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

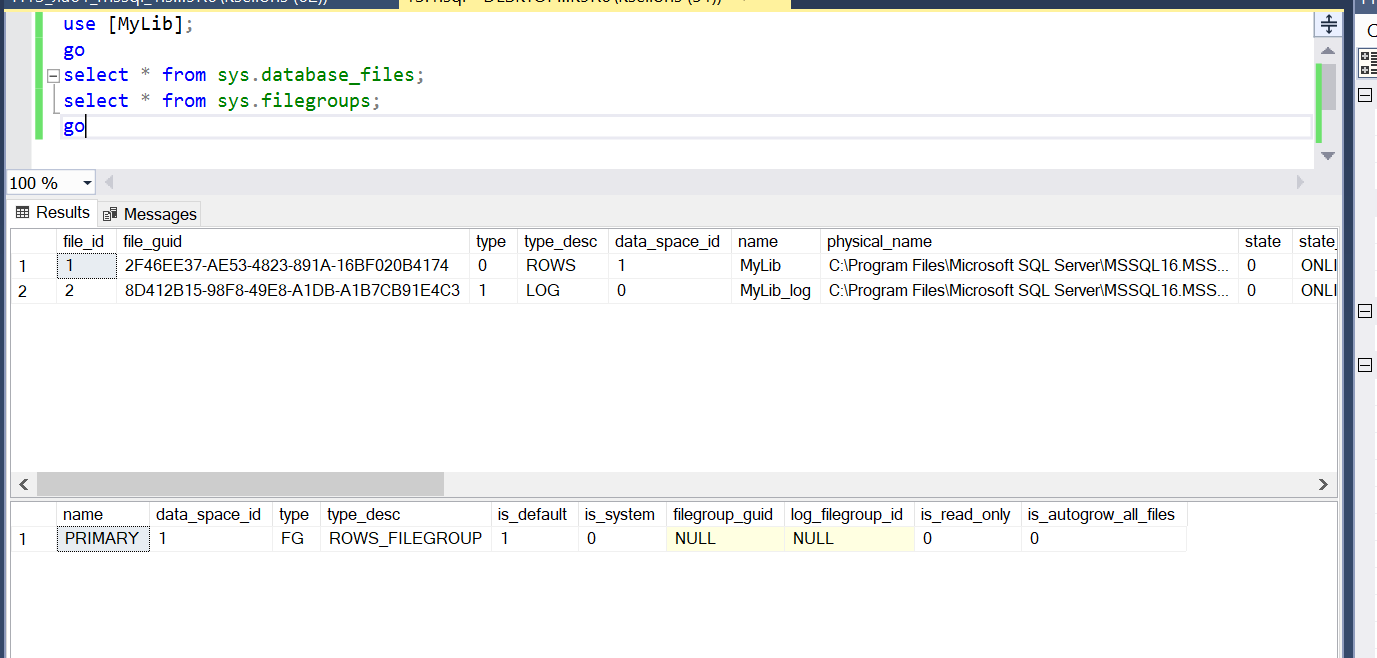
подпись

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

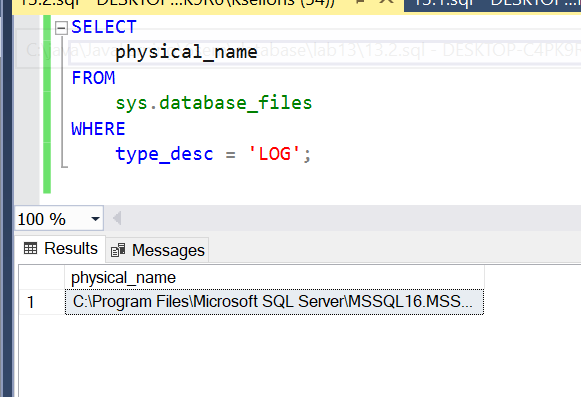
Санкт-Петербург, 2025

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

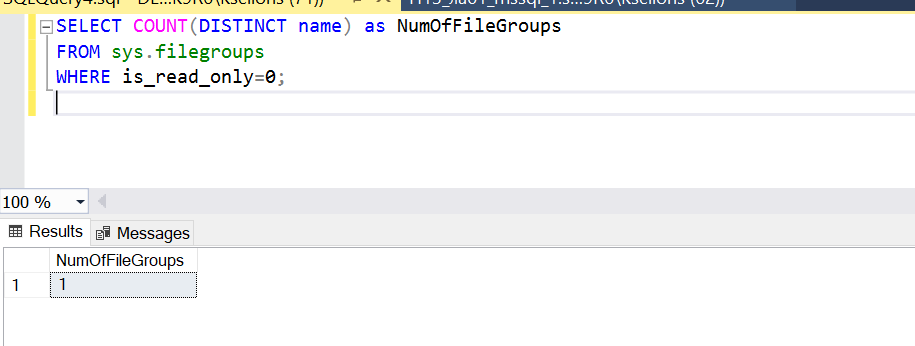
Выполните приведенный ниже код



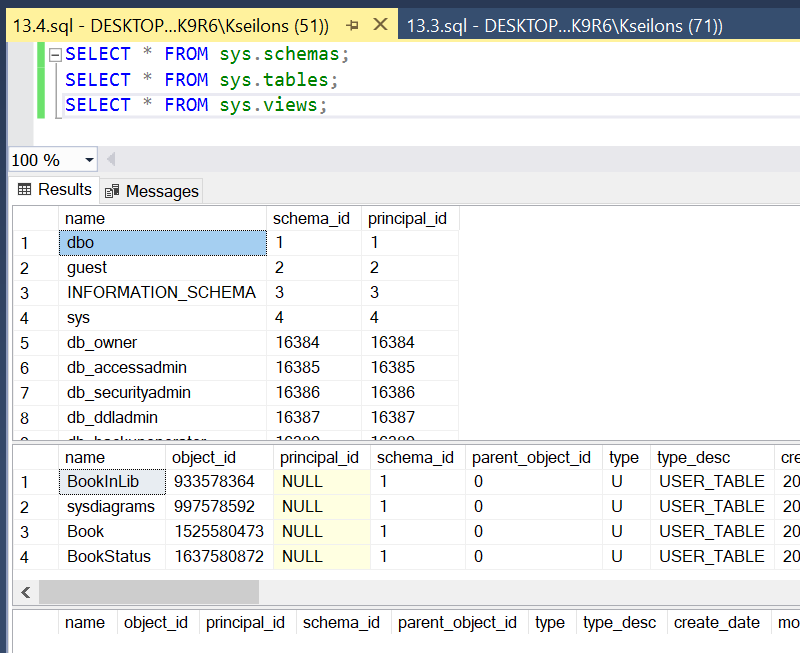
Задание. Напишите запрос, выводящий расположение файла (файлов) журнала на диске. Чтобы отобрать нужные записи, используйте условие на столбец type или type\_desc.



Задание. Напишите запрос, подсчитывающий количество файловых групп вашей базы данных, доступных на чтение и запись.

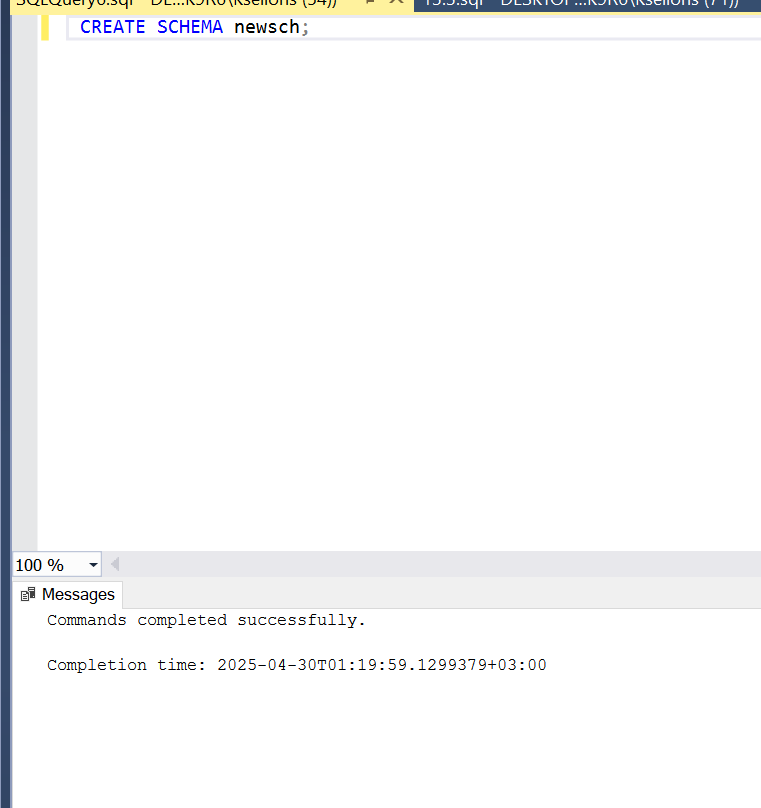


Используя системные представления sys.schemas, sys.tables, sys.views, получите сведения о схемах, таблицах и представлениях, определенных в вашей базе данных.

