Создать триггер, запрещающий изменение данных в столбцах e\_name и salary таблицы EMP в выходные, в нерабочие часы с 18:00 до 8:00 и по четвергам с 12:00 до 13:00.

Создать представление, которое включает следующие данные: фамилия сотрудника, имя сотрудника, название отдела, зарплата, дата приёма на работу. (таблицы DEPT, EMP)

Создать триггер instead off для обновления данных, содержащихся в базовых таблицах.

**1. Триггер: запрет на изменение** E\_NAME **и** SALARY **по условиям**

CREATE OR REPLACE TRIGGER trg\_block\_emp\_update

BEFORE UPDATE OF e\_name, salary ON EMP

FOR EACH ROW

DECLARE

e\_restrict EXCEPTION;

v\_day VARCHAR2(10);

v\_hour NUMBER;

v\_min NUMBER;

BEGIN

SELECT TRIM(UPPER(TO\_CHAR(SYSDATE, 'DAY'))),

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

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

INTO v\_day, v\_hour, v\_min

FROM dual;

IF v\_day IN ('SATURDAY', 'SUNDAY') THEN

RAISE e\_restrict;

ELSIF v\_hour < 8 OR v\_hour >= 18 THEN

RAISE e\_restrict;

ELSIF v\_day = 'THURSDAY' AND v\_hour = 12 THEN

RAISE e\_restrict;

END IF;

EXCEPTION

WHEN e\_restrict THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Запрещено изменять имя или зарплату в нерабочее время.');

END;

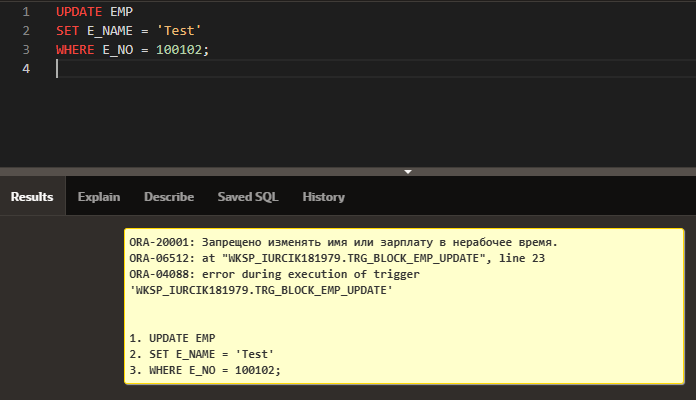
/

**Проверка**:

UPDATE EMP

SET E\_NAME = 'Test'

WHERE E\_NO = 100102;



**2. Представление: фамилия, отдел, зарплата, дата приёма**

CREATE OR REPLACE VIEW emp\_dept\_view AS

SELECT

e.e\_name,

d.d\_name,

e.salary,

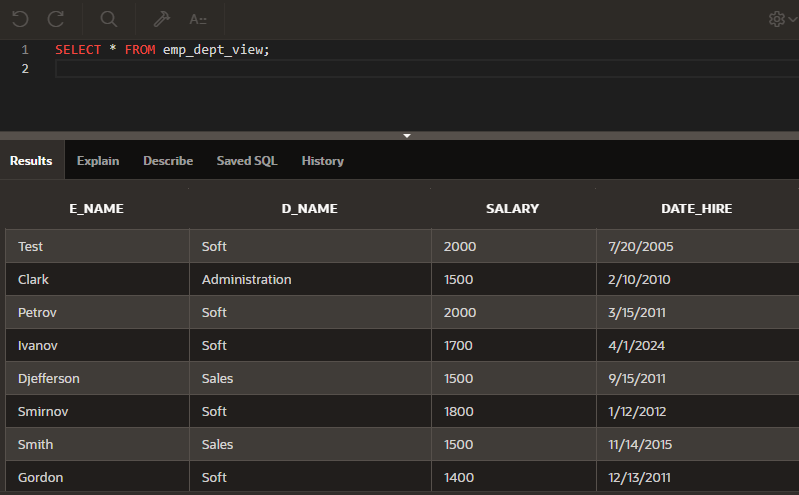
e.date\_hire

FROM EMP e

JOIN DEPT d ON e.d\_no = d.d\_no;

