



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

**КУРСОВОЙ ПРОЕКТ**

**«Разработка базы данных для сервиса доставки»**

Выполнил:

студент 3-го курса специальности

09.02.07 «Информационные системы и программирование»

Масалов Никита Максимович

подпись: \_\_\_\_\_

Проверил:

преподаватель дисциплины,

преподаватель ЧУПО «ВШП»,

к.ф.н. Ткачев П.С.

оценка: \_\_\_\_\_

подпись: \_\_\_\_\_

Тверь, 2024 г.

## Содержание

Введение.....	3
1. Сервис доставки готовой еды.....	4
1.1. Доставка: Понятия и какие бывают.....	4
1.2. Изучение рынка доставки.....	6
1.3. Управление персональными данными в сервисах доставки..	9
2. База данных для сервиса доставки.....	12
2.1. Структура таблиц в базе данных.....	12
2.2. Бизнес-логика базы данных.....	16
Заключение.....	22
Список источников.....	23
Приложение 1.....	24
Приложение 2.....	25

## Введение

В современном мире, где скорость и эффективность становятся ключевыми факторами успеха, сервисы доставки играют важную роль в удовлетворении потребностей бизнеса и потребителей. С развитием интернет-торговли и ростом спроса на быстрые и надёжные способы доставки товаров, разработка эффективной системы управления доставкой становится все более актуальной. Одной из основных составляющих такой системы является база данных, которая обеспечивает хранение, обработку и анализ информации о заказах, клиентах, курьерах и маршрутах доставки.

Так же почти каждый в той или иной степени сталкивался с сервисами доставки как пример доставка еды или мебели. Сервисы доставки бывают разными и не только доставляют еду, к примеру сейчас очень часто люди стали пользоваться онлайн магазинами для доставки бытовых приборов, различных девайсов и даже предметов мебели.

Цель курсовой работы заключается в разработки базы данных для сервиса доставки готовой еды. В рамках данной работы будет проведён анализ существующих систем управления доставкой, определены требования к базе данных, а также создан прототип системы на основе выбранной СУБД.

И для достижения данной цели были поставлены такие задачи:

1. Изучить рынок. Что больше всего любят заказывать люди.
2. Изучить как работают сервисы доставки.
3. Определить какие данные может хранить в себе база данных.
4. Определить какие бизнес процессы могут проходить в базе данных.

## Сервис доставки готовой еды.

### 1.1. Доставка: Понятия и какие бывают

Прежде чем углубляться в работу самого сервиса доставки давайте дадим ему определение и изучим какие виды доставки бывают.

Доставка - это удобный сервис, который позволяет покупателям получить заказанные товары прямо к своему дверному проёму. Этот сервис также позволяет насладиться кулинарными изысками из любимых заведений, не выходя из дома. Весь процесс заказа и доставки осуществляется без лишних хлопот, что делает жизнь насыщенной и комфортной. Возможность получить все необходимое, не отрываясь от своих повседневных дел, является настоящим бонусом в современном ритме жизни.

Из понятия о том, что такое доставка следует, что это сервис который доставляет что то до дома, но какие же существуют виды доставки товара.

Существует несколько видов доставки товаров, которые могут различаться в зависимости от способа перевозки и времени доставки:

1. Доставка почтой - товар отправляется почтовым оператором и доставляется в почтовое отделение или по указанному адресу получателя.
2. Курьерская доставка - курьер доставляет товар непосредственно на указанный адрес получателя в удобное для него время.
3. Самовывоз - покупатель может самостоятельно забрать товар из пункта выдачи или магазина.
4. Экспресс-доставка - доставка товара в кратчайшие сроки, обычно в течение 24 часов.
5. Доставка грузов - специализированная доставка товаров больших размеров или в большом количестве с использованием грузового транспорта.

Кроме того, существуют различные услуги и технологии для обеспечения безопасности и удобства доставки, такие как отслеживание груза, страхование, упаковка и т.д. Каждый вид доставки имеет свои преимущества и недостатки, и выбор определённого способа зависит от конкретных потребностей и требований клиента.

Я затрону лишь курьерскую доставку так как в моей работе будет использоваться именно этот способ доставки. Использовать его я выбрал по причине того, что это самый популярный и удобный способ доставки. В определении я упоминал, что курьерская доставка доставляется при помощи курьера, но кто же такой сам курьер.