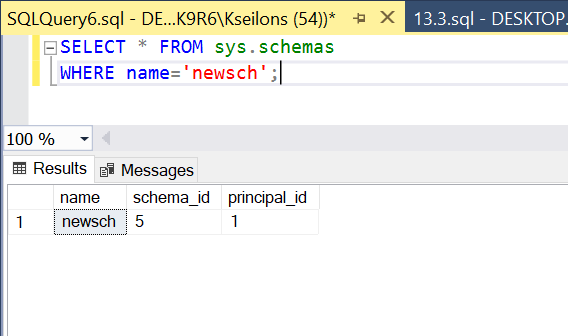


С помощью приведенного ниже кода создайте новую схему.

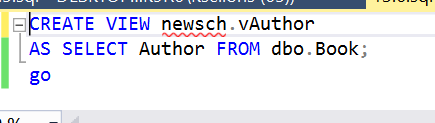
CREATE SCHEMA newsch;

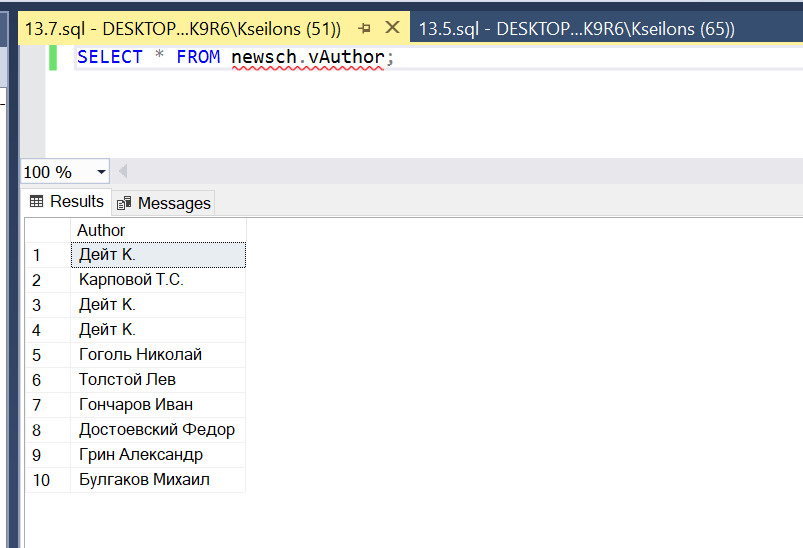


Проверьте, что данные о созданной схеме отображаются в представлении sys.schemas.

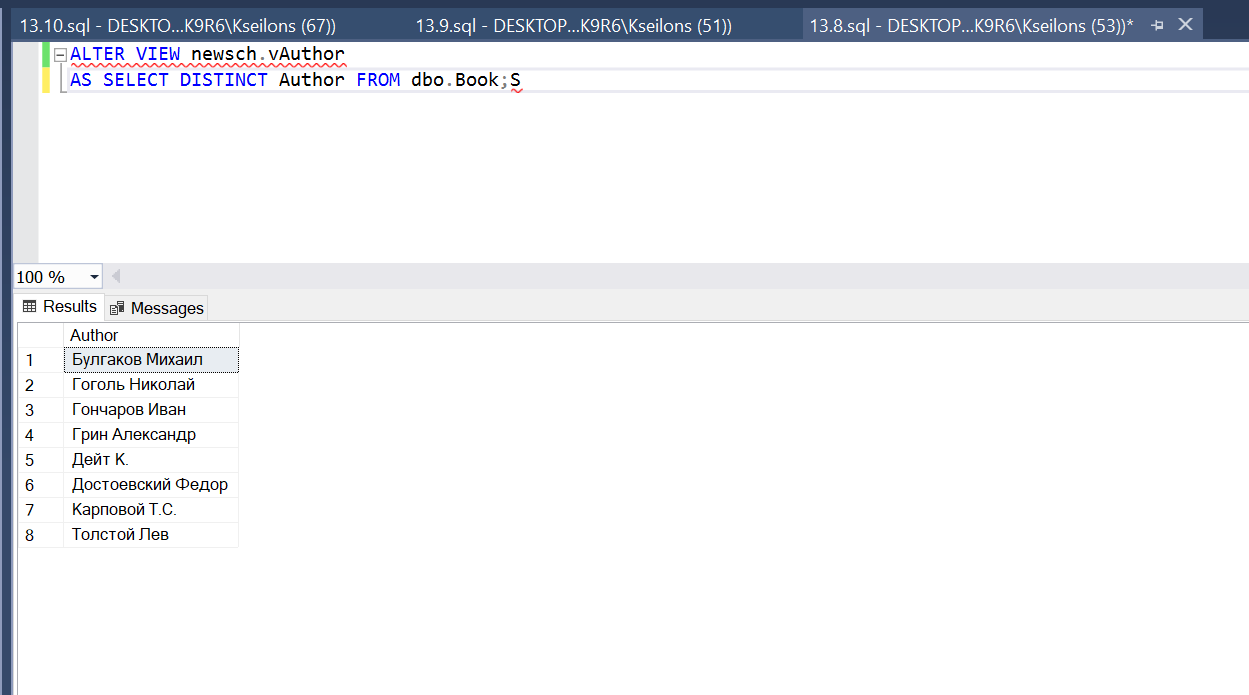


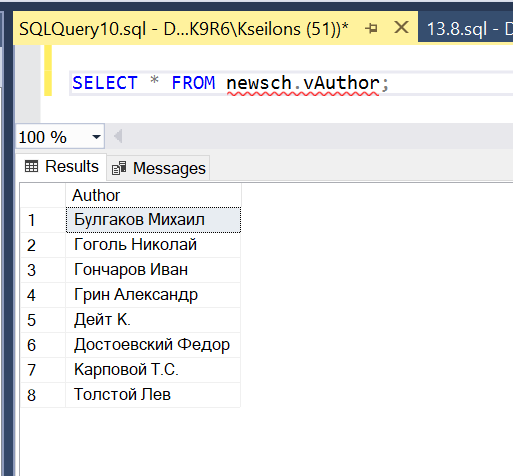
Задание. В схеме newsch создайте представление vAuthor, содержащее один столбец, в котором перечислены все авторы, упомянутые в таблице Book. Выберите данные из представления (не забудьте, что newsсh не является схемой по умолчанию, поэтому надо указывать имя представления вместе с схемой).

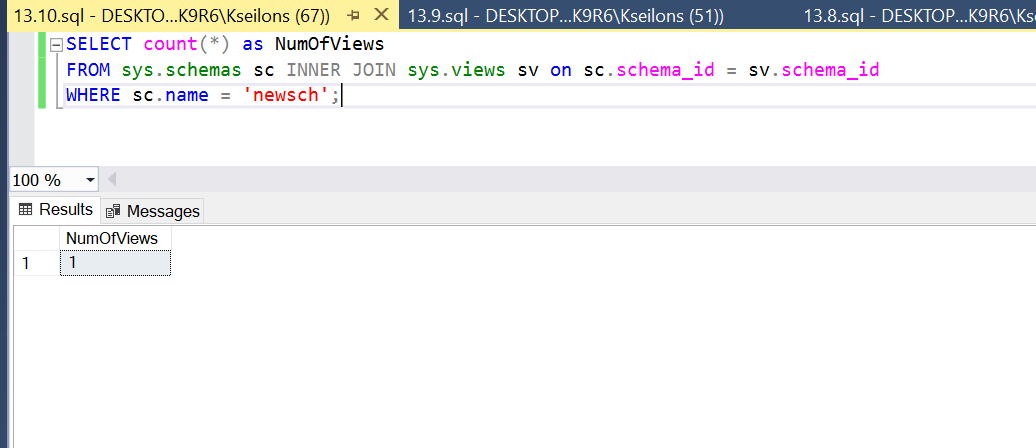




Задание. Измените определенное ранее представление таким образом, чтобы список авторов выводился без повторений. Снова выберите данные из представления. Задание. Используя рассмотренные ранее системные представления, посчитайте количество представлений в схеме newsch.







