Database Design
CS 6360.003: Health/Fitness Program

Due on Wednesday December 7, 2016 at 11:59pm

Instructor: Nurcan Yuruk

Hanlin He / Kai Kang (hxh160630 / kxk151230)

Contents

1 Requirement

The Fitness Systems main purpose is to assist the UT Dallas Fitness Center management. Management mainly care about these aspects: customer management, fitness program management, equipment management, and sport session management. The system shall include the following function and features.

1. The system shall keep track of customers profile. Basic information shall include name, address, phone, age, sexuality, height, weight. Each customer should have a customer id for identification. Each customer should have a membership id(valid date and expiration date) and a closet.

2. The system shall provide multiple fitness programs at different difficulty levels. The fitness program shall have the following information: name, type (aerobic/strength-/HIIT), list of equipment needed. For aerobic training, additional informations about the exercise intensity and minimal exercise time shall be available. In particular, there are four types of aerobic: treadmill, elliptical, exercise bike and rowing machine.

3. The system shall keep information about employees. Employees include trainer, equipment maintainer. Each trainer is eligible for some training programs.

4. Customer need to sign a contract to exercise in the Fitness Center. In the contract, customer can choose a fitness program. The system shall record the contract No.

5. The system shall keep information about all equipments, including name, price, vendor, model and age. each equipment's price can be implied by its vendor and model to identify.

6. Each equipment shall have one maintainer responsible for maintenance.

7. The system shall track and store the exercise sessions data for all customer, include exercise type, exercise session duration, average heart rate, maximum heart rate, estimate calories burnt. For aerobic training, types, distance and average pace shall be stored.

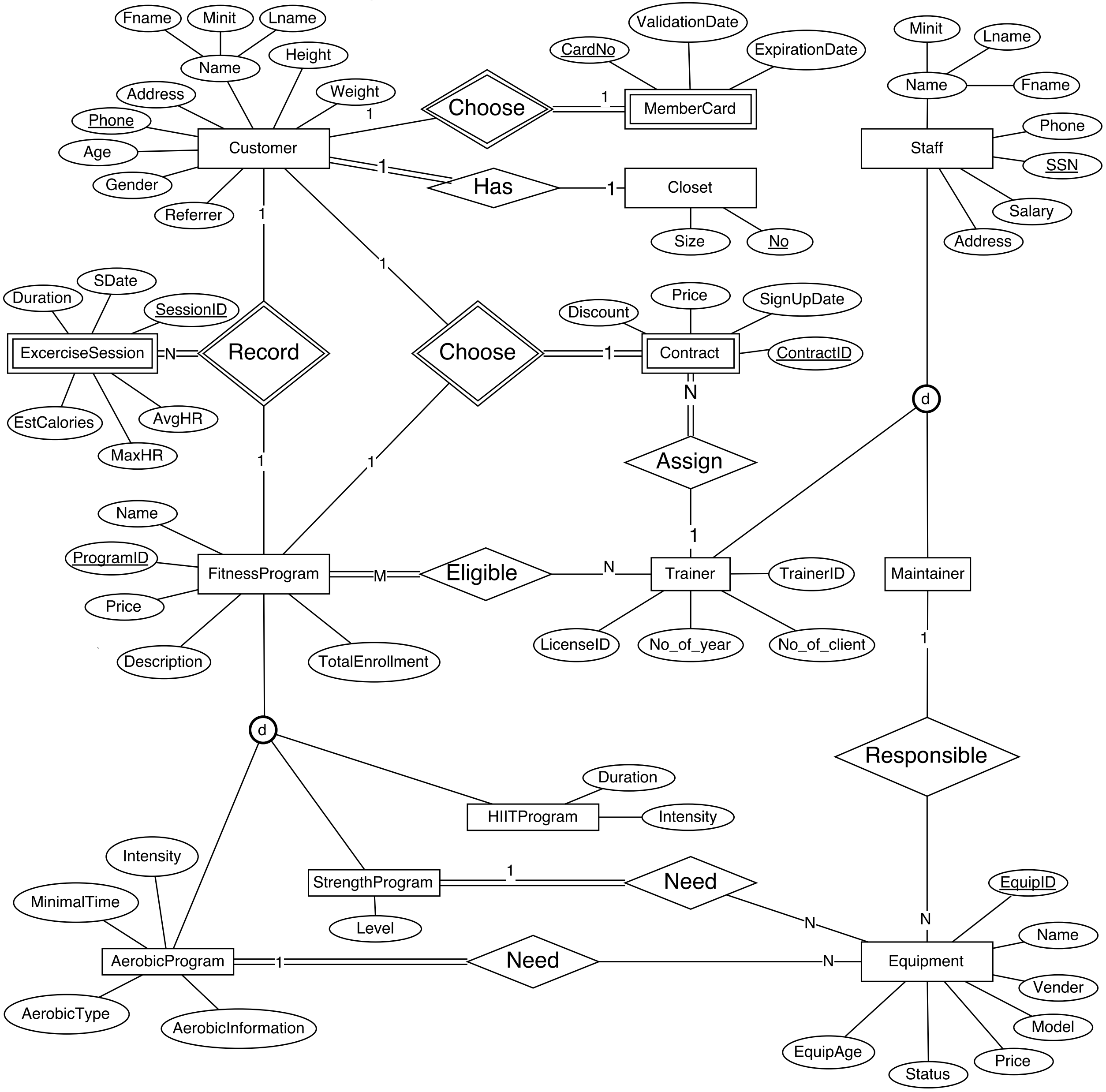
8. The system shall support occasional deal event, such as add bonus time for member registered during a specific period.

