

АНО ВО МЕЖДУНАРОДНЫЙ БАНКОВСКИЙ ИНСТИТУТ

им. Анатолия Собчака

[Кафедра прикладной информатики и моделирования экономических процессов](http://eos29.ibi.spb.ru/course/index.php?categoryid=9)

**КУРСОВАЯ РАБОТА**

по дисциплине "Базы данных"

на тему: «Проектирование и разработка реляционной базы данных средствами современных серверных СУБД»

**Проверил**:

доцент, к.п.н. Пискунова Т.Г.

**Выполнил:**

Арсентий Грищенко

Студент группы

114

Санкт-Петербург

2023

МЕЖДУНАРОДНЫЙ БАНКОВСКИЙ ИНСТИТУТ

Кафедра прикладной информатики и моделирования экономических процессов

ЗАДАНИЕ

**на курсовую работу по дисциплине «Базы данных»**

выдано студенту : Арсентий Грищенко

(Фамилия и инициалы)

ЭиПр наук 114 2**\_\_\_\_\_\_**

Факультет Гр. Курс

Тема работы: «Проектирование и разработка реляционной базы данных средствами современных клиент - серверных СУБД»

План выполнения курсовой работы:

|  |  |  |
| --- | --- | --- |
| Этапы курсовой работы: | Практические результаты этапа | Сроки |
| Выбор и утверждение темы курсовой работы. Составление задания на курсовую работу | Заполненный бланк Задания. |  |
| Анализ и описание предметной области | Описание раздела «Введение»: цели и задачи курсовой работы, краткое описание предметной области как области экономической деятельности организации или предприятия. |  |
| Постановка задачи и формализованное описание требований к разрабатываемой базе данных | Описание раздела «Постановка задачи», в котором приведен список требований (поставленных пользователем задач) к разрабатываемой базе данных |  |
| Разработка модели данных предметной области для разрабатываемой базы данных | Разработка структуры данных предметной области, для разрабатываемой БД. Описание структуры данных, определение типов даны, нормализация отношений БД приведенное в разделе “Постановка задачи и формализованное описание предметной области” |  |
| Разработать концептуальную, логическую модель базы данных | Описание раздела «Проектирование базы данных», в котором приведена ка концептуальная модель базы данных. |  |
| Реализация физической структуры БД в СУБД  MY SQL с помощью CASE-средств создать физическую модель базы данных; сгенерировать скрипт на SQL для реализации базы данных в СУБД; | Описание раздела «Проектирование базы данных», в котором приведена физическая структура БД и таблиц БД.  Физическая модель БД, реализованная и сохраненная с помощью CASE-средств. Генерация sql кода приведена в приложении |  |
| Разработать хранимые процедуры с входными и выходными параметрами;  разработать функции;  разработать триггеры | Описание раздела “Разработка хранимых процедур, функций и триггеров”. Сохранение на сервере выполненных разработок. |  |
| Оформление пояснительной записки | Сдача разработки и пояснительной записки на проверку преподавателя через ЕЭОС МБИ для получения допуска к защите. |  |

Руководитель курсовой работы: доцент, к.п.н. Пискунова Т.Г.

(Должность, фамилия, инициалы) (подпись)

# введение

### Тема курсовой работы:

Разработать базу данных фотоателье, реализующую следующие процессы: прием заказов на съемку (на документы и художественную), прием заказов съемки с выездом на место и учет исполнения заказов, учет оплаты и выдачи фотографий. Разработать запросы, перечень которых указан в разделе исходных данных по курсовой. Разработать хранимые процедуры: поиск клиента и сотрудника фотоателье по параметрам, внесение нового клиента фотоателье, подсчёта количества заказов и стоимости оказанных услуг.

Цель курсовой работы:

Спроектировать базу данных склада аптечных товаров на основе концептуальной, логической и физической модели. В качестве инструмента проектирования выбрать CASE-средство. Разработать базу данных на основе СУБД MySQL.

# Задачи курсовой работы:

* проанализировать и описать предметную область БД;
* разработать концептуальную модель базы данных с помощью CASE инструментария;
* на основе концептуальной модели, сгенерировать логическую и физическую модель базы данных с помощью CASE-инструментария;
* сгенерировать SQL-скрипт описания физической модели базы данных в MySQL;
* проверить правильность сгенерированного скрипта, проиллюстрировать диаграмму созданной базы данных;
* разработать запросы базы данных;
* разработать хранимые процедуры базы данных с входными и выходными параметрами;
* разработать триггеры базы данных;
* заполнить таблицы данными и подготовить к тестированию.

# ПОСТАНОВКА ЗАДАЧИ

Курсовая работа является практической разработкой, которая позволит получить навык применения методологического и инструментального аппарата для построения информационных систем на основе реляционных баз данных для решения задач в области экономики и менеджмента.

Курсовая работа формирует системный подход в освоении информационных технологий и выполняется как разработка информационного продукта в соответствии с его жизненным циклом. Работа должна последовательно реализовывать следующие этапы жизненного цикла:

1. этап анализа и описания предметной области БД;
2. этап постановки задачи и формализованного описания требований к БД;
3. этап разработки модели данных БД;
4. этап проектирования базы данных (БД);
5. этап разработки хранимых процедур и триггеров для БД;
6. этап документирования БД (пояснительная записка к курсовой работе).

# МОДЕЛИРОВАНИЕ ДАННЫХ

### **Анализ предметной области. Создание объектов базы данных**.

Перед началом проектирование концептуальной модели базы данных, необходимо продумать объекты, из которых она будет состоять. Предметная область данной курсовой работы – разработка базы данных для склада аптечных товаров. При анализе предметной области формируем сущности необходимые нам для создания БД: товар, поставщик, поставка, доставка, заказ, магазин, персонал, клиент. Для разработки базы данных использовалась MySQL WORKBENCH 8 CE. При создании объектов базы данных необходимо задать корректно имена сущностей и атрибутов с учетом ограничений MySQL. Коды объектов (Code) соответствуют идентификаторам объектов в физической модели и поэтому задаются с учетом ограничений: набор символов, содержащих латинские буквы, цифры и знаки подчеркивания и не содержащий недопустимых символов, длиной не более 128.

