

Nomor Kelompok : **05**

Kelas : **K01**

NIM : **18222013**

Nama : **Aththariq Lisan**

NIM : **18222087**

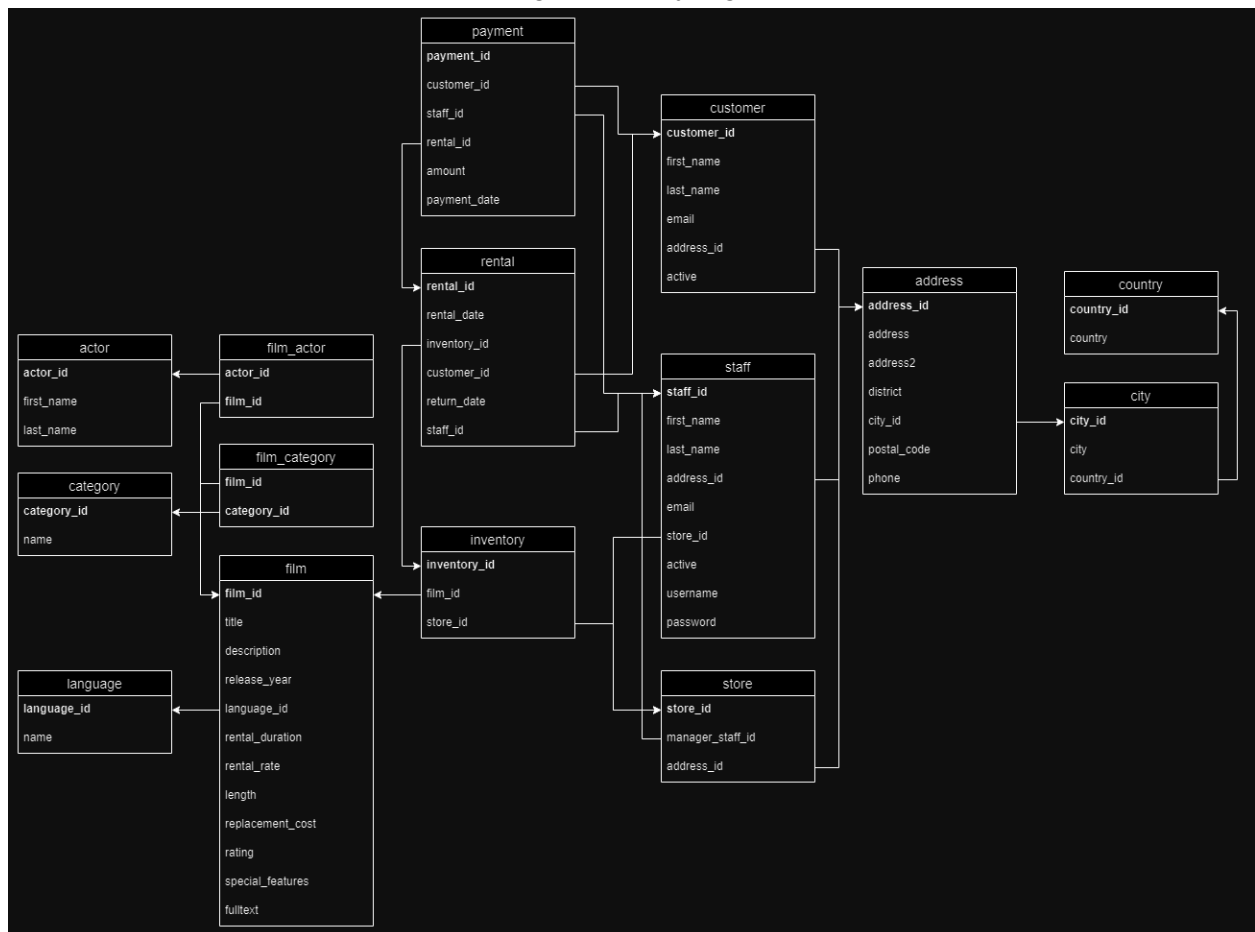
Nama : **Satria Wisnu Wibowo**

Lembar Kerja Praktikum 5 II2250 Manajemen Basis Data STI

Materi: *Security & Transaction*

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. Role

Pemilik Toko Pagila mempekerjakan HRD yang bertanggung jawab untuk mengatur administrasi kepegawaian. Salah satu tugas khusus yang diberikan kepada HRD adalah memberikan hampers kepada setiap staff yang berhasil mencapai target tertentu. Untuk itu, HRD memerlukan akses terhadap nama lengkap staff dan alamat lengkap rumahnya untuk keperluan pengiriman hampers.

