

**Образовательная автономная некоммерческая организация
высшего образования**

«МОСКОВСКИЙ ТЕХНОЛОГИЧЕСКИЙ ИНСТИТУТ»

ОТЧЕТ

о прохождении производственной практики

по профессиональному модулю ПМ.05 Сoadминистрирование и автоматизация
баз данных и серверов _____

шифр и номер группы

_____Иванов Иван Иванович_____

(Ф.И.О.)

Содержание:

1. Организационный этап (инструктаж по проведению практики)
2. Подготовительный этап (изучение организационной структуры объекта практики и особенностей деятельности выбранного банковского учреждения)
3. Исследовательский этап (сбор информации об объекте практики и анализ содержания источников информации по практике)
4. Проектный этап (экспериментально-практическая работа)
5. Аналитический этап (обработка и анализ полученной информации об объекте практики, предложения и рекомендации)

Отчетный этап (заполненные формы отчетности, документы, схемы, графики и прочее)

Структура отчета:

Введение

1. Характеристика базы практики, роль и место подразделения, в котором работал практикант в общей структуре организации, объем выполняемых подразделением работ и услуг в общем объеме операций и т.д.

2. Основная часть

Выполняется в соответствии с индивидуальным планом работы по соответствующему профессиональному модулю.

3. Заключение

Выводы и предложения. Необходимо разработать конкретные предложения по усовершенствованию организации работы базы практики в рамках соответствующего профессионального модуля, что, по сути, становится итогом пройденной практики. При этом сравниваются результаты теоретического обучения с наблюдениями и выводами по работе в конкретной организации.

4. Приложения

Документальное подтверждение отдельных разделов, положений отчета (заполненные формы отчетности, документы, схемы, графики и прочее).

5. Литература

Законодательная база, №№ инструкций, приказов, распоряжений, учебные пособия, учебники и другая литература.

Дата: _____

(Подпись, инициалы студента)

1. Анализ предметной области

Магазин продажи компьютерной техники должен быстро и качественно, как и любой другой магазин, обслуживать клиентов. База данных магазина будет использоваться как работниками, так и клиентами.

При работе с этой базой данных каждый клиент может получить информацию об интересующих его товарах.

Магазин компьютерной техники «ТехноТорг» начал свою работу в конце 2016 года в г. Москва. За это время он стал достаточно популярным за счет низких цен и хорошего качества продаваемой продукции.

Граждане России получили возможность заказывать следующие категории товаров:

- Ноутбуки
- Телефоны
- Планшеты
- Аксессуары
- Фотоаппараты
- Видеокамеры



• Рисунок 1. - Организационная структура магазина «ТехноТорг»

На рисунке 1 представлена организационная структура магазина «ТехноТорг», которая состоит из руководителя, бухгалтерии, менеджера, администратора, кассы, продавца и курьера.

Руководитель производит общее руководство производственно-хозяйственной и финансово-экономической деятельностью предприятия, организацию взаимодействия всех работников.

Для сотрудника магазина хранится следующая информация:

- ФИО
- Дата рождения
- Паспортные данные
- Телефон
- Адрес
- Дата приема на работу

При оформлении заказа клиентом, данные о нем также фиксируются и хранятся:

- ФИО
- Дата рождения
- Телефон
- Адрес
- Почтовый индекс

Данные о заказе:

- ФИО
- Телефон
- Товар
- Количество товара
- Дата заказа
- Адрес доставки
- Дата доставки
- Стоимость доставки
- Стоимость заказа

Данные о товарах:

- Наименование товара
- Категория
- Производитель
- Цена
- Срок гарантии

Данные в чеке:

- Код чека
- Товар
- Количество
- Сумма
- Дата покупки

2. Требования к разрабатываемой БД магазина «ТехноТорг»

В соответствии с ГОСТ 34.601-90 сформированы следующие требования:

С данной базой данных могут работать следующие группы пользователей:

- Администратор
- Менеджер
- Клиент

При работе с базой данных администратор может выполнять следующие задачи:

- вносить изменения в личные данные клиентов и работников
- добавлять или удалять информацию о товарах
- редактировать или добавлять информацию о заказах
- просматривать любую информацию

При работе с базой данных менеджер может выполнять следующие задачи:

- просматривать информацию по чекам
- добавлять информацию о чеках
- редактировать или добавлять информацию о заказах
- просматривать любую информацию

При работе с базой данных клиент может:

- просматривать информацию о заказах

Для данной базы данных требуется предусмотреть следующие

ограничения:

- работники не моложе 18 лет;
- у каждого сотрудника должны быть обязательно заполнены все данные;
- при заказе обязательно требуется заполнение полей ФИО и моб. Телефона;
- при заказе от 10 тыс. рублей доставка бесплатная.

