

Nomor :
Kelas : **K01**

Kelompok : **07**

NIM : **18222013**
Nama : **Aththariq Lisan Q. D. S.**

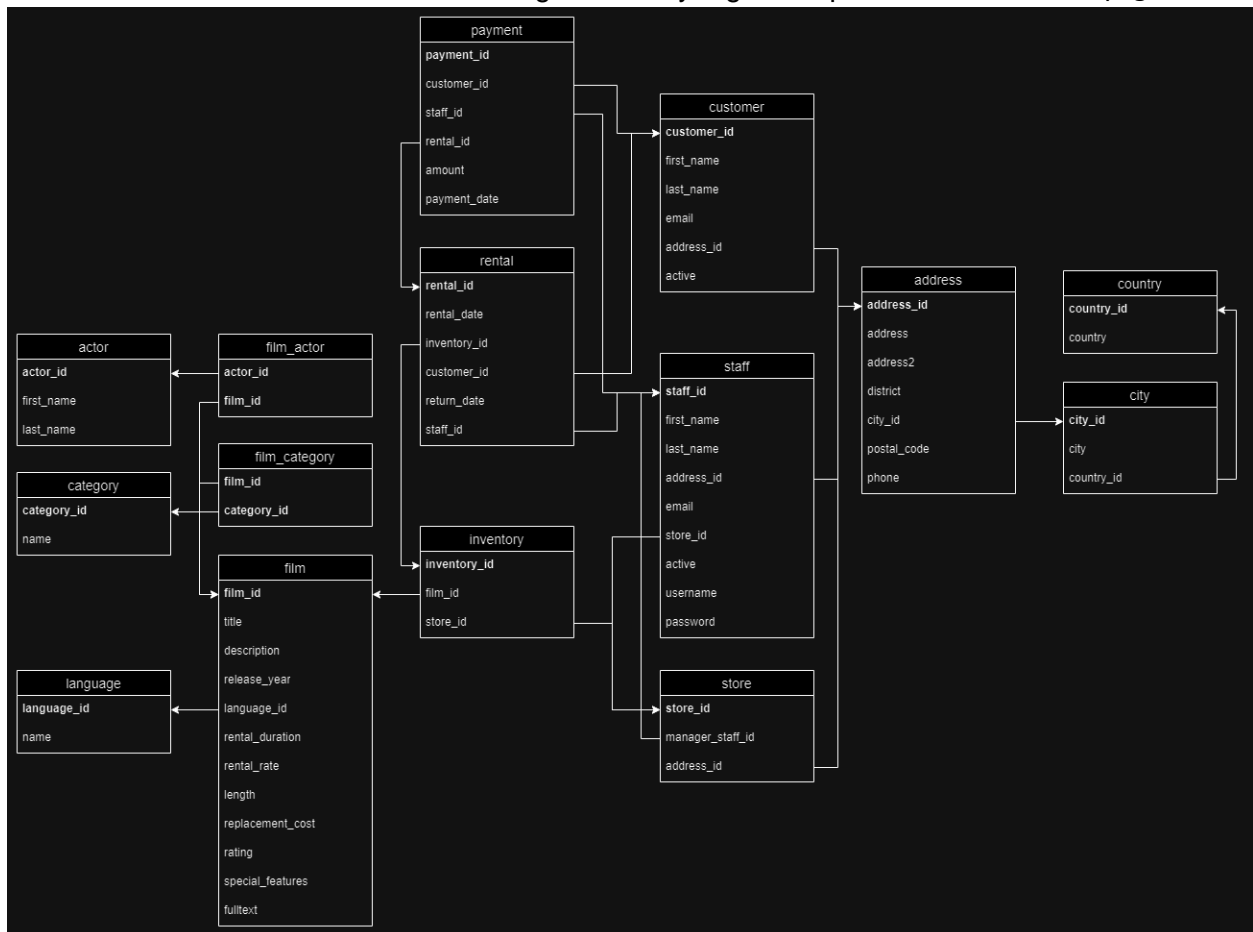
NIM : **18222061**
Nama : **Winata Tristan**

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 *pagila*.



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>pagila=# ALTER TABLE FILM ADD CONSTRAINT valid_rating check(rating in ('PG', 'PG-13', 'G', 'NC-17', 'R')); ALTER TABLE pagila=#</pre>
SS Pelanggaran Constraint	<pre>INSERT INTO FILM(FILM_ID, TITLE, DESCRIPTION, RELEASE_YEAR, LANGUAGE_ID, RENTAL_DURATION, RENTAL_RATE, LENGTH, REPLACEMENT_COST, RATING, SPECIAL_FEATURES, FULLTEXT) VALUES(510000, 'star wars', 'seru coy', '2006', 1, 1, 4.99, 1, 20.99, 'Dewasa', '{Trailer, Behind the Scene}', 'chamber');</pre> <pre>pagila=# INSERT INTO FILM(FILM_ID, TITLE, DESCRIPTION, RELEASE_YEAR, LANGUAGE_ID, RENTAL_DURATION, RENTAL_RATE, LENGTH, REPLACEMENT_COST, RATING, SPECIAL_FEATURES, FULLTEXT) VALUES(510000, 'star wars', 'seru coy', '2006', 1, 1, 4.99, 1, 20.99, 'Dewasa', '{Trailer, Behind the Scene}', 'chamber'); ERROR: invalid input value for enum mpaa_rating: "Dewasa" LINE 2: ... wars', 'seru coy', '2006', 1, 1, 4.99, 1, 20.99, 'Dewasa', ... ^</pre>
SS Tidak Melanggar	<pre>INSERT INTO FILM(FILM_ID, TITLE, DESCRIPTION, RELEASE_YEAR, LANGUAGE_ID,</pre>

