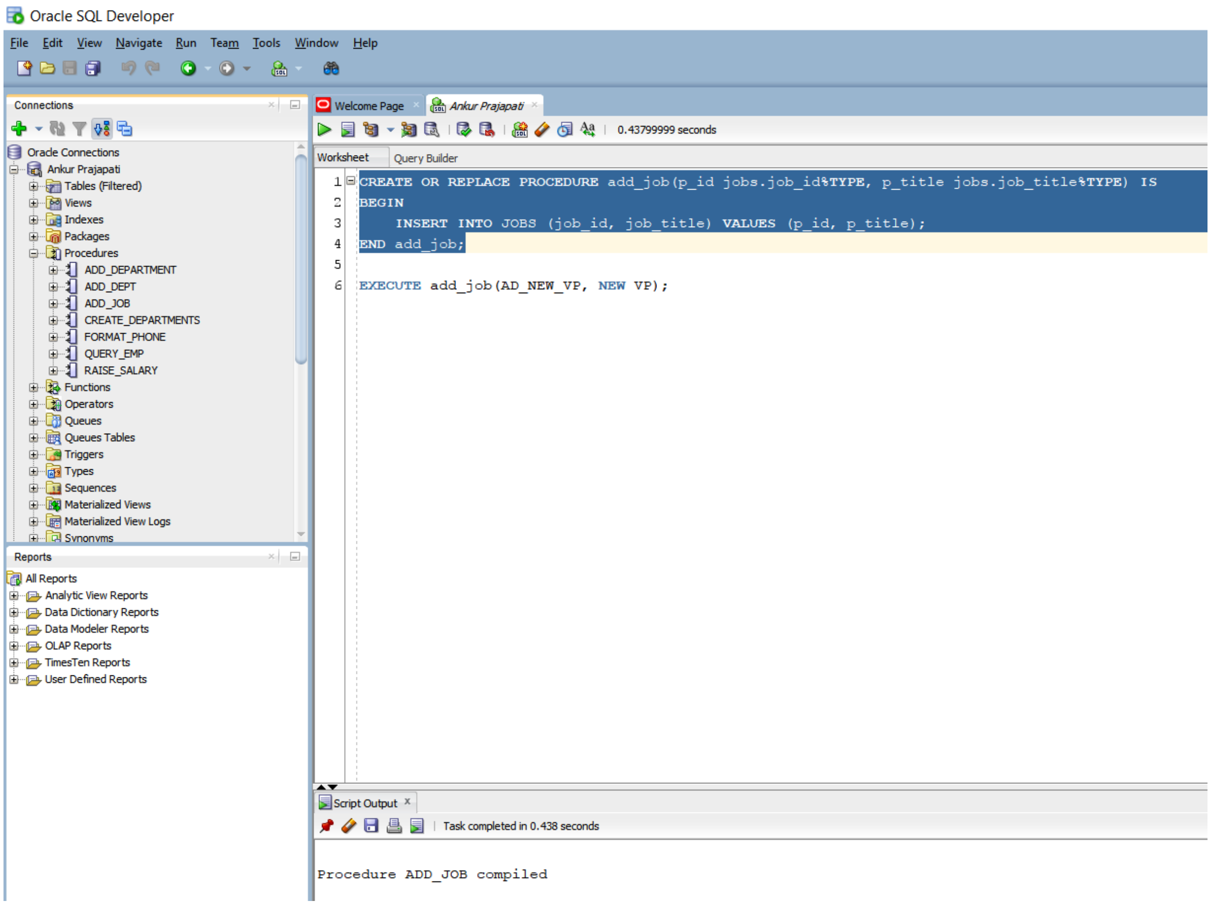
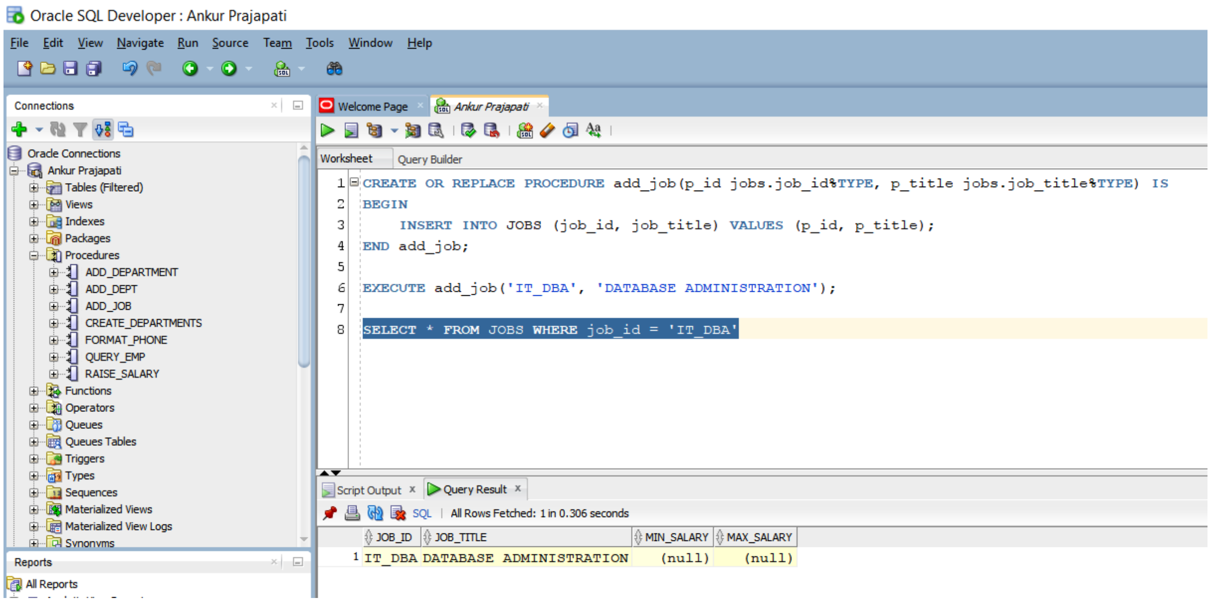
1. ADD_JOB Procedure:



```
CREATE OR REPLACE PROCEDURE add_job(p_id jobs.job_id%TYPE, p_title jobs.job_title%TYPE) IS
BEGIN
    INSERT INTO JOBS (job_id, job_title) VALUES (p_id, p_title);
END add_job;

EXECUTE add_job(AD_NEW_VP, NEW VP);
```

```
Procedure ADD_JOB compiled
```

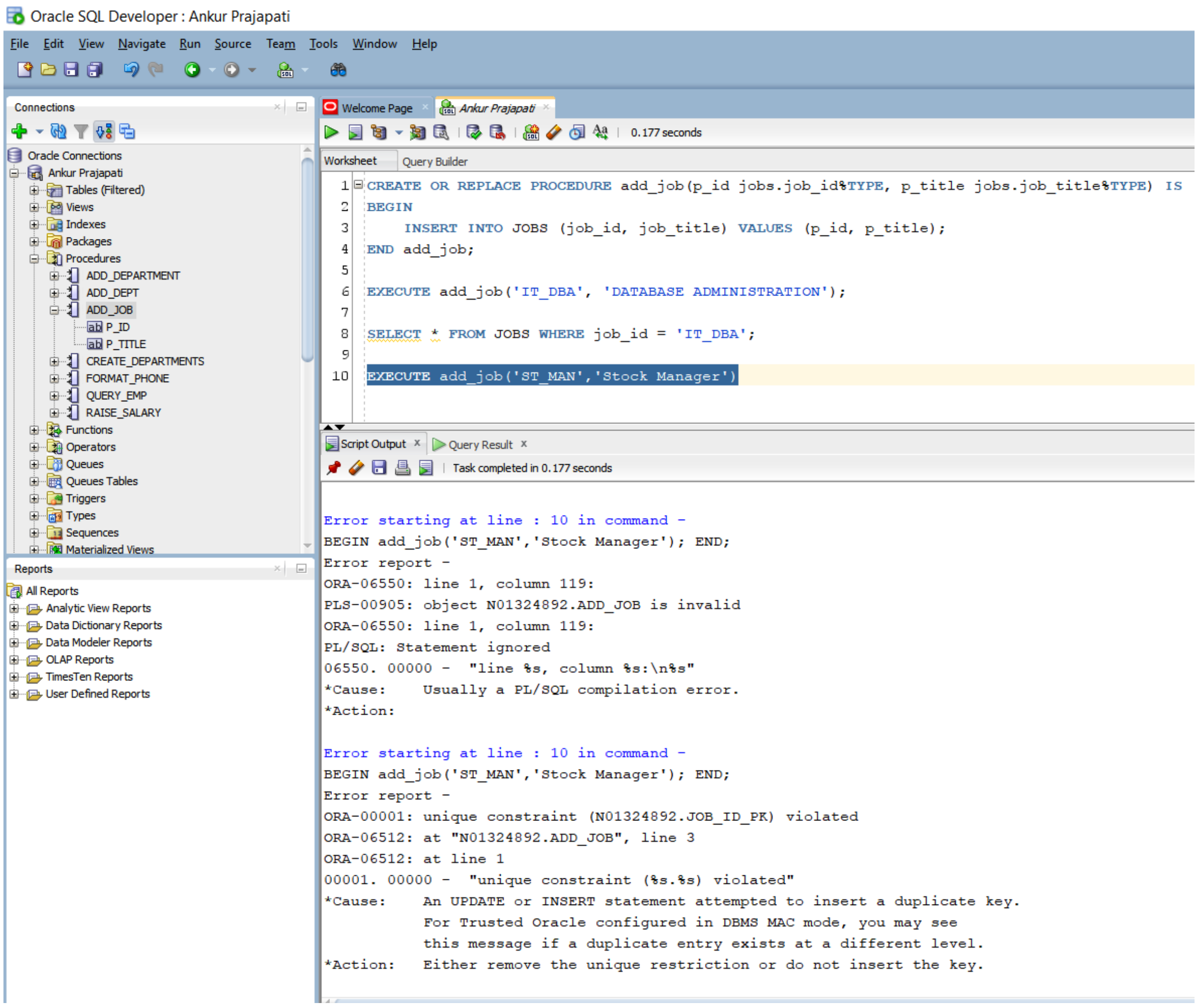
Here we are creating add_job procedure. As per the statement it will create new add_job procedure if it doesn’t exist otherwise it will drop it and again it will create. It will add job_id and job_title with using formal parameters p_id and p_title.



```
CREATE OR REPLACE PROCEDURE add_job(p_id jobs.job_id%TYPE, p_title jobs.job_title%TYPE) IS
BEGIN
    INSERT INTO JOBS (job_id, job_title) VALUES (p_id, p_title);
END add_job;

EXECUTE add_job('IT_DBA', 'DATABASE ADMINISTRATION');

SELECT * FROM JOBS WHERE job_id = 'IT_DBA'
```

| | JOB_ID | JOB_TITLE | MIN_SALARY | MAX_SALARY |
|---|---|---|---|---|
| 1 | IT_DBA | DATABASE ADMINISTRATION | (null) | (null) |

Here it will add new job with job_id having value of ‘IT_DBA’.

