

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.

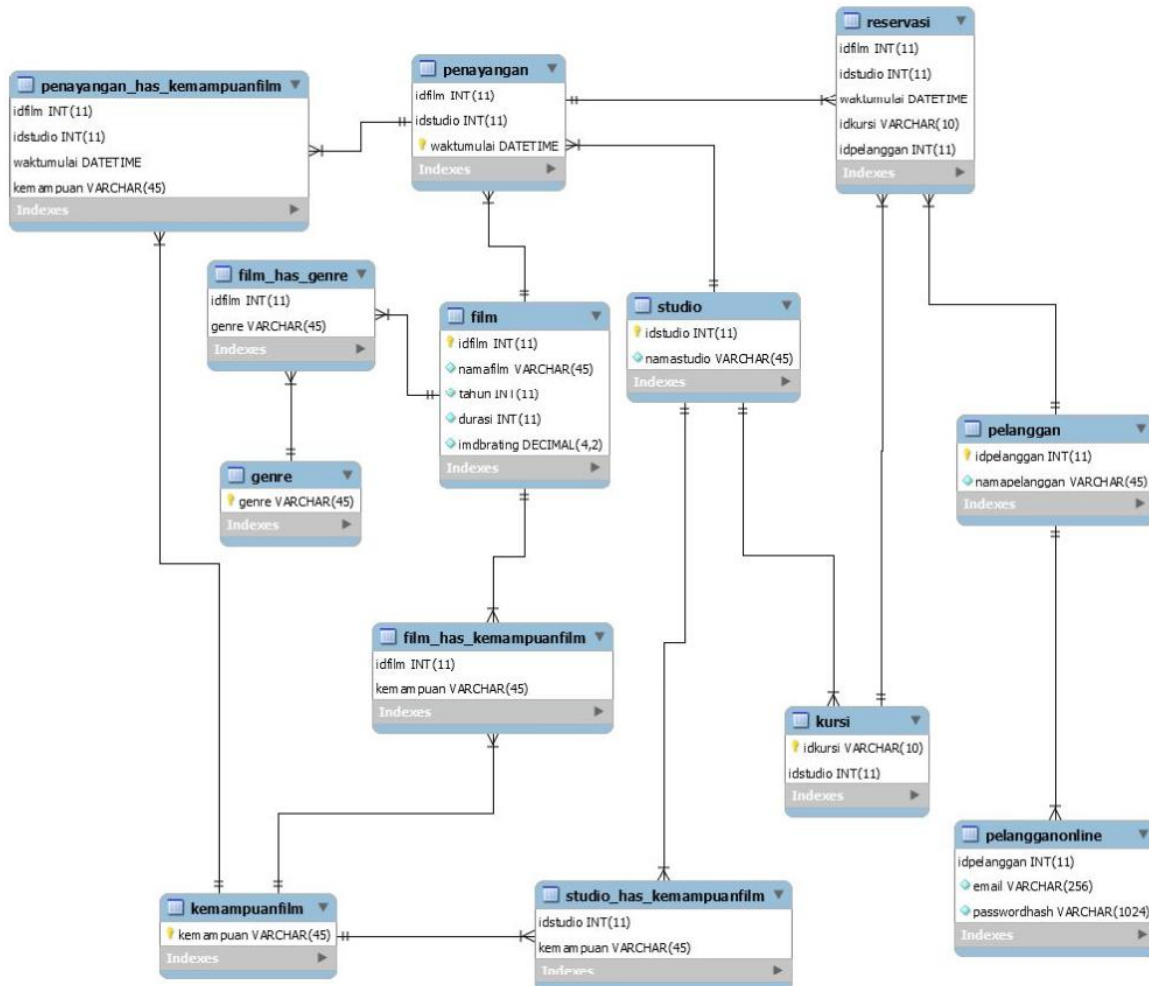
Pada data film, disimpan nama, tahun, durasi, dan rating IMDB-nya. Film juga dapat memiliki beberapa genre. Pada data studio, disimpan nama dan ID-nya, sedangkan untuk kursi, hanya disimpan ID-nya. Pada data pelanggan disimpan namanya. Jika pelanggan tersebut adalah pelanggan yang mendaftar secara online,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

count(idkursi)	Month(waktumulai)
12134	11
10852	1
9852	8
9061	5
8975	12
8059	10
7685	6
7565	9
7560	4
7553	3
7065	7
6267	2

