Федеральное агентство связи

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Сибирский государственный университет

телекоммуникаций и информатики»

**Лабораторная работа №4**

Выполнили: студенты IV курса

ИВТ, гр. ИП-713

Михеев Н.А.

Трусов К.В.

Устенко Д.А.

Проверил: старший преподаватель

кафедры ПМиК

Грязнов Н. Г.

Новосибирск, 2020 г.

1. Создайте триггер, который регистрирует вставку каждой строки в одну из ваших таблиц. При регистрации триггер должен сохранять, кроме названия команды и таблицы, имя того, кто вставляет строки, и дату и время вставки.

drop table MYSAL;

create table MYSAL(

    SNUM number(4),

    SNAME varchar2(10) NOT NULL,

    CITY  varchar2(10) NOT NULL,

    COMM  number(7,2) NOT NULL

);

INSERT INTO MYSAL VALUES (1001, 'Peel', 'London', .12);

INSERT INTO MYSAL VALUES (1002, 'Serres', 'San Jose', .13);

INSERT INTO MYSAL VALUES (1004, 'Motica', 'London', .11);

INSERT INTO MYSAL VALUES (1007, 'Rifkin', 'Barcelona', .15);

INSERT INTO MYSAL VALUES (1003, 'Axelrod', 'New York', .10);

drop table history;

create table history (

    command\_name varchar2(20) NOT NULL,

    table\_name varchar2(20) NOT NULL,

    user\_name varchar2(50) NOT NULL,

    time\_of\_insert  date

);

COMMIT;

CREATE TRIGGER dummy

    AFTER INSERT ON MYSAL

    FOR EACH ROW

BEGIN

    INSERT INTO history (command\_name, table\_name, user\_name, time\_of\_insert)

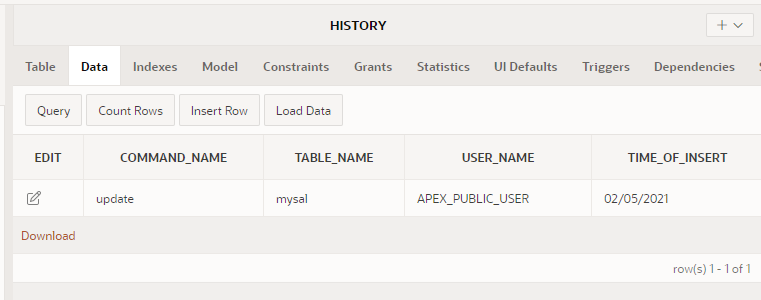
    VALUES ('update', 'mysal', user, sysdate);

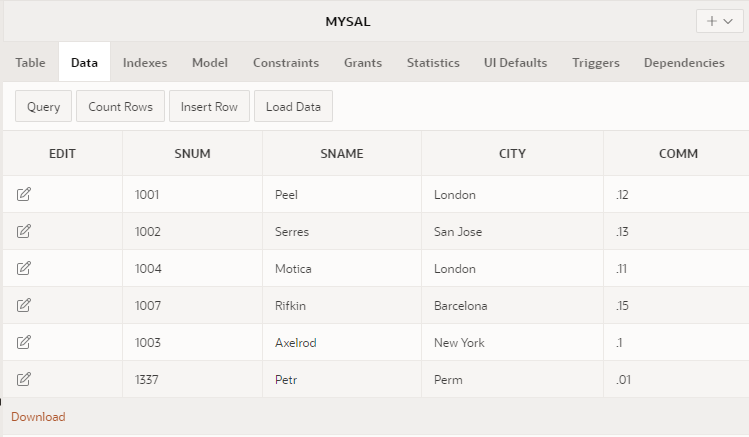
END;

