

ПРАКТИЧЕСКАЯ РАБОТА

«ТЕСТИРОВАНИЕ БАЗЫ ДАННЫХ»

Цель работы: приобретение практических навыков составления тестовых запросов к реляционной БД.

Для достижения поставленной цели требуется решить следующие **задачи**:

1. Разработать инфологическую модель БД для выбранной предметной области.
2. Разработать даталогическую модель БД в табличном виде.
3. Провести реализацию БД на сервере.
4. Разработать тестовые запросы к БД и проверить их выполнение.

Последовательность выполнения задания

1. Внимательно прочитать методические рекомендации по выполнению практической работы и выбрать один из трех вариантов предметных областей. Для каждой предметной области уже выделены сущности и их атрибуты.

Описание предметных областей по вариантам представлены в приложении к методическим указаниям. Так как задание выполняется в подгруппах количеством до 4 студентов включительно или, при желании, в индивидуальном порядке, то следует выбрать вариант предметной области. Варианты внутри одной учебной группы повторяться не должны. Фиксацию выбора проводит староста группы. Сообщать преподавателю, какой вариант выбран какой подгруппой, не требуется.

2. Разработать для выбранной предметной области инфологическую модель (ER-модель). При проектировании модели можно использовать любой сервис, текстовый или графический редактор, в котором предусмотрены графические элементы для любой нотации, например, Брахмана, Мартина или Чена. При установке связей должен быть определен тип связи: «один ко многим», «один к одному». Также должны быть выделены первичные ключи для каждой сущности.

3. Выполнить даталогическое проектирование в табличном виде (см. раздел 3). Определить тип и размер данных каждого атрибута.

4. Произвести физическую реализацию БД. Для этого можно воспользоваться облачным SQL-редактором (п.4.4) или, при желании более углубленного изучения, установить на персональный компьютер сервер MySQL (<https://dev.mysql.com/downloads/installer>).

5. Разработать программный код для реализации таблиц БД. Если планируется использовать СУБД на персональном компьютере, то необходимо сначала создать БД. Если задание выполняется облачным редактором, то БД создается автоматически, поэтому можно сразу переходить к реализации таблиц БД (см. листинги и рисунки раздела 3). Для обеспечения целостности БД также можно задать каскадное удаление и каскадное добавление данных.

6. Разработать программный код для заполнения таблиц правдоподобными данными. В каждой таблице должно быть не менее 7-10 записей. Сделать скриншоты, подтверждающие заполнение БД (см. раздел 3). Просмотр результата возможен, если каждый фрагмент программного кода выполняется в отдельной SQL-консоли.

Пример частичной реализации таблицы Путевки (Учетный номер, Код маршрута, Код клиента, Дата отправления, Длительность), с указанием атрибута в формате DATA, записью данных в таблицу и проверкой с помощью SELECT показан в следующем листинге.

```
CREATE TABLE Tours (  
    Tour_ID INT PRIMARY KEY,  
    Route_ID SMALLINT NOT NULL,  
    Client_ID INT NOT NULL,  
    Departure_Date DATE NOT NULL,  
    Duration SMALLINT NOT NULL  
);  
  
INSERT INTO Tours (Tour_ID, Route_ID, Client_ID, Departure_Date,  
Duration)  
VALUES  
    (1, 1, 101, '2023-07-01', 7),  
    (2, 2, 102, '2023-07-10', 5),  
    (3, 3, 103, '2023-07-15', 4),  
    (4, 4, 104, '2023-07-20', 10),
```

```
(5, 5, 105, '2023-07-25', 14),  
(6, 6, 106, '2023-08-01', 7);  
SELECT * FROM Tours;
```

7. Сформулировать и записать в текстовом формате 5 возможных запросов к БД. Примеры запросов показаны в п.4.4.

8. Реализовать программный код запросов. Каждый запрос реализуйте в отдельной SQL-консоли.