Ниже, в таблицах 1-8, представлены все созданные сущности с его атрибутами, жирным шрифтом обозначен атрибут, который планируется сделать ключевым.

Таблица 1. Сущность товар

|  |  |  |
| --- | --- | --- |
| товар | | |
| **Код\_товар** | артикул | наименование |
| Описание | цена |  |

Таблица 2. Сущность заказы

|  |  |  |
| --- | --- | --- |
| заказы | | |
| **Код\_заказа** | Дата\_заказа | отметка |

Таблица 3. персонал

|  |  |  |
| --- | --- | --- |
| покупатели | | |
| **Код\_покупателя** | фамилия | имя |
| отчество | адрес | почта |

Таблица 4. Сущность Клиенты

|  |  |  |  |
| --- | --- | --- | --- |
| сотрудники | | | |
| **Код\_сотрудника** | фамилия | имя | |
| отчество | почта | адрес |
| образование | Дата\_принятия |  |

Таблица 5. Сущность заказы-услуги

|  |  |
| --- | --- |
| Заказы-услуги | |
| количество | допцена |

Таблица 6. Сущность заказы-товары

|  |  |
| --- | --- |
| заказы-товары | |
| количество | допцена |

Таблица 7. Сущность услуги

|  |  |
| --- | --- |
| услуги | |
| **Код\_услуги** | наименование |
| описание | цена |

Таблица 8. Сущность группы товаров

|  |  |  |
| --- | --- | --- |
| Группы товаров | | |
| **Код\_группы** | название | телефон |

Названия кодов и типов данных атрибутов модели “фотоателье” приведена в таблице 9.

Таблица 9

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Атрибут | Тип | Длина |
| Код товара | код | int |  |
| Наименование товара | Наименование | Varchar(45) | 45 |
| Артикул товара | Артикул | Varchar(10) | 10 |
| описание товара | описание | Varchar(100) | 100 |
| Цена товара | цена | double |  |
| Код группы | Код группы | int |  |
| название | названиие | Varchar(45) | 45 |
| Количество товара | количество | Int |  |
| Доп цена за ед товара | Доп цена | Int |  |
| Код услуги | Код услуги | Int |  |
| Наименование услуги | Наименование | Varchar(45) | 45 |
| Описание услуги | описание | Varchar(100) | 100 |
| Количество услуг | количество | int |  |
| Доп цена за ед услуги | Доп цена | Double |  |
| Код сотрудника | Код\_сотрудника | Int |  |
| фамилия сотрудника | фамилия | Varchar(45) | 45 |
| имя сотрудника | имя | Varchar(45) | 45 |
| отчество сотрудника | отчество | Varchar(45) | 45 |
| Адрес сотрудника | адрес | Varchar(150) | 150 |
| Почта сотрудника | почта | Varchar45 | 45 |
| образование | образование | Varchar(45) |  |
| должность | должность | Varchar(45) | 45 |
| Дата\_принятия | Дата\_принятия | date |  |
| Код покупателя | код покупателя | int |  |
| фамилия сотрудника | фамилия | Varchar(45) | 45 |
| имя сотрудника | имя | Varchar(45) | 45 |
| отчество сотрудника | отчество | Varchar(45) | 45 |
| Адрес сотрудника | адрес | Varchar(150) | 150 |
| Почта сотрудника | почта | Varchar45 | 45 |
| Код заказа | Код\_заказа | Int |  |
| Дата\_заказа | Дата\_заказа | date |  |
| отметка | отметка | Enum(“да”,”нет”) |  |
| Дата\_окончания | Дата\_окончания | date |  |

### Создание концептуальной модели

В ходе работы была создана концептуальная модель будущей базы данных. Модель состоит из 8 сущностей и промежуточной таблицы. Концептуальная модель дает общее представление о предметной области. В концептуальной модели находят свое отражение все актуальные требования предметной области на стадии проектирования, также на этом этапе определяются задачи, которые будет решать пользователь с помощью базы данных. На рис. 1 представлена концептуальная модель. Концептуальная модель сохраняется в виде файла в формате .jpg.

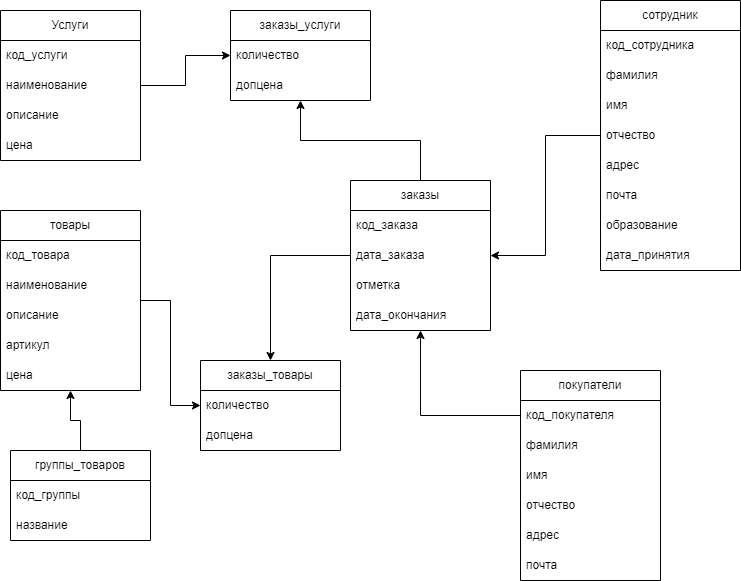


рис. 1. Концептуальная модель

### Физическая модель

Физическая модель БД определяет способ размещения данных в среде хранения и способы доступа к этим данным, которые поддерживаются на физическом уровне. Исторически первыми системами хранения и доступа были файловые структуры и системы управления файлами (СУФ), которые фактически являлись частью операционных систем. СУБД создавала над этими файловыми моделями свою надстройку, которая позволяла организовать всю совокупность файлов таким образом, чтобы она выглядела как единое целое и получала централизованное управление от СУБД. Однако непосредственный доступ осуществлялся на уровне файловых команд, которые СУБД использовала при манипуляции всеми файлами, составляющие хранимые данные одной или нескольких баз данных.

