

Nomor Kelompok :15

Kelas : K1

NIM :18222043

Nama : Ricky Wijaya

NIM : 18222089

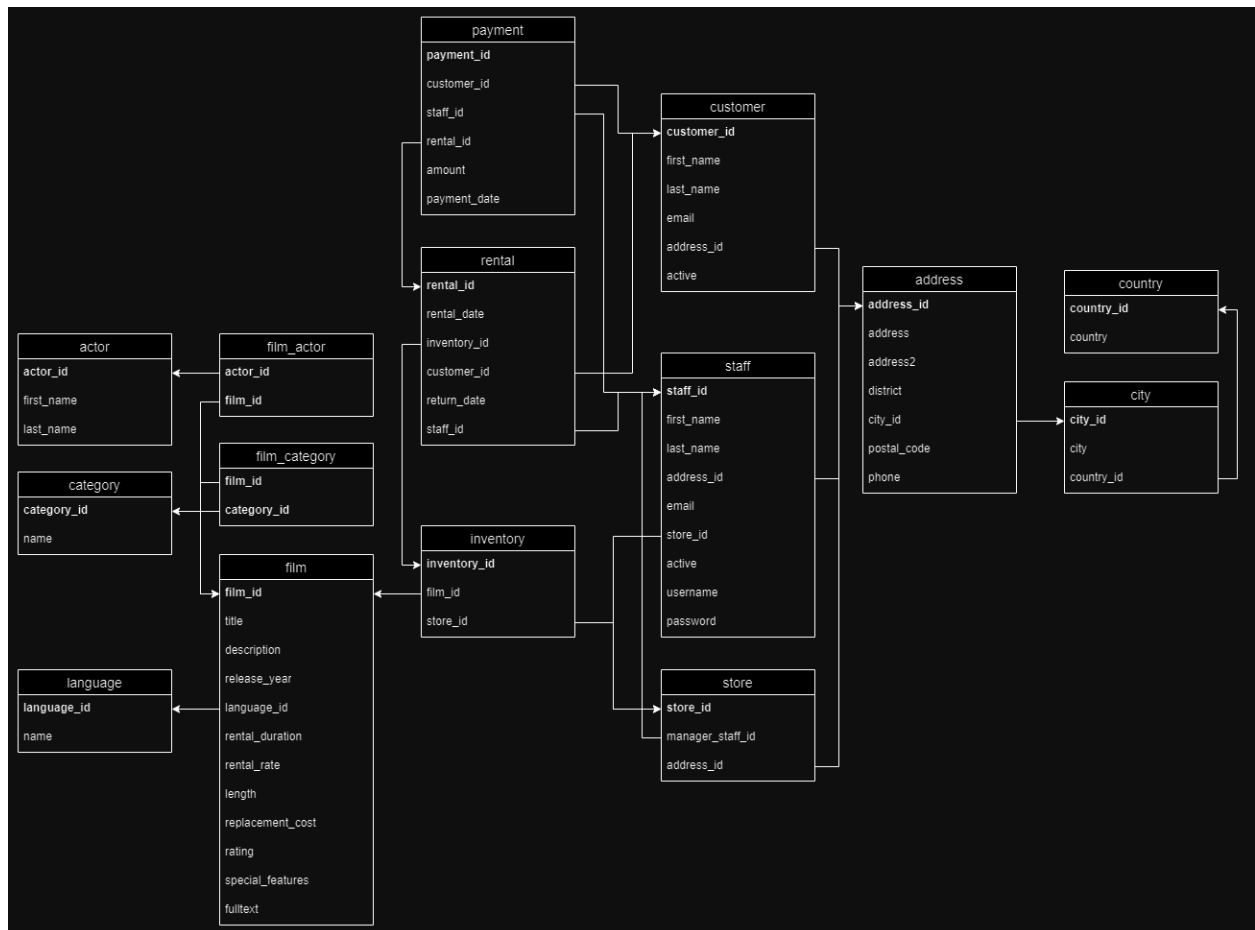
Nama : Muhammad Adli Arindra

Lembar Kerja Praktikum 4 II2250 Manajemen Basis Data STI

Materi: *Constraint & Trigger*

I. Skema Basis Data

Diberikan skema basis data sebagai berikut yang tersimpan dalam database pgila.



