

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-911
Давыдов И.П.
Шумихин Д.В.

Проверила: Дьячкова И.С.

Новосибирск, 2023 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 trigger dimka;
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 dimka
    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 (228, 'Vovan', 'Kemerovo', .01);
COMMIT;
```

Number ↑	Elapsed	Statement	Feedback	Rows
1	0.07	drop trigger dimka	Trigger dropped.	0
2	0.10	drop table MYSAL	Table dropped.	0
3	0.04	create table MYSAL (SNUM number(4), SNAME varchar2(10) NOT NULL, CITY varchar2(10) NOT NULL, COMM number(7,2) NOT NULL)	Table created.	0
4	0.01	INSERT INTO MYSAL VALUES (1001, 'Peel', 'London', .12)	1 row(s) inserted.	1
5	0.00	INSERT INTO MYSAL VALUES (1002, 'Serres', 'San Jose', .13)	1 row(s) inserted.	1
6	0.00	INSERT INTO MYSAL VALUES (1004, 'Motica', 'London', .11)	1 row(s) inserted.	1
7	0.00	INSERT INTO MYSAL VALUES (1007, 'Rifkin', 'Barcelona', .15)	1 row(s) inserted.	1
8	0.00	INSERT INTO MYSAL VALUES (1003, 'Axelrod', 'New York', .10)	1 row(s) inserted.	1
9	0.08	drop table history	Table dropped.	0
10	0.04	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)	Table created.	0
11	0.01	COMMIT	Statement processed.	0
12	0.03	CREATE TRIGGER dimka 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;	Trigger created.	0
13	0.02	INSERT INTO MYSAL VALUES (228, 'Vovan', 'Kemerovo', .01)	1 row(s) inserted.	1
14	0.00	COMMIT	Statement processed.	0


HISTORY												
Table	Data	Indexes	Model	Constraints	Grants	Statistics	UI Defaults	Triggers	Dependencies	SQL	REST	Sample Queries
<div>Query</div> <div>Count Rows</div> <div>Insert Row</div> <div>Load Data</div>												
EDIT		COMMAND_NAME				TABLE_NAME		USER_NAME			TIME_OF_INSERT	
		update				mysal		APEX_PUBLIC_USER			02/14/2023	

Рис. 1 – Вывод таблицы истории

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

```
drop trigger dummy;
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 OR REPLACE 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 (228, 'Vovan', 'Kemerovo', .01);
UPDATE MYSAL SET SNUM = 1122 WHERE SNUM = 228;
DELETE FROM MYSAL WHERE SNUM = 1122;
COMMIT;

```

Number ↑	Elapsed	Statement	Feedback	Rows
1	0.09	drop trigger dummy	Trigger dropped.	0
2	0.11	drop table MYSAL	Table dropped.	0
3	0.05	create table MYSAL(SNUM number(4), SNAME varchar2(1	Table created.	0
4	0.03	INSERT INTO MYSAL VALUES (1001, 'Peel', 'London', 12)	1 row(s) inserted.	1
5	0.00	INSERT INTO MYSAL VALUES (1002, 'Serres', 'San Jose', 13)	1 row(s) inserted.	1
6	0.00	INSERT INTO MYSAL VALUES (1004, 'Motica', 'London', 11)	1 row(s) inserted.	1
7	0.00	INSERT INTO MYSAL VALUES (1007, 'Rifkin', 'Barcelona', 15)	1 row(s) inserted.	1
8	0.00	INSERT INTO MYSAL VALUES (1003, 'Axelrod', 'New York', 10)	1 row(s) inserted.	1
9	0.07	drop table history	Table dropped.	0
10	0.04	create table history (command_name varchar2(20) NOT NUL	Table created.	0
11	0.01	COMMIT	Statement processed.	0
12	0.04	CREATE OR REPLACE TRIGGER dummy AFTER INSERT OR DELETE O	Trigger created.	0
13	0.03	INSERT INTO MYSAL VALUES (228, 'Vovan', 'Kemerovo', .01)	1 row(s) inserted.	1
14	0.01	UPDATE MYSAL SET SNUM = 1122 WHERE SNUM = 228	1 row(s) updated.	1
15	0.00	DELETE FROM MYSAL WHERE SNUM = 1122	1 row(s) deleted.	1
16	0.00	COMMIT	Statement processed.	0




HISTORY												
Table	Data	Indexes	Model	Constraints	Grants	Statistics	UI Defaults	Triggers	Dependencies	SQL	REST	Sample Queries
Query		Count Rows		Insert Row		Load Data						
EDIT		COMMAND_NAME				TABLE_NAME		USER_NAME			TIME_OF_INSERT	
		insert				mysal		APEX_PUBLIC_USER			02/14/2023	
		update				mysal		APEX_PUBLIC_USER			02/14/2023	
		delete				mysal		APEX_PUBLIC_USER			02/14/2023	

Рис. 2 – Вывод таблицы истории

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

