**АРХАНГЕЛЬСКИЙ КОЛЛЕДЖ ТЕЛЕКОММУНИКАЦИЙ (ФИЛИАЛ) СПбГУТ**

**(АКТ (ф) СПбГУТ)**

**Отчеты по практическим работам**

**по ОПБД**

Студент: Матигоров Н.И.

Группа: ИСПП-31

Преподаватель: Маломан Ю.С.

Архангельск 2025

**Практическая работа №1**

**Изучение правил преобразования ER-модели в реляционную модель данных**

**1 Цель работы**

1.1 Изучить правила преобразования ER-модели в реляционную модель данных с учетом вида связи;

1.2 Изучить процесс создания концептуальной модели данных и ее преобразования в логическую модель данных;

1.3 Научиться применять графические редакторы в процессе проектирования БД.

**2 Контрольные вопросы**

2.1 Что такое «сущность»?

Объект, информацию о котором нужно хранить в БД.

2.2 Что такое «атрибут»?

Это характеристика или свойство сущности или связи.

2.3 Что такое «ключевое поле»?

Ключевое поле в базе данных (первичный ключ) — это поле (или набор полей), значения которого однозначно определяют каждую запись в таблице.

2.4 Каково назначение первичных и внешних ключей?

Первичный ключ — это один или несколько столбцов таблицы, значения которых однозначно идентифицируют каждую запись (строку) в этой таблице.

Внешний ключ — это столбец или группа столбцов в одной таблице, которые указывают на первичный ключ другой (или той же самой) таблицы.

2.5 Что такое «связь»?

Связь, отображает взаимодействие между двумя сущностями.

2.6 Какие виды связей между сущностями существуют?

Один к одному; Один ко многим; Много ко многим.

2.7 Какие элементы входят в ER-диаграммы?

Сущности, связи, атрибуты.

**3 Вывод**

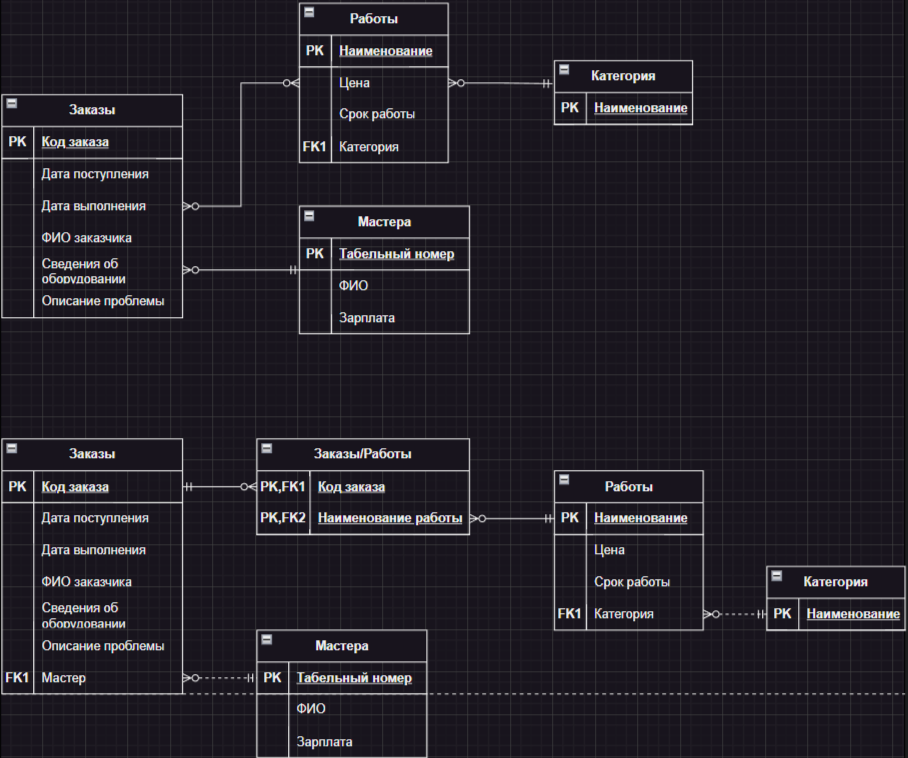
3.1 Я изучил правила преобразования ER-модели в реляционную модель данных с учетом вида связи;

3.2 Был изучен процесс создания концептуальной модели данных и ее преобразования в логическую модель данных;

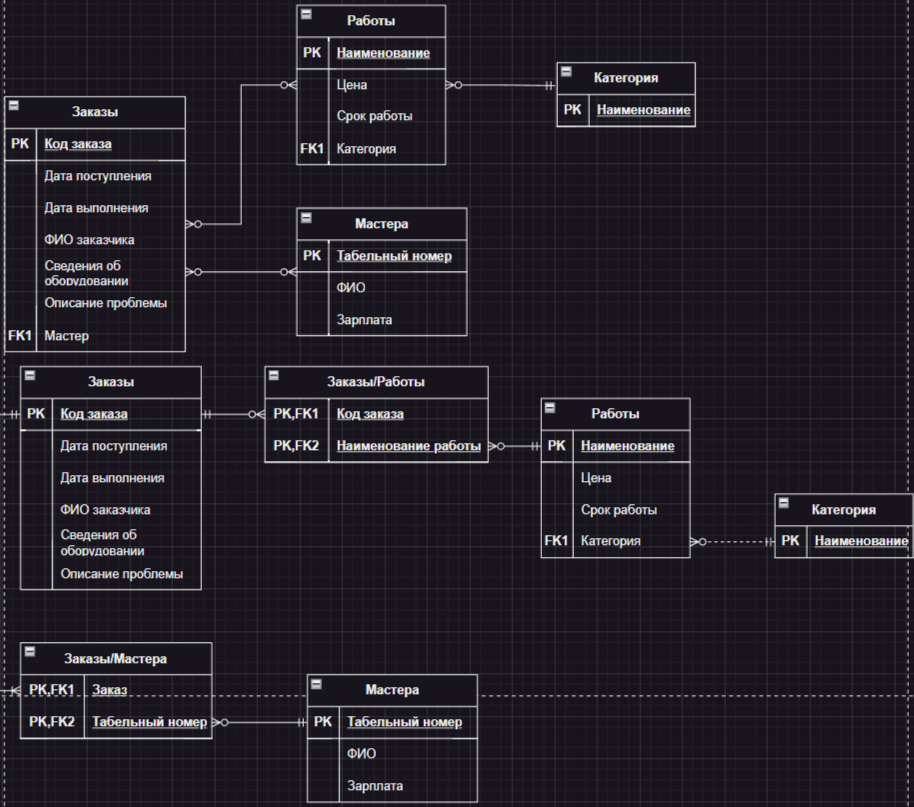
3.3 Я научился применять графические редакторы в процессе проектирования БД.

