Oracle Procedures

A procedure is a group of PL/SQL statements that can be called by name. The call specification (sometimes called call spec) specifies a java method or a third-generation language routine so that it can be called from SQL and PL/SQL.

Parameter

`IN`

An `IN` parameter is read-only. You can reference an `IN` parameter inside a procedure, but you cannot change its value. Oracle uses `IN` as the default mode. It means that if you don’t specify the mode for a parameter explicitly, Oracle will use the `IN` mode.

`OUT`

An `OUT` parameter is writable. Typically, you set a returned value for the `OUT` parameter and return it to the calling program. Note that a procedure ignores the value that you supply for an `OUT` parameter.

`INOUT`

An `INOUT` parameter is both readable and writable. The procedure can read and modify it.

Note that `OR REPLACE` option allows you to overwrite the current procedure with the new code.

Create procedure

```
DROP Table Employee;

CREATE TABLE Employee (
  Id INT,
  Name VARCHAR(15),
  Salary NUMBER(8, 2)
);
```

```
create or replace procedure Insert_Employee
(Id INT,Name VARCHAR,Salary NUMBER)
is
begin
```

```
INSERT INTO Employee VALUES(Id, Name,Salary);
end;

begin
   Insert_Employee(1002, 'Smith', 45000);
end;

select * from Employee;

begin
   Insert_Employee(1003, 'STEVE', 45000);
end;

select * from Employee;









//-----------------------------------------------------

CREATE OR REPLACE PROCEDURE Update_Salary
(
   i INT,s NUMBER
)
IS
BEGIN
  UPDATE Employee
  SET salary=s
  WHERE id=i;
  COMMIT;
  EXCEPTION
  WHEN OTHERS THEN
  DBMS_OUTPUT.PUT_LINE (SQLCODE);
  DBMS_OUTPUT.PUT_LINE (SQLERRM);
END;


begin
   Update_Salary(1002,1300);
end;
select * from Employee;

//-----------------------------------------------

CREATE OR REPLACE PROCEDURE SHOW_EMP
(
    P_EMP_ID Employee.id%TYPE,
    P_F_NAME OUT Employee.name%TYPE,
    P_SAL OUT Employee.salary%TYPE
 )
IS
BEGIN
   SELECT name, salary
   INTO P_F_NAME, P_SAL
   FROM
```

```
   Employee
   WHERE id=P_EMP_ID;
   EXCEPTION
   WHEN OTHERS THEN
   DBMS_OUTPUT.PUT_LINE(SQLCODE);
   DBMS_OUTPUT.PUT_LINE(SQLERRM);
END;

//-----------------------------------------------------------------------

CREATE OR REPLACE PROCEDURE SHOW_EMP
(
    P_EMP_ID Employee.id%TYPE,
    P_F_NAME OUT Employee.name%TYPE,
    P_SAL OUT Employee.salary%TYPE
 )
IS
BEGIN
   SELECT name, salary
   INTO P_F_NAME, P_SAL
   FROM
   Employee
   WHERE id=P_EMP_ID;
   EXCEPTION
   WHEN OTHERS THEN
   DBMS_OUTPUT.PUT_LINE(SQLCODE);
   DBMS_OUTPUT.PUT_LINE(SQLERRM);
END;



DECLARE
V_FIRST_NAME Employee.id%TYPE;
V_SAL Employee.salary%TYPE;
BEGIN
  SHOW_EMP(105,V_FIRST_NAME,V_SAL );
  DBMS_OUTPUT.PUT_LINE(V_FIRST_NAME);
  DBMS_OUTPUT.PUT_LINE(V_SAL);
END;



///-------------------------------------------------------
DROP TABLE PRODUCTS;
CREATE TABLE PRODUCTS
(
  PROD_ID NUMBER,
  PROD_NAME VARCHAR2(20),
  PROD_TYPE VARCHAR2(20),
  CONSTRAINT PRODUCTS_PK PRIMARY KEY (PROD_ID)
);

CREATE OR REPLACE PROCEDURE ADD_PRODUCTS
(
   P_PROD_ID NUMBER,
   P_PROD_NAME VARCHAR2:='UNKNOWN',