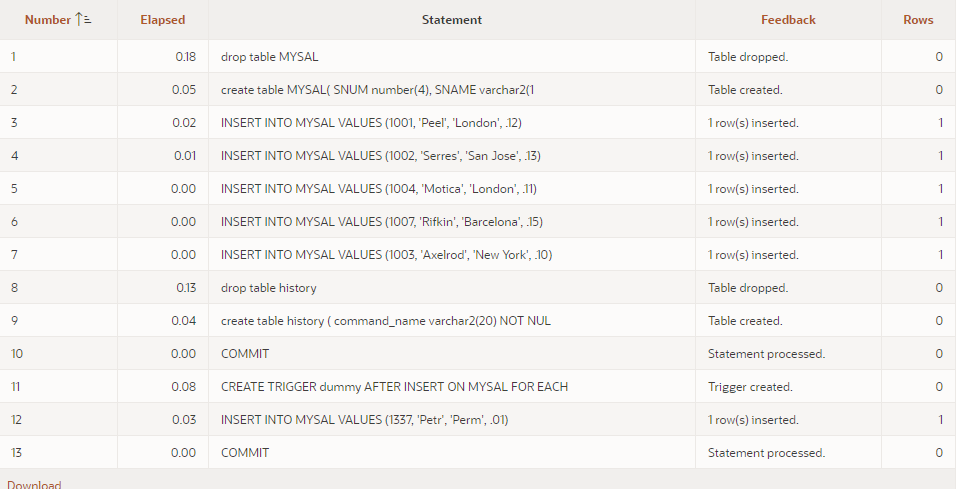
/

INSERT INTO MYSAL VALUES (1337, 'Petr', 'Perm', .01);

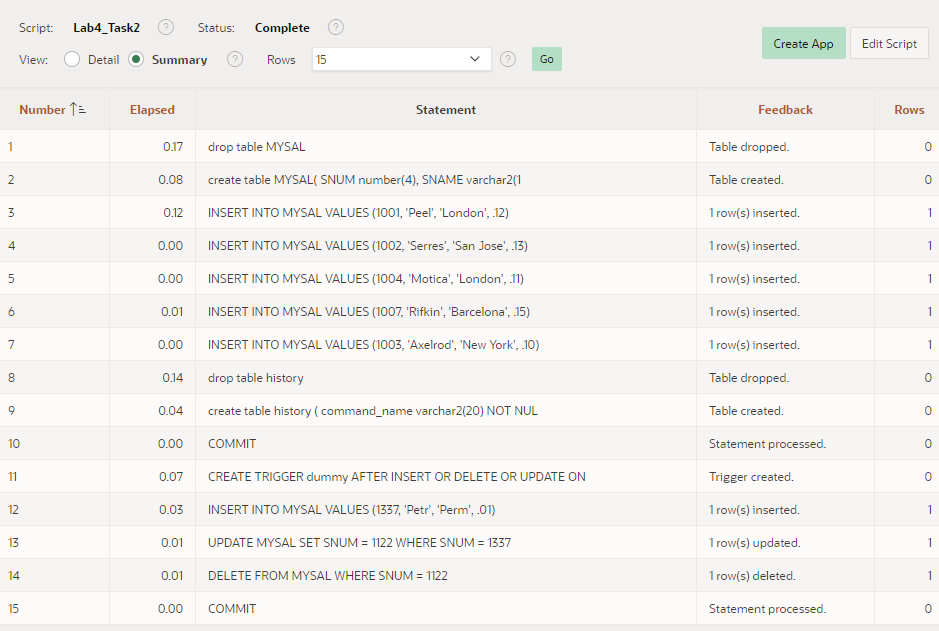
COMMIT;

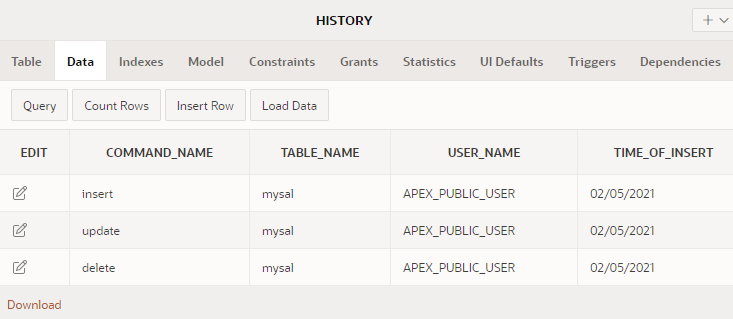
Для этого задания создаётся таблица MYSAL, которая полностью аналогична SAL, она заполняется данными. После этого происходит создание таблицы HISTORY, куда будут заноситься данные с помощью триггера. Триггер dummy срабатывает после работы команды INSERT над таблицей MYSAL.





