



МИНОБРНАУКИ РОССИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования  
**«МИРЭА – Российский технологический университет»**  
**РТУ МИРЭА**

---

Институт информационных технологий  
Кафедра промышленной информатики (ПИ)

**ОТЧЕТ ПО ПРАКТИЧЕСКИМ РАБОТАМ**  
по дисциплине  
**«Разработка баз данных»**

**Выполнил:** студент группы ИВБО-02-19

К. Ю. Денисов

**Принял:** инженер, ассистент

Д. С. Киселев

Работа выполнена «\_\_\_\_» \_\_\_\_\_ 202\_\_

«Зачтено» «\_\_\_\_» \_\_\_\_\_ 202\_\_

Москва 2021

# Содержание

<b>1</b>	<b>Построение модели данных</b>	<b>3</b>
1.1	Работа по методическим указаниям . . . . .	3
1.2	Индивидуальный вариант . . . . .	5
<b>2</b>	<b>Создание логической и физической модели данных</b>	<b>7</b>
2.1	Работа по методическим указаниям . . . . .	7
<b>3</b>	<b>Индивидуальный вариант</b>	<b>8</b>
<b>4</b>	<b>Создание базы данных</b>	<b>11</b>
4.1	Работа по методическим указаниям . . . . .	11

# 1 Построение модели данных

## 1.1 Работа по методическим указаниям

### Описание предметной области

Компания занимается производством и продажей небольших статуэток, раскрашиваемых вручную. Компания имеет несколько производственных направлений. Миниатюры изготавливаются из гипса, олова или алюминия.

Компания распространяет свои товары по трем каналам. Компания содержит пять собственных розничных магазинов. Помимо этого, компания владеет сайтом, на котором ведется online-торговля, и осуществляет оптовые поставки сторонним дистрибьюторам. Для анализа статистики, системой автоматизации производства, нужен интерактивный аналитический инструмент. Поэтому необходимо спроектировать и построить модель данных, которая станет хранилищем информации по производству.

В ходе производства изделий система автоматизации производства управляет всеми станками компании. Каждый станок реализует полный цикл производства изделий, включая:

- заполнение формы сырьем (гипсом, оловом или алюминием);
- затвердевание материала;
- удаление изделия из формы после затвердевания;
- при необходимости автоматизированная раскраска изделий (оловянные фигурки не раскрашиваются);
- сушку после покраски (при необходимости).

Покраска и сушка могут производиться за несколько этапов в зависимости от сложности изделия. По мере готовности изделия проходят проверку, выполняемую оператором станка.

Оператор станка регистрируется в системе. В ходе этого процесса оператор сообщает системе автоматизации производства тип производимых изделий и объем загруженного в машину сырья. Оператор также делает в системе запись при отбраковке изделий.

В ходе интервью необходимые для эффективного анализа статистики:

- число принятых изделий по объему сырья, видам изделий, машинам и
- время формовки и затвердевания по видам изделий, машинам и дням;

- покупки машины. Анализ файла-экспорта из системы автоматизации производства показал, что для каждого вида производимых изделий есть отдельная строка, в которой присутствует следующая информация:

- тип изделия;
- объем сырья;
- номер машины;
- личный номер оператора;
- время и дата начала производства (когда серия начата);
- время и дата окончания производства (когда серия закончена);
- флаг отбраковки.

По приведенному описанию предметной области построим ее модель в среде ErAssistant. Укажем линии связей, назначим им имена, укажем типы и кратность связей. В результате работы, модель примет вид, приведенный на Рисунке 1.