Курьер - это лицо или организация, которые занимаются доставкой деловых документов, писем, сообщений и малых объёмов груза. Курьеры работают в различных учреждениях, обеспечивая быструю и надёжную доставку важных бумаг или предоставляют услуги по доставке для других компаний и частных лиц. Мы разобрались с понятием доставки и её видами, но какие бывают сервисы доставки и что они могут доставлять. Существует множество сервисов доставки, каждый из которых специализируется на определённых услугах и товарах. Вот несколько популярных видов сервисов доставки:

Доставка еды: такие сервисы, как Delivery Club, Яндекс.Еда и СберМаркет предлагают доставку готовых блюд из ресторанов или кафе прямо к двери вашего дома или офиса. Вы выбираете блюдо из меню на сайте или в приложении, делаете заказ и оплачиваете его онлайн. После этого курьер забирает заказ у ресторана и доставляет его вам.

Доставка продуктов: сервисы доставки продуктов, такие как Яндекс.Лавка, iGooods и Перекресток.ру, позволяют заказать продукты из магазинов и получить их прямо у себя дома. Вы выбираете нужные товары на сайте, оплачиваете заказ и выбираете удобное время доставки. Курьер привозит продукты прямо к вашей двери.

Доставка товаров: такие сервисы как Ozon, Яндекс.Маркет и Wildberries предлагают доставку товаров из интернет-магазинов или частных. После оформления заказа на сайте или в мобильном приложении, товар упаковывается и передаётся курьерской службе. Затем курьер доставляет товар в указанное место это может быть пункт выдачи или же до дома.

## 1.2. Изучение рынка сервисов доставки.

Изучим поподробнее рынок доставок. Как я писал ранее я разрабатываю сервис по доставке готовой еды и какое существуют сервисы в этой области.

Известные сервисы в этой сфере это Delivery Club, Яндекс.Еда, СберМаркет, Dodo Pizza, Перекрёсток и другие. Эти сервисы предлагают широкий выбор заведений и меню для заказа и доставки еды прямо к двери, но какой же из этих сервисов самый популярный. В январе 2024 года Центр цифровой экспертизы Роскачества провёл исследование наиболее популярных мобильных приложений для доставки продуктов. Исследования проводились по 173 критериям и лучшими приложениями стали Перекресток Впрок гипермаркет и ВкусВилл: доставка продуктов. В качестве их достоинств они записали возможность платить наличными, возможность сделать дозаказ, и они функциональные.

Я изучил какие сервисы есть и какие среди них популярные и теперь можно ответить на вопрос какие блюда являются самыми популярными при заказе.

Яндекс.Еда подводила итоги по 2023 году и вот к каким выводам пришла. «Бургеры и пицца — традиционно одни из самых популярных категорий блюд в «Еде», и уходящий год не стал исключением. Бургеры и пиццу на сервисе чаще всего заказывают днем и вечером. Еще одна популярная категория блюд днем — супы. В ней по числу заказов лидируют борщ и том ям. Утром заказывают выпечку. А вот ночью многие предпочитают перекусить шаурмой — ее чаще всего выбирают после полуночи», — рассказали в компании. Топ 3 блюд стали картофель фри, хачапури по-аджарски, роллы «Филадельфия». Также в десятке лидеров оказались картофель по-деревенски, пицца «Пепперони», салат «Цезарь с курицей», борщ, сырники, палочки, пицца «Маргарита» и хачапури по-мегрельски. «В уходящем году выросла популярность хот-догов, шаурмы и кофе. Еще одним блюдом, на которое спрос сильно вырос за год, стали вафли — как сладкие, так и сырники. В 2023 году количество их заказов на сервисе увеличилось более чем в три раза относительно 2022 года. Это можно объяснить в том числе растущей на

сервисе популярностью завтраков: за последние два года число утренних заказов увеличилось на 40%», — отметили в компании. А вот среди национальных кухнь вошли: американская (бургеры, салаты, сэндвичи), японская, а также русская (блины, супы, выпечка).

Изучив популярные блюда я решил так же и затронуть какие бывают категории Бургеры, Суши, Пицца, Вок, Паста, Русская, Италия, Грузия, Десерты, Выпечка, Шашлык, Стейк, Шаурма, Морепродукты, Здоровая, Детское и так далее.

Изучив какие могут быть категории в моей базе данных и что в этих категориях может храниться. Далее нужно узнать а какая есть информация о продуктах в различных сервисах.

