**Отчет по лабораторной работе № 1**

**“Проектирование базы данных”**

**Тема “Склад”**

Список типичных бизнес-задач, связанных с управлением базой данных для системы учета товаров и заказов на складе:

1. Учет товаров и категорий товаров:

* Создание и обновление записей о товарах и их категориях.
* Мониторинг изменений в ассортименте товаров.

1. Управление поставщиками:

* Ведение списка поставщиков и их контактной информации.
* Регистрация новых поставщиков и обновление информации о существующих.

1. Складской учет:

* Отслеживание количества товаров на складе.
* Мониторинг и контроль за приходом и отгрузкой товаров.

1. Оформление заказов:

* Создание заказов от клиентов.
* Регистрация деталей заказов, включая выбранные товары и их количество.

1. Управление заказами:

* Отслеживание статусов заказов (например, "собрано", "отправлено", "доставлено").

1. Места хранения:

* Управление местами хранения для товаров на складе.
* Распределение товаров по определенным местам хранения.

1. Управление доступом:

* Назначение прав доступа для пользователей и ролей.

Список пользователей проекта:

1. Администраторы системы:

* Отвечают за управление и обслуживание базы данных и приложения.
* Назначают роли и разрешения пользователям.
* Могут решать технические вопросы и выполнять обновления системы.

1. Менеджеры по продажам:

* Используют систему для создания заказов и мониторинга статусов заказов клиентов.
* Могут генерировать отчеты о продажах и анализировать данные о продажах.

1. Клиенты:

* Размещают заказы и отслеживают статусы своих заказов.
* Могут просматривать ассортимент товаров и описания товаров.

1. Складские работники:

* Используют систему для учета товаров на складе.
* Осуществляют приемку, хранение и отгрузку товаров.
* Управляют местами хранения на складе.

1. Поставщики:

* Могут взаимодействовать с системой, например, для подтверждения заказов и предоставления информации о поставках.

1. Курьеры:

* Просматривают информацию о заказе, чтобы доставлять заказ.
* Могут изменять статус заказа.

1. Финансовые аналитики:

* Используют данные системы для анализа финансовой производительности, расчетов стоимости заказов и прибыли.

1. Технический персонал:

* Отвечает за обслуживание и поддержку системы, включая решение технических проблем и обновления.

1. Разработчики:

* Участвуют в разработке и поддержке приложения и базы данных.

1. Тестировщики:

* Отвечают за тестирование функциональности системы и обнаружение ошибок.

Срок проекта:

1. Определение и планирование проекта (до начала фактической разработки):

- Сроки: июль 2022 - сентябрь 2022.

- Задачи: определение требований, составление плана проекта, назначение ролей и ответственностей.

1. Разработка приложения и базы данных:

- Сроки: октябрь 2022 - июль 2023.

- Задачи: проектирование и создание базы данных, разработка приложения, тестирование и отладка, интеграция с другими системами, создание пользовательского интерфейса.

1. Тестирование и отладка:

- Сроки: август 2023 - сентябрь 2023.

- Задачи: тестирование функциональности, выявление и устранение ошибок, обеспечение безопасности данных.

1. Внедрение и обучение пользователей:

- Сроки: октябрь 2023 - ноябрь 2023.

- Задачи: установка системы на рабочие сервера, обучение пользователей работе с системой, настройка прав доступа.

1. Мониторинг и поддержка:

- Сроки: декабрь 2023 и далее.

- Задачи: мониторинг работы системы, предоставление технической поддержки, внесение изменений и улучшений по необходимости.

Необходимые ресурсы для разработки данного проекта:

1. Команда разработки:

- Программисты и разработчики баз данных.

- Тестировщики и инженеры по обеспечению качества.

1. Аппаратное и программное обеспечение:

- Серверы и оборудование для хранения данных и обеспечения работы системы.

- Лицензии для баз данных и разработки.

1. Бюджетные ресурсы:

- Финансирование для оплаты заработной платы команды разработки и приобретения необходимого оборудования и лицензий.

