

```
/*
Case Statement and Aggregate Function homework solution.
Your syntax does not have to match mine exactly.
What is important is that the result sets are the same.
*/
```

```
/*
1.
Display all tracks from the Track table and their associated media type name from the MediaType table.
Create two derived columns called MediaType and EncodingFormat.
Call the Track.Name column TrackName and the MediaType.Name column MediaName.
The result set will have 4 columns: TrackName, MediaName, MediaType and EncodingFormat
For MediaType if the media is a video then display "Video" otherwise display "Audio".
For EncodingFormat if the media name contains AAC then display "AAC", if it contains MPEG then display
"MPEG" otherwise display "Unknown".
*/
```

```
SELECT
    T.Name TrackName
    ,MT.Name MediaName
    ,CASE
        WHEN MT.MediaTypeId = 3 THEN 'Video'
        ELSE 'Audio'
    END MediaType
    ,CASE
        WHEN MT.Name LIKE '%mpeg%' THEN 'MPEG'
        WHEN MT.Name LIKE '%aac%' THEN 'AAC'
        ELSE 'Unknown'
    END EncodingFormat
FROM Track T
LEFT JOIN MediaType MT
    ON MT.MediaTypeId = T.MediaTypeId
ORDER BY MT.Name
```

```
/*
2.
Display the total track count for each Media type.
The result set will have 2 columns: MediaTypeName, TotalTracks.
Hint: The TotalTracks for an MPEG audio file should equal 3034.
*/
```

```
SELECT
    MT.Name AS MediaTypeName
    ,COUNT(*) AS TotalTracks
FROM Track T
JOIN MediaType MT
    ON MT.MediaTypeId = T.MediaTypeId
GROUP BY MT.Name
```

```

/*
3.
Sum the total sales for each Sales Support Agent grouped by year.
The result set should have 4 columns: FirstName, LastName, SaleYear, TotalSales.
Use Invoice.Total for TotalSales and Invoice.InvoiceDate for SaleYear.
Hint: You should return 15 rows. The TotalSales for Steve Johnson in 2009 should equal 164.34.
*/

```

```

SELECT
    E.FirstName
    ,E.LastName
    ,YEAR(InvoiceDate) SaleYear
    ,SUM(I.Total) TotalSales
FROM
Employee E
JOIN Customer C
    ON C.SupportRepId = E.EmployeeId
JOIN Invoice I
    ON I.CustomerId = C.CustomerId
GROUP BY
    E.FirstName
    ,E.LastName
    ,YEAR(InvoiceDate)
ORDER BY LastName, SaleYear

```

```

/*
4.
Display the highest amount paid by each customer for a single invoice.
The result set should have 3 columns: LastName, FirstName and MaxInvoice.
MaxInvoice should be derived from the Invoice.Total column.
Hint: Fynn Zimmermann's MaxInvoice should be 14.91.
*/

```

```

SELECT
    C.FirstName
    ,C.LastName
    ,MAX(Total) MaxInvoice
FROM Customer C
JOIN Invoice I
    ON I.CustomerId = C.CustomerId
GROUP BY
    C.FirstName
    ,C.LastName

```

```

/*
5.
Check customer postal codes to determine if they are numeric.
The result set should have 3 columns: Country, PostalCode and NumericPostalCode.
NumericPostalCode is a derived column.
If the PostalCode column has a numeric value then return "Yes".
If it does not have a numeric value then return "No".
If the PostalCode column has a NULL value then return "Unknown".
Order the results by NumericPostalCode and Country.
*/

```

```

SELECT
    Country
    ,PostalCode
    ,CASE
        WHEN PostalCode IS NULL THEN 'Unknown'
        WHEN ISNUMERIC(Postalcode) = 1 THEN 'Yes'
        WHEN ISNUMERIC(PostalCode) = 0 THEN 'No'
        END NumericPostalCode
FROM Customer
ORDER BY NumericPostalCode, Country

```

```

/*
6.
Find the customers whose total purchases are greater than 42 dollars.
The result set should have 3 columns: FirstName, LastName, TotalSales.
TotalSales is derived from the Invoice table.
*/
SELECT
    FirstName
    ,LastName
    ,SUM(I.Total) TotalSales
FROM Customer C
JOIN Invoice I
    ON C.CustomerId = I.CustomerId
GROUP BY
    FirstName
    ,LastName
HAVING SUM(I.Total) > 42
ORDER BY TotalSales DESC

/*
7.
Which artist has the most tracks in the database?
The result set should contain 1 column named TopArtist, and 1 row with the name of the artist.
(Note: Don't hard code the answer. I need to see the query logic.)
*/
SELECT TOP 1
    A.Name AS TopArtist
    --,count(*)
FROM Artist A
JOIN Album AL
    ON AL.ArtistId = A.ArtistId
JOIN Track T
    ON T.AlbumId = AL.AlbumId
GROUP BY A.Name
ORDER BY COUNT(*) DESC

/*
8.
Assign customers to groups using a derived column named CustomerGrouping.
The result set will have 3 columns: FirstName, LastName and CustomerGrouping.
Customers with a last name starting with A-G will be assigned to Group1.
Customers with a last name starting with H-M will be assigned to Group2.
Customers with a last name starting with N-S will be assigned to Group3.
Customers with a last name starting with T-Z will be assigned to Group4.
If there is no last name then the CustomerGrouping column should return NULL.
*/
SELECT
    FirstName
    ,LastName
    ,CASE
        WHEN LastName LIKE '[A-G]%' THEN 'Group1'
        WHEN LastName LIKE '[H-M]%' THEN 'Group2'
        WHEN LastName LIKE '[N-S]%' THEN 'Group3'
        WHEN LastName LIKE '[T-Z]%' THEN 'Group4'
        END CustomerGrouping
FROM Customer
ORDER BY LastName

```

```
/*
9.
List all the artists and a count of how many albums each artist has in the database.
The result set will have 2 columns: ArtistName and AlbumCount.
Order the results by AlbumCount and ArtistName.
*/
```

```
SELECT
    Name AS ArtistName
    ,count(Title) AS AlbumCount
FROM Artist A
LEFT JOIN Album AL
ON AL.ArtistId = A.ArtistId
GROUP BY Name
ORDER BY AlbumCount, ArtistName
```

```
/*
10.
Place employees in departments based on their title.
The result set will have 4 columns: FirstName, LastName, Title and Department.
Department is derived column with the following criteria:
If an employee's title contains "Sales" then their department is "Sales".
If an employee's title contains "IT" then their department is "Technology".
If an employee's title contains "Manager" then their department is "Management".
The Management department will override Sales and Technology for employee placement.
Order your results by Department.
*/
```

```
SELECT
    FirstName
    ,LastName
    ,Title
    ,CASE
        WHEN Title LIKE '%Manager%' THEN 'Management'
        WHEN Title LIKE '%Sales%' THEN 'Sales'
        WHEN Title LIKE '%IT%' THEN 'Technology'
        ELSE 'Unknown'
    END Department
FROM Employee
ORDER BY Department
```