Substitution Variables

What is a substitution variables (&)

Use substitution variables to:
Temporarily store values with single-ampersand (&) and double-ampersand (&&) substitution

Use substitution variables to supplement the following:
WHERE conditions
ORDER BY clauses
Column expressions
Table names
Entire SELECT statements

```
SELECT employee_id, last_name, salary, department_id
FROM EMPLOYEES
WHERE EMPLOYEE_ID = &EMPLOYEE_NUM; -- when using single & the variable will be discareded after is used
```

Use ' ' when using varchar

```
SELECT employee_id, first_name, last_name, salary, department_id
FROM EMPLOYEES
WHERE first_name = '&ename'
ORDER BY 2;
```

Another way to handle varchar to write the ' ' included in variable example 'Adam'

```
SELECT employee_id, first_name, last_name, salary, department_id
FROM EMPLOYEES
WHERE first_name = &ename
ORDER BY 2;
```

Specifying column names, expressions, and text
&column_name=salary &condition=salary>10000 &ORDER_COLUMN=employee_id

```
SELECT employee_id, last_name, job_id, &column_name
FROM EMPLOYEES
WHERE &condition
ORDER BY &ORDER_COLUMN;
```

DEFINE / UNDEFINE

Use the DEFINE command to create and assign a value to a variable
Use the UNDEFINE command to remove a variable

```
DEFINE EMPLOYEE_NUM = 200; -- this variable is valid for the session only

SELECT employee_id, last_name, salary, department_id
FROM employees
WHERE EMPLOYEE_ID = &EMPLOYEE_NUM;

UNDEFINE EMPLOYEE_NUM
```

ACCEPT / PROMPT

You can change the prompt message as follow
but it should executed as script

```
ACCEPT DEPT_ID PROMPT 'please enter dept id';
SELECT *
FROM EMPLOYEES
WHERE DEPARTMENT_ID = &DEPT_ID;

UNDEFINE DEPT_ID
```

```
ACCEPT emp_from PROMPT 'please enter EMPLOYEE from ';
ACCEPT emp_to PROMPT 'please enter EMPLOYEE to ';
SELECT * FROM EMPLOYEES
WHERE EMPLOYEE_ID BETWEEN &emp_from AND &emp_to;

UNDEFINE emp_from
UNDEFINE emp_to
```

Double-Ampersand / SET VERIFY / SET DEFINE

Using the Double-Ampersand Substitution Variable
The && define the variable and assign the value in the same time, then you can re-use it again

```
SELECT *
FROM DEPARTMENTS
WHERE DEMPARTMENT_ID =&& p;

UNDEFINE p
```

```
SELECT employee_id, last_name, job_id, &&column_name -- this =define column_namne
FROM EMPLOYEES
ORDER BY &COLUMN_NAME;

UNDEFINE column_name;
```

Use the VERIFY command to toggle the display of the substitution variable, both before and after SQL Developer replaces substitution variables with values:

Also should be executed as script

```
SET VERIFY ON
SELECT employee_id, last_name, salary
FROM employees
WHERE EMPLOYEES_ID = &E_NUM;
```

Using set define off
We use it to prevent oracle server to display the prompt window for specific reasons

```
SELECT *
FROM DEPARTMENTS
WHERE DEPARTMENT_NAME LIKE '%&t%'; -- here the oracle server suppose that & is variable
```

So we use SET DEFINE OFF

```
SET DEFINE OFF;

SELECT * FROM DEPARTMENTS
WHERE DEPARTMENT_NAME LIKE '%&t%';
```