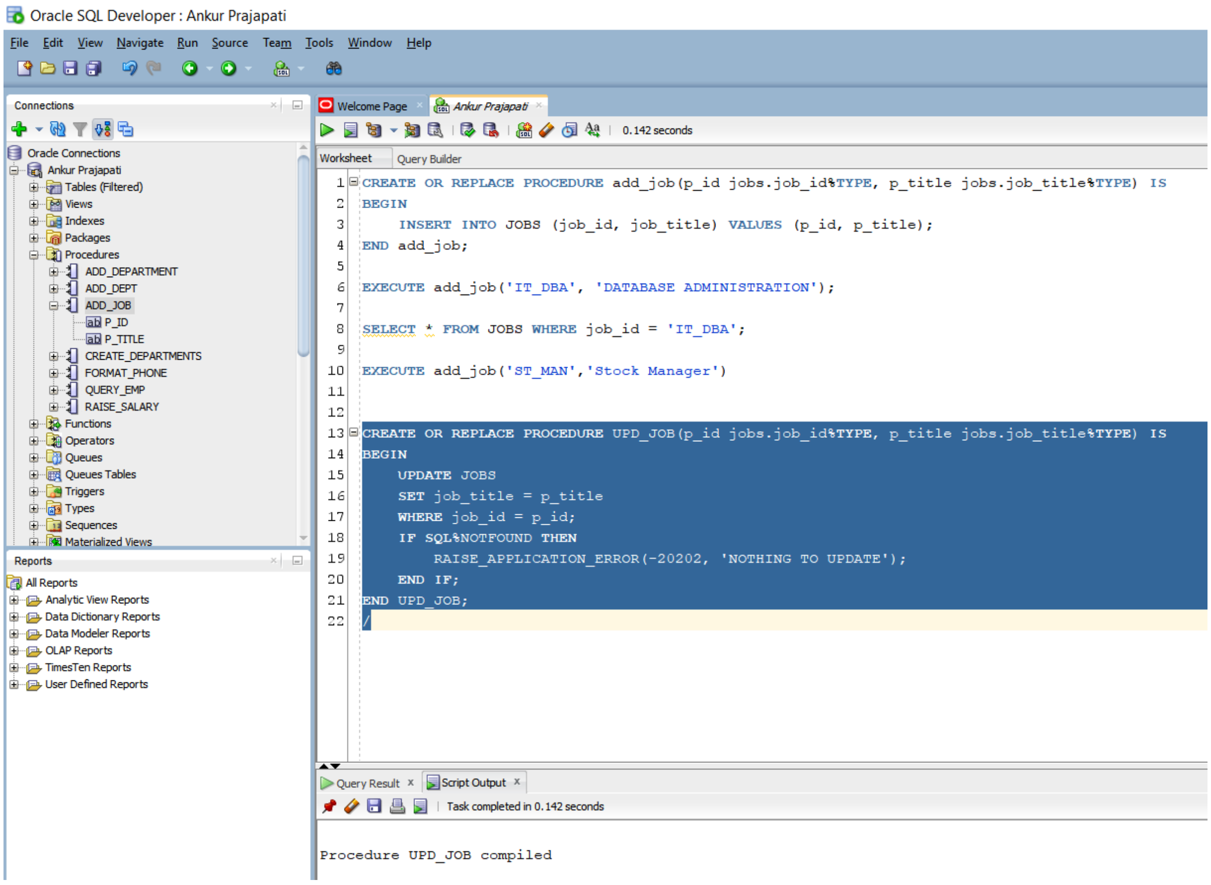


```
CREATE OR REPLACE PROCEDURE add_job(p_id jobs.job_id%TYPE, p_title jobs.job_title%TYPE) IS
BEGIN
    INSERT INTO JOBS (job_id, job_title) VALUES (p_id, p_title);
END add_job;

EXECUTE add_job('IT_DBA', 'DATABASE ADMINISTRATION');

SELECT * FROM JOBS WHERE job_id = 'IT_DBA';

EXECUTE add_job('ST_MAN','Stock Manager')
```

```
Error starting at line : 10 in command -
BEGIN add_job('ST_MAN','Stock Manager'); END;
Error report -
ORA-06550: line 1, column 119:
PLS-00905: object N01324892.ADD_JOB is invalid
ORA-06550: line 1, column 119:
PL/SQL: Statement ignored
06550. 00000 -  "line %s, column %s:\n%s"
*Cause:    Usually a PL/SQL compilation error.
*Action:

Error starting at line : 10 in command -
BEGIN add_job('ST_MAN','Stock Manager'); END;
Error report -
ORA-00001: unique constraint (N01324892.JOB_ID_PK) violated
ORA-06512: at "N01324892.ADD_JOB", line 3
ORA-06512: at line 1
00001. 00000 -  "unique constraint (%s.%s) violated"
*Cause:    An UPDATE or INSERT statement attempted to insert a duplicate key.
           For Trusted Oracle configured in DBMS MAC mode, you may see
           this message if a duplicate entry exists at a different level.
*Action:   Either remove the unique restriction or do not insert the key.
```

Here I am trying to add new job with job_id = ‘ST_MAN’ and job_title = ‘Stock Manager’ but it gives error of unique key constraint violation. It means there is same data with that job_id is available. It gives solution that either remove the unique restriction or do not insert the key.

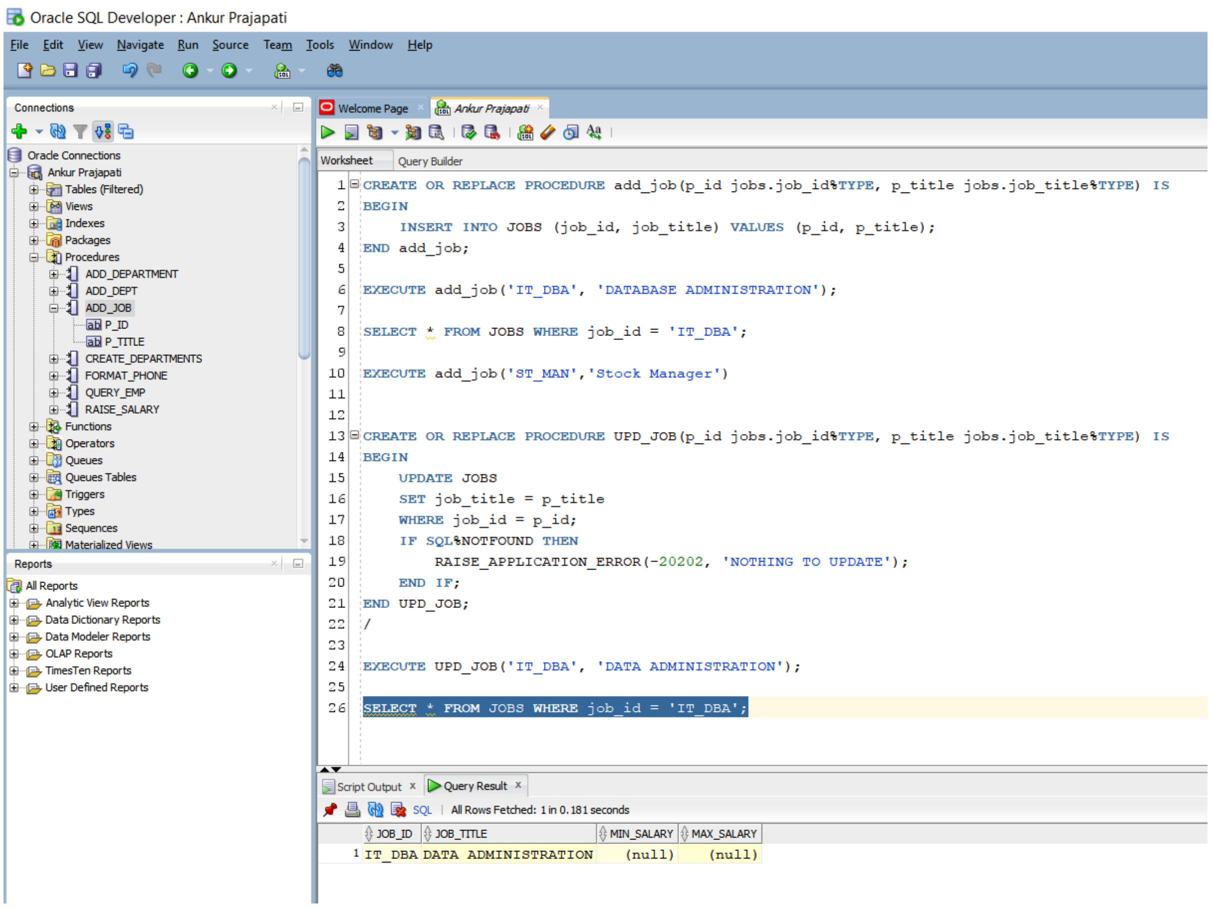
2. UPD_JOB Procedure:



```sql
CREATE OR REPLACE PROCEDURE add_job(p_id jobs.job_id%TYPE, p_title jobs.job_title%TYPE) IS
BEGIN
    INSERT INTO JOBS (job_id, job_title) VALUES (p_id, p_title);
END add_job;

EXECUTE add_job('IT_DBA', 'DATABASE ADMINISTRATION');

SELECT * FROM JOBS WHERE job_id = 'IT_DBA';

EXECUTE add_job('ST_MAN','Stock Manager')


CREATE OR REPLACE PROCEDURE UPD_JOB(p_id jobs.job_id%TYPE, p_title jobs.job_title%TYPE) IS
BEGIN
    UPDATE JOBS
    SET job_title = p_title
    WHERE job_id = p_id;
    IF SQL%NOTFOUND THEN
        RAISE_APPLICATION_ERROR(-20202, 'NOTHING TO UPDATE');
    END IF;
END UPD_JOB;
/
```

```
Procedure UPD_JOB compiled
```

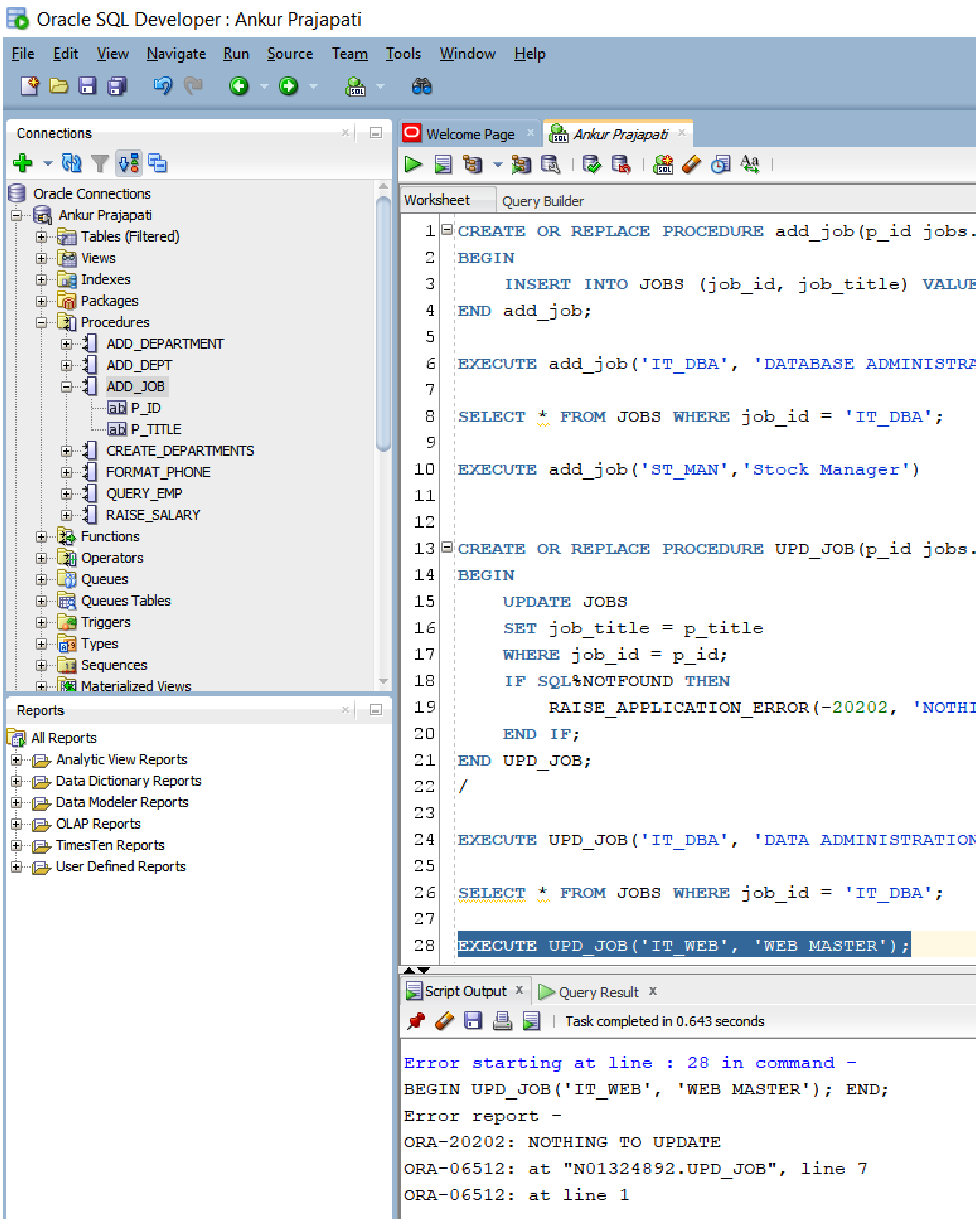
Here I am creating a procedure named UPD_JOB with formal parameters p_id with type of job_id, p_title with type of job_title. In this it sets job_title with p_title. If SQL%NOTFOUND then it will return message showing ‘Nothing To UPDATE’.
Basically, it job_id is not there then it won’t update and will return error.



```
CREATE OR REPLACE PROCEDURE add_job(p_id jobs.job_id%TYPE, p_title jobs.job_title%TYPE) IS
BEGIN
    INSERT INTO JOBS (job_id, job_title) VALUES (p_id, p_title);
END add_job;

EXECUTE add_job('IT_DBA', 'DATABASE ADMINISTRATION');

SELECT * FROM JOBS WHERE job_id = 'IT_DBA';

EXECUTE add_job('ST_MAN','Stock Manager')


CREATE OR REPLACE PROCEDURE UPD_JOB(p_id jobs.job_id%TYPE, p_title jobs.job_title%TYPE) IS
BEGIN
    UPDATE JOBS
    SET job_title = p_title
    WHERE job_id = p_id;
    IF SQL%NOTFOUND THEN
        RAISE_APPLICATION_ERROR(-20202, 'NOTHING TO UPDATE');
    END IF;
END UPD_JOB;
/

EXECUTE UPD_JOB('IT_DBA', 'DATA ADMINISTRATION');

SELECT * FROM JOBS WHERE job_id = 'IT_DBA';
```