В программной реализации запросов должны присутствовать:

- встроенные агрегатными функции, например `SUM()`, `Count()`, `Min()`, `Max()` и др.;
- операторы фильтрации данных, в частности, `LIKE`, `BETWEEN`, `DISTINCT`;
- атрибуты инструкции `SELECT`: `FROM`, `WHERE`, `GROUP BY`, `HAVING`, `ORDER BY`.

Если первичных данных для выполнения запросов не хватает, то дополните их. Программную реализацию запросов сохраните в виде листингов. Сделайте скриншоты выполнения. Сравните полученные результаты выборки с ожидаемыми. Ожидаемый результат – это тот результат, который возможно проверить визуально (см. раздел 3) на том количестве записей, которые находятся в БД. Важно понимать, что запросы созданы правильно и выводят именно те данные, которые требуются.

9. Сделать вывод в формате эссе о результатах выполнения заданий.

10. Выполнить отчет в соответствии с требованиями к структуре и наполнению.

Структура отчет по практической работе

1. Цель работы работы.
2. Выбранная предметная область и ее описание.
3. Скриншот разработанной ER-модели БД.
4. Даталогическая модель в табличном виде, где для каждой сущности определены атрибуты, формат и размер данных.
5. Описание физической среды реализации БД: онлайн-сервис или локальный сервер.
6. Программные коды реализации и заполнения таблиц БД с фактическим подтверждением выполненных заданий на 2-3 скриншотах. Каждый листинг следует оформить шрифтом Courier New 12, межстрочный интервал 1.0.

7. Список пяти разработанных запросов к БД в текстовом виде.
8. Листинги программной реализации каждого из запросов и скриншоты выполнения каждого из них. Код каждого запроса должен сопровождаться текстовой формулировкой запроса в виде комментария к программному коду (см. листинги п.4.4). Так будет проще проверить правильность программной реализации каждого запроса.
9. Вывод в формате эссе, в котором раскрывается цель выполненной работы, описываются знания и навыки, полученные в процессе ее выполнения, а также возникшие проблемы и пути их решения.
10. Список использованных источников.

Критерии оценивания практической работы

Оценивание практической работы выполняется по следующим критериям (табл.2.1). За выполнение работы выставляются баллы, которые потом переводятся в оценку в случае дифференцированного зачета или зачет/не зачет в случае зачета без оценки (табл.2.2).

Таблица 2.1

Критерии оценивания практической работы

№	Критерий	Рейтинговый балл
1	Приведена цель работы	Да - 0,5 балла Нет – 0 баллов
2	Приведено описание выбранной предметной области	Да – 0,5 балла Нет – 0 баллов
3	Разработана ER-модель и приведен ее скриншот из среды проектирования. Присутствуют все атрибуты, связи установлены, выделен атрибут, который будет первичным ключом	Да – 2 балл Нет – 0 баллов
4	Выбрана среда реализации БД	Да – 0,5 балла Нет – 0 баллов
5	Приведены программные коды для физической реализации таблиц БД и их наполнению. Сделаны скриншоты	Да – 2 балла Нет – 0 баллов
6	Выполнены 5 запросов к БД. Приведена их программная реализация. Сделаны скриншоты на каждый запрос	Да – 2 балла Нет – 0 баллов
7	В отчете присутствуют комментарии по ходу выполнения заданий	Да – 1 балл Нет – 0 баллов
8	Вывод написан в формате эссе, в котором сделаны объемные выводы по результатам работы	Да – 1 балл Нет – 0 баллов.
9	Список использованных источников	Да – 0,5 балла Нет – 0 баллов
Максимально возможный балл		10

Таблица 2.2

Перевод баллов в оценку

Баллы	Оценка	Зачет/не зачет
Менее 4 баллов	неудовлетворительно	Не зачет
4-6 баллов	удовлетворительно	Зачет
6-8 баллов	хорошо	
8-10 баллов	отлично	

3. Пример практической реализации БД в онлайн SQL-редакторе

Кейс. Рассмотрим трехтабличную базу данных, в которой сохраняют данные о клиентах, товарах и продажах магазина цифровой техники. Требуется осуществить физическую реализацию БД и протестировать правильность выполнения запросов.

