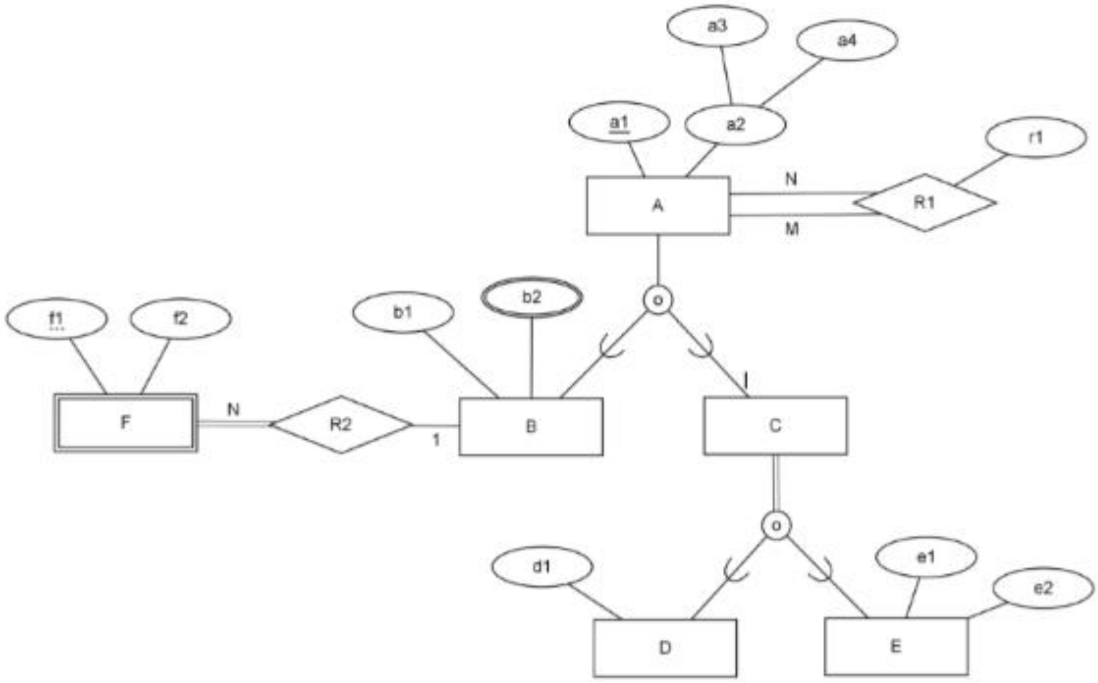
CS6131
DATABASE DESIGN

Name:___________________________ Date: ______________

ASSIGNMENT 2

SECTION A: STRUCTUED QUESTION [40 marks]

1. **[10m]** Consider the EER diagram:



a) Answer the following questions based on the EER diagram above:

i. Can a 'A' entity be neither a 'B' nor a 'C'? Explain your answer clearly.

Yes it can. It is not mandatory for an entity A to specialize in either subclass, since there is no double line between the A entity and the circle with the o in it on the ER diagram.

ii. Can there be a 'E' entity that is also a 'B'? Explain your answer clearly.