**4 Ход работы**

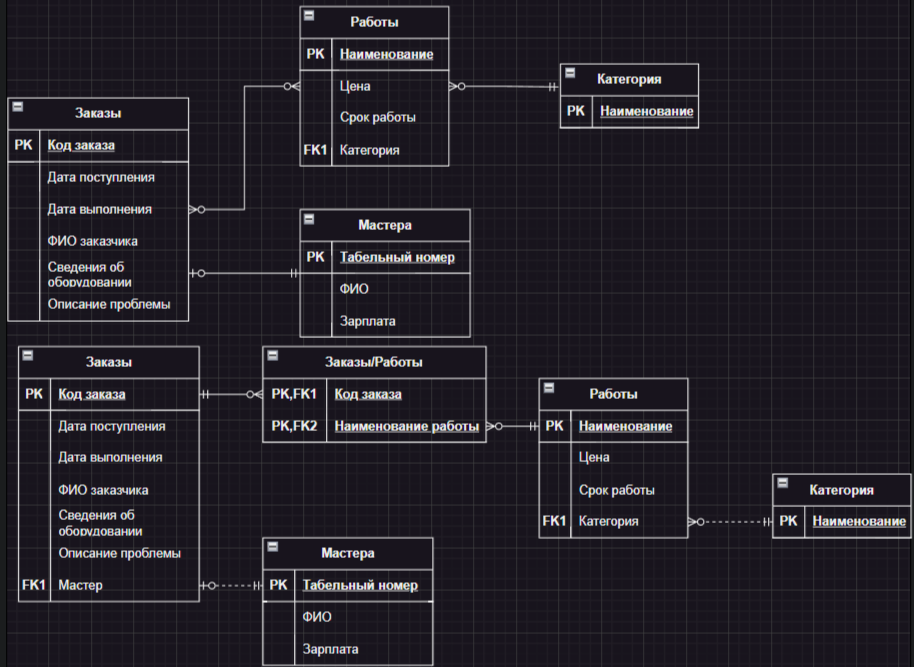
4.2 На странице 2 внести изменения в ER-модели с учетом того, что каждый заказ выполняется одним мастером.

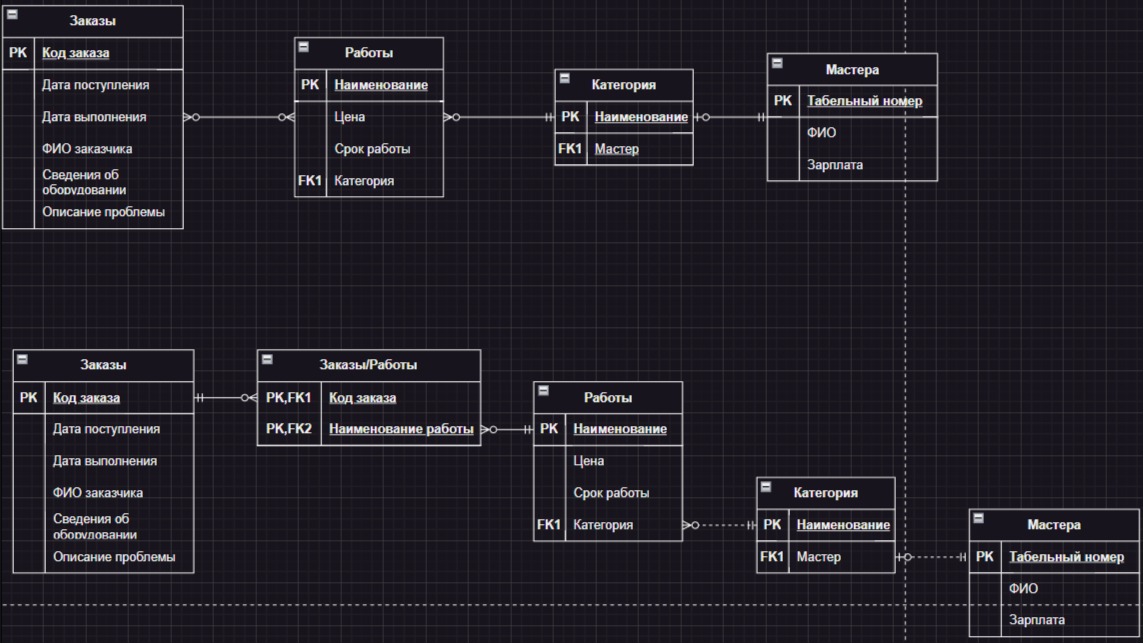


4.3 На странице 3 внести изменения в ER-модели с учетом того, что над выполнением одного заказа могут работать несколько мастеров. Преобразовать ее в реляционную модель.



4.4 На странице 4 внести изменения в ER-модели с учетом того, что каждый мастер выполняет определенную работу и только ее. Преобразовать ее в реляционную модель.



4.5 На странице 5 внести изменения в ER-модели с учетом того, что для выполнения каждой категории работы нанимается единственный мастер и только он выполняет работы этой категории. Информацию о предыдущем мастере, выполнявшем эту категорию работ, хранить не требуется.

**Практическая работа №2**

**Изучение процесса нормализации отношений**

**1 Цель работы**

1.1 Изучить процесс приведения отношений от ненормализованного вида к четвертой нормальной форме;

1.2 Изучить процесс декомпозии отношений.

**2 Контрольные вопросы**

2.1 Что называется первичным ключом отношения?

Первичным ключом отношения в реляционной модели данных называется набор атрибутов, значения которых однозначно определяют кортеж отношения.

2.2 Что называется внешним ключом отношения?

Внешний ключ отношения — это поле (или набор полей) в одной таблице, которое ссылается на первичный ключ в другой таблице.

2.3 В чем заключается процесс нормализации отношений?

Суть процесса заключается в устранении нежелательных функциональных зависимостей.

2.4 В каком случае атрибут А функционально зависит от атрибута В?

Атрибут А функционально зависит от атрибута В, если в каждый момент времени каждому значению А соответствует лишь одно значение В.

2.5 В каком случае атрибут А транзитивно зависит от атрибута В?

Атрибут С транзитивно зависит от атрибута А, если атрибут С зависит от атрибута В, а атрибут В зависит от атрибута А (при условии, что атрибут А функционально не зависит ни от атрибута В, ни от атрибута С).

2.6 Каким требованиям должно отвечать отношение, находящееся в 1НФ?

Отношение, находящееся в первой нормальной форме (1НФ), должно отвечать следующим требованиям:

* Все атрибуты в таблице должны быть простыми.
* Данные на пересечении строк и столбцов должны иметь исключительно скалярные значения.
* Не должно быть дублирующих строк.

2.7 Каким требованиям должно отвечать отношение, находящееся во 2НФ?

Отношение, находящееся во второй нормальной форме (2НФ), должно отвечать следующим требованиям:

* Быть в первой нормальной форме.
* **Если потенциальный ключ простой**, то есть состоит из единственного атрибута, то любая функциональная зависимость от него является неприводимой (полной).
* **Если потенциальный ключ составной**, то в отношении не должно быть неключевых атрибутов, зависящих от части составного потенциального ключа.
* **Запрещено наличие неключевых атрибутов**, которые вообще не зависят от потенциального ключа.

