CREATE TABLE PARENT(

SSN CHAR(5) NOT NULL,

FNAME VARCHAR(50) NOT NULL,

LNAME VARCHAR(50),

DOB DATE,

EMAIL VARCHAR(10),

CONTACTNO CHAR(10),

ADDRESS VARCHAR(30),

ZIPCODE CHAR(5),

PRIMARY KEY(SSN));

CREATE TABLE TOYS(

T\_ID INT NOT NULL,

TOY\_NAME VARCHAR(20),

DATE\_OF\_PURCHASE DATE,

COMPANY VARCHAR(10),

TOTAL\_QUANTITY INT,

PRIMARY KEY(T\_ID));

CREATE TABLE FOOD(

F\_ID INT NOT NULL,

FOOD\_NAME VARCHAR(20),

PRICE INT,

QUANTITY INT,

PRIMARY KEY(F\_ID));

CREATE TABLE JOB(

JOB\_ID INT NOT NULL,

BASIC\_SALARY INT,

PRIMARY KEY(JOB\_ID));

CREATE TABLE DEPARTMENT(

DEPT\_ID INT NOT NULL,

DEPT\_NAME VARCHAR(20),

MGR\_SSN CHAR(9),

PRIMARY KEY(DEPT\_ID));

CREATE TABLE CARETAKER(

CSSN CHAR(5) NOT NULL,

FNAME VARCHAR(50) NOT NULL,

LNAME VARCHAR(50),

DOB DATE,

EMAIL VARCHAR(10),

CONTACTNO CHAR(10),

JOBTYPE CHAR (1),

SUBJECT VARCHAR(20),

JOB\_ID INT,

MGR\_SSN CHAR(9),

DEPT\_ID INT,

PRIMARY KEY(CSSN),

FOREIGN KEY (JOB\_ID) REFERENCES JOB(JOB\_ID),

FOREIGN KEY (DEPT\_ID) REFERENCES DEPARTMENT(DEPT\_ID));

ADDING FOREIGN KEY CONSTRAINTS TO MGR\_SSN IN BOTH CARETAKER AND DEPARTMENT TABLES

ALTER TABLE CARETAKER ADD CONSTRAINT SUPERVISOR\_KEY FOREIGN KEY (MGR\_SSN) REFERENCES CARETAKER(CSSN);

ALTER TABLE DEPARTMENT ADD CONSTRAINT MGR\_KEY FOREIGN KEY (MGR\_SSN) REFERENCES CARETAKER(CSSN);

CREATE TABLE CLASSSECTION(

CLASS\_ID INT NOT NULL,

ROOM VARCHAR(10),

LOCATION VARCHAR(20),

CLASS\_TEACHER CHAR(9),

PRIMARY KEY(CLASS\_ID),

FOREIGN KEY (CLASS\_TEACHER) REFERENCES CARETAKER(CSSN));

CREATE TABLE CHILD(

CHILD\_ID INT NOT NULL,

FNAME VARCHAR(50) NOT NULL,

LNAME VARCHAR(50),

DOB DATE,

ENAME VARCHAR(10),

EPHONENO CHAR(10),

PSSN CHAR(9),

CLASS INT,

PRIMARY KEY (CHILD\_ID),

FOREIGN KEY (PSSN) REFERENCES PARENT(SSN),

FOREIGN KEY (CLASS) REFERENCES CLASSSECTION(CLASS\_ID));

CREATE TABLE TOY\_DISTRIBUTION(

TOY\_ID INT NOT NULL,

CLASS\_ID INT NOT NULL,

QUANTITY INT,

FOREIGN KEY (TOY\_ID) REFERENCES TOYS(T\_ID),

FOREIGN KEY (CLASS\_ID) REFERENCES CLASSSECTION(CLASS\_ID));

CREATE TABLE TUITION\_PAYMENT(

PAYMENT\_ID INT NOT NULL,

PAYMENT\_TYPE VARCHAR(20) NOT NULL,

PAYMENT\_DATE DATE,

AMOUNT INT,

CHILD\_ID INT,

PRIMARY KEY(PAYMENT\_ID),

FOREIGN KEY (CHILD\_ID) REFERENCES CHILD(CHILD\_ID));

CREATE TABLE TIMING(

TIME\_ID INT NOT NULL,

CSSN CHAR(9) NOT NULL,

TIMEIN TIMESTAMP,

TIMEOUT TIMESTAMP,

PRIMARY KEY(TIME\_ID),

FOREIGN KEY (CSSN) REFERENCES CARETAKER(CSSN));