3. Инфологическое проектирование базы данных

При проектировании на инфологическом уровне создается информационно-логическая модель, которая должна отвечать следующим требованиям:

- обеспечение наиболее естественных для человека способов сбора и предоставления той информации, которую предполагается хранить в создаваемой базе данных;
- корректность схемы БД (Адекватное отображение моделированной ПО);
- простота и удобство использования на следующих этапах проектирования, то есть информационно-логическая модель может легко отображаться на модели базы данных, которые поддерживаются известным СУБД (Сетевые, иерархические, реляционные и др.);
- информационно-логическая модель должна быть описана языком, понятным проектировщикам баз данных, программистам, администратору и будущим пользователям.

Суть инфологического моделирования состоит в выделении сущностей (Информационных объектов предметной области), которые подлежат хранению в базе данных, а также в определении характеристик объектов и взаимосвязей между ними.

Для базы данных «Магазин ТехноТорг» на основе проведенного

системного анализа предметной области выделены следующие сущности:

- продавец: сущность содержит информацию о продавцах, работающих в магазине;
- продажа товара: сущность содержит информацию о продаже товара;
- товар: сущность содержит информацию о товарах;
- категории: сущность содержит информацию о категории товара, продаваемого в магазине;
- производитель: сущность содержит информацию о производителях товаров;
- клиент: сущность содержит информацию о клиентах;
- заказ: сущность содержит информацию о заказе;
- доставка товара: сущность содержит информацию об осуществлении заказа;
- менеджеры: сущность содержит информацию о менеджерах, работающих в магазине.

Исходя из приведенных выше сущностей, построена инфологическая модель предметной области, которая представлена на рисунке 2.



Рисунок 2- Инфологическая модель предметной области

4. Логическое и даталогическое проектирование базы данных

Для логического проектирования выбрана реляционная модель данных,

т.к. она наиболее полно соответствует требованиям, предъявленным к разрабатываемой информационной системе:

- отсутствие дублируемой информации;
- поддержание целостности данных при вставке, удалении или изменении записей;
- возможность организации всех видов связи между отношениями 1:1, 1:М и М:М.

Целью даталогического проектирования является построение корректной схемы БД, ориентированную на реляционную модель. Корректной называется схема БД, в которой отсутствуют нежелательные зависимости между атрибутами отношений.

Процесс разработки корректной схемы РБД и является даталогическим проектированием.

При переходе от инфологической модели к реляционной модели была раскрыта связь М:М между отношениями «Продажа товара» и «Товар». Отношением-связкой стало отношение «Содержание чека». В нем в качестве FK были созданы атрибуты «Код товара» и «Код выбитого чека». Они вместе образуют уникальный идентифицирующий составной (композиционный) PK.

Исходя из приведённых выше отношений, построим схему получившейся БД (Рисунок 3):

Разрабатываемая база данных уже удовлетворяет требованиям третьей нормальной формы. Следовательно, процесс нормализации проводить не нужно.