2.8 Каким требованиям должно отвечать отношение, находящееся в 3НФ?

Отношение, находящееся в 3NF, должно отвечать следующим требованиям:

* **Быть во второй нормальной форме**
* **Не иметь транзитивных зависимостей между неключевыми атрибутами**.
* **Все атрибуты непервичного ключа должны напрямую зависеть от первичного ключа**, а не косвенно через другие атрибуты непервичного ключа.

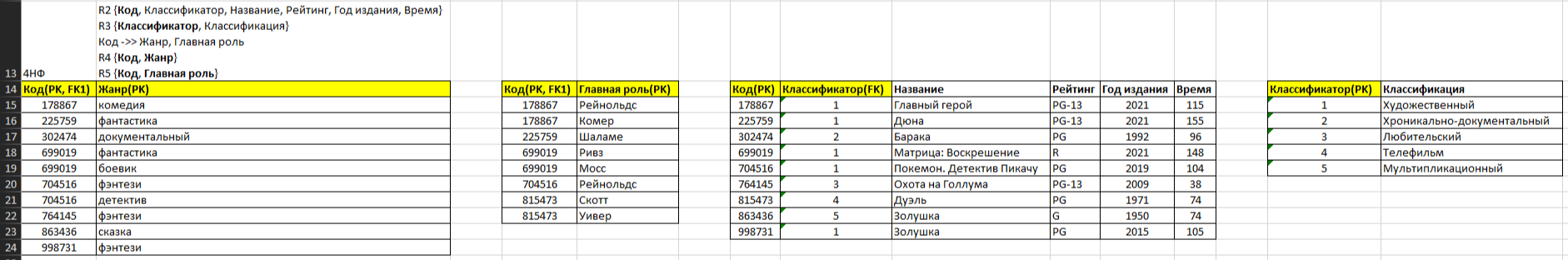
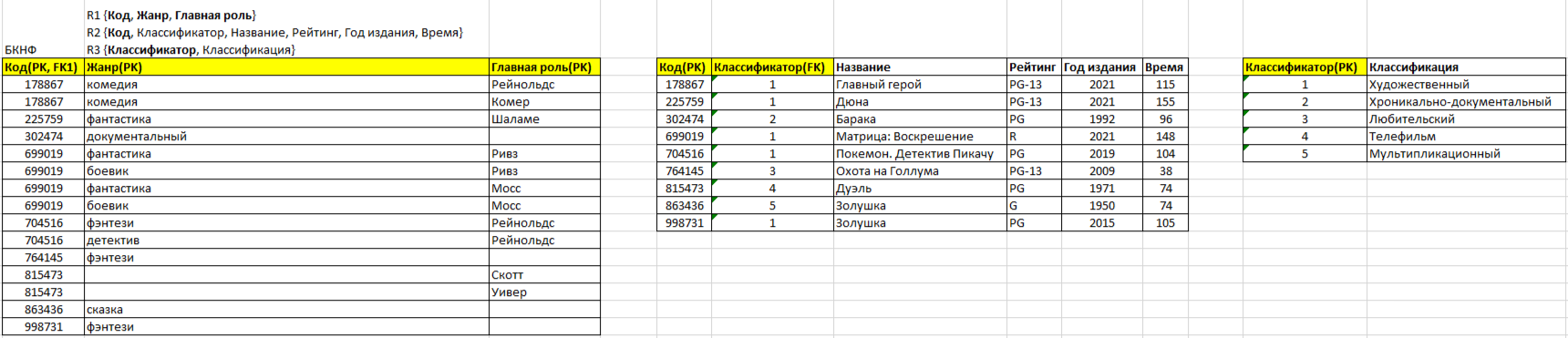
**3 Вывод**

3.1 Был изучен процесс приведения отношений от ненормализованного вида к четвертой нормальной форме;

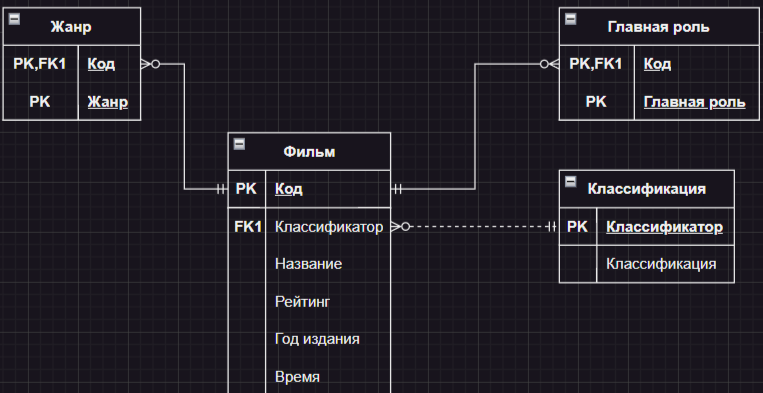
3.2 Я изучил процесс декомпозии отношений.

**4 Ход работы**

**4.4** Проверить отношения на соответствие нормальной форме Бойса-Кодда (БКНФ) и четвертой нормальной форме (4НФ). При несоответствии провести нормализацию.



5.5 Отобразить итоговую модель в формате Crow’s Foot.



**Практическая работа №3**

**Разработка физической модели данных**

**1 Цель работы**

1.1 Изучить процесс создания физической модели данных.

**2 Контрольные вопросы**

2.1 Что такое «физическая модель данных»?

Физическая модель данных — это представление логической модели в виде, соответствующем конкретной системе управления базами данных (СУБД).

2.2 Какие типы данных используются для хранения строковых значений?

Varchar, nvarchar, text

2.3 Какие типы данных используются для хранения целочисленных значений?

TINYINT, SMALLINT, INT , BIGINT.

2.4 Какие типы данных используются для хранения вещественных значений?

FLOAT, DOUBLE, REAL.

2.5 Какие типы данных используются для хранения логических значений?

BOOLEAN.

2.6 Какие типы данных используются для хранения даты и/или времени?

DATE, TIME, DATETIME.

2.7 Что такое «ограничение целостности»?

Ограничения целостности — это правила, которые помогают поддерживать точность, согласованность и надёжность данных в базе данных.

2.8 Какие виды ограничений целостности существуют?

Ограничения целостности данных и ограничения целостности ссылок.

**3 Вывод**

3.1 Я изучил процесс создания физической модели данных.

**Практическая работа №4**

**Разработка ER-модели БД в Microsoft Visio**

**1 Цель работы**

1.1 Изучить процесс создания концептуальной, логической и физической модели данных в MS Visio.

**2 Контрольные вопросы**

2.1 Что такое «MS Visio»?

Microsoft Visio — векторный графический редактор, редактор диаграмм и блок-схем для Windows.

2.2 В каких нотациях можно спроектировать схему БД в MS Visio?

Chen's, Crow's Foot, IDEF1X и UML.

2.3 Какие элементы доступны для настройки схемы БД в MS Visio?

Сущности, связи, атрибуты, индексы и ограничения.

2.4 Как задать в MS Visio первичный ключ?

В Microsoft Visio первичный ключ задаётся для столбца таблицы в окне «Свойства базы данных».

2.5 Как задать в MS Visio внешний ключ?

Внешний ключ в Microsoft Visio задаётся с помощью функции «Соединительная линия».

2.6 Как в MS Visio изменить тип связи на 1:1, М:М, 1:М?

Можно настроить в свойствах сущностию

2.7 Как в MS Visio изменить связь на идентифицирующую и не идентифицирующую?

Это доступно через настройки свойств связи в окне «Свойства базы данных».

**3 Вывод**

1.1 Был изучен процесс создания концептуальной, логической и физической модели данных в MS Visio.

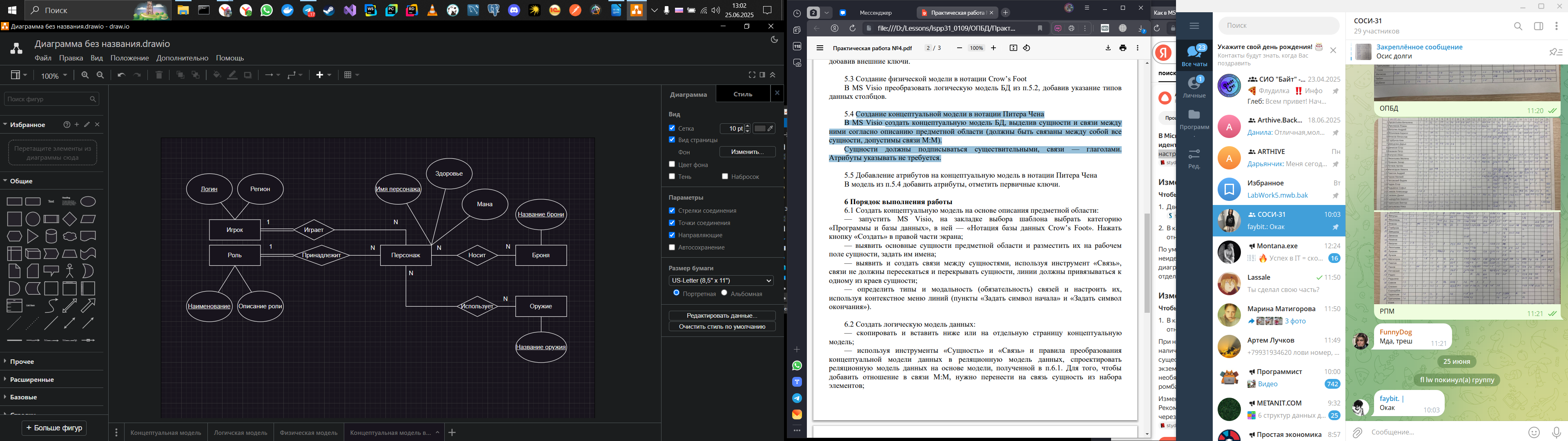
**4 Ход работы**

4.4 Создание концептуальной модели в нотации Питера Чена

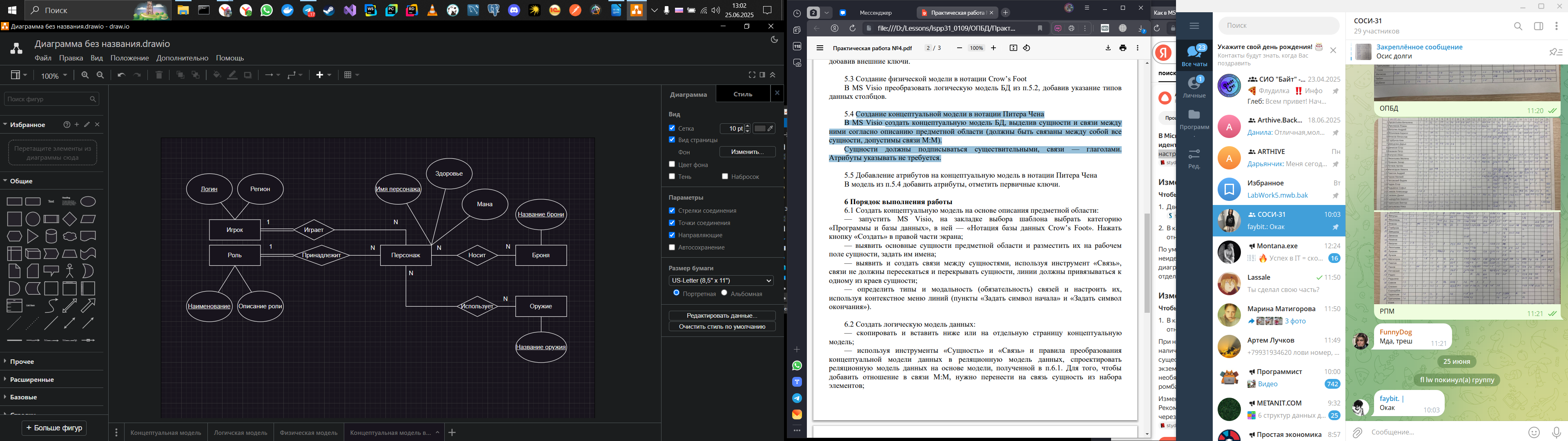
В MS Visio создать концептуальную модель БД, выделив сущности и связи между ними согласно описанию предметной области (должны быть связаны между собой все сущности, допустимы связи М:М).

Сущности должны подписываться существительными, связи — глаголами.

Атрибуты указывать не требуется.



5.5 Добавление атрибутов на концептуальную модель в нотации Питера Чена В модель из п.5.4 добавить атрибуты, отметить первичные ключи.



**Практическая работа №5**

**Разработка ER-модели БД в MySQL Workbench**

**1 Цель работы**

1.1 Научиться применять MySQL Workbench в процессе создания схем моделей БД;

1.2 Научиться представлять логическую модель данных согласно нотациям ERD и IDEF1X.

**2 Контрольные вопросы**

2.1 Что такое «сущность»?

Объект, информацию о котором нужно хранить в БД.

2.2 Что такое «атрибут»?

Это характеристика или свойство сущности или связи.

2.3 Что такое «ключевое поле»?

Ключевое поле в базе данных (первичный ключ) — это поле (или набор полей), значения которого однозначно определяют каждую запись в таблице.

2.4 Каково назначение первичных и внешних ключей?

Первичный ключ — это один или несколько столбцов таблицы, значения которых однозначно идентифицируют каждую запись (строку) в этой таблице.

