# вВедение

#### Тема курсовой работы:

Разработать базу данных базы данных отдела кадров фирмы, реализующую следующие процессы: учет личных данных сотрудников, учет функций кадровой службы (прием нового сотрудника на работу, различные операции с сотрудником: увольнение, перемещения по отделам предприятия), учет приказов по контингенту (сотрудникам). Разработать хранимые процедуры: внесение данных о новом сотруднике отдела, хранение личных данных о сотруднике, изменение статуса сотрудника (работает, уволен в отпуске), вычисление количества сотрудников конкретного отдела и всего предприятия в целом, проверка наличия сотрудника в конкретном отделе.

Цель курсовой работы:

Разработать базу данных склада на основе концептуальной, логической и физической модели. В качестве инструмента проектирования выбрать CASE-средство. Разработать базу данных на основе СУБД MYSQL WORKBENCH 8 CE.

#### 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 WORKBENCH 8 CE;
* 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. При анализе предметной области формируем сущности необходимые нам для создания БД: перераспределение, социальные\_льготы, отдел, личные\_сведения, паспорт, должность, вакансии, кадры. Для разработки базы данных использовалась Draw.IO и MYSQL WORKBENCH 8 CE. При создании объектов базы данных необходимо задать корректно имена сущностей и атрибутов с учетом ограничений MYSQL WORKBENCH 8 CE. Имена объектов в концептуальной модели (Name) могут быть набраны по-русски и содержать пробелы и другие недопустимые символы. 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 Перераспределение

|  |  |
| --- | --- |
| Перераспределение | |
| **Код\_ Перераспределение** | Код\_сотрудника |
| Перенаправлен\_от | Перенаправлен\_в |
| Дата\_перенаправления |  |

Таблица Социальные\_льготы

|  |  |
| --- | --- |
| Социальные\_льготы | |
| **Код\_записи** | Код\_сотрудника |
| наименование | Номер\_документа |
| Дата\_выдачи\_документа |  |

Таблица отдел

|  |  |
| --- | --- |
| отдел | |
| **Код\_отдела** | Название\_отдела |
| Глава\_отдела |  |

Таблица должность

|  |  |
| --- | --- |
| должность | |
| **Код\_должности** | Наименование\_должности |
| оклад |  |

Таблица паспорт

|  |  |
| --- | --- |
| паспорт | |
| **Код\_паспорта** | серия |
| номер | Дата\_выдачи |
| Дата\_выдачи | Код\_подразделения |

Таблица вакансии

|  |  |
| --- | --- |
| вакансии | |
| **Код\_вакансии** | Код\_должности |
| Код\_отдела | Дата\_обьявления |

Таблица кадры

|  |  |  |
| --- | --- | --- |
| кадры | | |
| **Код\_должности** | Код\_отдела | Код\_сотрудника |
| стаж | статус | Начало\_трудовой\_деятельности |

Таблица личные\_сведения

|  |  |  |
| --- | --- | --- |
| личные\_сведения | | |
| **Код\_сотрудника** | Фамилия | имя |
| отчество | паспорт | стаж |
| адресс | телефон | инн |

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

Таблица 9

|  |  |  |
| --- | --- | --- |
| Название | Атрибут | Тип |
| Код перераспределения | Код\_перераспределения | Int |
| Код сотрудника | Код\_сотрудника | Int |
| Перенаправлен от | Переренаправлен\_от | Int |
| Перенаправлен в | Перенаправлен\_в | Int |
| Дата перераспределения | Дата\_перераспределения | date |
| Код паспорта | Код\_паспорта | Int |
| серия | серия | Int |
| номер | номер | Int |
| Дата выдачи | дата\_выдачи | date |
| Код подразделения | код\_подразделения | INT |
| Код записи | Код\_записи | Int |
| Наименование льготы | наименование | Varchar(45) |
| Номер документа | номер\_документа | Varchar(45) |
| Дата выдачи документа | дата\_выдачи\_документа | date |
| Фамилия | Фамилия | Varchar(45) |
| Имя | Имя | Varchar(45) |
| Отчество | Отчество | Varchar(45) |
| Паспорт | Паспорт | int |
| Стаж | Стаж | Int |
| Адрес | адрес | Varchar(100) |
| Телефон | телефон | varchar(11) |
| ИНН | ИНН | varchar(11) |
| Код отдела | Код\_отдела | Int |
| Название отдела | Название отдела | Varchar(45) |
| Глава отдела | Глава отдела | int |
| Код должности | Код\_должности | int |
| Наименование должности | Наименование\_должности | Varchar(45) |
| оклад | оклад | Int |
| Стаж | Стаж | Int |
| статус | статус | Varchar(45) |
| Начало трудовой деятельности | начало\_трудовой\_деятельности | date |
| Конец трудовой деятельности | конец\_трудовой\_деятельности | date |
| Код вакансии | Код\_вакансии | Int |
| Дата объявления | Дата\_объявления | date |
| Дата окончания | Дата\_окончания | date |

#### 2.2 Создание концептуальной модели

В ходе работы с Case-инструментарием DRAW.IO была создана концептуальная модель будущей базы данных. Модель состоит из 8 сущностей и одной промежуточной таблицы, которая нужна для связи многие ко многим. 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 .png .