```

drop trigger dimka;
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 dimka
  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 (228, 'Vovan', 'Perm', .01);
INSERT INTO MYSAL VALUES (2000, 'Caesar', 'Rome', .99);
UPDATE MYSAL SET CITY = 'Rome' WHERE SNUM = 228;
COMMIT;

```

Number ↑	Elapsed	Statement	Feedback
1	0.03	drop trigger dimka	Trigger dropped.
2	0.10	drop table MYSAL	Table dropped.
3	0.05	create table MYSAL(SNUM number(4), SNAME varchar2(1	Table created.
4	0.04	INSERT INTO MYSAL VALUES (1001, 'Peel', 'London', 12)	1 row(s) inserted.
5	0.00	INSERT INTO MYSAL VALUES (1002, 'Series', 'San Jose', 15)	1 row(s) inserted.
6	0.01	INSERT INTO MYSAL VALUES (1004, 'Motica', 'London', 11)	1 row(s) inserted.
7	0.00	INSERT INTO MYSAL VALUES (1007, 'Rifkin', 'Barcelona', 15)	1 row(s) inserted.
8	0.00	INSERT INTO MYSAL VALUES (1003, 'Axelrod', 'New York', 10)	1 row(s) inserted.
9	0.05	drop table history	Table dropped.
10	0.05	create table history (command_name varchar2(20) NOT NUL	Table created.
11	0.01	COMMIT	Statement processed.
12	0.02	CREATE TRIGGER dimka AFTER INSERT OR DELETE OR UPDATE ON	Trigger created.
13	0.06	CREATE TRIGGER noRome BEFORE UPDATE OR INSERT ON MYSAL	Trigger created.
14	0.01	INSERT INTO MYSAL VALUES (228, 'Vovan', 'Perm', .01)	1 row(s) inserted.
15	0.01	INSERT INTO MYSAL VALUES (2000, 'Caesar', 'Rome', .99)	ORA-20001: Rome is not allowed here! ORA-06512: at "WKSP_SBDLESSON.NOROME", line 2 ORA-04088: error during execution of trigger 'WKSP_SBDLESSON.NOROME' ORA-06512: at "SYS.DBMS_SQL", line 1721
16	0.00	UPDATE MYSAL SET CITY = 'Rome' WHERE SNUM = 228	ORA-20001: Rome is not allowed here! ORA-06512: at "WKSP_SBDLESSON.NOROME", line 2 ORA-04088: error during execution of trigger 'WKSP_SBDLESSON.NOROME' ORA-06512: at "SYS.DBMS_SQL", line 1721
17	0.01	COMMIT	Statement processed.

Рис. 3.1 – Вывод ошибки, потому что нельзя добавлять новых продавцов с города Rome

Рис. 3.2 – Вывод таблицы истории

- 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;

```


Statement processed, 0.00 seconds

SELECT * FROM SAL_COPY

SNUM	SNAME	CITY	COMM
1001	Peel	London	.12
1002	Serres	San Jose	.13
1004	Motica	London	.11
1007	Rifkin	Barcelona	.15
1003	Axelrod	New York	.1

5 rows selected, 0.01 seconds

SELECT * FROM CUST_COPY

CNUM	CNAME	CITY	RATING	SNUM
2001	Hoffman	London	100	1001
2002	Giovanni	Rome	200	1003
2003	Liu	San Jose	200	1002
2004	Grass	Berlin	300	1002
2006	Clemens	London	100	1001
2008	Cisneros	San Jose	300	1007
2007	Pereira	Rome	100	1004

