CS 605.641 Principles of Database Systems
Midterm Examination Summer 2020
(Dates: 06/28/2020 – 07/01/2020; No Discussion, Collaboration, Consultation, Sharing)
Name: Brian Loughran

**Please answer all questions**

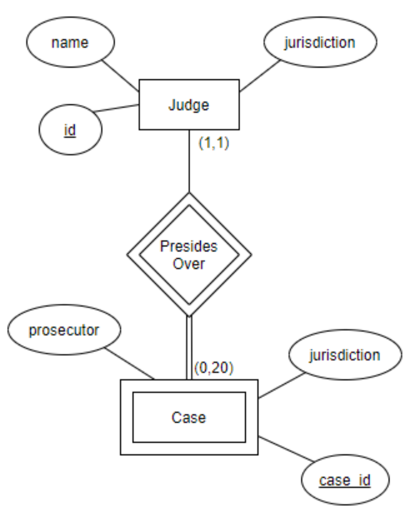
1. Answer the following database-related questions: (Total: 30 points, 5 points each)
   i) Explain what advantages NoSQL approaches have over traditional DBMS solutions. (3 points)

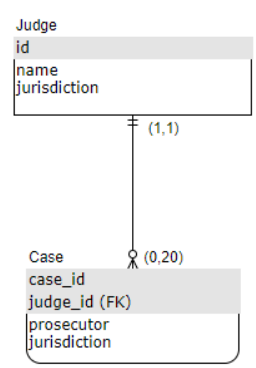
NoSQL approaches vary from traditional DBMS approaches such that they utilize a schema-less design to store data rather than traditional tables. NoSQL scales out horizontally in most cases better than DBMS, and is well suited for big data applications. Rather than following the traditional ACID rules of relational databases, NoSQL follows BASE (Basically Available, Soft state, Eventual consistency), which allow for different access patterns in scenarios where a different access pattern is needed.

Explain why NoSQL is not designed to replace relational DBMS solutions. (2 points)

NoSQL is designed as a complementary addition to relational DBMS solutions. There are use cases where it can work well such as some discussed above, but some of the tools in relational databases such as eliminating data anomalies at the database level, and doing relational joins at the database level, which almost always has better performance than at the application level. Relational databases also work wonderfully for a high percentage of data storage/access use cases, coupled with ACID rules provide a usable, safe, secure place to store data. Because of this, NoSQL is designed more as a complementary tool to relational DBMS, rather than a replacement.

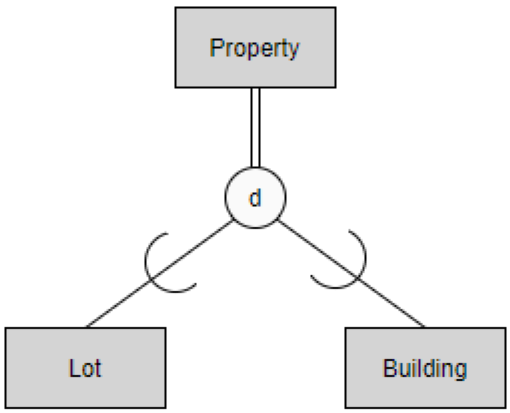
   ii) Draw an ERD that includes two entities: JUDGE with attributes (<u>id</u>, name, jurisdiction) and CASE with attributes (<u>case_id</u>, prosecutor, jurisdiction). The relationship between them is PRESIDES_OVER. A judge may preside over cases and can have a maximum of 20 cases; each case must be assigned to exactly one judge. Please **show cardinality ratios** using **both Chen's notation and IE notation** respectively. Note: You can just type text on two ERDs to show cardinality ratios. (5 points)



