

CS2102: Database Systems

Tutorial 2 — Entity-Relationship Model

Question 1

The Varsity International Network of Oenology wishes to computerise the management of the information about its members as well as to record the information they gather about various wines. Your company, Apasaja Private Limited, is commissioned by the Varsity International Network of Oenology to design and implement the relational schema of the database application. The organisation is big enough so that there could be several members with the same name. A card with a unique number is issued to identify each drinker. The contact address of each member is also recorded for the mailing of announcements and calls for meetings.

At most once a week, VINO organises a tasting session. At each session, the attending members taste several bottles. Each member records for each bottle his or her evaluation of the quality (very good, good, average, mediocre, bad, very bad) of each wine that she or he tastes. The evaluation may differ for the same wine from one drinker to another. Actual quality and therefore evaluation also varies from one to another bottle of a given wine. Every bottle that is opened during the tasting session is finished during that session.

Question 1

Each wine is identified by its name (“Parade D’Amour”), appellation (“Bordeaux”) and vintage (1990). Other information of interest about the wine is the degree of alcohol (11.5), where and by whom it has been bottled (“Mis en Bouteille par Amblard-Larolphie Negociant-Eleveur a Saint Andrede Cubzac (Gironde) - France”), the certification of its appellation if available (“Appellation Bordeaux Controlée”), and the country it comes from (produce of “France”).

Generally, there are or have been several bottles of the same wine in the cellar. For each wine, the bottles in the wine cellar of VINO are numbered. For instance, the cellar has 20 bottles numbered 1 to 20 of a Semillon from 1996 named Rumbalara. For documentation purposes VINO may also want to record wines for which it does not own bottles. The bottles are either available in the cellar or they have been tasted and emptied.

Question 1

- (a) Identify the entity sets. Justify your choice by quoting the sentences in the text that support it.

Recap - Entity Set

Real-world things or objects that are distinguishable from other objects

Typically **nouns**

User

Airport

Booking

Flight

Question 1

- (a) Identify the entity sets. Justify your choice by quoting the sentences in the text that support it.

member

*"... information about its **members** ..."*

wine

*"... record the information they gather about various **wines**."*

bottle

*"... there are or have been several **bottles** of the same wine ..."*

session

*"... VINO organises a tasting **session**."*

Question 1

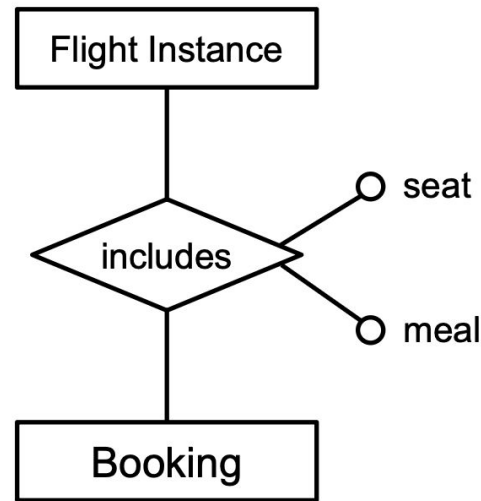
- (b) Identify the relationship sets and the entity sets that they associate. Justify your choice by quoting the sentences in the text that support it.

Recap - Relationship Set

Association among two or more entities

Can have their own attributes that further describe the relationship

Names are typically **verbs**

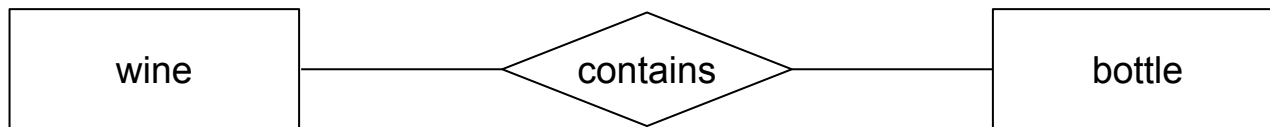


Question 1

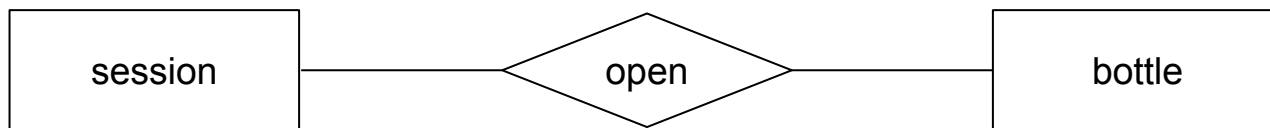
- (b) Identify the relationship sets and the entity sets that they associate. Justify your choice by quoting the sentences in the text that support it.



"... the attending members taste several bottles."



"... the attending members taste several bottles."



"... the attending members taste several bottles."

Question 1

- (c) For each entity set and relationship set identify its attributes. Justify your choice by quoting the sentences in the text that support it.

member

wine

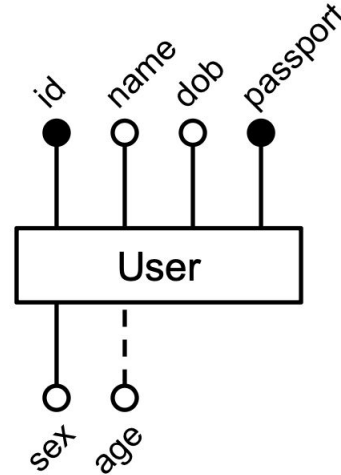
bottle

session

Recap - Attribute

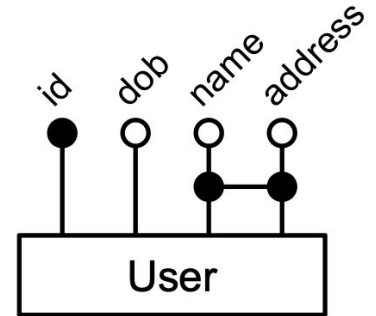
Types of attributes:

- Key attributes
- Derived attributes
- Normal attributes



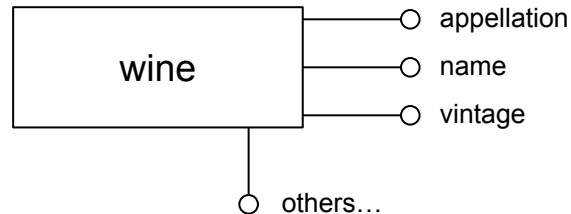
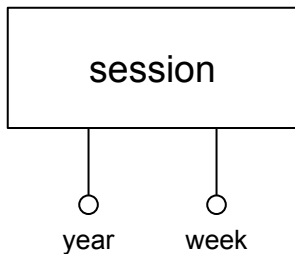
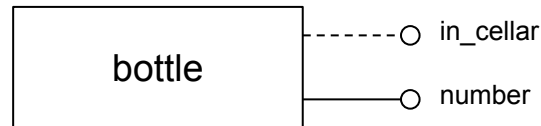
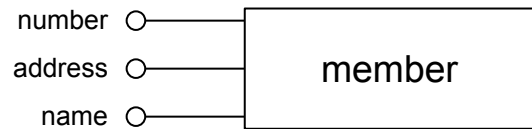
Composite key attributes:

- 2 or more attributes together uniquely identify each entity



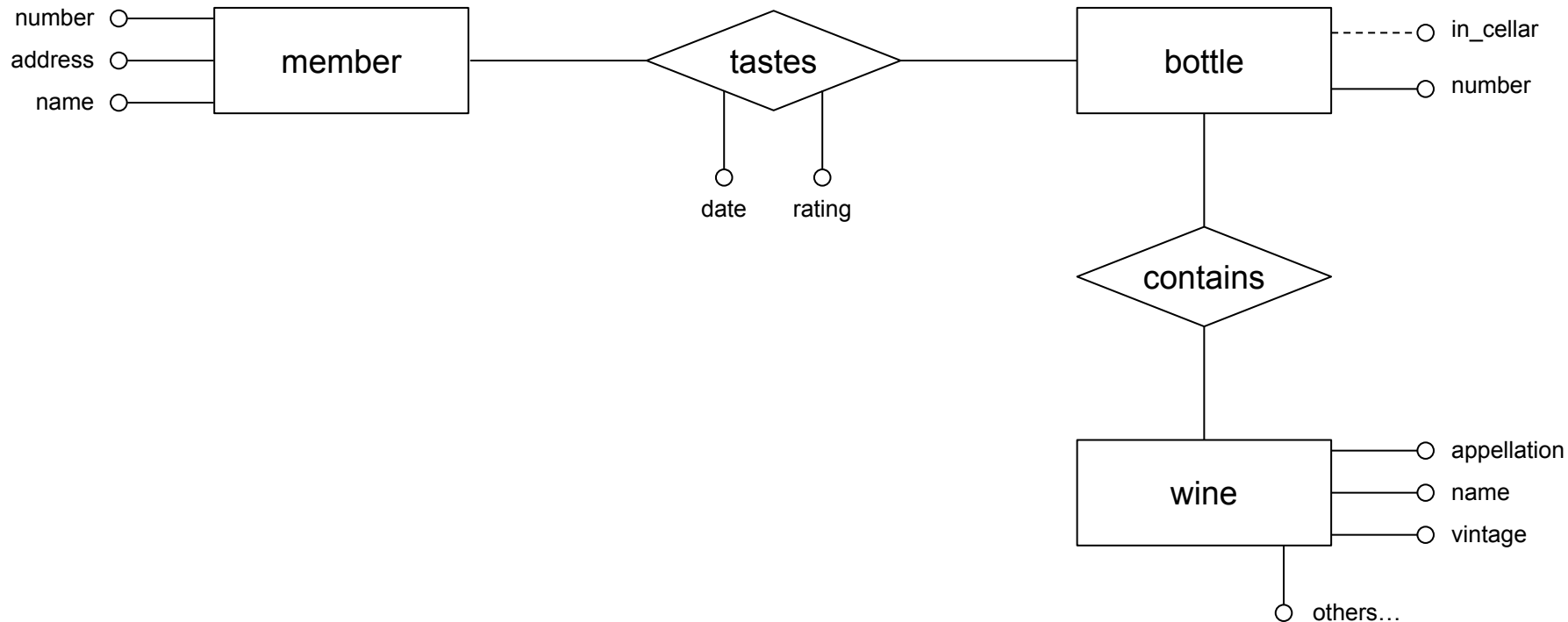
Question 1

- (c) For each entity set and relationship set identify its attributes. Justify your choice by quoting the sentences in the text that support it.



Question 1

(d) For each entity set, identify its keys.