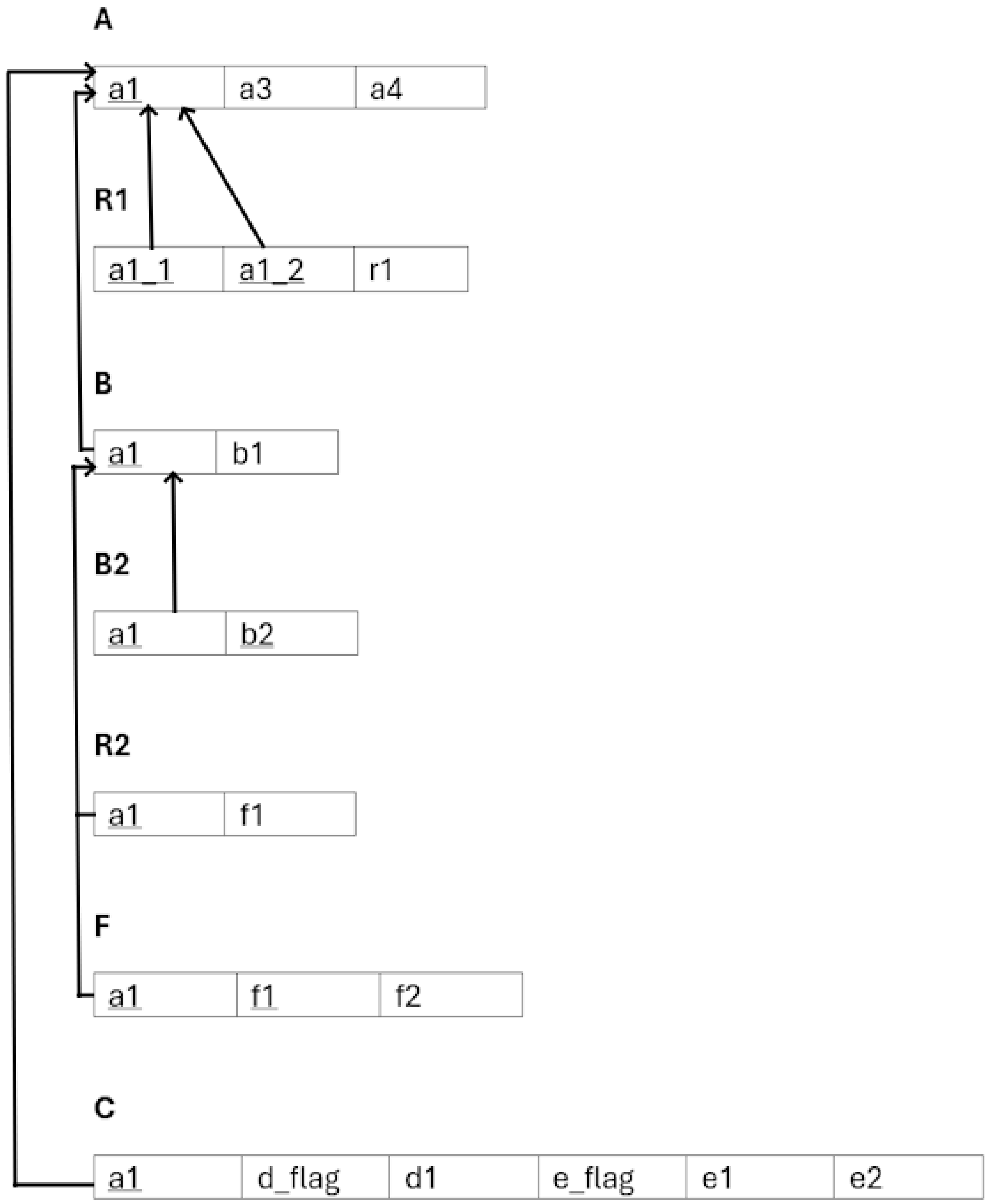
Yes. An entity can have overlapping subclasses (denoted by the o in the circle connected to entity A), which means that an entity A can be both, either or or neither a B or C. Since E is a subclass of C, an entity can be both an E and a B at the same time.

b) You are given the following information about the database design requirements:

- Most C entities are also D entities, and
- Most E entities are also D entities,
- Maintaining all the relevant integrity constraints are a priority.

i. Map the EER into a suitable relational schema.
Underline the primary key, and state down the foreign keys clearly by drawing appropriate arrows. Use the 'box-style' notation for your relational schema. The arrows showing foreign key should be done NEATLY and CLEARLY (i.e. NO overlapping arrows).

Your Diagram:



ii. Explain why the chosen mapping strategy for the inheritance of entities A, B, and C is more suitable compared to other possible strategies. Discuss the advantages and disadvantages of the selected approach over alternatives.

The chosen mapping strategy, multiple relations super classes and subclasses, is chosen because the inheritance is an optional, overlapping specialization. The advantages of this method is that there is no null values due to irrelevant data from other subclasses being in the same table and also more efficient queries when querying specific subclasses. The disadvantages is that there are more tables, which makes it harder to manage in general, as well as needing to perform multiple joins if an entity is of both subclasses, which is computationally inefficient.

iii. Explain why the chosen mapping strategy for the inheritance of entities C, D, and E is more suitable compared to other possible strategies. Discuss the advantages of the selected approach over alternatives.

The chosen mapping strategy, single relation with multiple type attribute, is chosen because of the statement in the question stating that most C entities are also D entities and most E entities are D entities. This means that there will be much fewer null values in the table. The advantages of using this method is that since all of the sub classes are contained within the same table, the insertion of data is much easier and more straightforward. The disadvantages would be that when querying a specific subclass, irrelevant columns from the other subclass will also be returned, which is an inefficiency in performance.

2. **[10m]** You are given a relation R(A, B, C, D, E, G). The following set *F* of functional dependency hold for this relation.

$$F = \{A \rightarrow BC, AB \rightarrow C, B \rightarrow C, CD \rightarrow E, E \rightarrow AC, G \rightarrow AB\}$$

a) Use Armstrong's Axiom to prove that BD $\rightarrow$ E is in F^+. State the axioms used in each step clearly.