|  | JOB_ID | JOB_TITLE | MIN_SALARY | MAX_SALARY |
|---|---|---|---|---|
| 1 | IT_DBA | DATA ADMINISTRATION | (null) | (null) |

Here I am trying to update job with job_id = ‘IT_DBA’. First it was named as ‘DATABASE ADMINISTRATION’ to ‘DATA ADMINISTRATION’.

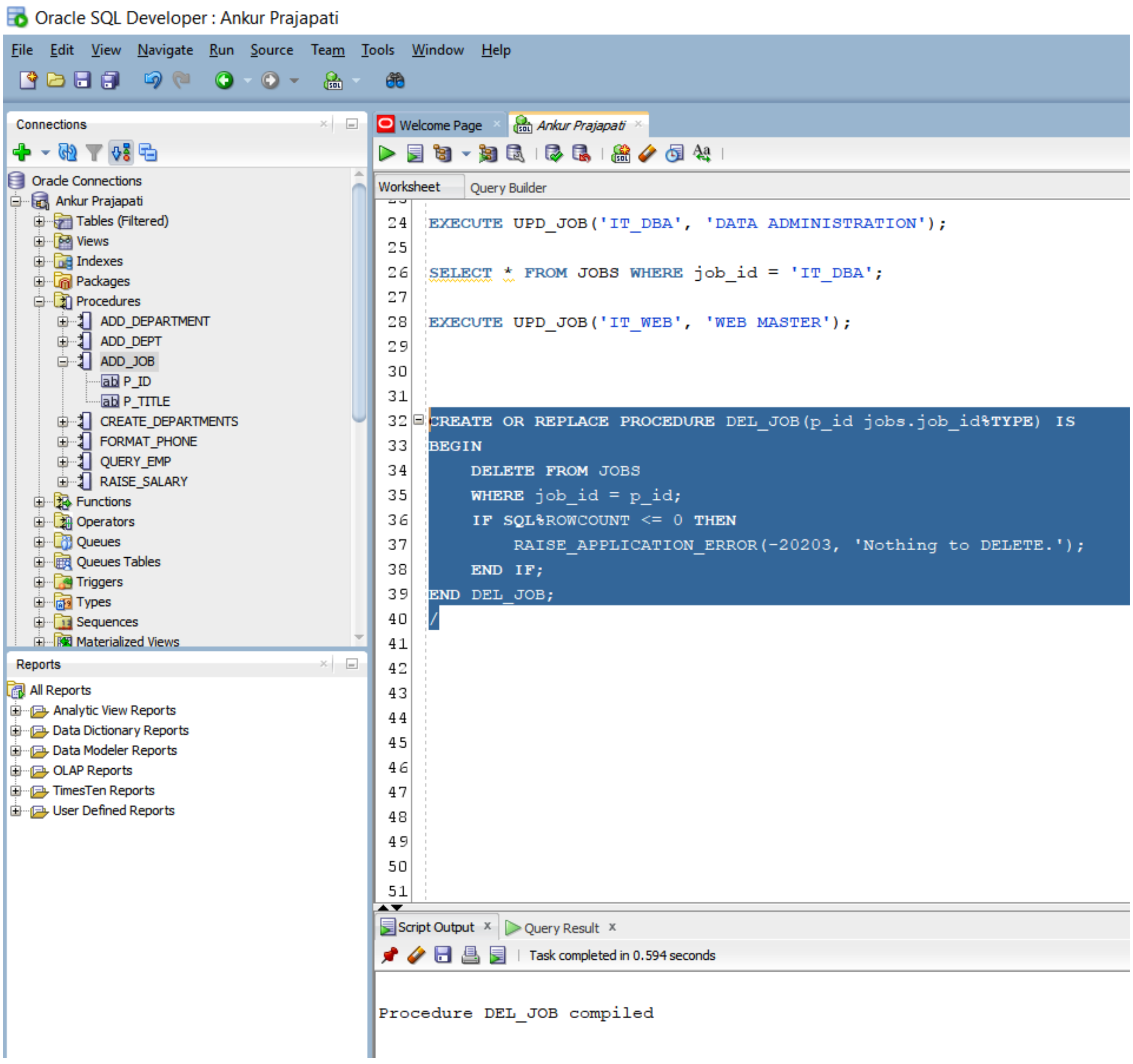


```
CREATE OR REPLACE PROCEDURE add_job(p_id jobs.
BEGIN
    INSERT INTO JOBS (job_id, job_title) VALUE
END add_job;

EXECUTE add_job('IT_DBA', 'DATABASE ADMINISTRA

SELECT * FROM JOBS WHERE job_id = 'IT_DBA';

EXECUTE add_job('ST_MAN','Stock Manager')


CREATE OR REPLACE PROCEDURE UPD_JOB(p_id jobs.
BEGIN
    UPDATE JOBS
    SET job_title = p_title
    WHERE job_id = p_id;
    IF SQL%NOTFOUND THEN
        RAISE_APPLICATION_ERROR(-20202, 'NOTHI
    END IF;
END UPD_JOB;
/

EXECUTE UPD_JOB('IT_DBA', 'DATA ADMINISTRATION

SELECT * FROM JOBS WHERE job_id = 'IT_DBA';

EXECUTE UPD_JOB('IT_WEB', 'WEB MASTER');
```

```
Error starting at line : 28 in command -
BEGIN UPD_JOB('IT_WEB', 'WEB MASTER'); END;
Error report -
ORA-20202: NOTHING TO UPDATE
ORA-06512: at "N01324892.UPD_JOB", line 7
ORA-06512: at line 1
```

Here I am trying to update job having job_id = 'IT_WEB' but it's not stored in database so it gives an error message notifying that 'Nothing to Update' associated to SQLERRORCode.

3. DEL_JOB Procedure:

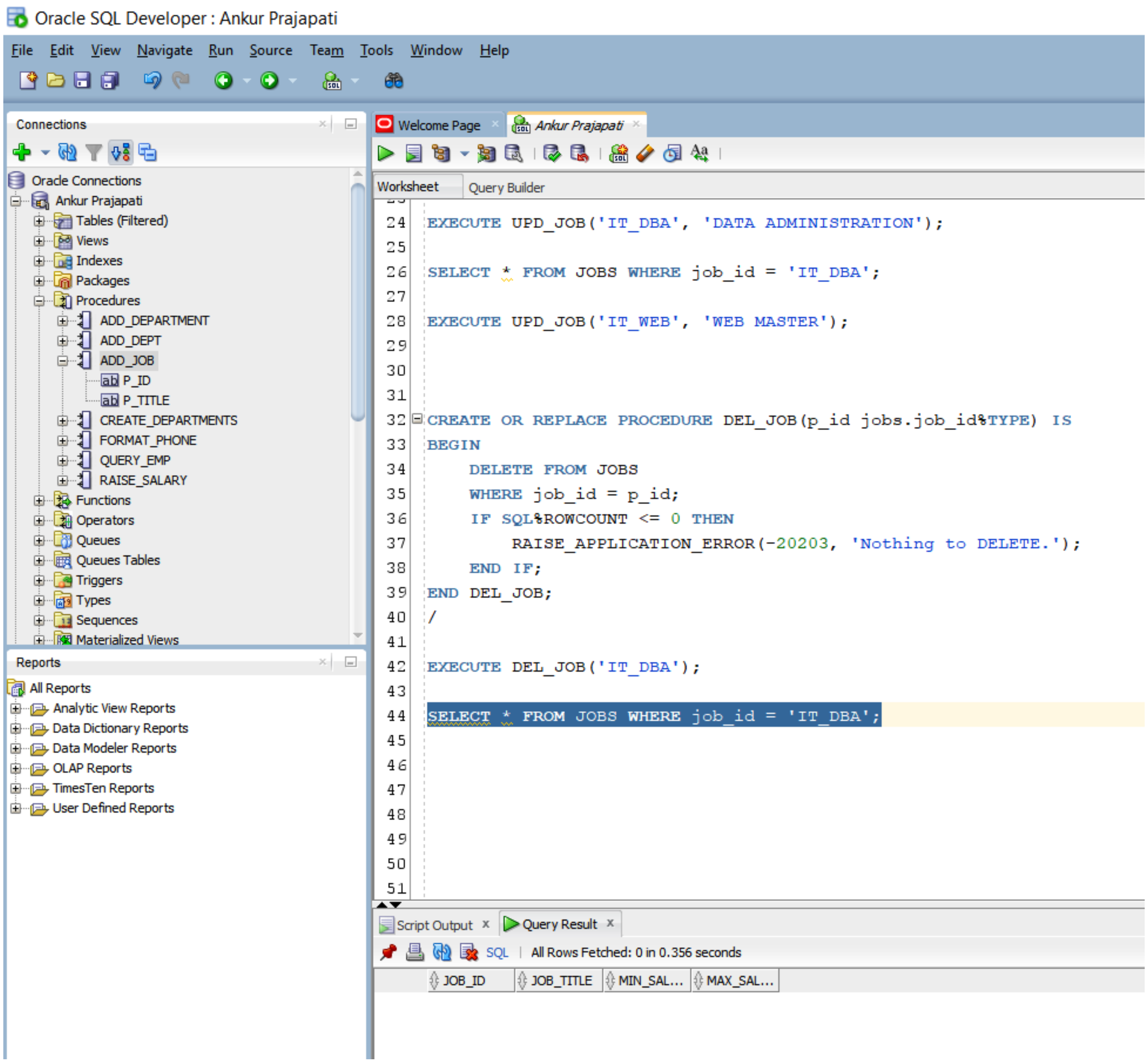


```
EXECUTE UPD_JOB('IT_DBA', 'DATA ADMINISTRATION');

SELECT * FROM JOBS WHERE job_id = 'IT_DBA';

EXECUTE UPD_JOB('IT_WEB', 'WEB MASTER');


CREATE OR REPLACE PROCEDURE DEL_JOB(p_id jobs.job_id%TYPE) IS
BEGIN
    DELETE FROM JOBS
    WHERE job_id = p_id;
    IF SQL%ROWCOUNT <= 0 THEN
        RAISE_APPLICATION_ERROR(-20203, 'Nothing to DELETE.');
    END IF;
END DEL_JOB;
/
```

Task completed in 0.594 seconds

```
Procedure DEL_JOB compiled
```

Here I am creating procedure named DEL_JOB with formal parameter p_id having type of job_id. It deletes job having same id in database.
If there is no row available to delete (SQL%ROWCOUNT is less than 0) then it will give error message that ‘Nothing To DELETE’ associated with SQLERRORCODE -20203.

