Class work on Lab 08 and 09

NOTE: write all the commands (each command in a new line) for the following tasks sequentially in a text file and save the file as [name].sql.

First RUN the script create2006.sql and then perform the following tasks using the tables created by create2006.sql script.

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.

```
IF SQL%NOTFOUND THEN
   RAISE_APPLICATION_ERROR(-20202, 'No job updated.');
 END IF;
```

2. In Create a procedure called DEL_JOB to delete a job. Provide an exception handling section so that if there is nothing to delete as ordered.

```
IF SQL%NOTFOUND THEN
   RAISE_APPLICATION_ERROR(-20203, 'No jobs deleted.');
 END IF;
```

3. Create and invoke a function named GET_JOB to return a job title.
   a. Create and compile a function called GET_JOB to return a job title.
   b. Create a VARCHAR2 host variable called TITLE, allowing a length of 35 characters. Invoke the function with SA_REP job ID to return the value in the host variable. Print the host variable to view the result. This can be achieved as follows and illustrates a different way of calling functions:

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

4. Create a function called GET_ANNUAL_COMP to return the annual salary computed from an employee's monthly salary and commission passed as parameters.
   Develop and store the function GET_ANNUAL_COMP, accepting parameter values for monthly salary and commission. Either or both values passed can be NULL, but the function should still return a non-NULL annual salary. Use the following basic formula to calculate the annual salary:
   (salary*12) + (commission_pct*salary*12)
5. Use the GET_ANNUAL_COMP function in a SELECT statement against the EMPLOYEES table for employees in department id 90.