

**Санкт-Петербургский государственный
университет телекоммуникаций
им. Профессора М.А. Бонч-Бруевича
Факультет ИСиТ**

Курсовая работа
по дисциплине «Управление данными»
на тему «Информационная система компьютерной фирмы»

Выполнил:
студент группы ИСТ-666
Ерохин Б. А.

Проверил:
Сабинин О. Ю

Содержание

Введение.....	3
1 Задание на проектирование.....	4
2 Описание предметной области.....	7
3 Процесс создания модели БД.....	8
3.1 Создание логической модели.....	8
3.2 Создание физической модели.....	11
3.3 Доказательства решённости всех поставленных в задании задач.....	13
3.4 Подтверждение нормализованности всех отношений.....	15
4 Отчет по ошибкам из Validator.....	16
5 Тексты и результаты выполнения запросов.....	17
Заключение.....	23

Введение

Информатизация общества и связанное с ней широкое распространение вычислительной техники и средств коммуникации устанавливают максимальный приоритет задаче, для создания специальных методов обработки данных: их поиск, защиту, обработку и хранение.

Большие объёмы информации практически невозможно обработать без специальных средств машинной обработки. В последнее время широкое распространение получили автоматизированные информационные системы: информационно-справочные, информационно-поисковые. Все они предназначены для регистрации, хранения и обработки данных с целью поиска и выдачи ответов на запросы пользователей. В большинстве случаев автоматизированные информационные системы разрабатывают как базы данных (БД).

Базы данных, которые широко используются на практике, - это совокупность специальных методов и математических, информационных, программных, языковых, организационных и технических средств для поддержки динамической информационной модели предметной отрасли с целью обеспечения информационных запросов пользователей. Скорость доступа к информации, которая хранится в базах данных и удобство работы с ней зависит от организации структуры хранения данных, вида представления, возможностей поиска.

1 Задание на проектирование

Ваша фирма занимается ремонтом и апгрейдингом персональных компьютеров. У вас имеется склад комплектующих и штат сотрудников по категориям. У вас есть один стационарный цех и несколько приемных пунктов. В приемных пунктах работают кроме приемщиков дежурные мастера, которые могут выполнять срочный ремонт.

1. Вы принимаете заказы, которые выполняются у вас в фирме стационарно или можете высылать ваших мастеров к заказчикам для диагностики и устранения неисправностей на месте.
2. Все виды ремонта имеют свою стоимость.
3. Стоимость ремонта складывается из стоимости комплектующих и стоимости ремонтных работ.
4. Ваша фирма дает гарантию на ремонт и если возвращается компьютер по гарантии, то его ремонт должен осуществлять тот, кто делал первый ремонт, при этом в объем его работ гарантийный ремонт не должен входить.
5. Ремонт выполняется только при наличие соответствующих комплектующих на складе.
6. В распоряжении фирмы имеется 2 микроавтобуса, которые могут забирать аппаратуру и доставлять заказы не место. Заказчики могут доставлять и забирать технику самостоятельно. За транспортные услуги взимается дополнительная плата.
7. В одном из приемных пунктов приемщик дополнительно может выполнять роль транспортного диспетчера.
8. При приеме заказа, который требует выезда мастера домой, необходимо предварительное согласование работы мастеров на выезде.

Ориентировочно для каждого выезда с диагностикой выделяется 1 час на работу + время на транспорт, при этом выездной мастер должен иметь возможность приема заказа на выезде, то есть он едет с ноутбуком, на котором установлена версия Вашей системы и принимает заказ, который попадает в БД. По приезде в центральный офис – он должен согласовать – синхронизировать все принятые заказы с существующими. При возникновении накладок, необходимо позвонить клиенту и сообщить о возможных изменениях по срокам выполнения заказа из-за накладок.

9. При планировании работы сборочных цехов необходимо учитывать загрузку их персонала. Для этого планируется работа на каждого исполнителя, на сборку и тестирование 1 компьютера отводится 2 человеко-дня .

Пользователями системы являются приемщики заказов в приемных пунктах, мастера, которые работают как стационарно, так и на выезде, шоферы.

Приемщика заказов должны иметь возможность оформить заказ с учетом наличия комплектующих и исполнителей- конкретного цеха и мастеров.

