

Kelas : K01

NIM : 18222043

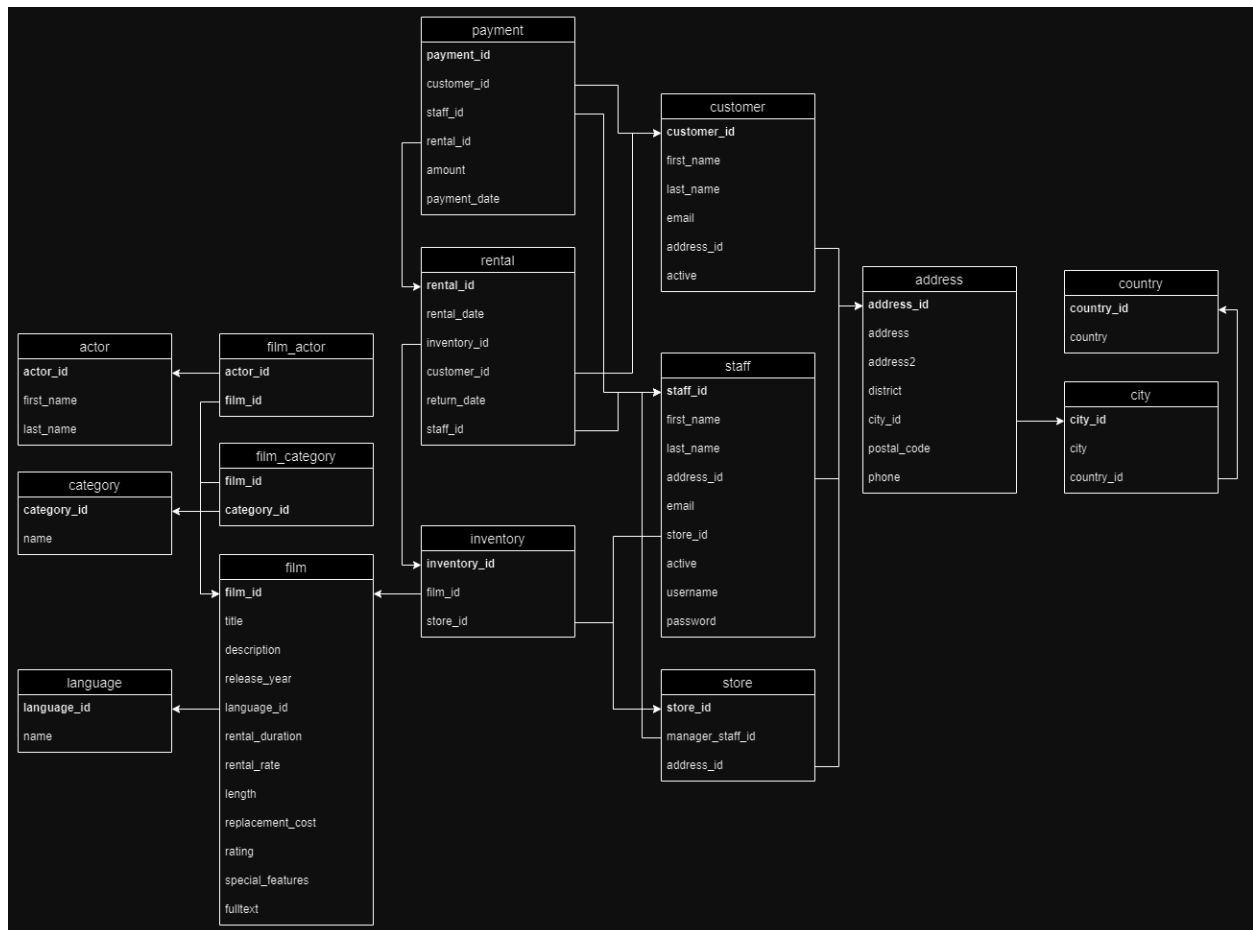
Nama : Ricky Wijaya

Lembar Kerja Praktikum 3 II2250 Manajemen Basis Data STI

Materi: *Functions & Stored Procedure*

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. Function

Toko pagila mengatur harga rental film dengan ketentuan 50% dari harga rental tersebut akan digunakan sebagai biaya operasional (sewa tempat, gaji pegawai, dsb), dan sisanya sebagai keuntungan rental tiap kali dilakukan penyewaan. Rental baru bisa dikatakan menghasilkan keuntungan ketika biaya beli film (80% dari biaya penggantian film) sudah tercover dari keuntungan yang didapatkan ketika pelanggan melakukan penyewaan atau dapat disebut sebagai break even point (BEP).

