**ASSIGNMENT 6**

CREATE TABLE VENDORS6 AS (SELECT \* FROM VENDORS);

CREATE TABLE INVOICES6 AS (SELECT \* FROM INVOICES);

ALTER TABLE VENDORS6 ADD CONSTRAINT VENDORS6\_PK PRIMARY KEY (VENDOR\_ID);

ALTER TABLE VENDORS6 ADD CONSTRAINT VENDORS6\_VENDOR\_NAME\_UQ UNIQUE (VENDOR\_NAME);

ALTER TABLE VENDORS6 ADD CONSTRAINT VENDORS6\_FK\_TERMS FOREIGN KEY (DEFAULT\_TERMS\_ID) REFERENCES TERMS (TERMS\_ID);

ALTER TABLE VENDORS6 ADD CONSTRAINT VENDORS6\_FK\_ACCOUNTS FOREIGN KEY (DEFAULT\_ACCOUNT\_NUMBER) REFERENCES GENERAL\_LEDGER\_ACCOUNTS (ACCOUNT\_NUMBER);

ALTER TABLE INVOICES6 ADD CONSTRAINT INVOICES6\_PK PRIMARY KEY (INVOICE\_ID);

ALTER TABLE INVOICES6 ADD CONSTRAINT INVOICES6\_FK\_VENDORS FOREIGN KEY (VENDOR\_ID) REFERENCES VENDORS6 (VENDOR\_ID);

ALTER TABLE INVOICES6 ADD CONSTRAINT INVOICES6\_FK\_TERMS FOREIGN KEY (TERMS\_ID) REFERENCES TERMS (TERMS\_ID);

ALTER TABLE INVOICES6 ADD BALANCE\_DUE NUMBER (9, 2);

DELETE FROM VENDORS6 WHERE VENDOR\_ID = 101;

BEGIN

EXECUTE IMMEDIATE

'CREATE OR REPLACE FUNCTION EXEC\_BALANCE\_DUE '

||'('

||' NUM1 IN NUMBER '

||', NUM2 IN NUMBER '

||', NUM3 IN NUMBER '

||') RETURN VARCHAR2 AS '

||'BEGIN'

||' RETURN NUM1- NUM2 - NUM3 ;'

||'END EXEC\_BALANCE\_DUE;'

;

END;

/

BEGIN

EXECUTE IMMEDIATE Q'[CREATE OR REPLACE TRIGGER TRIGGER\_CAL

BEFORE INSERT OR UPDATE ON INVOICES6

FOR EACH ROW

BEGIN

:NEW.BALANCE\_DUE := EXEC\_BALANCE\_DUE(:NEW.INVOICE\_TOTAL, :NEW.PAYMENT\_TOTAL, :NEW.CREDIT\_TOTAL); END;

]'; END;

DECLARE

nextVENDOR VENDORS6.VENDOR\_ID%TYPE;

nextINVOICE invoices6.INVOICE\_ID%TYPE;

BEGIN

SELECT MAX(VENDOR\_ID)+1 INTO nextVENDOR FROM VENDORS6;

SELECT MAX(INVOICE\_ID)+1 INTO nextINVOICE FROM INVOICES6;

INSERT INTO VENDORS6

VALUES (nextVENDOR,'Bhanumathi, LLC','Jackson','Apt#203','Chicago','IL','60616','(312) 589-8746','LLL','FFF',5,400);

INSERT INTO INVOICES6(INVOICE\_ID,VENDOR\_ID,INVOICE\_NUMBER,INVOICE\_DATE,INVOICE\_TOTAL,PAYMENT\_TOTAL,CREDIT\_TOTAL,TERMS\_ID,INVOICE\_DUE\_DATE,PAYMENT\_DATE,BALANCE\_DUE)

VALUES(nextINVOICE,nextVENDOR,'A1000',SYSDATE, 1000.10, 0, 0, 5, SYSDATE+30, NULL, 1000.10);

INSERT INTO INVOICES6 (INVOICE\_ID, VENDOR\_ID, INVOICE\_NUMBER, INVOICE\_DATE, INVOICE\_TOTAL, PAYMENT\_TOTAL, CREDIT\_TOTAL, TERMS\_ID,INVOICE\_DUE\_DATE, PAYMENT\_DATE, BALANCE\_DUE)

VALUES(nextINVOICE,nextVENDOR,'B1000',SYSDATE, 2000.10, 0, 0, 5, SYSDATE+30, NULL, 2000.10);

INSERT INTO INVOICES6 (INVOICE\_ID, VENDOR\_ID, INVOICE\_NUMBER, INVOICE\_DATE, INVOICE\_TOTAL, PAYMENT\_TOTAL, CREDIT\_TOTAL, TERMS\_ID,INVOICE\_DUE\_DATE, PAYMENT\_DATE, BALANCE\_DUE)

VALUES(nextINVOICE,nextVENDOR,'C1000',SYSDATE, 5000.10, 0, 0, 5, SYSDATE+30, NULL, 5000.10);

UPDATE INVOICES6

SET CREDIT\_TOTAL = ROUND(INVOICE\_TOTAL \* 0.15, 2)

WHERE VENDOR\_ID = nextVENDOR;

UPDATE INVOICES6

SET BALANCE\_DUE = NULL

WHERE VENDOR\_ID = nextVENDOR;

END;

/

SET LINESIZE 999

COLUMN VENDOR\_NAME FORMAT A15

COLUMN INVOICE\_TOTAL FORMAT A15

COLUMN CREDIT\_TOTAL FORMAT A15

COLUMN BALANCE\_DUE FORMAT A15

SELECT V.VENDOR\_ID,

V.VENDOR\_NAME,

I.INVOICE\_ID,

TO\_CHAR (I.INVOICE\_DATE,'MONTHDD, YYYY') AS INVOICE\_DATE,

TO\_CHAR (I.INVOICE\_DUE\_DATE, 'MONTHDD, YYYY') AS INVOICE\_DUE\_DATE,

TO\_CHAR (I.INVOICE\_TOTAL, '$999,999,999.00') AS INVOICE\_TOTAL,

TO\_CHAR (I.CREDIT\_TOTAL, '$999,999,999.00') AS CREDIT\_TOTAL,

TO\_CHAR (I.BALANCE\_DUE, '$999,999,999.00') AS BALANCE\_DUE

FROM VENDORS6 V

JOIN INVOICES6 I

ON V.VENDOR\_ID = I.VENDOR\_ID

WHERE V.VENDOR\_ID = (SELECT MAX (VENDOR\_ID) FROM VENDORS6);

DROP TABLE INVOICES6;

DROP TABLE VENDORS6;

DROP FUNCTION EXEC\_BALANCE\_DUE;

DROP TRIGGER TRIGGER\_CAL;