На некоторых сервисах есть описание продукта, его состав, цена, количество грамм, углеводы, белки, жиры, калории, срок хранения, условия хранения, оценки и кто производитель.

В создании своей базы данных я буду учитывать что в ней будет храниться.

Изучив популярные блюда и категории продуктов, следующим шагом является определение, какая информация о продуктах используется в различных сервисах доставки:

Описание продукта: Краткое описание блюда или продукта, включая его основные характеристики (например, состав, способ приготовления).

Цена: Цена продукта или блюда, которая может варьироваться в зависимости от ресторана или типа продукта.

Количество грамм: Вес продукта, который может быть важен для расчета порции и стоимости.

Пищевая ценность: Информация о содержании углеводов, белков, жиров, калорий и других пищевых компонентов. Это особенно важно для пользователей, следящих за питанием.

Срок хранения и условия хранения: Для продуктов, требующих специальных условий хранения или имеющих ограничения по срокам годности.

Оценки и отзывы: Отзывы пользователей и рейтинги блюд, которые могут помочь другим пользователям принимать решение о заказе.

Информация о производителе: Данные о поставщике или производителе продукта, что важно для прозрачности и доверия к сервису.

Эти аспекты помогут определить структуру базы данных, включая необходимые таблицы, поля и связи между ними. Например, вы можете создать таблицы для продуктов, ресторанов, пользователей, заказов и отзывов, а затем определить соответствующие атрибуты для каждого типа данных.



### 1.3 Управление персональными данными в сервисах доставки.

В данном параграфе я детальнее углублюсь в работу сервисов доставки.

Изучая материалы о современных сервисах доставки, я обратил внимание на ключевые аспекты автоматизации и оптимизации процессов доставки, которые играют важную роль в их функционировании. Давайте рассмотрим подробнее каждый из этих аспектов:

**Автоматизация процесса:** Программное обеспечение значительно упрощает и ускоряет выполнение рутинных задач, которые ранее выполнялись вручную или требовали значительных усилий. Это включает анализ текущих условий дорог, погодных условий и других факторов, которые могут повлиять на время и успешность доставки.

**Оптимизация маршрутов:** Система создает наиболее эффективные маршруты доставки, учитывая не только расстояние, но и текущие условия движения. Это помогает сократить время на доставку, сэкономить топливо и повысить общую эффективность процесса.

**Управление персоналом:** Отслеживание местоположения курьеров в реальном времени позволяет оптимизировать их маршруты, оценивать скорость доставки и эффективность работы. Такой подход обеспечивает более точное планирование и управление задачами, что особенно важно в условиях высокой нагрузки и изменчивости заказов.

**Отчётность и связь:** Встроенные в систему чаты и уведомления обеспечивают оперативное взаимодействие между курьерами и диспетчерами. Курьеры могут предоставлять отчёты о выполнении задач, обсуждать оперативные вопросы и быстро решать нестандартные ситуации, что способствует улучшению обслуживания клиентов.

**Повышение качества обслуживания:** Автоматизация позволяет достигать более высокой точности выполнения заказов, быстрее реагировать на изменения в условиях доставки и эффективнее использовать ресурсы. Это в конечном итоге приводит к улучшению качества обслуживания клиентов и укреплению репутации сервиса доставки.

Таким образом, использование современных технологий в управлении доставкой играет центральную роль в обеспечении эффективности, надежности и удобства для клиентов и персонала сервиса.

Далее стоит затронуть какую информацию берут у курьеров сервисы и зачем она им. В настоящей политике обработки персональных данных ООО "Деливери Клуб Партнёр" описывается процесс сбора, использования и защиты информации, которую компания получает от пользователей своего веб-сайта. Вот основные аспекты политики:

Цели и правовые основания обработки персональных данных:

Основная цель сбора данных (фамилии, имени, отчества, номеров телефонов и города проживания) - помощь в трудоустройстве курьером.

Дополнительно данные используются для уведомлений о новых продуктах, услугах, специальных предложениях и мероприятиях. Пользователи могут отказаться от таких уведомлений.

Обработка персональных данных происходит при заполнении специальных форм на сайте или при согласии пользователя на использование файлов "cookie" и JavaScript.

Защита и передача персональных данных:

Оператор обеспечивает безопасность персональных данных с помощью соответствующих правовых, организационных и технических мер.

Персональные данные не передаются третьим лицам, за исключением случаев, предусмотренных законодательством или с согласия пользователя.

Трансграничная передача персональных данных осуществляется с соблюдением требований о защите данных в других странах.