1. Временные ресурсы:

- Доступность команды и ресурсов на все фазы проекта в соответствии с расписанием.

Для обеспечения безопасности БД необходимо будет использовать:

1. Ограничение доступа:

Определять и ограничивать права доступа пользователей и ролей к данным и объектам в БД. Использовать принцип наименьших привилегий.

1. Аутентификация и авторизация:

Реализовать механизмы аутентификации для проверки подлинности пользователей. Управлять авторизацией для определения, какие действия пользователи могут выполнять в БД.

1. Шифрование данных:

Использовать шифрование для защиты данных, передаваемых между клиентами и сервером БД, а также данных внутри БД.

1. Мониторинг и журналирование:

Включать мониторинг безопасности для регистрации попыток несанкционированного доступа и других событий. Регулярно просматривать и анализировать журналы для выявления аномалий.

1. Резервное копирование и восстановление:

Создавать регулярные резервные копии данных и хранить их в безопасном месте. Разработать и протестировать процедуры восстановления данных в случае катастроф.

1. Ограничение сетевого доступа:

Использовать брандмауэры и другие средства для ограничения сетевого доступа к серверу БД только из доверенных сетей.

1. Обучение и осведомленность:

Обучать своих сотрудников и пользователей базы данных в вопросах безопасности, включая соблюдение безопасных практик при работе с данными.

Проанализировав список пользователей проекта, было определено его будет присутствовать 4 роли: Администратор склада, менеджер заказов, сотрудник склада и курьер. Ниже представлена диаграмма, которая демонстрирует возможности пользователей.

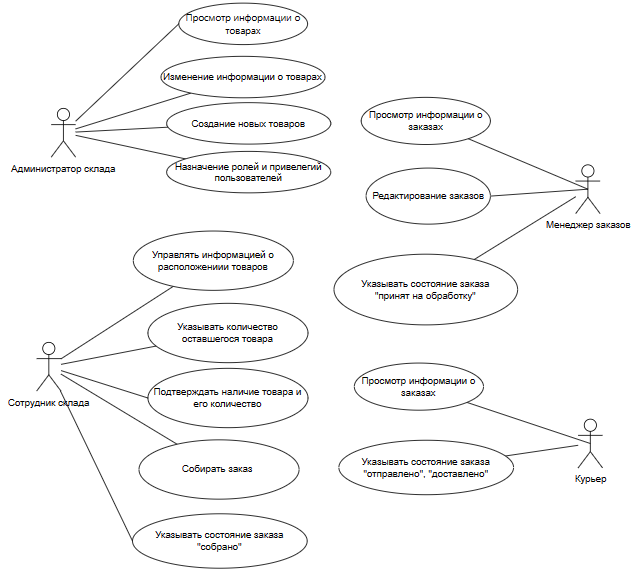


Рисунок 1 – Uml диаграмма пользовательских ролей

В ходе анализа было определено, что будут созданы следующие таблицы: Item, Category, StorageLocation, Warehouse, Supplier, Orders, OrderInf, Courier.

Для наилучшего понимания связей между таблицами и базой данных была разработана логическая схема разработанной базы данных со всеми связями:

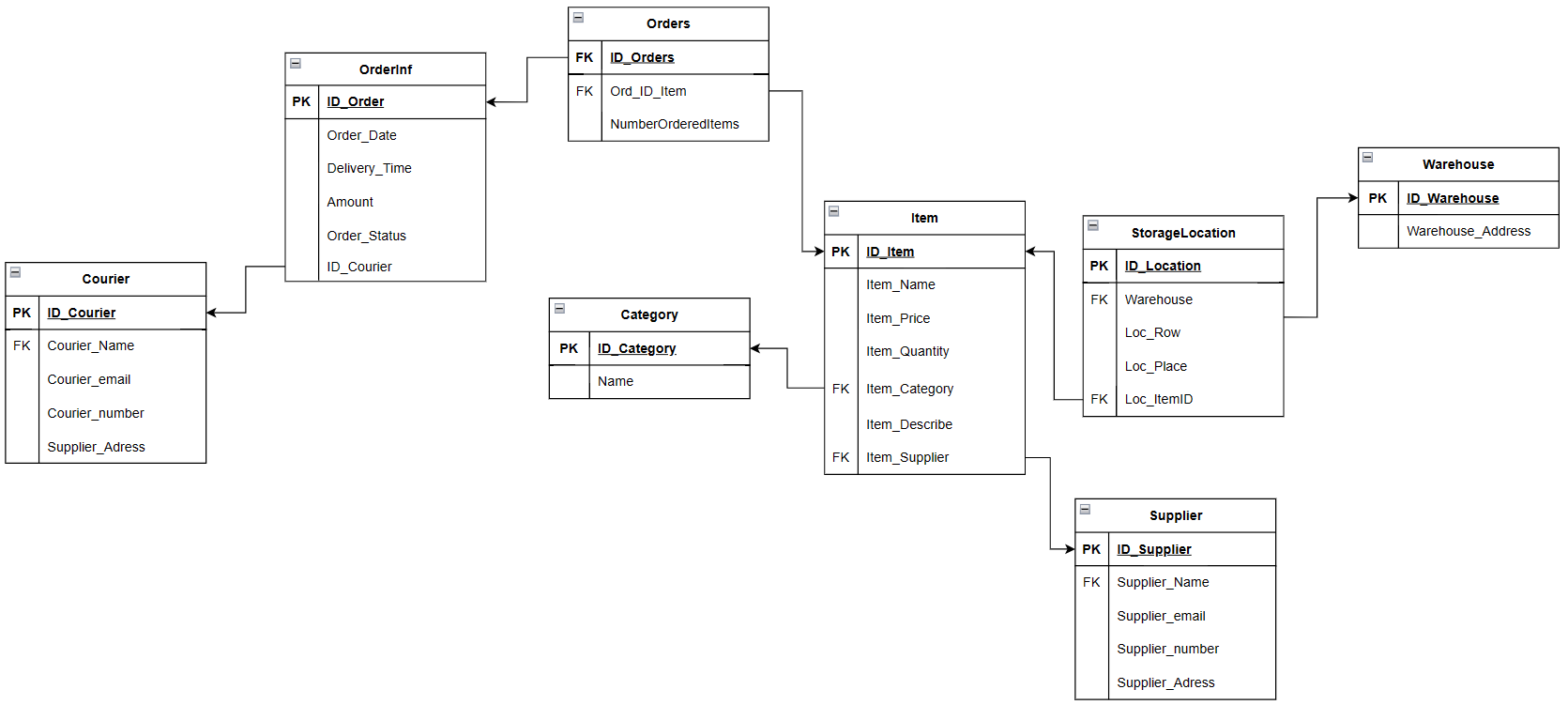


Рисунок 2 – Логическая схема БД

Для наилучшего понимания связей между таблицами и базой данных была разработана логическая схема разработанной базы данных со всеми связями:

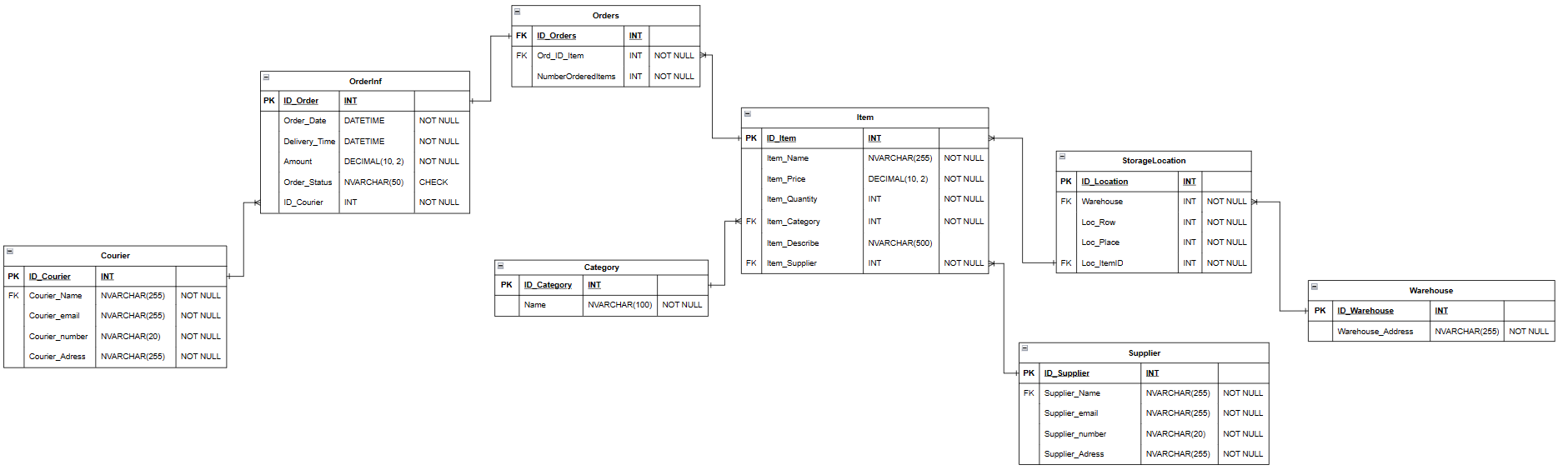


Рисунок 3 – Физическая схема БД

Нормальные формы до 4НФ включительно:

1. Таблица "Товары":

Нормализация: Таблица находится в первой нормальной форме (1NF), так как все атрибуты являются атомарными и уникальный идентификатор (ID товара) определяет каждую запись. 3NF: Все неключевые атрибуты функционально зависят от первичного ключа (ID товара). Нет транзитивных функциональных зависимостей. 4NF: Нет многозначных зависимостей.

1. Таблица "Категория товара":

Нормализация: Таблица также находится в первой нормальной форме (1NF) и второй нормальной форме (2NF), так как атрибут "Название" функционально зависит от первичного ключа (ID Категории). 3NF: Атрибут "Название" функционально зависит только от первичного ключа (ID Категории). 4NF: Нет многозначных зависимостей.

1. Таблица "Заказы":

Нормализация: Таблица находится в первой нормальной форме (1NF), и она также находится во второй нормальной форме (2NF), так как атрибуты не зависят от части составного первичного ключа (ID заказа и ID товара). 3NF: Атрибуты "ID товара" и "количество заказанных товаров" зависят только от первичного ключа (ID заказа).4NF: Нет многозначных зависимостей.

1. Таблица "Информация о заказе":

Нормализация: Таблица находится в первой нормальной форме (1NF), и она также находится во второй нормальной форме (2NF), так как атрибуты не зависят от части составного первичного ключа (ID заказа и ID товара). 3NF: Атрибуты "Дата заказа", "Дата доставки", "Статус заказа" и "Сумма заказа" зависят только от первичного ключа (ID заказа).4NF: Нет многозначных зависимостей.

1. Таблица "Места хранения":

Нормализация: Таблица находится в первой нормальной форме (1NF), так как все атрибуты атомарны и уникальный идентификатор (ID места) определяет каждую запись. 3NF: Атрибуты "номер склада", "ряд", "номер места" и "ID товара" зависят только от первичного ключа (ID места).4NF: Нет многозначных зависимостей.

1. Таблица "Склад":

Нормализация: Таблица находится в первой нормальной форме (1NF), и она находится во второй нормальной форме (2NF), так как атрибут "Адрес" функционально зависит от первичного ключа (Номер склада). 3NF: Атрибут "Адрес" зависит только от первичного ключа (Номер склада).4NF: Нет многозначных зависимостей.

1. Таблица "Поставщики":

Нормализация: Таблица также находится в первой нормальной форме (1NF) и второй нормальной форме (2NF), так как атрибут "Название поставщика" функционально зависит от первичного ключа (ID поставщика). 3NF: Атрибуты "Название поставщика", "Адрес поставщика", "Контактное лицо", "Телефон поставщика" и "Электронная почта поставщика" зависят только от первичного ключа (ID поставщика).4NF: Нет многозначных зависимостей.

Денормализованные атрибуты:

Order.Amount — используется для подсчета суммы заказов;

Item\_Quantity — хранит количество предметов;

NumberOrderedItems — подсчитывает количество заказанных предметов.

Код для создания БД со всеми таблицами в MS SQL Server:

|  |
| --- |
| --Create database MSCHOIAD\_labs  -- Категория товара  CREATE TABLE Category (  ID\_Category INT PRIMARY KEY,  Category\_Name NVARCHAR(100) NOT NULL  );  -- Поставщик  CREATE TABLE Supplier (  ID\_Supplier INT PRIMARY KEY,  Supplier\_Name NVARCHAR(255) NOT NULL,  Supplier\_email NVARCHAR(255),  Supplier\_number NVARCHAR(20),  Supplier\_Adress NVARCHAR(255)  );  -- Cклад  CREATE TABLE Warehouse (  ID\_Warehouse INT PRIMARY KEY,  Warehouse\_Address NVARCHAR(255)  );  -- Товары  CREATE TABLE Item (  ID\_Item INT PRIMARY KEY,  Item\_Name NVARCHAR(255) NOT NULL,  Item\_Category INT,  Item\_Price DECIMAL(10, 2) NOT NULL,  Item\_Quantity INT NOT NULL,  Item\_Describe NVARCHAR(500),  Item\_Supplier INT, -- Внешний ключ к "Поставщики"  FOREIGN KEY (Item\_Category) REFERENCES Category(ID\_Category),  FOREIGN KEY (Item\_Supplier) REFERENCES Supplier(ID\_Supplier)  );  -- Заказы  CREATE TABLE Orders (  ID\_Orders INT PRIMARY KEY,  Ord\_ID\_Item INT, -- Внешний ключ к "Товары"  NumberOrderedItems INT NOT NULL,  FOREIGN KEY (Ord\_ID\_Item) REFERENCES Item(ID\_Item)  );  -- Информация о заказе  CREATE TABLE OrderInf (  ID\_Order INT PRIMARY KEY,  Order\_Date DATETIME,  Delivery\_Time DATETIME,  Order\_status NVARCHAR(50) CHECK (Order\_status IN ('Собирается', 'Собрано', 'Отправлено', 'Доставлено')),  Amount DECIMAL(10, 2) NOT NULL,  FOREIGN KEY (ID\_Order) REFERENCES Orders(ID\_Orders)  );  -- Места хранения  CREATE TABLE StorageLocation (  ID\_Location INT PRIMARY KEY,  Warehouse INT, -- Внешний ключ к "Cклад"  Loc\_Row INT,  Loc\_Place INT,  Loc\_ItemID INT, -- Внешний ключ к "Товары"  FOREIGN KEY (Warehouse) REFERENCES Warehouse(ID\_Warehouse),  FOREIGN KEY (Loc\_ItemID) REFERENCES Item(ID\_Item)  ); |

Код для создания Бд со всеми таблицами в Oracle:

|  |
| --- |
| -- Создание таблицы "Категория товара"  CREATE TABLE Category (  ID\_Category NUMBER(10) PRIMARY KEY,  Category\_Name NVARCHAR2(100 CHAR) NOT NULL  );  -- Создание таблицы "Поставщик"  CREATE TABLE Supplier (  ID\_Supplier NUMBER(10) PRIMARY KEY,  Supplier\_Name NVARCHAR2(255 CHAR) NOT NULL,  Supplier\_email NVARCHAR2(255 CHAR) NOT NULL,  Supplier\_number NVARCHAR2(20 CHAR) NOT NULL,  Supplier\_Adress NVARCHAR2(255 CHAR) NOT NULL  );  -- Создание таблицы "Склад"  CREATE TABLE Warehouse (  ID\_Warehouse NUMBER(10) PRIMARY KEY,  Warehouse\_Address NVARCHAR2(255 CHAR) NOT NULL  );  -- Создание таблицы "Товары"  CREATE TABLE Item (  ID\_Item NUMBER(10) PRIMARY KEY,  Item\_Name NVARCHAR2(255 CHAR) NOT NULL,  Item\_Category NUMBER(10) NOT NULL,  Item\_Price NUMBER(10, 2) NOT NULL,  Item\_Quantity NUMBER(10) NOT NULL,  Item\_Describe NVARCHAR2(500 CHAR) NOT NULL,  Item\_Supplier NUMBER(10) NOT NULL, -- Внешний ключ к "Поставщики"  CONSTRAINT fk\_Item\_Category FOREIGN KEY (Item\_Category) REFERENCES Category(ID\_Category),  CONSTRAINT fk\_Item\_Supplier FOREIGN KEY (Item\_Supplier) REFERENCES Supplier(ID\_Supplier)  );  -- Создание таблицы "Заказы"  CREATE TABLE Orders (  ID\_Orders NUMBER(10) PRIMARY KEY,  Ord\_ID\_Item NUMBER(10) NOT NULL, -- Внешний ключ к "Товары"  NumberOrderedItems NUMBER(10) NOT NULL,  CONSTRAINT fk\_Ord\_ID\_Item FOREIGN KEY (Ord\_ID\_Item) REFERENCES Item(ID\_Item)  );  -- Создание таблицы "Информация о заказе"  CREATE TABLE OrderInf (  ID\_Order NUMBER(10) PRIMARY KEY,  Order\_Date TIMESTAMP NOT NULL,  Delivery\_Time TIMESTAMP NOT NULL,  Order\_status NVARCHAR2(50 CHAR) CHECK (Order\_status IN ('Собирается', 'Собрано', 'Отправлено', 'Доставлено')),  Amount NUMBER(10, 2) NOT NULL,  ID\_Courier NUMBER(10) NOT NULL,  CONSTRAINT fk\_ID\_Order FOREIGN KEY (ID\_Order) REFERENCES Orders(ID\_Orders),  CONSTRAINT fk\_ID\_Courier FOREIGN KEY (ID\_Courier) REFERENCES Courier(ID\_Courier)  );  -- Создание таблицы "Места хранения"  CREATE TABLE StorageLocation (  ID\_Location NUMBER(10) PRIMARY KEY,  Warehouse NUMBER(10) NOT NULL, -- Внешний ключ к "Склад"  Loc\_Row NUMBER(10) NOT NULL,  Loc\_Place NUMBER(10) NOT NULL,  Loc\_ItemID NUMBER(10) NOT NULL, -- Внешний ключ к "Товары"  CONSTRAINT fk\_Warehouse FOREIGN KEY (Warehouse) REFERENCES Warehouse(ID\_Warehouse),  CONSTRAINT fk\_Loc\_ItemID FOREIGN KEY (Loc\_ItemID) REFERENCES Item(ID\_Item)  );  -- Создание таблицы "Курьер"  CREATE TABLE Courier (  ID\_Courier NUMBER(10) PRIMARY KEY,  Courier\_Name NVARCHAR2(255 CHAR) NOT NULL,  Courier\_email NVARCHAR2(255 CHAR) NOT NULL,  Courier\_number NVARCHAR2(20 CHAR) NOT NULL,  Courier\_Adress NVARCHAR2(255 CHAR) NOT NULL  ); |

Вывод: в ходе выполнения данной лабораторной работы была разработана база данных по теме “Склад”. Был продуман и выполнен в виде диаграммы Use Case и логическая схема БД. На основе данной схемы были созданы БД и таблицы в MS SQL Serve и Oracle.