МИНОБРНАУКИ РОССИИ

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Российский государственный университет им.А.Н.Косыгина**

**(ТЕХНОЛОГИИ, ДИЗАЙН, ИСКУССТВО)»**

|  |
| --- |
| **Институт**\_\_мехатроники и информационных технологий  **Кафедра**\_\_\_автоматизированных систем обработки информации и управления\_\_ |
|  |

**ОТЧЕТ**

**о**  **производственной**  **практике**

**Уровень освоения основной**

**профессиональной**

**образовательной программы (ОПОП)**  академический бакалавриат

**Направление подготовки (специальность) 09.03.01 Информатика и**

**вычислительная техника**

**Профиль (специализация) Автоматизированные системы обработки информации и управления**

**Форма обучения** **очная**

**Способ прохождения практики**  **стационарная**

**Форма проведения практики**  **непрерывная**

**Сроки прохождения практики с «07» февраля 2022 г. по «18» июня 2022 г.**

**Место прохождения практики \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Отчет составил и сдал:** «\_23\_»\_мая\_2022 г.\_\_\_\_\_Лозбенев В. В.\_**\_\_\_\_\_\_\_**\_\_\_\_\_\_

*(фамилия, инициалы)*

группа \_МВА-119\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(*подпись практиканта)*

**Руководитель практики от университета** \_\_\_\_\_\_\_Монахов В. И., зав.каф.\_\_\_\_\_\_\_\_\_\_\_\_

*(фамилия, инициалы, должность)*

**ОЦЕНКА работы на практике** \_\_\_\_\_\_\_\_\_\_\_\_ «\_\_»\_*\_*2022\_ г.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

*(подпись руководителя практики от университета)*

**Москва, 2022 г.**

# Содержание

[Введение 3](#_Toc105861900)

[1. Описание объекта автоматизации и постановка задачи проектирования 3](#_Toc105861901)

[2. Выбор СУБД 4](#_Toc105861902)

[3. Проектирование БД 5](#_Toc105861903)

[3.1 Создание логической модели 5](#_Toc105861904)

[3.2 Создание физической модели 9](#_Toc105861905)

[3.3.Создание БД 11](#_Toc105861906)

[4. Создание представлений 16](#_Toc105861907)

[5. Создание триггеров 17](#_Toc105861908)

[Заключение 22](#_Toc105861909)

[Список использованных источников и литературы 24](#_Toc105861910)

[Приложения 25](#_Toc105861911)

[Приложение 1. Текст скрипт-файла схемы БД 25](#_Toc105861912)

[Приложение 2. Файлы для загрузки данных 29](#_Toc105861913)

[Приложение 3. Команды загрузки данных 32](#_Toc105861914)

[Приложение 4. Тексты представлений 34](#_Toc105861915)

[Приложение 5. Тексты триггеров 34](#_Toc105861916)

# Введение

В настоящее время быстро развивающиеся новые информационные технологии (ИТ) влекут за собой коренные изменения не только в производственных отраслях, но и в сфере познавательной деятельности, в частности в образовании. Во всем мире компьютер используется не только как предмет изучения, но и как средство обучения. Как показывает практика, из всех существующих технических средств обучения компьютер наиболее полно удовлетворяет дидактическим требованиям, так как обладает целым рядом дополнительных возможностей, позволяющих управлять процессом обучения, максимально адаптировать его к особенностям обучаемого.

Информационные системы – один из самых необходимых элементов знания, который требует умения работать с большими объемами информации в современном мире. Базы данных занимают основное место в информационных системах, так как позволяют оперировать большим количеством данных, используя при этом сравнительно низкие вычислительные мощности, и являются необходимым средством сохранения данных между последовательными запусками системы. СУБД) — специализированных программных средств, предназначенных для организации и ведения баз данных.

Реляционные БД [1] (РБД) — наиболее распространенный тип БД. Реляционная база данных представляет собой множество взаимосвязанных таблиц, каждая из которых содержит информацию об объектах определенного типа. Каждая строка таблицы включает данные об одном объекте (например, клиенте, автомобиле, документе), а столбцы таблицы содержат различные характеристики этих объектов – атрибуты (например, наименования и адреса клиентов, марки и цены автомобилей). Строки таблицы называются записями; все записи имеют одинаковую структуру – они состоят из полей, в которых хранятся атрибуты объекта.

Довольно часто под базой данных понимают множество данных, которые управляются специальной программной системой. Такие программные системы называются системами управления базами данных (СУБД).

В ходе практики были спроектированы логическая и физическая модели данных по предметной области, создана и наполнена значения база данных. Предусмотрен механизм автоматического заполнения значений колонок идентификаторов.

# 1. Описание объекта автоматизации и постановка задачи проектирования

Предметной областью является ведение учета офисной техники и расходных материалов.

Задача состоит в организации учета офисной техники и расходных материалов. Каждое оборудование и расходный материал имеет распределение по отделу и может списываться на склад сотрудником и поставляться контрагентом. Каждый расходный материал и оборудование имеет свою цену. По каждому сотруднику имеется карточка: ФИО, должность, название отдела. По каждому агенту имеются данные (название поставщика, контакты). Изменения в учетах оборудования и расходных материалов учитываются в книгах учета оборудования и расходных материалов, содержащие: номер документа, дату, название отдела, сотрудника, контрагента, код операции и стоимость.

Требования к разрабатываемой информационной системе: База данных должна содержать информацию о технике и расходных материалов, сотрудников, агентов, складских остатков. База данных должна не позволять вводить некорректные значения [2].

Данные должны удовлетворять основным требованиям предметной области.

# 2. Выбор СУБД

СУБД – специализированный комплекс программ, предназначенный для удобной и эффективной организации, контроля и администрирования баз данных [3].

Основные действия, которые пользователь может выполнять с помощью СУБД:

- создание структуры БД;

- заполнение БД информацией;

- изменение (редактирование) структуры и содержание БД:

- поиск информации в БД;

- сортировка данных;

- защита БД;

- проверка целостности БД.

Существуют различные СУБД.

MySQL, как и любая другая СУБД представляет собой программу-сервер, которая находится в памяти компьютера и обслуживает TCP порт. В случае с MySQL, номером порта будет являться число 3306. А клиентская программа, будь то CGI-приложение на Perl либо программный продукт на C, соединяется с СУБД по этому порту и посылает ему строчки на SQL. Тот в свою очередь их интерпретирует, выполняя необходимые действия, и отсылает результаты запроса обратно клиенту. Таким способом происходит общение сервера баз данных с клиентскими программами [4].

MуSQL имеет развитую систему доступа к базам данных. Пользователю базы данных может быть предоставлен доступ ко всей базе данных, отдельным таблицам и отдельным столбцам таблиц. Имеется разграничение на действия, которые может производить пользователь с записями. Для организации такой сложной (на первый взгляд) структуры доступа используется несколько таблиц в специальной базе данных. На основании значений этих таблиц выстраивается политика предоставления доступа.

База данных, которую сервер MуSQL использует для хранения внутренней информации о пользователях, по умолчанию имеет имя mуsql.

В течение последних десяти лет СУБД MySQL действительно прошла большой путь развития, превратившись в программный продукт мирового класса. В наши дни MySQL успешно конкурирует даже с такими наиболее полнофункциональными коммерческими приложениями, предназначенными для поддержки баз данных.

# 3. Проектирование БД

## 3.1 Создание логической модели

Логическая модель описывает понятия предметной области, их взаимосвязь, а также ограничения на данные, налагаемые предметной областью. Логическая модель данных является начальным прототипом будущей базы данных. Логическая модель строится в терминах информационных единиц, но без привязки к конкретной СУБД (СУБД – Система управления базами данных). Логическая модель разработана в Erwin Data Modeler [5].

Логическая модель описывает понятия предметной области, их взаимосвязь, а также ограничения на данные, налагаемые предметной областью. Логическая модель данных является начальным прототипом будущей базы данных. Логическая модель строится в терминах информационных единиц, но без привязки к конкретной СУБД.

Схема логической модели базы данных представлена на рис. 1.