1. Измените триггер из Задания 1 так, чтобы он регистрировал любые изменения таблицы, а не только вставку строк.





drop table MYSAL;

create table MYSAL(

    SNUM number(4),

    SNAME varchar2(10) NOT NULL,

    CITY  varchar2(10) NOT NULL,

    COMM  number(7,2) NOT NULL

);

INSERT INTO MYSAL VALUES (1001, 'Peel', 'London', .12);

INSERT INTO MYSAL VALUES (1002, 'Serres', 'San Jose', .13);

INSERT INTO MYSAL VALUES (1004, 'Motica', 'London', .11);

INSERT INTO MYSAL VALUES (1007, 'Rifkin', 'Barcelona', .15);

INSERT INTO MYSAL VALUES (1003, 'Axelrod', 'New York', .10);

drop table history;

create table history (

    command\_name varchar2(20) NOT NULL,

    table\_name varchar2(20) NOT NULL,

    user\_name varchar2(50) NOT NULL,

    time\_of\_insert  date

);

COMMIT;

CREATE TRIGGER dummy

    AFTER INSERT OR DELETE OR UPDATE ON MYSAL

    FOR EACH ROW

BEGIN

    IF INSERTING THEN

        INSERT INTO history (command\_name, table\_name, user\_name, time\_of\_insert)

        VALUES ('insert', 'mysal', user, sysdate);

    END IF;

    IF DELETING THEN

        INSERT INTO history (command\_name, table\_name, user\_name, time\_of\_insert)

        VALUES ('delete', 'mysal', user, sysdate);

    END IF;

    IF UPDATING THEN

        INSERT INTO history (command\_name, table\_name, user\_name, time\_of\_insert)

        VALUES ('update', 'mysal', user, sysdate);

    END IF;

END;

/

INSERT INTO MYSAL VALUES (1337, 'Petr', 'Perm', .01);

UPDATE MYSAL SET SNUM = 1122 WHERE SNUM = 1337;

DELETE FROM MYSAL WHERE SNUM = 1122;

COMMIT;

1. Создайте триггер, который не позволяет добавлять новых продавцов из Рима.

drop table MYSAL;

create table MYSAL(

    SNUM number(4),

    SNAME varchar2(10) NOT NULL,

    CITY  varchar2(10) NOT NULL,

    COMM  number(7,2) NOT NULL

);

INSERT INTO MYSAL VALUES (1001, 'Peel', 'London', .12);

INSERT INTO MYSAL VALUES (1002, 'Serres', 'San Jose', .13);

INSERT INTO MYSAL VALUES (1004, 'Motica', 'London', .11);

INSERT INTO MYSAL VALUES (1007, 'Rifkin', 'Barcelona', .15);

INSERT INTO MYSAL VALUES (1003, 'Axelrod', 'New York', .10);

drop table history;

create table history (

    command\_name varchar2(20) NOT NULL,

    table\_name varchar2(20) NOT NULL,

    user\_name varchar2(50) NOT NULL,

    time\_of\_insert  date

);

COMMIT;

CREATE TRIGGER dummy

    AFTER INSERT OR DELETE OR UPDATE ON MYSAL

    FOR EACH ROW

BEGIN

    IF INSERTING THEN

        INSERT INTO history (command\_name, table\_name, user\_name, time\_of\_insert)

        VALUES ('insert', 'mysal', user, sysdate);

    END IF;

    IF DELETING THEN

        INSERT INTO history (command\_name, table\_name, user\_name, time\_of\_insert)

        VALUES ('delete', 'mysal', user, sysdate);

    END IF;

    IF UPDATING THEN

        INSERT INTO history (command\_name, table\_name, user\_name, time\_of\_insert)

        VALUES ('update', 'mysal', user, sysdate);

    END IF;

END;

/

CREATE TRIGGER noRome

    BEFORE UPDATE OR INSERT ON MYSAL

    FOR EACH ROW

BEGIN

        IF :new.CITY = 'Rome' THEN raise\_application\_error(-20001, 'Rome is not allowed here!');

        END IF;

END;

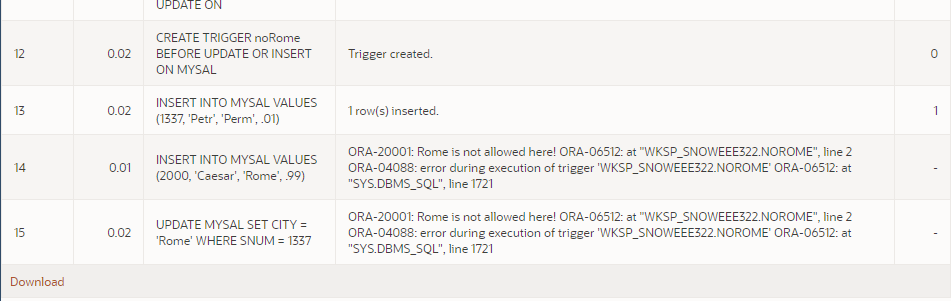
/

INSERT INTO MYSAL VALUES (1337, 'Petr', 'Perm', .01);