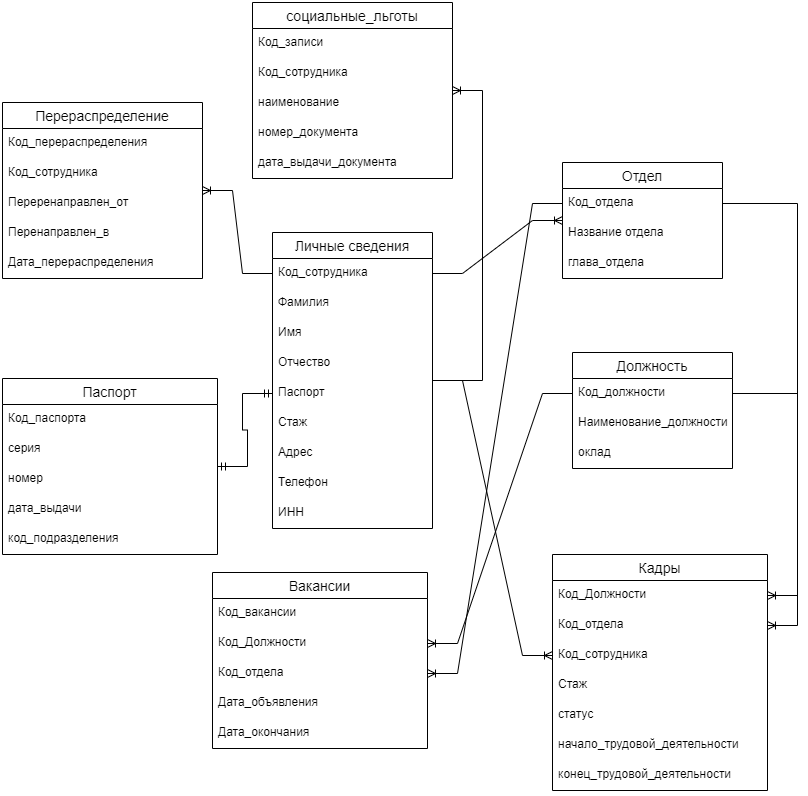


рис. Концептуальная модель магазина, созданная в draw.io

#### 2.3 Физическая модель

Физическая модель – логическая модель базы данных, выраженная в терминах языка описания данных конкретной СУБД.

Физическая модель базы данных содержит все детали, необходимые конкретной СУБД для создания базы: наименования таблиц и столбцов, типы данных, определения первичных и внешних ключей и т.п.

Физическая модель строится на основе концептуальной с учетом ограничений, накладываемых возможностями выбранной СУБД (в данном случае – My SQL).

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 представлена физическая модель.

