**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Саратовский государственный технический университет**

**имени Гагарина Ю.А.»**

Институт прикладных информационных технологий и коммуникаций

Направление «Информационные системы и технологии»

Кафедра «Прикладные информационные технологии»

**КОНТРОЛЬНАЯ РАБОТА**

по дисциплине «Управление данными»

Выполнил студент группы

б1-ИФСТипу-21

заочной формы обучения

Рамазанов Павел Андреевич

Номер зачетной книжки: 202583

Проверил: доцент кафедры ПИТ

Иванов А.В.

Саратов, 2022

**Задание № 1**

**Назначение базы данных. Анализ предметной области. Техническое задание на разработку БД**

**Учебный пример разработки базы данных**

Рассмотрим основные моменты проектирования базы данных "Сервисный центр по ремонту компьютерной техники" (вариант № 3).

**Назначение базы данных и анализ предметной области**

База данных предназначена для автоматизации основных информационных процессов в сервисном центре. К ним относятся: регистрация сломанной компьютерной техники, которую приносят в ремонт, фиксирование неисправности компьютерной техники, контактные данные клиента.

В настоящее время в сервисном центре используется бумажный учёт. Новая система позволит снизить трудоёмкость операций, увеличить их скорость и удобство.

С помощью сбора первичных документов и интервью с сотрудниками сервисного центра получены следующие сведения:

Для ведения учета техник, находящейся в ремонте, организации поиска требуемых решений и статистики в базе должны храниться сведения, часть которых размещается в стеллажных карточках (рис. 1).

|  |  |
| --- | --- |
|  | **Стеллажная карта**  Данные о ремонтируемом устройстве: \_\_\_\_\_\_\_  Тип неисправности: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Ответственный за диагностику: \_\_\_\_\_\_\_\_\_\_\_\_\_  № п/п заявки на ремонт: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Контактные данные клиента  Телефон: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  E-mail: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Возможность доставки клиенту после ремонта оборудования: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

Рисунок 1. Образец аннотированной каталожной карточки

Анализ поступающих заявок на ремонт компьютерной техники показывает, что для автоматизации процесса учета входящих заявок следует выделить следующие атрибуты обращений клиентов:

* Id.
* Дата обращения.
* Id устройства.
* Id неисправности устройства.
* Id ответственного за диагностику.
* Порядковый номер заявки на ремонт.
* Id контактных данных клиента.
* Возможность доставить отремонтированную технику клиенту (да/нет).
* Дата выполнения.
* Стоимость проведенных работ.

К атрибутам, которые характеризуют информацию о сломанном устройстве, относятся:

* Id.
* Название устройства.
* Тип устройства (ПК, ноутбук, принтер, сканер, МФУ, монитор).
* Изготовитель.
* Модель.
* Серийный номер.
* Год выпуска.
* Id детали, подлежащей замене.

К атрибутам неисправности устройства относятся:

* Id.
* Уровень серьезности поломки (низкий/средний/высокий).
* Возможность ремонта по гарантии (да/нет).
* Id устройства.

К атрибутам ответственного за диагностику устройства работника сервисного центра можно отнести:

* Id.
* Ф.И.О. работника.
* Должность.
* Уровень компетенций в ремонте оборудования (стажер/профессиональный техник).

Атрибуты контактных данных пользователя:

* Id.
* E-mail.
* Номер телефона.
* Адрес доставки.

Атрибуты поставщика запчастей для ремонта:

* Id.
* Название фирмы-поставщика.
* Адрес поставщика.
* Телефон поставщика.
* E-mail.
* Id менеджера сервисного центра, который сопровождает договор с поставщиком.

Атрибуты поставки:

* Id
* Название поставки.
* Номер поставки.
* Дата поставки.
* Поставщик.
* Для какой офисной техники осуществляется поставка.
* Стоимость поставки.
* Стоимость поставки без НДС.
* Ф. И. О. приемщика.
* Id поставщика.

Атрибуты деталей для ремонта:

* Id.
* Название детали.
* Код детали
* Для какой офисной техники подходит деталь.
* Количество на складе.
* Id поставки.

**Составление технического задания**

На основе проведённого анализа предметной области составляем техническое задание (ТЗ) в соответствие с ЕСПД ГОСТ 19.201-78.

––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––

**Оформление ТЗ - в соответствии с ГОСТ 19.201-78**

**Техническое задание**

**Введение**

Наименование - база данных «Сервисный центр по ремонту компьютерной техники)» (далее - БД).

Краткая характеристика - реляционная база данных для обеспечения деятельности сервисного центра при работе с клиентами.

**Основание для разработки**

Задание преподавателя на выполнение учебной лабораторной работы.

