



**Частное учреждение профессионального образования
«Высшая школа предпринимательства»
(ЧУПО «ВШП»)**

КУРСОВАЯ РАБОТА

«Разработка базы данных для магазина продуктов»

Выполнил:

студент 3-го курса специальности

09.02.07 «Информационные системы и
программирование»

Медведев Семён Сергеевич

подпись: _____

Проверил:

преподаватель дисциплины,

преподаватель ЧУПО «ВШП»

к.ф.н Ткачёв П.С.

оценка: _____

подпись: _____

Тверь 2025

Оглавление

Введение.....	3
Глава 1	4
Рассмотрт магазина продуктов как объект	4
Требования к базе данных	7
Выбор СУБД	8
Глава 2	13
Построение схем.....	13
Возможности MySQL Workbench.....	13
ER диаграмма	14
Реализация процессов	19
Типовые запросы:	20
Заключение.....	23
Список Литературы.....	24
Приложение 1	25
Приложение 2	29
Приложение 3	31
Приложение 4	32
Приложение 5	33

Введение

Актуальность

В наше время на каждом углу практически стоят продуктовые магазины. И все прекрасно знают как происходит продажа товаров.

Но скорее всего не многие знают, как устроен сам процесс. Поэтому я считаю, что продуктовый магазин “Бублик” покажет как и что происходит.

Определение цели работы

Тема моей курсовой работы “Разработка базы данных для магазина продуктов”.

Из темы работы мы явно понимаем, что главной целью у нас является разработка своей базы данных.

Постановка задач

1. Установить требования к разрабатываемой базе данных
2. Выбор СУБД
3. Визуализация схемы БД
4. Разработать ER - диаграмму, в которой будут находиться n -количество таблиц, а так же связи между ними
5. Заполнить таблицы данными

Объект исследования

В данной курсовой работе объектом исследования будет являться пошаговая разработка базы данных для “Магазина продуктов Бублик”

Глава 1

Рассмотрт магазина продуктов как объект

Для начала мы рассмотрим магазин как объект, дабы понять его функциональность и на основе этого выстроить ряд требований.

В магазинах продуктов, всегда много людей которые там работают. Мы предположим что в магазинах есть n - количество должностей.

Предположительно в магазине 4 должности: администратор, продавец, кладовщик кассир.

Рассмотрим каждую должность отдельно:

Администратор — данная должность в магазине является главной, поскольку этот человек следит за порядком и за работой своего коллектива. Администратор выполняет роль координатора всех бизнес-процессов и несёт ответственность за общие результаты деятельности магазина.

Основные функции администратора включают:

Планирование рабочих смен и составление графиков работы персонала

Контроль выполнения планов продаж и достижения ключевых показателей эффективности

Решение конфликтных ситуаций с покупателями и координация работы с жалобами

Взаимодействие с поставщиками и контроль соблюдения договорных обязательств

Анализ финансовых показателей и подготовка отчётности для руководства

Обеспечение соблюдения санитарных норм и требований безопасности

Координация процедур инвентаризации и контроль товарных остатков

Разработка и внедрение мер по повышению эффективности работы магазина

Продавец — данная должность уникальная, ведь она в себе включает обширный спектр работы. Он может выставлять товар, стоять за кассой, а также помогать кладовщику. Продавец является ключевой фигурой в формировании клиентского опыта и непосредственно влияет на лояльность покупателей.

Обязанности продавца:

Консультирование покупателей по ассортименту товаров и их характеристикам

Поддержание привлекательного внешнего вида торгового зала и выкладки товаров

Контроль сроков годности продукции и своевременное изъятие просроченных товаров

Ведение учёта движения товаров и информирование кладовщика о необходимости пополнения

Поддержание чистоты и порядка в торговом зале

Участие в процедурах приёмки товаров от поставщиков

Кладовщик — данная должность ответственна за скрытую часть магазина. Исходя из названия понятно, что по большей части своей работы занимается складом. Кладовщик обеспечивает бесперебойное функционирование цепи поставок и поддерживает оптимальный уровень товарных запасов.

Функции кладовщика:

Приёмка товаров от поставщиков с проверкой количества, качества и соответствия документам

Организация рационального размещения товаров на складе с учётом принципов логистики

Ведение складского учёта и контроль движения товарно-материальных ценностей

Обеспечение соблюдения температурных режимов хранения для различных категорий товаров

Проведение регулярных инвентаризаций и выявление расхождений в учёте

Подготовка товаров к перемещению в торговый зал согласно принципу ротации

Взаимодействие с транспортными компаниями и экспедиторами при получении грузов

Поддержание санитарного состояния складских помещений

Кассир — данная должность может как только выполнять одну функцию — стоять за кассой, но также может меняться заданиями с продавцом. Кассир занимает стратегически важную позицию, поскольку завершает цикл обслуживания покупателя и формирует финальное впечатление о магазине.