9. The system shall support clean up the exercise session data for long time idle user.

2 Enhanced Entity-Relation Diagram

The EER is shown in fig. 1.

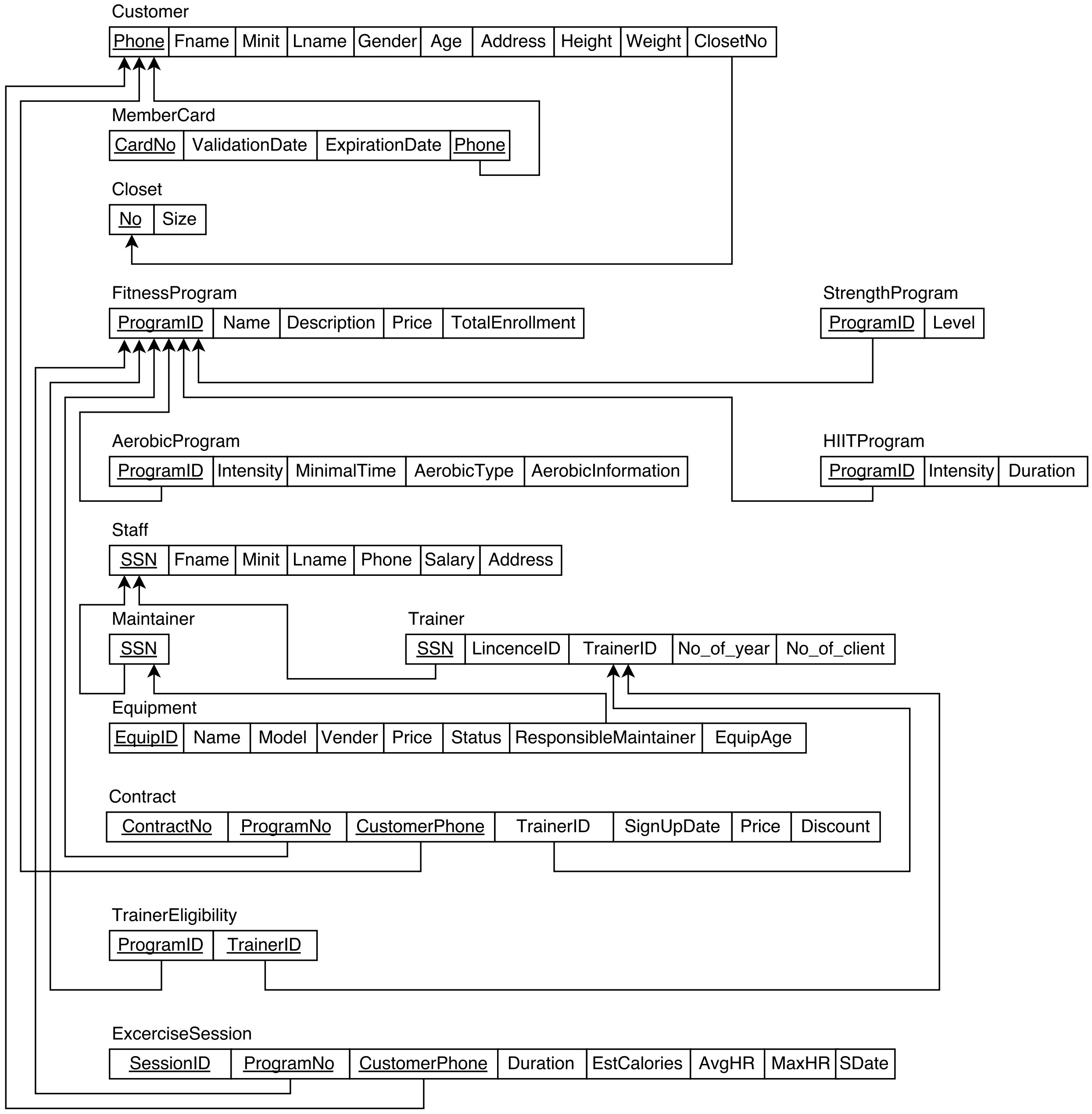
Figure 1: EER for Fitness Program



3 Mapping EER Diagram to Relational Schema and Normalization

