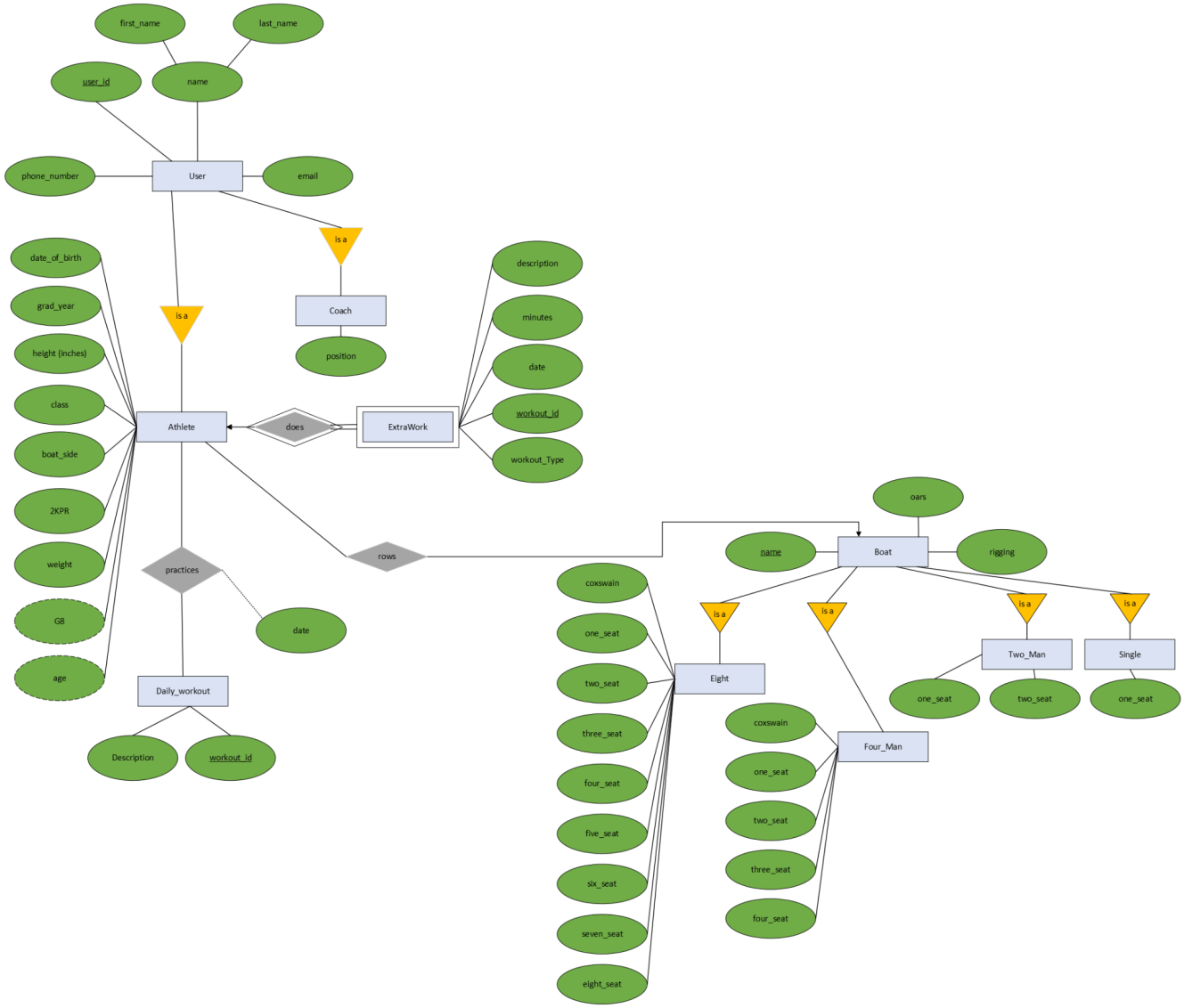
CS 4750 Milestone 1: DB design

Yusuf Cetin (yc5xw), Matthew Remigino (mr3ea), Maajid Husain (mah2ksc), Alex Becker (amb3yn)

Part 1: Develop E-R diagram

Our ER diagram comprises 10 tables describing users, boat types, extra workouts, and their relationships. Coaches and athletes are subclasses of User. Our four boat types are each subclasses of Boat. Each ExtraWork must exist with an Athlete. Athletes have two calculated attributes, G8 and age. G8 is a weight adjusted 2k score and age will be calculated from date of birth. An athlete can row in only one boat at a time, and a boat can have many athletes in it. A Daily workout is connected to athletes via a “practice relation”. Over the course of a season an athlete will complete many daily workouts, and each daily workout will have many athletes complete it in the form of practice.