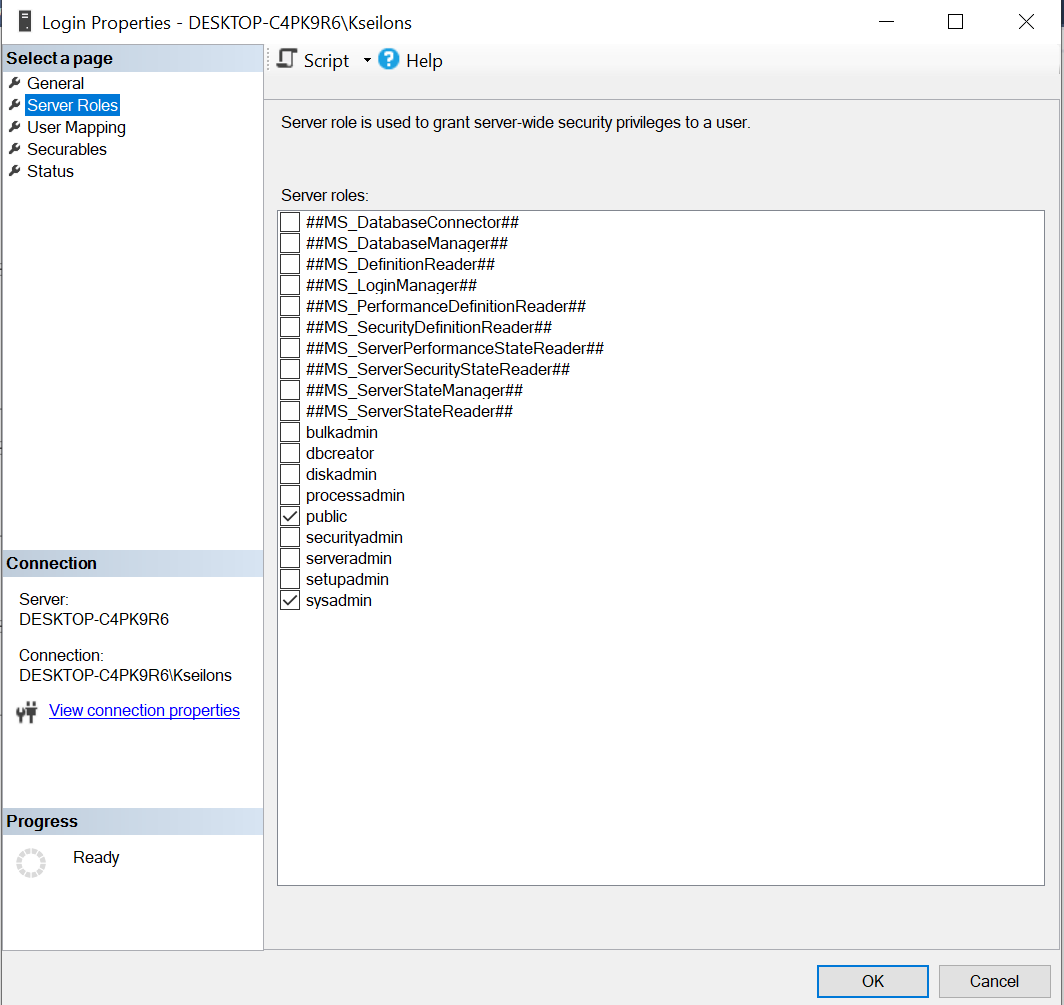
# ЛАБОРАТОРНАЯ РАБОТА № 2 (SQL SERVER) ТРИНАДЦАТОЙ НЕДЕЛИ КУРСА «УПРАВЛЕНИЕ ДАННЫМИ»

С помощью SQL Server Management Studio подключитесь к используемому экземпляру SQL Server. Проверьте установленный на сервере режим аутентификации

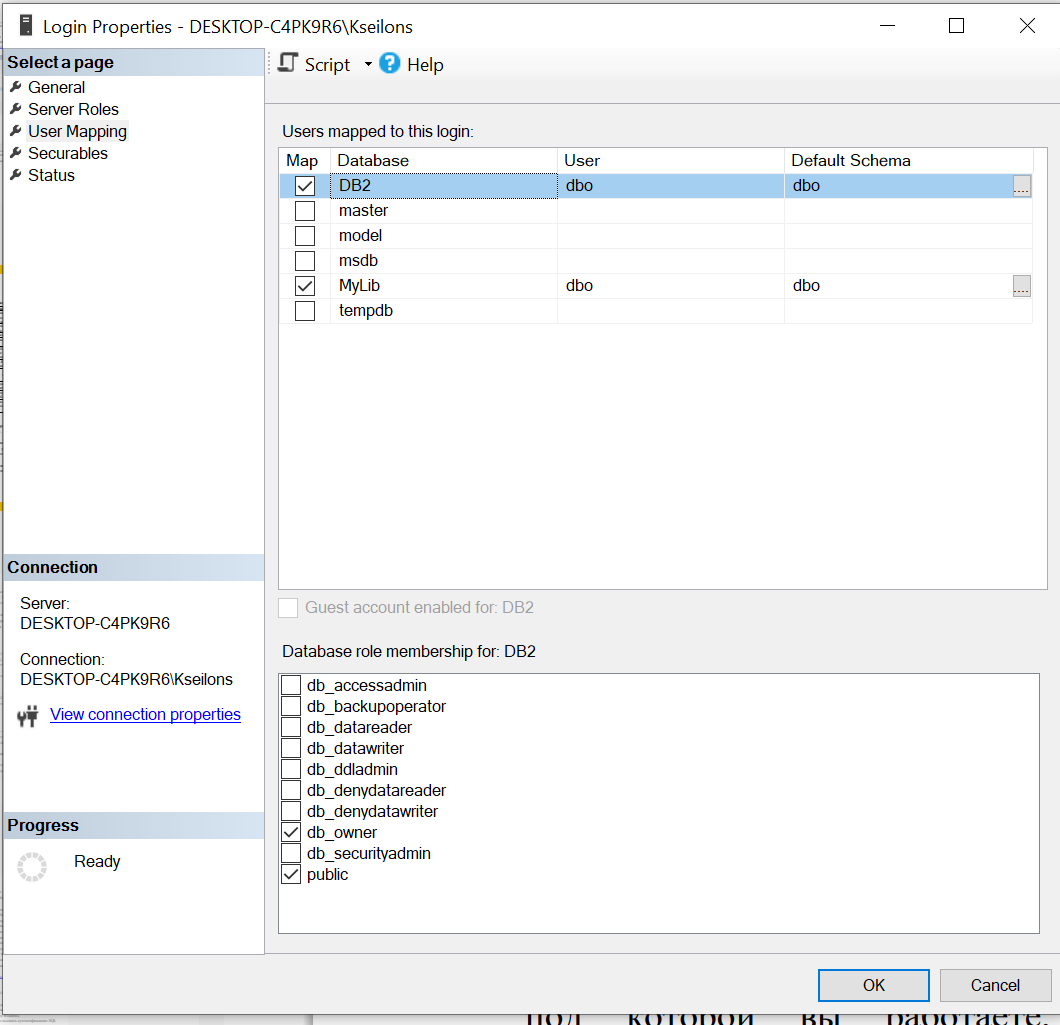
Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Контент, сгенерированный ИИ, может содержать ошибки.

В окне Object Explorer (по умолчанию – левая часть окна Management Studio) откройте список учетных записей (logins). На выполнение каких серверных ролей авторизована используемая вами учетная запись?



В каких базах данных сервера вашей учетной записи сопоставлены пользователи? На выполнение каких ролей они авторизованы?



В среде Management Studio создайте новую базу данных. Откройте список пользователей и ролей. Убедитесь, что учетная запись, под которой вы работаете, сопоставлена пользователю dbo, авторизованному на роль db\_owner

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.



Используя приведенный ниже скрипт, создайте в базе данных таблицы. Перед тем как запустить скрипт, уберите комментарий с первой строки и после ключевого слова use укажите имя вашей базы данных.

use newbase1

GO

CREATE TABLE dbo.Book (

book\_id int IDENTITY (1, 1) primary key,

Title varchar(50) NOT NULL,

Author varchar(50),

Publisher varchar(50),

[Year] smallint)

GO

CREATE TABLE dbo.Status (

Status\_id int IDENTITY (1, 1) primary key,

Status\_name varchar(50) NOT NULL )

GO

CREATE SCHEMA libr

GO CREATE TABLE libr.Book\_in\_lib (

lib\_id int primary key ,

book\_id int references dbo.Book,

status\_id int references dbo.[Status])

GO

Изображение выглядит как текст, снимок экрана, Шрифт, дисплей

Контент, сгенерированный ИИ, может содержать ошибки.

Создайте учётную запись Windows и учётную запись SQL Server для нее (в Management Studio: «Безопасность -> Имена входа» из контекстного меню «Создать имя входа»).

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

Контент, сгенерированный ИИ, может содержать ошибки.