Права пользователей и изменения в политике:

Пользователи имеют право на доступ к своим данным, их исправление, а также могут отозвать согласие на обработку персональных данных в любой момент.

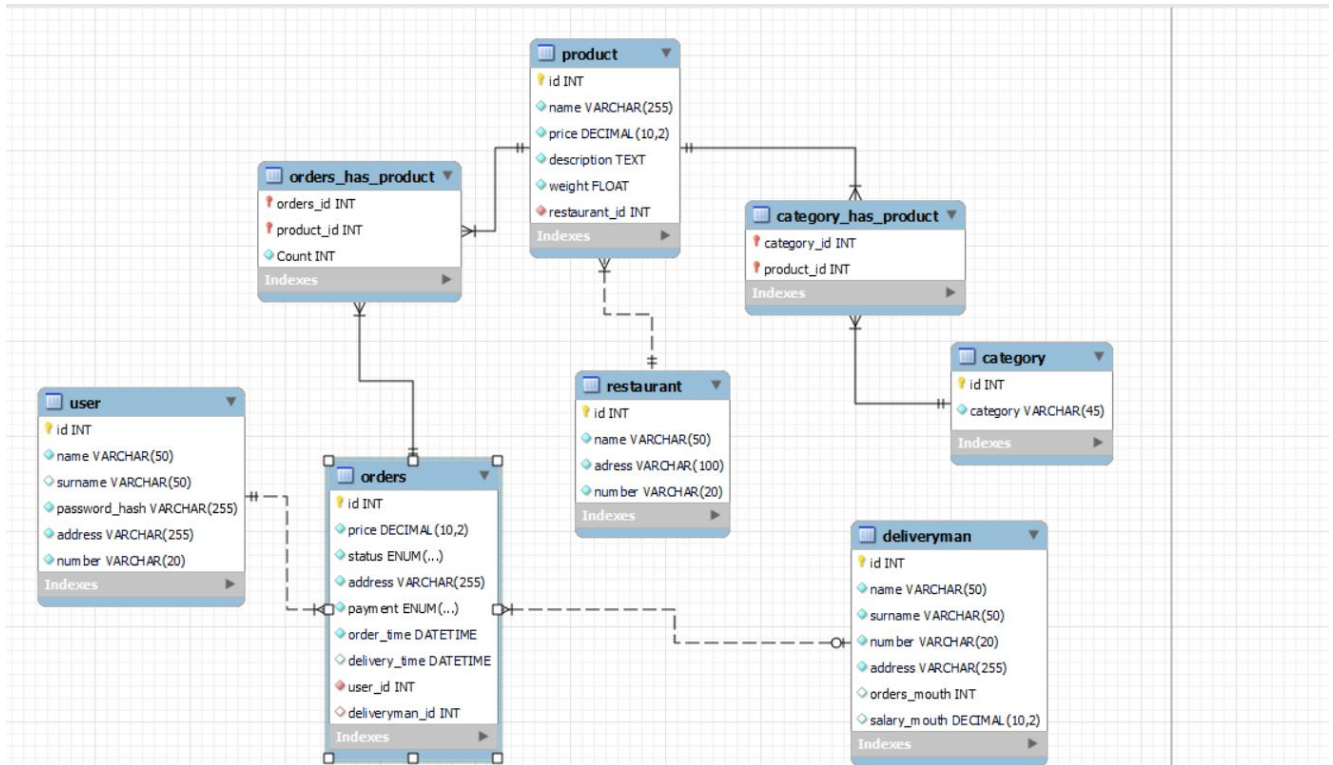
Оператор оставляет за собой право вносить изменения в политику обработки персональных данных. Актуальная версия политики всегда доступна на сайте компании.

Эта политика направлена на обеспечение защиты и соблюдения прав пользователей в контексте использования сервисов Delivery Club.

## 2. База данных для сервиса доставки

### 2.1. Структура таблиц в базе данных.

В этой главе я расскажу о разработке базы данных для сервиса доставки, которая была выполнена в рамках данного проекта. И сейчас я начну с таблиц в базе данных и их связей



Всего мною было создано 8 таблиц которые связаны между собой и про каждую из таблиц хочется рассказать по подробнее.