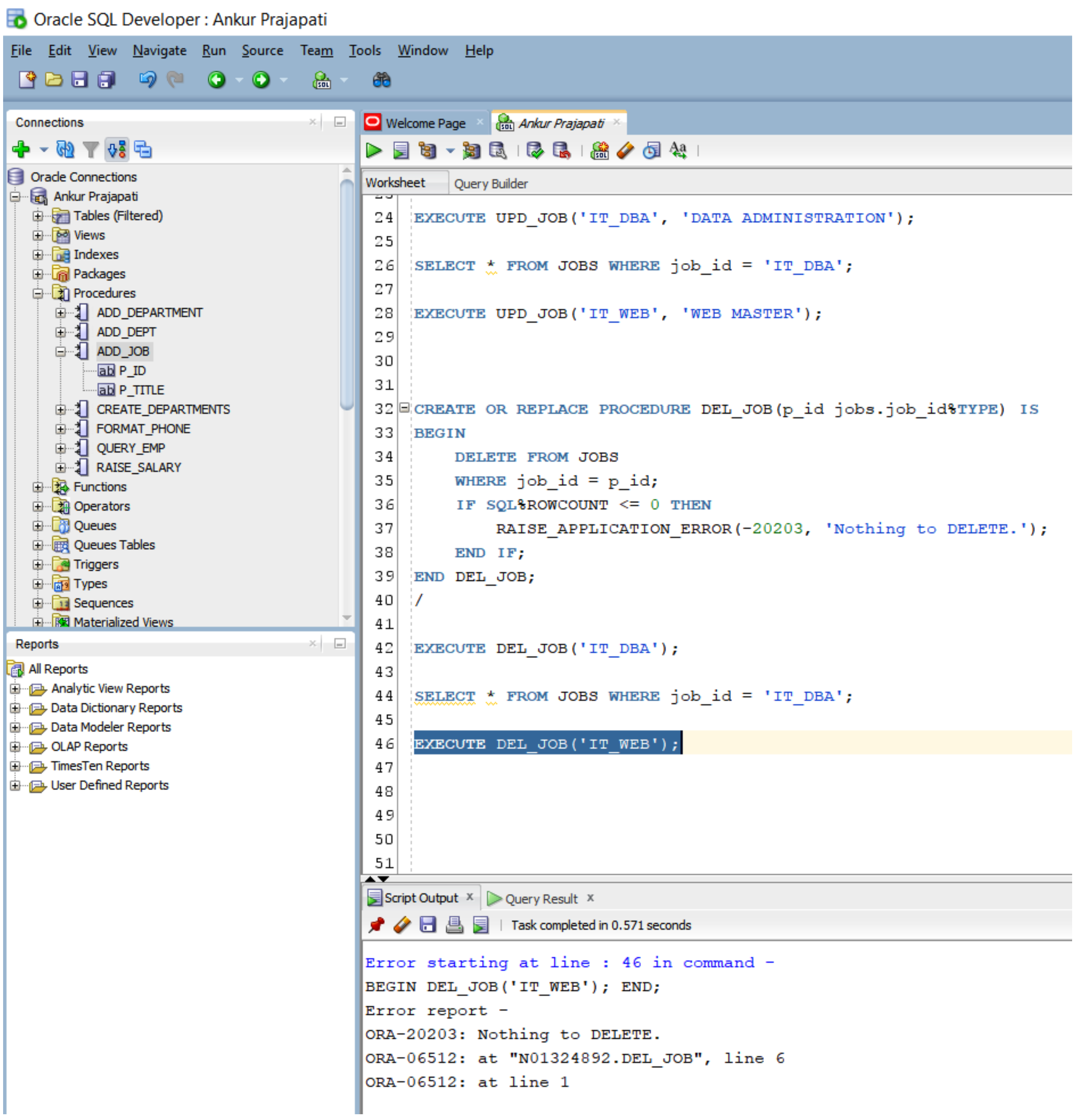


```
EXECUTE UPD_JOB('IT_DBA', 'DATA ADMINISTRATION');

SELECT * FROM JOBS WHERE job_id = 'IT_DBA';

EXECUTE UPD_JOB('IT_WEB', 'WEB MASTER');


CREATE OR REPLACE PROCEDURE DEL_JOB(p_id jobs.job_id%TYPE) IS
BEGIN
    DELETE FROM JOBS
    WHERE job_id = p_id;
    IF SQL%ROWCOUNT <= 0 THEN
        RAISE_APPLICATION_ERROR(-20203, 'Nothing to DELETE.');
    END IF;
END DEL_JOB;
/

EXECUTE DEL_JOB('IT_DBA');

SELECT * FROM JOBS WHERE job_id = 'IT_DBA';
```

All Rows Fetched: 0 in 0.356 seconds

| JOB_ID | JOB_TITLE | MIN_SAL... | MAX_SAL... |
|---|---|---|---|

There is data stored in database with job_id = ‘IT_DBA’ and I have deleted it from here. That’s why it is showing us the blank record.

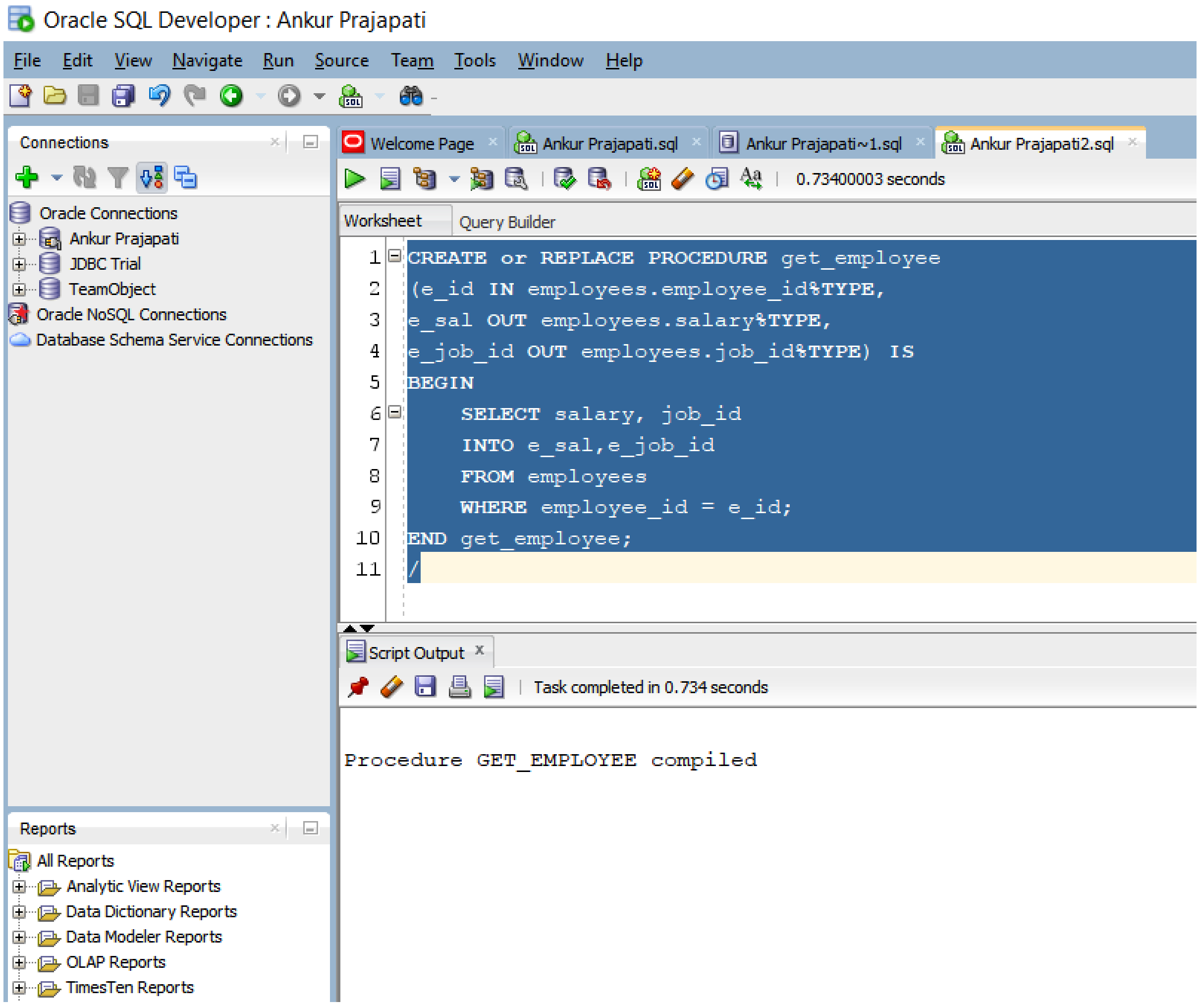


```
EXECUTE UPD_JOB('IT_DBA', 'DATA ADMINISTRATION');

SELECT * FROM JOBS WHERE job_id = 'IT_DBA';

EXECUTE UPD_JOB('IT_WEB', 'WEB MASTER');


CREATE OR REPLACE PROCEDURE DEL_JOB(p_id jobs.job_id%TYPE) IS
BEGIN
    DELETE FROM JOBS
    WHERE job_id = p_id;
    IF SQL%ROWCOUNT <= 0 THEN
        RAISE_APPLICATION_ERROR(-20203, 'Nothing to DELETE.');
    END IF;
END DEL_JOB;
/

EXECUTE DEL_JOB('IT_DBA');

SELECT * FROM JOBS WHERE job_id = 'IT_DBA';

EXECUTE DEL_JOB('IT_WEB');
```

```
Error starting at line : 46 in command -
BEGIN DEL_JOB('IT_WEB'); END;
Error report -
ORA-20203: Nothing to DELETE.
ORA-06512: at "N01324892.DEL_JOB", line 6
ORA-06512: at line 1
```

Here it will give error that there is nothing to delete because there is not record of job_id = 'IT_WEB'.

