CTEC 324 SCRIPTS:

**Intro Notes and Scripts:**

https://www.tutorialspoint.com/plsql/index.htm

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

select \* from cat;

drop table students cascade constraints;

create table students(

studentid varchar2(3),

lastname varchar2(15),

gpa number(5,2),

primary key (studentid));

desc students;

insert into students values ('555', 'Bryant', 4.00);

insert into students values ('777', 'Cupp', 2.33);

insert into students values ('888', 'Lacy', 2.10);

insert into students values ('999', 'Edwards', 2.43);

select \* from students;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

set serveroutput on;

begin

dbms\_output.put\_line('Hello World');

end;

/

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- Variables declare, assign and display

set SERVEROUTPUT ON SIZE UNLIMITED;

declare

lastname varchar2(15);

gpa number(5,2);

begin

lastname := 'Smith';

gpa := 3.75;

dbms\_output.put\_line('Last Name : ' || lastname );

dbms\_output.put\_line('GPA : ' || gpa );

end;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- using select into

declare

lastname varchar2(15);

gpa number(5,2);

begin

select lastname, gpa into lastname, gpa

from students

where studentid = 1;

dbms\_output\_put\_line('Last Name : ' || lastname );

dbms\_output\_put\_line('GPA : ' || gpa );

end;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- using datatype from fields in an exisiting table

declare

lastname students.lastname%TYPE;

gpa students.gpa%TYPE;

begin

select lastname, gpa into lastname, gpa

from students

where studentid = 1;

dbms\_output\_put\_line('Last Name : ' || lastname );

dbms\_output\_put\_line('GPA : ' || gpa );

end;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- use ROWTYPE

declare

student students%ROWTYPE;

begin

select \* into student from students

where studentid = 555;

dbms\_output.put\_line('Last Name : ' || student.lastname );

dbms\_output.put\_line('GPA : ' || student.gpa );

end;

**IF STATEMENT AND UPDATE**

https://www.tutorialspoint.com/plsql/index.htm

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

select \* from cat;

drop table students cascade constraints;

create table students(

studentid varchar2(3),

lastname varchar2(15),

gpa number(5,2),

primary key (studentid));

desc students;

insert into students values ('555', 'Bryant', 4.00);

insert into students values ('777', 'Cupp', 2.33);

insert into students values ('888', 'Lacy', 2.10);

insert into students values ('999', 'Edwards', 2.43);

select \* from students;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

set serveroutput on;

begin

dbms\_output.put\_line('Hello World');

end;

/

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- Variables â€“ declare, assign and display

set SERVEROUTPUT ON SIZE UNLIMITED;

declare

lastname varchar2(15);

gpa number(5,2);

begin

lastname := 'Smith';

gpa := 3.75;

dbms\_output.put\_line('Last Name : ' || lastname );

dbms\_output.put\_line('GPA : ' || gpa );

end;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- using select into

declare

lastname varchar2(15);

gpa number(5,2);

begin

select lastname, gpa into lastname, gpa

from students

where studentid = 1;

dbms\_output\_put\_line('Last Name : ' || lastname );

dbms\_output\_put\_line('GPA : ' || gpa );

end;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- using datatype from fields in an exisiting table

declare

lastname students.lastname%TYPE;

gpa students.gpa%TYPE;

begin

select lastname, gpa into lastname, gpa

from students

where studentid = 1;

dbms\_output\_put\_line('Last Name : ' || lastname );

dbms\_output\_put\_line('GPA : ' || gpa );

end;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- use ROWTYPE

declare

student students%ROWTYPE;

begin

select \* into student from students

where studentid = 555;

dbms\_output.put\_line('Last Name : ' || student.lastname );

dbms\_output.put\_line('GPA : ' || student.gpa );

end;

**LOOPS**

-- count and display # of students and today's date

set serveroutput on

DECLARE

numStudents NUMBER;

BEGIN

SELECT COUNT(studentid) INTO numStudents

FROM students;

dbms\_output.put\_line('Number of students is ' || numStudents );

dbms\_output.put\_line('Todays Date: ' || SYSDATE );