Тестировщику необходимо провести тестовые мероприятия методом «белого ящика», поэтому он должен обладать знаниями построения SQL-запросов. Рассмотрим последовательность проектирования и физической реализации БД на сервере. Чтобы не устанавливать MySQL-сервер на персональный компьютер, воспользуемся онлайн-редактором SQL, <https://sqliteonline.com/> или <https://www.programiz.com/sql/online-compiler/>. Такие SQL-редакторы предоставляют доступ к развернутому в облаке SQL-серверу.

В SQL-редакторе <https://sqliteonline.com/> возможен доступ к нескольким реляционным БД, в том числе к MariaDB¹ – СУБД с открытым исходным кодом, которую можно использовать в качестве замены MySQL. Она разрабатывается сообществом под лицензией GNU GPL и не содержит закрытых модулей.

На рис. 3.1 показано окно SQL-редактора, в котором требуется выбрать и создать соединение (connect) с сервером MariaDB. При необходимости демонстрационную БД с сервера можно удалить. Каждый запрос можно сформировать в новой SQL-консоли, добавление которой будет происходить при нажатии на символ «+». Для запуска запросов на выполнение используется Run.

¹ <https://mariadb.com/kb/ru/a-mariadb-primer/>

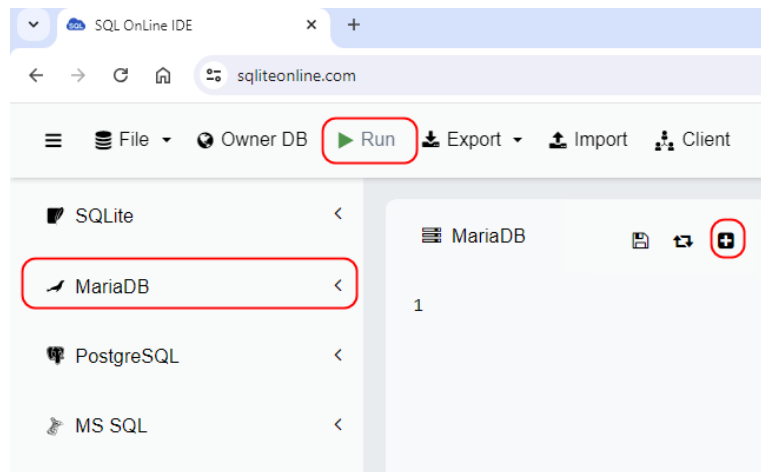


Рис.3.1. Окно SQL-редактора

1. На основе анализа предметной области выделено три сущности: клиенты (users), товары (product), продажи (invoice). В среде Draw.io разработана инфологическая модель (рис.3.2).