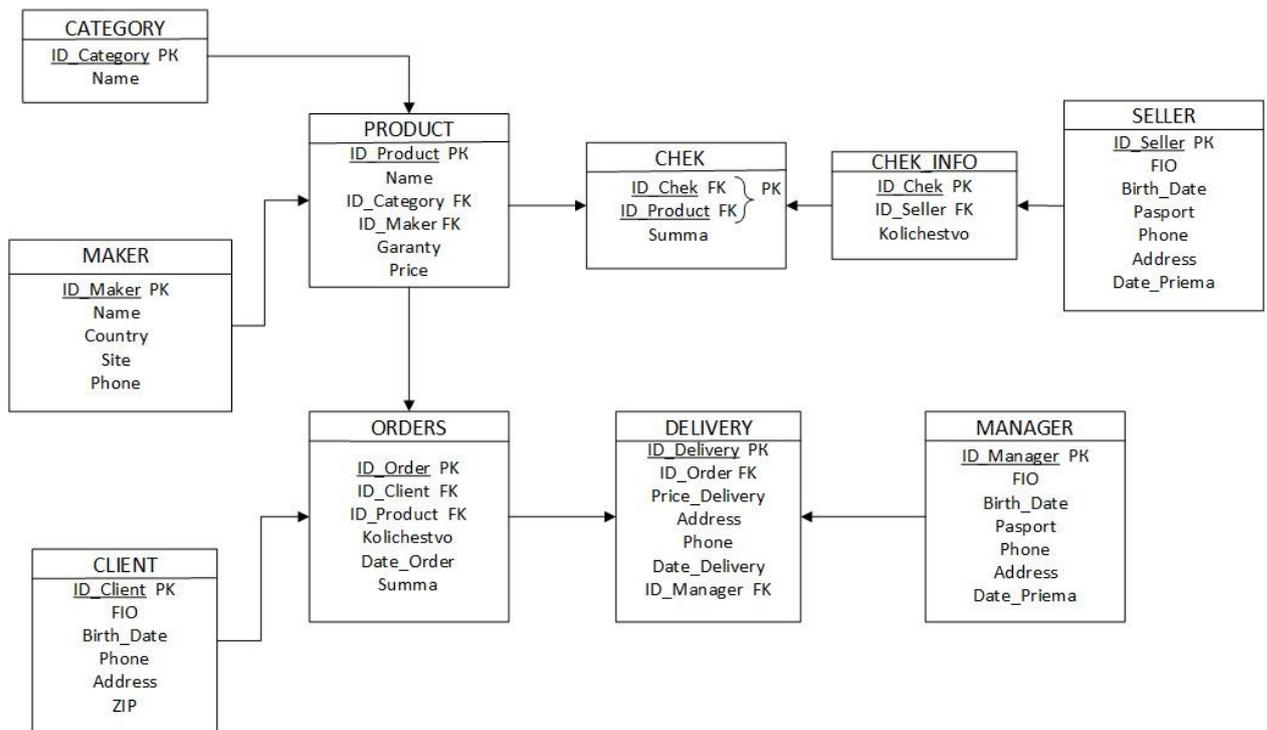


Рисунок 3 - Схема реляционной модели БД «Магазин ТехноТорг»

5. Физическое проектирование БД

На основе реляционной модели произведена программная реализация.

База данных содержит 10 таблиц:

- Продавец
- Клиент
- Заказ
- Доставка
- Менеджер
- Продажа товара
- Товар
- Категории
- Производитель
- Содержание чека

Таблица 1 - Продавцы (SELLER)

| ID_SELLER | FIO | BIRTH_DATE | PASPORT | PHONE | ADDRESS | DATE_PRIEMA |
|-----------|----------------|------------|------------|------------|------------------------------|-------------|
| 1 | Ivanov I.I. | 05/13/1991 | 6115348243 | 9051237816 | Moscow, Aviamotornaya, 34-8 | 02/05/2015 |
| 2 | Alekseeva N.D. | 04/21/1987 | 6211354218 | 9203211515 | Moscow, Volgogradskaya, 7-64 | 03/21/2016 |
| 3 | Smirnov N.I. | 09/21/1990 | 3208114586 | 9207421881 | Moscow, Avtozavodskaya, 13-5 | 01/17/2015 |
| 4 | Xoxlova I.V. | 02/01/1987 | 6512132476 | 9031283516 | Tula, Biryuzova, 13-17 | 05/30/2014 |
| 5 | Krasnova E.A. | 08/29/1992 | 3842132115 | 9037462835 | Moscow, Uznaya, 24-64 | 09/24/2016 |

Таблица 2 - Менеджер (MANAGER)

| ID_MANAGER | FIO | BIRTH_DATE | PASPORT | PHONE | ADDRESS | DATE_PRIEMA |
|------------|---------------|------------|------------|------------|--------------------------|-------------|
| 1 | Rublev A.I. | 01/13/1984 | 1235748212 | 9035722180 | Moscow, Svobodnaya, 30-2 | 12/13/2014 |
| 2 | Ivanova A.M. | 04/20/1987 | 6117841211 | 9213151694 | Moscow, Bolshaya, 18-24 | 01/17/2015 |
| 3 | Soboleva I.N. | 12/03/1991 | 3151821322 | 9037541118 | Moscow, Svobodj, 21-24 | 08/02/2015 |
| 4 | Vitko A.G. | 10/10/1990 | 6215312418 | 9203182111 | Moscow, Puskina, 31-21 | 09/15/2015 |

Таблица 3 - Клиенты (CLIENT)

| ID_CLIENT | FIO | BIRTH_DATE | PHONE | ADDRESS | ZIP |
|-----------|---------------|------------|------------|------------------------------|--------|
| 1 | Shirokov V.N. | 01/01/1980 | 9183212115 | Moscow, Vogogradskaya, 31-15 | 111352 |
| 2 | Malaxova O.Y. | 11/22/1990 | 9031823574 | Moscow, Komsomolskaya, 64-12 | 196182 |
| 3 | Nikolaev A.N. | 04/13/1989 | 9032183131 | Moscow, Zelenaya, 24-18 | 315112 |

Таблица 4 - Категории (CATEGORY)

| ID_CATEGORY | NAME |
|-------------|-------------|
| 1 | Telephone |
| 2 | Notebook |
| 3 | Planshet |
| 4 | Aksessuari |
| 5 | PhotoCamera |
| 6 | VideoCamera |

Таблица 5 - Производитель (MAKER)

| ID_MAKER | NAME | COUNTRY | SITE | PHONE |
|----------|--------|---------|----------------|--------------|
| 1 | ASUS | China | asusu.ru | 88001002727 |
| 2 | Huawei | China | shop.huawei.ru | 84953201221 |
| 3 | Lenovo | China | lenovo.com | 861058868888 |
| 4 | Nikon | Japan | nikon.ru | 84952216912 |
| 5 | Canon | Japan | canon.ru | 84952585601 |
| 6 | Xiaomi | China | ru-xiaomi.com | 84991109938 |
| 7 | Sony | Japan | sony-russia | 84951252446 |

