Jay Fuller

Final Project

```
--Drop Tables:
drop table order_details;
drop table orders;
drop table inventory;
drop table product;
drop table warehouse;
drop table dependents;
drop table employee;
drop table department;
drop table benefits;
drop table supplier;
drop table shipping;
drop table customer;

--Create Tables:
CREATE TABLE CUSTOMER
(CUST# INT NOT NULL ,
C_FNAME VARCHAR(20) NOT NULL ,
C_LNAME VARCHAR(20) NOT NULL ,
STREET VARCHAR(30) ,
CITY VARCHAR(20) ,
STATE CHAR(2) ,
ZIP CHAR(5) ,
PRIMARY KEY (CUST#));

--Customer
Insert into customer values('123','John','Smith','2000 Main Street','Oxford','MS','38655');
Insert into customer values('456','Leslie','Jones','555 North Road','Oxford','MS','38655');
Insert into customer values('1','Boss','Hog','1 Big House Street','Jackson','MS','39216');
Insert into customer values('13','John','Currence','13 Big Bad Breakfast Avenue','Oxford','MS','38655');
Insert into customer values('24','Yaboi','Tierney','15 Hospital Street','Chicago','IL','60605');
Insert into customer values('574','Jimmbo','Jimmy','256 St. Andrews Drive','Jackson','MS','39211');
Insert into customer values('7','Ash','Ketchum','151 Catch Em All Lane','Memphis','TN','37501');
Insert into customer values('99','Mike','Wazowsky','300 Monster Street','Oxford','MS','38655');
Insert into customer values('675','Houston','Nutt','125 Football Street','Tupelo','MS','38801');
Insert into customer values('111','Danny','Devito','250 Congress Park Drive','Boca Raton','FL','33445');

CREATE TABLE SHIPPING
(SHIPPING# INT NOT NULL ,
S_COMPANY VARCHAR(30) NOT NULL ,
S_COST INT NOT NULL ,
S_PHONE CHAR(10) ,
S_METHOD VARCHAR(20) ,
PRIMARY KEY (SHIPPING#));

--Shipping
Insert into shipping values('1','CompanyA','100','5551111111','Ground');
Insert into shipping values('2','CompanyA','200','5552222222','Ground');
Insert into shipping values('3','CompanyA','300','5553333333','Air');
Insert into shipping values('4','CompanyB','300','5554444444','Air');
Insert into shipping values('5','CompanyB','100','5556666666','Ground');
Insert into shipping values('6','CompanyA','200','8889999999','Ground');
Insert into shipping values('7','CompanyC','500','8887777777','Air');
Insert into shipping values('8','CompanyC','100','8885555555','Train');
Insert into shipping values('9','CompanyA','200','4449999999','Train');
Insert into shipping values('10','CompanyB','600','4448888888','Air');

CREATE TABLE SUPPLIER
(SUPPLIER_ID INT NOT NULL ,
SUPP_NAME VARCHAR(20) NOT NULL ,
SUPP_CITY VARCHAR(20) ,
SUPP_STATE CHAR(2) ,
SUPP_ZIP CHAR(5) ,
PRIMARY KEY (SUPPLIER_ID));

--Supplier
Insert into supplier values('22','SupplierA','Jackson','MS','39211');
Insert into supplier values('24','SupplierB','Jackson','MS','39211');
Insert into supplier values('26','SupplierC','Madison','MS','39110');
Insert into supplier values('28','SupplierD','Madison','MS','39110');
Insert into supplier values('33','SupplierE','Memphis','TN','37501');
Insert into supplier values('35','SupplierF','Memphis','TN','37501');
Insert into supplier values('37','SupplierG','Nashville','TN','37011');
Insert into supplier values('39','SupplierH','Boca Raton','FL','33445');
Insert into supplier values('44','SupplierI','Boca Raton','FL','33445');
Insert into supplier values('46','SupplierJ','Delray Beach','FL','33484');

CREATE TABLE BENEFITS
(POLICY# INT NOT NULL ,
HEALTHCARE VARCHAR(40) NOT NULL ,
DENTAL VARCHAR(3) ,
EYES VARCHAR(3) ,
PRESCRIPTION VARCHAR(10) ,
FAMILY_COVERAGE VARCHAR(10) ,
PRIMARY KEY (POLICY#));
```

```
--Benefits
Insert into benefits values('101','UnitedHealth','NO','NO','PARTIAL','SPOUSE');
Insert into benefits values('102','UnitedHealth','NO','YES','PARTIAL','ALL');
Insert into benefits values('103','UnitedHealth','YES','YES','FULL','ALL');
Insert into benefits values('201','Aetna','NO','NO','PARTIAL','SPOUSE');
Insert into benefits values('202','Aetna','NO','YES','PARTIAL','ALL');
Insert into benefits values('203','Aetna','YES','YES','FULL','ALL');