**Назначение разработки**

БД предназначена для автоматизации основных видов работ в сервисном центре. А именно: регистрация сломанной компьютерной техники, которую приносят в ремонт, фиксирование неисправности компьютерной техники, контактные данные клиента.

Она должна представлять собой информационную систему обеспечения деятельности работников сервисного центра.

**Требования к программе**

1. Требования к функциональным характеристикам

1.1. Общий состав информации, хранимой в БД

Хранение и регулярное обновление в БД как минимум следующей информации:

- информация, характеризующая данные поступающих обращений, в частности, вся информация из стеллажных карточек: *данные о принесенном устройстве, тип неисправности устройства, ответственный за диагностику, порядковый номер заявки на ремонт, контактные данные клиента, наличие возможности доставить отремонтированную технику клиенту*;

- информация, характеризующая данные о сломанном устройстве, в частности: *название устройства, тип устройства, изготовитель, модель, серийный номер, год выпуска*;

- информация, характеризующая неисправности устройства, в частности: *уровень серьезности поломки (низкий/средний/высокий), возможность ремонта по гарантии (да/нет), название магазина, предоставившего гарантийный ремонт*.

При этом БД обеспечивает: входной контроль данных, помощь при вводе данных в виде списков подстановки и масок ввода, подстановку значений по умолчанию и т.д.

1.2. Состав функций, обеспечивающих деятельность типового пользователя «Работник сервисного центра»

Работник сервисного центра должен иметь возможность выполнять с помощью форм ввода и поисковых форм следующие действия: *находить обращение клиента по номеру, заполнять форму на ремонт и диагностику техники, создавать заявку на доставку отремонтированной техники по адресу клиента, находить контактные данные клиента для связи по обращению*.

БД обеспечивает выполнение следующих типовых запросов сотрудника: *сортировка обращений, выводить список поступающих обращений и т.д*.

1.3. Состав функций, обеспечивающих деятельность типового пользователя «Руководитель сервисного центра»

Кроме функций работника сервисного центра, руководитель должен иметь возможность редактировать и удалять с помощью форм ввода следующую информацию: *информацию в обращении, личные данные пользователя*.

Руководитель должен иметь возможность выполнять следующие запросы: назначение сотрудника сервисного центра на ремонт техники в соответствии с должностью и уровнем компетенций, вывод количества обращений, закрепленных за каждым из сотрудников.

Руководитель должен иметь возможность строить следующие отчёты: *количество закрытых обращений – отремонтированной компьютерной техники каждым из сотрудников сервисного центра, общий процент обращений, успешно закрытых сотрудниками, процент премии сотрудников по результатам оценки работы сервиса клиентами.*

1.4. Общий состав функций

При обращении сотрудника к БД должна выводиться главная форма управления обращениями, позволяющая быстро выбрать нужную функцию *(функция поиска и сортировки поступающих обращений (формы, запросы и т.д.)).*

Разграничение прав доступа для разных групп сотрудников обеспечивается с помощью авторизации на основе логина и пароля.

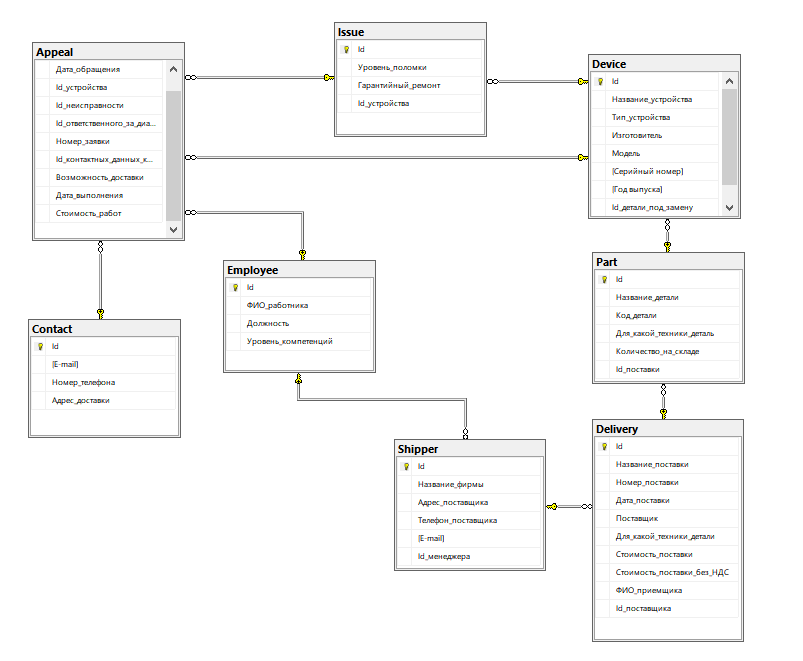
**Задание № 2**

**Концептуальная модель предметной области. ER-диаграмма. Язык концептуального моделирования (ЯКМ).**

**Разработка базы данных**

Проведённый анализ предметной области позволяет построить её концептуальную модель, то есть выделить **сущности** с их атрибутами и определить смысловые **связи** между сущностями.