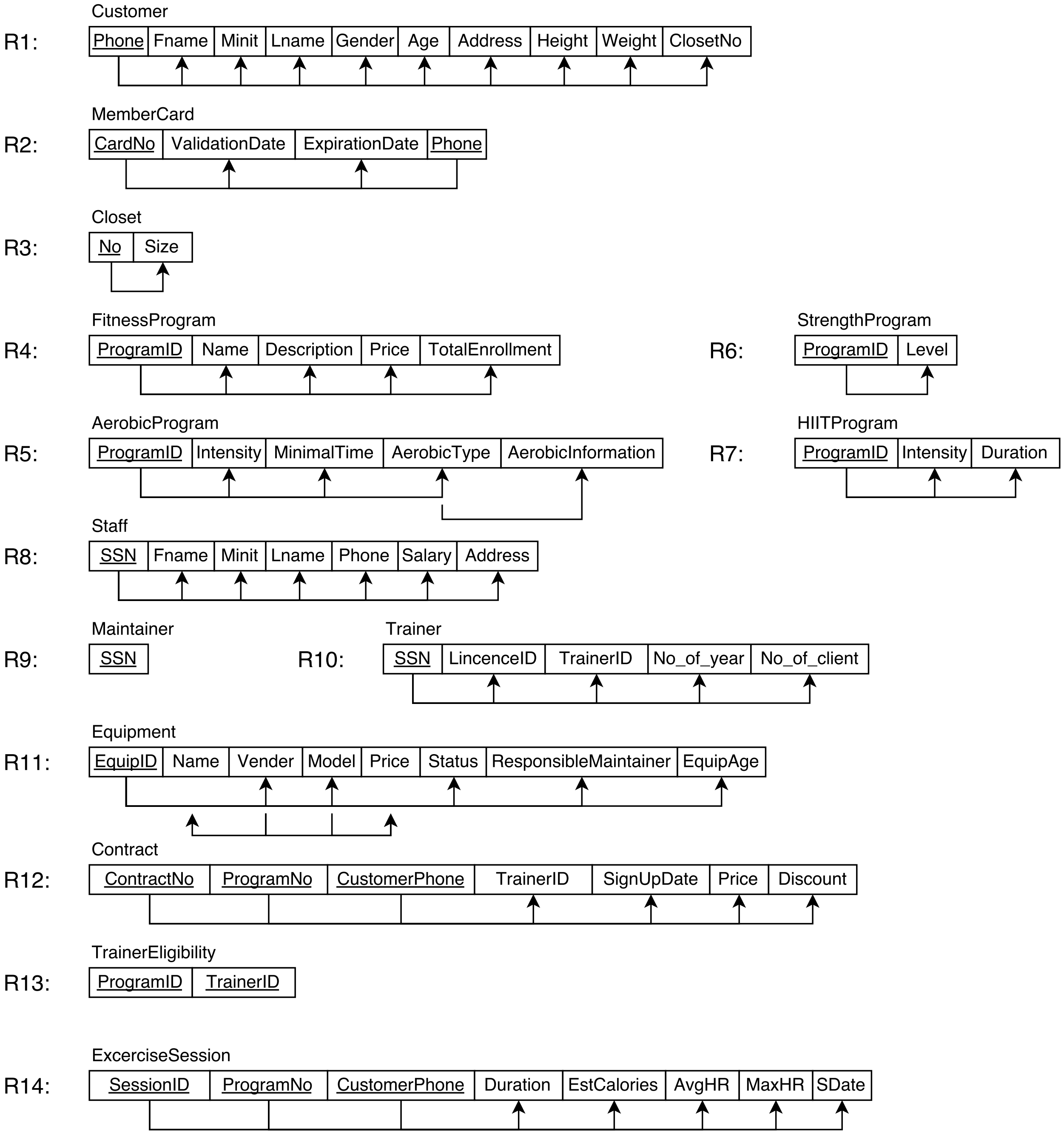
The relational schema mapped from EER is shown infig. 2.

Figure 2: Relational Schema after Mapping



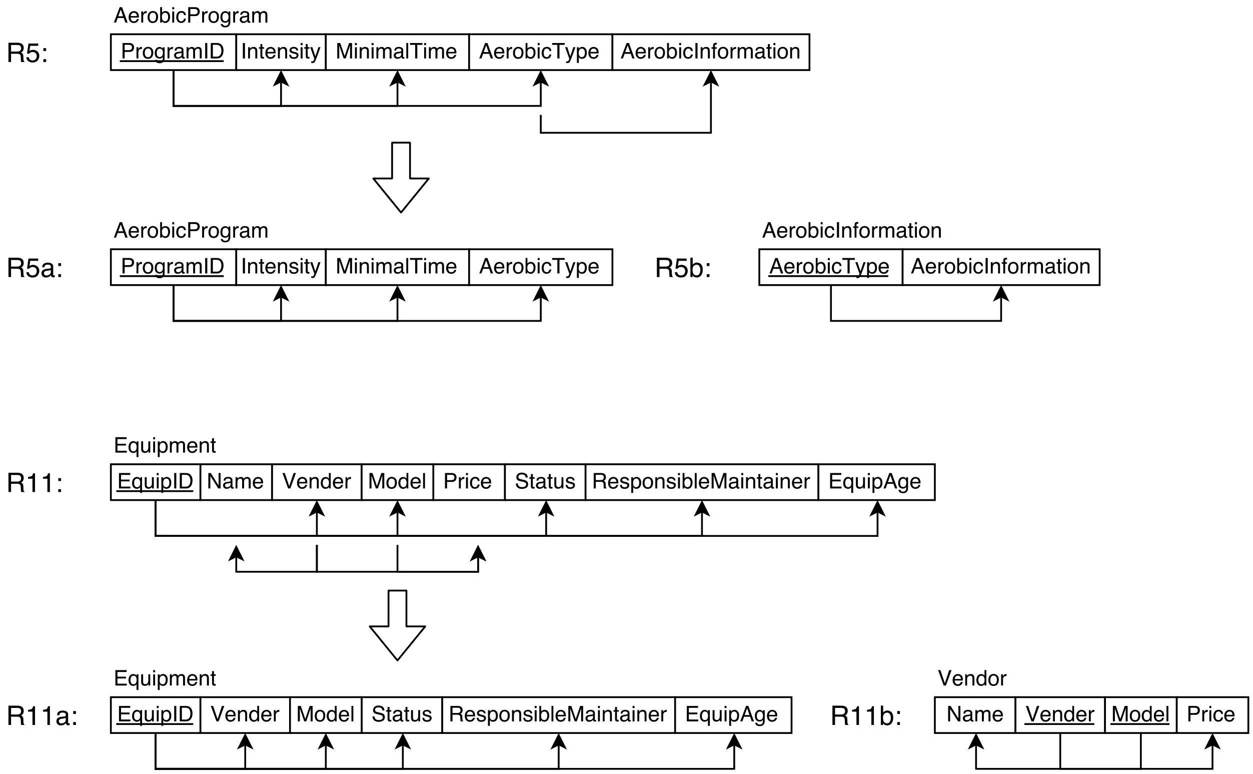
The functional dependencies in the schema is shown in fig. 3.