Insert into benefits values('301','Blue Cross/Blue Shield','NO','NO','PARTIAL','SPOUSE');
Insert into benefits values('302','Blue Cross/Blue Shield','NO','YES','PARTIAL','ALL');
Insert into benefits values('303','Blue Cross/Blue Shield','YES','NO','FULL','ALL');
Insert into benefits values('304','Blue Cross/Blue Shield','YES','YES','FULL','ALL');

CREATE TABLE DEPARTMENT
(DEPT_ID INT NOT NULL ,
DEPT_NAME VARCHAR(30) NOT NULL ,
DEPT_REGION VARCHAR(20) ,
MANAGER_ID INT NOT NULL ,
MGR_START_DATE VARCHAR(10) ,
PRIMARY KEY (DEPT_ID));

--Department
Insert into department values('99','Management','South','1234','2004-05-22');
Insert into department values('98','Administration','South','2345','2004-05-23');
Insert into department values('97','Research','South','3456','2006-09-19');
Insert into department values('96','Purchasing','East','4567','2004-12-03');
Insert into department values('95','Marketing','North','5678','2005-10-14');
Insert into department values('94','Human Resource','South','6789','2008-04-14');
Insert into department values('93','Accounting','South','7890','2004-06-26');
Insert into department values('92','Finance','North','8901','2004-08-22');
Insert into department values('91','Sales','East','9012','2006-01-29');
Insert into department values('90','Information Technology','South','4321','2009-10-31');

CREATE TABLE EMPLOYEE
(EMP_ID INT NOT NULL ,
EMP_FNAME VARCHAR(20) NOT NULL ,
EMP_LNAME VARCHAR(20) NOT NULL ,
SALARY INT ,
EMP_CITY VARCHAR(20) ,
EMP_STATE CHAR(2) ,
EMP_DEPT_NUM INT NOT NULL ,
POLICY# INT NOT NULL ,
PRIMARY KEY (EMP_ID) ,
FOREIGN KEY (EMP_DEPT_NUM) REFERENCES DEPARTMENT(DEPT_ID));

--Employee
Insert into employee values('1234','Aristides','Aquino','60000','Jackson','MS','99','102');
Insert into employee values('2345','Billy','Hamilton','65000','Jackson','MS','99','103');
Insert into employee values('3456','Curtis','Granderson','63000','Jackson','MS','98','102');
Insert into employee values('4567','Dee','Gordon','55000','Oxford','MS','97','102');
Insert into employee values('5678','Eugenio','Suarez','70000','Oxford','MS','93','302');
Insert into employee values('6789','Freddie','Freeman','55000','Oxford','MS','90','302');
Insert into employee values('7891','Giancarlo','Stanton','68000','Tupelo','MS','91','303');
Insert into employee values('8901','Trea','Turner','75000','Boca Raton','FL','96','304');
Insert into employee values('9012','Byron','Buxton','45000','Boca Raton','FL','91','201');
Insert into employee values('4321','Joey','Freshwater','99000','Boca Raton','FL','99','203');

CREATE TABLE DEPENDENTS
(EMP_ID INT NOT NULL REFERENCES EMPLOYEE(EMP_ID) ,
DEP_NAME VARCHAR(20) NOT NULL ,
SEX CHAR(1) ,
DOB VARCHAR(10) ,
RELATIONSHIP VARCHAR(10) NOT NULL ,
PRIMARY KEY (EMP_ID, DEP_NAME));

--Dependents
Insert into dependents values('1234','Allen','M','1989-10-31','Son');
Insert into dependents values('1234','Scott','M','1992-09-26','Son');
Insert into dependents values('1234','Ann','F','1965-04-13','Spouse');
Insert into dependents values('3456','Charles','M','1970-01-21','Spouse');
Insert into dependents values('5678','Ellen','F','1995-06-24','Daughter');
Insert into dependents values('5678','Eileen','F','1960-12-31','Spouse');
Insert into dependents values('6789','Frances','F','1992-03-09','Daughter');
Insert into dependents values('4321','Jimmy','M','1991-10-15','Son');
Insert into dependents values('4321','John','M','1994-12-26','Daughter');
Insert into dependents values('4321','Janice','F','1963-02-14','Spouse');

CREATE TABLE WAREHOUSE
(WAREHOUSE# INT NOT NULL ,
W_SIZE VARCHAR(20) NOT NULL ,
W_CITY VARCHAR(20) ,
W_STATE CHAR(2) ,
W_ZIP CHAR(5) ,
PRIMARY KEY (WAREHOUSE#));

--Warehouse
Insert into warehouse values('111','10000 sqft','Oxford','MS','38655');
Insert into warehouse values('222','20000 sqft','Oxford','MS','38655');
Insert into warehouse values('333','30000 sqft','Oxford','MS','38655');
Insert into warehouse values('444','40000 sqft','Jackson','MS','39216');
Insert into warehouse values('555','50000 sqft','Jackson','MS','39216');