Обязанности кассира:

Обслуживание покупателей на кассовом узле с использованием современного оборудования

Работа с различными формами оплаты: наличные, банковские карты, электронные платежи

Ведение точного учёта денежных средств и подготовка кассовых отчётов

Консультирование покупателей по акциям, скидкам и программам лояльности

Контроль правильности применения цен и скидок при оформлении покупок

Участие в процедурах инкассации и передачи выручки

Поддержание порядка в кассовой зоне и обеспечение комфортных условий для покупателей

Взаимодействие с системами фискального учёта и соблюдение требований налогового законодательства

Рассмотрев персонал, теперь рассмотрим сам магазин продуктов “Бублик”. Все магазины начинают с маленького товара -оборота и естественно с маленькой выручки, но если набрать опытных сотрудников и не очень, то магазин в большой скорости наберёт популярность, а в месте с ней придёт и выручка. Даже если в наш магазин не получается набрать опытных работников, тогда мы пойдём путём маркетинга. Ведь чем больше реклама магазина будет появляться на глаз у людей, тем быстрее он принесёт свой окуп, и тогда не важно будет есть ли там опытный состав или нет его.

Требования к базе данных

1. Создать реляционную базу данных.
2. В базе данных должны находиться таблицы, для упрощения связи.(перефразировать)
3. Таблицы должны быть связаны между собой, для упрощения взаимодействия

В своём магазине я могу использовать разные базы данных, но в таких сферах больше всего используются реляционные и NoSQL , базы данных.

Теперь рассмотрим каждую из них и выявим в каждой плюсы и минусы.

Реляционная база - это базы использующие таблицы в которых есть строки и столбцы.

Плюсы данной базы данных:

1. Согласованность и целостность данных
2. Точность данных
3. Безопасность
4. Доступ для нескольких сотрудников одновременно

Минусы:

1. Для обеспечения целостности данных используются длительные и кратковременные блокировки
2. Требуется большой объём физической памяти
3. Невысокая производительность

NoSQL - подходит для неструктурированных баз

Плюсы данной базы данных

1. Гибкость модели данных
2. Масштабируемость и распределённость
3. Высокая производительность
4. Поддержка различных типов данных
5. Минусы:
6. Ограниченная поддержка SQL
7. Сложность запросов
8. Недостаточная зрелость некоторых систем
9. Сниженная надёжность транзакций

Выбор СУБД

Рассматривая разные реляционные базы данных, я прихожу к выводу что MySQL подойдёт лучше всего. MySQL является простой СУБД подходящей для обширных запросов, и удобна в использовании с разными языками программирования.

1. MySQL оптимизирован для быстрой обработки запросов, и подходит для работ где используются большие объёмы. Так же он поддерживает

эффективные механизмы индексирования, что значительно ускоряет выполнение операций.

2. MySQL является одной из самых надёжных СУБД. Она обеспечивает целостность данных благодаря ACID - транзакций.

3. MySQL позволяет использовать вертикальное и горизонтальное масштабирование.

4. В MySQL используется понятный и простой синтаксис, которое поддерживает множество типов данных (JSON, геоданные, временные ряды)

5. MySQL используется на различных операционных системах и поддерживает разные языки программирования (Java, C++, Pethon)

6. MySQL благодаря своей популярности имеет большую документацию, где имеются много ответов на разные виды ошибок, а так же имеют готовые решения

Всё что я перечислил выше является только маленькой частью плюсов MySQL, и опираясь на выше изложенное ещё раз приходим к выводу, что данная СУБД является простой в использовании. Но так же не стоит забывать, что все программы не идеальны, поэтому рассмотрим теперь минусы.

1. Как я говорил ранее MySQL хорошо оптимизирована для быстрой обработки запросов, но она не подходит для каких-то сложных аналитических запросов

2. В масштабирование также есть маленькие проблемы, а именно при использовании горизонтального масштабирование, не получится настроить его автоматически, это придётся всё делать в ручную, но это тоже не так удобно как в других системах

3. Есть некоторые неудобства в использование JSON в MySQL. Ведь в выбранной мною СУБД работа JSON значительно медленнее чем в других Бд

4. Сложность настройки MySQL. Ведь если мы хотим, чтоб всё работало быстро, нужно сидеть и настраивать

5. Хотя MySQL и имеет бесплатные версии, но для коммерческих целей MySQL ограничивают, вводя платные пакеты

6. Так же MySQL при больших нагрузках может оставлять lag, чтоб такого не было требуются дополнительные решения

Теперь мы рассмотрели некоторые плюсы и минусы данной СУБД. Далее мы рассмотрим несколько областей где используется MySQL Workbench.

1. MySQL Workbench чаще всего используют для разработки и проектирования базы данных. Проще всего составлять ER - диаграммы, ведь в них сразу указываются сколько таблиц, какие там будут столбцы, а так же указывается связь между столбцами.

2. Также MySQL Workbench используется для администрирования серверов. Ведь там можно прописать какая роль какие данные может видеть та или иная роль в данной СУБД

3. MySQL Workbench помогает в оптимизации SQL - запросов. Опираясь на ER - диаграммы, он их считывает и выстраивает порядок чтения таблиц и затраченное время на их воспроизведения.

Так как у меня есть предположительно магазин товаров “Бублик”, то там я могу использовать как MySQL так и NoSQL.

Для начала рассмотрим их структуры, после чего придём к выводу какая же всё таки база данных мне больше подходит.

Структура MySQL заключается в том, что все данные хранятся в таблицах с определёнными столбцами и строками. Также для их взаимодействия используются специальные связи с помощью которых можно посмотреть какая таблица и какой столбец с чем связаны и что от чего зависит.