- Buatlah role bernama **hrd** yang memiliki akses sesuai dengan keperluan pengiriman hampers.
- Buatlah sebuah user dengan nama **hrd1** dengan password **maungirimhampers** lalu berikan akses role **hrd** kepada user tersebut.
- Login pada database sebagai user **hrd1** dan coba lakukan pencarian data nama dan alamat untuk staff yang memiliki id = 1.

HINT

- command login: `psql -U <user> <database> -h localhost`
- Dapat dibuat view untuk mengisolasi data yang perlu diakses (tuliskan query pembuatannya)

Jawaban:

Query Pembuatan Role dan Pemberian Akses Role (A)	<pre>CREATE ROLE hrd; GRANT SELECT ON staff TO hrd; GRANT SELECT ON address TO hrd; GRANT SELECT ON city TO hrd; GRANT SELECT ON country TO hrd;</pre>
Query Pembuatan User dan Pemberian Akses Role (B)	<pre>CREATE USER hrd1 LOGIN PASSWORD 'maungirimhampers' IN ROLE hrd;</pre>
Query pencarian data nama	<pre>SELECT concat(first_name, ' ', last_name), address, address2, district, city, country, postal_code, phone</pre>

dan alamat staff (C)	FROM staff NATURAL JOIN address NATURAL JOIN city NATURAL JOIN country WHERE staff_id = 1;
SS Hasil Query (C)	<pre> labdas=# CREATE ROLE hrd; GRANT SELECT ON staff TO hrd; GRANT SELECT ON address TO hrd; GRANT SELECT ON city TO hrd; GRANT SELECT ON country TO hrd; CREATE ROLE GRANT GRANT GRANT GRANT labdas=# CREATE USER hrd1 LOGIN PASSWORD 'maungirimhampers' IN ROLE hrd; CREATE ROLE labdas=> SELECT concat(first_name,' ',last_name), address, address2, district, city, country, postal_code, phone FROM staff NATURAL JOIN address NATURAL JOIN city NATURAL JOIN country WHERE staff_id = 1; concat address address2 district city country postal_code phone ----- ----- ----- ----- ----- ----- ----- ----- Mike Hillyer 23 Workhaven Lane Alberta Lethbridge Canada 14033335568 (1 row) </pre>

2. Role

Salah satu peran staff yang paling penting pada keberjalanan bisnis di Toko Pagila adalah seorang kasir. Seorang kasir memiliki tugas untuk **mencatat** dan **mengecek data rental baru** beserta **data pembayarannya**, namun tidak memiliki wewenang untuk menghapus maupun membarui data yang ada di dalam tabel 'Rental' maupun 'Payment'.

- Buatlah sebuah *role* dengan nama **cashier1** yang memiliki password **kAsiR2024**. Role ini memiliki akses untuk melihat dan menambahkan data pada terhadap tabel Rental. Selain itu, role ini juga memiliki akses untuk melihat dan menambahkan data pada tabel Payment. Setelah itu, login pada database sebagai user **cashier1**.
- Lakukan penambahan data pada tabel Rental (data yang diinput dibebaskan pada peserta praktikum). Setelah itu, hapus data yang baru saja Anda tambahkan.
- Tampilkan 10 data pada tabel Payment sebagai user **cashier1**. Setelah itu, hapus row data dengan payment_id = 24240.