Insert into warehouse values('666','60000 sqft','Jackson','MS','39216');
Insert into warehouse values('777','50000 sqft','Tupelo','MS','38801');
Insert into warehouse values('888','40000 sqft','Tupelo','MS','38801');
Insert into warehouse values('999','30000 sqft','Boca Raton','FL','33445');
Insert into warehouse values('1000','20000 sqft','Boca Raton','FL','33445');

CREATE TABLE PRODUCT
(PRODUCT# INT NOT NULL ,
P_NAME VARCHAR(20) NOT NULL ,
P_COLOR VARCHAR(20) ,
P_WEIGHT INT ,
SUPPLIER_ID INT NOT NULL REFERENCES SUPPLIER(SUPPLIER_ID),
PRIMARY KEY (PRODUCT#));

--Product
Insert into product values('1111','ProductX','Red','10','22');
Insert into product values('2222','ProductY','Blue','10','22');
Insert into product values('3333','ProductZ','Gray','40','44');
Insert into product values('4444','ProductA','Orange','30','33');
Insert into product values('5555','ProductB','Yellow','40','33');
Insert into product values('6666','ProductC','Green','80','33');
Insert into product values('7777','ProductJ','Purple','10','28');
Insert into product values('8888','ProductK','White','80','28');
Insert into product values('9999','ProductL','Black','45','24');
Insert into product values('10000','ProductF','Powder Blue','15','46');

CREATE TABLE INVENTORY
(WAREHOUSE# INT NOT NULL REFERENCES WAREHOUSE(WAREHOUSE#) ,
PRODUCT# INT NOT NULL REFERENCES PRODUCT(PRODUCT#) ,
QTY_IN_STOCK INT NOT NULL ,
QTY_AVAILABLE INT ,
QTY_ORDERED INT ,
PRIMARY KEY (WAREHOUSE#, PRODUCT#));

--Inventory
Insert into inventory values('111','1111','100','80','100');
Insert into inventory values('111','2222','200','200','40');
Insert into inventory values('111','3333','300','150','300');
Insert into inventory values('333','4444','150','100','25');
Insert into inventory values('444','5555','200','200','0');
Insert into inventory values('444','6666','75','75','225');
Insert into inventory values('555','5555','100','50','100');
Insert into inventory values('555','7777','700','400','500');
Insert into inventory values('888','9999','250','200','135');
Insert into inventory values('888','10000','150','55','123');

CREATE TABLE ORDERS
(ORDER# INT NOT NULL ,
O_DATE VARCHAR(20) ,
CUST# INT NOT NULL REFERENCES CUSTOMER(CUST#) ,
EMP_ID INT NOT NULL REFERENCES EMPLOYEE(EMP_ID) ,
SHIPPING# INT NOT NULL REFERENCES SHIPPING(SHIPPING#) ,
PRIMARY KEY (ORDER#));

--Orders
Insert into orders values('1','9/2019','123','1234','1');
Insert into orders values('2','9/2019','123','1234','2');
Insert into orders values('3','8/2019','123','1234','6');
Insert into orders values('4','10/2019','13','4567','4');
Insert into orders values('5','9/2019','13','4567','10');
Insert into orders values('6','9/2019','456','8901','7');
Insert into orders values('7','10/2019','456','8901','7');
Insert into orders values('8','10/2019','99','4321','3');
Insert into orders values('9','9/2019','99','2345','6');
Insert into orders values('10','10/2019','13','7890','10');

CREATE TABLE ORDER_DETAILS
(ORDER# INT NOT NULL REFERENCES ORDERS(ORDER#) ,
PRODUCT# INT NOT NULL REFERENCES PRODUCT(PRODUCT#) ,
UNIT_PRICE INT ,
QTY_P_PART INT ,
TOTAL_P_PART INT ,
PRIMARY KEY (ORDER#, PRODUCT#));

--Order Details
Insert into order_details values('1','1111','50','20','1000');
Insert into order_details values('2','1111','20','20','400');
Insert into order_details values('3','3333','25','10','250');
Insert into order_details values('4','6666','70','35','2450');
Insert into order_details values('5','7777','15','40','600');
Insert into order_details values('6','3333','40','10','400');
Insert into order_details values('7','2222','85','15','1275');
Insert into order_details values('8','4444','15','30','450');
Insert into order_details values('9','9999','50','50','2500');
Insert into order_details values('10','1111','33','20','660');
```

```
--1 inserts a new employee into the database from user input
--returns new employee info and their healthcare provider

CREATE OR REPLACE FUNCTION find_fname (p_emp_id employee.emp_id%type)