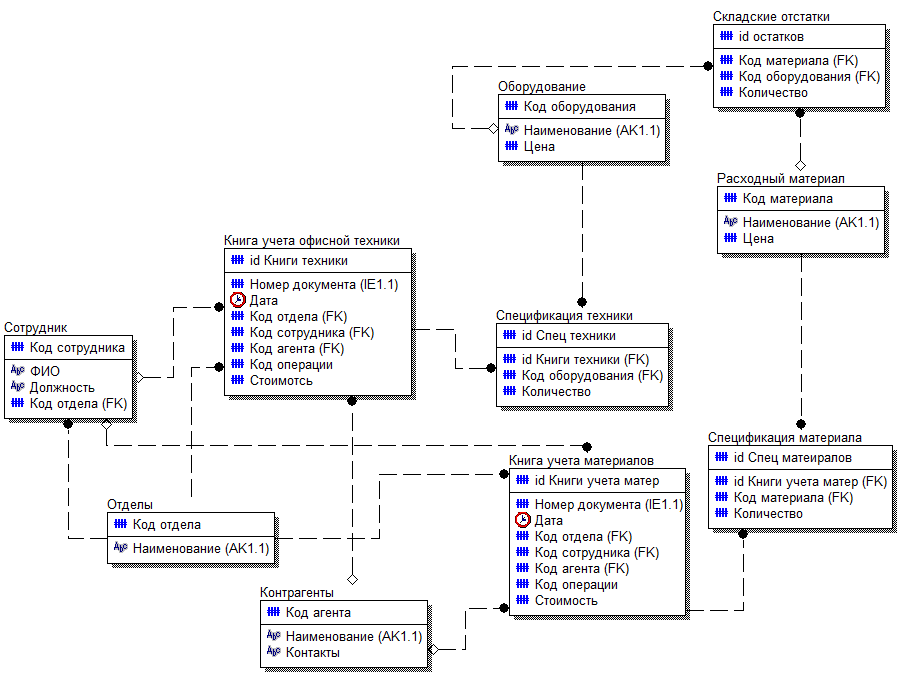


Рис. 1. Логическая модель базы данных

Перечень сущностей и их атрибутов представлен в таблицах 1-10.

Таблица 1 – Сотрудник

|  |  |  |
| --- | --- | --- |
| Название атрибута в информационной модели | Тип атрибута | Дополнительные сведения: PK, FK, поле ключа |
| Кот сотрудника | Number | PK |
| ФИО | String |  |
| Должность | String |  |
| Код отдела | Number | FK |

Таблица 2 – Контрагенты

|  |  |  |
| --- | --- | --- |
| Название атрибута в информационной модели | Тип атрибута | Дополнительные сведения: PK, FK, поле ключа |
| Код агента | Number | PK |
| Наименование | String |  |
| Контакты | String |  |

Таблица 3 – Отделы

|  |  |  |
| --- | --- | --- |
| Название атрибута в информационной модели | Тип атрибута | Дополнительные сведения: PK, FK, поле ключа |
| Код отдела | Number | PK |
| Наименование | String |  |

Таблица 4 – Книга учета офисной техники

|  |  |  |
| --- | --- | --- |
| Название атрибута в информационной модели | Тип атрибута | Дополнительные сведения: PK, FK, поле ключа |
| Id Книги техники | Number | PK |
| Номер документа | Number |  |
| Дата | Date |  |
| Код отдела | Number | FK |
| Код сотрудника | Number | FK |
| Код агента | Number | FK |
| Код операции | Number |  |
| Стоимость | Number |  |

Таблица 5 – Книга учета расходных материалов

|  |  |  |
| --- | --- | --- |
| Название атрибута в информационной модели | Тип атрибута | Дополнительные сведения: PK, FK, поле ключа |
| Id Книги учета материалов | Number | PK |
| Номер документа | Number |  |
| Дата | Date |  |
| Код отдела | Number | FK |
| Код сотрудника | Number | FK |
| Код агента | Number | FK |
| Код операции | Number |  |
| Стоимость | Number |  |

Таблица 6 – Спецификация техники

|  |  |  |
| --- | --- | --- |
| Название атрибута в информационной модели | Тип атрибута | Дополнительные сведения: PK, FK, поле ключа |
| Id Спецификации техники | Number | PK |
| Id Книги техники | Number | FK |
| Код оборудования | Number | FK |
| Количество | Number |  |

Таблица 7 – Спецификация расходных материалов

|  |  |  |
| --- | --- | --- |
| Название атрибута в информационной модели | Тип атрибута | Дополнительные сведения: PK, FK, поле ключа |
| Id спец материалов | Number | PK |
| Id книги учета материалов | Number | FK |
| Код материалов | Number | FK |

Продолжение таблицы 7

|  |  |  |
| --- | --- | --- |
| Количество | Number |  |

Таблица 8 – Оборудование

|  |  |  |
| --- | --- | --- |
| Название атрибута в информационной модели | Тип атрибута | Дополнительные сведения: PK, FK, поле ключа |
| Код оборудования | Number | PK |
| Наименование | String |  |
| Цена | Number |  |

Таблица 9 – Расходный материал

|  |  |  |
| --- | --- | --- |
| Название атрибута в информационной модели | Тип атрибута | Дополнительные сведения: PK, FK, поле ключа |
| Код материала | Number | PK |
| Наименование | String |  |
| Цена | Number |  |

Таблица 10 – Складские остатки

|  |  |  |
| --- | --- | --- |
| Название атрибута в информационной модели | Тип атрибута | Дополнительные сведения: PK, FK, поле ключа |
| Id остатков | Number | PK |
| Код материала | Number | FK |
| Код оборудования | Number | FK |
| Количество | Number |  |

Таблица 11 – Перечень дополнительных ключей

|  |  |  |  |
| --- | --- | --- | --- |
| Имя сущности | Имя ключа | Тип ключа (IE/AK) | Список атрибутов, входящих в ключ |
| Отдел | Наименование\_AK | АК | Наименование |
| Контрагенты | Наименование\_AK | AK | Наименование |
| Книга учета офисной техники | Номер\_документа\_IE | IE | Номер документа |
| Книга учета расходных материалов | Номер\_документа\_IE | IE | Номер документа |
| Оборудование | Наименование\_AK | AK | Наименование |
| Расходный материал | Наименование\_AK | AK | Наименование |

## 3.2 Создание физической модели

Физическая модель данных описывает реализацию объектов логической модели на уровне объектов конкретной базы данных. Физическая модель разрабатывалась в MySQL Workbench.

Построенная физическая модель приведена на рис. 2.

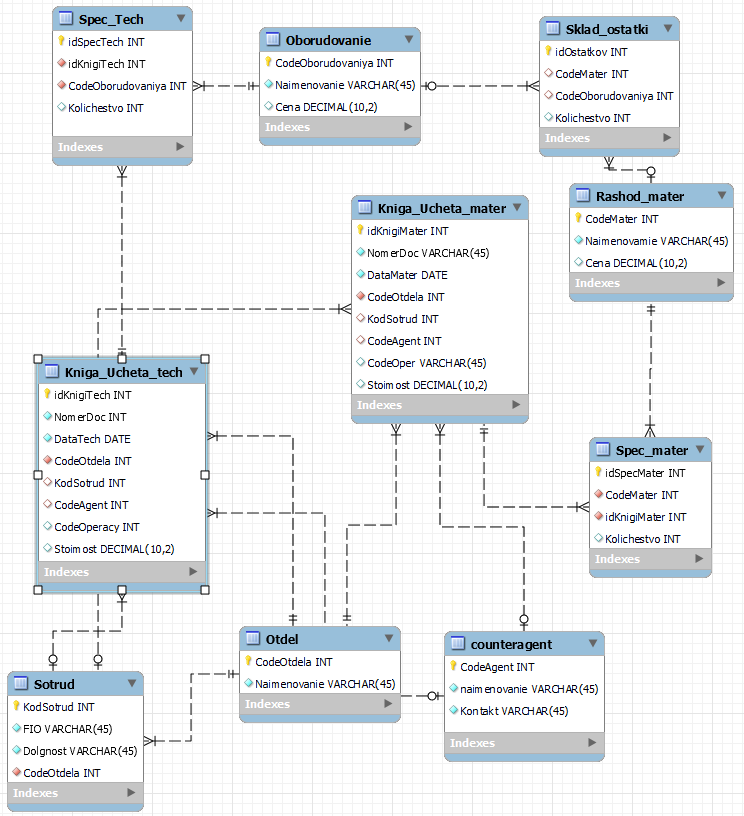


Рис. 2. Физическая модель базы данных

Перечень таблиц и их колонок представлен в таблице 9.

