**Московский государственный технический университет им.Н.Э.Баумана**

**ЗКЗАМЕНАЦИОННЫЙ БИЛЕТ № 5**

**По курсу \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_«Базы данных»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**/название дисциплины/**

1. Язык структурированных запросов SQL. Функции SQL. Пример.

SQL (Structured Query Language) - это язык программирования, используемый для взаимодействия с реляционными базами данных. SQL предоставляет набор команд и операторов для создания, изменения и извлечения данных из базы данных.

1. Команда SELECT:

- Используется для извлечения данных из базы данных.

- Пример: `SELECT \* FROM table\_name;` (извлекает все строки и столбцы из таблицы "table\_name").

2. Команда INSERT:

- Используется для добавления новых строк в таблицу.

- Пример: `INSERT INTO table\_name (column1, column2) VALUES (value1, value2);` (добавляет новую строку с указанными значениями в столбцы "column1" и "column2").

3. Команда UPDATE:

- Используется для обновления существующих строк в таблице.

- Пример: `UPDATE table\_name SET column1 = value1 WHERE condition;` (обновляет значение "column1" на "value1" для строк, удовлетворяющих условию).

4. Команда DELETE:

- Используется для удаления строк из таблицы.

- Пример: `DELETE FROM table\_name WHERE condition;` (удаляет строки, удовлетворяющие условию).

5. Команда CREATE:

- Используется для создания новой таблицы, базы данных или других объектов.

- Пример: `CREATE TABLE table\_name (column1 data\_type, column2 data\_type);` (создает новую таблицу с указанными столбцами и их типами данных).

6. Команда ALTER:

- Используется для изменения структуры существующей таблицы.

- Пример: `ALTER TABLE table\_name ADD column\_name data\_type;` (добавляет новый столбец с указанным именем и типом данных в таблицу).

7. Команда DROP:

- Используется для удаления таблицы, базы данных или других объектов.

- Пример: `DROP TABLE table\_name;` (удаляет таблицу "table\_name").

SQL предлагает различные функции, которые позволяют выполнять операции с данными в базе данных.

1. Агрегатные функции:

- SUM(): Вычисляет сумму значений в столбце.

- AVG(): Вычисляет среднее значение в столбце.

- COUNT(): Подсчитывает количество строк или значений в столбце.

- MAX(): Возвращает максимальное значение в столбце.

- MIN(): Возвращает минимальное значение в столбце.

2. Функции строки:

- CONCAT(): Соединяет две или более строк в одну.

- SUBSTRING(): Извлекает подстроку из строки.

- UPPER(): Преобразует строку в верхний регистр.

- LOWER(): Преобразует строку в нижний регистр.

- LENGTH(): Возвращает длину строки.

3. Функции даты и времени:

- NOW(): Возвращает текущую дату и время.

- DATE(): Извлекает дату из значения даты и времени.

- EXTRACT(): Извлекает часть даты или времени, такую как год, месяц или день.

- DATE\_FORMAT(): Форматирует значение даты и времени в определенном формате.

4. Функции условий:

- IF(): Возвращает одно значение, если условие истинно, и другое значение, если условие ложно.

- CASE(): Выполняет условное выражение и возвращает результат в зависимости от соответствия условиям.

5. Математические функции:

- ABS(): Возвращает абсолютное значение числа.

- ROUND(): Округляет число до заданного количества десятичных знаков.

- SQRT(): Возвращает квадратный корень числа.

- POWER(): Возвращает число, возведенное в заданную степень.

1. Разработка банка данных – программа и методика испытаний. Пример.

Разработка банка данных включает не только создание структуры базы данных, но и тестирование ее функциональности и производительности. Вот пример программы и методики испытаний для разработки банка данных:

Программа испытаний:

1. Подготовительная фаза:

- Определение целей и требований к базе данных.

- Разработка схемы базы данных и ее структуры.

- Создание тестовых данных и наборов данных для испытаний.

- Подготовка среды испытаний (установка базы данных, необходимых программ и инструментов).

2. Функциональное тестирование:

- Проверка создания таблиц, столбцов и ограничений целостности.

- Испытание операций вставки, обновления и удаления данных.

- Проверка запросов SQL и правильности результатов.

- Тестирование хранимых процедур, триггеров и функций.

- Проверка обработки ошибок и исключений.

3. Тестирование производительности:

- Измерение времени выполнения запросов различной сложности.

- Оценка времени отклика системы при множестве одновременных запросов.

- Тестирование производительности при работе с большим объемом данных.

- Определение пропускной способности и максимальной нагрузки системы.

4. Испытания безопасности:

- Проверка доступа к базе данных и разграничения прав доступа.

- Тестирование защиты от внешних атак, таких как SQL-инъекции.

- Проверка шифрования данных и обеспечения конфиденциальности.

- Анализ уязвимостей и устранение потенциальных угроз безопасности.

5. Тестирование масштабируемости и отказоустойчивости:

- Испытание работы базы данных при росте объема данных.

- Проверка распределенной архитектуры и репликации данных.

- Тестирование восстановления после сбоев и отказоустойчивости.

- Оценка производительности и доступности системы при нагрузке.

Методика испытаний может быть разработана в соответствии с требованиями и особенностями конкретного проекта. Она включает в себя определение тестовых случаев, шагов испытаний, ожидаемых результатов и критериев прохождения испытаний. Примером может быть использование инструментов автоматизации тестиров

ания, написание скриптов для проверки функций базы данных, использование мониторинга производительности и инструментов для анализа безопасности данных.

Билет рассмотрен и утвержден на заседании кафедры ИУ-5 протокол №9 от 27.04.2023

/№ протокола, дата/

Зав.каф. ИУ-5 В.И. Терехов