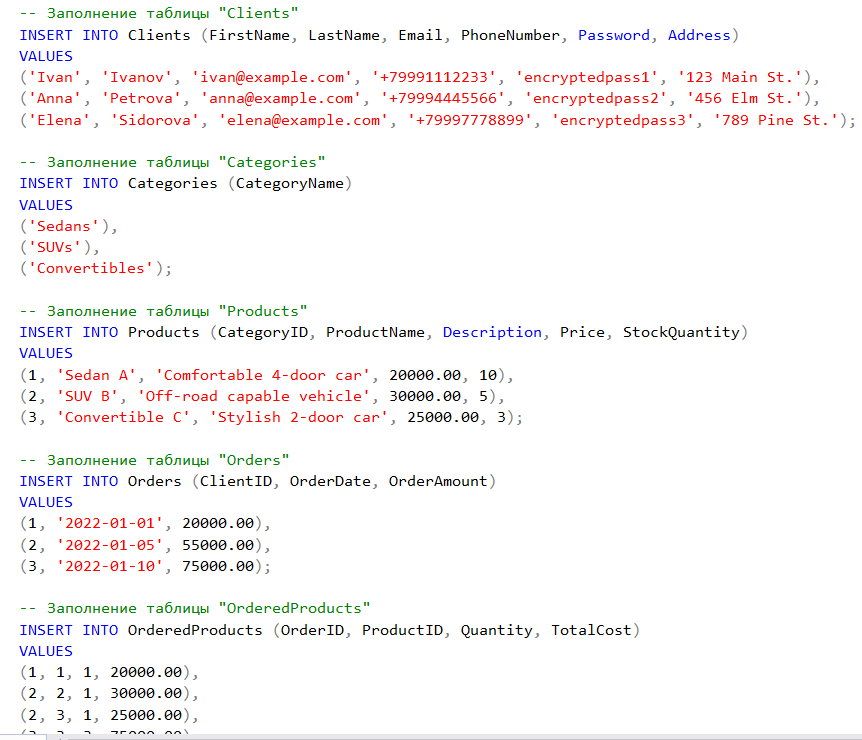
*Лабораторная работа № 4*

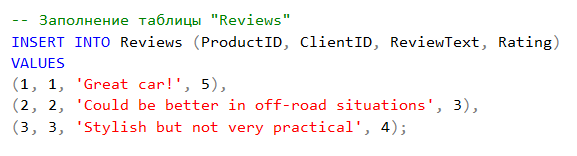
**Расширенные группировки. Аналитические функции.**

Задание:

1. Добавьте в обе базы данных достаточное число правдоподобных данных. Можно генерировать данные и/или использовать импорт.

MS SQL:





Oracle:

|  |
| --- |
| INSERT INTO Clients (FirstName, LastName, Email, PhoneNumber, Password, Address)  VALUES ('Иван', 'Иванов', 'ivan@example.com', '+71234567890', 'encryptedpassword1', 'Москва, ул. Ленина, 1');  INSERT INTO Clients (FirstName, LastName, Email, PhoneNumber, Password, Address)  VALUES ('Влад', 'Владов', 'vlad@example.com', '+71234567890', 'encrypt123password1', 'Минск, ул. Ленина, 1');  INSERT INTO Clients (FirstName, LastName, Email, PhoneNumber, Password, Address)  VALUES ('Мария', 'Петрова', 'maria@example.com', '+71234567891', 'encryptedpassword2', 'Санкт-Петербург, пр. Невский, 2');  -- Добавляем категории автомобилей  INSERT INTO Categories (CategoryName)  VALUES ('Седан');  INSERT INTO Categories (CategoryName)  VALUES ('Кроссовер');  INSERT INTO Categories (CategoryName)  VALUES ('Хэтчбек');  -- Добавляем автомобили  INSERT INTO Products (CategoryID, ProductName, Description, Price, StockQuantity)  VALUES (1, 'Toyota Camry', 'Седан, белого цвета, 2022 года выпуска', 2000000.00, 3);  INSERT INTO Products (CategoryID, ProductName, Description, Price, StockQuantity)  VALUES (2, 'Nissan Qashqai', 'Кроссовер, красного цвета, 2021 года выпуска', 1500000.00, 2);  INSERT INTO Products (CategoryID, ProductName, Description, Price, StockQuantity)  VALUES (3, 'Volkswagen Golf', 'Хэтчбек, синего цвета, 2020 года выпуска', 1400000.00, 4);  -- Добавляем заказы  INSERT INTO Orders (ClientID, OrderDate, OrderAmount)  VALUES (1, SYSDATE, 2000000.00);  INSERT INTO Orders (ClientID, OrderDate, OrderAmount)  VALUES (2, SYSDATE, 20000.00);  INSERT INTO Orders (ClientID, OrderDate, OrderAmount)  VALUES (21, SYSDATE, 400000.00);  -- Добавляем заказанные автомобили  INSERT INTO OrderedProducts (OrderID, ProductID, Quantity, TotalCost)  VALUES (1, 1, 1, 2000000.00);  INSERT INTO OrderedProducts (OrderID, ProductID, Quantity, TotalCost)  VALUES (21, 2, 1, 20000.00);  INSERT INTO OrderedProducts (OrderID, ProductID, Quantity, TotalCost)  VALUES (23, 3, 2, 800000.00);  -- Добавляем отзывы  INSERT INTO Reviews (ProductID, ClientID, ReviewText, Rating)  VALUES (1, 1, 'Отличный автомобиль, очень доволен покупкой!', 5); |

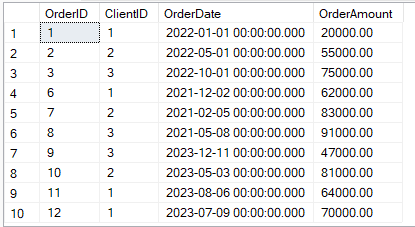
1. В этом задании результатом является **один запрос**. Найдите:

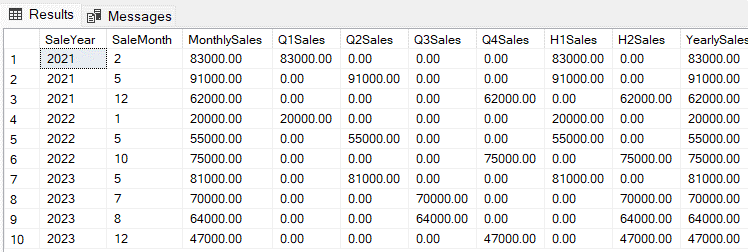
|  |  |
| --- | --- |
| Магазин | Вычисление итогов работы продавцов помесячно, за квартал, за полгода, за год. |

MS SQL

|  |
| --- |
| -- 4.1  -- quarter = 12/4=3 month, if jan-may - 1, apr-june - 2, july-sept - 3, oct-dec - 4  -- Common Table Expression (CTE) DateMetrics - breaks down order dates into year, month, quarter, and half-year components  WITH DateMetrics AS (  SELECT  OrderID,  OrderAmount,  DATEPART(YEAR, OrderDate) AS SaleYear,  DATEPART(MONTH, OrderDate) AS SaleMonth,  DATEPART(QUARTER, OrderDate) AS SaleQuarter,  -- to define wich first or second half a year  CASE  WHEN DATEPART(MONTH, OrderDate) <= 6 THEN 1  ELSE 2  END AS SaleHalfYear  FROM Orders  )  -- The main SELECT then aggregates sales data for each month and provides the results for monthly, quarterly, half-yearly, and yearly metrics.  SELECT  SaleYear,  SaleMonth,  SUM(OrderAmount) AS MonthlySales,  SUM(CASE WHEN SaleQuarter = 1 THEN OrderAmount ELSE 0 END) AS Q1Sales,  SUM(CASE WHEN SaleQuarter = 2 THEN OrderAmount ELSE 0 END) AS Q2Sales,  SUM(CASE WHEN SaleQuarter = 3 THEN OrderAmount ELSE 0 END) AS Q3Sales,  SUM(CASE WHEN SaleQuarter = 4 THEN OrderAmount ELSE 0 END) AS Q4Sales,  SUM(CASE WHEN SaleHalfYear = 1 THEN OrderAmount ELSE 0 END) AS H1Sales,  SUM(CASE WHEN SaleHalfYear = 2 THEN OrderAmount ELSE 0 END) AS H2Sales,  SUM(OrderAmount) AS YearlySales  FROM DateMetrics  GROUP BY SaleYear, SaleMonth  ORDER BY SaleYear, SaleMonth; |