Таблица 6 -Товар (PRODUCT)

| ID_PRODUCT | NAME | ID_CATEGORY | ID_MAKER | GARANTY | PRICE |
|------------|------------------|-------------|----------|---------|-------|
| 1 | ASUS GL52VX | 2 | 1 | 2 years | 57000 |
| 2 | Xiaomi Redmi 4 | 1 | 6 | 2 years | 7900 |
| 3 | Lenovo Vibe | 1 | 3 | 1 year | 5000 |
| 4 | Huawei Media Pad | 3 | 2 | 1 year | 7200 |

Таблица 7 - Продажа товара (CHEK_INFO)

| ID_CHEK | ID_SELLER | KOLICHESTVO |
|---------|-----------|-------------|
| 1 | 2 | 1 |
| 2 | 3 | 1 |

Таблица 8 -Содержание чека (CHEK)

| ID_CHEK | ID_PRODUCT | SUMMA |
|---------|------------|-------|
| 1 | 3 | 5700 |
| 2 | 4 | 7900 |

Таблица 9 - Доставка (DELIVERY)

| ID_DELIVERY | PRICE_DELIVERY | ADDRESS | PHONE | DATE_DELIVERY | ID_MANAGER |
|-------------|----------------|-------------------------------|------------|---------------|------------|
| 1 | 57000 | Moscow, Zelenaya, 25-18 | 9032183131 | 03/21/2016 | 3 |
| 2 | 7900 | Moscow, Komsomolskaya, 64-12 | 9031823574 | 04/02/2017 | 3 |
| 3 | 7900 | Moscow, Volgogradskaya, 31-15 | 9183212115 | 04/17/2017 | 1 |

Таблица 10 - Заказ (ORDERS)

| ID_ORDER | ID_PRODUCT | ID_CLIENT | KOLICHESTVO | DATE_ORDER | SUMMA | ID_DELIVERY |
|----------|------------|-----------|-------------|------------|-------|-------------|
| 1 | 1 | 1 | 3 | 03/10/2016 | 57000 | 1 |
| 2 | 2 | 2 | 1 | 03/31/2017 | 7900 | 2 |
| 3 | 2 | 1 | 1 | 04/11/2017 | 7900 | 3 |

6. Реализация ограничений. Безопасность и контроль.

С базой данных могут работать 3 типа пользователей: администратор, менеджер и клиент.

Администраторы организуют работу всей базы данных. Они имеют доступ к любой информации и могут изменять структуры таблиц и данные в них.

Менеджер имеет доступ к некоторым данным и может вводить необходимую для работы информацию.

Клиент может только просматривать необходимую ему информацию.

Для реализации ограничения «Бесплатная доставка при сумме заказа от 10 тысяч рублей» создадим завершающий триггер Orders_Trig.

```
CREATE OR REPLACE TRIGGER Orders_Trig
AFTER INSERT OR UPDATE ON ORDERS FOR EACH ROW
BEGIN
IF :new.SUMMA>10000 then
BEGIN
UPDATE DELIVERY SET Price_Delivery = 0 WHERE ID_Order=:new.ID_Order;
END;
END IF;
END;
```

Проверим его работу, изменив цену в 1 строке таблицы ORDERS. После изменения в таблице DELIVERY автоматически произошло изменение цены доставки на 0.

| ID_DELIVERY | PRICE_DELIVERY | ADDRESS | PHONE | DATE_DELIVERY | ID_MANAGER | ID_ORDER |
|-------------|----------------|-------------------------------|------------|---------------|------------|----------|
| 1 | 0 | Moscow, Zelenaya, 25-18 | 9032183131 | 03/21/2016 | 3 | 1 |
| 2 | 200 | Moscow, Komsomolskaya, 64-12 | 9031823574 | 04/02/2017 | 3 | 2 |
| 3 | 200 | Moscow, Volgogradskaya, 31-15 | 9183212115 | 04/17/2017 | 1 | 3 |

Второй триггер реализует подсчет «Сумма» чека в таблице «Продажа товара». Для этого нужно поле «Количество» из таблицы «Содержание чека» по «Коду товара» умножить на поле «Цена товара» из таблицы «Товар».

```
CREATE OR REPLACE TRIGGER Chek_T
AFTER INSERT OR UPDATE ON CHEK_INFO
FOR EACH ROW
DECLARE pCost INT;
BEGIN
SELECT PRICE INTO pCost
FROM PRODUCT
WHERE ID_Product = :new.ID_Product;
UPDATE CHEK
SET Summa = pCost * :new.Kolichestvo
WHERE ID_Chek = :new.ID_Chek;
```

END;

После изменения или добавления информации в таблицу Продажа товара (CHEL_INFO), в таблице Содержание чека будет меняться Сумма.

