**IF2240 Basis Data**

**Semester II 2021/2022**

Ujian Praktikum

Dipersiapkan oleh:

Asisten Laboratorium Basis Data

Pelaksanaan:

Waktu Mulai: Jumat, 18 Februari 2022 – 16.05 WIB

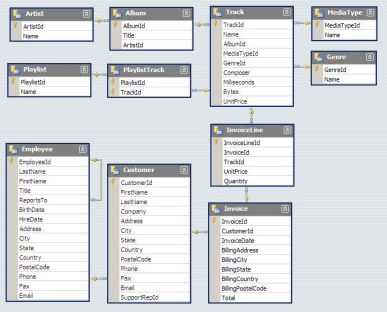
Waktu Selesai: Jumat, 18 Februari 2022 – 18.05 WIB

# I. Deskripsi Sistem

Sebuah toko musik “Basdat Music Square” melakukan pencatatan data track-track musik yang dimiliki, serta pembelian yang terjadi pada toko tersebut. Data tersebut digunakan untuk menganalisis minat masyarakat terhadap musik yang dijual. Satu track bisa masuk ke dalam banyak playlist, tetapi satu track hanya masuk ke satu album.

Toko musik juga menyimpan data pembelian dalam bentuk invoice. Invoice memiliki satu atau lebih invoice line yang merupakan detail pembelian tiap track. Setiap invoice juga akan mencatat identitas pembeli termasuk siapa penjual yang bertugas melayani pembeli tersebut.

# II. Skema Basis Data



|  |
| --- |
| Artist = (ArtistId, Name)  Album = (AlbumId, Title, ArtistId)  Playlist = (PlaylistId, Name)  PlaylistTrack = (PlaylistId, TrackId)  Track = (TrackId, Name, AlbumId, MediaTypeId, GenreId, Composer, Milliseconds, Bytes, UnitPrice)  MediaType = (MediaTypeId, Name)  Genre = (GenreId, Name)  Invoice = (InvoiceId, CustomerId, InvoiceDate, BillingAddress, BillingCity, BillingState, BillingCountry, BillingPostalCode, Total)  InvoiceLine = (InvoiceLineId, InvoiceId, TrackId, UnitPrice, Quantity)  Customer = (CustomerId, FirstName, LastName, Company, Address, City, State, Country, PostalCode, Phone, Fax, Email, SupportRepId)  Employee = (EmployeeId, FirstName, LastName, Title, ReportsTo, BirthDate, HireDate, Address, City, State, Country, PostalCode, Phone, Fax, Email) |

# III. Permasalahan

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

# III. Soal

Beberapa hal yang perlu diperhatikan dalam pengerjaan soal:

1. Untuk semua soal **tidak diperbolehkan untuk membuat view/tabel** **baru** terlebih dahulu **kecuali diinstruksikan pada soal**
2. Untuk **penggabungan dua string**, silakan gunakan fungsi **CONCAT** pada MariaDB

## Buatlah query untuk menampilkan nama negara dan jumlah pelanggan yang berdomisili di negara tersebut diurutkan menurun berdasarkan jumlah pelanggan.

**Jawaban query :**

|  |
| --- |
| SELECT country, COUNT(CustomerID) FROM Customer GROUP BY Country ORDER BY COUNT(CustomerID) desc; |

**Hasil screenshot :**

|  |
| --- |
|  |

## Buatlah query untuk menampilkan nama artis yang memiliki track yang terdapat pada playlist yang memiliki 3 jenis genre.

Catatan: Genre playlist adalah union dari genre seluruh track dalam playlist terkait.

**Jawaban query :**

|  |
| --- |
| SELECT Name From Artist NATURAL JOIN Album WHERE AlbumID IN (SELECT albumID FROM Track WHERE TrackId IN (SELECT TrackID FROM PlaylistTrack pt NATURAL JOIN Track GROUP BY PlaylistID HAVING COUNT(DISTINCT GenreID)=3)); |

**Hasil screenshot :**

|  |
| --- |
|  |

## Buatlah sebuah query untuk menampilkan nama artist dan total album yang dimiliki oleh artist tersebut dari artist yang memiliki lebih dari 5 album.