1. Построение ER-диаграммы



2. Запись модели на ЯКМ

**service\_centerDB**

**Appeal** (Id (PK), Дата\_обращения, Id\_устройства (FK), Id\_неисправности (FK), Id\_ответственного за диагностику (FK), Номер\_заявки, Id\_контактных\_данных\_клиента (FK), Возможность\_доставки, Дата\_выполнения, Стоимость\_работ).

**Contact** (Id (PK), E-mail, Номер\_телефона, Адрес\_доставки).

**Delivery** (Id (PK), Название\_поставки, Номер\_поставки, Дата\_поставки, Поставщик, Для\_какой\_техники\_детали, Стоимость\_поставки, Стоимость\_поставки\_без\_НДС, ФИО\_приемщика, Id\_поставщика (FK)).

**Device** (Id (PK), Название\_устройства, Тип\_устройства, Изготовитель, Модель, Серийный\_номер, Год\_выпуска, Id\_детали\_под\_замену (FK)).

**Employee** (Id (PK), ФИО\_работника, Должность, Уровень\_компетенций).

**Issue** (Id (PK), Уровень\_поломки, Гарантийный\_ремонт, Id\_устройства (FK)).

**Part** (Id (PK), Название\_детали, Код\_детали, Для\_какой\_техники\_деталь, Количество\_на\_складе, Id\_поставки (FK)).

**Shipper** (Id (PK), Название\_фирмы, Адрес\_поставщика, Телефон\_поставщика, E-mail, Id\_менеджера (FK)).

**Задание № 3**

**Логическая модель данных. Правила целостности**

Логическая модель базы данных «Сервисный центр» на языке T-SQL выглядит следующим образом.

/\* Создание базы данных. \*/

CREATE DATABASE service\_centerDB;

/\* Создание таблиц. \*/

CREATE TABLE Appeal

(

Id INT NOT NULL PRIMARY KEY IDENTITY,

[Дата\_обращения] DATE NOT NULL,

[Id\_устройства] INT NOT NULL,

[Id\_неисправности] INT NOT NULL,

[Id\_ответственного\_за\_диагностику] INT NOT NULL,

[Номер\_заявки] NCHAR(20) NOT NULL,

[Id\_контактных\_данных\_клиента] INT NOT NULL,

[Возможность\_доставки] NCHAR(10),

[Дата\_выполнения] DATE NOT NULL,

[Стоимость\_работ] FLOAT NOT NULL

);

CREATE TABLE Contact

(

Id INT NOT NULL PRIMARY KEY IDENTITY,

[E-mail] NCHAR(20),

[Номер\_телефона] NCHAR(15) NOT NULL,

[Адрес\_доставки] NCHAR(40) NOT NULL

);

CREATE TABLE Delivery

(

Id INT NOT NULL PRIMARY KEY IDENTITY,

[Название\_поставки] NCHAR(20) NOT NULL,

[Номер\_поставки] INT NOT NULL,

[Дата\_поставки] DATE NOT NULL,

[Поставщик] NCHAR(20) NOT NULL,

[Для\_какой\_техники\_детали] NCHAR(30) NOT NULL,

[Стоимость\_поставки] INT NOT NULL,

[Стоимость\_поставки\_без\_НДС] INT NOT NULL,

[ФИО\_приемщика] NCHAR(30) NOT NULL

);

CREATE TABLE Device

(

Id INT NOT NULL PRIMARY KEY IDENTITY,

[Название\_устройства] NCHAR(20),

[Тип\_устройства] NCHAR(10) NOT NULL,

[Иготовитель] NCHAR(20),

[Модель] NCHAR(10) NOT NULL,

[Серийный\_номер] NCHAR(10),

[Год\_выпуска] DATE

);

CREATE TABLE Employee

(

Id INT NOT NULL PRIMARY KEY IDENTITY,

[ФИО\_работника] NCHAR(30) NOT NULL,

[Должность] NCHAR(20) NOT NULL,

[Уровень\_кометенций] NCHAR(20) NULL

);

CREATE TABLE Employee

(

Id INT NOT NULL PRIMARY KEY IDENTITY,

[Уровень\_поломки] NCHAR(10) NOT NULL,

[Гарантийный\_ремонт] NCHAR(5) NOT NULL

);

