МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных технологий

Кафедра Информационные системы и технологии

Специальность 1-40 01 01 «Программное обеспечение информационных

технологий»

Специализация Программирование интернет приложений

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**К КУРСОВОМУ ПРОЕКТУ НА ТЕМУ:**

«База данных «Интернет-магазин алкогольных напитков» с реализацией технологии шифрования и маскирования БД»

Выполнил студент Шумский Евгений Сергеевич

(Ф.И.О.)

Руководитель проекта пр.-ст. Колмаков Михаил Владимирович

(учен. степень, звание, должность, Ф.И.О., подпись)

Заведующий кафедрой к.т.н., доц. Смелов В.В .

(учен. степень, звание, должность, Ф.И.О., подпись)

Курсовой проект защищен с оценкой

Минск 2021

Оглавление

[Введение 3](#_Toc90897774)

[1. Постановка задачи и анализ прототипов 4](#_Toc90897775)

[2. Разработка базы данных 5](#_Toc90897776)

[2.1 Проектирование базы данных 5](#_Toc90897777)

[2.2 Процедуры для решения поставленных задач 6](#_Toc90897778)

[2.3 Индексы 6](#_Toc90897779)

[2.4 Резервное копирование и восстановление базы данных 6](#_Toc90897780)

[3. Технология шифрования и маскирования данных в БД 8](#_Toc90897781)

[4. Импорт и экспорт данных 11](#_Toc90897782)

[5. Тестирование 12](#_Toc90897783)

[Заключение 13](#_Toc90897784)

[Список использованных источников 14](#_Toc90897785)

[Приложение А 15](#_Toc90897786)

[Диаграмма базы данных 15](#_Toc90897787)

[Приложение Б 16](#_Toc90897788)

[Хранимые процедуры для вывода данных 16](#_Toc90897789)

[Хранимые процедуры авторизации и регистрации пользователя 22](#_Toc90897790)

[Хранимые процедуры для работы с заказами 24](#_Toc90897791)

[Приложение В 26](#_Toc90897792)

[Приложение Г 27](#_Toc90897793)

# Введение

Целью данной работы была разработка реляционной базы данных на тему «Интернет-магазин алкогольных напитков». База данных должна быть составлена для работы онлайн магазина.

В связи с тем, что самым важным ресурсом современного мира становится информация, вполне закономерно развитие технологий в направлении хранения и управления данными.

Любое современное web-приложение невозможно представить без базы данных, работающей на основе одной из множества доступных СУБД.

Система управления базами данных (СУБД) — совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление, созданием и использованием баз данных.

База данных — это организованная структура, предназначенная для хранения информации, систематизированная таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью электронной вычислительной машины. Реляционная база данных — база данных, основанная на реляционной модели данных. В качестве СУБД для базы данных была выбрана Microsoft SQL Server, в связи с ее простотой, производительностью и надежностью.

# Постановка задачи и анализ прототипов

С развитием информационных технологий и интернета в частности начали появляться онлайн-магазины. Примечательной особенностью которых, в отличие от обычных магазинов, стало отсутствие необходимости физического присутствия покупателя в здании магазина. Таким образом они позволяют экономить огромное количество времени и сил покупателям.

Начнем рассмотрение аналогов с онлайн-магазина «WineStyle». Пример его интерфейса представлен на рисунке 1.1.

