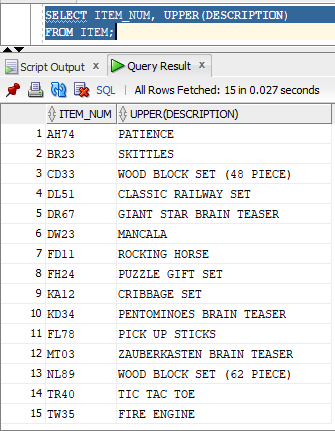
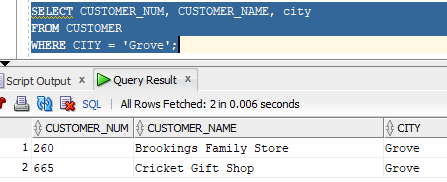
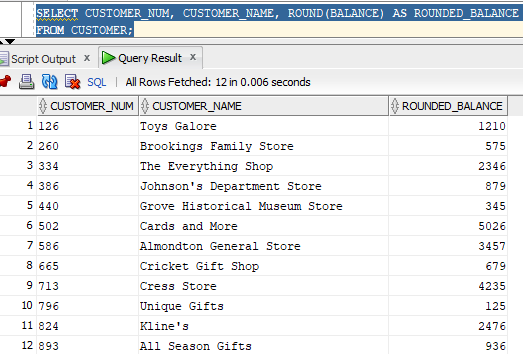
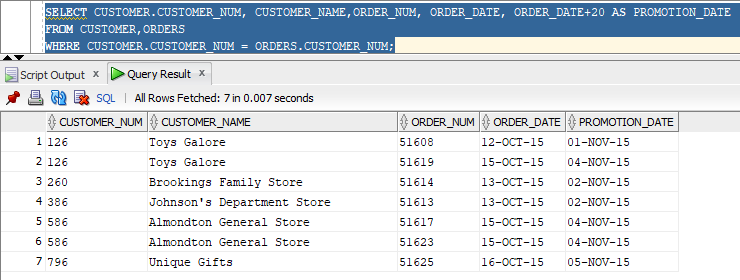
END OF CHAPTER ASSIGNMENT 8









A)CREATE OR REPLACE PROCEDURE DISP\_CUSTOMER\_NAME(I\_CUSTOMER\_NUM IN CUSTOMER.CUSTOMER\_NUM%TYPE) AS

I\_CREDIT\_LIMIT customer.credit\_limit%TYPE;

I\_CUSTOMER\_NUM CUSTOMER.CUSTOMER\_NUM%TYPE;

BEGIN

SELECT CUSTOMER\_NAME, CREDIT\_LIMIT

INTO I\_customer\_name, I\_CREDIT\_LIMIT

FROM CUSTOMER

WHERE CUSTOMER\_NUM = I\_CUSTOMER\_NUM;

DBMS\_OUTPUT.PUT\_LINE(I\_CUSTOMER\_NUM);

DBMS\_OUTPUT.PUT\_LINE(I\_CREDIT\_LIMIT);

END;/

b) CREATE OR REPLACE PROCEDURE DISP\_CUST\_ORDERS (I\_ORDER\_NUM IN ORDERS.ORDER\_NUM%TYPE) AS

I\_ORDER\_NUM ORDERS.ORDER\_NUM%TYPE;

I\_ORDER\_DATE ORDERS.ORDER\_NUM%TYPE;

I\_CUSTOMER\_NUM CUSTOMER.CUSTOMER\_NUM%TYPE;

I\_CUSTOMER\_NAME CUSTOMER.CUSTOMER\_NAME%TYPE;

BEGIN

SELECT ORDERS.ORDER\_NUM, ORDER\_DATE, CUSTOMER.CUSTOMER\_NUM, CUSTOMER\_NAME

FROM ORDERS, CUSTOMER

WHERE ORDERS.CUSTOMER\_NUM = CUSTOMER.CUSTOMER\_NUM;

DBMS\_OUTPUT.PUT\_LINE(I\_ORDER\_NUM);

DBMS\_OUTPUT.PUT\_LINE(I\_ORDER\_DATE);

DBMS\_OUTPUT.PUT\_LINE(I\_CUSTOMER\_NUM);

DBMS\_OUTPUT.PUT\_LINE(I\_CUSTOMER\_DATE);

END;

