

TRIGGER CODES

HU22CSEN0100287

Sai Ganesh Eswaraprasad

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\H-CSE-506-22> sqlplus hu22csen0100287/seswarap@//172.21.51.130:1521/orclpdb

SQL*Plus: Release 23.0.0.0.0 - Production on Wed Oct 9 15:27:00 2024
Version 23.4.0.24.05

Copyright (c) 1982, 2024, Oracle. All rights reserved.

Last Successful login time: Wed Oct 09 2024 10:20:33 +05:30

Connected to:
Oracle Database 21c Enterprise Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0

SQL> CREATE TABLE emp (employee_id NUMBER PRIMARY KEY, first_name VARCHAR2(50), salary NUMBER);
Table created.

SQL> create table salary_log1 (employee_id number, old_salary number, new_salary number, change_date date);
Table created.

SQL> create table customer1 (customer_id number primary key, customer_name varchar2(50));
Table created.

SQL> create table student1 (student_id number primary key, name varchar2(50), marks number, grade varchar2(2));
Table created.

SQL> create table voter (voter_id number primary key, name varchar2(50), age number);
Table created.

SQL> insert into emp (employee_id, first_name, salary) values (1, 'john', 50000);
1 row created.

SQL> insert into emp (employee_id, first_name, salary) values (2, 'jane', 60000);
1 row created.

SQL>
SQL> insert into customer1 (customer_id, customer_name) values (1, 'acme corp');
1 row created.

SQL> insert into customer1 (customer_id, customer_name) values (2, 'globex inc');
1 row created.

SQL>
SQL> insert into salary_log1 (employee_id, old_salary, new_salary, change_date) values (1, null, 50000, sysdate);
1 row created.

SQL> insert into salary_log1 (employee_id, old_salary, new_salary, change_date) values (2, null, 60000, sysdate);
1 row created.

SQL>
SQL> insert into student1 (student_id, name, marks, grade) values (1, 'alice', 95, 'a');
1 row created.

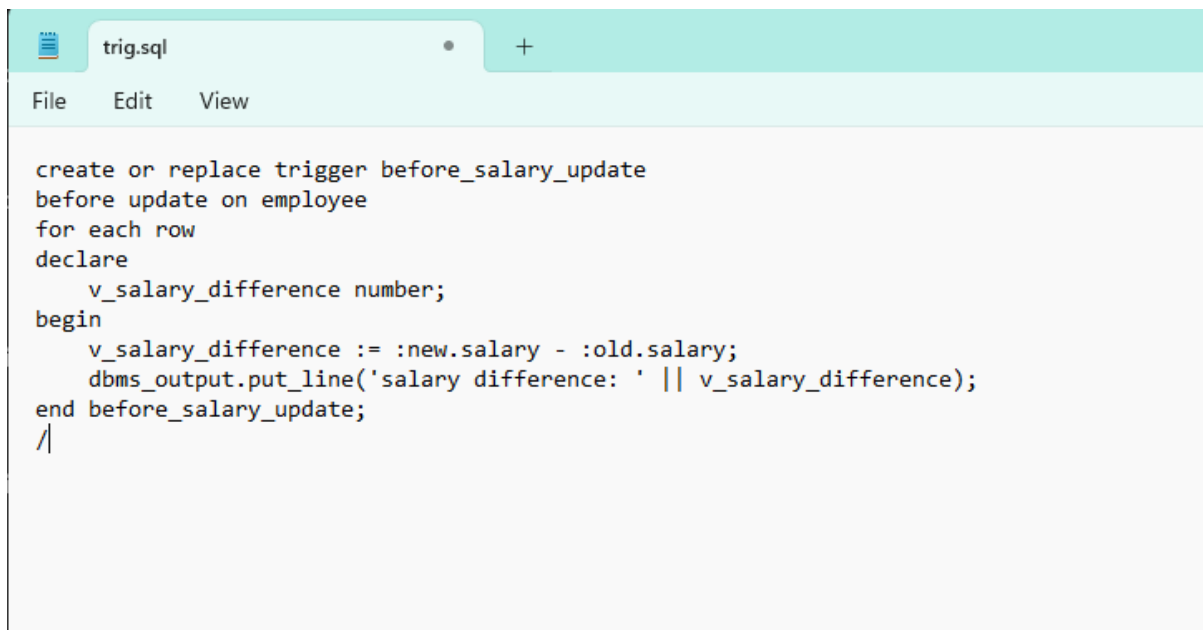
SQL> insert into student1 (student_id, name, marks, grade) values (2, 'bob', 88, 'b');
1 row created.

