**IF2240 Basis Data**

**Semester II 2020/2021**

Ujian Praktikum

Dipersiapkan oleh:

Asisten Laboratorium Basis Data

Pelaksanaan:

Waktu Mulai: Jumat, 26 Februari 2020 – 16.15 WIB

Waktu Selesai: Jumat, 26 Februari 2020 – 18.15 WIB

**I. Deskripsi Sistem**

Sebuah bioskop ingin menganalisis data yang mereka miliki sehingga mereka bisa menggunakannya untuk memberikan promosi dan marketing yang lebih tepat. Untuk itu, mereka memasukkan semua data pelanggan, film, infrastruktur (studio dan kursi), dan penayangan-penayangan film tersebut ke dalam suatu basis data yang terintegrasi.

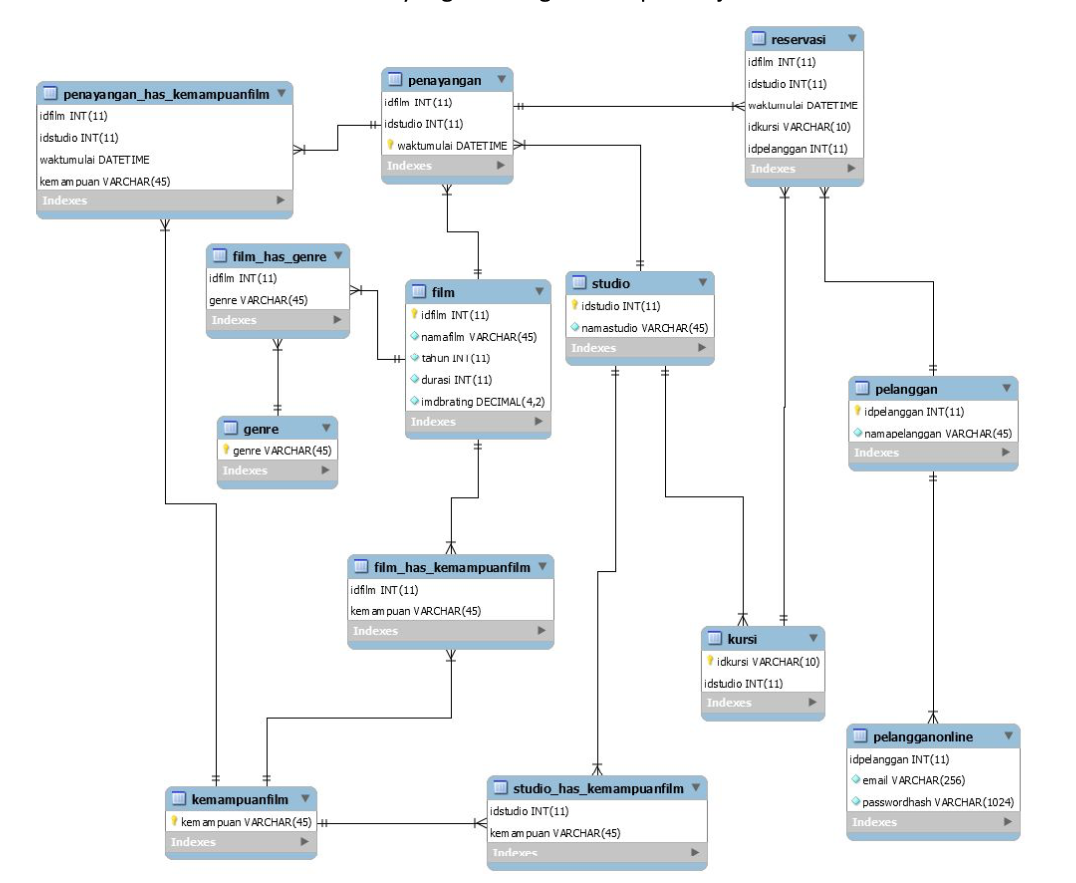
Pada data film, disimpan nama, tahun, durasi, dan rating IMDB-nya. Film juga dapat memiliki beberapa genre. Pada data studio, disimpan nama dan ID-nya, sedangkan untuk kursi, hanya disimpan ID-nya. Pada data pelanggan disimpan namanya. Jika pelanggan tersebut adalah pelanggan yang mendaftar secara online, disimpan juga email dan password-nya (dalam bentuk hash). Pada data penayangan, disimpan waktu mulainya, beserta film dan studio yang digunakan untuk penayangan.