Figure 3: Original Functional Dependencies


R1: Customer
Phone | Fname | Minit | Lname | Gender | Age | Address | Height | Weight | ClosetNo
R2: MemberCard
CardNo | ValidationDate | ExpirationDate | Phone
R3: Closet
No | Size
R4: FitnessProgram
ProgramID | Name | Description | Price | TotalEnrollment
R6: StrengthProgram
ProgramID | Level
R5: AerobicProgram
ProgramID | Intensity | MinimalTime | AerobicType | AerobicInformation
R7: HIITProgram
ProgramID | Intensity | Duration
R8: Staff
SSN | Fname | Minit | Lname | Phone | Salary | Address
R9: Maintainer
SSN
R10: Trainer
SSN | LincenceID | TrainerID | No_of_year | No_of_client
R11: Equipment
EquipID | Name | Vender | Model | Price | Status | ResponsibleMaintainer | EquipAge
R12: Contract
ContractNo | ProgramNo | CustomerPhone | TrainerID | SignUpDate | Price | Discount
R13: TrainerEligibility
ProgramID | TrainerID
R14: ExcerciseSession
SessionID | ProgramNo | CustomerPhone | Duration | EstCalories | AvgHR | MaxHR | SDate

We can see from fig. 3 that $R5$ and $R11$ violates 3NF. To normalize these relations, we need to remove the transitive functional dependencies in the relation. The normalization process is shown in fig. 4, $R5a$, $R5b$ substitute the original $R5$, $R11a$ and $R11b$ substitute the original $R11$.

Figure 4: Normalization Process



4 SQL

The CREATE table command is as follow.

```
--Create tables.
CREATE TABLE Closet (
  No VARCHAR2(10),
  ClosetSize VARCHAR2(10) NOT NULL,
  PRIMARY KEY (No)
);
/
CREATE TABLE Customer (
  Phone VARCHAR2(10),
  Fname VARCHAR2(20) NOT NULL,
  Minit CHAR,
  Lname VARCHAR2(20) NOT NULL,
  Gender CHAR NOT NULL,
  Age NUMBER NOT NULL,
  Height FLOAT,
  Weight FLOAT,
  ClosetNo VARCHAR2(10),
  PRIMARY KEY (Phone),
  FOREIGN KEY (ClosetNo) REFERENCES CLOSET(No) ON DELETE CASCADE
);
/
CREATE TABLE MemberCard (
  CardNo VARCHAR2(10),
  Phone VARCHAR2(10),
  ValidationDate DATE NOT NULL,
  ExpirationDate DATE NOT NULL,
  PRIMARY KEY (CardNo, Phone),
  FOREIGN KEY (Phone) REFERENCES Customer(Phone) ON DELETE CASCADE
);
/
CREATE TABLE FitnessProgram (
  ProgramID NUMBER,
  Name VARCHAR2(30) NOT NULL,
  Description VARCHAR2(100) NOT NULL,
  Price FLOAT NOT NULL,
  TotalEnrollment NUMBER DEFAULT 0,
  PRIMARY KEY (ProgramID)
);
/
CREATE TABLE AerobicInformation (
  AerobicType VARCHAR2(20),
  AerobicInformation VARCHAR2(300) NOT NULL,
  PRIMARY KEY (AerobicType)
);
/
CREATE TABLE AerobicProgram (
  ProgramID   NUMBER,
  Intensity   VARCHAR2(15) NOT NULL,
  MinimalTime NUMBER NOT NULL,
  AerobicType VARCHAR2(20),
  PRIMARY KEY (ProgramID),
  FOREIGN KEY (ProgramID) REFERENCES FitnessProgram (ProgramID) ON DELETE CASCADE ,
  FOREIGN KEY (AerobicType) REFERENCES AerobicInformation(AerobicType) ON DELETE CASCADE ,
  CONSTRAINT maximumMinimalTime CHECK (MinimalTime < 90)
);
/
CREATE TABLE StrengthProgram (
  ProgramID NUMBER,
  StrengthLevel VARCHAR2(15) NOT NULL,
  PRIMARY KEY (ProgramID),
  FOREIGN KEY (ProgramID) REFERENCES FitnessProgram (ProgramID) ON DELETE CASCADE
);
/
CREATE TABLE HIITProgram (
  ProgramID NUMBER,
  Intensity VARCHAR2(15) NOT NULL,
  Duration NUMBER NOT NULL,
  PRIMARY KEY (ProgramID),
  FOREIGN KEY (ProgramID) REFERENCES FitnessProgram (ProgramID) ON DELETE CASCADE ,
  CONSTRAINT maximumDuration CHECK (Duration < 90)
);
/
CREATE TABLE Staff (
  SSN CHAR(9),
  Fname VARCHAR2(20) NOT NULL,
  Minit CHAR,
  Lname VARCHAR2(20) NOT NULL,
  Phone VARCHAR2(10) NOT NULL,
  Salary NUMBER NOT NULL,
  Address VARCHAR2(60) NOT NULL,
  PRIMARY KEY (SSN),
  CONSTRAINT minSalary CHECK (Salary > 5000)
);
/
CREATE TABLE Maintainer (
  SSN CHAR(9),
  PRIMARY KEY (SSN),
  FOREIGN KEY (SSN) REFERENCES Staff(SSN) ON DELETE CASCADE
);
/
CREATE TABLE Trainer (
  SSN CHAR(9),
  LicenseID CHAR(10),
  TrainerID NUMBER UNIQUE NOT NULL,
  No_of_years NUMBER NOT NULL,
  No_of_clients NUMBER DEFAULT 0,
  PRIMARY KEY (SSN),
  FOREIGN KEY (SSN) REFERENCES Staff(SSN) ON DELETE CASCADE
);
```

```
/
CREATE TABLE Vendor (
  Name VARCHAR2(20) NOT NULL,
  VendorName VARCHAR2(20),
  Model VARCHAR2(20),
  Price FLOAT NOT NULL,
  PRIMARY KEY (VendorName, Model)
);
CREATE TABLE Equipment (
  EquipID VARCHAR(10),
  Vendor VARCHAR2(20),
  Model VARCHAR2(20),
  Status VARCHAR2(20) NOT NULL,
  ResponsibleMaintainer CHAR(9),
  EquipAge NUMBER DEFAULT 0,
  PRIMARY KEY (EquipID),
  FOREIGN KEY (ResponsibleMaintainer) REFERENCES Maintainer(SSN) ON DELETE CASCADE,
  FOREIGN KEY (Vendor, Model) REFERENCES Vendor(VendorName, Model) ON DELETE CASCADE
);
/
CREATE TABLE Contract (
  ContractNo CHAR(10),
  ProgramNo NUMBER,
  CustomerPhone VARCHAR2(10),
  TrainerID NUMBER NOT NULL,
  SignUpDate DATE NOT NULL,
  Price FLOAT NOT NULL,
  Discount Float,
  PRIMARY KEY (ContractNo, ProgramNo, CustomerPhone),
  FOREIGN KEY (ProgramNo) REFERENCES FitnessProgram(ProgramID) ON DELETE CASCADE,
  FOREIGN KEY (CustomerPhone) REFERENCES Customer(Phone) ON DELETE CASCADE,
  FOREIGN KEY (TrainerID) REFERENCES Trainer(TrainerID) ON DELETE CASCADE
);
/
CREATE TABLE TrainerEligibility (
  ProgramID NUMBER,
  TrainerID NUMBER,
  PRIMARY KEY (ProgramID, TrainerID),
  FOREIGN KEY (ProgramID) REFERENCES FitnessProgram(ProgramID) ON DELETE CASCADE ,
  FOREIGN KEY (TrainerID) REFERENCES Trainer(TrainerID) ON DELETE CASCADE
);
/
CREATE TABLE ExerciseSession (
  SessionID CHAR(15),
  ProgramID NUMBER,
  CustomerPhone VARCHAR2(10),
  Duration NUMBER NOT NULL,
  EstCalories FLOAT NOT NULL,
  AvgHR NUMBER NOT NULL,
  MaxHR NUMBER NOT NULL,
  SDATE DATE NOT NULL,
  PRIMARY KEY (SessionID, ProgramID, CustomerPhone),
  FOREIGN KEY (ProgramID) REFERENCES FitnessProgram(ProgramID) ON DELETE CASCADE ,
  FOREIGN KEY (CustomerPhone) REFERENCES Customer(Phone) ON DELETE CASCADE
);
```

