Name

CWID

Homework Assignment 3

November 8th, 2019

Due on November 18th, 2019 11:59pm (CST)

CS425 - Database Organization Results

Please leave this empty!

3.1 | 3.2 | 3.3 | Sum

Instructions

- Try to answer all the questions using what you have learned in class

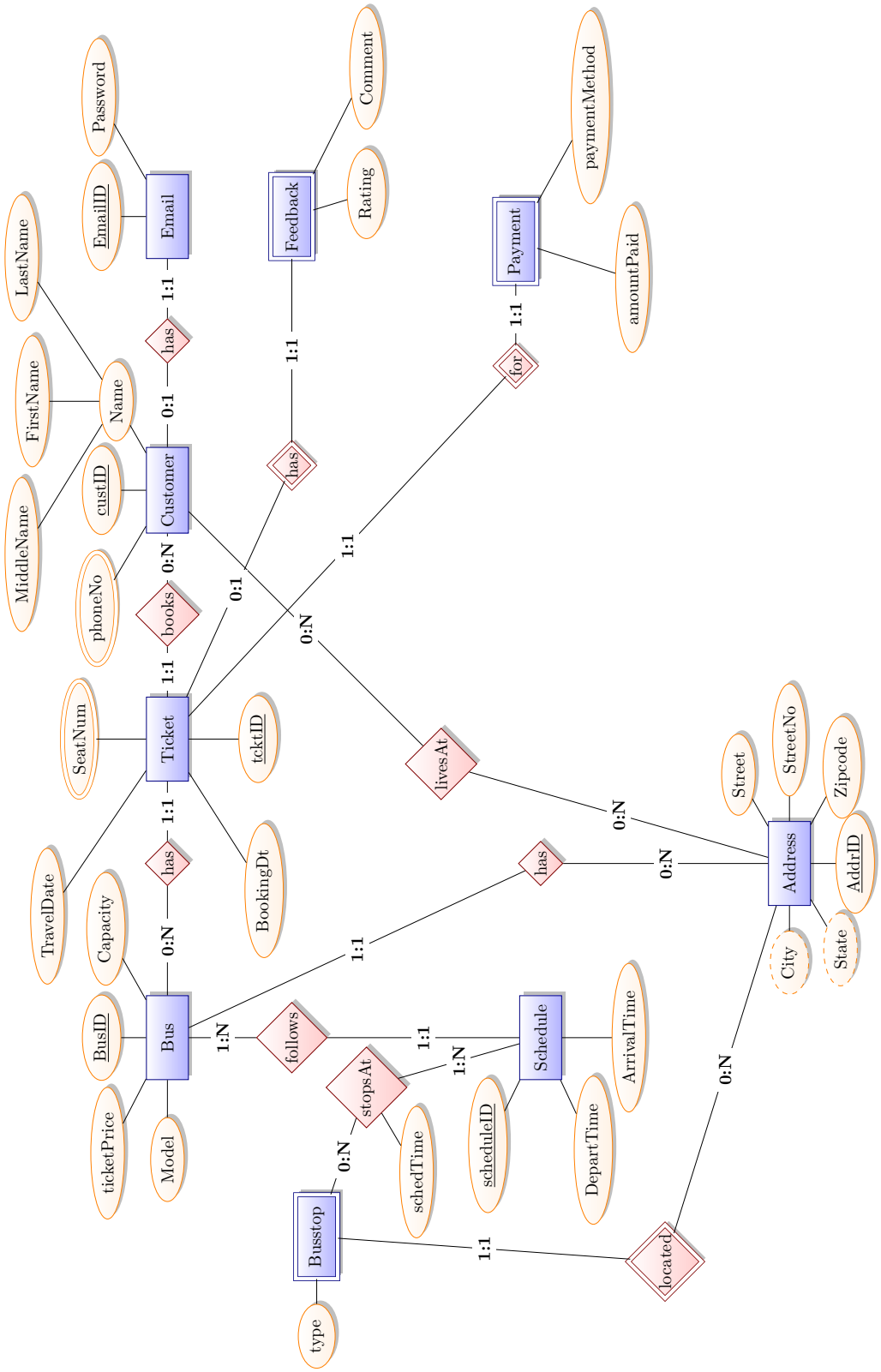
Part 3.1 Modeling (Total: 35 Points)

Question 3.1.1 (35 Points)

Build a conceptional model for an **Online Bus Booking System**. The solution should be presented as an **ER-diagram**. Design your ER diagram with the following requirements.

