**CAR SALES REPORT – SQL JOINS**

**Here four tables are:**

* CarBrands,
* CarModels,
* CarSales, and
* Customers

**Inner Join:**

INNER JOIN combines rows from two or more tables based on a related column between them. It returns only the rows where there is a match between the columns in both tables.

SELECT CarBrands.BrandName, CarModels.ModelName, CarSales.SaleDate, CarSales.SalePrice

FROM CarBrands

INNER JOIN CarModels ON CarBrands.BrandID = CarModels.BrandID

INNER JOIN CarSales ON CarModels.ModelID = CarSales.ModelID;

A screenshot of a computer

Description automatically generated

We are selecting data from the CarBrands, CarModels, and CarSales tables.

The INNER JOIN is used to join the CarBrands and CarModels tables based on the BrandID column, ensuring that only rows with matching BrandID values in both tables are returned.

Then, another INNER JOIN is used to join the resulting set with the CarSales table based on the ModelID column, ensuring that only rows with matching ModelID values in all three tables are returned.

The resulting dataset contains columns from all three tables, but only the rows where there is a match across all joined tables. Any rows where there is no match in any of the joined tables are excluded from the result.

**Left Join:**

The LEFT JOIN returns all rows from the left table (the table mentioned first in the query) and the matched rows from the right table (the table mentioned second in the query). If there is no match, the result is NULL on the right side.

SELECT CarBrands.BrandName, CarModels.ModelName, CarSales.SaleDate, CarSales.SalePrice

FROM CarBrands

LEFT JOIN CarModels ON CarBrands.BrandID = CarModels.BrandID

LEFT JOIN CarSales ON CarModels.ModelID = CarSales.ModelID;

A screenshot of a computer

Description automatically generated

We are selecting data from the CarBrands, CarModels, and CarSales tables.

The first LEFT JOIN is used to join the CarBrands table with the CarModels table based on the BrandID column. It ensures that all rows from CarBrands are returned, and only matching rows from CarModels are included. If there's no matching BrandID in CarModels, the columns from CarModels will be NULL in the result set.

Then, another LEFT JOIN is used to join the resulting set with the CarSales table based on the ModelID column. This ensures that all rows from the previously joined set are returned, and only matching rows from CarSales are included. If there's no matching ModelID in CarSales, the columns from CarSales will be NULL in the result set.

The result of the LEFT JOIN will include all rows from CarBrands, and any matching rows from CarModels and CarSales. If there's no match in CarModels or CarSales, NULL values will be used for the respective columns.

**Right Join:**

RIGHT JOIN returns all rows from the right table (the table mentioned second in the query) and the matched rows from the left table (the table mentioned first in the query). If there is no match, the result is NULL on the left side.

SELECT CarBrands.BrandName, CarModels.ModelName, CarSales.SaleDate, CarSales.SalePrice

FROM CarBrands

RIGHT JOIN CarModels ON CarBrands.BrandID = CarModels.BrandID

RIGHT JOIN CarSales ON CarModels.ModelID = CarSales.ModelID;

A screenshot of a computer

Description automatically generated

We are selecting data from the CarBrands, CarModels, and CarSales tables.

The first RIGHT JOIN is used to join the CarModels table with the CarBrands table based on the BrandID column. It ensures that all rows from CarModels are returned, and only matching rows from CarBrands are included. If there's no matching BrandID in CarBrands, the columns from CarBrands will be NULL in the result set.

Then, another RIGHT JOIN is used to join the resulting set with the CarSales table based on the ModelID column. This ensures that all rows from the previously joined set are returned, and only matching rows from CarSales are included. If there's no matching ModelID in CarSales, the columns from CarSales will be NULL in the result set.

The result of the RIGHT JOIN will include all rows from CarModels, and any matching rows from CarBrands and CarSales. If there's no match in CarBrands or CarSales, NULL values will be used for the respective columns.

**Full Outer Join:**

FULL OUTER JOIN combines the results of both left and right joins, ensuring that all rows from both tables are included in the result set, even if there is no matching row in the other table.

SELECT CarBrands.BrandName, CarModels.ModelName, CarSales.SaleDate, CarSales.SalePrice

FROM CarBrands

FULL OUTER JOIN CarModels ON CarBrands.BrandID = CarModels.BrandID

FULL OUTER JOIN CarSales ON CarModels.ModelID = CarSales.ModelID;

A screenshot of a computer

Description automatically generated

Here in the query, the CarBrands table is joined with the CarModels table using a full outer join based on the BrandID.

Then, the result of that join is further joined with the CarSales table using a full outer join based on the ModelID.

This ensures that all rows from both CarBrands and CarModels, as well as all rows from CarSales, are included in the result set, with matching rows from other tables where applicable, and NULL values where there's no match.

**List of customers who bought cars:**

SELECT

Customers.CustomerID,

Customers.CustomerName,

CarBrands.BrandName,

CarModels.ModelName,

CarSales.SaleDate,

CarSales.SalePrice

FROM

Customers

INNER JOIN

CarSales ON Customers.SaleID = CarSales.SaleID

INNER JOIN

CarModels ON CarSales.ModelID = CarModels.ModelID

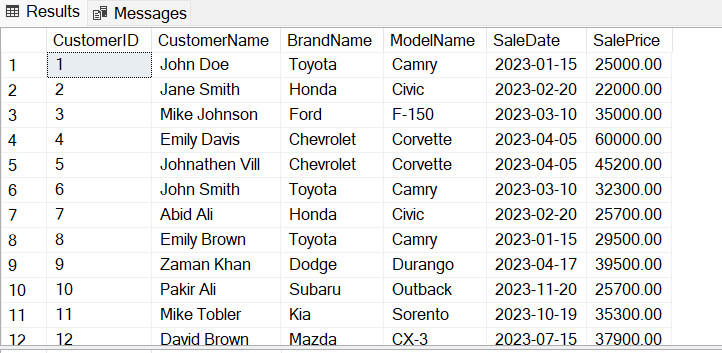
INNER JOIN

CarBrands ON CarModels.BrandID = CarBrands.BrandID;

SELECT DISTINCT C1.CustomerID, C1.CustomerName

FROM Customers C1

INNER JOIN Customers C2 ON C1.CustomerID = C2.CustomerID AND C1.SaleID <> C2.SaleID;

****

**Inner Join with Aggregation**:

Here we want to find the total sales amount for each brand:

SELECT CB.BrandName, SUM(CS.SalePrice) AS TotalSales

FROM CarBrands CB

INNER JOIN CarModels CM ON CB.BrandID = CM.BrandID

INNER JOIN CarSales CS ON CM.ModelID = CS.ModelID

GROUP BY CB.BrandName;

**A screenshot of a computer

Description automatically generated**

**Left Join with Filtering:**

If we want to list all customers along with their purchased car details:

SELECT C.CustomerName, CB.BrandName, CM.ModelName, CS.SaleDate, CS.SalePrice

FROM Customers C

LEFT JOIN CarSales CS ON C.SaleID = CS.SaleID

LEFT JOIN CarModels CM ON CS.ModelID = CM.ModelID

LEFT JOIN CarBrands CB ON CM.BrandID = CB.BrandID;

A screenshot of a computer

Description automatically generated