МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ   
ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «ВГТУ», ВГТУ)

ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И КОМПЬЮТЕРНОЙ БЕЗОПАСНОСТИ

КАФЕДРА КОМПЬЮТЕРНЫХ ИНТЕЛЛЕКТУАЛЬНЫХ ТЕХНОЛОГИЙ  
ПРОЕКТИРОВАНИЯ

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №7  
  
По дисциплине Системы хранения и обработки данных

Тема: Разработка логической структуры базы данных

Выполнил работу студент группы мИИВТ-231 Ахлестин А.И. (подпись) Фамилия, инициалы

Принял \_\_\_\_\_\_\_\_\_\_\_\_Короленко В.В.  
 (подпись) Фамилия, инициалы

Защищена\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Оценка\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Воронеж 2023

Цель работы: изучить основы логического проектирования базы данных, освоить процесс разработки логической структуры базы данных и построения диаграммы «сущность-связь».

Задание на лабораторную работу:

1. Определение сущности для проекта и их атрибутов;
2. Выделение ключевых атрибутов;
3. Определение связей между сущностями и типов связей;
4. Построение диаграммы сущность-связь.

Ход выполнения задания:

# 1 Определение сущности для проекта и их атрибутов

# В базах данных, сущность представляет собой объект или явление, которое может быть идентифицировано и о котором необходимо хранить данные. Для проекта было выделено 4 типа сущностей: “ВРАЧИ”, “ПАЦИЕНТЫ”, “ПРИЕМ ПАЦИЕНТОВ” и “КВИТАНЦИЯ”.

# Врачи являются ключевыми участниками системы платного приема в поликлинике. Их основные характеристики, такие как ФИО и специальность, необходимы для их однозначной идентификации и предоставления информации пациентам.

# Пациенты - это клиенты поликлиники, для которых ведется врачебная документация. Их личная информация, такая как ФИО, дата рождения и адрес, является неотъемлемой частью базы данных для предоставления медицинских услуг.

# Сущность "ПРИЕМ ПАЦИЕНТОВ" связывает врачей и пациентов в рамках конкретного приема. Информация о стоимости приема и дате приема необходима для учета финансов и организации приемов.

# Сущность "КВИТАНЦИИ" позволяет отслеживать финансовые операции, связанные с оплатой приемов. Включение этой сущности обеспечивает систему учета платежей и возможность расчета заработной платы для врачей.

# 2 Выделение ключевых атрибутов

# Выделим атрибуты для каждой сущности:

# ВРАЧИ: Врач\_ID (PK), ФИО врача, Специальность врача, Процент отчисления на зарплату;

# ПАЦИЕНТЫ: Пациент\_ID (PK), Фамилия пациента, Имя пациента, Отчество пациента, Дата рождения пациента, Адрес пациента;

# ПРИЕМ ПАЦИЕНТОВ: Прием\_ID (PK), Врач\_ID (FK), Пациент\_ID (FK), Дата приема, Стоимость приема.

# Для каждой сущности были выделены внешние и первичные ключи (FK и PK). Выбор первичных ключей (PK) обеспечивает уникальность записей в каждой из сущностей, а внешних ключей (FK) обеспечивает связи между этими сущностями, что важно для поддержания целостности данных в базе данных.

# Рассмотрим ключи для каждой из сущностей.

# ВРАЧИ:

# Врач\_ID (PK): Уникальный идентификатор врача.

# Этот ключ обеспечивает уникальную идентификацию каждого врача в системе. Использование уникального идентификатора упрощает ссылки на врачей из других таблиц и обеспечивает уникальность данных в данной таблице.

# ПАЦИЕНТЫ:

# Пациент\_ID (PK): Уникальный идентификатор пациента.

# Аналогично ключу врачей, уникальный идентификатор пациента гарантирует, что каждый пациент может быть однозначно идентифицирован в системе.

# ПРИЕМ ПАЦИЕНТОВ:

# Прием\_ID (PK): Уникальный идентификатор приема.

# Данный ключ обеспечивает уникальную идентификацию каждого приема. Он позволяет легко ссылаться на конкретные приемы и использовать их как внешний ключ в других связанных таблицах.

# Врач\_ID (FK): Внешний ключ, связывающий прием с конкретным врачом.

# Этот внешний ключ устанавливает связь между приемом и врачом, позволяя эффективно отслеживать, какой врач проводил конкретный прием.

# Пациент\_ID (FK): Внешний ключ, связывающий прием с конкретным пациентом.

# Аналогично ключу врачей, внешний ключ пациента обеспечивает связь между приемом и конкретным пациентом, которому он был проведен.

# КВИТАНЦИИ:

# Квитанция\_ID (PK): Уникальный идентификатор квитанции.

# Уникальный идентификатор квитанции гарантирует уникальность каждой квитанции в системе.

# Прием\_ID (FK): Внешний ключ, связывающий квитанцию с конкретным приемом.

# Этот внешний ключ связывает квитанцию с определенным приемом, позволяя отслеживать, за какой прием была выставлена квитанция.