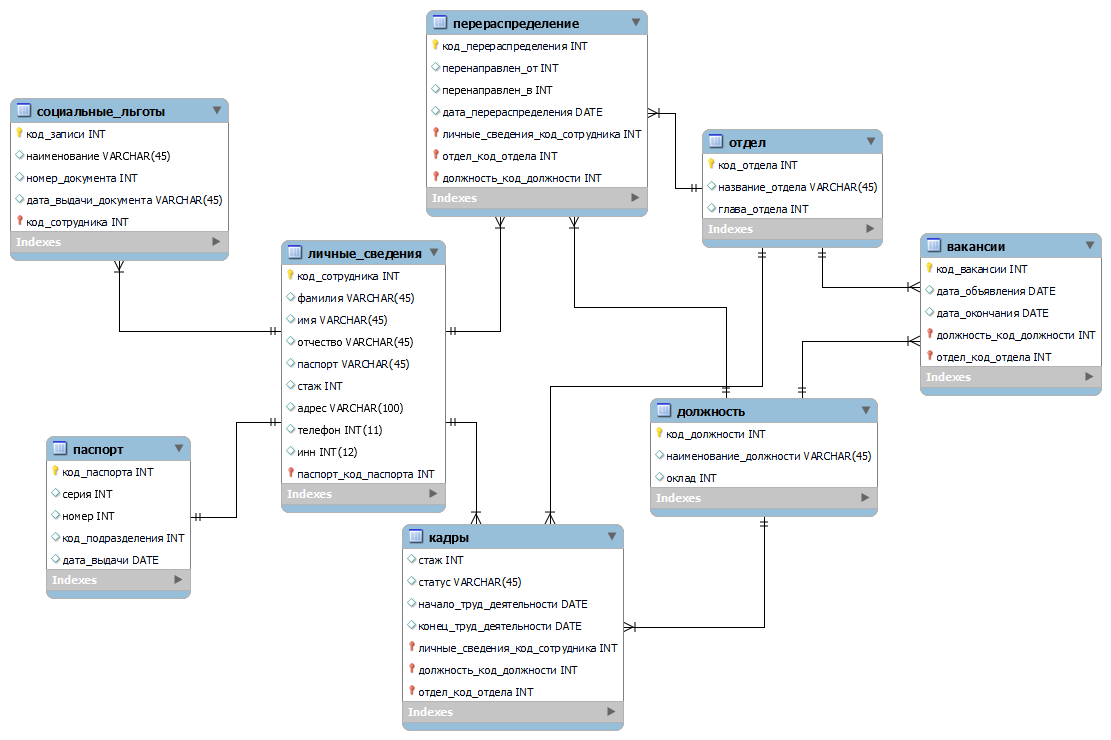


рис. Физическая модель в MySQL Workbench 8

#### 2.4 Описание связей 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.

В таблице 12 показаны все связи.

|  |  |  |
| --- | --- | --- |
| **Таблица** | **Связь** | **Таблица** |
| Личные\_сведения | Один ко многим | Социальные\_льготы |
| Личные\_сведения | Один к одному | паспорт |
| Личные\_сведения | Один ко многим | Кадры |
| Личные\_сведения | Один ко многим | перераспределение |
| Отдел | Один ко многим | перераспределение |
| Отдел | Один ко многим | Кадры |
| Отдел | Один ко многим | Вакансии |
| Должность | Один ко многим | перераспределение |
| Должность | Один ко многим | Кадры |
| Должность | Один ко многим | Кадры |

Дальше вся работа будет происходит на сервере.

#### 2.4 Заполнение данных

Заполнение данных будут выполняться без использования триггеров.

Заполнение таблиц. Рисунки 3 - 13.