Физическую схему базы данных создаётся в MySQL Workbench. Физические модели определяют способы размещения данных в среде хранения и способы доступа к этим данным, которые поддерживаются на физическом уровне. На рис. 2 представлен скриншот внутренней среды разработки модели.

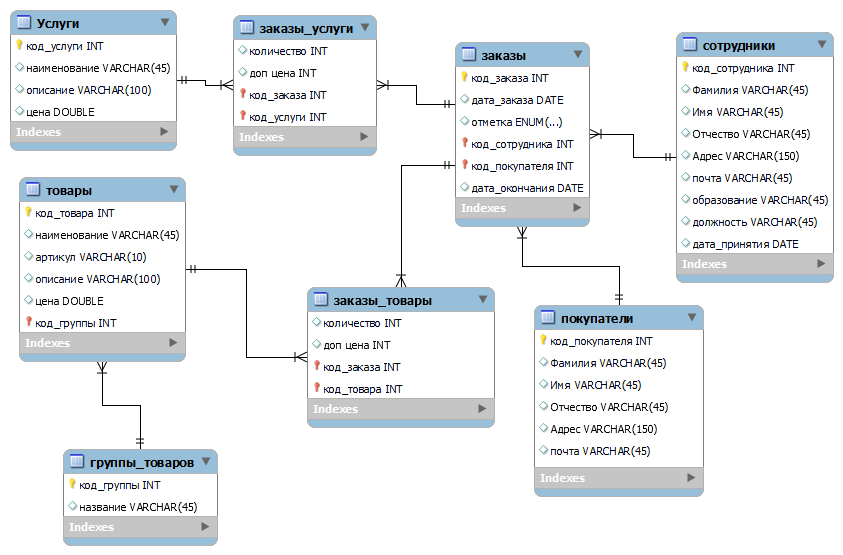


рис. 2

### Связи между таблицами

Все связи в физической модели идентифицирующие, т.к. экземпляр сущности-потомка однозначно определяется своей связью (отношением) с сущностью-родителем. В таком случае связь каждый экземпляр подчиненной сущности идентифицируется значениями атрибутов родительской сущности. Это означает, что экземпляр подчиненной сущности зависит от родительской сущности и не может существовать без экземпляра родительской сущности. В идентифицирующем отношении единственный экземпляр родительской сущности связан с множеством экземпляров подчиненной.

### Выгрузка на сервер

Для того, чтобы загрузить на сервер, поэтапно выполняются следующе действия, которые показаны на рисунках 3 - 6.

