Fitness Center

CS 6360.005 - Database Design - F15

JOYDEEP ROY (jxr150430)
YOGESH DARJI (ypd150030)
AKSHAY THAKARE (ast140230)

1. Detailed Project Description

This project considers all the details of a small-medium scale fitness center. The Fitness center for which the Database system is designed is a multi-branch fitness center which houses exercise equipment for the purpose of physical exercise. It also includes facilities like cardio workout session, group exercise classes (like aerobics, yoga etc.), personal training and also houses sauna and steam shower facilities.

Why is Fitness center important?

In today's world, health and fitness plays and important role. Managing the fitness center to fulfill he needs of users is really important.
Database system can at times becomes complex as there are many relations involved among different entities. In this project we show case the complex Fitness Center System by a simplified Entity-Relationship Diagram and make sure that the Relational Schema fulfills all the needs of Fitness Center.

Requirements Analysis

The Users of the Fitness Center Database Application are as below:

- CUSTOMERS
  - Customer can register at the Fitness Center. While registering he will mention his/her preferred batch (timing) and the facilities he want to sign up for.
  - Customers from their online portal can access their Batch Details, Personal Trainer information, Branch Details and can get a copy of their latest fitness report.
  - Customer can update their profile online. They can modify their batch, trainer or the type of facilities they want to use.
  - Customer can upgrade their membership
  - Customer can access different batches details, the trainers in those batches, and also how full the batch is and can accordingly switch to a different batch.
- EMPLOYEES
  - Manager
    - Manager will manage the entire Fitness Branch
    - Manager will manage the Trainer, Clerk and assign work to them
    - Manager would be responsible for creating Batches
  - Trainer
    - Trainer will run the batches and also provide Personal Training to Customers
  - Clerk
    - Clerk will manage all the clerical activities like manage Customer Report and maintaining Equipment.

Assumptions and Constraints in the designed Fitness Center

- **Facilities** in this fitness center are like packages or membership levels.
  For example: Sauna/Steam Bath, Group Exercise Facility (Includes Jumba, Aerobics, Yoga, Pilates, and Self Defense classes), Cardio Facilities and Weight Training.

  One **Customer** can register for one **Facility**.
  So, for example if he/she can register for:

  a. Sauna/Steam Bath+ Group Exercise Facility +Cardio Facilities +Weight Training.
  b. Group Exercise Facility +Cardio Facilities +Weight Training.
  c. Cardio Facilities +Weight Training.
  d. Weight Training.

  Also, all the branches may/may not offer all the facilities/packages.

- One **Clerk** can maintain multiple **Customer Report.** There may also be a clerk who maintains no customer report.

- One **Clerk** can maintain multiple **Equipments.** There may also be a clerk who maintains no equipment.

- One **Fitness-Branch** will have multiple **Equipments.**

- Only the latest **Fitness Report** for each **Customer** will be stored in the system and will be available for viewing.

- A Batch in the Fitness Center is like the timing and type. For Example, we can have morning batch starting at 9:00 am to 10 am and another batch 11 am to 12 pm.

  A **Customer** would be registered to attend only one **Batch.** He may later change his batch through online portal subject to how filled it is and availability.