Buatlah sebuah function untuk menghitung break even point (berapa kali film disewa agar menutupi biaya pembelian film tersebut) dengan menerima argumen berupa harga beli dan keuntungan tiap melakukan rental film tersebut.

Lalu buatlah query untuk menampilkan judul film, tahun rilis, durasi rental, dan break even point (BEP) dari film tersebut dan urutkan berdasarkan BEP terkecil.

HINT : Bulatkan hasil fungsi BEP ke atas karena penyewaan hanya bisa dilakukan dalam bilangan bulat.

HINT TAMBAHAN : IF DALAM POSTGRESQL

Jawaban:

Query Pembuatan Function	<pre>CREATE OR REPLACE FUNCTION BEP(x real, y real) RETURNS integer AS \$\$ DECLARE hasilbep integer; BEGIN IF y > 0 THEN hasilbep := CEILING((0.8*x)/(0.5*y)); ELSE hasilbep := -1; END IF; RETURN hasilbep; END; \$\$ LANGUAGE plpgsql;</pre> <p>– asumsi jika rental_rate bernilai 0 maka akan</p>
---------------------------------	--

	menghasilkan BEP -1																																																																																																				
SS Query Pembuatan Function	<pre>prak3=# CREATE OR REPLACE FUNCTION BEP(x real, y prak3(# real) prak3-# RETURNS integer AS \$\$ prak3\$# DECLARE prak3\$# hasilbep integer; prak3\$# BEGIN prak3\$# IF y > 0 THEN prak3\$# hasilbep := CEILING((0.8*x)/(0.5*y)); prak3\$# ELSE prak3\$# hasilbep := -1; prak3\$# END IF; prak3\$# RETURN hasilbep; prak3\$# END; prak3\$# \$\$ prak3-# LANGUAGE plpgsql; CREATE FUNCTION</pre>																																																																																																				
Query Penerapan Function	SELECT title, release_year, rental_duration, BEP(replacement_cost,rental_rate) BEP FROM film ORDER BY BEP asc;																																																																																																				
SS Hasil Penerapan Function	<pre>prak3=# SELECT title, release_year, rental_duration, prak3-# BEP(replacement_cost,rental_rate) BEP FROM film prak3-# ORDER BY BEP asc;</pre> <table><thead><tr><th>title</th><th>release_year</th><th>rental_duration</th><th>bep</th></tr></thead><tbody><tr><td>Chicago North</td><td>2006</td><td>6</td><td>4</td></tr><tr><td>Super Wyoming</td><td>2006</td><td>5</td><td>4</td></tr><tr><td>Videotape Arsenic</td><td>2006</td><td>4</td><td>4</td></tr><tr><td>Sting Personal</td><td>2006</td><td>3</td><td>4</td></tr><tr><td>Trap Guys</td><td>2006</td><td>3</td><td>4</td></tr><tr><td>Kissing Dolls</td><td>2006</td><td>3</td><td>4</td></tr><tr><td>North Tequila</td><td>2006</td><td>4</td><td>4</td></tr><tr><td>Edge Kissing</td><td>2006</td><td>5</td><td>4</td></tr><tr><td>Scarface Bang</td><td>2006</td><td>3</td><td>4</td></tr><tr><td>Heartbreakers Bright</td><td>2006</td><td>3</td><td>4</td></tr><tr><td>Streetcar Intentions</td><td>2006</td><td>5</td><td>4</td></tr><tr><td>Dude Blindness</td><td>2006</td><td>3</td><td>4</td></tr><tr><td>October Submarine</td><td>2006</td><td>6</td><td>4</td></tr><tr><td>Whale Bikini</td><td>2006</td><td>4</td><td>4</td></tr><tr><td>Memento Zoolander</td><td>2006</td><td>4</td><td>4</td></tr><tr><td>Paths Control</td><td>2006</td><td>3</td><td>4</td></tr><tr><td>Sabrina Midnight</td><td>2006</td><td>5</td><td>4</td></tr><tr><td>Elizabeth Shane</td><td>2006</td><td>7</td><td>4</td></tr><tr><td>Vampire Whale</td><td>2006</td><td>4</td><td>4</td></tr><tr><td>Matrix Snowman</td><td>2006</td><td>6</td><td>4</td></tr><tr><td>Maiden Home</td><td>2006</td><td>3</td><td>4</td></tr><tr><td>Fellowship Autumn</td><td>2006</td><td>6</td><td>4</td></tr><tr><td>Yentl Idaho</td><td>2006</td><td>5</td><td>4</td></tr><tr><td>Impossible Prejudice</td><td>2006</td><td>7</td><td>4</td></tr></tbody></table>	title	release_year	rental_duration	bep	Chicago North	2006	6	4	Super Wyoming	2006	5	4	Videotape Arsenic	2006	4	4	Sting Personal	2006	3	4	Trap Guys	2006	3	4	Kissing Dolls	2006	3	4	North Tequila	2006	4	4	Edge Kissing	2006	5	4	Scarface Bang	2006	3	4	Heartbreakers Bright	2006	3	4	Streetcar Intentions	2006	5	4	Dude Blindness	2006	3	4	October Submarine	2006	6	4	Whale Bikini	2006	4	4	Memento Zoolander	2006	4	4	Paths Control	2006	3	4	Sabrina Midnight	2006	5	4	Elizabeth Shane	2006	7	4	Vampire Whale	2006	4	4	Matrix Snowman	2006	6	4	Maiden Home	2006	3	4	Fellowship Autumn	2006	6	4	Yentl Idaho	2006	5	4	Impossible Prejudice	2006	7	4