CREATE TABLE Part

(

Id INT NOT NULL PRIMARY KEY IDENTITY,

[Название\_детали] NCHAR(20) NOT NULL,

[Код\_детали] NCHAR(10) NOT NULL,

[Для\_какой\_техники\_деталь] NCHAR(30) NOT NULL,

[Количество\_на\_складе] INT NOT NULL

);

CREATE TABLE Shipper

(

Id INT NOT NULL PRIMARY KEY IDENTITY,

[Название\_фирмы] NCHAR(20) NOT NULL,

[Адрес\_поставщика] NCHAR(30) NOT NULL,

[Телефон\_поставщика] NCHAR(15) NOT NULL,

[E-mail] NCHAR(20) NOT NULL

);

/\* Добавление столбцов для установления зависимостей между таблицами. \*/

ALTER TABLE dbo.Device

ADD Id\_детали\_под\_замену INT NOT NULL;

ALTER TABLE dbo.Issue

ADD Id\_устройства INT NOT NULL;

ALTER TABLE dbo.Delivery

ADD Id\_поставщика INT NOT NULL;

ALTER TABLE dbo.Part

ADD Id\_поставки INT NOT NULL;

ALTER TABLE dbo.Shipper

ADD Id\_менеджера INT NOT NULL;

/\* Добавление внешних ключей для установления зависимостей между таблицами. \*/

ALTER TABLE dbo.Appeal

ADD CONSTRAINT FK\_Appeal\_Device FOREIGN KEY (Id\_устройства) REFERENCES dbo.Device(Id);

ALTER TABLE dbo.Appeal

ADD CONSTRAINT FK\_Appeal\_Issue FOREIGN KEY (Id\_неисправности) REFERENCES dbo.Issue(Id);

ALTER TABLE dbo.Appeal

ADD CONSTRAINT FK\_Appeal\_Employee FOREIGN KEY (Id\_ответственного\_за\_диагностику) REFERENCES dbo.Employee(Id);

ALTER TABLE dbo.Appeal

ADD CONSTRAINT FK\_Appeal\_Contact FOREIGN KEY (Id\_контактных\_данных\_клиента) REFERENCES dbo.Contact(Id);

ALTER TABLE dbo.Device

ADD CONSTRAINT FK\_Device\_Part FOREIGN KEY (Id\_детали\_под\_замену) REFERENCES dbo.Part(Id);

ALTER TABLE dbo.Issue

ADD CONSTRAINT FK\_Issue\_Device FOREIGN KEY (Id\_устройства) REFERENCES dbo.Device(Id);

ALTER TABLE dbo.Part

ADD CONSTRAINT FK\_Part\_Delivery FOREIGN KEY (Id\_поставки) REFERENCES dbo.Delivery(Id);

ALTER TABLE dbo.Delivery

ADD CONSTRAINT FK\_Delivery\_Shipper FOREIGN KEY (Id\_поставщика) REFERENCES dbo.Shipper(Id);

ALTER TABLE dbo.Shipper

ADD CONSTRAINT FK\_Shipper\_Employee FOREIGN KEY (Id\_менеджера) REFERENCES dbo.Employee(Id);

Заполнение БД необходимыми данными на языке T-SQL выглядит следующим образом:

- для сущности **Employee**: INSERT Employee VALUES (0, 'Ivanov\_PA', 'Trainee', 'Without\_experience');

- для сущности **Shipper**: INSERT Shipper VALUES ('Pavlov\_Details', 'Moscow, pr. Lenina, 9', '+79991110211', 'pavlov123@mail.ru', 4);

- для сущности **Delivery**: INSERT INTO VALUES ('Matrix\_for\_monitors', '121012', '2015-10-21', 'PC\_Repair', 'Monitor', 120000, 100000, 'Sergeev\_SV', 2);

- для сущности **Part**: INSERT PartVALUES ('Matrix\_101A', '001001', 'Monitor', 110, 1);

- для сущности **Device**: INSERT Device VALUES ('Acer\_209VX', 'Monitor', 'Acer', '209VX', 'S1XC245B80', '2013-05-01', 1);

- для сущности **Issue**: INSERT Issue VALUES ('Critical', 'Yes', 1);

- для сущности **Contact**: INSERT Contact VALUES ('npetrov@yandex.ru', '+79098431201', 'Moscow, Burova, 8');

- для сущности **Appeal**: INSERT Appeal VALUES ('2021-10-05', 1, 1, 0, '10101a', 1, 'Yes', '2021-10-15', 15000);

Теперь следует проверить, не нарушены ли в данном проекте какие-либо принципы нормализации, т.е. что любое неключевое поле каждой таблицы:

* функционально зависит от полного первичного ключа, а не от его части (если ключ составной);
* не имеет функциональной зависимости от другого неключевого поля.