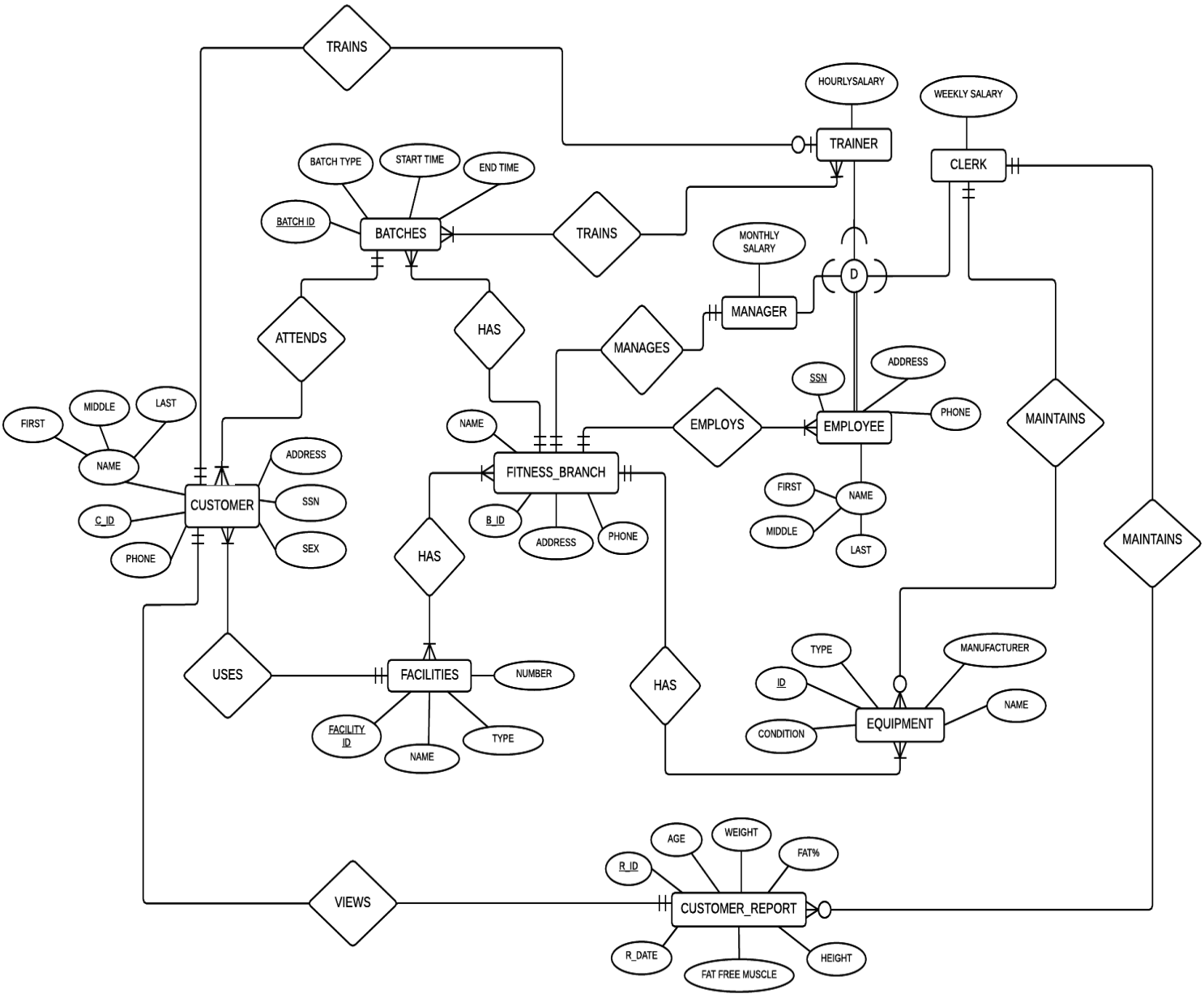
- A **Batch** will be trained by multiple **trainers** and also a trainer can train multiple Batches.

- A **Customer** may or may not have a **Personal Trainer.**

- An **Employee** can be a **Manager**, **Trainer** or a **Clerk.**

- There is One **Manager** for each **Fitness-Branch.**

2. EER Diagram



The Entities and Relationship in the above EER diagram are as below:

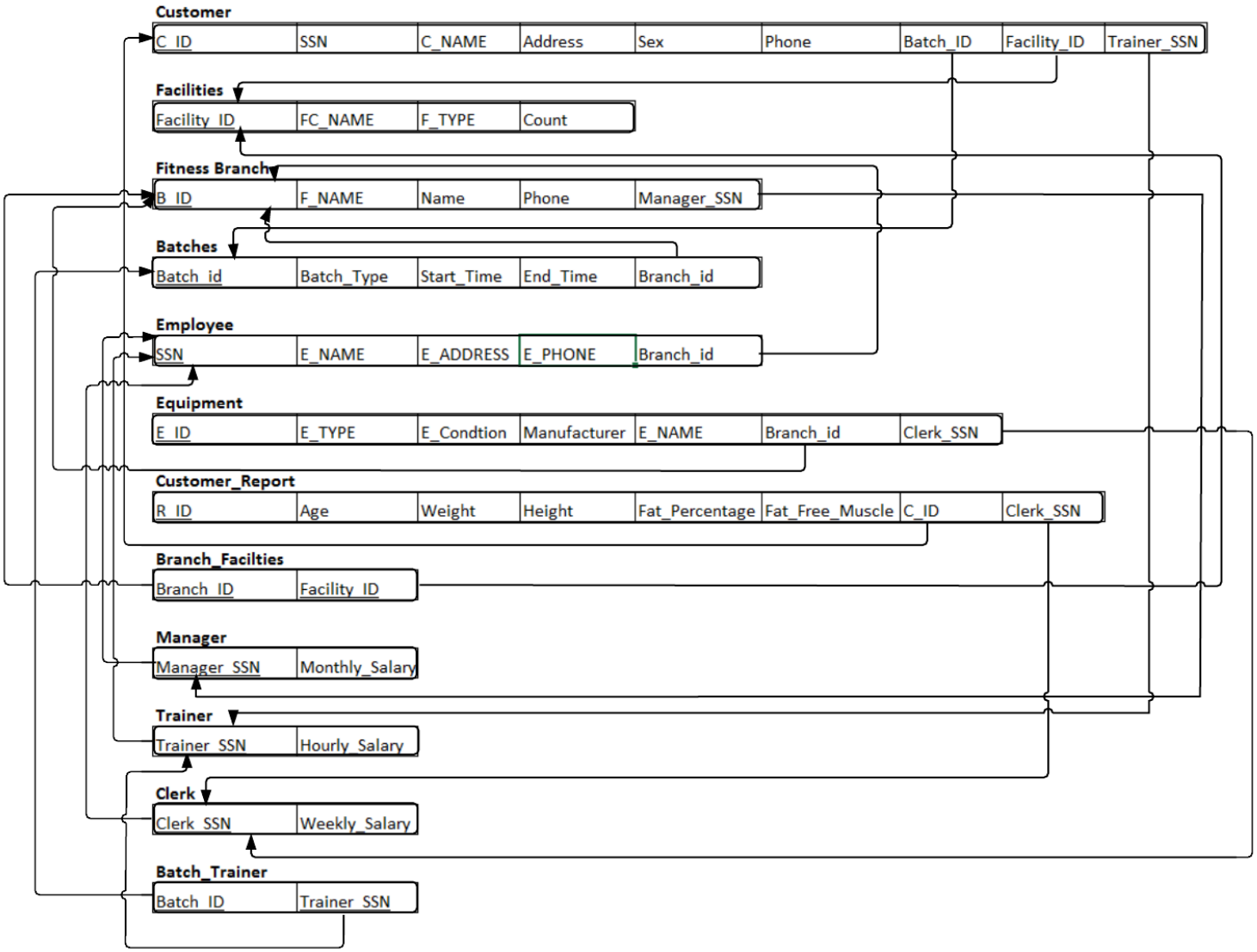
Entities:

1. **Customer -** A Customer entity represents a particular customer. Its primary key is C_ID to distinguish different customers in the database system.
2. **Batches** – A Batch entity represents a particular instance of a Batch uniquely identified by its Batch_ID. It includes details like Start/End time of the batch and the Batch Type.
3. **Facilities –** A Facility is associated with the Facility Type like Sauna, Cardio or Weight Training. It is identified by its Facility_ID.
4. **Fitness Branch –** This is an instance of a Fitness Branch. It will be storing the details of the Branch like address, phone number, and name and will be uniquely identified by Branch Id (B_ID).
5. **Employee –** Employee entity represents any employee in the fitness center and may be classified as Manager, Trainer or Clerk. In general it will be storing the employee details like Name, SSN, address, phone number etc. Specializations like **Manager**, **Clerk** and **Trainer** will have monthly, weekly and hourly salaries. SSN is the key field for this entity.
6. **Equipment –** Represents the various equipment in the fitness center. It will have details like Type, Manufacturer, and Name and will also store the condition of the equipment.
7. **Customer Report –** A Customer report will store the latest details of a customer's progress in the fitness center. It will store Age, weight, height, fat %, fat free muscle mass, and report date. Each customer report will be identified by a unique report id (R_ID).

Relationships:

1. **Customer – Batches:** One customer will be registered in one batch. One Batch may have many customers.
2. **Customer – Trainer:** One customer can have one personal trainer. A trainer can be a personal trainer to only one customer.
3. **Customer – Customer Report:** One customer will be having a single report.
4. **Customer – Facilities:** One customer can register for a single particular facility (package). Multiple customers can have similar facilities.
5. **Batches – Trainer:** One single batch can have multiple trainers. A single trainer can also train multiple batches.
6. **Batches – Fitness_Branch:** One Fitness Branch may have multiple batches. A particular Batch will be in a specific fitness branch.
7. **Fitness_Branch – Employee:** One Fitness Branch can have multiple employee. A single employee will belong to a particular fitness branch.
8. **Fitness_Branch – Manager:** One Manager will be managing a single Fitness Branch. Each branch will have only one manager.
9. **Fitness_Branch – Equipments:** One Fitness Branch will have multiple equipment. A single equipment will belong to a particular fitness branch.
10. **Clerk – Equipment:** One clerk may/may not maintain multiple equipment. One equipment will be maintained by only one clerk.
11. **Clerk – Customer Report:** One clerk may/may not maintain multiple customer report. One customer report will be maintained by only one clerk.