Таблица 9 – Перечень сущностей и их атрибутов модели БД

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя сущности** | **Имя атрибута** | **Тип** | **PK,FK,IE,AK** |
| Sotrud | KodSotrud | Integer | PK |
| FIO | Varchar(45) |  |
| Dolgnost | Varchar(45) |  |
| CodeOtdela | Integer | FK |
| Counteragent | CodAgent | Integer | PK |
| Naimenovanie | Varchar(45) | AK1.1 |
| Kontakt | Varchar(45) |  |
| Otdel | CodeOtdela | Integer | PK |
| Naimenovanie | Varchar(45) | AK1.1 |
| Kniga\_ucheta\_tech | IdKnigiTech | Integer | PK |
| NomerDoc | Integer | IE1.1 |
| DateTech | Date |  |
| CodeOtdela | Integer | FK |
| KodSotrud | Integer | FK |
| CodeAgenta | Integer | FK |
| CodeOperacy | Integer |  |
| Stoimost | Decimal(10,2) |  |
| Kniga\_ucheta\_mater | IdKnigiMater | Integer | PK |
| NomerDoc | Integer | IE1.1 |
| DateMater | Date |  |
| CodeOtdela | Integer | FK |
| KodSotrud | Integer | FK |
| CodeAgenta | Integer | FK |
| CodeOperacy | Integer |  |
| Stoimost | Decimal(10,2) |  |
| Spec\_tech | IdSpecTech | Integer | PK |
| IdKnigiTech | Integer | FK |
| CodeOborudovaniya | Integer | FK |
| Kolichestvo | Integer |  |
| Spec\_mater | IdSpecMater | Integer | PK |
| IdKnigiMater | Integer | FK |
| CodeMater | Integer | FK |
| Kolichestvo | Integer |  |
| Oborudovanie | CodeOborudovaniya | Integer | PK |
| Naimenovanie | Varchar(45) | AK |
| Cena | Decimal(10,2) |  |
| Rashod\_mater | CodeMater | Integer | PK |
| Naimenovanie | Varchar(45) | AK |
| Cena | Decimal(10,2) |  |
| Sklad\_ostatki | idOstatkov | Integer | PK |
| CodeMater | Integer | FK |
| CodeOborudovaniya | Integer | FK |
| Kolichestvo | Integer |  |

## 3.3.Создание БД

Для производственной практики была выбрана СУБД MySQL, так как в ней просто и понятно создаются представления и добавляются данные.

После создания физической модели данных выполняется проектирование БД [6]. Для этого используется MySQL Connection — средство администрирования баз данных.

Сначала создадим подключение (рис. 3).

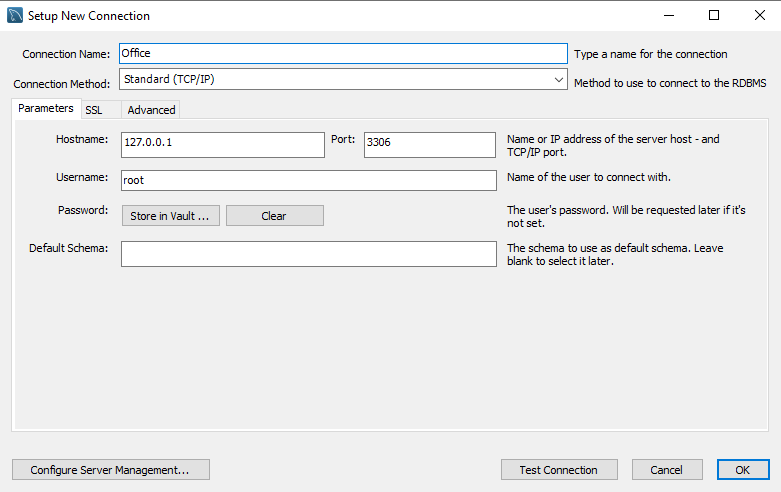


Рис. 3. Создание БД

По умолчанию владельцем базы данных является пользователь root со всеми правами доступа. Был создан новый пользователь Admin с правами администратора. Для этого в браузере администрации был выделен пункт «Добавления аккаунта». В открывшемся окне задается имя аккаунта и его пароль (рис. 4). В перечне привилегий галочка установлена на все права пользователя (рис. 5).

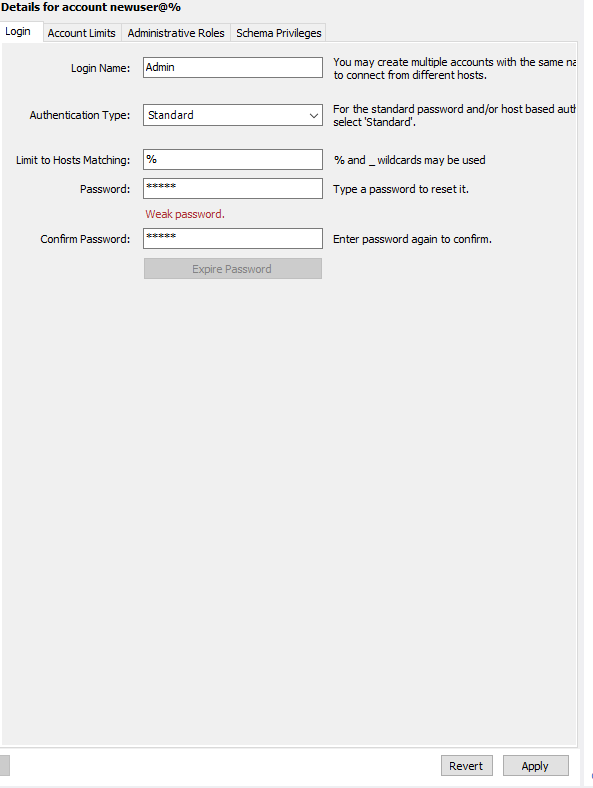


Рис. 4. Создание пользователя

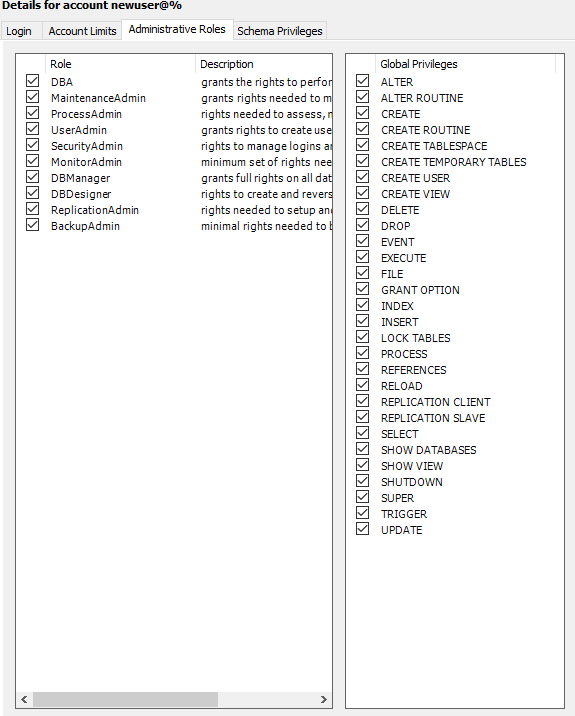


Рис. 5. Создание пользователя

Далее был создан сервер. Для этого в пункте меню «Файл» было выбрано «Создать соединение», указано название подключения, указан адрес БД (127.0.0.1), порт подключения (3306) и пользователь (рис. 6).

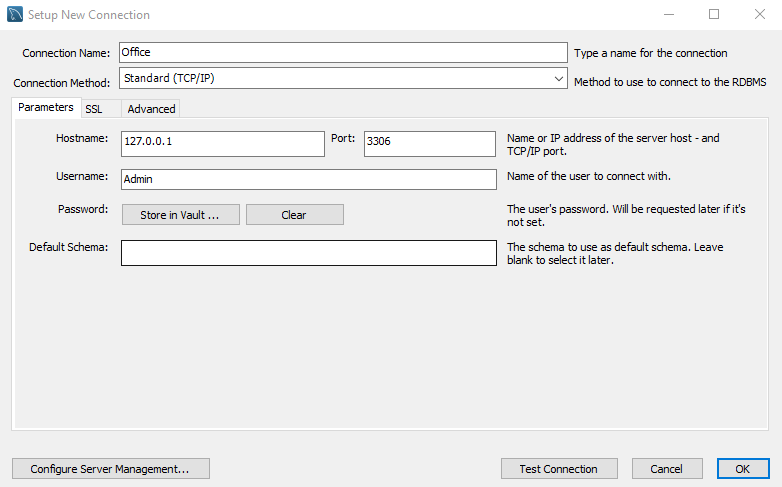


Рис. 6. Создание подключения

После создание подключение был выполнен вход в него через права администратора. Далее в MySQL Workbench был выгружен скрипт файл создания БД на основе диаграммы и выполнен в командной строке SQL (рис. 7).

