МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
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

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №7

По дисциплине: «Системы хранения и обработки данных»

Тема: «Разработка логической структуры базы данных»

Выполнил работу студент группы мИИВТ-251: Вытовтова Н.Ю.

подпись, дата

Воронеж 2025

Содержание

1. Введение………………………………………………………………………………...3

2. Порядок выполнения учебной задачи.……………………………..…………………3

2.1 Основные понятия теории баз данных. Основные модели данных.…..…..3

2.2 Изучение средства для разработки диаграмм draw.io.……………………..6

2.3 Определение основных сущностей для разрабатываемой  
базы данных …..........................................................................................................8

2.4 Определение атрибутов сущностей...……………………………………….8

2.5 Выделение ключевых атрибутов сущностей ………………………………9

2.6 Определение связей между сущностями ……………………………………10

2.7 Пояснение выбора типов связей …………………………………………….11

2.8 Введение дополнительных сущностей……………………………………..11

2.9 Построение диаграммы сущность-связь ……………………………….…11

3. Заключение…………………………………………………………………………....13

4 Ответы на контрольные вопросы………………………………………………….....14

Список используемой литературы……………………………………………………19

**1 Введение**

Целью данной лабораторной работы является изучение основы логического проектирования базы данных, освоение процесса разработки логической структуры базы данных и построения диаграммы «сущность-связь».

Задачей лабораторной работы является выполнение следующих действий:

* определение сущности для проекта в соответствии с индивидуальным заданием и их атрибуты;
* выделение ключевых атрибутов;
* определение связей между сущностями и типов связей;
* построение диаграммы сущность-связь для отображения логической структуры базы данных.

**2 Порядок выполнения учебной задачи**

2.1 Основные понятия теории баз данных. Основные модели данных

*Автоматизированная информационная система (АИС)* – это система, в которой представление, хранение и обработка информации осуществляются с помощью вычислительной техники. Центральным элементом любой АИС является база данных, которую необходимо постоянно поддерживать в актуальном состоянии.

*Данные* – представление объектов реального мира и их свойств в формализованном виде, пригодном для хранения, передачи, интерпретации или обработки. В случае использования данных для уменьшения неопределенности знаний о каком-либо объекте данные превращаются в информацию.

*База данных (БД)* – совокупность данных, организованных по определенным правилам, предусматривающим общие принципы описания, хранения и манипулирования данными, независимая от прикладных программ.

*Ведение базы данных* – деятельность по обновлению, восстановлению и перестройке структуры базы данных с целью обеспечения ее целостности, сохранности и эффективности использования.

*Система управления базами данных (СУБД)* – совокупность программ и языковых средств, предназначенных для управления данными в базе данных, ведения базы данных и обеспечения взаимодействия ее с прикладными программами.

*Сущность* – это представление набора реальных или абстрактных объектов (людей, вещей, мест, событий т. д.), которые имеют общие атрибуты или характеристики. Каждая сущность должна иметь наименование, выраженное существительным в единственном числе.

*Атрибут сущности* – это именованная характеристика, являющаяся некоторым свойством сущности.

*Экземпляр сущности* – это конкретный представитель данной сущности.

*Связь* – это некоторая ассоциация между двумя сущностями. Одна сущность может быть связана с другой сущностью или сама с собою. Связи позволяют по одной сущности находить другие сущности, связанные с ней.

*Модель данных (МД)* – совокупность правил порождения структур данных в базе данных, операций над ними, а также ограничений целостности, определяющих допустимые связи и значения данных, последовательность их изменения.

Самой первой МД, которая использовалась для построения концептуальных схем была иерархическая модель. Вслед за ней появились сетевые модели. Потом ER-модели, и, как итог развития моделей, возникли реляционные и постреляционные модели.

*Иерархическая модель данных (ИМД)* – это модель, в которой абстрактные понятия находятся в отношении предшествования таким образом, что каждому понятию соответствует только один предшественник (родитель).

Достоинство ИМД: данные, отражающие общие свойства совокупности конкретных данных не дублируются.

Недостаток ИМД связан с дублированием данных в случае их однотипности.

