**Министерство образования и науки Российской Федерации**

Федеральное государственное автономное образовательное учреждение

высшего образования

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Инженерная школа информационных технологий и робототехники

Направление Информационные системы и технологии

Профиль Геоинформационные системы

**ОТЧЕТ**

**по индивидуальному заданию**

Проектирование БД в заданной предметной области

по дисциплине «Управление данными»

Выполнил: студент гр. 8И5Б \_\_\_\_\_\_\_\_\_\_\_\_ Смирнов П.О.

\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_ 2018 г.

(дата сдачи отчета)

Проверил: Ст. Преподаватель каф. ИСТ \_\_\_\_\_\_\_\_\_\_\_\_ Лепустин А.В.

\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_ 2018 г.

(дата проверки отчета)

|  |  |
| --- | --- |
| Параметры | Балл |
| 1. Соответствие отчета стандарту. 2. Проектирование перечня отчётов. 3. Концептуальная схемы. 4. Физическая схема БД в СУБД. 5. Наполнение БД данными. 6. Корректность и оптимальность запросов. |  |
| Всего баллов |  |

Томск 2018 г.

# Цель работы

Получить основные навыки выполнения полного цикла проектирования и создания БД.

# Задание

Автомойка

Автомойка оказывает определенный перечень услуг (с прейскурантом и временем выполнения). Клиенты должны предварительно записаться, при этом указав набор желаемых услуг. Фактически оказанные услуги могут отличаться от заказа (например, для выполнения одной услуги потребовалось предварительно оказать другую или клиент отказался от одной из услуг). Информация об оплате заказа также должна сохраняться в БД. В распоряжении мойки имеется несколько боксов, в которых машины могут обслуживаться параллельно разными сотрудниками. Все сотрудники могут выполнять все виды работ. У клиентов могут быть предпочтения как по сотрудникам, так и по номеру бокса.

Клиент так же может иметь несколько автомобилей или же менять их с течением времени. Автомобили также классифицируются по категориям (зависит от габаритов, а не цены автомобиля) описание которых находится в таблице отдельной таблице. Стоимость и длительность услуги на автомойке зависят от категории, чем она выше, тем больше цена и дольше идет работа, и не зависят от конкретного сотрудника. У авто мойщиков 10 часовой рабочий день, нормой считается 4 рабочих дня в неделю, итого 120 часов в месяц). Но авто мойщик не может работать больше 300 часов в месяц. Услуги по предварительной записи и фактический набор оказанных услуг могут отличаться, от этого зависит суммарная стоимость и длительность выполнения заказа. Оплата может производиться по факту выполнения или же с задержкой, но только одним платежом. Фактическая стоимость заказа рассчитывается путем сложения стоимости всех оказанных услуг. Сезон года хранится в отдельной таблице с названием сезона и порядковым номером месяца в году. К одной записи прикрепляется только один автомобиль. При записи клиент может указать приоритетный бокс (по номеру бокса) или же приоритетного авто мойщика (по его фамилии). Если приоритеты не указаны, то бокс и авто мойщик указывается автоматически менеджером.

# Порядок выполнения работы

# Часть 1. Ознакомление с предметной областью, и планировка перечня отчётов.

В базе данных будет хранится информация об клиентах, их автомобилях, сотрудниках, услугах автомойки и записях клиентов на услуги. Для клиентов это:

1. Фамилия
2. Имя
3. Отчество
4. Номер телефона
5. Предпочитаемый сотрудник
6. Предпочитаемый бокс
7. Номер, марка и модель автомобилей

Для сотрудников:

1. Фамилия
2. Имя
3. Отчество
4. Количество отработанных часов за каждую месяц в году

Для услуг:

1. Название
2. Стоимость
3. Длительность
4. Категория

Для записей:

1. Номер клиента
2. Номер сотрудника
3. Номер бокса
4. Номер автомобиля
5. Дата записи на услугу
6. Фактическая дата оказания услуги
7. Выбранные при записи услуги
8. Фактически оказанные услуги
9. Дата оплаты услуги

Отчёты:

1. Топ 5 самых заказываемых услуг за определенный год?
2. Вывести ФИО клиента и марку автомобиля на котором он посещал автомойку, когда за один заказ он приобрел две и более услуги стоимостью дороже 1000 рублей каждая.
3. Клиенты, которые посещали автомойку более одного раза за определенный месяц в году и на каких автомобилях, если они разные, то вывести кол-во посещений для каждого автомобиля.
4. Какая услуга является часто оказываемой для каждой марки автомобилей для каждого сезона года.
5. Вывести последние 5 записей рабочего, у которого больше всего записей, которые были оплачены с запозданием (за полчаса и более после факта записи) с указанием их даты и разницы во времени (когда произошел факт оплаты и факт записи)
6. Вывести выручку в определенном месяце определенного года и разницу с выручкой прошлого года
7. Вывести для каждого рабочего категорию автомобиля для которой он оказал наибольшее количество услуг с указанием количества за предыдущий месяц.
8. Вывести ФИО рабочих, которые перерабатывали (работали больше определенного кол-ва часов) два месяца подряд
9. Вывести топ 3 рабочих при работе которых клиент чаще всего заказывает дополнительные услуги и их количество
10. Таблица статистики количества оказанных услуг авто мойщиком для каждой категории автомобилей.

# Часть 2. Проектирование схем БД

После ознакомления с предметной областью разработана концептуальная схема в Toad Data Modeler. Схема показана на рисунке 1.

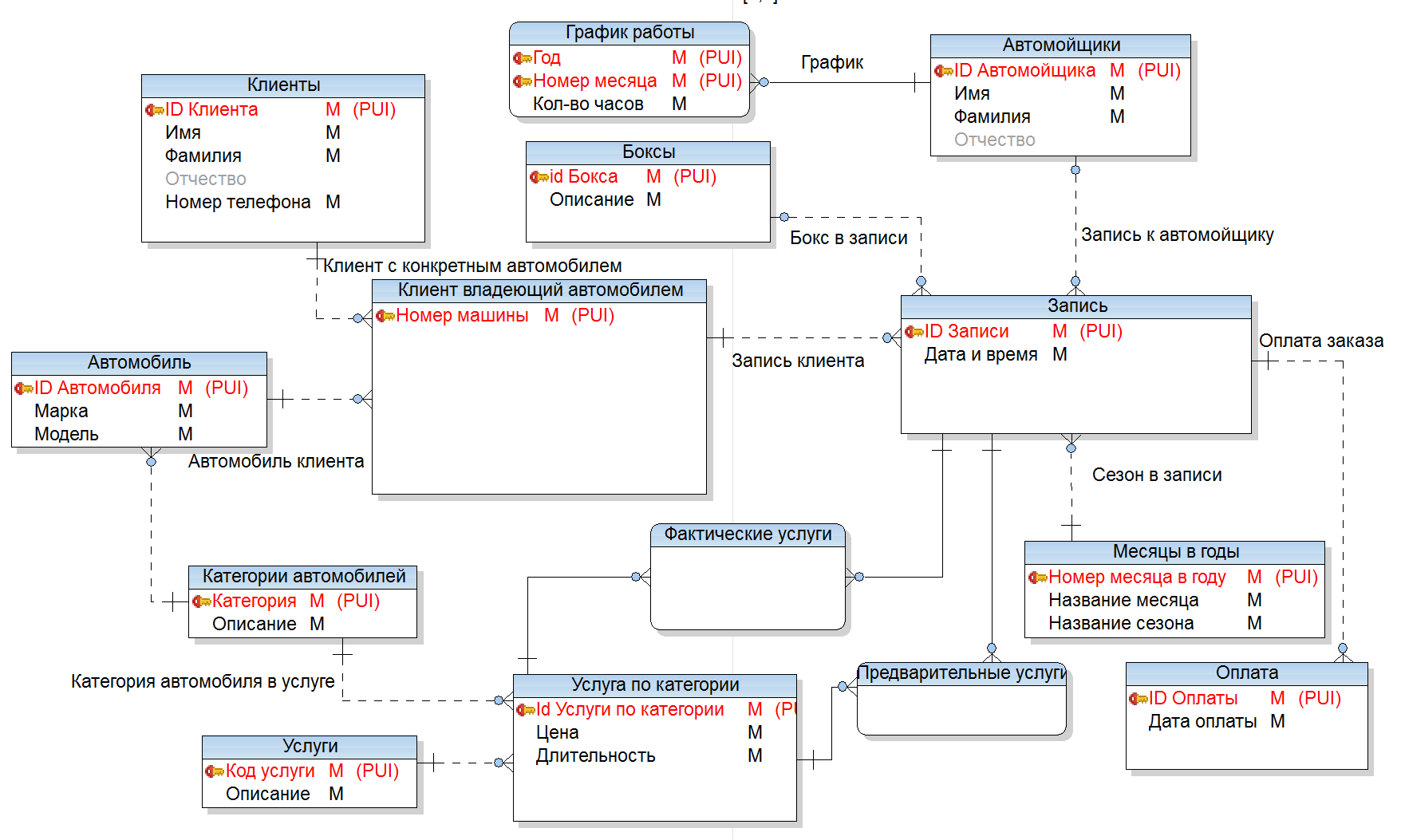


Рисунок 1 – Концептуальная схема БД

Затем схема конвертирована в логическую (Рисунок 2).

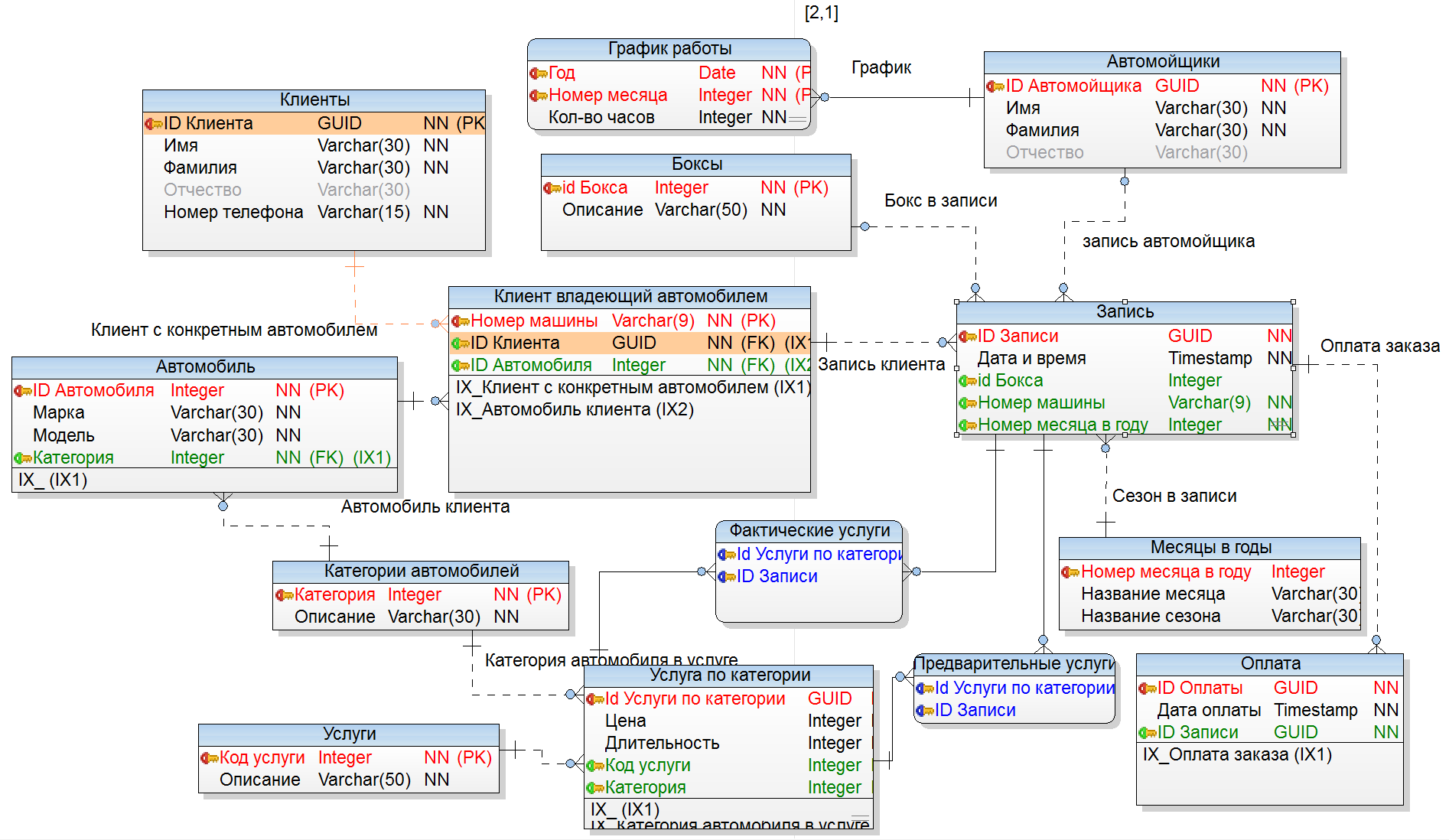


Рисунок 2 – Логическая схема БД

После схема конвертирована в физическую (для MS SQL Server 2016), сгенерирован и выполнен сценарий создания БД в целевой СУБД.

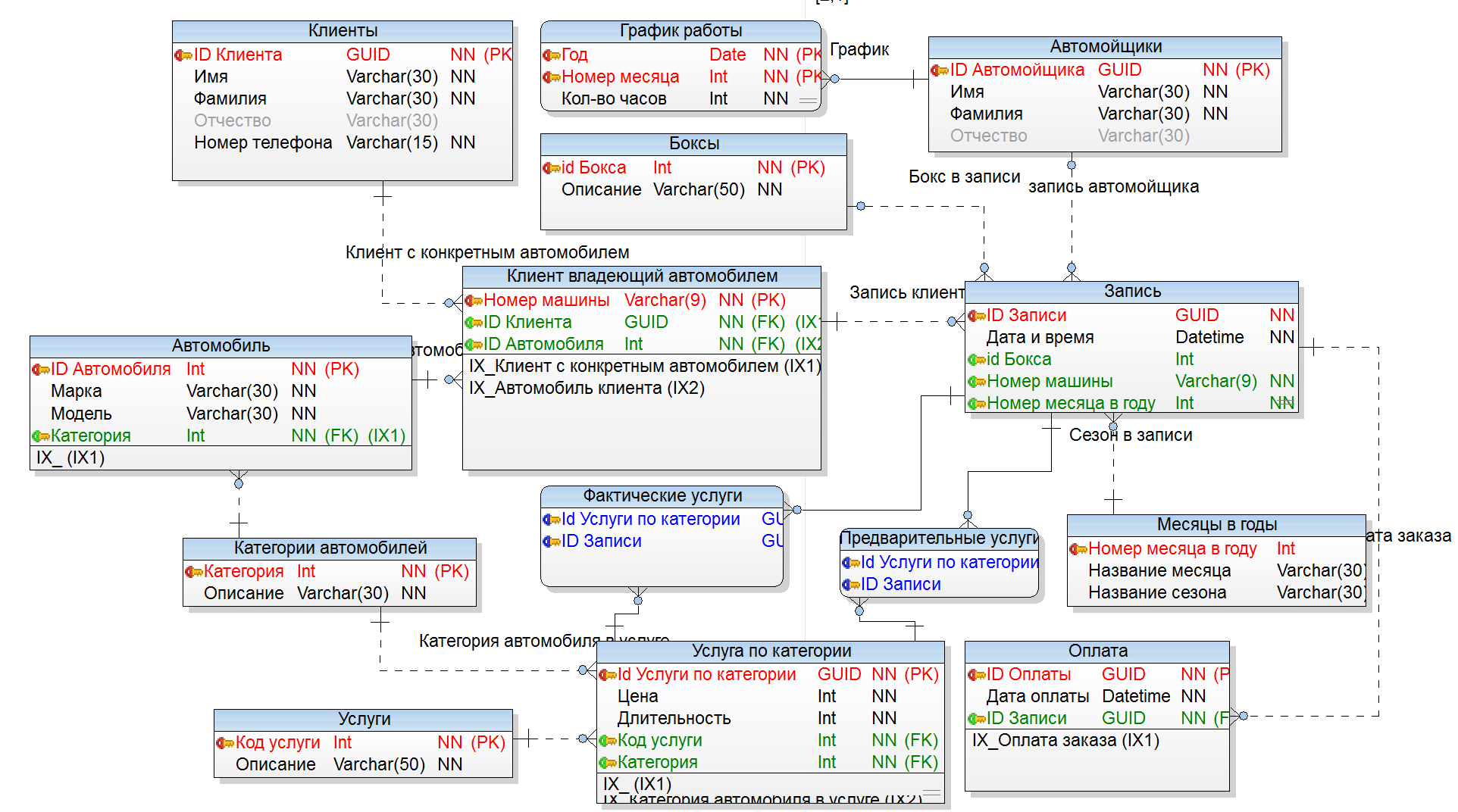


Рисунок 3 – Физическая схема БД



Рисунок 4 – Диаграмма БД

Также на данном этапе добавим ограничения на столбцы, например, зададим ограничение на диапазоны значений в таблице WorkSchedule, которые могут храниться в столбце. А для таблицы CarClient установим ограничение на номер автомобиля, чтобы пользователь мог ввести только значения корректного формата.

ALTER TABLE WorkSchedule

ADD CONSTRAINT NumberOfHours CHECK (NumberOfHours <= 300),

CONSTRAINT Years CHECK (WorkSchedule.Years >= 2016),

CONSTRAINT Months CHECK (WorkSchedule.Months >= 1 AND WorkSchedule.Months <= 12);

ALTER TABLE CarClientSet

ADD CONSTRAINT CarNumber CHECK (CarNumber LIKE '[0-9][A-Z][A-Z][A-Z][0-9][0-9][0-9][0-9]'

OR dist\_id LIKE '[0-9][A-Z][A-Z][A-Z][0-9][0-9][0-9][0-9][0-9]');

# Часть 3. Наполнение БД данными

После создания БД наполнена осмысленными данными. Примеры заполнения таблиц показаны на рисунках 5-7.

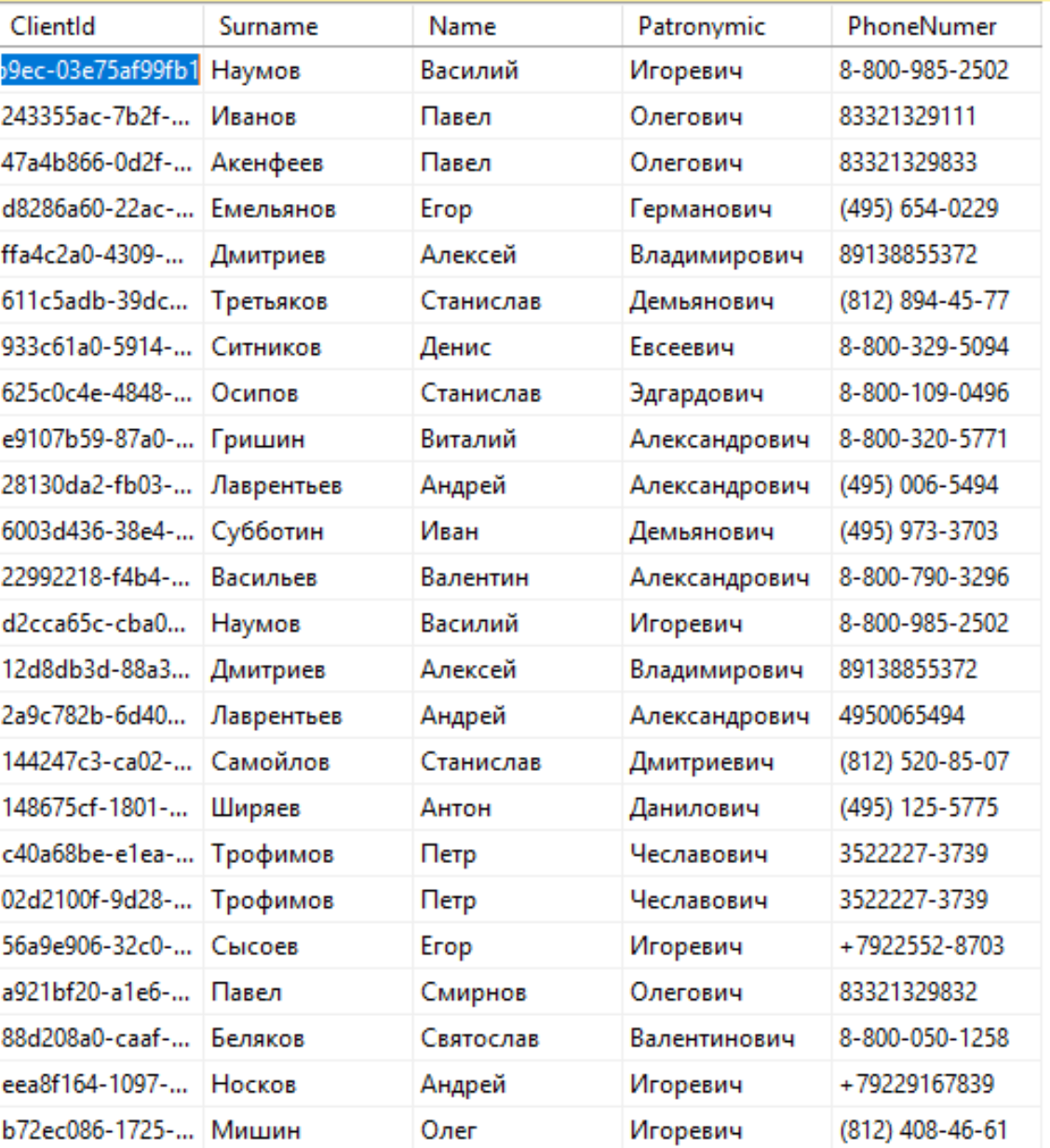


Рисунок 5 – Фрагмент заполнения таблицы «Clients»



Рисунок 6 – Фрагмент заполнения таблицы «Records»

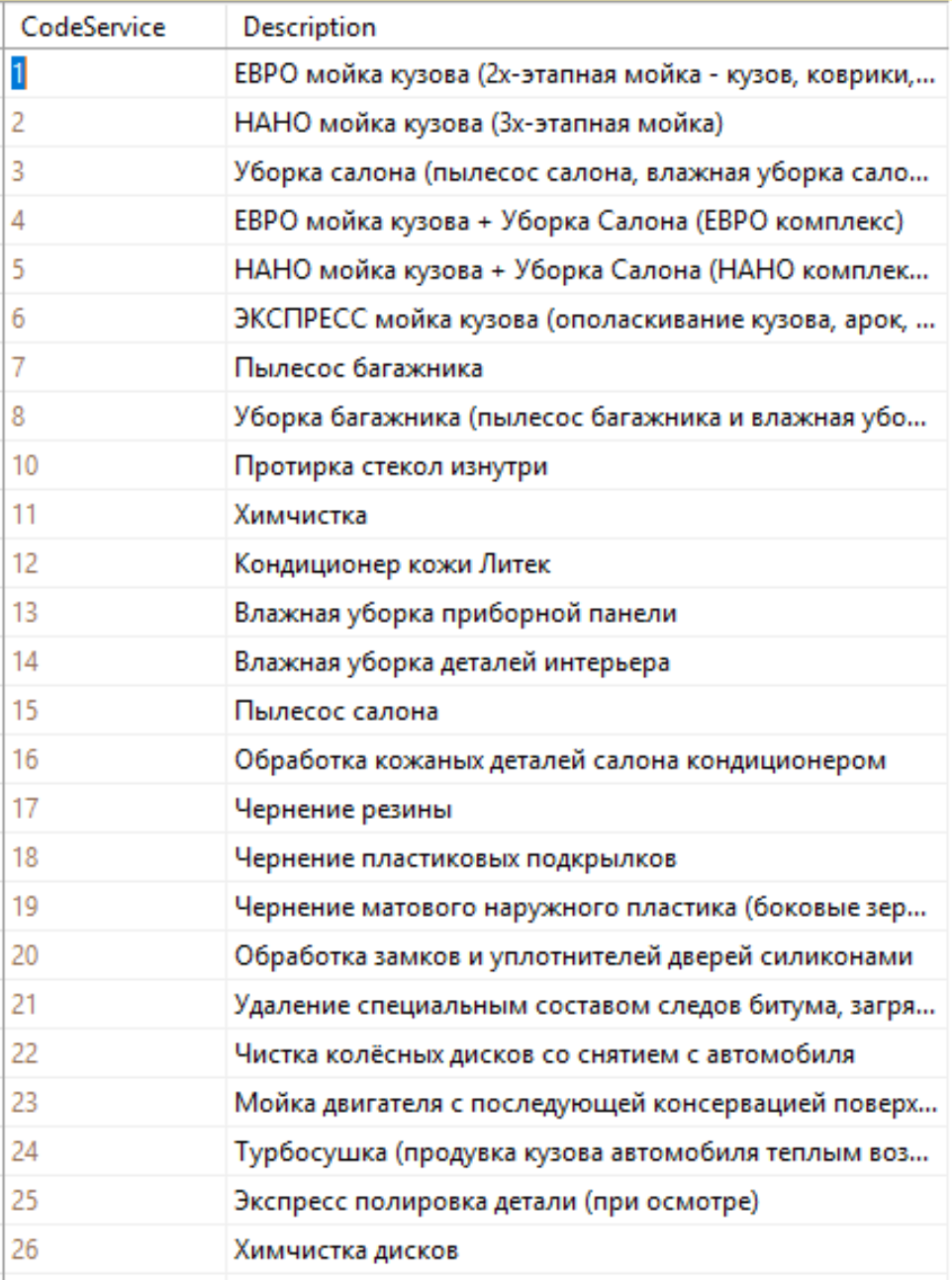


Рисунок 7 – Фрагмент заполнения таблицы «Services»

# Часть 4. Создание ХП и представлений. Настройка прав доступа

Создано представление для отображения категории автомобилей для каждого сезона года и кол-во посещений в них.

CREATE VIEW CategoriesBySeasons AS

SELECT CategoryOfCarSet.Description AS 'Категория автомобиля',

COUNT(OrderServicesByCategory.Order\_OrderId) AS 'Кол-во посещений',

Seasons.Season AS 'Сезон'

FROM ServicesByCategorySet

INNER JOIN CategoryOfCarSet ON CategoryOfCarSet.CategoryId = ServicesByCategorySet.CategoryOfCarCategoryId

INNER JOIN OrderServicesByCategory ON OrderServicesByCategory.ServicesByCategory\_ServiceId = ServicesByCategorySet.ServiceId

INNER JOIN OrderSet ON OrderSet.OrderId = OrderServicesByCategory.Order\_OrderId

INNER JOIN RecordSet ON RecordSet.RecordId = OrderSet.RecordRecordId

INNER JOIN Seasons ON Seasons.NumberOfMount = RecordSet.NumberOfSeason

GROUP BY CategoryOfCarSet.Description, Seasons.Season

Выборка из представления показана на рисунке 8.

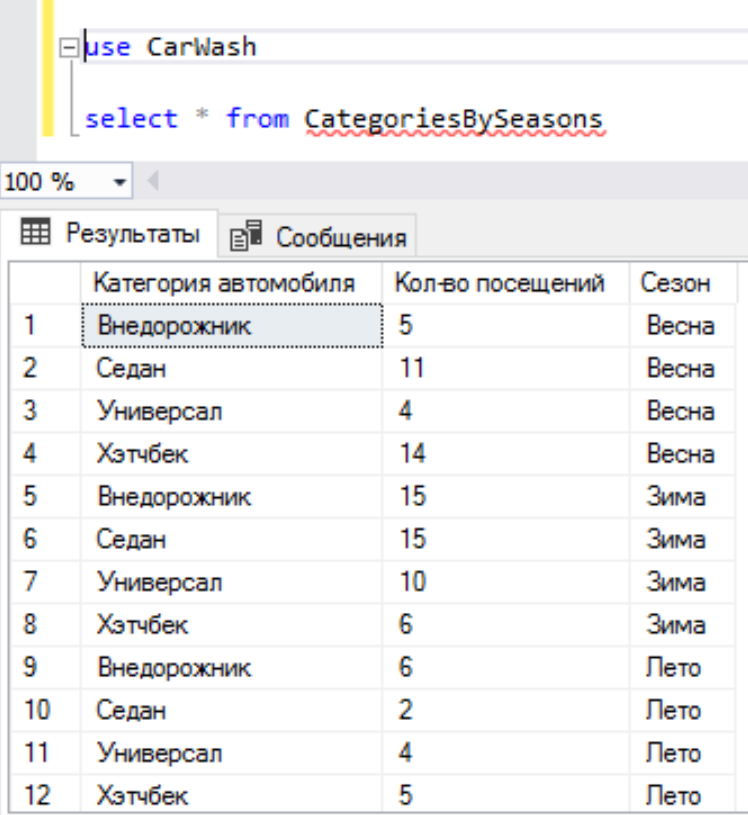


Рисунок 8 – Результат выборки из представления

Процедура для отображения клиента(ов), который(ые) пользовались определенной услугой в определенный месяц:

CREATE PROCEDURE ClientsInMonths

(@service VARCHAR(MAX), @month INT ) AS

SELECT ClientSet.Surname AS 'Фамилия рабочего'

FROM RecordSet

INNER JOIN CarClientSet ON CarClientSet.CarNumber = RecordSet.CarClientCarNumber

INNER JOIN ClientSet ON CarClientSet.ClientClientId = ClientSet.ClientId

INNER JOIN OrderSet ON OrderSet.RecordRecordId = RecordSet.RecordId

INNER JOIN OrderServicesByCategory ON OrderServicesByCategory.Order\_OrderId = OrderSet.OrderId

INNER JOIN ServicesByCategorySet ON ServicesByCategorySet.ServiceId = OrderServicesByCategory.ServicesByCategory\_ServiceId

INNER JOIN ServicesSet ON ServicesSet.CodeService = ServicesByCategorySet.ServicesCodeService

WHERE ServicesSet.Description = @service AND MONTH(RecordSet.DateTime) = @month

Результат выполнения хранимой процедуры для услуги «Антидождь» в июне показан на рисунке 9.

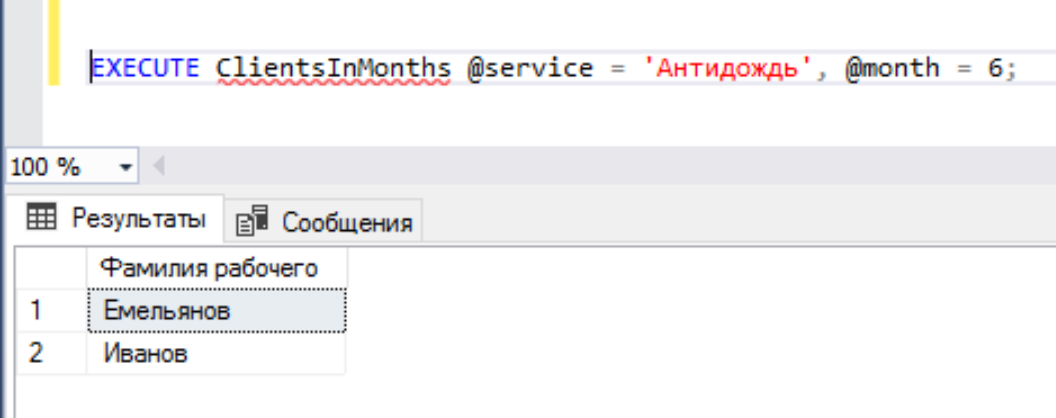


Рисунок 9 – Результат выполнения хранимой процедуры

В базе данных созданы три пользователя: «Администратор», «Менеджер» и «Клиент».

CREATE LOGIN OurAdmin

WITH PASSWORD = 'admin123',

DEFAULT\_DATABASE = CarWash

CREATE LOGIN OurManager

WITH PASSWORD = 'manager123',

DEFAULT\_DATABASE = CarWash

CREATE LOGIN OurClient

WITH PASSWORD = 'client123',

DEFAULT\_DATABASE = CarWash

CREATE USER Администратор FOR LOGIN OurAdmin;

CREATE USER Менеджер for LOGIN OurManager;

CREATE USER Клиент for LOGIN OurClient;

Для пользователя «Администратор» предоставлено право делать все действия для всех для таблиц и представлений, а также разрешено запускать хранимые процедуры.

GRANT SELECT TO Администратор

GRANT INSERT TO Администратор

GRANT UPDATE TO Администратор

GRANT EXEC TO Администратор

Для пользователя «Менеджер» предоставлено право делать SELECT, INSERT, UPDATE для всех таблиц и представлений, а также запуск хранимой процедуры «ClientsInMonths». Но для таблиц ServicesByCategorySet и WorkSchedule запрещено делать INSERT и UPDATE для того, чтобы менеджер не мог влиять на цены услуг и количество отработанных часов работником.

GRANT SELECT TO Менеджер

GRANT INSERT TO Менеджер

GRANT UPDATE TO Менеджер

GRANT EXEC ON ClientsInMonths TO Менеджер

DENY INSERT ON ServicesByCategorySet TO Менеджер

DENY INSERT ON WorkSchedule TO Менеджер

DENY UPDATE ON ServicesByCategorySet TO Менеджер

DENY UPDATE ON WorkSchedule TO Менеджер

Для пользователя «Клиент» предоставлено право делать SELECT только из таблиц с услугами.

GRANT SELECT ON ServicesByCategorySet TO Клиент

GRANT SELECT ON CategoryOfCarSet TO Клиент

GRANT SELECT ON ServicesSet TO Клиент

# Часть 5. Написание запросов для формирования отчётов

* + - 1. Топ 5 самых заказываемых услуг за определенный год?

CREATE PROCEDURE MostOrderedServices

(@year VARCHAR(4)) AS

SELECT TOP 5 ServicesSet.Description as 'Услуга',

COUNT(ServicesByCategorySet.ServicesCodeService) AS 'Кол-во заказов в данном году'

FROM ServicesByCategorySet

INNER JOIN ServicesSet ON ServicesSet.CodeService = ServicesByCategorySet.ServicesCodeService

INNER JOIN OrderServicesByCategory ON OrderServicesByCategory.ServicesByCategory\_ServiceId = ServicesByCategorySet.ServiceId

INNER JOIN OrderSet ON OrderSet.OrderId = OrderServicesByCategory.Order\_OrderId

INNER JOIN PaymentSet ON PaymentSet.OrderOrderId = OrderSet.OrderId

where CONVERT(datetime, PaymentSet.DataTime , 113)

BETWEEN CONVERT(datetime,'01/01/'+@year,104) AND CONVERT(datetime,'31/12/'+@year,104)

GROUP BY ServicesSet.Description

ORDER BY COUNT(ServicesByCategorySet.ServicesCodeService) DESC

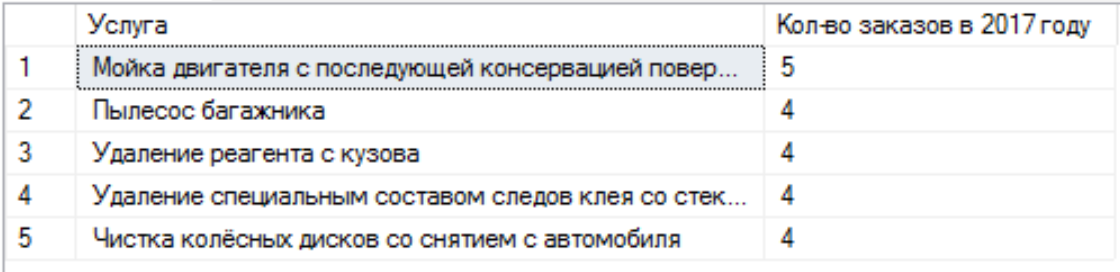


Рисунок 10 – Результат выборки

2. Какая услуга является часто оказываемой для каждой марки автомобилей для каждого сезона года.

SELECT ServicesSet.Description AS 'Название услуги', Seasons.Season AS 'Сезон',

COUNT(OrderServicesByCategory.ServicesByCategory\_ServiceId) AS 'Кол-во оказанных раз'

FROM ServicesByCategorySet

INNER JOIN ServicesSet ON ServicesSet.CodeService = ServicesByCategorySet.ServicesCodeService

INNER JOIN OrderServicesByCategory ON OrderServicesByCategory.ServicesByCategory\_ServiceId = ServicesByCategorySet.ServiceId

INNER JOIN OrderSet ON OrderSet.OrderId = OrderServicesByCategory.Order\_OrderId

INNER JOIN RecordSet ON RecordSet.RecordId = OrderSet.RecordRecordId

INNER JOIN Seasons ON Seasons.NumberOfMount = RecordSet.NumberOfSeason

GROUP BY ServicesSet.Description, Seasons.Season

ORDER BY ServicesSet.Description

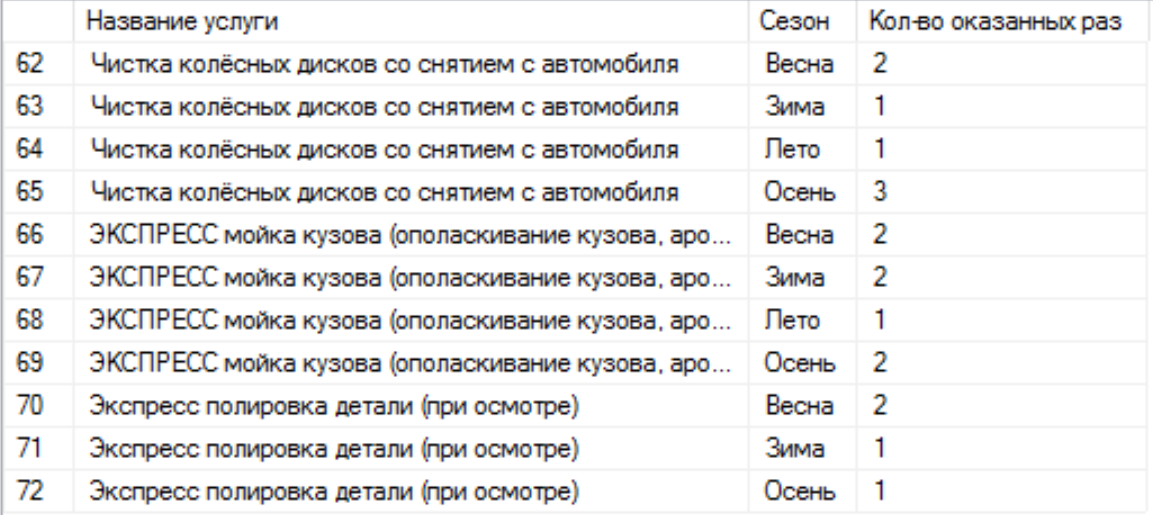


Рисунок 11 – Результат выборки

3. Клиенты, которые посещали автомойку более одного раза за определенный месяц в году и на каких автомобилях, если они разные, то вывести кол-во посещений для каждого автомобиля.

SELECT CONCAT(ClientSet.Surname,' ', ClientSet.Name,' ', ClientSet.Patronymic) AS 'Ф.И.О.',

ROW\_NUMBER() OVER(PARTITION BY CONCAT(ClientSet.Surname,' ', ClientSet.Name,' ', ClientSet.Patronymic)

ORDER BY CarSet.Mark, CarSet.Model) AS '№ п/п',

CarSet.Mark AS 'Марка', CarSet.Model AS 'Модель', COUNT(RecordSet.RecordId) AS 'Кол-во посещений'

FROM RecordSet

INNER JOIN CarClientSet ON CarClientSet.CarNumber = RecordSet.CarClientCarNumber

INNER JOIN ClientSet ON ClientSet.ClientId = CarClientSet.ClientClientId

INNER JOIN CarSet ON CarSet.CarId = CarClientSet.CarCarId

GROUP BY CONCAT(ClientSet.Surname,' ', ClientSet.Name,' ', ClientSet.Patronymic), CarSet.Mark, CarSet.Model

ORDER BY CONCAT(ClientSet.Surname,' ', ClientSet.Name,' ', ClientSet.Patronymic)



Рисунок 12 – Результат выборки

4. Вывести ФИО клиента и марку автомобиля на котором он посещал автомойку, когда за один заказ он приобрел две и более услуги стоимостью дороже 1000 рублей каждая, с датой данного посещения.

SELECT CONCAT(ClientSet.Surname,' ', ClientSet.Name,' ', ClientSet.Patronymic) AS FIO, RecordSet.DateTime AS 'Дата посещения',

CarSet.Mark AS 'Марка автомобиля', SUM(ServicesByCategorySet.Price) AS 'Общая стоимость заказа'

FROM RecordSet

INNER JOIN OrderSet ON OrderSet.RecordRecordId = RecordSet.RecordId

INNER JOIN OrderServicesByCategory ON OrderSet.OrderId = OrderServicesByCategory.Order\_OrderId

INNER JOIN ServicesByCategorySet ON ServicesByCategorySet.ServiceId = OrderServicesByCategory.ServicesByCategory\_ServiceId

INNER JOIN CarClientSet ON CarClientSet.CarNumber = RecordSet.CarClientCarNumber

INNER JOIN ClientSet ON ClientSet.ClientId = CarClientSet.ClientClientId

INNER JOIN CarSet ON CarSet.CarId = CarClientSet.CarCarId

INNER JOIN ServicesSet ON ServicesSet.CodeService = ServicesByCategorySet.ServicesCodeService

WHERE ServicesByCategorySet.Price > 1000

GROUP BY CONCAT(ClientSet.Surname,' ', ClientSet.Name,' ', ClientSet.Patronymic),

RecordSet.DateTime, CarSet.Mark, RecordSet.RecordId

HAVING COUNT(OrderServicesByCategory.ServicesByCategory\_ServiceId) >= 2

ORDER BY CONCAT(ClientSet.Surname,' ', ClientSet.Name,' ', ClientSet.Patronymic)

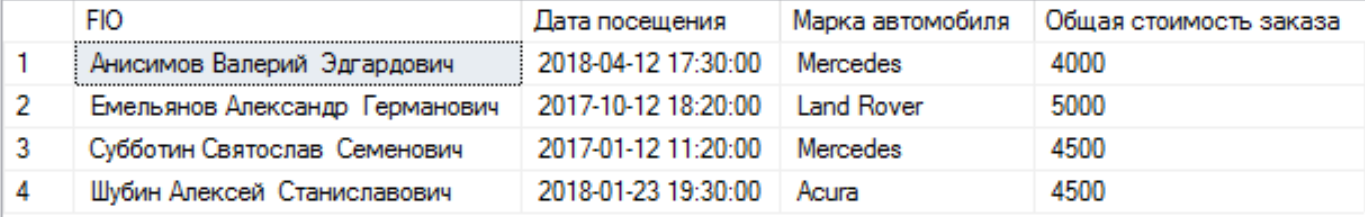


Рисунок 13 – Результат выборки

1. Вывести последние 5 записей рабочего, у которого больше всего записей, которые были оплачены с запозданием (за полчаса и более после факта записи) с указанием их даты и разницы во времени (когда произошел факт оплаты и факт записи)

SELECT TOP 5 RecordSet.CarClientCarNumber AS 'Запись', PaymentSet.DataTime AS 'Дата оплаты',

DATEDIFF(MINUTE,RecordSet.DateTime, PaymentSet.DataTime) AS 'Разница во времени'

FROM RecordSet

INNER JOIN OrderSet ON OrderSet.RecordRecordId = RecordSet.RecordId

INNER JOIN PaymentSet ON PaymentSet.OrderOrderId = OrderSet.OrderId

WHERE RecordSet.CarWasherCarWasherId =

(SELECT TOP 1 CarWasherSet.CarWasherId AS 'Автомойщик'

FROM RecordSet

INNER JOIN OrderSet ON OrderSet.RecordRecordId = RecordSet.RecordId

INNER JOIN PaymentSet ON PaymentSet.OrderOrderId = OrderSet.OrderId

INNER JOIN CarWasherSet ON CarWasherSet.CarWasherId = RecordSet.CarWasherCarWasherId

WHERE DATEDIFF(MINUTE,RecordSet.DateTime, PaymentSet.DataTime) >= 30

GROUP BY CarWasherSet.CarWasherId

ORDER BY COUNT(DATEDIFF(MINUTE,PaymentSet.DataTime, RecordSet.DateTime)) DESC)

ORDER BY PaymentSet.DataTime DESC

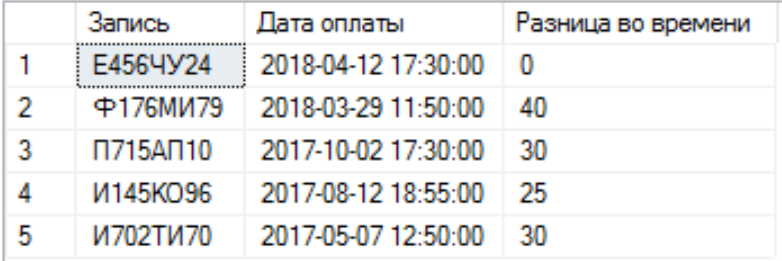


Рисунок 14 – Результат выборки

1. Вывести выручку в определенном месяце определенного года и разницу с выручкой прошлого года

CREATE PROCEDURE DeferenceBetweenTwoYears

(@\_year INT, @\_month INT) AS

DECLARE @PASTYEAR INT

SET @PASTYEAR = @\_year - 1;

WITH PastYear(outputsum, \_month, \_year) AS

(SELECT SUM(ServicesByCategorySet.Price), MONTH(RecordSet.DateTime), YEAR(RecordSet.DateTime)

FROM RecordSet

INNER JOIN OrderSet ON OrderSet.RecordRecordId = RecordSet.RecordId

INNER JOIN OrderServicesByCategory ON OrderServicesByCategory.Order\_OrderId = OrderSet.OrderId

INNER JOIN ServicesByCategorySet ON ServicesByCategorySet.ServiceId = OrderServicesByCategory.ServicesByCategory\_ServiceId

INNER JOIN Seasons ON RecordSet.NumberOfSeason = Seasons.NumberOfMount

WHERE MONTH(RecordSet.DateTime) = @\_month AND YEAR(RecordSet.DateTime) = @PASTYEAR

GROUP BY MONTH(RecordSet.DateTime), YEAR(RecordSet.DateTime))

SELECT @\_month AS 'Месяц', @\_year AS 'Год', SUM(ServicesByCategorySet.Price) AS 'Выручка в текущем году',

SUM(ServicesByCategorySet.Price) - PastYear.outputsum AS 'Разница с прошлым годом'

FROM RecordSet

INNER JOIN OrderSet ON OrderSet.RecordRecordId = RecordSet.RecordId

INNER JOIN OrderServicesByCategory ON OrderServicesByCategory.Order\_OrderId = OrderSet.OrderId

INNER JOIN ServicesByCategorySet ON ServicesByCategorySet.ServiceId = OrderServicesByCategory.ServicesByCategory\_ServiceId

INNER JOIN Seasons ON RecordSet.NumberOfSeason = Seasons.NumberOfMount

inner join PastYear on RecordSet.NumberOfSeason = PastYear.\_month

WHERE MONTH(RecordSet.DateTime) = @\_month AND YEAR(RecordSet.DateTime) = @\_year

GROUP BY PastYear.outputsum

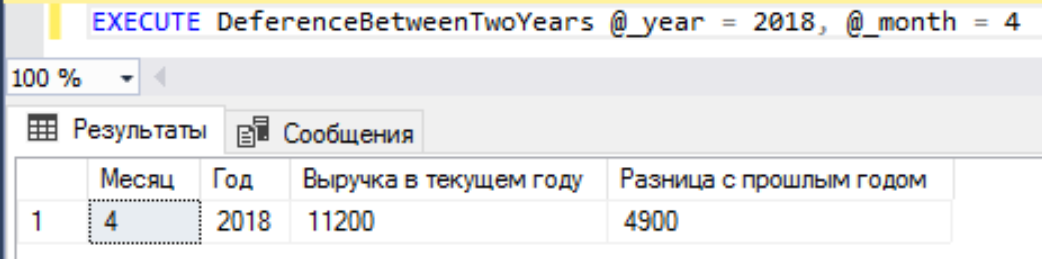


Рисунок 15 – Результат выборки

1. Вывести для каждого рабочего категорию автомобиля для которой он оказал наибольшее количество услуг с указанием количества за предыдущий месяц.

SELECT TOP(1) WITH TIES CarWasherSet.Surname AS 'Фамилия рабочего',

CategoryOfCarSet.Description AS 'Категория автомобиля',

COUNT(OrderServicesByCategory.ServicesByCategory\_ServiceId) AS 'Кол-во заказов'

FROM RecordSet

INNER JOIN CarWasherSet ON CarWasherSet.CarWasherId = RecordSet.CarWasherCarWasherId

INNER JOIN OrderSet ON OrderSet.RecordRecordId = RecordSet.RecordId

INNER JOIN OrderServicesByCategory ON OrderServicesByCategory.Order\_OrderId = OrderSet.OrderId

INNER JOIN ServicesByCategorySet ON ServicesByCategorySet.ServiceId = OrderServicesByCategory.ServicesByCategory\_ServiceId

INNER JOIN CategoryOfCarSet ON CategoryOfCarSet.CategoryId = ServicesByCategorySet.CategoryOfCarCategoryId

WHERE (RecordSet.DateTime) > dateadd(MONTH, -1, GETDATE())

GROUP BY CarWasherSet.Surname, CategoryOfCarSet.Description

order by ROW\_NUMBER() over(partition by CarWasherSet.Surname

order by count(ServicesByCategorySet.ServicesCodeService) desc)

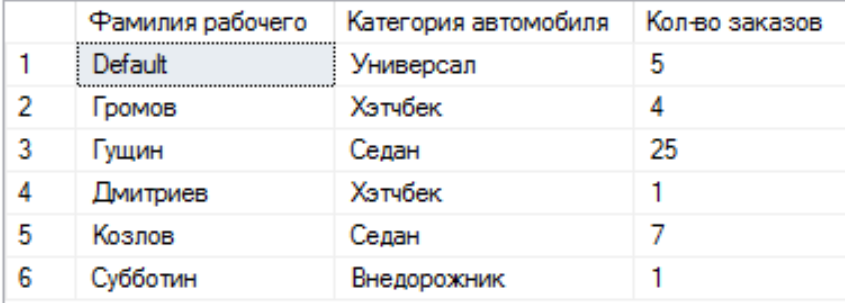


Рисунок 16 – Результат выборки

1. Вывести ФИО рабочих, которые перерабатывали (работали больше определенного кол-ва часов) два месяца подряд

SELECT (CarWasherSet.Surname +' ' +CarWasherSet.Name + '' + CarWasherSet.Patronymic) AS 'Автомойщик'

FROM RecordSet

INNER JOIN CarWasherSet ON RecordSet.CarWasherCarWasherId = CarWasherSet.CarWasherId

INNER JOIN WorkSchedule as work1 ON work1.CarWasherId = CarWasherSet.CarWasherId

INNER JOIN WorkSchedule AS work2 ON work2.CarWasherId = CarWasherSet.CarWasherId AND

((work1.Month + 1 = work2.Month AND work1.Year = work2.Year)

OR ((work1.Month = 12 AND work2.Month = 1)) AND (work1.Year + 1 = work2.Year))

WHERE work1.NumberOfHours > 120 AND (work2.NumberOfHours > 120)

GROUP BY CarWasherSet.Surname, CarWasherSet.Name, CarWasherSet.Patronymic

ORDER BY CarWasherSet.Surname

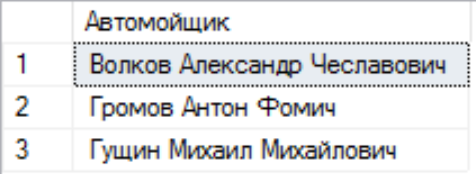


Рисунок 17 – Результат выборки

1. Вывести топ 3 рабочих при работе которых клиент чаще всего заказывает дополнительные услуги и их количество

SELECT TOP 5 CarWasherSet.Surname, count(distinct OrderServicesByCategory.ServicesByCategory\_ServiceId) -

count(distinct RecordServicesByCategory.ServicesByCategory\_ServiceId) AS 'Кол-во дополнительных услуг'

FROM RecordSet

INNER JOIN CarWasherSet ON CarWasherSet.CarWasherId = RecordSet.CarWasherCarWasherId

INNER JOIN OrderSet ON OrderSet.RecordRecordId = RecordSet.RecordId

INNER JOIN OrderServicesByCategory ON OrderServicesByCategory.Order\_OrderId = OrderSet.OrderId

LEFT JOIN RecordServicesByCategory ON OrderServicesByCategory.Order\_OrderId = RecordServicesByCategory.Order\_OrderId and

(OrderServicesByCategory.ServicesByCategory\_ServiceId = RecordServicesByCategory.ServicesByCategory\_ServiceId)

GROUP BY CarWasherSet.Surname

ORDER BY count(distinct OrderServicesByCategory.ServicesByCategory\_ServiceId) -

count(distinct RecordServicesByCategory.ServicesByCategory\_ServiceId) DESC

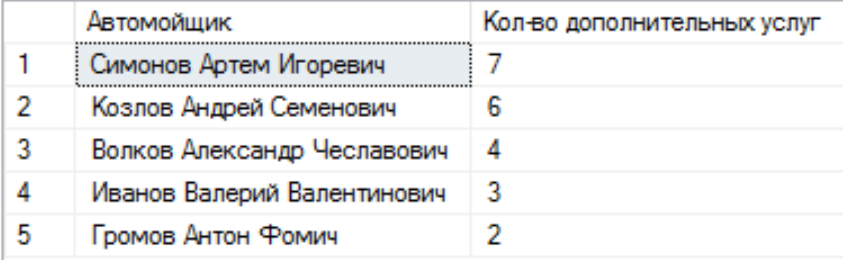


Рисунок 18 – Результат выборки

1. Таблица статистики количества оказанных услуг авто мойщиком для каждой категории автомобилей.

DECLARE @columns VARCHAR(MAX), @sql VARCHAR(MAX);

SET @columns = '';

SELECT @columns += ', ' + QUOTENAME(col.[Название категории])

FROM

(SELECT CategoryOfCarSet.Description AS 'Название категории'

FROM ServicesByCategorySet

INNER JOIN CategoryOfCarSet ON CategoryOfCarSet.CategoryId = ServicesByCategorySet.CategoryOfCarCategoryId

INNER JOIN OrderServicesByCategory ON OrderServicesByCategory.ServicesByCategory\_ServiceId = ServicesByCategorySet.ServiceId

INNER JOIN OrderSet ON OrderSet.OrderId = OrderServicesByCategory.Order\_OrderId

INNER JOIN RecordSet ON RecordSet.RecordId = OrderSet.RecordRecordId

GROUP BY CategoryOfCarSet.Description) AS col

SET @sql= 'SELECT \*

FROM

(SELECT CarWasherSet.Surname AS Автомойщик, COUNT(OrderServicesByCategory.ServicesByCategory\_ServiceId) AS SERVICES, CategoryOfCarSet.Description AS CARS

FROM RecordSet

INNER JOIN CarWasherSet ON CarWasherSet.CarWasherId = RecordSet.CarWasherCarWasherId

INNER JOIN OrderSet ON RecordSet.RecordId = OrderSet.RecordRecordId

INNER JOIN OrderServicesByCategory ON OrderServicesByCategory.Order\_OrderId = OrderSet.OrderId

INNER JOIN ServicesByCategorySet ON ServicesByCategorySet.ServiceId = OrderServicesByCategory.ServicesByCategory\_ServiceId

INNER JOIN CategoryOfCarSet ON CategoryOfCarSet.CategoryId = ServicesByCategorySet.CategoryOfCarCategoryId

GROUP BY CarWasherSet.Surname, CategoryOfCarSet.Description

) AS temp

PIVOT(MAX(temp.SERVICES) FOR temp.CARS IN (' +STUFF(@columns, 1, 2, '') +')) RESULT;';

EXECUTE(@sql)

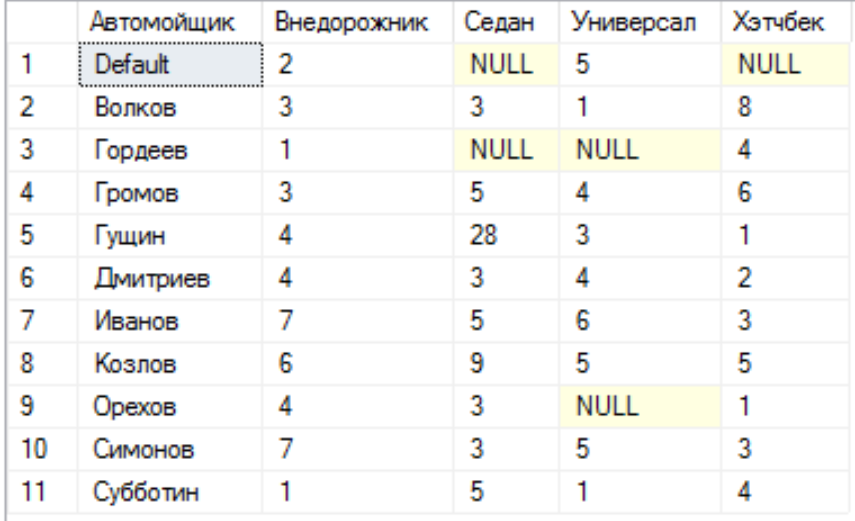


Рисунок 19 – Результат выборки

# Вывод

В ходе данной работы были получены основные навыки выполнения полного цикла проектирования и создания БД. Непосредственно создана БД по рассматриваемой теме, заполнена необходимыми данными. Также были написаны необходимые запросы для данной БД.