**Jawaban query :**

|  |
| --- |
| SELECT Name, COUNT(ArtistID) AS "Jumlah Album" FROM Artist a NATURAL JOIN Album al WHERE ArtistID IN (SELECT ArtistID FROM Album GROUP BY ArtistId HAVING COUNT(AlbumID)>5) GROUP BY ArtistID; |

**Hasil screenshot :**

|  |
| --- |
|  |

## Buatlah query untuk menampilkan id playlist, nama playlist, beserta total lagu yang terdapat pada playlist tersebut **dari semua playlist yang ada** diurutkan secara menurun berdasarkan jumlah lagunya.

**Jawaban query :**

|  |
| --- |
| SELECT PlaylistID, Name, COUNT(TrackID) FROM Playlist pl NATURAL JOIN PlaylistTrack GROUP BY PlaylistID ORDER BY COUNT(TrackID) desc; |

**Hasil screenshot :**

|  |
| --- |
|  |

## Tunjukkan customerid secara unik yang pernah membeli track dengan genre ‘Jazz’ namun tidak pernah membeli track dengan genre ‘Bossa Nova’. Urutkan hasil secara menaik.

Catatan: Gunakan set operations.

**Jawaban query :**

|  |
| --- |
| SELECT DISTINCT CustomerID FROM Invoice WHERE InvoiceID IN (SELECT InvoiceID FROM InvoiceLine WHERE TrackID IN (SELECT TrackID FROM Track WHERE GenreID IN (SELECT GenreID FROM Genre WHERE Name LIKE "Jazz"))) AND InvoiceID NOT IN (SELECT InvoiceID FROM InvoiceLine WHERE TrackID IN (SELECT TrackID FROM Track WHERE GenreID IN (SELECT GenreID FROM Genre WHERE Name LIKE "Bossa Nova"))) ORDER BY (CustomerID) asc; |

**Hasil screenshot :**

|  |
| --- |
|  |

## Miles Davis ingin melakukan upgrade untuk seluruh albumnya. Tambahkan album untuk Miles Davis dengan semua album yang sudah dimilikinya ditambah dengan “ (Expanded Edition)”. Id untuk album-album baru tersebut merupakan id album lama ditambah 1000. Buktikan dengan menampilkan nama artist beserta nama album untuk seluruh album milik Miles Davis.

**Jawaban query :**

|  |
| --- |
| INSERT INTO Album SELECT AlbumID+1000,CONCAT(Title," ","Expanded Edition"), ArtistId FROM Album WHERE ArtistID IN (SELECT ArtistID FROM Artist Where Name LIKE "Miles Davis"); |

**Hasil screenshot :**

\*screenshot saat melakukan update dan setelahnya (sesuai dengan bukti pada instruksi soal).

|  |
| --- |
| **Sebelum**    **Saat**    **Sesudah** |

## Buatlah sebuah view dengan nama "Jazzy" untuk Track yang memiliki genre Jazz dan telah dibeli sebanyak 2 kali atau lebih. Dalam view tersebut, data yang disertakan adalah TrackId, nama lagu tersebut sebagai "nama\_lagu", genre lagu tersebut sebagai "genre\_lagu", serta jumlah\_pembelian\_lagu tersebut. Setelah itu, tampilkan data yang ditampung dalam view tersebut

**Jawaban query :**

|  |
| --- |
| CREATE VIEW Jazzy AS (SELECT t.TrackID, t.Name as "nama\_lagu", g.Name as "genre\_lagu", COUNT(InvoiceLineId) FROM Track t INNER JOIN Genre g ON g.genreID = t.genreID INNER JOIN InvoiceLine il ON t.trackID = il.TrackID GROUP BY il.TrackID HAVING COUNT(InvoiceLineId)>=2);  Query OK, 0 rows affected (0.018 sec) |

**Hasil screenshot :**

|  |
| --- |
|  |

## Buatlah sebuah view dengan nama "not\_purchased" untuk Track yang belum pernah dibeli sama sekali, memiliki Id Album di atas 289, serta nilai kolom Composer tidak Null. Dalam view tersebut, data yang disertakan adalah TrackId, nama lagu sebagai "nama\_lagu", genre lagu sebagai "nama\_genre", serta komposer lagu tersebut. Setelah itu, tampilkan data yang ditampung dalam view tersebut.

**Jawaban query :**

|  |
| --- |
| CREATE VIEW not\_purchased AS (SELECT t.TrackId, t.Name as "nama\_lagu", g.Name as "genre\_lagu", t.composer FROM Track t INNER JOIN Genre g ON t.GenreID = g.GenreID WHERE t.Composer IS NOT NULL AND t.AlbumID>289 AND T.TrackID NOT IN (SELECT TrackID FROM InvoiceLine)); |

**Hasil screenshot :**

|  |
| --- |
|  |

## Buatlah query untuk mengubah nama composer menjadi "Basdat Rock Composer" dari track yang composernya tidak diketahui dan bergenre “Rock”.