II. Soal

(**Note:** Pastikan telah terdapat database bernama *pagila* di dalam komputer yang digunakan. Jika belum, buatlah sebuah database bernama *pagila* dan import *pagila.sql* ke dalam database tersebut!)

1. Constraint

Toko Pagila menyediakan film dengan berbagai rating. Tipe-tipe rating yang ada adalah sebagai berikut:

```
rating: ['PG', 'PG-13', 'G', 'NC-17', 'R']
```

Untuk memastikan data yang dimasukkan ke dalam basis data benar. Buatlah:

- Buatlah sebuah check constraint bernama *valid_rating* yang akan memeriksa apakah record baru untuk rating yang dimasukkan adalah salah satu di antara tipe rating film yang ada di daftar film milik toko Pagila.
- Buatlah contoh penambahan *record* pada tabel Film yang melanggar *check constraint*.
- Buatlah contoh penambahan *record* pada tabel Film yang tidak melanggar *check constraint*.
- Hapus kembali *record* yang baru saja dimasukkan.

Jawaban:


Query Pembuatan Constraint	ALTER TABLE film ADD CONSTRAINT valid_rating CHECK (rating IN ('PG', 'PG-13', 'G', 'NC-17', 'R'));
SS Pembuatan Constraint	<pre>labdas=# ALTER TABLE film ADD CONSTRAINT valid_rating CHECK (rating IN ('PG', 'PG-13', 'G', 'NC-17', 'R')); ALTER TABLE</pre>
SS Pelanggaran Constraint	<pre>prak4=# INSERT INTO film VALUES (11111,'filmcoba','siksakubur',2004,1,1,2,3,4,'PGG',{coba},'cobaa aja'); ERROR: invalid input value for enum mpa_rating: "PGG" LINE 1: ...UES (11111,'filmcoba','siksakubur',2004,1,1,2,3,4,'PGG',{co...</pre>
SS Tidak Melanggar Constraint	<pre>prak4=# INSERT INTO film VALUES (11111,'filmcoba','siksakubur',2004,1,1,2,3,4,'PG',{coba},'cobaa aja'); INSERT 0 1</pre>

2. Constraint

Toko Pagila menyadari pentingnya pencatatan umur pelanggan untuk kebutuhan pemasaran.

- Buatlah sebuah kolom baru bernama age pada tabel customer yang akan menyimpan data umur pelanggan. Kolom tersebut akan menyimpan tipe data integer dan harus menerapkan constraint NOT NULL serta memiliki nilai default sebesar 0.
- Selanjutnya, buatlah sebuah check constraint yang akan memeriksa apakah nilai umur yang dimasukkan bernilai valid. Data umur bernilai valid apabila berada pada rentang nilai 0 hingga 100.
- Terakhir, buatlah sebuah contoh penambahan data pada tabel pasien yang melanggar check constraint yang telah dibuat pada bagian sebelumnya.

Jawaban:

Query Penambahan Kolom & Pembuatan Constraint	Pembuatan kolom <pre>ALTER TABLE customer ADD COLUMN age INT NOT NULL DEFAULT 0;</pre> Pembuatan constraint <pre>ALTER TABLE customer ADD CONSTRAINT check_age CHECK (age BETWEEN 0 AND 100);</pre>
Query Contoh Insert yang gagal	<pre>INSERT INTO customer VALUES (12345, 'Ricky', 'Wijaya', 'rickywijaya@gmail.com', 4000, true, -80);</pre>
SS Query Contoh Insert	 <pre>prakt4=# INSERT INTO customer VALUES (12345, 'Ricky', 'Wijaya', 'rickywijaya@gmail.com', 4000, true, -80); ERROR: new row for relation "customer" violates check constraint "check_age" DETAIL: Failing row contains (12345, Ricky, Wijaya, rickywijaya@gmail.com, 4000, t, -80).</pre>

3. Constraint

Pagila menyimpan alamat para customernya dalam tabel address, namun terdapat kesalahan skema pada tabel tersebut. Kolom postal_code pada tabel address memiliki tipe varchar.

