

ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ И НАУКИ ГОРОДА МОСКВЫ

Государственное автономное образовательное учреждение

высшего образования города Москвы

«Московский городской педагогический университет»

(ГАОУ ВО МГПУ)

Институт цифрового образования

Департамент информатики, управления и технологий

Лабораторная №2 (19 вариант)

по дисциплине «Базы данных»

на тему:

Разработка ER-диаграммы для заданной предметной области. Применение
нормализации к созданной модели данных.

Выполнила:

студент группы ЦИБ-241

Чайковская Я.М.

Проверил:

Босенко Т.М.

Москва 2025

Цель работы:

1. овладеть практическими навыками анализа бизнес-требований для проектирования баз данных;
2. научиться строить логическую, даталогическую (реляционную) и физическую модели данных с использованием Oracle SQL Developer Data Modeler;
3. освоить принципы нормализации данных для устранения избыточности и аномалий, приводя структуру таблиц к третьей нормальной форме (3NF);
4. научиться генерировать SQL-скрипт из физической модели и выполнять его на сервере MySQL с помощью MySQL Workbench.

Задача индивидуального задания: БД управляет лекарственными препаратами (название, производитель, цена), их партиями (срок годности), поставщиками и продажами, фиксируя, какой препарат и когда был продан.

Предметная область: Процесс учета и управления лекарственными препаратами в аптечной организации. Основная цель — обеспечение контроля за наличием, движением и продажей лекарственных средств. В систему заносятся данные о лекарственных препаратах, включая их наименование, производителя и цену. Каждое лекарственное средство поступает в виде партии, для которой указываются срок годности, дата поступления, количество и поставщик. Информация о поставщиках включает их название, контактные данные и реквизиты, что позволяет отслеживать, от кого была получена каждая партия лекарств. При продаже фиксируется, какой именно препарат (из какой партии) был реализован, дата продажи, количество проданных единиц и сумма продажи.

Практическая часть:

Логическая ER-диаграмма из Oracle SQL Developer Data Modeler.

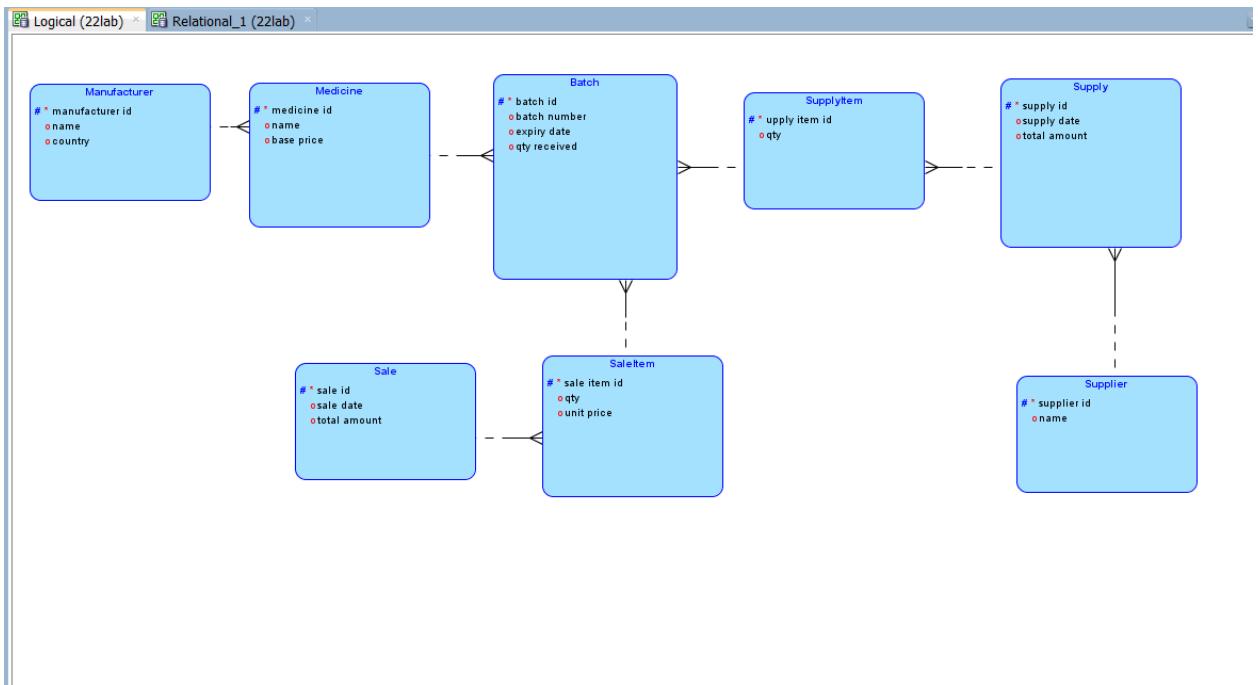


Рисунок 1. логическая ER-диаграмма из Oracle SQL Developer Data Modeler.

