**Отчет**

По итоговому проекту

**Выполнили:**

Бобринский Павел,

Внукова Екатерина

## Раздел 1. Описание базы данных.

Предположим, что существует некоторая компания, которая занимается продажей автомобилей, под названием “Автомобильный салон VSHE”. Наша база данных представляет собой укороченную версию полноценной БД данной компании. Она включает в себя 5 таблиц, которые мы назвали “Employees”, “Departaments”, “Clients”, “Deals”, “car\_mark”, “Car\_types”. Таблица “Employees” хранит всю информацию по каждому сотруднику компании (ID сотрудника, Имя, должность, заработная плата, пол, номер отдела). В таблице “Clients” находится также вся необходимая информация по каждому клиенту (ID клиента, Номер телефона, дата рождения). Таблица “Departaments” отражает названия отделов, количество сотрудников в каждом отделе, план, а также факт продаж для каждого отдела. В таблице “ car\_mark ” мы отразили всю информацию об автомобилях компании, точнее о тех, которые есть в наличие. Сюда вошли такие столбцы как марка автомобиля, количество автомобилей в наличие, пробег каждой машины и ID автомобиля, и ID марки. Также есть таблица с древовидной структурой, хранимая в виде вложенных множеств. Называется она “Car\_types”. Она состоит из ID машины, типа машины и два столбца, отмечающиеся за выход и вход в узел. Наконец, последняя таблица “Deals” показывает взаимодействие между сотрудниками и клиентами в виде предоставления услуг купли-продаж. Таблица содержит информацию об ID сотрудника и ID клиента, заключившие сделку, дату покупки, ID автомобиля, количество проданных машин за 1 сделку и ID сделки.

Основная идея нашего проекта такова: мы попытались представить, что в данной компании мы работаем аналитиками данных и перед нами стоит задача: проанализировать в компании отдел продаж и подобрать наиболее эффективную премиальную систему (или систему бонусов), которая будет способствовать не только увеличению чистой прибыли компании, но и будет стимулировать отдел продаж чаще заключать сделки и больше продавать машин.

## Раздел 2. ER- диаграмма.

На рис.2 представлена ER-диаграмма по базе данных.

**Сущности**.

В нашей диаграмме 6 сущностей, которые соответствуют названию всех таблиц:

1. “Employees”;
2. “Departaments”;
3. “Clients”;
4. “Deals”;
5. “car\_mark”;
6. “Car\_types”.

**Атрибуты сущностей.**

Для сущности “Employees” атрибутами являются:

* ID сотрудника (ID\_employee);
* Имя сотрудника (Name);
* Должность сотрудника (Position);
* Зарплата сотрудника (Salary);
* Пол сотрудника (Sex);
* Номер отдела (DepNo);

Для сущности “Departaments” атрибутами являются:

* Номер отдела (DepNo);
* Количество сотрудников (Amount);
* План продаж (Sales\_plan);
* Факт продаж (Fact\_of\_sales);

Для сущности “Clients” атрибутами являются:

* ID клиента (ID\_client);
* Номер телефона (PhoneNum);
* Дата рождения клиента (Birthday);

Для сущности “Deals” атрибутами являются:

* ID сотрудника (ID\_employee);
* ID клиента (ID\_client);
* Дата продажи автомобиля (Deal\_date);
* ID марки (ID\_mark);
* Количество автомобилей (Amount);
* ID сделки (ID\_deal);

Для сущности “car\_mark” атрибутами являются:

* Марка машины (Mark);
* Количество машин (Quantity);
* Пробег машины (Mileage);
* ID марки (ID\_mark);
* ID автомобиля (ID\_car);

Для сущности “Car\_types” атрибутами являются:

* ID машины (Id\_car);
* Тип машины (typecar);
* Порядок вхождение (lft);
* Номер выхода из узла (rgt);

**Связи между сущностями.**

Employees(N) --- Clients(P)

* Здесь происходит взаимодействие между покупателем и продавцом с целью совершения сделки (сюда входит обсуждение понравившейся модели, цены и условий страхования).

