Ahmad Habibie Marjan

18222082

**Lembar Kerja Praktikum 2**

**IF2140 Pemodelan Basis Data**

**Waktu: 100 menit**

**07.10 - 08.50 WIB**

1. **Soal**

**Bagian 1**

Kerjakan bagian ini terlebih dahulu sebelum pindah ke bagian 2. Query dan SS yang diminta sama dengan praktikum sebelumnya. Pastikan SS yang dilampirkan sesuai dengan ketentuan yang telah tercantum pada **file Panduan Praktikum 2**.

**Pada bagian I, SS minimal meliputi 5 baris awal termasuk query DAN 5 baris terakhir termasuk jumlah row**

1. Buatlah sebuah query yang menunjukkan id dan nama lengkap dari semua aktor yang pernah memainkan peran di film “Sunrise League”. DIURUT ascending berdasarkan nama lengkap aktor.

**HINT:**

* 1. Gunakan CONCAT(kolom1, <pemisah>, kolom2) untuk menggabungkan dua nilai kolom
     1. Contoh: CONCAT(animal\_1, ‘ ‘, animal\_2)
     2. Hasil: “Cat Dog”

| Query | select actor\_id, concat(first\_name,' ',last\_name) as full\_name from actor order by full\_name; |
| --- | --- |
| SS |  |

1. Gina Degeneres merupakan seorang aktor yang sukses. Ia berperan di banyak film dengan genre yang beragam. Gina Degeneres memiliki id aktor = 107. Tampilkan kategori film dan jumlah film yang memiliki kategori terkait yang pernah diperankan oleh Gina Degeneres. Urutkan secara menurun berdasarkan jumlah film masing-masing kategori!

| Query | select name as categories, count(film\_id) as total\_film from category natural join film\_category natural join film\_actor where actor\_id=107 group by name; |
| --- | --- |
| SS |  |

1. Tampilkan nama lengkap dan total melakukan penyewaan atau rental dari setiap customer yang memiliki nama diawali dengan huruf 'S'. Data yang ditampilkan hanya mencakup penyewaan yang dilakukan oleh customer tersebut antara tanggal '2005-05-26' dan '2005-05-28' (inklusif). Pastikan bahwa customer yang tidak melakukan penyewaan pada tanggal tersebut juga turut ditampilkan dengan total penyewaan 0 (jika ada). Hasil akhir diurutkan berdasarkan total penyewaan secara menurun!

**HINT:**

* 1. Gunakan CONCAT(kolom1, <pemisah>, kolom2) untuk menggabungkan dua nilai kolom
     1. Contoh: CONCAT(animal\_1, ‘ ‘, animal\_2)
     2. Hasil: “Cat Dog”

| Query | select concat(first\_name,' ', last\_name) as full\_name, count(rental\_id) as total\_rental from customer natural left outer join rental where first\_name like 'S%' group by customer\_id order by total\_rental desc; |
| --- | --- |
| SS |  |

1. Tuan Wan sedang menjalani internship di perusahaan Sakila. Tugas pertama yang diberikan oleh bosnya adalah untuk melakukan investigasi terhadap beberapa rental\_id mencurigakan. Bantulah Tuan Wan dengan menemukan rental\_id, rental\_date, dan return\_date yang memenuhi kriteria berikut:
   1. Rental tersebut dilakukan oleh customer yang memiliki nama diawali dengan huruf ‘S’ di store\_id = 2
   2. Rental tersebut dipinjam dan dikembalikan pada hari yang sama (selisih waktu peminjaman kurang dari 1 hari)

Terlepas dari 2 kriteria di atas, rental yang juga dicurigai oleh bosnya adalah rental yang dikembalikan sebelum tanggal '2005-05-26' dan ditangani oleh staff aktif (active=1). Siapkan data rental yang dicurigai untuk Tuan Wan agar bosnya senang dengan kinerja Tuan Wan dan mengangkatnya menjadi karyawan tetap!

**HINT:**

1. Gunakan Set Operation!
2. Gunakan datediff(date1, date2) yang akan menghasilkan date1-date2 atau selisih hari dari dua date tersebut

| Query |  |
| --- | --- |
| SS |  |

1. Lagi-lagi Pak Iyal kembali berulah, Ia ingin mencari nama lengkap customer yang aktif dan pernah melakukan rental film yang berkategori Sci-Fi atau Action selama lebih dari 9 hari dalam sekali peminjaman.

**(Hanya boleh menggunakan maksimal 5 tabel saja)**

**HINT**:

1. Gunakan datediff(date1, date2) yang akan menghasilkan date1-date2 atau selisih hari dari dua date tersebut
2. terdapat customer yang meminjam film yang sama berkali-kali
3. Output hanya berupa nama lengkap customer saja
4. customer aktif ditandai dengan atribut active = 1

| Query |  |
| --- | --- |
| SS |  |

1. Tampilkan judul film, durasi film, dan jumlah aktor yang bermain pada film tersebut yang memenuhi semua kriteria berikut:
   1. Memiliki nilai replacement costnya lebih besar dari beberapa nilai replacement cost film berkategori ‘Children’.
   2. Memiliki durasi (length) yang lebih besar atau sama dengan rata-rata durasi film dengan special feature ‘Behind the Scenes’ per kategorinya.

Tampilkan hanya 15 baris pertama saja yang diurutkan secara menurun berdasarkan jumlah aktornya.

**HINT:**

Gunakan limit X untuk menampilkan X baris pertama dari data yang didapat

| Query |  |
| --- | --- |
| SS |  |

1. Gunakan WITH clause untuk membuat relasi sementara **actions\_films** yang berisi id aktor dan judul film yang pernah dimainkan aktor tersebut serta memiliki kategori ‘Action’. Dengan memanfaatkan relasi sementara tersebut buatlah query untuk menampilkan **nama lengkap aktor** dan **jumlah film yang pernah dimainkan** oleh setiap aktor.

**HINT**:

1. Gunakan concat(kolom1, <string>, kolom2) untuk menggabungkan nama depan dan nama belakang
   1. Contoh: CONCAT(‘Praktikum’, ‘ ‘, ‘PBD’) menghasilkan “Praktikum PBD”)

| Query |  |
| --- | --- |
| SS |  |

**Bagian 2**

Kerjakan bagian ini setelah selesai mengerjakan bagian 1. Terdapat 3 file SS yang diminta pada bagian ini, yaitu sebelum, saat, dan sesudah melakukan modifikasi.

**Pada bagian II, SS minimal meliputi hasil query seleksi sebelum modifikasi atau pembuatan view, SS saat melakukan modifikasi atau pembuatan view, dan SS seleksi setelah modifikasi atau pembuatan view. SS saat pastikan terdapat tulisan ‘Query OK’ dan berapa rows yang dimodifikasi**

1. Hapuslah record dari film\_actor yang memiliki actor\_id lebih dari 100!

**Tunjukkan SS sebelum, SS Query yang telah dijalankan, dan SS sesudah record ditambahkan!**

**SS sebelum dan sesudah dibebaskan selama memperlihatkan perbedaan antara kondisi awal dan akhir setelah record ditambahkan**

| Query Modifikasi | delete from film\_actor where actor\_id>100; |
| --- | --- |
| Query Seleksi | select \* from film\_actor; |
| SS sebelum |  |
| SS saat modifikasi |  |
| SS sesudah |  |

1. Di data sakila, tercatat bahwa film “Lucky Flying” termasuk dalam kategori “Music”. Ternyata, film “Lucky Flying” juga termasuk dalam kategori “Comedy”. Namun, di pencatatan data Sakila, film “Lucky Flying” belum termasuk dalam kategori “Comedy”. Buatlah query untuk **menambahkan record** film “Lucky Flying” sehingga juga termasuk dalam kategori “Comedy”.

**Tunjukkan SS sebelum, SS Query yang telah dijalankan, dan SS sesudah record ditambahkan!**

***SS sebelum dan sesudah dibebaskan selama memperlihatkan perbedaan antara kondisi awal dan akhir setelah record ditambahkan!***

| Query Modifikasi |  |
| --- | --- |
| Query Seleksi |  |
| SS sebelum |  |
| SS saat modifikasi |  |
| SS sesudah |  |

1. Gani adalah salah satu pekerja toko rental film Sakila, dia terpesona oleh salah satu pelanggan yang meminjam film saat dia sedang bekerja. Gani ingin PDKT ke pelanggan tersebut, untuk itu dia ingin mengetahui film apa saja yang pelanggan tersebut suka supaya Gani bisa menikmati film yang sama.

Sayangnya, Gani hanya mengingat huruf pertama nama depan pelanggan tersebut yaitu ‘S’, huruf pertama nama terakhirnya yaitu ‘W’, huruf terakhir nama depannya yaitu ‘N’, dan huruf terakhir nama akhirnya sama yaitu ‘N’, dan bahwa pelanggan tersebut masih aktif (active = 1).

