

Билет №5

Теоретическая часть:

1 вопрос:

Пользователи взаимодействуют с базами данных для решения различных прикладных задач, используя различные способы и инструменты:

1. Использование приложений:

Настольные приложения, веб-приложения, мобильные приложения.

2. Использование языков запросов:

(Structured Query Language), NoSQL query languages.

3. Использование инструментов бизнес-аналитики (BI):

BI-платформы, OLAP-кубы.

4. Использование API (Application Programming Interface): REST API.

В зависимости от уровня доступа и функциональности, пользователи баз данных могут иметь разные роли: администратор базы данных DBA, разработчик базы данных, аналитик данных, конечный пользователь.

2 вопрос:

База данных (БД) — это организованная коллекция структурированных данных, хранящихся электронно в компьютерной системе. Она предназначена для эффективного хранения, управления и доступа к информации.

Основные компоненты базы данных:

1. Данные: факты, значения и информация, структурированные данные.

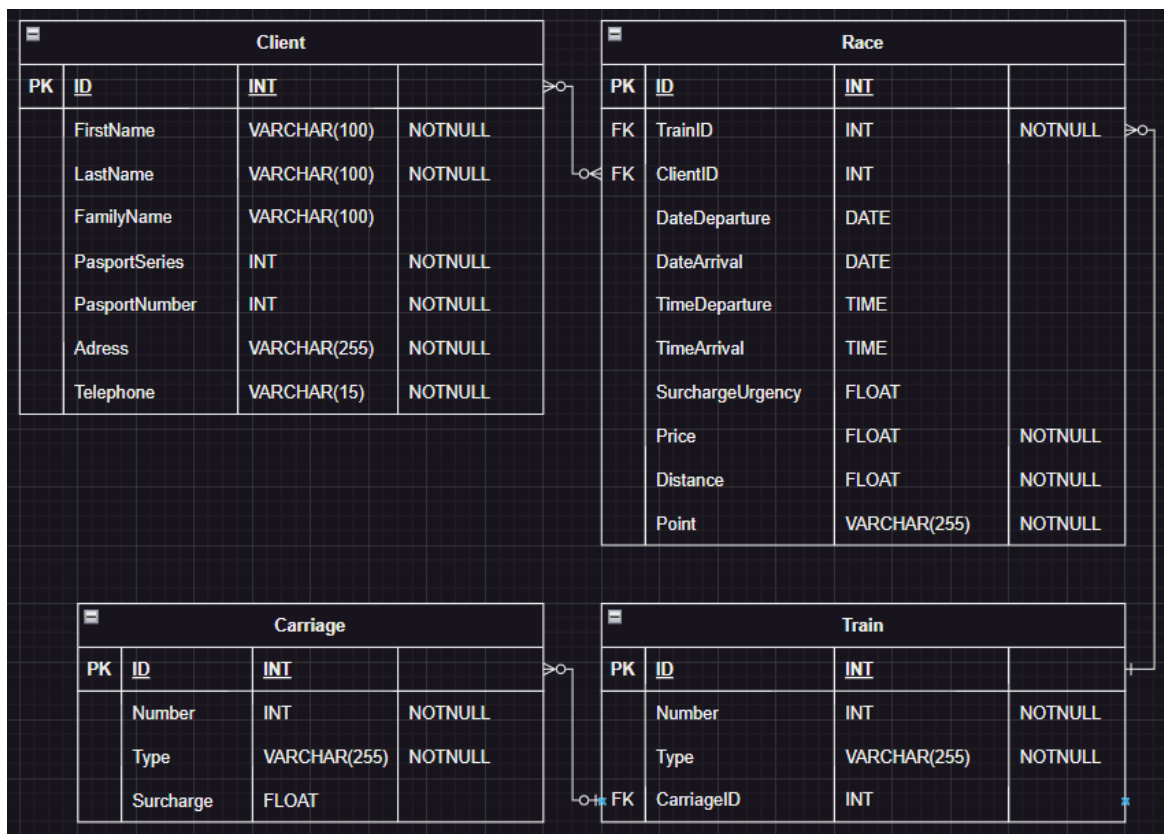
2. Схема данных: описание структуры данных, метаданные.

3. Система управления базами данных (СУБД): программное обеспечение, СУБД (MySQL, PostgreSQL, Oracle Database, Microsoft SQL Server, MongoDB).

4. Язык запросов (SQL - Structured Query Language, NoSQL query languages)

5. Приложения и инструменты: программы для работы с БД, инструменты бизнес-аналитики (BI), другие приложения.