Структура NoSQL заключается в том что в ней нет никаких таблиц и связей. Она хранит все данные в виде документов.

Масштабируемость MySQL обычно это вертикальная она работает за счёт увлечения мощности сервера

Масштабируемость NoSQL обычно горизонтальная и она работает за счёт добавление серверов и распределение равномерных нагрузок на каждый из них

Производительность MySQL подходит больше всего для сложных запросов, ведь ей это позволяет мощност SQL запросов

Производительность NoSQL так же можно использовать для сложных запросов, но есть некоторые условия. Они должны быть структурированными и распределены на разные сервера.

Что должно быть реализовано в магазине продуктов ?

Собранные данные позволяют мне перейти к созданию базы данных.Список реализаций:

1. Регистрация новых сотрудников, имение их должностей,а так же их удаление
2. Просмотр количества товаров на кладе
3. Заказ товаров у поставщика

Исходя из всего рассмотренного, я останавливаюсь на выборе MySQL так как рассмотрев все плюсы и минусы данной СУБД. Она нам подходит для реализации наших требований.

Вывод

Проведенный анализ магазина продуктов "Бублик" как объекта автоматизации показал, что данное предприятие представляет собой сложную систему взаимосвязанных процессов, требующих координации и оптимизации. Существующие методы управления, основанные на ручном труде и простейших инструментах, не позволяют в полной мере использовать потенциал бизнеса и создают препятствия для его развития.

Внедрение современной информационной системы управления на базе реляционной базы данных позволит не только автоматизировать рутинные

операции, но и создать основу для стратегического развития магазина. Система должна учитывать специфику розничной торговли продуктами питания, обеспечивать надежность и безопасность данных, а также предоставлять удобные инструменты для работы персонала различной квалификации.

Таким образом, магазин "Бублик" является подходящим объектом для внедрения информационной системы, которая поможет повысить эффективность работы, улучшить качество обслуживания покупателей и создать конкурентные преимущества на рынке розничной торговли.

Глава 2

Построение схем

В начале своей работы я выяснял какая СУБД лучше будет мне подходить для реализации моего проекта, и пришёл к выводу, что MySQL Workbench подходит как нельзя лучше.

MySQL Workbench - это инструмент которые включает себя много функций, одними из которых являются:

Визуальное проектирование - составление ER диаграм

Создание БД - генерация кода из ER диаграм

Эксплуатация БД - доступ к администрированию и настройки БД

Возможности MySQL Workbench

1. Моделирование данных - включает в себя несколько аспектов, а именно создание ER диаграмм, добавление таблиц, определение связей, установка ключей.

Теперь рассмотрим всё по отдельности.

1. ER диаграммы - включает в себя визуальную наглядность базы данных.

2. Добавление таблиц - позволяет создавать имя таблицы, а так же её колонки, так же может указать тип данных и их атрибуты.

3. Определение связей - есть 3 основных типа связи, один к одному, один ко многим и многие ко многим.

4. Установка ключей - показывает уникальность каждой записи

2. Администрирование - включает в себя также несколько аспектов настройка сервера, управление пользователями, экспорт и импорт данных, мониторинг производительности.

1. Настройка сервера - она позволяет просматривать все настройки сервера, а также смотреть логи сервера.

2. Управления пользователями - позволяет создавать/удалять пользователей, так же может менять права для разных пользователей

3. Экспорт и импорт данных - позволяет копировать разные схемы с данными и вставлять их в другие БД

4. Мониторинг производительности - как понятно из названия, то можно просматривать производительность

3. Разработка - в MySQL Workbench есть визуальный редактор, который позволяет создавать, выполнять и редактировать запросы, так же есть возможность управлять подключением. Ещё есть обозреватель объектов, который позволяет перемещаться между таблицами и изменять, создавать и удалять их.

ER диаграмма

ER диаграмма - это схема в которой отображаются все сущности и как они между собой связаны.

В своей ER диаграмме я разработал 6 таблиц, которые будут отображать сущность моей работы.

1. “tovar” - данная таблица показывает сущность товара. В ней содержатся данные о наименовании товара, сроке и условия хранения.

2. “postavka” - данная таблица показывает сущность поставки. В ней содержатся наименования поставки, адрес по которому надо доставить товар, а так же телефон фирмы, которая занимается поставками.

3. “torgovlya” - данная таблица показывает сущность торговли. В ней содержатся данные о коде товара (предположим, что у нас маркированный товар) дата продажи товара, а так же содержит столбец о количестве проданного товара.

4. “sklad” - данная таблица показывает сущность склада. В ней содержится данные о коде накладной и о остаточном количестве товара на складе.

5. “nakladnaya” - данная таблица показывает сущность накладной. В ней содержится код товара, дата изготовления, стоимость закупа определённого товара, количество продуктов, которое приехало от закупщика.

6. “sotrydniki” - данная таблица показывает сущность сотрудников. В ней содержатся данные должности имена, фамилии, отчества сотрудников.