RETURN employee.emp_fname%type
IS
ret_name employee.emp_fname%type;

BEGIN
select emp_fname into ret_name
from employee
where p_emp_id = emp_id;
return ret_name;
END;

/

CREATE OR REPLACE FUNCTION find_lname (p_emp_id employee.emp_id%type)
RETURN employee.emp_lname%type
IS
ret_name employee.emp_lname%type;

BEGIN
select emp_lname into ret_name
from employee
where p_emp_id = emp_id;
return ret_name;
END;
```

```
/

CREATE OR REPLACE FUNCTION find_sal (p_emp_id employee.emp_id%type)
RETURN employee.salary%type
IS
ret_sal employee.salary%type;

BEGIN
select salary into ret_sal
from employee
where p_emp_id = emp_id;
return ret_sal;
END;

/

CREATE OR REPLACE FUNCTION find_city (p_emp_id employee.emp_id%type)
RETURN employee.emp_city%type
IS
ret_city employee.emp_city%type;

BEGIN
select emp_city into ret_city
from employee
where p_emp_id = emp_id;
return ret_city;
END;
```

```
/

CREATE OR REPLACE FUNCTION find_state (p_emp_id employee.emp_id%type)
RETURN employee.emp_state%type
IS
ret_state employee.emp_state%type;

BEGIN
select emp_state into ret_state
from employee
where p_emp_id = emp_id;
return ret_state;
END;

/

CREATE OR REPLACE FUNCTION find_benefit (p_emp_id employee.emp_id%type)
RETURN benefits.healthcare%type
IS
ret_healthcare benefits.healthcare%type;

BEGIN
select b.healthcare into ret_healthcare
from benefits b, employee e
where p_emp_id = e.emp_id
and b.policy# = e.policy#;
return ret_healthcare;
```

```
END;

/

SET SERVEROUTPUT ON;
DECLARE
eempid  employee.emp_id%type;
finame  employee.emp_fname%type;
laname  employee.emp_lname%type;
empid   employee.emp_id%type;
empsal  employee.salary%type;
empcity employee.emp_city%type;
empstate    employee.emp_state%type;
fname   employee.emp_fname%type;
lname   employee.emp_lname%type;
health  benefits.healthcare%type;
city    employee.emp_city%type;
state   employee.emp_state%type;
sal     employee.salary%type;

BEGIN
--user inputs
finame:='&FirstName';
laname:='&LastName';
empid:='&EmployeeID';
empsal:='&Salary';
empcity:='&City';
empstate:='&State';
```

```
DBMS_OUTPUT.PUT_LINE('');
DBMS_OUTPUT.PUT_LINE('');

insert into employee values(empid, finame, laname, empsal, empcity, empstate, '93', '303');

DBMS_OUTPUT.PUT_LINE('Here is the new employee you have entered.');
DBMS_OUTPUT.PUT_LINE('');
fname:=find_fname(empid);
lname:=find_lname(empid);
sal:=find_sal(empid);
city:=find_city(empid);
state:=find_state(empid);
health:=find_benefit(empid);
DBMS_OUTPUT.PUT_LINE('Employee Name: '||trim(fname)||' '||trim(lname));
DBMS_OUTPUT.PUT_LINE('Location: '||city||' '||state);
DBMS_OUTPUT.PUT_LINE('Salary: $'||sal);
DBMS_OUTPUT.PUT_LINE('Healthcare Provider: '||health);
END;

--Delete from employee where emp_id = [whatever id user entered]

--2 discount amount calculations; enter A, B, or C for code
--enter 1-10 for order number
SET SERVEROUTPUT ON

DECLARE
discount_code CHAR(1);
```

```
total_amt order_details.total_p_part%type;
disc_amt NUMBER(5,2) :=0;
order_num order_details.order#%type;

BEGIN
  order_num :='&OrderNumber';
  discount_code :='&DiscountCode';

  select total_p_part into total_amt
  from order_details
  where order# = order_num;

  IF
    discount_code = 'A'
  THEN
    disc_amt := total_amt * .05;
  ELSIF discount_code = 'B' THEN
    disc_amt := total_amt * .15;
  ELSIF discount_code = 'C' THEN
    disc_amt := total_amt * .15;
  END IF;

DBMS_OUTPUT.PUT_LINE('Total Order Cost: '||total_amt);
DBMS_OUTPUT.PUT_LINE('Discount Code: '||discount_code);
DBMS_OUTPUT.PUT_LINE('Discount Amount: '||disc_amt);

END;
```

```
--3 Procedure for entering new department info
--Prompt user for all department info

CREATE OR REPLACE PROCEDURE Insert_Department
(p_dept_id IN department.dept_id%type,
p_dept_name IN department.dept_name%type,
p_dept_region IN department.dept_region%type,
p_manager_id IN department.manager_id%type,
p_mgr_start_date IN department.mgr_start_date%type)
IS
BEGIN
insert into department values(p_dept_id, p_dept_name, p_dept_region, p_manager_id, p_mgr_start_date);