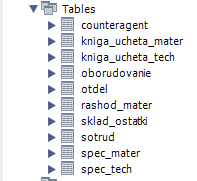


Рис. 7. Выгрузка таблиц

В окно SQL были написаны команды загрузки данных (приложение 3). Данные были успешно загружены (рис. 8).

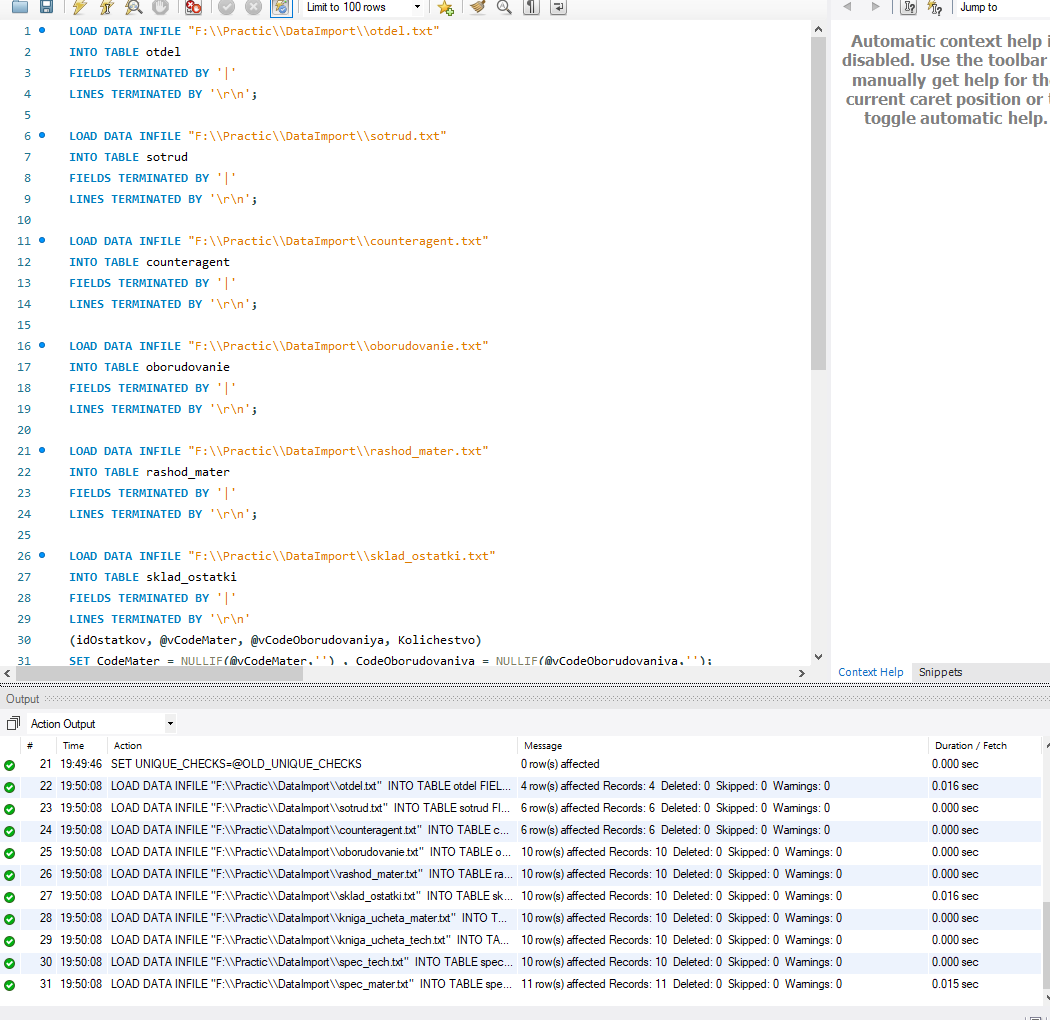


Рис. 8. Вывод информации о выгрузки данных

Созданные таблицы приведены в приложении 3.

Для проверки правильности загрузки расчетных полей таблицы составляется оператор запроса SELECT. Результат работы оператора с выборкой представлен на рис. 9. Критерием правильности загрузки является совпадение значений ячеек «Stoimost» и «StoimostNEW».

select km.idKnigiMater, km.Stoimost,

(select sum(rm.Cena\*sm.Kolichestvo) from Spec\_mater as sm

where rm.codemater = sm.CodeMater ) as StoimostNEW,

(Select count(\*) from spec\_mater as sm where sm.idKnigiMater = km.idKnigiMater) as Count

From Kniga\_Ucheta\_mater as km, Rashod\_mater as rm, Spec\_mater as sm

where sm.idKnigiMater = km.idKnigiMater and rm.codemater = sm.CodeMater

group by km.idKnigiMater

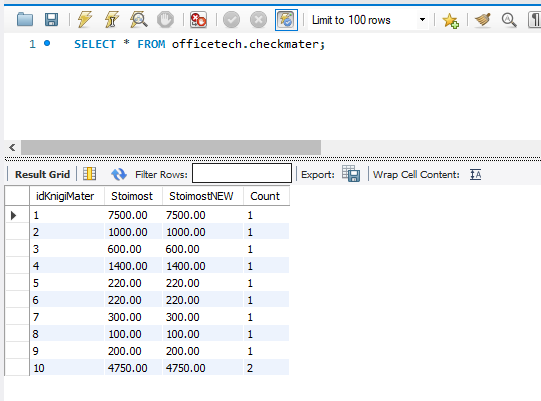


Рис. 9. Результаты проверки загрузки вычисляемого поля

# 4. Создание представлений

Представление для просмотра данных основной таблицы рассмотрено на примере учетной книге техники с добавлением данных из справочника техники, результат представлено на рис. 10. Текст представления приведен в приложении 4.

SELECT \* FROM officetech.uchet\_tech

where nomerdoc =211 and Datatech = "2021-06-26";

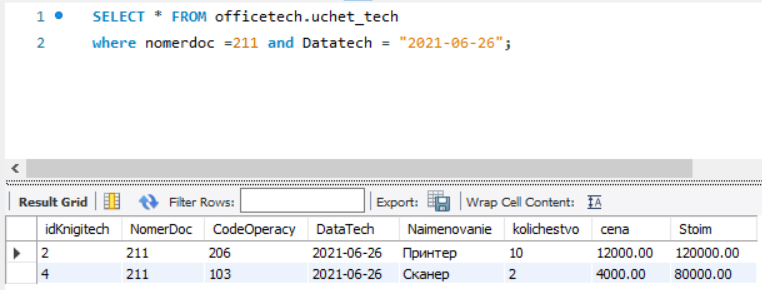


Рис. 10. Результат работы представления 1

Результат представления, показывающего информацию о количестве поставок и перевозок за определенное время с подсчетом количества и стоимости группировкой по двум полям представлено на рис. 11. Текст представления приведен в приложении 4.

SELECT \* FROM officetech.slog\_view;

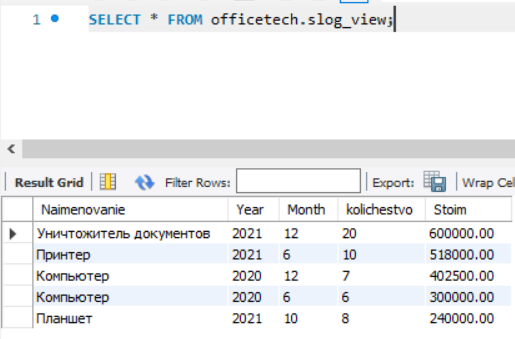


Рис. 11. Результат работы представления 3

# 5. Создание триггеров

Триггер - хранимая процедура особого типа, которую пользователь не вызывает непосредственно, а исполнение которой обусловлено действием по модификации данных: добавлением (INSERT), удалением (DELETE) строки в заданной таблице, или изменением (UPDATE) данных в определённом столбце заданной таблицы реляционной базы данных. Триггер запускается сервером автоматически при попытке изменения данных в таблице, с которой он связан. Все производимые им модификации данных рассматриваются как выполняемые в [транзакции](https://dic.academic.ru/dic.nsf/ruwiki/12167), в которой выполнено действие, вызвавшее срабатывание триггера. Соответственно, в случае обнаружения ошибки или нарушения целостности данных может произойти откат этой транзакции.

Триггер для заполнения полей перед фиксацией данных в таблице представлен на рис.12. Текст создания триггера приведен в приложении 5.

Выполним команду.