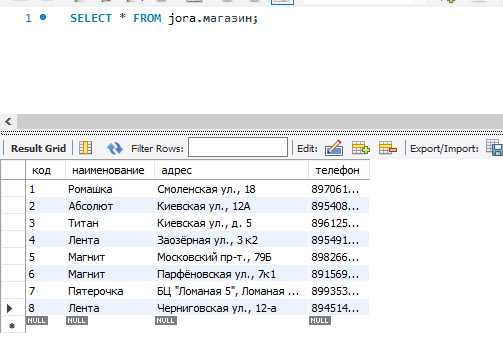


рис. Заполнение данных в таблицу магазин

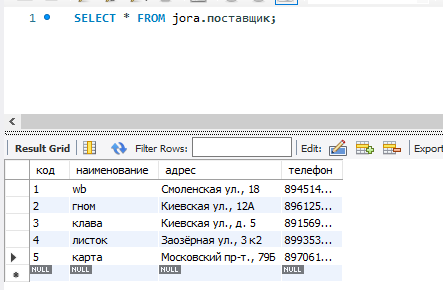


рис. Заполнение данных в таблицу поставщик

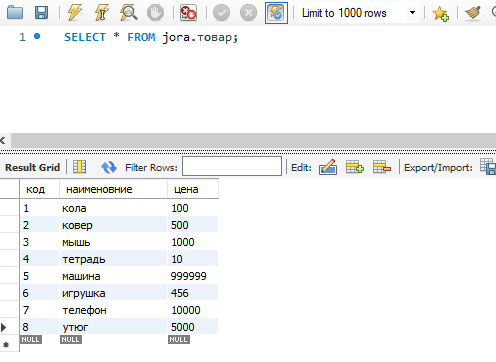


рис. Заполнение данных в таблицу товар

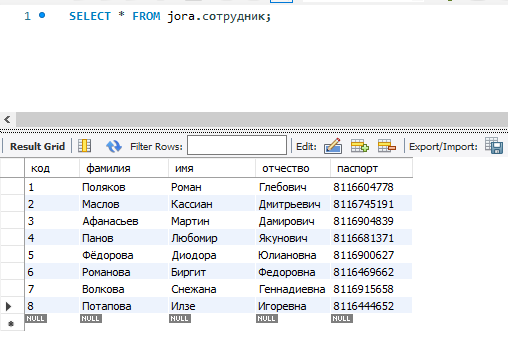


рис. Заполнение данных в таблицу сотрудник

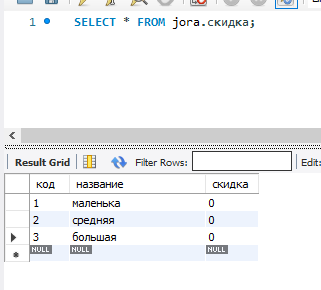


рис. Заполнение данных в таблицу скидка

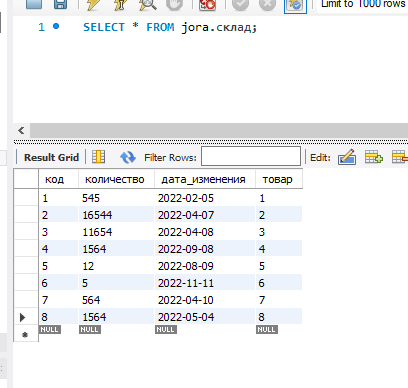


рис. Заполнение данных в таблицу склад

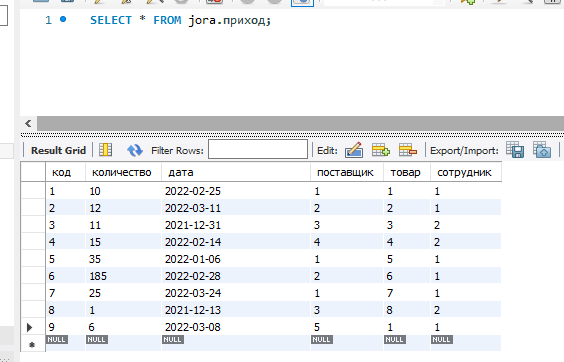


рис. Заполнение данных в таблицу приход

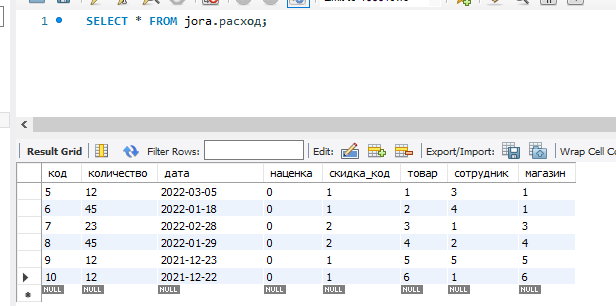


рис. Заполнение данных в таблицу расход

данные добавлялись с помощью запроса INSERT INTO рис.11 – 13

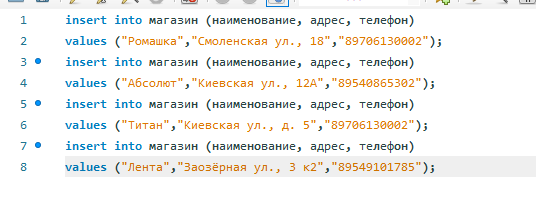


рис.

### Запросы

MySQL запрос – это обращение к базе данных MySQL, с помощью которого мы можем реализовать: получение, изменение, удаление, сортировку, добавление, и другие манипуляции с данными базы.

1. Запрос выводит все товары магазина. Рис.14

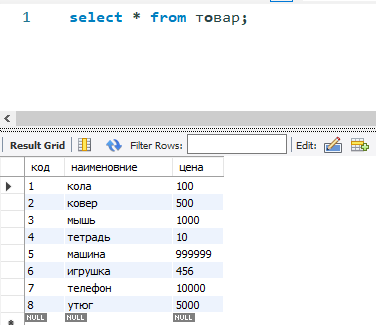


рис. Все товары

1. Запрос показывает количество сотрудников. Рис.15

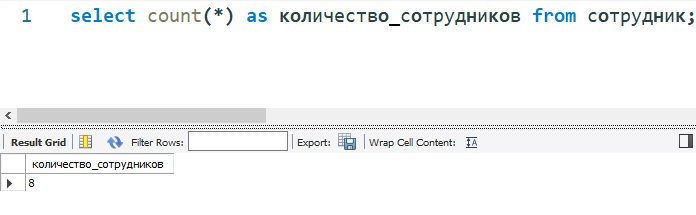


рис. количество сотрудников

1. Запрос, показывающий количество уникальных магазинов которые есть в расходе . Рис.16

