlukan tabel ketiga, yang disebut tabel asosiasi atau tautan.







Joining Multiple Tables

- **Inner Join:** mengambil data yang memiliki nilai yang sesuai di table lain.
- **Left/Right Join:** pilih baris dari satu tabel yang mungkin atau mungkin tidak memiliki baris yang sesuai di tabel lain.
- **Full Outer Join:** untuk menemukan baris dalam tabel yang tidak memiliki baris yang cocok di tabel lain.
- **etc...**





Live Coding

Task Manage Database

Buatlah API dari aplikasi berikut : [Tickitz Movie](#)

- Requirements:
 - Database (Movie, Schedule, Booking, Genre)
 - CRUD
 - Query Film berdasarkan nama
 - Query Film berdasarkan nama, tahun rilis, genre
 - Pagination (optional)
 - Table User (optional)
- Gunakan Bahasa Inggris untuk nama File dan Fungsi
- Presentasikan apa yang telah Anda lakukan dalam minggu ini – Selasa, 30 Mei 2023
 - Jelaskan dengan bahasa kalian sendiri apa itu dan konsep tentang:
 - Sql
 - Postgres
 - Transaction
 - Example Relation
 - Ceritakan dan Demokan tentang proyek ini



Jangan lupa isi feedback yaa

<https://bit.ly/feedbackTechnicalTrainerFazztrack>