Министерство высшего образования и науки РФ

Санкт–Петербургский политехнический университет Петра Великого

Институт компьютерных наук и кибербезопасности

Высшая школа компьютерных технологий и информационных систем

**Отчет**

по дисциплине «Базы данных»

**Выполнил:**

студент группы 5130902/20201 \_\_\_\_\_\_\_\_\_\_\_\_\_ Леванов В.С.

подпись

**Проверил:**

доцент, к.т.н \_\_\_\_\_\_\_\_\_\_\_\_ Нестеров С.А.

подпись

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_ 2025г.

Санкт-Петербург

2025

Оглавление

**Элементы оглавления не найдены.**

**Urait**

Test 7:

Изображение выглядит как текст, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.  
Test 8: Изображение выглядит как текст, снимок экрана, дизайн

Контент, сгенерированный ИИ, может содержать ошибки.

**Конспект по теме SQL: Сортировка, Индексы, Объединение, Группировка**

**1. Сортировка данных (ORDER BY)**

* **По умолчанию** результат SELECT не упорядочен.
* Сортировку можно задать с помощью ORDER BY.
* Формы задания:
  + Имя столбца
  + Псевдоним столбца (AS)
  + Порядковый номер столбца
* Можно комбинировать разные формы в одном ORDER BY.

**Дополнительные опции:**

* ASC (по возрастанию, значение по умолчанию)
* DESC (по убыванию)
* NULLS FIRST / NULLS LAST (расположение NULL)

Изображение выглядит как текст, программное обеспечение, Мультимедийное программное обеспечение, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

**2. Индексы**

**Что такое индекс?**

Индекс — структура данных в БД для быстрого поиска строк.

**Преимущества:**

* Быстрый поиск
* Оптимизация выполнения запросов
* Быстрая сортировка/группировка
* Поддержка ограничений (PRIMARY KEY, UNIQUE)

**Виды индексов:**

* **B-tree (B-дерево):** для диапазонных запросов.
* **Hash-индекс:** быстрый поиск по точному совпадению.
* **Bitmap-индекс:** для полей с малым числом уникальных значений.
* **Полнотекстовые индексы, Пространственные индексы**.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

**Недостатки индексов:**

* Дополнительное место на диске
* Ухудшение производительности при вставке/обновлении/удалении
* Не всегда используются в запросах

**3. Объединение данных (UNION)**

* Объединяет несколько запросов, увеличивая количество строк (столбцов — нет).
* Требуется одинаковое количество столбцов в объединяемых запросах.
* Столбцы могут иметь разные типы данных.
* По умолчанию убираются дубликаты (UNION DISTINCT).
* Для сохранения всех строк — UNION ALL.
* Сортировка возможна только по позициям столбцов.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

**4. Группировка (GROUP BY, HAVING)**

**GROUP BY**

* Объединяет строки с одинаковыми значениями в группы.
* Используется с агрегатными функциями: COUNT(), AVG(), SUM(), MIN(), MAX().

**Правила:**

* Все неагрегированные поля должны быть указаны в GROUP BY.
* После WHERE фильтрации.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

**HAVING**

* Фильтрует сгруппированные данные (аналог WHERE, но для агрегированных данных).
* Может использовать агрегатные функции.

Изображение выглядит как текст, программное обеспечение, Мультимедийное программное обеспечение, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

**Ответы на контрольные вопросы**

**Раздел: Сортировка**

1. **Надо ли использовать упорядочивание, если мы добавляли записи в правильном порядке?**
   * Да, нужно. Без ORDER BY СУБД **не гарантирует порядок** выборки, даже если данные вставлены в нужной последовательности.
2. **Что произойдет, если указать несколько полей для сортировки?**
   * Результат будет сначала отсортирован по первому полю, затем по второму внутри групп одинаковых значений первого поля, и так далее.
3. **Можно ли упорядочивать по одному полю по возрастанию, а по другому — по убыванию?**
   * Да, можно. Для каждого столбца можно независимо указать ASC или DESC.
4. **Куда будут помещаться значения NULL при сортировке?**
   * По умолчанию — **NULLS FIRST** (в начало). Можно изменить на NULLS LAST.
5. **Что можно указывать в качестве элемента сортировки?**
   * Имя столбца, его псевдоним, порядковый номер в списке выборки или выражение.

**Раздел: Индексы**

1. **Для чего применяются индексы?**
   * Для ускорения поиска, сортировки, группировки и обеспечения уникальности данных.
2. **Какие структуры данных лежат в основе индексов?**
   * Обычно это **B-деревья** (B-tree) или **хэш-таблицы**.
3. **Назовите известные типы индексов.**
   * B-tree индексы, Hash-индексы, Bitmap-индексы, Полнотекстовые индексы, Пространственные индексы.
4. **Какие команды используются для создания индексов?**
   * CREATE INDEX, CREATE UNIQUE INDEX, CREATE INDEX COMPUTED BY.
5. **Как можно изменять уже созданный индекс?**
   * С помощью ALTER INDEX имя {ACTIVE | INACTIVE} (активация/деактивация).
6. **Назовите преимущества использования индексов.**
   * Быстрый поиск данных, оптимизация выполнения запросов, поддержка ограничений целостности.
7. **Назовите недостатки применения индексов.**
   * Дополнительное место на диске, замедление операций вставки/обновления/удаления, не всегда применяются в запросах.

**Раздел: Объединение (UNION)**

1. **Что будет, если объединяемые запросы будут содержать разное количество столбцов?**
   * Произойдёт ошибка: **число столбцов должно быть одинаковым**.
2. **Что будет, если соответствующие столбцы объединяемых запросов будут иметь разные названия?**
   * Названия столбцов берутся из первого запроса. Разные названия **не проблема**, важен только порядок.
3. **Что будет, если соответствующие столбцы объединяемых запросов будут иметь разные типы данных?**
   * Допустимо, но могут быть приведены к совместимому типу автоматически или возникнет ошибка приведения типов.
4. **Можно ли упорядочить объединение нескольких запросов и как?**
   * Да, но в ORDER BY можно указывать **только позиции столбцов** (например, ORDER BY 1, 2).
5. **Как управлять подавлением дубликатов в результате объединения и что является поведением по умолчанию?**
   * По умолчанию дубликаты подавляются (UNION = UNION DISTINCT). Чтобы сохранить все строки — использовать UNION ALL.

**Раздел: Группировка (GROUP BY и HAVING)**

1. **Опишите алгоритм работы группировки записей.**
   * После выполнения WHERE данные разбиваются на группы по указанным полям, затем к каждой группе применяются агрегатные функции (SUM, AVG, COUNT и т.д.).
2. **Какие ограничения накладывает группировка записей на список выбора?**
   * В списке SELECT могут быть только агрегатные функции и те поля, которые указаны в GROUP BY.
3. **Что допускается использовать в списке выбора при группировке записей?**
   * Агрегатные функции и столбцы, перечисленные в GROUP BY.
4. **Чем определяется количество записей, возвращаемых запросами с группировкой?**
   * Количеством уникальных комбинаций значений полей, указанных в GROUP BY.
5. **Для чего используется предложение HAVING?**
   * Для фильтрации сгруппированных данных (например, выбрать только те группы, где сумма больше 100).
6. **Чем HAVING отличается от WHERE?**
   * WHERE фильтрует строки **до** группировки, а HAVING — **после** группировки.
7. **В каких случаях рекомендуется использовать WHERE, а не HAVING?**
   * Если фильтрация касается **отдельных строк** до агрегирования, лучше использовать WHERE для повышения производительности.