Анализ сущностей Appeal, Contact, Delivery, Device, Employee, Issue, Part, Shipper, состоящих из составного ключа и нескольких неключевых полей, показал, что в них нет функциональных связей между неключевыми полями. Последние же не зависят функционально от какой-либо части составного ключа.

**Задание № 4**

**Создание запросов на языке SQL**

**4.1. Запросы на выборку данных**

Для начала продемонстрируем пример работы БД с запросами на выборку данных из таблиц.

4.1.1. Запросы для типового пользователя «Работник сервисного центра»

Запрос № 1. В качестве первого демонстрационного запроса выберем все данные из таблицы Appeal, отсортировав их по дате обращения, следующим запросом:

SELECT \* FROM Appeal

ORDER BY Дата\_обращения;

На рисунке 1 представлена выборка всех данных из таблицы Appeal.

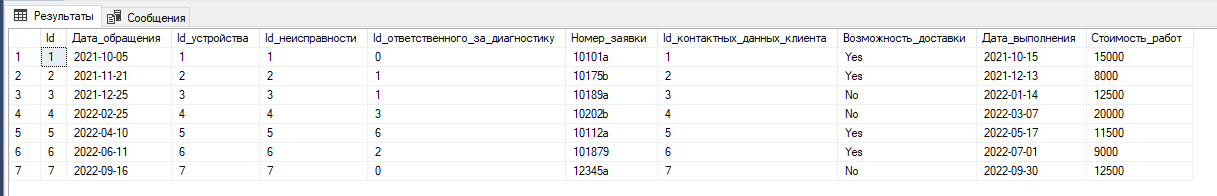


Рисунок 1 – Выборка данных из таблицы Appeal

Запрос № 2. Из таблицы Appeal выберем данные, чтобы посмотреть возможность доставки отремонтированной техники клиентам по указанным адресам. Сортировка производится по id клиента:

SELECT Номер\_заявки, Id\_контактных\_данных\_клиента, Возможность\_доставки

FROM Appeal

ORDER BY Id\_контактных\_данных\_клиента;

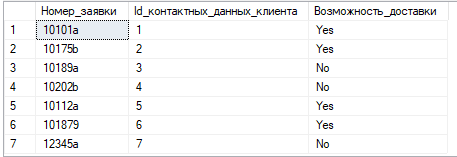


Рисунок 2 – Проверка возможности доставки техники клиентам

Запрос № 3. Для того чтобы написать клиентам электронное письмо с запросом на доставку по адресу, выберем те строки таблицы Contact, где указан E-mail клиентов:

SELECT \* FROM Contact WHERE [E-mail] IS NOT NULL;

На рисунке 3 изображен результат запроса на языке T-SQL.

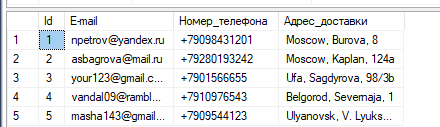


Рисунок 3 – Выборка строк таблицы Contact с указанным E-mail

4.1.2. Запросы для типового пользователя «Руководитель сервисного центра»

Запрос № 1. Для начала посмотрим список всех поставок деталей в сервисный центр, отсортировав их по дате поставки, следующим запросом:

SELECT \* FROM Delivery

ORDER BY Дата\_поставки;

На рисунке 1 представлена выборка всех данных из таблицы Delivery.

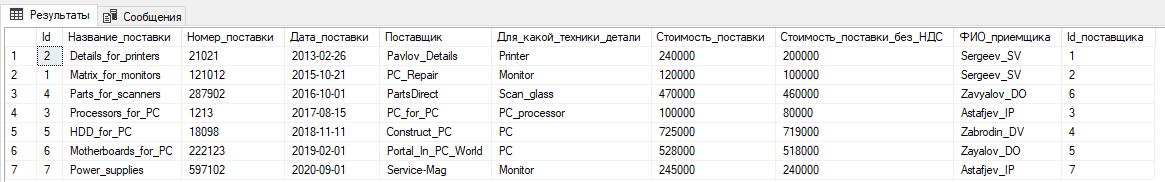


Рисунок 1 – Выборка данных из таблицы Delivery

Запрос № 2. Из таблицы Part выберем данные, чтобы посмотреть, сколько деталей осталось на складе. Сортировка производится по названию деталей:

SELECT \* FROM Part

ORDER BY Название\_детали;

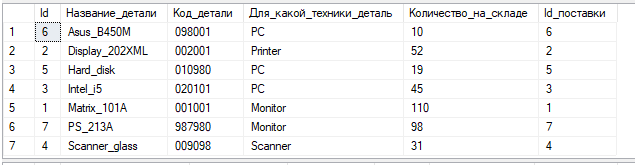


Рисунок 2 – Проверка наличия деталей на складе

Запрос № 3. Из таблицы Device выберем только данные о компьютерах, находящихся в ремонте, чтобы посмотреть их серийные номера:

SELECT \* FROM Device

WHERE Тип\_устройства = 'PC'

ORDER BY Серийный\_номер;

На рисунке 3 изображен результат запроса на языке T-SQL.