Insert into Kniga\_ucheta\_tech (idKnigiTech,KodSotrud,CodeOtdela,CodeOperacy)

VALUES (12,2,1,219);

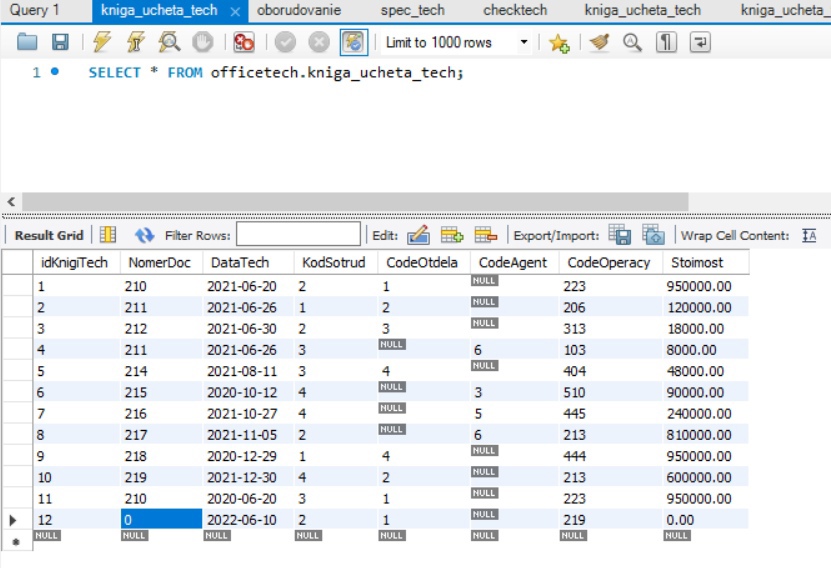


Рис. 12. Результат работы триггера

Триггер для контроля вводимых данных. При ошибочной попытке ввода операции в книге вывод информации об ошибке. Текст создания триггера и операторов SQL для проверки работы представлены в приложении 5.

Результат представлен на рис. 13.

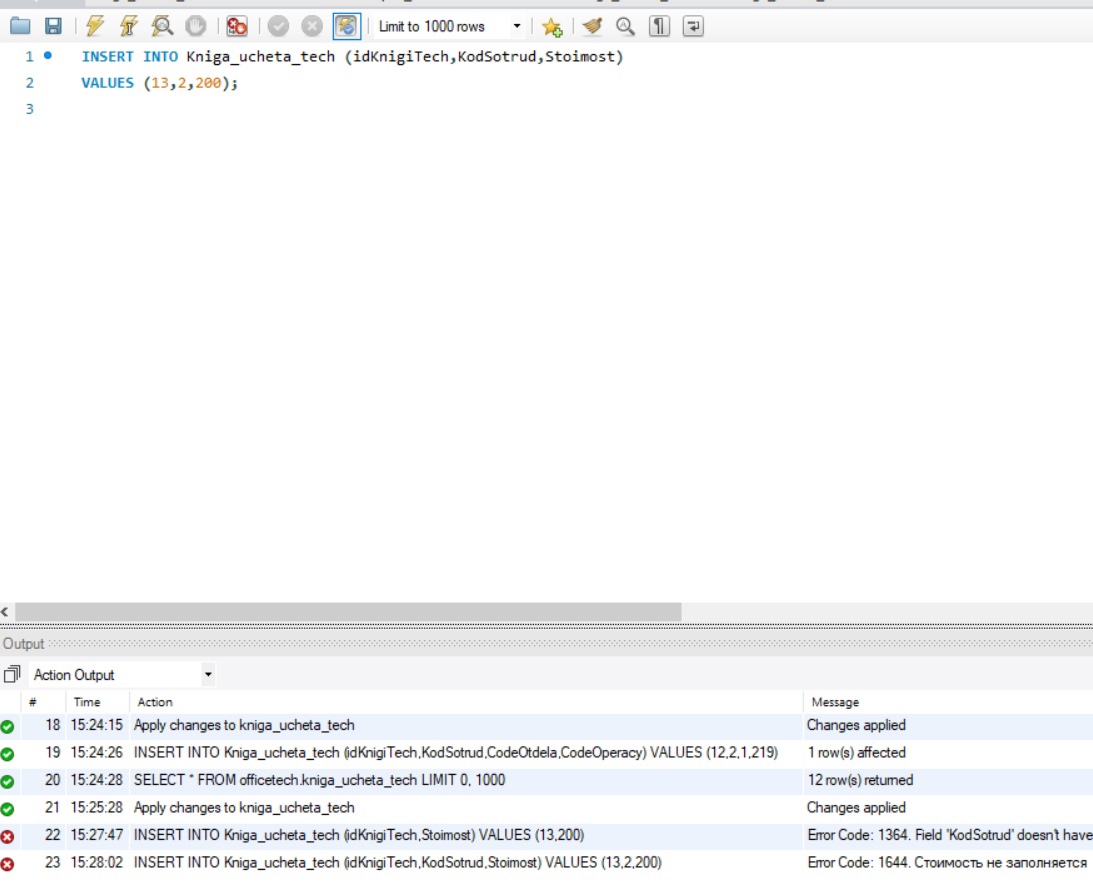


Рис. 13. Результат работы триггера контроля

Триггер для контроля событий представлен на рис. 14. Текст создания триггера и операторов SQL для проверки работы представлены в приложении 5.

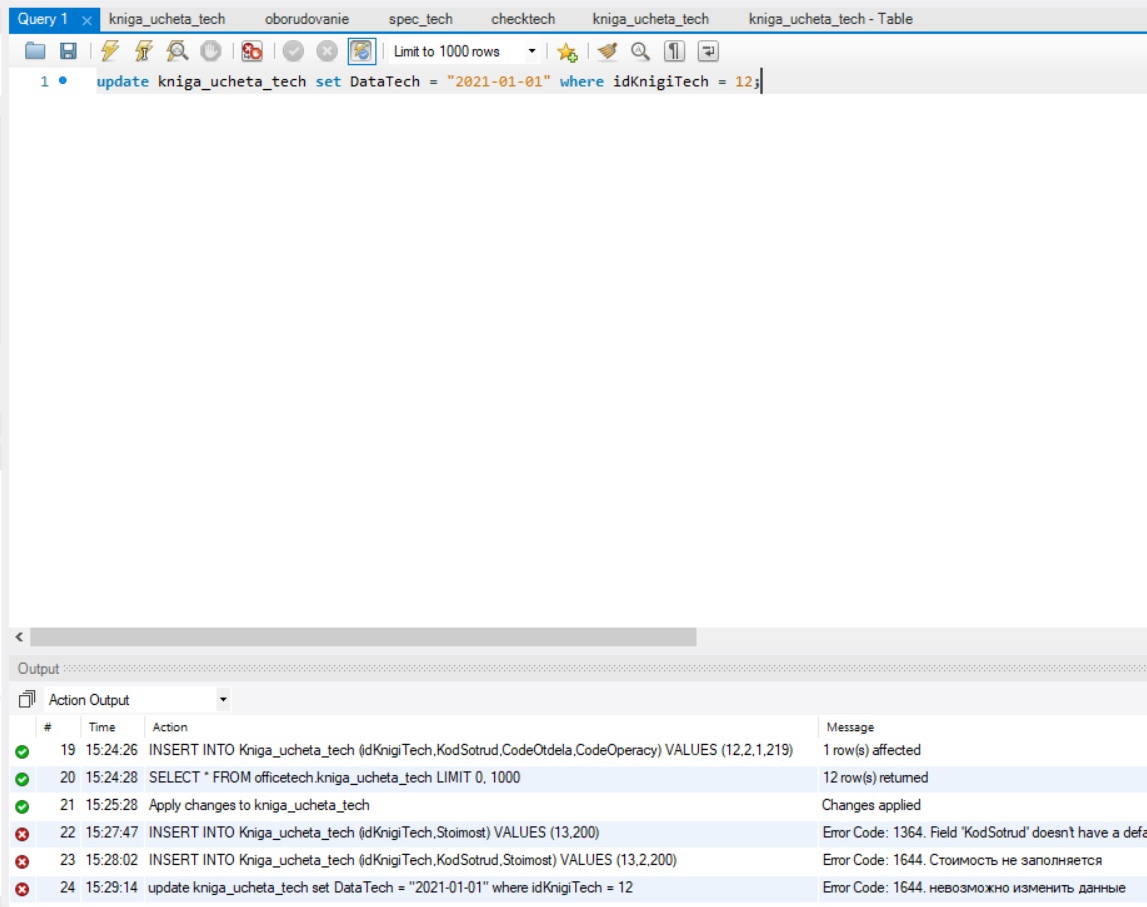


Рис. 14. Результат работы триггера контроля

Триггеры для изменения (перерасчета) вычисляемых полей в связанных таблицах представлен на рис. 15. Текст создания триггера и операторов SQL для проверки работы представлены в приложении 5.