Таблица «user» предназначена для хранения информации о пользователях которые пользуются сервисом доставки. Можно заметить данные что она в себе хранит такие как имя пользователя, фамилия, адрес и номер а так же пароль а точнее его хэш чтобы если даже база данных утекла или к ней получит доступ злоумышленники они не смогли войти под видом пользователей. Для связи с пользователем в случае каких либо проблем храниться номер телефона пользователя. Адрес нужен чтобы доставить курьеры знали куда доставлять товар. Так же для общения с пользователем спрашивают его имя и при желании пользователя можно написать фамилию. Таблица «user» связана только с таблицей «orders» как один ко многим.

Таблица «orders» предназначена для хранения данных о заказах, сделанных пользователями через сервис доставки. Эта таблица содержит всю необходимую информацию о каждом заказе, чтобы обеспечить правильную обработку и доставку. В таблице «orders» хранятся следующие данные: уникальный идентификатор каждого заказа, стоимость заказа, статус заказа, который может быть одним из следующих значений: "в корзине", "принят", "отклонен", "готовится", "передан для доставки", "в доставке", "доставлен". Также в таблице есть адрес доставки, по которому нужно доставить заказ, и способ оплаты, который может быть "Онлайн" или "Наличными".

Время и дата заказа фиксируются для отслеживания временных рамок, а время и дата доставки помогают управлять своевременностью выполнения заказов. Идентификатор пользователя связывает таблицу «orders» с таблицей «user», что позволяет отслеживать, какой пользователь сделал заказ. Эта связь «один ко многим» означает, что один пользователь может сделать несколько заказов, но каждый заказ относится только к одному пользователю. Идентификатор курьера связывает таблицу «orders» с таблицей «deliveryman» и позволяет отслеживать, какой курьер отвечает за доставку конкретного заказа.

Таблица «orders» также связана с таблицей «product» через таблицу-связку «orders\_has\_product», что позволяет отслеживать, какие продукты включены в каждый заказ.

Таблица «product» предназначена для хранения данных о продуктах, доступных для заказа через сервис доставки. В этой таблице хранятся такие данные, как наименование продукта, его цена, описание, вес, и идентификатор ресторана, из которого этот продукт. Каждому продукту присваивается уникальный идентификатор, который позволяет однозначно идентифицировать его среди других продуктов. Наименование продукта помогает пользователям выбрать нужный товар из списка. Цена указывается с точностью до двух знаков после запятой, что позволяет учитывать даже небольшие изменения стоимости. Описание продукта содержит информацию о его составе и других характеристиках, которые могут быть важны для пользователя. Вес продукта

указывается для того, чтобы пользователи знали, сколько они получают за свою оплату.

Таблица «product» связана с таблицей «restaurant» через идентификатор ресторана, что позволяет определить, из какого именно ресторана можно заказать данный продукт. Эта связь «многие к одному» означает, что один ресторан может предлагать множество различных продуктов, но каждый продукт относится только к одному ресторану.

Кроме того, таблица «product» связана с таблицей «orders» через таблицу-связку «orders\_has\_product». Это позволяет отслеживать, какие продукты включены в каждый заказ, и обеспечивает возможность управления содержимым заказов. Также таблица «product» связана с таблицей «category» через таблицу-связку «category\_has\_product», что позволяет классифицировать продукты по категориям, таким как «бургеры», «пицца», «суши» и т.д., и облегчает поиск продуктов для пользователей.

Таблица «category» предназначена для хранения данных о категориях продуктов, доступных в сервисе доставки. В этой таблице хранятся такие данные, как уникальный идентификатор категории и наименование категории.

Наименование категории используется для классификации продуктов, что облегчает поиск и фильтрацию товаров для пользователей.

Таблица «restaurant» предназначена для хранения данных о ресторанах, предлагающих свои продукты через сервис доставки. В этой таблице хранятся такие данные, как наименование ресторана, адрес, и номер телефона.

Наименование ресторана используется для отображения информации о ресторане в интерфейсе пользователя. Адрес ресторана указывается для определения его местоположения, что может быть важно для планирования доставки. Номер телефона ресторана хранится для связи по вопросам заказа и доставки.

Таблица «deliveryman» предназначена для хранения данных о курьерах, работающих в сервисе доставки. В этой таблице хранятся такие данные, как имя курьера, фамилия, адрес, номер телефона, количество выполненных заказов за месяц и заработная плата за месяц.

Имя и фамилия используются для идентификации курьера в системе и взаимодействия с ним. Адрес курьера может использоваться для логистических целей, таких как распределение маршрутов. Номер телефона хранится для оперативной связи с курьером в процессе выполнения заказов.