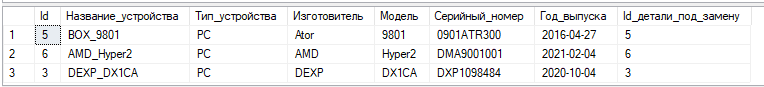


Рисунок 3 – Выборка данных о компьютерах из таблицы Device

**4.2. Запросы на изменение данных**

После выборки данных покажем примеры запросов на изменение данных в таблицах БД.

4.2.1. Запросы для типового пользователя «Работник сервисного центра»

Запрос № 4. В случае ухода сотрудника сервисного центра в отпуск или на больничный можно назначить другого работника на ремонт конкретной техники следующим запросом:

UPDATE Appeal

SET Id\_ответственного\_за\_диагностику=3 WHERE Id\_ответственного\_за\_диагностику=1;

На рисунке 1 представлены измененные данные в столбце [Id\_ответственного\_за\_диагностику].

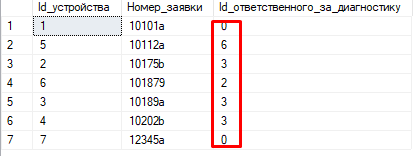
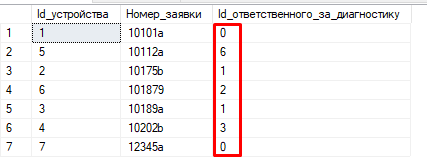


Рисунок 1 – Изменение данных в таблице Appeal

Запрос № 5. Для того чтобы БД правильно функционировала с бухгалтерским ПО сервисного центра, нужно изменить данные таблицы Contact таким образом, чтобы вместо значения NULL в столбце E-mail было значение «no-email». Сделать это можно следующим образом:

UPDATE Contact

SET [E-mail]='no e-mail' WHERE [E-mail] IS NULL;

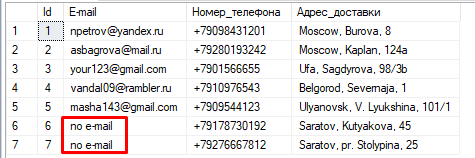
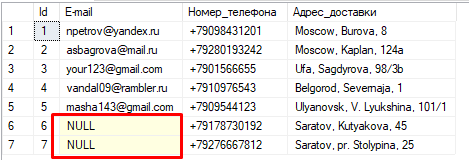


Рисунок 2 – Заполнение пустых ячеек в таблице Contact

Запрос № 6. В связи с изменением политики доставки техники клиентам после её ремонта требуется изменить данные в БД таким образом, что при стоимости работ менее 15 000 доставка техники клиенту отсутствует. Осуществляется это следующим запросом:

UPDATE Appeal

SET Возможность\_Доставки='No' WHERE Стоимость\_работ < 15000;

UPDATE Appeal

SET Возможность\_Доставки='Yes' WHERE Стоимость\_работ >= 15000;

На рисунке 3 изображен результат запроса на языке T-SQL.

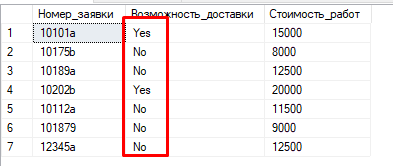
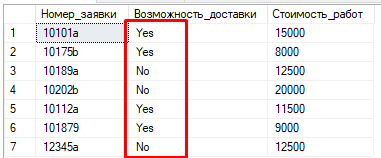


Рисунок 3 – Выборка строк таблицы Contact с указанным E-mail

4.2.2. Запросы для типового пользователя «Руководитель сервисного центра»

Запрос № 4. В связи с недавним изменением процесса предоставления гарантии на купленную технику магазинами электроники меняется политика сервисного центра в сфере ремонта по гарантии – если поломка не критична для техники, то ремонт по гарантии больше не предоставляется. Для актуализации информации в БД создадим следующий запрос:

UPDATE Issue

SET Гарантийный\_ремонт='No' WHERE Уровень\_поломки != 'Critical';

UPDATE Issue

SET Гарантийный\_ремонт='Yes' WHERE Уровень\_поломки = 'Critical';

На рисунке 1 представлено изменение столбцов в таблице Issue.

