Praktikum I: Basic SQL

Dipersiapkan oleh:

Asisten Laboratorium Basis Data

Pelaksanaan:

Waktu Mulai: Jumat, 4 Februari 2022 – 16.00 WIB

Waktu Selesai: Jumat, 4 Februari 2022 – 18.00 WIB

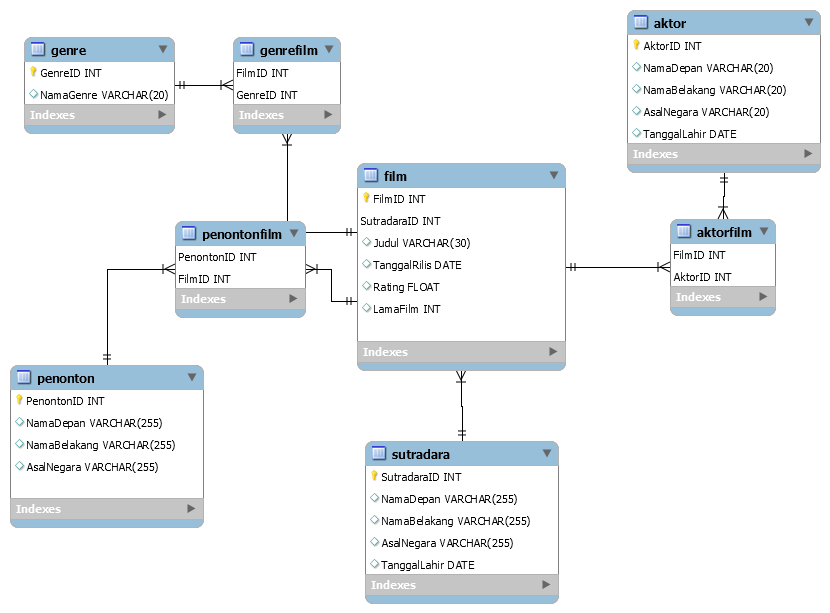
# I. Permasalahan

Buatlah basis data dengan nama “film”. Gunakan basis data tersebut, kemudian masukkan (import) script SQL prak1.sql yang telah diberikan bersama deskripsi tugas ini.

# 

# II. Skema Basis Data

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

****

|  |
| --- |
| film = (FilmID, SutradaraID, Judul, TanggalRilis, Rating, LamaFilm)  sutradara = (SutradaraID, NamaDepan, NamaBelakang, AsalNegara, TanggalLahir)  aktor = (AktorID, NamaDepan, NamaBelakang, AsalNegara, TanggalLahir)  genre = (GenreID, NamaGenre)  penonton = (PenontonID, NamaDepan, NamaBelakang, AsalNegara)  aktorfilm = (FilmID, AktorID)  genrefilm = (FilmID, GenreID)  penontonfilm = (PenontonID, FilmID) |

# III. Soal

**\*Perhatikan bahwa untuk semua soal tidak diperbolehkan untuk membuat view/tabel baru terlebih dahulu.**

**\*Untuk penggabungan dua string, silakan gunakan fungsi CONCAT pada MariaDB**

## Buatlah query yang menampilkan nama lengkap (NamaLengkap), asal negara, dan tanggal lahir dari aktor yang lahir sebelum Indonesia merdeka.

**Jawaban query :**

|  |
| --- |
| SELECT CONCAT(a.NamaDepan,' ' ,a.NamaBelakang) AS NamaLengkap, a.AsalNegara, a.TanggalLahir FROM aktor a WHERE a.TanggalLahir < '1945-07-17'; |

**Hasil screenshot :**

|  |
| --- |
|  |

## Buatlah query untuk menampilkan nama lengkap (NamaLengkap) dari 10 aktor termuda dan tanggal lahirnya.

**Jawaban query :**

|  |
| --- |
| SELECT CONCAT (a.NamaDepan,' ',a.NamaBelakang) AS NamaLengkap , a.TanggalLahir FROM aktor a ORDER BY a.TanggalLahir desc LIMIT 10; |

**Hasil screenshot :**

|  |
| --- |
|  |

## Tampilkan rata-rata rating dari film yang rilis setelah tahun 2000 dan memiliki durasi lebih dari 200 menit.

## *Hint: Gunakan fungsi YEAR(column) untuk mengambil data tahun dari sebuah kolom bertipe DATE.*

**Jawaban query :**

|  |
| --- |
| SELECT AVG(f.Rating) AS Rata\_Rata\_Rating FROM film f WHERE YEAR(f.TanggalRilis) > 2000 AND f.LamaFilm > 200; |

**Hasil screenshot :**

|  |
| --- |
|  |

## Buatlah query untuk menampilkan nama dan jumlah aktor dari 5 negara yang memiliki jumlah aktor terbanyak. Terurut berdasarkan negara dengan jumlah aktor terbanyak.

**Jawaban query :**

|  |
| --- |
| SELECT a.AsalNegara, COUNT(a.AktorID) AS Jumlah\_Aktor FROM aktor a GROUP BY a.AsalNegara ORDER BY COUNT(a.AktorID) desc LIMIT 5; |

**Hasil screenshot :**

|  |
| --- |
|  |

## Buatlah query untuk menampilkan ID sutradara, nama lengkap sutradara, umur sutradara, dan jumlah film yang dibuat oleh sutradara tersebut, dari sutradara yang berumur 45 tahun atau lebih muda dan telah membuat lebih dari 10 film.