| ID_CHEK | ID_PRODUCT | SUMMA |
|---------|------------|-------|
| 1 | 3 | 11400 |
| 2 | 4 | 7900 |

Третий триггер: если страной производителя является Америка, то цена на его товар устанавливается на отметке 5000 р.

```
CREATE OR REPLACE TRIGGER Trig2  
AFTER UPDATE ON MAKER FOR EACH ROW  
BEGIN  
IF :new.Country='America' THEN  
BEGIN  
UPDATE PRODUCT SET Price = '5000'  
WHERE ID_Maker=:new.ID_Maker;  
END;  
END IF;  
END;
```

В Oracle имеется целый ряд механизмов для идентификации и верификации пользователей. Самый простой из них – обязательное указание пользователем своих имени и пароля при каждом подключении. Эта верификация должна выполняться независимо от того, какое внешнее интерфейсное средство используется для доступа к базе данных. Идея состоит в том, чтобы допустить пользователей к работе со средствами базы данных только после того, как он установит санкционированное соединение с ней. Имя пользователя и пароль сверяются с указанными в таблице SYS.USERS, куда пароль заносится в зашифрованной форме.

В большинстве приложений баз данных существуют разные категории пользователей, которые работают с разными частями системы и имеют разные права на просмотр и изменение данных. В простом случае может быть всего два класса пользователей: те, кто вводит данные, и менеджеры, выполняющие запросы к данным. Но в большинстве случаев существует несколько категорий пользователей, и функциональные возможности, к которым они должны иметь

доступ, пересекаются. В таких ситуациях можно избежать дублирования работы, создав одно приложение с меню или панелью инструментов, вид и содержимое которой зависят от задач конкретного пользователя.

Если подключаться к базе данных могут лишь уполномоченные пользователи, и они могут запускать только те модули, на выполнение которых им явно предоставлено право, то нужно подумать о следующем уровне безопасности – ограничении доступа этих пользователей к данным.

Для добавления пользователя в базу данных администратор базы данных создает учетную запись с именем пользователя и паролем. Каждому пользователю присваивается профиль — характеристика предельных объемов системных ресурсов, которые могут быть выделены данному пользователю. Сюда входит лимит совокупного процессорного времени, предоставляемого в течение одного сеанса или за один вызов Oracle, и другие подобные ограничения.

В Oracle имеются системные и объектные привилегии. Системные привилегии — это права на выполнение общих задач, таких как SELECT ANY TABLE и UPDATE ANY TABLE. Объектные привилегии относятся к действиям с определенными элементами базы данных — таблицами, представлениями и последовательностями. Для предоставления привилегий другому пользователю можно использовать оператор GRANT.

Выводы

В ходе работы было выполнено индивидуальное задание:

- выполнен анализ предприятия;
- выполнен анализ предметной области;
- обоснован выбранный и разработанный проект БД;
- построена инфологическая модель;
- построена реляционная модель;
- была создана БД, отвечающая основным требованиям, обозначенным

в разделе проектирования

- определена технология и выбрана среда программирования;
- разработан программный код БД;
- проведены ограничения безопасности БД по определенному сценарию;

Разработана реляционная база данных, содержащая элементы автоматизации и обработки данных. В ее составе: 10 таблиц, 3 триггера.

Список используемых источников

Основная литература:

1. Романенко М.Г. Системы электронного документооборота : учебное пособие (лабораторный практикум) / составители М. Г. Романенко. — Ставрополь : Северо-Кавказский федеральный университет, 2019. — 109 с. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/92745.html>
2. Сизова, О. В. Управление электронным предприятием : учебное пособие / О. В. Сизова, О. П. Смирнова. — Саратов : Ай Пи Ар Медиа, 2019. — 143 с. — ISBN 978-5-4497-0209-8. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/83659.html>
3. Степанова, Е. Н. Система электронного документооборота (облачное решение) : учебное пособие / Е. Н. Степанова. — Москва : Ай Пи Ар Медиа, 2021. — 182 с. — ISBN 978-5-4497-0767-3. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/101357.html>

Дополнительная литература:

4. Гладкий, А. А. Компьютер для индивидуального предпринимателя. Как вести учет быстро, легко и безошибочно : практическое пособие : [16+] / А. А. Гладкий. — Москва ; Берлин : Директ-Медиа, 2020. — 217 с. : ил. — Режим доступа: по подписке. — URL: <https://biblioclub.ru/index.php?page=book&id=577163>
5. Яковенко В.С. Учет в торговле : учебное пособие / В. С. Яковенко, Е. И. Костюкова, М. Н. Татаринова, О. В. Ельчанинова ; под редакцией В. С. Яковенко. — Ставрополь : Секвойя, 2019. — 81 с. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/109413.html>

Программный код базы данных

Создание таблицы Клиенты

```
create table CLIENT (  
  ID_Client int PRIMARY KEY,  
  FIO varchar(25) NOT NULL,  
  Birth_Date date NOT NULL,  
  Phone varchar(20) NOT NULL,  
  Address varchar(50) NOT NULL,  
  ZIP int NOT NULL);
```