commit;
END;
/
SET SERVEROUTPUT ON
DECLARE
deptid department.dept_id%type;
deptname department.dept_name%type;
deptregion department.dept_region%type;
deptmgrid department.manager_id%type;
deptmgrstart department.mgr_start_date%type;
BEGIN
deptid :='&DepartmentID';
deptname :='&DepartmentName';
deptregion :='&DepartmentRegion';
```

```
deptmgrid :='&ManagerID';
deptmgrstart :='&ManagerStartDate';
Insert_Department(deptid, deptname, deptregion, deptmgrid, deptmgrstart);
END;



--4 Find number of orders an employee is associated with
--Enter 1234, 2345, 3456, 4567, or 8901

CREATE OR REPLACE FUNCTION f_emp_orders (f_emp employee.emp_id%type)
RETURN orders.order#%type
IS
ret_num_orders orders.order#%type;

BEGIN

select count(order#) into ret_num_orders from orders o, employee e
where e.emp_id = o.emp_id
and e.emp_id = f_emp;

return ret_num_orders;
END;

/
SET SERVEROUTPUT ON
DECLARE
emp_num employee.emp_id%type :='&Employee_ID';
```

```
num_orders orders.order#%type;
BEGIN
   num_orders := f_emp_orders(emp_num);
   DBMS_OUTPUT.PUT_LINE('Employee Number '||emp_num||' works on '||num_orders);
END;

--5 incomplete
CREATE OR REPLACE PROCEDURE p_product
(p_product IN product.product#%type,
p_id OUT product.product#%type,
p_name OUT product.p_name%type,
p_color OUT product.p_color%type,
p_weight OUT product.p_weight%type,
p_supp OUT product.supplier_id%type)
IS
BEGIN
select product#, p_name, p_color, p_weight, supplier_id
into p_id, p_name, p_color, p_weight, p_supp
from product
where product# = p_product;
END;
/
CREATE OR REPLACE PROCEDURE p_supplier
(p_supplier IN supplier.supplier_id%type,
p_sname OUT supplier.supp_name%type,
p_scity OUT supplier.supp_city%type,
p_sstate OUT supplier.supp_state%type)
IS
```

```
BEGIN
select supp_name, supp_city, supp_state
into p_sname, p_scity, p_sstate
from supplier
where supplier_id = p_supplier;
END;
/
SET SERVEROUTPUT ON
DECLARE
pid_fk EXCEPTION;
PRAGMA Exception_Init(pid_fk,+0100);
pid product.product#%type;
p_id product.product#%type;
p_name product.p_name%type;
p_color product.p_color%type;
p_weight product.p_weight%type;
p_supp product.supplier_id%type;
p_sname supplier.supp_name%type;
p_scity supplier.supp_city%type;
p_sstate supplier.supp_state%type;
bl number;
tot number;
tot2 number;
tot3 number;
dif number;
BEGIN
pid:='&product_id';
p_product(pid, p_id,p_name,p_color,p_weight,p_supp);
```

```
p_supplier(p_supp,p_sname, p_scity, p_sstate);
DBMS_OUTPUT.PUT_LINE('Product ID; '||p_id);
DBMS_OUTPUT.PUT_LINE('Name: '||p_name);
DBMS_OUTPUT.PUT_LINE('Color: '||p_color);
DBMS_OUTPUT.PUT_LINE('Weight: '||p_weight);
DBMS_OUTPUT.PUT_LINE('Supplier ID: '||p_supp);
DBMS_OUTPUT.PUT_LINE('Supplier Name: '||p_sname);
DBMS_OUTPUT.PUT_LINE('Supplier City: '||p_scity);
DBMS_OUTPUT.PUT_LINE('Supplier State: '||p_sstate);
END;


--6 big block
--using multiple cursors to go through database to categorize employees by department
--also lists their healthcare provider
--one cursor with no data to show what happens when no employee there
SET SERVEROUTPUT ON;
DECLARE
CURSOR cur1 IS --cursor for department 99
select
e.emp_fname, e.emp_lname, b.healthcare, dept_name
from
employee e, department, benefits b
where
e.emp_dept_num = dept_id
and
e.policy# = b.policy#
and
```

```
e.emp_dept_num = '99';

CURSOR cur2 IS --cursor for department 98
select
e.emp_fname, e.emp_lname, b.healthcare, dept_name
from
employee e, department, benefits b
where
e.emp_dept_num = dept_id
and
e.policy# = b.policy#
and
e.emp_dept_num = '98';

CURSOR cur3 IS --cursor for department 91
select
e.emp_fname, e.emp_lname, b.healthcare, dept_name
from
employee e, department, benefits b
where
e.emp_dept_num = dept_id
and
e.policy# = b.policy#
and
e.emp_dept_num = '91';

CURSOR cur4 IS --cursor for department 90
select
```

```
e.emp_fname, e.emp_lname, b.healthcare, dept_name

from

employee e, department, benefits b

where

e.emp_dept_num = dept_id

and

e.policy# = b.policy#

and

e.emp_dept_num = '90';


