

Procedure: Syntax

```
CREATE [OR REPLACE] PROCEDURE procedure_name
[(argument1 [mode1] datatype1,
 argument2 [mode2] datatype2,
 . . .)]
IS|AS
procedure_body;
```

ORACLE

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

Procedure: Syntax

The slide shows the syntax for creating procedures. In the syntax:

procedure_name	Is the name of the procedure to be created
argument	Is the name given to the procedure parameter. Every argument is associated with a mode and data type. You can have any number of arguments separated by commas.
mode	Mode of argument: IN (default) OUT IN OUT
datatype	Is the data type of the associated parameter. The data type of parameters cannot have explicit size; instead, use %TYPE.
Procedure_body	Is the PL/SQL block that makes up the code

Creating a Procedure

```
***
CREATE TABLE dept AS SELECT * FROM departments;
CREATE PROCEDURE add_dept IS
  v_dept_id dept.department_id%TYPE;
  v_dept_name dept.department_name%TYPE;
BEGIN
  v_dept_id:=280;
  v_dept_name:='ST-Curriculum';
  INSERT INTO dept(department_id,department_name)
  VALUES (v_dept_id,v_dept_name);
  DBMS_OUTPUT.PUT_LINE(' Inserted ' || SQL%ROWCOUNT
  || ' row ');
END;
```

ORACLE

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

Creating a Procedure

In the code example, the add_dept procedure inserts a new department with department ID 280 and department name ST-Curriculum.

In addition, the example shows the following:

The declarative section of a procedure starts immediately after the procedure declaration and does not begin with the DECLARE keyword.

The procedure declares two variables, dept_id and dept_name.

The procedure uses the implicit cursor attribute or the SQL%ROWCOUNT SQL attribute to verify that the row was successfully inserted. A value of 1 should be returned in this case.

Note: See the following page for more notes on the example.

Procedure: Example

Note

When you create any object, the entries are made to the user_objects table.

When the code in the slide is executed successfully, you can check the user_objects table for the new objects by issuing the following command:

```
SELECT object_name,object_type FROM user_objects;
```

Results		
Script Output Explain Autotrace		
Results:		
	OBJECT_NAME	OBJECT_TYPE
41	COPY_EMP	TABLE
42	DEPT	TABLE
43	GREET	PROCEDURE
44	ADD_DEPT	PROCEDURE
45	MY_SEQ	SEQUENCE

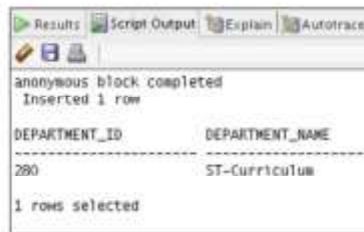
The source of the procedure is stored in the user_source table. You can check the source for the procedure by issuing the following command:

```
SELECT * FROM user_source WHERE name='ADD_DEPT';
```

Results					
Script Output Explain Autotrace DBMS Output OWA Output					
Results:					
	NAME	TYPE	LINE	TEXT	
1	ADD_DEPT	PROCEDURE	1	PROCEDURE add_dept IS	
2	ADD_DEPT	PROCEDURE	2	v_dept_id dept.department_id%TYPE;	
3	ADD_DEPT	PROCEDURE	3	v_dept_name dept.department_name%TYPE;	
4	ADD_DEPT	PROCEDURE	4	BEGIN	
5	ADD_DEPT	PROCEDURE	5	v_dept_id:=280;	
6	ADD_DEPT	PROCEDURE	6	v_dept_name:='ST-Curriculum';	
7	ADD_DEPT	PROCEDURE	7	INSERT INTO dept(department_id,department_name)	
8	ADD_DEPT	PROCEDURE	8	VALUES(v_dept_id,v_dept_name);	
9	ADD_DEPT	PROCEDURE	9	DBMS_OUTPUT.PUT_LINE(' Inserted ' SQL%ROWCOUNT ' row');	
10	ADD_DEPT	PROCEDURE	10	END;	

Invoking a Procedure

```
***  
BEGIN  
  add_dept;  
END;  
/  
SELECT department_id, department_name FROM dept  
WHERE department_id=280;
```



The screenshot shows the 'Results' tab of an Oracle SQL Developer window. It displays the message 'anonymous block completed' and 'Inserted 1 row'. Below this, a table shows the result of the SELECT statement: one row with DEPARTMENT_ID 280 and DEPARTMENT_NAME ST-Curriculum. At the bottom, it states '1 rows selected'.

DEPARTMENT_ID	DEPARTMENT_NAME
280	ST-Curriculum

ORACLE

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

Invoking the Procedure

The slide shows how to invoke a procedure from an anonymous block. You must include the call to the procedure in the executable section of the anonymous block. Similarly, you can invoke the procedure from any application, such as a Forms application or a Java application. The SELECT statement in the code checks to see whether the row was successfully inserted.

You can also invoke a procedure with the SQL statement CALL <procedure_name>.