Воспользовавшись сущностями, которые указаны выше получилась такая ER-диаграмма

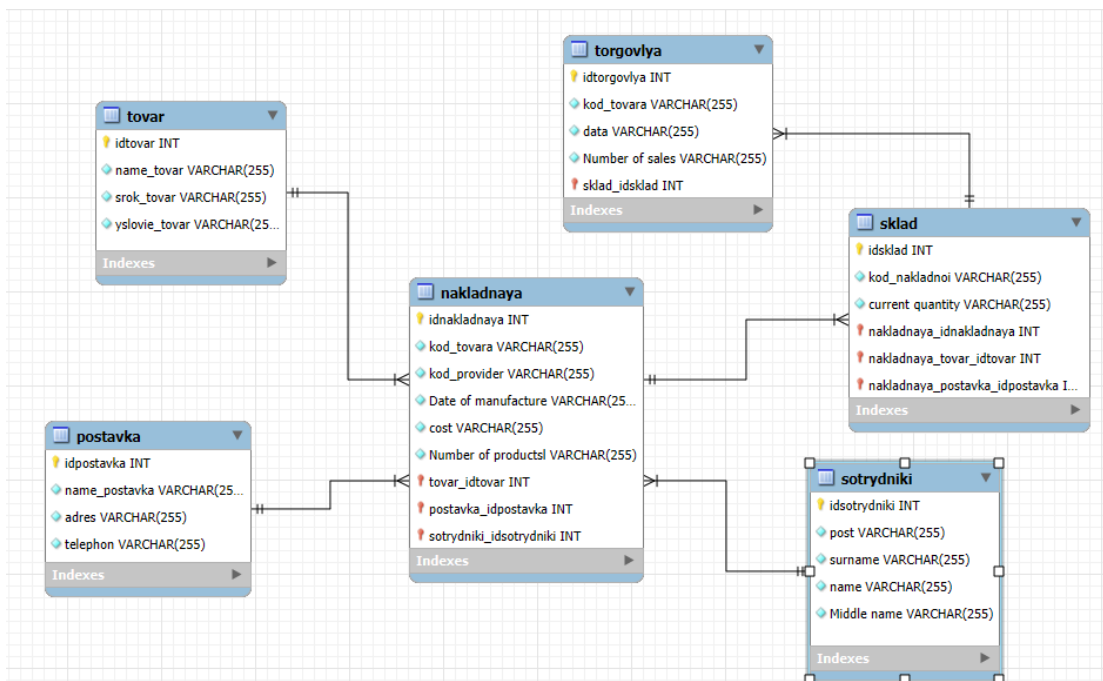


Рисунок1

Таблица nakladnaya

Данная таблица является главной, ведь в ней содержатся практически все данные.

Начиная от кода накладной и заканчивая нумерацией товара

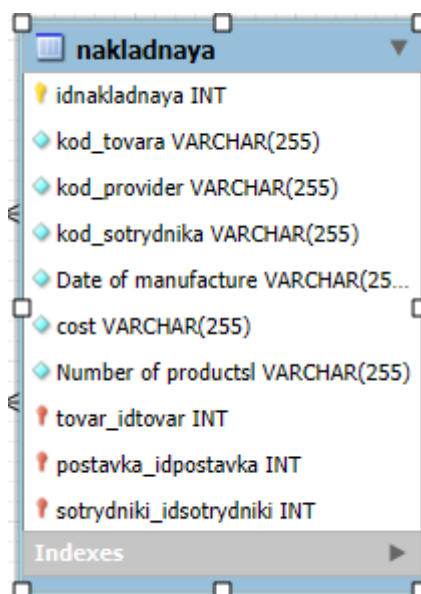


Рисунок2

Таблица `nakladnaya` содержит в себе колонки :
`idnakladnaya`, `kod_tovara`, `kod_provider`, `kod_sotrydnika`, `Date of manufacture`, `cost`,
`Number of productsl`

1. Для поля `idnakladnaya` я выбрал тип данных `INT`, так как он является первичным ключом.
2. Для поля `kod_tovara` был выбран тип данных `VARCHAR(255)`, так как он будет содержать в себе номер накладной. Данный номер может состоять как из букв, так и из цифр
3. Для поля `kod_sotrydnika` был выбран тип данных `VARCHAR(255)`, из-за того что в коде сотрудника находится 20 символом
4. Для поля `Date of manufacture` был выбран тип данных `VARCHAR(255)`, потому что в дате изготовления товара используются не только цифры, но и другие знаки по типу точки или двоеточия
5. Для поля `cost` был выбран тип данных `VARCHAR(255)`, так как при покупке товаром может выходить разная сумма и указывать другой тип данных, здесь попросту не имеет смысла
6. Для поля `Number of productsl` был выбран тип данных `VARCHAR(255)`, так как у меня учитывается за 1 единицу не коробка, а каждый товар в отдельности

Таблица `postavka`

Данная таблица полностью отвечает за поставку товара в магазин “Бублик”

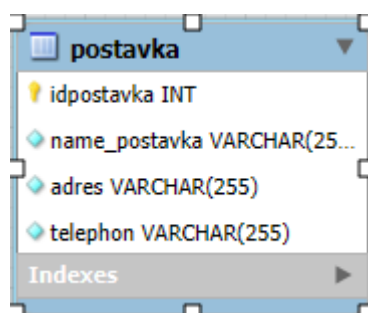


Рисунок3

Таблица `postavka` содержит в себе колонки: `idpostavka`, `name_postavka`, `adres`, `telephon`