*Hint: Untuk mencari perbedaan waktu, gunakan fungsi TIMESTAMPDIFF(), e.g. TIMESTAMPDIFF(YEAR, “1970-01-01”, “2022-01-01”) akan mengembalikan nilai 52, dan untuk mendapatkan tanggal hari ini, fungsi CURDATE() dapat digunakan.*

**Jawaban query :**

|  |
| --- |
| SELECT s.SutradaraID, CONCAT(s.NamaDepan,' ',s.NamaBelakang) AS NamaLengkap, TIMESTAMPDIFF(YEAR,s.TanggalLahir,"2022-02-04") AS Umur, COUNT(f.FilmID) AS JumlahFIlm FROM film f NATURAL JOIN sutradara s WHERE TIMESTAMPDIFF(YEAR,s.TanggalLahir,"2022-02-04")<=45 GROUP BY s.SutradaraID; |

**Hasil screenshot :**

|  |
| --- |
|  |

## Buatlah query untuk menampilkan nama lengkap dari 10 sutradara yang memiliki rata-rata rating film yang disutradarainya di atas rata-rata seluruh film, kemudian diurutkan membesar berdasarkan rata-rata rating filmnya.

**Jawaban query :**

|  |
| --- |
| SELECT CONCAT(s.NamaDepan,' ',s.NamaBelakang) FROM film f NATURAL JOIN sutradara s WHERE f.rating > (SELECT AVG(rating) FROM FILM) GROUP BY s.SutradaraID ORDER BY AVG(f.rating) LIMIT 10; |

**Hasil screenshot :**

|  |
| --- |
|  |

## Buatlah query untuk menampilkan nama belakang aktor yang pernah bermain dalam film yang memiliki rating di atas 4.7 dan memiliki nama belakang yang sama dengan sutradara yang berasal dari Indonesia.

*CATATAN: Gunakan set operations.*

**Jawaban query :**

|  |
| --- |
|  |

**Hasil screenshot :**

|  |
| --- |
|  |

## Buatlah query untuk menampilkan 10 baris pertama dari ID penonton, nama lengkap penonton, asal negara penonton, dan jumlah film yang sudah dilihat oleh masing-masing penonton untuk penonton yang berasal dari negara Indonesia ataupun China, diurutkan secara menurun berdasarkan jumlah film yang ditonton, kemudian menaik berdasarkan ID penonton.

**Jawaban query :**

|  |
| --- |
| SELECT p.PenontonID, CONCAT(p.NamaDepan,' ',p.NamaBelakang) AS NamaLengkap, p.AsalNegara, COUNT(p.PenontonID) FROM penonton p NATURAL JOIN penontonfilm pf WHERE p.AsalNegara LIKE "China" OR p.asalNegara LIKE "Indonesia" GROUP BY p.PenontonID ORDER BY COUNT(p.PenontonID) asc; |

**Hasil screenshot :**

|  |
| --- |
|  |

## Buatlah query untuk menampilkan id film, judul film, rating film, dan nama lengkap sutradara dari film yang memiliki rating lebih besar dari 4.95 terurut menurun berdasarkan ratingnya.

**Jawaban query :**

|  |
| --- |
| SELECT f.FilmID, f.Judul, f.Rating, CONCAT(s.NamaDepan,' ',s.NamaBelakang) FROM film f NATURAL JOIN sutradara s WHERE f.Rating>4.95 ORDER BY f.Rating desc; |

**Hasil screenshot :**

|  |
| --- |
|  |

## Buatlah query untuk menampilkan id film, judul film, dan jumlah penonton dari semua film yang pernah dimainkan oleh aktor bernama “Nicole Montgomery”.

**Jawaban query :**

|  |
| --- |
| SELECT f.FilmID, f.Judul, COUNT(pf.PenontonID) AS JumlahPenonton FROM penontonfilm pf NATURAL JOIN film f NATURAL JOIN aktorfilm af NATURAL JOIN aktor a WHERE a.NamaDepan LIKE "Nicole" AND a.NamaBelakang LIKE "Montgomery"; |

**Hasil screenshot :**

|  |
| --- |
|  |

## Tampilkan judul film beserta sutradaraid untuk film dengan sutradara yang belum pernah menyutradarai film dengan rating lebih dari 4. Jika terdapat lebih dari 1 film dengan judul dan sutradaraid yang sama, tampilkan salah satu saja. Hasil diurutkan berdasarkan sutradaraid secara ascending.

**Jawaban query :**

|  |
| --- |
| SELECT DISTINCT f.Judul, s.SutradaraID FROM film f NATURAL JOIN sutradara s WHERE s.SutradaraID NOT IN (SELECT f.SutradaraID FROM film f WHERE f.Rating>4.0) GROUP BY s.SutradaraID; |

**Hasil screenshot :**

|  |
| --- |
|  |

## Tampilkan id aktor dan jumlah film yang dibintanginya yang memiliki rating lebih dari rata-rata. Tampilkan jumlah film sebagai ‘above\_avg\_movie’. Urutkan hasil berdasarkan id aktor secara ascending.

**Jawaban query :**

|  |
| --- |
| SELECT af.AktorId, COUNT(f.filmID) AS above\_avg\_movie FROM film f NATURAL JOIN aktorfilm af WHERE f.Rating > (SELECT AVG(rating)FROM film) GROUP BY af.aktorID ORDER BY COUNT(f.filmID) asc; |

**Hasil screenshot :**

|  |
| --- |
|  |

**Shape, rectangle

Description automatically generated**