Внешний ключ — это столбец или группа столбцов в одной таблице, которые указывают на первичный ключ другой (или той же самой) таблицы.

2.5 Что такое «связь»?

Связь, отображает взаимодействие между двумя сущностями.

2.6 Какие виды связей между сущностями существуют?

Один к одному; Один ко многим; Много ко многим.

2.7 Какие элементы входят в ER-диаграммы?

Сущности, связи, атрибуты.

2.8 Для чего применяются ER-диаграммы?

ER-диаграммы применяются для визуализации структуры данных в системе.

**3 Вывод**

3.1 Я научился применять MySQL Workbench в процессе создания схем моделей БД;

3.2 Я научился представлять логическую модель данных согласно нотациям ERD и IDEF1X.

**4 Ход работы**

4.1 Спроектировать в MySQL Workbench EER-модель базы данных «Ремонт ПК»

Изучить описание предметной области «Компьютерный сервисный центр» и

спроектировать в MySQL Workbench EER-модель базы данных «Ремонт ПК»

методом «сущность-связь» согласно указанной в описании информации:

Сервисный центр принимает заказы на выполнение работ в соответствии с имеющимся прейскурантом цен. Заказчик приносит оборудование и описывает проблему, работник оформляет заказ, при выполнении заказа может потребоваться проделать несколько работ (работы в заказе не могут дублироваться).

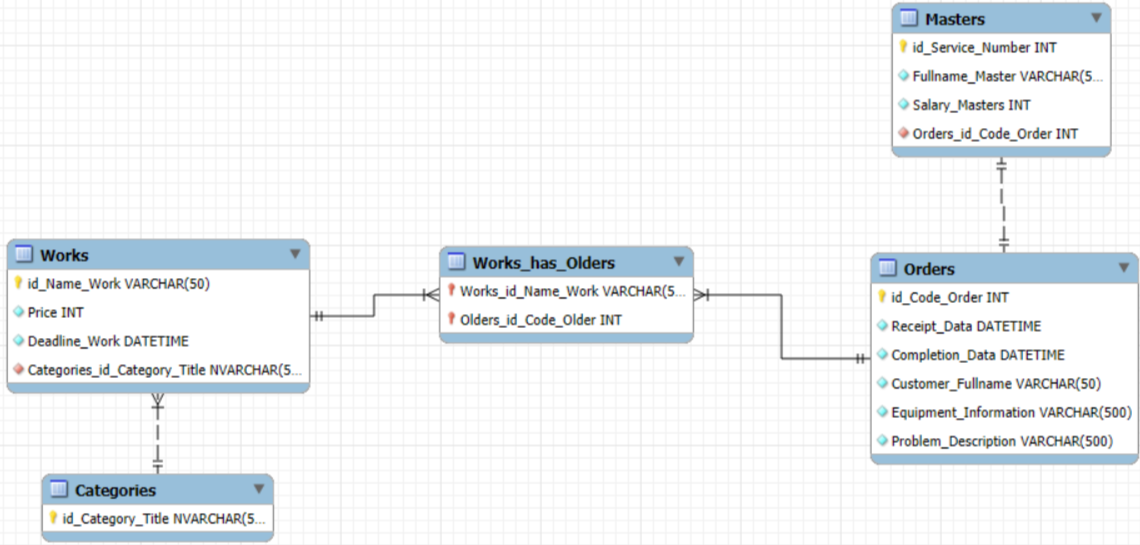
В БД должна храниться следующая информация:

- о заказах: код заказа (уникален для каждого заказа), даты поступления и выполнения заказа, ФИО заказчика (база клиентов не ведется), сведения об оборудовании, описание проблемы,

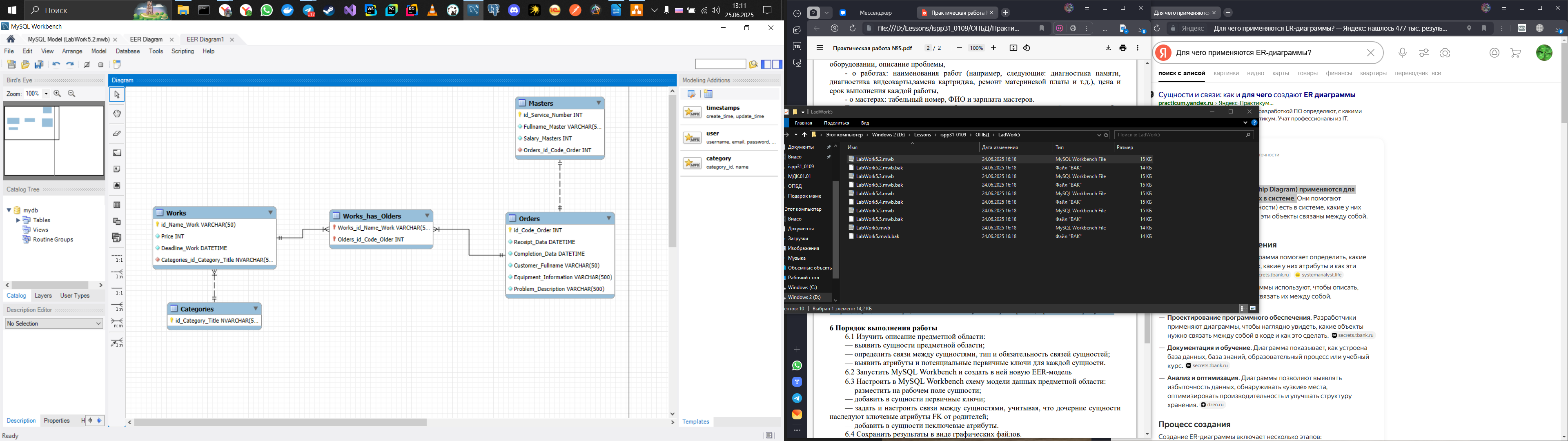
- о работах: наименования работ (например, следующие: диагностика памяти, диагностика видеокарты,замена картриджа, ремонт материнской платы и т.д.), цена и срок выполнения каждой работы,

- о мастерах: табельный номер, ФИО и зарплата мастеров.

