Database Design Project

CMPT 306

Nicole Rae

Table of Contents

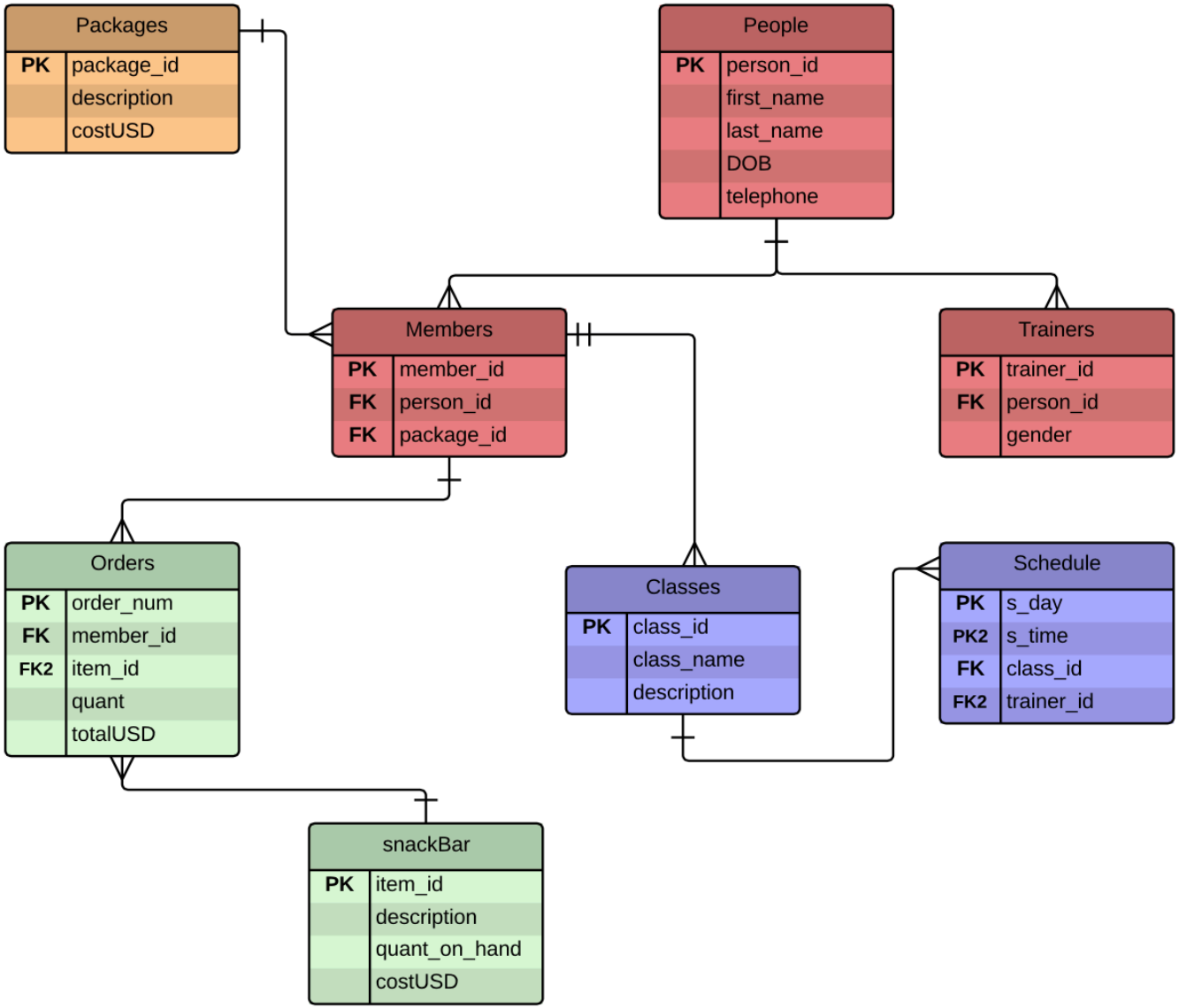
EXECUTIVE SUMMARY **3**
ENTITY – RELATIONSHIP DIAGRAM **4**
CREATE TABLES
PACKAGES 5
PEOPLE 6
MEMBERS 7
TRAINERS 8
CLASSES 9
SCHEDULE 10
SNACK BAR 11
ORDERS 12
VIEWS **13**
REPORTS **15**
SECURITY **17**
KNOWN PROBLEMS **18**
FUTURE ENHANCEMENTS **19**

Executive Summary

This database design was created with the needs and demands of Crunch Fitness in mind to enable the business to run swiftly and smoothly. Due to the many aspects that make up the gym as a whole, it is critical to have a database system constructed to store the data of everyday business activities. One of the objectives is to keep track of both the several different types of memberships offered, along with payment and contact information of new and current members.