Create Sequence ID_Cl Increment by 1 start with 1;

```
Insert into CLIENT (ID_Client, FIO, Birth_Date, Phone, Address, ZIP)  
  Values (ID_Cl.NextVal, 'Shirokov V.N.', TO_DATE('1980/01/01', 'YYYY/MM/DD'),  
  '9183212115', 'Moscow, Volgogradskaya, 31-15', '111352');  
Insert into CLIENT (ID_Client, FIO, Birth_Date, Phone, Address, ZIP)  
  Values (ID_Cl.NextVal, 'Malaxova O.U.', TO_DATE('1990/11/22', 'YYYY/MM/DD'),  
  '9031823574', 'Moscow, Komsomolskaya, 64-12', '196182');  
Insert into CLIENT (ID_Client, FIO, Birth_Date, Phone, Address, ZIP)  
  Values (ID_Cl.NextVal, 'Nikolaev A.N.', TO_DATE('1989/04/13', 'YYYY/MM/DD'),  
  '9032183131', 'Moscow, Zelenaya, 25-18', '315112');
```

Создание таблицы Продавцы

```
create table SELLER (  
  ID_Seller int PRIMARY KEY,  
  FIO varchar(25) NOT NULL,  
  Birth_Date date NOT NULL,  
  Pasport int NOT NULL,  
  Phone varchar(20) NOT NULL,  
  Address varchar(50) NOT NULL,  
  Date_Priema date NOT NULL);
```

Create Sequence ID_Sel Increment by 1 start with 1;

```
Insert into SELLER (ID_Seller, FIO, Birth_Date, Pasport, Phone, Address, Date_Priema)  
  Values (ID_Sel.NextVal, 'Ivanov I.I.',  
  TO_DATE('1991/05/13', 'YYYY/MM/DD'), '6115348243', '9051237816', 'Moscow,  
  Aviamotornaya, 34-8', TO_DATE('2015/02/05', 'YYYY/MM/DD'));  
Insert into SELLER (ID_Seller, FIO, Birth_Date, Pasport, Phone, Address, Date_Priema)  
  Values (ID_Sel.NextVal, 'Alekseeva N.D.',  
  TO_DATE('1987/04/21', 'YYYY/MM/DD'), '6211354218', '9203211515', 'Moscow,  
  Volgogradskaya, 7-64', TO_DATE('2016/03/21', 'YYYY/MM/DD'));  
Insert into SELLER (ID_Seller, FIO, Birth_Date, Pasport, Phone, Address, Date_Priema)  
  Values (ID_Sel.NextVal, 'Smirnov N.I.',  
  TO_DATE('1990/09/21', 'YYYY/MM/DD'), '3208114586', '9207421881', 'Moscow,  
  Avtozavodskaya, 16-5', TO_DATE('2015/01/17', 'YYYY/MM/DD'));  
Insert into SELLER (ID_Seller, FIO, Birth_Date, Pasport, Phone, Address, Date_Priema)  
  Values (ID_Sel.NextVal, 'Xoxlova I.V.',  
  TO_DATE('1987/02/01', 'YYYY/MM/DD'), '6512132476', '9031283516', 'Tula, Biruzova,  
  13-17', TO_DATE('2014/05/30', 'YYYY/MM/DD'));  
Insert into SELLER (ID_Seller, FIO, Birth_Date, Pasport, Phone, Address, Date_Priema)  
  Values (ID_Sel.NextVal, 'Krasnova E.A.',
```

```
TO_DATE('1992/08/29','YYYY/MM/DD'),'3842132115','9037422835','Moscow,
Uznaya, 24-64', TO_DATE('2016/09/24','YYYY/MM/DD')));
```

Создание таблицы Менеджеры

```
create table MANAGER (
  ID_Manager int PRIMARY KEY,
  FIO varchar(25) NOT NULL,
  Birth_Date date NOT NULL,
  Pasport int NOT NULL,
  Phone varchar(20) NOT NULL,
  Address varchar(50) NOT NULL,
  Date_Priema date NOT NULL);
```

Create Sequence ID_Man Increment by 1 start with 1;

```
Insert into MANAGER (ID_Manager, FIO, Birth_Date, Pasport, Phone, Address, Date_Priema)
  Values (ID_Man.NextVal,'Rublev A.I.',
  TO_DATE('1984/01/13','YYYY/MM/DD'),'1235748212','9035721801','Moscow,
  Svobodnaya, 30-2', TO_DATE('2014/12/13','YYYY/MM/DD')));
```

```
Insert into MANAGER (ID_Manager, FIO, Birth_Date, Pasport, Phone, Address, Date_Priema)
  Values (ID_Man.NextVal,'Ivanova A.M.',
  TO_DATE('1987/04/20','YYYY/MM/DD'),'6117841211','9213151694','Moscow,
  Bolshaya, 18-24', TO_DATE('2015/01/17','YYYY/MM/DD')));
```

```
Insert into MANAGER (ID_Manager, FIO, Birth_Date, Pasport, Phone, Address, Date_Priema)
  Values (ID_Man.NextVal,'Soboleva I.N.',
  TO_DATE('1991/12/03','YYYY/MM/DD'),'3151821322','9037541118','Moscow,
  Svobodi, 21-24', TO_DATE('2015/08/02','YYYY/MM/DD')));
```

```
Insert into MANAGER (ID_Manager, FIO, Birth_Date, Pasport, Phone, Address, Date_Priema)
  Values (ID_Man.NextVal,'Vitko A.G.',
  TO_DATE('1990/10/10','YYYY/MM/DD'),'6215312418','9203182111','Moscow,
  Pushkina, 31-21', TO_DATE('2016/09/15','YYYY/MM/DD')));
```