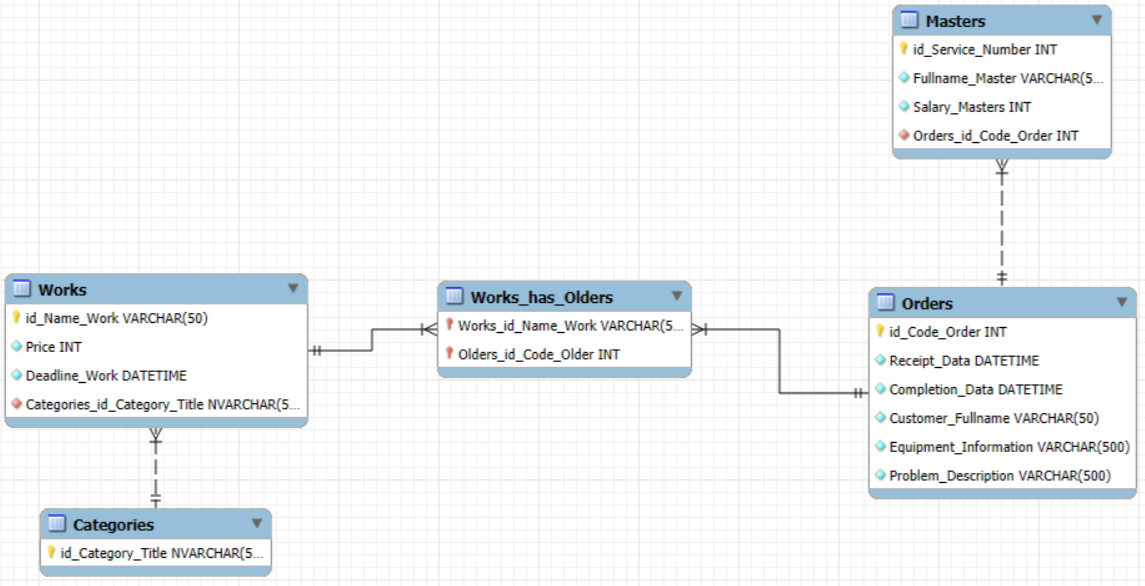
Также известно, что каждая работа относится к определенной категории работ (например, следующие: диагностика, ремонт, замена, настройка и т.д.).



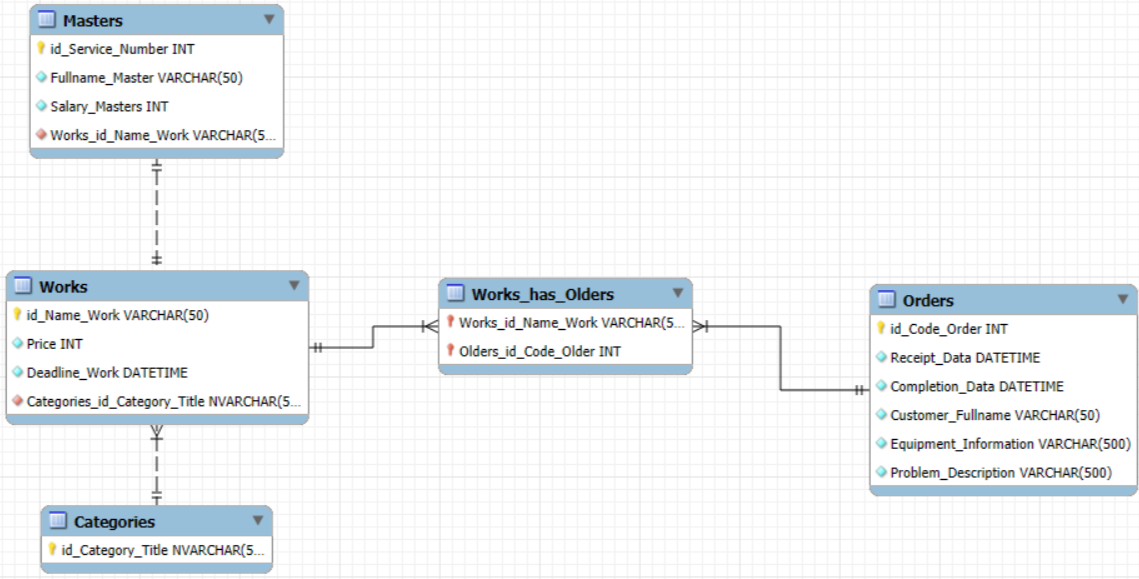
4.2 Создать копию файла с EER-моделью из п.5.1 и добавить в нее новую связь с учетом того, что каждый заказ выполняется одним мастером.



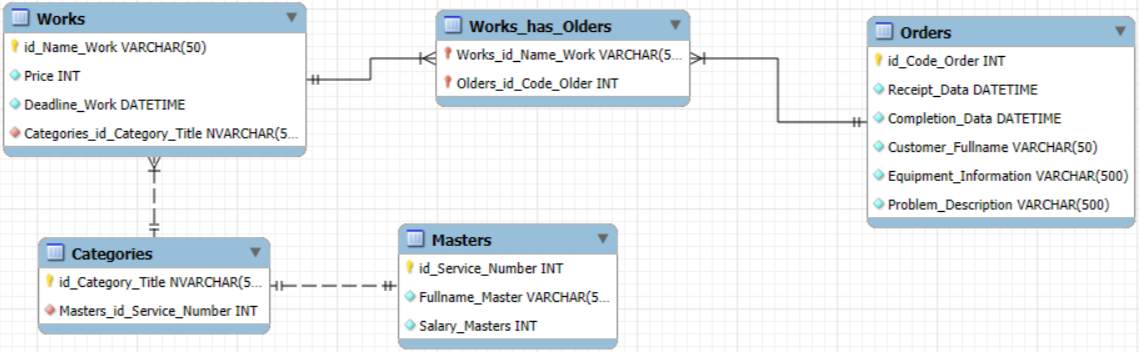
4.3 Создать копию файла с EER-моделью из п.5.1 и добавить в нее новую связь с учетом того, что над выполнением одного заказа могут работать несколько мастеров.



4.4 Создать копию файла с EER-моделью из п.5.1 и добавить в нее новую связь с учетом того, что каждый мастер выполняет определенную работу и только ее.



4.5 Создать копию файла с EER-моделью из п.5.1 и добавить в нее новую связь с учетом того, что для выполнения каждой категории работы нанимается единственный мастер и только он выполняет работы этой категории. Информацию о предыдущем мастере, выполнявшем эту категорию работ, хранить не требуется.



**Практическая работа №6**

**Создание БД MySQL**

**1 Цель работы**

1.1 Научиться создавать физическую модель данных БД MySQL, используя CASE-средство MySQL Workbench.

1.2 Научиться создавать БД MySQL, используя CASE-средство MySQL

**2 Контрольные вопросы**

2.1 Для чего применяется MySQL Workbench?

MySQL Workbench — инструмент для визуального проектирования баз данных (БД) для системы MySQL.

2.2 Как создать EER-модель в MySQL Workbench?

Чтобы создать новую EER-модель в MySQL Workbench, нужно:

1. Открыть MySQL Workbench.
2. Установить соединение с базой данных. Если база существующая, выбрать «Database» → «Connect to Database» и ввести учётные данные (хост, имя пользователя, пароль). Если базы нет, можно создать новую.
3. Создать новую EER-модель. Перейти в меню «File» → «New Model» — откроется пустой холст для проектирования базы данных.

2.3 Из каких элементов состоит EER-модель?

Сущности, связи, атрибуты.

2.4 Что такое «прямой инжиниринг»?

Прямой инжиниринг — это процесс разработки, производства и внедрения новых товаров, технологий или систем с начальной стадии концепции до выпуска на рынок и реализации.