CURSOR cur5 IS --cursor for department 94

select

e.emp_fname, e.emp_lname, b.healthcare, dept_name

from

employee e, department, benefits b

where

e.emp_dept_num = dept_id

and

e.policy# = b.policy#

and

e.emp_dept_num = '94';


TYPE type_emp_dept IS RECORD(

fn  employee.emp_fname%type,

ln  employee.emp_lname%type,

bn  benefits.healthcare%type,

dp  department.dept_name%type

);
```

```
rec_emp_dept    type_emp_dept;


BEGIN

OPEN cur1;

rec_emp_dept.dp:='99';

DBMS_OUTPUT.PUT_LINE('Management ');

DBMS_OUTPUT.PUT_LINE('');

DBMS_OUTPUT.PUT_LINE('');


   LOOP

   FETCH cur1 INTO rec_emp_dept;

   EXIT WHEN cur1%notfound;

   DBMS_OUTPUT.PUT_LINE('First Name: '||rec_emp_dept.fn);

   DBMS_OUTPUT.PUT_LINE('Last Name: '||rec_emp_dept.ln);

   DBMS_OUTPUT.PUT_LINE('Healthcare: '||rec_emp_dept.bn);

   DBMS_OUTPUT.PUT_LINE('');

   END LOOP;

DBMS_OUTPUT.PUT_LINE('===============================================================================
=========================================');

   DBMS_OUTPUT.PUT_LINE('');

   DBMS_OUTPUT.PUT_LINE('');

CLOSE cur1;

OPEN cur2;

rec_emp_dept.dp:='98';

DBMS_OUTPUT.PUT_LINE('Administration ');

DBMS_OUTPUT.PUT_LINE('');

DBMS_OUTPUT.PUT_LINE('');
```

```
   LOOP
   FETCH cur2 INTO rec_emp_dept;
   EXIT WHEN cur2%notfound;
   DBMS_OUTPUT.PUT_LINE('First Name: '||rec_emp_dept.fn);
   DBMS_OUTPUT.PUT_LINE('Last Name: '||rec_emp_dept.ln);
   DBMS_OUTPUT.PUT_LINE('Healthcare: '||rec_emp_dept.bn);
   DBMS_OUTPUT.PUT_LINE('');
   END LOOP;

DBMS_OUTPUT.PUT_LINE('=======================================================================
=====================================');
   DBMS_OUTPUT.PUT_LINE('');
   DBMS_OUTPUT.PUT_LINE('');
CLOSE cur2;
OPEN cur3;
rec_emp_dept.dp:='91';
DBMS_OUTPUT.PUT_LINE('Sales ');
DBMS_OUTPUT.PUT_LINE('');
DBMS_OUTPUT.PUT_LINE('');

   LOOP
   FETCH cur3 INTO rec_emp_dept;
   EXIT when cur3%notfound;
   DBMS_OUTPUT.PUT_LINE('First Name: '||rec_emp_dept.fn);
   DBMS_OUTPUT.PUT_LINE('Last Name: '||rec_emp_dept.ln);
   DBMS_OUTPUT.PUT_LINE('Healthcare: '||rec_emp_dept.bn);
   DBMS_OUTPUT.PUT_LINE('');
   END LOOP;
```

```
DBMS_OUTPUT.PUT_LINE('===============================================================
=================================');
   DBMS_OUTPUT.PUT_LINE('');
   DBMS_OUTPUT.PUT_LINE('');
CLOSE cur3;
OPEN cur4;
rec_emp_dept.dp:='90';
DBMS_OUTPUT.PUT_LINE('Information Technology ');
DBMS_OUTPUT.PUT_LINE('');
DBMS_OUTPUT.PUT_LINE('');


   LOOP
   FETCH cur4 INTO rec_emp_dept;
   EXIT WHEN cur4%notfound;
   DBMS_OUTPUT.PUT_LINE('First Name: '||rec_emp_dept.fn);
   DBMS_OUTPUT.PUT_LINE('Last Name: '||rec_emp_dept.ln);
   DBMS_OUTPUT.PUT_LINE('Healthcare: '||rec_emp_dept.bn);
   DBMS_OUTPUT.PUT_LINE('');
   END LOOP;

DBMS_OUTPUT.PUT_LINE('===============================================================
=================================');
   DBMS_OUTPUT.PUT_LINE('');
   DBMS_OUTPUT.PUT_LINE('');
CLOSE cur4;
OPEN cur5;
rec_emp_dept.dp:='94';
DBMS_OUTPUT.PUT_LINE('Human Resources ');
DBMS_OUTPUT.PUT_LINE('');
DBMS_OUTPUT.PUT_LINE('');


   LOOP

   FETCH cur5 INTO rec_emp_dept;

   EXIT WHEN cur5%notfound;

   DBMS_OUTPUT.PUT_LINE('First Name: '||rec_emp_dept.fn);

   DBMS_OUTPUT.PUT_LINE('Last Name: '||rec_emp_dept.ln);

   DBMS_OUTPUT.PUT_LINE('Healthcare: '||rec_emp_dept.bn);

   DBMS_OUTPUT.PUT_LINE('');

   END LOOP;

DBMS_OUTPUT.PUT_LINE('========================================================================
======================================');

   DBMS_OUTPUT.PUT_LINE('');

   DBMS_OUTPUT.PUT_LINE('');

CLOSE cur5;

END;




--7 counts the number of employees in a city

--displays employee name and number in that city


CREATE OR REPLACE PACKAGE pg_count

IS

PROCEDURE p_count

(p_count IN employee.emp_city%type,

p_p OUT number);

END pg_count;

/
```

```
CREATE OR REPLACE PACKAGE body pg_count AS

PROCEDURE p_count

(p_count IN employee.emp_city%type,

p_p OUT number)

IS


BEGIN

select count(emp_city)

into p_p

from employee

