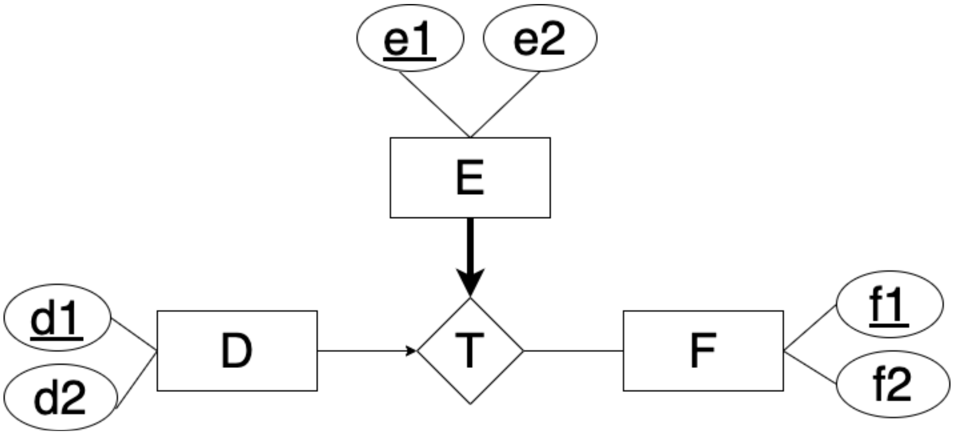
CPSC 304: Introduction to Relational Databases
In-Class Exercise Solution

Suppose that we have a ternary relationship T between entity sets D, E, and F such that D has a key constraint and E has a key constraint and total participation; these are the only constraints. D has attributes d1 and d2, with d1 being the key; E has attributes e1 and e2, with e1 being the key; and F has attributes f1 and f2, with f1 being the key. T has no descriptive attributes. All attributes are integers. Write SQL statements that create tables corresponding to this information so as to capture as many of the constraints as possible. If you cannot capture some constraint, explain why.

First, let's draw the ER diagram corresponding to this problem. As a reminder, when we use the term "key constraints", we are referring to the arrows in an ER diagram.



Now, let's first create the relational schemas for this diagram.

- F(f1, f2)
- D(d1, d2)
- ET(e1, e2, d1, f1), d1 unique, d1 and f1 not null

Common questions:

- Why is the T relation with E? Can it not be with D?
  The information about the T relationship is stored with E because of the participation constraint. We can use a NOT NULL constraint on the foreign keys to D and F to ensure every E tuple has an associated D and F value.

- Why does the relation containing the representation of entity set D not have a unique constraint?
  Since d1 is a primary key, it is already guaranteed to be unique.

- Why does the relation containing the representation of entity set D not contain information about E and F?
  The information relating to relationship T has already been stored in the relation containing the representation of entity set E. There is no need to duplicate information.

- Why is there a unique constraint on d1 in ET?
  The key constraint on D tells us that a given D entity can only participate in the relationship set T at most once. In order to ensure that this is true, we need to have the UNIQUE constraint in E.

Answer:

```
CREATE TABLE E_T (
    e1 int,
    e2 int,
    d1 int NOT NULL,
    f1 int NOT NULL,
    PRIMARY KEY (e1),
    UNIQUE (d1),
    FOREIGN KEY (d1) REFERENCES D,
    FOREIGN KEY (f1) REFERENCES F
)

CREATE TABLE D (
    d1 int,
    d2 int,
    PRIMARY KEY (d1)
)

CREATE TABLE F (
    f1 int,
    f2 int,
    PRIMARY KEY (f1)
)
```