2.5 Как провести прямой инжиниринг, используя MySQL Workbench?

1. Создать модель базы данных в MySQL Workbench. Можно использовать как новую модель, так и существующую базу данных — для этого нужно выбрать «Создать модель из существующей базы данных» или «Открыть существующую модель».
2. Настроить параметры прямого инжиниринга. Например, можно указать, нужно ли создавать отдельные запросы для индексов, или исключить их из скрипта. Также можно выбрать, сохранять ли скрипты INSERT для таблиц (если в них есть данные).
3. Запустить мастер прямого инжиниринга. Он будет последовательно задавать шаги, например, выбор объектов для экспорта или проверку скрипта.
4. Сохранить скрипт после завершения мастера. Если нужно, можно отредактировать его перед сохранением.
5. Использовать сохранённый скрипт для создания или изменения базы данных. Например, его можно запустить через клиент MySQL или выполнить в MySQL Workbench, подключившись к серверу.

**3 Вывод**

3.1 Я научился создавать физическую модель данных БД MySQL, используя CASE-средство MySQL Workbench.

3.2 Я научился создавать БД MySQL, используя CASE-средство MySQL

**Практическая работа №7**

**Изучение способов обеспечения целостности данных в СУБД MySQL**

**1 Цель работы**

1.1 Изучить способы обеспечения целостности данных в БД MySQL Server, используя CASE-средство MySQL Workbench.

**2 Контрольные вопросы**

2.1 Как в MySQL Workbench указать автоинкрементные значения?

В схеме базы данных у нужной сущности в свойствах выбрать автоинкрементирование(поставить галочку) у нужного столбца.

2.2 Как в MySQL Workbench указать значения по умолчанию?

В схеме базы данных у нужной сущности в свойствах у нужного столбца вписать значения по умолчанию.

2.3 Как в MySQL Workbench указать обязательные для заполнения столбцы?

Поставить галочку у Not Null (NN).

2.4 Как создаются индексы в MySQL Workbench?

Индексы в MySQL Workbench создаются во вкладке «Indexes» интерфейса управления таблицей.

2.5 Какие виды индексов можно создать в таблицах MySQL?

Уникальные, составные, кластерные, префиксные, полнотекстовые, пространственные.

2.6 Как в MySQL Workbench указать настройки внешних ключей?

Настройки внешних ключей в MySQL Workbench указываются во вкладке «Foreign Keys» редактора таблиц.

2.7 Как в MySQL Workbench указать уникальность значений в одном столбце?

Поставить галочку у UQ в редакторе таблиц.

2.8 Как в MySQL Workbench указать уникальность значений в наборе столбцов?

Уникальность нескольких столбцов можно создать в MySQL Workbench во вкладке «Indexes» интерфейса управления таблицей.

**3 Вывод**

3.1 Были изучены способы обеспечения целостности данных в БД MySQL Server, используя CASE-средство MySQL Workbench.

**Практическая работа №8**

**Создание БД Microsoft SQL Server**

**1 Цель работы**

1.1 Изучить процесс создания таблиц и связей между ними в СУБД Microsoft SQL Server (MSSQL).

1.2 Научиться работать в среде SQL Server Management Studio (SSMS).

**2 Контрольные вопросы**

2.1 Что такое SQL Server Management Studio?

SQL Server Management Studio — это бесплатная интегрированная среда от Microsoft для управления инфраструктурой Microsoft SQL Server.

2.2 Какие виды авторизации поддерживаются в MS SQL Server?

Аутентификация Windows, смешанный режим аутентификации.

2.3 Как создать диаграмму в SQL Server Management Studio?

1. Открыть SSMS и подключиться к базе данных.
2. В левой части окна найти раздел «Объекты», затем выбрать базу данных.
3. Щёлкнуть правой кнопкой мыши по папке «Диаграммы базы данных» и выбрать «Создать диаграмму из существующей базы данных».

2.4 Как добавить таблицы в SQL Server Management Studio?

В диаграмме БД выбрать таблицы для диаграммы и нажать «Добавить».

2.5 Как создать связи между таблицами в SQL Server Management Studio?

1. Открыть конструктор таблиц для нужной таблицы в обозревателе объектов (Object Explorer).
2. В меню конструктора выбрать «Связи» (Relationships).
3. В диалоговом окне «Связи внешнего ключа» (Foreign Key Relationships) нажать кнопку «Добавить».

**3 Вывод**

1.1 Был изучен процесс создания таблиц и связей между ними в СУБД Microsoft SQL Server (MSSQL).

1.2 Я научился работать в среде SQL Server Management Studio (SSMS).

**Практическая работа №9**

**Изучение способов обеспечения целостности данных в СУБД Microsoft SQL Server**

**1 Цель работы**

1.1 Изучить способы обеспечения целостности данных в БД Microsoft SQL Server, используя SQL Server Management Studio (SSMS).

**2 Контрольные вопросы**

2.1 Как в SSMS указать автоинкрементные значения?

Написать SQL-запрос с созданием у столбца ограничение «Автоинременции».

2.2 Как в SSMS указать значения по умолчанию?

Написать SQL-запрос с созданием у столбца ограничение «Значение по умолчанию» и указать данное значение.

2.3 Как в SSMS указать необязательные для заполнения столбцы?

При создании не создавать ограничение «NOT NULL».

2.4 Как в SSMS указать настройки внешних ключей?

Написать SQL-запрос с созданием у столбца ограничение «Внешний ключ» и указать какой столбец является внешним ключом и на какую таблицу ссылается.

2.5 Как в SSMS указать уникальность значений?

Написать SQL-запрос с созданием у столбца ограничение «Уникальное значение».

**3 Вывод**

3.1 Были изучены способы обеспечения целостности данных в БД Microsoft SQL Server, используя SQL Server Management Studio (SSMS).

**Практическая работа №10**

**Создание SQL-запросов на модификацию схемы БД**

**1 Цель работы**

1.1 Изучить процесс разработки DDL-команд в БД Microsoft SQL Server, используя SQL Server Management Studio (SSMS).

**2 Контрольные вопросы**

2.1 Какое предложение инструкции CREATE TABLE позволяет определить первичный ключ?

имя\_столбца PRIMARY KEY или CONSTRAINT имя\_ограничения PRIMARY KEY (имя\_столбца)

2.2 Какое предложение инструкции CREATE TABLE позволяет определить внешний ключ?

CONSTRAINT имя\_ограничения FOREIGN KEY (имя\_слобца) REFERENCES имя\_таблицы (имя\_столбца),

2.3 Как в инструкции CREATE TABLE задать ограничения для значений столбца?

CHECK (имя\_столбца условие)

2.4 Каково назначение предложения CONSTRAINT?

Создание ограничения.

2.5 В каких случаях нельзя удалить столбец, таблицу?

При наличии у данного столбца ограничений.