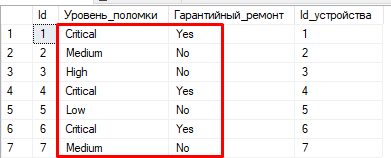
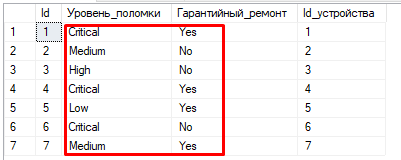


Рисунок 1 – Актуализация таблицы Issue

Запрос № 5. Недавно у поставщика «Детали Павлова» возросли цены на поставку деталей. Нужно изменить стоимость последней поставки на основе нового прайс-листа следующим запросом:

UPDATE Delivery

SET Стоимость\_поставки += 15000 WHERE Поставщик = 'Pavlov\_Details';

UPDATE Delivery

SET Стоимость\_поставки\_без\_НДС += 1500 WHERE Поставщик = 'Pavlov\_Details';

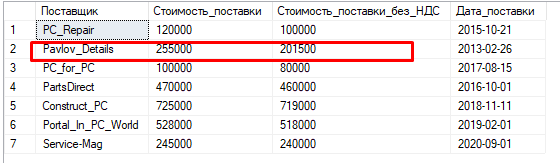
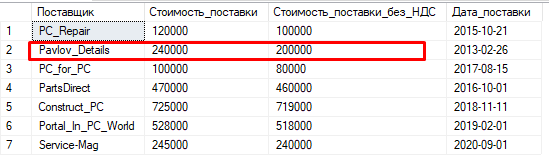


Рисунок 2 – Актуализация данных для поставщика «Pavlov\_Details» таблицы Delivery

Запрос № 6. Менеджер сервисного центра Маслова Е.П. сообщила о том, что срочно вынуждена взять больничный. Вместо неё на время больничного договорами с поставщиками будет заниматься инженер Астафьев Л.П. Актуализировать информацию в таблице Shipper можно таким запросом:

UPDATE Shipper SET Id\_менеджера = 5;

На рисунке 3 изображен результат запроса на языке T-SQL.

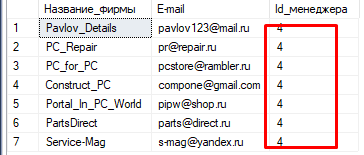
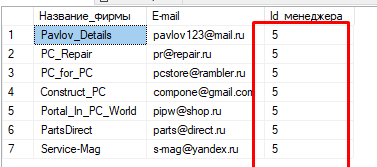


Рисунок 3 – Замена менеджера в таблице Shipper

**4.3. Запросы с использованием группировки (GROUP BY, HAVING)**

4.3.1. Запросы для типового пользователя «Работник сервисного центра»

Запрос № 7. Требуется рассчитать минимальную, общую и среднюю прибыль с отремонтированной техники, при этом нужно учесть параметр возможности доставки техники клиенту после ремонта. Сделать такую выборку можно следующим запросом:

SELECT Возможность\_доставки, MIN(Стоимость\_работ) As Минимальная\_прибыль, SUM(Стоимость\_работ) As Общая\_прибыль, AVG(Стоимость\_работ) As Средняя\_прибыль

FROM Appeal

GROUP BY Возможность\_доставки;

На рисунке 1 представлена выборка по результату запроса.

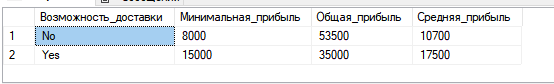


Рисунок 1 – расчет минимальной, общей и средней прибыли с выполненных заказов

Запрос № 8. Продемонстрируем использование оператора HAVING. Создадим запрос для типов устройств, которые находятся в ремонте и количество которых больше одного, отсортировав по количеству в порядке убывания:

SELECT Тип\_устройства, COUNT(\*) As Количество

FROM Device

GROUP BY Тип\_устройства

HAVING COUNT(\*) > 1

ORDER BY Количество DESC;

На рисунке 2 изображен результат описанного запроса.