**Проверка**:

SELECT \* FROM emp\_dept\_view;



**3. Замещающий триггер INSTEAD OF UPDATE + пакет (без ID)**

Пакет (с закрытой переменной и обработкой ошибок)

CREATE OR REPLACE PACKAGE emp\_pkg AS

PROCEDURE update\_employee(

p\_name IN EMP.e\_name%TYPE,

p\_dept IN DEPT.d\_name%TYPE,

p\_salary IN EMP.salary%TYPE,

p\_hire\_date IN EMP.date\_hire%TYPE

);

END emp\_pkg;

/

CREATE OR REPLACE PACKAGE BODY emp\_pkg AS

-- Закрытая переменная для d\_no

v\_dno DEPT.d\_no%TYPE;

PROCEDURE update\_employee(

p\_name IN EMP.e\_name%TYPE,

p\_dept IN DEPT.d\_name%TYPE,

p\_salary IN EMP.salary%TYPE,

p\_hire\_date IN EMP.date\_hire%TYPE

) IS

e\_no\_such\_dept EXCEPTION;

e\_no\_such\_emp EXCEPTION;

BEGIN

-- Получить номер отдела

SELECT d\_no INTO v\_dno

FROM DEPT

WHERE d\_name = p\_dept;

-- Обновить данные сотрудника

UPDATE EMP

SET

salary = p\_salary,

date\_hire = p\_hire\_date,

d\_no = v\_dno

WHERE e\_name = p\_name;

IF SQL%ROWCOUNT = 0 THEN

RAISE e\_no\_such\_emp;

END IF;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RAISE\_APPLICATION\_ERROR(-20010, 'Такой отдел не найден.');

WHEN e\_no\_such\_emp THEN

RAISE\_APPLICATION\_ERROR(-20011, 'Сотрудник не найден.');

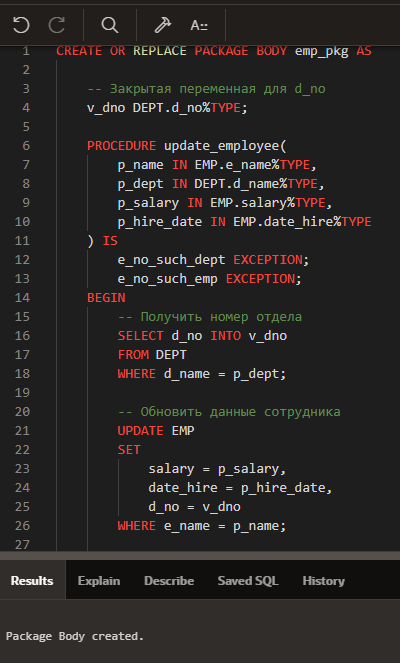
WHEN OTHERS THEN

RAISE\_APPLICATION\_ERROR(-20012, 'Ошибка обновления.');

END update\_employee;

END emp\_pkg;