2.6 Для чего предназначена команда ALTER TABLE?

Команда ALTER TABLE предназначена для изменения структуры таблицы базы данных.

2.7 Для чего предназначена команда DROP TABLE?

Для удаления таблицы.

**3 Вывод**

3.1 Был изучен процесс разработки DDL-команд в БД Microsoft SQL Server, используя SQL Server Management Studio (SSMS).

**Практическая работа №11**

**Создание SQL-запросов на модификацию данных**

**1 Цель работы**

1.1 Изучить процесс создания таблиц и связей между ними в Microsoft SQL Server (MSSQL).

1.2 Научиться работать в среде SQL Server Management Studio (SSMS)

**2 Контрольные вопросы**

2.1 Каковы причины ошибок, связанных с выполнением DML-команд на модификацию данных?

Нарушение ограничений или ввода значений, доступ к таблице и к базе данных.

2.2 После какого ключевого слова в запросе на обновление записывается обновляемое поле?

SET

2.3 Для чего используется команда INSERT?

Для вставки записи в таблицу.

2.4 Для чего используется команда UPDATE

Для обновления записи в таблице.

2.5 Для чего используется команда DELETE?

Для удаления записи в таблице.

**3 Вывод**

3.1 Был изучен процесс создания таблиц и связей между ними в Microsoft SQL Server (MSSQL).

3.2 Я научился работать в среде SQL Server Management Studio (SSMS).

**Практическая работа №12**

**Создание однотабличных SQL-запросов на выборку данных**

**1 Цель работы**

1.1 Изучить процесс создания однотабличных SQL-запросов на выборку.

**2 Контрольные вопросы**

2.1 Какова структура инструкции SELECT?

SELECT имя\_столбцов FROM имя\_таблицы

2.2 Что указывается после ключевого слова SELECT?

SELECT имя\_таблицы

2.3 Чем отличается структура от класса?

FROM имя\_таблицы

2.4 Что указывается после ключевого слова GROUP BY?

GROUP BY имя\_столбцов

2.5 Что указывается после ключевого слова ORDER BY?

ORDER BY имя\_слобцов

**3 Вывод**

3.1 Я изучил процесс создания однотабличных SQL-запросов на выборку.

**Практическая работа №13**

**Создание однотабличных SQL-запросов на выборку с фильтрацией данных**

**1 Цель работы**

1.1 Изучить процесс фильтрации данных в SQL-запросах на выборку.

**2 Контрольные вопросы**

2.1 Что указывается после ключевого слова WHERE?

Условия фильтрации.

2.2 Что указывается после ключевого слова HAVING?

Условия включения записей в группы.

2.3 В чем отличие между разделами WHERE и HAVING?

WHERE используется для фильтрации, а HAVING используется для включения записей в группы.

2.4 Что указывается после ключевого слова IN?

Список значений, которые должны соответствовать определённым условиям.

2.5 Что указывается после ключевого слова LIKE?

После ключевого слова LIKE в SQL указывается выражение, содержащее шаблон подстроки для поиска.

2.6 Что указывается после ключевого слова BETWEEN?

указывается имя поля и два значения, которые определяют диапазон.

**3 Вывод**

3.1 Я изучил процесс фильтрации данных в SQL-запросах на выборку.

**Практическая работа №14**

**Создание многотабличных SQL-запросов на выборку данных**

**1 Цель работы**

1.1 Изучить процесс создавать многотабличных SQL-запросов на выборку.

**2 Контрольные вопросы**

2.1 Что указывается после ключевого слова FROM?

Имя таблицы.

2.2 Каков синтаксис инструкции объединения двух таблиц?

(INNER/LEFT/RIGHT/CROSS) JOIN имя\_таблицы ON условия объединения

2.3 Какие виды объединения таблиц существуют?

INNER/LEFT/RIGHT/CROSS.

2.4 Чем отличается объединение LEFT JOIN от RIGHT JOIN?

LEFT JOIN создаёт левое внешнее соединение. В результат попадают совпадающие по ключу данные обеих таблиц и все записи из левой таблицы, для которых не нашлось пары в правой. Если в правой таблице нет соответствующей строки, то возвращается NULL для значений правой таблицы. RIGHT JOIN создаёт правое внешнее соединение. В результат попадают все записи второй (правой) таблицы, даже если они не соответствуют записям в первой (левой) таблице.

2.5 Что такое подзапрос?

Это запрос, который используется внутри другого запроса.

2.6 Для чего применяется ключевое слово UNION?

Ключевое слово UNION в языке SQL применяется для объединения результатов двух и более запросов SELECT в единую таблицу.

**3 Вывод**

3.1 Был изучен процесс создавать многотабличных SQL-запросов на выборку.

**Практическая работа №15**

**Применение встроенных функций SQL**

**1 Цель работы**

1.1 Изучить процесс создания SQL-запросов, содержащих встроенные функции в СУБД MS SQL Server.

**2 Контрольные вопросы**

2.1 В чем преимущество применения встроенных функций?

Избавления от повторного использования кода, обработка записей на стороне БД.

2.2 Какие виды системных функций имеются в MSSQL?

Математические, даты и времени, строковые, преобразования типов, оконные.

2.3 Каким образом указываются передаваемые в функцию параметры?

имя\_функции(список параметров).

**3 Вывод**

3.1 Я изучил процесс создания SQL-запросов, содержащих встроенные функции в СУБД MS SQL Server.

**4 Ход работы**

4.3 Уменьшить на 10% цену всех товаров, которые не заказывались в текущем году.

UPDATE Book

SET Price = Price \* 0.9

WHERE BookId IN (SELECT b.BookId

FROM Book b

LEFT JOIN OrderedBook ob ON b.BookId = ob.BookId JOIN [Order] o ON ob.OrderId = o.OrderId

WHERE YEAR(OrderDate) != YEAR(GETDATE()) OR OrderDate IS NULL);

4.6 Вывести на экран всю информацию о покупателях:

- если телефон не указан, при выводе должен быть написан «–» (прочерк),

- если адрес пустая строка, при выводе должно быть NULL-значение.

SELECT

CustomerId,

[Login],

Surname,

[Name],

NULLIF(TRIM([Address]), '') [Address],

ISNULL(Phone, '-') Phone

FROM Customer;

4.8 Вывести на экран список авторов и названий книг, сгруппированных по заказам.

SELECT STRING\_AGG(CONCAT(Surname, ' ', [Name], ' ', Title), '; ')

FROM OrderedBook ob

JOIN Book b ON ob.BookId = b.BookId

JOIN Author a ON b.AuthorId = a.AuthorId

GROUP BY OrderId;

4.10 Вывести на экран отсортированный по группам и названиям список всех книг, пронумерованный по порядку. Для каждого жанра должна быть своя нумерация.

SELECT ROW\_NUMBER() OVER (PARTITION BY Genre ORDER BY Title) AS #, Genre, Title

FROM Book;