Создайте пользователя в вашей базе данных (в примерах ниже имя пользователя – «ns»), в качестве схемы по умолчанию выберите dbo. В Management Studio это можно сделать из графического интерфейса (контекстное меню узла Security для выбранной БД, там New…-> User) или выполнив оператор CREATE USER. Например (если схема не указана, подразумевается dbo)

Изображение выглядит как текст, снимок экрана, Шрифт, число

Контент, сгенерированный ИИ, может содержать ошибки.

1. Добавьте этого пользователя в роль db\_datareader. Это можно сделать или через графический интерфейс, или с помощью системной хранимой процедуры sp\_addrolemember, первым параметром которой будет имя роли, а вторым – имя пользователя.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Контент, сгенерированный ИИ, может содержать ошибки.

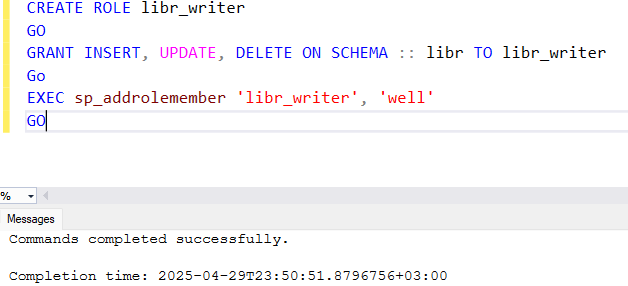
Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Контент, сгенерированный ИИ, может содержать ошибки.

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

Контент, сгенерированный ИИ, может содержать ошибки.

1. Создадим новую роль уровня базы данных и добавим ей разрешение на удаление (DELETE), изменение (UPDATE) и добавление данных (INSERT) в объекты схемы libr. Добавим нашего пользователя к этой роли. Все эти действия надо выполняться с правами администратора или владельца базы данных. Как и в предыдущем случае, все это можно сделать в графическом интерфейсе или запуском скрипта.



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

Контент, сгенерированный ИИ, может содержать ошибки.

1. С помощью оператора CREATE VIEW создайте представление, выбирающее из таблицы Book книги, изданные не ранее 2000 года. Предоставьте пользователю с ограниченными правами возможность изменять и добавлять подобные книги (GRANT INSERT, UPDATE). При этом возможности изменять прочие записи таблицы и добавлять книги, изданные до 2000 года, он иметь не должен. Последнее условие должна 10 контролировать указанная вами в CREATE VIEW опция (какая именно – разберитесь по справке и материалам лекции). Обратите внимание, что предоставлять разрешения вы должны на представление.

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Контент, сгенерированный ИИ, может содержать ошибки.

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

1. Подключитесь к базе как пользователь HR и продолжайте использовать это соединение в дальнейшем
2. Определите имена всех представлений, принадлежащих пользователю HR с помощью представления USER\_VIEWS.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Контент, сгенерированный ИИ, может содержать ошибки.

1. Создайте представление с именем EMP\_VIEW для просмотра следующих данных о всех сотрудниках, приписанных к отделам: номер, фамилия, дата приёма на работу, номер\_руководителя, фамилия\_руководителя, название отдела. Дайте столбцам с номером и фамилией сотрудника имена emp\_id и emp\_lastname, а столбцам с номером и фамилией руководителя – имена man\_id и man\_last\_name.

Изображение выглядит как текст, электроника, снимок экрана, дисплей

Контент, сгенерированный ИИ, может содержать ошибки.

1. Выберите все данные из представления, чтобы убедиться, что оно отображает корректную информацию.

Изображение выглядит как текст, электроника, снимок экрана, дисплей

Контент, сгенерированный ИИ, может содержать ошибки.

1. Снова выполните запрос из п.2. Убедитесь, что в списке появилось имя только что созданного представления.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Контент, сгенерированный ИИ, может содержать ошибки.

1. Выведите полные имена всех таблиц (в формате .), к которым текущий пользователь oraclelabs имеет доступ. Исключите имена таблиц, принадлежащих пользователю.

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

Контент, сгенерированный ИИ, может содержать ошибки.

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

1. С помощью представлений словаря данных USER\_SYS\_PRIVS, USER\_TAB\_PRIVS и USER\_ROLE\_PRIVS просмотрите список привилегий и ролей, выданных пользователю.

Изображение выглядит как текст, снимок экрана, линия, График

Контент, сгенерированный ИИ, может содержать ошибки.

1. Выдайте пользователю oraclelabs привилегии на просмотр таблиц employees и departments. Какого уровня эти привилегии (системные или объектные)?

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

Контент, сгенерированный ИИ, может содержать ошибки.

1. Подключитесь к базе как пользователь oraclelabs и убедитесь, что вы можете просматривать таблицы employees и departments в схеме HR, но не может просмотреть таблицу locations

Изображение выглядит как текст, снимок экрана, число, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, число, Параллельный

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Контент, сгенерированный ИИ, может содержать ошибки.

1. Для того чтобы было удобнее обращаться к таблицам пользователя hr, пользователь oraclelabs решает создать для них синонимы с помощью команд вида:

CREATE [OR REPLACE] SYNONYM FOR

но получает ошибку ORA-01031: insufficient privileges.

Для создания синонимов требуется обладать привилегией CREATE SYNONYM. Какого уровня эта привилегия? Кто может её выдать? Выдайте привилегию пользователю oraclelabs.

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

Контент, сгенерированный ИИ, может содержать ошибки.

1. От имени пользователя oraclelabs создайте синонимы hremp и hrdept для таблиц employees и departments в схеме HR. Выберите данные о номерах, фамилиях и названиях отделов сотрудников, обращаясь к таблицам через синонимы.

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

Контент, сгенерированный ИИ, может содержать ошибки.

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

Контент, сгенерированный ИИ, может содержать ошибки.

1. Отберите привилегию CREATE SYNONYM у пользователя oraclelabs. Проверьте, что доступ к таблицам через синонимы сохраняется



1. От имени пользователя HR предоставьте пользователю oraclelabs привилегии на обновление имён, фамилий и адресов эл. почты сотрудников. Будьте внимательны, он не должен иметь прав обновлять другие данные.

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

Контент, сгенерированный ИИ, может содержать ошибки.

1. От имени пользователя oraclelabs выберите имя и оклад сотрудника с номером 135. Затем одной командой UPDATE обновите имя сотрудника с номером 135 на Ivan и понизьте оклад на 200. Какие данные были обновлены? Обновите имя сотрудника с номером 135 на Ivan. Проверьте изменения, затем отмените их с помощью команды ROLLBACK

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

Контент, сгенерированный ИИ, может содержать ошибки.

1. Пользователь HR решает, что пользователь oraclelabs не должен видеть финансовые и некоторые другие данные о сотрудниках. Что нужно сделать, чтобы предоставить пользователю oraclelabs читать только столбцы employee\_id, last\_name, first\_name, email, job\_id, manager\_id и department\_id таблицы EMPLOYEES? Учтите, что привилегия SELECT не может быть выдана на определённые столбцы. После выполнения необходимых действий проверьте, что пользователь oraclelabs не может читать данные из столбцов salary и commission\_pct

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Контент, сгенерированный ИИ, может содержать ошибки.

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

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, число, Шрифт, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

1. От имени пользователя HR создайте представление DEPT50, содержащее номера, фамилии и номера отделов всех сотрудников отдела 50. Назовите столбцы представления empno, employee и deptno. Запретите операцию перевода служащего в другой отдел через представление.

Изображение выглядит как текст, электроника, снимок экрана, дисплей

Контент, сгенерированный ИИ, может содержать ошибки.