Untuk memastikan data yang diinput memiliki nilai yang benar, lakukanlah hal-hal berikut:

- Ubah tipe data pada kolom *postal_code* menjadi *integer*.

- b. Berikan check constraint untuk memastikan kolom postal_code memiliki nilai dalam rentang 1000 - 10000. ($0 \leq \text{postal_code} \leq 100000$)
- c. Buatlah sebuah contoh penambahan data pada tabel address yang melanggar check constraint yang telah dibuat pada bagian sebelumnya.

Jawaban:

<div>Detail Schema Tabel address (Before)</div>	<div><pre>prak4=# \d address;</pre><table><thead><tr><th>Column</th><th>Type</th><th>Table "public.address"</th><th>Collation</th><th>Nullable</th><th>Default</th></tr></thead><tbody><tr><td>address_id</td><td>integer</td><td></td><td></td><td>not null</td><td>nextval('address_address_id_seq'::regclass)</td></tr><tr><td>address</td><td>character varying(50)</td><td></td><td></td><td>not null</td><td></td></tr><tr><td>address2</td><td>character varying(50)</td><td></td><td></td><td></td><td></td></tr><tr><td>district</td><td>character varying(20)</td><td></td><td></td><td>not null</td><td></td></tr><tr><td>city_id</td><td>smallint</td><td></td><td></td><td>not null</td><td></td></tr><tr><td>postal_code</td><td>character varying(10)</td><td></td><td></td><td></td><td></td></tr><tr><td>phone</td><td>character varying(20)</td><td></td><td></td><td>not null</td><td></td></tr></tbody></table></div>	Column	Type	Table "public.address"	Collation	Nullable	Default	address_id	integer			not null	nextval('address_address_id_seq'::regclass)	address	character varying(50)			not null		address2	character varying(50)					district	character varying(20)			not null		city_id	smallint			not null		postal_code	character varying(10)					phone	character varying(20)			not null	
Column	Type	Table "public.address"	Collation	Nullable	Default																																												
address_id	integer			not null	nextval('address_address_id_seq'::regclass)																																												
address	character varying(50)			not null																																													
address2	character varying(50)																																																
district	character varying(20)			not null																																													
city_id	smallint			not null																																													
postal_code	character varying(10)																																																
phone	character varying(20)			not null																																													
<div>Query Perubahan Type Kolom</div>	<div><pre>UPDATE address SET postal_code = '12345' WHERE postal_code = '';</pre><pre>ALTER TABLE address ALTER COLUMN postal_code TYPE integer using (postal_code::integer);</pre></div>																																																
<div>Query Pembuatan Constraint</div>	<div><pre>ALTER TABLE address ADD CONSTRAINT check_postal_code CHECK (postal_code BETWEEN 0 AND 100000);</pre></div>																																																