Recap - Weak Entity Set

A weak entity can only be uniquely identified by considering the primary key of the owner entity

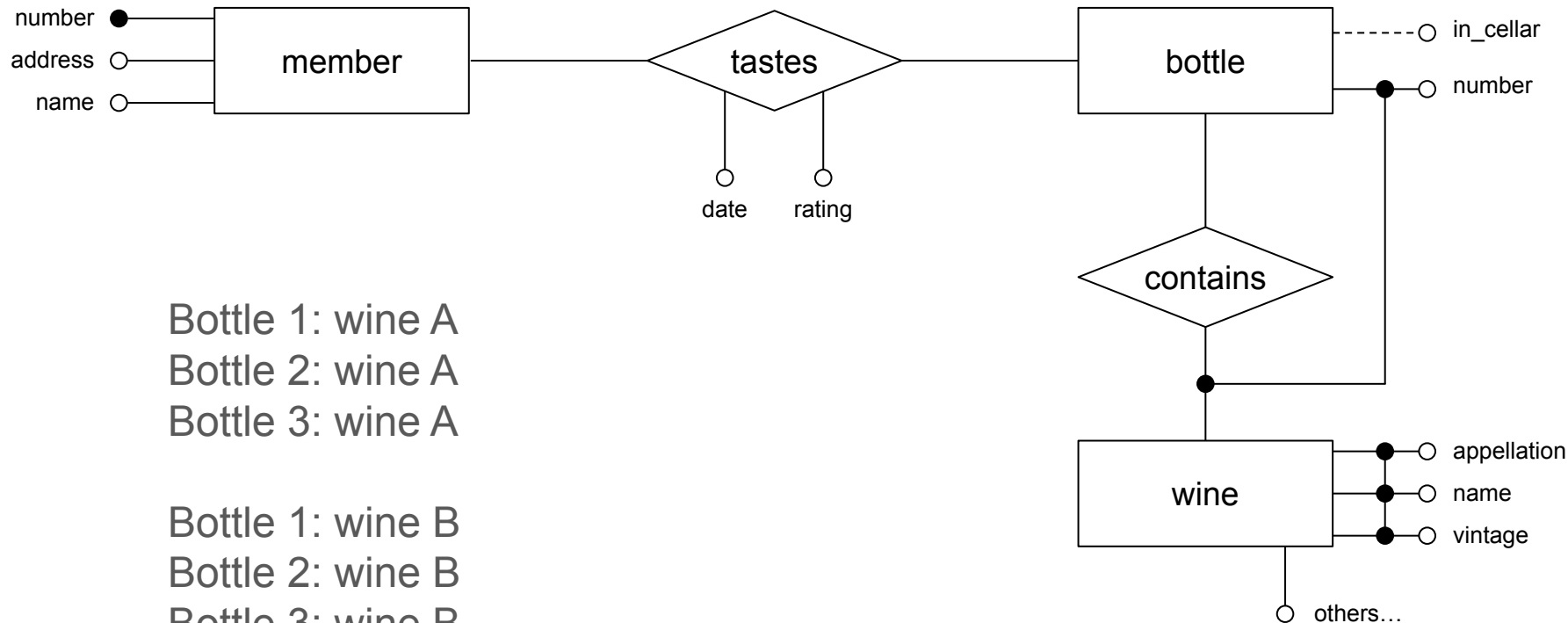
A weak entity's existence depends on the existence of its owner entity

- Requirements

- Many-to-one relationship (identifying relationship) from weak entity set to owner entity set
(one-to-one possible but less common)
- Weak entity set must have (1, 1) attached to identifying relationship

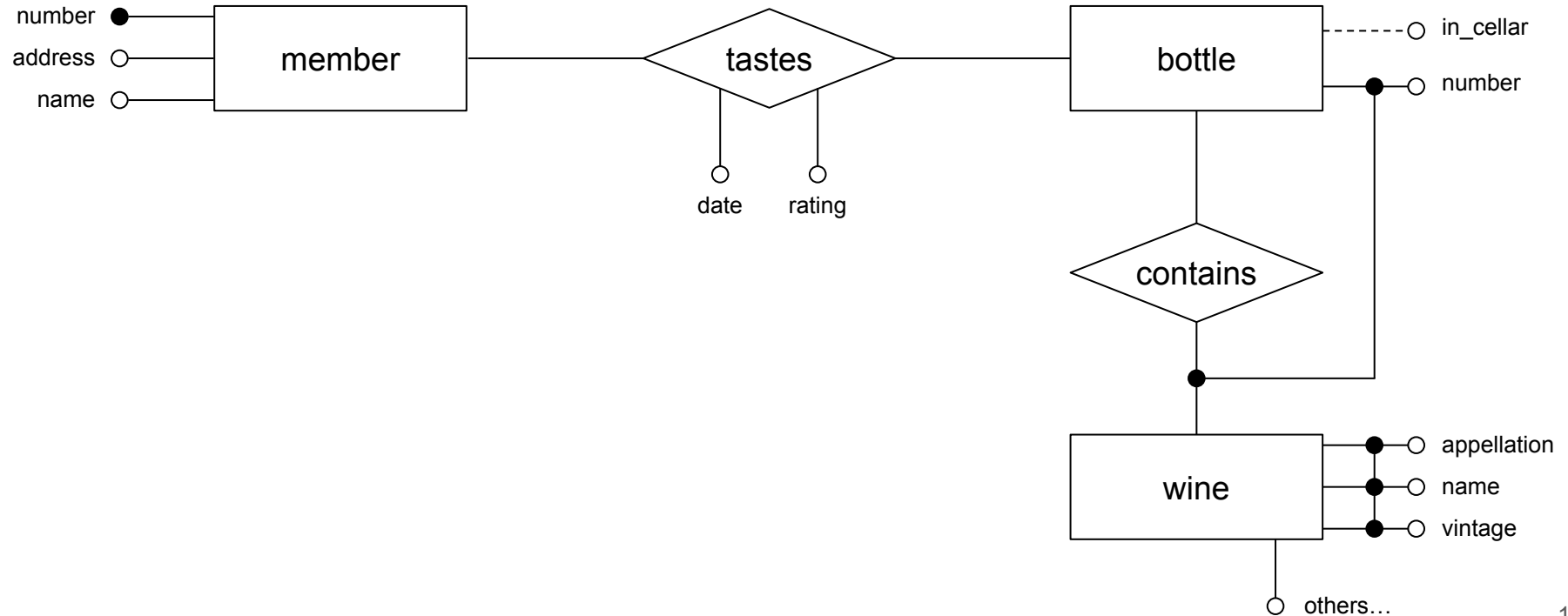
Question 1

(d) For each entity set, identify its keys.



Question 1

(e) For each entity set and each relationship set in which it participates, indicate the minimum and maximum participation constraints.

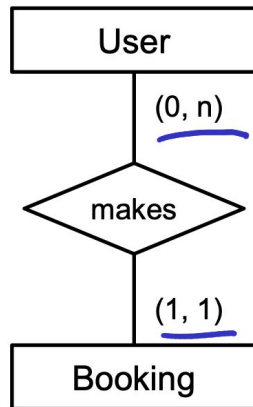


Recap - Cardinality

Participation constraints → describe how often an entity can participate in a relationship

3 basic cardinality constraints:

- Many-to-many
- Many-to-one
- One-to-one

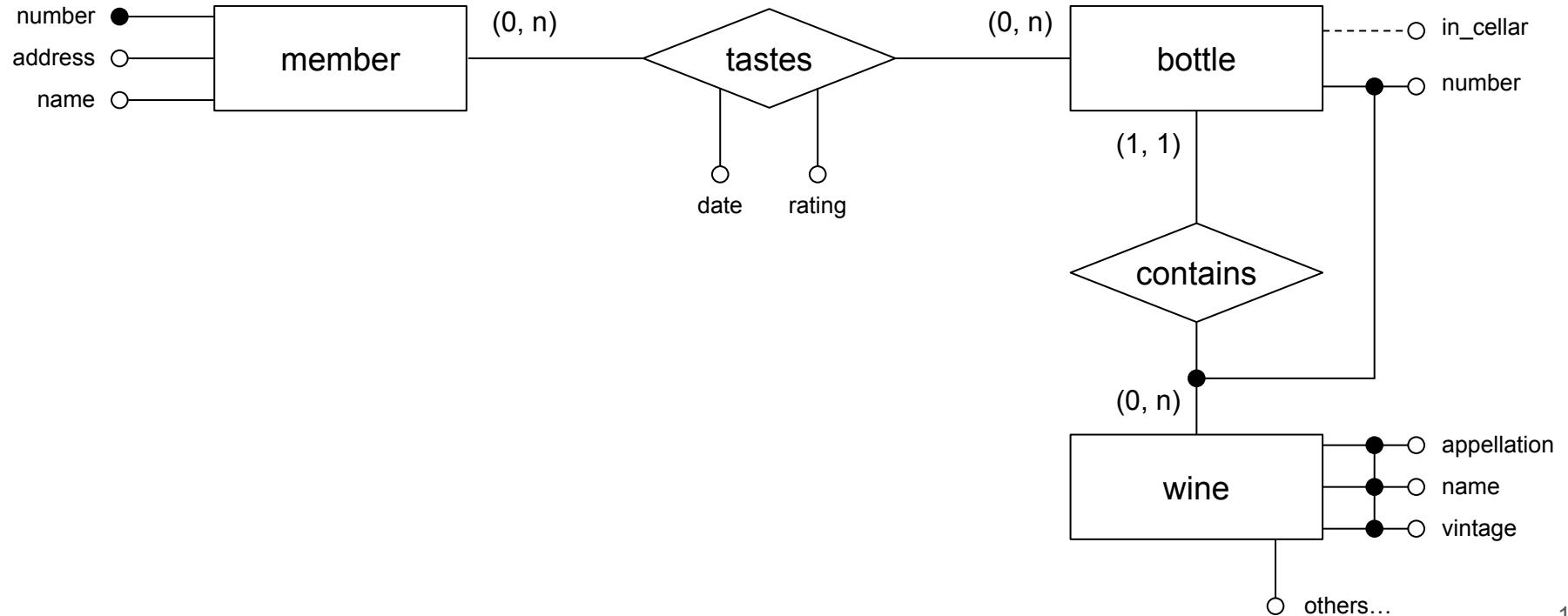


Interpretation

- Each user can make multiple bookings
(but not every user must have made a booking)
- Each booking was done by exactly one user
(implies that each booking is associated with a user)