Мастера должны иметь возможность установить отметку о принятии заказа к исполнению и о исполнении заказа, то есть готовности его.

Шоферы должны иметь возможность посмотреть, что им на текущую дату запланировано из поездок по городу и отчитаться о выполнении развозки выполненных работ или принятии новых заказов.

Создать запросы:

1. для определения поездок, запланированных шоферу на конкретный день;
2. для определения заказов, принятых к исполнению в конкретный день;

3. для определения списка ремонтов с максимальной стоимостью;
4. для определения списка сотрудников, обслуживших наибольшее количество клиентов за заданный интервал времени;
5. для определения общего списка сотрудников и клиентов с указанием статуса (сотрудник или клиент) и общего количества сотрудников, и общего количества клиентов;
6. для определения списка клиентов, стоимость заказов которых выше средней стоимости заказов;
7. для определения списка услуг, названия которых содержат два или более слов;
8. для определения списка комплектующих, имеющихся на складе в наибольшем количестве;
9. для определения для конкретного заказа требуемых комплектующих;
10. для определения заказов, для которых нет хотя бы одной комплектующей на складе.

2 Описание предметной области

Предметной областью данной работы является база фирмы, которая занимается ремонтом и апгрейдингом персональных компьютеров. Рассматривая данную тему, необходимо выделить основные сущности, которые являются ключевыми для создания модели данной системы.

В фирме работают мастера, которые занимаются ремонтом, диспетчеры, которые принимают заказы и водители, которые занимаются перевозкой техники. На основе такой фирмы была создана база данных учета заявок компьютерной фирмы, которая содержит информацию о ремонте и апгрейде компьютеров.

Информационная система позволит максимально автоматизировать работу по ведению учета заказов и их выполнения, по наличию необходимых комплектующих и персонала, который сможет выполнить работу. Всё это позволит существенно уменьшить трудоемкость работы.

3 Процесс создания модели БД

3.1 Создание логической модели

В рассматриваемой системе можно выделить следующие сущности:

1. Сотрудники — включает в себя информацию о сотруднике (Код сотрудника, Фамилия, Имя, Отчество, Специализация).
2. Клиенты — включает в себя информацию о клиенте (Код клиента, Фамилия, Имя, Отчество, Телефон, Адрес).
3. Ремонтные работы — включает в себя информацию о видах ремонтных работ (Код типа ремонта, Название, Стоимость).
4. Комплектующие — включает в себя информацию о имеющихся комплектующих (Код комплектующей, Название, Количество, Тип).
5. Автопарк — включает в себя информацию об автомобилях в компании (Код автомобиля, Тип, Стоимость)
6. Приём — включает в себя информацию о принятых заказах (Код сотрудника, Код заказчика, Код приёмного пункта, Код автомобиля, Время приёма, Время выполнения)

Между сущностями существуют следующие связи:

1. у 1 клиента может быть несколько заказов, но 1 заказ может быть только у 1 заказчика (1:M);
2. над 1 заказом может работать только 1 сотрудник, но 1 сотрудник может работать над несколькими заказами (1:M);
3. для 1 заказа может быть заказано несколько ремонтных работ и 1 ремонтная работа может быть заказана для нескольких заказов (M:M);
4. для 1 заказа может быть заказано несколько комплектующих и 1 комплектующая может быть заказана для нескольких заказов (M:M);

5. к 1 заказу может быть прикреплена только одна машина, но 1 машина может обслуживать сразу несколько заказов (1:M);

Логическая модель базы данных представлена на рисунке 1.

