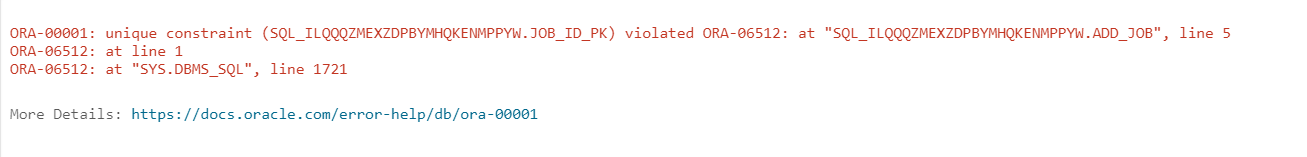
**Zhumagali Kanat IT2-2003**

**Practice 6**

1. Create, compile, and invoke the ADD\_JOB procedure and review the results.

* + - 1. Create a procedure called ADD\_JOB to insert a new job into the JOBS table. Provide the ID and job title using two parameters.
      2. Compile the code, and then invoke the procedure with IT\_DBA as the job ID and Database Administrator as the job title. Query the JOBS table and view the results.
      3. Invoke your procedure again, passing a job ID of ST\_MAN and a job title of Stock Manager. What happens and why?



create procedure add\_job(

j\_id in jobs.job\_id%type,

j\_title in jobs.job\_title%type) is

begin

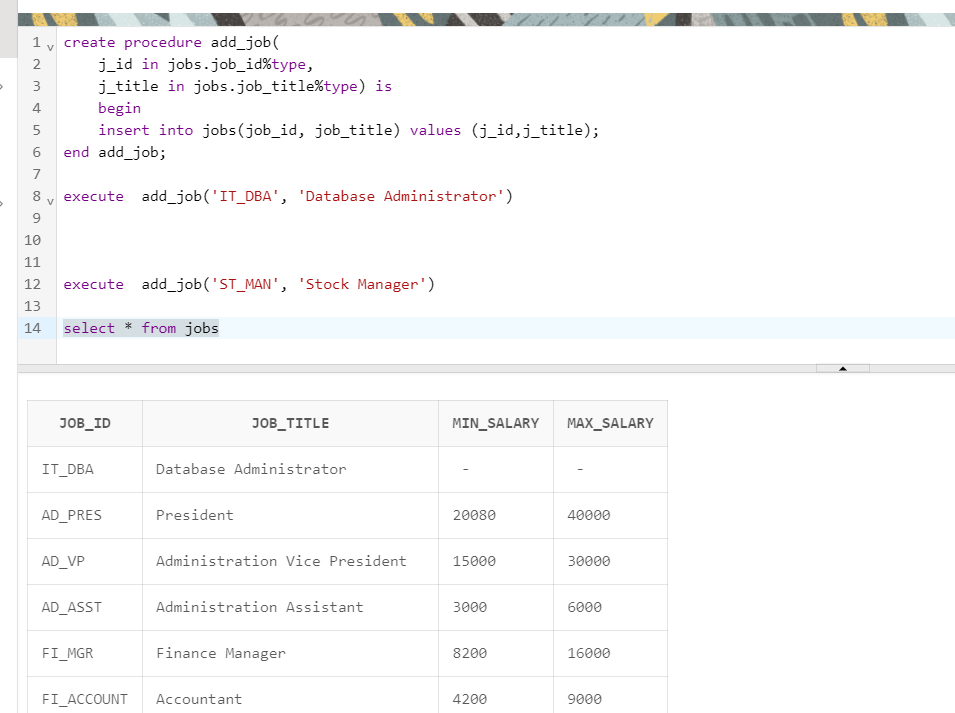
insert into jobs(job\_id, job\_title) values (j\_id,j\_title);

end add\_job;

execute add\_job('IT\_DBA', 'Database Administrator')

