**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.

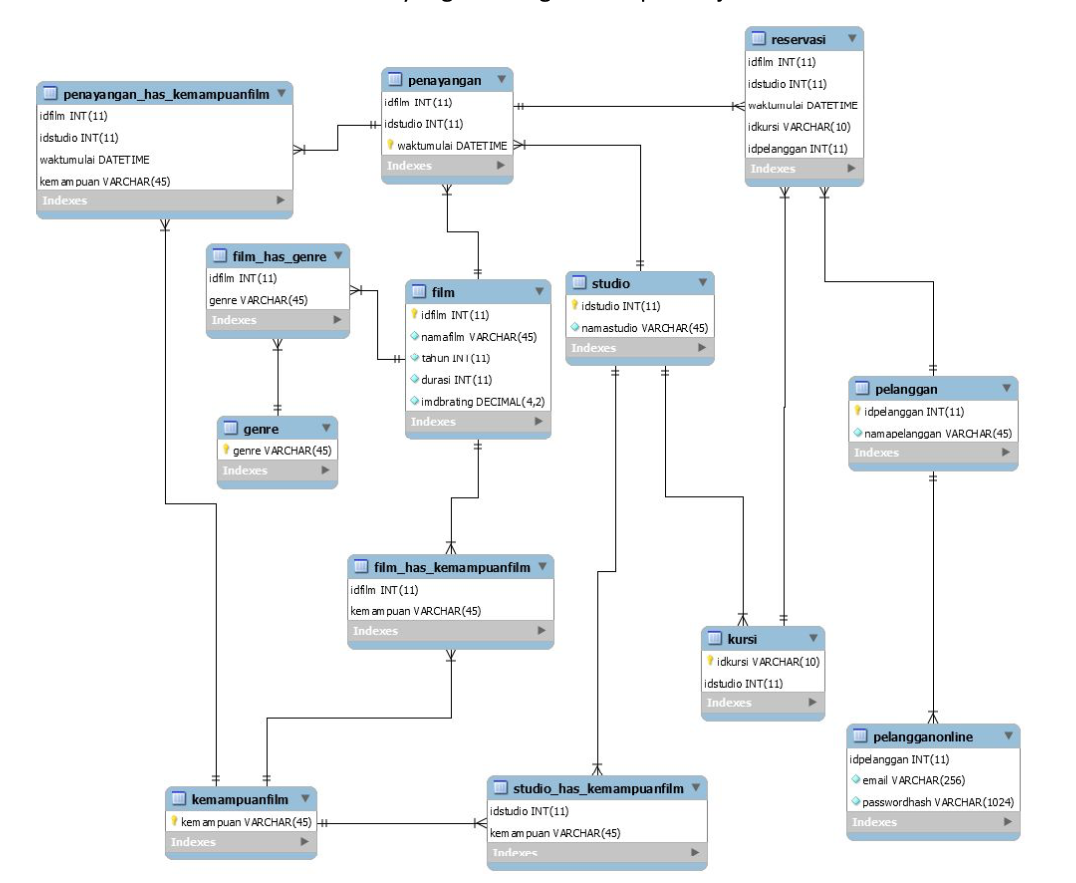
Pada data film, disimpan nama, tahun, durasi, dan rating IMDB-nya. Film juga dapat memiliki beberapa genre. Pada data studio, disimpan nama dan ID-nya, sedangkan untuk kursi, hanya disimpan ID-nya. Pada data pelanggan disimpan namanya. Jika pelanggan tersebut adalah pelanggan yang mendaftar secara online, 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

|  |
| --- |
| Asumsi : Tanpa perlu menghitung per tahunnya juga  SELECT count(idkursi),Month(waktumulai) FROM reservasi GROUP BY Month(waktumulai) ORDER BY count(idkursi) 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

|  |
| --- |
| Genre yang dirilis setelah tahun 1995 hanya music , sisanya pernah rilis di tahun sebelumnya  With genre1995 as (SELECT genre,min(tahun) FROM film NATURAL JOIN film\_has\_genre Group By genre Having min(tahun) > 1995 )  SELECT genre as Genre, count(idpelanggan) as Tiket FROM  film NATURAL JOIN film\_has\_genre NATURAL JOIN reservasi WHERE tahun >1995 and genre in (SELECT genre FROM genre1995) GROUP BY genre ; |