3. Relational Schema



4. Functional Dependencies

CUSTOMER
C_ID → {SSN, C_NAME, ADDRESS, SEX, PHONE, BATCH_ID, FACILITY_ID, TRAINER_SSN}

FACILITIES
FACILITY_ID → {FC_NAME, F_TYPE, COUNT}

FITNESS_BRANCH
B_ID → {ADDRESS, F_NAME, PHONE, MANAGER_SSN}

BATCHES
BATCH_ID → {BATCH_TYPE, START_TIME, END_TIME, BRANCH_ID}

F_EMPLOYEE
SSN → {E_NAME, E_ADDRESS, E_PHONE, BRANCH_ID}

EQUIPMENT
E_ID → {AGE, E_CONDITION, MANUFACTURER, E_NAME, BRANCH_ID, CLERK_SSN}

CUSTOMER_REPORT
R_ID → {AGE, WEIGHT, HEIGHT, FAT_PERCENTAGE, FAT_FREE_MUSCLE, C_ID, CLERK_SSN}

MANAGERS
MANAGER_SSN → {MONTHLY_SALARY}

TRAINER
TRAINER_SSN → HOURLY_SALARY

CLERK
CLERK_SSN → WEEKLY_SALARY

5. Database Normalization

From our database it seems that there can be one functional dependency in Customer Table for SSN → {C_NAME, ADDRESS, SEX, PHONE}.But here we have no information that depends on it or its components. Thus splitting them is not desired as that may increase number of queries unnecessarily. Hence the database is already in 3 NF.

6. Final Relational Schema

As there is no normalization the final relational schema is same as above.

7. SQL

a) Table create queries:

CUSTOMER	CREATE TABLE CUSTOMER (C_ID VARCHAR2(5) NOT NULL, SSN NUMBER(9) NOT NULL, C_NAME VARCHAR2(30) NOT NULL, ADDRESS VARCHAR2(100), SEX VARCHAR2(5) NOT NULL, PHONE NUMBER(12), BATCH_ID VARCHAR2(5), FACILITY_ID VARCHAR2(5), TRAINER_SSN VARCHAR2(5), PRIMARY KEY(C_ID));
FACILITIES	CREATE TABLE FACILITIES (FACILITY_ID VARCHAR2(5) NOT NULL, FC_NAME VARCHAR2(20), F_TYPE VARCHAR2(30), COUNT NUMBER(5), PRIMARY KEY(FACILITY_ID));
FITNESS_BRANCH	CREATE TABLE FITNESS_BRANCH (B_ID VARCHAR2(5) NOT NULL, ADDRESS VARCHAR2(50) NOT NULL, F_NAME VARCHAR2(30) NOT NULL, PHONE NUMBER(12) NOT NULL, MANAGER_SSN NUMBER(9), PRIMARY KEY(B_ID));
BATCHES	CREATE TABLE BATCHES (BATCH_ID VARCHAR2(5) NOT NULL, BATCH_TYPE VARCHAR2(50) NOT NULL, START_TIME VARCHAR2(5) NOT NULL, END_TIME VARCHAR2(5), BRANCH_ID VARCHAR2(5) NOT NULL, PRIMARY KEY(BATCH_ID));
BATCH_TRAINER	CREATE TABLE BATCH_TRAINER (BATCH_ID VARCHAR2(5) NOT NULL, TRAINER_SSN NUMBER(9) NOT NULL, PRIMARY KEY (BATCH_ID, TRAINER_SSN));

CLERK	CREATE TABLE CLERK (CLERK_SSN NUMBER(9) NOT NULL, WEEKLY_SALARY NUMBER(9) NOT NULL, PRIMARY KEY (CLERK_SSN));
TRAINER	CREATE TABLE TRAINER (TRAINER_SSN NUMBER(9) NOT NULL, HOURLY_SALARY NUMBER(9) NOT NULL, PRIMARY KEY (TRAINER_SSN));
MANAGERS	CREATE TABLE MANAGERS (MANAGER_SSN NUMBER(9) NOT NULL, MONTHLY_SALARY NUMBER(9) NOT NULL, PRIMARY KEY (MANAGER_SSN));
BRANCH_FACILITIES	CREATE TABLE BRANCH_FACILITIES (BRANCH_ID VARCHAR2(5) NOT NULL, FACILITY_ID VARCHAR2(5) NOT NULL, PRIMARY KEY (BRANCH_ID, FACILITY_ID));
CUSTOMER_REPORT	CREATE TABLE CUSTOMER_REPORT (R_ID VARCHAR2(5) NOT NULL, AGE NUMBER(3) NOT NULL, WEIGHT NUMBER(5,2) NOT NULL, HEIGHT NUMBER(5,2) NOT NULL, FAT_PERCENTAGE NUMBER(3,2) NOT NULL, FAT_FREE_MUSCLE NUMBER(3,2) NOT NULL, C_ID VARCHAR2(5) NOT NULL, CLERK_SSN NUMBER(9), PRIMARY KEY (R_ID));
EQUIPMENT	CREATE TABLE EQUIPMENT (E_ID VARCHAR2(5) NOT NULL, E_TYPE VARCHAR2(20) NOT NULL, E_CONDITION VARCHAR2(20), MANUFACTURER VARCHAR2(20), E_NAME NUMBER(3,2) NOT NULL, BRANCH_ID VARCHAR2(5) NOT NULL, CLERK_SSN NUMBER(9), PRIMARY KEY (E_ID));

