**MINISTERUL EDUCAȚIEI, CULTURII ȘI CERCETĂRII**

**AL REPUBLICII MOLDOVA**

**Centrul de Excelenţă în Informatică şi Tehnologii Informaţionale**

**RAPORT**

**LA PRACTICA TEHNOLOGICĂ SPECIALITATEA "Programare şi analiza produselor program"**

**TEMA : "Продажа компьютерной техники"**

A elaborat elevul

*Терзи Кирилл,* P-1834r

Conducătorul practicii

*Şarapanovscaia Irina*

Chişinău 2020

# Содержание

[**Введение**. 3](#_Toc36457122)

[**Постановка задачи** 4](#_Toc36457123)

[*Словесное описание* 4](#_Toc36457124)

[*Техническое задание по проекту* 4](#_Toc36457125)

[**Проектирование базы данных** 4](#_Toc36457126)

[*ER диаграмма* 4](#_Toc36457127)

[*Нормализация* 4](#_Toc36457128)

[*Реляционная модель* 4](#_Toc36457129)

[**Создание Базы данных** 4](#_Toc36457130)

[*Диаграмма Базы данных* 4](#_Toc36457131)

[*Таблицы Базы данных* 4](#_Toc36457132)

[*Команды SQL создания БД и таблиц* 4](#_Toc36457133)

[*Запросы* 4](#_Toc36457134)

[*Представления* 4](#_Toc36457135)

[*Хранимые процедуры* 4](#_Toc36457136)

[**Создание приложения** 4](#_Toc36457137)

[*Авторизация* 4](#_Toc36457138)

[*Описание интерфейса приложения* 4](#_Toc36457139)

[Просмотр 4](#_Toc36457140)

[Редактирование 4](#_Toc36457141)

[Вычисления 4](#_Toc36457142)

[*Создание отчетов* 5](#_Toc36457143)

[**Поддержка приложения** 5](#_Toc36457144)

[*Создание резервных копий* 5](#_Toc36457145)

[*Восстановление данных* 5](#_Toc36457146)

[**Безопасность данных** 5](#_Toc36457147)

[**Создание установочного файла** 5](#_Toc36457148)

[**Руководство пользователя** 5](#_Toc36457149)

[**Вывод** 6](#_Toc36457150)

[**Библиография** 7](#_Toc36457151)

[**Приложение** 8](#_Toc36457152)

# **Введение**.

Общее описание практики

Технологическая практика преследует достижение общих целей:

- накопление студентами практических навыков и опыта создания общей структуры визуального приложения;

- настройка компонентов визуального приложения;

- управление компонентами визуального приложения;

- разработка визуального приложения в соответствии с техническими условиями;

- поддержка визуальных приложений;

- использование визуального программирования для разработки приложений;

- изучение механизма подключения базы данных к программируемому приложению;

- применение методов восстановления данных и создание резервных копий;

- соблюдение стандартов, касающихся способа оформления документов.

Период практики

Время практики

Перечисление использованных инструментов(Visual Studio 2017, SQL Manager Server Studio и.д.)

# **Постановка задачи**

## *Словесное описание*

В разработанной базе данных будут храниться данные о пользователях, товарах и товарах, которые были приобретены. В базе данных будет использовано 8 таблиц разных типов товаров: *процессоры, SSD накопители, HDD накопители, видеокарты, водяное охлаждение, башенное охлаждение, блоки питания, оперативная память*. У каждой из этих таблиц есть идентичные поля данных: ID *товара, название товара, цена*. Остальные атрибуты таблиц с товарами зависят от типа товара. В таблице ***Пользователи*** будет храниться вся необходимая информация о покупателе: ID пользователя, *логин, имя, фамилия, почта, пароль(зашифрованный), роль пользователя, дату создания и дату последней авторизации.* В таблице ***Корзина*** будет храниться информация о товарах, которые пользователь желает приобрести, и она содержит поля: *ID,* ID пользователя, ID *товара, количество товара и дату добавления товара.* Весь товар, находящийся в ***Корзине***, можно приобрести и в таблице ***Заказы*** будут заполнены поля: *ID,* ID пользователя, ID *товара, цена, количество товара и дату платежа.*