Практическая часть:



Билет №6

Теоретическая часть:

1 вопрос:

База данных (БД) — это организованная коллекция структурированных данных, хранящихся электронно в компьютерной системе. Она предназначена для эффективного хранения, управления и доступа к информации.

Основные компоненты базы данных:

1. Данные: факты, значения и информация, структурированные данные.
2. Схема данных: описание структуры данных, метаданные.
3. Система управления базами данных (СУБД): программное обеспечение, СУБД (MySQL, PostgreSQL, Oracle Database, Microsoft SQL Server, MongoDB).
4. Язык запросов (SQL - Structured Query Language, NoSQL query languages)
5. Приложения и инструменты: программы для работы с БД, инструменты бизнес-аналитики (BI), другие приложения.

2 вопрос:

Дублирование данных (data duplication) - это наличие одинаковых или очень похожих данных в нескольких местах внутри одной или нескольких систем. Это может привести к различным проблемам, таким как несогласованность данных, увеличение затрат на хранение и обработку, а также сложности с обеспечением целостности данных.

Основные виды дублирования данных:

1. Полное дублирование (Exact duplicates) - абсолютно идентичные копии данных хранятся в разных местах.

Пример: одна и та же таблица клиентов хранится как в CRM-системе, так и в системе бухгалтерского учета.

2. Частичное дублирование (Partial duplicates) - данные похожи, но не идентичны, могут иметь небольшие отличия в формате, полноте или значениях.

Пример: информация о клиенте хранится в двух разных системах, но в одной системе указан только номер телефона, а в другой - полный адрес и email.

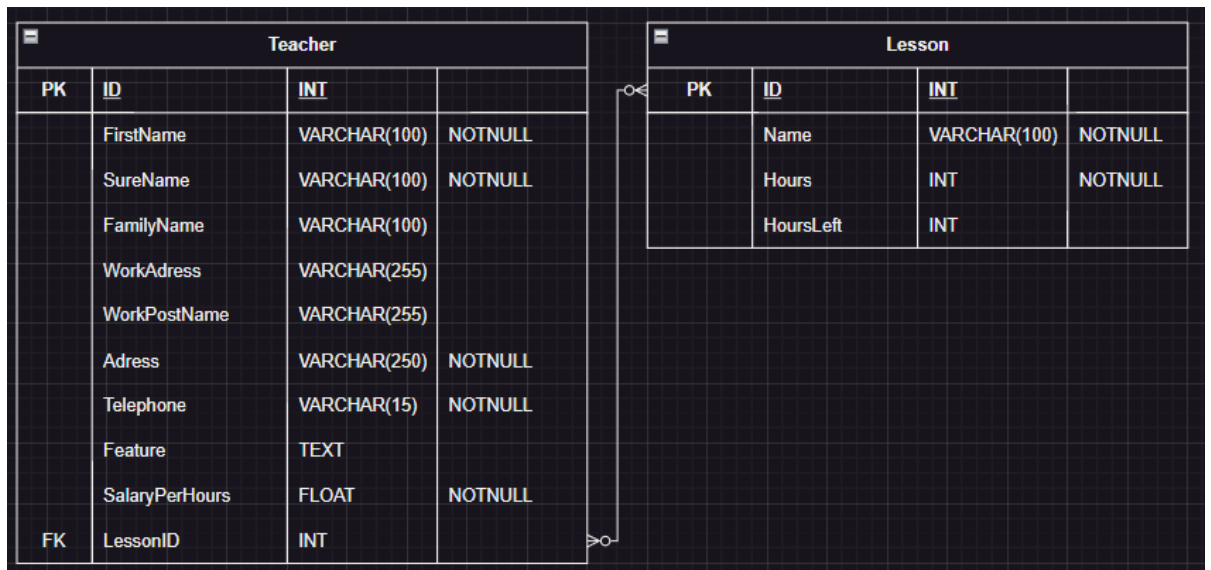
3. Структурное дублирование (Structural duplicates) - данные хранятся в разных структурах или форматах, но представляют собой одну и ту же информацию.

Пример: информация о продукте хранится в одной системе в виде таблицы, а в другой - в виде XML-файла.

4. Семантическое дублирование (Semantic duplicates) - данные кажутся разными, но на самом деле представляют одно и то же понятие или сущность.

Пример: в одной системе используется термин "клиент", а в другой - "покупатель" для обозначения одного и того же понятия.

Практическая часть:



Билет №7

Теоретическая часть:

1 вопрос:

1НФ (Первая нормальная форма):

- Убедитесь, что каждый столбец содержит атомарные значения (неделимые).
- Убедитесь, что в таблице нет повторяющихся групп столбцов.