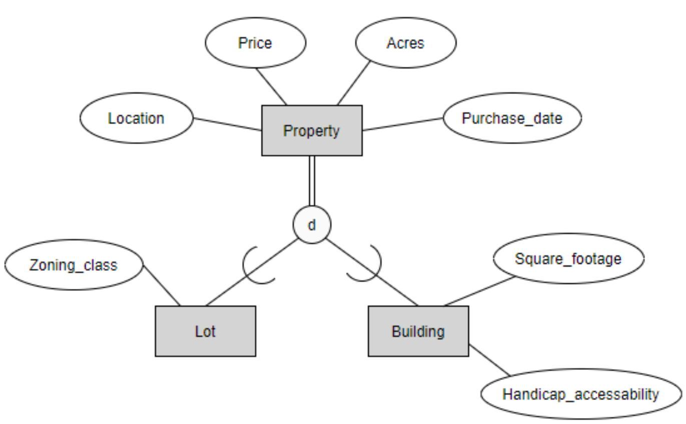
iii) Create an EERD using a total category for a PROPERTY entity type that may be a LOT or a BUILDING entity type. (2 Points)

Property may be either a Lot or Building. Property has a total constraint, specifying that it cannot be neither or both. This is achieved with the disjoint constraint and the total constraint. Note this EERD does not have attributes for each entity for simplicity sake.




In a generalization process if you have a LOT and a BUILDING, then how will you come up with your EERD? (Note: This is a different design) (2 Points). Please explain your design. (1 point)

To come up with a EERD for similar requirements using a generalization process we should identify common attributes between subclasses and attach those entities to the superclass. This is in contrast to specialization, where as many attributes as possible are stored in the superclass. For example for the Lot and Building entities, some common Property attributes can be a part of the Property superclass such as price, location, acres, and purchase date. Some attributes may not be able to be migrated to the superclass, like zoning class (residential, commercial, etc.) for Lot, or square footage and handicap accessibility for Property. This is an extension of the previous diagram showing the Lot, Property and Building EERD with a disjoint constraint.



iv) From the COMPANY database discussed in the class, what is the operation type specified in the Relational Algebra to retrieve the names of employees who work on all the projects that 'John Smith' works on. (1 point)

After performing a few project operations to get John's SSN and some selects to get the projects John works on, as well a project to get the list of every employee and what they work on, the operation type specified in relational algebra to retrieve the names of employees who work on all the projects John Smith works on, is division. You can divide the list of employees who work on a project by John Smith's projects to get the requested result.

Note: You could also *probably do an intersection, however division works well for this type of query.

If one relation has M tuples and N attributes and the other has J tuples and K attributes, what are the number of tuples and attributes for the Cartesian product for these two relations? (2 points)

The number of tuples after performing a Cartesian product operation for the given scenario is M*J. The number of attributes in each tuple for the given scenario is N+K.

If one relation has $M_1$ attributes and the other has $M_2$ attributes with *two* attributes specified as a join condition on each relation, how many attributes for the EQUIJOIN and how many attributes for the NATURAL JOIN (assuming $M_1 > 2$ and $M_2 > 2$)? (2 points)

The main difference between EQUIJOIN and NATURAL JOIN is that in NATURAL JOIN the join contains redundant attribute(s) in the result relation, while in EQUIJOIN the redundant attribute(s) of the second relation are excluded in the resulting relation. Thus, for NATURAL JOIN, the result would contain ($M_1 + M_2$) attributes, while for EQUIJOIN the result would contain ($M_1 + M_2 - 2$) attributes.

v) What are relational constraints defined in a relational database schema? (4 points) Which relational constraint defines a relationship between two relations? (1 point)

There are 4 main types of constraints defined in the relational database schema. The 4 types of constraints are named and described below:

- Domain Constraints: Include data types and ranges associated with a domain. An example could be a range of integers, a format for a date, a format for a SSN (9 numbers), etc.
- Key Constraints and Constraints on Null: The key constraint specifies that the unique identifier for each tuple cannot have the same value as any other tuple in the relation. This is helpful when querying by the unique identifier. Further, other fields can be specified to not accept null, for example for required fields on a form, etc.
- Entity Integrity Constraints: The unique identifier discussed earlier cannot be null. This would result in that tuple being impossible to identify.