END;

/

---------------------------------

-- adding a new table for in-class demo

drop table dailyStudentCount cascade constraints;

create table dailyStudentCount

( theDate DATE,

numStudents number(2),

primary key (theDate));

---------------------------------

-- PL- SQL for in-class demo: storing the "daily student count"

set serveroutput on;

DECLARE

v\_numStudents NUMBER;

BEGIN

SELECT COUNT(studentid) INTO v\_numStudents

FROM students;

dbms\_output.put\_line('Number of students is ' || v\_numStudents );

dbms\_output.put\_line('Todays Date: ' || SYSDATE );

INSERT INTO dailyStudentCount(theDate, numStudents)

VALUES(SYSDATE, v\_numStudents);

END;

/

---------------------------------

-- adding a new table for lab 4

drop table studentAccounts cascade constraints;

create table studentAccounts

( accountid number(2),

startDate DATE,

username varchar2(20),

primary key (accountid));

**CURSORS**

set SERVEROUTPUT ON SIZE UNLIMITED;

DECLARE

v\_studentid students.studentid%TYPE;

v\_lastname students.lastname%TYPE;

v\_gpa students.gpa%TYPE;

CURSOR stud\_cursor IS

select studentid, lastname, gpa

from students;

BEGIN

OPEN stud\_cursor;

LOOP

fetch stud\_cursor into v\_studentid, v\_lastname, v\_gpa;

EXIT WHEN stud\_cursor%NOTFOUND;

dbms\_output.put\_line('Student ID: ' || v\_studentid);

dbms\_output.put\_line('Last Name : ' || v\_lastname);

dbms\_output.put\_line('GPA : ' || v\_gpa);

END LOOP;

CLOSE stud\_cursor;

END;

/

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

2ND EXAMPLE

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

set SERVEROUTPUT ON SIZE UNLIMITED;

DECLARE

r\_product products%rowtype;

CURSOR c\_product (low\_price NUMBER, high\_price NUMBER)

IS

SELECT \*

FROM products

WHERE list\_price BETWEEN low\_price AND high\_price;

BEGIN

-- show mass products

dbms\_output.put\_line('Mass products: ');

OPEN c\_product(50,100);

LOOP

FETCH c\_product INTO r\_product;

EXIT WHEN c\_product%notfound;

dbms\_output.put\_line(r\_product.product\_name || ': ' ||r\_product.list\_price);

END LOOP;

CLOSE c\_product;

-- show luxury products

dbms\_output.put\_line('Luxury products: ');

OPEN c\_product(800,1000);

LOOP

FETCH c\_product INTO r\_product;

EXIT WHEN c\_product%notfound;

dbms\_output.put\_line(r\_product.product\_name || ': ' ||r\_product.list\_price);

END LOOP;

CLOSE c\_product;

END;

/

**FUNCTIONS**

drop function circle\_area;

-- Function for caclulating the area of a circle

CREATE OR REPLACE FUNCTION circle\_area(radius NUMBER)

RETURN NUMBER IS

pi CONSTANT NUMBER (7,3) := 3.14;

area NUMBER (7,3);

BEGIN

-- Area of a Circle pi \* r \* r;

area := pi \* (radius \* radius);

RETURN area;

END;

/

**TRIGGERS**

-- trigger for students table - insert or delete or update

create or replace trigger tr\_students

before insert or delete or update on students

for each row

enable

declare

v\_user varchar2(20);

begin

select user INTO v\_user from dual;

if INSERTING then

dbms\_output.put\_line (' One row inserted by ' || v\_user);

elsif DELETING then

dbms\_output.put\_line (' One row deleted by ' || v\_user);

elsif UPDATING then

dbms\_output.put\_line (' One row updated by ' || v\_user);

else

dbms\_output.put\_line (' Error opertion by ' || v\_user);

end if;

end;

-- sql statements to check trigger

set serveroutput on;

select \* from students;

insert into students values ('777', 'Packer',3.25);

select \* from students;

delete from students where studentid = '777';

select \* from students;