1. Для поля `idpostavka` был выбран тип данных `INT`, так как он является первичным ключом для данной таблицы
2. Для поля `name_postavka` был выбран тип данных `VARCHAR(255)`, так как наименование поставки может включать все разные символы, это могут быть не только цифры и буквы, а так же и другие знак.
3. Для поля `adres` был выбран тип данных `VARCHAR(255)`, так как названия улиц и городов могут иметь разное количество символов.
4. Для поля `telephon` был выбран тип данных `VARCHAR(255)`, так как это номер телефона

Таблица `tovar`

В этой таблице мы можем увидеть все данные о товаре

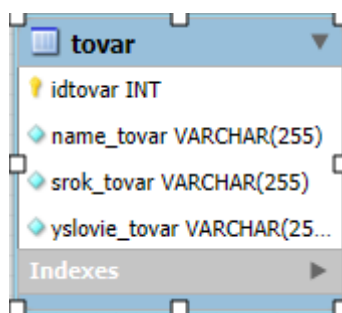


Рисунок4

Таблица `tovar` содержит в себе колонки: `idtovar`, `name_tovar`, `srok_tovar`, `yslovie_tovar`

1. Для поля `tovar` был выбран тип данных `INT`, так как он является первичным ключом для данной таблицы
2. Для поля `name_tovar` был выбран тип данных `VARCHAR(255)`, так как наименование товара может быть разной длины
3. Для поля `srok_tovar` был выбран тип данных `VARCHAR(255)`, так как срок товара как я описывал раньше состоит не только из цифр, но и других символов
4. Для поля `yslovie_tovar` был выбран тип данных `VARCHAR(255)`, так как в условие товара прописывается большой текст и на каждом товаре он может быть разным

Таблица sklad

Данная таблица отвечает за склад магазина “Бублик”

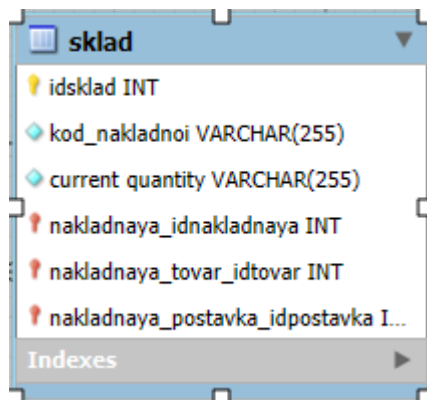


Рисунок5

Таблица sklad содержит в себе колонки: idsklad,kod_nakladnoi,current quantity

1. Для поля sklad был выбран тип данных INT, так как он является первичным ключом для данной таблицы
2. Для поля kod_nakladnoi был выбран тип данных VARCHAR(255), так как он может быть разной длины и содержать в себе разные символы
3. Для колонки current quantity был выбран тип данных VARCHAR(255), так как остаток товара может быть разным и не имеет смысла устанавливать какие-либо границы

Таблица sotrydniki

В этой таблице можно посмотреть все данные о сотрудниках магазина

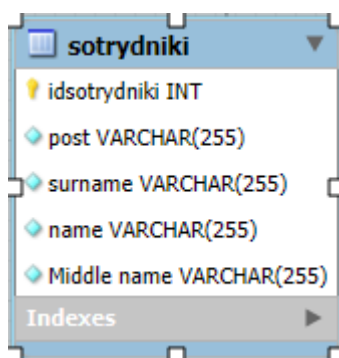


Рисунок6

Таблица sotrydniki содержит в себе колонки:idsotrydniki, post,surname, name ,middle name

1. Для поля `idsotrydniki` был выбран тип данных `INT`, так как он является первичным ключом для данной таблицы
2. Для поля `post` был выбран тип данных `VARCHAR(255)`, так как наименование должности разны по символам
3. Для поля `surname` был выбран тип данных `VARCHAR(255)`, так как у всех людей разные фамилии и содержат разное количество символов
4. Для поля `name` был выбран тип данных `VARCHAR(255)`, так как тут такая же ситуация как и с фамилиями
5. Для поля `Middle name` был выбран тип данных `VARCHAR(255)`, так как имеет разное количество символов

Связь в таблицах

Разрабатывая свою базу данных с использованием ER - диаграммы я использовал связь между таблицами "один ко многим". Ведь рассматривая мой магазин она подходит лучше всего.

Как были реализованы все таблицы смотрите (приложение 1)

Реализация процессов

1. Типовые запросы:

`SELECT` - этот запрос позволяет извлекать данные как из одной таблицы так и из множества

`VALUES` - этот запрос позволяет указывать значения и может их использовать во время запроса

`FROM` - этот запрос указывает таблицы откуда берутся данные

`JOIN` - этот запрос объединяет данные из разных таблиц

`WHERE` - фильтрует данные

2. Транзакции - это последовательность выполнения операций, они либо выполняются все успешно, либо не выполняются вообще (приложение 2)

3. Представление - это таблица, которая хранит в себе данные, которые выполняются запросом. В представлении данные никакие не хранятся, они там появляются только в момент запроса (приложение 3)

4. Триггер - это процедура, которую пользователь не вызывает

5. Пользовательская функция - это функция которую создаёт пользователь и выполняет определённую операцию (приложение 4)