Setiap film, studio, dan penayangan bisa ditandai memiliki suatu kemampuan khusus, seperti 3D dan 60FPS. Suatu penayangan bisa memiliki kemampuan tertentu hanya jika studio dan filmnya memiliki kemampuan tersebut juga.

Suatu pelanggan bisa melakukan reservasi terhadap beberapa kursi pada suatu penayangan. Reservasi tersebut disimpan sebagai hubungan antara pelanggan, kursi, dan penayangan.

**II. Skema Basis Data**

Berikut skema relasional basis data berdasarkan deskripsi sistem yang telah dijabarkan pada bagian sebelumnya.



|  |
| --- |
| film = (idfilm, namafilm, tahun, durasi, imdbrating)  film\_has\_genre = (idfilm, genre)  film\_has\_kemampuanfilm = (idfilm, kemampuan)  genre = (genre)  kemampuanfilm = (kemampuan)  kursi = (idkursi, idstudio)  pelanggan = (idpelanggan, namapelanggan)  pelangganonline = (idpelanggan, email, passwordhash)  studio = (idstudio, namastudio)  studio\_has\_kemampuanfilm = (idstudio, kemampuan)  reservasi = (idfilm, idstudio, waktumulai, idkursi, idpelanggan)  penayangan = (idfilm, idstudio, waktumulai)  penayangan\_has\_kemampuanfilm = (idfilm, idstudio, waktumulai, kemampuan) |

**III. Permasalahan**

Buatlah sebuah basis data lalu masukkan (*import*) script SQL uprak.sql yang telah diberikan bersama deskripsi tugas ini.

**IV. Soal**

Lakukan hal-hal berikut.

1. Buatlah query untuk menampilkan berapa jumlah kursi yang direservasi per bulan, urutkan dari bulan yang memiliki reservasi kursi paling banyak.

Query

|  |
| --- |
| select count(idKursi) as jumlah\_kursi, month(waktumulai) as bulan from reservasi r group by month(waktumulai) order by jumlah\_kursi desc; |

Hasil Screenshot

|  |
| --- |
|  |

1. Di bioskop, tiket hanya dapat dipesan melalui reservasi. Buatlah query yang dapat menampilkan genre film yang dirilis selepas tahun 1995 sebagai “Genre” dan jumlah tiket film terjual dari genre tersebut sebagai “Tiket Terjual”!

Query

|  |
| --- |
| select genre as Genre, count(idfilm) as Tiket\_Terjual from  film\_has\_genre g natural inner join film f natural inner join penayangan p natural inner join reservasi r  where tahun > 1995  group by Genre; |

Hasil Screenshot

|  |
| --- |
|  |

1. Buatlah query untuk menampilkan id, nama, dan jumlah kursi dari studio yang memiliki kemampuan 3D dan memiliki jumlah kursi lebih dari 150

Query

|  |
| --- |
| select idstudio, namastudio, count(idkursi) as jumlah\_kursi  from studio s natural inner join kursi k natural inner join studio\_has\_kemampuanfilm shk  where kemampuan = “3D”  group by idstudio  having count(idkursi) > 150; |

Hasil Screenshot

|  |
| --- |
|  |

1. Buatlah query untuk menampilkan nama, tahun rilis, durasi, rating, dan jumlah penonton dari film yang dirilis sebelum tahun reformasi, memiliki rating di atas 7, dan telah ditonton lebih dari 352 orang. Urutkan dari film yang memiliki rating paling tinggi

Query

|  |
| --- |
| select namafilm, tahun, durasi, imdbrating, count(idkursi) as jumlah\_penonton  from film f natural inner join penayangan p natural inner join reservasi r  where imdbrating > 7 and tahun < 1998  group by idfilm  having count(idkursi) > 352  order by imdbrating desc; |

Hasil Screenshot

|  |
| --- |
|  |

1. Buatlah sebuah query untuk menampilkan id film yang ditayangkan di studio 10 hingga studio 19 pada bulan januari hingga september tahun 2015 dan memiliki rating lebih dari 6. Gunakanlah set operation!

Query

|  |
| --- |
| (select idfilm  from film f natural inner join penayangan p natural inner join studio s  where idstudio in (select idstudio from studio where idstudio < 19 and idstudio > 8)  and month(waktumulai) >= 1 and month(waktumulai) <= 9)  intersect  (select idfilm  from film f  where tahun = 2015 and imdbrating > 6); |

Hasil Screenshot

|  |
| --- |
|  |

1. Buatlah sebuah view bernama “marcus\_movie” yang menampilkan judul film sebagai “Judul”, rating IMDB sebagai “Rating Film”, dan tahun rilis sebagai “Tahun” dari film-film yang pernah dilihat oleh pelanggan yang memiliki nama depan Marcus! Tunjukkan juga bahwa view berhasil dibuat!

