**Богданов Юрий I2302**

**Вариант №2**

1. Создать триггер, запрещающий ввод данных в таблицу SALE в выходные дни с 17:00 до 18:00 (30 б.)
2. Создать представление с данными о продажах (БД №1): дата продажи, фамилия клиента, фамилия продавца, название товара, количество проданного товара.(10 б.)
3. Создать триггер для ввода данных в БД данных о новом клиенте через созданное представление.(60 б.)

**1. Триггер, запрещающий вставку в SALE в выходные с 17:00 до 18:00**

CREATE OR REPLACE TRIGGER trg\_block\_sale\_insert

BEFORE INSERT ON SALE

FOR EACH ROW

DECLARE

e\_block\_insert EXCEPTION;

v\_day VARCHAR2(10);

v\_hour NUMBER;

BEGIN

SELECT TRIM(UPPER(TO\_CHAR(SYSDATE, 'DAY', 'NLS\_DATE\_LANGUAGE=ENGLISH'))),

TO\_NUMBER(TO\_CHAR(SYSDATE, 'HH24'))

INTO v\_day, v\_hour

FROM dual;

IF v\_day IN ('SATURDAY', 'SUNDAY') AND v\_hour = 17 THEN

RAISE e\_block\_insert;

END IF;

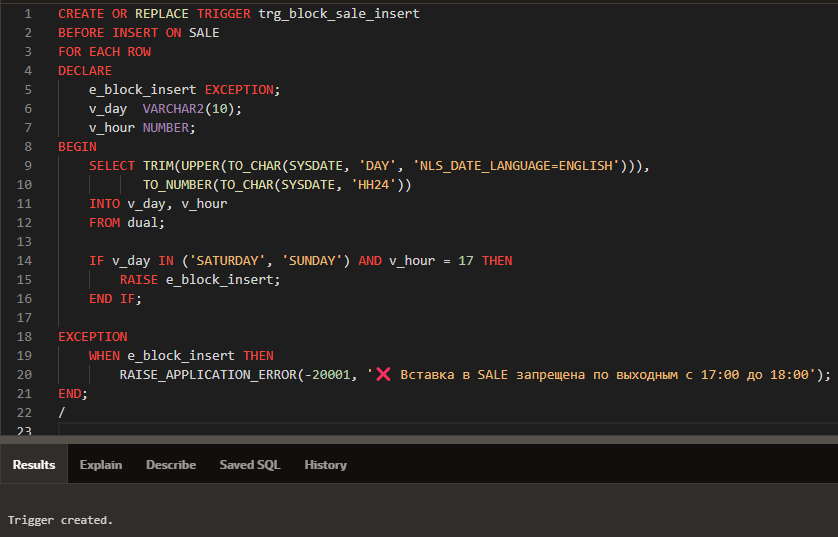
EXCEPTION

WHEN e\_block\_insert THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Вставка в SALE запрещена по выходным с 17:00 до 18:00');

END;

/



**Проверка**

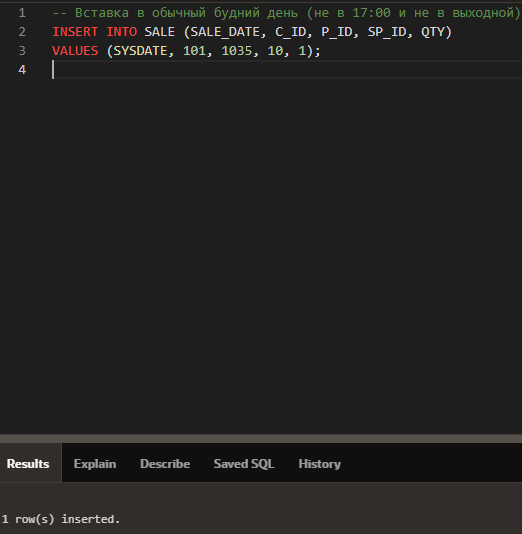
**Цель:** запретить вставку в таблицу SALE по выходным с 17:00 до 18:00.

**Успешная проверка:**

-- Вставка в обычный будний день (не в 17:00 и не в выходной)

INSERT INTO SALE (SALE\_DATE, C\_ID, P\_ID, SP\_ID, QTY)

VALUES (SYSDATE, 101, 1035, 10, 1);



Строка добавлена успешно.