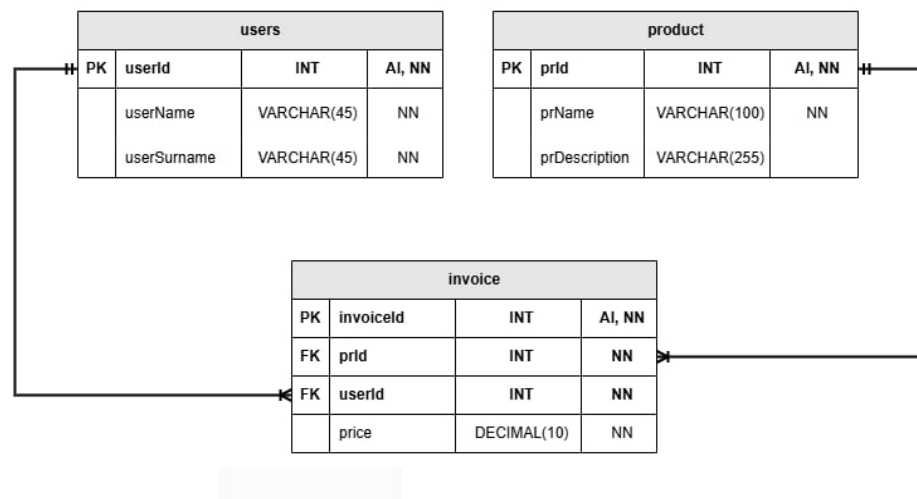


Рис.3.2. Инфологическая модель для предметной области «Магазин»

2. Для каждой сущности выявлены атрибуты, выделены первичные ключи, определены типы связей (табл.3.1-3.3). В таблицах приняты следующие сокращения: NN (Not Null) – поле с обязательным заполнением, AI (Avto Increment) – поле типа счетчик, ПК – первичный ключ, ВК – внешний ключ.

Таблица 3.1

Определение атрибутов сущности **users**

Атрибуты	Тип данных	Размер (бит)	Комментарии
userId	INT	32	ПК, NN, AI
userName	VARCHAR	45	Имя клиента, NN
userSurname	VARCHAR	45	Фамилия клиента, NN

Таблица 3.2

Определение атрибутов сущности **product**

Атрибуты	Тип данных	Размер (бит)	Комментарии
prId	INT	32	ПК, NN, AI
prName	VARCHAR	100	название товара, NN
prDescription	VARCHAR	255	Описание товара

Таблица 3.3

Определение атрибутов сущности **invoice**

Атрибуты	Тип данных	Размер (бит)	Комментарии
invoiceId	INT	32	ПК, NN, AI
prId	INT	32	БК, NN
userId	INT	32	БК, NN
price	DECIMAL	10, 2	стоимость товара, NN

3. Для реализации БД на SQL-сервере в консоли следует записать и затем выполнить программный код по созданию таблиц БД (см. листинг Реализация таблиц и структуры БД). Для соединения таблиц между собой по внешним ключам в явном виде прописаны поля, которые участвуют в соединении. Дополнительной команды на создание БД в данном онлайн-редакторе не требуется.

Листинг. Реализация таблиц и структуры БД

```
CREATE TABLE
users (
    userId INT AUTO_INCREMENT PRIMARY KEY,
    userName VARCHAR(45) Not Null,
    userSurname VARCHAR(45) Not Null
);
```

```

CREATE TABLE product(
    prId INT AUTO_INCREMENT PRIMARY KEY,
    prName VARCHAR(100) Not Null,
    prDescription VARCHAR(255) Not Null
);

CREATE TABLE invoice(
    invoiceId INT AUTO_INCREMENT PRIMARY KEY,
    prId INT Not Null,
    userId INT Not Null,
    price DECIMAL(10) Not Null,
    FOREIGN KEY (userId) REFERENCES users(userId) ON DELETE
CASCADE,
    FOREIGN KEY (prId) REFERENCES product(prId) ON DELETE
CASCADE
);

```

В результате будут реализованы таблицы и структура БД (рис.3.3).