INSERT INTO MYSAL VALUES (2000, 'Caesar', 'Rome', .99);

UPDATE MYSAL SET CITY = 'Rome' WHERE SNUM = 1337;

COMMIT;



1. Создайте свои копии таблиц SAL, CUST и ORD. Создайте скрипт, в котором удалите все данные из созданных копий таблиц следующим образом: из первой и второй таблиц - командой DELETE, а из третьей - командой TRUNCATE. Затем отмените изменения командой ROLLBACK и выведите содержимое таблиц. Выполните скрипт и объясните результат.

DROP TABLE SAL\_COPY;

DROP TABLE CUST\_COPY;

DROP TABLE ORD\_COPY;

create table SAL\_COPY

  (SNUM number(4),

   SNAME varchar2(10) NOT NULL,

   CITY  varchar2(10) NOT NULL,

   COMM  number(7,2) NOT NULL);

--

--  создание таблицы заказчиков - CUST

--

create table CUST\_COPY

  (CNUM number(4),

   CNAME varchar2(10) NOT NULL,

   CITY  varchar2(10) NOT NULL,

   RATING number(3) NOT NULL,

   SNUM number(4));

--

--  создание таблицы заказов - ORD

--

create table ORD\_COPY

  (ONUM number(4),

   AMT  number(7,2) NOT NULL,

   ODATE date NOT NULL,

   CNUM number(4),

   SNUM number(4));

INSERT INTO sal\_COPY

  VALUES (1001, 'Peel', 'London', .12);

INSERT INTO sal\_COPY

  VALUES (1002, 'Serres', 'San Jose', .13);

INSERT INTO sal\_COPY

  VALUES (1004, 'Motica', 'London', .11);

INSERT INTO sal\_COPY

  VALUES (1007, 'Rifkin', 'Barcelona', .15);

INSERT INTO sal\_COPY

  VALUES (1003, 'Axelrod', 'New York', .10);

--

--  заполнение таблицы cust

--

INSERT INTO cust\_COPY

  VALUES (2001, 'Hoffman', 'London', 100, 1001);

INSERT INTO cust\_COPY

  VALUES (2002, 'Giovanni', 'Rome', 200, 1003);

INSERT INTO cust\_COPY

  VALUES (2003, 'Liu', 'San Jose', 200, 1002);

INSERT INTO cust\_COPY

  VALUES (2004, 'Grass', 'Berlin', 300, 1002);

INSERT INTO cust\_COPY

  VALUES (2006, 'Clemens', 'London', 100, 1001);

INSERT INTO cust\_COPY

  VALUES (2008, 'Cisneros', 'San Jose', 300, 1007);

INSERT INTO cust\_COPY

  VALUES (2007, 'Pereira', 'Rome', 100, 1004);

--

--  заполнение таблицы ord

--

INSERT INTO ord\_COPY

  VALUES (3001, 18.69,   to\_date('03.01.2010','dd.mm.yyyy'), 2008, 1007);

INSERT INTO ord\_COPY

  VALUES (3003, 767.19,  to\_date('03.01.2010','dd.mm.yyyy'), 2001, 1001);

INSERT INTO ord\_COPY

  VALUES (3002, 1900.11, to\_date('03.01.2010','dd.mm.yyyy'), 2007, 1004);

INSERT INTO ord\_COPY

  VALUES (3005, 5160.45, to\_date('03.01.2010','dd.mm.yyyy'), 2003, 1002);

INSERT INTO ord\_COPY

  VALUES (3006, 1098.16, to\_date('03.01.2010','dd.mm.yyyy'), 2008, 1007);

INSERT INTO ord\_COPY

  VALUES (3009, 1713.23, to\_date('04.01.2010','dd.mm.yyyy'), 2002, 1003);

INSERT INTO ord\_COPY

  VALUES (3007, 75.75,   to\_date('04.01.2010','dd.mm.yyyy'), 2004, 1002);

INSERT INTO ord\_COPY

  VALUES (3008, 4723.00, to\_date('05.01.2010','dd.mm.yyyy'), 2006, 1001);

INSERT INTO ord\_COPY

  VALUES (3010, 1309.95, to\_date('06.01.2010','dd.mm.yyyy'), 2004, 1002);

