Невретдинов Марат 2-ИП

Отчёт по практике МДК 11.01 Технология разработки и защиты баз данных

**Задание:**

Разработать клиент-серверное приложение, серверная часть которой

реализована на MySQL Server (в моём случае на MsSql), представляющая собой модель предметной области в соответствии с вариантом задания. В рамках заданной предметной области реализовать заданную (по варианту) схему отношений, т.е. выделить сущности и их атрибуты, так чтобы связи между сущностями соответствовали представленной схеме.

**Тема:**

Учет трудовых ресурсов (кадров) предприятия инспектором отдела кадров.

Так как моя тема связана с отделом кадров, то я выделил для своей БД 4 сущности:

**Отдел кадров:**

В отделе кадров хранится информация о id отдела кадров (где он является первичным ключом), его названии, адреса, номера телефона и о количестве сотрудников.

**Сотрудники:**

Сотрудники также имеют свой id (где он является первичным ключом), внешний ключ в виде id отдела кадров для вычисления в каком отделе он работает. Также имеют имя, фамилию и телефон.

**Должности:**

У должности имеется свой id (где он является первичным ключом), внешний ключ в виде id сотрудника. Также присутствует время работы по полным годам, какая должность и ЗП.

**Документы:**

У документов имеются свои id (где они являются первичным ключом), внешний ключ в виде id сотрудника. Также присутствует название документа, его официальная достоверность и дата регистрации.

1. **Составной многотабличный запрос с параметром, включающий соединение таблиц и CASE-выражение**

SELECT TOP 10 Orders.Id, RentId, TotalAmount,

CASE

WHEN TotalAmount >= 500 AND TotalAmount < 750 THEN 'Заказы от 500 до 750 р'

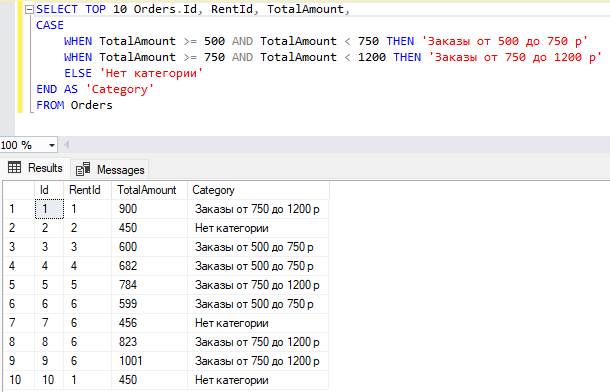
WHEN TotalAmount >= 750 AND TotalAmount < 1200 THEN 'Заказы от 750 до 1200 р'

ELSE 'Нет категории'

END AS 'Category'

FROM Orders

Результат запроса отображен на рисунке 1.



**Рисунок 1 –** Результат Case запроса

1. **На основе обновляющего представления (многотабличного VIEW), в котором критерий упорядоченности задает пользователь при выполнении**

SELECT dbo.Orders.Id, dbo.CarBrands.Name

AS CarBrand, dbo.Users.FIrstName AS DriverName, dbo.Customers.Name

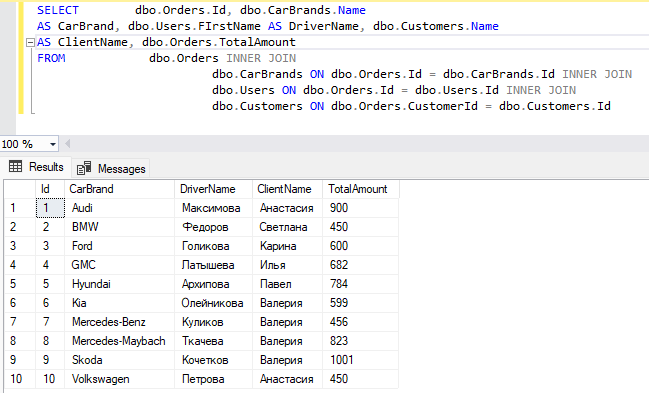
AS ClientName, dbo.Orders.TotalAmount

FROM dbo.Orders INNER JOIN

dbo.CarBrands ON dbo.Orders.Id = dbo.CarBrands.Id INNER JOIN

dbo.Users ON dbo.Orders.Id = dbo.Users.Id INNER JOIN

dbo.Customers ON dbo.Orders.CustomerId = dbo.Customers.Id



**Рисунок 2 –** Результатвыборки VIEW запроса

1. **Запрос, содержащий коррелированные и некоррелированные подзапросы в разделах SELECT, FROM и WHERE (в каждом хотя бы по одному)**

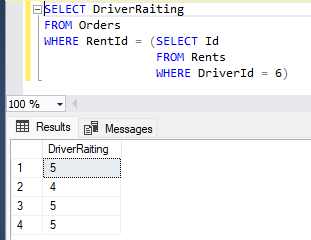
SELECT DriverRaiting

FROM Orders

WHERE RentId = (SELECT Id

FROM Rents

WHERE DriverId = 6)