Constraint	RENTAL_DURATION, RENTAL_RATE, LENGTH, REPLACEMENT_COST, RATING, SPECIAL_FEATURES, FULLTEXT) VALUES(500000, 'star wars', 'seru coy', '2006', 1, 1, 4.99, 1, 20.99, 'G', '{Trailer, Behind the Scene}', 'chamber'); <pre> pagila=# INSERT INTO FILM(FILM_ID, TITLE, DESCRIPTION, RELEASE_YEAR, LANGUAGE_ID, RENTAL_DURATION, RENTAL_RATE, LENGTH, REPLACEMENT_COST, RATING, SPECIAL_FEATURES, FULLTEXT) VALUES(500000, 'star wars', 'seru coy', '2006', 1, 1, 4.99, 1, 20.99, 'G', '{Trailer, Behind the Scene}', 'cha mber'); INSERT # 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	ALTER TABLE CUSTOMER ADD COLUMN AGE INT NOT NULL DEFAULT 0; <pre> pagila=# ALTER TABLE CUSTOMER ADD COLUMN AGE INT NOT NULL DEFAULT 0; ALTER TABLE </pre> ALTER TABLE CUSTOMER ADD CONSTRAINT CHECK_AGE CHECK(AGE>=0 AND AGE <=100); <pre> pagila=# ALTER TABLE CUSTOMER ADD CONSTRAINT CHECK_AGE CHECK(AGE>=0 AND AGE <=100); ALTER TABLE </pre>
Query Contoh Insert yang gagal	INSERT INTO CUSTOMER(CUSTOMER_ID, FIRST_NAME, LAST_NAME, EMAIL, ADDRESS_ID, ACTIVE, AGE) VALUES(1000000, 'jokowi', 'widodo', 'jokowiganteng@gmail.com', 1, TRUE, 101);

SS Query Contoh Insert

```
pagila=# INSERT INTO CUSTOMER(CUSTOMER_ID, FIRST_NAME, LAST_NAME, EMAIL, ADDRESS_ID, ACTIVE, AGE)
VALUES(1000000, 'jokowi', 'widodo', 'jokowiganteng@gmail.com', 1, TRUE, 101);
ERROR: new row for relation "customer" violates check constraint "check_age"
DETAIL: Failing row contains (1000000, jokowi, widodo, jokowiganteng@gmail.com, 1, t, 101).
```

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

Jawaban:

Detail Schema Tabel address (Before)

```
pagila=# \d address;
          Table "public.address"
   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) |           | not null |              |
 district    | character varying(20) |           | not null |              |
 city_id     | smallint       |           | not null |              |
 postal_code | character varying(10) |           | not null |              |
 phone       | character varying(20) |           | not null |              |
Indexes:
    "address_pkey" PRIMARY KEY, btree (address_id)
    "idx_fk_city_id" btree (city_id)
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
```

Query Perubahan Type Kolom

```
ALTER TABLE address
ALTER COLUMN postal_code TYPE INTEGER
USING postal_code::integer;
```

Query Pembuatan Constraint

```
UPDATE address SET postal_code = '0'
WHERE postal_code = '';
```

```
pagila=# UPDATE address SET postal_code = '0'
WHERE postal_code = '';
UPDATE 4
pagila=# ALTER TABLE address
ALTER COLUMN postal_code TYPE INTEGER
USING postal_code::integer;
ALTER TABLE
```

```
ALTER TABLE ADDRESS
ADD CONSTRAINT CHECK_POSTAL_CODE
CHECK(postal_code >= 0 AND postal_code <= 100000);
```

```
pagila=# ALTER TABLE ADDRESS
ADD CONSTRAINT CHECK_POSTAL_CODE CHECK(postal_code >= 0 AND postal_code <= 100000);
ALTER TABLE
```

