Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и кибербезопасности

Высшая школа «Компьютерных технологий и информационных систем»

**ОТЧЕТ по лабораторной работе №**13

по дисциплине «Базы данных»

**Выполнил:**

студент группы 5130902/20201 \_\_\_\_\_\_\_\_\_\_\_\_\_ А. И. Сафонов

подпись

**Проверил:**

Кандидат тех. наук, доцент \_\_\_\_\_\_\_\_\_\_\_\_ С. А. Нестеров

подпись

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

Санкт-Петербург, 2025

# ЛАБОРАТОРНАЯ РАБОТА № 1 (SQL SERVER) ТРИНАДЦАТОЙ НЕДЕЛИ КУРСА «УПРАВЛЕНИЕ ДАННЫМИ». - делаете на своем сервере или на сервере в классе

# ЛАБОРАТОРНАЯ РАБОТА № 2 (SQL SERVER) ТРИНАДЦАТОЙ НЕДЕЛИ КУРСА «УПРАВЛЕНИЕ ДАННЫМИ»

# ЛАБОРАТОРНАЯ РАБОТА 13.1 (ORACLE). ЗНАКОМСТВО С ПРЕДСТАВЛЕНИЯМИ СЛОВАРЯ ДАННЫХ

# ЛАБОРАТОРНАЯ РАБОТА 13.2 (ORACLE). РОЛИ И ПРИВИЛЕГИИ. УПРАВЛЕНИЕ ДОСТУПОМ К ОБЪЕКТАМ СХЕМЫ

# ЛАБОРАТОРНАЯ РАБОТА № 1 (SQL SERVER) ЧЕТЫРНАДЦАТОЙ НЕДЕЛИ КУРСА «УПРАВЛЕНИЕ ДАННЫМИ»

1) обновление двух или более записей одним оператором update, так чтобы одно из изменений не могло быть произведено (транзакция должна быть целиком откачена);

**Изображение выглядит как текст, число, линия, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.**

2) оформите те же по сути изменения в виде отдельных операторов, проверьте их работу;

Изображение выглядит как текст, электроника, снимок экрана, дисплей

Контент, сгенерированный ИИ, может содержать ошибки.

3) измените данные так, чтобы они были в первоначальном стоянии (в графическом интерфейсе или написав соответствующий код); оформите последовательность из одного оператора update и одного оператора insert в виде именованной явной транзакции, по-отдельности эти операторы должны выполняться успешно; транзакцию завершите откатом, проверьте результат;

Изображение выглядит как текст, снимок экрана, Шрифт, дисплей

Контент, сгенерированный ИИ, может содержать ошибки.

4) между операторами update и insert кода транзакции, созданной на предыдущем шаге, поставьте оператор select, выбирающий все записи из соответствующей таблицы; после отката транзакции выполните аналогичный select; убедитесь, что внутренний select «видит» производимые изменения, т.е. внутри транзакции эти данные доступны;

Изображение выглядит как текст, снимок экрана, число, линия

Контент, сгенерированный ИИ, может содержать ошибки.

5) дополните транзакцию точкой сохранения установленной после внутреннего select; выполните откат до точки сохранения (в ROLLBACK TRAN указываете имя точки сохранения, а не транзакции), ваш код должен выглядеть примерно так:

Изображение выглядит как текст, снимок экрана, число, линия

Контент, сгенерированный ИИ, может содержать ошибки.

6) обратите внимание, что на момент окончания задания 5 у вас транзакция не завершена; откройте новое окно и выполните в нем следующий код:

use MyLib1; --здесь имя вашей базы GO SELECT \* FROM BookInLib;

Запрос должен «зависнуть», ожидая пока запущенная в п. 5) транзакция не снимет с записей блокировки. Это связано с тем, что используемый по умолчанию уровень изоляции - READ COMMITTED – не дает произвести «грязное чтение», т.е. чтение параллельно выполняющейся задачей данных, измененных еще не завершенной транзакцией. Завершите запущенную в п. 5) транзакцию откатом, убедитесь, что запущенный в другом окне select прочитал данные из таблицы, при этом данные соответствуют состоянию после отката транзакции;

A screenshot of a computer

AI-generated content may be incorrect.Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Контент, сгенерированный ИИ, может содержать ошибки.Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Контент, сгенерированный ИИ, может содержать ошибки.

7) выполните действия, аналогичные предыдущему примеру, также откатив транзакцию к точке сохранения, но пока не завершая ее; в параллельном окне выполните запрос

SELECT \* FROM BookInLib with (NOLOCK);

Указание NOLOCK разрешает чтение результатов незафиксированных транзакций. Таким образом, мы не ожидали окончания выполнения параллельной транзакции (запрос отработал сразу), но получили «грязное чтение» - был считан результат, незафиксированной транзакции, которую мы планировали отменить, и вы должны увидеть изменения, выполненные оператором UPDATE.

Изображение выглядит как текст, электроника, снимок экрана, дисплей

Контент, сгенерированный ИИ, может содержать ошибки.Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Контент, сгенерированный ИИ, может содержать ошибки.

# ЛАБОРАТОРНАЯ РАБОТА (Oracle) ЧЕТЫРНАДЦАТОЙ НЕДЕЛИ КУРСА «УПРАВЛЕНИЕ ДАННЫМИ». ОСНОВЫ УПРАВЛЕНИЯ ТРАНЗАКЦИЯМИ

# 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.

# Нестеров Базы данных

Test 7:

A screenshot of a computer

AI-generated content may be incorrect.

Test 8:

A screenshot of a computer

AI-generated content may be incorrect.