Если в иерархических структурах запись-потомок должна иметь ровно одного предка, то в *сетевых структурах данных* потомок может иметь любое число предков. Связи между записями в сетевой МД выполняются в виде указателей, то есть каждая запись хранит ссылку на другую однотипную запись (или признак конца списка) и ссылки на списки подчиненных записей.

Достоинство сетевых МД: данные имеют четкую структуру.

Недостатком сетевых МД является большое количество дополнительной информации о связях.

*Реляционная модель данных (РМД)* – это модель, в которой данные можно представить в виде отношений, изменяющихся во времени. Традиционно в реляционных системах отношением называют таблицу, кортежем – строку таблицы, а атрибутом – столбец. При этом атрибуты имеют уникальные имена в рамках одного отношения.

Достоинство реляционной модели заключается в простоте для понимания, наглядности и удобстве физической реализации на ЭВМ.

Недостатки реляционной модели данных: модель не допускает представления объектов со сложной структурой, поскольку в ее рамках возможно моделирование лишь с помощью двумерных таблиц. Данные об объектах содержатся, как правило, во многих таблицах, что значительно замедляет обработку данных.

В последнее время при разработке БД активно используются такие модели, как *постреляционная, объектно-ориентированная, объектно-реляционная и многомерная модели.*

*Постреляционная модель* данных в общем случае представляет собой расширенную реляционную модель, снимающую ограничение неделимости значений полей. То есть, допускаются многозначные поля, значения которых состоят из подзначений. Набор значений многозначных полей считается самостоятельной таблицей, встроенной в основную.

Достоинство постреляционной модели данных: возможность представления связанных реляционных таблиц одной постреляционной таблицей.

Недостаток постреляционной модели: сложность в обеспечении целостности данных.

*Объектно-ориентированная модель данных (ООМД)* представляет структуру, которую можно изобразить графически в виде дерева, узлами которого являются объекты.

Каждый объект характеризуется уникальным идентификатором, состоянием и поведением. Состояние объекта определяется множеством значений его свойств (атрибутов). Поведение объекта описывают методы, называемые процедурами. То есть, составной частью описания объекта являются процедуры, способные производить действия над атрибутами объекта в случае наступления тех или иных событий. Объекты могут объединяться в классы. Экземпляры одного класса отличаются лишь значениями своих свойств, но не своими методами. Методы устанавливаются при определении класса.

Для выполнения действий над объектами применяются объектно-ориентированные механизмы – наследование, инкапсуляция, полиморфизм.

*Объектно-реляционная модель данных (ОРМД)* является гибридной моделью, сочетающей возможности реляционной модели с объектными свойствами данных. В ОРМД используются такие объектно-ориентированные компоненты, как инкапсуляция, полиморфизм, наследование и т.п. Отличительная особенность объектно-реляционной модели от ООМД состоит в том, что она основана на стратегии реляционной модели.

2.2 Изучение средства для разработки диаграмм draw.io

Draw.io – это кроссплатформенное программное обеспечение с открытым исходным кодом, предназначенное для создания векторных диаграмм и визуализации данных.