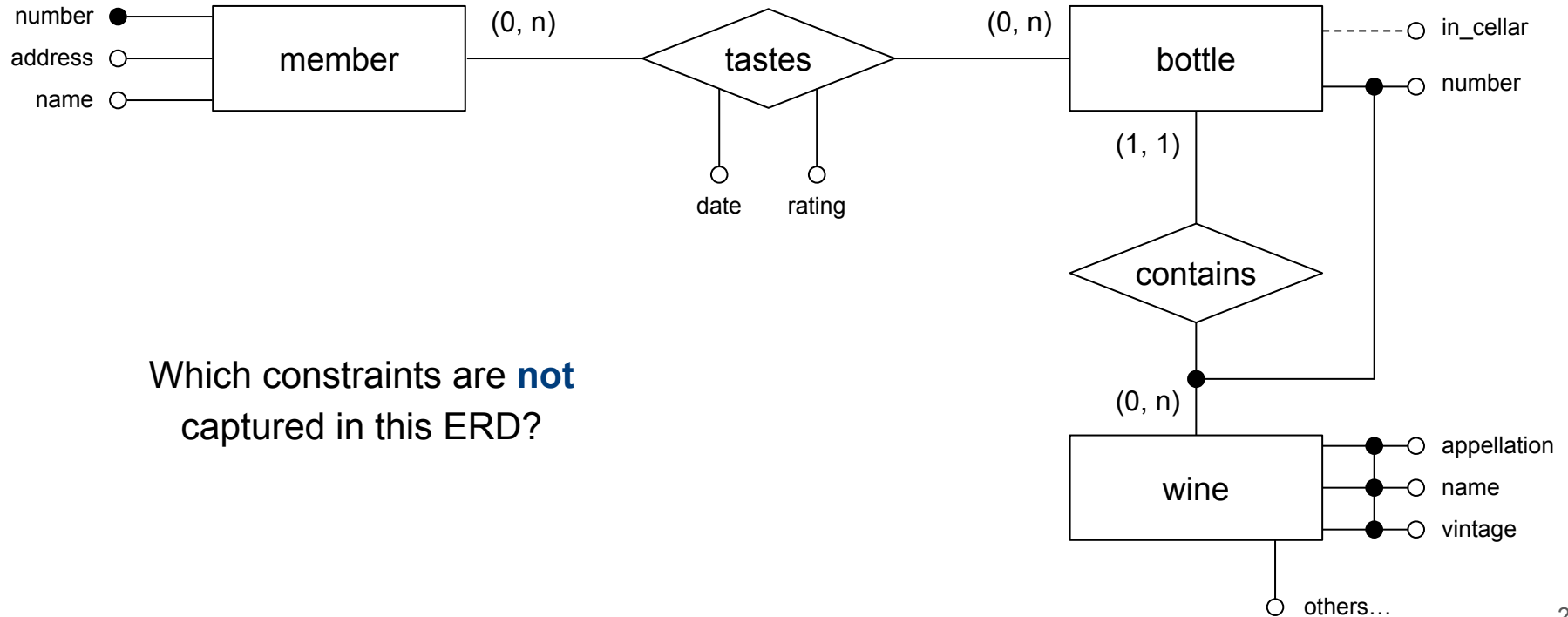
Question 1

(e) For each entity set and each relationship set in which it participates, indicate the minimum and maximum participation constraints.



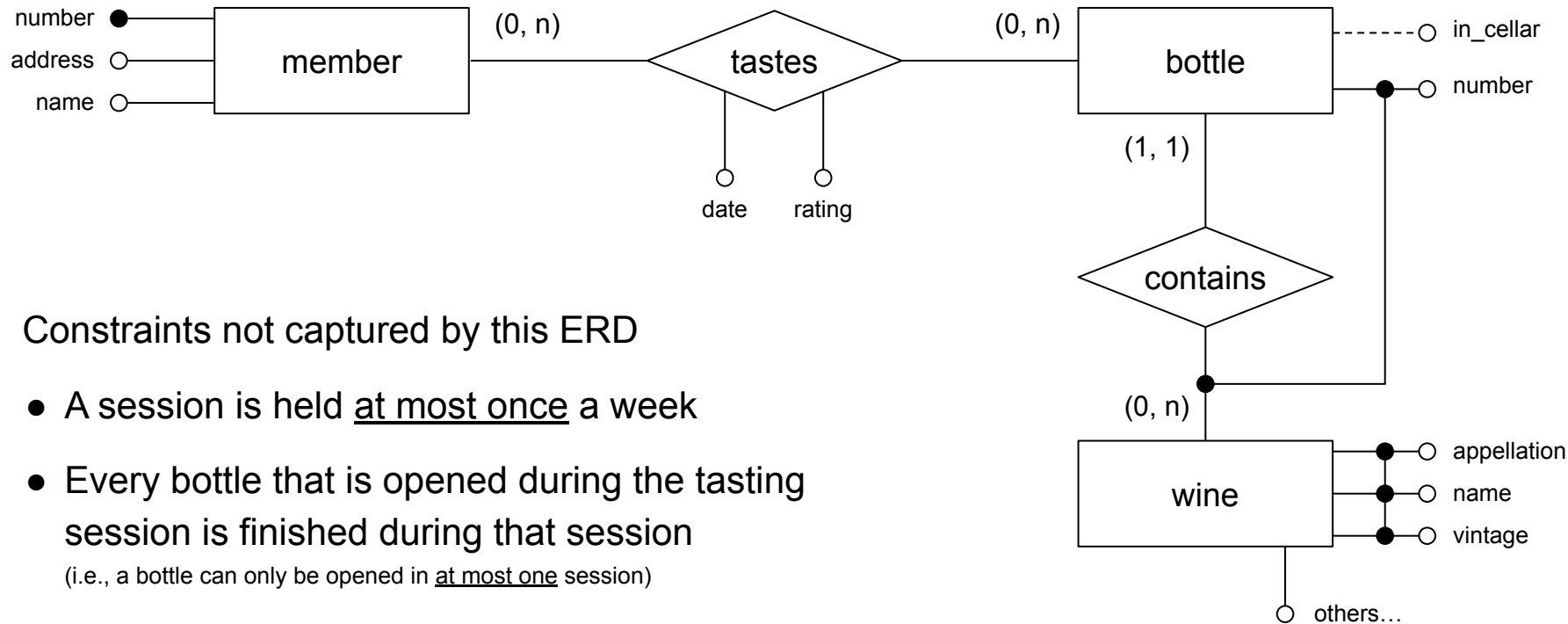
Question 1

(f) Draw the corresponding entity-relationship diagram with the key and participation constraints. Indicate in English the constraints that cannot be captured, if any.

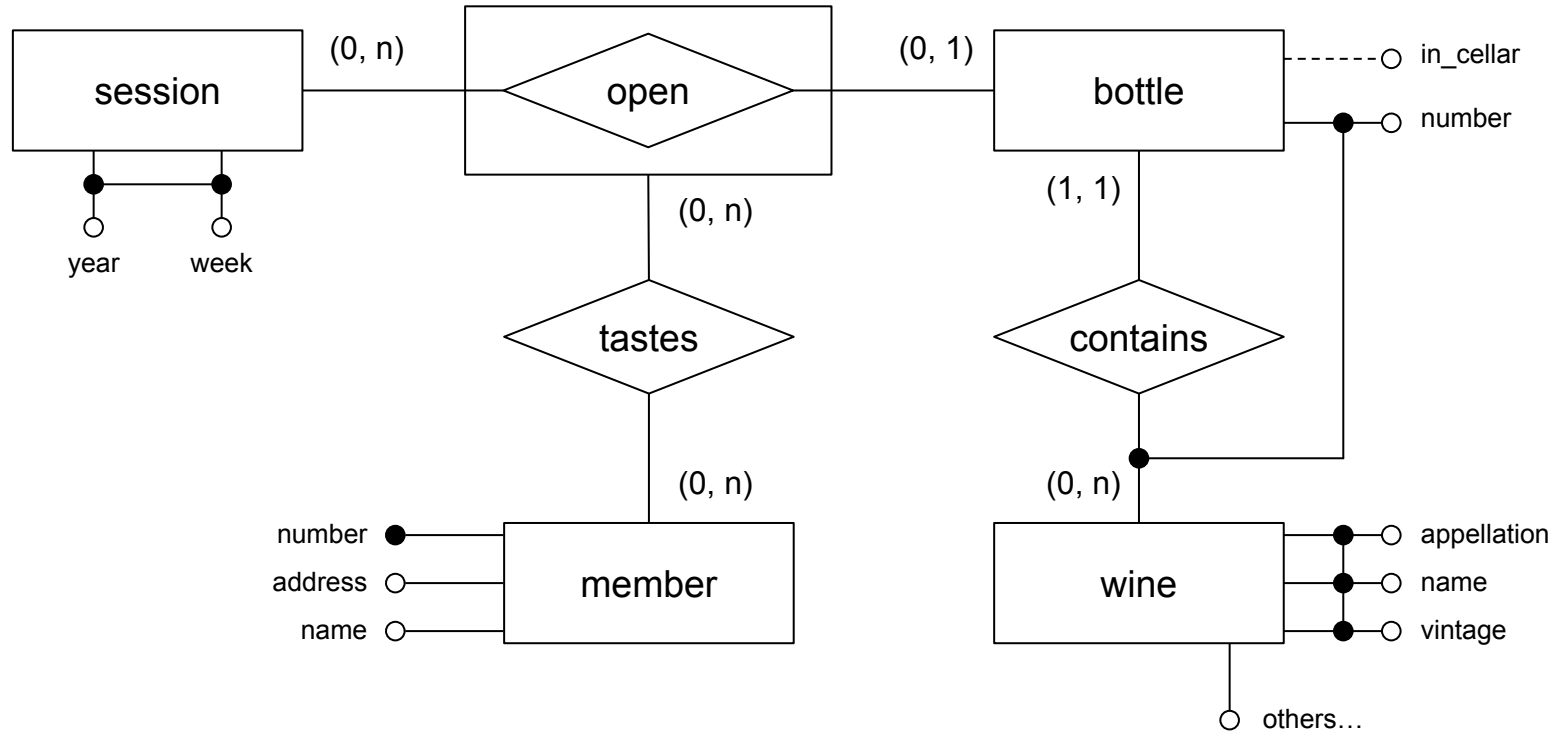


Question 1

(f) Draw the corresponding entity-relationship diagram with the key and participation constraints. Indicate in English the constraints that cannot be captured, if any.