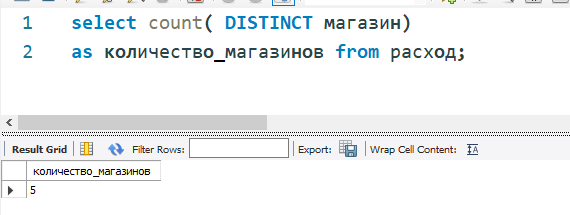


рис. количество магазинов

1. Запрос выводит товар, которого больше всего на складе. Рис.17

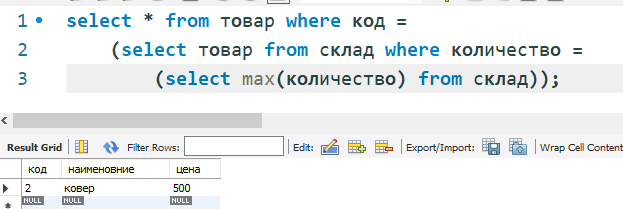


рис. товар, которого больше всего на складе

1. Запрос выводит фио сотрудников. Рис.18

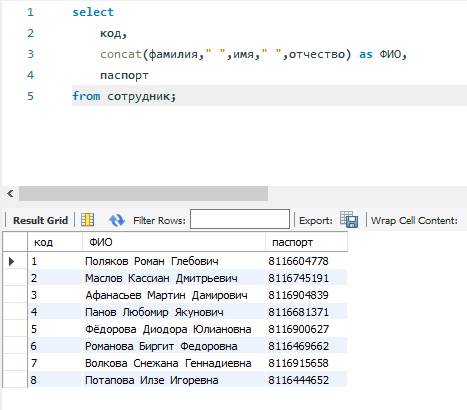


рис. фио сотрудников

1. Запрос количество записей, у которых нет скидка.Рис.19

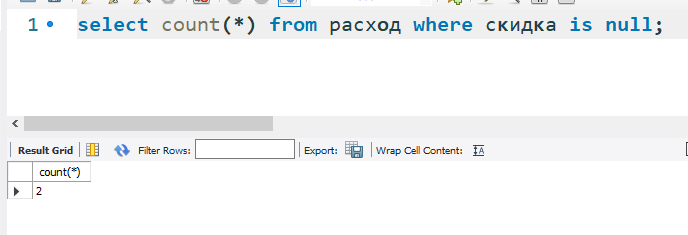


рис. расход без скидки

1. Запросы, выводящие всех поставщиков у которых наименование заканчивается на а Рис.20