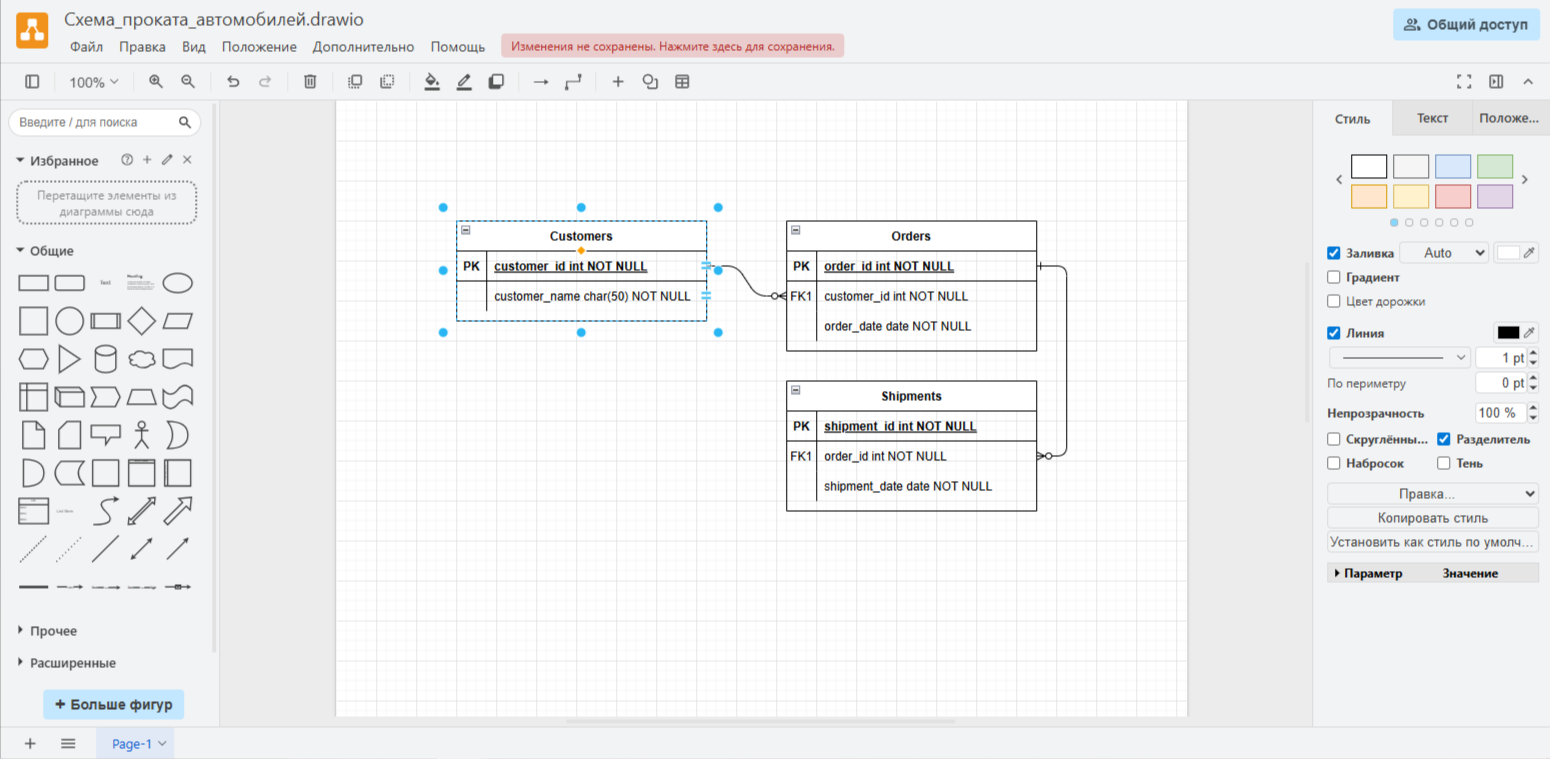


Рис. 1 – Средство для разработки диаграмм draw.io

Ключевые характеристики и особенности draw.io:

1. Архитектура и доступность. Приложение является веб-сервисом, не требующим установки и регистрации, что обеспечивает высокую доступность. Также существуют десктопные версии для основных операционных систем.

2. Функциональность. Программа предоставляет обширную библиотеку стандартизированных графических элементов (нотация BPMN, UML, схемы сетей, блок-схемы и др.) для построения структурно-логических диаграмм. Интерфейс реализован по принципу drag-and-drop, что упрощает процесс проектирования.

3. Интеграция и экспорт. Ключевым преимуществом является глубокая интеграция с облачными сервисами хранения данных (Google Drive, OneDrive) и платформами (например, Confluence, Jira). Поддерживается экспорт результатов работы в широком спектре форматов, включая векторные (SVG, PDF) и растровые (PNG, JPEG), а также собственный XML-формат для последующего редактирования.

4. Лицензирование. Распространение осуществляется на бесплатной основе под лицензией Apache 2.0, что делает инструмент доступным для образовательных, коммерческих и научных целей без лицензионных отчислений.

2.3 Определение основных сущностей для разрабатываемой базы данных

На основе анализа предметной области и предложенного набора таблиц я выделяю следующие сущности:

1. Clients (Клиенты) - Хранение информации о клиентах, берущих автомобили в прокат. Это фундаментальная сущность для любого бизнеса, работающего с клиентами.