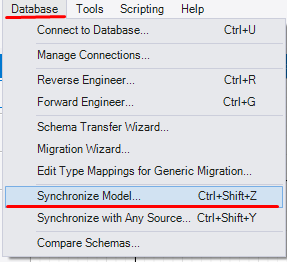


рис. 3

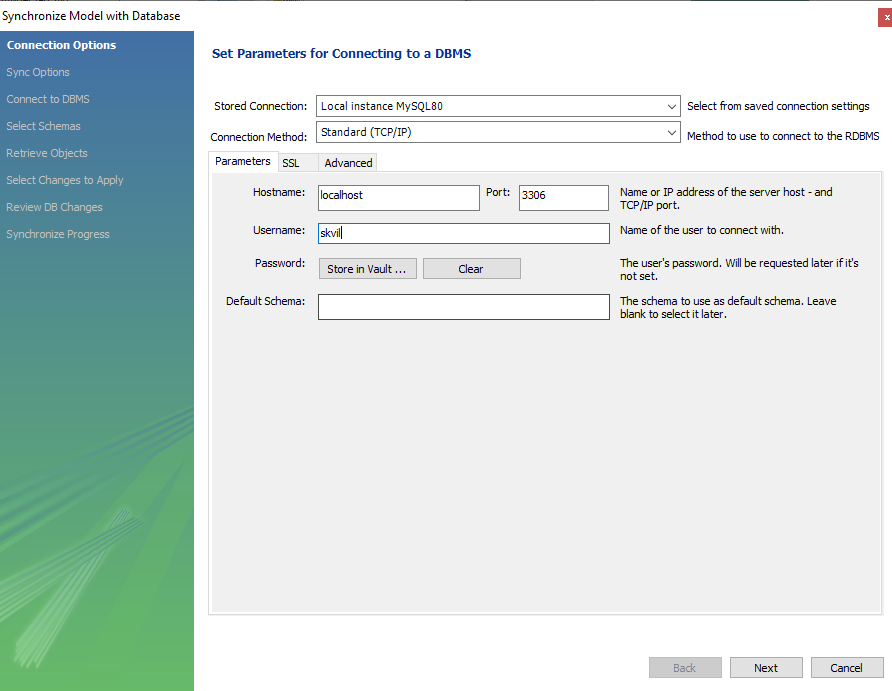


рис. 4

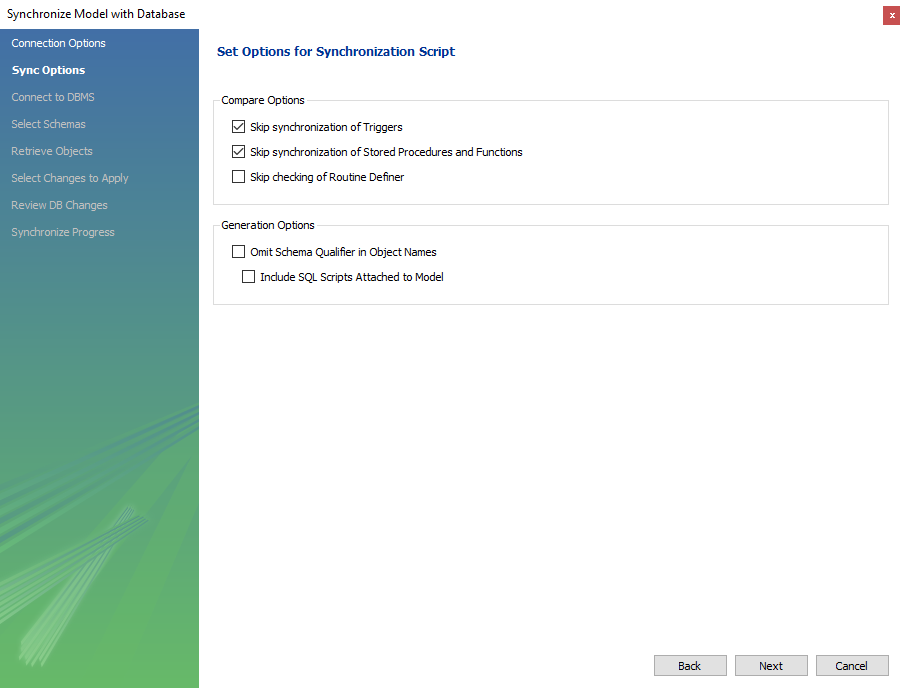


рис. 5

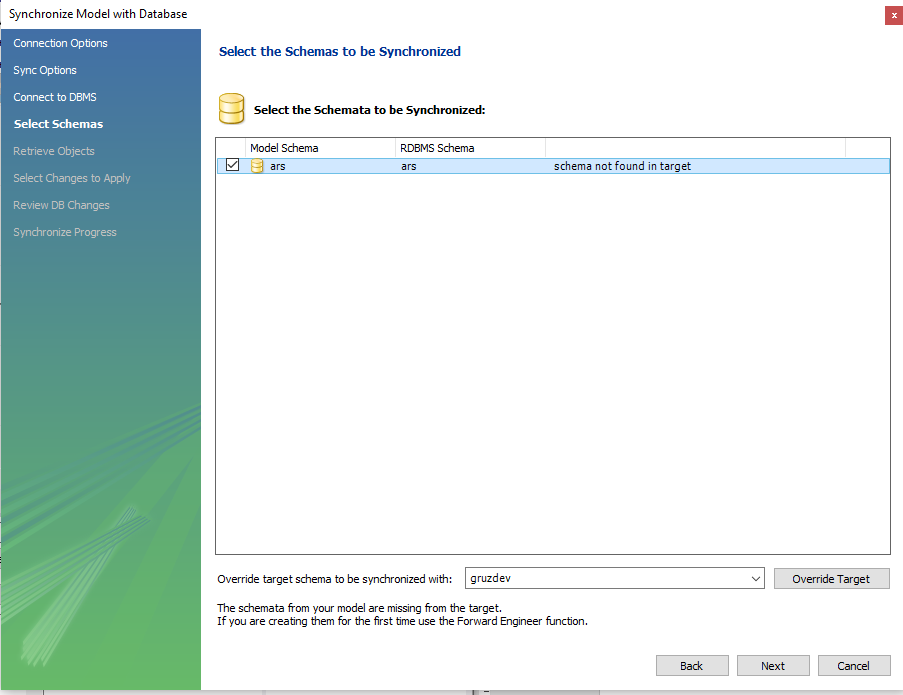


рис. 6

Далее проверяется на сервере рис.7 .

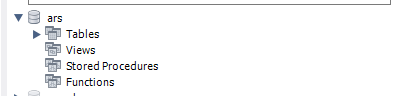


рис. 7

### Заполнение данных

На рисунках 8 -16 показаны все данные всех таблиц.