## *Техническое задание по проекту*

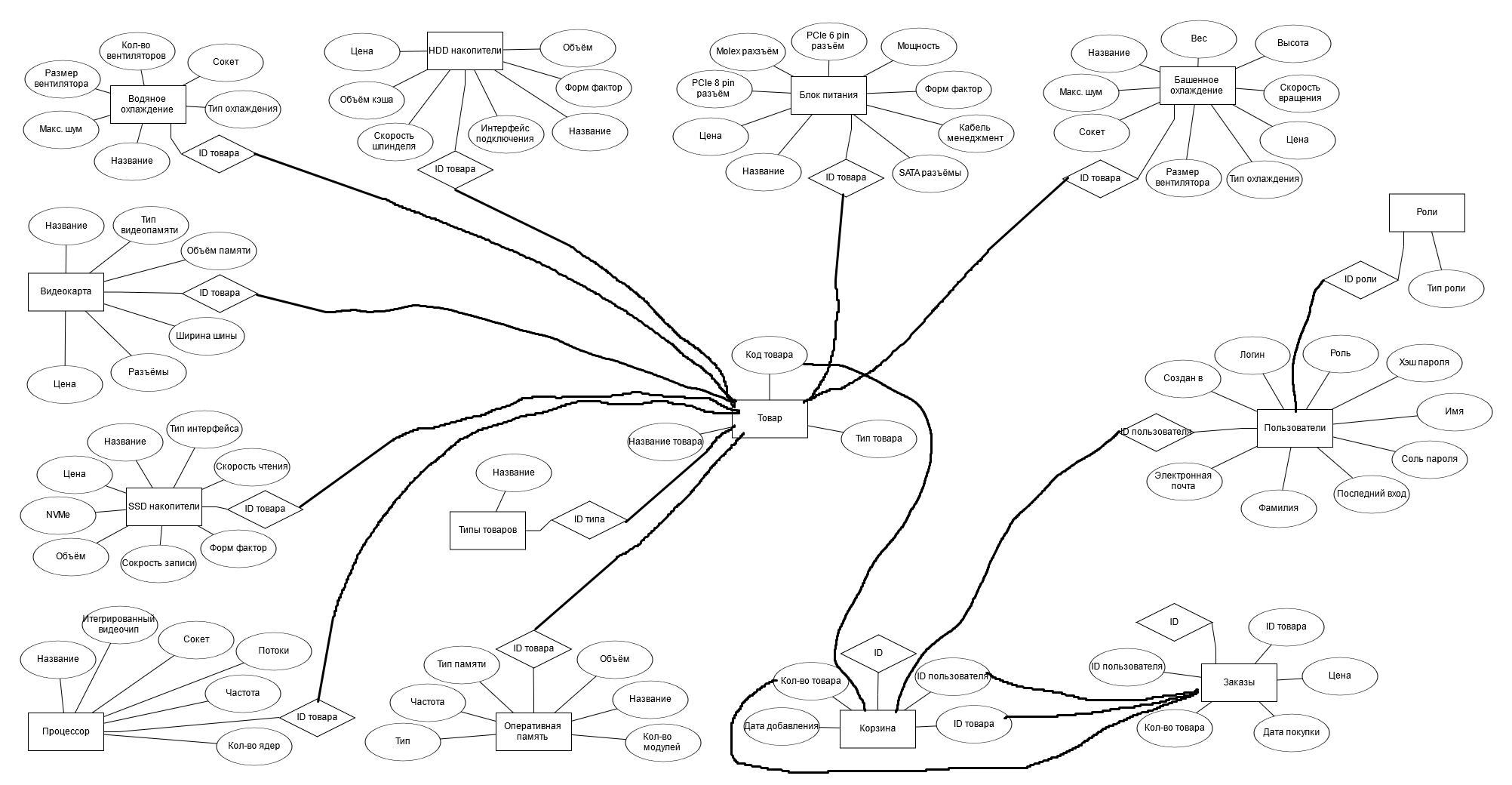
Для приобретения товаров необходимо зарегистрироваться в приложении, создав новый, уникальный логин и пароль, ввести своё имя, фамилию и электронную почту. После авторизации используя логин и пароль необходимо зайти в приложение. После авторизации появляется возможность добавить товар в корзину и после совершить покупку товаров, которые находятся в корзине. После оплаты пользователю предоставляется чек, в котором указаны: имя, фамилия, название продукта, количество, цена и дата осуществления платежа.