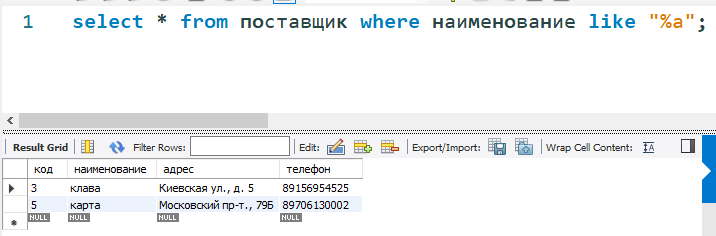


рис. наименование заканчивается на а

1. Запрос расход с окончательной ценой. Рис.21

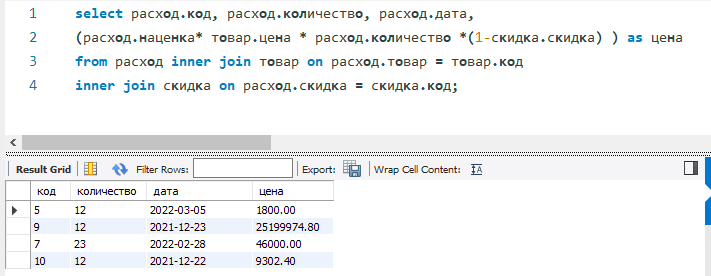


рис. расход с окончательной ценой

1. Запрос выводит приход, но заменяет коды. Рис.22

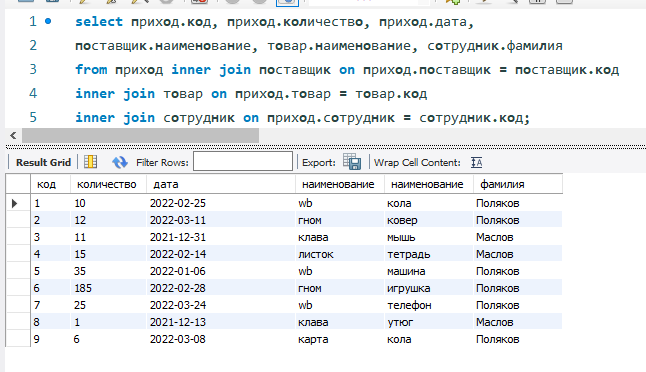


рис. приход

1. Запрос выводит день недели. Рис.23

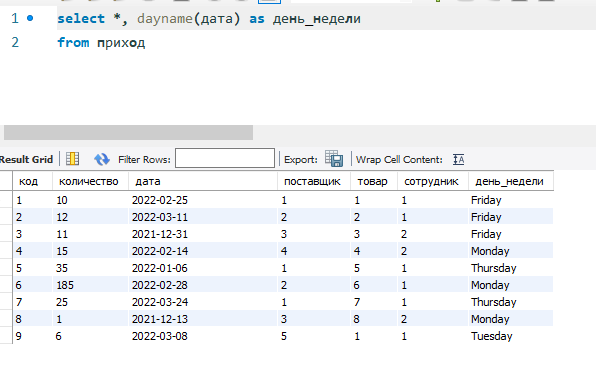


рис. День недели

1. Запрос выводит цену без скидки. Рис.24

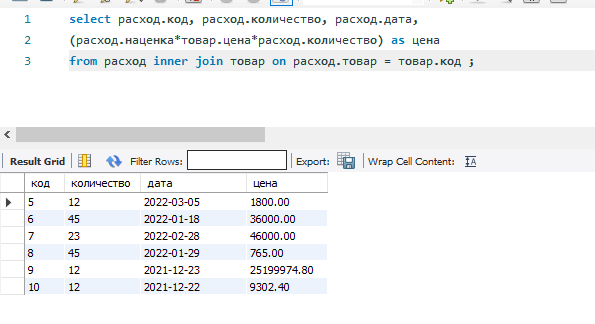


рис. Возраст клиентов

### процедуры

В MySQL 5 есть много новых функций, одной из самых весомых из которых является создание хранимых процедур.

Хранимая процедура - способ инкапсуляции повторяющихся действий. В хранимых процедурах можно объявлять переменные, управлять потоками данных, а также применять другие техники программирования.

Было создано 5 процедур.

1. Процедура – внесение данных о новом товаре. Рис.25-26.

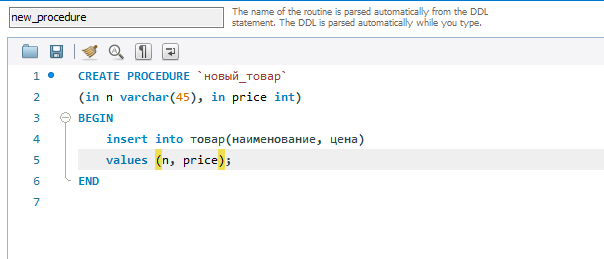


рис. Создание процедуры «внесение данных о новом товаре»



рис. Выполнение процедуры

1. Процедура внесение данных о сотруднике. Рис. 27-28.

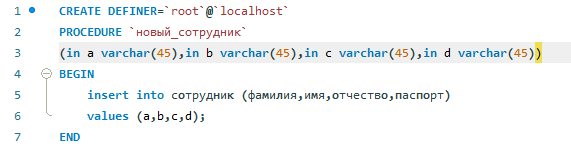


рис. Создание процедуры внесение данных о сотруднике

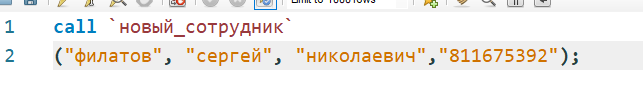


рис. Выполнение процедуры

1. Процедура, который ведёт подсчёт количества и стоимость товара определённого ассортимента. Рис.29-30

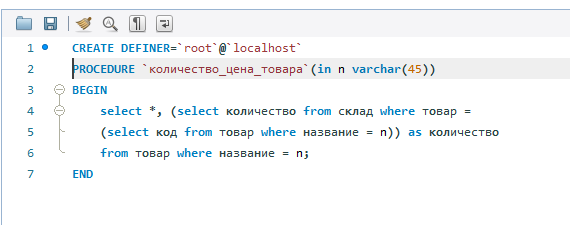


рис. Создание процедуры «Количество и цена товара»

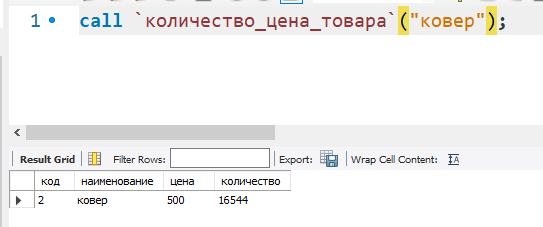


рис. Выполнение процедуры

1. Процедура вычисление количества сотрудников магазина в целом. Рис.31-32

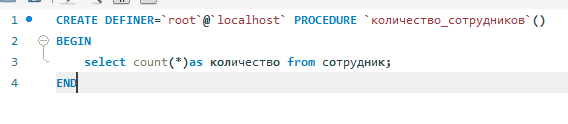


рис. Создание процедуры «Количество сотрудников»