<div>Detail Schema</div> <div>Tabel address</div> <div>(After)</div>	<div><pre>pgsql=# \d address;</pre><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</div> <div>Pelanggaran</div> <div>Constraint</div>	<div><div>INSERT INTO address(address_id, address, address2, district, city_id, postal_code, phone)</div><div>VALUES (10000000, 'dago', 'tubis', 'bandung', 1, -1, '0812345678');</div><div>VALUES (10000000, 'dago', 'tubis', 'bandung', 1, -1, '0812345678');</div><div>ERROR: new row for relation "address" violates check constraint "check_postal_code"</div><div>DETAIL: Failing row contains (10000000, dago, tubis, bandung, 1, -1, 0812345678).</div><div>pgsql=# []</div></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*

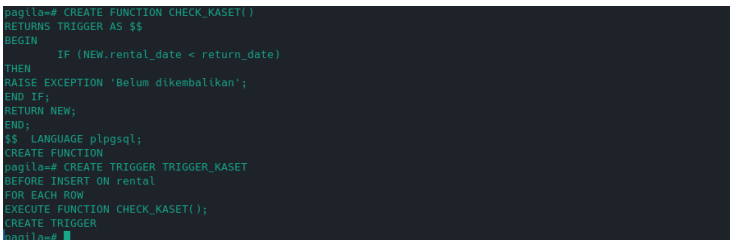
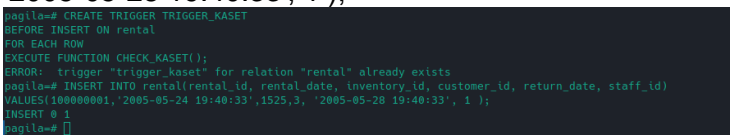
Jawaban:

SS Query Pembuatan Constraint	<pre>ALTER TABLE RENTAL ADD CONSTRAINT VALID_RETURN CHECK(DATE_PART('day', return_date - rental_date) <= 9 AND DATE_PART('day', return_date - rental_date) >= 0);</pre> <pre>pagila=# ALTER TABLE RENTAL ADD CONSTRAINT VALID_RETURN CHECK(DATE_PART('day', return_date - rental_date) <= 9 AND DATE_PART('day', return_date - rental_date) >= 0); ALTER TABLE pagila=#</pre>
Query Contoh Insert dan Berikan SS nya	<pre>INSERT INTO RENTAL(rental_id, rental_date, inventory_id, customer_id, return_date, staff_id) VALUES(100000001,'2005-05-28 19:40:33',1525,3, '2005-05-24 19:40:33', 1);</pre> <pre>pagila=# INSERT INTO RENTAL(rental_id, rental_date, inventory_id, customer_id, return_date, staff_id) VALUES(100000001, '2005-05-28 19:40:33', 1525, 3, '2005-05-24 19:40:33', 1); ERROR: new row for relation "rental" violates check constraint "valid_return" DETAIL: Failing row contains (100000001, 2005-05-28 19:40:33, 1525, 3, 2005-05-24 19:40:33, 1). pagila=#</pre>
Hapus Constraint	<pre>ALTER TABLE rental DROP CONSTRAINT valid_return;</pre> <pre>DETAIL: Failing row contains (100000001, 2005-05-28 19:40:33, 1525, 3, 2005-05-24 19:40:33, 1). pagila=# ALTER TABLE rental DROP CONSTRAINT valid_return; ALTER TABLE pagila=#</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:

Query pembuatan trigger	<pre>CREATE OR REPLACE FUNCTION CHECK_KASET() RETURNS TRIGGER AS \$\$ BEGIN IF (NEW.rental_date < old.return_date) THEN RAISE EXCEPTION 'Belum dikembalikan'; END IF; RETURN NEW; END; \$\$ LANGUAGE plpgsql; CREATE TRIGGER TRIGGER_KASET BEFORE INSERT ON rental FOR EACH ROW EXECUTE FUNCTION CHECK_KASET();</pre> 
Penambahan entry berhasil	<pre>INSERT INTO rental(rental_id, rental_date, inventory_id, customer_id, return_date, staff_id) VALUES(100000001,'2005-05-24 19:40:33',1525,3, '2005-05-28 19:40:33', 1);</pre> 
Penambahan entry gagal	<pre>INSERT INTO rental(rental_id, rental_date, inventory_id, customer_id, return_date, staff_id) VALUES(100000001,'2005-05-29 19:40:33',1525,3, '2005-05-28 19:40:33', 1);</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**
- 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	<pre> CREATE TABLE city_archive AS SELECT * FROM city WHERE FALSE; CREATE OR REPLACE FUNCTION CHECK_CITY() RETURNS TRIGGER AS \$\$ BEGIN INSERT INTO city_archive VALUES (old.*) RETURN old; END; \$\$ LANGUAGE plpgsql; CREATE TRIGGER TRIGGER_CITY AFTER DELETE ON CITY FOR EACH ROW EXECUTE FUNCTION CHECK_CITY(); DELETE FROM city WHERE city_id NOT IN (SELECT DISTINCT city_id FROM address); </pre>
--------------------------------	---

	SELECT * FROM city_archive;
Penambahan <i>entry</i> berhasil	
Penambahan <i>entry</i> gagal	

III. Pembagian Tugas

NIM	Nama	Tugas