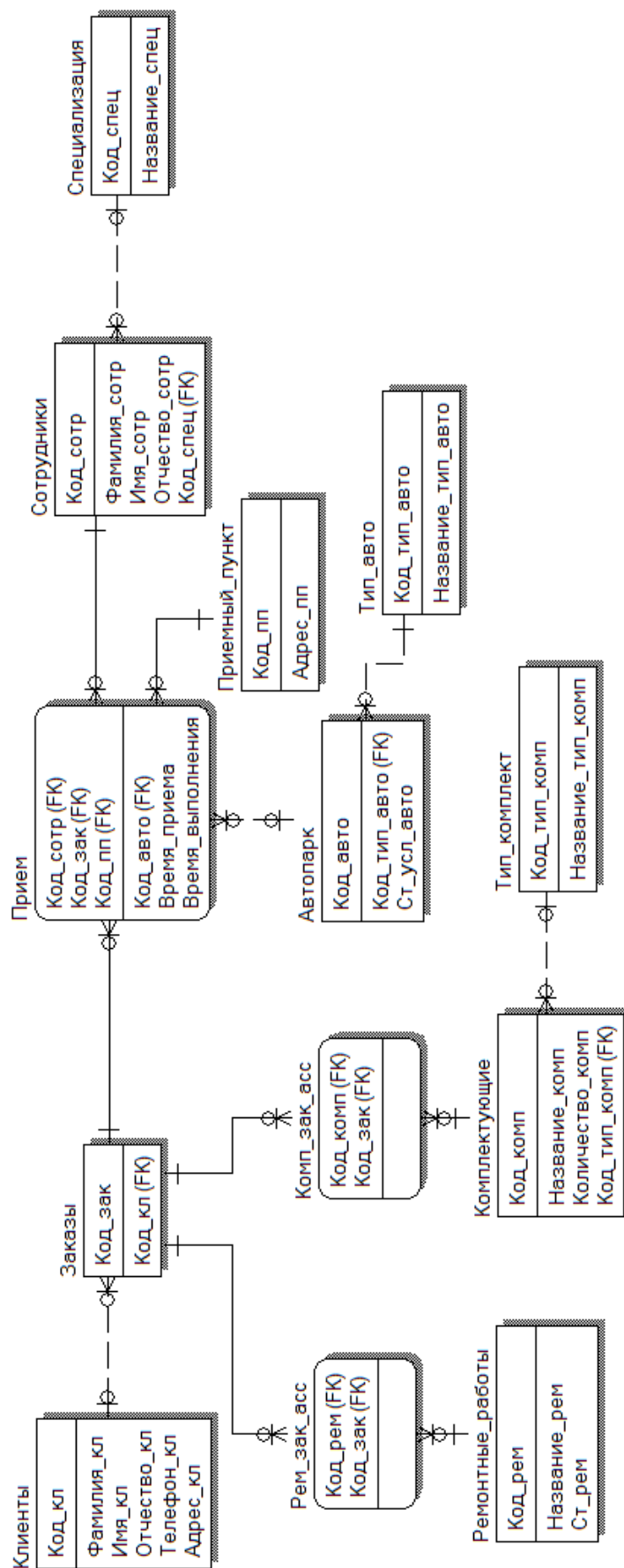


Рисунок 1

3.2 Создание физической модели

Для реализации связи М:М (многие ко многим) вводятся ассоциативные таблицы:

1. Для связи между таблицами Заказы и Ремонтные работы вводится таблица Рем_зак_асс.
2. Для связи между таблицами Заказы и Комплектующие вводится таблица Комп_зак_асс.

Физическая модель представлена на рисунке 2.