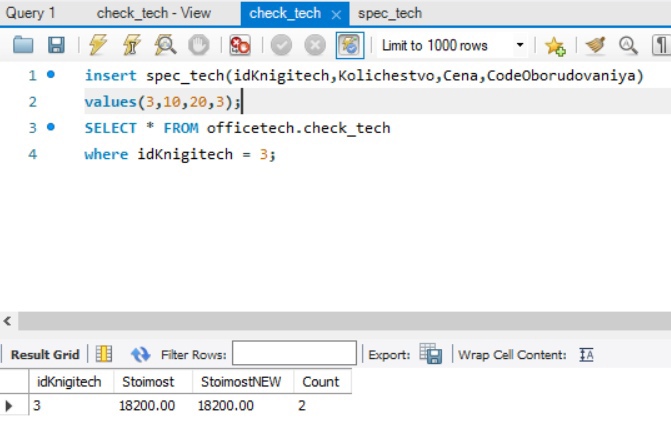


Рис. 15. Результат тестирования триггера на добавление записей

Триггер на обновление записей представлен на рис. 16. Текст создания триггера и операторов SQL для проверки работы представлены в приложении 5.

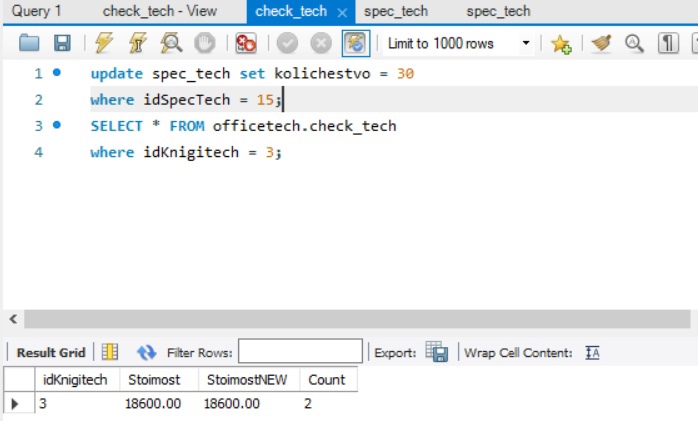


Рис. 16. Результат тестирования триггера на изменение записей

Триггер на удаление записей представлен на рис. 17. Текст создания триггера и операторов SQL для проверки работы представлены в приложении 5.

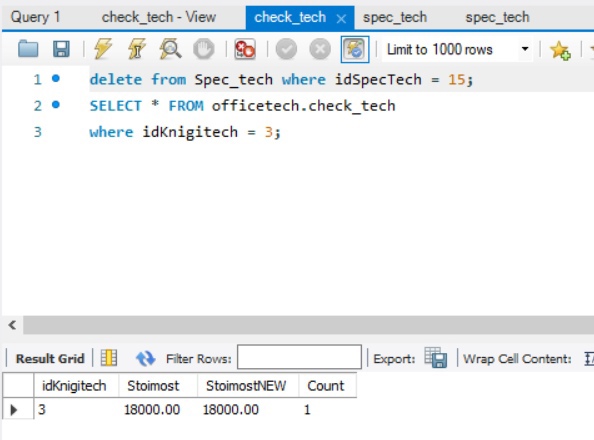


Рис. 17. Результат тестирования триггера на удаление записей

# Заключение

В рамках практики были получены навыки проектирования реальных баз данных. С развитием новых экономических отношений возросла потребность в аналитической работе. Возникает необходимость в накоплении фактов, опыта, знаний в каждой конкретной отрасли экономики и в управленческой деятельности. На первый план выдвигается заинтересованность в тщательном исследовании конкретных экономических, коммерческих, производственных ситуаций с целью принятия в оперативном порядке экономически обоснованных и наиболее приемлемых решений.

На основе БД сформировался новый пласт информационных технологий, которые эффективно используются во многих областях деятельности человека. Организации также нуждаются в специально разработанных БД.

В результате прохождения практики были выполнены поставленные цели. Изучен объект автоматизации и предметная область проектирования. Разработано формализованное описание предметной области.

Исходя из предметной области поставленной задачи, была разработана база данных для автоматизации учета расходных материалов и офисного оборудования. БД создана в программе MySQL.

# Список использованных источников и литературы

1. Что такое база данных. Информационный источник (<https://www.oracle.com/cis/database/what-is-database/>) Дата обращения: 15.05.22
2. Тестирование базы данных – Краткое руководство. Информационный источник (https://coderlessons.com/tutorials/kachestvo-programmnogo-obespecheniia/izuchite-testirovanie-bazy-dannykh/testirovanie-bazy-dannykh-kratkoe-rukovodstvo?) Дата обращения: 15.05.22
3. Администрирование баз данных Информационный источник (https://zhidba.ru/index.php/administrirovanie-baz-dannykh?) Дата обращения: 18.05.22
4. Официальный сайт MySQL. Информационный источник (https://www.mysql.com/) Дата обращения: 18.05.22

5. Анализ функциональных возможностей MySql – Информационный источник (<https://studfile.net/preview/5405689/page:3/>) Дата обращения: 20.05.22

6. MySQL Connection Информационный источник (<https://dev.mysql.com/doc/refman/8.0/en/connecting.html>) Дата обращения: 04.06.22

# Приложения

## Приложение 1. Текст скрипт-файла схемы БД

-- MySQL Script generated by MySQL Workbench

-- Fri Jun 10 13:04:54 2022

-- Model: New Model Version: 1.0

-- MySQL Workbench Forward Engineering

SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;

SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;

SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_ENGINE\_SUBSTITUTION';

-- -----------------------------------------------------

-- Schema OfficeTech

-- -----------------------------------------------------

-- -----------------------------------------------------

-- Schema OfficeTech

-- -----------------------------------------------------

CREATE SCHEMA IF NOT EXISTS `OfficeTech` DEFAULT CHARACTER SET utf8 ;

USE `OfficeTech` ;

-- -----------------------------------------------------

-- Table `OfficeTech`.`Otdel`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `OfficeTech`.`Otdel` (

`CodeOtdela` INT NOT NULL,

`Naimenovanie` VARCHAR(45) NOT NULL,

PRIMARY KEY (`CodeOtdela`),

UNIQUE INDEX `CodeOtdela\_UNIQUE` (`CodeOtdela` ASC) VISIBLE,

UNIQUE INDEX `Naimenovanie\_UNIQUE` (`Naimenovanie` ASC) VISIBLE)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `OfficeTech`.`Sotrud`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `OfficeTech`.`Sotrud` (

`KodSotrud` INT NOT NULL,

`FIO` VARCHAR(45) NOT NULL,

`Dolgnost` VARCHAR(45) NOT NULL,

`CodeOtdela` INT NOT NULL,

PRIMARY KEY (`KodSotrud`),

INDEX `fk\_Sotrud\_Otdel1\_idx` (`CodeOtdela` ASC) VISIBLE,

CONSTRAINT `fk\_Sotrud\_Otdel1`

FOREIGN KEY (`CodeOtdela`)

REFERENCES `OfficeTech`.`Otdel` (`CodeOtdela`)

ON DELETE CASCADE

ON UPDATE CASCADE)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `OfficeTech`.`counteragent`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `OfficeTech`.`counteragent` (

`CodeAgent` INT NOT NULL,

`naimenovanie` VARCHAR(45) NOT NULL,

`Kontakt` VARCHAR(45) NOT NULL,

PRIMARY KEY (`CodeAgent`),

UNIQUE INDEX `naimenovanie\_UNIQUE` (`naimenovanie` ASC) VISIBLE)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `OfficeTech`.`Kniga\_Ucheta\_tech`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `OfficeTech`.`Kniga\_Ucheta\_tech` (

`idKnigiTech` INT NOT NULL AUTO\_INCREMENT,

`NomerDoc` INT NOT NULL DEFAULT 0,

`DataTech` DATE NULL,

`KodSotrud` INT NOT NULL,

`CodeOtdela` INT NULL,

`CodeAgent` INT NULL,

`CodeOperacy` INT NULL,

`Stoimost` DECIMAL(10,2) NULL DEFAULT 0,

PRIMARY KEY (`idKnigiTech`),

INDEX `fk\_Kniga\_Ucheta\_tech\_Sotrud\_idx` (`KodSotrud` ASC) VISIBLE,

INDEX `fk\_Kniga\_Ucheta\_tech\_Otdel1\_idx` (`CodeOtdela` ASC) VISIBLE,

INDEX `fk\_Kniga\_Ucheta\_tech\_counteragent1\_idx` (`CodeAgent` ASC) VISIBLE,

CONSTRAINT `fk\_Kniga\_Ucheta\_tech\_Sotrud`

FOREIGN KEY (`KodSotrud`)