<div>Detail Schema Tabel address (After)</div>	<div><table><thead><tr><th>Column</th><th>Type</th><th>Collation</th><th>Nullable</th><th>Default</th></tr></thead><tbody><tr><td>address_id</td><td>integer</td><td></td><td>not null</td><td>nextval('address_address_id_seq'::regclass)</td></tr><tr><td>address</td><td>character varying(50)</td><td></td><td>not null</td><td></td></tr><tr><td>address2</td><td>character varying(50)</td><td></td><td></td><td></td></tr><tr><td>district</td><td>character varying(20)</td><td></td><td>not null</td><td></td></tr><tr><td>city_id</td><td>smallint</td><td></td><td>not null</td><td></td></tr><tr><td>postal_code</td><td>integer</td><td></td><td></td><td></td></tr><tr><td>phone</td><td>character varying(20)</td><td></td><td>not null</td><td></td></tr></tbody></table><pre>Indexes: "address_pkey" PRIMARY KEY, btree (address_id) "idx_fk_city_id" btree (city_id) Check constraints: "check_postal_code" CHECK (postal_code >= 0 AND postal_code <= 100000) Foreign-key constraints: "fk_address_city" FOREIGN KEY (city_id) REFERENCES city(city_id) Referenced by: TABLE "customer" CONSTRAINT "customer_address_id_fkey" FOREIGN KEY (address_id) REFERENCES address(address_id) ON UPDATE CASCADE ON DELETE RESTRICT TABLE "staff" CONSTRAINT "staff_address_id_fkey" FOREIGN KEY (address_id) REFERENCES address(address_id) ON UPDATE CASCADE ON DELETE RESTRICT TABLE "store" CONSTRAINT "store_address_id_fkey" FOREIGN KEY (address_id) REFERENCES address(address_id) ON UPDATE CASCADE ON DELETE RESTRICT</pre></div>	Column	Type	Collation	Nullable	Default	address_id	integer		not null	nextval('address_address_id_seq'::regclass)	address	character varying(50)		not null		address2	character varying(50)				district	character varying(20)		not null		city_id	smallint		not null		postal_code	integer				phone	character varying(20)		not null									
Column	Type	Collation	Nullable	Default																																													
address_id	integer		not null	nextval('address_address_id_seq'::regclass)																																													
address	character varying(50)		not null																																														
address2	character varying(50)																																																
district	character varying(20)		not null																																														
city_id	smallint		not null																																														
postal_code	integer																																																
phone	character varying(20)		not null																																														
<div>SS Insert Pelanggaran Constraint</div>	<div><pre>prak4=# INSERT INTO address VALUES (12345,'jalanjalan',NULL,'Bandung',300,-10,'081081081'); ERROR: new row for relation "address" violates check constraint "check_postal_code" DETAIL: Failing row contains (12345, jalanjalan, null, Bandung, 300, -10, 081081081). prak4=# []</pre></div>																																																