Вот список всех Заказов:



Сначала идет сортировка по году, а потом по месяцам:

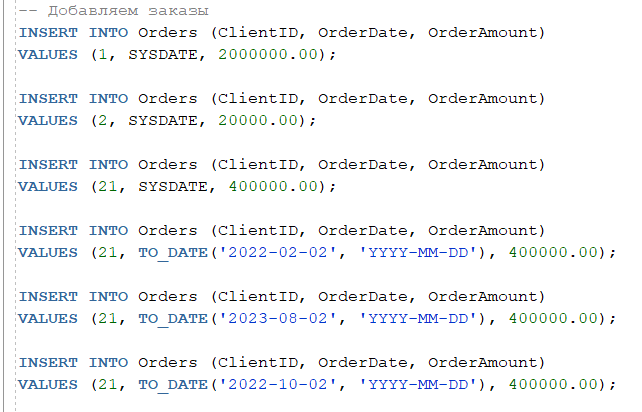
Oracle

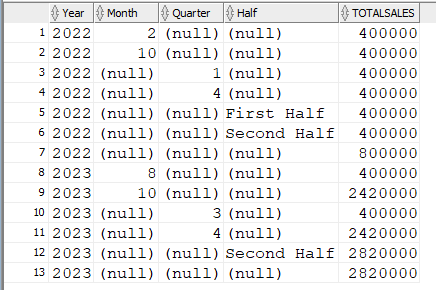
|  |
| --- |
| WITH SalesAggregation AS (  -- Monthly sales  SELECT  EXTRACT(YEAR FROM OrderDate) AS "Year",  EXTRACT(MONTH FROM OrderDate) AS "Month",  NULL AS "Quarter",  NULL AS "Half",  SUM(OrderAmount) AS TotalSales  FROM Orders  GROUP BY  EXTRACT(YEAR FROM OrderDate),  EXTRACT(MONTH FROM OrderDate)    UNION ALL    -- Quarterly sales  SELECT  EXTRACT(YEAR FROM OrderDate) AS "Year",  NULL AS "Month",  TO\_NUMBER(TO\_CHAR(OrderDate, 'Q')) AS "Quarter",  NULL AS "Half",  SUM(OrderAmount) AS TotalSales  FROM Orders  GROUP BY  EXTRACT(YEAR FROM OrderDate),  TO\_NUMBER(TO\_CHAR(OrderDate, 'Q'))    UNION ALL    -- Half yearly sales  SELECT  EXTRACT(YEAR FROM OrderDate) AS "Year",  NULL AS "Month",  NULL AS "Quarter",  CASE  WHEN EXTRACT(MONTH FROM OrderDate) <= 6 THEN 'First Half'  ELSE 'Second Half'  END AS "Half",  SUM(OrderAmount) AS TotalSales  FROM Orders  GROUP BY  EXTRACT(YEAR FROM OrderDate),  CASE  WHEN EXTRACT(MONTH FROM OrderDate) <= 6 THEN 'First Half'  ELSE 'Second Half'  END    UNION ALL    -- Yearly sales  SELECT  EXTRACT(YEAR FROM OrderDate) AS "Year",  NULL AS "Month",  NULL AS "Quarter",  NULL AS "Half",  SUM(OrderAmount) AS TotalSales  FROM Orders  GROUP BY  EXTRACT(YEAR FROM OrderDate)  )  SELECT \*  FROM SalesAggregation  ORDER BY  "Year",  "Month" NULLS LAST,  "Quarter" NULLS LAST,  "Half" NULLS LAST; |