**Проверка блокировки:**

Чтобы проверить, можно временно "смоделировать" системное время:

-- Эмулируем триггер вручную, подставив SATURDAY + 17:00 (только для тестов)

DECLARE

v\_day VARCHAR2(10);

v\_hour NUMBER;

BEGIN

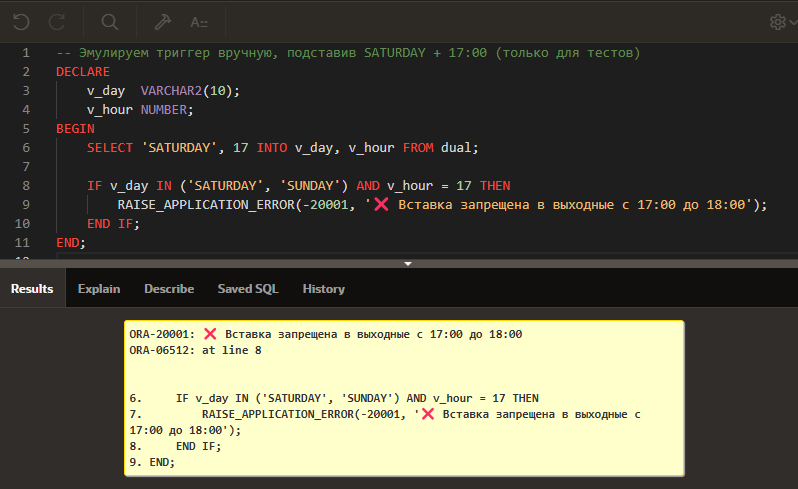
SELECT 'SATURDAY', 17 INTO v\_day, v\_hour FROM dual;

IF v\_day IN ('SATURDAY', 'SUNDAY') AND v\_hour = 17 THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Вставка запрещена в выходные с 17:00 до 18:00');

END IF;

END;



**2. Представление с данными о продажах**

CREATE OR REPLACE VIEW sales\_info\_view AS

SELECT

s.sale\_date,

c.c\_name AS client\_name,

sp.sp\_name AS seller\_name,

p.p\_desc AS product\_name,

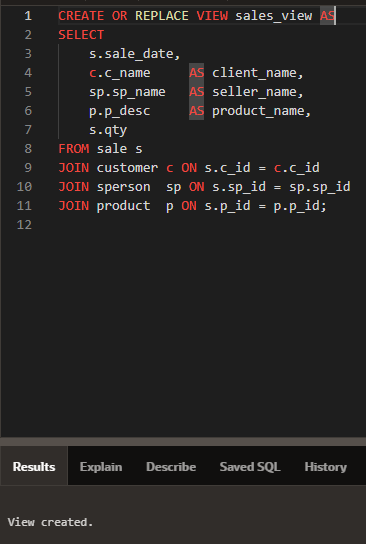
s.qty

FROM SALE s

JOIN CUSTOMER c ON s.c\_id = c.c\_id

JOIN SPERSON sp ON s.sp\_id = sp.sp\_id

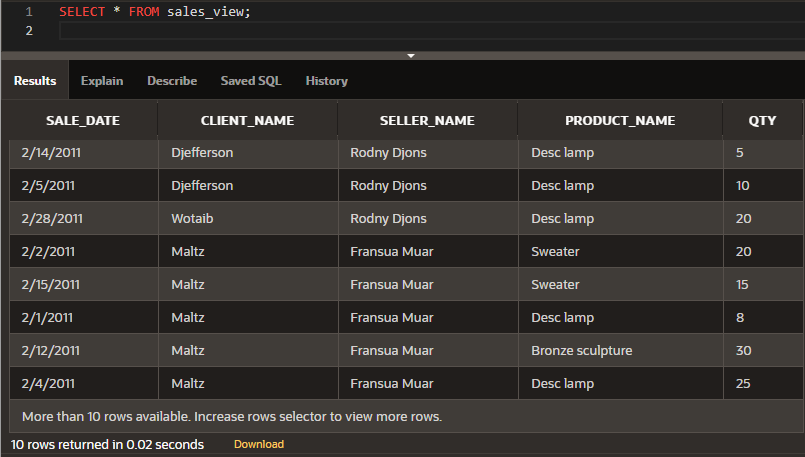
JOIN PRODUCT p ON s.p\_id = p.p\_id;



