

----- Handling the exception-----

```
declare
  v_name varchar2(6);
begin
  select first_name into v_name from employees where employee_id = 50;
  dbms_output.put_line('Hello');
exception
  when no_data_found then
    dbms_output.put_line('There is no employee with the selected id');
end;
```

----- Handling multiple exceptions-----

```
declare
  v_name varchar2(6);
  v_department_name varchar2(100);
begin
  select first_name into v_name from employees where employee_id = 100;
  select department_id into v_department_name from employees where first_name = v_name;
  dbms_output.put_line('Hello '|| v_name || '. Your department id is : '|| v_department_name );
exception
  when no_data_found then
    dbms_output.put_line('There is no employee with the selected id');
  when too_many_rows then
    dbms_output.put_line('There are more than one employees with the name '|| v_name);
    dbms_output.put_line('Try with a different employee');
end;
```

----- when others then example-----

```
declare
  v_name varchar2(6);
  v_department_name varchar2(100);
begin
  select first_name into v_name from employees where employee_id = 103;
  select department_id into v_department_name from employees where first_name = v_name;
  dbms_output.put_line('Hello '|| v_name || '. Your department id is : '|| v_department_name );
exception
  when no_data_found then
    dbms_output.put_line('There is no employee with the selected id');
  when too_many_rows then
    dbms_output.put_line('There are more than one employees with the name '|| v_name);
    dbms_output.put_line('Try with a different employee');
  when others then
    dbms_output.put_line('An unexpected error happened. Connect with the programmer..');
end;
```

----- sqlerrm & sqlcode example-----

```
declare
  v_name varchar2(6);
  v_department_name varchar2(100);
begin
  select first_name into v_name from employees where employee_id = 103;
  select department_id into v_department_name from employees where first_name = v_name;
  dbms_output.put_line('Hello '|| v_name || '. Your department id is : '|| v_department_name );
exception
  when no_data_found then
    dbms_output.put_line('There is no employee with the selected id');
  when too_many_rows then
    dbms_output.put_line('There are more than one employees with the name '|| v_name);
```

```

    dbms_output.put_line('Try with a different employee');
when others then
    dbms_output.put_line('An unexpected error happened. Connect with the programmer..');
    dbms_output.put_line(sqlcode || ' ---> ' || sqlerrm);
end;
----- Inner block exception example-----
declare
    v_name varchar2(6);
    v_department_name varchar2(100);
begin
    select first_name into v_name from employees where employee_id = 100;
begin
    select department_id into v_department_name from employees where first_name = v_name;
exception
    when too_many_rows then
        v_department_name := 'Error in department_name';
end;
    dbms_output.put_line('Hello ' || v_name || '. Your department id is : ' || v_department_name );
exception
    when no_data_found then
        dbms_output.put_line('There is no employee with the selected id');
    when too_many_rows then
        dbms_output.put_line('There are more than one employees with the name ' || v_name);
        dbms_output.put_line('Try with a different employee');
    when others then
        dbms_output.put_line('An unexpected error happened. Connect with the programmer..');
        dbms_output.put_line(sqlcode || ' ---> ' || sqlerrm);
end;
/
select * from employees where first_name = 'Steven';

```

-----Handling non predefine Exeption-----

```

begin
    UPDATE employees_copy set email = null where employee_id = 100;
end;

```

-----HANDLING a nonpredefined exception-----

```

declare
    cannot_update_to_null exception;
    pragma exception_init(cannot_update_to_null,-01407);
begin
    UPDATE employees_copy set email = null where employee_id = 100;
exception
    when cannot_update_to_null then
        dbms_output.put_line('You cannot update with a null value!');
end;

```

-----Handling And Raising User Define Exeption-----

----- creating a user defined exception-----

```

declare
    too_high_salary exception;
    v_salary_check pls_integer;
begin

```

```

select salary into v_salary_check from employees where employee_id = 100;
if v_salary_check > 20000 then
    raise too_high_salary;
end if;
--we do our business if the salary is under 2000
dbms_output.put_line('The salary is in an acceptable range');
exception
when too_high_salary then
    dbms_output.put_line('This salary is too high. You need to decrease it.');
```

----- raising a predefined exception-----

```

declare
    too_high_salary exception;
    v_salary_check pls_integer;
begin
    select salary into v_salary_check from employees where employee_id = 100;
    if v_salary_check > 20000 then
        raise invalid_number;
    end if;
    --we do our business if the salary is under 2000
    dbms_output.put_line('The salary is in an acceptable range');
exception
when invalid_number then
    dbms_output.put_line('This salary is too high. You need to decrease it.');
```

----- raising inside of the exception-----

```

declare
    too_high_salary exception;
    v_salary_check pls_integer;
begin
    select salary into v_salary_check from employees where employee_id = 100;
    if v_salary_check > 20000 then
        raise invalid_number;
    end if;
    --we do our business if the salary is under 2000
    dbms_output.put_line('The salary is in an acceptable range');
exception
when invalid_number then
    dbms_output.put_line('This salary is too high. You need to decrease it.');
```

-----Using_Rise_Application_Error()Procedure-----

```

declare
    too_high_salary exception;
    v_salary_check pls_integer;
begin
    select salary into v_salary_check from employees where employee_id = 100;
    if v_salary_check > 20000 then
        --raise too_high_salary;
        raise_application_error(-20243,'The salary of the selected employee is too high!');
```

```

    dbms_output.put_line('The salary is in an acceptable range');
exception
    when too_high_salary then
        dbms_output.put_line('This salary is too high. You need to decrease it.');
```

end;

----- raise inside of the exception section-----

```

declare
    too_high_salary exception;
    v_salary_check pls_integer;
begin
    select salary into v_salary_check from employees where employee_id = 100;
    if v_salary_check > 20000 then
        raise too_high_salary;
    end if;
    --we do our business if the salary is under 2000
    dbms_output.put_line('The salary is in an acceptable range');
```

exception

```

    when too_high_salary then
        dbms_output.put_line('This salary is too high. You need to decrease it.');
```

raise_application_error(-01403,'The salary of the selected employee is too high!',true);

end;

```

-----
-----
-----
```