title	release_year	rental_duration	bep																																																																																																		
Chicago North	2006	6	4																																																																																																		
Super Wyoming	2006	5	4																																																																																																		
Videotape Arsenic	2006	4	4																																																																																																		
Sting Personal	2006	3	4																																																																																																		
Trap Guys	2006	3	4																																																																																																		
Kissing Dolls	2006	3	4																																																																																																		
North Tequila	2006	4	4																																																																																																		
Edge Kissing	2006	5	4																																																																																																		
Scarface Bang	2006	3	4																																																																																																		
Heartbreakers Bright	2006	3	4																																																																																																		
Streetcar Intentions	2006	5	4																																																																																																		
Dude Blindness	2006	3	4																																																																																																		
October Submarine	2006	6	4																																																																																																		
Whale Bikini	2006	4	4																																																																																																		
Memento Zoolander	2006	4	4																																																																																																		
Paths Control	2006	3	4																																																																																																		
Sabrina Midnight	2006	5	4																																																																																																		
Elizabeth Shane	2006	7	4																																																																																																		
Vampire Whale	2006	4	4																																																																																																		
Matrix Snowman	2006	6	4																																																																																																		
Maiden Home	2006	3	4																																																																																																		
Fellowship Autumn	2006	6	4																																																																																																		
Yentl Idaho	2006	5	4																																																																																																		
Impossible Prejudice	2006	7	4																																																																																																		
*boleh SS awal dan akhir row saja jika terlalu panjang																																																																																																					

	Watership Frontier	2006	6	47
	Ridgemont Submarine	2006	3	47
	Rock Instinct	2006	4	47
	Roman Punk	2006	7	47
	Fury Murder	2006	3	47
	Lesson Cleopatra	2006	3	47
	Voyage Legally	2006	6	47
	Sunset Racer	2006	6	47
	Storm Happiness	2006	6	47
	Double Wrath	2006	4	47
	Magnolia Forrester	2006	4	47
	Treasure Command	2006	3	47
	Goldfinger Sensibility	2006	3	49
	Graffiti Love	2006	3	49
	Hills Neighbors	2006	5	49
	Hollywood Anonymous	2006	7	49
	Feud Frogmen	2006	6	49
	Gilmore Boiled	2006	5	49
	Loathing Legally	2006	4	49
	Honey Ties	2006	3	49
	Everyone Craft	2006	4	49
	Japanese Run	2006	6	49
	Cruelty Unforgiven	2006	7	49
	Tracy Cider	2006	3	49
	Song Hedwig	2006	3	49
	Ballroom Mockingbird	2006	6	49
	Bonnie Holocaust	2006	4	49
	Patient Sister	2006	7	49
	Earth Vision	2006	7	49
	Clyde Theory	2006	4	49
	Clockwork Paradise	2006	7	49
	Arabia Dogma	2006	6	49
	River Outlaw	2006	4	49
	Sassy Packer	2006	6	49
	(1000 rows)			