Crunch holds numerous exercise classes taught by various personal trainers. This database will allow for the weekly scheduling of the date, time, and instructor for each class. Lastly, is the tracking of the daily transactions made within the fitness center's snack bar service, providing beverages, snacks, and health supplements for their members. The purpose of this documentation is to outline each particular entity and its fields within the relational database created to meet the needs of Crunch Fitness.

Entity Relationship Diagram



Create Statements

Packages

```
create table Packages
(
        package_id      integer   not null,
        description     text      not null,
        costUSD         money     not null,
        primary key  (package_id)
);
```

Functional Dependencies

Package_Id → Description, CostUSD

Sample Data

package_id	description	costUSD
001	Regular	$65.00
002	Non-Peak	$75.50
003	Peak	$90.00

People

```
create table People
(
        person_id       integer   not null,
        first_name      text      not null,
        last_name       money     not null,
        DOB             date      not null,
        Telephone       char(15)  not null,
        primary key  (person_id)
);
```

Functional Dependencies

Person_Id → First_Name, Last_Name, DOB, Telephone

Sample Data

person_id	first_name	last_name	DOB	telephone
10100000	Megan	Stacklum	1992-03-28	821-121-2179
10100000	Sean	Maddie	1986-06-07	914-240-7779
10100000	Sharon	Holy	1988-10-21	901-923-0086
10100000	Chrissy	Higgle	1993-04-14	201-423-2463
10100000	Jimmy	Roberts	1992-02-28	392-219-2494
10100000	Jessie	Reespo	1990-01-02	845-316-4496

Members

```
create table Members
(
    member_id       integer       not null,
    person_id       integer       not null references People(person_id),
    package_id      integer       not null references Packages(package_id),
    primary key  (member_id)
);
```

Functional Dependencies

Member_Id → Person_Id, Package_Id

Sample Data

member_id	person_id	package_id
000001	10100000	002
000002	10100001	001
000003	10100014	002
000004	10100003	003
000005	10100004	002
000006	10100008	001
000007	10100006	001

Trainers

```
create table Trainers
(
    trainer_id      integer       not null,
    person_id       integer       not null references People(person_id),
,   gender          char(8)       not null,
    constraint   check_gender  check (gender = 'Male' or gender = 'Female'),
    primary key  (trainer_id)
);
```

Functional Dependencies

Trainer_Id → Person_Id, Gender

Sample Data

trainer_id	person_id	gender
00	10100005	Female
01	10100011	Male
02	10100012	Female
03	10100018	Male
04	10100023	Female

Classes

```
create table Classes
(
    class_id        integer       not null,
    class_name      text          not null,
    description     text          not null,
    primary key  (class_id)
);
```

Functional Dependencies

Class_Id → Class_Name, Description

Sample Data

class_id	class_name	description
550	Overdrive	high intensity interval training
551	Cardio Tai Box	tai box with upbeat music
552	Tread & Shed	treadmill cardio to met away calories
553	Absolution	core strengthening solution
554	Beach Body	total body conditioning

Schedule

```
create table Schedule
(
    s_day        date      not null,
    s_time       time      not null,
    class_id     integer   not null references Classes  (class_id),
    trainer_id   integer   not null references Trainers (trainer_id),
    primary key  (s_day, s_time)
);
```

Functional Dependencies

S_Day, S_Time → Class_Id, Trainer_Id

Sample Data

s_day	s_time	class_id	trainer_id
2015-04-12	07:00:00	550	2
2015-04-12	12:30:00	557	0
2015-04-12	15:30:00	559	3

s_day	s_time	class_id	trainer_id
2015-04-13	15:30:00	553	2
2015-04-13	18:00:00	551	5
2015-04-13	20:00:00	555	3

Snack Bar

```
create table snackBar
(
    item_id         date      not null,
    description     text      not null,
    quant_on_hand   integer   not null,
    costUSD         money     not null,
    primary key (item_id)
);
```

Functional Dependencies

Item_ID → Description, Quant_on_Hand, CostUSD

Sample Data

item_id	description	quant_on_hand	costUSD
1	small water	500	$1.00
2	large water	500	$2.00
3	gatorade	450	$3.00
4	powerade	300	$2.50
5	power bar	250	$4.00
6	peanuts	200	$3.50
7	protein shake mix	300	$6.50