2. Cars (Автомобили) - Хранение информации об автомобильном парке компании. Это основная сущность, которая является "товаром" компании.

3. Rentals (Прокаты) - Сущность для регистрации факта проката автомобиля клиенту. Эта сущность является "операционной" и связывает клиентов и автомобили, фиксируя детали сделки.

4. InsurancePolicies (Страховые полисы) - Новая сущность. По условию задачи страховой взнос выплачивается ежегодно. Один автомобиль может быть застрахован несколько раз за разные годы. Связь между "Автомобилем" и "Страховым полисом" - "один ко многим" (один автомобиль может иметь много полисов на разные годы). Это более корректно отражает бизнес-процесс, чем хранение просто "страховой стоимости" в сущности Cars.

Такой набор сущностей покрывает все данные из технического задания и правильно моделирует бизнес-процессы: кто (Clients), что (Cars), когда и на какой срок (Rentals), а также как обеспечивается страховая защита (InsurancePolicies).

2.4 Определение атрибутов сущностей

1. Сущность Clients (Клиенты). Атрибуты сущности:

- client\_id (Идентификатор клиента);

- last\_name (Фамилия);

- first\_name (Имя);

- patronymic (Отчество);

- passport\_serial (Серия паспорта);

- passport\_number (Номер паспорта)

2. Сущность Cars (Автомобили). Атрибуты сущности:

- car\_id (Идентификатор автомобиля);

- model (Модель автомобиля);

- color (Цвет автомобиля);

- manufacture\_year (Год выпуска);

- license\_plate (Государственный номер);

- rental\_cost\_per\_day (Стоимость одного дня проката).

3. Сущность: InsurancePolicies (Страховые полисы). Атрибуты сущности:

- policy\_id (Идентификатор полиса);

- car\_id (Идентификатор автомобиля, внешний ключ);

- insurance\_value (Страховая стоимость автомобиля на момент оформления полиса);

- start\_date (Дата начала действия полиса);

- end\_date (Дата окончания действия полиса).

4. Сущность: Rentals (Прокаты). Атрибуты сущности:

- rental\_id (Идентификатор проката);

- client\_id (Идентификатор клиента, внешний ключ);

- car\_id (Идентификатор автомобиля, внешний ключ);

- start\_date (Дата начала проката);

- rental\_days (Количество дней проката);

- total\_cost (Общая стоимость проката) – вычисляемое поле: rental\_cost\_per\_day \* rental\_days. Может быть сохранено в БД для истории.

2.5 Выделение ключевых атрибутов сущностей

1. Clients.

Первичный ключ (PK): client\_id является искусственным идентификатором. Он более эффективен для связей и индексирования, чем использование составного ключа из серии и номера паспорта.

Альтернативный ключ (Unique) – passport\_serial, passport\_number гарантирует уникальность клиента в системе, так как паспортные данные уникальны.

2. Cars.

Первичный ключ (PK): car\_id является стандартной и надежной практикой.

3. InsurancePolicies.

Первичный ключ (PK): policy\_id.

Внешний ключ (FK): car\_id -> Cars(car\_id) связывает полис с конкретным автомобилем.

4. Rentals.

Первичный ключ (PK): rental\_id.

Внешний ключ (FK): client\_id -> Clients(client\_id) связывает прокат с конкретным клиентом.

Внешний ключ (FK): car\_id -> Cars(car\_id) связывает прокат с конкретным автомобилем.

2.6 Определение связей между сущностями

1. Clients – Rentals: Один ко многим (1:M). Один клиент может брать машину в прокат много раз. Одна запись о прокате принадлежит только одному клиенту.

2. Cars – Rentals: Один ко многим (1:M). Один автомобиль может быть взят в прокат много раз (в разные даты). Одна запись о прокате соответствует только одному автомобилю.

3. Cars – InsurancePolicies: Один ко многим (1:M). Один автомобиль может иметь несколько страховых полисов (на разные годы). Один полис принадлежит только одному автомобилю.

4. Clients – Cars: Многие ко многим (M:N). Изначальная связь. Один клиент может арендовать много разных автомобилей. Один автомобиль может быть арендован многими клиентами. Эта связь требует разрешения через промежуточную сущность.