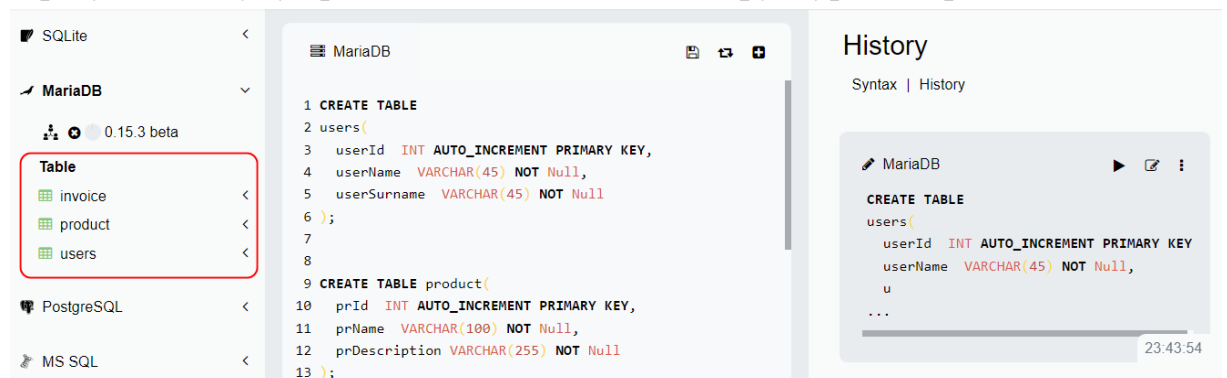


Рис.4.42. Создание структуры БД

4. Заполнение БД записями можно провести на основе кода, представленного ниже. Программный код следует записать в новой SQL-консоли, вызвав ее нажатием на «+». Для выполнения кода нажимаем на «Run». Результат показан на рис.3.4.

Листинг. Добавление записей в таблицы:

```

INSERT INTO users (userId, userName, userSurname)
VALUES
    (1, 'Петр', 'Антошин'),
    (2, 'Сергей', 'Пастухов'),
    (3, 'Анна', 'Рокотова');
SELECT * FROM users;

INSERT INTO product (prId, prName, prDescription)
VALUES

```

```

(1, 'Смартфон', 'средство связи'),
(2, 'Ноутбук', 'рабочая станция'),
(3, 'Телевизор', 'каналы о природе'),
(4, 'Наушники', 'слушаем подкасты'),
(5, 'Кондиционер', ''),
(6, 'Кофемашина', 'для души');
SELECT * FROM product;

INSERT INTO invoice (invoiceId, prId, userId, price)
VALUES
(1, 1, 1, 100),
(2, 2, 2, 200),
(3, 1, 2, 100),
(4, 3, 1, 300),
(5, 4, 1, 50),
(6, 5, 3, 350),
(7, 1, 3, 100);
SELECT * FROM invoice;

```

The screenshot shows a database management interface with two main panels. The left panel displays SQL code for inserting data into a 'users' table and selecting all records. The right panel shows the 'History' of executed queries. Below the SQL editor, a table view displays the inserted data.

id	userId	userName	userSurname
1	1	Петр	Антошин
2	2	Сергей	Пастухов
3	3	Анна	Рокотова

Рис.3.4. Заполнение таблиц данными

5. Для тестирования сформируем несколько запросов к БД и проверим правильность их выполнения. Визуализация результата возможна, если каждый запрос выполняется в отдельной SQL-консоли.

Обратим внимание, что в листинге с примерами запросов к БД использованы сокращенные названия таблиц, в частности, `u.username` вместо атрибута `userName` таблицы `users`.

Листинг. Запросы к БД:

```

-- Подсчитать общее количество поданных товаров
SELECT count(prId) AS 'Количество проданных товаров'