Будет два типа пользователей:

1. ***Покупатель*** - имеет возможность просматривать товар из разных таблиц, добавлять товар в корзину и затем осуществлять покупку товаров.
2. ***Администратор***, помимо возможностей обычного пользователя, также имеет возможность изменять значения в таблицах (INSERT, UPDATE, DELETE).

# **Проектирование базы данных**

## *ER диаграмма*

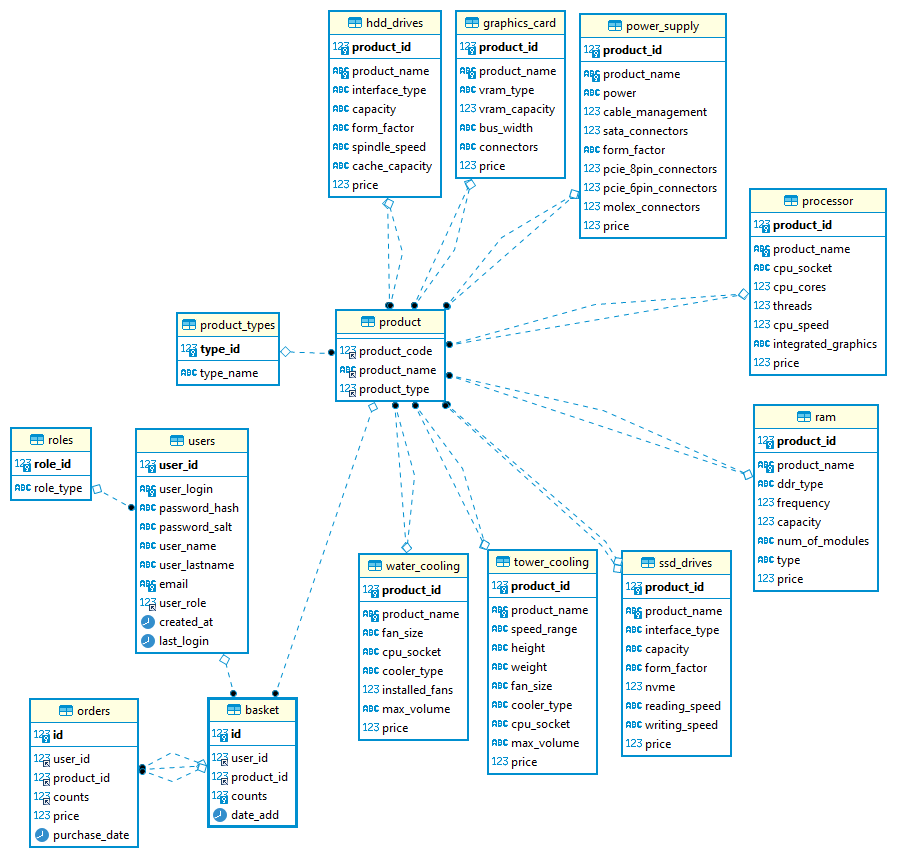


## *Нормализация*

Моя база данных находится в третьей нормальной форме и не нуждается в дальнейшей нормализации для нынешних целей и возможностей программы.

Требование третьей нормальной формы (3NF) заключается в том, чтобы в таблицах отсутствовала транзитивная зависимость. Таблица должна содержать правильные неключевые столбцы.