2.7 Пояснение выбора типов связей

Связи «один-ко-многоим» 1:M (Clients-Rentals, Cars-Rentals, Cars-InsurancePolicies) – это наиболее естественное и правильное отражение бизнес-логики. Прокат (Rental) является "транзакционной" сущностью, которая фиксирует событие взаимодействия двух независимых сущностей (Client и Car). Аналогично, полис (InsurancePolicy) является событием, связанным с сущностью Car.

Связь «многие ко многим» M:N (Clients-Cars): Это изначальная, неразрешенная связь. Она напрямую не реализуема в реляционной БД и служит указанием на необходимость создания связующей таблицы (которой в создаваемой модели как раз и является Rentals).

2.8 Введение дополнительных сущностей

Связь «многие ко многим» между Clients и Cars уже разрешена путем введения промежуточной сущности Rentals. Каждая запись в Rentals содержит внешние ключи на client\_id и car\_id, что позволяет установить, какой клиент арендовал какой автомобиль и когда. Таким образом, связь M:N заменена на две связи 1:M.

2.9 Построение диаграммы сущность-связь

Для отображения логической структуры проектируемой базы данных с использованием средства для разработки диаграмм draw.io была построена диаграмма:

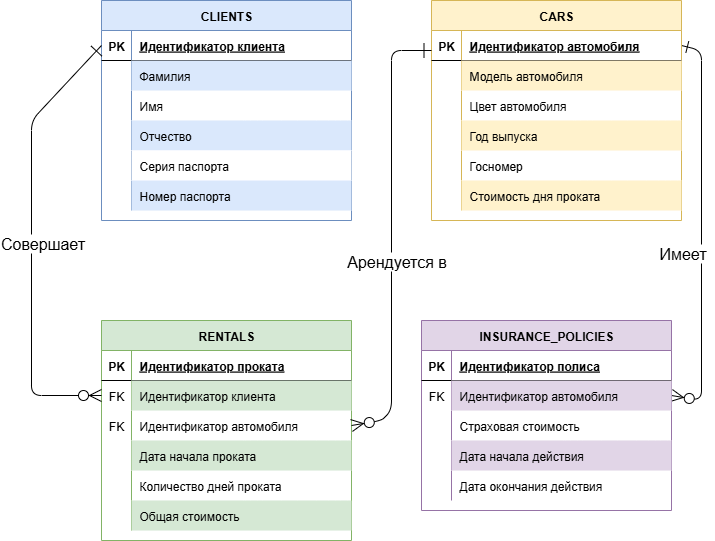


Рис. 2 – Диаграммы сущность-связь

Выбор сущностей для диаграммы сущность-связь для базы данных «Прокат автомобилей» (Рис. 2) осуществлялся в следующем порядке:

1. CLIENTS (Клиенты) - хранит информацию о клиентах проката. Это фундаментальная сущность для учета клиентской базы.

2. CARS (Автомобили) - содержит данные об автомобильном парке компании. Это основная сущность, представляющая «товар» компании.

3. RENTALS (Прокаты) - операционная сущность, фиксирующая факт аренды автомобиля клиентом. Связывает клиентов и автомобили.

4. INSURANCE\_POLICIES (Страховые полисы) - добавлена для корректного отражения процесса ежегодного страхования автомобилей.

Выбор атрибутов осуществлялся в следующем порядке:

1. Введены суррогатные ключи (client\_id, car\_id, rental\_id, policy\_id) для обеспечения надежности связей

2. Атрибуты соответствуют техническому заданию (ФИО, паспортные данные, модель, цвет, год выпуска и т.д.)

3. Добавлено вычисляемое поле total\_cost для ускорения формирования отчетов.

Выбор связей был сделан в связи со следующим обоснованием:

1. CLIENTS → RENTALS (1:М) - один клиент может совершать множество прокатов.

2. CARS → RENTALS (1:М) - один автомобиль может арендоваться многократно.

3. CARS → INSURANCE\_POLICIES (1:М) - один автомобиль может иметь несколько страховых полисов.

Для отражения логики взаимодействия сущностей были выбраны ключи:

1. Первичные ключи (PK) - суррогатные идентификаторы (id) для обеспечения стабильности связей

2. Внешние ключи (FK) - обеспечивают ссылочную целостность данных

Данная модель обеспечивает целостность данных, исключает аномалии и эффективно обслуживает бизнес-процессы проката автомобилей.