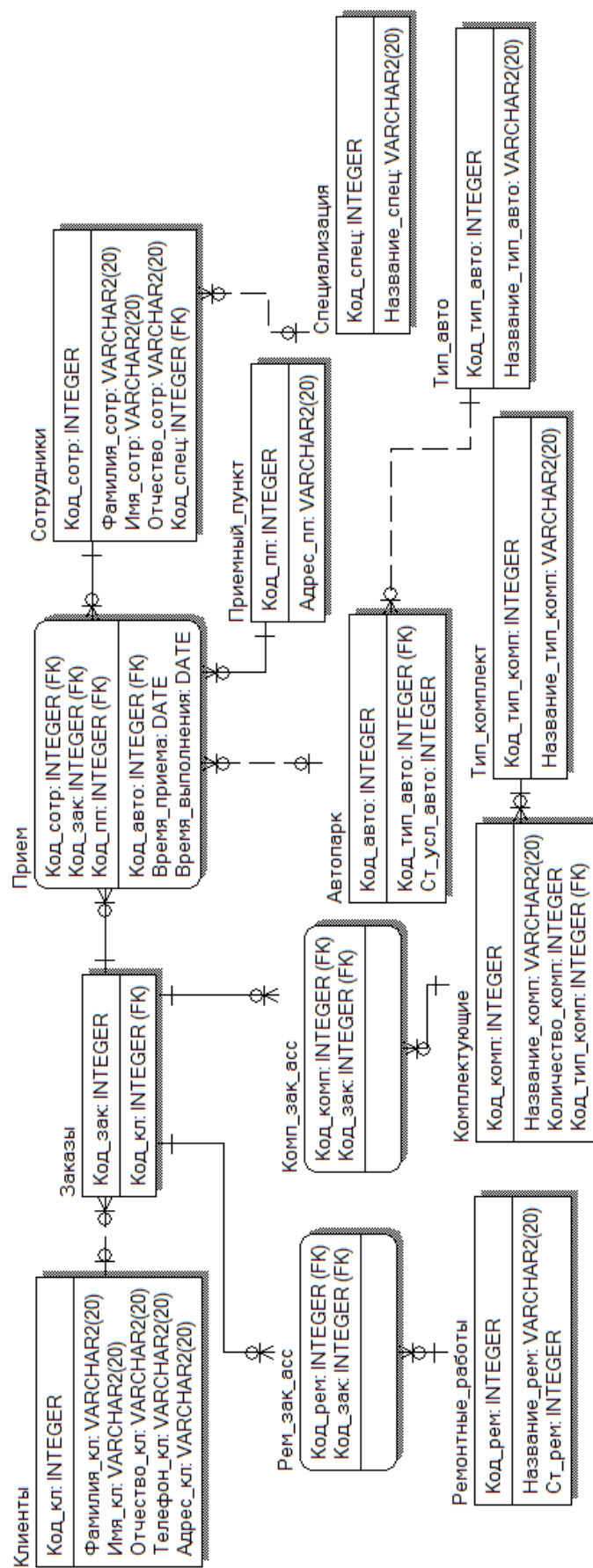


Рисунок 2

3.3 Доказательства решённости всех поставленных в задании задач

Для некоторых полей вводятся значения по умолчанию и условия проверки вводимых пользователем значений (Ст_рем ≥ 0 и так далее).

Для связей в ассоциативных таблицах устанавливаются те типы связей, которые представлены на рисунках 3 и 4.

При удалении данных из дочерней таблицы никаких действий ни с родительской, ни с дочерней таблицей производиться не будет (NONE). При попытке обновить или вставить в дочернюю таблицу система выдаст ошибку, это позволит сохранить информацию (RESTRICT). Действие в связи SET NULL присваивает полям, которые зависят от изменившихся данных, значение NULL.

Relationship: R/13 (Заказы to Рем_зак_acc)	
Name: R/13	
General Definition Rolename RI Actions UDP	
RI Actions	
Child Delete:	NONE
Child Insert:	RESTRICT
Child Update:	RESTRICT
Parent Delete:	RESTRICT
Parent Insert:	NONE
Parent Update:	RESTRICT

Рисунок 3

Relationships [X]

Relationship: R/31 (Автопарк to Прием) ▼

Name: R/31 [New...] [Delete]

General | Definition | Rolename | **RI Actions** | UDP

RI Actions

Child Delete:	NONE ▼	Parent Delete:	SET NULL ▼
Child Insert:	SET NULL ▼	Parent Insert:	NONE ▼
Child Update:	NO ACTION ▼	Parent Update:	SET NULL ▼

[OK] [Cancel]

Рисунок 4

3.4 Подтверждение нормализованности всех отношений

Согласно требованию первой нормальной формы, в таблице все строки должны быть различны и все элементы внутри ячеек должны быть атомарны. Отсюда следует, что спроектированная модель находится в первой нормальной форме.

Таблицы также находятся и во второй нормальной форме, так как любое её поле, не входящее в состав первичного ключа, функционально полно зависит от первичного ключа.

Также можно сказать, что эти таблицы будут находиться в третьей нормальной форме, потому что неключевые атрибуты зависят от составного ключа нетранзитивно, эти атрибуты взаимно не зависимы, что удовлетворяет правилу третьей нормальной формы.

Таблицы соответствуют нормальной форме Бойса-Кодда, так как потенциальный ключ в отношениях является детерминантом для любой функциональной зависимости.