2НФ (Вторая нормальная форма):

- Таблица должна соответствовать 1НФ.
- Все не ключевые атрибуты должны полностью зависеть от первичного ключа.
- Не должно быть частичной зависимости не ключевых атрибутов от части составного первичного ключа.

2 вопрос:

Основные операторы DDL:

- CREATE - используется для создания объектов базы данных, таких как таблицы, индексы, представления и т.д.

CREATE TABLE table_name (

column1 datatype,

column2 datatype,

column3 datatype,

....

);

- DROP - используется для удаления объектов базы данных.

DROP TABLE table;

- ALTER - используется для изменения структуры существующих объектов базы данных.

ALTER TABLE Customers

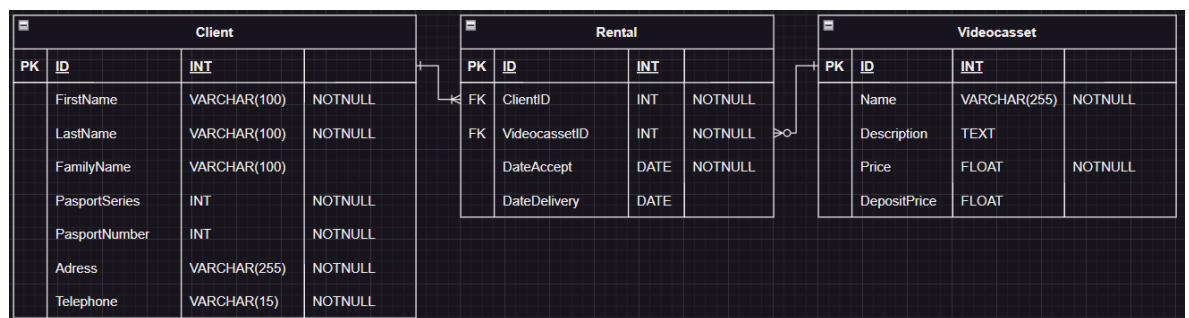
ADD Address NVARCHAR(50) NULL;

- TRUNCATE - используется для удаления всех данных из таблицы, но не удаляет саму таблицу.

TRUNCATE TABLE table_name;

- COMMENT - используется для добавления комментариев к объектам базы данных.

Практическая часть:



Билет №8

Теоретическая часть:

1 вопрос:

Запрос к базе данных (Database Query) - это команда или набор команд, используемых для взаимодействия с базой данных с целью извлечения, модификации или удаления информации. Запросы к базе данных обычно формулируются с использованием языка запросов, такого как SQL (Structured Query Language).

Основные виды запросов к базе данных:

1. Запросы на выборку (SELECT): используются для извлечения данных из одной или нескольких таблиц базы данных. Они могут включать фильтрацию, сортировку и агрегирование данных.
2. Запросы на обновление (UPDATE): предназначены для изменения существующих данных в таблице базы данных.

3. Запросы на вставку (INSERT): используются для добавления новых данных в таблицу базы данных.

4. Запросы на удаление (DELETE): предназначены для удаления существующих данных из таблицы базы данных.

Отличительные особенности запросов к базе данных:

1. Универсальность: запросы к базе данных могут обрабатывать данные из различных источников и в различных форматах.

2. Гибкость: запросы могут быть настроены для выполнения специфических задач и удовлетворения потребностей пользователей.

3. Производительность: запросы оптимизированы для работы с большими объемами данных и обеспечения быстрого доступа к информации.

Назначение запросов к базе данных:

1. Извлечение данных: запросы позволяют пользователям получать необходимую информацию из базы данных в соответствии с заданными критериями.

2. Управление данными: запросы используются для добавления, изменения и удаления данных в базе данных, что позволяет поддерживать актуальность и целостность информации.

3. Анализ данных: запросы могут включать агрегирование и статистические функции, что позволяет выполнять анализ данных и принимать обоснованные решения на основе этого анализа.

2 вопрос:

Система управления базами данных (СУБД) - это комплекс программных средств, который обеспечивает управление данными, хранимыми в базах данных, и взаимодействие с пользователями и приложениями, использующими эти данные. СУБД предоставляет интерфейс для создания, модификации, удаления и извлечения данных, а также для управления структурой базы данных и обеспечения целостности и безопасности данных.