Количество выполненных заказов за месяц и заработная плата за месяц являются важными показателями, которые используются не только для расчёта эффективности работы курьера, но и его заработная плата в зависимости от того сколько заказов сделал курьер.

Таблица «deliveryman» связана с таблицей «orders». Эта связь «один ко многим» означает, что один курьер может выполнять множество различных заказов, но каждый заказ назначается только одному курьеру. Такая структура позволяет отслеживать, какие заказы выполняются каждым курьером, и управлять процессом доставки в сервисе.

## 2.2. Бизнес-логика базы данных.

В этом параграфе я затрону бизнес-логику и какие процессы могут проходить в базе данных для сервиса доставки. Начну с типовых запросов которые отражают бизнес процессы проходящие в базе данных.

Запрос на получение самых популярных продуктов:

```
SELECT p.name, COUNT(*) AS order_count
FROM orders_has_product op
JOIN product p ON op.product_id = p.id
GROUP BY op.product_id
ORDER BY order_count DESC
LIMIT 10;
```

Этот запрос определяет самые популярные продукты на основе количества заказов. Это важно для оптимизации запасов, маркетинговых кампаний, обновления меню и повышения общей выручки. Понимание предпочтений клиентов улучшает их опыт, увеличивает удовлетворенность и количество повторных заказов.

Запрос на получение самого дешёвого блюда из категории Пиццы:

```
SELECT *
FROM mydb.category_has_product
join product ON category_has_product.product_id = product.id
join category on category_has_product.Category_id = category.id
where category = "Пиццы"
order by product.price ASC
limit 1;
```

Этот запрос находит самое дешёвое блюдо в категории "Пиццы". Знание такой информации важно для установки ценовой политики, создания специальных предложений и привлечения клиентов, ищущих экономичные варианты.

Запрос на просмотр заказов, которые ещё не взяли курьеры:

```
SELECT *
FROM mydb.orders
```



```
WHERE orders.deliveryman_id IS NULL;
```

Этот запрос выявляет заказы, которые ещё не назначены курьерам для доставки. Он важен для оперативного распределения заказов среди курьеров, обеспечивая своевременную доставку и минимизируя время ожидания клиентов. Информация о нераспределенных заказах помогает оптимизировать рабочие процессы и улучшать общую эффективность доставки, поддерживая баланс нагрузки между курьерами.

Запрос на просмотр курьера, который доставил больше всего заказов:

```
SELECT *
```

```
FROM mydb.deliveryman
```

```
ORDER BY deliveryman.orders_mouth DESC LIMIT 1;
```

Этот запрос выявляет курьера, который доставил больше всего заказов. Информация о лучших курьерах помогает в мотивации сотрудников, управлении персоналом и планировании рабочих графиков.

Запрос на просмотр заказов из ресторана "Кафе "Гриль Хаус"":

```
SELECT product.name, orders.status, orders.order_time
```

```
FROM mydb.orders_has_product
```

```
JOIN product ON orders_has_product.product_id = product.id
```

```
JOIN mydb.orders ON orders_has_product.orders_id = orders.id
```

```
WHERE product.restaurant_id =
```

```
(SELECT id FROM mydb.restaurant WHERE name = 'Кафе "Гриль Хаус"');
```

Этот запрос позволяет получить информацию о заказах, сделанных в ресторане "Кафе "Гриль Хаус"". Анализ таких заказов важен для управления запасами, понимания спроса на продукцию и эффективного планирования работы ресторана. Далее рассмотрим транзакцию:

```
START TRANSACTION;
```

```
SELECT address INTO @user_address FROM user WHERE id = 3;
```

```
INSERT INTO orders (price, status, address, payment, order_time, user_id) VALUES  
(0.00, 'accepted', @user_address, 'Online', NOW(), 3);
```

```
SET @order_id = LAST_INSERT_ID();
```

```

INSERT INTO orders_has_product (orders_id, product_id, Count) VALUES
(@order_id, 4, 2), (@order_id, 3, 1);
SELECT SUM(p.price * o.Count) INTO @total_price FROM orders_has_product o
JOIN product p ON o.product_id = p.id WHERE o.orders_id = @order_id;
UPDATE orders SET price = @total_price, order_time = NOW()
WHERE id = @order_id;
COMMIT;

```

Эта транзакция создаёт новый заказ на доставку и автоматически рассчитывает его общую стоимость. Он начинает транзакцию, извлекает адрес пользователя, вставляет информацию о заказе и связях с продуктами, вычисляет общую стоимость заказа на основе цен и количества продуктов, затем обновляет запись заказа с рассчитанной ценой и завершает транзакцию.