- Referential Integrity Constraint: This constraint specifies between two relations that the primary key of the parent relation migrates to the child relation as a foreign key to maintain referential integrity between the relations. The foreign key references the primary key in the referenced relation.

As discussed above, the relational constraint which defines the relationship between two relations is the Referential Integrity Constraint.

vi) Specify the following queries on the COMPANY relational database schema shown in Figure 5.5 (page 161, 7th edition), using the relational operators. Also show the result of each query as it would apply to the database state of Figure 5.6 (page 162, 7th edition).

EMPLOYEE

| FNAME | MINIT | LNAME | SSN | BDATE | ADDRESS | SEX | SALARY | SUPERSSN | DNO |
|---|---|---|---|---|---|---|---|---|---|
| John | B | Smith | 123456789 | 1965-01-09 | 731 Fondren, Houston, TX | M | 30000 | 333445555 | 5 |
| Franklin | T | Wong | 333445555 | 1955-12-08 | 638 Voss, Houston, TX | M | 40000 | 888665555 | 5 |
| Alicia | J | Zelaya | 999887777 | 1968-01-19 | 3321 Castle, Spring, TX | F | 25000 | 987654321 | 4 |
| Jennifer | S | Wallace | 987654321 | 1941-06-20 | 291 Berry, Bellaire, TX | F | 43000 | 888665555 | 4 |
| Ramesh | K | Narayan | 666884444 | 1962-09-15 | 975 Fire Oak, Humble, TX | M | 38000 | 333445555 | 5 |
| Joyce | A | English | 453453453 | 1972-07-31 | 5631 Rice, Houston, TX | F | 25000 | 333445555 | 5 |
| Ahmad | V | Jabbar | 987987987 | 1969-03-29 | 980 Dallas, Houston, TX | M | 25000 | 987654321 | 4 |
| James | E | Borg | 888665555 | 1937-11-10 | 450 Stone, Houston, TX | M | 55000 | null | 1 |

DEPT_LOCATIONS

| DNUMBER | DLOCATION |
|---|---|
| 1 | Houston |
| 4 | Stafford |
| 5 | Bellaire |
| 5 | Sugarland |
| | Houston |

DEPARTMENT

| DNAME | DNUMBER | MGRSSN | MGRSTARTDATE |
|---|---|---|---|
| Research | 5 | 333445555 | 1988-05-22 |
| Administration | 4 | 987654321 | 1995-01-01 |
| Headquarters | 1 | 888665555 | 1981-06-19 |

WORKS_ON

| ESSN | PNO | HOURS |
|---|---|---|
| 123456789 | 1 | 32.5 |
| 123456789 | 2 | 7.5 |
| 666884444 | 3 | 40.0 |
| 453453453 | 1 | 20.0 |
| 453453453 | 2 | 20.0 |
| 333445555 | 2 | 10.0 |
| 333445555 | 3 | 10.0 |
| 333445555 | 10 | 10.0 |
| 333445555 | 20 | 10.0 |
| 999887777 | 30 | 30.0 |
| 999887777 | 10 | 10.0 |
| 987987987 | 10 | 35.0 |
| 987987987 | 30 | 5.0 |
| 987654321 | 30 | 20.0 |
| 987654321 | 20 | 15.0 |
| 888665555 | 20 | null |

PROJECT

| PNAME | PNUMBER | PLOCATION | DNUM |
|---|---|---|---|
| ProductX | 1 | Bellaire | 5 |
| ProductY | 2 | Sugarland | 5 |
| ProductZ | 3 | Houston | 5 |
| Computerization | 10 | Stafford | 4 |
| Reorganization | 20 | Houston | 1 |
| Newbenefits | 30 | Stafford | 4 |