# 3 Определение связей между сущностями и типов связей

# В данной базе данных "Платный прием в поликлинике", сущности связаны следующим образом:

# ВРАЧИ и ПРИЕМ ПАЦИЕНТОВ связаны “Один ко многим”, что позволяет каждому врачу провести несколько приемов, но каждый прием будет проводиться только одним врачом.

# ПАЦИЕНТЫ и ПРИЕМ ПАЦИЕНТОВ также связаны “Один ко многим”, чтобы каждый пациент мог пройти несколько приемов, но каждый прием относился только к одному пациенту.

# На данный момент в базе данных есть связи "многие ко многим" между ВРАЧАМИ и ПРИЕМАМИ ПАЦИЕНТОВ. Чтобы избежать этого, введем дополнительную сущность "ПРИЕМ", которая выступит в качестве промежуточной таблицы для связи врачей и пациентов.

# ВРАЧИ и ПРИЕМ ПАЦИЕНТОВ будут связаны с ПРИЕМОМ при помощи связи “Один ко многим”, чтобы каждый врач мог иметь множество приемов, но каждый прием относится только к одному врачу и каждый пациент мог иметь множество приемов, но каждый прием относится только к одному пациенту.

# 4 Построение диаграммы сущность-связь

# Построим схему базы данных. Схема базы данных отображена на рисунке 1.

# 

# Рисунок 1 – Схема базы данных

# На диаграмме изображена схема базы данных, которая описывает взаимодействие пациентов и врачей через прием. Каждый врач может провести множество приемов, а пациент может записаться на множество приемов. Прием пациентов содержит в себе информацию о всех существующих приемах и пациентах, которые на них записаны. При помощи приема пациентов мы можем получить сам прием и узнать данные о пациенте, враче, стоимости и дате прохождения. Квитанция позволяет узнать о всю информацию о приёме.

Вывод: в процессе выполнения работы были освоены основы работы с draw.io, а также получены навыки построения схем базы данных.

Контрольные вопросы:

1. Определения:

Данные: Это факты, представленные в форме, пригодной для хранения, передачи и обработки компьютерной системой.

База данных: Совокупность взаимосвязанных данных, хранящихся электронными средствами и организованных таким образом, чтобы они были легко доступны и обрабатываемы.

СУБД (Система Управления Базами Данных): Программное обеспечение, предназначенное для создания, управления и манипулирования базами данных.

Ведение базы данных: Это процесс организации, хранения и управления данными в базе данных, включая добавление, обновление и извлечение информации.

1. Отличие данных от информации:

Данные - это факты и статистика, представленные без контекста.

Информация - это данные, организованные и представленные в контексте, что делает их полезными для принятия решений.

1. Отличие базы данных от банка данных и СУБД:

База данных: Хранилище данных с определенной структурой.

Банк данных: Совокупность взаимосвязанных баз данных.

СУБД: Программное обеспечение для управления базами данных.

1. Основные компоненты банка данных и их назначение:

База данных: Хранилище данных.

Метаданные: Информация о структуре и описании данных.

Язык данных: Средства для определения, манипулирования и запросов к данным.

1. Классификация АИС по типу хранимых данных:

АИС с текстовой информацией.

АИС с числовой информацией.

АИС с графической информацией.

АИС с звуковой информацией.

1. Трехуровневая архитектура ANSI/SPARC:

Внешний уровень (уровень зрения пользователя): Описывает, как видят данные конечные пользователи.

Концептуальный уровень (уровень данных): Содержит описание структуры всей базы данных.

Внутренний уровень (физический уровень): Связан с тем, как данные хранятся и обрабатываются на уровне физических устройств.

1. Определения схем БД:

Внешняя схема БД: Описывает, как данные видны конкретному пользователю или группе пользователей.

Концептуальная схема БД: Описывает структуру всей базы данных независимо от конкретных пользователей.

Внутренняя схема БД: Описывает, как данные физически хранятся и организованы.

1. Особенности иерархической модели организации данных:

Деревообразная структура данных.

Записи связаны "предок-потомок".

Пример: IMS (Information Management System).

1. Особенности сетевой модели организации данных:

Графовая структура данных.

Записи связаны "родитель-потомок" или "владелец-член".

Пример: CODASYL (Conference on Data Systems Languages).

1. Особенности многомерной модели организации данных:

Данные представлены в виде "кубов".

Каждое измерение куба представляет собой атрибут данных.

Пример: OLAP (Online Analytical Processing).

1. Особенности постреляционной модели организации данных:

Данные организованы в виде таблиц.

Связи между таблицами устанавливаются с использованием ключей.

Пример: MySQL, Oracle, Microsoft SQL Server.

1. Неструктурированные данные:

Это данные, которые не имеют фиксированной структуры или организации.

Например, текстовые документы, аудио- и видеозаписи.

1. Преимущество колоночной СУБД по сравнению с реляционной:

Экономия места за счет хранения данных по столбцам.

Улучшенная производительность при агрегатных запросах.

1. Связь между таблицами в реляционной СУБД:

Связь устанавливается с использованием внешних ключей, которые связывают столбцы одной таблицы с другой.

1. Решение проблемы дублирующих записей:

Использование уникальных ключей для идентификации записей.

Нормализация базы данных для уменьшения дублирования данных.