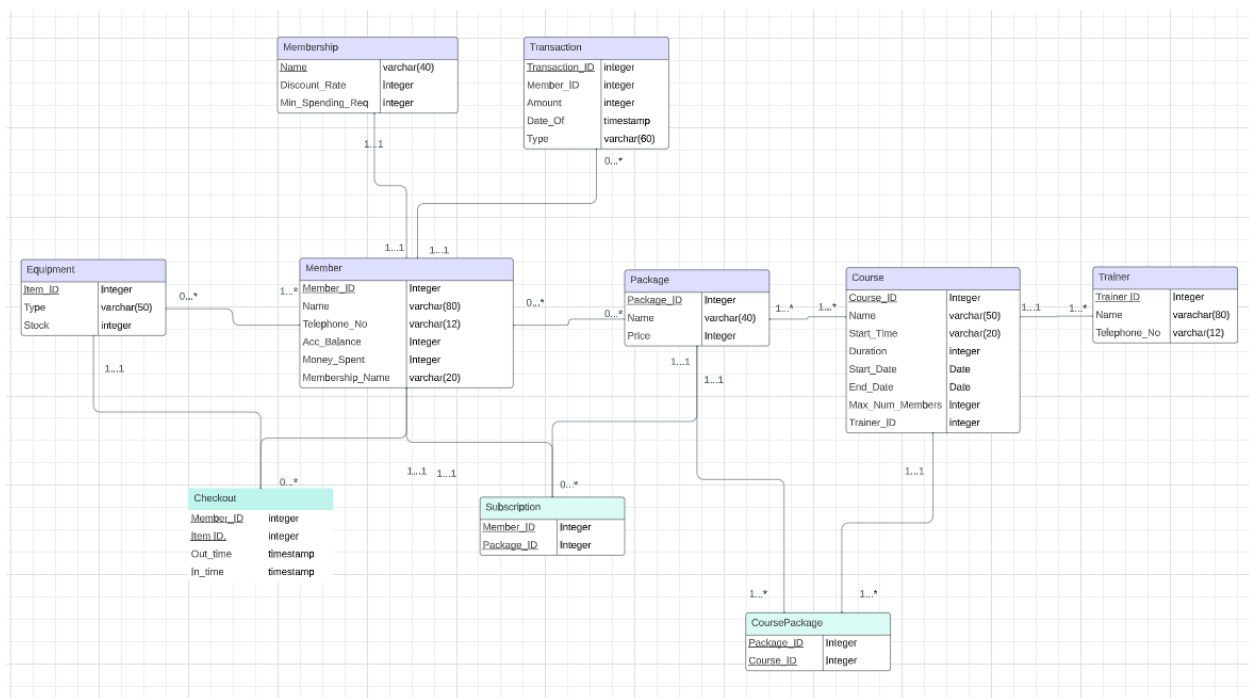Edan Uccetta, Jake Gridley, Lex Castaneda
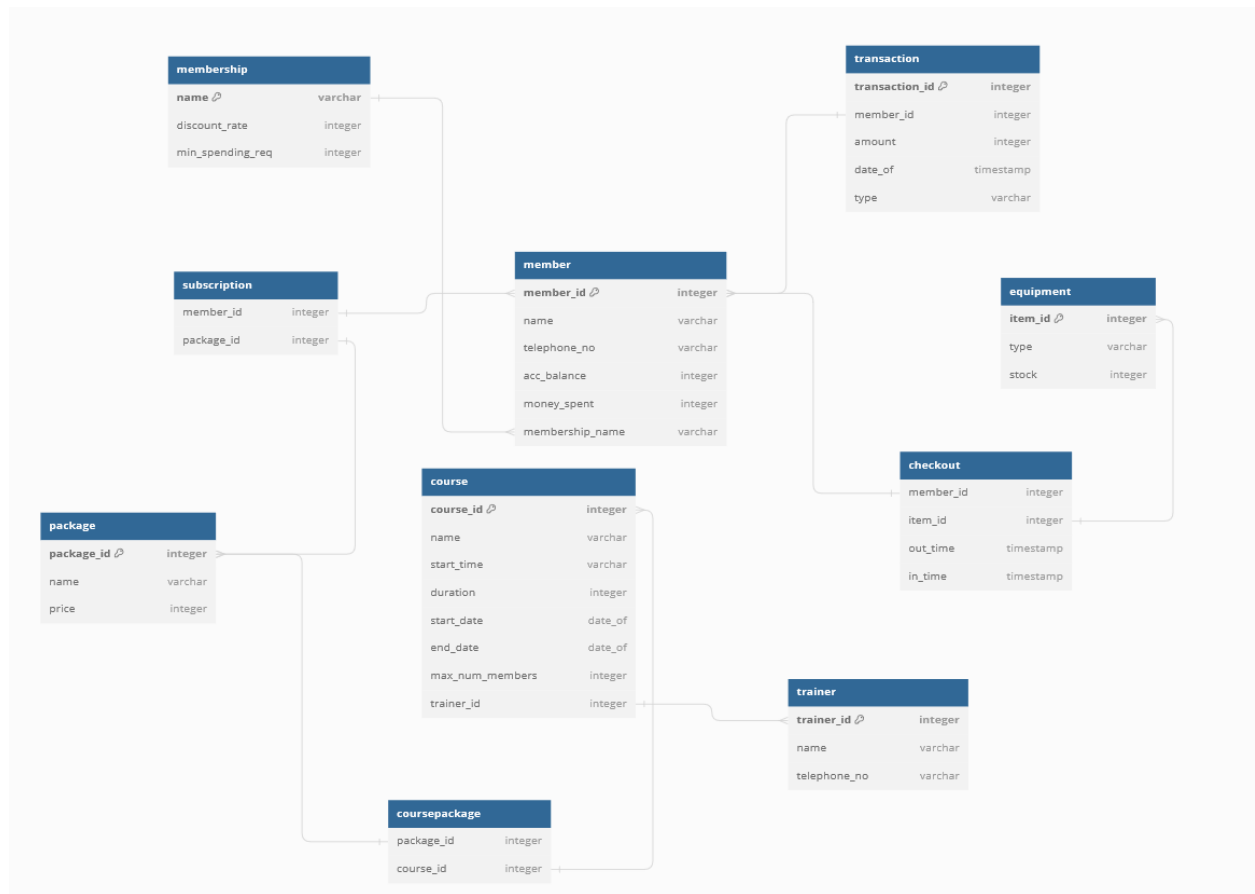
12/5/2023