Jawaban:

Query Pembuatan Role (A)	<pre> CREATE ROLE cashier1 LOGIN PASSWORD 'kAsiR2024'; GRANT SELECT, INSERT ON rental TO cashier1; GRANT SELECT, INSERT ON payment TO cashier1; </pre>
--------------------------	--

Query pada Tabel Rental (B)	<pre>INSERT INTO rental(rental_id, rental_date, inventory_id, customer_id, return_date, staff_id) VALUES (100000, '2006-05-24 22:54:33', 1525, 459, '2006-05-28 19:40:33', 1); DELETE FROM rental WHERE rental_id = 100000;</pre>
SS Query pada Tabel Rental (B)	<pre>labdas=> INSERT INTO rental(rental_id, rental_date, inventory_id, customer_id, return_date, staff_id) VALUES (100000, '2006-05-24 22:54:33', 1525, 459, '2006-05-28 19:40:33', 1); INSERT 0 1 labdas=> labdas=> DELETE FROM rental labdas-> WHERE rental_id = 100000; ERROR: permission denied for table rental labdas=></pre>
Query pada Tabel Payment (C)	<pre>SELECT * FROM payment LIMIT 10; DELETE FROM payment WHERE payment_id = 24240;</pre>
SS Query pada Tabel Payment (C)	<pre>labdas=> SELECT * FROM payment LIMIT 10; payment_id customer_id staff_id rental_id amount payment_date ----- 17503 341 2 1520 7.99 2007-02-15 22:25:46.996577 17504 341 1 1778 1.99 2007-02-16 17:23:14.996577 17505 341 1 1849 7.99 2007-02-16 22:41:45.996577 17506 341 2 2829 2.99 2007-02-19 19:39:56.996577 17507 341 2 3130 7.99 2007-02-20 17:31:48.996577 17508 341 1 3382 5.99 2007-02-21 12:33:49.996577 17509 342 2 2190 5.99 2007-02-17 23:58:17.996577 17510 342 1 2914 5.99 2007-02-20 02:11:44.996577 17511 342 1 3081 2.99 2007-02-20 13:57:39.996577 17512 343 2 1547 4.99 2007-02-16 00:10:50.996577 (10 rows) labdas=> DELETE FROM payment WHERE payment_id = 24240; ERROR: permission denied for table payment labdas=></pre>

3. Role

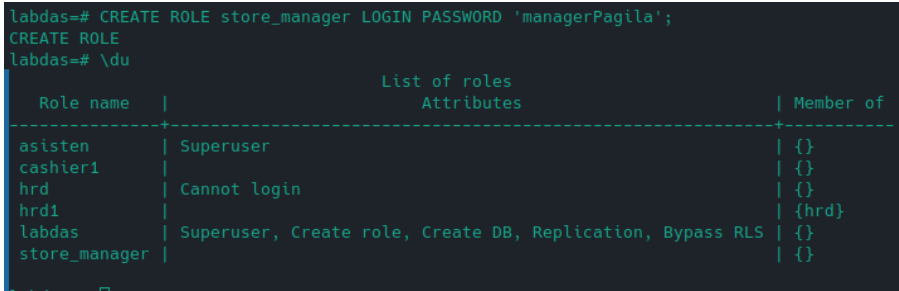
Di toko ini juga terdapat seorang **Store Manager** yang bertanggungjawab untuk mengelola toko. Beberapa tugas **Store Manager** yaitu menyesuaikan stok film di inventory apabila ada film yang baru masuk, serta memastikan proses rental dan payment berjalan lancar. Store Manager **hanya perlu melihat** apakah data pada **rental** dan **payment** sudah sesuai. Selain itu, setiap kali ada film baru, store manager perlu **melihat** data **film**, kemudian **menambahkan** data film baru maupun **menyesuaikan** jumlah stok film pada **inventory**.

- A. Berdasarkan deskripsi tersebut, akses privilege apa saja yang sesuai dengan deskripsi? Sebutkan nama tabel, tipe privilege, dan alasannya.

- B. Buatlah sebuah role bernama **store_manager** dengan password **managerPagila**, dan berikan akses privilege kepada role tersebut berdasarkan jawaban Anda pada bagian A.