Создание таблицы Категории

```
create table CATEGORY (
  ID_Category int PRIMARY KEY,
  Name varchar(25) NOT NULL);
```

Create Sequence ID_Cat Increment by 1 start with 1;

```
Insert into CATEGORY (ID_Category, Name)
  Values (ID_Cat.NextVal, 'Telephone');
```

```
Insert into CATEGORY (ID_Category, Name)
  Values (ID_Cat.NextVal, 'Notebook');
```

```
Insert into CATEGORY (ID_Category, Name)
  Values (ID_Cat.NextVal, 'Planshet');
```

```
Insert into CATEGORY (ID_Category, Name)
  Values (ID_Cat.NextVal, 'Aksessuari');
```

```
Insert into CATEGORY (ID_Category, Name)
  Values (ID_Cat.NextVal, 'PhotoCamera');
```

```
Insert into CATEGORY (ID_Category, Name)
  Values (ID_Cat.NextVal, 'VideoCamera');
```

Создание таблицы Производитель
create table MAKER (
ID_Maker int PRIMARY KEY,
Name varchar(25) NOT NULL,
Country varchar(25) NOT NULL,
Site varchar(50) NOT NULL,
Phone int NOT NULL);

Create Sequence ID_Maker Increment by 1 start with 1;

Insert into MAKER (ID_Maker, Name, Country, Site, Phone)
Values (ID_Maker.NextVal, 'ASUS','China','asus.ru','88001002787');
Insert into MAKER (ID_Maker, Name, Country, Site, Phone)
Values (ID_Maker.NextVal, 'Huawei','China','shop.huawei.ru','84953201221');
Insert into MAKER (ID_Maker, Name, Country, Site, Phone)
Values (ID_Maker.NextVal, 'Lenovo','China','lenovo.com','861058868888');
Insert into MAKER (ID_Maker, Name, Country, Site, Phone)
Values (ID_Maker.NextVal, 'Nikon','Japan','nikon.ru','84952216912');
Insert into MAKER (ID_Maker, Name, Country, Site, Phone)
Values (ID_Maker.NextVal, 'Canon','Japan','canon.ru','84952585601');
Insert into MAKER (ID_Maker, Name, Country, Site, Phone)
Values (ID_Maker.NextVal, 'Xiaomi','China','ru-xiaomi.com','84991109938');
Insert into MAKER (ID_Maker, Name, Country, Site, Phone)
Values (ID_Maker.NextVal, 'PocketBook','Ukraine','pocketbook-int.com',
'88007009327');
Insert into MAKER (ID_Maker, Name, Country, Site, Phone)
Values (ID_Maker.NextVal, 'Sony','Japan','sony-russia.com','84951252446');

Создание таблицы Продажа товара
create table CHEK_INFO (
ID_Chek int PRIMARY KEY,
ID_Seller int NOT NULL,
Kolichestvo int NOT NULL);

Create Sequence ID_Chek Increment by 1 start with 1;

alter table CHEK_INFO

ADD CONSTRAINT SellerFK FOREIGN KEY (ID_Seller) REFERENCES SELLER;

Insert into CHEK_INFO (ID_Chek, ID_Seller, Kolichestvo)
Values (ID_Chek.NextVal, '2','5000');
Insert into CHEK_INFO (ID_Chek, ID_Seller, Kolichestvo)
Values (ID_Chek.NextVal, '3','7200');

Создание таблицы Товары
create table PRODUCT (
ID_Product int PRIMARY KEY,
Name varchar(30) NOT NULL,
ID_Category int NOT NULL,
ID_Maker int NOT NULL,
Garanty varchar(15) NOT NULL,
Price int NOT NULL);

```
Create Sequence ID_Pr Increment by 1 start with 1;
alter table PRODUCT
  ADD CONSTRAINT CategoryFK FOREIGN KEY (ID_Category) REFERENCES
  CATEGORY;
alter table PRODUCT
  ADD CONSTRAINT MakerFK FOREIGN KEY (ID_Maker) REFERENCES
  MAKER;
```

```
Insert into PRODUCT (ID_Product, Name, ID_Category, ID_Maker, Garanty, Price)
  Values (ID_Pr.NextVal, 'ASUS GL552VX','2','1','2 year','57000');
Insert into PRODUCT (ID_Product, Name, ID_Category, ID_Maker, Garanty, Price)
  Values (ID_Pr.NextVal, 'Xiaomi Redmi 4','1','6','2 year','7900');
Insert into PRODUCT (ID_Product, Name, ID_Category, ID_Maker, Garanty, Price)
  Values (ID_Pr.NextVal, 'Lenovo Vibe','1','3','1 year','5000');
Insert into PRODUCT (ID_Product, Name, ID_Category, ID_Maker, Garanty, Price)
  Values (ID_Pr.NextVal, 'Huawei Nedia Pad','3','2','1 year','7200');
```