SQL>
SQL> insert into voter (voter_id, name, age) values (1, 'charlie', 25);
1 row created.

SQL> insert into voter (voter_id, name, age) values (2, 'david', 30);
1 row created.

SQL> ed trig.sql

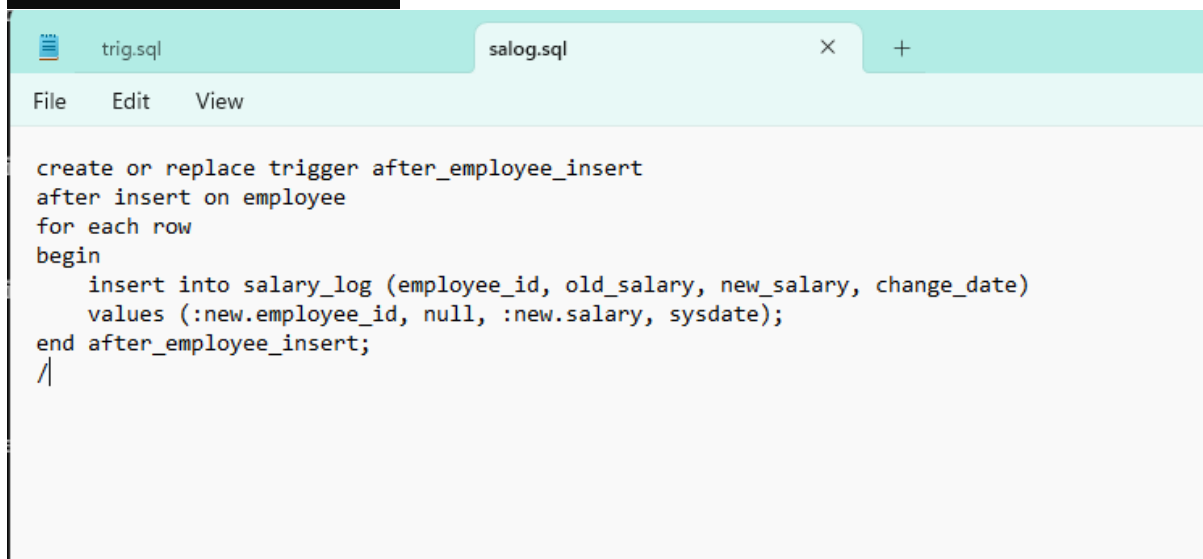
SQL> @ trig.sql
```



```
create or replace trigger before_salary_update
before update on employee
for each row
declare
    v_salary_difference number;
begin
    v_salary_difference := :new.salary - :old.salary;
    dbms_output.put_line('salary difference: ' || v_salary_difference);
end before_salary_update;
/
```