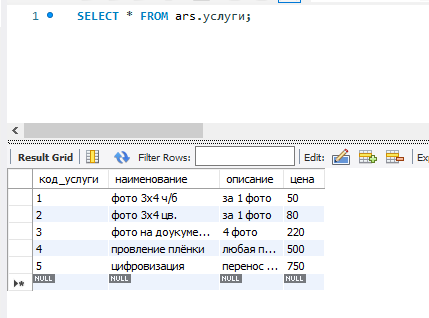


рис. 8 Таблица услуги

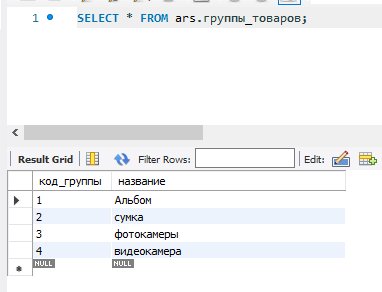


рис. 9 Таблица группы товаров

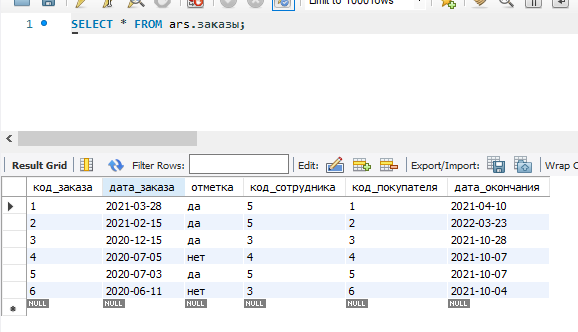


рис. 10 Таблица заказы

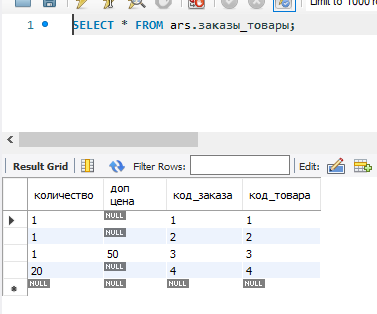


рис. 11 Таблица заказы\_товары

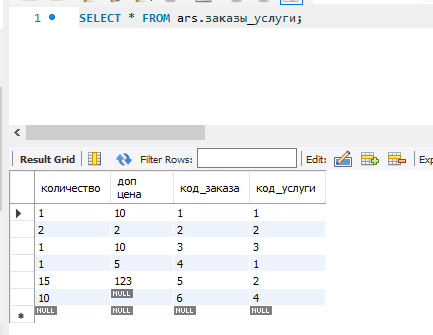


рис. 12 Таблица заказы\_услуги

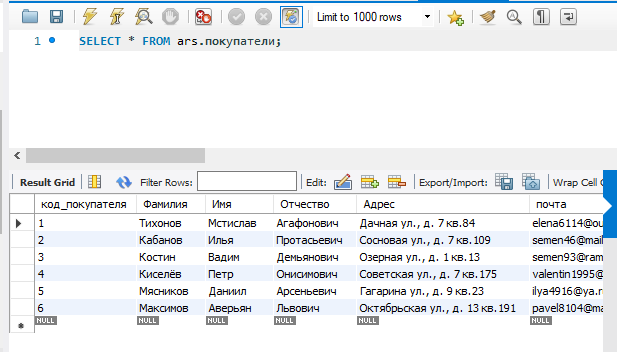


рис. 13 Таблица покупатели

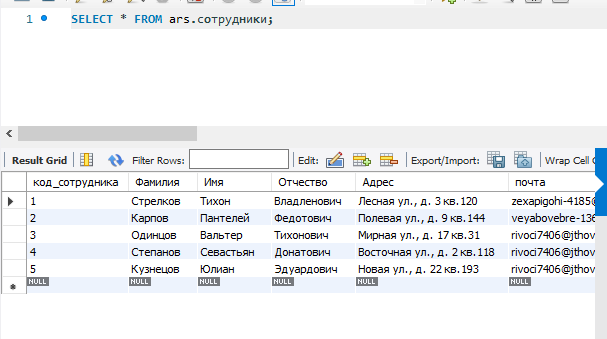


рис. 14 Таблица сотрудники

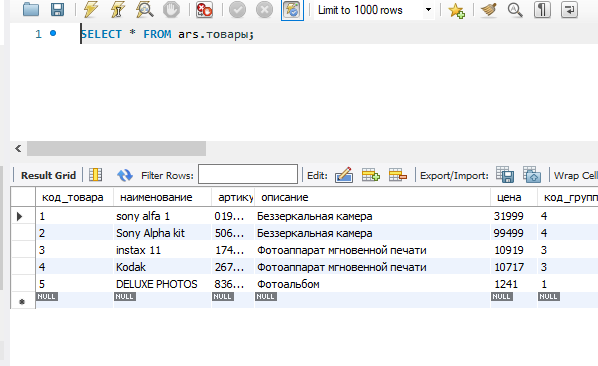


рис. 15 Таблица товары

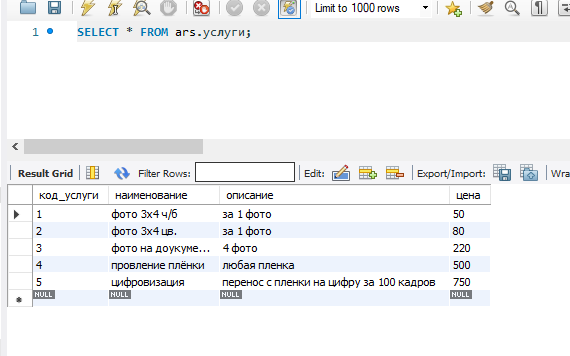


рис. 16 Таблица услуги

# запросы

На рисунках 17 - 27 представлены запросы на выборку.

1.Запрос показывает все товары со всеми полями. Рис.17.

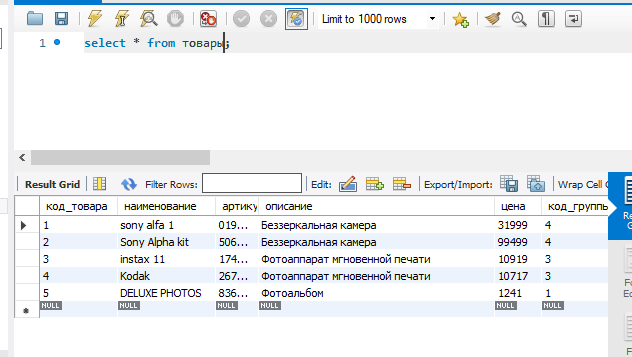


рис. 17 все товары

2. Запрос, показывающий сотрудника. Рис.18.

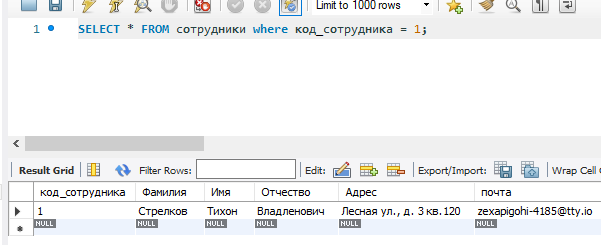


рис. 18 сотрудник с определенным кодом

3.Запрос показывает стаж сотрудника. Рис.19.

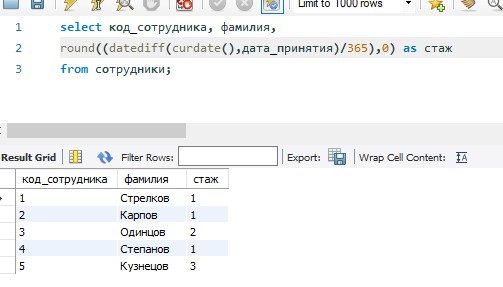


рис. 19 стаж сотрудников

4. Данный запрос ФИО персонала. Рис.20.

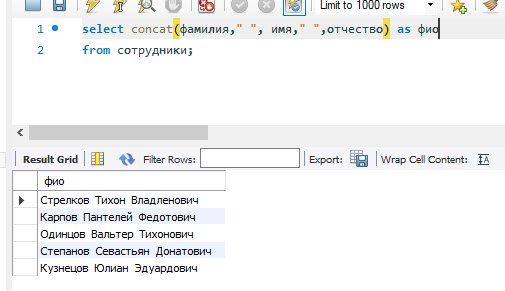


рис. 20 ФИО персонала

5. Вывод поля, у которых группа товаров «фотокамеры». Рис.21.