INSERT INTO ord\_COPY

  VALUES (3011, 9891.88, to\_date('06.01.2010','dd.mm.yyyy'), 2006, 1001);

--

COMMIT;

TRUNCATE TABLE ORD\_COPY;

DELETE FROM SAL\_COPY;

DELETE FROM CUST\_COPY;

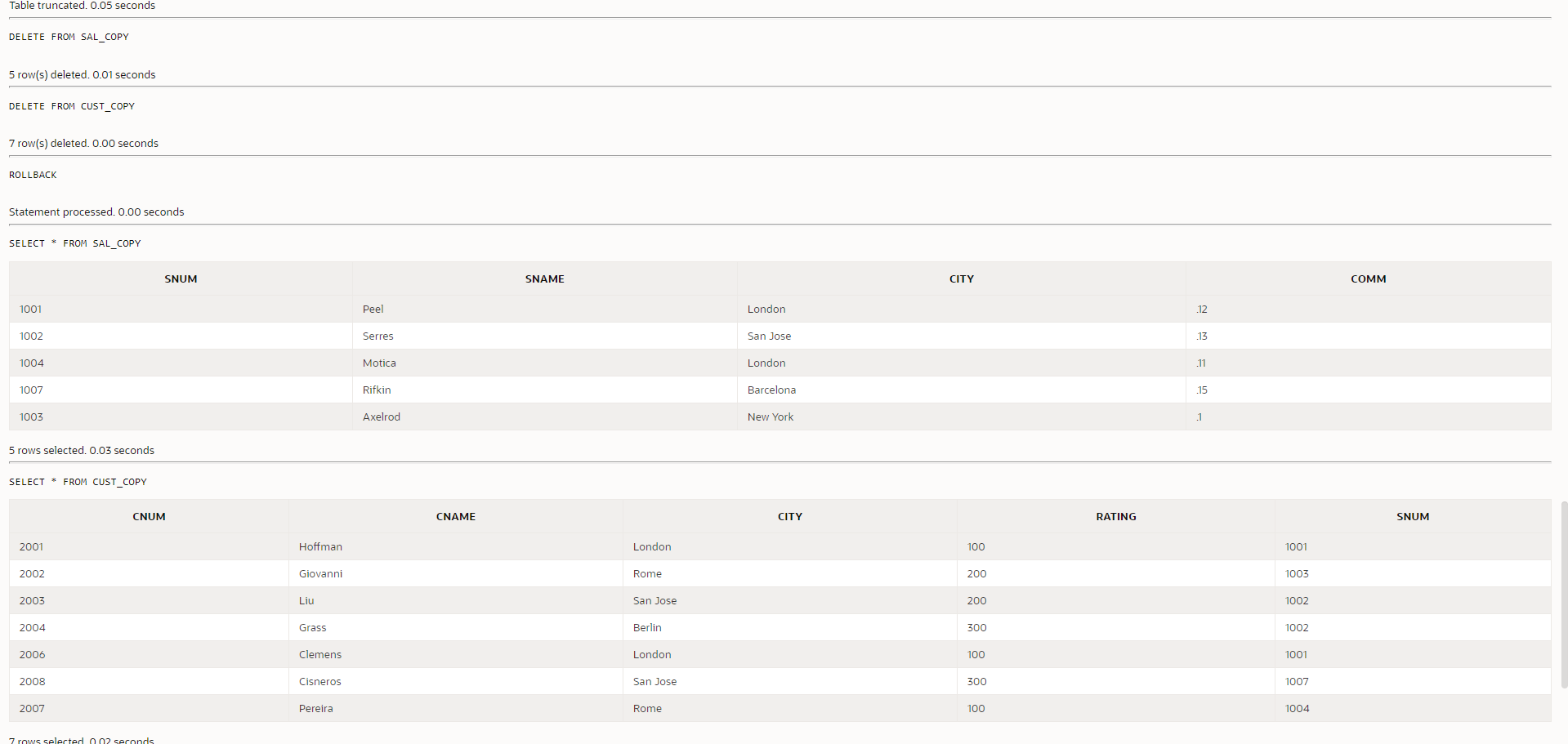
ROLLBACK;

SELECT \* FROM SAL\_COPY;

SELECT \* FROM CUST\_COPY;

SELECT \* FROM ORD\_COPY;

После TRUNCATE идёт неявная операция COMMIT, поэтому результат TRUNCATE нельзя откатить с помощью ROLLBACK. Если первые две операции будут DELETE, а за ними будет следовать TRUNCATE, то в таком случае транзакция будет подтверждена, и откат не будет возможен. Все записи в таблицах будут удалены. Если же сначала совершить TRUNCATE, а затем DELETE, то мы можем совершить откат двух последних операций DELETE.