4. Constraint

Dalam memasukkan data rental, terdapat tanggal peminjaman dan tanggal pengembalian yang harus dimasukkan. Toko Pagila membuat aturan baru bahwa pengembalian harus dilakukan maksimal 9 hari setelah peminjaman dilakukan. Toko Pagila menjamin bahwa tidak ada pelanggan yang melakukan pengembalian lebih dari 9 hari setelah peminjaman dilakukan. Selain itu, data tanggal pengembalian yang dimasukkan tidak boleh berupa tanggal yang terjadi

sebelum peminjaman dilakukan. Toko pagila ingin memastikan konsistensi data rental. Oleh karena itu, Toko Pagila memerlukan bantuan Anda untuk memastikannya.

- Buatlah sebuah check constraint yang akan memeriksa apakah data tanggal pengembalian yang dimasukkan bernilai benar. Data tanggal pengembalian yang benar adalah tanggal yang terjadi setelah peminjaman dilakukan, serta memiliki perbedaan return_date dan rental_date tidak lebih dari 9 hari.
- Buatlah sebuah contoh penambahan data pada tabel rental yang melanggar check constraint yang telah dibuat pada bagian sebelumnya.
- Hapuslah constraint yang baru saja anda buat.

Hint: Gunakan DATE_PART('day', <end date> - <start date>) untuk mencari perbedaan return_date dan rental_date

Jawaban:

SS Query Pembuatan Constraint	<pre>prak4=# ALTER TABLE rental ADD CONSTRAINT check_date CHECK (DATE_PART('day', return_date - rental_date) <= 9 AND return_date > rental_date); ALTER TABLE</pre>
Query Contoh Insert dan Berikan SS nya	<pre>INSERT INTO rental VALUES(123456, '2004-05-24 00:00:00', 1525, 459, '2005-05-28 19:40:33 prak4'# ', 1);</pre> <pre>prak4=# INSERT INTO rental VALUES(123456, '2004-05-24 00:00:00', 1525, 459, '2005-05-28 19:40:33 prak4'# ', 1); ERROR: new row for relation "rental" violates check constraint "check_date"</pre>
Hapus Constraint	<pre>prak4=# ALTER TABLE rental DROP CONSTRAINT check_date; ALTER TABLE</pre>

5. Trigger

Hingga hari ini, Toko Pagila sudah memiliki dua cabang. Setiap cabang memiliki banyak inventory untuk menyimpan kaset film. Penjaga toko perlu memastikan bahwa kaset film yang hendak disewa oleh pelanggan tersedia di inventory-nya. Dengan kata lain, kaset di inventory tersebut tidak sedang disewa oleh pelanggan lain. Oleh karena itu, Toko Pagila ingin menerapkan trigger sebelum sewa dilakukan, yaitu dengan memastikan tanggal sewa lebih besar daripada tanggal pengembalian terakhir di inventory tersebut. Buatlah query untuk menerapkan trigger tersebut! Buktikan dengan penambahan entry yang berhasil dan gagal!

Jawaban:

<p>Query pembuatan trigger</p>	<pre>CREATE OR REPLACE FUNCTION check_tanggal() RETURNS trigger LANGUAGE plpgsql AS \$\$ BEGIN IF NEW.rental_date > (SELECT return_date FROM rental WHERE NEW.inventory_id = rental.inventory_id ORDER BY return_date DESC LIMIT 1) THEN RETURN NEW; ELSE RAISE EXCEPTION 'Tanggal tidak valid!'; END IF; END; \$\$;</pre> <p>Trigger</p> <pre>CREATE TRIGGER check_date BEFORE INSERT ON rental FOR EACH ROW EXECUTE FUNCTION check_tanggal();</pre>
<p>Penambahan entry berhasil</p>	<pre>prak4=# INSERT INTO rental VALUES (123456, '2005-08-27 00:00:00', 1525, 127, '2005-08-26', 1); INSERT 0 1</pre>
<p>Penambahan entry gagal</p>	<pre>prak4=# INSERT INTO rental VALUES (123456, '2005-08-22 00:00:00', 1525, 127, '2005-08-25', 1); ERROR: Tanggal tidak valid!</pre>

6. Trigger

Pegawai dari Toko Pagila menyadari terdapat beberapa instances data yang disimpan namun tidak perlu, misalnya seperti di tabel city. Oleh sebab itu, Toko Pagila ingin merapikan database-nya dengan mengarsipkan data-data yang tidak dibutuhkan itu pada tabel yang berbeda dari tabel aktifnya.

- Buatlah sebuah *query* untuk membuat tabel baru dengan nama *city_archive* dengan atribut yang sama dengan tabel *city*.
- Buatlah sebuah *trigger* untuk menambahkan data baru pada tabel *city_archive* setiap menghapus data dari tabel *city*.

HINT:

- **Buat Function terlebih dahulu sebelum membuat Trigger**
- c. Buatlah *query* untuk menghapus data pada tabel *city* yang tidak digunakan di tabel *address*. Lalu perlihatkan semua data yang ada di dalam tabel *city_archive*.

Jawaban:

Query pembuatan trigger	Function CREATE OR REPLACE FUNCTION archiving() RETURNS trigger LANGUAGE plpgsql AS \$\$ BEGIN INSERT INTO city_archive VALUES (NEW.city_id, NEW.city, NEW.country_id); END; \$\$;
Penambahan entry berhasil	
Penambahan entry gagal	

III. Pembagian Tugas

NIM	Nama	Tugas
18222089	Muhammad Adli ARindra	
18222043	Ricky Wijaya	