**Проверка:**

SELECT \* FROM sales\_info\_view;

**Цель:** получить данные о продажах (дата, клиент, продавец, товар, количество).



**3. Ввод нового клиента через представление + пакет + INSTEAD OF-триггер**

**3.1. Представление customer\_view (без ID):**

CREATE OR REPLACE VIEW customer\_view AS

SELECT

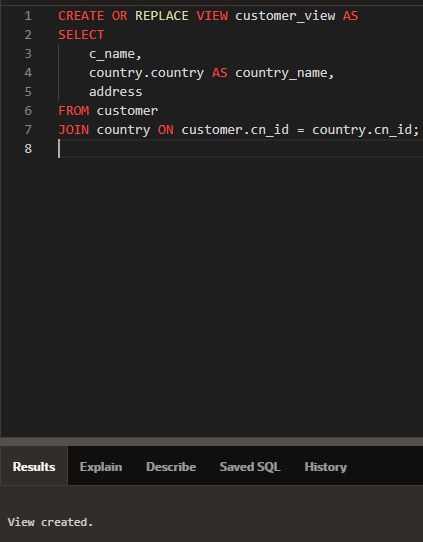
c\_name,

country.country AS country\_name,

address

FROM customer

JOIN country ON customer.cn\_id = country.cn\_id;



**3.2. Пакет с закрытой переменной и обработкой ошибок:**

CREATE OR REPLACE PACKAGE customer\_pkg AS

PROCEDURE insert\_customer(

p\_name IN CUSTOMER.c\_name%TYPE,

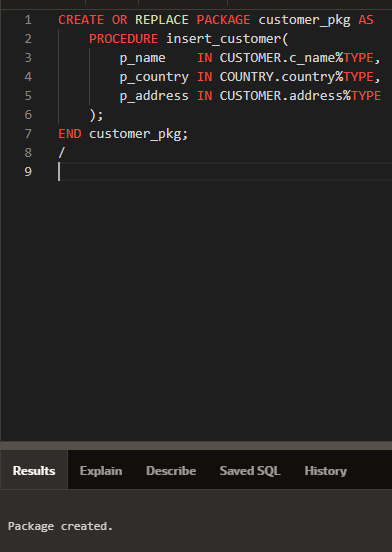
p\_country IN COUNTRY.country%TYPE,

p\_address IN CUSTOMER.address%TYPE

);

END customer\_pkg;

/



CREATE OR REPLACE PACKAGE BODY customer\_pkg AS

-- закрытая переменная

v\_cn\_id COUNTRY.cn\_id%TYPE;

PROCEDURE insert\_customer(

p\_name IN CUSTOMER.c\_name%TYPE,

p\_country IN COUNTRY.country%TYPE,

p\_address IN CUSTOMER.address%TYPE

) IS

v\_c\_id CUSTOMER.c\_id%TYPE;

e\_country\_not\_found EXCEPTION;

e\_customer\_exists EXCEPTION;

BEGIN

-- Поиск страны

SELECT cn\_id INTO v\_cn\_id

FROM COUNTRY

WHERE country = p\_country;

-- Проверка дубликата

SELECT COUNT(\*) INTO v\_c\_id

FROM CUSTOMER

WHERE c\_name = p\_name AND address = p\_address;

IF v\_c\_id > 0 THEN

RAISE e\_customer\_exists;

END IF;

-- Новый ID

SELECT MAX(c\_id) + 1 INTO v\_c\_id FROM CUSTOMER;

-- Вставка

INSERT INTO CUSTOMER (c\_id, c\_name, cn\_id, address)

VALUES (v\_c\_id, p\_name, v\_cn\_id, p\_address);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RAISE\_APPLICATION\_ERROR(-20010, 'Страна не найдена.');

WHEN e\_customer\_exists THEN

RAISE\_APPLICATION\_ERROR(-20020, 'Такой клиент уже есть.');