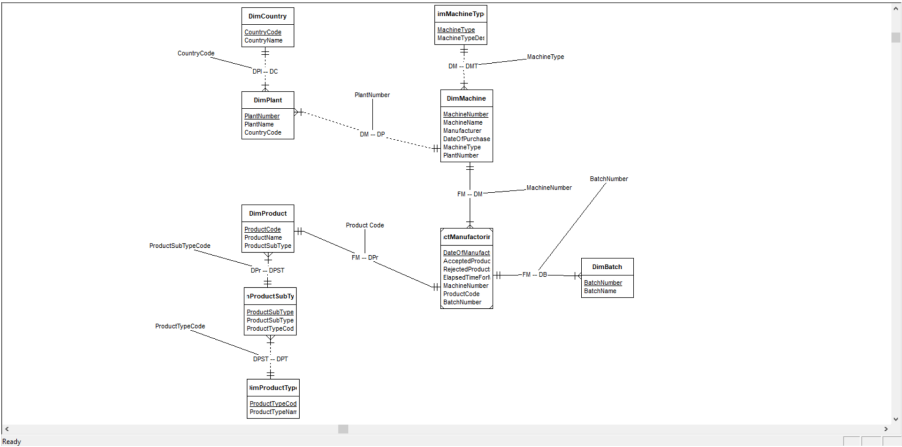


Рисунок 1 – Модель данных Производства со связями

## 1.2 Индивидуальный вариант

### Описание предметной области

В качестве индивидуального задания была выбрана реализация модели информационной системы по хранению и анализу данных предприятия, занимающегося сборкой и поставкой спортивных велосипедов для конечных потребителей по индивидуальному заказу.

Предприятие располагает широким выбором компонентов и комплектующих для сборки велосипедов следующих типов:

- Дорожный;
- Горный;
- Кросс-кантри;
- Эндуро;
- Прогулочный.

Предприятие может предложить сконфигурировать сборку велосипеда, отдельно выбрав каждый из предложенных компонентов:

- Рама;
- Вилка;
- Руль;
- Трансмиссия;
- Колеса;
- Тормозная система.

Контроль над выполнением работ по сборке велосипеда проводится в виде учета всех операций по сборке, настройке и тестированию, проводимых на территории предприятия ее сотрудниками. При этом в каждой записи содержится следующая информация о проведенных работах:

1. Внутренний номер изделия;
2. Время;
3. Этап работ;
4. Название цеха;
5. Имя мастера;
6. Статус;
7. Примечание.

На предприятии ведется учет всех компонентов велосипедов. В базе дан-

ных предприятия хранится информация о каждом компоненте, приобретенном у партнеров или изготовленном самостоятельно. В независимости от типа компонента он обладает общей информацией о наименовании производителя, месте и времени изготовления, типе и рекомендованной розничной цене.

Также каждый компонент имеет особые сведения, присущие данному типу детали.

## Построение модели

После приведения общих сведений о роде деятельности предприятия, факторизируем модель данных информационной системы предприятия в среде ErAssistant. См. Рисунок 2.

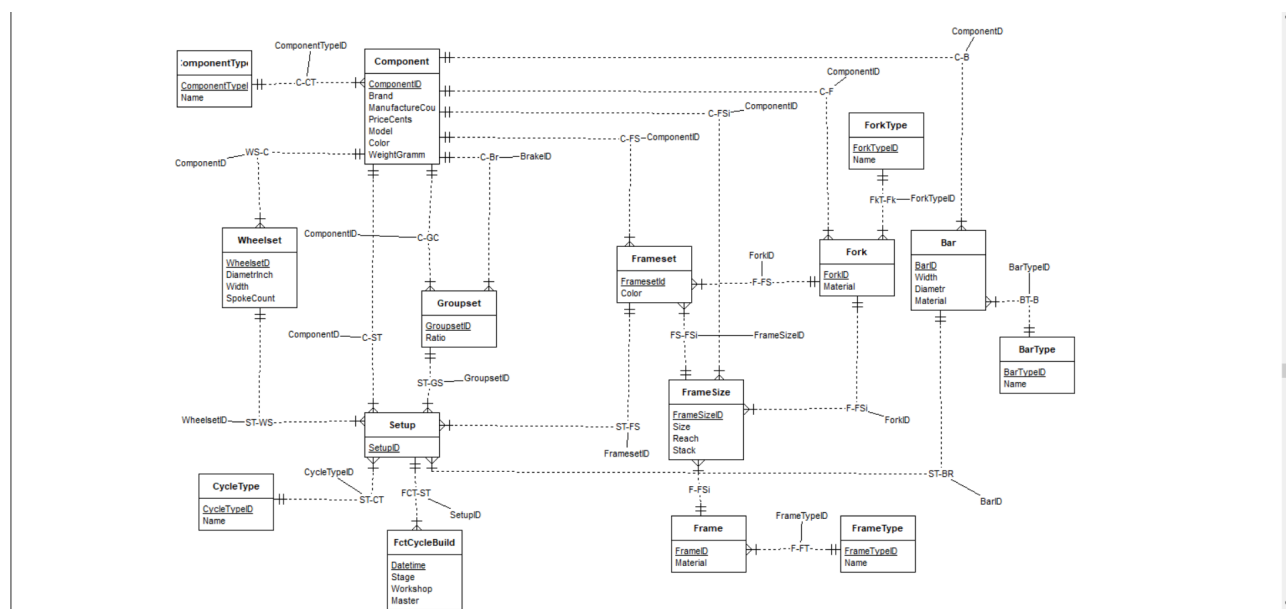


Рисунок 2 — Модель данных вело-предприятия

Укажем названия связей, их идентификаторы и кратность, исходя из вида отношений, выстроенных между сущностями.