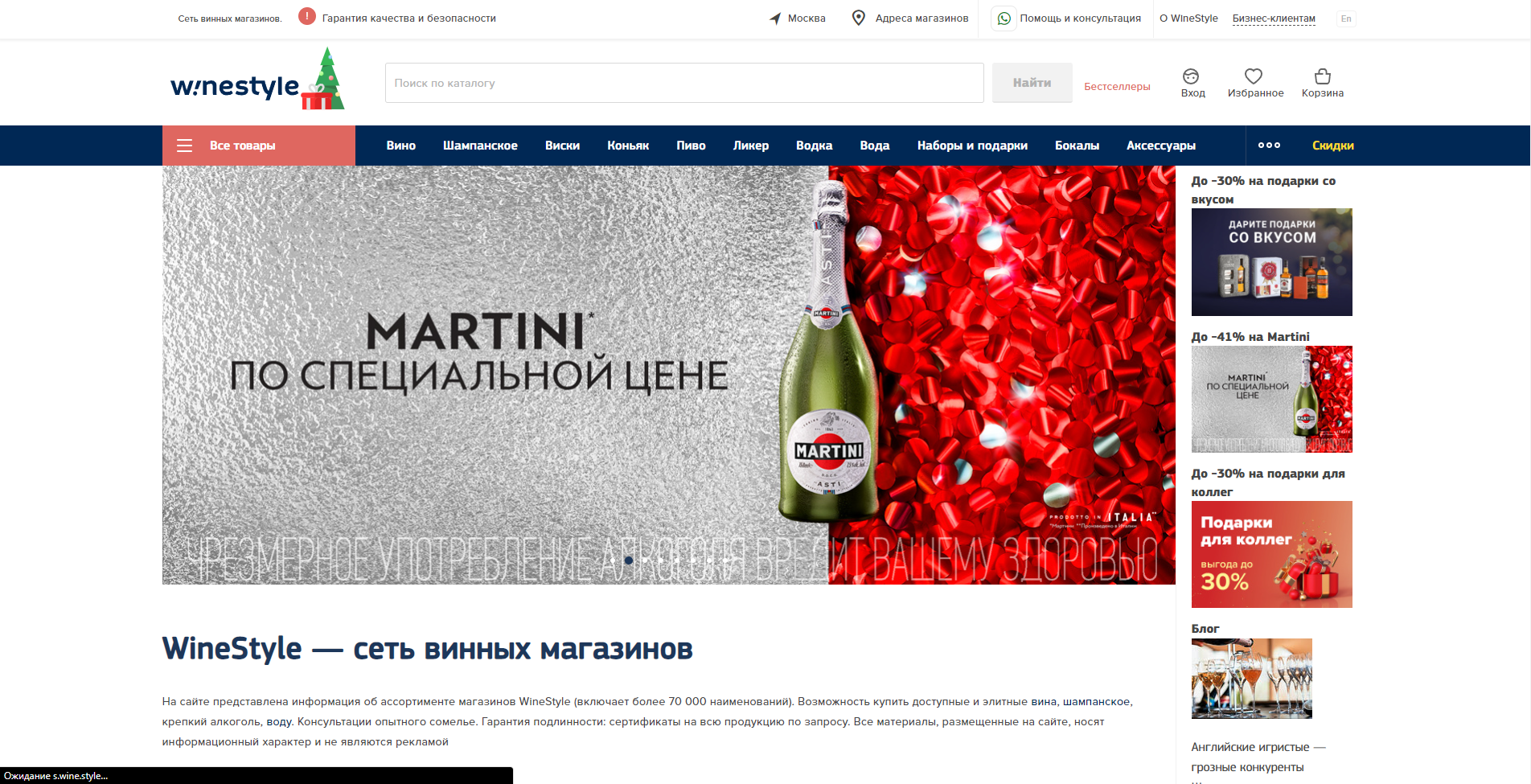


Рисунок 1.1 – Интерфейс сайта онлайн-магазина «WineStyle»

По итогам изучения аналогов были сделаны выводы, что разрабатываемая база данных должна обладать следующими возможностями:

* Регистрация и авторизация пользователя
* Добавление и удаление товара из корзины
* Оформление заказа
* Отслеживание состояния текущего заказа
* Просмотр истории заказов

# Разработка базы данных

2.1 Проектирование базы данных

Для реализации необходимого функционала была создана база данных, таблицы которой можно разделить на 3 логические группы: таблицы для хранения информации о продуктах, таблицы для хранения информации о заказах пользователей и таблицы для хранения пользовательских данных.

Рассмотрим таблицы базы данных на примере их диаграммы представленной на рисунке 2.1, а также в приложении А.

