Health and Fitness Club Management System
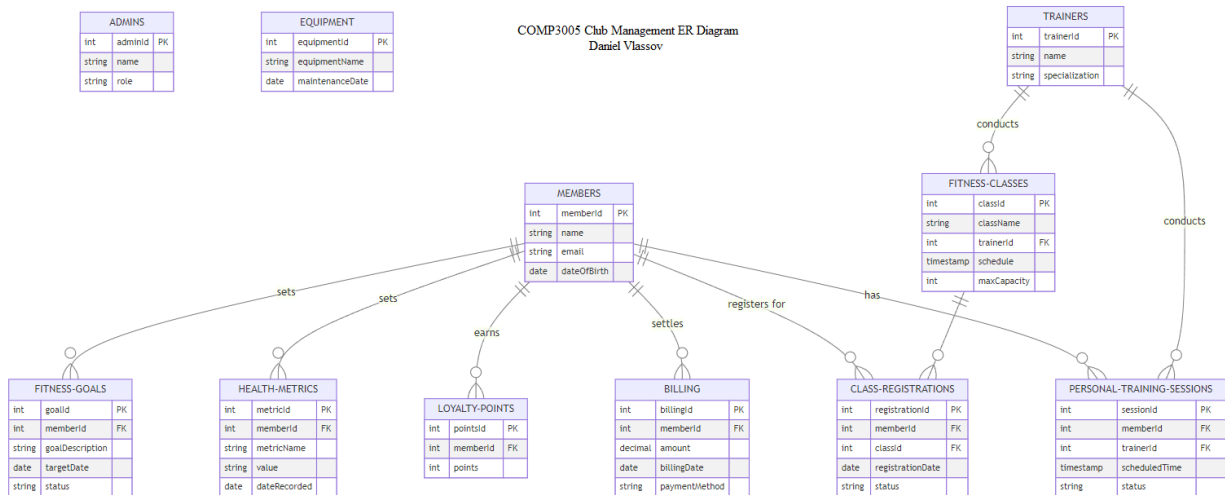
Daniel Vlassov 101222428

Carleton University

COMP3005 - Database Management Systems

Conceptual Design

ER Diagram (ER_Final.png):

COMP3005 Club Management ER Diagram
Daniel Vlassov

ADMINS

int	adminId	PK
string	name	
string	role	

EQUIPMENT

int	equipmentId	PK
string	equipmentName	
date	maintenanceDate	

TRAINERS

int	trainerId	PK
string	name	
string	specialization	

MEMBERS

int	memberId	PK
string	name	
string	email	
date	dateOfBirth	

FITNESS-CLASSES

int	classId	PK
string	className	
int	trainerId	FK
timestamp	schedule	
int	maxCapacity	

FITNESS-GOALS

int	goalId	PK
int	memberId	FK
string	goalDescription	
date	targetDate	
string	status	

HEALTH-METRICS

int	metricId	PK
int	memberId	FK
string	metricName	
string	value	
date	dateRecorded	

LOYALTY-POINTS

int	pointsId	PK
int	memberId	FK
int	points	

BILLING

int	billingId	PK
int	memberId	FK
decimal	amount	
date	billingDate	
string	paymentMethod	

CLASS-REGISTRATIONS

int	registrationId	PK
int	memberId	FK
int	classId	FK
date	registrationDate	
string	status	