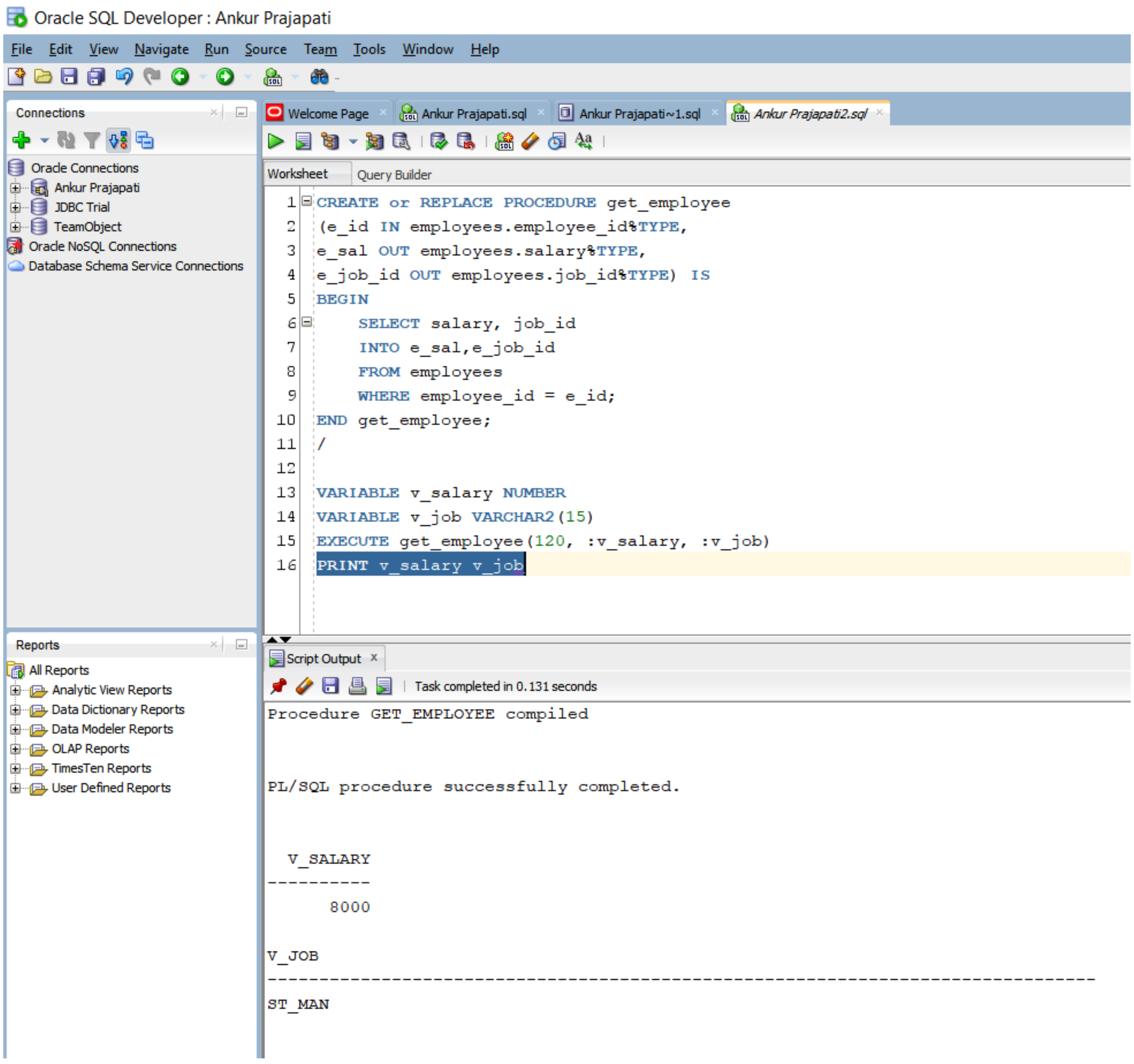
4. GET_EMPLOYEE Procedure:



```
CREATE or REPLACE PROCEDURE get_employee
(e_id IN employees.employee_id%TYPE,
e_sal OUT employees.salary%TYPE,
e_job_id OUT employees.job_id%TYPE) IS
BEGIN
    SELECT salary, job_id
    INTO e_sal,e_job_id
    FROM employees
    WHERE employee_id = e_id;
END get_employee;
/
```

```
Procedure GET_EMPLOYEE compiled
```

Here I am creating get_employee procedure with formal parameters e_Id with type of employees.employee_id, e_sal with type of employees.salary and e_job_id with type of employees.job_id where employee_id(actual parameter) is same as e_id(formal parameters).

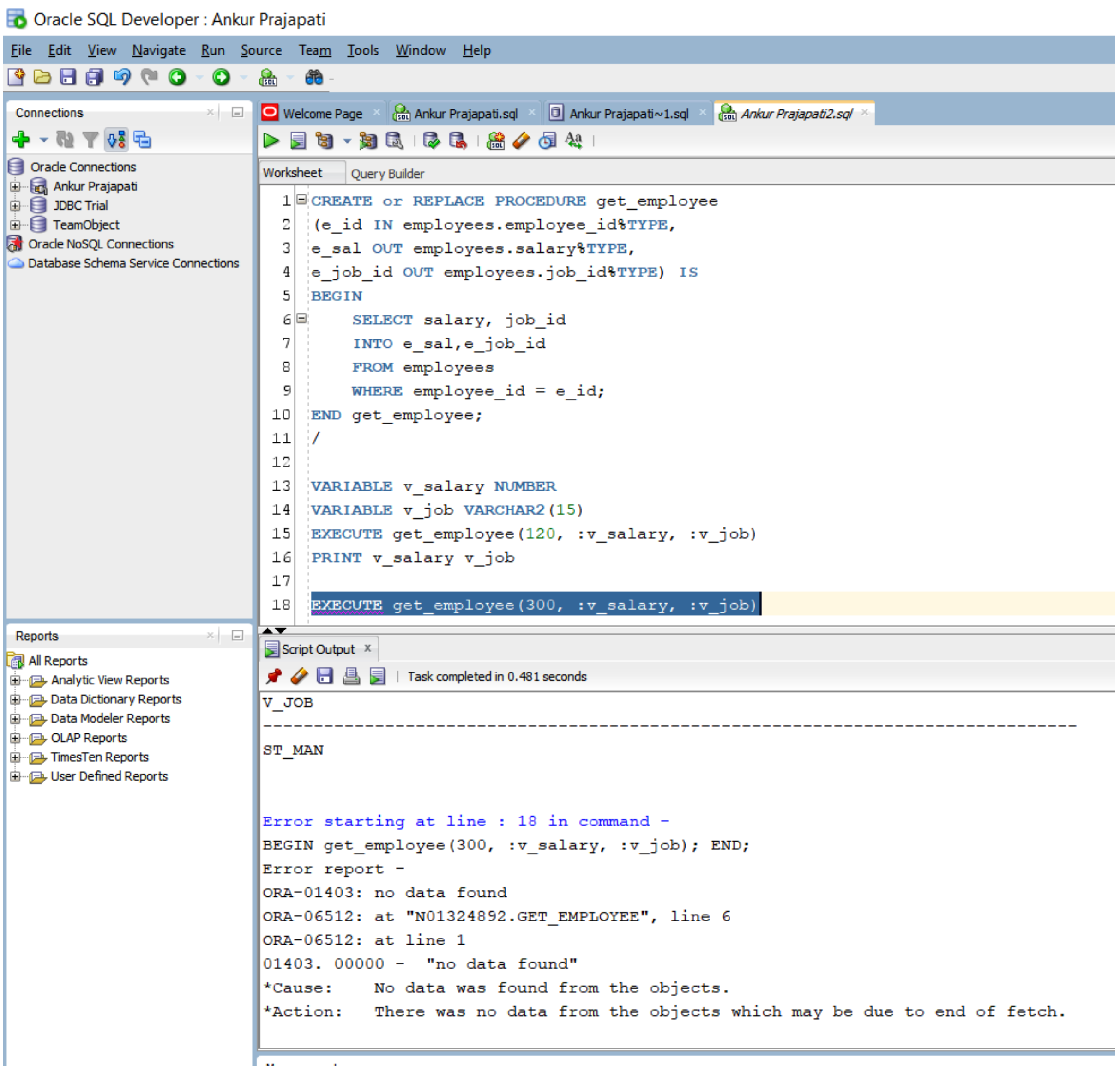


```
CREATE or REPLACE PROCEDURE get_employee
(e_id IN employees.employee_id%TYPE,
e_sal OUT employees.salary%TYPE,
e_job_id OUT employees.job_id%TYPE) IS
BEGIN
    SELECT salary, job_id
    INTO e_sal,e_job_id
    FROM employees
    WHERE employee_id = e_id;
END get_employee;
/

VARIABLE v_salary NUMBER
VARIABLE v_job VARCHAR2(15)
EXECUTE get_employee(120, :v_salary, :v_job)
PRINT v_salary v_job
```

```
Procedure GET_EMPLOYEE compiled


PL/SQL procedure successfully completed.


  V_SALARY
----------
      8000

V_JOB
--------------------------------------------------------------------------------
ST_MAN
```

Here v_salary of NUMBER type and v_job of VARCHAR2 type is declared.

Here employee_id is 120 and based on that salary and job_id associated with it will be printed using bind variable. Note that here e_id is of TYPE IN which means we have to give it, where v_salary and v_job is of TYPE OUT that means it will give data to us.



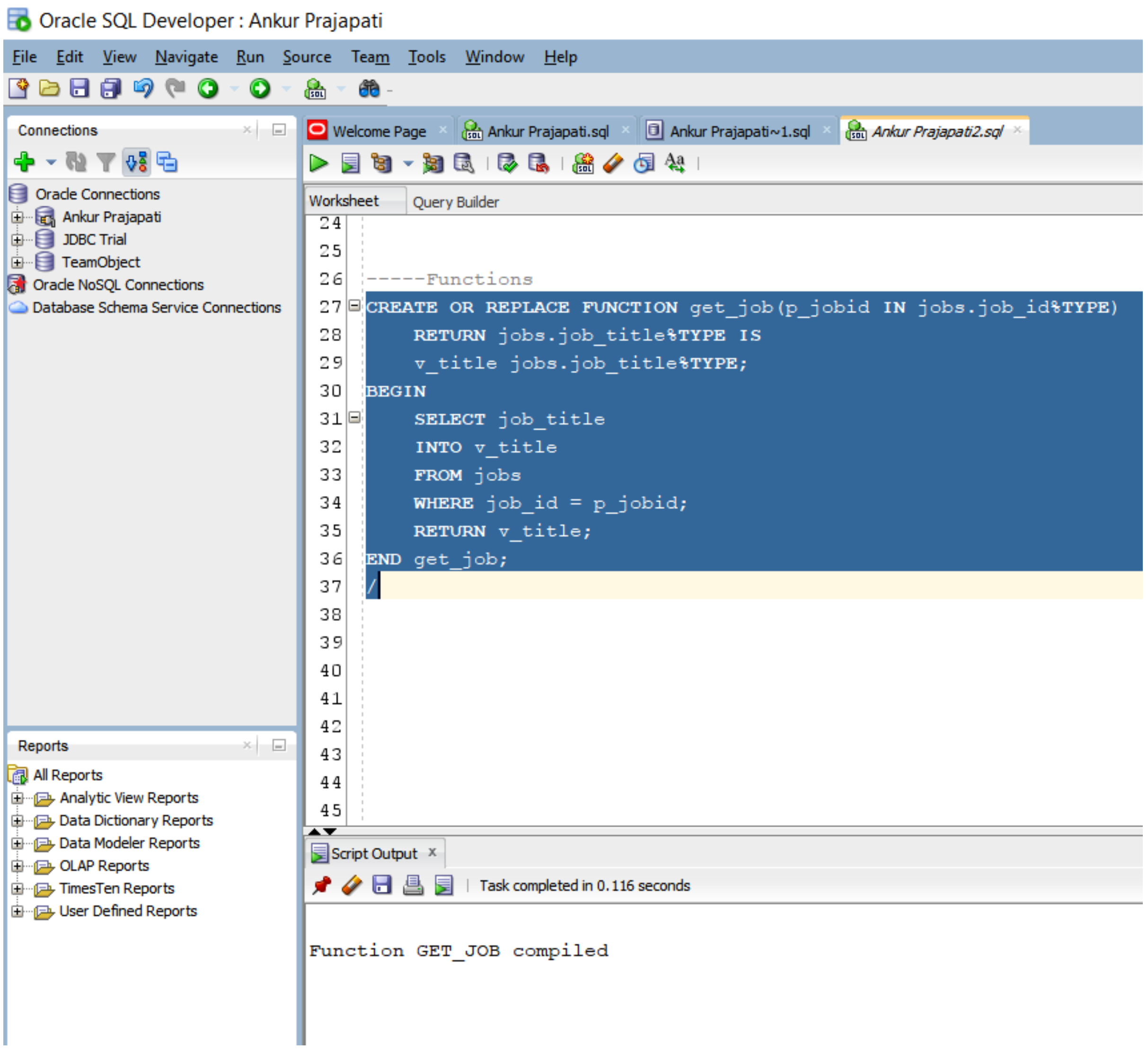
```
CREATE or REPLACE PROCEDURE get_employee
(e_id IN employees.employee_id%TYPE,
e_sal OUT employees.salary%TYPE,
e_job_id OUT employees.job_id%TYPE) IS
BEGIN
    SELECT salary, job_id
    INTO e_sal,e_job_id
    FROM employees
    WHERE employee_id = e_id;
END get_employee;
/

VARIABLE v_salary NUMBER
VARIABLE v_job VARCHAR2(15)
EXECUTE get_employee(120, :v_salary, :v_job)
PRINT v_salary v_job

EXECUTE get_employee(300, :v_salary, :v_job)
```

```
V_JOB
--------------------------------------------------------------------------------
ST_MAN


Error starting at line : 18 in command -
BEGIN get_employee(300, :v_salary, :v_job); END;
Error report -
ORA-01403: no data found
ORA-06512: at "N01324892.GET_EMPLOYEE", line 6
ORA-06512: at line 1
01403. 00000 -  "no data found"
*Cause:    No data was found from the objects.
*Action:   There was no data from the objects which may be due to end of fetch.
```

Here we tried to fetch data from employees table having employee id 300. But as you can see, we have no data for this. So that it gives an error no data found.