F_EMPLOYEE	CREATE TABLE F_EMPLOYEE (SSN NUMBER(9) NOT NULL, E_NAME VARCHAR2(20) NOT NULL, E_ADDRESS VARCHAR2(55) NOT NULL, E_PHONE VARCHAR2(10), BRANCH_ID VARCHAR2(5), PRIMARY KEY (SSN));

b) Table alter queries (Adding the Foreign Keys):

CUSTOMER	ALTER TABLE CUSTOMER ADD CONSTRAINT FK_CUS_BATCHID FOREIGN KEY(BATCH_ID) REFERENCES BATCHES (BATCH_ID) ON DELETE SET NULL; **Note:** If a particular batch is deleted the Batch_ID in customer is set to null. i.e. Customer needs to register in new batch.
	ALTER TABLE CUSTOMER ADD CONSTRAINT FK_CUS_TSSN FOREIGN KEY(TRAINER_SSN) REFERENCES TRAINER (TRAINER_SSN) ON DELETE SET NULL; **Note:** If Trainer is deleted then Trainer_SSN in customer is set to NULL. So customer can choose a new personal trainer if wanted.
	ALTER TABLE CUSTOMER ADD CONSTRAINT FK_CUS_FID FOREIGN KEY(FACILITY_ID) REFERENCES FACILITIES (FACILITY_ID) ON DELETE SET NULL; **Note:** If Facility (Package) is removed then Facility_ID is set to NULL and customer will be notified to choose a different package.
FITNESS_BRANCH	ALTER TABLE FITNESS_BRANCH ADD CONSTRAINT FK_FBRANCH_MSSN FOREIGN KEY(MANAGER_SSN) REFERENCES MANAGERS(MANAGER_SSN) ON DELETE SET NULL; **Note:** If a Fitness Branch is deleted, a customer will be asked to register in a new fitness branch.

BATCHES	ALTER TABLE BATCHES ADD CONSTRAINT FK_BATCH_BID FOREIGN KEY(BRANCH_ID) REFERENCES FITNESS_BRANCH (B_ID) ON DELETE CASCADE; **Note:** If a particular Fitness Branch is deleted, then all its associated batches are also deleted.
F_EMPLOYEE	ALTER TABLE F_EMPLOYEE ADD CONSTRAINT FK_EMP_BID FOREIGN KEY(BRANCH_ID) REFERENCES FITNESS_BRANCH (B_ID) ON DELETE SET NULL; **Note:** If a Fitness Branch is deleted, then the Branch_ID for all its employees are set to NULL.
EQUIPMENT	ALTER TABLE EQUIPMENT ADD CONSTRAINT FK_EQP_BID FOREIGN KEY(BRANCH_ID) REFERENCES FITNESS_BRANCH (B_ID) ; **Note:** If a Fitness Branch is deleted, then the Branch_ID for the Equipments are set to NULL.
	ALTER TABLE EQUIPMENT ADD CONSTRAINT FK_EQP_CSSN FOREIGN KEY(CLERK_SSN) REFERENCES CLERK(CLERK_SSN) ON DELETE SET NULL; **Note:** If a Fitness Branch is deleted, then the Branch_ID for the Equipments are set to NULL. So, all the free equipment can be either stored in storage room or allocated to other branches.
CUSTOMER_REPORT	ALTER TABLE CUSTOMER_REPORT ADD CONSTRAINT FK_CR_CID FOREIGN KEY(C_ID) REFERENCES CUSTOMER(C_ID) ON DELETE CASCADE; **Note:** If a Customer is deleted, its associated customer report is also deleted form the system.
	ALTER TABLE CUSTOMER_REPORT ADD CONSTRAINT FK_CR_CSSN FOREIGN KEY(CLERK_SSN) REFERENCES CLERK(CLERK_SSN) ON DELETE SET NULL;