## 

## Buktikan dengan screenshot sebelum dan sesudah dilakukan perubahan. Pastikan screenshot memperlihatkan juga bagian jumlah rows.

**Jawaban query :**

|  |
| --- |
| UPDATE TRACK SET composer = Basdat Rock Composer WHERE composer IS NULL AND genreID is IN (SELECT GenreID From Genre WHERE Name LIKE "ROCK"); |

**Hasil screenshot :**

|  |
| --- |
| **Sebelum**    **Saat**    **Sesudah** |

## Para artis sedang mempromosikan lagu-lagu mereka menjadi lebih murah untuk paket bundling album. Buatlah query untuk mengubah harga setiap unit track yang terdapat pada album yang memiliki lebih dari 50 track menjadi 10% lebih murah.

## 

## Buktikan dengan screenshot sebelum dan sesudah dilakukan perubahan. Pastikan screenshot memperlihatkan juga bagian jumlah rows.

**Jawaban query :**

|  |
| --- |
| UPDATE Track SET UnitPrice = UnitPrice\*0.9 WHERE AlbumID IN (SELECT AlbumID FROM Track GROUP BY AlbumID HAVING COUNT(TrackID)>50); |

**Hasil screenshot :**

|  |
| --- |
| **Sebelum**      **Saat**    **Sesudah** |

## Buatlah query untuk menghapus data invoiceline yang memiliki track dengan nama artist ‘Chico Science & Nação Zumbi’ dan memiliki judul album ‘Afrociberdelia’.

## 

## Tunjukkan hasilnya (berupa 2 screenshot) dengan cara menampilkan id invoiceline, nama artist dan nama album yang memiliki track dengan nama artist ‘Chico Science & Nação Zumbi’ dan memiliki judul album ‘Afrociberdelia’, Sebelum dan Sesudah query delete.

**Jawaban query :**

|  |
| --- |
| DELETE FROM InvoiceLine WHERE InvoiceLineID IN (SELECT il.InvoiceLineId FROM Artist a NATURAL JOIN Album al INNER JOIN Track t ON al.AlbumID = t.AlbumID INNER JOIN InvoiceLine il ON t.TrackId = il.TrackID WHERE a.Name LIKE "Chico Science & Nação Zumbi" AND al.Title LIKE "Afrociberdelia");  Query menampilkan:  SELECT il.InvoiceLineId, a.Name, al.Title FROM Artist a NATURAL JOIN Album al INNER JOIN Track t ON al.AlbumID = t.AlbumID INNER JOIN InvoiceLine il ON t.TrackId = il.TrackID WHERE a.Name LIKE "Chico Science & Nação Zumbi" AND al.Title LIKE "Afrociberdelia"; |

**Hasil screenshot :**

|  |
| --- |
| **Sebelum**    **Saat**    **Sesudah** |

## Buatlah query untuk mengubah tabel customer dengan spesifikasi sebagai berikut:

## Mengubah nama tabel menjadi newcustomer

## Menghapus kolom phone

## Menambahkan kolom whatsapp TEPAT setelah kolom email dengan tipe data varchar dan diberikan nilai default 08123456789 ke semua data yang ada

## Menghapus key dengan nama IFK\_CustomerSupportRepId

## Menghapus Foreign Key dengan nama FK\_CustomerSupportRepId

## Query bisa dilakukan per masing masing spesifikasi (total ada 5 query) atau dilakukan 1 query mencakup 5 spesifikasi di atas. Tunjukkan hasilnya berupa screenshot describe tabel customer yang baru dan jalankan perintah “show create table newcustomer” lalu screenshot. (Total 2 Screenshot)

## 

## Constraint: Jawablah soal ini dengan urutan paling TERAKHIR, jika anda tidak ingin menjawab soal ini di urutan terakhir, silahkan untuk restore ulang database music setelah menjawab soal ini untuk menjawab soal lainnya.

**Jawaban query :**

|  |
| --- |
| RENAME TABLE customer TO newcustomer;  ALTER TABLE newcustomer DROP phone; |

**Hasil screenshot :**

|  |
| --- |
|  |