/



**INSTEAD OF триггер для представления** emp\_dept\_view

CREATE OR REPLACE TRIGGER trg\_emp\_dept\_view\_update

INSTEAD OF UPDATE ON emp\_dept\_view

FOR EACH ROW

BEGIN

emp\_pkg.update\_employee(

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

p\_dept => :NEW.d\_name,

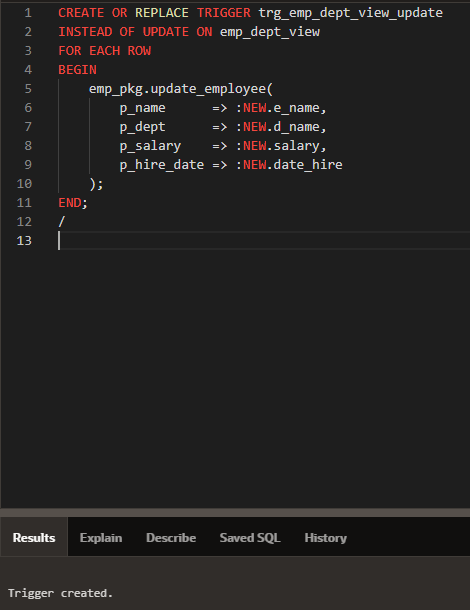
p\_salary => :NEW.salary,

p\_hire\_date => :NEW.date\_hire

);

END;

/



**Проверка:**

Обновляем данные через представление:

UPDATE emp\_dept\_view

SET

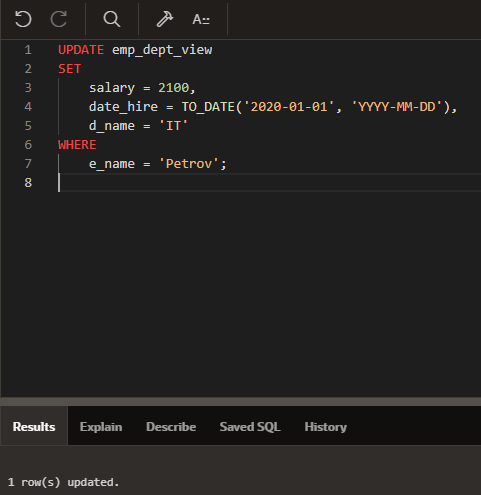
salary = 2100,

date\_hire = TO\_DATE('2020-01-01', 'YYYY-MM-DD'),

d\_name = 'IT'

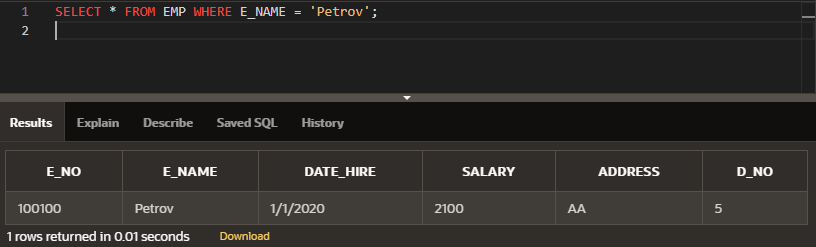
WHERE

e\_name = 'Petrov';



**Проверка:**

SELECT \* FROM EMP WHERE E\_NAME = 'Petrov';



**Дата найма обновилась (была 15/3/2011)**

**Проверка ошибок**

Попробуем ввести несуществующий отдел:

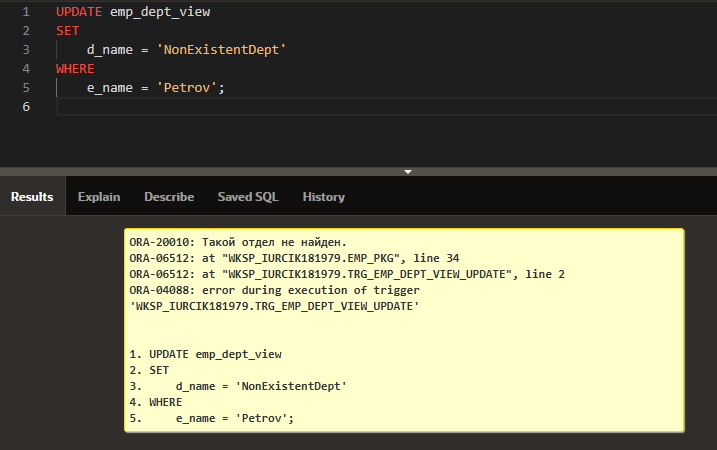
UPDATE emp\_dept\_view

SET

d\_name = 'NonExistentDept'

WHERE

e\_name = 'Petrov';

****