На данном этапе выполнения работы мы реализовали модель данных будущего хранилища данных, планируемого к применению в компании, занимающейся сборкой и поставкой велосипедов по индивидуальному заказу.

## 2 Создание логической и физической модели данных

### 2.1 Работа по методическим указаниям

#### Теоретический материал

На этапе инфологического проектирования базы данных должна быть построена модель предметной области, не привязанная к конкретной СУБД, понятная не только разработчикам информационной системы, но и экономистам, менеджерам и другим специалистам. В то же время модель предметной области должна максимально точно отражать семантику предметной области и позволять легко перейти к модели данных конкретной СУБД.

Таковыми моделями являются модели «сущность-связь». Известно несколько методологий построения моделей «сущность-связь». Наибольшее распространение получила методология IDEF1X. Рассмотрим построение моделей "сущность-связь ориентируясь на продукт СА ERwin Data Modeler. ERwin имеет два уровня представления модели:

**Логический уровень** — это уровень, соответствующий инфологическому этапу проектирования и не привязанный к конкретной СУБД. Модели логического уровня оперируют с понятиями сущностей, атрибутов и связей, которые на этом уровне именуются на естественном языке (в нашем случае — на русском) так, как они называются в реальном мире.

**Физический уровень** — это отображение логической модели на модель данных конкретной СУБД. Одной логической модели может соответствовать несколько физических моделей. Причем, Erwin (как и другие CASE-системы проектирования баз данных) позволяет автоматизировать отображение логической модели на физическую.

Модель «сущность-связь» строится в виде диаграммы «сущность-связь», основными компонентами которой являются сущности (Entity) и связи (Relationship). Отсюда происходят часто используемые названия модели (ER-модель) и диаграммы (ER-диаграмма).

В ходе данной практической работы нами была описана и построена логическая и физическая модель данных средствами ПО ERwin Data Modeler.

Порядок построения модели данных в среде ERwin рассмотрим на приме-

ре автоматизированной информационной системы "Реализация средств вычислительной техники предназначенной для учета продаж настольных компьютеров по заказам клиентов.

Создание модели данных начинается с разработки логической модели, которая должна представлять состав сущностей предметной области с перечнем атрибутов и отношений между ними.

Результат разработки логической модели данных системы «Реализация средств вычислительной техники», предназначенной для учета продаж настольных компьютеров по заказам клиентов приведен на Рисунке 3.

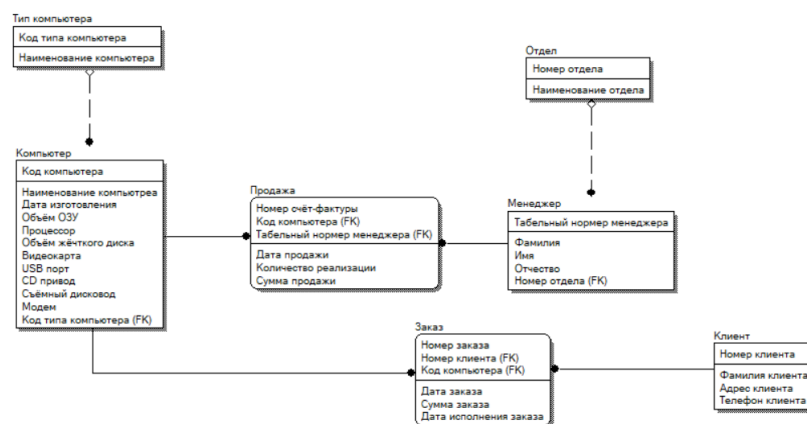


Рисунок 3 — Логическая модель данных системы «Реализация средств вычислительной техники»

Для построения физической модели данных системы, следует определиться с СУБД, в которой будет реализована модель. При построении физической модели данных следует учитывать формальную теория представления и обработки данных в конкретной системе управления базами данных (СУБД).

В данной практической работе в качестве СУБД выбрана MySQL.

Приступим к построению физической модели данных системы «Реализация средств вычислительной техники». Результат работы можно видеть на рисунке 4.