Practice 3: Creating Functions:

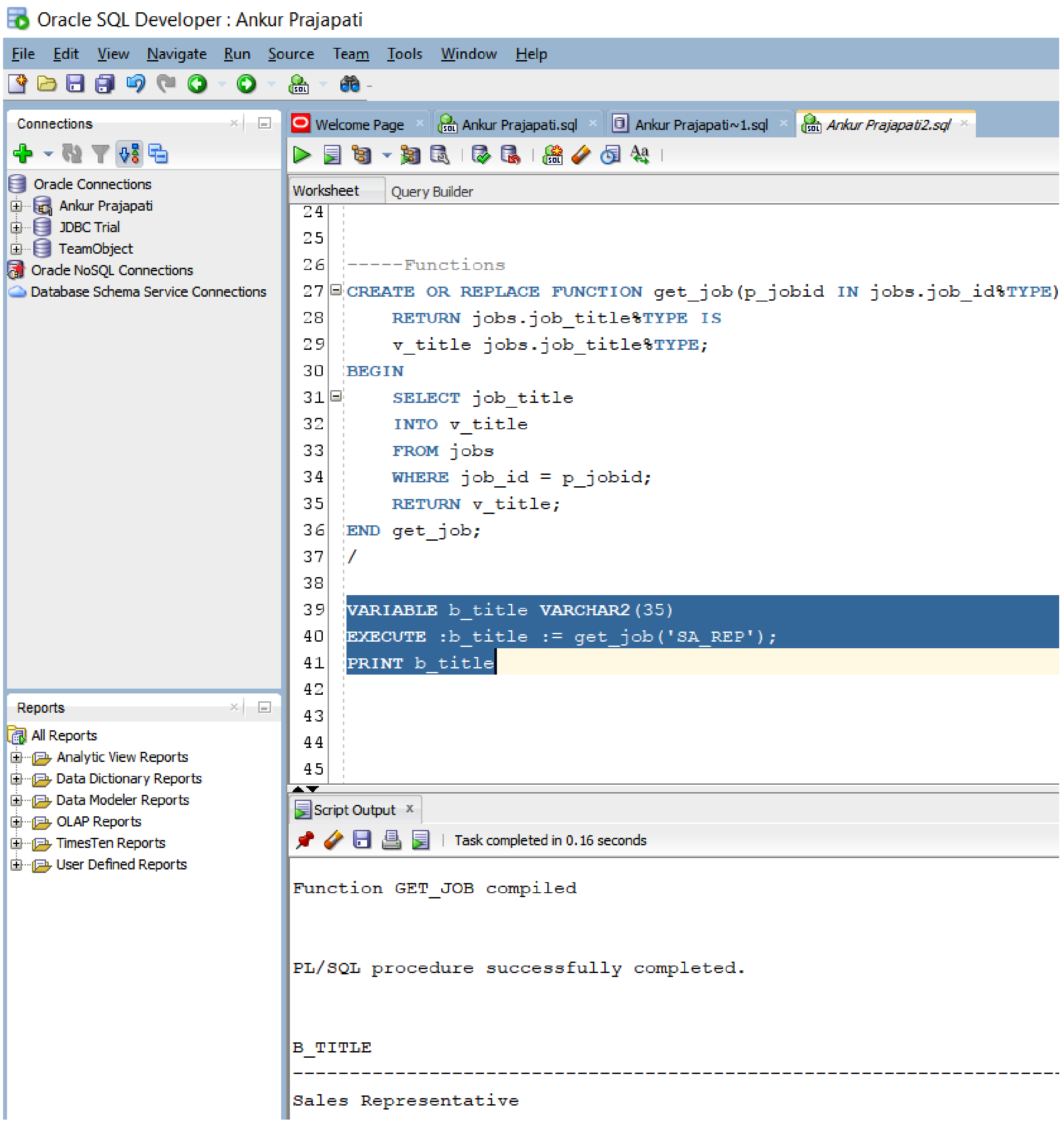
1. GET_JOB Function:



```
-----Functions
CREATE OR REPLACE FUNCTION get_job(p_jobid IN jobs.job_id%TYPE)
    RETURN jobs.job_title%TYPE IS
    v_title jobs.job_title%TYPE;
BEGIN
    SELECT job_title
    INTO v_title
    FROM jobs
    WHERE job_id = p_jobid;
    RETURN v_title;
END get_job;
/
```

```
Function GET_JOB compiled
```

GET_JOB is compiled

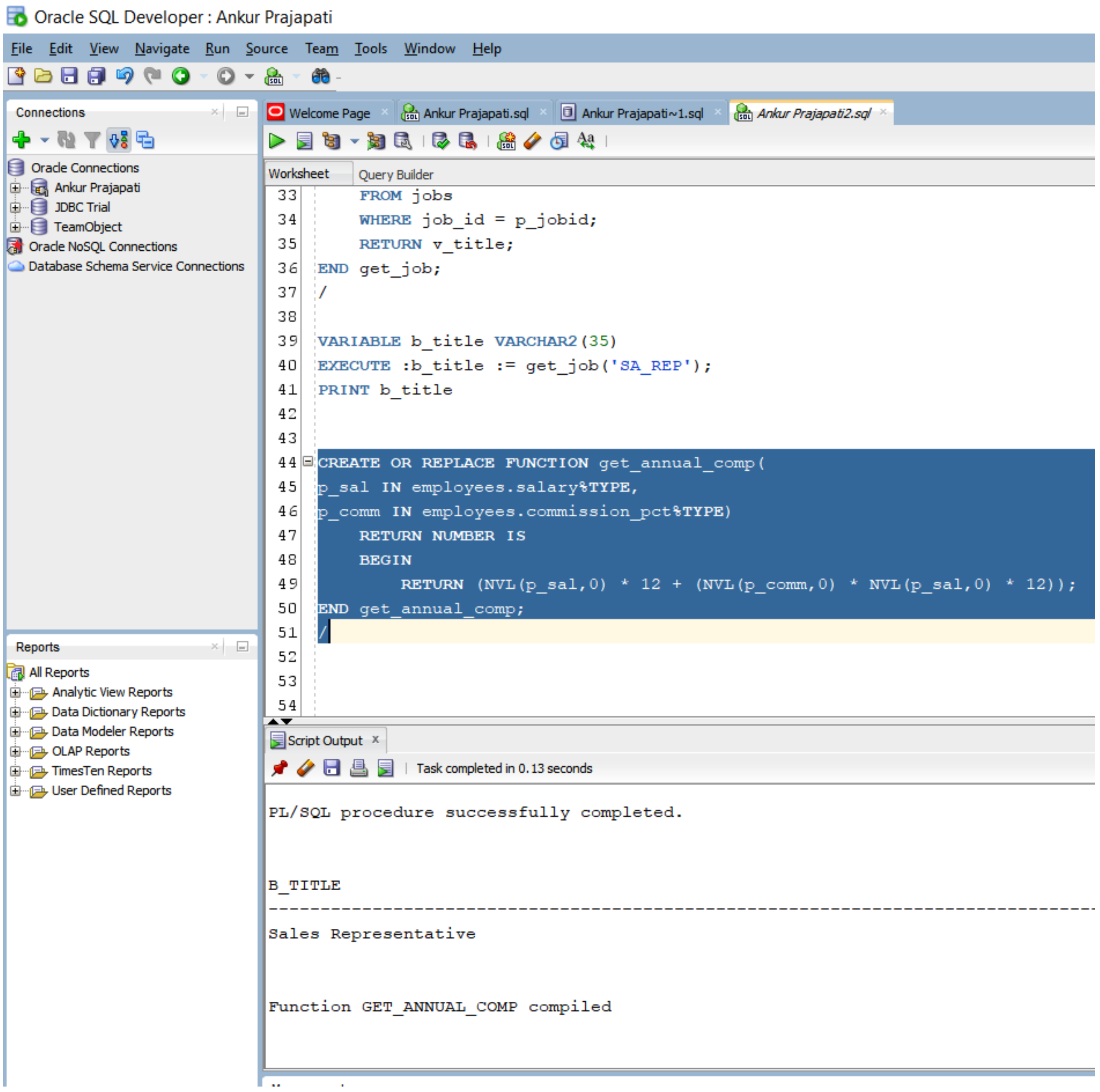


```
-----Functions
CREATE OR REPLACE FUNCTION get_job(p_jobid IN jobs.job_id%TYPE)
    RETURN jobs.job_title%TYPE IS
    v_title jobs.job_title%TYPE;
BEGIN
    SELECT job_title
    INTO v_title
    FROM jobs
    WHERE job_id = p_jobid;
    RETURN v_title;
END get_job;
/

VARIABLE b_title VARCHAR2(35)
EXECUTE :b_title := get_job('SA_REP');
PRINT b_title
```

```
Function GET_JOB compiled


PL/SQL procedure successfully completed.


B_TITLE
--------------------------------------------------------------------
Sales Representative
```

Here it gives job title from given job_id.
For example, SA_REP to SALES REPRESENTATIVE.