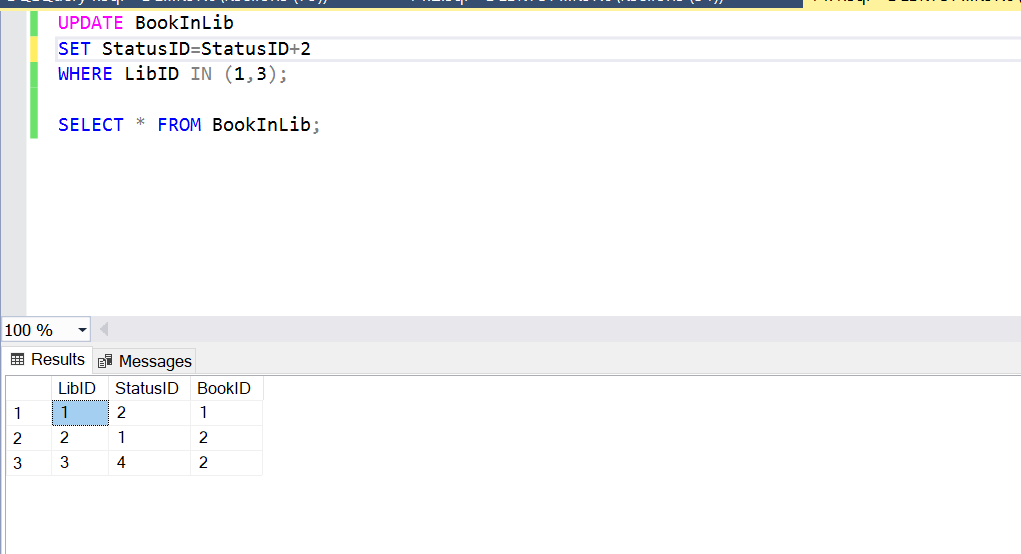
1. Убедитесь, что через представление не получится изменить номер отдела служащего по фамилии Matos на 80, но получится изменить его фамилию на Ivanov. Не забудьте отменить произведённые изменения командой ROLLBACK

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

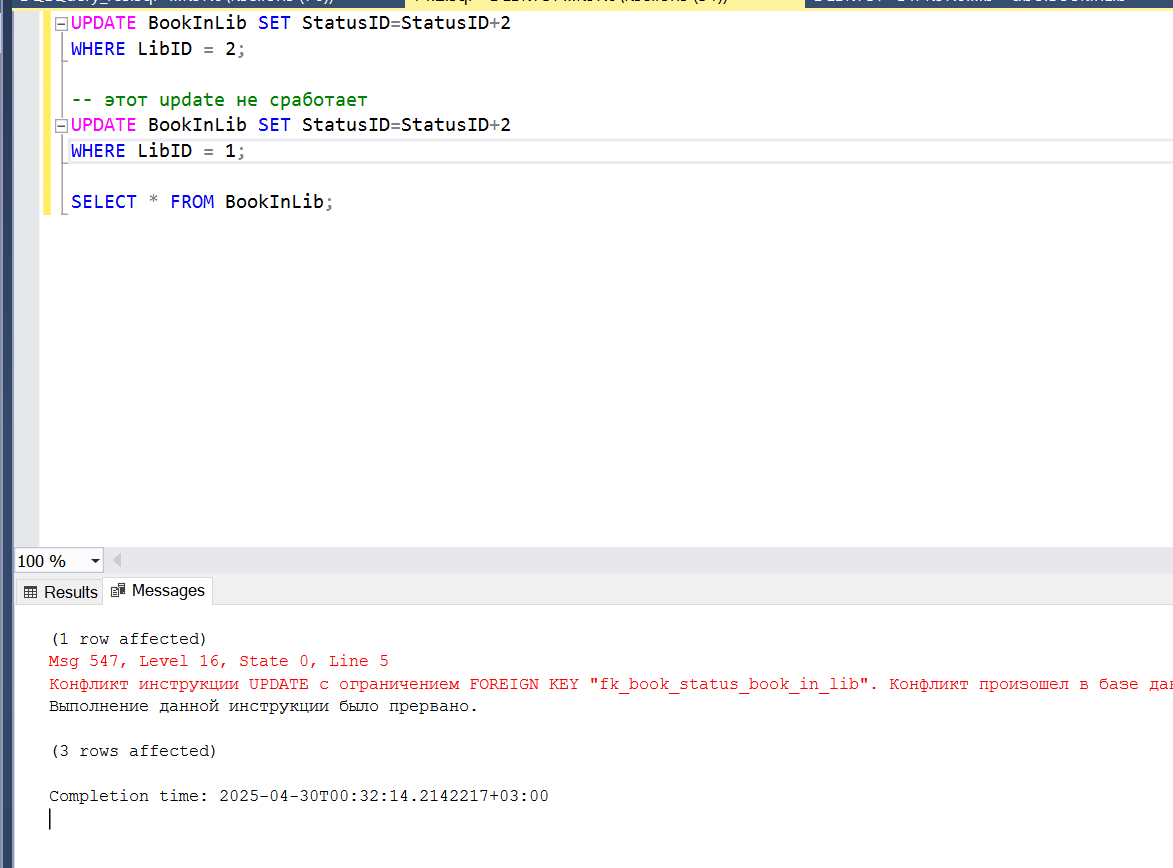
Контент, сгенерированный ИИ, может содержать ошибки.

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

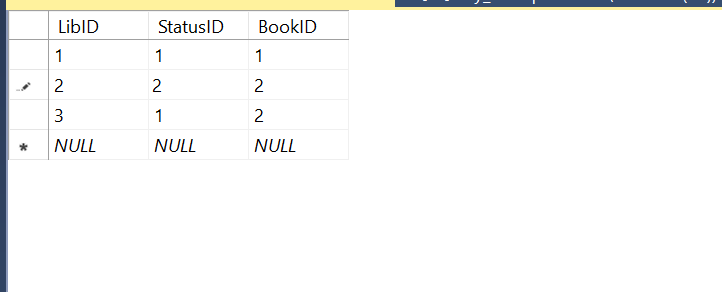
1) обновление двух или более записей одним оператором update, так чтобы одно из изменений не могло быть произведено (транзакция должна быть целиком откачена);