Jawaban:

Untuk screenshot nomor B, perlihatkan role yang dibuat dengan perintah \du dan privilege yang telah dibuat dengan menggunakan perintah \dp <nama_tabel>

Tabel, Tipe Privilege dan Alasan (A)	<p>Tabel Rental: Select, karena store manager hanya perlu melihat apakah data pada rental sudah sesuai sehingga privilege yang perlu diberikan hanyalah SELECT untuk melihat data tersebut.</p> <p>Tabel Payment: Select, karena store manager hanya perlu melihat apakah data pada payment sudah sesuai sehingga privilege yang perlu diberikan hanyalah SELECT untuk melihat data tersebut.</p> <p>Tabel Film: SELECT INSERT , karena store manager perlu melihat data film yang menggunakan privilege SELECT kemudian menambahkan data film dengan privilege INSERT.</p> <p>Tabel Inventory: Update, karena store manager perlu menyesuaikan jumlah stok film pada inventory sehingga privilege yang perlu diberikan hanyalah UPDATE untuk mengubah data tersebut.</p>																												
Query Pembuatan Role & Screenshot (B)	<pre>CREATE ROLE store_manager LOGIN PASSWORD 'managerPagila';</pre>  <p>The screenshot shows the PostgreSQL command prompt with the following commands and output:</p> <pre>labdas=# CREATE ROLE store_manager LOGIN PASSWORD 'managerPagila'; labdas=# \du</pre> <table><tr><th>Role name</th><th>List of roles</th><th>Attributes</th><th>Member of</th></tr><tr><td>asisten</td><td>Superuser</td><td></td><td>{ }</td></tr><tr><td>cashier1</td><td></td><td></td><td>{ }</td></tr><tr><td>hrd</td><td>Cannot login</td><td></td><td>{ }</td></tr><tr><td>hrd1</td><td></td><td></td><td>{hrd}</td></tr><tr><td>labdas</td><td>Superuser, Create role, Create DB, Replication, Bypass RLS</td><td></td><td>{ }</td></tr><tr><td>store_manager</td><td></td><td></td><td>{ }</td></tr></table>	Role name	List of roles	Attributes	Member of	asisten	Superuser		{ }	cashier1			{ }	hrd	Cannot login		{ }	hrd1			{hrd}	labdas	Superuser, Create role, Create DB, Replication, Bypass RLS		{ }	store_manager			{ }
Role name	List of roles	Attributes	Member of																										
asisten	Superuser		{ }																										
cashier1			{ }																										
hrd	Cannot login		{ }																										
hrd1			{hrd}																										
labdas	Superuser, Create role, Create DB, Replication, Bypass RLS		{ }																										
store_manager			{ }																										
Query Pemberian Akses Privilege & Screenshot (C)	<pre>GRANT SELECT ON rental TO store_manager; GRANT SELECT ON payment TO store_manager; GRANT SELECT, INSERT ON film TO store_manager; GRANT UPDATE ON inventory TO store_manager;</pre>																												

Schema	Name	Type	Access privileges	Column privileges	Policies
public	rental	table	asisten=arwdDxt/asisten+ cashier1=ar/asisten + store_manager=r/asisten		
(1 row)					

Schema	Name	Type	Access privileges	Column privileges	Policies
public	payment	table	asisten=arwdDxt/asisten+ cashier1=ar/asisten + store_manager=r/asisten		
(1 row)					

Schema	Name	Type	Access privileges	Column privileges	Policies
public	film	table	asisten=arwdDxt/asisten + store_manager=ar/asisten		
(1 row)					

Schema	Name	Type	Access privileges	Column privileges	Policies
public	inventory	table	asisten=arwdDxt/asisten+ store_manager=w/asisten		
(1 row)					

4. Role

Toko Pagila ingin memastikan validitas pencatatan rental, oleh karena itu mereka akan menunjuk seorang **rental manager** untuk mengawasi dan mengelola data rental yang ada. Untuk itu seorang **rental manager** harus bisa **melihat data rental yang sudah ada, memperbaikinya, menghapus data rental yang tidak valid, dan juga menambahkan data rental yang baru**. Selain itu, **rental manager** juga dapat memastikan data pembayaran sesuai dengan data rental, sehingga dapat **melihat, memperbaiki, menghapus, dan menambahkan data pembayaran** sesuai data rental. **Rental manager** juga harus memastikan **semua relasi yang berhubungan dengan data rental dan data payment** harus memiliki data yang valid dengan cara melihat data pada relasi tersebut.