REFERENCES `OfficeTech`.`Sotrud` (`KodSotrud`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `fk\_Kniga\_Ucheta\_tech\_Otdel1`

FOREIGN KEY (`CodeOtdela`)

REFERENCES `OfficeTech`.`Otdel` (`CodeOtdela`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `fk\_Kniga\_Ucheta\_tech\_counteragent1`

FOREIGN KEY (`CodeAgent`)

REFERENCES `OfficeTech`.`counteragent` (`CodeAgent`)

ON DELETE CASCADE

ON UPDATE CASCADE)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `OfficeTech`.`Kniga\_Ucheta\_mater`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `OfficeTech`.`Kniga\_Ucheta\_mater` (

`idKnigiMater` INT NOT NULL AUTO\_INCREMENT,

`NomerDoc` VARCHAR(45) NOT NULL DEFAULT 0,

`DataMater` DATE NULL,

`KodSotrud` INT NOT NULL,

`CodeOtdela` INT NULL,

`CodeAgent` INT NULL,

`CodeOper` VARCHAR(45) NULL DEFAULT 0,

`Stoimost` DECIMAL(10,2) NULL DEFAULT 0,

PRIMARY KEY (`idKnigiMater`),

INDEX `fk\_Kniga\_Ucheta\_mater\_Sotrud1\_idx` (`KodSotrud` ASC) VISIBLE,

INDEX `fk\_Kniga\_Ucheta\_mater\_Otdel1\_idx` (`CodeOtdela` ASC) VISIBLE,

INDEX `fk\_Kniga\_Ucheta\_mater\_counteragent1\_idx` (`CodeAgent` ASC) VISIBLE,

CONSTRAINT `fk\_Kniga\_Ucheta\_mater\_Sotrud1`

FOREIGN KEY (`KodSotrud`)

REFERENCES `OfficeTech`.`Sotrud` (`KodSotrud`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `fk\_Kniga\_Ucheta\_mater\_Otdel1`

FOREIGN KEY (`CodeOtdela`)

REFERENCES `OfficeTech`.`Otdel` (`CodeOtdela`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `fk\_Kniga\_Ucheta\_mater\_counteragent1`

FOREIGN KEY (`CodeAgent`)

REFERENCES `OfficeTech`.`counteragent` (`CodeAgent`)

ON DELETE CASCADE

ON UPDATE CASCADE)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `OfficeTech`.`Oborudovanie`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `OfficeTech`.`Oborudovanie` (

`CodeOborudovaniya` INT NOT NULL,

`Naimenovanie` VARCHAR(45) NOT NULL,

PRIMARY KEY (`CodeOborudovaniya`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `OfficeTech`.`Spec\_Tech`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `OfficeTech`.`Spec\_Tech` (

`idSpecTech` INT NOT NULL AUTO\_INCREMENT,

`idKnigiTech` INT NOT NULL,

`CodeOborudovaniya` INT NOT NULL,

`Kolichestvo` INT NULL DEFAULT 0,

`Cena` DECIMAL(10,2) NULL,

PRIMARY KEY (`idSpecTech`),

INDEX `fk\_Spec\_Tech\_Kniga\_Ucheta\_tech1\_idx` (`idKnigiTech` ASC) VISIBLE,

INDEX `fk\_Spec\_Tech\_Oborudovanie1\_idx` (`CodeOborudovaniya` ASC) VISIBLE,

CONSTRAINT `fk\_Spec\_Tech\_Kniga\_Ucheta\_tech1`

FOREIGN KEY (`idKnigiTech`)

REFERENCES `OfficeTech`.`Kniga\_Ucheta\_tech` (`idKnigiTech`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `fk\_Spec\_Tech\_Oborudovanie1`

FOREIGN KEY (`CodeOborudovaniya`)

REFERENCES `OfficeTech`.`Oborudovanie` (`CodeOborudovaniya`)

ON DELETE CASCADE

ON UPDATE CASCADE)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `OfficeTech`.`Rashod\_mater`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `OfficeTech`.`Rashod\_mater` (

`CodeMater` INT NOT NULL,

`Naimenovamie` VARCHAR(45) NOT NULL,

PRIMARY KEY (`CodeMater`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `OfficeTech`.`Sklad\_ostatki`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `OfficeTech`.`Sklad\_ostatki` (

`idOstatkov` INT NOT NULL AUTO\_INCREMENT,

`CodeMater` INT NULL,

`CodeOborudovaniya` INT NULL,

`Kolichestvo` INT NULL DEFAULT 0,

PRIMARY KEY (`idOstatkov`),

INDEX `fk\_Sklad\_ostatki\_Rashod\_mater1\_idx` (`CodeMater` ASC) VISIBLE,

INDEX `fk\_Sklad\_ostatki\_Oborudovanie1\_idx` (`CodeOborudovaniya` ASC) VISIBLE,

CONSTRAINT `fk\_Sklad\_ostatki\_Rashod\_mater1`

FOREIGN KEY (`CodeMater`)

REFERENCES `OfficeTech`.`Rashod\_mater` (`CodeMater`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `fk\_Sklad\_ostatki\_Oborudovanie1`

FOREIGN KEY (`CodeOborudovaniya`)

REFERENCES `OfficeTech`.`Oborudovanie` (`CodeOborudovaniya`)

ON DELETE CASCADE

ON UPDATE CASCADE)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `OfficeTech`.`Spec\_mater`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `OfficeTech`.`Spec\_mater` (

`idSpecMater` INT NOT NULL AUTO\_INCREMENT,

`CodeMater` INT NOT NULL,

`idKnigiMater` INT NOT NULL,

`Kolichestvo` INT NULL DEFAULT 0,

`Cena` DECIMAL(10,2) NULL DEFAULT 0,

PRIMARY KEY (`idSpecMater`),

INDEX `fk\_Spec\_mater\_Kniga\_Ucheta\_mater1\_idx` (`idKnigiMater` ASC) VISIBLE,

INDEX `fk\_Spec\_mater\_Rashod\_mater1\_idx` (`CodeMater` ASC) VISIBLE,

CONSTRAINT `fk\_Spec\_mater\_Kniga\_Ucheta\_mater1`

FOREIGN KEY (`idKnigiMater`)

REFERENCES `OfficeTech`.`Kniga\_Ucheta\_mater` (`idKnigiMater`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `fk\_Spec\_mater\_Rashod\_mater1`

FOREIGN KEY (`CodeMater`)

REFERENCES `OfficeTech`.`Rashod\_mater` (`CodeMater`)

ON DELETE CASCADE

ON UPDATE CASCADE)

ENGINE = InnoDB;

SET SQL\_MODE=@OLD\_SQL\_MODE;

SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;