12 rows in set (0.067 sec)

- 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 ;
```

- 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

idstudio	namastudio	count(idkursi)
2	Studio 3	180
6	Studio 7	180
10	Studio 11	180
13	Studio 14	180
14	Studio 15	200
16	Studio 17	275
17	Studio 18	220

7 rows in set (0.004 sec)

4. 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

namafilam	tahun	durasi	imdbrating	count(idpelanggan)
Unforgotten	1900	45	7.90	1043
The Crow	1994	98	7.60	1069
Se7en	1995	127	8.60	1289
Schindler's List	1993	185	8.90	1300
Scream: The TV Series	1900	45	7.30	897
The Dead Zone	1900	60	7.50	1010
Jackie Brown	1997	154	7.50	1188
Major League	1989	107	7.20	948
Pulp Fiction	1994	178	8.90	1349
Nikita	1900	60	7.70	1224
Veronica Mars	1900	44	8.40	1333
The Apostle	1997	134	7.20	1051
Psych	1900	44	8.40	1053
Leaving Las Vegas	1995	111	7.60	1279
M*A*S*H	1900	25	8.40	1080
The Secret	1900	45	7.30	940
Red River	1948	126	7.80	954
Blazing Saddles	1974	93	7.80	986
Niagara	1953	92	7.10	831
Bambi	1942	70	7.40	883
Welcome to the Dollhouse	1995	88	7.50	1006
Pandora's Box	1929	110	8.00	1116
Intolerance: Love's Struggle Throughout the A	1916	123	8.00	1059
Ordet	1955	126	8.10	1405

24 rows in set (0.013 sec)

- 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

idfilm
28
488
562

3 rows in set (0.005 sec)

6. 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

```
MariaDB [uprak]> select * from marcus_movie;  
+-----+-----+-----+  
| Judul          | Rating Film | Tahun |  
+-----+-----+-----+  
| Formula 51     | 6.30       | 2001  |  
| Swing Vote     | 6.10       | 2008  |  
| 40 Days and 40 Nights | 5.60       | 2002  |  
| Scream: The TV Series | 7.30       | 1900  |  
| Niagara        | 7.10       | 1953  |  
+-----+-----+-----+  
5 rows in set (0.153 sec)
```

7. 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

nama_genre	jumlah_penayangan
Action	171
Adventure	162
Comedy	240
Crime	286
Drama	592
Mystery	136
Romance	191
Thriller	294

8 rows in set (0.017 sec)

8. 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 😊

9. 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

```
Query OK, 19 rows affected (0.105 sec)  
Rows matched: 19  Changed: 19  Warnings: 0  
MariaDB [uprak]> SELECT idfilm,imdbrating FROM film NATURAL JOIN reser  
vasi GROUP BY idfilm HAVING count(idpelanggan) > 500 and imdbrating <6  
.00;  
Empty set (0.058 sec)
```

10. 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

```
Query OK, 1 row affected (0.021 sec)  
Rows matched: 1  Changed: 1  Warnings: 0  
+-----+-----+  
| idstudio | namastudio |  
+-----+-----+  
|      17 | Studio Basdat |  
+-----+-----+  
1 row in set (0.000 sec)
```

11. Buatlah query untuk menghapus data reservasi pada studio yang bernama "Studio 1" dan pelanggan bernama "Wm Yapple"

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

```
MariaDB [uprak]> DELETE FROM reservasi WHERE idpelanggan=(SELECT idpelanggan FROM pelanggan WHERE namapelanggan = "WM YAPLE") and idstudio=(SELECT idstudio FROM studio WHERE namastudio="Studio 1");
Query OK, 3 rows affected (0.105 sec)
```

12. 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

```
MariaDB [uprak]> show tables;
```

Tables_in_uprak
film
film_has_genre
film_has_kemampuanfilm
genre
karyawan
kemampuanfilm
kursi
marcus_movie
pelanggan
pelangganonline
penayangan
penayangan_has_kemampuanfilm
reservasi
studio
studio_has_kemampuanfilm

```
15 rows in set (0.001 sec)
```

```
MariaDB [uprak]> describe karyawan;
```

Field	Type	Null	Key	Default	Extra
idkaryawan	int(11)	NO	PRI	NULL	
nama	varchar(45)	YES		NULL	
idstudio	int(11)	YES	MUL	NULL	

```
3 rows in set (0.083 sec)
```