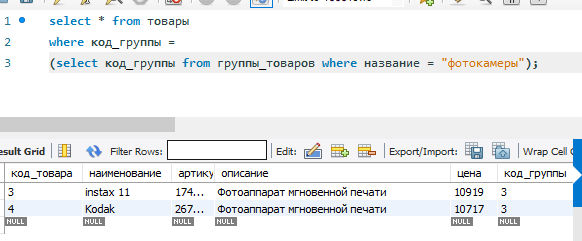


рис. 21 поля у которых группа товаров «фотокамеры»

6. Вывод заказы, покупателя и сотрудника. Рис.22.

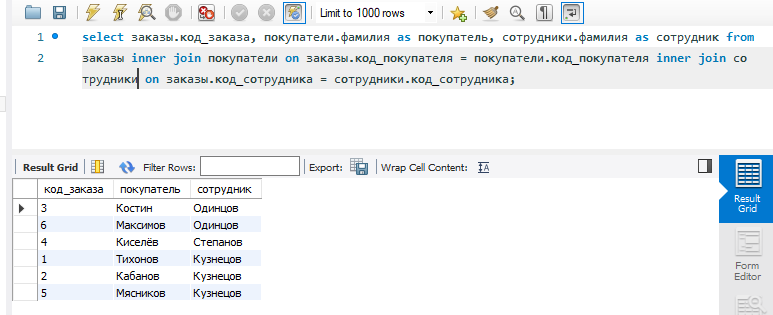


рис. 22 Вывод заказы, покупателя и сотрудника

7. Запрос выводит количество всех заказов. Рис.23.

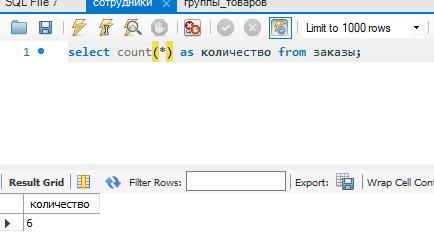


рис. 23 количество всех заказов

8. Сумма заказов на услуги. Рис.24.

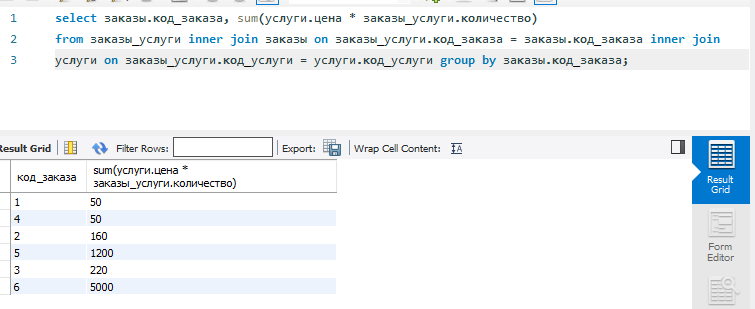


рис. 24 Сумма заказов на услуги

9. Запрос выводит всех клиентов, у которых в фамилии есть «а». Рис.25.

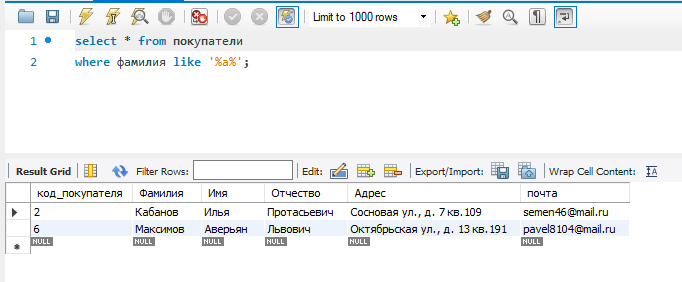


рис. 25 выводит всех клиентов, у которых в фамилии есть «а»

10. Сумма заказов на товары Рис.26.

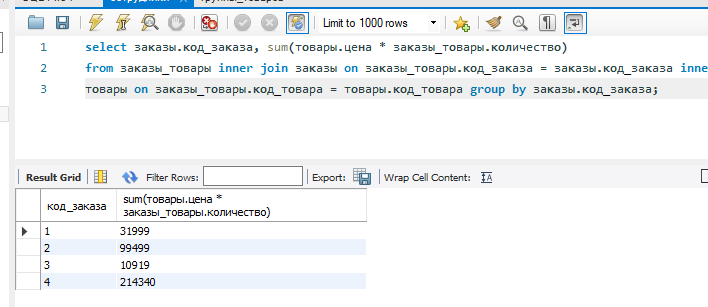


рис. 26 возраст вступления на работу

11. Количество дней для выполнения заказа. Рис.27.

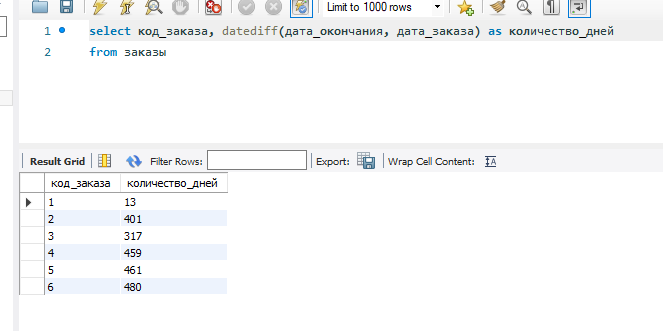


рис. 27 Количество дней для выполнения заказа

# 4. ПРОЦЕДУРЫ

Итак, хранимые процедуры в SQL — это аналог функций в других языках программирования. Хранимые процедуры могут выполнять действия над данными автоматически: вывод данных, удаление, изменение — то, что мы делали вручную.

Хранимые процедуры в СУБД MySQL можно создать несколькими способами. В данной работе все процедуры создаются с sql консоли, Также как и все запросы SELECT, INSERT и т.д.