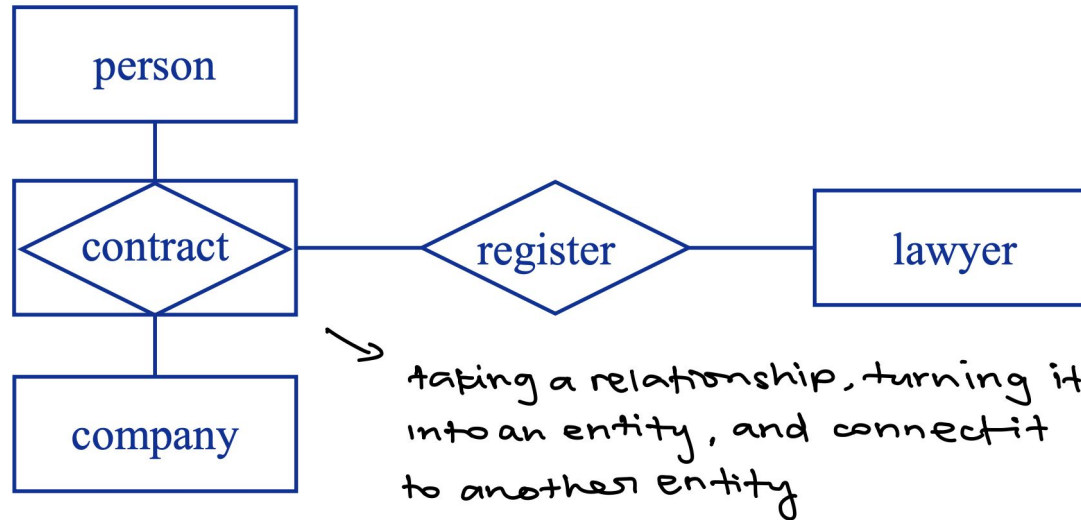
Question 1



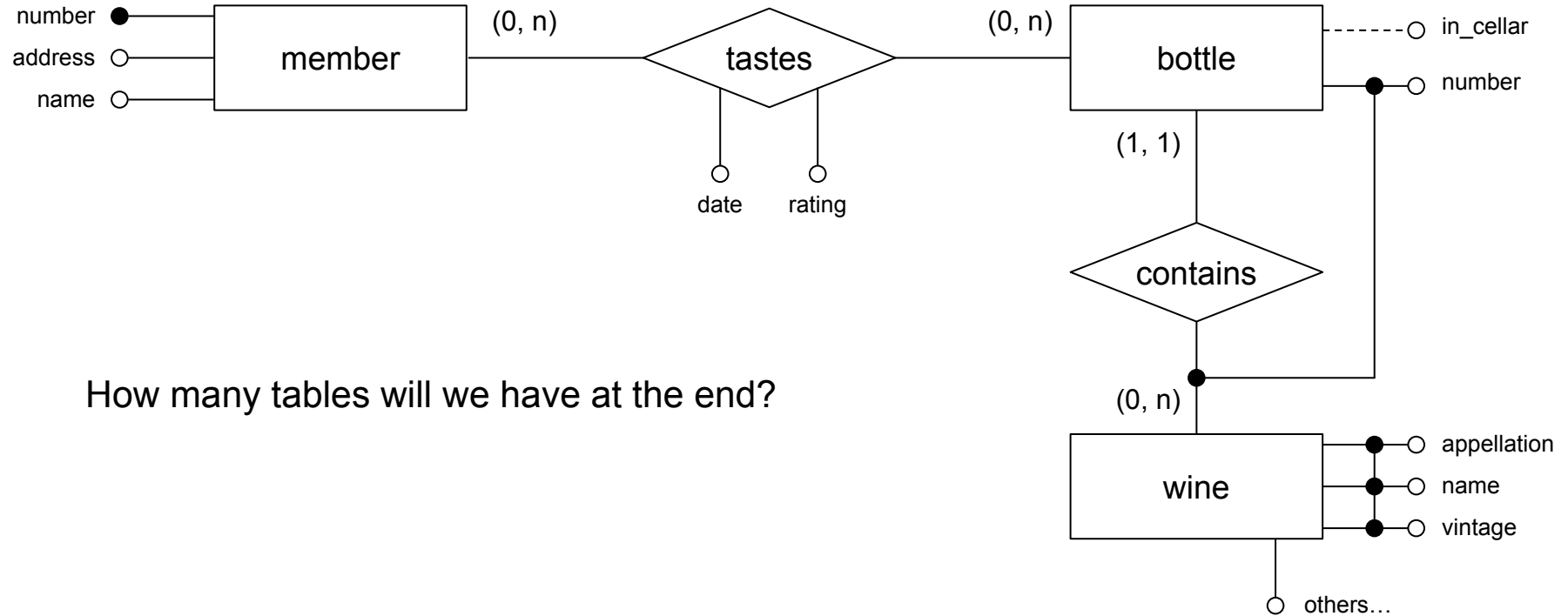
Recap - Aggregation

Taking a relationship, turning it into an entity, and connecting it to another entity

The box around the *contract* diamond should not touch

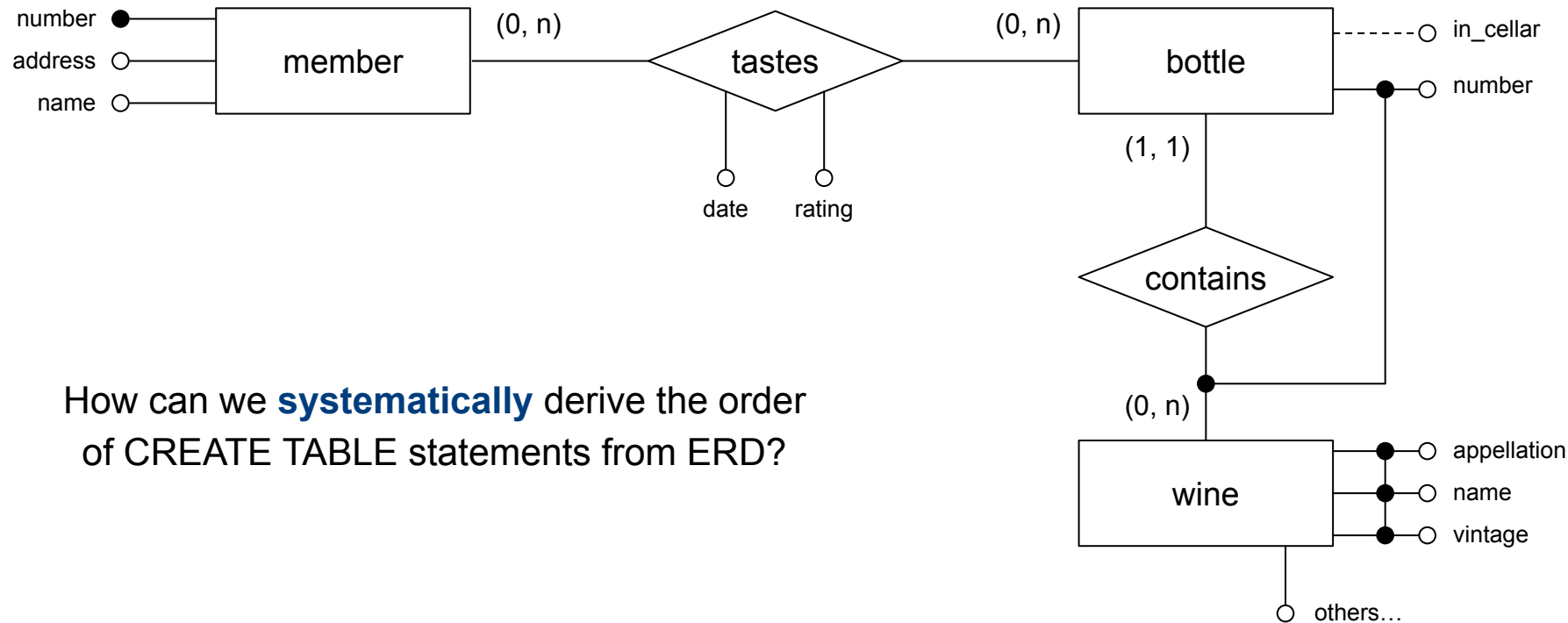


Question 2



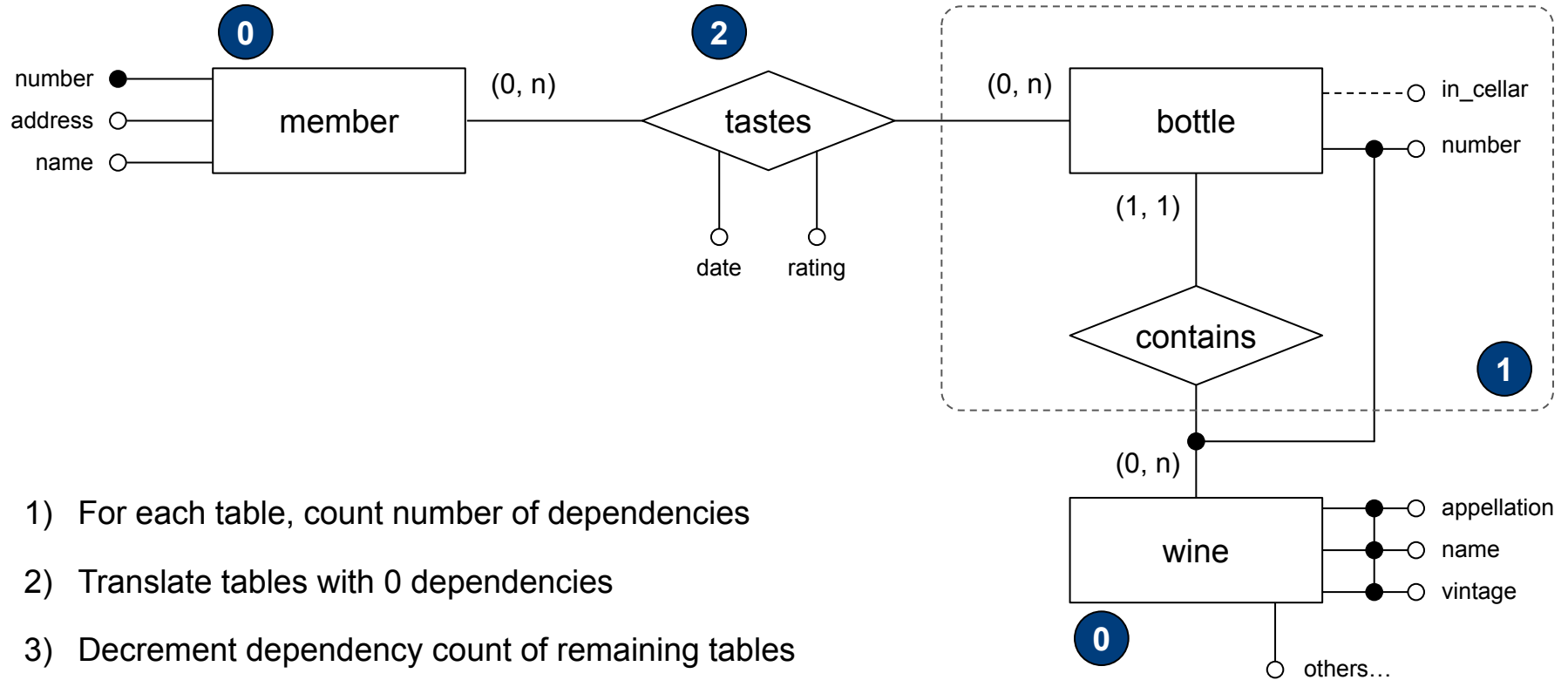
How many tables will we have at the end?

Question 2



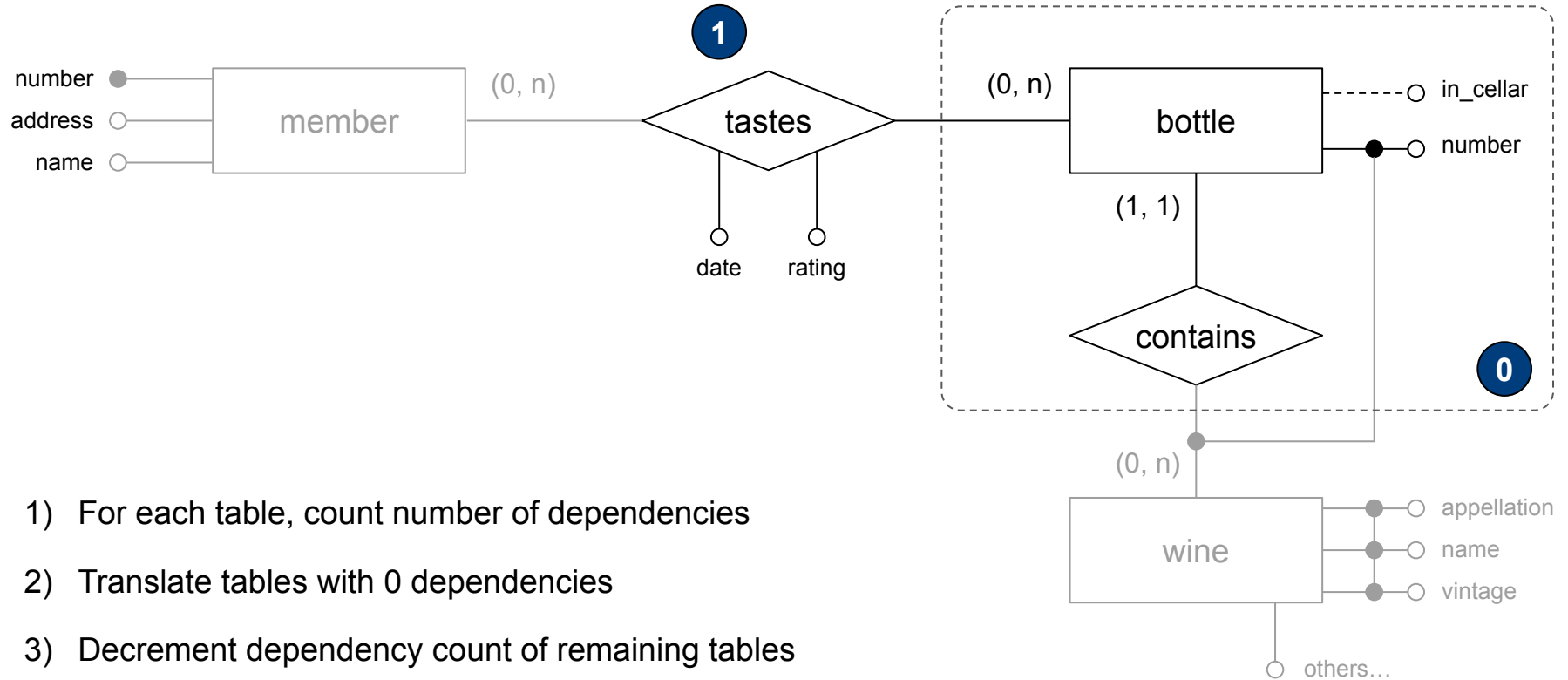
How can we **systematically** derive the order of CREATE TABLE statements from ERD?

Question 2



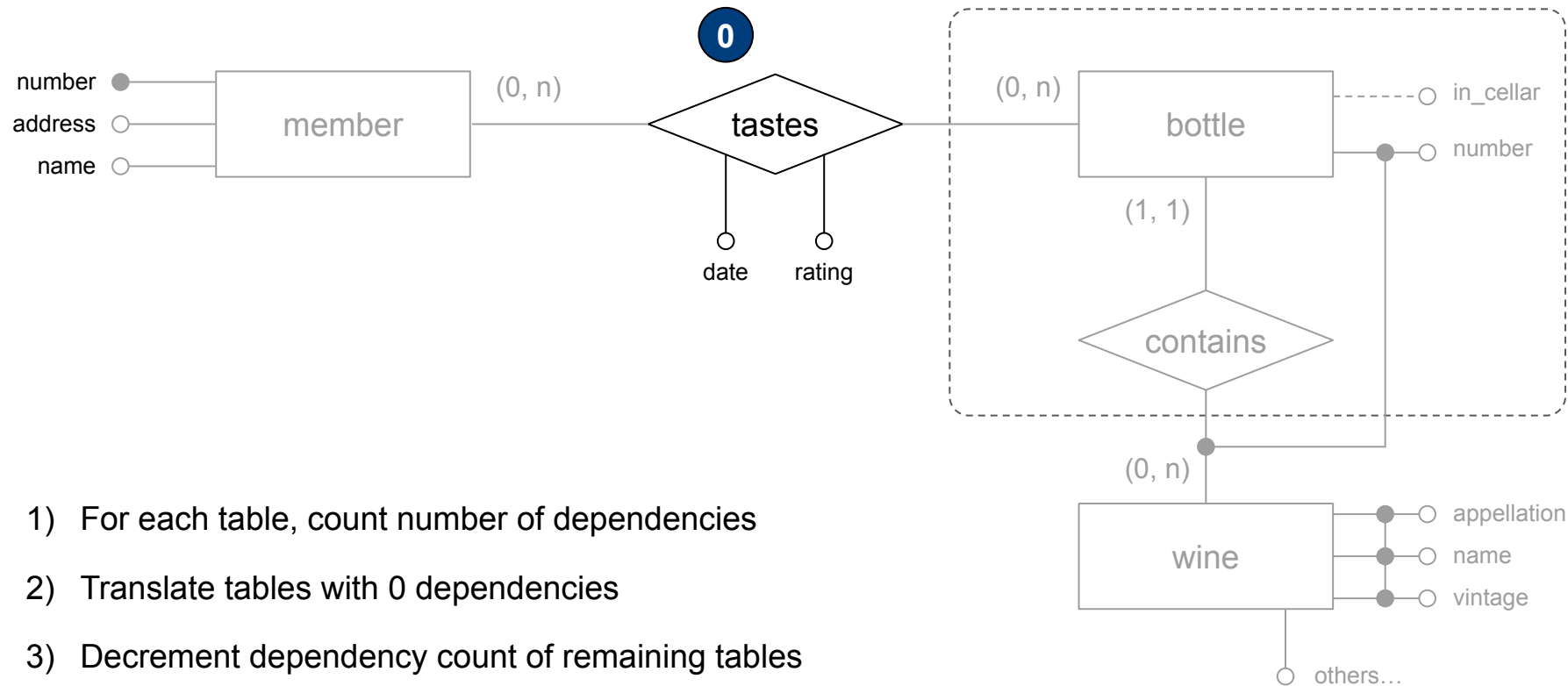
- 1) For each table, count number of dependencies
- 2) Translate tables with 0 dependencies
- 3) Decrement dependency count of remaining tables

Question 2



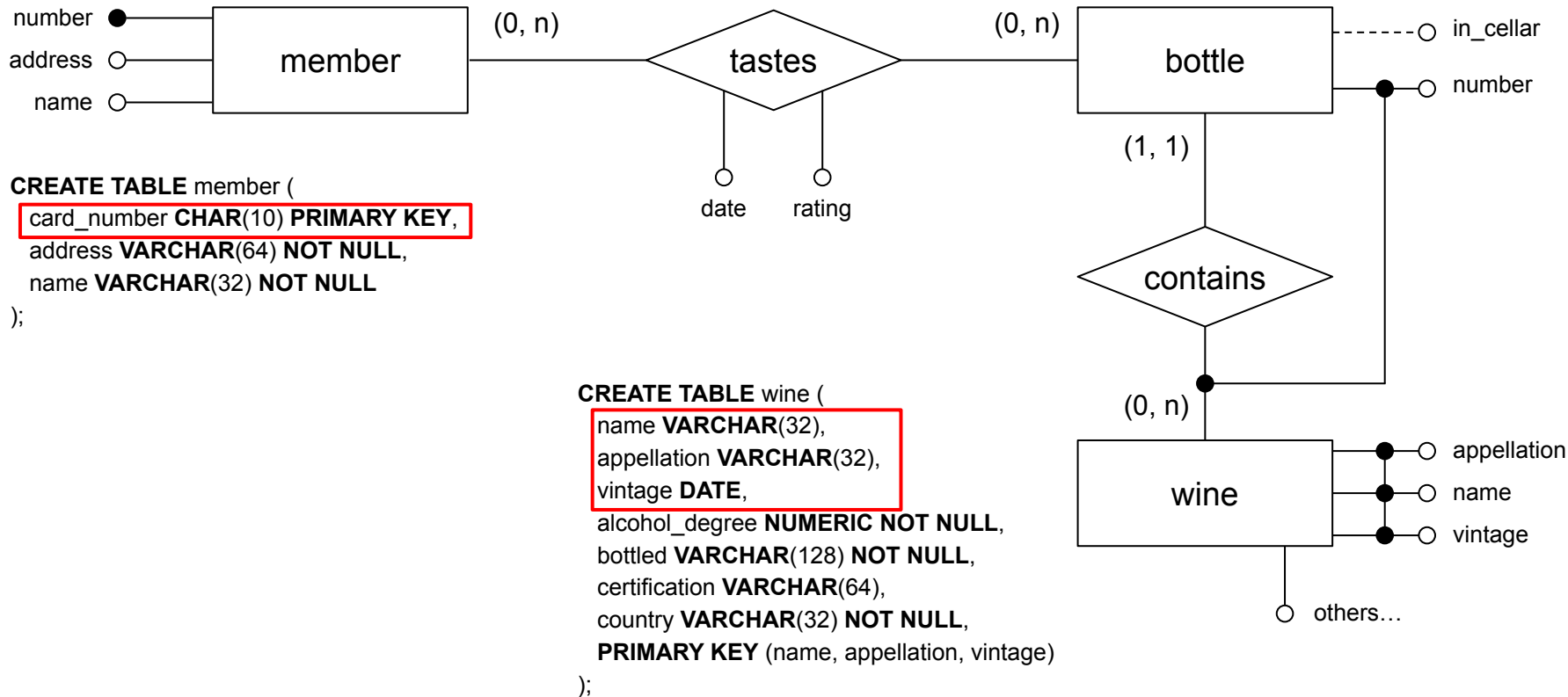
- 1) For each table, count number of dependencies
- 2) Translate tables with 0 dependencies
- 3) Decrement dependency count of remaining tables

Question 2

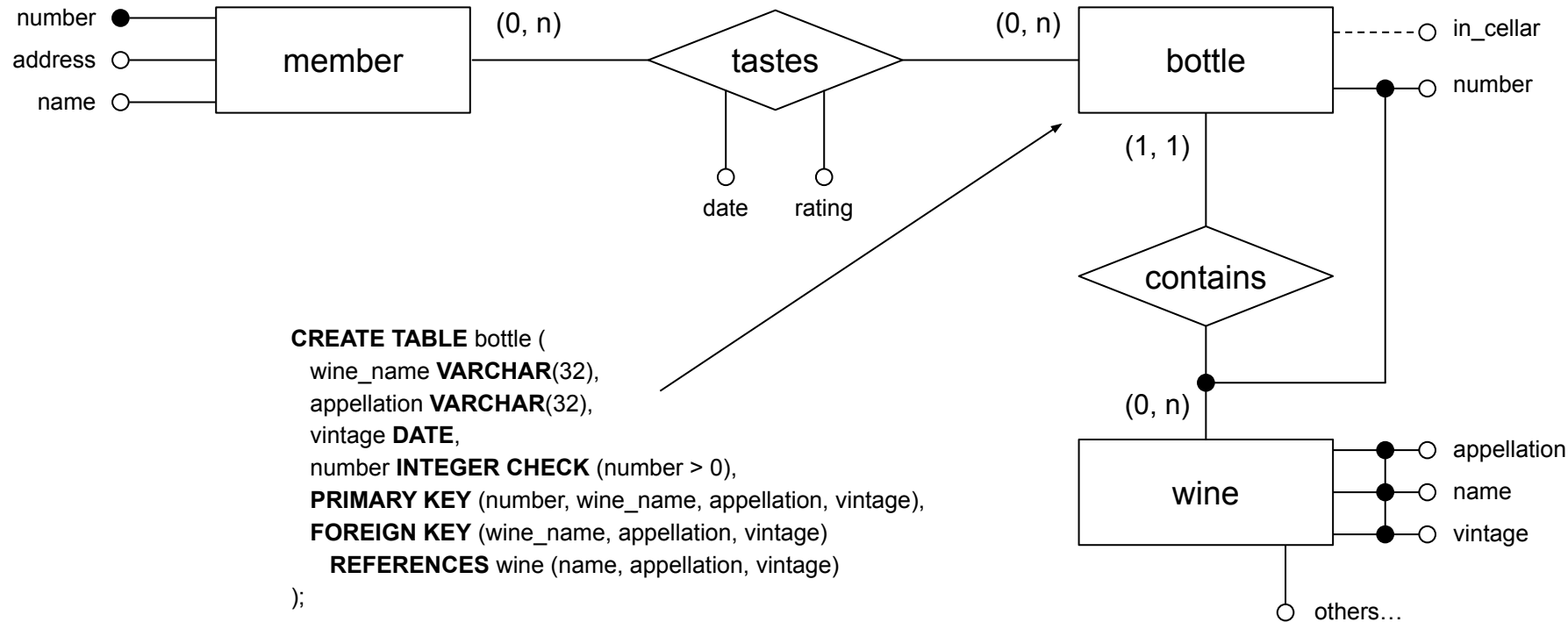


- 1) For each table, count number of dependencies
- 2) Translate tables with 0 dependencies
- 3) Decrement dependency count of remaining tables

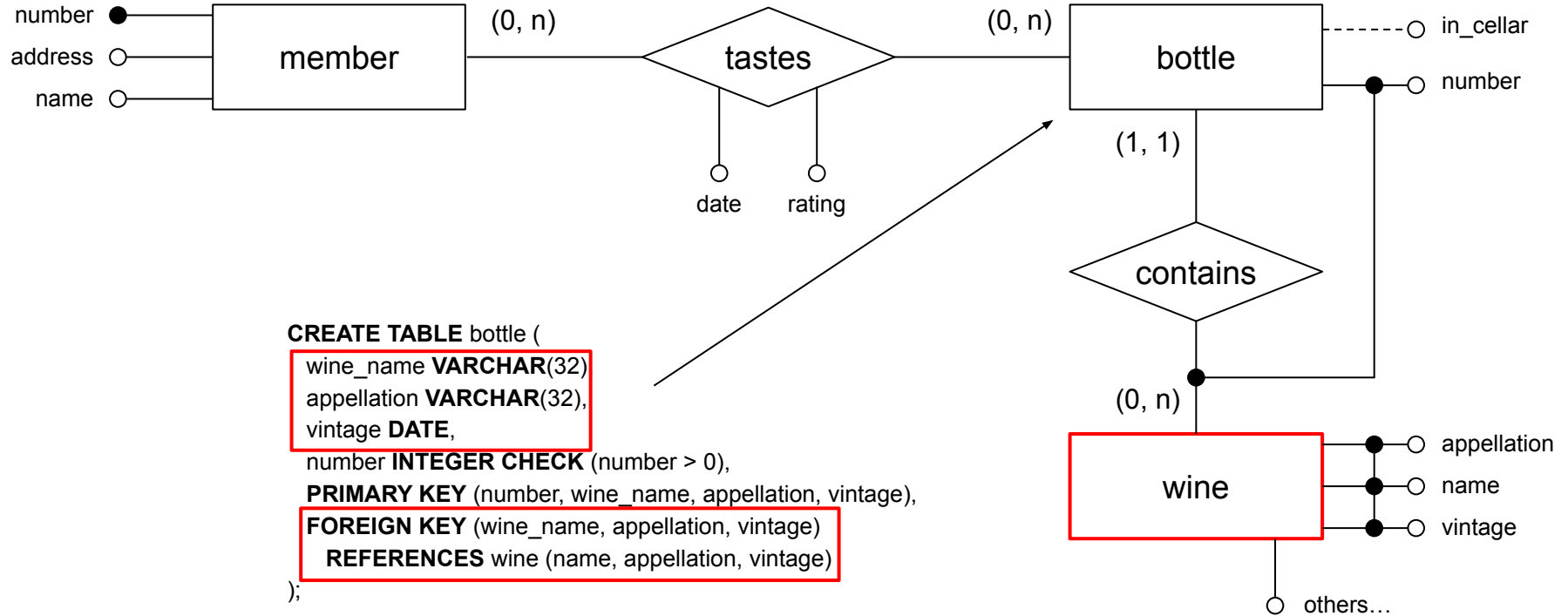
Question 2



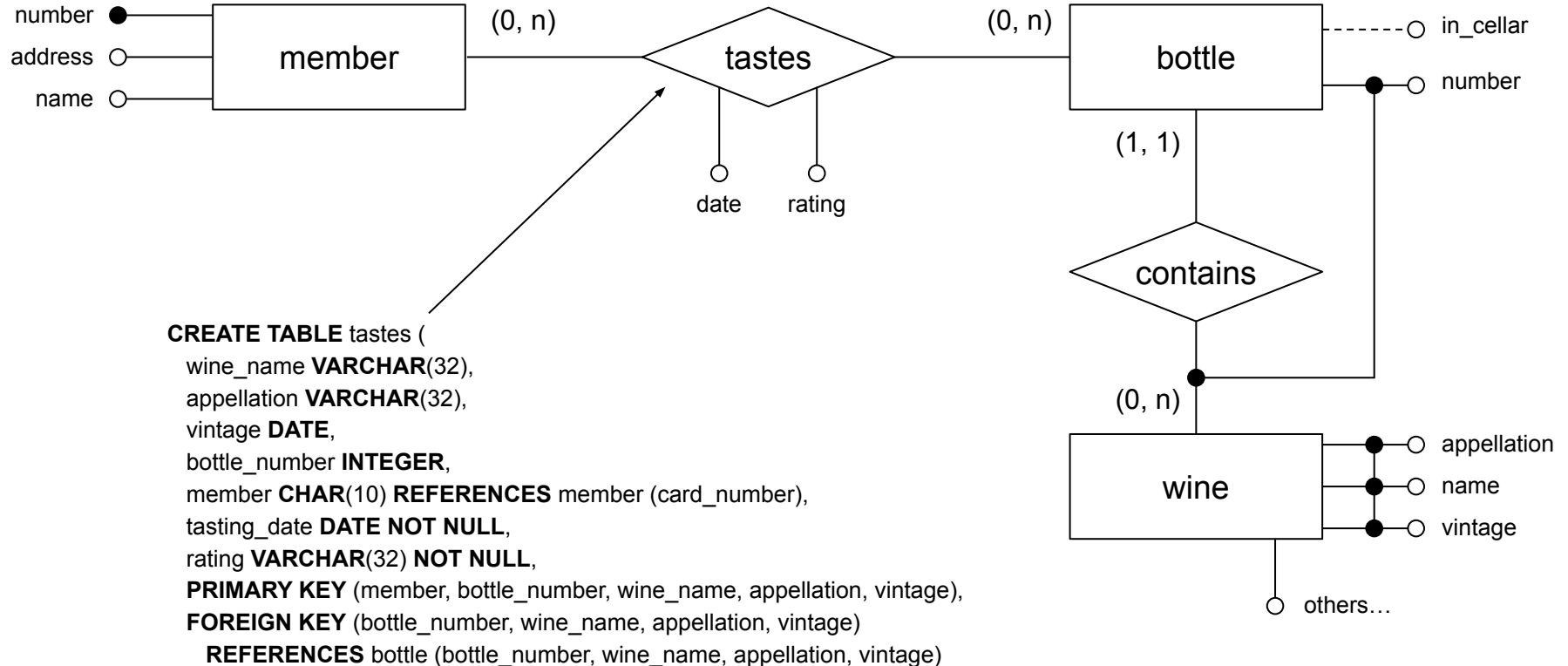
Question 2



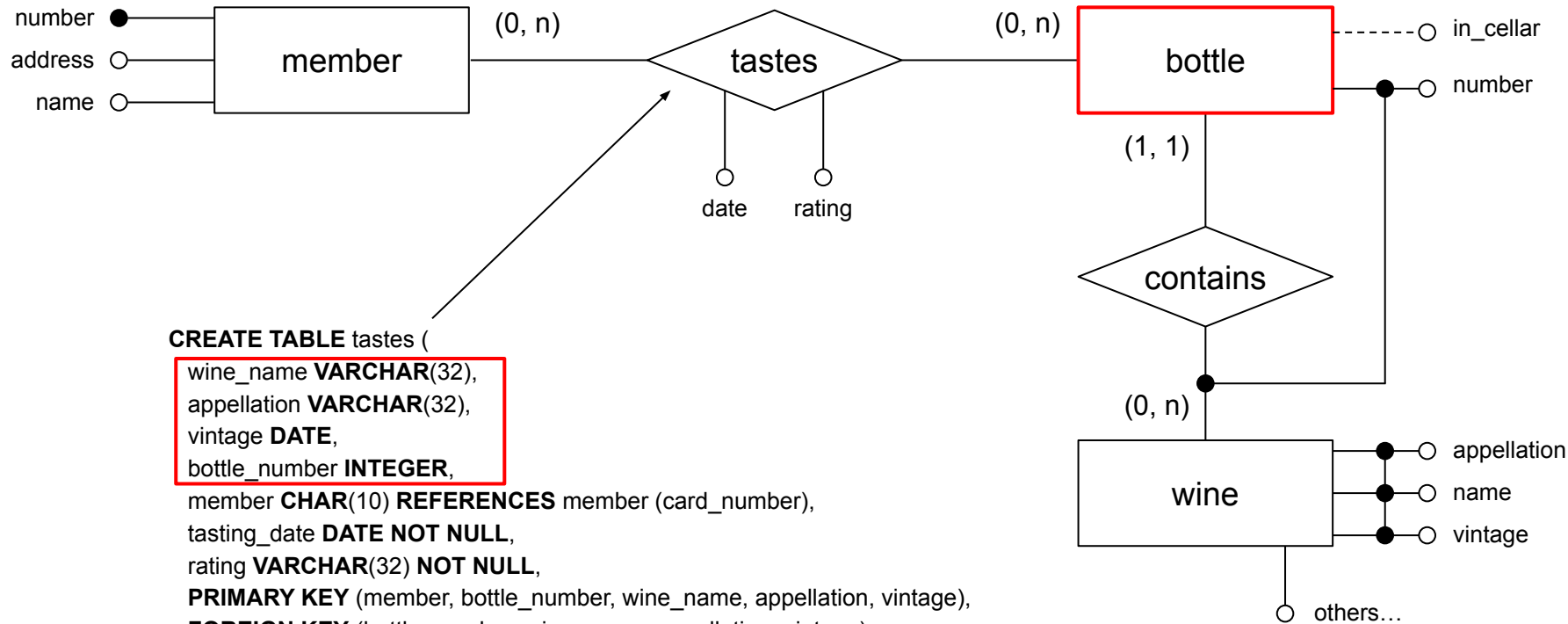
Question 2



Question 2

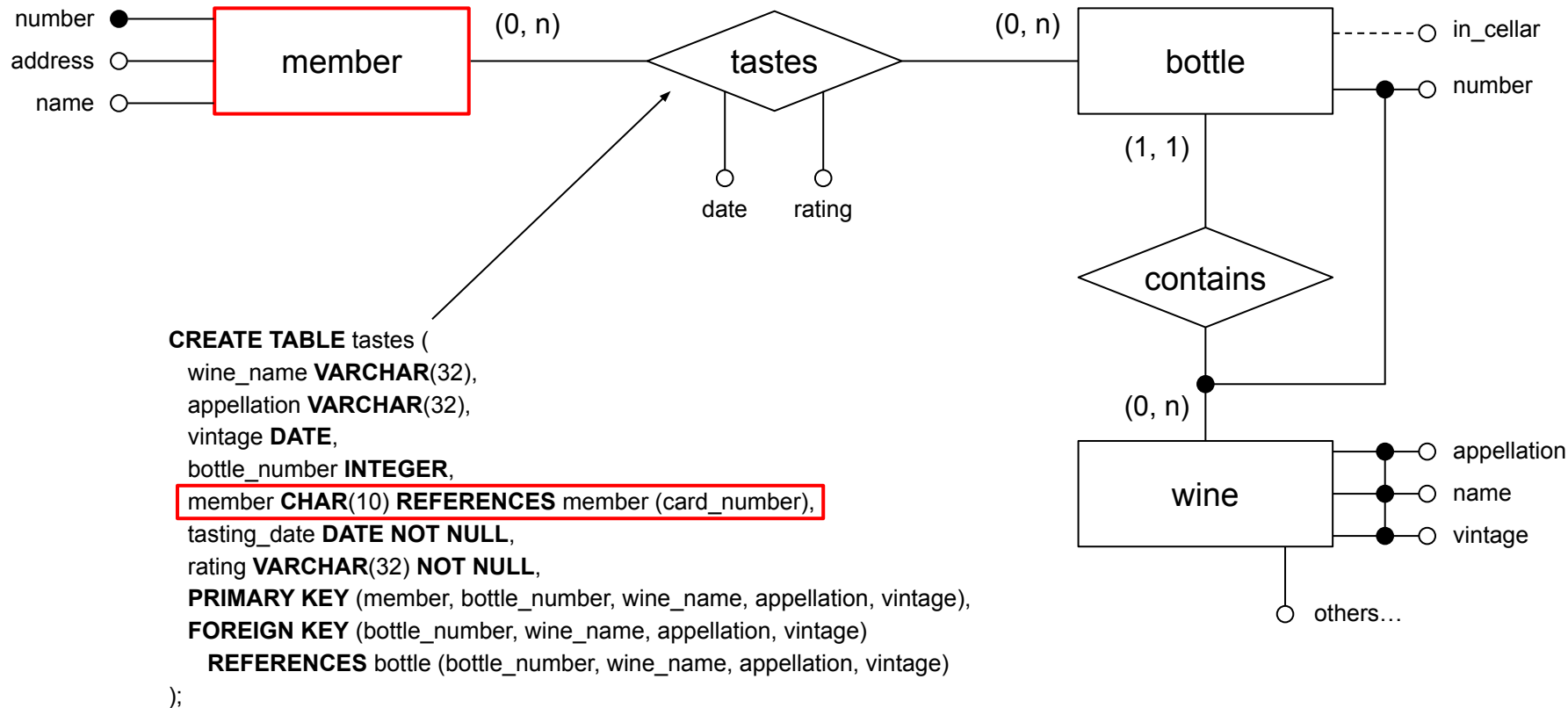


Question 2



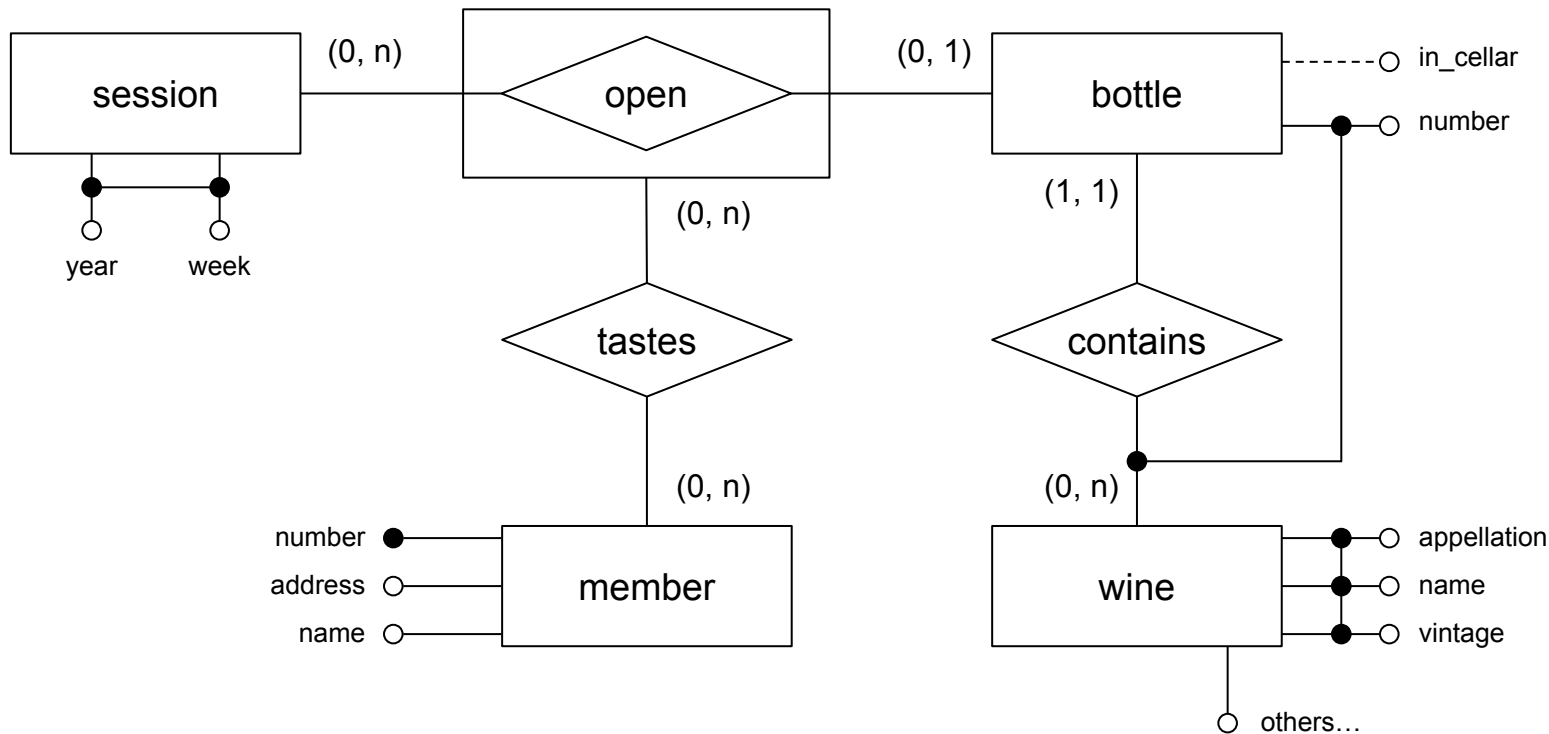
```
CREATE TABLE tastes (  
  wine_name VARCHAR(32),  
  appellation VARCHAR(32),  
  vintage DATE,  
  bottle_number INTEGER,  
  member CHAR(10) REFERENCES member (card_number),  
  tasting_date DATE NOT NULL,  
  rating VARCHAR(32) NOT NULL,  
  PRIMARY KEY (member, bottle_number, wine_name, appellation, vintage),  
  FOREIGN KEY (bottle_number, wine_name, appellation, vintage)  
    REFERENCES bottle (bottle_number, wine_name, appellation, vintage)  
);
```


Question 2



Question 2

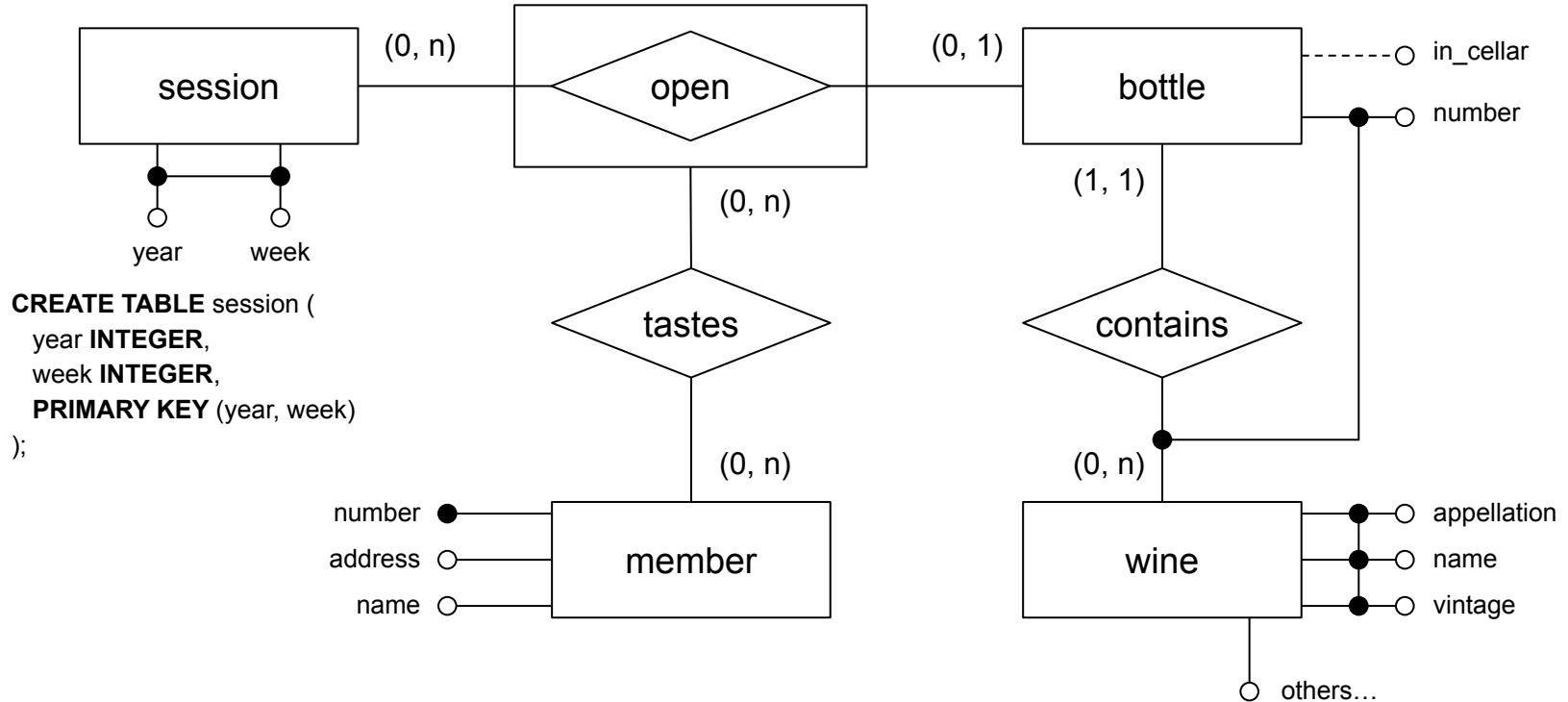
How many tables will we have at the end?



Which tables will be the same as before?

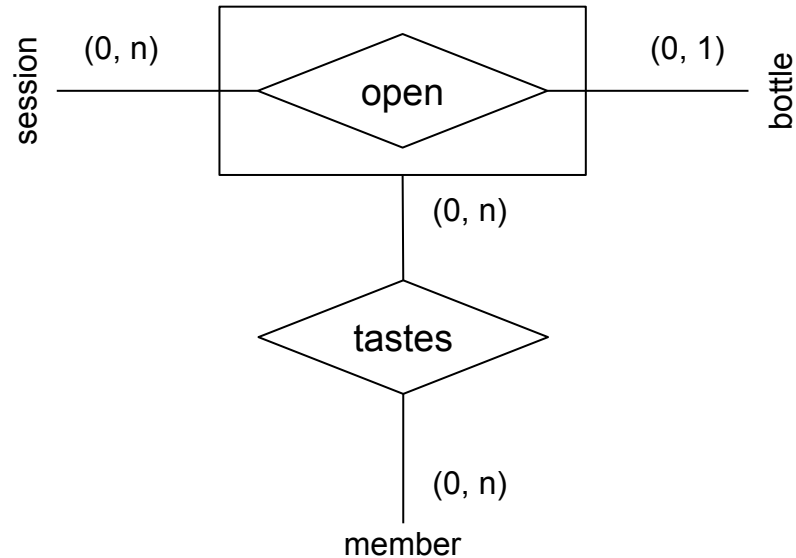
Question 2

How many table will we have at the end?



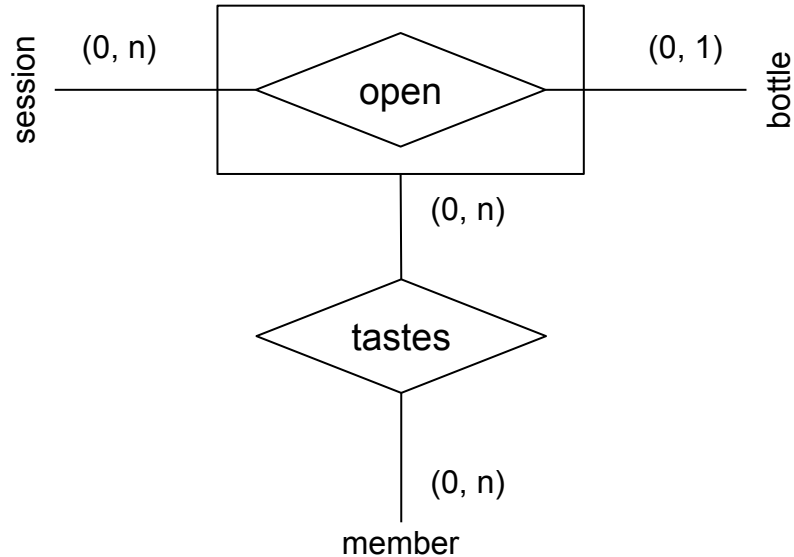
Which tables will be the same as before?

Question 2



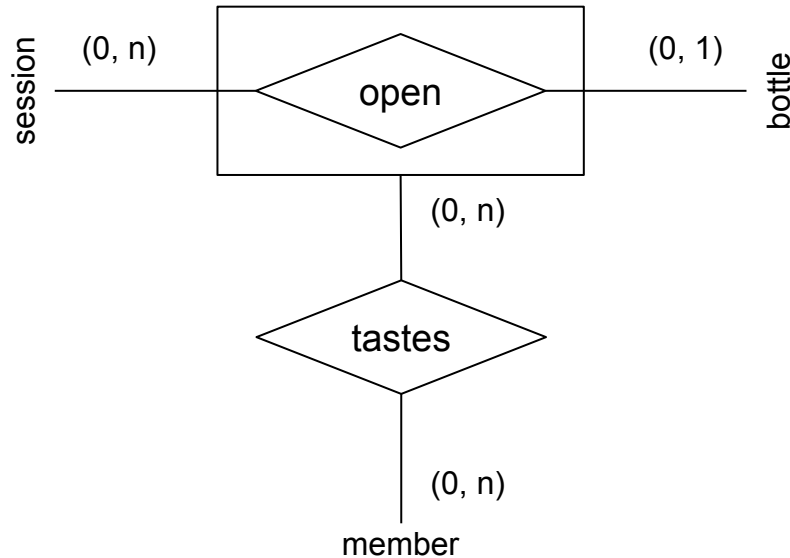
```
CREATE TABLE open (  
  wine_name VARCHAR(32),  
  appellation VARCHAR(32),  
  vintage DATE,  
  bottle_number INTEGER,  
  session_year INTEGER NOT NULL,  
  session_week INTEGER NOT NULL,  
  PRIMARY KEY (bottle_number, wine_name, appellation, vintage),  
  FOREIGN KEY (session_year, session_week)  
    REFERENCES session (year, week),  
  FOREIGN KEY (bottle_number, wine_name, appellation, vintage)  
    REFERENCES bottle (number, wine_name, appellation, vintage)  
)
```

Question 2



```
CREATE TABLE tastes (  
  wine_name VARCHAR(32),  
  appellation VARCHAR(32),  
  vintage DATE,  
  bottle_number INTEGER,  
  member CHAR(10) REFERENCES member (card_number),  
  rating VARCHAR(32) NOT NULL,  
  PRIMARY KEY (member, bottle_number, wine_name, appellation, vintage),  
  FOREIGN KEY (bottle_number, wine_name, appellation, vintage)  
    REFERENCES open (bottle_number, wine_name, appellation, vintage)  
);
```

Question 2



```
CREATE TABLE tastes (  
  wine_name VARCHAR(32),  
  appellation VARCHAR(32),  
  vintage DATE,  
  bottle_number INTEGER,  
  member CHAR(10) REFERENCES member (card_number),  
  rating VARCHAR(32) NOT NULL,  
  PRIMARY KEY (member, bottle_number, wine_name, appellation, vintage),  
  FOREIGN KEY (bottle_number, wine_name, appellation, vintage)  
    REFERENCES open (bottle_number, wine_name, appellation, vintage)  
);
```

```
CREATE TABLE tastes (  
  wine_name VARCHAR(32),  
  appellation VARCHAR(32),  
  vintage DATE,  
  bottle_number INTEGER,  
  member CHAR(10) REFERENCES member (card_number),  
  tasting_date DATE NOT NULL,  
  rating VARCHAR(32) NOT NULL,  
  PRIMARY KEY (member, bottle_number, wine_name, appellation, vintage),  
  FOREIGN KEY (bottle_number, wine_name, appellation, vintage)  
    REFERENCES bottle (bottle_number, wine_name, appellation, vintage)  
);
```

See you next week!