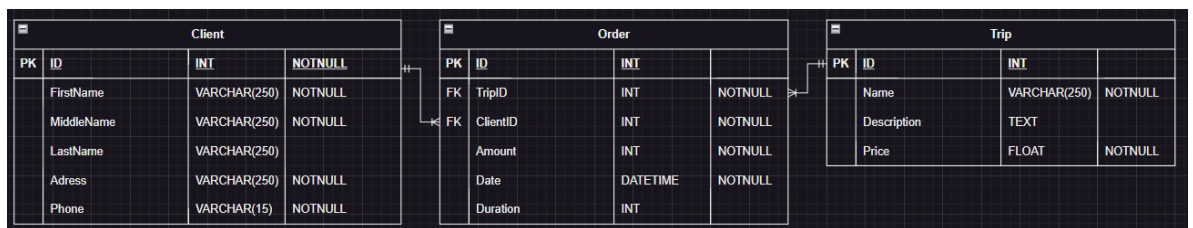
Назначение СУБД заключается в следующем: хранение данных, управление данными, обеспечение целостности данных, безопасность данных, взаимодействие с пользователями и приложениями.

Классификация СУБД может быть основана на различных критериях, таких как модель данных, способ хранения данных, архитектура и другие.

Неклассические СУБД:

1. NoSQL СУБД: Используют нереляционные модели данных, такие как документная, ключ-значение, графовая и широколокационная. Примеры: MongoDB, Cassandra, Redis, Neo4j.
2. Встраиваемые СУБД: Предназначены для использования в мобильных устройствах и встраиваемых системах. Примеры: SQLite, Berkeley DB.
3. Временные СУБД: Хранение данных с временной меткой, используются для анализа временных рядов. Примеры: InfluxDB, TimescaleDB.
4. Высоконагруженные СУБД: Специализированные системы, оптимизированные для работы с большими объемами данных и высокими нагрузками. Примеры: Google Bigtable, Amazon DynamoDB.
5. Многомодельные СУБД: Поддерживают несколько моделей данных в рамках одной системы. Примеры: OrientDB, ArangoDB.

Практическая часть:



Билет №9

Теоретическая часть:

1 вопрос:

Классические модели представления данных: Реляционная модель данных (Relational Model), Иерархическая модель данных (Hierarchical Model), Сетевая модель данных (Network Model), Семантическая модель данных (Semantic Model): Обеспечивает более высокий уровень абстракции для представления данных, включая концепции, отношения и ограничения. Примерами семантических моделей являются Entity-Relationship (ER) модель и Unified Modeling Language (UML).

Современные модели представления данных: Модель NoSQL (Not Only SQL), Документные базы данных (Document Stores), Графовые базы данных (Graph Databases), Квазиреляционные базы данных (Wide-column stores), Кэшные базы данных (Key-value stores), Модель данных NewSQL, Модель данных в памяти (In-memory databases), Модель данных на основе графов (Graph-based models), Модель данных на основе временных рядов (Time-series databases).

2 вопрос:

Реляционная модель данных - это фундаментальный принцип организации и представления данных в виде таблиц (отношений), которые состоят из строк (кортежей) и столбцов (атрибутов). В основе реляционной модели лежит математическое понятие отношения, что обеспечивает строгую структуру данных и позволяет эффективно обрабатывать их с помощью языка SQL.

Основные свойства реляционной модели данных:

1. Данные организованы в виде таблиц (отношений).
2. Каждая таблица имеет уникальное имя и состоит из строк (кортежей) и столбцов (атрибутов).
3. Каждый столбец имеет уникальное имя и определенный тип данных.
4. Строки в таблице уникальны и не упорядочены.
5. Столбцы в таблице упорядочены и имеют фиксированное количество.
6. Взаимосвязь между таблицами устанавливается с помощью общих атрибутов (ключей).
7. Постреляционная модель данных (Post-Relational Model) - это направление в развитии систем управления базами данных, которое появилось в ответ на ограничения реляционной модели. Постреляционные модели данных стремятся объединить преимущества реляционной модели с другими моделями данных, такими как объектно-ориентированная, сетевая и иерархическая.

Отличия реляционной и постреляционной модели данных:

1. Постреляционные модели данных могут поддерживать несколько способов хранения данных, включая реляционные таблицы, объекты, массивы и другие структуры.
2. В постреляционных моделях данных могут использоваться разные языки запросов и программирования, а не только SQL.
3. Постреляционные модели данных могут предоставлять более гибкие и сложные механизмы взаимодействия между данными, включая наследование, агрегацию и композицию объектов.
4. Постреляционные модели данных могут быть более эффективными при работе с большими объемами данных и сложными запросами, так как они могут использовать иные алгоритмы и структуры данных.

Практическая часть:

