

```

/*
1.
Provide a report displaying the 10 artists with the most sales from July 2011 through June 2012.
Do not include any video tracks in the sales.
Display the Artist's name and the total sales for the year.
Include ties for 10th if there are any.
*/
SELECT TOP 10 WITH TIES
    A.Name AS ArtistName
    ,SUM(IL.UnitPrice * IL.Quantity) AS Total
FROM Artist A
JOIN Album AL
    ON AL.ArtistId = A.ArtistId
JOIN Track T
    ON T.AlbumId = AL.AlbumId
    AND T.MediaTypeId <> 3
JOIN InvoiceLine IL
    ON IL.TrackId = T.TrackId
JOIN Invoice I
    ON I.InvoiceId = IL.InvoiceId
WHERE I.InvoiceDate BETWEEN '7/1/2011' AND '6/30/2012'
GROUP BY A.Name
ORDER BY Total DESC

```

/*

2.

Provide a report displaying the total sales for all Sales Support Agents grouped by year and quarter. Include data from January 2010 through June 2012. Each year has 4 Sales Quarters divided as follows:

Jan-Mar: Quarter 1

Apr-Jun: Quarter 2

Jul-Sep: Quarter 3

Oct-Dec: Quarter 4

The Sales Quarter column should display its values as First, Second, Third, Fourth.

The data needs to be ordered by the employee name, the calendar year, and the sales quarter.

The sales quarter order should be numeric and not alphabetical (e.g. "Third" comes before "Fourth").

*/

SELECT

E.FirstName+' '+E.LastName AS [Employee Name]

,YEAR(InvoiceDate) AS [Calendar Year]

,CASE DATEPART(QUARTER,I.InvoiceDate)

WHEN 1 THEN 'First'

WHEN 2 THEN 'Second'

WHEN 3 THEN 'Third'

WHEN 4 THEN 'Fourth'

END AS [Sales Quarter]

,MAX(I.Total) AS [Highest Sale]

,COUNT(I.Total) AS [Number of Sales]

,SUM(I.Total) AS [Total Sales]

FROM

Employee E

JOIN Customer C

ON C.SupportRepId = E.EmployeeId

JOIN Invoice I

ON I.CustomerId = C.CustomerId

WHERE I.InvoiceDate BETWEEN '01/01/2010' AND '06/30/2012'

GROUP BY

E.FirstName

,E.LastName

,YEAR(InvoiceDate)

,DATEPART(QUARTER,I.InvoiceDate)

ORDER BY [Employee Name],[Calendar Year],DATEPART(QUARTER,I.InvoiceDate)

```
/*
3.
The Sales Reps have discovered duplicate Playlists in the database. The duplicates have the same
Playlist name, but have a higher Playlist ID.
Write a SELECT statement that will return the duplicate Playlist IDs and Names, as well as any
associated Track IDs if they exist.
Your result set will be marked for deletion so it must be accurate.
*/
```

```
SELECT
    P.Name AS [Playlist Name]
    ,P.PlaylistId AS [Playlist ID]
    ,PT.TrackId AS [Track ID]
FROM Playlist P
LEFT JOIN PlaylistTrack PT
    ON PT.PlaylistId = P.PlaylistId
WHERE EXISTS(
SELECT * --name, max(PlaylistId),min(PlaylistId), count(*)
FROM Playlist P2
GROUP BY P2.Name
HAVING COUNT(*)>1
AND MAX(P2.PlaylistId) = P.PlaylistId
)
```

```

/*
4.
Management would like to view Artist popularity by Country.
Provide a report that displays the Customer country and the Artist name.
Determine the total number of tracks sold by an artist to each country, and the total unique tracks by
artist sold to each country.
Include a column that shows the difference between the track count and the unique track count.
Include the total revenue which will be the cost of the track multiplied by the number of tracks
purchased.
Include a column that shows whether the tracks are audio or video (Hint: Videos have a MediaTypeId =3).
The range of data will be between July 2009 and June 2013.
Order the results by Country, Track Count and Artist Name.
*/
SELECT
    Country
    ,A.Name AS [Artist Name]
    ,COUNT(*) AS [Track Count]
    ,COUNT(DISTINCT T.Name) AS [Unique Track Count]
    ,COUNT(*) - COUNT(DISTINCT T.Name) AS [Count Difference]
    ,SUM(IL.UnitPrice * IL.Quantity) AS [Total Revenue]
    ,CASE WHEN M.MediaTypeId = 3 THEN 'Video' ELSE 'Audio' END AS [Media Type]
FROM Customer C
JOIN Invoice I
    ON I.CustomerId = C.CustomerId
JOIN InvoiceLine IL
    ON IL.InvoiceId = I.InvoiceId
JOIN Track T
    ON T.TrackId = IL.TrackId
JOIN Album AL
    ON AL.AlbumId = T.AlbumId
JOIN Artist A
    ON A.ArtistId = AL.ArtistId
JOIN MediaType M
    ON M.MediaTypeId = T.MediaTypeId
WHERE I.InvoiceDate BETWEEN '07/01/2009' AND '6/30/2013'
GROUP BY
    Country
    ,A.Name
    ,CASE WHEN M.MediaTypeId = 3 THEN 'Video' ELSE 'Audio' END
ORDER BY
    Country
    ,[Track Count] DESC
    ,A.Name

```

```

/*
5.
HR wants to plan birthday celebrations for all employees in 2016.
They would like a list of employee names and birth dates, as well as the day of the week the birthday
falls on in 2016.
Celebrations will be planned the same day as the birthday if it falls on Monday through Friday.
If the birthday falls on a weekend then the celebration date needs to be set on the following Monday.
Provide a report that displays the above date logic.
The column formatting needs to be the same as in the example below.
(Hint: This is a tough one. I used 7 different functions in my solution.
You will need to nest functions inside other functions. Don't worry about accounting for leap
birthdays in your script.)
*/
SELECT
    FirstName + ' ' + LastName AS [Full Name]
    --Converts the birth date to U.S. standard.
    ,CONVERT(varchar,BirthDate,101) AS [Birth Date]
    --Breaks out the Day and Month from BirthDate and merges them back together with 2016 using
DATEFROMPARTS.
    ,CONVERT(varchar,DATEFROMPARTS(2016,Month(BirthDate),Day(BirthDate)),101) AS [Birth Day 2016]
    --Finds the day of week in 2016 using DATENAME.
    ,DATENAME(weekday,DATEFROMPARTS(2016,Month(BirthDate),Day(BirthDate))) AS [Birth Day of Week]
    -- Checks for weekend dates using DATEPART and if found moves them to Monday using DATEADD.
    --Conerts the birth date to U.S. standard.
    ,CONVERT(varchar,(CASE
        WHEN DATEPART(weekday,DATEFROMPARTS(2016,Month(BirthDate),Day(BirthDate))) = 1
        THEN DATEADD(DAY,1,DATEFROMPARTS(2016,Month(BirthDate),Day(BirthDate)))
        WHEN DATEPART(weekday,DATEFROMPARTS(2016,Month(BirthDate),Day(BirthDate))) = 7
        THEN DATEADD(DAY,2,DATEFROMPARTS(2016,Month(BirthDate),Day(BirthDate)))
        ELSE DATEFROMPARTS(2016,Month(BirthDate),Day(BirthDate))
        END),101) AS [Celebration Date]
    --A copy of the above CASE statement encapsulated in a DATENAME function.
    ,DATENAME(weekday,(CASE
        WHEN DATEPART(weekday,DATEFROMPARTS(2016,Month(BirthDate),Day(BirthDate))) = 1
        THEN DATEADD(DAY,1,DATEFROMPARTS(2016,Month(BirthDate),Day(BirthDate)))
        WHEN DATEPART(weekday,DATEFROMPARTS(2016,Month(BirthDate),Day(BirthDate))) = 7
        THEN DATEADD(DAY,2,DATEFROMPARTS(2016,Month(BirthDate),Day(BirthDate)))
        ELSE DATEFROMPARTS(2016,Month(BirthDate),Day(BirthDate))
        END
    )) AS [Celebration Day of Week]
FROM Employee

```