2. Function

Toko Pagila berencana untuk menerapkan diskon pada biaya sewa film berdasarkan dua kriteria: **popularitas kategori film** dan **nilai rating** dari film tersebut. Diskon ditentukan dengan aturan berikut:

- Film yang termasuk dalam 5 besar kategori film terpopuler (kategori film yang paling banyak dipinjam) akan menerima diskon sebesar 20%.
- Film yang memiliki **rental rate** lebih dari 4 akan menerima diskon sebesar 5%.
- Film yang memenuhi kedua kriteria di atas akan menerima diskon sebesar 25%.

Tampilkan 10 film dengan diskon terbesar

Catatan : Diperbolehkan membuat view untuk membantu perhitungan

Jawaban:

Query Pembuatan Function	<pre>CREATE VIEW populerkategori AS SELECT fc.category_id, COUNT(*) AS rental_count FROM rental r JOIN inventory i ON r.inventory_id = i.inventory_id JOIN film_category fc ON i.film_id = fc.film_id GROUP BY fc.category_id</pre>
---------------------------------	---

	<pre> ORDER BY rental_count DESC LIMIT 5; CREATE FUNCTION diskon(film_category_id int, rating numeric) RETURNS numeric AS \$\$ DECLARE diskon NUMERIC := 0; populer BOOLEAN; BEGIN SELECT EXISTS (SELECT 1 FROM populerkategori WHERE category_id = film_category_id) INTO populer; IF populer THEN IF rating > 4 THEN diskon := 0.25; ELSE diskon := 0.20; END IF; ELSE IF rating > 4 THEN diskon := 0.05; END IF; END IF; RETURN diskon; END; \$\$ LANGUAGE plpgsql; </pre>
Query Penerapan Function	<pre> SELECT f.title, diskon(fc.category_id, f.rental_rate) AS diskonsewa FROM film f JOIN film_category fc ON f.film_id = fc.film_id ORDER BY diskonsewa DESC LIMIT 10; </pre>

SS Hasil Penerapan Function

```
pagila=# SELECT f.title, hitung_diskon(fc.category_id, f.rental_rate) AS diskonsewa
FROM film f JOIN film_category fc
ON f.film_id = fc.film_id
ORDER BY diskonsewa DESC
LIMIT 10;
 title | diskonsewa
-----|-----
 Bilko Anonymous | 0.25
 California Birds | 0.25
 Attacks Hate | 0.25
 Bikini Borrowers | 0.25
 Blindness Gun | 0.25
 Bubble Grosse | 0.25
 Apache Divine | 0.25
 American Circus | 0.25
 Aladdin Calendar | 0.25
 Caribbean Liberty | 0.25
(10 rows)
```

3. Function

Toko Pagila saat ini menjalin kerja sama dengan beberapa negara, diantaranya adalah India, Indonesia, dan Japan. Kerja sama ini akan berupaya untuk menyejahterakan staff Toko Pagila yang sering melayani rental dari customer yang berasal dari negara-negara tersebut dengan memberikan insentif tambahan dengan aturan sebagai berikut.

1. Setiap transaksi yang berasal dari negara India, maka staff akan diberikan insentif sebesar 1% dari amount transaksi.
2. Setiap transaksi yang berasal dari negara Indonesia, maka staff akan diberikan insentif sebesar 0,5% dari amount transaksi.
3. Setiap transaksi yang berasal dari negara Japan, maka staff akan diberikan insentif sebesar 2% dari amount transaksi.

Buatlah sebuah function untuk menghitung total insentif yang didapatkan oleh setiap staff dengan menerima argumen berupa staff_id.

Lalu, buatlah query untuk menampilkan keseluruhan informasi dari tabel staff dan tampilkan juga total insentif yang mereka dapatkan.