5 Trigger

5.1 Update Total Enrollment of Program

Each time a record was added to the CONTRACT table, the *TotalEnrollment* in table FITNESSPROGRAM of the particular program should add 1.

The trigger is defined as follow:

```
CREATE OR REPLACE TRIGGER addTotalEnrollmentTrigger
  AFTER INSERT ON CONTRACT
      FOR EACH ROW
BEGIN
  UPDATE FITNESSPROGRAM
    SET TOTALENROLLMENT = TOTALENROLLMENT + 1
    WHERE PROGRAMID = :new.PROGRAMNO;
  UPDATE TRAINER
    SET NO_OF_CLIENTS = NO_OF_CLIENTS + 1
    WHERE TRAINERID = :new.TRAINERID;
END;/
```

5.2 Log all the changes about Customer

Each time a record was added/updated/deleted in Customer table, record the modification in Customer_Log table.

The trigger is defined as follow:

```
CREATE TABLE Customer_Log ( log_date DATE, action VARCHAR2(50),
  old_Phone VARCHAR2(10), old_Gender CHAR, old_Age NUMBER,
  old_Height FLOAT, old_Weight FLOAT, old_ClosetNo VARCHAR2(10),
  new_Phone VARCHAR2(10), new_Gender CHAR, new_Age NUMBER,
  new_Height FLOAT, new_Weight FLOAT, new_ClosetNo VARCHAR2(10)
);

CREATE OR REPLACE TRIGGER customerLogTrigger
  AFTER INSERT OR UPDATE OR DELETE ON CUSTOMER
      FOR EACH ROW
DECLARE
  log_action  Customer_Log.action%TYPE;
BEGIN
  IF INSERTING THEN
    log_action := 'Insert';
    INSERT INTO Customer_Log
    VALUES (SYSDATE, log_action, null, null, null, null, null, null,
            :new.PHONE, :new.GENDER, :new.AGE,
            :new.HEIGHT, :new.WEIGHT, :new.CLOSETNO );
  ELSIF UPDATING THEN
    log_action := 'Update';
    INSERT INTO Customer_Log
    VALUES (SYSDATE, log_action,
            :old.PHONE, :old.GENDER, :old.AGE,
            :old.HEIGHT, :old.WEIGHT, :old.CLOSETNO,
            :new.PHONE, :new.GENDER, :new.AGE,
            :new.HEIGHT, :new.WEIGHT, :new.CLOSETNO );
  ELSIF DELETING THEN
    log_action := 'Delete';
    INSERT INTO Customer_Log
    VALUES (SYSDATE, log_action,
            :old.PHONE, :old.GENDER, :old.AGE,
            :old.HEIGHT, :old.WEIGHT, :old.CLOSETNO,
            null, null, null, null, null, null );
  ELSE
    DBMS_OUTPUT.PUT_LINE('This code is not reachable.');
  END IF;
END;/
```