У таблиц отсутствуют многозначные зависимости атрибутов, не являющиеся функциональными зависимостями, поэтому условие четвёртой нормальной формы соблюдается.

4 Отчет по ошибкам из Validator

После проверки созданной модели программой ERwin Data Model Validator, были обнаружены следующие ошибки (Рисунок 5):

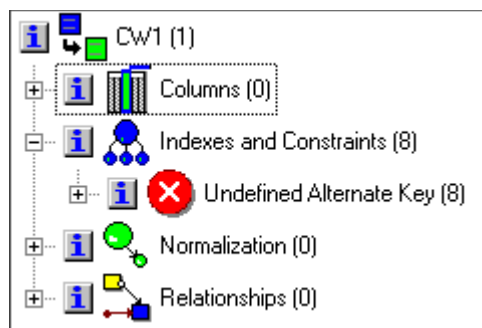


Рисунок 5

Ошибка Undefined Alternate Key означает то, что в таблице отсутствует альтернативный ключ. Исправить эту ошибку достаточно просто: нужно добавить альтернативный ключ в таблицу.

После добавления необходимых альтернативных ключей, все ошибки были исправлены (Рисунок 6):

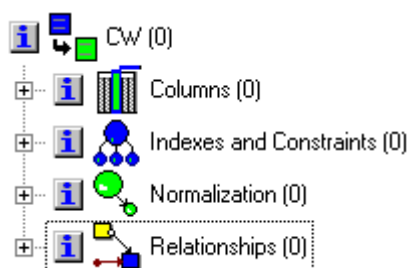


Рисунок 6

5 Тексты и результаты выполнения запросов

Запрос для определения поездок, запланированных шоферу на конкретный день

```
SELECT
  прием.код_зак AS "Код заказа",
  клиенты.фамилия_кл AS "Фамилия",
  клиенты.имя_кл AS "Имя",
  клиенты.отчество_кл AS "Отчество",
  клиенты.телефон_кл AS "Телефон",
  клиенты.адрес_кл AS "Адрес"
FROM
  прием
INNER JOIN заказы ON
  заказы.код_зак = прием.код_зак
INNER JOIN клиенты ON
  клиенты.код_кл = заказы.код_кл
WHERE
  код_авто IS NOT NULL AND
  время_приема = to_date('08.05.15', 'dd.mm.yy');
```

	Код заказа	Фамилия	Имя	Отчество	Телефон	Адрес
1	2	Петров	Пётр	Петрович	71234567890	Москва
2	3	Петров	Пётр	Петрович	71234567890	Москва

Рисунок 7

Запрос для определения заказов, принятых к исполнению в конкретный день

```
SELECT
  код_зак AS "Код заказа"
FROM
  прием
WHERE
  время_приема = to_date('08.05.15', 'dd.mm.yy');
```

	Код заказа
1	2
2	3

Рисунок 8

Запрос для определения списка ремонтов с максимальной стоимостью

```
SELECT
  код_зак AS "Код заказа",
  sum(ст_рем) AS "Стоимость ремонта"
FROM
  рем_зак_асс
INNER JOIN ремонтные_работы ON
```

```

    ремонтные_работы.код_рем = рем_зак_асс.код_рем
GROUP BY
    код_зак
HAVING
    sum(ст_рем) =
    (
        SELECT
            max(sum(ст_рем))
        FROM
            рем_зак_асс
        INNER JOIN ремонтные_работы ON
            ремонтные_работы.код_рем = рем_зак_асс.код_рем
        GROUP BY
            код_зак
    )

```

	Код заказа	Стоимость ремонта
1	4	5000

Рисунок 9

Запрос для определения списка сотрудников, обслуживших наибольшее количество клиентов за заданный интервал времени

```

SELECT
    сотрудники.фамилия_сотр AS "Фамилия",
    сотрудники.имя_сотр AS "Имя",
    сотрудники.отчество_сотр AS "Отчество"
FROM
    сотрудники
INNER JOIN прием ON
    прием.код_сотр = сотрудники.код_сотр
GROUP BY
    прием.код_сотр,
    сотрудники.фамилия_сотр,
    сотрудники.имя_сотр,
    сотрудники.отчество_сотр
HAVING
    count(прием.код_сотр) =
    (
        SELECT
            max(count(код_сотр))
        FROM
            прием
        WHERE
            время_приема BETWEEN to_date('01.05.15', 'dd.mm.yy') AND
            to_date('20.05.15', 'dd.mm.yy')
        GROUP BY
            код_сотр
    )