Реляционная модель и нормализация.

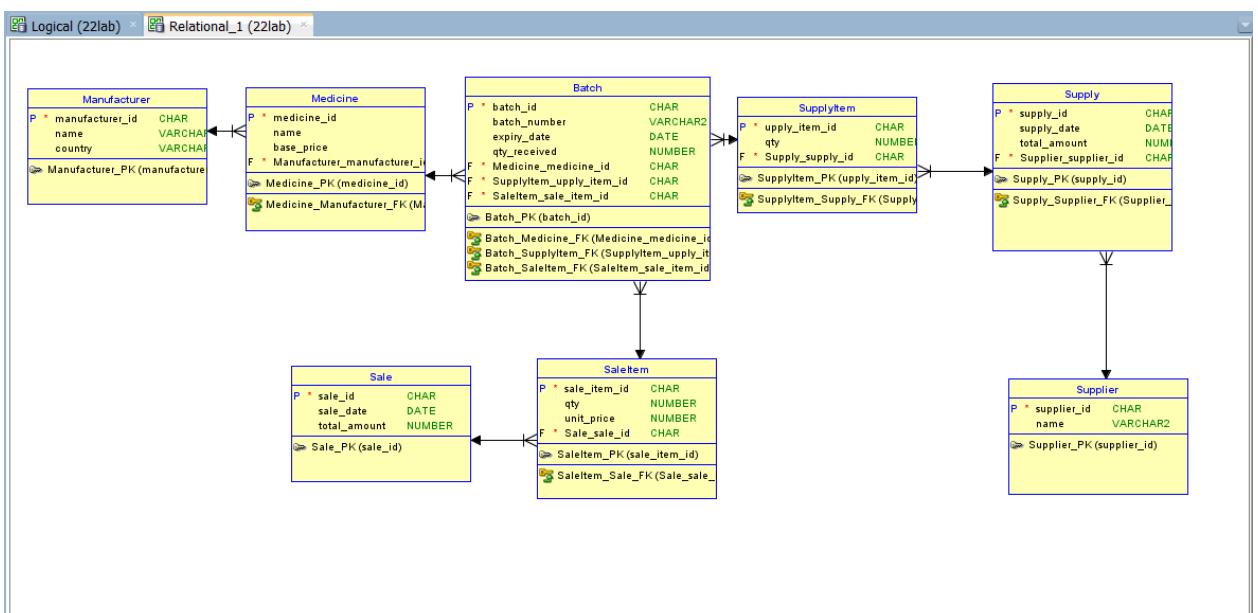
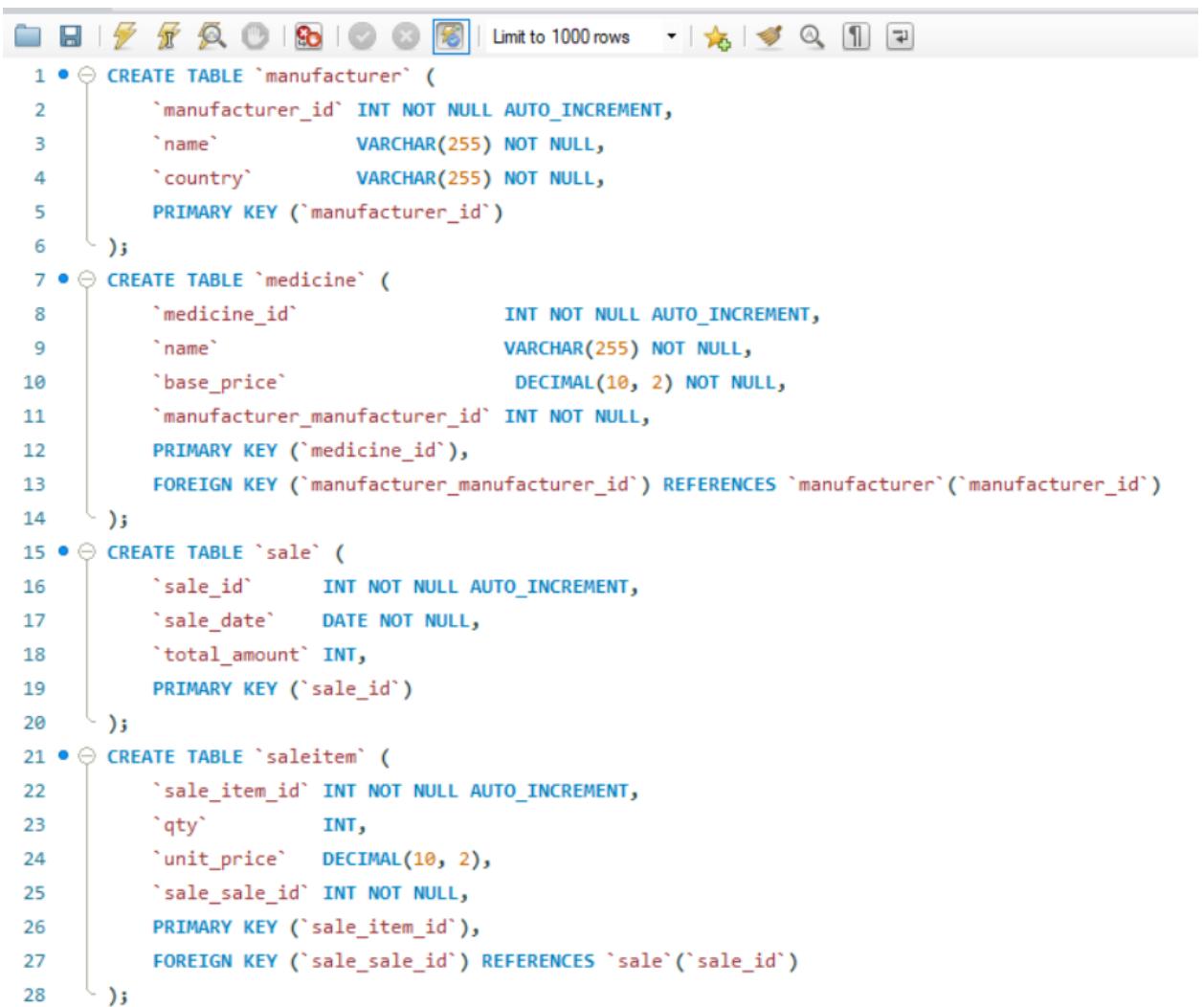


Рисунок 2. реляционная модель из Oracle SQL Developer Data Modeler.

Все таблицы базы данных находятся в третьей нормальной форме (3NF), потому что:

1. Все атрибуты атомарны (1NF выполнена);
2. Все неключевые атрибуты зависят только от первичного ключа (2NF выполнена);
3. Нет транзитивных зависимостей между неключевыми атрибутами (3NF выполнена);
4. Связанные данные (о поставщике, производителе, продажах и партиях) вынесены в отдельные таблицы и соединяются через внешние ключи.

Физическая модель и SQL-скрипт.



The screenshot shows the MySQL Workbench interface with the SQL editor tab active. The code area contains the following SQL script:

```
1 • CREATE TABLE `manufacturer` (
2     `manufacturer_id` INT NOT NULL AUTO_INCREMENT,
3     `name`           VARCHAR(255) NOT NULL,
4     `country`        VARCHAR(255) NOT NULL,
5     PRIMARY KEY (`manufacturer_id`)
6 );
7 • CREATE TABLE `medicine` (
8     `medicine_id`      INT NOT NULL AUTO_INCREMENT,
9     `name`             VARCHAR(255) NOT NULL,
10    `base_price`       DECIMAL(10, 2) NOT NULL,
11    `manufacturer_id` INT NOT NULL,
12    PRIMARY KEY (`medicine_id`),
13    FOREIGN KEY (`manufacturer_id`) REFERENCES `manufacturer`(`manufacturer_id`)
14 );
15 • CREATE TABLE `sale` (
16     `sale_id`         INT NOT NULL AUTO_INCREMENT,
17     `sale_date`       DATE NOT NULL,
18     `total_amount`   INT,
19     PRIMARY KEY (`sale_id`)
20 );
21 • CREATE TABLE `saleitem` (
22     `sale_item_id`   INT NOT NULL AUTO_INCREMENT,
23     `qty`            INT,
24     `unit_price`    DECIMAL(10, 2),
25     `sale_sale_id`  INT NOT NULL,
26     PRIMARY KEY (`sale_item_id`),
27     FOREIGN KEY (`sale_sale_id`) REFERENCES `sale`(`sale_id`)
28 );
```

Рисунок 3. финальный SQL-код часть 1

```
29 • CREATE TABLE `supplier` (
30     `supplier_id` INT NOT NULL AUTO_INCREMENT,
31     `name`         VARCHAR(255),
32     PRIMARY KEY (`supplier_id`)
33 );
34 • CREATE TABLE `supply` (
35     `supply_id`      INT NOT NULL AUTO_INCREMENT,
36     `supply_date`    DATE,
37     `total_amount`   INT,
38     `supplier_supplier_id` INT NOT NULL,
39     PRIMARY KEY (`supply_id`),
40     FOREIGN KEY (`supplier_supplier_id`) REFERENCES `supplier`(`supplier_id`)
41 );
42 • CREATE TABLE `supplyItem` (
43     `supply_item_id`  INT NOT NULL AUTO_INCREMENT,
44     `qty`             INT,
45     `supply_supply_id` INT NOT NULL,
46     PRIMARY KEY (`supply_item_id`),
47     FOREIGN KEY (`supply_supply_id`) REFERENCES `supply`(`supply_id`)
48 );
49 • CREATE TABLE `batch` (
50     `batch_id`        INT NOT NULL AUTO_INCREMENT,
51     `batch_number`    VARCHAR(255),
52     `expiry_date`    DATE,
53     `qty_received`   INT,
54     `medicine_medicine_id` INT NOT NULL,
55     `supplyitem_supply_item_id` INT NOT NULL,
56     `saleitem_sale_item_id` INT NOT NULL,
57     PRIMARY KEY (`batch_id`),
58     FOREIGN KEY (`medicine_medicine_id`) REFERENCES `medicines`(`medicine_id`),
59     FOREIGN KEY (`supplyitem_supply_item_id`) REFERENCES `supplyitem`(`supply_item_id`),
60     FOREIGN KEY (`saleitem_sale_item_id`) REFERENCES `saleitem`(`sale_item_id`)
61 );
```

Рисунок 4. финальный SQL-код часть 2.

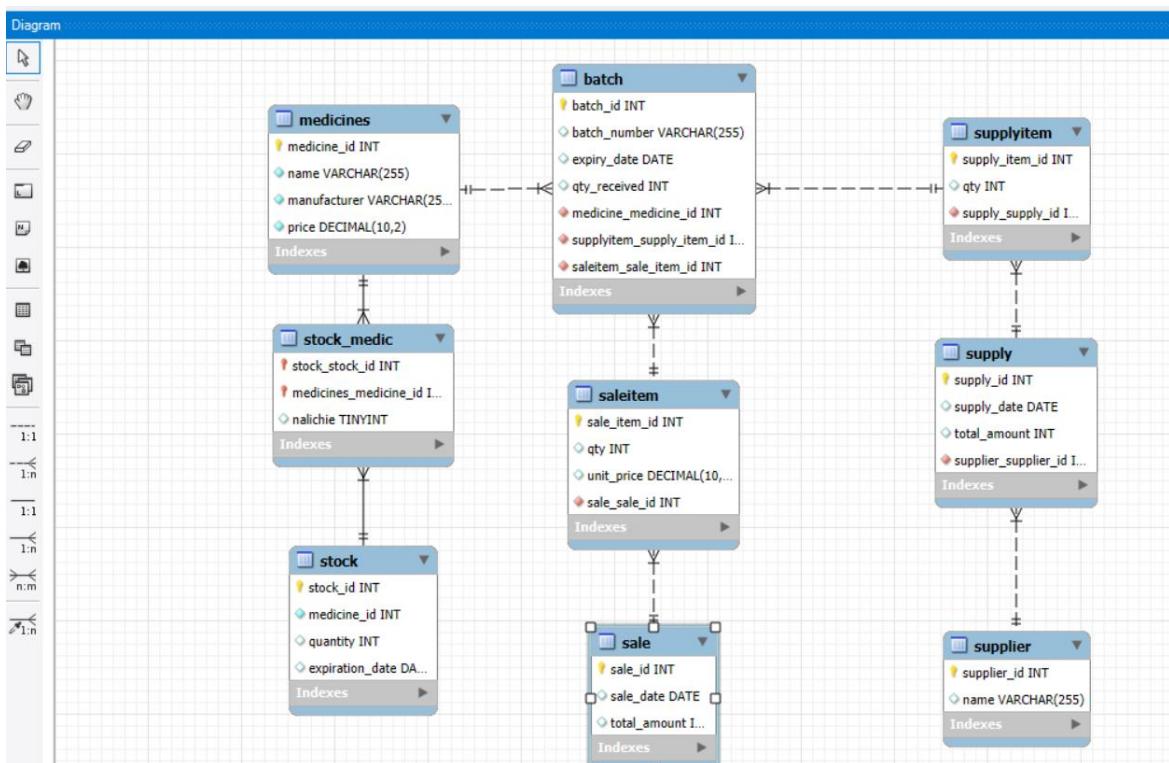


Рисунок 5. выполнение SQL-кода

Выводы:

В ходе выполнения работы были приобретены практические навыки анализа бизнес-требований и проектирования баз данных. На основе исходных данных были построены логическая, реляционная (даталогическая) и физическая модели данных с использованием Oracle SQL Developer Data Modeler. При проектировании структуры таблиц были применены принципы нормализации, что позволило привести базу данных к третьей нормальной форме (3NF) и устраниТЬ избыточность данных. На основании физической модели был сгенерирован SQL-скрипт, который успешно выполнен на сервере MySQL с помощью MySQL Workbench. Полученная база данных соответствует требованиям предметной области и обеспечивает целостность и согласованность данных.