Note: Dapat menggunakan CASE WHEN

Jawaban:

Query Pembuatan Function

```
CREATE OR REPLACE FUNCTION
calculate_incentive(staff_id INT)
RETURNS NUMERIC AS $$
DECLARE
total_incentive NUMERIC := 0;
BEGIN
total_incentive := (
SELECT
SUM(
CASE
```

	<pre>WHEN country.country = 'India' THEN payment.amount * 0.01 WHEN country.country = 'Indonesia' THEN payment.amount * 0.005 WHEN country.country = 'Japan' THEN payment.amount * 0.02 ELSE 0 END) FROM payment JOIN rental ON payment.rental_id = rental.rental_id JOIN customer ON rental.customer_id = customer.customer_id JOIN address ON customer.address_id = address.address_id JOIN city ON address.city_id = city.city_id JOIN country ON city.country_id = country.country_id WHERE payment.staff_id = calculate_incentive.staff_id); RETURN total_incentive; END; \$\$ LANGUAGE plpgsql;</pre>
--	---

	<pre>prak3=# CREATE OR REPLACE FUNCTION prak3=# calculate_incentive(staff_id INT) prak3=# RETURNS NUMERIC AS \$\$ prak3\$# DECLARE prak3\$# total_incentive NUMERIC := 0; prak3\$# BEGIN prak3\$# total_incentive := (prak3\$# SELECT prak3\$# SUM(prak3\$# CASE prak3\$# WHEN country.country = 'India' THEN prak3\$# payment.amount * 0.01 prak3\$# WHEN country.country = 'Indonesia' THEN prak3\$# payment.amount * 0.005 prak3\$# WHEN country.country = 'Japan' THEN prak3\$# payment.amount * 0.02 prak3\$# ELSE 0 prak3\$# END prak3\$#) prak3\$# FROM prak3\$# payment prak3\$# JOIN rental ON payment.rental_id = prak3\$# rental.rental_id prak3\$# JOIN customer ON rental.customer_id = prak3\$# customer.customer_id prak3\$# JOIN address ON customer.address_id = prak3\$# address.address_id prak3\$# JOIN city ON address.city_id = city.city_id prak3\$# JOIN country ON city.country_id = prak3\$# country.country_id prak3\$# WHERE prak3\$# payment.staff_id = calculate_incentive.staff_id prak3\$#); prak3\$# RETURN total_incentive; prak3\$# END; prak3\$# \$\$ LANGUAGE plpgsql; CREATE FUNCTION</pre>																														
Query Penerapan Function	<pre>SELECT s.*, calculate_incentive(staff.staff_id) AS total_insentif FROM staff s;</pre>																														
SS Hasil Penerapan Function	<pre>pgsql=# SELECT s.s, hitung_incentif(s.staff_id) AS total_incentif FROM staff s;</pre> <table><thead><tr><th>staff_id</th><th>first_name</th><th>last_name</th><th>address_id</th><th>email</th><th>store_id</th><th>active</th><th>username</th><th>password</th><th>total_incentif</th></tr></thead><tbody><tr><td>1</td><td>Mike</td><td>Hillier</td><td>3</td><td>Mike.Hillier@bakilistaff.com</td><td>1</td><td>t</td><td>Mike</td><td>8db223708679ca88dbd6464eaccd8a963d55376A</td><td>64.52035</td></tr><tr><td>2</td><td>Jon</td><td>Stephens</td><td>4</td><td>Jon.Stephens@bakilistaff.com</td><td>2</td><td>t</td><td>Jon</td><td>8db223708679ca88dbd6464eaccd8a963d55376A</td><td>64.99869</td></tr></tbody></table> <p>(2 rows)</p>	staff_id	first_name	last_name	address_id	email	store_id	active	username	password	total_incentif	1	Mike	Hillier	3	Mike.Hillier@bakilistaff.com	1	t	Mike	8db223708679ca88dbd6464eaccd8a963d55376A	64.52035	2	Jon	Stephens	4	Jon.Stephens@bakilistaff.com	2	t	Jon	8db223708679ca88dbd6464eaccd8a963d55376A	64.99869
staff_id	first_name	last_name	address_id	email	store_id	active	username	password	total_incentif																						
1	Mike	Hillier	3	Mike.Hillier@bakilistaff.com	1	t	Mike	8db223708679ca88dbd6464eaccd8a963d55376A	64.52035																						
2	Jon	Stephens	4	Jon.Stephens@bakilistaff.com	2	t	Jon	8db223708679ca88dbd6464eaccd8a963d55376A	64.99869																						

4. Procedure

Toko Pagila ingin memperbaiki sistem penyewaan film mereka dengan menambahkan fitur untuk mengelola transaksi penyewaan film secara otomatis. Oleh karena itu, Toko Pagila ingin membuat prosedur untuk menangani transaksi penyewaan film, termasuk pembuatan catatan penyewaan dan pembayaran.