PERSONAL-TRAINING-SESSIONS

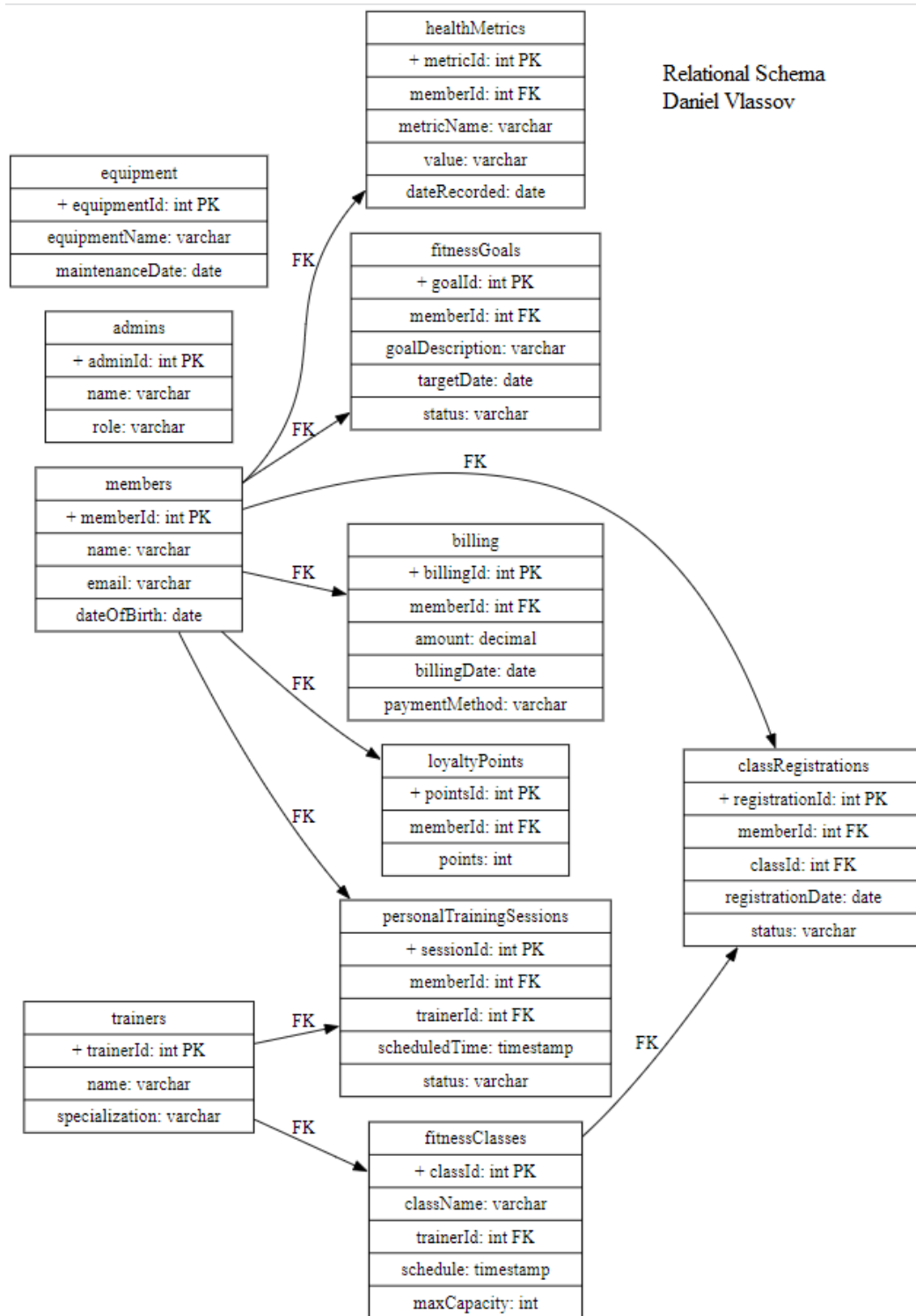| int	sessionId	PK
int	memberId	FK
int	trainerId	FK
timestamp	scheduledTime	
string	status	

Specifications:

Members should be able to register and manage their profiles, setting personal fitness goals and inputting health metrics. The platform will also enable members to schedule, reschedule, or cancel personal training sessions with certified trainers. Members can register for group fitness classes, workshops, and other events.
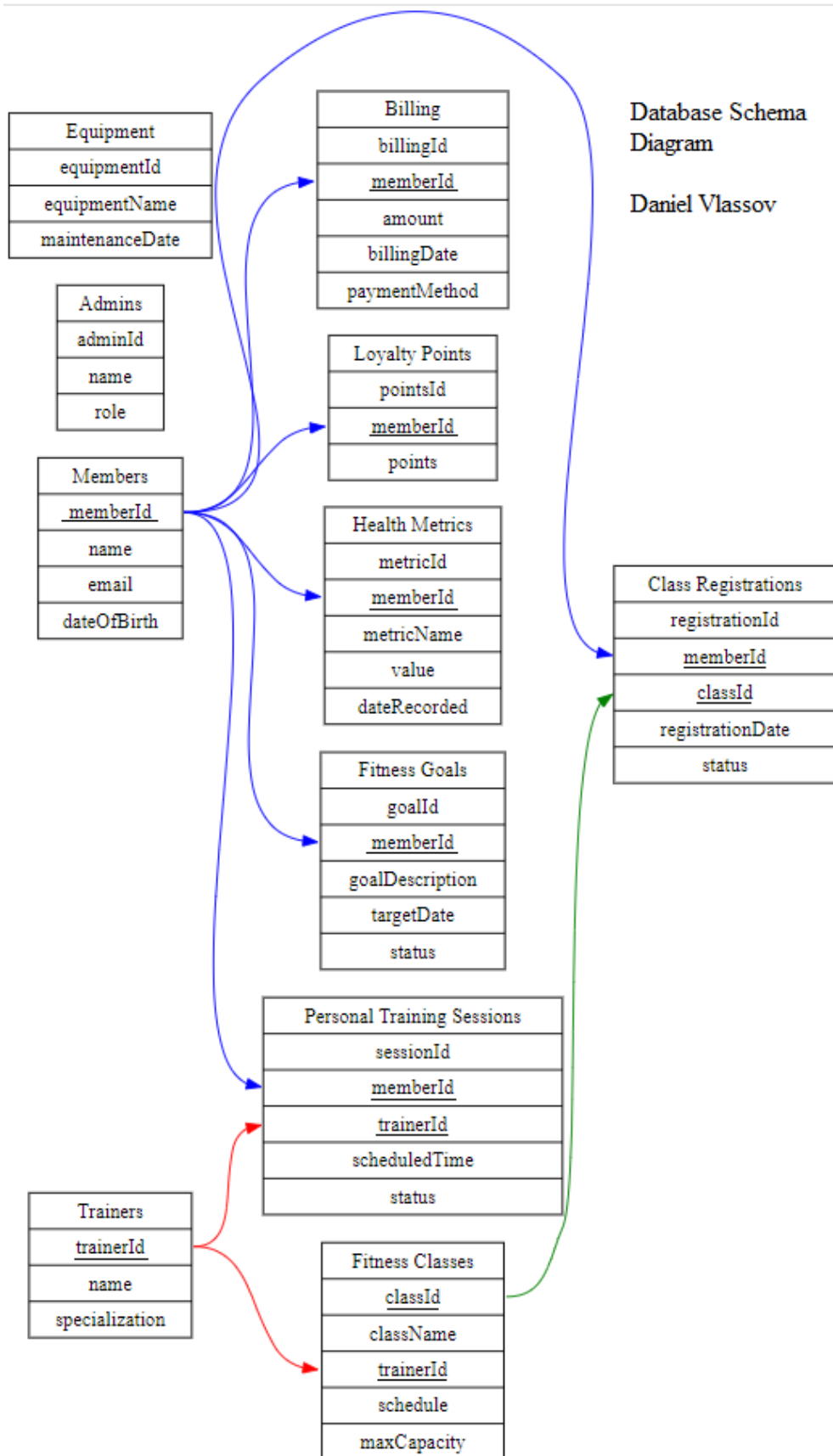
Assumptions:

- Each member can set multiple fitness goals and health metrics

- Trainers are associated with fitness classes and personal training sessions

- Loyalty points are earned through billing transactions

- Class registrations are linked to members and specific fitness classes

- Personal training sessions are scheduled between a member and a trainer

Reduction & Normalization of Schema (to 3NF)

Reducing to schemas from ER: (diagrams/Relational_Schema.png)

Equipment

equipmentId
equipmentName
maintenanceDate

Billing

billingId
memberId
amount
billingDate
paymentMethod

Database Schema
Diagram

Daniel Vlassov

Admins

adminId
name
role

Loyalty Points

pointsId
memberId
points

Members

memberId
name
email
dateOfBirth

Health Metrics

metricId
memberId
metricName
value
dateRecorded

Class Registrations

registrationId
memberId
classId
registrationDate
status

Fitness Goals

goalId
memberId
goalDescription
targetDate
status

Personal Training Sessions

sessionId
memberId
trainerId
scheduledTime
status

Trainers

trainerId
name
specialization

Fitness Classes

classId
className
trainerId
schedule
maxCapacity

(diagrams/Database_Schema_Diagram.png) ^^

Normalization and functional dependencies:

Members:

memberId -> {name, email, dateOfBirth}

1NF: only atomic values

2NF: memberId is the only key, attributes are fully functionally dependent on it

3NF: non-key attributes do not depend on each other.

Trainers:

trainerId -> {name, specialization}

1NF: Atomic

2NF: Complete dependency on trainerId.

3NF: No attribute is dependent on another non-prime attribute.

FitnessClasses:

classId -> {className, trainerId, schedule, maxCapacity}

1NF: Like above, passes.

2NF: Like above, passes.

3NF: Like above, passes.

PersonalTrainingSessions:

sessionId -> {memberId, trainerId, scheduledTime, status}

1NF: Like above, passes.

2NF: Like above, passes.

3NF: scheduledTime and status are not dependent on memberId or trainerId (only sessionId) so this example with multiple FK's passes as no transitive dependencies.

ClassRegistrations:

registrationId -> {memberId, classId, registrationDate, status}

1NF: Like above, passes.

2NF: Like above, passes.

3NF: Like above, passes.

Equipment:

equipmentId -> {equipmentName, maintenanceDate}

1NF: Like above, passes.

2NF: Like above, passes.

3NF: Like above, passes.

Billing:

billingId -> {memberId, amount, billingDate, paymentMethod}

1NF: Like above, passes.

2NF: Like above, passes.

3NF: Like above, passes.

LoyaltyPoints:

pointsId -> {memberId, points}

1NF: Like above, passes.

2NF: Like above, passes.

3NF: Like above, passes.

HealthMetrics:

metricId -> {memberId, metricName, value, dateRecorded}

1NF: Like above, passes.

2NF: Like above, passes.

3NF: Like above, passes.

FitnessGoals:

goalId -> {memberId, goalDescription, targetDate, status}

1NF: Like above, passes.

2NF: Like above, passes.

3NF: Like above, passes.

Proven we have the Database Schema normalized to 3NF

DDL DML and Queries are in the /sql folder which can be run through PgAdmin4