```
SQL> ed sallog.sql
```

```
SQL> @ sallog.sql
```

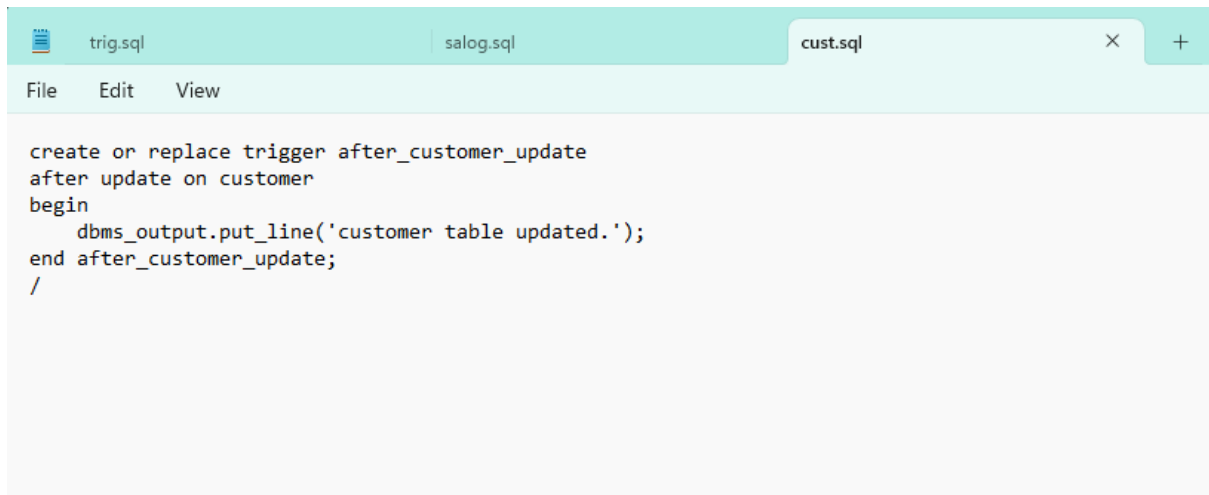


```
create or replace trigger after_employee_insert
after insert on employee
for each row
begin
    insert into salary_log (employee_id, old_salary, new_salary, change_date)
    values (:new.employee_id, null, :new.salary, sysdate);
end after_employee_insert;
/
```

```
SQL> ed cust.sql
```

```
SQL> @ cust.sql
```

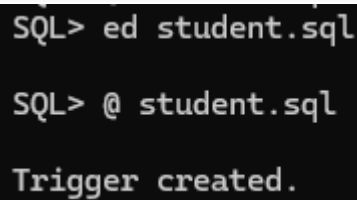
```
Trigger created.
```



The screenshot shows a SQL editor window with three tabs: trig.sql, salog.sql, and cust.sql. The trig.sql tab is active, displaying the following SQL code:

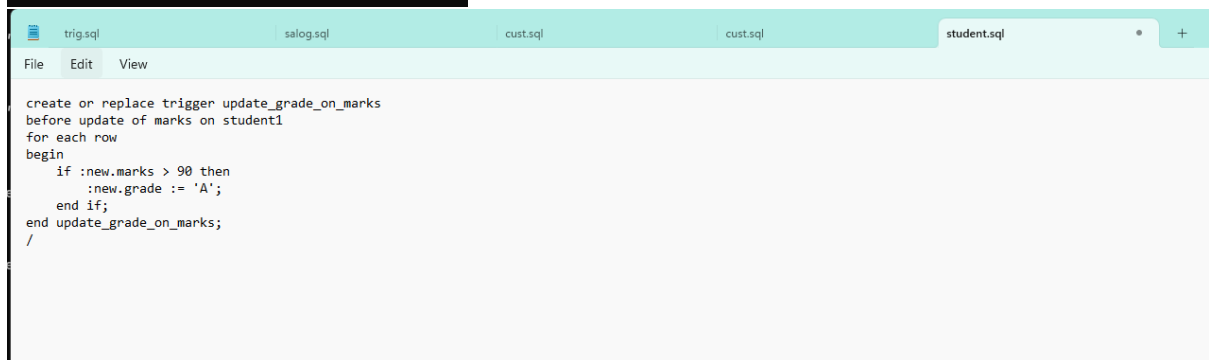
```
create or replace trigger after_customer_update
after update on customer
begin
    dbms_output.put_line('customer table updated.');
```

The code is partially visible, ending with 'end after_customer_update;' and a forward slash on the next line.



A terminal window showing the following commands and output:

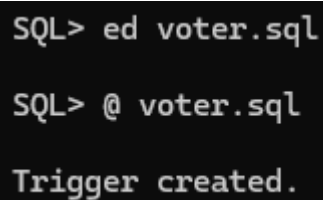
```
SQL> ed student.sql
SQL> @ student.sql
Trigger created.
```



The screenshot shows a SQL editor window with five tabs: trig.sql, salog.sql, cust.sql, cust.sql, and student.sql. The student.sql tab is active, displaying the following SQL code:


```
create or replace trigger update_grade_on_marks
before update of marks on student1
for each row
begin
    if :new.marks > 90 then
        :new.grade := 'A';
    end if;
end update_grade_on_marks;
```

The code is partially visible, ending with a forward slash on the next line.



A terminal window showing the following commands and output:

```
SQL> ed voter.sql
SQL> @ voter.sql
Trigger created.
```



The screenshot shows a SQL editor window with seven tabs: trig.sql, salog.sql, cust.sql, cust.sql, student.sql, student.sql, and voter.sql. The voter.sql tab is active, displaying the following SQL code:

```
create or replace trigger check_voting_eligibility
before insert on voter
for each row
begin
    if :new.age < 18 then
        raise_application_error(-20001, 'age must be 18 or older to be eligible to vote.');
```

The code is partially visible, ending with 'end check_voting_eligibility;' and a forward slash on the next line.