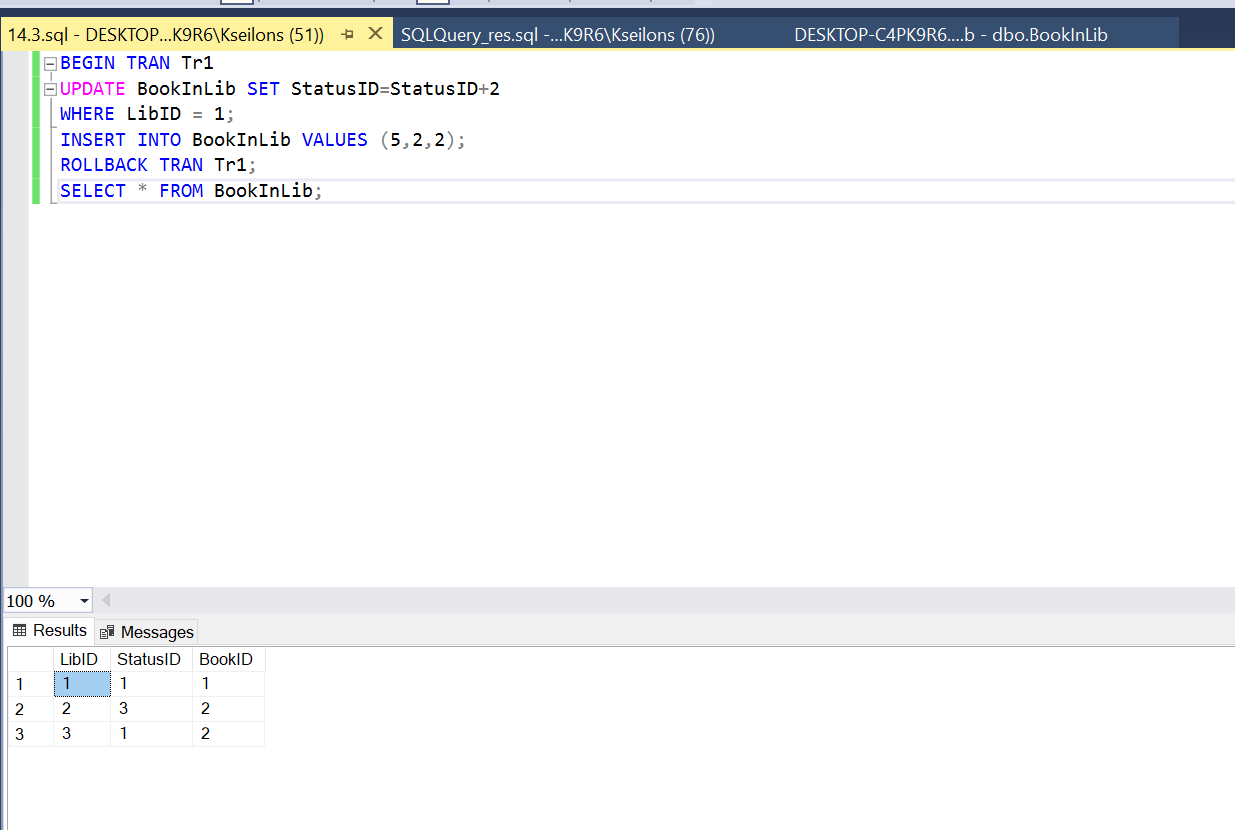
****

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

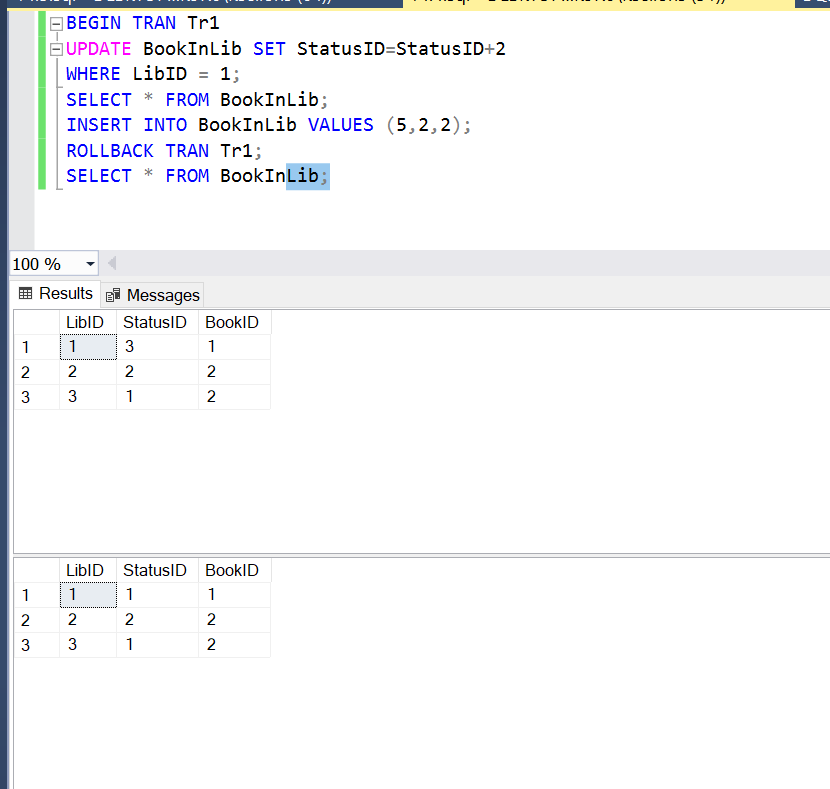


3) измените данные так, чтобы они были в первоначальном стоянии (в графическом интерфейсе или написав соответствующий код); оформите последовательность из одного оператора update и одного оператора insert в виде именованной явной транзакции, по-отдельности эти операторы должны выполняться успешно; транзакцию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

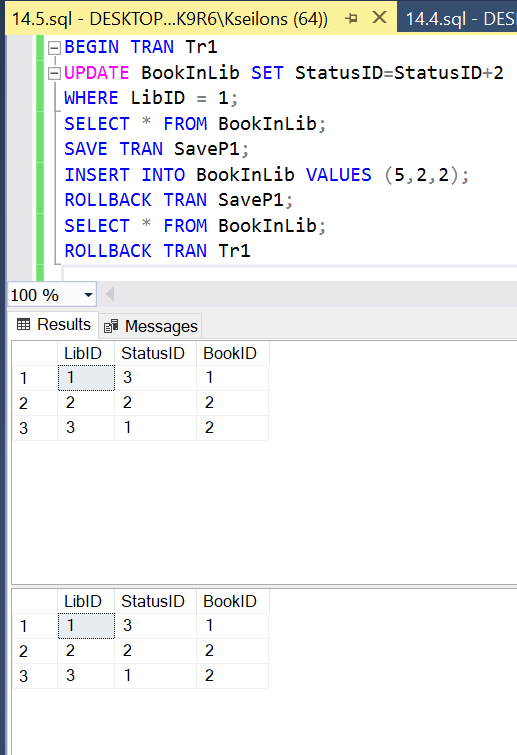




4) между операторами update и insert кода транзакции, созданной на предыдущем шаге, поставьте оператор select, выбирающий все записи из соответствующей таблицы; после отката транзакции выполните аналогичный select; убедитесь, что внутренний select «видит» производимые изменения, т.е. внутри транзакции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



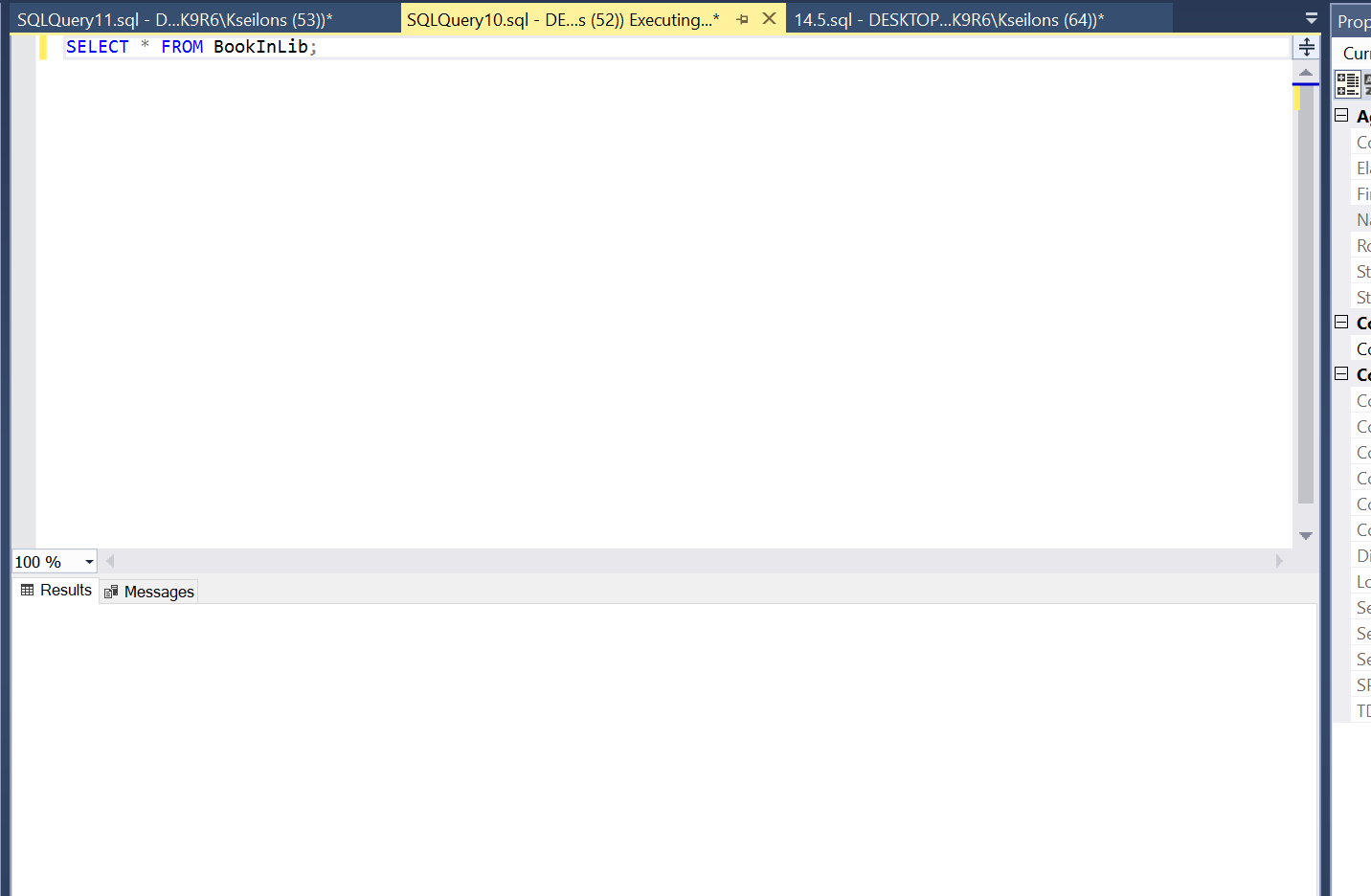
5) дополните транзакцию точкой сохранения установленной после внутреннего select; выполните откат до точки сохранения (в ROLLBACK TRAN указываете имя точки сохранения, а не транзакции), ваш код должен выглядеть примерно так:



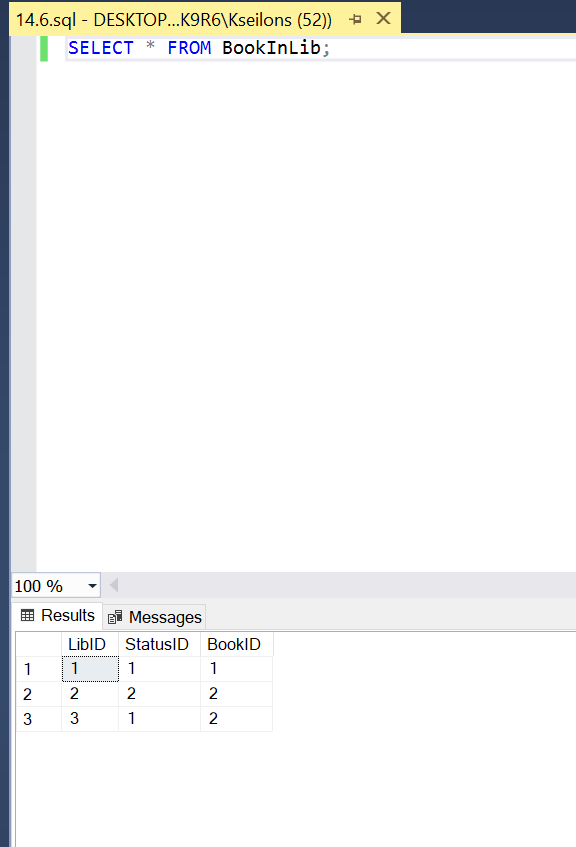
6) обратите внимание, что на момент окончания задания 5 у вас транзакция не завершена; откройте новое окно и выполните в нем следующий код:

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;

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

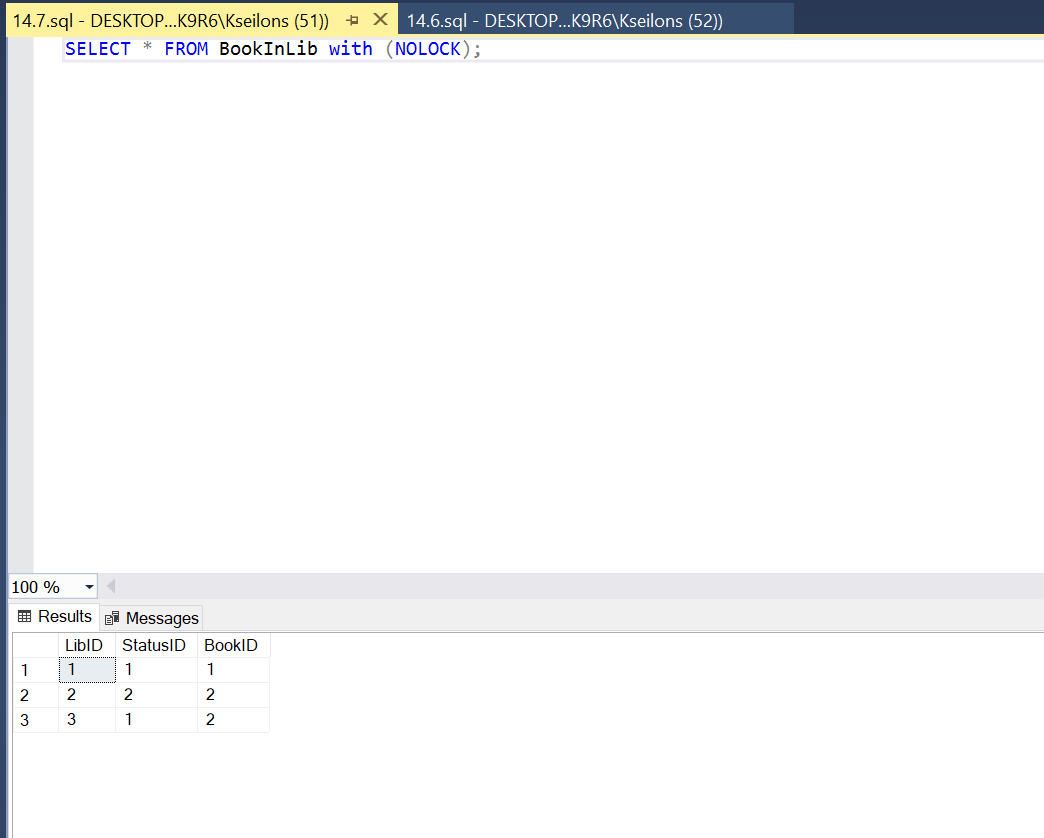
Контент, сгенерированный ИИ, может содержать ошибки.



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);

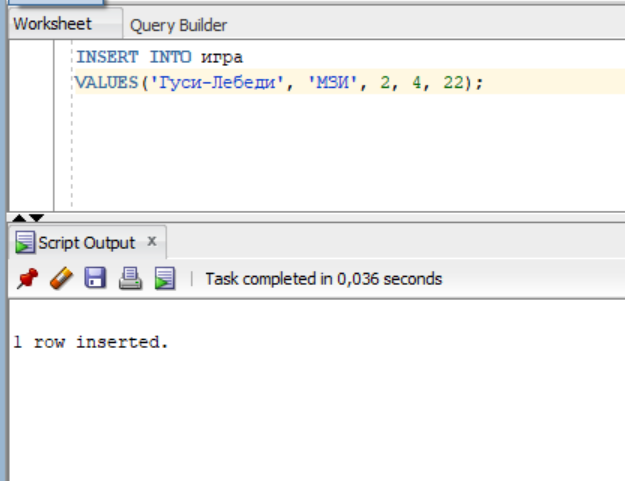
Указание NOLOCK разрешает чтение результатов незафиксированных транзакций. Таким образом, мы не ожидали окончания выполнения параллельной транзакции (запрос отработал сразу), но получили «грязное чтение» - был считан результат, незафиксированной транзакции, которую мы планировали отменить, и вы должны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



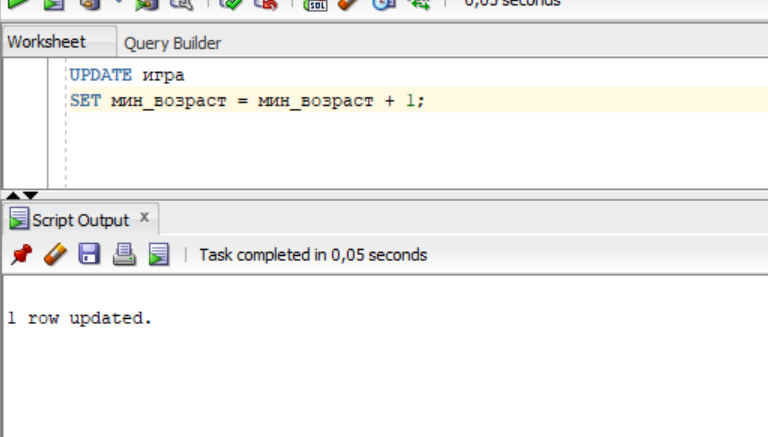
# ЛАБОРАТОРНАЯ РАБОТА (Oracle) ЧЕТЫРНАДЦАТОЙ НЕДЕЛИ КУРСА «УПРАВЛЕНИЕ ДАННЫМИ». ОСНОВЫ УПРАВЛЕНИЯ ТРАНЗАКЦИЯМИ

**Часть 1. Управление транзакциями (INSERT, UPDATE, ALTER TABLE, ROLLBACK)**

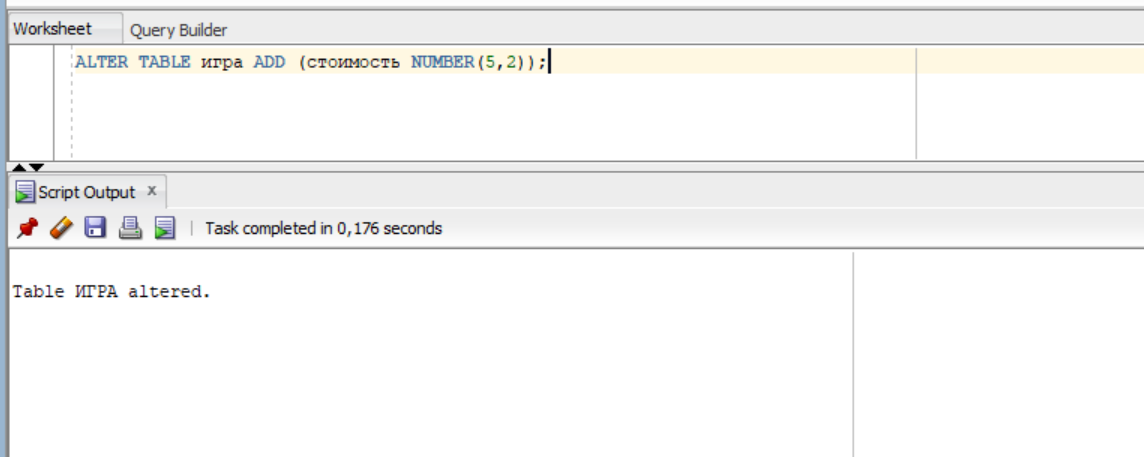
Добавь новую строку в таблицу Игра:



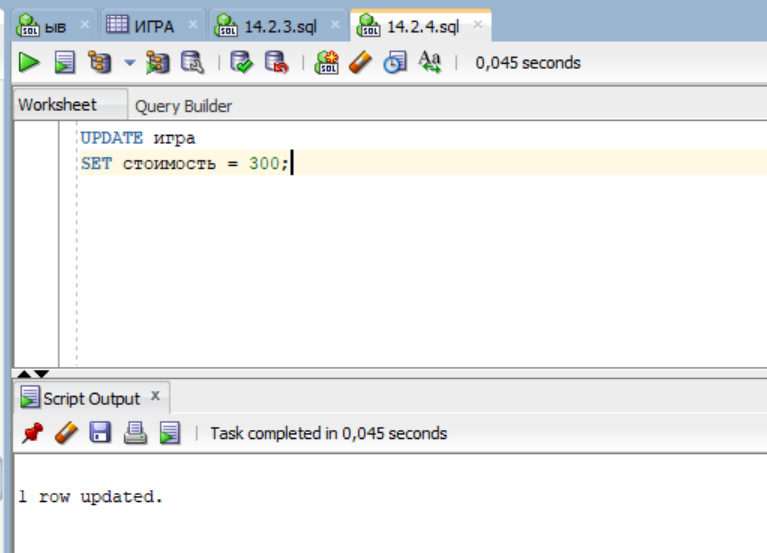
Увеличь минимальный возраст всех игр на 1:



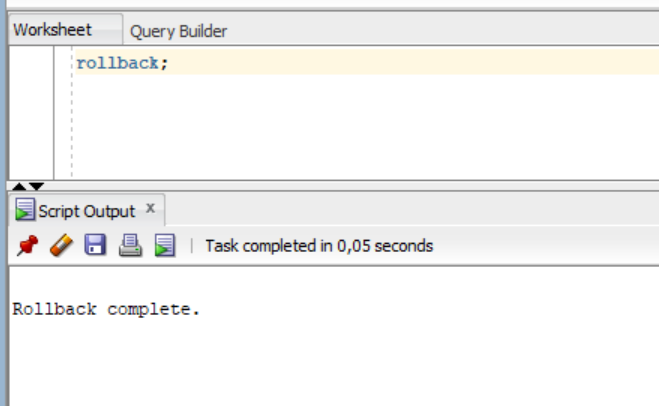
Добавление нового столбца "стоимость" в таблицу Игра:



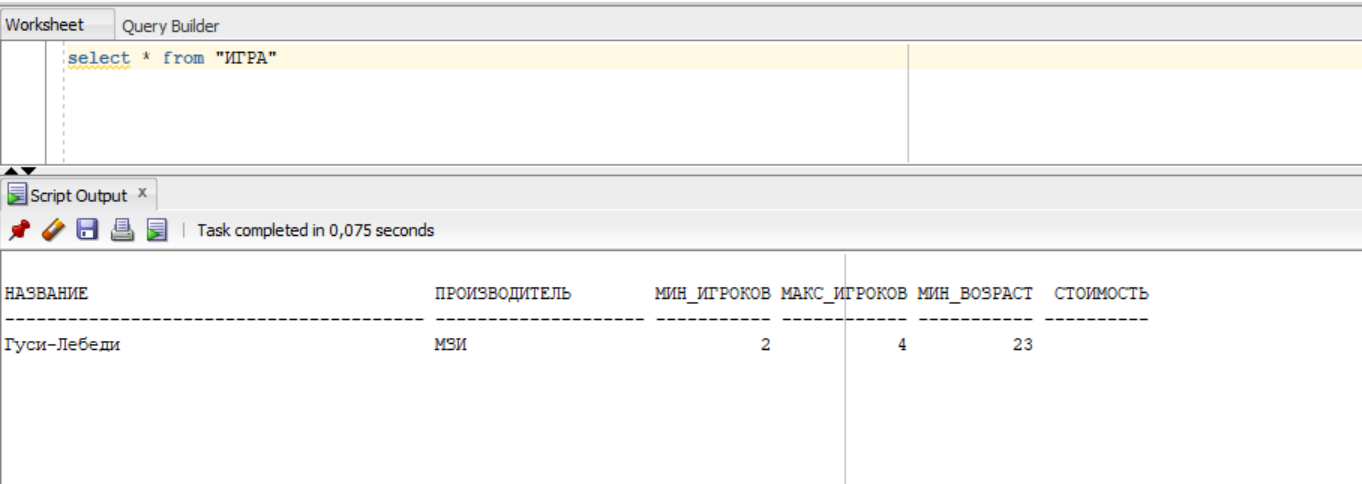
Установка стоимости всех игр равной 300:

**

Выполнение команды отката:

**

Проверка содержимого таблицы Игра:

**

**Вывод по части 1:**

* Все DML-команды (INSERT, UPDATE) до ALTER TABLE были зафиксированы, так как команда ALTER TABLE выполнила скрытый COMMIT.
* UPDATE стоимости был отменён командой ROLLBACK.
* Изменение структуры таблицы (ALTER TABLE) не откатывается.

**Часть 2. Использование CREATE SCHEMA для создания представления**

Попытка создания представления и выдачи прав несуществующему пользователю:

Изображение выглядит как текст, электроника, снимок экрана, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

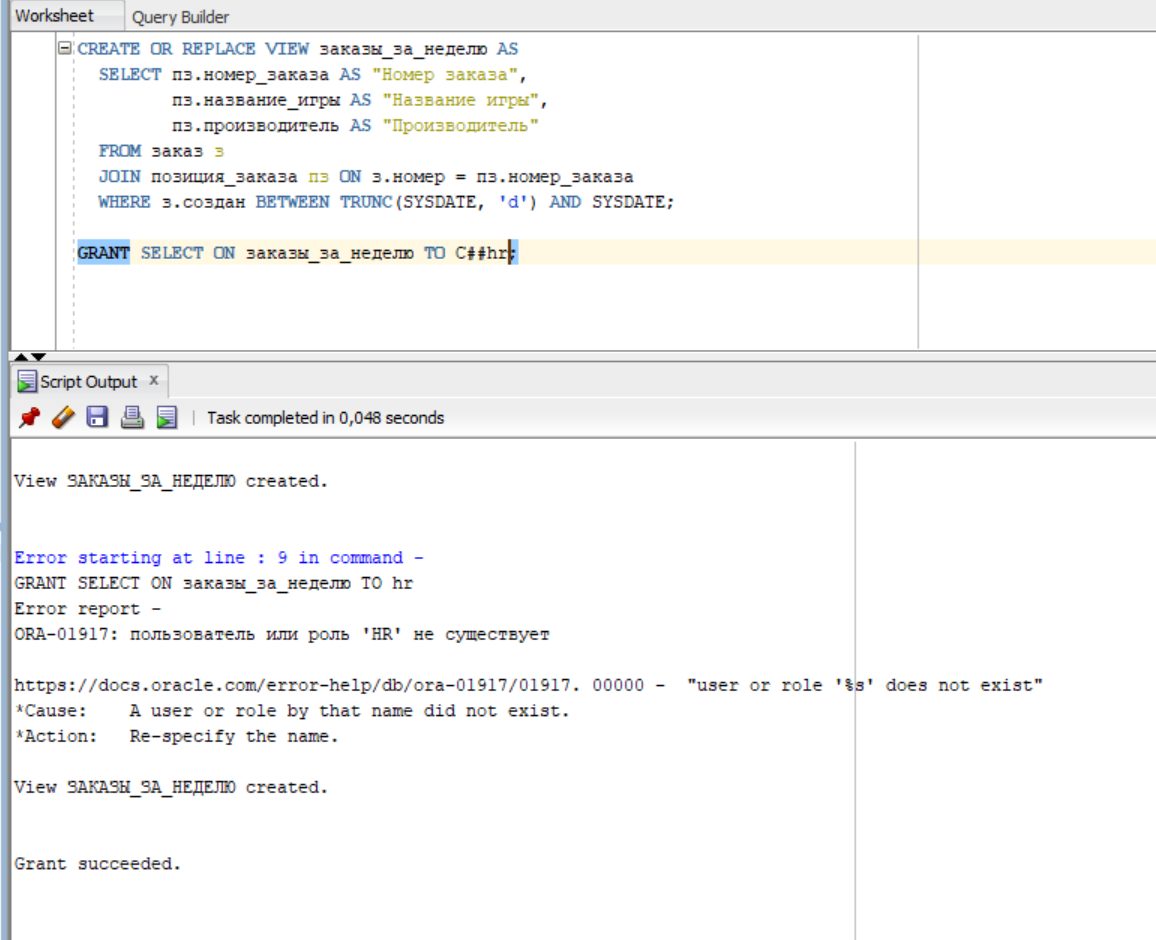
Ошибка: пользователь superstar не существует.  
Транзакция отменена. Представление не создано.

Проверка наличия представления:





Корректное создание представления с выдачей прав пользователю oraclelabs:



Подключись как HR и посмотри содержимое представления:





# 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.