**3 Заключение**

В ходе выполнения данной лабораторной работы были успешно достигнуты все поставленные учебные задачи и приобретены практические навыки работы с таким средством для разработки диаграмм как draw.io.

Построенная в ходе лабораторной работы диаграмма наглядно отображает логическую структуру базы данных «Прокат автомобилей».

На диаграмме удалось отобразить:

1. Сущности: Clients, Cars, Rentals, InsurancePolicies.

2. Ключи: Для каждой сущности определен первичный ключ PK (например, client\_id). Внешние ключи FK (например, client\_id в таблице Rentals) четко показаны и связаны с соответствующими первичными ключами.

3. Связи:

- Clients 1 ──< Rentals («Один клиент может иметь много прокатов»).

- Cars 1 ──< Rentals («Один автомобиль может быть во многих прокатах»).

- Cars 1 ──< InsurancePolicies («Один автомобиль может иметь много страховых полисов»).

Таким образом, draw.io – это эффективное, доступное и безопасное решение для задач визуального моделирования в академической и профессиональной среде, сочетающее простоту использования с достаточной для большинства задач функциональностью.

**4 Ответы на контрольные вопросы**

1) Дайте определения следующим понятиям: данные, база данных, СУБД, ведение базы данных.

*Данные* – представление объектов реального мира и их свойств в формализованном виде, пригодном для хранения, передачи, интерпретации или обработки.

*База данных (БД)* – совокупность данных, организованных по определенным правилам, предусматривающим общие принципы описания, хранения и манипулирования данными, независимая от прикладных программ.

*Система управления базами данных (СУБД)* – совокупность программ и языковых средств, предназначенных для управления данными в базе данных, ведения базы данных и обеспечения взаимодействия ее с прикладными программами.

*Ведение базы данных* – деятельность по обновлению, восстановлению и перестройке структуры базы данных с целью обеспечения ее целостности, сохранности и эффективности использования.

2) В чем отличие данных от информации?

Данные — это набор необработанных, несистематизированных фактов (символы, числа, сигналы), не несущих смысловой нагрузки без контекста.

Информация — это обработанные, структурированные и интерпретированные данные, которые становятся осмысленными, полезными и позволяют принимать решения.

3) В чем отличие базы данных от банка данных и СУБД?

База данных (БД) – это упорядоченная совокупность структурированных данных, относящихся к определенной предметной области.

Банк данных – это расширенное понятие, включающее саму БД, аппаратное обеспечение, математическое и программное обеспечение (включая СУБД), персонал и пользователей.

СУБД (Система Управления Базами Данных) — это конкретное программное обеспечение, предназначенное для создания, ведения и использования БД.

4) Назовите основные компоненты банка данных и их назначение.

* Информационная база (БД): Хранение данных.
* Программные средства (СУБД, СУБД, приложения): Управление данными, доступом и их обработка.
* Технические средства (аппаратное обеспечение): Обеспечение функционирования всей системы.
* Лингвистические средства (языки, словари): Обеспечение интерфейса между пользователем и системой.
* Организационно-методические средства (документация, правила): Регламентация работы системы.
* Персонал (администраторы, разработчики, пользователи): Эксплуатация и использование системы.

5) Классифицируйте АИС по типу хранимых данных.

* Фактографические системы: Хранят структурированные данные в строго определенном формате (реквизиты: имя, дата, число).
* Документальные системы: Хранят неструктурированные или слабоструктурированные данные (документы, тексты, изображения, видео).
* Геоинформационные системы (ГИС): Хранят пространственные данные, привязанные к координатам.

6) Что понимается под трехуровневой архитектурой ANSI/SPARC?

Это концепция, разделяющая описание данных в БД на три уровня абстракции для обеспечения:

* Независимости данных от программ: Изменения на одном уровне не требуют изменений на других.
* Физической независимости: Изменение способа хранения данных не влияет на логические схемы и приложения.
* Логической независимости: Изменение логической структуры данных (напр., добавление таблицы) не влияет на внешние представления и приложения.

7) Дайте определения внешней схеме БД, концептуальной схеме БД, внутренней схеме БД.