Bantu Gani memikat perhatian pelanggan tersebut dengan membuat sebuah view yang menunjukkan **customer\_id, nama lengkap, id rental, id film, dan judul film** yang pernah dirental pelanggan tersebut

* Maksimal menggunakan 3 join
* Cantumkan dua SS, satu untuk pembuatan view dan satu untuk seleksi data di view
* Cantumkan dua query, satu untuk pembuatan view dan satu untuk seleksi data di view

**HINT**:

1. Gunakan concat(kolom1, <string>, kolom2) untuk menggabungkan nama depan dan nama belakang
2. Pelanggan yang aktif memiliki nilai active = 1

| Query Pembuatan View |  |
| --- | --- |
| Query Seleksi View |  |
| SS sebelum |  |
| SS saat pembuatan |  |
| SS sesudah |  |

1. Pemilik rental sangat menyukai film dengan kategori ‘Animation’ sehingga ia tidak rela jika film tersebut dihilangkan. Untuk mencegah hal tersebut pemilik rental ingin menaikkan biaya penggantian (replacement cost) seluruh film kategori tersebut dengan kondisi:

* Naik 100% untuk biaya penggantian yang lebih kecil dari 10
* Naik 70% untuk biaya penggantian yang lebih kecil dari 20
* Naik 20% untuk selebihnya

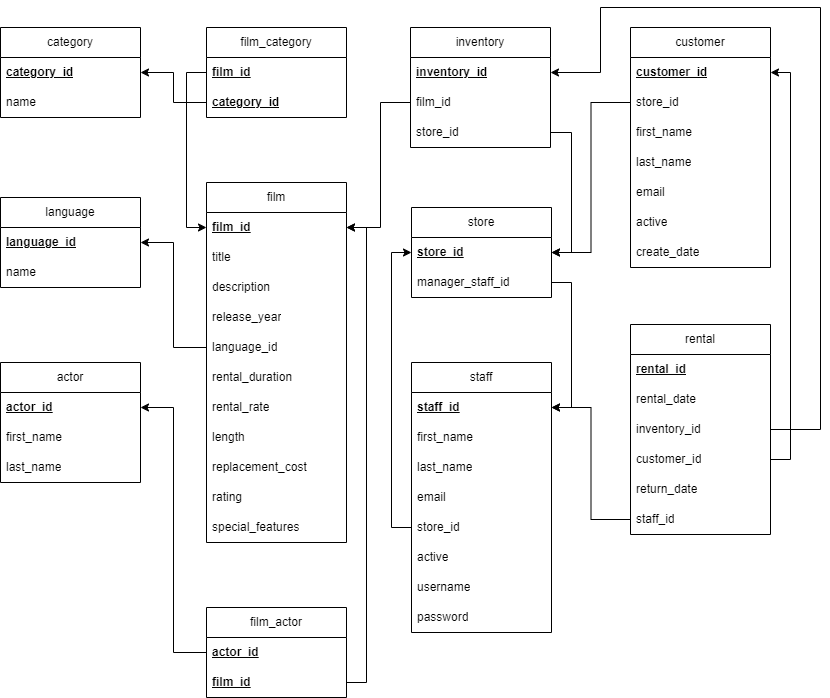
Bantulah pemilik rental untuk mengupdate database tersebut.

**Tunjukkan SS sebelum, SS Query yang telah dijalankan, dan SS sesudah record ditambahkan!**

***SS sebelum dan sesudah dibebaskan selama memperlihatkan perbedaan antara kondisi awal dan akhir setelah record ditambahkan***

| Query Modifikasi |  |
| --- | --- |
| Query Seleksi |  |
| SS sebelum |  |
| SS saat modifikasi |  |
| SS sesudah |  |

1. **Skema Basis Data**



| actor = (**actor\_id**, first\_name, last\_name)  film\_actor = (**film\_id, actor\_id**)  film = (**film\_id**, title, description, release\_year, language\_id, rental\_duration, rental\_rate, length, replacement\_cost, rating, special\_features)  film\_category = (**film\_id**, **category\_id**)  category = (**category\_id**, name)  inventory = (**inventory\_id**, film\_id, store\_id)  store = (**store\_id**, manager\_staff\_id)  staff = (**staff\_id**, first\_name, last\_name, email, store\_id, active, username, password)  rental = (**rental\_id**, rental\_date, inventory\_id, customer\_id, return\_date, staff\_id)  customer = (**customer\_id**, store\_id, first\_name, last\_name, email, active, create\_date) |
| --- |