DEPENDENT

| ESSN | DEPENDENT_NAME | SEX | BDATE | RELATIONSHIP |
|---|---|---|---|---|
| 333445555 | Alice | F | 1986-04-05 | DAUGHTER |
| 333445555 | Theodore | M | 1983-10-25 | SON |
| 333445555 | Joy | F | 1958-05-03 | SPOUSE |
| 987654321 | Abner | M | 1942-02-28 | SPOUSE |
| 123456789 | Michael | M | 1988-01-04 | SON |
| 123456789 | Alice | F | 1988-12-30 | DAUGHTER |
| 123456789 | Elizabeth | F | 1967-05-05 | SPOUSE |

*Figure 5.6*

(a) Retrieve the names of employees in department 5 who work more than 10 hours per week on the 'ProductX' project. (3 points)

Note: The symbol ⋈ is my best attempt at matching the symbol for theta join in the below expressions.

$$ProjectX \leftarrow \sigma_{name="ProjectX"}(Project)$$
$$EmpsWorkX \leftarrow ProjectX \bowtie_{name=ProjectName}(Works\_On)$$
$$EmpsOver10Hrs \leftarrow Employee \bowtie_{Ssn=Essn}(\sigma_{Hours>10}(EmpsWorkX))$$
$$Result \leftarrow \pi_{Fname,Lname}(\sigma_{DepartmentNumber=5}(EmpsOver10Hrs))$$

Name: Brian Loughran

Query Result:

| Fname | Lname |
|---|---|
| John | Smith |
| Joyce | English |

(b) Find the names of employees that are directly supervised by 'Franklin Wong' (2 points)

$$WongSSN \leftarrow \pi_{SSN}(\sigma_{Fname="Franklin" \ AND \ Lname="Wong"}(Employee)$$
$$Result \leftarrow \pi_{Fname,Lname}(Employee \ltimes_{SuperSSN=\ WongSSN}(WongSSN)$$

Query Result:

| Fname | Lname |
|---|---|
| John | Smith |
| Narayan | Ramesh |
| Joyce | English |

2. A college needs a database that supports its placement office to help college students finding out potential employers. The primary purpose of the database is to schedule interviews and facilitate searches by students and companies. The requirements are the followings:

- Student data include a unique identifier, a name, phone number, an email address, a web address, a major, a minor, and a GPA.
- The placement office maintains a standard list of positions based on the Labor Department's list of occupations. Position data include a unique position identifier and a position description.
- Company data include a unique company identifier, a company name, and a list of positions and interviewers. Each company must map its positions into the position list maintained by the placement office. For each available position, the company lists the cities in which positions are available.
- Interviewer data include a unique interviewer identifier, a name, a phone, an email address, and a web address. Each interviewer works for one company.
- An interview includes a unique interview identifier, a date, a time, a location (building and room), an interviewer, and a student.
- An interview may be setup without knowing the interviewer
- An interview is setup with the student.

Design a database with ERD using **IE notation**. This E/R diagram should include entities, relationships, attributes, and constraints. You need to specify primary keys and foreign keys for all entities. You also need to specify the rationale for your design. (**25 points**)

