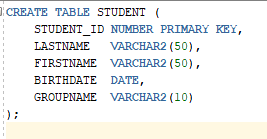
1. Создайте таблицу, имеющую несколько атрибутов, один из которых первичный ключ.



Изображение выглядит как текст, Шрифт, снимок экрана, белый

Контент, сгенерированный ИИ, может содержать ошибки.

1. Заполните таблицу данными (10 шт.).

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, число, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

1. Создайте BEFORE – триггер уровня оператора на события INSERT, DELETE и UPDATE. Этот и все последующие триггеры должны выдавать сообщение на серверную консоль (DMS\_OUTPUT) со своим собственным именем.





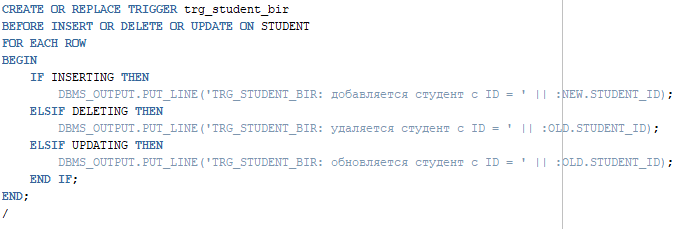
Изображение выглядит как текст, снимок экрана, Шрифт, линия

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, Шрифт, белый

Контент, сгенерированный ИИ, может содержать ошибки.

1. Создайте BEFORE-триггер уровня строки на события INSERT, DELETE и UPDATE.
2. Примените предикаты INSERTING, UPDATING и DELETING.



Изображение выглядит как текст, Шрифт, снимок экрана, белый

Контент, сгенерированный ИИ, может содержать ошибки.

1. Разработайте AFTER-триггеры уровня оператора на события INSERT, DELETE и UPDATE.

Изображение выглядит как текст, Шрифт, линия, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

1. Разработайте AFTER-триггеры уровня строки на события INSERT, DELETE и UPDATE.

Изображение выглядит как текст, Шрифт, число, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, Шрифт, снимок экрана, белый

Контент, сгенерированный ИИ, может содержать ошибки.

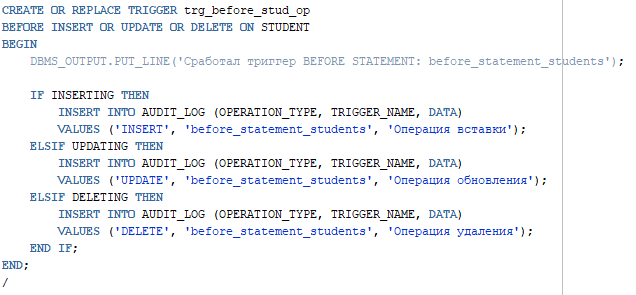
1. Создайте таблицу с именем AUDIT. Таблица должна содержать поля: OperationDate, OperationType (операция вставки, обновления и удаления), TriggerName(имя триггера), Data (строка со значениями полей до и после операции).

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Контент, сгенерированный ИИ, может содержать ошибки.Изображение выглядит как текст, Шрифт, белый, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

1. Измените все триггеры таким образом, чтобы они регистрировали все операции с исходной таблицей в таблице AUDIT.



Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.Изображение выглядит как текст, снимок экрана, Шрифт, число

Контент, сгенерированный ИИ, может содержать ошибки.Изображение выглядит как текст, Шрифт, снимок экрана, линия

Контент, сгенерированный ИИ, может содержать ошибки.

1. Выполните операцию, нарушающую целостность таблицы по первичному ключу. Выясните, зарегистрировал ли триггер это событие. Объясните результат.



Изображение выглядит как текст, снимок экрана, Шрифт, число

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, Шрифт, линия, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

1. Удалите (drop) исходную таблицу. Объясните результат. Добавьте триггер, запрещающий удаление исходной таблицы.





Изображение выглядит как текст, снимок экрана, Шрифт, число

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, Шрифт, снимок экрана, белый

Контент, сгенерированный ИИ, может содержать ошибки.

1. Удалите (drop) таблицу AUDIT. Просмотрите состояние триггеров с помощью SQL-DEVELOPER. Объясните результат. Измените триггеры.





Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

1. Создайте представление над исходной таблицей. Разработайте INSTEAD OF UPDATE-триггер. Триггер должен добавлять новую строку в таблицу, а старую помечать как недействительную.





Изображение выглядит как текст, снимок экрана, Шрифт, число

Контент, сгенерированный ИИ, может содержать ошибки.

1. Продемонстрируйте, в каком порядке выполняются триггеры.



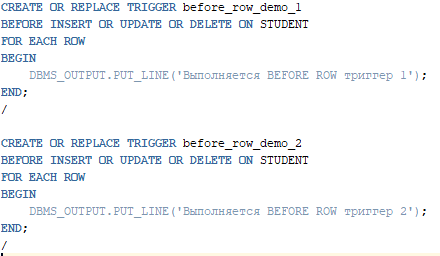
Изображение выглядит как текст, Шрифт, снимок экрана, белый

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Контент, сгенерированный ИИ, может содержать ошибки.

1. Создайте несколько триггеров одного типа, реагирующих на одно и то же событие, и покажите, в каком порядке они выполняются. Измените порядок выполнения этих триггеров.



Изображение выглядит как текст, снимок экрана, Шрифт, линия

Контент, сгенерированный ИИ, может содержать ошибки.Изображение выглядит как текст, Шрифт, снимок экрана, линия

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Контент, сгенерированный ИИ, может содержать ошибки.Изображение выглядит как текст, снимок экрана, Шрифт, линия

Контент, сгенерированный ИИ, может содержать ошибки.\

ОТВЕТЫ НА ВОПРОСЫ

**1. Перечислите типы триггеров, поддерживаемых Oracle**

Oracle поддерживает **триггеры следующих типов**:

**📌 По времени срабатывания:**

* **BEFORE** — до выполнения DML-операции.
* **AFTER** — после выполнения DML-операции.
* **INSTEAD OF** — вместо DML-операции (только для представлений).

**📌 По уровню:**

* **ROW-level (на уровне строки)** — FOR EACH ROW (срабатывает для каждой строки).
* **STATEMENT-level (на уровне оператора)** — по умолчанию (срабатывает один раз для всей операции).

**📌 По типу операций:**

* DML-триггеры: INSERT, UPDATE, DELETE
* DDL-триггеры: CREATE, ALTER, DROP, TRUNCATE и др. (называются **system triggers**)
* Триггеры на события БД: LOGON, LOGOFF, STARTUP, SHUTDOWN, SERVERERROR

**2. Поясните правило: триггер является частью транзакции**

Это означает, что:

* **Триггер исполняется внутри той же транзакции**, что и вызывающая его операция.
* Все действия, выполненные триггером, **будут отменены**, если транзакция откатится (ROLLBACK).
* Примеры:
  + Если INSERT активирует триггер, и потом срабатывает ROLLBACK, то **и вставка, и действия триггера отменятся**.
  + Если COMMIT выполнен, то и операция, и работа триггера будут сохранены в БД.

**3. Перечислите привилегии, необходимые для работы с триггерами**

Для создания и использования триггеров нужны следующие привилегии:

**📌 Системные:**

* CREATE TRIGGER — для создания триггера в своей схеме.
* ALTER ANY TRIGGER, DROP ANY TRIGGER — для управления чужими триггерами.
* ADMINISTER DATABASE TRIGGER — для создания системных триггеров (например, на STARTUP, LOGON и т.п.).

**📌 Объектные:**

* Привилегии на таблицу, к которой применяется триггер:
  + INSERT, UPDATE, DELETE — если триггер использует соответствующие операции.
  + SELECT — если в триггере используется выборка данных.

**4. Перечислите события, на которые могут срабатывать DML-триггеры**

Триггеры DML срабатывают на:

* INSERT
* UPDATE
* DELETE

Могут быть объединены:

AFTER INSERT OR UPDATE OR DELETE ON STUDENT

Также можно ограничить по колонке:

AFTER UPDATE OF salary ON EMPLOYEES

**5. Поясните принцип применения INSTEAD OF-триггеров в Oracle**

**INSTEAD OF-триггеры** применяются **только к представлениям (VIEW)** и нужны для **реализации DML-операций на представлениях**, когда они сами по себе не поддерживают INSERT, UPDATE, DELETE.

**🔹 Принцип:**

* Такой триггер **перехватывает DML-запрос** к представлению и **выполняет вместо него свои действия** (например, вставку в базовые таблицы).

**🔹 Пример:**

CREATE OR REPLACE TRIGGER trg\_update\_view

INSTEAD OF UPDATE ON my\_view

FOR EACH ROW

BEGIN

UPDATE base\_table

SET col1 = :NEW.col1

WHERE id = :OLD.id;

END;

/

Это позволяет **работать с представлением как с таблицей**, даже если оно агрегирует или объединяет данные из нескольких источников.