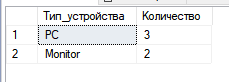


Рисунок 2 – запрос с использованием оператора HAVING

4.3.2. Запросы для типового пользователя «Руководитель сервисного центра»

Запрос № 7. Рассчитаем минимальную, максимальную и среднюю цену по каждой поставке деталей в сервисный центр, учитывая, что средняя цена поставки будет без копеек (0 коп.), и отсортировав по средней цене в порядке убывания, следующим запросом:

SELECT Для\_какой\_техники\_детали, MIN(Стоимость\_поставки) As Минимальная\_цена, MAX(Стоимость\_поставки) As Максимальная\_цена, AVG(Стоимость\_поставки) As Средняя\_цена

FROM Delivery

GROUP BY Для\_какой\_техники\_детали

HAVING AVG(Стоимость\_поставки) % 10 = 0

ORDER BY Средняя\_цена DESC;

На рисунке 1 изображена выборка из таблицы при использовании описанного запроса.

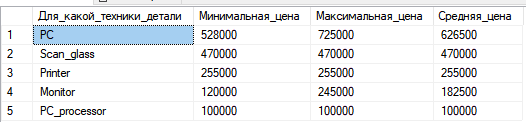


Рисунок 1 – результат запроса T-SQL

Запрос № 8. Из таблицы Delivery выбрать сотрудников сервисного центра, которые приняли более одной поставки деталей. Сделать это можно следующим запросом:

SELECT ФИО\_приемщика, COUNT(Название\_поставки) As Количество\_принятых\_поставок

FROM Delivery

GROUP BY ФИО\_приемщика

HAVING COUNT(Название\_поставки) > 1;

На рисунке 2 представлен результат запроса.

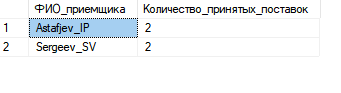


Рисунок 2 – выборка приемщиков, принявших более одной поставки деталей

**4.4. Соединение таблиц (JOIN)**

4.4.1. Запрос для типового пользователя «Работник сервисного центра»

Запрос № 9. Представим, что работнику сервисного центра требуется посмотреть выполненные заказы по столбцам:

- тип устройства;

- год выпуска;

- производитель.

При этом необходимо учесть, что выборку нужно осуществить при условии того, что техника была отремонтирована по гарантии и отсортировать выборку по году выпуска в порядке убывания. Для такого запроса соединим таблицы Device и Issue следующим запросом:

SELECT D.Id, D.Тип\_устройства, D.Год\_выпуска, D.Изготовитель

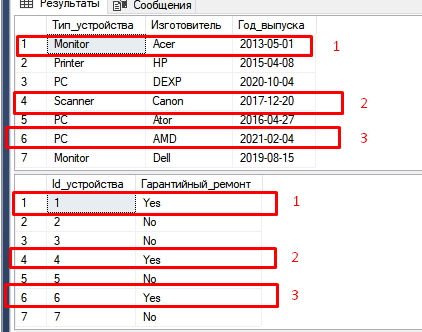
FROM Device AS D

JOIN Issue AS I ON I.Id = D.Id

WHERE Гарантийный\_ремонт = 'Yes'

ORDER BY Год\_выпуска DESC;

На рисунке 1 представлен результат соединения двух таблиц по заданным критериям.



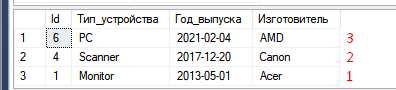


Рисунок 1 – соединение таблиц Device и Issue при помощи оператора JOIN

4.4.2. Запрос для типового пользователя «Руководитель сервисного центра»

Запрос № 9. Руководителю сервисного центра требуется выяснить, какие поставщики оформили продажу деталей для сервисного центра на сумму больше 250 000 рублей без НДС. Такой запрос пишется при помощи соединения таблиц Shipper и Delivery. Отсортируем выборку по стоимости поставки в порядке убывания:

SELECT S.Название\_фирмы, Адрес\_поставщика, [E-mail]

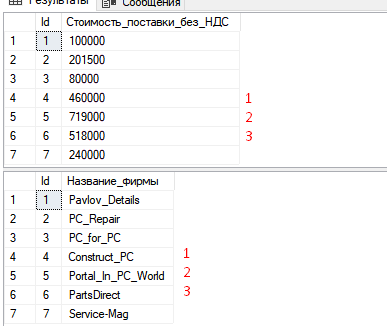
FROM Shipper AS S

JOIN Delivery AS D ON D.Id = S.Id

WHERE D.Стоимость\_поставки\_без\_НДС > 250000

ORDER BY D.Стоимость\_поставки DESC;

На рисунке 1 изображено соединение таблиц при помощи написанного запроса T-SQL.



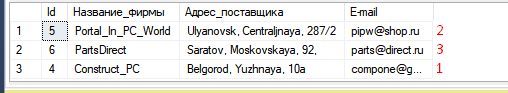


Рисунок 1 – выборка посредством соединения таблиц Shipper и Delivery оператором JOIN