6. Обработчик исключений - это блок кода, который находит ошибки и реагирует на них(приложение 5)

7. Создание ролей

Мы разобрались что и как работает, а теперь посмотрим это на практике

Типовые запросы:

Вывести всех поставщиков

```
SELECT * FROM mydb.postavka;
```

добавление поставщиков

```
INSERT INTO mydb.postavka (idpostavka, name_postavka, adres, telephon)
```

```
VALUES (1, 'Поставщик АО Ромашка', 'г. Москва, ул. Примерная, д.10', '+7 (495) 123-45-67');
```

Вывести все товары

```
SELECT * FROM mydb.tovar;
```

добавление товара

```
INSERT INTO mydb.tovar (idtovar, name_tovar, srok_tovar, yslovie_tovar)
```

```
VALUES (1, 'Молоко 3.2%', '7 дней', 'Хранить при температуре от +2 до +6°C');
```

Получить список всех накладных с привязанными товарами, поставщиками и сотрудниками

```
SELECT
```

```
    n.idnakladnaya,
```

```
    t.name_tovar,
```

```
    p.name_postavka,
```

```

s.surname,
s.name,
s.`Middle name`,
n.`Date of manufacture`,
n.cost,
n.`Number of products`
FROM mydb.nakladnaya n
JOIN mydb.tovar t ON n.tovar_idtovar = t.idtovar
JOIN mydb.postavka p ON n.postavka_idpostavka = p.idpostavka
JOIN mydb.sotrydniki s ON n.sotrydniki_idsotrydniki = s.idsotrydniki;
Добавление сотрудников в таблицу
INSERT INTO mydb.sotrydniki (idsotrydniki, post, surname, name, `Middle name`)
VALUES
(1, 'Менеджер', 'Иванов', 'Алексей', 'Петрович'),
(2, 'Кладовщик', 'Петрова', 'Мария', 'Игоревна'),
(3, 'Бухгалтер', 'Сидоров', 'Дмитрий', 'Алексеевич');
Просмотр всех сотрудников:
SELECT * FROM mydb.sotrydniki;

```

4. Триггер

```

DELIMITER $$
CREATE TRIGGER trg_after_insert_torgovlya
AFTER INSERT ON mydb.torgovlya
FOR EACH ROW
BEGIN
    DECLARE msg VARCHAR(255);
    SET msg = CONCAT('Продажа товара с кодом ', NEW.kod_tovara, '
зарегистрирована.');
```

```

    SELECT msg AS message;
END$$

```

DELIMITER

7.Создание ролей

Создадим роль Администратор и покупатель

Администратор

-- Создание роли admin

CREATE ROLE 'admin';

-- Назначение всех привилегий на всю базу данных

GRANT ALL PRIVILEGES ON mydb.* TO 'admin';

Покупатель

-- Создание роли pokupatel

CREATE ROLE 'pokupatel';

-- Назначение прав на чтение определённых таблиц

GRANT SELECT ON mydb.tovar TO 'pokupatel';

GRANT SELECT ON mydb.sklad TO 'pokupatel';

GRANT SELECT ON mydb.torgovlya TO 'pokupatel';

Заключение

Вывод о работе с MySQL

Работа в MySQL Workbench относительно не сложная, когда есть хотябы минимальный набор знаний. В своей работе я попробовал решить несколько определённых задач, и хочу сказать что MySQL Workbench очень удобная и гибкая. Ведь в ней есть понятный интерфейс и помощь в упрощении задач.

Выполнение цели и задачи

Цель - этой работы заключалась в разработке базы данных для магазина продуктов "Бублик", так же в самом начале были поставлены задачи, которые выполнены успешно

1. Разработать ER - диаграмму, создать таблицы с определёнными данными, а так же указать между ними связь
2. Так же нужно было реализовать определённый функционал, а именно: типовые запросы, транзакцию, хранимую процедуру, триггер, пользовательскую функцию и обработчик исключений

Список Литературы

- 1.Бейли Б. MySQL 8.0: руководство для начинающих / Б. Бейли. — СПб. : Питер, 2020. — 352 с.
- 2.Горев Д. В. Проектирование баз данных : учебное пособие / Д. В. Горев. — М. : Академия, 2018. — 240 с.
- 3.Кузнецов М. С. Основы реляционных баз данных и SQL / М. С. Кузнецов. — М. : БХВ-Петербург, 2019. — 416 с.
- 4.Шаронов В. Г. MySQL : практическое руководство / В. Г. Шаронов. — М. : ДМК Пресс, 2021. — 288 с.
- 5.Хорстманн К., Корнелл Г. Java и базы данных : программирование и взаимодействие / К. Хорстманн, Г. Корнелл. — М. : Диалектика, 2017. — 560 с.
- 6.Документация по MySQL Workbench [Электронный ресурс]. — Oracle Corporation, 2024. — Режим доступа: <https://dev.mysql.com/doc/workbench/en/>, свободный. — (дата обращения: 24.06.2025).
- 7.Вершинин А. Системы управления базами данных : конспект лекций / А. Вершинин. — М. : Инфра-М, 2020. — 208 с.
- 8.Столбов В. А. Основы информационных систем и технологий / В. А. Столбов. — М. : Юрайт, 2021. — 352 с.
- 9.Date С. J. Введение в системы баз данных / С. J. Date ; пер. с англ. — М. : Вильямс, 2019. — 1024 с.
- 10.SQL Tutorial [Электронный ресурс]. — W3Schools. — Режим доступа: <https://www.w3schools.com/sql/>, свободный. — (дата обращения: 24.06.2025).