- Buatlah prosedur untuk menangani transaksi penyewaan film. Prosedur ini harus menerima parameter seperti tanggal sewa, id inventory, id customer, id staff, durasi sewa, dan tarif sewa (rental rate). Prosedur ini

harus secara otomatis menambahkan catatan penyewaan dan membuat catatan pembayaran apabila pembayaran dilakukan di depan (

- b. . Jumlah yang dibayar didapatkan dari $\text{rental_duration} * \text{rental_rate}$
- c. Lakukan panggilan pada prosedur tersebut dengan parameter yang sesuai. Gunakan tanggal hari ini sebagai rental date.
- d. Buat query untuk menampilkan data dari tabel rental dan payment setelah prosedur berhasil dipanggil.

Jawaban:

Query Pembuatan Prosedur	<pre>CREATE OR REPLACE PROCEDURE soal_4(tglsewa date, inventid int, customerid int, idstaf int, durasisewa int, tarifsewa numeric(4,2), upfront boolean) LANGUAGE plpgsql AS \$\$ DECLARE idrental int; idpayment int; BEGIN SELECT (max(rental_id) +1) FROM rental into Idrental; SELECT (max(payment_id)+1) FROM payment into Idpayment; INSERT INTO rental VALUES (idrental, tglsewa, inventid, customerid, tglsewa, idstaf); IF(upfront) THEN INSERT INTO payment VALUES(idpayment, customerid, idstaf, idrental, durasisewa * tarifsewa, tglsewa); END IF; END;</pre>
---------------------------------	--

SS Query Pembuatan Prosedur	<pre>prak3=# CREATE OR REPLACE PROCEDURE soal_4(tglsewa prak3(# date, inventid int, customerid int, idstaf int, durasisewa int, prak3(# tarifsewa numeric(4,2), upfront boolean) prak3-# LANGUAGE plpgsql prak3-# AS prak3-# \$\$ prak3\$# DECLARE prak3\$# idrental int; prak3\$# idpayment int; prak3\$# BEGIN prak3\$# SELECT (max(rental_id) +1) FROM rental into prak3\$# Idrental; prak3\$# prak3\$# SELECT (max(payment_id)+1) FROM payment into prak3\$# Idpayment; prak3\$# prak3\$# INSERT INTO rental VALUES (idrental, tglsewa, prak3\$# inventid, customerid, tglsewa, idstaf); prak3\$# IF(upfront) THEN prak3\$# INSERT INTO payment VALUES(idpayment, prak3\$# customerid, idstaf, idrental, durasisewa * tarifsewa, prak3\$# tglsewa); prak3\$# END IF; prak3\$# END; prak3\$# prak3\$# \$\$; CREATE PROCEDURE</pre>
Query Pemanggilan Prosedur	<pre>CALL soal_4('2024-04-02', 1, 459, 1, 10, 3.5, true);</pre>
Query untuk Menampilkan Data & SS	<pre>prak3=# CALL soal_4('2024-04-02', 1, 459, 1, 10, 3.5, true); CALL</pre> <pre>SELECT * FROM payment WHERE payment_id = (SELECT MAX(payment_id) FROM payment);</pre> <pre>prak3=# SELECT * FROM payment prak3-# WHERE payment_id = (prak3(# SELECT MAX(payment_id) FROM payment); payment_id customer_id staff_id rental_id amount payment_date -----+-----+-----+-----+-----+----- 32099 459 1 16050 35.00 2024-04-02 00:00:00 (1 row)</pre> <pre>SELECT * FROM rental WHERE rental_id = (SELECT MAX(rental_id) FROM payment);</pre> <pre>prak3=# SELECT * FROM rental prak3-# WHERE rental_id = (prak3(# SELECT MAX(rental_id) FROM payment); rental_id rental_date inventory_id customer_id return_date staff_id -----+-----+-----+-----+-----+----- 16050 2024-04-02 00:00:00 1 459 2024-04-02 00:00:00 1 (1 row)</pre>

5. Procedure

Berkurangnya attention span masyarakat masa kini membuat manajemen Toko Pagila berpikir untuk meningkatkan promosi film dengan durasi pendek. Oleh karena itu, Toko Pagila ingin menandai film-film mana yang memiliki durasi kurang dari 60 menit sebagai film pendek untuk menjadi bahan marketing.