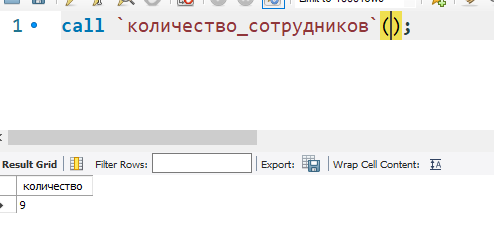


рис. Выполнение процедуры

1. Процедура на увольнение сотрудника. Рис.33-34.

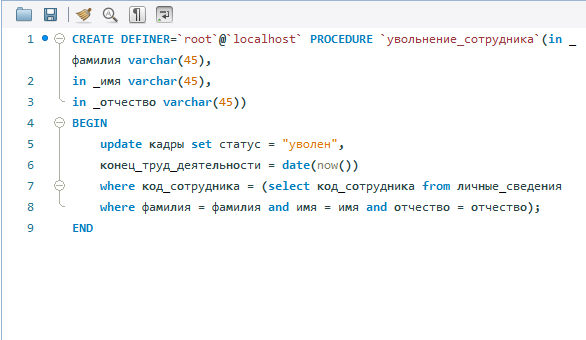


рис. Создание процедуры «увольнение\_сотрудника»



рис. Выполнение процедуры

### Функции

1. Функция «статус сотрудника». Рис.35-36.

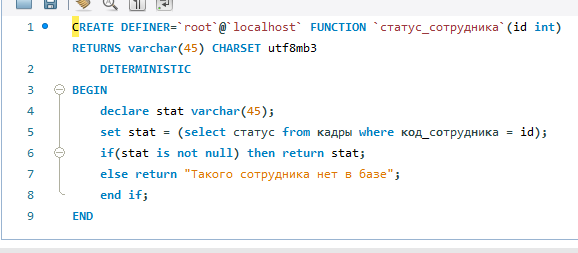


рис. Создание функции «статус сотрудника»

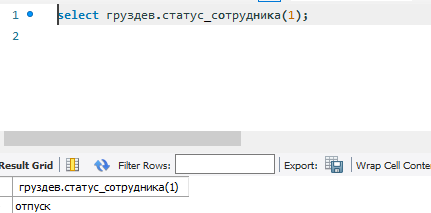


рис. Вызов функции

1. Функция, выводящая фио сотрудника. Рис.37-38

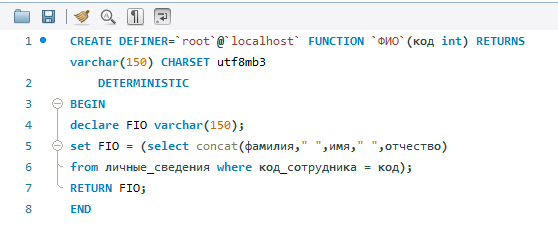


рис. Создание функции фио

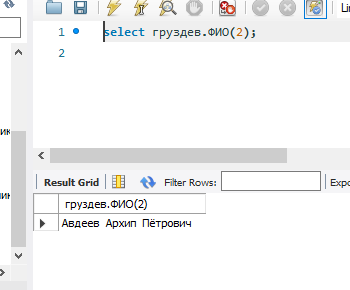


рис. Вызов функции

1. Функция, выводящая количество лет, между датами. Рис.39-40

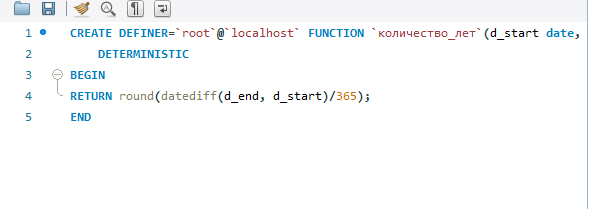


рис. Создание функции количество лет

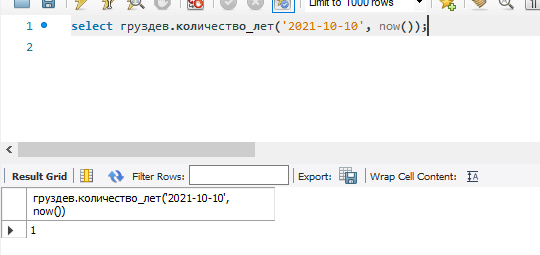


рис. Вызов функции

### триггеры

Триггер — это откомпилированная SQL-процедура, исполнение которой обусловлено наступлением определенных событий внутри реляционной базы данных. События, к которым можно привязывать триггеры в MySQL: INSERT, UPDATE, DELETE. Время, в которое будет исполнен триггер может быть: BEFORE (до наступления заданного события) или AFTER (после него).