7 rows selected, 0.01 seconds

7 rows selected. 0.01 seconds

SELECT * FROM ORD_COPY

no data found

Рис. 4.1 – Вывод содержимого таблиц после ROLLBACK

- 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;
```

SELECT * FROM USER_TAB_PRIVS

GRANTEE	OWNER	TABLE_NAME	GRANTOR	PRIVILEGE	GRANTABLE	HIERARCHY	COMMON	TYPE	INHERITED
PUBLIC	WKSP_SBDLESSON	VIEW_1	WKSP_SBDLESSON	SELECT	NO	NO	NO	VIEW	NO
PUBLIC	SYS	WKSP_SBDLESSON	WKSP_SBDLESSON	INHERIT PRIVILEGES	NO	NO	NO	USER	NO

Рис. 5.1 – Набор привилегий до выдачи

GRANT select ON history TO PUBLIC

Statement processed, 0.01 seconds

SELECT * FROM USER_TAB_PRIVS

GRANTEE	OWNER	TABLE_NAME	GRANTOR	PRIVILEGE	GRANTABLE	HIERARCHY	COMMON	TYPE	INHERITED
PUBLIC	WKSP_SBDLESSON	VIEW_1	WKSP_SBDLESSON	SELECT	NO	NO	NO	VIEW	NO
PUBLIC	WKSP_SBDLESSON	HISTORY	WKSP_SBDLESSON	SELECT	NO	NO	NO	TABLE	NO
PUBLIC	SYS	WKSP_SBDLESSON	WKSP_SBDLESSON	INHERIT PRIVILEGES	NO	NO	NO	USER	NO

Рис. 5.2 – Набор привилегий после выдачи

```
REVOKE select ON history FROM PUBLIC
```

Statement processed. 0.02 seconds

```
SELECT * FROM USER_TAB_PRIVS
```

GRANTEE	OWNER	TABLE_NAME	GRANTOR	PRIVILEGE	GRANTABLE	HIERARCHY	COMMON	TYPE	INHERITED
PUBLIC	WKSP_SBDLESSON	VIEW_1	WKSP_SBDLESSON	SELECT	NO	NO	NO	VIEW	NO
PUBLIC	SYS	WKSP_SBDLESSON	WKSP_SBDLESSON	INHERIT PRIVILEGES	NO	NO	NO	USER	NO

Рис. 5.3 – Набор привилегий после отнятия

6. Используя представления словаря данных, определите все привилегии, которыми вы обладаете при проведении сеанса работы в Oracle APEX. Создайте роль и присвойте ей привилегии на чтение и изменение Вашей таблицы.

```
CREATE ROLE VOVKA
```

```
CREATE ROLE VOVKA
```

ORA-01031: insufficient privileges ORA-06512: at "SYS.WWV_DBMS_SQL_APEX_220200", line 828 ORA-06512: at "SYS.DBMS_SYS_SQL", line 1658 ORA-06512: at "SYS.WWV_DBMS_SQL_APEX_220200", line 813 ORA-06512: at "APEX_220200.WWV_FLOW_DYNAMIC_EXEC", line 2046

Рис. 6 – Невозможно создать роль, так как не хватает прав

Дополнительные задания

br13
1. Создайте триггер, который регистрирует только вставки и удаления строк в таблице Покупателей. При регистрации необходимо фиксировать тип команды, имя пользователя, дату и время.

```
drop trigger dummy;

drop table CUST_COPY;

--

--  создание таблицы заказчиков - CUST

--

create table CUST_COPY
  (CNUM number(4),
   CNAME varchar2(10) NOT NULL,
   CITY  varchar2(12) NOT NULL,
   RATING number(3) NOT NULL,
   SNUM number(4));

--

--  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);

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 ON cust_COPY
```

```
    FOR EACH ROW
```

```
BEGIN
```

```
    IF INSERTING THEN
```

```
        INSERT INTO history (command_name, table_name, user_name, time_of_insert)
```

```
        VALUES ('insert', 'cust_copy', user, sysdate);
```

```
    END IF;
```

```
    IF DELETING THEN
```

```
        INSERT INTO history (command_name, table_name, user_name, time_of_insert)
```