- Berdasarkan deskripsi tersebut, akses privilege apa saja yang diberikan kepada rental manager? Sebutkan nama tabel, tipe privilege, dan alasannya.
- Buatlah sebuah role bernama **rental_manager** dengan password stijayajayajaya, dan berikan akses privilege kepada role tersebut berdasarkan jawaban Anda pada bagian A.

Jawaban:

Untuk screenshot nomor B, perhatikan role yang dibuat dengan perintah `\du` dan privilege yang telah dibuat dengan menggunakan perintah `\dp <nama_tabel>`

Tabel, Tipe Privilege dan Alasan (A)	<p>Tabel Rental: SELECT INSERT UPDATE DELETE, karena rental manager harus bisa melihat data rental yang sudah ada, memperbaikinya, menghapus data rental yang tidak valid, dan juga menambahkan data rental yang baru.</p> <p>Tabel Payment: SELECT INSERT UPDATE DELETE, karena rental manager harus bisa melihat, memperbaiki, menghapus, dan menambahkan data pembayaran sesuai data rental.</p> <p>Tabel Inventory, Staff, Customer: SELECT, karena rental manager harus memastikan semua relasi yang berhubungan dengan data rental dan data payment yang harus memiliki data yang valid dengan cara melihat data pada relasi tersebut.</p>																								
Query Pembuatan Role & Screenshot (B)	<pre>CREATE ROLE rental_manager LOGIN PASSWORD 'stijayajayajaya';</pre> <div><pre>labdas=# CREATE ROLE rental_manager LOGIN PASSWORD 'stijayajayajaya'; CREATE ROLE labdas=# \du</pre><table><thead><tr><th>Role name</th><th>List of roles Attributes</th><th>Member of</th></tr></thead><tbody><tr><td>asisten</td><td>Superuser</td><td>{ }</td></tr><tr><td>cashier1</td><td></td><td>{ }</td></tr><tr><td>hrd</td><td>Cannot login</td><td>{ }</td></tr><tr><td>hrd1</td><td></td><td>{ {hrd}</td></tr><tr><td>labdas</td><td>Superuser, Create role, Create DB, Replication, Bypass RLS</td><td>{ }</td></tr><tr><td>rental_manager</td><td></td><td>{ }</td></tr><tr><td>store_manager</td><td></td><td>{ }</td></tr></tbody></table></div>	Role name	List of roles Attributes	Member of	asisten	Superuser	{ }	cashier1		{ }	hrd	Cannot login	{ }	hrd1		{ {hrd}	labdas	Superuser, Create role, Create DB, Replication, Bypass RLS	{ }	rental_manager		{ }	store_manager		{ }
Role name	List of roles Attributes	Member of																							
asisten	Superuser	{ }																							
cashier1		{ }																							
hrd	Cannot login	{ }																							
hrd1		{ {hrd}																							
labdas	Superuser, Create role, Create DB, Replication, Bypass RLS	{ }																							
rental_manager		{ }																							
store_manager		{ }																							
Query Pemberian Akses Privilege & Screenshot (C)	<pre>GRANT SELECT, INSERT, UPDATE, DELETE ON rental TO rental_manager;</pre> <pre>GRANT SELECT, INSERT, UPDATE, DELETE ON payment TO rental_manager;</pre> <pre>GRANT SELECT ON inventory TO rental_manager;</pre> <pre>GRANT SELECT ON staff TO rental_manager;</pre> <pre>GRANT SELECT ON customer TO rental_manager;</pre>																								

```

labdas=# \dp rental
      Access privileges
Schema | Name | Type | Access privileges | Column privileges | Policies
-----+-----+-----+-----+-----+-----
public | rental | table | asisten=arwdDxt/asisten +|                  |
        |      |      | cashier1=ar/asisten      +|                  |
        |      |      | store_manager=r/asisten  +|                  |
        |      |      | rental_manager=arwd/asisten |                  |
(1 row)

labdas=# \dp payment
      Access privileges
Schema | Name | Type | Access privileges | Column privileges | Policies
-----+-----+-----+-----+-----+-----
public | payment | table | asisten=arwdDxt/asisten +|                  |
        |      |      | cashier1=ar/asisten      +|                  |
        |      |      | store_manager=r/asisten  +|                  |
        |      |      | rental_manager=arwd/asisten |                  |
(1 row)

labdas=# \dp inventory
      Access privileges
Schema | Name | Type | Access privileges | Column privileges | Policies
-----+-----+-----+-----+-----+-----
public | inventory | table | asisten=arwdDxt/asisten +|                  |
        |      |      | store_manager=w/asisten +|                  |
        |      |      | rental_manager=r/asisten |                  |
(1 row)

labdas=# \dp staff
      Access privileges
Schema | Name | Type | Access privileges | Column privileges | Policies
-----+-----+-----+-----+-----+-----
public | staff | table | asisten=arwdDxt/asisten +|                  |
        |      |      | hrd=r/asisten          +|                  |
        |      |      | rental_manager=r/asisten |                  |
(1 row)

labdas=# \dp customer
      Access privileges
Schema | Name | Type | Access privileges | Column privileges | Policies
-----+-----+-----+-----+-----+-----
public | customer | table | asisten=arwdDxt/asisten +|                  |
        |      |      | rental_manager=r/asisten |                  |
(1 row)