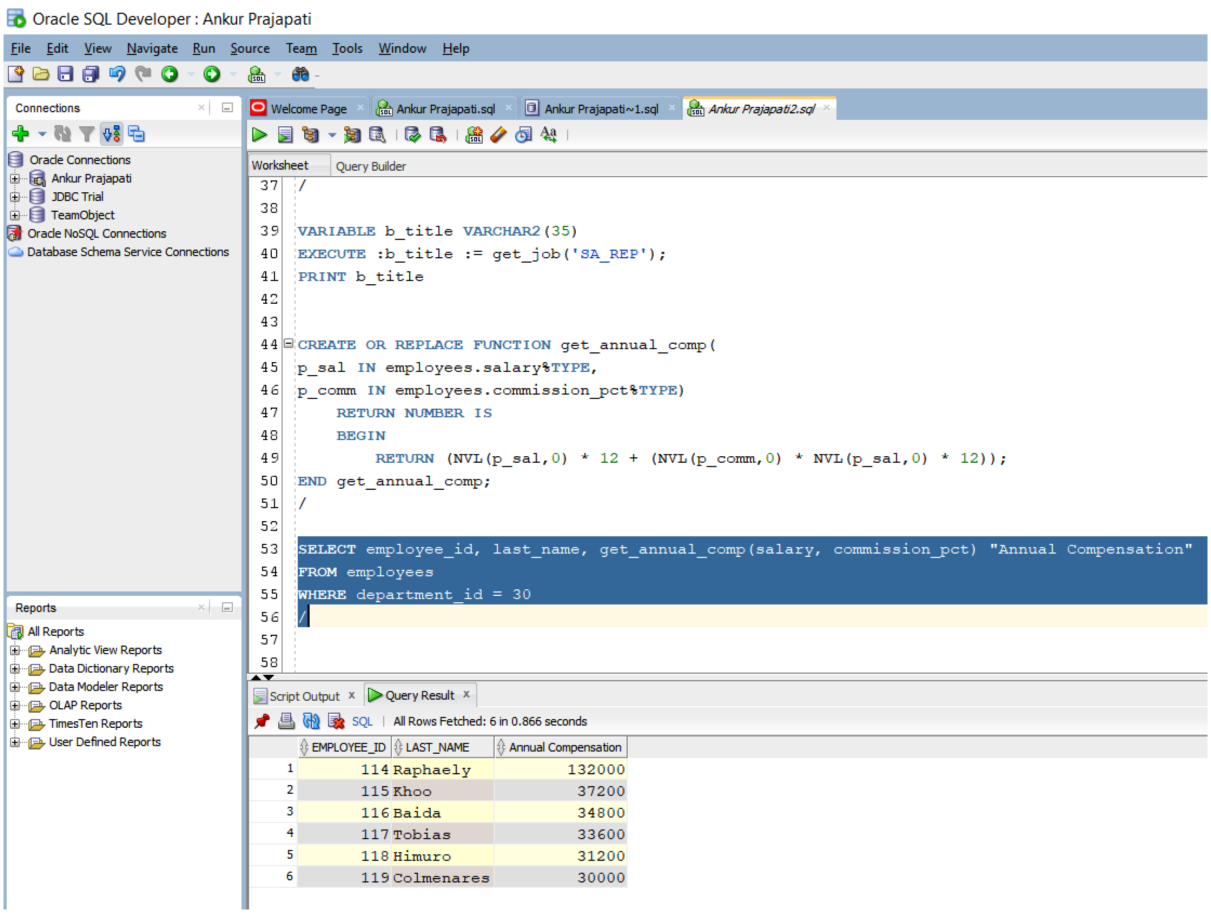
2. GET_ANNUAL_COMP Function:



```
    FROM jobs
    WHERE job_id = p_jobid;
    RETURN v_title;
END get_job;
/

VARIABLE b_title VARCHAR2(35)
EXECUTE :b_title := get_job('SA_REP');
PRINT b_title


CREATE OR REPLACE FUNCTION get_annual_comp(
p_sal  IN employees.salary%TYPE,
p_comm  IN employees.commission_pct%TYPE)
    RETURN NUMBER IS
    BEGIN
        RETURN (NVL(p_sal,0) * 12 + (NVL(p_comm,0) * NVL(p_sal,0) * 12));
END get_annual_comp;
/
```

```
PL/SQL procedure successfully completed.


B_TITLE
--------------------------------------------------------------------------------
Sales Representative


Function GET_ANNUAL_COMP compiled
```

Get_annual_comp is compiled. And it returns annual compound salary.
Note that here NVL will ignore null values and it converts it to 0.



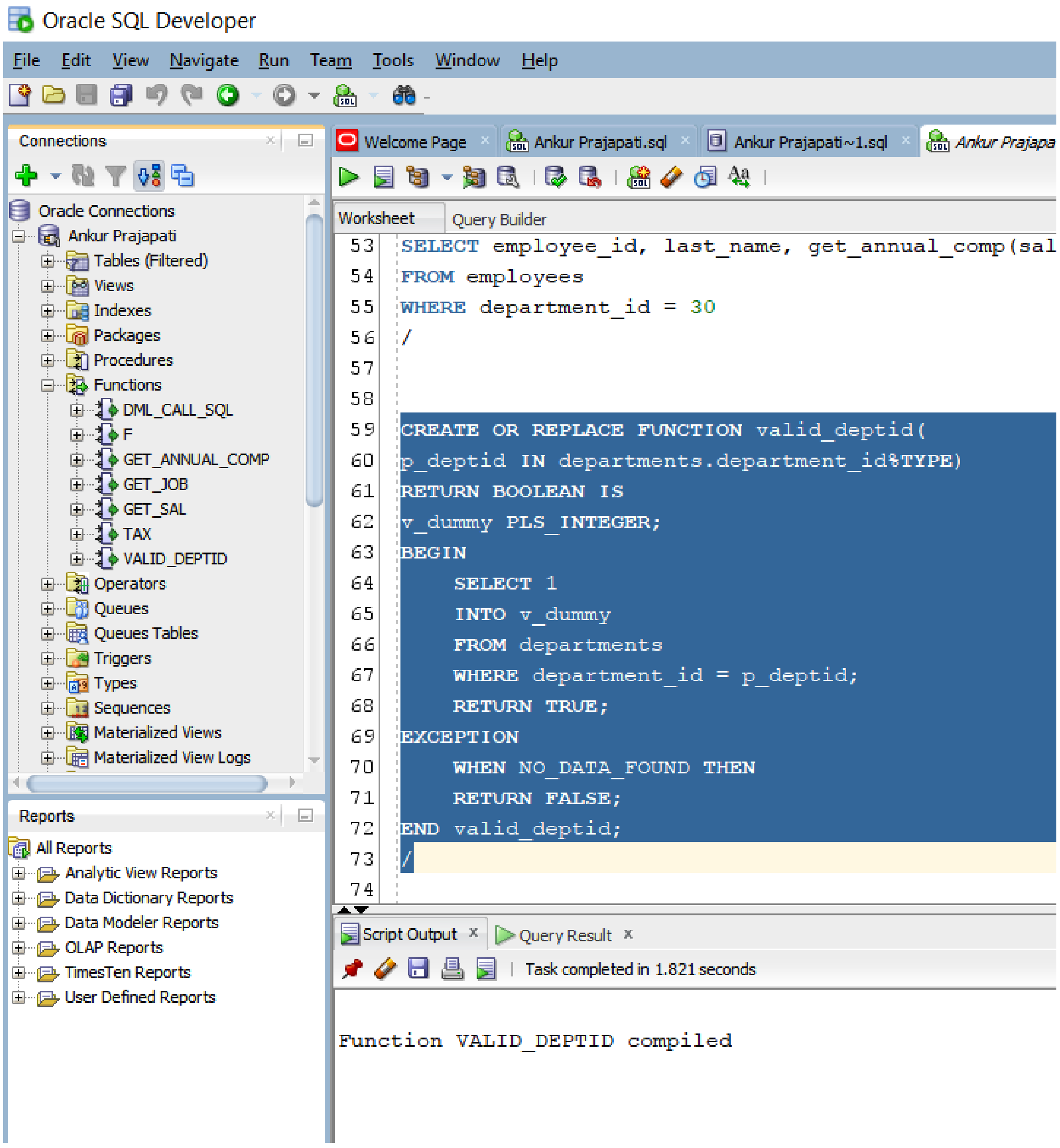
```
/

VARIABLE b_title VARCHAR2(35)
EXECUTE :b_title := get_job('SA_REP');
PRINT b_title


CREATE OR REPLACE FUNCTION get_annual_comp(
p_sal IN employees.salary%TYPE,
p_comm IN employees.commission_pct%TYPE)
    RETURN NUMBER IS
    BEGIN
        RETURN (NVL(p_sal,0) * 12 + (NVL(p_comm,0) * NVL(p_sal,0) * 12));
END get_annual_comp;
/

SELECT employee_id, last_name, get_annual_comp(salary, commission_pct) "Annual Compensation"
FROM employees
WHERE department_id = 30
/
```

All Rows Fetched: 6 in 0.866 seconds

|   | EMPLOYEE_ID | LAST_NAME | Annual Compensation |
|---|---|---|---|
| 1 | 114 | Raphaely | 132000 |
| 2 | 115 | Khoo | 37200 |
| 3 | 116 | Baida | 34800 |
| 4 | 117 | Tobias | 33600 |
| 5 | 118 | Himuro | 31200 |
| 6 | 119 | Colmenares | 30000 |

Fetching data of employees with id, last name and annual compensation where department id is 30.

3. ADD_EMPLOYEE:



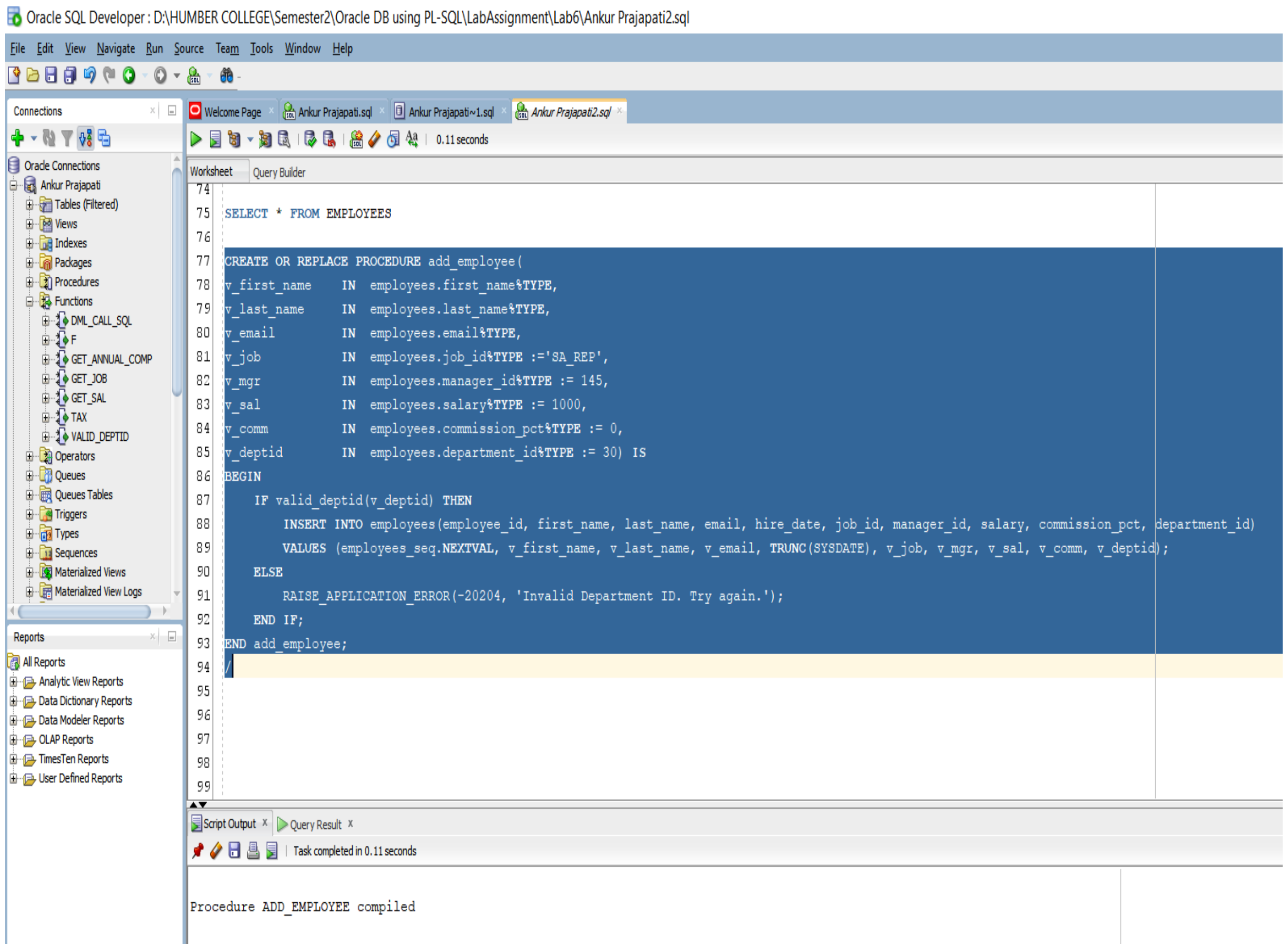
```
SELECT employee_id, last_name, get_annual_comp(sal
FROM employees
WHERE department_id = 30
/


CREATE OR REPLACE FUNCTION valid_deptid(
p_deptid IN departments.department_id%TYPE)
RETURN BOOLEAN IS
v_dummy PLS_INTEGER;
BEGIN
    SELECT 1
    INTO v_dummy
    FROM departments
    WHERE department_id = p_deptid;
    RETURN TRUE;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
    RETURN FALSE;
END valid_deptid;
/
```

```
Function VALID_DEPTID compiled
```

Creation of valid_deptid function.

We are selecting 1 into v-dummy which is PLS_INTEGER from departments table if the department id is there with given department id(p_deptid).