## *Реляционная модель*



# **Создание Базы данных**

## *Диаграмма Базы данных*

## *Таблицы Базы данных*

## 

Таблица 1 Graphics card

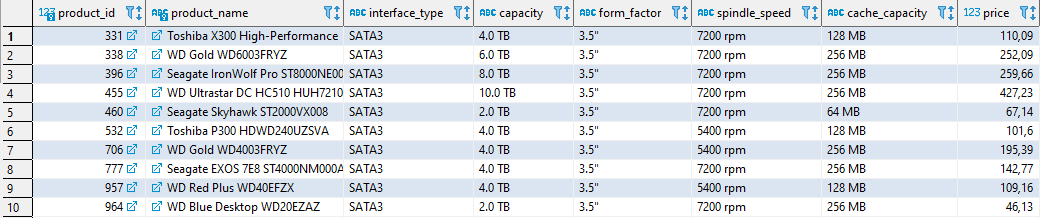


Таблица 2 HDD drives

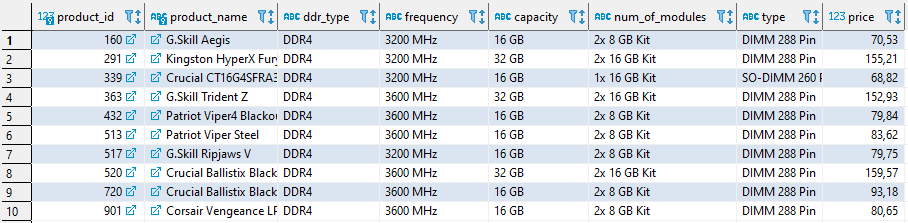


Таблица 3 RAM

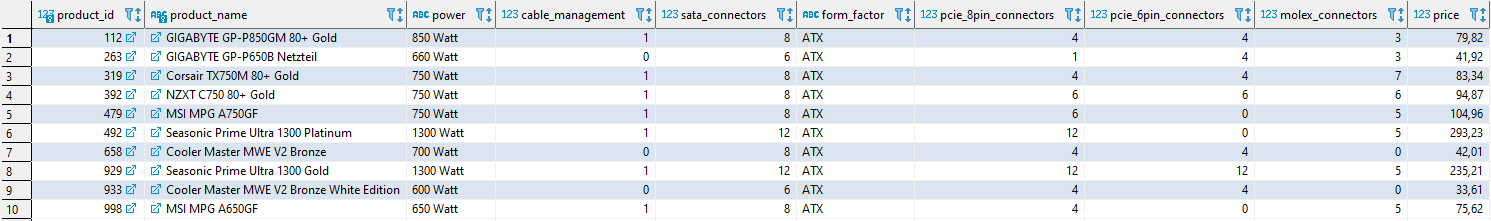


Таблица 4 Power supply

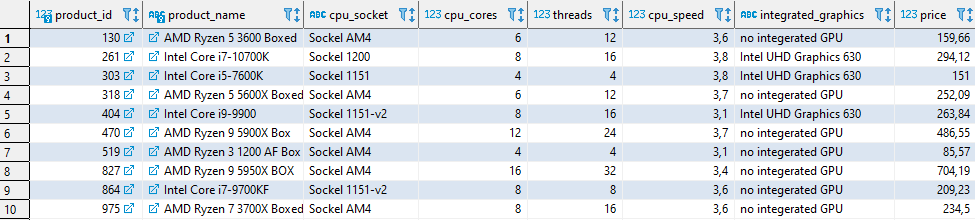


Таблица 5 Processor

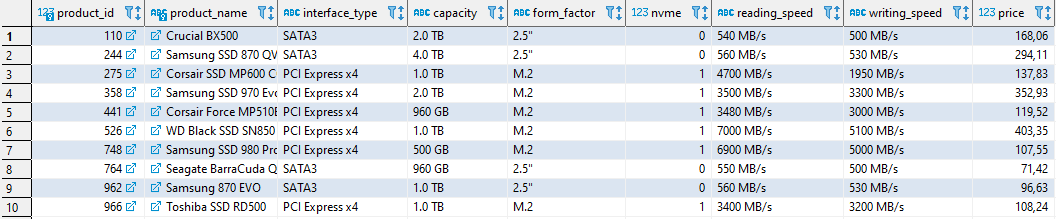


Таблица 6 SSD drives

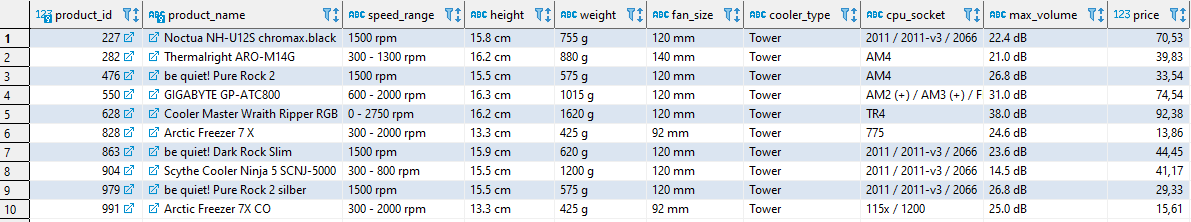


Таблица 7 Tower cooling

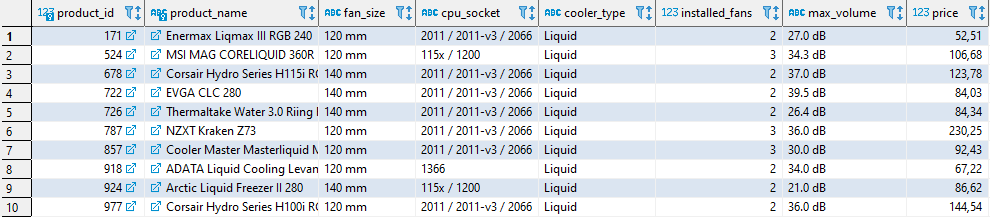


Таблица 8 Water cooling

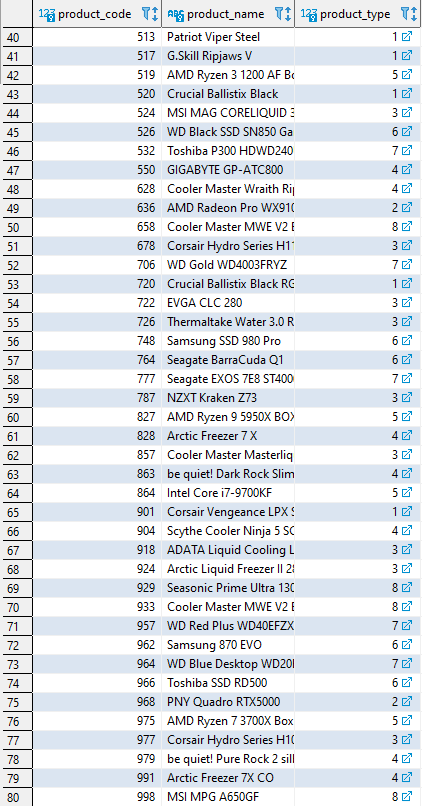


Таблица 9 Product

## *Команды SQL создания БД и таблиц*

**CREATE** **TABLE** `ram` **(**

`product\_id` int **NOT** **NULL,**

`product\_name` varchar**(**40**)** **NOT** **NULL,**

`ddr\_type` varchar**(**12**)** **NOT** **NULL,**

`frequency` varchar**(**20**)** **NOT** **NULL,**

`capacity` varchar**(**10**)** **NOT** **NULL,**

`num\_of\_modules` varchar**(**12**)** **not** **null,**

`**type**` varchar**(**15**)** **not** **null,**

`price` float **NOT** **NULL,**

**PRIMARY** **KEY** **(**`product\_id`**),**

**CONSTRAINT** FK\_product\_id **UNIQUE** **(**product\_id**),**

**CONSTRAINT** FK\_product\_name **UNIQUE** **(**product\_name**),**

**FOREIGN** **KEY** **(**product\_id**)** **REFERENCES** product**(**product\_code**)** **ON** **DELETE** **CASCADE,**

**FOREIGN** **KEY** **(**product\_name**)** **REFERENCES** product**(**product\_name**)** **ON** **DELETE** **CASCADE**

**);**

**CREATE** **TABLE** power\_supply **(**

product\_id INT **NOT** **NULL,**

product\_name VARCHAR**(**50**)** **NOT** **NULL,**

**power** VARCHAR**(**15**)** **NOT** **NULL,**

cable\_management BOOLEAN **NOT** **NULL,**

sata\_connectors INT **NOT** **NULL,**

form\_factor VARCHAR**(**15**)** **NOT** **NULL,**

pcie\_8pin\_connectors INT **NOT** **NULL,**

pcie\_6pin\_connectors INT **NOT** **NULL,**

molex\_connectors INT **NOT** **NULL,**

price FLOAT **NOT** **NULL,**