Query

|  |
| --- |
| create view marcus\_movie as  select distinct namafilm as Judul, imdbrating as “Rating Film”, tahun as Tahun  from film f natural inner join penayangan p natural inner join reservasi r natural inner join pelanggan  where namapelanggan like “Marcus %”;  select \* from marcus\_movie; |

Hasil Screenshot

|  |
| --- |
|  |

1. Satu film dapat memiliki lebih dari satu genre. Buatlah View dengan nama ‘above\_avg\_genre\_penayangan’ yang menampilkan genre sebagai ‘nama\_genre’ dan jumlah penayangannya sebagai ‘jumlah penayangan’ untuk genre film yang memiliki jumlah penayangan di atas rata-rata. Tampilkan hasil akhir viewnya!

Query

|  |
| --- |
| create view above\_avg\_genre\_penayangan as  select genre as nama\_genre, count(p.idfilm) as jumlah\_penayangan  from film f natural inner join film\_has\_genre fhg natural inner join penayangan p  group by genre  having count(p.idfilm) > avg(jumlah\_penayangan);  select \* from above\_avg\_genre\_penayangan; |

Hasil Screenshot

|  |
| --- |
|  |

1. Veronica Doheny selalu membuat sebuah reservasi dengan pacarnya dengan menonton film yang tayang di tahun baru. Tetapi, mereka tidak akan menonton film yang memiliki rating dibawah 7. Namun, pada pada malam tahun baru, selalu saja terdapat bug sehingga transaksi yang ia lakukan tidak tercatat. Bantu pegawai untuk menambahkan record yang hilang tersebut. (Kerjakan hanya dengan 1 query saja, untuk letak kursi dibebaskan asalkan bersebelahan dengan asumsi seluruh kursi masih kosong)

Query

|  |
| --- |
| insert into reservasi  (select idfilm, idstudio, waktumulai, 1, idpelanggan  from penayangan natural inner join film natural inner join pelanggan  where namapelanggan = “Veronica Doheny” and imdbrating >= 7 and month(waktumulai) = 12 and day(waktumulai) = 31  union  select idfilm, idstudio, waktumulai, 2, idpelanggan  from penayangan natural inner join film natural inner join pelanggan  where namapelanggan = “Veronica Doheny” and imdbrating >= 7 and month(waktumulai) = 12 and day(waktumulai) = 31); |

Hasil Screenshot

|  |
| --- |
|  |

1. IMDB ingin memperbarui rating beberapa film berdasarkan jumlah penontonnya. Buatlah query untuk mengupdate rating film yang memiliki jumlah penonton diatas 500 dan rating lebih kecil dari 6.00 menjadi 6.00. Tunjukkan bahwa tidak ada lagi film yang memiliki penonton diatas 500 yang memiliki rating dibawah 6.00.

Query

|  |
| --- |
| update film  set imdbrating = 6.00  from penayangan p natural inner join reservasi r  where imdbrating < 6.00  group by idfilm  having count(idkursi) > 500;  select \* from film f natural inner join penayangan p natural inner join reservasi r  having count(idkursi) > 500; |

Hasil Screenshot

|  |
| --- |
|  |

1. Buatlah query untuk memperbarui nama studio yang mempunyai jumlah penayangan sebanyak 39 menjadi “Studio Basdat”

Query

|  |
| --- |
| update studio  set namastudio = “Studio Basdat”  where idstudio in (select idstudio from penayangan group by idstudio having count(idstudio = 39)); |

Hasil Screenshot

|  |
| --- |
|  |

1. Buatlah query untuk menghapus data reservasi pada studio yang bernama “Studio 1” dan pelanggan bernama “Wm Yaple”

Query

|  |
| --- |
| delete from reservasi  where  idstudio in (select idstudio from studio where namastudio = “Studio 1”)  and  idpelanggan in (select idpelanggan from pelanggan where namapelanggan = “Wm Yaple”); |

Hasil Screenshot

|  |
| --- |
|  |

1. Buatlah sebuah table karyawan yang berisi id karyawan, nama, id studio. Perhatikan kolom yang harus direferensikan ke table lain.

Query

|  |
| --- |
| create table if not exists karyawan(  idkaryawan int(10) primary key,  nama varchar(50),  idstudio int(11),  constraint FK\_karyawan  foreign key(idstudio) references studio(idstudio)  on update cascade on delete cascade); |

Hasil Screenshot

|  |
| --- |
|  |