*Внешняя схема* (уровень представления): Описание части БД с точки зрения конкретного пользователя или приложения.

*Концептуальная схема* (логический уровень): Полное, интегрированное описание всех данных БД без привязки к их физическому хранению.

*Внутренняя схема* (физический уровень): Описание способа физического хранения данных на носителе (структуры файлов, индексы, методы доступа).

8) Каковы особенности иерархической модели организации данных?

Иерархическая модель данных (ИМД) – это модель, в которой абстрактные понятия находятся в отношении предшествования таким образом, что каждому понятию соответствует только один предшественник (родитель).

Достоинство ИМД: данные, отражающие общие свойства совокупности конкретных данных не дублируются.

Недостаток ИМД связан с дублированием данных в случае их однотипности.

9) Каковы особенности сетевой модели организации данных?

Если в иерархических структурах запись-потомок должна иметь ровно одного предка, то в сетевых структурах данных потомок может иметь любое число предков. Связи между записями в сетевой МД выполняются в виде указателей, то есть каждая запись хранит ссылку на другую однотипную запись (или признак конца списка) и ссылки на списки подчиненных записей.

Достоинство сетевых МД: данные имеют четкую структуру.

Недостатком сетевых МД является большое количество дополнительной информации о связях.

10) Каковы особенности многомерной модели организации данных?

Структура: Данные организованы в виде гиперкубов (измерений и фактов).

Особенности: Предназначена для систем оперативной аналитической обработки (OLAP). Позволяет быстро агрегировать и анализировать большие объемы данных по различным измерениям (Время, Регион, Товар и т.д.).

11) Каковы особенности постреляционной модели организации данных?

Постреляционная модель данных в общем случае представляет собой расширенную реляционную модель, снимающую ограничение неделимости значений полей. То есть, допускаются многозначные поля, значения которых состоят из подзначений. Набор значений многозначных полей считается самостоятельной таблицей, встроенной в основную.

Достоинство постреляционной модели данных: возможность представления связанных реляционных таблиц одной постреляционной таблицей.

Недостаток постреляционной модели: сложность в обеспечении целостности данных.

12) Что относится к неструктурированным данным?

Данные, не имеющие предопределенной модели или структуры. Например: текстовые документы, электронные письма, видео-, аудиофайлы, изображения, сообщения в соцсетях, веб-страницы.

13) В чем преимущество использования колоночной СУБД по сравнению с реляционной?

Основное преимущество – эффективность при выполнении запросов агрегации и аналитики (OLAP) над большими объемами данных. Данные хранятся по колонкам, а не по строкам, что позволяет:

- быстро читать и агрегировать данные только по нужным колонкам.

- эффективно применять сжатие.

14) Каким образом осуществляется связь между таблицами в реляционной СУБД?

Связь осуществляется через первичные (Primary Key) и внешние ключи (Foreign Key). Внешний ключ в одной таблице ссылается на первичный ключ в другой таблице, обеспечивая целостность данных и логическую связь.

15) Каким образом на этапе проектирования решается проблема дублирующих записей в таблице?

Проблема решается путем нормализации — процесса устранения избыточности данных путем декомпозиции таблиц и вынесения повторяющихся данных в отдельные таблицы с последующим установлением связей между ними через ключи.

Список используемой литературы

1. ГОСТ 7.32-2017 Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления

2. Грас Д. 2021 Data Science Наука о данных с нуля: Пер. с англ., 2 е изд. перераб., 416 с., СПб БХВ Петербург

3. Коннолли, Т. Базы данных: проектирование, реализация и сопровождение. Теория и практика / Т. Коннолли, К. Бегг. — 3-е изд. — М. : Вильямс, 2003. — 1436 с. — ISBN 5-8459-0527-1.

4. Кузьмин, А. В. Введение в базы данных : учебное пособие / А. В. Кузьмин, И. А. Фомин. — СПб. : СПбГУАП, 2017. — 124 с. — ISBN 978-5-8088-1196-7.

5. Тарасов, С. В. Базы данных. Язык SQL : учебное пособие / С. В. Тарасов. — М. : СОЛОН-ПРЕСС, 2020. — 176 с. — ISBN 978-5-91359-365-4.