**PRIMARY** **KEY** **(**product\_id**),**

**CONSTRAINT** FK\_product\_id **UNIQUE** **(**product\_id**),**

**CONSTRAINT** FK\_product\_name **UNIQUE** **(**product\_name**),**

**FOREIGN** **KEY** **(**product\_id**)** **REFERENCES** product**(**product\_code**)** **ON** **DELETE** **CASCADE,**

**FOREIGN** **KEY** **(**product\_name**)** **REFERENCES** product**(**product\_name**)** **ON** **DELETE** **CASCADE**

**);**

**CREATE** **TABLE** hdd\_drives **(**

product\_id INT **NOT** **NULL,**

product\_name VARCHAR**(**40**)** **NOT** **NULL,**

interface\_type VARCHAR**(**10**)** **NOT** **NULL,**

capacity VARCHAR**(**10**)** **NOT** **NULL,**

form\_factor VARCHAR**(**10**)** **NOT** **NULL,**

spindle\_speed VARCHAR**(**10**)** **NOT** **NULL,**

cache\_capacity VARCHAR**(**10**)** **NOT** **NULL,**

price FLOAT **NOT** **NULL,**

**PRIMARY** **KEY** **(**product\_id**),**

**CONSTRAINT** FK\_product\_id **UNIQUE** **(**product\_id**),**

**CONSTRAINT** FK\_product\_name **UNIQUE** **(**product\_name**),**

**FOREIGN** **KEY** **(**product\_id**)** **REFERENCES** product**(**product\_code**)** **ON** **DELETE** **CASCADE,**

**FOREIGN** **KEY** **(**product\_name**)** **REFERENCES** product**(**product\_name**)** **ON** **DELETE** **CASCADE**

**);**

**CREATE** **TABLE** ssd\_drives **(**

product\_id INT **NOT** **NULL,**

product\_name VARCHAR**(**40**)** **NOT** **NULL,**

interface\_type VARCHAR**(**20**)** **NOT** **NULL,**

capacity VARCHAR**(**10**)** **NOT** **NULL,**

form\_factor VARCHAR**(**10**)** **NOT** **NULL,**

nvme BOOLEAN **NOT** **NULL,**

reading\_speed VARCHAR**(**10**)** **NOT** **NULL,**

writing\_speed VARCHAR**(**10**)** **NOT** **NULL,**

price FLOAT **NOT** **NULL,**

**PRIMARY** **KEY** **(**product\_id**),**

**CONSTRAINT** FK\_product\_id **UNIQUE** **(**product\_id**),**

**CONSTRAINT** FK\_product\_name **UNIQUE** **(**product\_name**),**

**FOREIGN** **KEY** **(**product\_id**)** **REFERENCES** product**(**product\_code**)** **ON** **DELETE** **CASCADE,**

**FOREIGN** **KEY** **(**product\_name**)** **REFERENCES** product**(**product\_name**)** **ON** **DELETE** **CASCADE**

**);**

**CREATE** **TABLE** tower\_cooling **(**

product\_id INT **NOT** **NULL,**

product\_name VARCHAR**(**40**)** **NOT** **NULL,**

speed\_range VARCHAR**(**25**)** **NOT** **NULL,**

height VARCHAR**(**10**)** **NOT** **NULL,**

weight VARCHAR**(**10**)** **NOT** **NULL,**

fan\_size VARCHAR**(**10**)** **NOT** **NULL,**