1.Создание процедуры, которая вносит нового клиента.28.

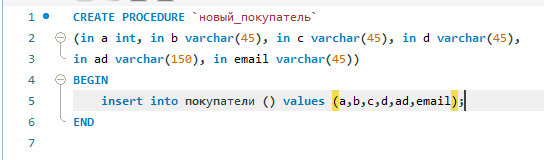


рис. 28 новый клиент

Проверка процедуры Рисунки 29-31

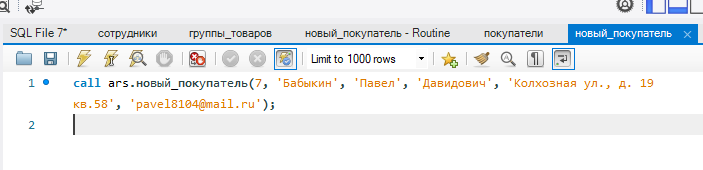


рис. 29



рис. 30

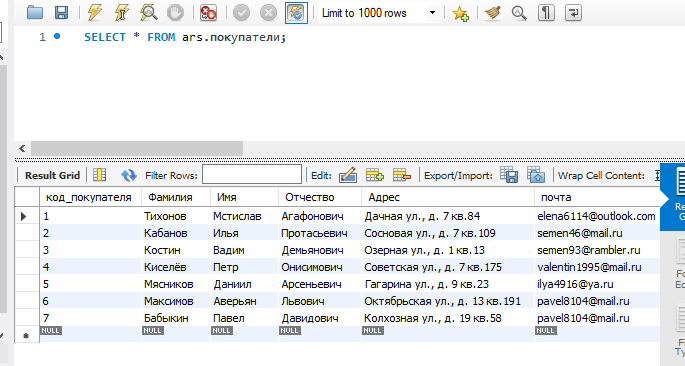


рис. 31

1. Процедура на создание внесение новых данных о сотруднике. Рис.32

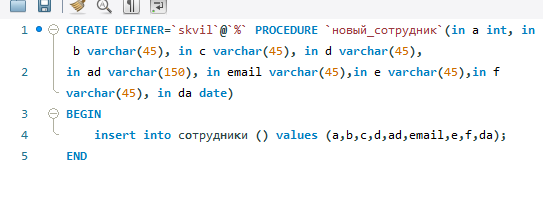


рис. 32 новый сотрудник

Проверка процедуры. Рисунки 33 - 35.

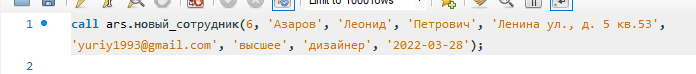


рис. 33



рис. 34

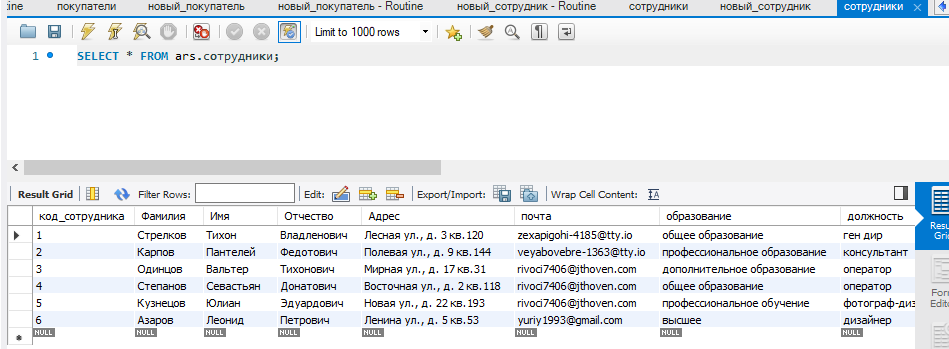


рис. 35

3.Процедура, добавляет новый товар. Рис36.

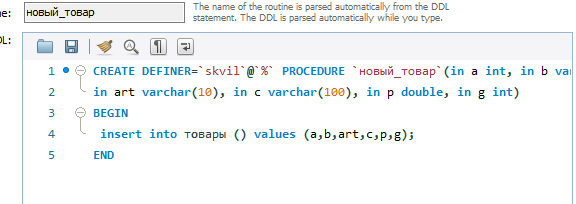


рис. 36 новый товар

Проверка процедуры. Рисунки 37 - 38.

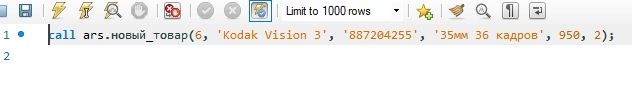


рис. 37

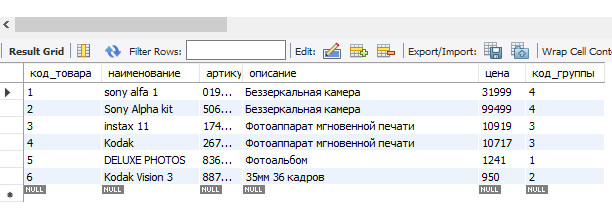


рис. 38

1. Процедура показывает все заказы у определенного клиента. Рис.39

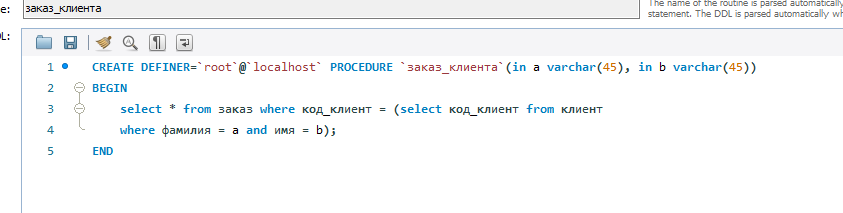


рис. 39

Проверка процедуры. Рис. 40.