Employees(N) --- Departaments (M)

* Каждый сотрудник в компании работает в определенном отделе.

Employees(N) --- Deals(K)

* Сотрудники оформляют необходимые документы и подписывают договор для заключения сделки.

Clients(P) --- car\_mark(C)

* Клиенты делают то, ради чего они пришли в автосалон – покупают автомобиль.

Clients(P) --- Deals(K)

* Без бумажной работы в сделке на покупку не обойтись, поэтому клиенты, как представители стороны потребителя, должны подписать определенные документы и заключить сделку официально.

car\_mark (K) --- Deals(K)

* После официального заключения сделки происходит уже передача денежных средств и приобретения клиентом нового автомобиля.

car\_mark(K) --- Car\_types(D)

* Car\_types – это параметры машины, которые влияют на ее ценник.

Теперь запишем все в виде регуляционной схемы и обозначим первичные, внешние и составные ключи и типы столбцов.

Будем обозначать первичные ключи нижним подчеркиванием, а внешние ключи выделять **жирным шрифтом.**

Employees {[ ID\_employee: string, Name: string, Position: string, Salary: int, Sex: string, **DepNo**: string]}

Departaments {[ DepNo: string, Amount: int, Sales\_plan: int, Fact\_of\_sales: int]}

Clients {[ID\_client: string, PhoneNum: string, Birthday: date]}

Deals {[**ID\_client**: string, **ID\_employee**: string, Deal\_date: date, **ID\_mark**: int, Amont: int, ID\_deal: string]}

car\_mark {[Mark: string, Quantity: int, Mileage: int, **ID\_car**: string, ID\_mark: int ]}

Car\_types{[ ID\_car: string, typecar: string, lft: int, rgt: int ]}

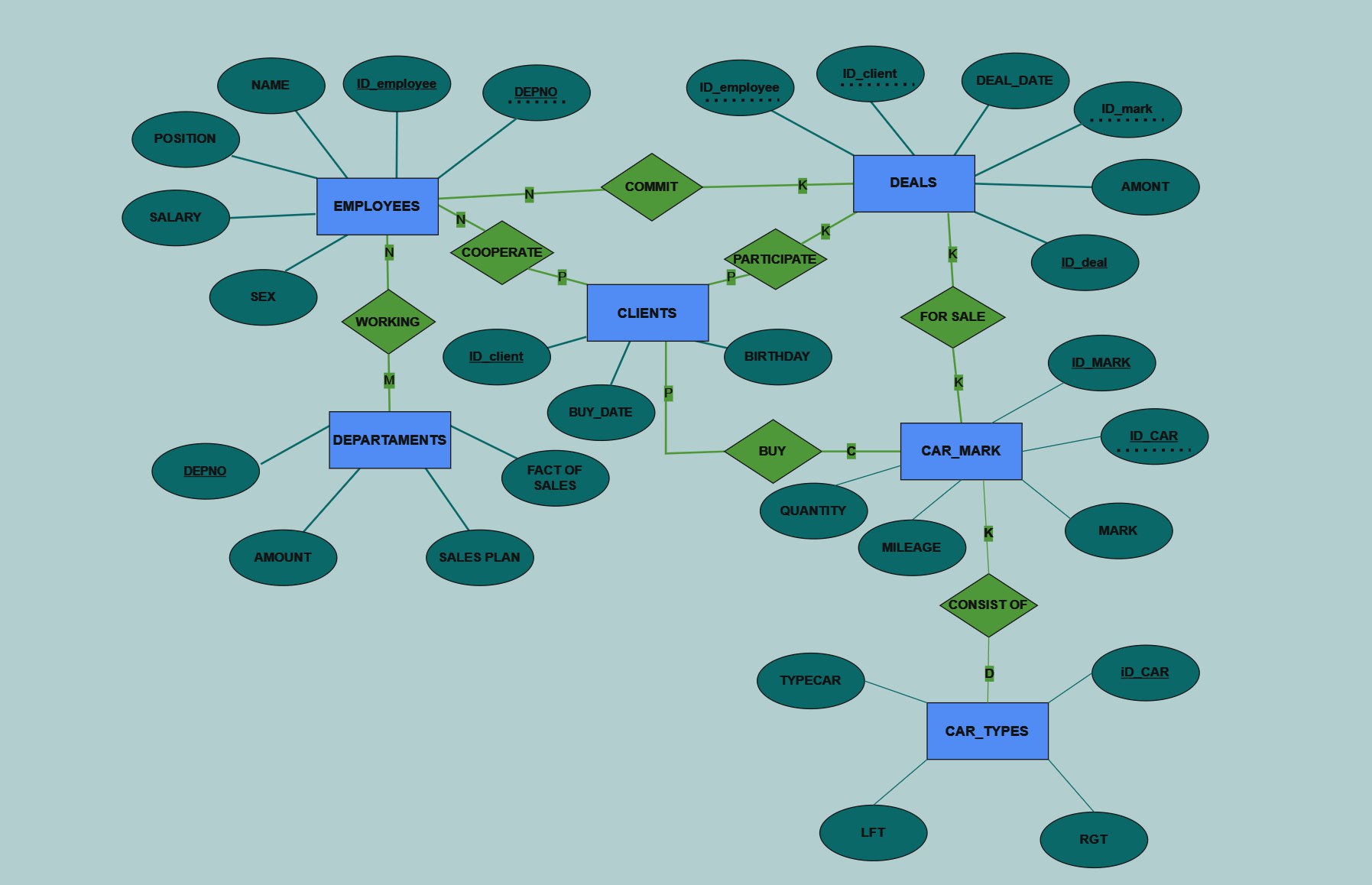


Рис.2

## Раздел 3. Синтаксис SQL для создания и заполнения таблиц данными.

**Весь синтаксис SQL выполнялся на сайте** [**https://sqliteonline.com/**](https://sqliteonline.com/)

--Создаем таблицу Клиенты

DROP TABLE IF EXISTS Clients;

CREATE TABLE Clients (

ID\_client VARCHAR(200) PRIMARY Key,

PhoneNum VARCHAR(200),

Birthday Date);

--Заполняем поля таблицы Клиенты

insert into Clients values ('01', '7-916-230-32-35', '25-08-1980' );

insert into Clients values ('02', '7-903-543-32-11','13-02-1995' );

insert into Clients values ('03', '7-968-456-85-13', '15-08-2000' );

insert into Clients values ('04', '7-916-525-27-14','02-06-1979' );

insert into Clients values ('05', '7-968-853-37-15','09-11-2001' );

insert into Clients values ('06', '7-916-635-27-27', '23-08-1999' );

insert into Clients values ('07', '7-903-862-63-73', '01-07-1988' );

insert into Clients values ('08', '7-916-258-89-26','22-12-1975' );

insert into Clients values ('09', '7-968-269-32-83','03-09-2002' );

insert into Clients values ('010', '7-903-247-19-26', '15-03-1998' );

insert into Clients values ('011', '7-916-852-01-37','21-05-2001' );

insert into Clients values ('012', '7-916-824-00-00','25-08-2000' );

--Создаем таблицу Отделы

DROP TABLE IF EXISTS Departaments;

CREATE TABLE Departaments (

Amount INT,

Sales\_plan INT,

Fact\_of\_sales INT DEFAULT 0,

DepNo VARCHAR(160) PRIMARY key);

--Заполняем поля таблицы Отделы

insert into Departaments values (5,70000000 , 0 , '01Продажи');

insert into Departaments values (5,70000000 , 0 , '02Продажи');

insert into Departaments values (3,30000000 , 0 , '03Продажи');

insert into Departaments values (4,60000000 , 0 , '04Продажи');

insert into Departaments values (2,15000000 , 0 , '05Продажи');

--Создаем таблицу Сотрудники

DROP TABLE IF EXISTS Employees;

CREATE TABLE Employees (

ID\_employee VARCHAR(180),

Name VARCHAR(180) NOT NULL,

Position VARCHAR(100),

Salary INT,

Sex VARCHAR(3),

DepNo VARCHAR(160) REFERENCES Departaments(DepNo),

PRIMARY KEY (ID\_employee, DepNo),

CHECK (Sex IN ('м', 'ж')),

FOREIGN KEY (DepNo) REFERENCES Departaments (DepNo));

--Заполняем поля таблицы Сотрудники

insert into Employees values ('101','Петров Иван', 'генеральный директор' , 200000, 'м', '01Продажи');

insert into Employees values ('301','Александрова Алина', 'менеджер по продажам' , 100000, 'ж', '03Продажи');

insert into Employees values ('201','Мельников Андрей', 'менеджер по продажам' , 25000, 'м', '02Продажи');

insert into Employees values ('302','Филимонов Андрей', 'менеджер по продажам' , 250000, 'м', '03Продажи');

insert into Employees values ('102','Андреева Анна', 'менеджер по продажам' , 250000, 'ж', '01Продажи');

insert into Employees values ('303','Карпов Тихон', 'менеджер по продажам' , 300000, 'м', '03Продажи');

insert into Employees values ('401','Некрасов Дмитрий', 'менеджер по продажам' , 40000, 'м', '04Продажи');

Insert into Employees values ('402','Иванов Максим', 'менеджер по продажам' , 200000, 'м', '04Продажи');

insert into Employees values ('103','Орлов Максим', 'менеджер по продажам' , 30000, 'м', '01Продажи');

insert into Employees values ('202','Касаткина Милана', 'менеджер по продажам' , 30000, 'ж', '02Продажи');

insert into Employees values ('403','Тарасов Артем', 'менеджер по продажам' , 200000, 'м', '04Продажи');

insert into Employees values ('501','Сизова Екатерина', 'менеджер по продажам' , 20000, 'ж', '05Продажи');

insert into Employees values ('502','Фокин Виктор', 'менеджер по продажам' , 30000, 'м', '05Продажи');

insert into Employees values ('404','Борисова Мария', 'менеджер по продажам' , 27000, 'ж', '04Продажи');

insert into Employees values ('104','Грекова Елена', 'менеджер по продажам' , 35000, 'ж', '01Продажи');

insert into Employees values ('203','Золотова Евгения', 'менеджер по продажам' , 35000, 'ж', '02Продажи');

insert into Employees values ('204','Маслова Анастасия', 'менеджер по продажам' , 20000, 'ж', '02Продажи');

insert into Employees values ('105','Корнеев Виктор', 'менеджер по продажам' , 25000, 'м', '01Продажи');

insert into Employees values ('205','Сергеев Филипп', 'менеджер по продажам' , 35000, 'м', '02Продажи');

--Создаем таблицу Тип машин

DROP TABLE IF EXISTS Car\_types;

CREATE TABLE Car\_types(

Id\_car VARCHAR(180) PRIMARY KEY,

typecar VARCHAR(180) NOT NULL,

lft INT,

rgt INT,

CHECK (lft > 0),

CHECK (rgt > 1),

CHECK (lft < rgt)

);

--Заполняем поля таблицы Тип машин

insert into Car\_types values ('1', 'Sedan', 1, 8);

INSERT INTO Car\_types VALUES ('4', 'All-Wheel Drive', 2, 3);

INSERT INTO Car\_types VALUES ('5', 'Rear-Wheel Drive', 4, 5);

INSERT INTO Car\_types VALUES ('6', 'Front-Wheel Drive', 6, 7);

insert into Car\_types values ('2', 'Hatchback', 9, 16);

INSERT INTO Car\_types VALUES ('7', 'All-Wheel Drive', 10, 11);

INSERT INTO Car\_types VALUES ('8', 'Rear-Wheel Drive', 12, 13);

INSERT INTO Car\_types VALUES ('9', 'Front-Wheel Drive', 14, 15);

insert into Car\_types values ('3', 'Jeep', 17, 24);

INSERT INTO Car\_types VALUES ('10', 'All-Wheel Drive', 18, 19);

INSERT INTO Car\_types VALUES ('11', 'Rear-Wheel Drive', 20, 21);

INSERT INTO Car\_types VALUES ('12', 'Front-Wheel Drive', 22, 23);

--Создаем таблицу Марка машин

DROP TABLE IF EXISTS car\_mark;

CREATE TABLE car\_mark (

Id\_mark INT,

Id\_car VARCHAR(180),

Mark VARCHAR(100),

cost INT,

Quantity INT,

PRIMARY KEY (id\_mark, id\_car),

FOREIGN KEY (Id\_car) REFERENCES car\_types(Id\_car) );

--Заполняем поля таблицы Марка машин

insert into car\_mark values (1, '4', 'Audi', 5000000, 6);

insert into car\_mark values (2, '10', 'BMW', 10000000, 3);

insert into car\_mark values (3, '7', 'Volkswagen', 4000000, 15);

insert into car\_mark values (4, '8', 'Lada', 1000000, 16);

insert into car\_mark values (5, '8', 'Renault', 1500000, 12);

insert into car\_mark values (6, '10', 'Audi', 8500000, 9);

insert into car\_mark values (7, '6', 'Jaguar', 5500000, 10);

insert into car\_mark values (8, '5', 'KIA', 2500000, 11);

insert into car\_mark values (9, '12', 'HAVAL', 3000000, 11);

insert into car\_mark values (10, '9', 'BMW', 3300000, 5);

insert into car\_mark values (11, '11', 'Geely', 2700000, 16);

insert into car\_mark values (12, '5', 'Volkswagen', 2600000, 12);

insert into car\_mark values (13, '4', 'Mercedes', 8000000, 3);

insert into car\_mark values (14, '7', 'Hyundai', 6000000, 4);

insert into car\_mark values (15, '10', 'Mercedes', 1100000, 4);

insert into car\_mark values (16, '4', 'BMW', 12000000, 2);

insert into car\_mark values (17, '10', 'Land Rover', 4500000, 6);

--Создаем таблицу Сделки

DROP TABLE IF EXISTS Deals;

CREATE TABLE Deals (

Id\_deal VARCHAR(200) PRIMARY KEY,

ID\_client VARCHAR(200) ,

ID\_employee VARCHAR(180),

Id\_mark INT,

Sale\_date Date,

Amount INT,

CHECK (Amount > 0),

FOREIGN KEY (ID\_client) REFERENCES Clients(ID\_client),

FOREIGN KEY (ID\_employee) REFERENCES Employees(ID\_employee),

FOREIGN KEY (Id\_mark) REFERENCES car\_mark(Id\_mark)

);

--Заполняем поля таблицы Сделки

insert into Deals values ('001', '02', '301', 9, '11-08-2023', 1);

insert into Deals values ('002', '010', '502', 5, '12-07-2023', 2);

insert into Deals values ('003', '012', '403', 8, '17-12-2023', 3);

insert into Deals values ('004', '03', '103', 10, '16-12-2023', 4);

insert into Deals values ('005', '01', '105', 11,'18-01-2023', 8);

insert into Deals values ('006', '05', '205', 12, '21-01-2023', 1);

insert into Deals values ('007', '012', '302', 7, '29-04-2023', 3);

insert into Deals values ('008', '04', '401', 6, '15-10-2023', 1);

insert into Deals values ('009', '06', '202', 5, '10-11-2023', 2);

insert into Deals values ('0010', '08', '403', 8, '03-03-2023', 4);

insert into Deals values ('0011', '09', '204', 9, '01-02-2023', 2);

insert into Deals values ('0012', '06', '201', 10, '18-07-2023', 1);

insert into Deals values ('0013', '07', '402', 2, '03-08-2023', 1);

insert into Deals values ('0014', '01', '202', 13,'11-06-2023', 3);

insert into Deals values ('0015', '012', '203', 3, '10-01-2023', 1);

insert into Deals values ('0016', '01', '104', 14, '04-02-2023', 2);

insert into Deals values ('0017', '04', '404', 1, '02-12-2023', 3);

insert into Deals values ('0018', '07', '501', 15,'06-09-2023', 1);

insert into Deals values ('0019', '09', '303', 17, '08-11-2023', 1);

insert into Deals values ('0020', '02', '102', 16, '09-04-2023', 2);

insert into Deals values ('0021', '05', '401', 6, '13-12-2023', 1);

**Проверяем наш код на соответствие критериям:**

В каждой таблице не менее 5 строк;



Для каждой таблицы должен быть установлен первичный ключ( Employees – ID\_employee, Deals – ID\_deal, Clients – ID\_client, Departaments – DepNo, Car\_mark – ID\_mark, car\_types – ID\_car);



Не менее чем в 2-х таблицах должен быть составной первичный ключ ( В таблице Empoyees составной первичный ключ – ID\_employee, DepNo; В таблице Car\_mark составной первичный ключ - ID\_mark, ID\_car);



В базе данных должно быть не меньше 3-х внешних ключей( DepNo, ID\_employee, ID\_client, ID\_mark, ID\_car);



Picture backgroundДля части таблиц необходимо установить ограничения целостности ( прописали Check);

Picture backgroundВ одной из таблиц должна быть древовидная структура, хранимая в виде списка (Наша древовидная структура реализована в виде вложенных множеств (сущность car\_types). Данная таблица реализована так: У нас есть три типа кузова, у каждого кузова есть номер входа(lft) и номер выхода(rgt). К примеру «sedan» lft=1, rgt=(заполним позже). Заходим в данный атрибут и после имеем три разветвления: «All-Wheel-Drive» lft=2, rgt=3; и оставшиеся два типа привода, в которые мы заходим и сразу выходим, далее получаем, что у “sedan” rgt=8, далее по аналогичной схеме заполняем оставшиеся два типа кузова);

## Раздел 4. Запросы

--Наши запросы

--Запрос 1

--В таблице Departaments у нас есть столбец Fact\_of\_sales, который нужно заполнить. То есть мы хотим отследить

--за 2023 год на какую сумму осуществлялись продажи для каждого из отдела. Для этого нужно отследить количество

--проданных автомобителей для каждого сотрудника из таблицы Deals и умножить на стоимость машины из таблицы

--car\_mark. Также не забыть о том, что каждый сотрудник мог совершить более одной сделки и в каждом отделе работает

--несколько сотрудников.

UPDATE Departaments

SET Fact\_of\_sales = (

SELECT SUM(deal.Amount \* cm.cost)

FROM Deals deal

INNER JOIN car\_mark cm ON deal.Id\_mark = cm.Id\_mark

INNER JOIN Employees e ON deal.ID\_employee = e.ID\_employee

WHERE e.DepNo = Departaments.DepNo

GROUP BY e.DepNo

);

SELECT \* from Departaments;

--Запрос 2

--Добавляем стобец summ\_count в Employee, чтобы посчитать сумму на которые заключили сделки сотрудники

ALTER TABLE Employees add COLUMN summ\_count int DEFAULT 0;

SELECT \* from Employees;

--Отобразить на какую сумму продал автомобили каждый из сотрудников.

UPDATE Employees

set summ\_count = (

SELECT SUM(d.Amount \* cm.cost) AS 'Сумма\_продаж'

FROM Deals d

INNER JOIN car\_mark cm ON d.Id\_mark = cm.Id\_mark

INNER JOIN Employees e ON d.ID\_employee = e.ID\_employee

WHERE Employees.ID\_employee = d.ID\_employee

);

--Выводим сотрудников без NULL в summ\_coint

SELECT \* from Employees

WHERE Employees.summ\_count is not NULL;

--Запрос 3

--Посчитать какой процент плана продаж выполнил каждый работник по отделам

SELECT e.id\_employee, e.name, e.depno, e.summ\_count, d.sales\_plan,

ROUND((CAST(e.summ\_count as FLOAT) / d.sales\_plan) \* 100, 2) AS percentage

FROM Departaments d

INNER JOIN Employees e ON e.DepNo = d.DepNo

WHERE e.summ\_count IS NOT NULL

ORDER BY percentage Asc;

select \* From Employees;

--Запрос 4

--Теперь нужно посчитать чистую прибыль компании на данный момент. Для этого нужно сложить

--суммы всех проданных машин по отделам в качестве доходов, посчитать суммы всех зарплат и вычесть их

-- из прибыли и вычесть также остальные издержки( на оборудование, аренду и другое). Условно будет считать

-- все остальные издержки равными 4 000 000.

select SUM(d.fact\_of\_sales) - SUM(e.Salary) - 4000000 As "Чистая прибыль компании"

from Departaments d

NATURAL join Employees e;

-- Получили 870848000

--Запрос 5

--Начинаем подбирать оптимальную бонусную систему

-- Начнем с самой простой. К зарплатам сотрудников отдела продаж будем добавлять 2% от суммы проданных машин

ALTER TABLE Employees add COLUMN new\_salary int DEFAULT 0;

UPDATE Employees

set new\_salary = Salary + (summ\_count \* 0.02 );

select \* From Employees;

-- Снова посчитаем выручку компании

select SUM(d.fact\_of\_sales) - SUM(e.new\_salary) - 4000000 As "Чистая прибыль компании"

from Departaments d

NATURAL join Employees e;

--Получили 877022000

-- Запрос 6

-- Попробуем другую бонусную систему. Будем действовать по формеле Зарплата = Fn + (Факт продаж - Rмин) \* Kn

-- Идея такая : мы поделим всех сотрудников по диапазонам продаж( Rmin и Rmax). Для каждого диапазона будет создан свой

-- коэфициент Kn и своя констаната Fn.

-- Добавляем новый столбец salary1

ALTER TABLE Employees ADD COLUMN salary1 INT DEFAULT 0;

UPDATE Employees

SET salary1 = CASE

WHEN summ\_count >= 1 AND summ\_count <= 3500000 THEN 0 + (summ\_count - 1) \* 0.1

WHEN summ\_count > 3500000 AND summ\_count <= 10000000 THEN 80000 + (summ\_count - 3500001) \* 0.08

WHEN summ\_count > 10000000 AND summ\_count <= 20000000 THEN 120000 + (summ\_count - 10000001) \* 0.05

WHEN summ\_count > 20000000 AND summ\_count <= 30000000 THEN 150000 + (summ\_count - 20000001) \* 0.03

ELSE 20000 + (summ\_count - 30000001) \* 0.01

END;

select \*FROM Employees;

-- Считаем чистую прибыль компании

select SUM(d.fact\_of\_sales) - SUM(e.salary1) - 4000000 As "Чистая прибыль компании"

from Departaments d

NATURAL join Employees e;

-- Получили 877132001.21

-- Заметим, что в последнем случае мы получили максимальную прибыль, а это значит, что бонусная система

-- напрямую влияет на прибыль компании

--Запрос 7

-- Вывести топ-5 сотрудников, которые чаще всего покупают автомобили (постоянные клиенты)

SELECT e.ID\_employee, e.Name AS Имя, COUNT(d.Id\_deal) AS "Количество купленных машин"

FROM Employees e

JOIN Deals d ON e.ID\_employee = d.ID\_employee

GROUP BY e.ID\_employee, e.Name

ORDER BY "Количество купленных машин" DESC

LIMIT 5;

-- Запрос 8

-- Вывести информацию о сотрудниках, работающих в отделах, которые перевыплнили свой план продаж

-- с сортировкой по убыванию общей суммы продаж

WITH DepartmentSales AS (

SELECT d.DepNo, d.Sales\_plan, d.Fact\_of\_sales

FROM Departaments d

WHERE d.Fact\_of\_sales > d.Sales\_plan

),

EmployeeSales AS (

SELECT e.ID\_employee, e.Name, e.Position, e.DepNo, COUNT(deal.Id\_deal) AS TotalDeals, SUM(deal.Amount) AS TotalSales

FROM Employees e

JOIN Deals deal ON e.ID\_employee = deal.ID\_employee

GROUP BY e.ID\_employee, e.Name, e.Position, e.DepNo

)

SELECT es.Name, es.Position, es.DepNo, es.TotalDeals, es.TotalSales, ds.Sales\_plan, ds.Fact\_of\_sales

FROM EmployeeSales es

JOIN DepartmentSales ds ON es.DepNo = ds.DepNo

ORDER BY es.TotalSales DESC;

-- Запрос 9

-- Определить какие марки автомобилей чаще всего покупали и какие клиенты с информацией о машинах и клиентах.

WITH BestCar AS (

SELECT cm.Mark, cm.Id\_car, COUNT(d.Id\_deal) AS "Куплено"

FROM car\_mark cm

JOIN Deals d ON cm.Id\_mark = d.Id\_mark

GROUP BY cm.Mark, cm.Id\_car

ORDER BY "Куплено" DESC

LIMIT 5

)

SELECT c.ID\_client, c.PhoneNum as "Номер телефона", c.Birthday as "Дата рождения", mbc.Mark as "Марка", mbc.Id\_car, mbc. "Куплено"

FROM Clients c

JOIN Deals d ON c.ID\_client = d.ID\_client

JOIN car\_mark cm ON d.Id\_mark = cm.Id\_mark

JOIN BestCar mbc ON cm.Mark = mbc.Mark AND cm.Id\_car = mbc.Id\_car;

--Запрос 10

-- Найти сотрудников с наибольшими продажами и вычислить среднюю их зарплату после начисления бонусов

SELECT

e.ID\_employee,

e.Name,

e.DepNo,

e.summ\_count,

e.Salary1,

AVG(e.Salary1) OVER() AS 'Средняя\_зарплата'

FROM Employees e

WHERE e.summ\_count IS NOT NULL AND e.summ\_count > 20000000

ORDER BY e.summ\_count DESC;

SELECT \* from Employees;

--Запрос 11

-- Вывести топ 6 самых продаваемых машин из Deals, их тип, цену, количество продаж и количество сделок, в которых они участвовали

SELECT

cm.Mark,

cm.cost,

ct.typecar,

COUNT(d.Id\_deal) AS Number\_of\_Deals,

SUM(d.Amount) AS Total\_Quantity\_Sold

FROM Deals d

INNER JOIN car\_mark cm ON d.Id\_mark = cm.Id\_mark

INNER JOIN Car\_types ct ON cm.Id\_car = ct.Id\_car

GROUP BY cm.Mark, cm.cost, ct.typecar

ORDER BY Total\_Quantity\_Sold DESC

LIMIT 6;

--Запрос 12

--Вывести клиентов, которые совершали больше всего, а также, кто их обслуживал, какие машины покупали и их тип

SELECT

c.ID\_client,

c.PhoneNum,

c.Birthday,

e.Name AS Employee\_Name,

cm.Mark AS Car\_Mark,

ct.typecar AS Car\_Type

FROM Clients c

inner JOIN Deals d ON c.ID\_client = d.ID\_client

inner JOIN Employees e ON d.ID\_employee = e.ID\_employee

inner JOIN car\_mark cm ON d.Id\_mark = cm.Id\_mark

inner JOIN Car\_types ct ON cm.Id\_car = ct.Id\_car

GROUP BY c.ID\_client, c.PhoneNum, c.Birthday, e.Name, cm.Mark, ct.typecar

LIMIT 10;

**Подведем итог:** В данной компании необходимо заменить премиальную систему для отдела продаж, поскольку это не только позволит увеличивать в разумных пределах зарплаты для тех сотрудников, которые смогли больше продать, но и увеличит чистую прибыль компании за счет как текущего распределения бюджета, так и дальнейшего стимула сотрудников больше продавать.