CSC 460

Program 4 Design Specifications

1. Conceptual Database Design

We chose to implement most constraints in the Java side rather than the SQL side, so implementation of the tables was relatively straightforwards. The first attempt at an ER diagram didn't have Subscription, but we quickly ran into the necessity to link members to packages for querying and adding purposes, so it was added.

2. Logical Database Design

membership
name 🔑	varchar
discount_rate	integer
min_spending_req	integer

transaction
transaction_id 🔑	integer
member_id	integer
amount	integer
date_of	timestamp
type	varchar

subscription
| member_id | integer |
| package_id | integer |

member
member_id 🔑	integer
name	varchar
telephone_no	varchar
acc_balance	integer
money_spent	integer
membership_name	varchar

equipment
item_id 🔑	integer
type	varchar
stock	integer

checkout
member_id	integer
item_id	integer
out_time	timestamp
in_time	timestamp

package
package_id 🔑	integer
name	varchar
price	integer

course
course_id 🔑	integer
name	varchar
start_time	varchar
duration	integer
start_date	date_of
end_date	date_of
max_num_members	integer
trainer_id	integer

trainer
trainer_id 🔑	integer
name	varchar
telephone_no	varchar

coursepackage
| package_id | integer |
| course_id | integer |

3. Normalization Analysis

All our tables adhere to Third Normal Form. No attribute besides the primary key functionally determines another attribute in any table in any case.

4. Query Description

Our custom query joins nearly every table in the database to tally up the total amount a member, chosen by the program user, has spent on each individual trainer (I.E. the total of all transactions paying for courses staffed by that trainer). This query could be used in aggregate to see which trainers tend to succeed most, or so a member could use the program to better understand where their own money is going (and check their total spending over a duration).