where emp_city = p_count;

END p_count;

END; -- package end

/


SET SERVEROUTPUT ON

DECLARE

   CURSOR j IS -- cursor for jackson

   select emp_fname, emp_lname

   from employee

   where emp_city ='Jackson';


   CURSOR o IS -- cursor for oxford

   select emp_fname, emp_lname

   from employee

   where emp_city ='Oxford';


   CURSOR t IS -- cursor for tupelo
   select emp_fname, emp_lname
   from employee
   where emp_city ='Tupelo';

   CURSOR b IS -- cursor for boca
   select emp_fname, emp_lname
   from employee
   where emp_city ='Boca Raton';

   TYPE type_emp_city IS RECORD(
   firstname   employee.emp_fname%type,
   lastname    employee.emp_lname%type);
   rec_emp_city    type_emp_city;
   ct number;

BEGIN
OPEN j;
   DBMS_OUTPUT.PUT_LINE('List of employees located in Jackson ');
   LOOP
   FETCH j into rec_emp_city;
   EXIT WHEN j%notfound;
   DBMS_OUTPUT.PUT_LINE('Employee Name: '||trim(rec_emp_city.firstname)||'
'||trim(rec_emp_city.lastname));
   END LOOP;
   pg_count.p_count('Jackson',ct);
   DBMS_OUTPUT.PUT_LINE('');
   DBMS_OUTPUT.PUT_LINE('Total employees in Jackson: '||ct);
   ct:=0;
   CLOSE j;
   DBMS_OUTPUT.PUT_LINE('');
   DBMS_OUTPUT.PUT_LINE('');

OPEN o;
   DBMS_OUTPUT.PUT_LINE('List of employees located in Oxford ');
   LOOP
   FETCH o into rec_emp_city;
   EXIT WHEN o%notfound;
   DBMS_OUTPUT.PUT_LINE('Employee Name: '||trim(rec_emp_city.firstname)||' '||trim(rec_emp_city.lastname));
   END LOOP;
   pg_count.p_count('Oxford',ct);
   DBMS_OUTPUT.PUT_LINE('');
   DBMS_OUTPUT.PUT_LINE('Total employees in Oxford: '||ct);
   ct:=0;
   CLOSE o;
   DBMS_OUTPUT.PUT_LINE('');
   DBMS_OUTPUT.PUT_LINE('');

OPEN t;
   DBMS_OUTPUT.PUT_LINE('List of employees located in Tupelo ');
   LOOP
   FETCH t into rec_emp_city;
   EXIT WHEN t%notfound;
   DBMS_OUTPUT.PUT_LINE('Employee Name: '||trim(rec_emp_city.firstname)||' '||trim(rec_emp_city.lastname));
   END LOOP;
   pg_count.p_count('Tupelo',ct);
   DBMS_OUTPUT.PUT_LINE('');
   DBMS_OUTPUT.PUT_LINE('Total employees in Tupelo: '||ct);

   ct:=0;

   CLOSE t;

   DBMS_OUTPUT.PUT_LINE('');

   DBMS_OUTPUT.PUT_LINE('');


OPEN b;

   DBMS_OUTPUT.PUT_LINE('List of employees located in Boca Raton ');

   LOOP

   FETCH b into rec_emp_city;

   EXIT WHEN b%notfound;

   DBMS_OUTPUT.PUT_LINE('Employee Name: '||trim(rec_emp_city.firstname)||'
'||trim(rec_emp_city.lastname));

   END LOOP;

   pg_count.p_count('Boca Raton',ct);

   DBMS_OUTPUT.PUT_LINE('');

   DBMS_OUTPUT.PUT_LINE('Total employees in Boca Raton: '||ct);

   ct:=0;

   CLOSE b;

   DBMS_OUTPUT.PUT_LINE('');

   DBMS_OUTPUT.PUT_LINE('');

END;


--8 Changes an employees city, state, and deptnum and updates their info in the employee table

--error if incorrect emp_id entered

--enter any valid employee id (1234,2345,3456, etc.)

--drop table employee3;

CREATE TABLE employee3 as select * from employee; --create employee table to edit

/
```

```
SET SERVEROUTPUT ON;
DECLARE
new_city_pk EXCEPTION;
   PRAGMA EXCEPTION_INIT(new_city_pk, -02291);
empid   employee3.emp_id%type;
efn     employee3.emp_fname%type;
eln     employee3.emp_lname%type;
sal     employee3.salary%type;
cit     employee3.emp_city%type;
estate  employee3.emp_state%type;
edeptn  employee3.emp_dept_num%type;
pol     employee3.policy#%type;
prevcity    employee3.emp_city%type;
prevstate   employee3.emp_state%type;
prevdept    employee3.emp_dept_num%type;

BEGIN
empid:='&Employee_ID';

insert into department
values ('13','New York Office','North','4321','10-30-2000');

select e.emp_id, e.emp_fname, e.emp_lname, e.salary, e.emp_city, e.emp_state, e.emp_dept_num, e.policy#
into empid, efn, eln, sal, prevcity, prevstate, prevdept, pol
from employee3 e
where e.emp_id = empid;

update employee3
```

```
set emp_city = 'New York City', emp_state = 'NY', emp_dept_num = '13'
where emp_id = empid;

select e.emp_id, e.emp_lname, e.emp_fname, e.emp_city, e.emp_state, e.emp_dept_num
into empid, eln, efn, cit, estate, edeptn
from employee3 e