- Tambahkan atribut `is_short_movie` (boolean) pada tabel `film`. Dengan definisi durasi film pendek yang ditentukan manajemen, buatlah prosedur untuk memberikan value `is_short_movie` sesuai dengan apakah film tersebut termasuk film pendek (`true`) atau bukan (`false`).
- Lakukan panggilan pada prosedur tersebut!
- Buat query untuk menampilkan data dari tabel `film` setelah prosedur berhasil dipanggil.

Jawaban:

Query Penambahan kolom	ALTER TABLE film ADD COLUMN is_short_movie boolean;
Query Pembuatan Prosedur	CREATE OR REPLACE PROCEDURE no_5() LANGUAGE sql AS \$\$ UPDATE film Set is_short_movie = (length < 60); \$\$;
SS Query Pembuatan Prosedur	<pre>prak3=# CREATE OR REPLACE PROCEDURE no_5() prak3-# LANGUAGE sql prak3-# AS \$\$ prak3\$# prak3\$# UPDATE film prak3\$# Set is_short_movie = (length < 60); prak3\$# \$\$ prak3-# ; CREATE PROCEDURE Time: 65,832 ms</pre>
Query Pemanggilan Prosedur	CALL no_5();
Query untuk Menampilkan Data & SS	SELECT title, length, is_short_movie FROM film;

prak3=# SELECT title,length, is_short_movie FROM film;		
title	length	is_short_movie
-----+-----+-----		
Chamber Italian	117	f
Grosse Wonderful	49	t
Opposite Necklace	92	f
Scarface Bang	102	f
Airport Pollock	54	t
Bright Encounters	73	f
Young Language	183	f
Youth Kick	179	f
Zhivago Core	105	f
Zoolander Fiction	101	f
Zorro Ark	50	t
Academy Dinosaur	86	f
Ace Goldfinger	48	t
Adaptation Holes	50	t
Affair Prejudice	117	f
African Egg	130	f
Agent Truman	169	f
Airplane Sierra	62	f
Alabama Devil	114	f
Aladdin Calendar	63	f
Alamo Videotape	126	f
Alaska Phantom	136	f
Date Speed	104	f
Women Dorado	126	f
Won Dares	105	f
Wonderful Drop	126	f
Wonderland Christmas	111	f
Wonka Sea	85	f
Words Hunter	116	f
Worker Tarzan	139	f
Working Microcosmos	74	f
World Leathernecks	171	f
Worst Banger	185	f
Wrath Mile	176	f
Wrong Behavior	178	f
Wyoming Storm	100	f
Yentl Idaho	86	f
(1000 rows)		
Time: 3,002 ms		

6. Procedure

Toko Pagila ingin membuat strategi bisnis baru berupa membership untuk meningkatkan loyalitas customer mereka. Terdapat 2 kategori membership, yaitu premium dan standard. Membership premium memiliki benefit yang lebih banyak dibandingkan dengan membership standard. Untuk memudahkan pemberian benefit, toko pagila ingin melakukan penambahan atribut membership pada tabel customer. Customer yang termasuk dalam membership premium merupakan customer yang memiliki total amount payment di atas 160.00.

- Tambahkan atribut `membership_category` (varchar (25)) pada tabel `customer`. Buatlah prosedur untuk memberikan value `membership_category` sesuai dengan kategori membership (premium / standard) dari customer terkait.
- Lakukan panggilan pada prosedur tersebut.
- Buat query untuk menampilkan data dari tabel `customer` setelah prosedur berhasil dipanggil. Tampilkan juga data customer dengan kategori membership premium.