```
        VALUES ('delete', 'cust_copy', user, sysdate);
```

```
    END IF;
```

```
END;
```

```
/
```

```
INSERT INTO cust_COPY VALUES (2005, 'Igor', 'Novosibirsk', 500, 1008);
```

```
DELETE FROM cust_COPY WHERE SNUM = 1008;
```

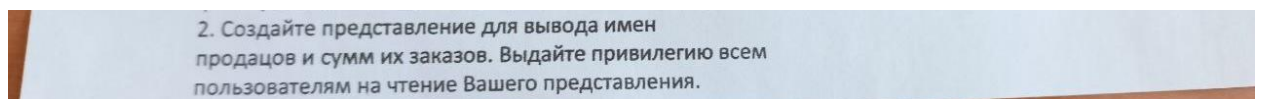
```
COMMIT;
```

Number ↑	Elapsed	Statement	Feedback
1	0.04	drop trigger dummy	Trigger dropped.
2	0.11	drop table CUST_COPY	Table dropped.
3	0.06	create table CUST_COPY (CNUM number(4), CNAME varchar2(Table created.
4	0.02	INSERT INTO cust_COPY VALUES (2001, 'Hoffman', 'London', 1	1 row(s) inserted.
5	0.00	INSERT INTO cust_COPY VALUES (2002, 'Giovanni', 'Rome', 20	1 row(s) inserted.
6	0.01	INSERT INTO cust_COPY VALUES (2003, 'Liu', 'San Jose', 200	1 row(s) inserted.
7	0.00	INSERT INTO cust_COPY VALUES (2004, 'Grass', 'Berlin', 300	1 row(s) inserted.
8	0.00	INSERT INTO cust_COPY VALUES (2006, 'Clemens', 'London', 1	1 row(s) inserted.
9	0.00	INSERT INTO cust_COPY VALUES (2008, 'Cisneros', 'San Jose'	1 row(s) inserted.
10	0.00	INSERT INTO cust_COPY VALUES (2007, 'Pereira', 'Rome', 100	1 row(s) inserted.
11	0.06	drop table history	Table dropped.
12	0.05	create table history (command_name varchar2(20) NOT NUL	Table created.
13	0.01	COMMIT	Statement processed.
14	0.03	CREATE TRIGGER dummy AFTER INSERT OR DELETE ON cust_COPY	Trigger created.
15	0.03	INSERT INTO cust_COPY VALUES (2005, 'Igor', 'Novosibirsk', 5	1 row(s) inserted.
16	0.01	DELETE FROM cust_COPY WHERE SNUM = 1008	1 row(s) deleted.
17	0.00	COMMIT	Statement processed.

Рис. 1.1 – Всё отработало хорошо

HISTORY												
Table	Data	Indexes	Model	Constraints	Grants	Statistics	UI Defaults	Triggers	Dependencies	SQL	REST	Sample Queries
Query		Count Rows		Insert Row		Load Data						
EDIT		COMMAND_NAME			TABLE_NAME			USER_NAME			TIME_OF_INSERT	
		insert			cust_copy			APEX_PUBLIC_USER			02/14/2023	
		delete			cust_copy			APEX_PUBLIC_USER			02/14/2023	

Рис. 1.2 – Вывод таблицы истории



```
CREATE OR REPLACE VIEW view_1 (salname, sum_amt) AS SELECT sal.sname, sum(ord.amt)
FROM sal, ord WHERE ord.snum = sal.snum GROUP BY sal.sname;
```

```
GRANT select ON view_1 TO PUBLIC;
```

Number ↑	Elapsed	Statement	Feedback
1	0.04	CREATE OR REPLACE VIEW view_1 (salname, sum_amt) AS SELECT s	View created.
2	0.02	GRANT select ON view_1 TO PUBLIC	Statement processed.

Рис. 2.1 – Всё отработало хорошо, всем пользователям разрешено просматривать представление

VIEW_1	
View	Code
Data	Grants
UI Defaults	Dependencies
SQL	REST
Query	Count Rows
Insert Row	Load Data
SALNAME	SUM_AMT
Axelrod	1713.23
Motica	1900.11
Peel	15382.07
Rifkin	1116.85
Serres	6546.15

Рис. 2.2 – Наше представление для вывода имен продавцов и сумм их заказов