**Рисунок 3 –** Результат запроса

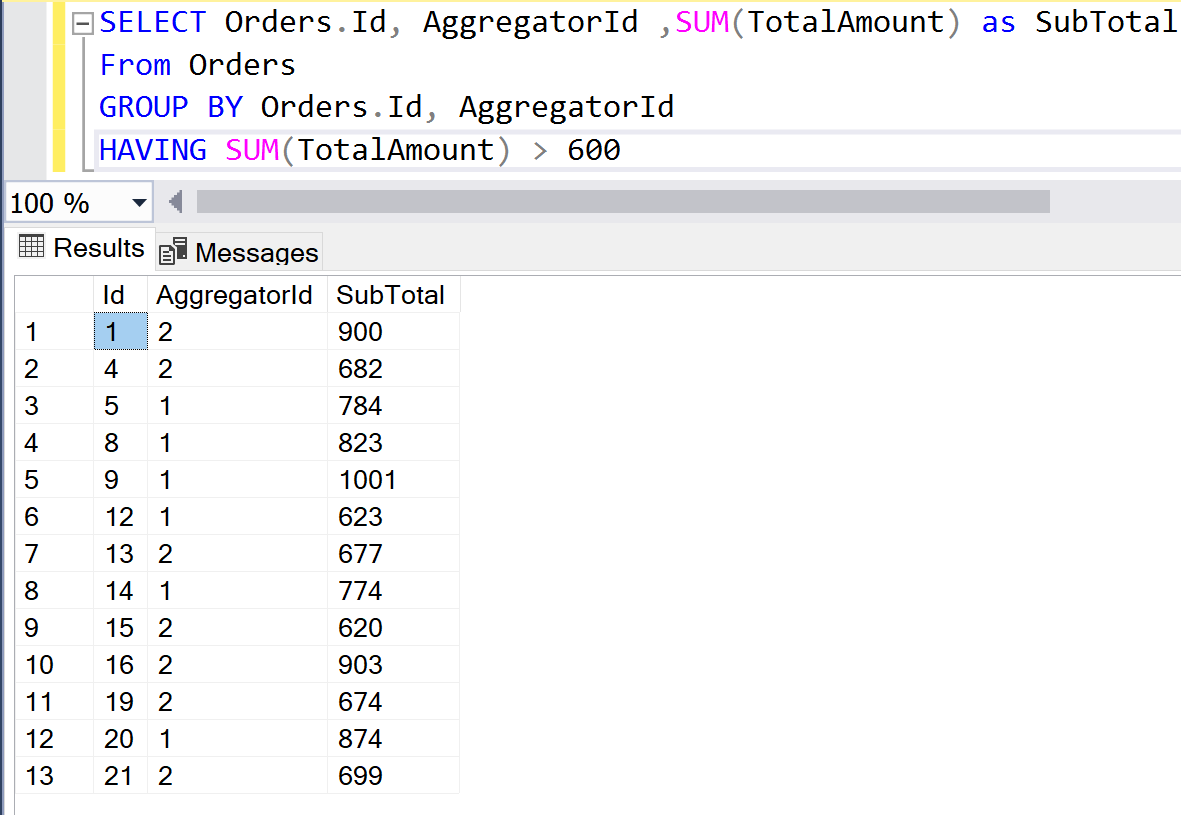
1. **Многотабличный запрос, содержащий группировку записей, агрегативные функции и параметр, используемый в разделе HAVING**

SELECT Orders.Id, AggregatorId ,SUM(TotalAmount) as SubTotal

From Orders

GROUP BY Orders.Id, AggregatorId

HAVING SUM(TotalAmount) > 600



**Рисунок 4 –** Результат запроса Having

1. **Запрос, содержащий предикат ANY, SOME или ALL**

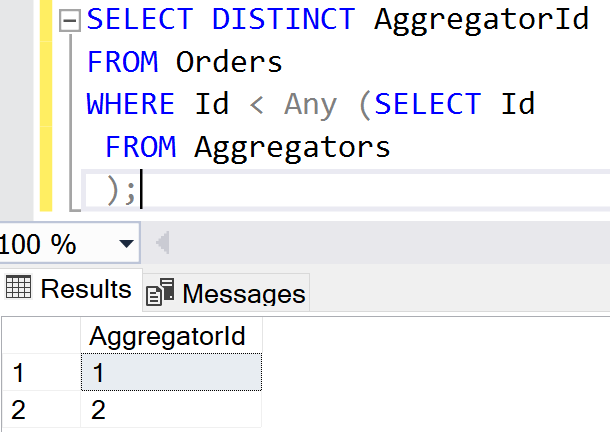
SELECT DISTINCT AggregatorId

FROM Orders

WHERE Id < Any (SELECT Id

FROM Aggregators

);



**Рисунок 5 –** Результат запроса Any