1. Выдайте привилегию на чтение одной Вашей таблицы всем пользователям БД. Отнимите выданные Вами привилегии и посмотрите, как изменился набор привилегий.

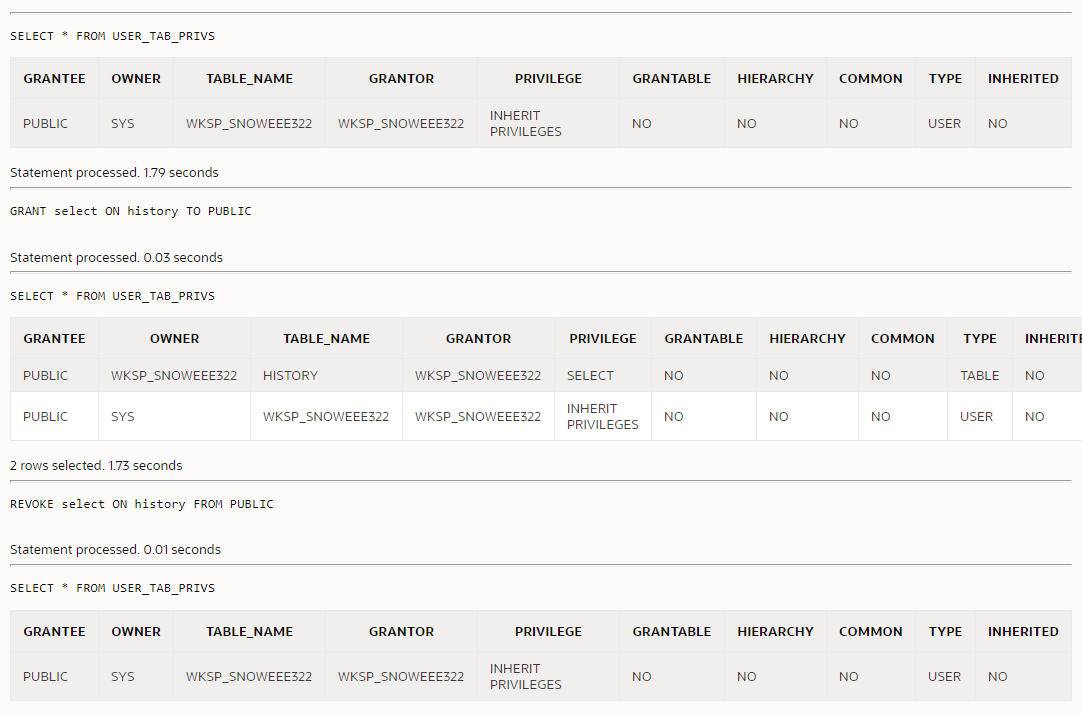
SELECT \* FROM USER\_TAB\_PRIVS;

GRANT select ON history TO PUBLIC;

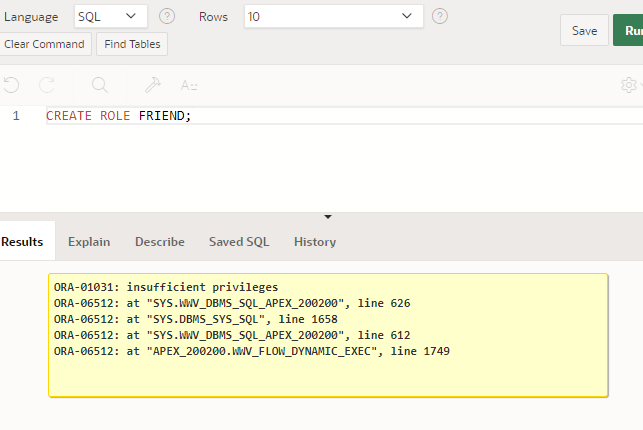
SELECT \* FROM USER\_TAB\_PRIVS;

REVOKE select ON history FROM PUBLIC;

SELECT \* FROM USER\_TAB\_PRIVS;



1. Создайте роль и присвойте ей привилегии на чтение и изменение вашей таблицы.



Последнее задание не было выполнено по причине недостаточных прав.