```

5. Transaction

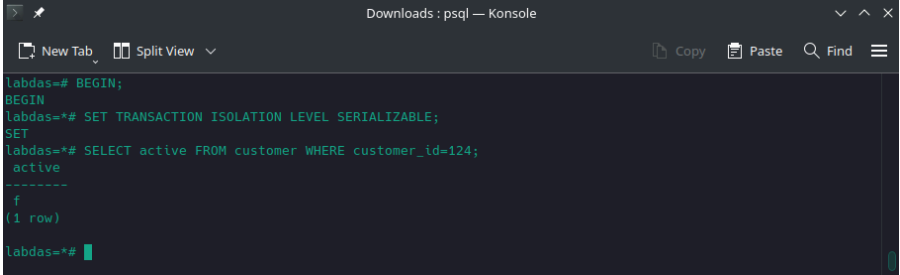
Toko Pagila memiliki 2 *staff* baru yang bertugas untuk mengatur database Toko Pagila. Terdapat *customer* dengan *customer_id* = 124 yang menelpon *store manager* untuk meminta bantuan mengubah statusnya dari tidak aktif menjadi aktif secepatnya. *Store manager* memberikan instruksi kepada *staff* A dan *staff* B untuk mengubah status *customer* tersebut. *Staff* yang lebih dahulu mengubah status *customer* tersebut akan mendapatkan bonus dari *store manager*. Dengan ambisi mendapatkan bonus, *staff* A dan *staff* B melakukan perubahan status *customer* secara bersamaan.

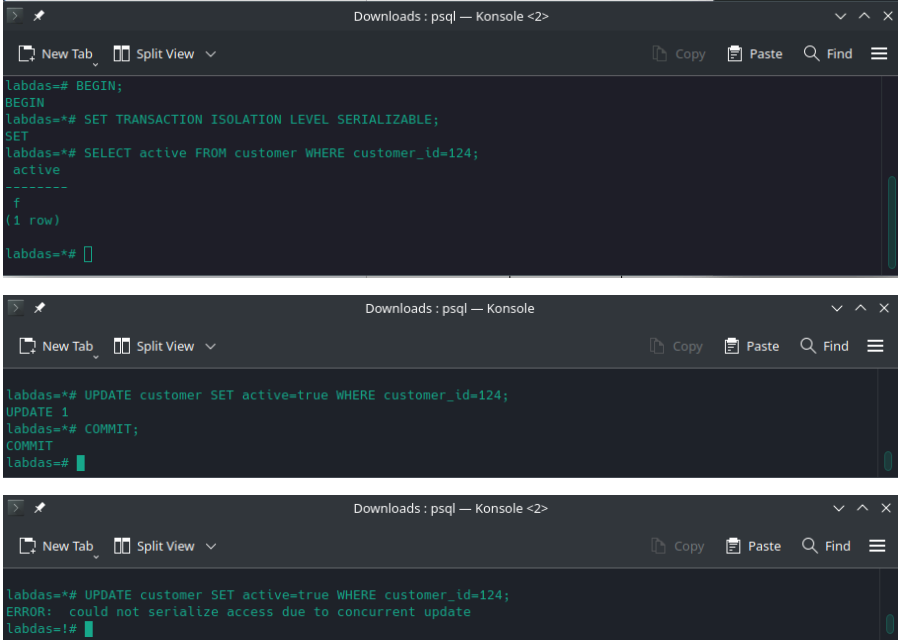
- Tentukan transaction isolation level yang cocok untuk menangani konsistensi data untuk kasus di atas, sertakan penjelasan isolation level yang dipilih.
- Lakukan simulasi SELECT dan UPDATE status active data customer dengan id 124 menjadi true secara bersamaan.