There is an Exception block is also included if we don’t have data then it will return false.



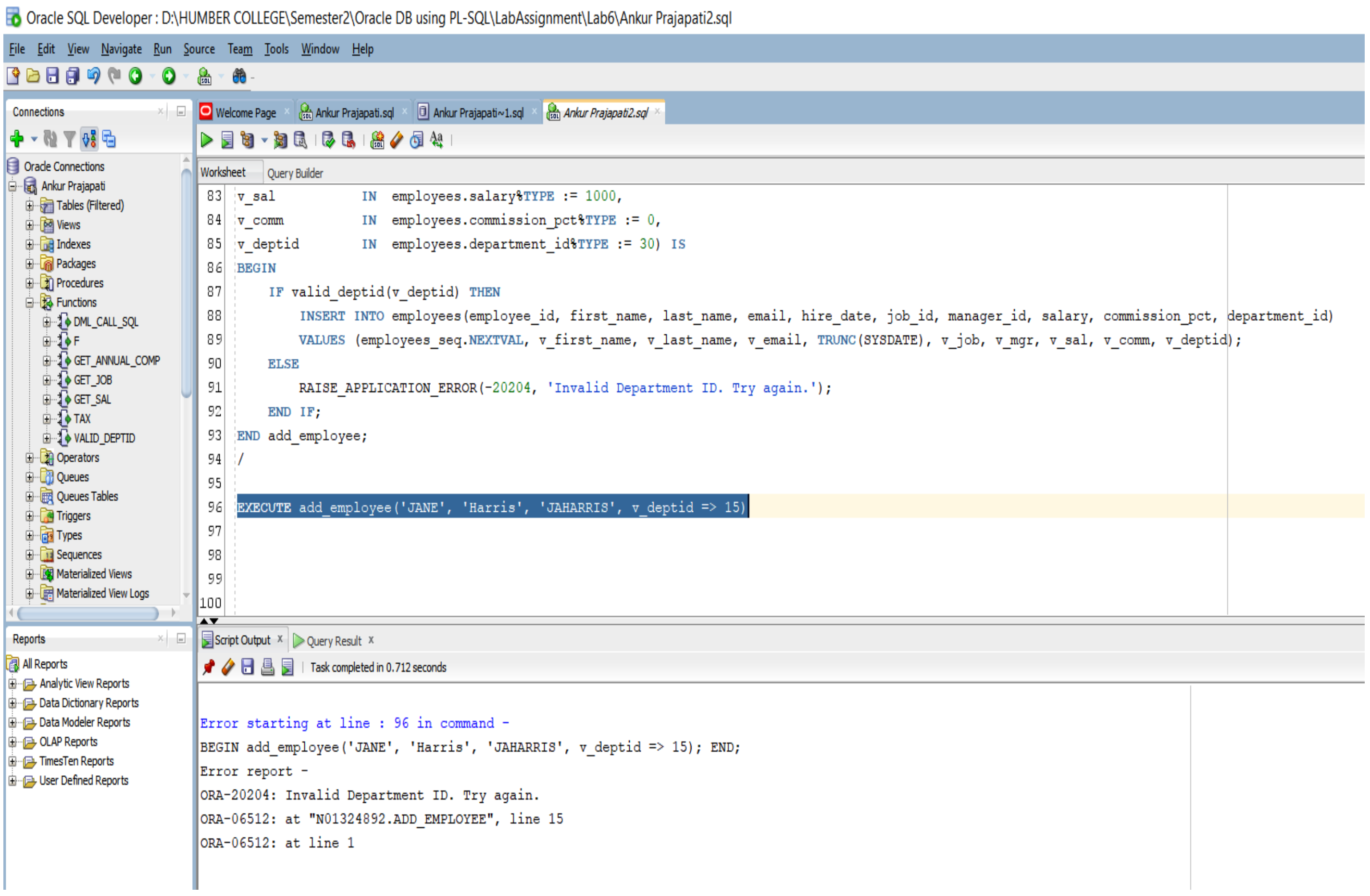
Oracle SQL Developer : D:\HUMBER COLLEGE\Semester2\Oracle DB using PL-SQL\LabAssignment\Lab6\Ankur Prajapati2.sql

```
SELECT * FROM EMPLOYEES

CREATE OR REPLACE PROCEDURE add_employee(
v_first_name    IN  employees.first_name%TYPE,
v_last_name     IN  employees.last_name%TYPE,
v_email         IN  employees.email%TYPE,
v_job           IN  employees.job_id%TYPE :='SA_REP',
v_mgr           IN  employees.manager_id%TYPE := 145,
v_sal           IN  employees.salary%TYPE := 1000,
v_comm          IN  employees.commission_pct%TYPE := 0,
v_deptid        IN  employees.department_id%TYPE := 30) IS
BEGIN
    IF valid_deptid(v_deptid) THEN
        INSERT INTO employees(employee_id, first_name, last_name, email, hire_date, job_id, manager_id, salary, commission_pct, department_id)
        VALUES (employees_seq.NEXTVAL, v_first_name, v_last_name, v_email, TRUNC(SYSDATE), v_job, v_mgr, v_sal, v_comm, v_deptid);
    ELSE
        RAISE_APPLICATION_ERROR(-20204, 'Invalid Department ID. Try again.');
    END IF;
END add_employee;
/
```

Task completed in 0.11 seconds

```
Procedure ADD_EMPLOYEE compiled
```

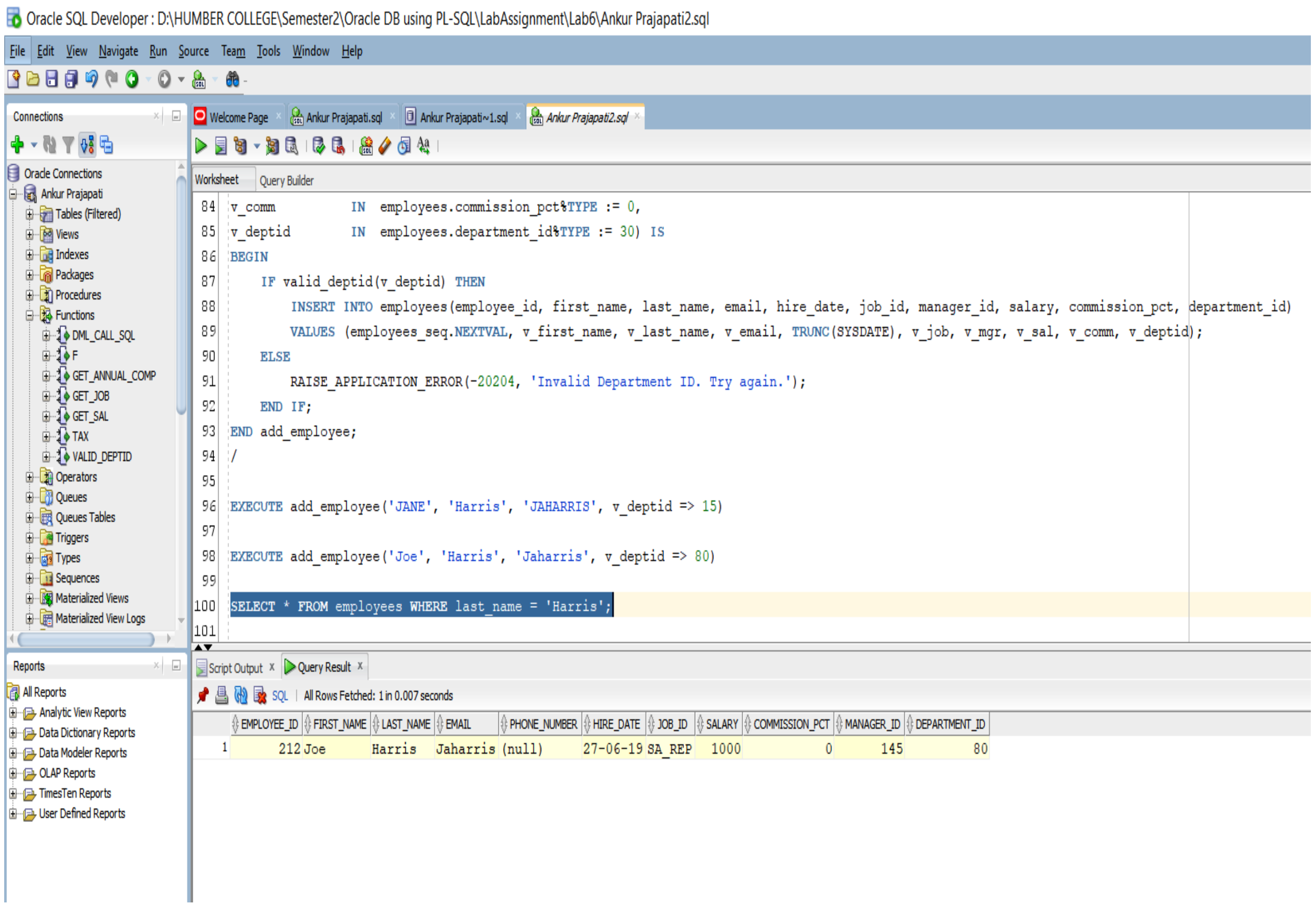
Creation of add_employee procedure. First, we declared all the formal parameters using data type from employees table.
In executable section, there is one IF condition which checks if the valid_deptid(v_deptid) is true or not. If it is true it will insert values into employees table. If it is not true then it will raise an application error notifying Invalid Department ID. Try again.



```
v_sal          IN  employees.salary%TYPE := 1000,
v_comm         IN  employees.commission_pct%TYPE := 0,
v_deptid       IN  employees.department_id%TYPE := 30) IS
BEGIN
    IF valid_deptid(v_deptid) THEN
        INSERT INTO employees(employee_id, first_name, last_name, email, hire_date, job_id, manager_id, salary, commission_pct, department_id)
        VALUES (employees_seq.NEXTVAL, v_first_name, v_last_name, v_email, TRUNC(SYSDATE), v_job, v_mgr, v_sal, v_comm, v_deptid);
    ELSE
        RAISE_APPLICATION_ERROR(-20204, 'Invalid Department ID. Try again.');
    END IF;
END add_employee;
/

EXECUTE add_employee('JANE', 'Harris', 'JAHARRIS', v_deptid => 15)
```

```
Error starting at line : 96 in command -
BEGIN add_employee('JANE', 'Harris', 'JAHARRIS', v_deptid => 15); END;
Error report -
ORA-20204: Invalid Department ID. Try again.
ORA-06512: at "N01324892.ADD_EMPLOYEE", line 15
ORA-06512: at line 1
```

Here add_employee is executed and due to department id miss matching it gives invalid department id.



```
v_comm       IN  employees.commission_pct%TYPE := 0,
v_deptid     IN  employees.department_id%TYPE := 30) IS
BEGIN
    IF valid_deptid(v_deptid) THEN
        INSERT INTO employees(employee_id, first_name, last_name, email, hire_date, job_id, manager_id, salary, commission_pct, department_id)
        VALUES (employees_seq.NEXTVAL, v_first_name, v_last_name, v_email, TRUNC(SYSDATE), v_job, v_mgr, v_sal, v_comm, v_deptid);
    ELSE
        RAISE_APPLICATION_ERROR(-20204, 'Invalid Department ID. Try again.');
    END IF;
END add_employee;
/

EXECUTE add_employee('JANE', 'Harris', 'JAHARRIS', v_deptid => 15)

EXECUTE add_employee('Joe', 'Harris', 'Jaharris', v_deptid => 80)

SELECT * FROM employees WHERE last_name = 'Harris';

```

| | EMPLOYEE_ID | FIRST_NAME | LAST_NAME | EMAIL | PHONE_NUMBER | HIRE_DATE | JOB_ID | SALARY | COMMISSION_PCT | MANAGER_ID | DEPARTMENT_ID |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 212 | Joe | Harris | Jaharris | (null) | 27-06-19 | SA_REP | 1000 | 0 | 145 | 80 |

Adding new record here in table.

It prints the inserted record here.