Хранимая процедура для добавления пользователя

```
DELIMITER //
```

```

CREATE PROCEDURE `mydb`.`add_user` (
    IN p_name VARCHAR(50),
    IN p_surname VARCHAR(50),
    IN p_password_hash VARCHAR(255),
    IN p_address VARCHAR(255),
    IN p_number VARCHAR(20)
)
BEGIN
    INSERT INTO `mydb`.`user` (`name`, `surname`, `password_hash`, `address`,
`number`)
        VALUES (p_name, p_surname, p_password_hash, p_address, p_number);
END //
DELIMITER ;

```

Представление (приложение 2):

Это представление "order\_details" объединяет данные о заказах, пользователях, курьерах, продуктах и ресторанах для полного описания каждого заказа.

Включает идентификатор заказа, цену, статус, адрес доставки, способ оплаты, время заказа и доставки, данные пользователя (имя, фамилия, адрес), данные курьера (имя, фамилия), данные о продукте (идентификатор, название, цена, описание, вес) и данные о ресторане (идентификатор, название, адрес, номер телефона), а также количество каждого продукта в заказе.

Триггер:

```
DELIMITER //
```

```
CREATE PROCEDURE update_order_status(
```

```
    IN order_id INT,
```

```
    IN new_status ENUM('in the basket', 'accepted', 'rejected', 'prepare', 'transferred for  
delivery', 'in delivery', 'delivered')
```

```
)
```

```
BEGIN
```

```
    DECLARE EXIT HANDLER FOR SQLEXCEPTION
```

```
    BEGIN
```

```
        SELECT 'Ошибка при обновлении статуса заказа. Возможно, заказ с  
таким идентификатором не существует.' AS error_message;
```

```
    END;
```

```
    UPDATE `mydb`.`orders`
```

```
    SET status = new_status
```

```
    WHERE id = order_id;
```

```
END //
```

```
DELIMITER ;
```

Этот триггер `update_order_status` срабатывает после обновления записи в таблице `orders` базы данных `mydb`. Он автоматически обновляет статус заказа (`status`) на новое значение (`new_status`), определенное в соответствии с переданными параметрами, когда изменяется статус заказа.

Если произойдет ошибка SQL в процессе обновления, триггер автоматически обрабатывает исключение с помощью обработчика ошибок, который возвращает понятное сообщение о возможной причине ошибки.

Триггер обеспечивает автоматизацию процесса обновления статуса заказа в базе данных, что упрощает управление и поддержку заказов.

```
DELIMITER //
```

```
CREATE FUNCTION get_orders_for_restaurant(restaurantId INT) RETURNS TEXT
BEGIN
    DECLARE order_list TEXT;
    SELECT GROUP_CONCAT(CONCAT('Заказ ', ': ', p.name) ORDER BY o.id
SEPARATOR '\n')
    INTO order_list
    FROM `mydb`.`orders` o
    JOIN `mydb`.`orders_has_product` ohp ON o.id = ohp.orders_id
    JOIN `mydb`.`product` p ON ohp.product_id = p.id
    WHERE p.restaurant_id = restaurantId
        AND o.status NOT IN ('in the basket', 'rejected', 'delivered');
    RETURN order_list;
END //
DELIMITER ;
```

Эта функция `get_orders_for_restaurant` возвращает список заказов для указанного ресторана в формате текста. Функция принимает один параметр `restaurantId`, идентификатор ресторана.

Обработчик исключений:

```
VARCHARDELIMITER //
CREATE PROCEDURE update_order_status(
    IN order_id INT,
    IN new_status ENUM('in the basket', 'accepted', 'rejected', 'prepare', 'transferred for delivery', 'in
delivery', 'delivered')
)
BEGIN
    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        SELECT 'Ошибка при обновлении статуса заказа. Возможно, заказ с таким
идентификатором не существует.' AS error_message;
    END;
    UPDATE `mydb`.`orders`
    SET status = new_status
```

```
WHERE id = order_id;  
END //  
DELIMITER ;
```

Этот обработчик исключений предназначен для процедуры `update_order_status`, которая обновляет статус заказа в таблице `orders` базы данных `mydb`. Обработчик `EXIT HANDLER FOR SQLEXCEPTION` активируется в случае возникновения ошибки SQL во время выполнения операции обновления.