1. A $\rightarrow$ BC (given)
2. AB $\rightarrow$ C (given)
3. B $\rightarrow$ C (given)
4. CD $\rightarrow$ E (given)
5. E $\rightarrow$ AC (given)
6. G $\rightarrow$ AB (given)
7. BD -> CD (augmentation on 3)
8. BD -> E (transitivity on 7 and 4)

b) Find all candidate key(s) for the relation R with *F*. You will need to prove that the candidate key(s) listed is indeed a candidate key. Show your working clearly.

DG

D^+= {D} G^+= {ABCG} DG^+= {ABCDEG}

Since neither D^+ or G^+ is minimal, DG is minimal and hence a candidate key

c) Use the algorithm discussed in the notes to find the minimal cover of the above set ***F*** of functional dependencies. Show your working clearly (i.e. Step 1, 2, 3). Write down the remaining set of functional dependencies at the end of each step.

Step 1: Make RHS of each FD into a single attribute

F = { A → B, A → C, AB → C, B → C, CD → E, E → A, E → C, G → A, G → B }

Step 2: Eliminate redundant attributes from LHS

Consider AB -> C and CD -> E

Find A^+ under F

A^+ = A

= AB (A -> B)

= ABC (A -> C)

Since A^+ already contains C, AB -> C is redundant and we can reduce it to A -> C, which is already in F, and remove AB -> C.

Since there is no other way to derive E (neither C^+ nor D^+ contains E), CD -> E is necessary and hence will remain.

F = { A → B, A → C, B → C, CD → E, E → A, E → C, G → A, G → B }

Step 3: Delete redundant FDs from F

Let T = F

Compute A^+ under T minus A -> C

A^+ = A

= AB (A -> B)

= ABC (B -> C)

Since A^+ contains C, this implies A -> C is redundant.

Compute G^+ under T minus G -> B

G^+ = G

= AG (G -> A)

= ABG (A -> B)

= ABCG (B -> C)

Since G^+ contains B, this implies G -> B is redundant.

Compute E^+ under T minus E -> C

E^+ = E

= AE (E -> A)

= ABE (A -> B)

= ABCE (B -> C)

Since E^+ contains C, this implies E -> C is redundant.

Hence, the minimal cover is

$F = \{$ $A \rightarrow B, B \rightarrow C, CD \rightarrow E, E \rightarrow A, G \rightarrow A$ $\}$

d) Explain why the relation is not in 3NF. Hence, give a 3NF decomposition of R with *F*.

Looking at the first FD, A -> B, A is not the superkey, which violates the 2nf condition. Therefore, the relation is not in 2nf so it cannot be in 3nf (since being 2nf is part of the conditions of being 3nf).

R1(A, B)
R2(B, C)
R3(C, D, E)
R4(E, A)
R5(G, A)
R6(D, G)

3. **[10m]** You have been hired to design a database for **nushFit**, a company that owns and operates multiple gyms. The database will be used to manage various aspects of its operations. The system must accommodate the following requirements:

   Each gym branch has a unique branch name, one or more phone numbers, and an address consisting of the street number, street name, and ZIP code. Every gym branch is required to have exactly one designated manager who is responsible for overseeing its operations.

   Employees are uniquely identified by their employee id and we also store their Social Security Number (SSN), their full name, comprising of first name and last name. Employees may work at multiple gym locations, and the database must track the percentage of time they spend at each branch. For instance, an employee may work 50% at one gym and 50% at another. Some employees may specialize in a specific role, and each employee can have at most one specialization. The available specializations include manager, receptionist, and personal trainer. Managers are responsible for overseeing one or more gyms, with the condition that each gym must have exactly one manager. Personal trainers may hold multiple certifications such as yoga, aerobics, or sports nutrition, which must be recorded in the system. Receptionists handle front-desk operations, and the system must record their shift timing, languages spoken and whether they are certified in first aid.

   Customers are uniquely identified by their SSN, and their personal details, including name, birthdate, and age, must be stored. A customer can hold memberships at multiple gym locations, and each gym can have multiple customers associated with it. Customers also have the ability to invite guests to the gym, and the database must track these guest associations.

   Guests, unlike customers, do not have independent gym memberships but are instead linked to a specific inviting customer. The system must store relevant guest information such as their name and age while ensuring that each guest is associated with exactly one customer. It is assumed that no two guests invited by the same customer will have the same name.

   Design an ER/EER diagram for the above gym database.

Your Diagram: