**-- The query finds the names of artists who have not made any albums at all.**

SELECT ARTIST.Name

FROM ARTIST

LEFT JOIN ALBUM ON ALBUM.ARTISTID = ARTIST.ArtistId

WHERE ALBUM.AlbumId IS NULL;

**--The query finds the artists who did not record any tracks in the Latin genre.**

SELECT DISTINCT Artist.Name

FROM Artist

WHERE NOT EXISTS (

SELECT 1

FROM Album

JOIN Track ON Album.AlbumId = Track.AlbumId

JOIN Genre ON Track.Genreid = Genre.GenreID

WHERE Artist.ArtistId = Album.ArtistId AND Genre.Name = 'Latin'

);

**--The query finds the video track with the longest length.**

SELECT TRACK.Name

FROM TRACK

JOIN MEDIATYPE ON MEDIATYPE.MediaTypeId = TRACK.MediaTypeId

WHERE MEDIATYPE.Name = 'Protected MPEG-4 video file' AND TRACK.Milliseconds = (SELECT MAX(TRACK.Milliseconds) FROM TRACK);

**--The query finds the names of customers who live in the same city as the top employee (the one not managed by anyone).**

SELECT CUSTOMER.FirstName, CUSTOMER.LastName

FROM CUSTOMER

WHERE CUSTOMER.City = (SELECT EMPLOYEE.City FROM EMPLOYEE WHERE ReportsTo IS NULL);

**--The query finds the managers of employees who are supporting Brazilian customers.**

SELECT EMPLOYEE.FirstName, EMPLOYEE.LastName

FROM EMPLOYEE

WHERE EMPLOYEE.EmployeeId = (SELECT EMPLOYEE.ReportsTo

FROM EMPLOYEE

JOIN CUSTOMER ON EMPLOYEE.EmployeeId = CUSTOMER.SupportRepId

WHERE CUSTOMER.Country = 'Brazil');

**--The query finds the total number of audio tracks purchased by German customers and the total price paid for them.**

SELECT

COUNT(INVOICELINE.TrackId), SUM(INVOICELINE.UnitPrice)

FROM INVOICELINE

JOIN TRACK ON INVOICELINE.TrackId = TRACK.TrackId

JOIN MEDIATYPE ON MEDIATYPE.MediaTypeId = TRACK.MediaTypeId

JOIN INVOICE ON INVOICE.InvoiceId = INVOICELINE.InvoiceId

JOIN CUSTOMER ON INVOICE.CustomerId = CUSTOMER.CustomerId

WHERE MEDIATYPE.Name LIKE '%Audio%'

AND CUSTOMER.Country = 'Germany';

**--The query finds the space in bytes occupied by the "Grunge" playlist and the total cost of the playlist (calculated as the sum of the prices of its tracks).**

SELECT SUM(TRACK.Bytes), SUM(TRACK.UnitPrice)

FROM TRACK

JOIN PLAYLISTTRACK ON TRACK.TrackId = PLAYLISTTRACK.TrackId

JOIN PLAYLIST ON PLAYLIST.PlaylistId = PLAYLISTTRACK.PlaylistId

WHERE PLAYLIST.Name = 'Grunge';

**--The query finds the playlists that do not contain any tracks by the artists "Black Sabbath" or "Chico Buarque."**

SELECT Playlist.Name, Playlist.PlaylistId

FROM Playlist

WHERE NOT EXISTS (

SELECT 1

FROM PlaylistTrack

JOIN Track ON Playlisttrack.TrackId = Track.TrackId

JOIN Album ON Track.AlbumId = Album.AlbumId

JOIN Artist ON Album.ArtistId = Artist.ArtistId

WHERE playlisttrack.PlaylistId = playlist.PlaylistId

AND (Artist.Name = 'Black Sabbath' OR Artist.Name = 'Chico Buarque')

);

**--The query finds the names and countries of customers who are supported by an employee who was younger than 35 when hired.**

SELECT CUSTOMER.FirstName, CUSTOMER.LastName, CUSTOMER.Country

FROM CUSTOMER

JOIN EMPLOYEE ON CUSTOMER.SupportRepId = EMPLOYEE.EmployeeId

WHERE (JULIANDAY(EMPLOYEE.HireDate) - JULIANDAY(EMPLOYEE.BirthDate)) / 365 < 35;

**-- The query finds the name of the German customer who has paid the most in total, excluding the company name.**

SELECT CUSTOMER.FirstName, CUSTOMER.LastName

FROM CUSTOMER

JOIN INVOICE ON CUSTOMER.CustomerId = INVOICE.CustomerId

WHERE CUSTOMER.Country = 'Germany' AND CUSTOMER.Company IS NULL

GROUP BY CUSTOMER.CustomerId

HAVING SUM(INVOICE.Total) = (

SELECT MAX(TotalAmount)

FROM (

SELECT CUSTOMER.CustomerId, SUM(INVOICE.Total) AS TotalAmount

FROM CUSTOMER

JOIN INVOICE ON CUSTOMER.CustomerId = INVOICE.CustomerId

WHERE CUSTOMER.Country = 'Germany' AND CUSTOMER.Company IS NULL

GROUP BY CUSTOMER.CustomerId

) );

**--The query finds the names and ages of employees who support more than 5 customers.**

CREATE VIEW BIRTH AS

SELECT EmployeeId,

((JULIANDAY('now') - JULIANDAY(EMPLOYEE.BirthDate)) / 365) AS Age

FROM EMPLOYEE;

SELECT EMPLOYEE.FirstName, BIRTH.Age

FROM EMPLOYEE

JOIN BIRTH ON EMPLOYEE.EmployeeId = BIRTH.EmployeeId

JOIN CUSTOMER ON EMPLOYEE.EmployeeId = CUSTOMER.SupportRepId

GROUP BY EMPLOYEE.EmployeeId, EMPLOYEE.FirstName, BIRTH.Age

HAVING COUNT(CUSTOMER.SupportRepId) > 5;

**--The query finds the manager who manages the most employees but is also managed by someone else (excluding employees who do not have managers).**

SELECT e1.FirstName, e1.LastName

FROM EMPLOYEE e1

JOIN EMPLOYEE e2 ON e1.EmployeeId = e2.ReportsTo

WHERE e1.ReportsTo IS NOT NULL

GROUP BY e1.EmployeeId

ORDER BY COUNT(e2.ReportsTo) DESC

LIMIT 1

**-- The query finds the names of artists who have more than 5 tracks.**

SELECT ARTIST.Name

FROM ARTIST

JOIN ALBUM ON ALBUM.ArtistId = ARTIST.ArtistId

JOIN TRACK ON ALBUM.AlbumId = TRACK.AlbumId

GROUP BY ARTIST.ArtistId

HAVING COUNT(TRACK.TrackId) > 5

**-- The query finds the playlist(s) that contain the most tracks by the artist "AC/DC."**

CREATE VIEW COUNTER AS

SELECT PLAYLIST.PlaylistId, COUNT(TRACK.TrackId) AS TRACK\_COUNT

FROM PLAYLIST

JOIN PLAYLISTTRACK ON PLAYLIST.PlaylistId = PLAYLISTTRACK.PlaylistId

JOIN TRACK ON PLAYLISTTRACK.TrackId = TRACK.TrackId

JOIN ALBUM ON ALBUM.AlbumId = TRACK.AlbumId

JOIN ARTIST ON ALBUM.ArtistId = ARTIST.ArtistId

WHERE ARTIST.Name = 'AC/DC'

GROUP BY PLAYLIST.PlaylistId;

SELECT PlaylistId, TRACK\_COUNT

FROM COUNTER

WHERE TRACK\_COUNT = (SELECT MAX(TRACK\_COUNT) FROM COUNTER);