```

```
FROM invoice;

-- Найти все товары на букву «К»
SELECT
    prId,
    prName
FROM product
WHERE prName like 'K%';

-- Найти, сколько товаров купил клиент с ID=1
SELECT
    userId,
    count(*) AS 'Количество товаров'
FROM invoice
WHERE userId = 1;

-- Найти всех клиентов, которые не покупали товары на букву
«К»
SELECT
    DISTINCT u.userId,
    u.username
FROM    invoice AS i
        INNER JOIN users AS u
            ON i.userId = u.userId
        INNER JOIN product AS p
            ON i.prId = p.prId
WHERE p.prName NOT like 'K%';

-- Подсчитать количество продаж по каждому товару
SELECT
    p.prId,
    p.prName AS 'Товар',
    count(p.prId) AS 'Количество продаж'
FROM invoice AS i
        INNER JOIN product AS p
            ON i.prId = p.prId
GROUP BY p.prId;

-- Выбрать максимальное количество товаров, купленных одним
клиентом

SELECT
    MAX(tmp.Количество) as 'Максимальное количество'
FROM (
```

```
SELECT
    userId,
    count(prId) as 'Количество'
FROM invoice
GROUP BY userId
) as tmp;
```

```
-- Найти всех клиентов, которые купили смартфон
```

```
SELECT u.userId,
       u.username,
       p.prId,
       p.prname
FROM   invoice AS i
       INNER JOIN users AS u
           ON i.userId = u.userId
       INNER JOIN product AS p
           ON i.prId = p.prId
WHERE  p.prname = 'Смартфон' ;
```

```
-- Вывести клиента с максимальным количеством товаров,
начинающихся не на букву «К»
```

```
SELECT
    u.userId,
    u.username,
    count(i.prId) AS 'Количество'
FROM   invoice AS i
       INNER JOIN users AS u
           ON i.userId = u.userId
       INNER JOIN product AS p
           ON i.prId = p.prId
WHERE  p.prName NOT like 'K%'
GROUP BY u.userId, u.username
ORDER BY count(i.prId) DESC
LIMIT 1;
```

```
-- Вывести товары, выручка по которым больше 200
```

```
SELECT
    p.prId,
    p.prName AS 'Товар',
    sum(i.price) AS 'Выручка'
FROM invoice AS i
```

```

INNER JOIN product AS p
      ON i.prId = p.prId
GROUP BY p.prId
HAVING Выручка > 200;

```

Пример результата выполнения запроса «Вывести товары, выручка по которым больше 200» показан на рис. 3.5.

prId	Товар	Выручка
1	Смартфон	300
3	Телевизор	300
5	Кондиционер	350

Рис.3.5. Результат выполнения запроса

Для предварительной подготовки программного кода запросов используют текстовые редакторы, например Notepad++. Если необходимо оформить SQL-код в соответствии со стандартами записи, применяют программные сервисы, в частности <https://codebeautify.org/sqlformatter> (рис. 3.6).

Рис.3.6. Результат преобразования записи SQL-кода

ПРИЛОЖЕНИЕ

ВАРИАНТЫ ДЛЯ ПРАКТИЧЕСКОЙ РАБОТЫ

Вариант 1.

1. Книги (Код книги, Название, Автор, Жанр).
2. Читатели (Код читателя, Фамилия, Имя, Телефон).
3. Выданные книги (Учетный номер выдачи, Код книги, Код читателя, Дата выдачи, Дата возврата, Цена проката).

Вариант 2.

1. Маршруты (Код маршрута, Название, Дальность, Оплата).
2. Водители (Код водителя, Фамилия, Имя, Пол, Стаж).
3. Выполнения рейсы (Код рейса, Код маршрута, Код водителя, Дата отправки, Количество дней в пути).

Вариант 3.

1. Сотрудники (Код сотрудника, Фамилия, Имя, Оклад).
2. Виды работ (Код вида, Описание, Оплата за день).
3. Выполненные работы (Код работы, Код сотрудника, Код вида работы, Количество дней, Описание работы).

Вариант 4.

1. Авиарейсы (Код рейса, Название рейса, Авиакомпания)
2. Билеты на авиарейсы (Код билета, Код рейса, Имя пассажира, Место).
3. Журнал прилета (Учетный номер, Код рейса, Код билета, Тип воздушного судна, Стоимость билета).

Вариант 5.

1. Автомобили (Код автомобиля, Марка, Стоимость автомобиля).
 2. Клиенты (Код клиента, Фамилия, Имя, Телефон).
- Выданные автомобили (Учетный номер, Код автомобиля, Код клиента, Дата выдачи, Стоимость проката в день).