	Note: If a Clerk is deleted, then the Clerk_ID for all the customer reports he/she made will be set to NULL.
BRANCH_FACILITIES	ALTER TABLE BRANCH_FACILITIES ADD CONSTRAINT FK_BF_FID FOREIGN KEY(FACILITY_ID) REFERENCES FACILITIES(FACILITY_ID) ON DELETE CASCADE; **Note**: If a Fitness Facility is deleted, then the corresponding record in Branch_Facilities is also deleted.
	ALTER TABLE BRANCH_FACILITIES ADD CONSTRAINT FK_BF_BID FOREIGN KEY(BRANCH_ID) REFERENCES FITNESS_BRANCH(B_ID) ON DELETE CASCADE; **Note**: If a Fitness Facility is deleted, then the corresponding record in Branch_Facilities is also deleted.
MANAGERS	ALTER TABLE MANAGERS ADD CONSTRAINT FK_MGR_SSN FOREIGN KEY(MANAGER_SSN) REFERENCES F_EMPLOYEE(SSN) ON DELETE CASCADE; **Note**: If an employee who is manager is deleted, then the corresponding record in Managers table is also deleted.
CLERK	ALTER TABLE CLERK ADD CONSTRAINT FK_CLERK_SSN FOREIGN KEY(CLERK_SSN) REFERENCES F_EMPLOYEE(SSN) ON DELETE CASCADE; **Note**: If an employee who is clerk is deleted, then the corresponding record in Clerk table is also deleted.
TRAINER	ALTER TABLE TRAINER ADD CONSTRAINT FK_TRAINER_SSN FOREIGN KEY(TRAINER_SSN) REFERENCES F_EMPLOYEE(SSN) ON DELETE CASCADE; **Note**: If an employee who is trainer is deleted, then the corresponding record in Trainers table is also deleted.

BATCH_TRAINER	ALTER TABLE BATCH_TRAINER ADD CONSTRAINT FK_BTRNR_BID FOREIGN KEY(BATCH_ID) REFERENCES BATCHES(BATCH_ID) ON DELETE CASCADE; **Note**: If a batch is deleted, then the corresponding records in Batch_Trainer are also deleted.
	ALTER TABLE BATCH_TRAINER ADD CONSTRAINT FK_BTRNR_TSSN FOREIGN KEY(TRAINER_SSN) REFERENCES TRAINER(TRAINER_SSN) ON DELETE CASCADE; **Note**: If a Trainer is deleted, then the corresponding records in Batch_Trainer are also deleted.

c) Table insert queries:

CUSTOMER	INSERT INTO CUSTOMER VALUES('C01', 987654321, 'YOGESH', 'RICHARSON', 'MALE', 4696643693, 'B001', 'F001', 111111111); INSERT INTO CUSTOMER VALUES('C02', 987654322, 'AKSHAY', 'DALLAS', 'MALE', 4696643694, 'B002', 'F002', 111111112); INSERT INTO CUSTOMER VALUES('C03', 987654323, 'JOY', 'PLANO', 'FEMALE', 4696643678, 'B003', 'F003', 111111113); INSERT INTO CUSTOMER VALUES('C04', 987654324, 'JOHN', 'GARLAND', 'MALE', 4696643698, 'B004', 'F004', 111111114); INSERT INTO CUSTOMER VALUES('C05', 987654325, 'DAVID', 'ALLEN', 'FEMALE', 4696643673, 'B005', 'F005', 111111115);
FACILITIES	INSERT INTO FACILTIES VALUES('F01', 'CARDIO','INDOOR', 25); INSERT INTO FACILTIES VALUES('F02', 'STEAM','INDOOR', 30); INSERT INTO FACILTIES VALUES('F03', 'SONA','INDOOR', 10); INSERT INTO FACILTIES VALUES('F04', 'CROSS-FIT','INDOOR', 22); INSERT INTO FACILTIES VALUES('F05', 'CYCLING','OUTDOOR', 45);