See next page for ER diagram


first_name
last_name
user_id
name
phone_number
User
email
Is a
date_of_birth
grad_year
Is a
Coach
description
minutes
height (inches)
position
class
Athlete
does
ExtraWork
date
workout_id
boat_side
workout_Type
2KPR
weight
practices
rows
oars
name
Boat
rigging
coxswain
Is a
Is a
Is a
Is a
G8
date
one_seat
two_seat
Eight
Two_Man
Single
age
Daily_workout
three_seat
four_seat
five_seat
six_seat
seven_seat
eight_seat
one_seat
two_seat
one_seat
coxswain
Four_Man
one_seat
two_seat
three_seat
four_seat
Description
workout_id

Part 2: Convert E-R diagram into tables

1. Athlete(user_id, first_name, last_name, email, phone_number, date_of_birth, grad_year, height, class, boat_side, 2KPR, weight, g8, age)
2. Coach(user_id, first_name, last_name, email, phone_number, position)
3. ExtraWork(athlete_user_id, workout_id, minutes, date, workout_type, description)
4. Single(name, oars, rigging, one_seat)
5. TwoMan(name, oars, rigging, one_seat, two_seat)
6. FourMan(name, oars, rigging, coxswain, one_seat, two_seat, three_seat, four_seat)
7. Eight(name, oars, rigging, coxswain, one_seat, two_seat, three_seat, four_seat, five_seat, six_seat, seven_seat, eight_seat)
8. DailyWorkout(workout_ID, description)
9. RowsIn(athlete_user_id, boat_name)
10. Practices(athlete_user_id, daily_workout_ID, date)

Part 3: Decompose tables using 3NF or BCNF

All tables are decomposed using 3NF

1. Athlete

Table Keys:

```
user_id: A
first_name: B
last_name: C
email: D
phone_number: E
date_of_birth: F
grad_year: G
height: I
class: J
boat_side: K
2KPR: L
weight: M
g8: N
age: O
```

FDs:

```
{ A → BCDEFGHIJKLMNO,
  LM → N,
  F → O }
```

Calculate F_c

1. Copy FD	2. No reflexivity	3. Remove extraneous attributes
A → BCDEFGHIJKLMNO LM → N F → O		A → BCDEFGHIJKLMNO LM → N F → O

Tables:

ABCDEFGHIJKLMNO // LMN // FO

AthleteData // ErgData // AgeData

Prove 3NF:

1. Each column is atomic and flat
2. No partial dependency
3. Dependency preserving:
    a. A → BCDEFGHIJKLMNO is satisfied by ABCDEFGHIJKLMNO
    b. LM → N is satisfied by LMN
    c. F → O is satisfied by FO
4. Lossless Join

a. $ABCDEFGHIJKLMNO \cap LMN = LMN \,\&\, LM\ is\ SK\ of\ LMN$
b. $ABCDEFGHIJKLMNO \cap FO = FO \,\&\, F\ is\ SK\ of\ FO$

2. Coach

Table Keys:

```
user_id: A
first_name: B
last_name: C
email: D
phone_number: E
position: F
```

FDs:

```
{ A → BCDEF }
```

Calculate F_c:

1. Copy FD `A → BCDEF`	2. No reflexivity	3. No extraneous attributes to remove

Tables:

ABCDEF

Coach

Prove 3NF:

1. Each column is atomic and flat
2. No partial dependency
3. The LHS of the FD in the relation, A, is a SK

3. ExtraWork

Table key:

```
athlete_user_id: A
workout_id: B
minutes: C
date: D
workout_type: E
description: F
```

FDs:

```
{ AB → CDEF }
```

Calculate F_c:

1. Copy FD `AB → CDEF`	2. No reflexivity	3. No extraneous attributes to remove

Tables:
ABCDEF
ExtraWork

Prove 3NF:

1. Each column is atomic and flat
2. No partial dependency
3. The LHS of the FD in the relation, A, is a SK

4. Single

Table Keys:

```
name: A
oars: B
rigging: C
one_seat: D
```

FDs:

```
{ A → BCD}
```

Calculate F_c:

1. Copy FD `A → BCD`	2. No reflexivity	3. No extraneous attributes to remove

Tables:
ABCD
Single

Prove 3NF:

1. Each column is atomic and flat
2. No partial dependency
3. The LHS of the FD in the relation, A, is a SK

5. TwoMan

Table Keys:

```
name: A
oars: B
rigging: C
one_seat: D
```

`two_seat: E`

FDs:
`{ A → BCDE}`

Calculate F_c:

1. Copy FD `A → BCDE`	2. No reflexivity	3. No extraneous attributes to remove

Tables:
ABCDE
TwoMan

Prove 3NF:
1. Each column is atomic and flat
2. No partial dependency
3. The LHS of the FD in the relation, A, is a SK

6. FourMan

Table Keys:
`name: A`
`oars: B`
`rigging: C`
`coxswain: D`
`one_seat: E`
`two_seat: F`
`three_seat: G`
`four_seat: H`

FDs:
`{ A → BCDEFG}`

Calculate F_c:

1. Copy FD `A → BCDEFGH`	2. No reflexivity	3. No extraneous attributes to remove

Tables:
ABCDEFGH
FourMan

Prove 3NF:
1. Each column is atomic and flat

2. No partial dependency
3. The LHS of the FD in the relation, A, is a SK

7. Eight

Table Keys:

```
name: A
oars: B
rigging: C
coxswain: D
one_seat: E
two_seat: F
three_seat: G
four_seat: H
five_seat: I
six_seat: j
seven_seat: k
eight_seat: L
```

FDs:

```
{ A → BCDEFGHIJ}
```

Calculate F_c:

1. Copy FD `A → BCDEFGHIJ`	2. No reflexivity	3. No extraneous attributes to remove

Tables

ABCDEFGHIJ

TwoMan

Prove 3NF:

1. Each column is atomic and flat
2. No partial dependency
3. The LHS of the FD in the relation, A, is a SK

8. DailyWorkout

Table Keys:

```
workout_id: A
description: B
```

FDs:

```
{ A → B}
```

Calculate F_c:

1. Copy FD `A → B`	2. No reflexivity	3. No extraneous attributes to remove

Tables:
AB
DailyWorkout

Prove 3NF:

1. Each column is atomic and flat
2. No partial dependency
3. The LHS of the FD in the relation, A, is a SK

9. RowsIn

Table Keys:

```
athlete_user_id: A
boat_name: B
```

FDs:

```
{ A → B}
```

Calculate F_c:

1. Copy FD `A → B`	2. No reflexivity	3. No extraneous attributes to remove

Tables:
AB
RowsIn

Prove 3NF:

1. Each column is atomic and flat
2. No partial dependency
3. The LHS of the FD in the relation, A, is a SK

10. Practices

Table Keys:

```
athlete_user_id: A
daily_workout_id: B
date: C
```

FDs:
{ AB → C}

Calculate F_c:

1. Copy FD AB→ C	2. No reflexivity	3. No extraneous attributes to remove

Tables:
ABC
Practices

Prove 3NF:

1. Each column is atomic and flat
2. No partial dependency
3. The LHS of the FD in the relation, AB, is a SK

Post normalization schema:

1. AthleteData(user_id, first_name, last_name, email, phone_number, date_of_birth, grad_year, height, class, boat_side, 2KPR, weight, g8, age)
2. ErgData(2KPR, weight, g8)
3. AgeData(date_of_birth, age)
4. Coach(user_id, first_name, last_name, email, phone_number, position)
5. ExtraWork(athlete_user_id, workout_id, minutes, date, workout_type, description)
6. Single(name, oars, rigging, one_seat)
7. TwoMan(name, oars, rigging, one_seat, two_seat)
8. FourMan(name, oars, rigging, coxswain, one_seat, two_seat, three_seat, four_seat)
9. Eight(name, oars, rigging, coxswain, one_seat, two_seat, three_seat, four_seat, five_seat, six_seat, seven_seat, eight_seat)
10. DailyWorkout(workout_ID, description)
11. RowsIn(athlete_user_id, boat_name)
12. Practices(athlete_user_id, daily_workout_ID, date)

Contact Info

Alex Becker, amb3yn@virginia.edu
Yusuf Cetin, yc5xw@virginia.edu
Maajid Husain, mah2ksc@virginia.edu
Matt Remigino, mr3ea@virginia.edu