cooler\_type VARCHAR**(**10**)** **NOT** **NULL,**

cpu\_socket VARCHAR**(**40**)** **NOT** **NULL,**

max\_volume VARCHAR**(**10**)** **NOT** **NULL,**

price FLOAT **NOT** **NULL,**

**PRIMARY** **KEY** **(**product\_id**),**

**CONSTRAINT** FK\_product\_id **UNIQUE** **(**product\_id**),**

**CONSTRAINT** FK\_product\_name **UNIQUE** **(**product\_name**),**

**FOREIGN** **KEY** **(**product\_id**)** **REFERENCES** product**(**product\_code**)** **ON** **DELETE** **CASCADE,**

**FOREIGN** **KEY** **(**product\_name**)** **REFERENCES** product**(**product\_name**)** **ON** **DELETE** **CASCADE**

**);**

**CREATE** **TABLE** water\_cooling **(**

product\_id INT **NOT** **NULL,**

product\_name VARCHAR**(**40**)** **NOT** **NULL,**

fan\_size VARCHAR**(**10**)** **NOT** **NULL,**

cpu\_socket VARCHAR**(**25**)** **NOT** **NULL,**

cooler\_type VARCHAR**(**10**)** **NOT** **NULL,**

installed\_fans INT **NOT** **NULL,**

max\_volume VARCHAR**(**10**)** **NOT** **NULL,**

price FLOAT **NOT** **NULL,**

**PRIMARY** **KEY** **(**product\_id**),**

**CONSTRAINT** FK\_product\_id **UNIQUE** **(**product\_id**),**

**CONSTRAINT** FK\_product\_name **UNIQUE** **(**product\_name**),**

**FOREIGN** **KEY** **(**product\_id**)** **REFERENCES** product**(**product\_code**)** **ON** **DELETE** **CASCADE,**

**FOREIGN** **KEY** **(**product\_name**)** **REFERENCES** product**(**product\_name**)** **ON** **DELETE** **CASCADE**

**);**

**CREATE** **TABLE** `processor` **(**

`product\_id` int **NOT** **NULL,**

`product\_name` varchar**(**40**)** **NOT** **NULL,**

`cpu\_socket` varchar**(**20**)** **NOT** **NULL,**

`cpu\_cores` int **NOT** **NULL,**

`threads` int **NOT** **NULL,**

`cpu\_speed` float **NOT** **NULL,**

`integrated\_graphics` varchar**(**30**)** **not** **null,**

`price` float **NOT** **NULL,**

**PRIMARY** **KEY** **(**`product\_id`**),**

**CONSTRAINT** FK\_product\_id **UNIQUE** **(**product\_id**),**

**CONSTRAINT** FK\_product\_name **UNIQUE** **(**product\_name**),**

**FOREIGN** **KEY** **(**product\_id**)** **REFERENCES** product**(**product\_code**)** **ON** **DELETE** **CASCADE,**

**FOREIGN** **KEY** **(**product\_name**)** **REFERENCES** product**(**product\_name**)** **ON** **DELETE** **CASCADE**

**);**

**CREATE** **TABLE** `graphics\_card` **(**

`product\_id` int **NOT** **NULL,**