6 Procedure

6.1 Delete old Exercise Session Data for Customer

According to requirement, the database should only keep exercise session data of non-member customer for two years, three for customer with expired membership, and forever for current member. Therefore, there would be a circumstance that we want to delete exercise session data of two years old for customer without member, as well as the exercise session of three years old for customer with expired member card. The procedure is defined as follow:

```
CREATE OR REPLACE PROCEDURE deleteSessionOfIdleCustomer
  ( WithoutMemerCard      IN NUMBER DEFAULT 730,
    WithExpireMemerCard   IN NUMBER DEFAULT 1095 ) AS
  CURSOR CustomerWithoutMemberCard IS
    SELECT  CUSTOMER.PHONE
    FROM    CUSTOMER
    WHERE   PHONE NOT IN (SELECT PHONE FROM MEMBERCARD);
  CURSOR CustomerWithExpireMemberCard IS
    SELECT  CUSTOMER.PHONE
    FROM    CUSTOMER
    WHERE   PHONE IN (
      SELECT PHONE FROM MEMBERCARD WHERE EXPIRATIONDATE < SYSDATE);
  thisWithoutCard CustomerWithoutMemberCard%ROWTYPE;
  thisExpiredCard CustomerWithExpireMemberCard%ROWTYPE;
BEGIN
  OPEN CustomerWithoutMemberCard;
  LOOP
    FETCH CustomerWithoutMemberCard INTO thisWithoutCard;
    EXIT WHEN CustomerWithoutMemberCard%NOTFOUND;
    DELETE FROM EXERCISESESSION
      WHERE CUSTOMERPHONE = thisWithoutCard.PHONE
        AND SDATE < SYSDATE - WithoutMemerCard;
   END LOOP;
  CLOSE CustomerWithoutMemberCard;
  OPEN CustomerWithExpireMemberCard;
  LOOP
    FETCH CustomerWithExpireMemberCard INTO thisExpiredCard;
    EXIT WHEN CustomerWithExpireMemberCard%NOTFOUND;
    DELETE FROM EXERCISESESSION
      WHERE CUSTOMERPHONE = thisExpiredCard.PHONE
        AND SDATE < SYSDATE - WithExpireMemerCard;
   END LOOP;
  CLOSE CustomerWithExpireMemberCard;
END;/
```

6.2 Add Bonus for Specific Member

Assume a deal event want to add some bonus time for member registered during specific period. The procedure is defined as follow:

```
CREATE OR REPLACE PROCEDURE addBonusTime
  ( StartDate IN MEMBERCARD.VALIDATIONDATE%TYPE,
    EndDate IN MEMBERCARD.VALIDATIONDATE%TYPE,
    bonusTime IN NUMBER DEFAULT 90 ) AS
  CURSOR EligibleMember IS
    SELECT MEMBERCARD.CARDNO
    FROM MEMBERCARD
    WHERE VALIDATIONDATE >= StartDate
      AND VALIDATIONDATE <= EndDate
      AND EXPIRATIONDATE - VALIDATIONDATE > 180;
  thisMember EligibleMember%ROWTYPE;
BEGIN
  OPEN EligibleMember;
  LOOP
    FETCH EligibleMember INTO thisMember;
    EXIT WHEN EligibleMember%NOTFOUND;

    UPDATE MEMBERCARD
      SET EXPIRATIONDATE = EXPIRATIONDATE + bonusTime;
   END LOOP;
  CLOSE EligibleMember;
END;/
```

7 Business Rule

7.1 Check the Age of the Equipment

Assume not allowing a equipment older than 5 years, the constraint is defined as follow:

```
ALTER TABLE EQUIPMENT
ADD CONSTRAINT CheckEquipAge
  CHECK (EquipAge <= 5);
```

7.2 Check the TotalEnrollment of a Fitness Program

Assume not allowing a fitness program's total enrollment shall not exceed 50, the constraint is defined as follow:

```
ALTER TABLE FitnessProgram
  ADD CONSTRAINT CheckTOTALENROLLMENT
  CHECK (TOTALENROLLMENT <= 50);
```

7.3 Check the Number of client of a Fitness Trainer

Assume not allowing a fitness trainer's client number shall not exceed 50, the constraint is defined as follow:

```
ALTER TABLE Trainer
  ADD CONSTRAINT CheckNo_of_clients
  CHECK (No_of_clients <= 50);
```