Вариант 6.

1. Пациенты (Код, Фамилия, Имя, Пол, Год рождения).

2. Врачи (Код врача, Фамилия, Специализация, Категория).

3. Обращения к врачу (Учетный номер, Код пациента, Код врача, Дата обращения, Оплата за прием)

Вариант 7.

1. Ткани (Код ткани, Тип ткани, Цвет).

2. Изделия (Код изделия, Вид изделия, Размер).

3. Заказы (Код заказа, Код изделия, Код ткани, Цена изделия, Дата заказа).

Вариант 8.

1. Маршруты путешествий (Код маршрута, Страна, Климат, Стоимость).

2. Клиенты (Код клиента, Фамилия, Имя, Телефон).

3. Путевки (Учетный номер, Код маршрута, Код клиента, Дата отправления, Длительность)

Вариант 9.

1. Клиенты (Код клиента, Фамилия, Имя, Телефон).

2. Номера в отеле (Код номера, Тип номера, Цена).

3. Поселение (Код поселения, Код клиента, Код номера, Дата поселения, Количество дней)

Вариант 10.

1. Товары (Код товара, Название, Цена).

2. Клиенты (Код клиента, Фамилия, Имя, Телефон).

3. Продажи (Код продажи, Код товара, Код клиента, Дата продажи, Количество).

Вариант 11.

1. Виды услуг химчистки (Код вида услуг, Тип белья, Стоимость).

2. Клиенты (Код клиента, Фамилия, Имя, Телефон).

3. Заказы (Код заказа, Код вида услуги, Код клиента, Дата приема, Цена услуги).

Вариант 12.

1. Изделия (Код изделия, Тип (серьги, кольца, броши, браслеты), Цена).
2. Материалы (Код материала, Название (платина, золото, серебро и др.), Цена за грамм).
3. Продажи (Код продажи, Код изделия, Код материала, Дата продажи, Стоимость покупки).

Вариант 13.

1. Клиенты (Код клиента, Фамилия, Имя, Телефон).
2. Стрижки (Код стрижки, Название, Цена).
3. Парикмахерские услуги (Код услуги, Код клиента, Код стрижки, Дата, Цена).

Вариант 14.

1. Фирмы (Код фирмы, Название фирмы, Вид деятельности);
2. Услуги юридические (Код услуги, Название (договоры купли-продажи долей, договоры купли-продажи квартир и т.д.), Стоимость).
3. Сделки нотариальной конторой (Код сделки, Код клиента, Код услуги, Дата, Сумма к оплате).

Вариант 15.

1. Филиал (Код филиала, Наименование, Адрес, Телефон).
2. Вид страхования (Код вида страхования, Наименование (страхование автотранспорта, страхование домашнего имущества, добровольное медицинское страхование), Страховая сумма).
3. Договора (Учетный номер, Код филиала, Код вида страхования, Дата заключения, Скидка).

Вариант 16.

1. Вид изделия (Код изделия, Вид изделия (шкаф, стол, сервант), материал).
2. Клиенты (Код клиента, Фамилия, Имя, Телефон).

3. Услуги по реставрации мебели (Код услуги, Код клиента, Код изделия Дефект, Дата обращения клиента в мастерскую, Цена за работу).

17 вариант.

1. Цех производства (Код цеха, Название, Директор, Адрес производства).

2. Полуфабрикат (Код полуфабриката, Название продукта, Цена)

3. Поставка полуфабрикатов (Код поставки, Код полуфабриката, Код цеха, Название кафе, Дата поставки в кафе, Стоимость доставки).

18 вариант.

1. Сведения о водителе (Код водителя, Фамилия, Имя, Номер водительского удостоверения).

2. Автомашина (Код машины, Марка, Номер машины, Год выпуска).

3. Дорожные происшествия (Код события, Код машины, Код водителя, Дата происшествия, Сумма ущерба).