`product\_name` varchar**(**40**)** **NOT** **NULL,**

`vram\_type` varchar**(**12**)** **NOT** **NULL,**

`vram\_capacity` int **NOT** **NULL,**

`bus\_width` varchar**(**15**)** **NOT** **NULL,**

`connectors` varchar**(**40**)** **NOT** **NULL,**

`price` float **NOT** **NULL,**

**PRIMARY** **KEY** **(**`product\_id`**),**

**CONSTRAINT** FK\_product\_id **UNIQUE** **(**product\_id**),**

**CONSTRAINT** FK\_product\_name **UNIQUE** **(**product\_name**),**

**FOREIGN** **KEY** **(**product\_id**)** **REFERENCES** product**(**product\_code**)** **ON** **DELETE** **CASCADE,**

**FOREIGN** **KEY** **(**product\_name**)** **REFERENCES** product**(**product\_name**)** **ON** **DELETE** **CASCADE**

**);**

**CREATE** **TABLE** product **(**

product\_code INT **NOT** **NULL,**

product\_name VARCHAR**(**50**)** **NOT** **NULL,**

product\_type INT **NOT** **NULL,**

**CONSTRAINT** FK\_product\_name **UNIQUE** **(**product\_name**),**

**CONSTRAINT** FK\_product\_code **UNIQUE** **(**product\_code**),**

**primary** **key** **(**product\_code**,** product\_name**,** product\_type**),**

**foreign** **key** **(**product\_type**)** **references** product\_types**(type\_id)**

**);**

**CREATE** **TABLE** product\_types **(**

**type\_id** INT **NOT** **NULL,**

**type\_name** VARCHAR**(**20**)** **NOT** **NULL,**

**PRIMARY** **KEY** **(type\_id)**

**);**

**CREATE** **TABLE** basket **(**

id INT **NOT** **NULL,**

**user\_id** bigint unsigned **NOT** **NULL,**

product\_id int **NOT** **NULL,**

counts int **NOT** **NULL,**

**date\_add** datetime **NOT** **NULL,**

**PRIMARY** **KEY** **(**id**),**

**CONSTRAINT** FK\_counts\_basket **UNIQUE** **(**counts**),**

**FOREIGN** **KEY** **(**product\_id**)** **REFERENCES** product**(**product\_code**)** **,**

**FOREIGN** **KEY** **(user\_id)** **REFERENCES** users**(user\_id)**

**);**

**CREATE** **TABLE** orders **(**

id INT **NOT** **NULL,**

**user\_id** BIGINT UNSIGNED **NOT** **NULL,**

product\_id INT **NOT** **NULL,**

counts INT **NOT** **NULL,**

price FLOAT **NOT** **NULL,**

purchase\_date DATETIME **NOT** **NULL,**