### 3 Индивидуальный вариант

Приступим к построению логической модели данных системы «Велосипедное предприятие». В соответствии с моделью, реализованной в ходе первой практической работы, добавим в рабочую область следующие сущности:



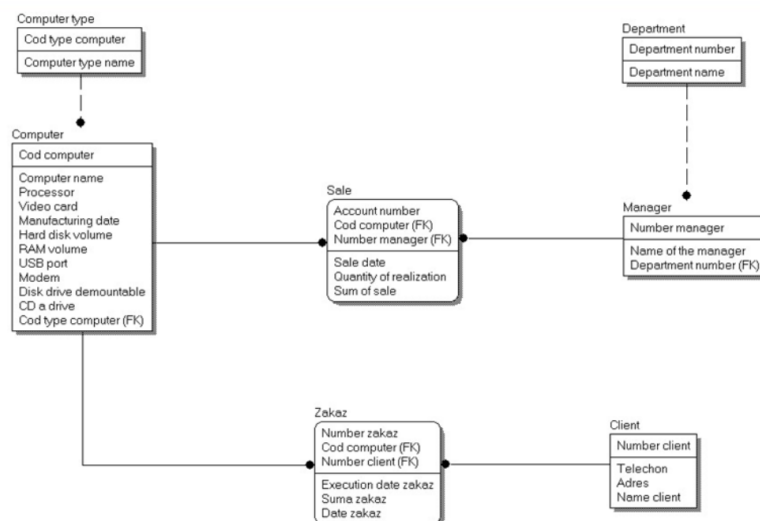


Рисунок 4 — Физическая модель данных системы «Реализация средств вычислительной техники»

- Component;
- FrameInfo;
- Frame;
- Frameset;
- FrameSize;
- Fork;
- ComponentType;
- Wheelset;
- Groupset;
- Brake;
- FctCycleBuild;
- CycleType;
- Bar;
- Setup.

Добавим связи между сущностями в соответствии с ранее построенной моделью. Логическая модель системы «Велосипедное предприятие» приведена на рисунке 5.

После уточнения типов данных, выбранных в соответствии с предметной областью и спецификой СУБД MySQL, построим физическую модель данных (см. рисунок 6).

После реализации физической и логической модели можно приступать к реализации модели данной системы в СУБД MySQL.

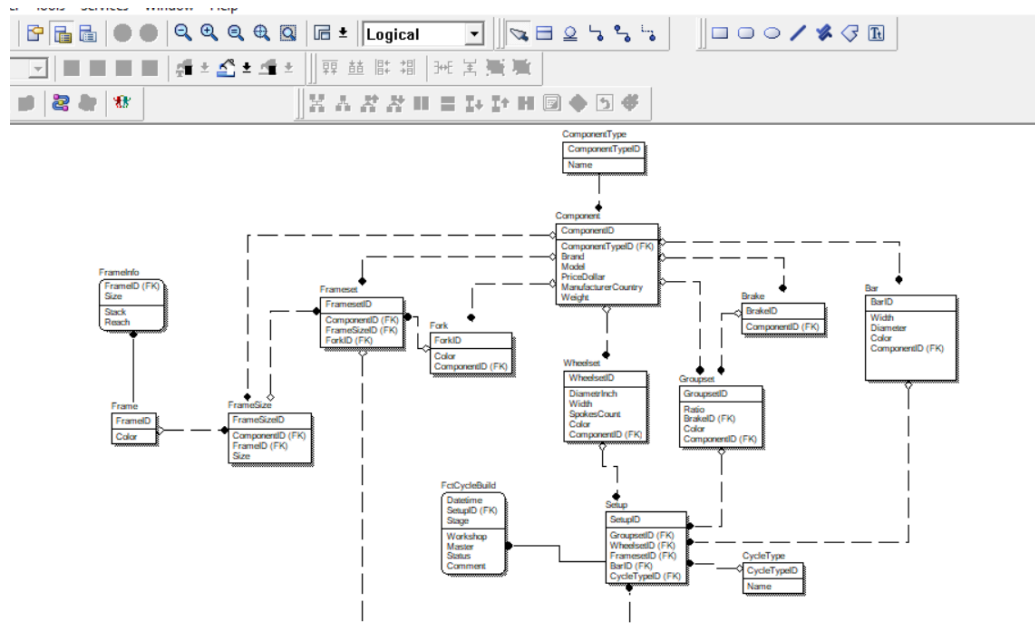


Рисунок 5 — Логическая модель данных системы «Велосипедное предприятие»