Orders

```
create table Orders
(
    order_num   date      not null,
    person_id   text      not null references People  (person_id),
    item_id     integer   not null references snackBar (item_id),
    quant       integer   not null,
    totalUSD    money     not null,
    primary key (order_num)
);
```

Functional Dependencies

Order_Num → Person_Id, Item_Id, Quant, TotalUSD

Sample Data

order_num	person_id	item_id	quant	totalUSD
111111	10100004	1	2	$2.00
111112	10100001	6	1	$3.50
111113	10100018	13	1	$24.00
111114	10100013	2	1	$2.00
111115	10100005	5	1	$4.00
111116	10100022	20	1	$60.00
111117	10100017	9	2	$35.00

Views

Members Overview

```
create view memberOverview AS
    select first_name, last_name, members.person_id,
    member_id as memberNumber, telephone
    from people
    inner join members
    on people.person_id = members.person_id;
```

Sample Data

first_name	last_name	person_id	memberNumber	telephone
Megan	Stacklum	10100000	1	821-121-2179
Sean	Maddie	10100001	2	914-249-7779
Sharon	Holy	10100002	19	901-923-0086
Chrissy	Higgle	10100000	4	201-423-2463
Jimmy	Roberts	10100000	5	392-219-4292
Hannah	Petron	10100000	7	845-743-2144
Annie	Morris	10100000	8	891-309-2302

Trainers Overview

```
create view trainerOverview AS
    select first_name, last_name, trainers.person_id,
    trainer_id as trainerumber, telephone
    from people
    inner join trainers
    on people.person_id = trainers.person_id;
```

Sample Data

first_name	last_name	person_id	trainerNumber	telephone
Jessie	Reespo	10100005	0	845-316-4496
Nicholas	Diccilo	10100005	1	845-293-1034
Victoria	Panico	10100005	2	832-491-0234
Joey	Kelly	10100005	3	914-754-2168
Kevin	Geeder	10100005	4	201-677-3268
Sally	Badger	10100005	5	345-666-4326

Reports

Daily Class Schedule

This report will display the exercise classes and the times that they are being held for a specified date.

```
select s_day as 'Day',
       s_time as 'Start Time',
       class_name as 'Class'
from schedule
join classes
on schedule.s_day = '2015-04-13'
limit 5;
```

Sample Data

Day	Start Time	Class
2015-04-13	07:00	Beach Body
2015-04-13	12:30	Boom Move It
2015-04-13	15:30	Absolution
2015-04-13	18:00	Cardio Tai Box
2015-04-13	20:00	Retro-Robics

Current Inventory

This specific report would typically be used on a weekly basis. Its purpose would be allowing the fitness center to keep track of the current inventory of the Snack Bar. Additionally, popular purchased items would be determined by viewing the amount of orders that have been placed for that specific item.

```
select description      as 'Item',
       quant_on_hand    as 'Quantity on Hand',
       count(order_num) as 'Number of Orders'
from snackBar
inner join Orders
on snackBar.item_id = Orders.item_id;
```

Sample Data

Item	Quantity on Hand	Number of Orders
2	500	1
16	250	2
20	200	2

Security

The level of security within the database is determined by the roles created and the privileges each are given.

Admin

```
create role Admin
    grant all on all tables
    in schema public
    to Admin;
```

Staff

```
create role Staff
    grant select, insert on all tables
    in schema public
    to Staff;
```

Known Problems

In addition to the athletic trainers that work at Crunch, there are also numerous employees that are responsible for the day-to-day care of the fitness center. It is obvious, however, that not each and every staff member can be working at the same time. Therefore, it is realistic that the company would have a way to plan on a monthly basis the scheduled shifts for each employee along with the amount of hours they worked, pay rate, etc.

When a member is at the gym and they need something quick and nutritious to eat, the chances are high that they will approach the Snack Bar and purchase a protein bar. It is also very likely that with a protein bar, some type of beverage would be expected. In my current database design, a member is certainly able to purchase as many items as they wish, however, for every unique item that exists a new order would be placed. Therefore, it would be more logical to allow for multiple items to make up one order number.

Future Enhancements

Due to the limited amount of spots available for each exercise class offered, a future enhancement would be a way to keep track of the current spots available for all classes. If a particular class reaches its limit, no additional members would be able to sign up.

Although the trainers at Crunch Fitness are the instructors for the classes offered, they also hold one-on-one sessions with the members to allow for a more personalized work out. Another future enhancement would be the ability to schedule each of these personal sessions and update the trainer's availability. In keeping track of the member's sessions with their trainers, we would have the possibility to implement a reward system. By showing the clients that with hard work comes rewards, a more positive and motivated individual will be the desirable outcome