**PRIMARY** **KEY** **(**id**),**

**FOREIGN** **KEY** **(user\_id)** **REFERENCES** basket**(user\_id),**

**FOREIGN** **KEY** **(**product\_id**)** **REFERENCES** basket**(**product\_id**),**

**FOREIGN** **KEY** **(**counts**)** **REFERENCES** basket**(**counts**)**

**);**

**CREATE** **TABLE** `users` **(**

`**user\_id**` bigint unsigned **NOT** **NULL** AUTO\_INCREMENT**,**

`user\_login` varchar**(**60**)** **NOT** **NULL,**

`password\_hash` varchar**(**200**)** **NOT** **NULL,**

`password\_salt` varchar**(**20**)** **NOT** **NULL,**

`**user\_name**` varchar**(**60**)** **NOT** **NULL,**

`user\_lastname` varchar**(**60**)** **NOT** **NULL,**

`email` varchar**(**100**)** **NOT** **NULL,**

`user\_role` int **NOT** **NULL,**

`created\_at` datetime **NOT** **NULL,**

`last\_login` datetime **NOT** **NULL,**

**PRIMARY** **KEY** **(**`**user\_id**`**),**

**UNIQUE** **KEY** `**user\_id**` **(**`**user\_id**`**),**

**UNIQUE** **KEY** `user\_login` **(**`user\_login`**),**

**UNIQUE** **KEY** `email` **(**`email`**),**

**KEY** `user\_role` **(**`user\_role`**),**

**CONSTRAINT** `users\_ibfk\_1` **FOREIGN** **KEY** **(**`user\_role`**)** **REFERENCES** `roles` **(**`role\_id`**)**

**);**

**CREATE** **TABLE** `roles` **(**

`role\_id` int **NOT** **NULL,**

`role\_type` varchar**(**60**)** **NOT** **NULL,**

**PRIMARY** **KEY** **(**`role\_id`**),**

**UNIQUE** **KEY** `role\_id` **(**`role\_id`**)**

**);**

## *Запросы*

## *Представления*

## *Хранимые процедуры*

# **Создание приложения**

## *Авторизация*

Описание

Скриншот

Фрагмент кода

## *Описание интерфейса приложения*

Описание какие операции (действия) в каждом пункте меню

Скриншот

Фрагмент кода

Например, могут быть такие пункты меню:

### Просмотр

### Редактирование

### Вычисления

И т.д

## *Создание отчетов*

# **Поддержка приложения**

## *Создание резервных копий*

## *Восстановление данных*

# **Безопасность данных**

Авторизация

Транзакции

Распределение ролей

# **Создание установочного файла**

# **Руководство пользователя**

Описание пунктов меню или Help

# **Вывод**

Общие впечатления от практики

Что понравилось

Что не понравилось

Какие трудности встретил

Что помогло

Какие есть предложения по практике

# **Библиография**

# **Приложение**

Код программы