```

	Фамилия	Имя	Отчество
1	Журавель	Ярослав	Михайлович
2	Скачков	Исай	Александрович

Рисунок 10

Запрос для определения списка клиентов, стоимость заказов которых выше средней стоимости заказов

```

SELECT
    клиенты.код_кл AS "Код клиента",
    клиенты.фамилия_кл AS "Фамилия",
    клиенты.имя_кл AS "Имя",
    клиенты.отчество_кл AS "Отчество",
    клиенты.телефон_кл AS "Телефон",
    клиенты.адрес_кл AS "Адрес"
FROM
    клиенты
INNER JOIN заказы ON
    заказы.код_кл = клиенты.код_кл
INNER JOIN рем_зак_асс ON
    рем_зак_асс.код_зак = заказы.код_зак
INNER JOIN ремонтные_работы ON
    ремонтные_работы.код_рем = рем_зак_асс.код_рем
GROUP BY
    клиенты.код_кл,
    клиенты.фамилия_кл,
    клиенты.имя_кл,
    клиенты.отчество_кл,
    клиенты.телефон_кл,
    клиенты.адрес_кл
HAVING
    sum(ремонтные_работы.ст_рем) >
    (
        SELECT
            avg(sum(ремонтные_работы.ст_рем))
        FROM
            рем_зак_асс
        INNER JOIN ремонтные_работы ON
            ремонтные_работы.код_рем = рем_зак_асс.код_рем
        GROUP BY
            рем_зак_асс.код_зак
    )

```

	Код клиента	Фамилия	Имя	Отчество	Телефон	Адрес
1	3	Сидоров	Сидор	Сидорович	78887776655	Выборг
2	2	Петров	Пётр	Петрович	71234567890	Москва

Рисунок 12

Запрос для определения списка услуг, названия которых содержат два или более слов

```
SELECT
    код_рем AS "Код услуги",
    название_рем AS "Название услуги"
FROM
    ремонтные_работы
WHERE
    length(название_рем) - length(replace(название_рем, ' ', '')) + 1 >= 2
```

	Код услуги	Название услуги
1	2	Замена комплектующих

Рисунок 13

Запрос для определения списка комплектующих, имеющихся на складе в наибольшем количестве

```
SELECT
    код_комп AS "Код",
    название_комп AS "Название"
FROM
    комплектующие
WHERE
    количество_комп =
    (
        SELECT
            max(количество_комп)
        FROM
            комплектующие
    )
```

	Код услуги	Название услуги
1	2	Замена комплектующих

Рисунок 14

Запрос для определения для конкретного заказа требуемых комплектующих

```
SELECT
    комплектующие.название_комп AS "Название"
FROM
    комп_зак_асс
INNER JOIN комплектующие ON
    комплектующие.код_комп = комп_зак_асс.код_комп
WHERE
    комп_зак_асс.код_зак = 2
GROUP BY
    комплектующие.название_комп
```

	Код	Название
1	6	ВИД1
2	14	ЖДЗ

Рисунок 15

Запрос для определения заказов, для которых нет хотя бы одной комплектующей на складе.

```
SELECT
    комп_зак_асс.код_зак AS "Заказ"
FROM
    комп_зак_асс
INNER JOIN комплектующие ON
    комплектующие.код_комп = комп_зак_асс.код_комп
WHERE
    комплектующие.количество_комп = 0
GROUP BY
    комп_зак_асс.код_зак
```

Название
1 МП2
2 МП3

Рисунок 16

Заключение

В результате выполнения работы была создана информационная система компьютерной фирмы, отвечающая всем, поставленным в задаче, требованиям. В процессе проверки базы данных аномалий выявлено не было, все таблицы нормализованы.

В ходе данной курсовой работы было осуществлено построение модели базы данных в ERwin Data Modeler и проверено на наличие ошибок проектирования в ERwin Data Model Validator. Была произведена нормализация таблиц, прямое и обратное проектирование, заполнение таблиц данными и написание SQL запросов.