8 Appendix

Here is some sample data.

```
--Sample Data
--Closet Data
INSERT INTO CLOSET (NO, CLOSETSIZE) VALUES ('0000000001','Small');/
INSERT INTO CLOSET (NO, CLOSETSIZE) VALUES ('0000000002','Small');/
INSERT INTO CLOSET (NO, CLOSETSIZE) VALUES ('0000000003','Small');/
INSERT INTO CLOSET (NO, CLOSETSIZE) VALUES ('0000000004','Small');/
INSERT INTO CLOSET (NO, CLOSETSIZE) VALUES ('0000000005','Middle');/
INSERT INTO CLOSET (NO, CLOSETSIZE) VALUES ('0000000006','Middle');/
INSERT INTO CLOSET (NO, CLOSETSIZE) VALUES ('0000000007','Middle');/
INSERT INTO CLOSET (NO, CLOSETSIZE) VALUES ('0000000008','Middle');/
INSERT INTO CLOSET (NO, CLOSETSIZE) VALUES ('0000000009','Large');/
INSERT INTO CLOSET (NO, CLOSETSIZE) VALUES ('0000000010','Large');/
INSERT INTO CLOSET (NO, CLOSETSIZE) VALUES ('0000000011','Large');/
INSERT INTO CLOSET (NO, CLOSETSIZE) VALUES ('0000000012','Large');/
--Customer Data
INSERT INTO CUSTOMER (PHONE, FNAME, MINIT, LNAME, GENDER, AGE, HEIGHT,
    WEIGHT, CLOSETNO)
  VALUES ('4695621000','San','M','Zhang','M',25,5.10,180,'0000000001')
     ;/
INSERT INTO CUSTOMER (PHONE, FNAME, MINIT, LNAME, GENDER, AGE, HEIGHT,
    WEIGHT, CLOSETNO)
  VALUES ('4695622000','Si','M','Li','M',26,6.3,170,'0000000006');/
INSERT INTO CUSTOMER (PHONE, FNAME, MINIT, LNAME, GENDER, AGE, HEIGHT,
    WEIGHT, CLOSETNO)
  VALUES ('4695623000','Wu','M','Wang','F',30,4.10,130,'0000000010');/
```

```sql
INSERT INTO CUSTOMER (PHONE, FNAME, MINIT, LNAME, GENDER, AGE, HEIGHT, WEIGHT, CLOSETNO)
  VALUES (’4695624000’,’John’,’M’,’Keats’,’M’,55,5.6,150,’0000000011’);
--MemberCard Data
INSERT INTO MEMBERCARD (CARDNO, PHONE, VALIDATIONDATE, EXPIRATIONDATE)
  VALUES (’1000000001’,’4695621000’,’30-MAR-16’,’30-MAR-17’);/
INSERT INTO MEMBERCARD (CARDNO, PHONE, VALIDATIONDATE, EXPIRATIONDATE)
  VALUES (’1000000002’,’4695623000’,’01-OCT-15’,’01-OCT-16’);/
--FitnessProgram Data
INSERT INTO FITNESSPROGRAM (PROGRAMID, NAME, DESCRIPTION, PRICE)
  VALUES (1,’Jogging’,’Jogging is good for life.’,100.0);/
INSERT INTO FITNESSPROGRAM (PROGRAMID, NAME, DESCRIPTION, PRICE)
  VALUES (2,’Insanity’,’Fight for life.’,120.0);/
INSERT INTO FITNESSPROGRAM (PROGRAMID, NAME, DESCRIPTION, PRICE)
  VALUES (3,’Music Cycling’,’Cycling is good for life.’,80.0);/
INSERT INTO FITNESSPROGRAM (PROGRAMID, NAME, DESCRIPTION, PRICE)
  VALUES (4,’Power’,’Strength is good for life.’,130.0);/
INSERT INTO FITNESSPROGRAM (PROGRAMID, NAME, DESCRIPTION, PRICE)
  VALUES (5,’T25’,’HIIT is good for life.’,110.0);/
--AerobicInformation Data
INSERT INTO AEROBICINFORMATION (AEROBICTYPE, AEROBICINFORMATION)
  VALUES (’Jogging’,’Jogging is a form of trotting or running at a slow or leisurely pace.’);/
INSERT INTO AEROBICINFORMATION (AEROBICTYPE, AEROBICINFORMATION)
  VALUES (’Cycling’,’Cycling, also called bicycling or biking, is the use of bicycles for transport, recreation, exercise or sport.’);/
--AerobicProgram Data
INSERT INTO AEROBICPROGRAM (PROGRAMID, INTENSITY, MINIMALTIME, AEROBICTYPE)
  VALUES (1,’Low’,30,’Jogging’);/
INSERT INTO AEROBICPROGRAM (PROGRAMID, INTENSITY, MINIMALTIME, AEROBICTYPE)
  VALUES (3,’Middle’,20,’Cycling’);/
--StrengthProgram Data
INSERT INTO STRENGTHPROGRAM (PROGRAMID, STRENGTHLEVEL) VALUES (4,’Expert’);/
--HIITProgram Data
INSERT INTO HIITPROGRAM (PROGRAMID, INTENSITY, DURATION)
  VALUES (2,’High’,60);/
INSERT INTO HIITPROGRAM (PROGRAMID, INTENSITY, DURATION)
  VALUES (5,’Middle’,40);/
--Staff Data
INSERT INTO STAFF (SSN, FNAME, MINIT, LNAME, PHONE, SALARY, ADDRESS)
```

```