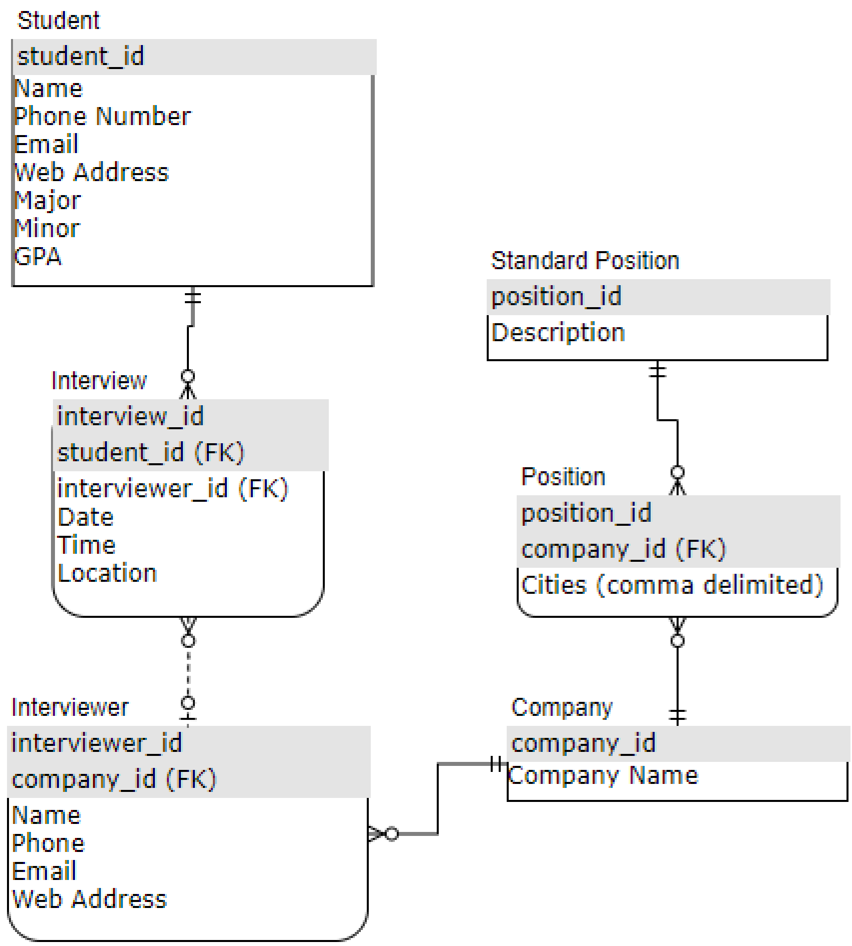
Name: Brian Loughran

The six entities to be managed for this database system are Student, Standard Position, Company Interviewer, Interview, and Position. The first four are as described in the above requirements with their own primary key and attributes. All four save for Interviewer are considered to be strong entities, since they can exist independent of any other entity. This exception is made for Interviewer, since the interviewer must work for a company according to the requirements. Because Interviewer is a weak entity the primary key for Company is mapped to Interview as a foreign key. Interview is an entity that is created to manage the many to many relationship between Student and Interviewer, each of which can have more than one interview scheduled. Interview is also where the date, time and location for the interview are stored, and Interview also receives the primary key of Student and Interviewer as foreign keys. It is assumed that location can be stored in a single string for Interview. The last entity, Position, manages the many to many relationship between Standard Position (the one based on the Labor Department's list of occupations) and Company, thus receives the primary key of Standard Position as Company as foreign keys.

All relationships in the ERD are considered identifying relationships, since they are each between a strong entity and a weak entity except the relationship between Interview and Interviewer, which is a non-identifying relationship, since Interview does not necessarily have to be assigned an Interviewer. The relationship between Student and Interview is 1 to 0/many. Each Interview must have exactly one Student, and each Student may have 0 or more Interviews. The relationship between Interviewer and Interview is 0/1 to 0/many, since each interview may be set up with or without an Interviewer, and each Interviewer can do between 0 and many Interviews. The relationship between Interviewer and Company is 0/many to 1. Each Interviewer must have exactly one Company, and each Company can have 0 or many Interviewers. The relationship between Company and Position is 1 to 0/many. Each Company can have 0 or many Positions, however each Position must be assigned to a Company. Finally, the relationship between Standard Position and Position is also 1 to 0/many. Each Standard Position can be mapped to 0 or many positions, however each Position must be assigned to a Standard Position.

These are the assumptions made to create the following ERD:

Name: Brian Loughran



Name: Brian Loughran

3. A nationwide health club provides a broad range of fitness services for its members at several major US locations at a discount. A member may be an individual member or a company member that allows the company's employees to use the services. An individual member will pay his/her own fees for services, a company employee will be paid for by his/her company. The payment will be charged by credit cards.

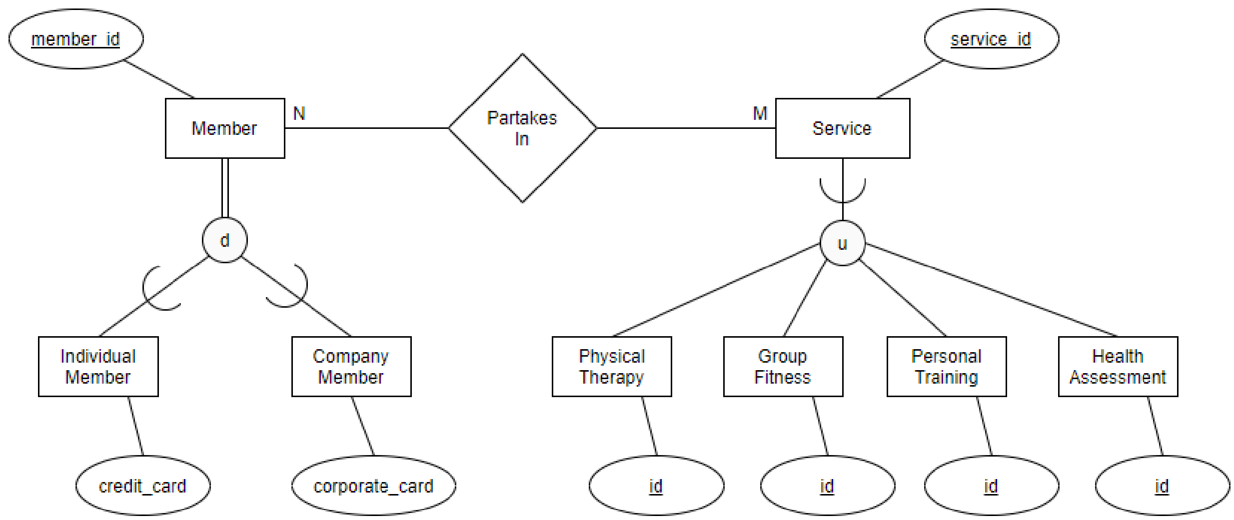
   The health club offers the following categories of services:
   - Physical Therapy – Massage, Sports medicine through a network of physicians
   - Group Fitness – aquatics, aerobics, weight classes
   - Personal Training – weight loss, toning, weightlifting, cardio
   - Health Assessments – BMI, cardiovascular, muscle tone

   Use **Enhanced ER Diagram** (EERD) to design a database to track a member to use their services. The EERD must include some EERD notations such as superclass/subclass, categories, and other constraints. You **only** need to include **key attributes** for the entities (such as Physical Therapy and so on) and relationships to reduce complexity of your design. You do not need to identify attributes associated with Physical_Therapy and other entities.

   Please clearly state any assumptions you make and your design rationale. You may use any graphical tools to draw your EERD or you can use hand-writing to clearly show your EERD (**20 points**)