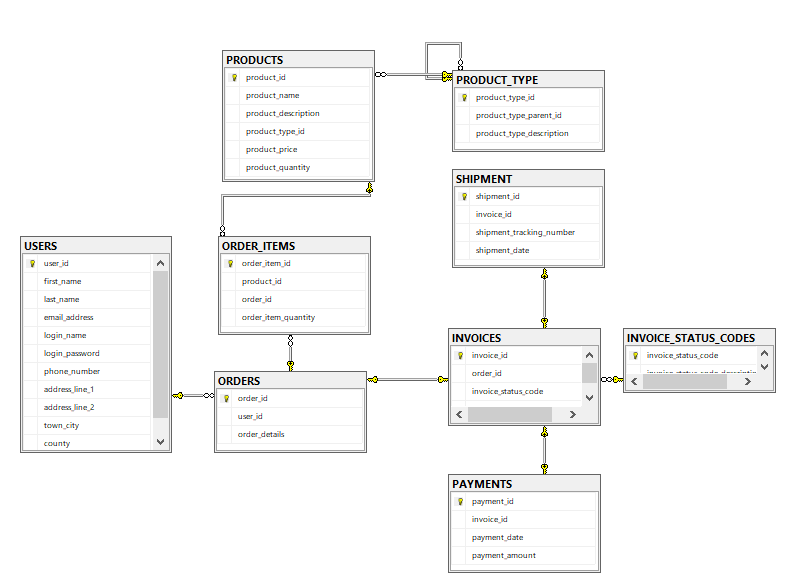


Рисунок 2.1 – Диаграмма таблиц базы данных

База данных состоит из 8 таблиц, связанных между собой внешними ключами.

Таблица «*USERS*» хранит в себе всю информацию о пользователях, включая пароль, логин, электронную почту и т.д.

Таблица «PRODUCTS» хранит информацию о продуктах, имеющихся в наличии, такую как: название продукта, описание продукта, тип продукта, цена продукта и его количество.

Таблица «*ORDERS*» хранит в себе информацию о заказах, включая номер заказа, идентификатор пользователя и описание заказа.

Таблица «ORDER\_ITEMS» хранит в себе информацию о продуктах из каждого заказа, включая номер заказа, идентификатор продукта, его количество и т. д.

Таблица «*PRODUCT\_TYPE*» содержит типы продукции и их описание. Типы могут наследовать другие типы таким образовывая свою иерархию.

Таблица «*INVOICES*» содержит информацию о оплаченных заказах, включая номер заказа, статус оплаты, дата оплаты и т. д.

Таблица «*INVOICE\_STATUS\_CODES*» содержит описание статусов оформления заказа.

Таблица «*PAYMENTS*» содержит информацию связанную с оплатой заказа, включая дату оплаты, цену заказа и т. д.

Таблица «*SHIPMENT*» содержит информацию связанную с доставкой заказа.

* 1. Процедуры для решения поставленных задач

Хранимая процедура – объект базы данных, представляющий собой набор SQL-инструкций, который компилируется один раз и хранится на сервере.

При разработке курсового проекта было создано большое количество процедур для следующих целей:

1. Выборка данных из таблиц;
2. Выборка данных по поисковому запросу;
3. Заполнение таблиц 100 000 строк;
4. Вход в аккаунт;
5. Удаление данных из таблиц;
6. Добавление данных в таблицы;
7. Экспорт и импорт таблицы PRODUCTS в формате xml;

Наиболее используемые процедуры представлены в приложении Б.

* 1. Индексы

Для оптимизации работы с поисковыми запросами, необходимо использовать индексы. Так, для работы с наиболее часто используемыми таблицами, были разработаны наборы индексов по наиболее используемым выборкам. Создание индексов представлено в приложении В.

2.4 Резервное копирование и восстановление базы данных

Для копирования базы данных была разработана процедура export\_BD. Для этого используется конструкция backup database, которая экспортирует базу данных в файл с расширением bak. Эта процедура представлена ниже на рисунке 2.2.

******

Рисунок 2.2 – Пример создания процедуры export\_DB

Для восстановления базы данных была разработана процедура import\_DB. Для этого используется конструкция restore database, которая экспортирует базу данных в файл с расширением bak. Эта процедура представлена ниже на рисунке 2.3.