CREATE TABLE FOODORDEREDBY (

STAFFSSN CHAR(9) NOT NULL,

FOOD\_ID INT NOT NULL,

DATEORDERED DATE,

FOREIGN KEY (STAFFSSN) REFERENCES CARETAKER(CSSN),

FOREIGN KEY (FOOD\_ID) REFERENCES FOOD(F\_ID));

CREATE TABLE FOODRECORD(

CHILD\_ID INT NOT NULL,

FOOD\_ID INT NOT NULL,

PRIMARY KEY(CHILD\_ID,FOOD\_ID),

FOREIGN KEY (CHILD\_ID) REFERENCES CHILD(CHILD\_ID),

FOREIGN KEY (FOOD\_ID) REFERENCES FOOD(F\_ID));

CREATE TABLE SALARYCOMPUTATION(

CARETAKERSSN CHAR(9) NOT NULL,

TIMINGID INT NOT NULL,

JOBID INT NOT NULL,

SALARY INT,

PRIMARY KEY(CARETAKERSSN,TIMINGID,JOBID),

FOREIGN KEY (CARETAKERSSN) REFERENCES CARETAKER(CSSN),

FOREIGN KEY (TIMINGID) REFERENCES TIMING(TIME\_ID),

FOREIGN KEY (JOBID) REFERENCES JOB(JOB\_ID));

CREATE OR REPLACE TRIGGER TOYS\_UPDATE

AFTER INSERT ON TOY\_DISTRIBUTION

FOR EACH ROW

BEGIN

UPDATE TOYS

SET TOTAL\_QUANTITY=TOTAL\_QUANTITY - :NEW.QUANTITY

WHERE T\_ID=:NEW.TOY\_ID;

END;

CREATE OR REPLACE PROCEDURE CLASS\_LIST

( CLASS\_ID IN CHILD.CLASS %TYPE) AS

CHILD\_ID CHILD.CHILD\_ID%TYPE;

CHILD\_FNAME CHILD.FNAME%TYPE;

CURSOR RESULT IS

(

SELECT CHILD\_ID,FNAME

FROM CHILD WHERE CLASS\_ID=CLASS

);

BEGIN

OPEN RESULT;

LOOP

FETCH RESULT INTO CHILD\_ID,CHILD\_FNAME;

EXIT WHEN (RESULT%NOTFOUND);

DBMS\_OUTPUT.PUT\_LINE

(

'CHILD\_ID IS ' || CHILD\_ID || ' CHILD NAME IS '||CHILD\_FNAME

);

END LOOP;

CLOSE RESULT;

END;

BEGIN

CLASS\_LIST(1);

END;

CREATE OR REPLACE PROCEDURE NO\_OF\_CARETAKERS

(DEPT IN NUMBER) AS

RES NUMBER:=0;

BEGIN

SELECT COUNT(CSSN) INTO RES

FROM CARETAKER

WHERE DEPT\_ID =DEPT;

DBMS\_OUTPUT.PUT\_LINE('TOTAL NUMBER OF EMPLOYEES WORKING IN THIS DEPT ARE '||RES);

END;

BEGIN

NO\_OF\_CARETAKERS(2)

END;

CREATE OR REPLACE PROCEDURE TOTAL\_FEE

(ID IN NUMBER) AS

FEE NUMBER:=0;

BEGIN

SELECT SUM(AMOUNT) INTO FEE

FROM TUITION\_PAYMENT

WHERE CHILD\_ID =ID;

DBMS\_OUTPUT.PUT\_LINE('TOTAL FEES PAID BY CHILD IS '||FEE);

END;

create or replace TRIGGER SALARY\_COMPUTE

AFTER INSERT ON TIMING

FOR EACH ROW

DECLARE

BASICSALARY INT;

SALARYCOMPUTED INT;

CSSN CHAR(9);

JID INT;

BEGIN

SELECT C.CSSN,C.JOB\_ID,J.BASIC\_SALARY INTO CSSN,JID,BASICSALARY FROM CARETAKER C JOIN JOB J ON J.JOB\_ID = C.JOB\_ID

WHERE C.CSSN = :NEW.CSSN;

SALARYCOMPUTED := (8\*BASICSALARY)/10000;

INSERT INTO SALARYCOMPUTATION

VALUES(CSSN,:NEW.TIME\_ID,JID, SALARYCOMPUTED);

END;