Приложение 1

Реализация БД

```
-- MySQL Script generated by MySQL Workbench
-- Sun May 18 15:52:48 2025
-- 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_I
N_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_S
UBSTITUTION';

CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8 ;
USE `mydb` ;

CREATE TABLE IF NOT EXISTS `mydb`.`postavka` (
  `idpostavka` INT NOT NULL,
  `name_postavka` VARCHAR(255) NOT NULL,
  `adres` VARCHAR(255) NOT NULL,
  `telephon` VARCHAR(255) NOT NULL,
  PRIMARY KEY (`idpostavka`))
ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `mydb`.`tovar` (
  `idtovar` INT NOT NULL,
  `name_tovar` VARCHAR(255) NOT NULL,
  `srok_tovar` VARCHAR(255) NOT NULL,
  `yslovie_tovar` VARCHAR(255) NOT NULL,
  PRIMARY KEY (`idtovar`))
ENGINE = InnoDB;
```

```

CREATE TABLE IF NOT EXISTS `mydb`.`sotrydniki` (
  `idsotrydniki` INT NOT NULL,
  `post` VARCHAR(255) NOT NULL,
  `surname` VARCHAR(255) NOT NULL,
  `name` VARCHAR(255) NOT NULL,
  `Middle name` VARCHAR(255) NOT NULL,
  PRIMARY KEY (`idsotrydniki`))
ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `mydb`.`nakladnaya` (
  `idnakladnaya` INT NOT NULL,
  `kod_tovara` VARCHAR(255) NOT NULL,
  `kod_provider` VARCHAR(255) NOT NULL,
  `kod_sotrydnika` VARCHAR(255) NOT NULL,
  `Date of manufacture` VARCHAR(255) NOT NULL,
  `cost` VARCHAR(255) NOT NULL,
  `Number of productsl` VARCHAR(255) NOT NULL,
  `tovar_idtovar` INT NOT NULL,
  `postavka_idpostavka` INT NOT NULL,
  `sotrydniki_idsotrydniki` INT NOT NULL,
  PRIMARY KEY (`idnakladnaya`, `tovar_idtovar`, `postavka_idpostavka`,
`sotrydniki_idsotrydniki`),
  INDEX `fk_nakladnaya_tovar1_idx` (`tovar_idtovar` ASC) VISIBLE,
  INDEX `fk_nakladnaya_postavka1_idx` (`postavka_idpostavka` ASC) VISIBLE,
  INDEX `fk_nakladnaya_sotrydniki1_idx` (`sotrydniki_idsotrydniki` ASC) VISIBLE,
  CONSTRAINT `fk_nakladnaya_tovar1`
    FOREIGN KEY (`tovar_idtovar`)
    REFERENCES `mydb`.`tovar` (`idtovar`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,

```

```

CONSTRAINT `fk_nakladnaya_postavka1`
    FOREIGN KEY (`postavka_idpostavka`)
    REFERENCES `mydb`.`postavka` (`idpostavka`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
CONSTRAINT `fk_nakladnaya_sotrydniki1`
    FOREIGN KEY (`sotrydniki_idsotrydniki`)
    REFERENCES `mydb`.`sotrydniki` (`idsotrydniki`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
CREATE TABLE IF NOT EXISTS `mydb`.`sklad` (
    `idsklad` INT NOT NULL,
    `kod_nakladnoi` VARCHAR(255) NOT NULL,
    `current quantity` VARCHAR(255) NOT NULL,
    `nakladnaya_idnakladnaya` INT NOT NULL,
    `nakladnaya_tovar_idtovar` INT NOT NULL,
    `nakladnaya_postavka_idpostavka` INT NOT NULL,
    PRIMARY KEY (`idsklad`, `nakladnaya_idnakladnaya`, `nakladnaya_tovar_idtovar`,
`nakladnaya_postavka_idpostavka`),
    INDEX `fk_sklad_nakladnaya1_idx` (`nakladnaya_idnakladnaya` ASC,
`nakladnaya_tovar_idtovar` ASC, `nakladnaya_postavka_idpostavka` ASC) VISIBLE,
    CONSTRAINT `fk_sklad_nakladnaya1`
        FOREIGN KEY (`nakladnaya_idnakladnaya` , `nakladnaya_tovar_idtovar` ,
`nakladnaya_postavka_idpostavka`)
        REFERENCES `mydb`.`nakladnaya` (`idnakladnaya` , `tovar_idtovar` ,
`postavka_idpostavka`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION)

```

```

ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `mydb`.`torgovlya` (
  `idtorgovlya` INT NOT NULL,
  `kod_tovara` VARCHAR(255) NOT NULL,
  `data` VARCHAR(255) NOT NULL,
  `Number of sales` VARCHAR(255) NOT NULL,
  `sklad_idsklad` INT NOT NULL,
  PRIMARY KEY (`idtorgovlya`, `sklad_idsklad`),
  INDEX `fk_torgovlya_sklad1_idx` (`sklad_idsklad` ASC) VISIBLE,
  CONSTRAINT `fk_torgovlya_sklad1`
    FOREIGN KEY (`sklad_idsklad`)
    REFERENCES `mydb`.`sklad` (`idsklad`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