/

c)

d) CREATE OR REPLACE PROCEDURE DISP\_CUST\_ORDERS(I\_ORDER\_NUM IN ORDERS.ORDER\_NUM%TYPE,

I\_ORDER\_DATE IN ORDERS.ORDER\_DATE%TYPE) AS

BEGIN

UPDATE ORDERS

SET ORDER\_DATE = I\_ORDER\_DATE

WHERE ORDER\_NUM = I\_ORDER\_NUM;

END;

/

e) CREATE OR REPLACE PROCEDURE DISP\_CUST\_ORDERS(I\_ORDER\_NUM IN ORDERS.ORDER\_NUM%TYPE) AS

BEGIN

DELETE

FROM ORDER\_LINE

WHERE ORDER\_NUM = I\_ORDER\_NUM;

BEGIN

DELETE

FROM ORDERS

WHERE ORDER\_NUM = I\_ORDER\_NUM;

END;

/

6) CREATE OR REPLACE PROCEDURE DISP\_ITEM\_ORDS (I\_CATEGORY IN ITEM.ITEM\_NUM%TYPE) AS

I\_ITEM\_NUM ITEM.ITEM\_NUM%TYPE;

I\_DESCRIPTION ITEM.ITEM\_NUM%TYPE;

I\_STOREHOUSE\_NUM ITEM.ITEM\_NUM%TYPE;

I\_PRICE ITEM.ITEM\_NUM%TYPE;

CURSOR ORDSGROUP IS

SELECT ITEM.ITEM\_NUM, DESCRIPTION, STOREHOUSE\_NUM, PRICE

FROM ITEM

WHERE ITEM\_NUM = I\_ITEM\_NUM;

BEGIN

OPEN ORDSGROUP;

LOOP

FETCH ORDSGROUP INTO I\_ITEM\_NUM,I\_DESCRIPTION,I\_STOREHOUSE\_NUM,I\_PRICE;

EXIT WHEN ORDSGROUP%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE(I\_ITEM\_NUM);

DBMS\_OUTPUT.PUT\_LINE(I\_DESCRIPTION);

DBMS\_OUTPUT.PUT\_LINE(I\_STOREHOUSE\_NUM);

DBMS\_OUTPUT.PUT\_LINE(I\_PRICE);

END LOOP;

CLOSE ORDSGROUP;

END;

/

7)a) strSQL = “DELETE FROM ORDERS WHERE ORDER\_NUM = ‘”

StrSQL = strSQL & I\_ORDER\_NUM

StrSQL = strSQL & “’”

DoCmd.RUNSQL strSQL

END FUNCTION (?)

b)Public Function RepUpdate(I\_ORDER\_DATE, I\_ORDER\_NUM)

Dim StrSQL As String

StrSQL=”UPDATE ORDERS SET ORDER\_DATE = ‘”

StrSQL= strSQL & I\_ORDER\_DATE

StrSQL= strSQL & ‘ WHERE I\_ORDER\_NUM = ‘”

StrSQL= strSQL & ORDER\_NUM

StrSQL= strSQL & “’”

DoCmd.RunSQL strSQL

End Function(?)

8) CREATE OR REPLACE PROCEDURE ITEM\_PRICE (I\_ITEM\_NUM IN ITEM.ITEM\_NUM%TYPE,

I\_PRICE IN ITEM.PRICE%TYPE) AS

BEGIN

UPDATE ITEM

SET PRICE = I\_PRICE

WHERE CATEGORY = I\_CATEGORY

END;

/

BEGIN

ITEM\_PRICE(26.95, ‘AH74’)

END;

/

9) CREATE OR REPLACE TRIGGER ADD\_CUSTOMER\_BALANCE

AFTER INSERT ON REP FOR EACH ROW

BEGIN

UPDATE REP

SET ON\_REP =ON\_COMMISSION \* :NEW.BALANCE

WHERE REP.REP\_NUM = CUSTOMER.REP\_NUM

10) The MONTHSBETWEEN function is used to obtain the number of months between two inputted dates. If date one is later than date two, the result equals positive. If both dates are around the same time the result is an integer. The CURRENTDATE function returns the current datetime in the user’s timezone. SYSDATE returns the current date and time of the database server with a time zone that may differ. SQL SERVER- GETDATE() ACCESS- SELECT DATE.

a) SELECT ORDER\_NUM, SYSDATE AS TODAYS DATE, ORDER\_DATE, ROUND(SYSDATE – ORDER\_DATE, 2) AS DAYS PAST

FROM ORDERS;

b) In access, Date() is used when dates are required. Now() is used for datetime values.