   P_PROD_TYPE VARCHAR2 DEFAULT 'Unknown'
)
```

```
IS
BEGIN
  INSERT INTO PRODUCTS VALUES (P_PROD_ID, P_PROD_NAME ,P_PROD_TYPE);
  DBMS_OUTPUT.PUT_LINE (P_PROD_ID||' '||P_PROD_NAME||'  INSERTED ' );
  COMMIT;
  EXCEPTION
  WHEN OTHERS THEN
  DBMS_OUTPUT.PUT_LINE ('ERROR IN INSERT '||P_PROD_ID||' '||P_PROD_NAME);
  DBMS_OUTPUT.PUT_LINE (SQLCODE);
  DBMS_OUTPUT.PUT_LINE (SQLERRM);
END;


BEGIN
ADD_PRODUCTS (10,'Bajaj');
ADD_PRODUCTS (10,'Laptop');
ADD_PRODUCTS (20,'Samsung');
END;

SELECT * FROM PRODUCTS;
```

```
create or replace procedure Insert_Employee
(Id INT,Name VARCHAR,Salary NUMBER)
is
begin
INSERT INTO Employee VALUES(Id, Name,Salary);
end;

begin
   Insert_Employee(1002, 'Smith', 45000);
end;

select * from Employee;
```

```
begin
   Insert_Employee(1003, 'STEVE', 45000);
end;

select * from Employee;









//-----------------------------------------------------

CREATE OR REPLACE PROCEDURE Update_Salary
(
   i INT,s NUMBER
)
IS
BEGIN
  UPDATE Employee
  SET salary=s
  WHERE id=i;
  COMMIT;
  EXCEPTION
  WHEN OTHERS THEN
  DBMS_OUTPUT.PUT_LINE (SQLCODE);
  DBMS_OUTPUT.PUT_LINE (SQLERRM);
END;


begin
   Update_Salary(1002,1300);
end;
select * from Employee;

//-----------------------------------------------

CREATE OR REPLACE PROCEDURE SHOW_EMP
(
    P_EMP_ID Employee.id%TYPE,
    P_F_NAME OUT Employee.name%TYPE,
    P_SAL OUT Employee.salary%TYPE
 )
IS
BEGIN
   SELECT name, salary
   INTO P_F_NAME, P_SAL
   FROM
   Employee
   WHERE id=P_EMP_ID;
   EXCEPTION
   WHEN OTHERS THEN
   DBMS_OUTPUT.PUT_LINE(SQLCODE);
   DBMS_OUTPUT.PUT_LINE(SQLERRM);
END;

//-------------------------------------------------------------------------
```

```
CREATE OR REPLACE PROCEDURE SHOW_EMP
(
    P_EMP_ID Employee.id%TYPE,
    P_F_NAME OUT Employee.name%TYPE,
    P_SAL OUT Employee.salary%TYPE
 )
IS
BEGIN
   SELECT name, salary
   INTO P_F_NAME, P_SAL
   FROM
   Employee
   WHERE id=P_EMP_ID;
   EXCEPTION
   WHEN OTHERS THEN
   DBMS_OUTPUT.PUT_LINE(SQLCODE);
   DBMS_OUTPUT.PUT_LINE(SQLERRM);
END;



DECLARE
V_FIRST_NAME Employee.id%TYPE;
V_SAL Employee.salary%TYPE;
BEGIN
  SHOW_EMP(105,V_FIRST_NAME,V_SAL );
  DBMS_OUTPUT.PUT_LINE(V_FIRST_NAME);
  DBMS_OUTPUT.PUT_LINE(V_SAL);
END;



///-------------------------------------------------------
DROP TABLE PRODUCTS;
CREATE TABLE PRODUCTS
(
  PROD_ID NUMBER,
  PROD_NAME VARCHAR2(20),
  PROD_TYPE VARCHAR2(20),
  CONSTRAINT PRODUCTS_PK PRIMARY KEY (PROD_ID)
);

CREATE OR REPLACE PROCEDURE ADD_PRODUCTS
(
   P_PROD_ID NUMBER,
   P_PROD_NAME VARCHAR2:='UNKNOWN',
   P_PROD_TYPE VARCHAR2 DEFAULT 'Unknown'
)
IS
BEGIN
  INSERT INTO PRODUCTS VALUES (P_PROD_ID, P_PROD_NAME ,P_PROD_TYPE);
  DBMS_OUTPUT.PUT_LINE (P_PROD_ID||' '||P_PROD_NAME||'  INSERTED ' );
  COMMIT;
  EXCEPTION
  WHEN OTHERS THEN
  DBMS_OUTPUT.PUT_LINE ('ERROR IN INSERT '||P_PROD_ID||' '||P_PROD_NAME);
  DBMS_OUTPUT.PUT_LINE (SQLCODE);
```

```
  DBMS_OUTPUT.PUT_LINE (SQLERRM);
END;


BEGIN
ADD_PRODUCTS (10,'Bajaj');
ADD_PRODUCTS (10,'Laptop');
ADD_PRODUCTS (20,'Samsung');
END;

SELECT * FROM PRODUCTS;
```

Drop Procedore

```
DROP PROCEDURE Insert_Employee;
```