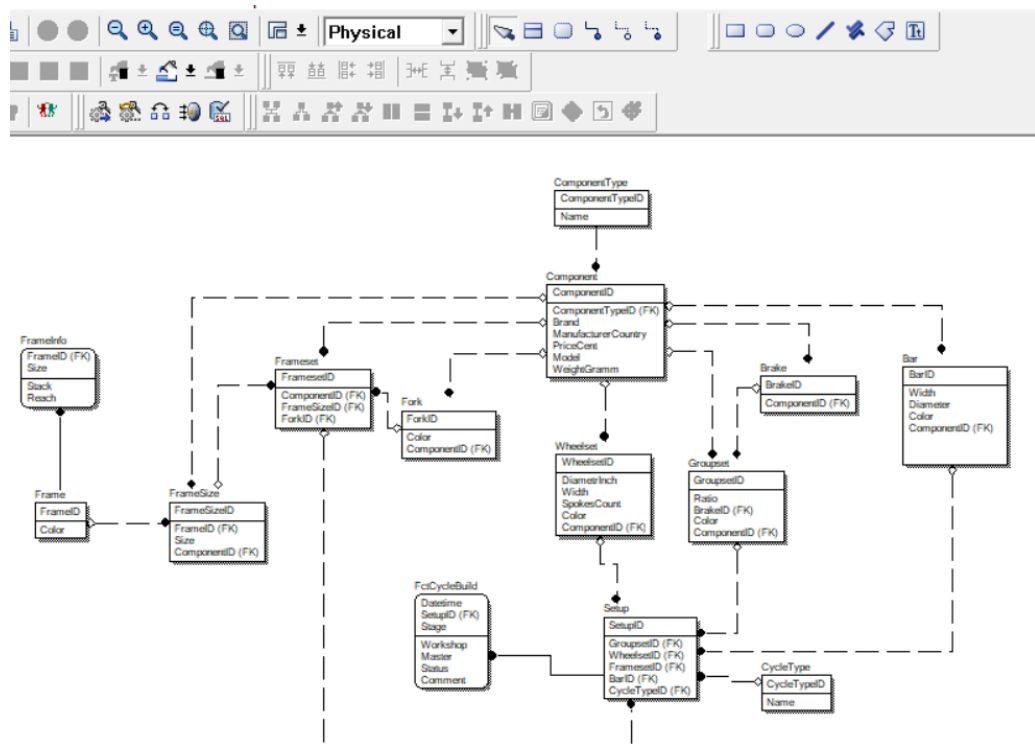


Рисунок 6 — Физическая модель данных системы «Велосипедное предприятие»

## 4 Создание базы данных

### 4.1 Работа по методическим указаниям

В ходе данной практической работы была создана база данных `forum`, которая хранит в себе сведения о пользователях форума и размещенных ими темах.

Помимо суперпользователя `root`, был создан пользователь `denilai`, под которым производятся все манипуляции с данными.

Создадим базу данных `forum` с помощью команды `create database forum`;

Создадим таблицу `users` с помощью данных команд (см. рисунок 7):

```
mysql> create table users (  
-> id_user int (10) AUTO_INCREMENT,  
-> name varchar (20) NOT NULL,  
-> email varchar (50) NOT NULL,  
-> password varchar (15) NOT NULL,  
-> PRIMARY KEY (id_user));  
Query OK, 0 rows affected (0.16 sec)
```

Рисунок 7

Создадим таблицу `topics` с помощью данных команд (см. рисунок 8):

```
mysql> create table topics (  
-> id_topic int (10) AUTO_INCREMENT,  
-> topic_name varchar(100) NOT NULL,  
-> id_author int(10) NOT NULL,  
-> PRIMARY KEY (id_topic),  
-> FOREIGN KEY (id_author) REFERENCES users (id_user));  
Query OK, 0 rows affected (0.13 sec)
```

Рисунок 8

Создадим таблицу `posts` с помощью данных команд (см. рисунок 9):

```
mysql> create table posts (  
-> id_post int (10) AUTO_INCREMENT,  
-> message text NOT NULL,  
-> id_author int (10) NOT NULL,  
-> id_topic int (10) NOT NULL,  
-> primary key (id_post),  
-> foreign key (id_author) references users (id_user),  
-> foreign key (id_topic) references topics (id_topic));  
Query OK, 0 rows affected (0.17 sec)
```

Рисунок 9