- The database should record information about Customers, Emails, Addresses, Tickets, Buses, Bus Schedules, Bus Stops, Purchase Feedback, and Payments.
- A **Customer** has a name which consists of *firstName*, *middleName* and *lastName*. **Customers** are identified by a unique *custID*. A **Customer** can have one or more *phoneNumbers*.
  - **Customers** can book any number of *Tickets* (including none). **Customers** may provide *Feedback* for each booking (optional). For every booking, a **Customer** has to make a *Payment*.
- An **Address** consists of a unique *addrID*, *street*, *streetNumber*, *city*, *state* and *zipcode*. The attributes *city* and *state* can be derived from the attribute *zipcode*.
  - A **Customer** can be associated with any number of **Addresses** and there may be multiple **Customers** living at the same **Address**. There may be some **Addresses** which are not be associated with any **Customer**.
- An **Email** consists of unique *emailID* and a *Password*.
  - A *Customer* may or may not have an **Email** and every **Email** belongs to a single owner (customer). We assume that a **Customer** can have only one **Email**.
- A **Bus** is identified using a *busNumber*. A **Bus** has a *capacity, model* and a *ticketPrice*.
  - A **Bus** may follow several **Schedules**.
- A **Bus Stop** is uniquely identified by the *Address* it is located at. A **Bus Stop** has a *type* (either *sheltered* or *simple*).
- A **Schedule** includes an *ArrivalTime*, **DepartureTime**, and a unique *scheduleID*.
  - A **Schedule** is associated with one or more **Bus Stops**. For each association, we record the *scheduled time* of the bus stopping at this **Bus Stop**.
- A **Ticket** has its unique *ticketId*. Tickets also have a *bookingDate* and *travelDate*. A **Ticket** may have one or more *seatNumbers*.
  - Each **Ticket** may be associated with a single **Feedback**.
  - Every **Ticket** has a **Bus** associated with it, while a **Bus** may have many different **Tickets** associated with itself.
- A **Payment** is identified by the *Ticket* for which the payment was made. It consists of the *amountPaid* and *paymentMethod* (Credit Card, E-Check, etc.)
- Every **Purchase Feedback** is uniquely identified by the **Ticket** for which the feedback is given. For each **Purchase Feedback** we store a *rating* and *comment*.