where e.emp_id = empid;

DBMS_OUTPUT.PUT_LINE('');
DBMS_OUTPUT.PUT_LINE('');
DBMS_OUTPUT.PUT_LINE('');
DBMS_OUTPUT.PUT_LINE('');
DBMS_OUTPUT.PUT_LINE('Last Name: '||trim(eln));
DBMS_OUTPUT.PUT_LINE('First Name: '||trim(efn));
DBMS_OUTPUT.PUT_LINE('Previous City: '||prevcity);
DBMS_OUTPUT.PUT_LINE('Previous State: '||prevstate);
DBMS_OUTPUT.PUT_LINE('Salary: '||sal);
DBMS_OUTPUT.PUT_LINE('Previous Department Number: '||prevdept);
DBMS_OUTPUT.PUT_LINE('Policy #: '||pol);
DBMS_OUTPUT.PUT_LINE('');
DBMS_OUTPUT.PUT_LINE('New City: '||cit);
DBMS_OUTPUT.PUT_LINE('New State: '||estate);
DBMS_OUTPUT.PUT_LINE('New Department Number: '||edeptn);
DBMS_OUTPUT.PUT_LINE('');
DBMS_OUTPUT.PUT_LINE('');
DBMS_OUTPUT.PUT_LINE('');

EXCEPTION
```

```
   WHEN new_city_pk THEN
   DBMS_OUTPUT.PUT_LINE('That Employee ID was not found.');
   DBMS_OUTPUT.PUT_LINE('Please try a different Employee ID.');
END;


--9 Uses a trigger to count the number of employees each time one is deleted
--uses a cursor to go through the table to fetch all employee names

--drop table employee4;
--drop table test_emp4;

create table employee4 as select * from employee;
/
create table test_emp4 --create table for trigger
(user_name varchar2(20),
date_time timestamp(6),
num_records number);
/
CREATE OR REPLACE TRIGGER test_deemp4 --delete trigger
after delete on employee4
declare
num_records Number:=0;
BEGIN
   delete from test_emp4;
   select count(*) into num_records from employee4;
   insert into test_emp4 values(user, sysdate, num_records);
END;
```

```
/
SET SERVEROUTPUT ON
DECLARE
CURSOR cur IS
select emp_fname, emp_lname
from employee4;
validemp_fk EXCEPTION;
PRAGMA Exception_Init(validemp_fk,+0100);
TYPE emp4 IS RECORD(
   fname   employee4.emp_fname%type,
   lname   employee4.emp_lname%type
   );
rec emp4;
eid employee4.emp_id%type;
nr number;
pr number;
BEGIN
eid:='&enter_employeeid';
select count(emp_id)
into nr
from employee4;
open cur;
loop
FETCH cur INTO rec;
EXIT WHEN cur%notfound;
DBMS_OUTPUT.PUT_LINE('Employee Name: '||trim(rec.fname)||' '||trim(rec.lname));
END LOOP;
DBMS_OUTPUT.PUT_LINE('The current total number of employees is: '||nr);
```

```
CLOSE cur;
DBMS_OUTPUT.PUT_LINE('');
nr:=0;
DBMS_OUTPUT.PUT_LINE('If the employee id is in the employee4 table it will show more info. If not there is an exception.');
DBMS_OUTPUT.PUT_LINE('');


select count(emp_id)
into pr
from employee4
where emp_id=eid;
if(pr=1)then
DBMS_OUTPUT.PUT_LINE('Now the employee will be deleted.');
delete from employee4
where emp_id = eid;
select num_records
into nr
from test_emp4
where date_time=sysdate;
OPEN cur;
   LOOP
   FETCH cur INTO rec;
   EXIT WHEN cur%notfound;
   DBMS_OUTPUT.PUT_LINE('Employee Name: '||trim(rec.fname)||' '||trim(rec.lname));
   END LOOP;
   DBMS_OUTPUT.PUT_LINE('The current number of employees is: '||nr);
CLOSE cur;
ELSE
RAISE validemp_fk;
```

```
END IF;
DBMS_OUTPUT.PUT_LINE('');
DBMS_OUTPUT.PUT_LINE('');
DBMS_OUTPUT.PUT_LINE('');
DBMS_OUTPUT.PUT_LINE('The employee id is not in the employee4 table. Please try again.');
END;
```

Output:

(Input 1234 to test; enter 1234 again to test exception)

```
Employee Name: Aristides Aquino
Employee Name: Billy Hamilton
Employee Name: Curtis Granderson
Employee Name: Dee Gordon
Employee Name: Eugenio Suarez
Employee Name: Freddie Freeman
Employee Name: Giancarlo Stanton
Employee Name: Trea Turner
Employee Name: Byron Buxton
Employee Name: Joey Freshwater
The current total number of employees is: 10
```

If the employee id is in the employee4 table it will show more info. If not there is an exception.

Now the employee will be deleted.

```
Employee Name: Billy Hamilton
Employee Name: Curtis Granderson
Employee Name: Dee Gordon
Employee Name: Eugenio Suarez
Employee Name: Freddie Freeman

Employee Name: Giancarlo Stanton

Employee Name: Trea Turner

Employee Name: Byron Buxton

Employee Name: Joey Freshwater

The current number of employees is: 9


The employee id is not in the employee4 table. Please try again.
```