Gunakan 2 terminal untuk melakukan simulasi. (anggap masing-masing terminal sebagai 1 user yang berbeda.)

- c. Dari simulasi yang telah dilakukan. Jelaskan apa yang terjadi dalam simulasi tersebut.

Jawaban:

Isolation Level yang dipilih	Serializable
Penjelasan Isolation Level yang dipilih	<p>Dalam kasus ini, menggunakan isolation level Serializable sangat penting karena melibatkan dua transaksi yang akan mengubah data yang sama secara bersamaan. Dengan menggunakan isolation level yang lebih rendah, seperti Read Committed atau Repeatable Read, dapat menyebabkan masalah konsistensi data, seperti pembacaan yang kotor atau penulisan yang hilang. Dengan isolation level Serializable, sistem memastikan bahwa transaksi berjalan secara serial dan tidak ada pembacaan yang kotor atau penulisan yang hilang, sehingga menjaga konsistensi data dan mencegah konflik antara dua transaksi yang bersamaan, yaitu perubahan status active oleh dua staf dalam waktu yang sama.</p>
SS Simulasi	<pre>BEGIN SET TRANSACTION ISOLATION LEVEL SERIALIZABLE; SELECT active FROM customer WHERE customer_id=124; UPDATE customer SET active=true WHERE customer_id=124;</pre> 

	 <pre> labdas=# BEGIN; BEGIN labdas=# SET TRANSACTION ISOLATION LEVEL SERIALIZABLE; SET labdas=# SELECT active FROM customer WHERE customer_id=124; active ----- f (1 row) labdas=# labdas=# UPDATE customer SET active=true WHERE customer_id=124; UPDATE 1 labdas=# COMMIT; COMMIT labdas=# labdas=# UPDATE customer SET active=true WHERE customer_id=124; ERROR: could not serialize access due to concurrent update labdas=# </pre>
Penjelasan Simulasi	<p>Dengan menggunakan transaction isolation level Serializable, sistem akan memastikan bahwa transaksi dari Staff A dan Staff B dijalankan secara bergantian, sehingga hanya satu transaksi yang berhasil mengubah status customer dengan ID 124 menjadi aktif, sementara yang lainnya akan menunggu.</p> <p>Jika Staff A berhasil mengubah status customer dengan ID 124 menjadi aktif, maka ketika Staff B mencoba untuk melakukan update, transaksi Staff B akan terblokir (blocked) dan harus menunggu hingga transaksi Staff A selesai.</p> <p>Sebaliknya, jika Staff B berhasil mengubah status customer dengan ID 124 menjadi aktif, maka transaksi Staff A akan terblokir dan harus menunggu hingga transaksi Staff B selesai.</p> <p>Dengan demikian, sistem memastikan bahwa hanya satu transaksi yang berhasil mengubah status customer dengan ID 124 menjadi aktif, sementara yang lainnya ditolak untuk mencegah konflik dan menjaga konsistensi data.</p>

III. Pembagian Tugas

NIM	Nama	Tugas
18222013	Aththariq Lisan	Navigator & Driver
18222087	Satria Wisnu Wibowo	Navigator & Driver