Jawaban:

Query Penambahan kolom	<code>ALTER TABLE customer ADD COLUMN membership_category VARCHAR(25);</code>
Query Pembuatan Prosedur	<pre>CREATE OR REPLACE PROCEDURE update_membership() AS \$\$ BEGIN UPDATE customer SET membership_category = CASE WHEN (SELECT SUM(amount) FROM payment WHERE customer_id = customer.customer_id) > 160.00 THEN 'premium' ELSE 'standard' END; END; \$\$ LANGUAGE plpgsql;</pre>
SS Query Pembuatan Prosedur	<pre>prak3=# CREATE OR REPLACE PROCEDURE update_membership() prak3=# AS \$\$ prak3\$# BEGIN prak3\$# UPDATE customer prak3\$# SET membership_category = prak3\$# CASE WHEN (prak3\$# SELECT SUM(amount) prak3\$# FROM payment prak3\$# WHERE customer_id = customer.customer_id prak3\$#) > 160.00 THEN 'premium' prak3\$# ELSE 'standard' prak3\$# END; prak3\$# END; prak3\$# \$\$ LANGUAGE plpgsql; CREATE PROCEDURE</pre>
Query Pemanggilan Prosedur	<pre>CALL update_membership();</pre> <pre>prak3=# CALL update_membership(); CALL prak3=#</pre>
Query untuk Menampilkan Data &	<code>SELECT * FROM customer LIMIT 20;</code>

SS

```
prak3=# SELECT * FROM customer LIMIT 20;
```

customer_id	first_name	last_name	email	address_id	active	membership_category
524	Jared	Ely	jared.ely@sakilacustomer.org	530	t	standard
1	Mary	Smith	mary.smith@sakilacustomer.org	5	t	standard
2	Patricia	Johnson	patricia.johnson@sakilacustomer.org	6	t	standard
3	Linda	Williams	linda.williams@sakilacustomer.org	7	t	standard
4	Barbara	Jones	barbara.jones@sakilacustomer.org	8	t	standard
5	Elizabeth	Brown	elizabeth.brown@sakilacustomer.org	9	t	standard
6	Jennifer	Davis	jennifer.davis@sakilacustomer.org	10	t	standard
7	Maria	Miller	maria.miller@sakilacustomer.org	11	t	standard
8	Susan	Wilson	susan.wilson@sakilacustomer.org	12	t	standard
9	Margaret	Moore	margaret.moore@sakilacustomer.org	13	t	standard
10	Dorothy	Taylor	dorothy.taylor@sakilacustomer.org	14	t	standard
11	Lisa	Anderson	lisa.anderson@sakilacustomer.org	15	t	standard
12	Nancy	Thomas	nancy.thomas@sakilacustomer.org	16	t	standard
13	Karen	Jackson	karen.jackson@sakilacustomer.org	17	t	standard
14	Betty	White	betty.white@sakilacustomer.org	18	t	standard
15	Helen	Harris	helen.harris@sakilacustomer.org	19	t	standard
17	Donna	Thompson	donna.thompson@sakilacustomer.org	21	t	standard
18	Carol	Garcia	carol.garcia@sakilacustomer.org	22	t	standard
19	Ruth	Martinez	ruth.martinez@sakilacustomer.org	23	t	standard
20	Sharon	Robinson	sharon.robinson@sakilacustomer.org	24	t	standard

(20 rows)

SELECT * FROM customer WHERE
membership_category = 'premium';

```
prak3=# SELECT * FROM customer WHERE membership_category = 'premium';
```

customer_id	first_name	last_name	email	address_id	active	membership_category
137	Rhonda	Kennedy	rhonda.kennedy@sakilacustomer.org	141	t	premium
144	Clara	Shaw	clara.shaw@sakilacustomer.org	148	t	premium
148	Eleanor	Hunt	eleanor.hunt@sakilacustomer.org	152	t	premium
178	Marion	Snyder	marion.snyder@sakilacustomer.org	182	t	premium
181	Ana	Bradley	ana.bradley@sakilacustomer.org	185	t	premium
236	Marcia	Dean	marcia.dean@sakilacustomer.org	240	t	premium
403	Mike	Way	mike.way@sakilacustomer.org	408	t	premium
410	Curtis	Irbby	curtis.irby@sakilacustomer.org	415	t	premium
459	Tommy	Collazo	tommy.collazo@sakilacustomer.org	464	t	premium
522	Arnold	Havens	arnold.havens@sakilacustomer.org	528	t	premium
526	Karl	Seal	karl.seal@sakilacustomer.org	532	t	premium

(11 rows)

III. Pembagian Tugas

NIM	Nama	Tugas
18222043	Ricky Wijaya	