FITNESS_BRANCH	INSERT INTO FITNESS_BRANCH VALUES('BR01', 'RICHARDSON','GALAXY', 234123432, 333333331); INSERT INTO FITNESS_BRANCH VALUES('BR02', 'DALLAS','BLUESTAR', 234123439, 333333332); INSERT INTO FITNESS_BRANCH VALUES('BR03', 'PLANO','GROUNDFLY', 251232123, 333333333); INSERT INTO FITNESS_BRANCH VALUES('BR04', 'GARLAND','GAUANT', 258976321, 333333334); INSERT INTO FITNESS_BRANCH VALUES('BR05', 'ALLEN','KCLOCK', 258976321, 333333335);
BATCHES	INSERT INTO BATCHES VALUES('BT01','MORNING', '06:00', '10:00', 'BR01'); INSERT INTO BATCHES VALUES('BT02','MORNING', '06:00', '10:00', 'BR01'); INSERT INTO BATCHES VALUES('BT03','MORNING', '08:00', '11:00', 'BR01'); INSERT INTO BATCHES VALUES('BT04','MORNING', '08:00', '10:00', 'BR04'); INSERT INTO BATCHES VALUES('BT05','MORNING', '18:00', '21:00', 'BR05');
BATCH_TRAINER	INSERT INTO BATCH_TRAINER VALUES('BT01','444444441'); INSERT INTO BATCH_TRAINER VALUES('BT02','444444442'); INSERT INTO BATCH_TRAINER VALUES('BT03','444444443'); INSERT INTO BATCH_TRAINER VALUES('BT04','444444444'); INSERT INTO BATCH_TRAINER VALUES('BT05','444444445');
CLERK	INSERT INTO CLERK VALUES(222222221, 200); INSERT INTO CLERK VALUES(222222222, 300); INSERT INTO CLERK VALUES(222222223, 250); INSERT INTO CLERK VALUES(222222224, 400); INSERT INTO CLERK VALUES(222222225, 300);
TRAINER	INSERT INTO TRAINER VALUES(444444441, 20); INSERT INTO TRAINER VALUES(444444442, 22); INSERT INTO TRAINER VALUES(444444443, 24); INSERT INTO TRAINER VALUES(444444444, 28); INSERT INTO TRAINER VALUES(444444445, 30);
MANAGERS	INSERT INTO MANAGERS VALUES(333333331, 2000); INSERT INTO MANAGERS VALUES(333333332, 2500); INSERT INTO MANAGERS VALUES(333333333, 3000); INSERT INTO MANAGERS VALUES(333333334, 2200); INSERT INTO MANAGERS VALUES(333333335, 2000);
BRANCH_FACILITIES	INSERT INTO BRANCH_FACILITIES VALUES('BR01', 'F01'); INSERT INTO BRANCH_FACILITIES VALUES('BR02', 'F02');

	INSERT INTO BRANCH_FACILITIES VALUES('BR03', 'F03'); INSERT INTO BRANCH_FACILITIES VALUES('BR04', 'F04'); INSERT INTO BRANCH_FACILITIES VALUES('BR05', 'F05');
CUSTOMER_REPORT	INSERT INTO CUSTOMER_REPORT VALUES('R01', 23, 78, 5, 10, 12, 'C01', 222222221); INSERT INTO CUSTOMER_REPORT VALUES('R02', 23, 88, 5, 10, 12, 'C01', 222222222); INSERT INTO CUSTOMER_REPORT VALUES('R03', 23, 73, 5, 10, 12, 'C01', 222222223); INSERT INTO CUSTOMER_REPORT VALUES('R04', 23, 78, 5, 10, 12, 'C01', 222222221); INSERT INTO CUSTOMER_REPORT VALUES('R05', 23, 78, 5, 10, 12, 'C01', 222222221);
EQUIPMENT	INSERT INTO EQUIPMENT VALUES('EP01','CARDIO', 'GOOD', 'ALLEN', 'TREADMILL', 'BR01',222222221); INSERT INTO EQUIPMENT VALUES('EP02','CARDIO', 'EXCELLENT', 'ALLEN', 'TREADMILL', 'BR01',222222221); INSERT INTO EQUIPMENT VALUES('EP03','CARDIO', 'IN REPAIR', 'ALLEN', 'TREADMILL', 'BR01',222222222); INSERT INTO EQUIPMENT VALUES('EP04','CYCLING', 'GOOD', 'ALLEN', 'TREADMILL', 'BR01',222222223); INSERT INTO EQUIPMENT VALUES('EP05','DUMBLES', 'GOOD', 'ALLEN', 'TREADMILL', 'BR01',222222224);
F_EMPLOYEE	INSERT INTO F_EMPLOYEE VALUES(333333331, 'THOMAS', 'MCCALLUM', 9999999999, 'BR01'); INSERT INTO F_EMPLOYEE VALUES(222222221, 'TOMMY', 'IRVING', '9999999999', 'BR01'); INSERT INTO F_EMPLOYEE VALUES(444444441, 'DUKES', 'ARLINGTON', 9999999999', 'BR01'); INSERT INTO F_EMPLOYEE VALUES(333333331, 'ALEX', 'NEWYORK', '9999999999', 'BR01'); INSERT INTO F_EMPLOYEE VALUES(333333331, 'CHAN', 'NEW CAMPBELL', '9999999999', 'BR01'); INSERT INTO F_EMPLOYEE VALUES(333333334,'ALEX', 'NEW CAMPBELL', '9999999969', 'BR04'); INSERT INTO F_EMPLOYEE VALUES(333333335,'CHUNG', 'NEW CAMPBELL', '9999999959', 'BR05');

d) PL/SQL:

1. Procedure to generate a report of Equipments for particular Branch ID
IF condition is set to ALL, then ALL the EQUIPMENTS of that particular BRANCH will be displayed
ELSE we can filter on specific EQUIPMENT condition like 'Went for Repair', 'Working', etc

```
CREATE OR REPLACE PROCEDURE EQ_STATUS_CHECK (BRANCHID IN
EQUIPMENT.BRANCH_ID%TYPE, EQCOND IN EQUIPMENT.E_CONDITION%TYPE) AS
BEGIN
DECLARE
  B_ID EQUIPMENT.BRANCH_ID%TYPE;
  E_COND EQUIPMENT.E_CONDITION%TYPE;
  EQDESC EQUIPMENT%ROWTYPE;
  CURSOR EQUIPMENT_DETAILS IS SELECT * FROM EQUIPMENT WHERE
BRANCH_ID = BRANCHID;
  BEGIN
    OPEN EQUIPMENT_DETAILS;
    DBMS_OUTPUT.PUT_LINE('EQUIPMENT DETAILS FOR BRANCH '
||BRANCHID);
      LOOP
        FETCH EQUIPMENT_DETAILS INTO EQDESC;
        EXIT WHEN (EQUIPMENT_DETAILS%NOTFOUND);
        IF(EQCOND = 'ALL') THEN
          DBMS_OUTPUT.PUT_LINE(' '||EQDESC.E_ID||'
'||EQDESC.E_NAME||' '||EQDESC.MANUFACTURER||' '||EQDESC.CLERK_SSN);
        ELSE
          IF(EQCOND = EQDESC.E_CONDITION) THEN
                DBMS_OUTPUT.PUT_LINE(' '||EQDESC.E_ID||'
'||EQDESC.E_NAME||' '||EQDESC.MANUFACTURER||' '||EQDESC.CLERK_SSN);
          END IF;
        END IF;

      END LOOP;
    CLOSE EQUIPMENT_DETAILS;
  END;
END EQ_STATUS_CHECK;


SET SERVEROUTPUT ON;
BEGIN
EQ_STATUS_CHECK('BR01','IN REPAIR');
END;
```

2. PL/SQL Query to raise the hourly salary of Trainer by 20%

```
CREATE OR REPLACE PROCEDURE INCREMENT_SALARY AS
BEGIN
DECLARE
THISEMP TRAINER%ROWTYPE;
CURSOR C IS
SELECT * FROM TRAINER
FOR UPDATE;
BEGIN
OPEN C;
LOOP
FETCH C INTO THISEMP;
EXIT WHEN C%NOTFOUND;
UPDATE TRAINER SET HOURLY_SALARY=THISEMP.HOURLY_SALARY +
0.20*THISEMP.HOURLY_SALARY WHERE CURRENT OF C;
END LOOP;
CLOSE C;
END;
END INCREMENT_SALARY;
```

3. Retrieve number of Customers IN each Batch

```
CREATE OR REPLACE PROCEDURE BT_STATUS_CHECK (BID IN
CUSTOMER.BATCH_ID%TYPE,COUNTS OUT NUMBER)
AS
BEGIN
BEGIN
  SELECT COUNT(*)  INTO  COUNTS FROM CUSTOMER WHERE BATCH_ID= BID;
END;
END BT_STATUS_CHECK;
```

4. Describe Person as 'Not Healthy', 'Healthy' by Fat Percentage (If fat percentage less than 15, healthy or else not healthy.

```
CREATE OR REPLACE PROCEDURE EQ_PERSON_HEALTH_CHECK (CUSTID IN
CUSTOMER.C_ID%TYPE)AS
BEGIN
DECLARE
  FAT_P CUSTOMER_REPORT.FAT_PERCENTAGE%TYPE;
  CURSOR REPORT_DETAILS IS SELECT FAT_PERCENTAGE FROM
CUSTOMER_REPORT WHERE C_ID = CUSTID;
  BEGIN
    OPEN REPORT_DETAILS;
        LOOP
```

```
          FETCH REPORT_DETAILS INTO FAT_P;
          EXIT WHEN (REPORT_DETAILS%NOTFOUND);
          IF FAT_P < 15 THEN
              DBMS_OUTPUT.PUT_LINE('HEALTHY');
          ELSE
              DBMS_OUTPUT.PUT_LINE('NOT HEALTHY');
          END IF;
        END LOOP;
      CLOSE REPORT_DETAILS;
    END;
  END EQ_PERSON_HEALTH_CHECK;
```