## Заключение

В ходе курсовой работы была разработана база данных для сервиса доставки готовой еды, что позволит оптимизировать процессы управления заказами и повысить удовлетворённость клиентов. Анализ существующих систем управления доставкой показал, что эффективная база данных является критически важной составляющей для обеспечения оперативности и надёжности сервиса. Основные компоненты разработанной базы данных включают таблицы для хранения информации о пользователях, заказах, продуктах, ресторанах, а также связанные с ними данные о доставке и платежах. Эта структура обеспечивает полную информационную поддержку всех этапов процесса доставки — от размещения заказа пользователем до его финальной доставки курьером. Основные требования к базе данных определялись на основе анализа рынка и специфики услуги доставки готовой еды. Важными аспектами являются хранение персональных данных пользователей в соответствии с требованиями безопасности, а также обеспечение точности и актуальности информации о продуктах и заказах.

В процессе разработки базы данных были реализованы функциональные возможности, включая автоматизацию процессов управления заказами, оптимизацию маршрутов доставки, управление персоналом и взаимодействие с клиентами через встроенные чаты и уведомления. Это позволяет значительно повысить эффективность работы сервиса и улучшить качество обслуживания. Разработанная база данных ориентирована на масштабируемость и гибкость, что позволит адаптировать систему к изменяющимся потребностям рынка и расширять функциональность в будущем. Рекомендуется дальнейшее совершенствование системы на основе обратной связи от пользователей и мониторинга изменений в индустрии доставки.

Таким образом, разработка базы данных для сервиса доставки готовой еды представляет собой важный шаг в направлении улучшения качества обслуживания и конкурентоспособности на рынке. Это позволит компании эффективно управлять заказами, повышать удовлетворённость клиентов и расширять ассортимент предлагаемых услуг.

## Список источников

1. Политика в отношении обработки персональных данных - Режим доступа: <https://courier-delivery-club.ru/privacy.html>
2. Статья на тему -Что заказывали в «Яндекс Еде» в 2023 году- Режим доступа: <https://www.bfm.ru/news/540662>
3. Рейтинг доставок продуктов - Режим доступа: <https://rskrf.ru/ratings/tekhnika-i-elektronika/mobilnye-prilozheniya/zakaz-produktov/#>
4. Оптимизация процесса доставки: роль управления курьерами - Режим доступа: [https://logistic.tools/blog/optimizaciya\\_processa\\_dostavki\\_rol\\_upravleniya\\_kurierami](https://logistic.tools/blog/optimizaciya_processa_dostavki_rol_upravleniya_kurierami)

## Приложение 1



Ссылка для ручного ввода:

<https://github.com/NikitaMasalov/KursachBD>



## Приложение 2

```
CREATE VIEW order_details AS

SELECT

    orders.id AS order_id,

    orders.price AS order_price,

    orders.status AS order_status,

    orders.address AS order_address,

    orders.payment AS order_payment,

    orders.order_time AS order_time,

    orders.delivery_time AS delivery_time,

    user.id AS user_id,

    user.name AS user_name,

    user.surname AS user_surname,

    user.address AS user_address,

    deliveryman.id AS deliveryman_id,

    deliveryman.name AS deliveryman_name,

    deliveryman.surname AS deliveryman_surname,

    product.id AS product_id,

    product.name AS product_name,

    product.price AS product_price,

    product.description AS product_description,

    product.weight AS product_weight,

    restaurant.id AS restaurant_id,

    restaurant.name AS restaurant_name,
```

```
restaurant.address AS restaurant_address,  
  
restaurant.number AS restaurant_number,  
  
orders_has_product.Count AS product_count  
  
FROM orders  
  
JOIN user ON orders.user_id = user.id  
  
LEFT JOIN deliveryman ON orders.deliveryman_id = deliveryman.id  
  
JOIN orders_has_product ON orders.id = orders_has_product.orders_id  
  
JOIN product ON orders_has_product.product_id = product.id  
  
JOIN restaurant ON product.restaurant_id = restaurant.id
```

## Антиплагиат



Ссылка для ручного ввода:

[http://antiplagius.ru/checkout/unique/report/8897112?email=maksimovicnikita049%40gmail.com&utm\\_nooverride=1](http://antiplagius.ru/checkout/unique/report/8897112?email=maksimovicnikita049%40gmail.com&utm_nooverride=1)