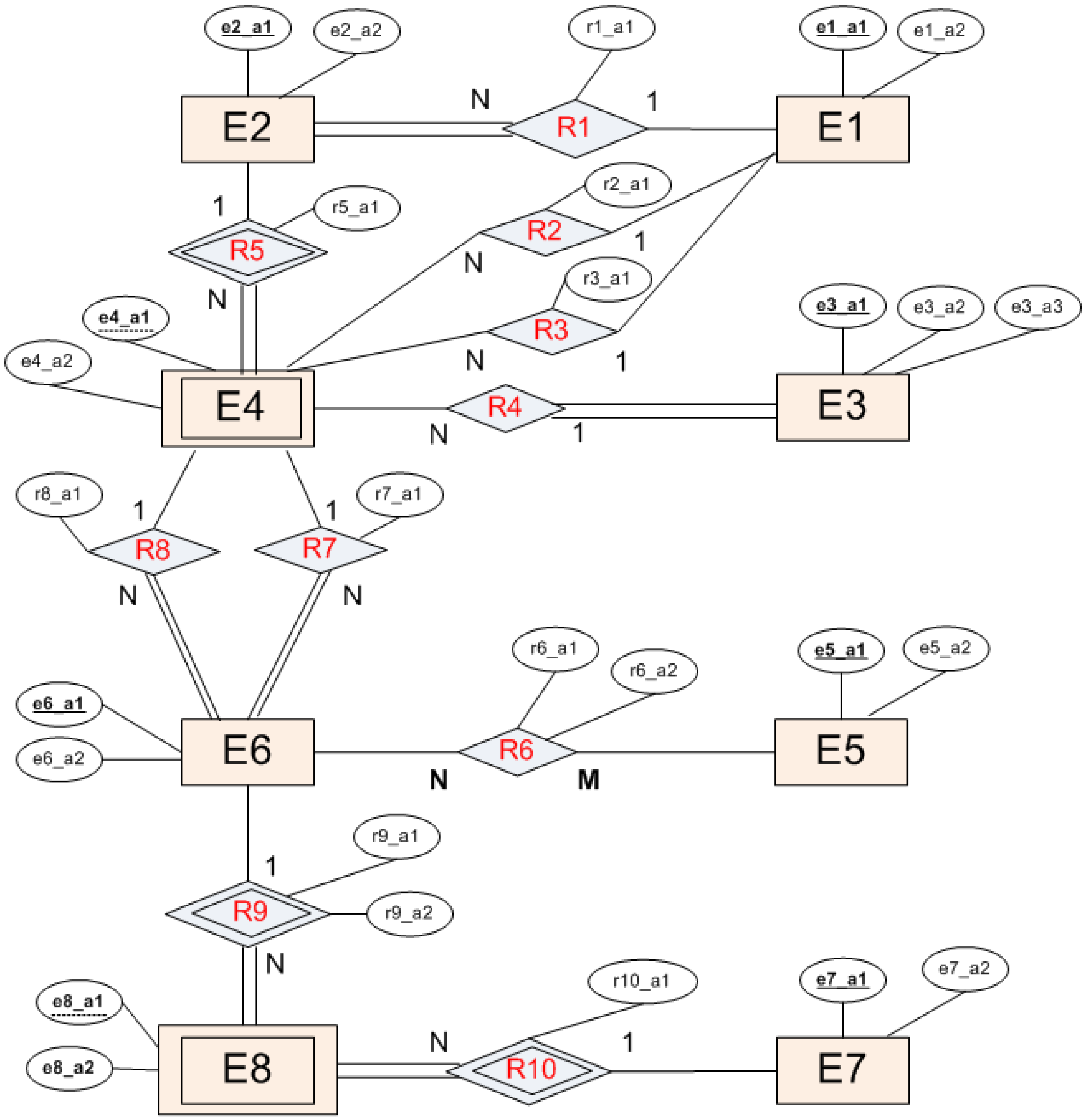
This database description is best handled using an EERD, since using some EERD concepts such as superclasses, categories, and other constraints. For example, a Member may either be an Individual Member or a Company Member, which is a good place to use a disjoint constraint to specify that a Member can be one or the other. Further, each Service offered by the club can be a Union of the categories Physical Therapy, Group Fitness, Personal Training or Health Assessment. Each of those 4 categories can have their own set of attributes, but for simplicity this EERD shows only their primary keys. Both Member and Service are strong entities, since they can exist independently.
Member and Service have a N:M relationship between the two called Partakes In. The relationship is non-identifying, since both Member and Service are strong entities. The relationship is also partial participation between Member and Service, since each Member can Partake In 0 to N Services, and each Service can have 0 to M Members Partake In the Service. The EERD describing this is shown below:

CS 605.641 Principles of Database Systems
Midterm Examination Summer 2020
(Dates: 06/28/2020 – 07/01/2020; No Discussion, Collaboration, Consultation, Sharing)
Name: Brian Loughran



Name: Brian Loughran

4. Convert the following ERD in Chen's notation to an ERD in **IE notation**. All entities, attributes, and relationships are artificial. (**25 points**)




You need to clearly show strong and weak entities, foreign key migrations, total participations and partial participations, and cardinality ratios.

Name: Brian Loughran