Solution

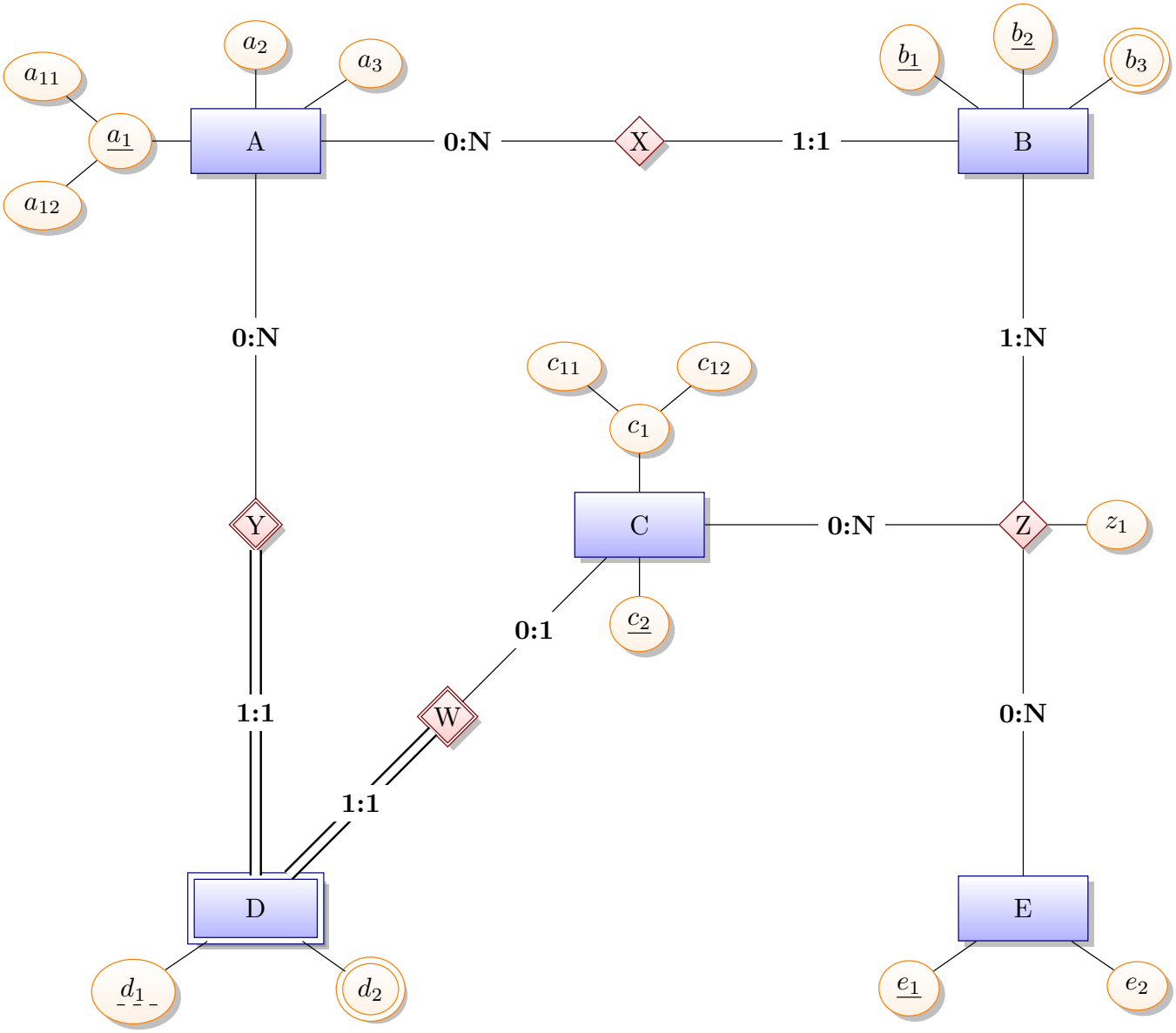

Password
EmailID
Email
LastName
FirstName
Name
MiddleName
custID
Customer
phoneNo
Comment
Rating
Feedback
paymentMethod
amountPaid
Payment
SeatNum
Ticket
tcktID
BookingDt
TravelDate
Capacity
BusID
Bus
ticketPrice
Model
type
Busstop
schedTime
scheduleID
DepartTime
ArrivalTime
Schedule
Street
StreetNo
Zipcode
AddrID
City
State
Address
books
has
for
follows
stopsAt
livesAt
located
0:N
1:1
0:1
1:N

Part 3.2 Translation of ER into Relational Model (Total: 35 Points)

Question 3.2.1 (35 Points)

Take the following ER-model and translate it into a relational schema using the rules presented in class. Present the relational schema as an SQL script (assume that all attributes are of data type `INT`). Present the results of the following intermediate steps in this order:

1. Translate strong entities + unnest composite attributes
2. Translate weak entities
3. Translated multi-valued attributes
4. Translate relationships



Solution

1st Step (strong entities)

```
CREATE TABLE A (
   a11 INT,
   a12 INT,
   a2 INT,
   a3 INT,
   PRIMARY KEY(a11, a12)
);

CREATE TABLE B (
   b1 INT,
   b2 INT,
   PRIMARY KEY (b1,b2)
);

CREATE TABLE C (
   c2 INT PRIMARY KEY,
   c11 INT,
   c12 INT
);

CREATE TABLE E (
   e1 INT PRIMARY KEY,
   e2 INT
);
```

2nd Step (weak entities)

```
CREATE TABLE A (
   a11 INT,
   a12 INT,
   a2 INT,
   a3 INT,
   PRIMARY KEY(a11, a12)
);

CREATE TABLE B (
   b1 INT,
   b2 INT,
   PRIMARY KEY (b1, b2)
);

CREATE TABLE C (
   c2 INT PRIMARY KEY,
   c11 INT,
   c12 INT
);

CREATE TABLE D (
   d1 INT,
   PRIMARY KEY (a11, a12, c2, d1),
   FOREIGN KEY a11, a12 REFERENCES A,
   FOREIGN KEY c2 REFERENCES C
);

CREATE TABLE E (
   e1 INT PRIMARY KEY,
   e2 INT
);
```

3rd Step (multivalued attributes)

```
CREATE TABLE A (
   a11 INT,
   a12 INT,
   a2 INT,
   a3 INT,
   PRIMARY KEY(a11, a12)
);

CREATE TABLE B (
   b1 INT,
   b2 INT,
   PRIMARY KEY (b1, b2)
);

CREATE TABLE B3 (
   b1 INT,
   b2 INT,
   b3 INT,
   FOREIGN KEY (b1, b2) REFERENCES B,
   PRIMARY KEY (b1, b2, b3)
);

CREATE TABLE C (
   c2 INT PRIMARY KEY,
   c11 INT,
   c12 INT
);

CREATE TABLE D (
   d1 INT,
   PRIMARY KEY (a11, a12, c2, d1),
   FOREIGN KEY a11, a12 REFERENCES A,
   FOREIGN KEY c2 REFERENCES C
);

CREATE TABLE D2 (
   d1 INT,
   d2 INT,
   PRIMARY KEY (a11, a12, c2, d1, d2),
   FOREIGN KEY a11, a12 REFERENCES A,
   FOREIGN KEY c2 REFERENCES C
);

CREATE TABLE E (
   e1 INT PRIMARY KEY,
   e2 INT
);
```

4th Step (relationships)

```
CREATE TABLE A (
   a11 INT,
   a12 INT,
   a2 INT,
   a3 INT,
   PRIMARY KEY(a11, a12)
);

CREATE TABLE B (
   b1 INT,
   b2 INT,
   a11 INT,
   a12 INT,
   PRIMARY KEY (b1, b2),
   FOREIGN KEY (a11, a12) REFERENCES A,
);

CREATE TABLE B3 (
   b1 INT,
   b2 INT,
   b3 INT,
   FOREIGN KEY (b1, b2) REFERENCES B,
   PRIMARY KEY (b1, b2, b3)
);

CREATE TABLE C (
   c2 INT PRIMARY KEY,
   c11 INT,
   c12 INT
);

CREATE TABLE D (
   d1 INT,
   PRIMARY KEY (a11, a12, c2, d1),
   FOREIGN KEY a11, a12 REFERENCES A,
   FOREIGN KEY c2 REFERENCES C
);

CREATE TABLE D2 (
   d1 INT,
   d2 INT,
   PRIMARY KEY (a11, a12, c2, d1, d2),
   FOREIGN KEY a11, a12 REFERENCES A,
   FOREIGN KEY c2 REFERENCES C
);

CREATE TABLE E (
   e1 INT PRIMARY KEY,
   e2 INT
);

CREATE TABLE Z (
   b1 INT,
```

```
    b2 INT,
    c2 INT,
    e1 INT,
    z1 INT,
    PRIMARY KEY (b1, b2, c2, e1),
    FOREIGN KEY (b1, b2) REFERENCES B,
    FOREIGN KEY (c2) REFERENCES C,
    FOREIGN KEY (e1) REFERENCES E
);
```

Part 3.3 Normalization (Total: 30 Points)

Question 3.3.1 (30 Points)

Considering the following relations, determine the candidate key(s) and normal form for each relation (note that a relation can be in multiple normal forms). Please consider the following normal forms: 1NF, 2NF, 3NF and BCNF.

1. $R(A, B, C, D)$ and the Functional Dependencies are $C \rightarrow B, B \rightarrow D, A \rightarrow BD$
2. $R(A, B, C, D, E)$ and the Functional Dependencies are $A \rightarrow B, B \rightarrow C, AC \rightarrow D$
3. $R(A, B, C, D, E)$ and the Functional Dependencies are $A \rightarrow BC, D \rightarrow E, AB \rightarrow CD$
4. $R(A, B, C, D, E)$ and the Functional Dependencies are $A \rightarrow C, B \rightarrow C, AB \rightarrow DE, BC \rightarrow AD$
5. $R(A, B, C, D, E)$ and the Functional Dependencies are $AC \rightarrow BE, A \rightarrow D, B \rightarrow A, BC \rightarrow D$
6. $R(A, B, C, D)$ and the Functional Dependencies are $C \rightarrow B, B \rightarrow AC, A \rightarrow BD$

Solution

1. Candidate key(s) is AC. The relation is in 1NF only.
2. Candidate key(s) is AE. The relation is in 1NF only.
3. Candidate key(s) is A. The relation is in 1NF and 2NF.
4. Candidate key(s) is B. The relation is in 1NF and 2NF.
5. Candidate key(s) is AC. The relation is in 1NF only.
6. Candidate key(s) is A or B or C. The relation is in 1NF, 2NF, 3NF and BCNF.