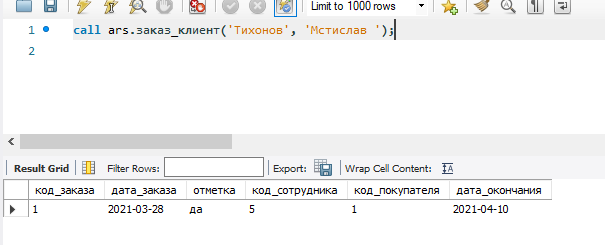


рис. 40

# Функции

В MySQL функция - это хранимая программа, в которую вы можете передавать параметры и возвращать значение.

Функция отличается от процедуры. Таблица 11

Таблица 11

|  |  |
| --- | --- |
| Функция | Процедура |
| Функция должна возвращать значение | Хранимая процедура может как возвращать, так и не возвращать значение. |
| Функции не могут возвращать несколько результирующих наборов. | Хранимая процедура может сформировать и вернуть несколько результирующих наборов данных. |
| Функции можно использовать в операторе SELECT. | Процедуры нельзя использовать в операторе SELECT и во всех его секциях (WHERE, JOIN, HAVING и т.д.), так как процедуры вызываются с помощью команды EXECUTE или EXEC. |
| Из функции нельзя вызвать хранимые процедуры. | В хранимых процедурах можно вызывать и функции, и другие хранимые процедуры. |
| В функциях можно использовать только табличные переменные, временные таблицы использовать не получится. | В хранимых процедурах можно использовать как табличные переменные, так и временные таблицы |
| В функциях можно использовать только входные параметры. | В хранимых процедурах можно использовать как входные, так и выходные параметры. |
| В функциях можно использовать только оператор SELECT на выборку данных.  Операторы DML (INSERT, UPDATE, DELETE) для модификации данных использовать нельзя. | В хранимых процедурах можно использовать оператор SELECT, а также операторы DML (INSERT, UPDATE, DELETE) для модификации данных. |

1. Функция возвращает ФИО. Рис.43



рис. 43 количество работ у мастера

Проверка функции рис. 44.

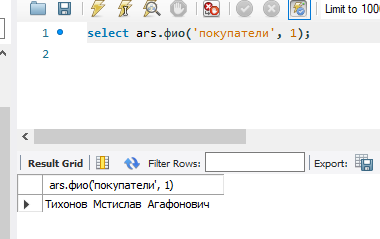


рис. 44

1. Функция показывает сумму всех заказов у клиента.Рис.45.

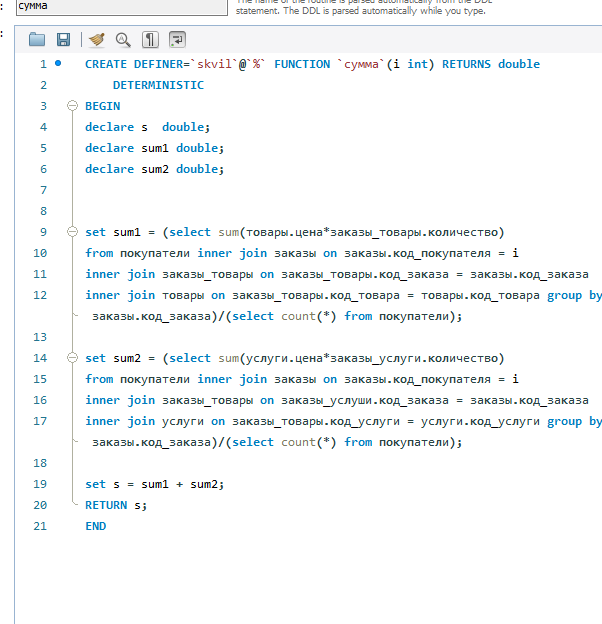


рис. 45 Функция показывает сумму всех заказов у клиента

Проверка функции. Рис.46.

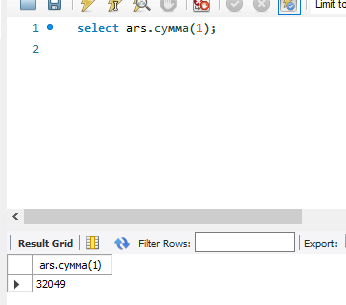


рис. 46

# триггеры

Триггер в MySQL — это определяемая пользователем SQL-команда, которая автоматически вызывается во время операций INSERT, DELETE или UPDATE. Код триггера связан с таблицей и уничтожается после удаления таблицы.

1. Триггер изменяет дату на текущую если в отметку у заказ меняется на да. Рис.47

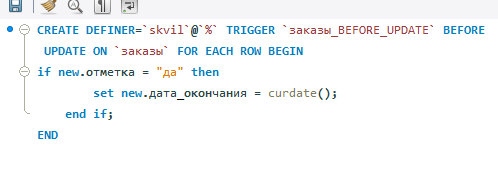


рис. 47 текущая дата

# Заключение

Основываясь на выполненной курсовой работе, можно прийти к выводу о том, что СУБД MySQL, которая служит в качестве системы управления базами данных, является очень хорошим инструментом для того, чтобы создавать собственные реляционные базы данных.

В данной курсовой работе была спроектирована информационная система, автоматизирующая работу фотоателье, реализующую следующие процессы: прием заказов на съемку (на документы и художественную), прием заказов съемки с выездом на место и учет исполнения заказов, учет оплаты и выдачи фотографий. Разработать запросы, перечень которых указан в разделе исходных данных по курсовой. Разработать хранимые процедуры: поиск клиента и сотрудника фотоателье по параметрам, внесение нового клиента фотоателье, подсчёта количества заказов и стоимости оказанных услуг.

Выполняя курсовую работу, были созданы 8 сущностей, а также разнообразные запросы на выборку, функции, триггеры и хранимые процедуры. Кроме того, были проведены тесты, для того чтобы проверить автоматизацию базы данных.

# Список литературы

* Изучаем SQL. Генерация, выборка и обработка данных, 3-е изд./

Алан Болье; пер. с англ. И.В. Красикова. — Киев. : “Диалектика”,

2021. — 402 с.: ил. — Парал. тит . англ.