Hasil Screenshot

|  |
| --- |
| With genre1995 as (SELECT genre,min(tahun) FROM film NATURAL JOIN film\_has\_genre Group By genre Having min(tahun) > 1995 )  SELECT genre as Genre, count(idpelanggan) as Tiket FROM  film NATURAL JOIN film\_has\_genre NATURAL JOIN reservasi WHERE tahun >1995 and genre in (SELECT genre FROM genre1995) GROUP BY genre ; |

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

Query

|  |
| --- |
| WITH 3d as (SELECT DISTINCT idstudio FROM studio NATURAL JOIN studio\_has\_kemampuanfilm WHERE kemampuan ="3D")  SELECT idstudio, namastudio, count(idkursi) FROM studio NATURAL JOIN kursi WHERE idstudio in (SELECT idstudio FROM 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(idpelanggan) FROM film NATURAL JOIN reservasi WHERE imdbrating > 7 and tahun <1998 GROUP BY idfilm ; |

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 NATURAL JOIN penayangan WHERE idstudio between 10 and 19)  INTERSECT  (SELECT idfilm FROM film WHERE imdbrating > 6)  INTERSECT  (SELECT idfilm FROM film NATURAL JOIN penayangan WHERE waktumulai Between"2015-1-1" and "2015-9-30"); |

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 NATURAL JOIN reservasi NATURAL JOIN pelanggan WHERE namapelanggan LIKE "Marcus %" or namapelanggan = "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

|  |
| --- |
| SELECT genre as nama\_genre, count(idstudio) as jumlah\_penayangan FROM film NATURAL JOIN film\_has\_genre NATURAL JOIN penayangan GROUP BY nama\_genre HAVING jumlah\_penayangan >(SELECT avg(jumlah\_penayangan) FROM(SELECT genre as nama\_genre, count(idstudio) as jumlah\_penayangan FROM film NATURAL JOIN film\_has\_genre NATURAL JOIN penayangan GROUP BY nama\_genre) as jar) ; |

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 \* FROM(  (SELECT idfilm,idstudio,waktumulai,"B1" as idkursi, idpelanggan from penayangan,pelanggan WHERE month(waktumulai)=01 and day(waktumulai)=01 and namapelanggan="Veronica Doheny")  Union  (SELECT idfilm,idstudio,waktumulai,"B2" as idkursi, idpelanggan from penayangan,pelanggan WHERE month(waktumulai)=12 AND day(waktumulai)=31 and namapelanggan="Veronica Doheny")); |

Hasil Screenshot

|  |
| --- |
| Gatau Error Kenapa ☺ |

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 WHERE idfilm in(  SELECT idfilm FROM film NATURAL JOIN reservasi GROUP BY idfilm HAVING count(idpelanggan) > 500) and imdbrating <6.00;  SELECT idfilm,imdbrating FROM film NATURAL JOIN reservasi GROUP BY idfilm HAVING count(idpelanggan) > 500 and imdbrating <6.00; |

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(waktumulai)=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 idpelanggan=(SELECT idpelanggan FROM pelanggan WHERE namapelanggan = "WM YAPLE") and idstudio=(SELECT idstudio FROM studio WHERE namastudio="Studio 1"); |

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 karyawan(  idkaryawan INT,  nama varchar(45) charset utf8,  idstudio INT,  PRIMARY KEY(idkaryawan),  FOREIGN KEY(idstudio) REFERENCES studio(idstudio)  ); |

Hasil Screenshot

|  |
| --- |
|  |