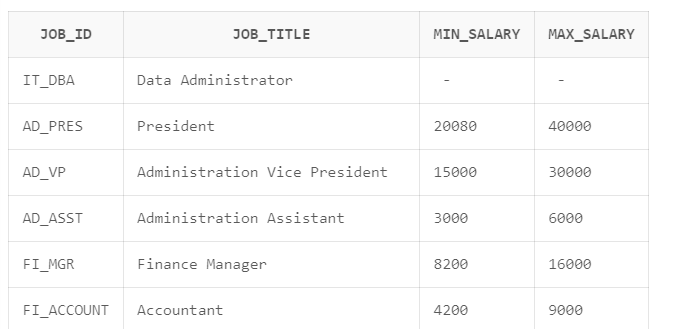
execute add\_job('ST\_MAN', 'Stock Manager')

select \* from jobs



2. Create a procedure called UPD\_JOB to modify a job in the JOBS table.

* + - 1. Create a procedure called UPD\_JOB to update the job title. Provide the job ID and a new title using two parameters. Include the necessary exception handling if no update occurs.
      2. Compile the procedure. Invoke the procedure to change the job title of the job ID IT\_DBA to Data Administrator. Query the JOBS table and view the results.
      3. Test the exception handling section of the procedure by trying to update a job that does not exist. You can use the job ID IT\_WEB and the job title Web Master.



create or replace procedure upd\_job(

j\_id jobs.job\_id%type,

j\_ntitle jobs.job\_title%type) is

begin

update jobs set job\_title = j\_ntitle where job\_id = j\_id;

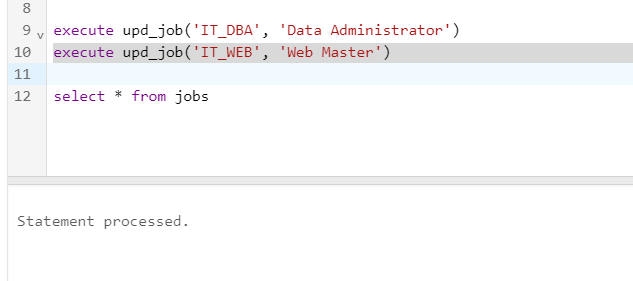
if(sql%rowcount = 0) then dbms\_output.put\_line ('No data found for updating...');

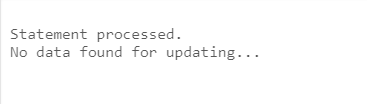
else dbms\_output.put\_line('Nice! ');

end if;

exception when others then dbms\_output.put\_line('error');

end upd\_job;





3. Create a procedure called DEL\_JOB to delete a job from the JOBS table.

* + - 1. Create a procedure called DEL\_JOB to delete a job. Include the necessary exception handling code if no job is deleted.
      2. Compile the code; invoke the procedure using the job ID IT\_DBA. Query the JOBS table and view the results.
      3. Test the exception handling section of the procedure by trying to delete a job that does not exist. Use the IT\_WEB as the job ID. You should get the message that you included in the exception handling section of the procedure as the output.

create or replace procedure del\_job(

j\_id jobs.job\_id%type

) is

begin

delete from jobs where job\_id = j\_id;

if(sql%rowcount = 0) then dbms\_output.put\_line ('No data found for deleting...');

else dbms\_output.put\_line('Nice! ');

end if;

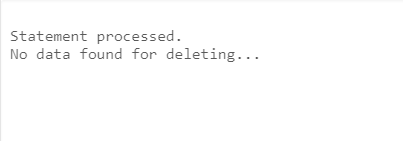
exception

when others then dbms\_output.put\_line('error');

end del\_job;

execute del\_job('IT\_DBA')

execute del\_job('IT\_WEB')

4. Create a procedure called GET\_EMPLOYEE to query the EMPLOYEES table, retrieving the salary and job ID for an employee when provided with the employee ID.

a. Create a procedure that returns a value from the SALARY and JOB\_ID columns for a specified employee ID. Compile the code and remove syntax errors, if any.

b. Create an anonymous block and call the procedure using variables for the two OUT parameters – one for the salary and the other for the job ID. Display the salary and job ID for employee ID 120.

c. Invoke the procedure again, passing an EMPLOYEE\_ID of 300. What happens and why?

create or replace procedure get\_employee (

e\_id in employees.employee\_id%type,

j\_id out employees.job\_id%type,

salary out employees.salary%type

) is

begin

select salary, job\_id into salary, j\_id from employees where employee\_id = e\_id;

exception

when no\_data\_found then dbms\_output.put\_line('No data found for this id');

when others then dbms\_output.put\_line('Error');

end get\_employee;

declare

emp\_job employees.job\_id%type;

emp\_sal employees.salary%type;

begin

get\_employee(120, emp\_job, emp\_sal);

dbms\_output.put\_line('Job ID: ' || emp\_job || ' Salary: ' || emp\_sal);

end;