1. Данный триггер обновит отдел в таблице кадры. Рис. 41 – 44.

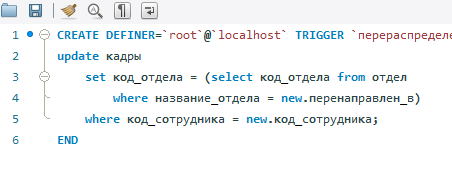


рис. Создание триггера «перераспределения»

Далее проверяется триггер.

Изначально код\_отдела у 1 записи равен 1. Рис.42

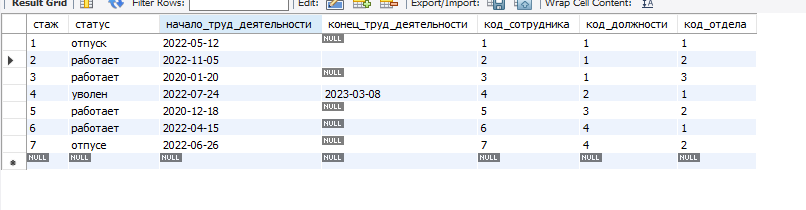


рис. Кадры

в таблицу перераспределение вводится новая запись. Рис.43

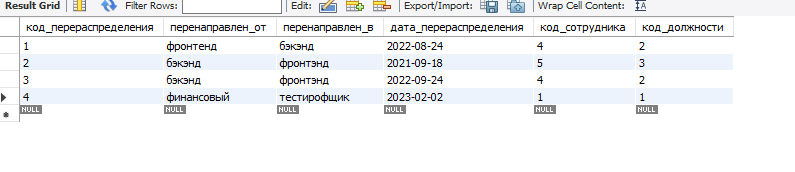


рис. таблица перераспределение

Дальше проверяется код\_отдела. Рис.44

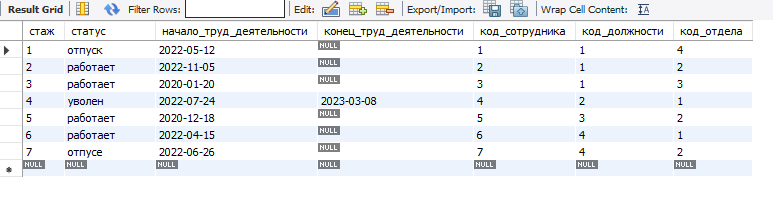


рис. Данные таблицы кадры

### Заключение

В курсовой работе создана база данных для склада.

В данной курсовой работе была спроектирована информационная система, автоматизирующая работу отдела кадров фирмы, реализующую следующие процессы: учет личных данных сотрудников, учет функций кадровой службы (прием нового сотрудника на работу, различные операции с сотрудником: увольнение, перемещения по отделам предприятия), учет приказов по контингенту (сотрудникам). Разработать хранимые процедуры: внесение данных о новом сотруднике отдела, хранение личных данных о сотруднике, изменение статуса сотрудника (работает, уволен в отпуске), вычисление количества сотрудников конкретного отдела и всего предприятия в целом, проверка наличия сотрудника в конкретном отделе

Также для автоматизации были разработаны: 5 процедур, 3 функции, 1 триггер. Также были проведены тесты.

### Список используемой литературы

1. Прокушев, Я. Е. Базы данных: практикум для студентов, обучающихся по специальностям и направлениям подготовки 09.00.00 «Информатика и вычислительная техника», 10.00.00 «Информационная безопасность», а также направлению подготовки 38.03.05 «Бизнес-Информатика» / Я. Е. Прокушев; под редакцией Т. С. Кулакова. — Санкт-Петербург: Интермедия, 2018. — 240 c. — ISBN 978-5-4383-0149-3. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: https://www.iprbookshop.ru/73639.html (дата обращения: 26.12.2021). — Режим доступа: для авторизир. Пользователей

2. Стасышин, В. М. Разработка информационных систем и баз данных: учебное пособие для СПО / В. М. Стасышин. — Саратов: Профобразование, 2020. — 100 c. — ISBN 978-5-4488-0527-1. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: https://www.iprbookshop.ru/87389.html (дата обращения: 26.12.2021). — Режим доступа: для авторизир. пользователей. - DOI: https://doi.org/10.23682/87389