```

Приложение 2

Транзакции

```
START TRANSACTION;

INSERT INTO nakladnaya (
    idnakladnaya, kod_tovara, kod_provider, kod_sotrydnika,
    `Date of manufacture`, cost, `Number of products`,
    tovar_idtovar, postavka_idpostavka, sotrydniki_idsotrydniki
)
VALUES (
    1001, 'T-123', 'P-456', 'S-789',
    '2025-06-20', '1500', '100',
    1, 1, 1
);

INSERT INTO sklad (
    idsklad, kod_nakladnoi, `current quantity`,
    nakladnaya_idnakladnaya,
    nakladnaya_tovar_idtovar,
    nakladnaya_postavka_idpostavka
)
VALUES (
    5001, 'NK-1001', '100',
    1001, 1, 1
);

INSERT INTO trgovlya (
    idtorgovlya, kod_tovara, data, `Number of sales`, sklad_idsklad
)
VALUES (
    7001, 'T-123', '2025-06-20', '5', 5001
);
```

```
UPDATE sklad  
SET `current quantity` = CAST(`current quantity` AS UNSIGNED) - 5  
WHERE idsklad = 5001;  
COMMIT;  
DELIMITER ;
```

Приложение 3

Представление

```
CREATE OR REPLACE VIEW `mydb`.`view_sales_info` AS
SELECT
    t.idtorgovlya,
    t.kod_tovara,
    t.`data` AS date_of_sale,
    t.`Number of sales` AS quantity_sold,
    sk.`current quantity` AS current_stock,
    tv.name_tovar AS product_name,
    p.name_postavka AS provider_name,
    s.surname AS employee_surname,
    s.name AS employee_name,
    s.`Middle name` AS employee_middlename
FROM mydb.torgovlya t
JOIN mydb.sklad sk ON t.sklad_idsklad = sk.idsklad
JOIN mydb.nakladnaya n ON sk.nakladnaya_idnakladnaya = n.idnakladnaya
AND sk.nakladnaya_tovar_idtovar = n.tovar_idtovar
AND sk.nakladnaya_postavka_idpostavka = n.postavka_idpostavka
JOIN mydb.tovar tv ON n.tovar_idtovar = tv.idtovar
JOIN mydb.postavka p ON n.postavka_idpostavka = p.idpostavka
JOIN mydb.sotrydniki s ON n.sotrydniki_idsotrydniki = s.idsotrydniki;
```

Приложение 4

Пользовательская функция

DELIMITER \$\$

CREATE FUNCTION mydb.get_tovar_name_by_id(p_idtovar INT)

RETURNS VARCHAR(255)

DETERMINISTIC

READS SQL DATA

BEGIN

 DECLARE tovar_name VARCHAR(255);

 SELECT name_tovar INTO tovar_name

 FROM mydb.tovar

 WHERE idtovar = p_idtovar;

 RETURN tovar_name;

END\$\$

DELIMITER ;

SELECT mydb.get_tovar_name_by_id(1);

Приложение 5

Обработчик исключений

DELIMITER \$\$

CREATE PROCEDURE mydb.sell_product(

 IN p_sklad_id INT,

 IN p_quantity INT

)

BEGIN

 DECLARE current_qty INT;

 DECLARE insufficient_stock CONDITION FOR SQLSTATE '45000';

 DECLARE CONTINUE HANDLER FOR insufficient_stock

 BEGIN

 SELECT 'Недостаточно товара на складе!' AS error_message;

 END;

 SELECT CAST(`current quantity` AS UNSIGNED)

 INTO current_qty

 FROM mydb.sklad

 WHERE idsklad = p_sklad_id;

 IF current_qty < p_quantity THEN

 SIGNAL SQLSTATE '45000'

 SET MESSAGE_TEXT = 'Not enough stock';

 ELSE

 UPDATE mydb.sklad

 SET `current quantity` = CAST(current_qty - p_quantity AS CHAR)

 WHERE idsklad = p_sklad_id;

 END IF;

END\$\$

DELIMITER ;

CALL mydb.sell_product(1, 5);

Уважаемый пользователь!

Обращаем ваше внимание, что система Антиплагиус отвечает на вопрос, является тот или иной фрагмент текста заимствованным или нет. Ответ на вопрос, является ли заимствованный фрагмент именно плагиатом, а не законной цитатой, система оставляет на ваше усмотрение.

Отчет о проверке № 9534584

Дата загрузки: 2025-06-25 01:47:22
Пользователь: volker17@mail.ru, ID: 9534584

Отчет предоставлен сервисом «Антиплагиат»
на сайте antiplagius.ru/

Информация о документе

№ документа: 9534584
Имя исходного файла: 6Д.docx
Размер файла: 0.18 МБ
Размер текста: 26360
Слов в тексте: 4032
Число предложений: 513

Информация об отчете

Дата: 2025-06-25 01:47:22 - Последний готовый отчет
Оценка оригинальности: 98%
Заимствования: 2%

98.89%

1.11%

Источники:

Доля в тексте	Ссылка
---------------	--------

Информация о документе: