```
--1)
SELECT e1.FirstName, e2.FirstName
FROM Employee e1
JOIN Employee e2
ON e1.EmployeeId = e2.ReportsTo
--2)
SELECT *
FROM Track
SELECT t.Name, g.Name as Genre, t.Milliseconds
FROM Track t
LEFT JOIN Genre g
ON t.GenreId = g.GenreId
--3)
INSERT INTO InvoiceLine (InvoiceId, TrackId, UnitPrice, Quantity)
VALUES (2,2,0.99,1050)
--4)
SELECT t.TrackId, t.Name, a.Name as Artist, SUM(il.TrackId) AS QuantityTotal
FROM Track t
JOIN Album al
ON t.AlbumId = al.AlbumId
JOIN Artist a
ON a.ArtistId = al.ArtistId
JOIN InvoiceLine il
ON t.TrackId = il.TrackId
GROUP BY t.TrackId, t.Name, a.Name
HAVING SUM(il.TrackId) > 1000
ORDER BY t.TrackId
--5) SELECT i.CustomerId, COUNT(DISTINCT i.InvoiceId) AS CustomerInvoice, SUM(il.UnitPrice)
AS Totalinvoice
FROM Invoice i
JOIN InvoiceLine il
ON il.InvoiceId = i.InvoiceId
GROUP BY i.CustomerId
ORDER BY Totalinvoice DESC
--6)
SELECT *
FROM Employee e
SELECT e.EmployeeId, e.LastName, e.FirstName, e.Title, COUNT(c.CustomerId) AS Customers
FROM Employee e
JOIN Customer c
ON e.EmployeeId = c.SupportRepId
GROUP BY e.FirstName
--7)
SELECT g.Name, COUNT(t.GenreId) AS TotalTrack
FROM Genre g
JOIN Track t
ON t.GenreId = g.GenreId
GROUP BY g.Name
HAVING TotalTrack > 50
ORDER BY TotalTrack
--8)
SELECT c.CustomerId, c.Firstname, SUM(i.Total) AS TotalSpent
```

```
FROM Customer c
JOIN Invoice i
ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
ORDER BY TotalSpent DESC
--9)
SELECT pt.PlaylistId, COUNT(pt.TrackId) AS TotalTrack, p.Name
FROM PlaylistTrack pt
JOIN Playlist p
ON p.PlaylistId = pt.PlaylistId
GROUP BY pt.PlaylistId
HAVING TotalTrack >= 20
--10)  SELECT t.Name, t.TrackId,t.AlbumId
FROM Track t
WHERE t.AlbumId IS NULL
```