## Приложение 2. Файлы для загрузки данных

*Контрагенты*

1|Суперпоставщик|+7(495)598-49-05

2|ОптоОффис|+7(495)306-87-88

3|Мерлион|+7(495)124-51-10

4|Оборудование Оптом|+7(495)389-12-28

5|Всё для оффиса|+7(495)459-32-78

6|Айтон|+7(495)051-27-75|\N

Книга учета материалов

1|200|2021-01-12|1|2||213|7500

2|201|2021-01-19|2||1|313|1000

3|202|2021-01-25|5|1||105|600

4|203|2021-02-20|4||4|106|1400

5|204|2021-02-28|2|4||404|60

6|205|2021-03-12|1|4||510|160

7|206|2021-03-26|3||2|312|300

8|207|2021-04-05|3|2||213|100

9|208|2021-04-11|1|3||443|200

10|209|2021-05-13|2||6|223|4750|\N

Книга учета техники

1|210|2021-06-20|2|1||223|300000

2|211|2021-06-26|1|2||206|120000

3|212|2021-06-30|2|3||313|18000

4|211|2021-06-26|3||6|103|8000

5|214|2021-08-11|3|4||404|48000

6|215|2020-10-12|4||3|510|90000

7|216|2021-10-27|4||5|445|240000

8|217|2021-11-05|2||6|213|810000

9|218|2020-12-29|1|4||444|402500

10|219|2021-12-30|4|2||213|600000

11|210|2020-06-20|3|1||223|300000|\N

Оборудование

1|Принтер

2|Сканер

3|Компьютер

4|Планшет

5|Уничтожитель документов

6|Пакетный ламинатор

7|Цифровой диктофон

8|SIP телефон

9|Кондиционер

10|Видеокамера|\N

Расходный материал

1|Ручки

2|Картридж

3|Скотч

4|Скобы для степлеров

5|Ножницы

6|Скрепки

7|Клей

8|Резинки

9|Степлер

10|Наклейки|\N

Отделы

1|Маркетинг

2|Отдел сбыта

3|Производственный

4|Продажи|\N

Складские остатки

1|1||5

2||2|0

3|3||10

4||4|5

5|5||3

6||6|1

7|7||1

8||8|2

9|9||12

10||10|7|\N

Сотрудники

1|Кулакова Людмила Протасьевна|Менеджер|1

2|Рябова Агнесса Михайловна|Секретарь|3

3|Федотов Эдуард Ефимович|Глав.Бугхалтер|1

4|Баранов Лаврентий Артемович|Дир. по персоналу|3

5|Селезнёва Лада Антоновна|Нач. отдела продаж|4

6|Жукова Анастасия Семеновна|Нач. отдела по производству|3|\N

Спецификация материалов

1|2|1|15|500

2|5|2|10|100

3|4|3|4|150

4|7|4|7|200

5|10|5|3|20

6|10|6|8|20

7|6|7|10|30

8|3|8|2|50

9|1|9|4|50

10|9|10|13|250

11|9|10|6|250|\N

Спецификация техники

1|2|1|10|12000

2|4|2|2|4000

3|9|3|7|50000

4|7|4|8|30000

5|10|5|20|30000

6|5|6|4|12000

7|4|7|9|8000

8|3|8|2|9000

9|6|9|5|18000

10|9|10|7|7500

11|11|3|6|50000

12|1|3|6|50000|\N

## Приложение 3. Команды загрузки данных

LOAD DATA INFILE "F:\\Practic\\DataImport\\otdel.txt"

INTO TABLE otdel

FIELDS TERMINATED BY '|'

LINES TERMINATED BY '\r\n';

LOAD DATA INFILE "F:\\Practic\\DataImport\\sotrud.txt"

INTO TABLE sotrud

FIELDS TERMINATED BY '|'

LINES TERMINATED BY '\r\n';

LOAD DATA INFILE "F:\\Practic\\DataImport\\counteragent.txt"

INTO TABLE counteragent

FIELDS TERMINATED BY '|'

LINES TERMINATED BY '\r\n';

LOAD DATA INFILE "F:\\Practic\\DataImport\\oborudovanie.txt"

INTO TABLE oborudovanie

FIELDS TERMINATED BY '|'

LINES TERMINATED BY '\r\n';

LOAD DATA INFILE "F:\\Practic\\DataImport\\rashod\_mater.txt"

INTO TABLE rashod\_mater

FIELDS TERMINATED BY '|'

LINES TERMINATED BY '\r\n';

LOAD DATA INFILE "F:\\Practic\\DataImport\\sklad\_ostatki.txt"

INTO TABLE sklad\_ostatki

FIELDS TERMINATED BY '|'

LINES TERMINATED BY '\r\n'

(idOstatkov, @vCodeMater, @vCodeOborudovaniya, Kolichestvo)

SET CodeMater = NULLIF(@vCodeMater,'') , CodeOborudovaniya = NULLIF(@vCodeOborudovaniya,'');

LOAD DATA INFILE "F:\\Practic\\DataImport\\kniga\_ucheta\_mater.txt"

INTO TABLE kniga\_ucheta\_mater

FIELDS TERMINATED BY '|'

LINES TERMINATED BY '\r\n'

(idKnigiMater,NomerDoc,DataMater,KodSotrud,@vCodeOtdela,@vCodeAgent,CodeOper,Stoimost)

SET CodeOtdela = NULLIF(@vCodeOtdela,'') , CodeAgent = NULLIF(@vCodeAgent,'');

LOAD DATA INFILE "F:\\Practic\\DataImport\\kniga\_ucheta\_tech.txt"

INTO TABLE kniga\_ucheta\_tech

FIELDS TERMINATED BY '|'

LINES TERMINATED BY '\r\n'

(idKnigiTech,NomerDoc,DataTech,KodSotrud,@vCodeOtdela,@vCodeAgent,CodeOperacy,Stoimost)

SET CodeOtdela = NULLIF(@vCodeOtdela,'') , CodeAgent = NULLIF(@vCodeAgent,'');

LOAD DATA INFILE "F:\\Practic\\DataImport\\spec\_tech.txt"

INTO TABLE spec\_tech

FIELDS TERMINATED BY '|'

LINES TERMINATED BY '\r\n';

LOAD DATA INFILE "F:\\Practic\\DataImport\\spec\_mater.txt"

INTO TABLE spec\_mater

FIELDS TERMINATED BY '|'

LINES TERMINATED BY '\r\n';

## Приложение 4. Тексты представлений

Представление 1

select kt.idKnigitech, kt.NomerDoc,kt.CodeOperacy,kt.DataTech, ob.Naimenovanie,st.kolichestvo,ob.cena,

sum(ob.Cena\*st.kolichestvo) as Stoim

From Kniga\_Ucheta\_tech as kt

Join Spec\_tech st on kt.idKnigiTech = st.idKnigitech

Join oborudovanie ob on ob.CodeOborudovaniya = st.CodeOborudovaniya

group by kt.CodeOperacy

Представление 2

select ob.Naimenovanie, Year(kt.DataTech) as Year,Month(kt.DataTech)as Month,st.kolichestvo,

sum(ob.Cena\*st.kolichestvo) as Stoim

From Kniga\_Ucheta\_tech as kt

Join Spec\_tech st on kt.idKnigiTech = st.idKnigitech

Join oborudovanie ob on ob.CodeOborudovaniya = st.CodeOborudovaniya

group by Year(kt.DataTech), Month(kt.DataTech)

having Stoim > 200000

order by Stoim desc

## Приложение 5. Тексты триггеров

Триггер для заполнения

CREATE DEFINER=`admin`@`%` TRIGGER `kniga\_ucheta\_tech\_BEFORE\_INSERT` BEFORE INSERT ON `kniga\_ucheta\_tech` FOR EACH ROW BEGIN

IF (New.DataTech is null) THEN

SET New.DataTech = current\_date();

END IF;

END

Триггер для контроля вводимых данных

CREATE DEFINER=`admin`@`%` TRIGGER `kniga\_ucheta\_tech\_AFTER\_INSERT` AFTER INSERT ON `kniga\_ucheta\_tech` FOR EACH ROW BEGIN

IF (New.Stoimost>0)THEN

SIGNAL SQLSTATE VALUE '99999' SET MESSAGE\_TEXT='Стоимость не заполняется';

END IF;

END

Триггер для контроля событий

CREATE DEFINER=`admin`@`%` TRIGGER `kniga\_ucheta\_tech\_BEFORE\_UPDATE` BEFORE UPDATE ON `kniga\_ucheta\_tech` FOR EACH ROW BEGIN

if new.dataTech < current\_date()

then signal sqlstate value '99999'

set message\_text = 'невозможно изменить данные';

end if;

END

Триггеры для изменения (перерасчета) вычисляемых полей

CREATE DEFINER=`admin`@`%` TRIGGER `spec\_tech\_AFTER\_INSERT` AFTER INSERT ON `spec\_tech` FOR EACH ROW BEGIN

UPDATE Kniga\_ucheta\_tech SET

Stoimost = Stoimost + (New.Cena \* New.Kolichestvo)

WHERE idKnigiTech = New.idKnigiTech;

END

Триггер на обновление

CREATE DEFINER=`admin`@`%` TRIGGER `spec\_tech\_AFTER\_UPDATE` AFTER UPDATE ON `spec\_tech` FOR EACH ROW BEGIN

UPDATE Kniga\_ucheta\_tech SET

Stoimost = Stoimost - (Old.Cena \* Old.Kolichestvo) + (New.Cena \* New.Kolichestvo)

WHERE idKnigiTech = New.idKnigiTech;

END

Триггер на удаление

CREATE DEFINER=`admin`@`%` TRIGGER `spec\_tech\_AFTER\_DELETE` AFTER DELETE ON `spec\_tech` FOR EACH ROW BEGIN

UPDATE Kniga\_ucheta\_tech SET

Stoimost = Stoimost - (Old.Cena \* Old.Kolichestvo)

WHERE idKnigiTech = Old.idKnigiTech;

END