Первые две строки - выводятся месяцы для 2022: 2, 10.

Вторые две строки – кварталы. Т.к. 2 – 1 квартал, а 10 месяц – 4 квартал.

Третьи две строки – полугоды. Далее идет тоже самое для 2023 года.





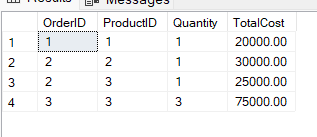
|  |  |
| --- | --- |
| Магазин | Вычисление итогов работы продавцов за определенный период:   * объем продаж; * сравнение их с общим объемом продаж (в %); * сравнение с наилучшим объемом продаж (в %). |

CTE (Common Table Expression) — это временный результат запроса, который существует только во время выполнения одного SQL-запроса. CTE предоставляет средства для разбиения сложных SQL-запросов на простые блоки, что делает запросы более понятными и обеспечивает возможность повторного использования.

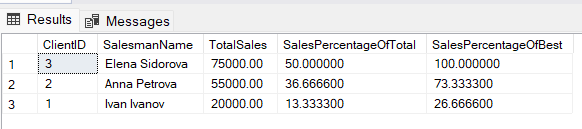
MS SQL

|  |
| --- |
| WITH SalesSummary AS (  SELECT  o.ClientID,  SUM(op.TotalCost) AS TotalSales  FROM Orders o  JOIN OrderedProducts op ON o.OrderID = op.OrderID  GROUP BY o.ClientID  ),  TotalSales AS (  SELECT SUM(TotalSales) AS GrandTotalSales FROM SalesSummary  ),  MaxSales AS (  SELECT MAX(TotalSales) AS BestSales FROM SalesSummary  )  SELECT  c.ClientID,  c.FirstName + ' ' + c.LastName AS SalesmanName,  ss.TotalSales,  (ss.TotalSales / ts.GrandTotalSales \* 100) AS SalesPercentageOfTotal,  (ss.TotalSales / ms.BestSales \* 100) AS SalesPercentageOfBest  FROM SalesSummary ss  JOIN Clients c ON ss.ClientID = c.ClientID  CROSS JOIN TotalSales ts  CROSS JOIN MaxSales ms  ORDER BY ss.TotalSales DESC; |

Total cost for every Client (select \* from OrderedProduct)



The result of final query



ORACLE

|  |
| --- |
| WITH SalesSummary AS (  SELECT  o.ClientID,  SUM(op.TotalCost) AS TotalSales  FROM Orders o  JOIN OrderedProducts op ON o.OrderID = op.OrderID  GROUP BY o.ClientID  ),  TotalSales AS (  SELECT SUM(TotalSales) AS GrandTotalSales FROM SalesSummary  ),  MaxSales AS (  SELECT MAX(TotalSales) AS BestSales FROM SalesSummary  )  SELECT  c.ClientID,  c.FirstName || ' ' || c.LastName AS SalesmanName,  ss.TotalSales,  (ss.TotalSales / ts.GrandTotalSales \* 100) AS SalesPercentageOfTotal,  (ss.TotalSales / ms.BestSales \* 100) AS SalesPercentageOfBest  FROM SalesSummary ss  JOIN Clients c ON ss.ClientID = c.ClientID  CROSS JOIN TotalSales ts  CROSS JOIN MaxSales ms  ORDER BY ss.TotalSales DESC; |