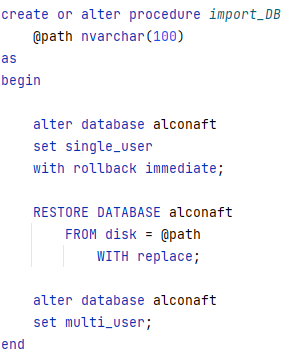


Рисунок 2.3 – Пример создания процедуры import\_DB

Для запуска и исполнения данной процедуры необходимо ее создавать системной базе данных master. Это необходимо для избегания коллизий при восстановлении. Так же восстановление невозможно, при наличии подключенных к базе пользователей, поэтому переводим базу данных в монопольный режим с помощью set single\_user и отменяем все изменения через параметр rollback immediate. После восстановления возвращаем базу данных в прежнее состояние через set multi\_user.

# Технология шифрования и маскирования данных в БД

Защита информации от посторонних – это шифрование. Шифрование представляет собой способ скрытия данных с помощью ключа или пароля. Это делает данные бесполезными без соответствующего ключа или пароля для дешифрования.

Защита должна всегда начинаться с разграничения прав доступа, но, даже в случае обхода системы управления правами, шифрование поможет защитить информацию. Например, когда база данных попадет в чужие руки, украденная или изъятая информация будет бесполезна, если она была предварительно зашифрована.

В *Microsoft SQL Server* 2008 впервые реализовано прозрачное шифрование баз данных (*Transparent Data Encryption*). Прозрачное шифрование кодирует базы данных целиком. Когда страница данных записывается из оперативной памяти на диск, она шифруется. Когда страница загружается обратно в оперативную память, она расшифровывается. Таким образом, база данных на диске оказывается полностью зашифрованной, а в оперативной памяти – нет. Основным преимуществом *TDE* является то, что шифрование и дешифрование выполняются абсолютно прозрачно для приложений. Использовать преимущества шифрования может любое приложение, использующее для хранения своих данных *Microsoft SQL Server*. При этом модификации или доработки приложения не потребуется.

В *Microsoft SQL* *Server* 2012 функции шифрования были улучшены и расширены. Для увеличения надежности криптозащиты и уменьшения нагрузки на систему применяется специальная иерархия ключей.

Пример включения прозрачного шифрования данных представлен на рисунке 3.1.

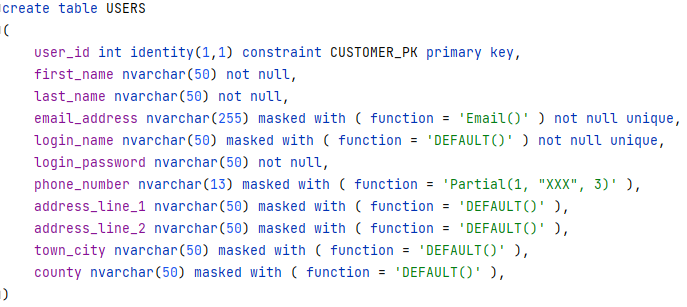


Рисунок 3.1 – Пример включения прозрачного шифрования данных

Динамическое маскирование данных (DDM) ограничивает возможность раскрытия конфиденциальных данных за счет маскирования этих данных для непривилегированных пользователей. Оно позволяет значительно упростить проектирование и написание кода для системы безопасности в приложении.

Динамическое маскирование данных помогает предотвратить несанкционированный доступ к конфиденциальным данным, позволяя клиентам задать объем раскрываемых конфиденциальных данных с минимальным влиянием на уровень приложения. DDM можно настроить для отдельных полей базы данных, чтобы скрыть конфиденциальные данные в результирующих наборах запросов. При использовании DDM данные в базе данных не изменяются. DDM легко использовать с существующими приложениями, так как правила маскирования применяются к результатам запроса. Многие приложения могут маскировать конфиденциальные данные без изменения существующих запросов.

Пример создания таблицы USERS с включением маскирования данных представлен на рисунке 3.2.



# Импорт и экспорт данных

Для импорта таблицы PRODUCTS в формате xml была разработана процедура import\_products. Для формирования xml в select запросе используется конструкция FOR XML. Для вывода в файл была использована расширенная хранимая процедура xp\_cmdshell. Процедура import\_products представлена ниже на рисунке 4.1. Отрывок из xml представлен в приложении Г.



Рисунок 4.1 – Пример создания процедуры import\_procedure

Для импорта данных в таблицу PRODUCTS, из файла формата xml, была разработана процедура export\_products. Для получения xml файла и последующего разбора со вставкой используется конструкция FROM OPENROWSET совместно с параметром BULK. Процедура export\_products представлена ниже на рисунке 4.2.



Рисунок 4.2 – Пример создания процедуры export\_products

# Тестирование

Для тестирования базы данных были проведены запросы для определения разницы времени в обработке запроса. Наиболее часто используемой является процедура login\_user, позволяющая авторизоваться пользователю.

Время выполнения процедуры и план выполнения запроса показаны на рисунках 5.1 и 5.2 соответственно.



Рисунок 5.1 – Время выполнения запроса при 100000 строках

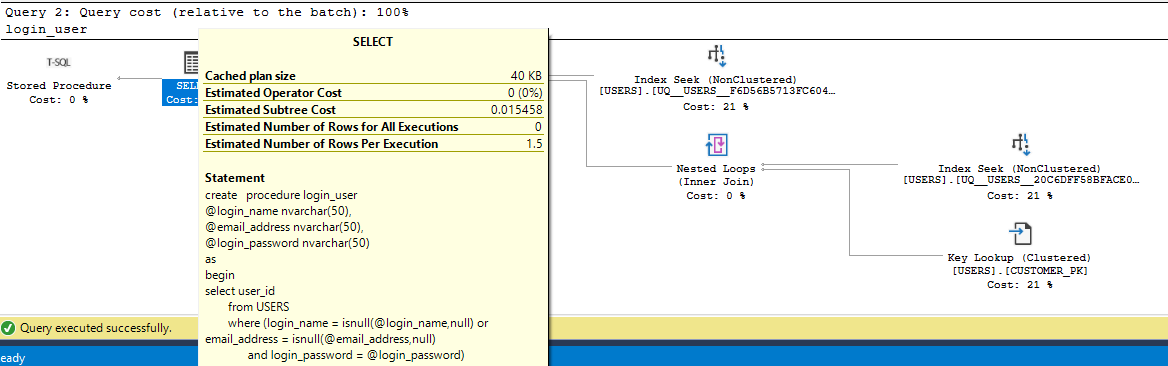


Рисунок 5.2 – План выполнения запроса при 100000 строках

Для 100000 строк результаты по времени выполнения были 75 мс., а по предполагаемой стоимости запроса – 0,015.

Подведя итоги тестирования производительности базы данных для различного объема данных в таблице можно сделать вывод, что приложение будет корректно работать с объемом выборки до 100000 значений.

# Заключение

В рамках работы над проектом был проведен обзор аналогичных решений и программных продуктов, спроектирована архитектура и структура базы данных, проведено тестирование производительности запросов к таблицам базы данных при различном объеме данных.

Разработанная база банных содержит в себе 8 связанных таблиц. Разработка базы данных проводилась в рамках *Microsoft SQL Server* – система управления реляционными базами данных. Данная СУБД используется для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия.

Одной из главных задач данной курсовой работы - освоение технологии шифрование и маскирование данных. Данная технология и ее возможности были успешно изучены и реализованы в процессе разработки базы данных.

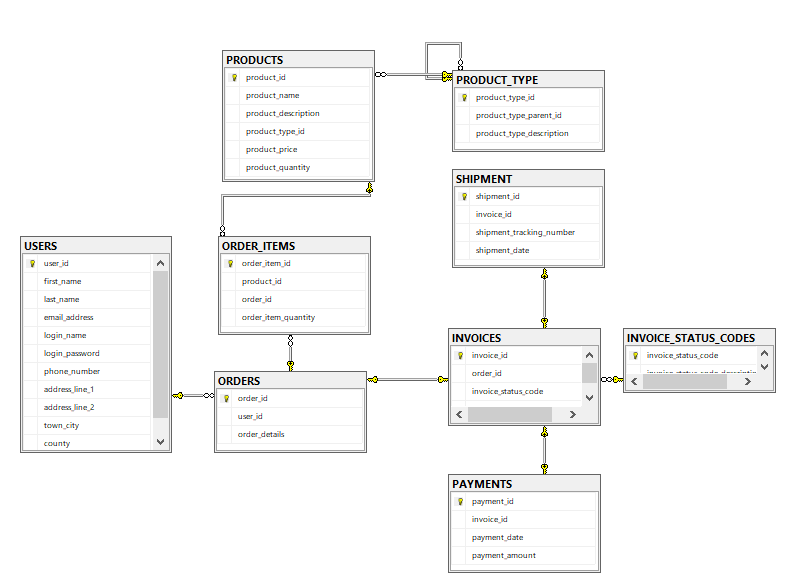
База данных является законченной, хотя и возможна доработка и расширение функционала. Данная база данных соответствует представленной задаче и отвечает всем необходимым требованиям.

# Список использованных источников

1. BACKUP (Transact-SQL) [Электронный ресурс] / Microsoft. – Режим доступа: https://docs.microsoft.com/en-us/sql/t-sql/statements/backup-transact-sql?view=sql-server-ver15. – Дата доступа: 15.11.2021.
2. Restore a Database Backup Using SSMS [Электронный ресурс] / Microsoft. – Режим доступа: https://docs.microsoft.com/en-us/sql/relational-databases/backup-restore/restore-a-database-backup-using-ssms?view=sql-server-ver15. – Дата доступа: 15.11.2021.
3. Transparent Data Encryption (TDE) SSMS [Электронный ресурс] / Microsoft. – Режим доступа: <https://docs.microsoft.com/en-us/sql/relational-databases/security/encryption/transparent-data-encryption?view=sql-server-ver15>
4. Dynamic Data Masking (DDM) [Электронный ресурс] / Microsoft. – Режим доступа: https://docs.microsoft.com/en-us/sql/relational-databases/security/dynamic-data-masking?view=sql-server-ver15

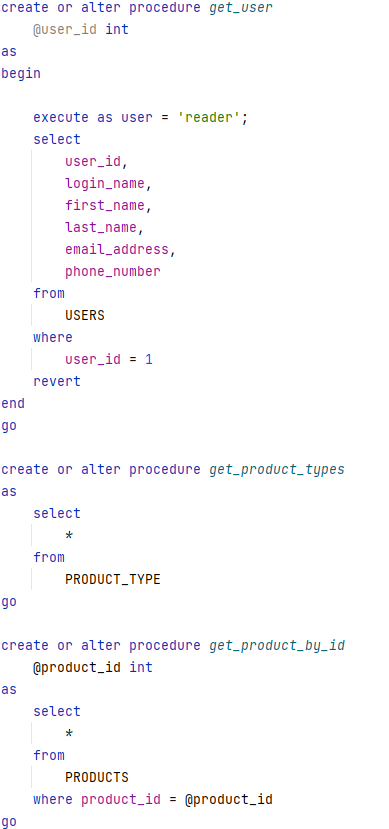
# Приложение А

Диаграмма базы данных

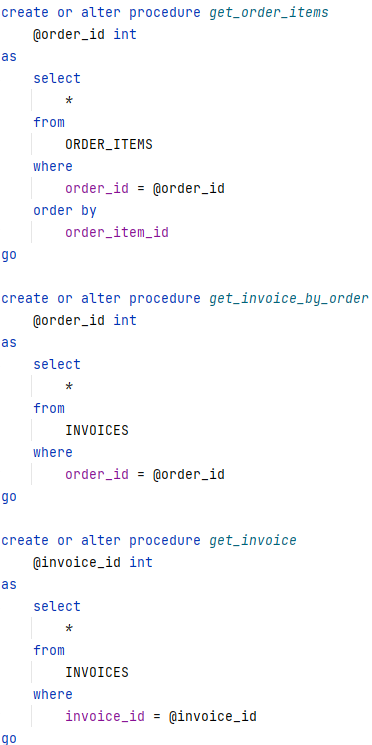


# Приложение Б

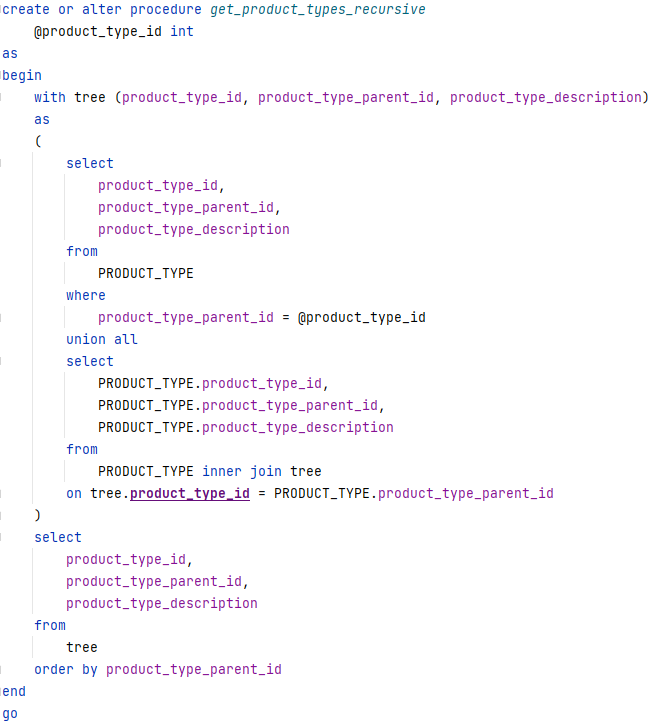
Хранимые процедуры для вывода данных

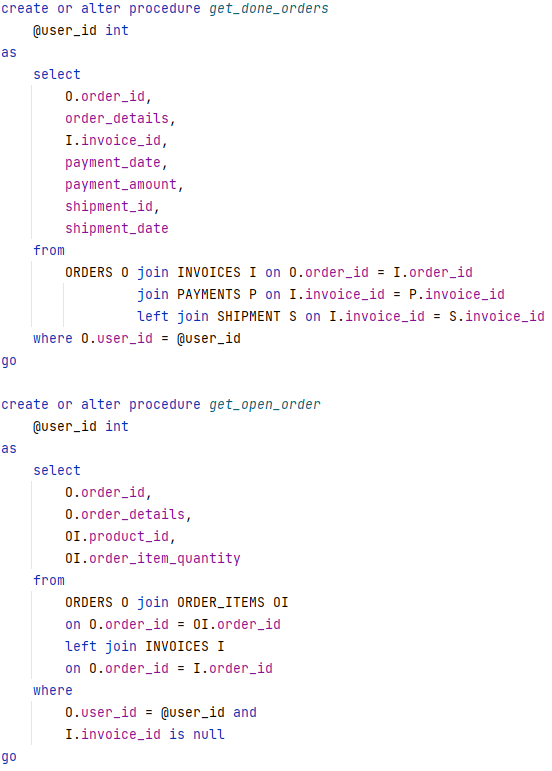
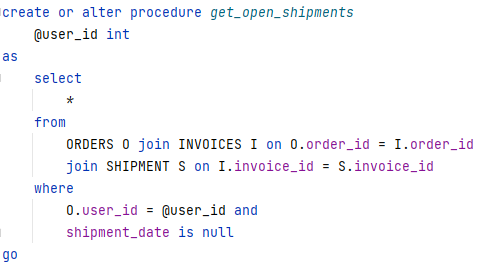




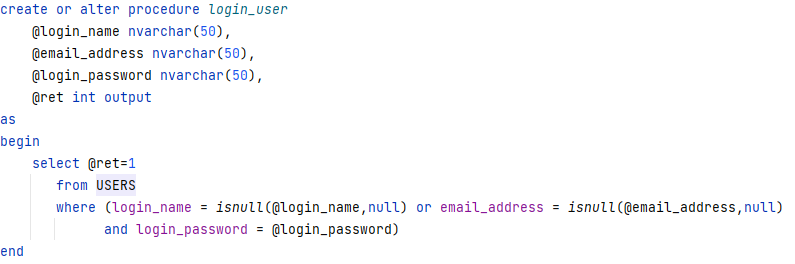






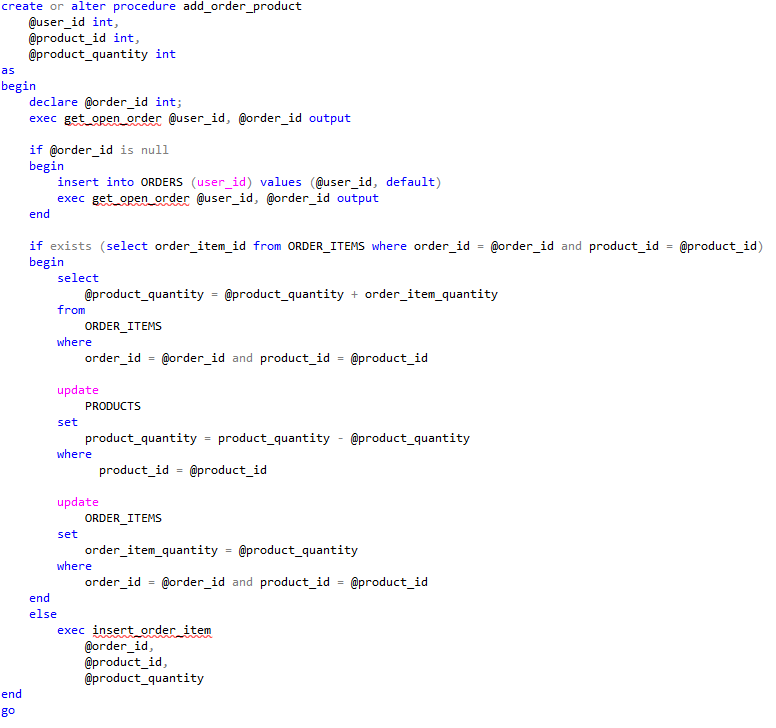
 

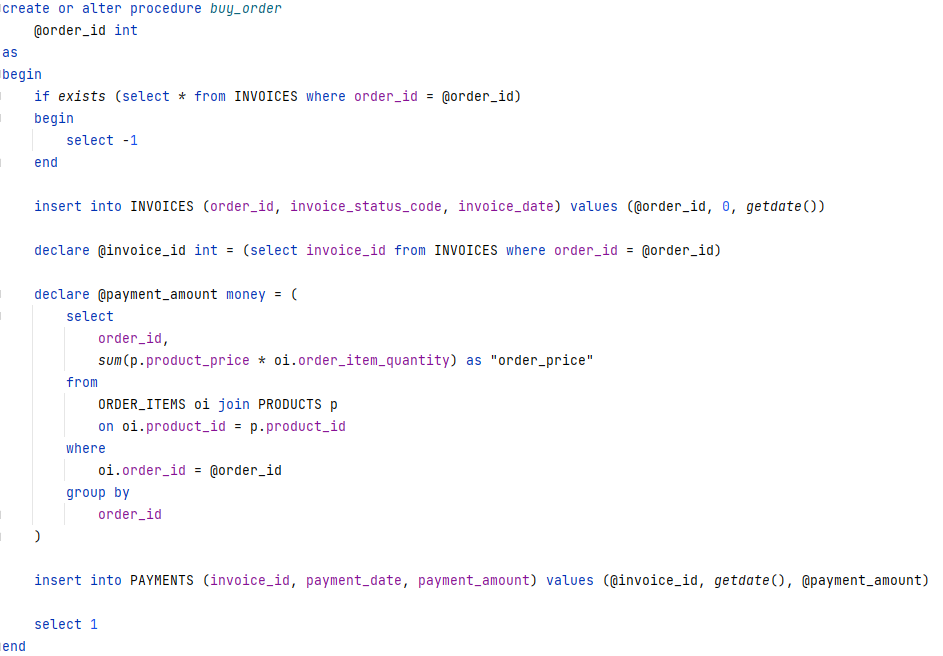
Хранимые процедуры авторизации и регистрации пользователя





Хранимые процедуры для работы с заказами







# Приложение В



# Приложение Г