Создание таблицы Информация о чеке

```
create table CHEK (
  ID_Chek int NOT NULL,
  ID_Product int NOT NULL,
  Summa int NOT NULL);

alter table CHEK ADD CONSTRAINT PK_CEK PRIMARY KEY (ID_Chek,
ID_Product);
alter table CHEK
  ADD CONSTRAINT ChekFK FOREIGN KEY (ID_Chek) REFERENCES
  CHEK_INFO;
alter table CHEK
  ADD CONSTRAINT MProduct_FK FOREIGN KEY (ID_PRODUCT)
  REFERENCES PRODUCT;
```

```
Insert into CHEK (ID_Chek, ID_Product, Summa)
  Values ('1', '3','1');
Insert into CHEK (ID_Chek, ID_Product, Summa)
  Values ('2', '4','1');
```

Создание таблицы Доставка

```
create table DELIVERY (
  ID_Delivery int PRIMARY KEY,
  Price_Delivery int NOT NULL,
  Address varchar(50) NOT NULL,
  Phone int NOT NULL,
  Date_Delivery date NOT NULL,
  ID_Manager int NOT NULL,
  ID_Order int NOT NULL);

Create Sequence ID_Del Increment by 1 start with 1;
alter table DELIVERY
  ADD CONSTRAINT ManagerFK FOREIGN KEY (ID_Manager) REFERENCES
  MANAGER;
```

```
Insert into DELIVERY (ID_Delivery, ID_Delivery, Address, Phone, Date_Delivery,
ID_Manager, ID_Order)
```

```
    Values (ID_Del.NextVal,'57000','Moscow, Zelenya, 25-18', '9032183131',
    TO_DATE('2016/03/20','YYYY/MM/DD'),'3');
```

```
Insert into DELIVERY (ID_Delivery, ID_Delivery, Address, Phone, Date_Delivery,
ID_Manager, ID_Order)
```

```
    Values (ID_Del.NextVal,'7900','Moscow, Komsomolskaya, 64-12', '9031823574',
    TO_DATE('2017/04/20','YYYY/MM/DD'),'3');
```

```
Insert into DELIVERY (ID_Delivery, ID_Delivery, Address, Phone, Date_Delivery,
ID_Manager, ID_Order)
```

```
    Values (ID_Del.NextVal,'7900','Moscow, Volgogradskaya, 31-5','9183212115',
    TO_DATE('2017/04/13','YYYY/MM/DD'),'1');
```

Создание таблицы Заказы

```
create table ORDERS (
    ID_Order int PRIMARY KEY,
    ID_Product int NOT NULL,
    ID_Client int NOT NULL,
    Kolichestvo int NOT NULL,
    Date_Order date NOT NULL,
    Summa int NOT NULL);
```

Create Sequence ID_Or Increment by 1 start with 1;

```
alter table ORDERS
```

```
    ADD CONSTRAINT ProductFK FOREIGN KEY (ID_Product) REFERENCES
    PRODUCT;
```

```
alter table ORDERS
```

```
    ADD CONSTRAINT ClientFK FOREIGN KEY (ID_Client) REFERENCES CLIENT;
```

```
alter table ORDERS
```

```
    ADD CONSTRAINT DeliveryFK FOREIGN KEY (ID_Delivery) REFERENCES
    DELIVERY;
```

```
Insert into ORDERS (ID_Order, ID_Product, ID_Client, Kolichestvo, Date_Order,
Summa, ID_Delivery)
```

```
    Values
```

```
(ID_Or.NextVal,'2','3','1',TO_DATE('2016/03/20','YYYY/MM/DD'),'57000','1');
```

```
Insert into ORDERS (ID_Order, ID_Product, ID_Client, Kolichestvo, Date_Order,
Summa, ID_Delivery)
```

```
    Values
```

```
(ID_Or.NextVal,'3','2','1',TO_DATE('2017/03/31','YYYY/MM/DD'),'7900','2');
```

```
Insert into ORDERS (ID_Order, ID_Product, ID_Client, Kolichestvo, Date_Order,
Summa, ID_Delivery)
```

```
    Values (ID_Or.NextVal,'3','1',
```

```
'1',TO_DATE('2017/04/11','YYYY/MM/DD'),'7900','3');
```