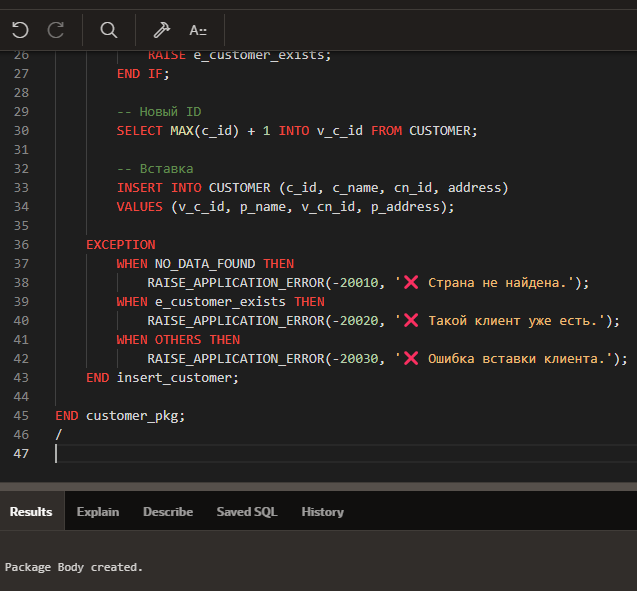
WHEN OTHERS THEN

RAISE\_APPLICATION\_ERROR(-20030, 'Ошибка вставки клиента.');

END insert\_customer;

END customer\_pkg;

/



3.3. INSTEAD OF-триггер на представление:

CREATE OR REPLACE TRIGGER trg\_insert\_customer\_view

INSTEAD OF INSERT ON customer\_view

FOR EACH ROW

BEGIN

customer\_pkg.insert\_customer(

p\_name => :NEW.c\_name,

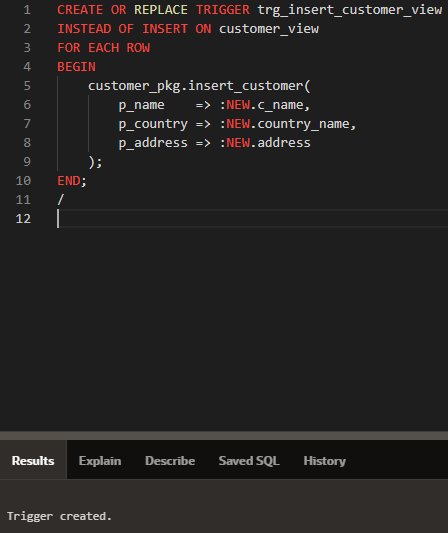
p\_country => :NEW.country\_name,

p\_address => :NEW.address

);

END;

/



**Проверка:**

**Цель:** вставка нового клиента в таблицу CUSTOMER через представление customer\_view.

**Успешная вставка:**

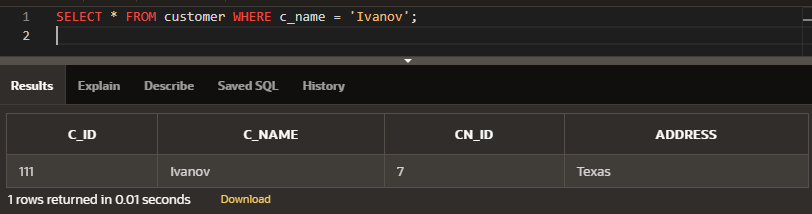
INSERT INTO customer\_view (c\_name, country\_name, address)

VALUES ('Ivanov', 'USA', 'Texas');



Клиент добавлен в CUSTOMER.

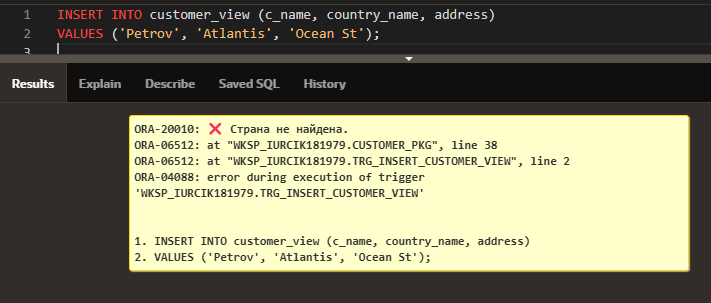
SELECT \* FROM customer WHERE c\_name = 'Ivanov';



**Проверка: страна не существует**

INSERT INTO customer\_view (c\_name, country\_name, address)

VALUES ('Petrov', 'Atlantis', 'Ocean St');



**Проверка: клиент уже существует**

INSERT INTO customer\_view (c\_name, country\_name, address)

VALUES ('Ivanov', 'USA', 'Texas');