  VALUES (’200000001’,’Jon’,’M’,’Steward’,’4695624000’,6000,’900 Civic Center Dr, Richardson, TX 75080’);/
INSERT INTO STAFF (SSN, FNAME, MINIT, LNAME, PHONE, SALARY, ADDRESS)
  VALUES (’200000002’,’Trevor’,’M’,’Noah’,’4695625000’,7000,’800 West Campbell Road, Richardson, TX 75080’);/
INSERT INTO STAFF (SSN, FNAME, MINIT, LNAME, PHONE, SALARY, ADDRESS)
  VALUES (’200000003’,’Steven’,’M’,’Colbert’,’4695626000’,8000,’6400 Frankford Rd, Dallas, TX 75252’);/
--Maintainer Data
INSERT INTO MAINTAINER (SSN) VALUES (’200000003’);/
--Trainer Data
INSERT INTO TRAINER (SSN, LICENSEID, TRAINERID, NO_OF_YEARS)
  VALUES (’200000001’,’6000000001’,1,16);/
INSERT INTO TRAINER (SSN, LICENSEID, TRAINERID, NO_OF_YEARS)
  VALUES (’200000002’,’6000000002’,2,1);/
--Vendor Data
INSERT INTO VENDOR (NAME, VENDORNAME, MODEL, PRICE)
  VALUES (’Treadmill’,’Decathlon’,’TR0001’,210);/
INSERT INTO VENDOR (NAME, VENDORNAME, MODEL, PRICE)
  VALUES (’Treadmill’,’Decathlon’,’TR0002’,310);/
INSERT INTO VENDOR (NAME, VENDORNAME, MODEL, PRICE)
  VALUES (’Cycle Trainer’,’Decathlon’,’CL0004’,110);/
INSERT INTO VENDOR (NAME, VENDORNAME, MODEL, PRICE)
  VALUES (’Cycle Trainer’,’Nordic’,’CT0001’,150);/
--Equipment Data
INSERT INTO EQUIPMENT (EQUIPID, VENDOR, MODEL, STATUS, RESPONSIBLEMAINTAINER)
  VALUES (’E000000001’,’Decathlon’,’TR0001’,’In Use’,’200000003’);/
INSERT INTO EQUIPMENT (EQUIPID, VENDOR, MODEL, STATUS, RESPONSIBLEMAINTAINER)
  VALUES (’E000000002’,’Decathlon’,’TR0001’,’Idle’,’200000003’);/
INSERT INTO EQUIPMENT (EQUIPID, VENDOR, MODEL, STATUS, RESPONSIBLEMAINTAINER)
  VALUES (’E000000003’,’Decathlon’,’TR0002’,’In Use’,’200000003’);/
INSERT INTO EQUIPMENT (EQUIPID, VENDOR, MODEL, STATUS, RESPONSIBLEMAINTAINER)
  VALUES (’E000000004’,’Decathlon’,’CL0004’,’In Use’,’200000003’);/
INSERT INTO EQUIPMENT (EQUIPID, VENDOR, MODEL, STATUS, RESPONSIBLEMAINTAINER)
  VALUES (’E000000005’,’Nordic’,’CT0001’,’In Use’,’200000003’);/
--Contract Data
INSERT INTO CONTRACT (CONTRACTNO, PROGRAMNO, CUSTOMERPHONE, TRAINERID, SIGNUPDATE, PRICE, DISCOUNT)
```

```
  VALUES (’0000000001’,1,’4695621000’,1,’30-MAR-16’,100,1);/
INSERT INTO CONTRACT (CONTRACTNO, PROGRAMNO, CUSTOMERPHONE, TRAINERID,
    SIGNUPDATE, PRICE, DISCOUNT)
  VALUES (’0000000002’,4,’4695623000’,2,’01-OCT-15’,130,1);/
--TrainerEligibility Data
INSERT INTO TRAINERELIGIBILITY (PROGRAMID, TRAINERID) VALUES (1,1);/
INSERT INTO TRAINERELIGIBILITY (PROGRAMID, TRAINERID) VALUES (3,1);/
INSERT INTO TRAINERELIGIBILITY (PROGRAMID, TRAINERID) VALUES (1,2);/
INSERT INTO TRAINERELIGIBILITY (PROGRAMID, TRAINERID) VALUES (2,2);/
INSERT INTO TRAINERELIGIBILITY (PROGRAMID, TRAINERID) VALUES (3,2);/
INSERT INTO TRAINERELIGIBILITY (PROGRAMID, TRAINERID) VALUES (4,2);/
INSERT INTO TRAINERELIGIBILITY (PROGRAMID, TRAINERID) VALUES (5,2);/
--ExerciseSession Data
INSERT INTO EXERCISESESSION (SESSIONID, PROGRAMID, CUSTOMERPHONE,
    DURATION, ESTCALORIES, AVGHR, MAXHR, SDATE)
  VALUES (’000000000000001’,1,’4695621000’,60,900,160,180,SYSDATE);/
INSERT INTO EXERCISESESSION (SESSIONID, PROGRAMID, CUSTOMERPHONE,
    DURATION, ESTCALORIES, AVGHR, MAXHR, SDATE)
  VALUES (’000000000000002’,1,’4695621000’,60,900,160,180,SYSDATE-2000)
      ;/
INSERT INTO EXERCISESESSION (SESSIONID, PROGRAMID, CUSTOMERPHONE,
    DURATION, ESTCALORIES, AVGHR, MAXHR, SDATE)
  VALUES (’000000000000003’,1,’4695622000’,60,900,160,180,SYSDATE-500)
      ;/
INSERT INTO EXERCISESESSION (SESSIONID, PROGRAMID, CUSTOMERPHONE,
    DURATION, ESTCALORIES, AVGHR, MAXHR, SDATE)
  VALUES (’000000000000004’,1,’4695622000’,60,900,160,180,SYSDATE-1000)
      ;/
```