

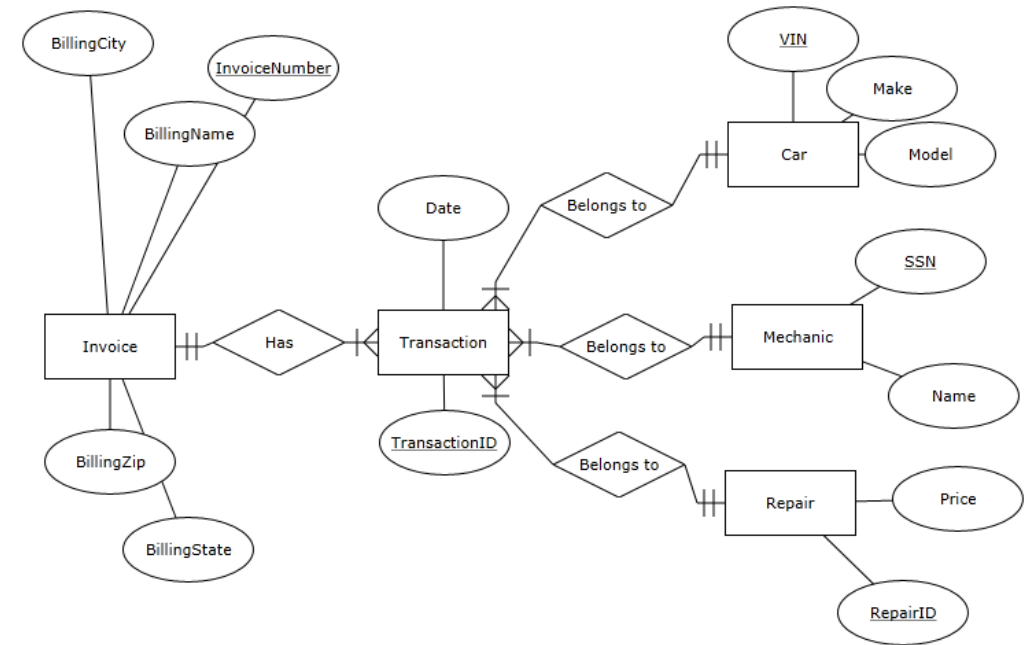
# Common Mistakes

- Fail to identify all entities
  - Some relationships between entities are missing
  - Some attributes are misplaced
- Forget to include a primary key
  - No primary key at all
  - Use “first name, last name” as primary key
- Cardinality
  - Do not specify cardinality at all
  - Only specify maximum cardinality
  - Specify the cardinality incorrectly (most tricky)

# Assignment Instructions

- **Read Instructions carefully**

- Create ER Model (namely ERD)
- Primary key
- 1:1, 1:m, m:m (maximum cardinality)
  - ✓ Do we need to specify minimum cardinality?
- Two types of attributes: entity attributes and relationship attributes



Create an ER model for each scenario. Make sure that you read the description carefully. Your diagram should reflect all entities, attributes, and relationships in the description. You should make sure each entity has a primary key (a unique identifier). Use the relationship types we used in class (one-to-one, one-to-many, or many-to-many). Don't forget, attributes can describe both entities and relationships.

# Assignment Instructions – Continued

- How to draw ERD?
  - Use ERDPlus
- How to submit your assignment?
  - Submit electronically via blackboard
  - Include your name at the top of a word document

You must submit your diagrams electronically, and they cannot be hand-drawn. Use ERDPlus to create your diagrams (<http://www.erdplus.com>). It's free, and there is a short YouTube tutorial on the site that shows you how to export the diagrams as a graphic which can be placed into a Word document.

# Scenario 1: Park (1)

- What is purpose of this database?
  - Provide periodic safety reports
  - Be ready for “surprise” audits
- What are the entities?
  - Ride
  - Inspection
  - Incidents
- Why don't we include those nouns safety reports, information record?

**P1**

Schufflyland Park is a medium-sized amusement park in suburban Philadelphia. The park is required to provide periodic *safety reports* to the Commonwealth of Pennsylvania certifying its rides have passed *inspection*. They must also keep a *record* of all safety-related incidents (such as accidents) for each ride. This *information* also has to be ready for “surprise” audits that can occur any time.

# Scenario 1: Park (2)

- Ride
  - Name, Type, Ride ID
- Operator
  - First name, Last Name, Operator ID
- Incident
  - Date, Time, Description, Incident ID

P2

An incident involves both a ride and an operator. All operators and rides are included in the database whether or not they've been involved in a safety incident, and the date and shift are recorded each time an operator runs a ride at the park. A ride is described by a name and type. An operator is described by their first and last name. When an incident occurs, its date and time are recorded along with a description of the incident.

# Scenario 1: Park (3)

- Certification

- Title, Certification ID

- What do we learn from this paragraph except entity certification?

- ✓ Sentences in green?

- ✓ And the rest?

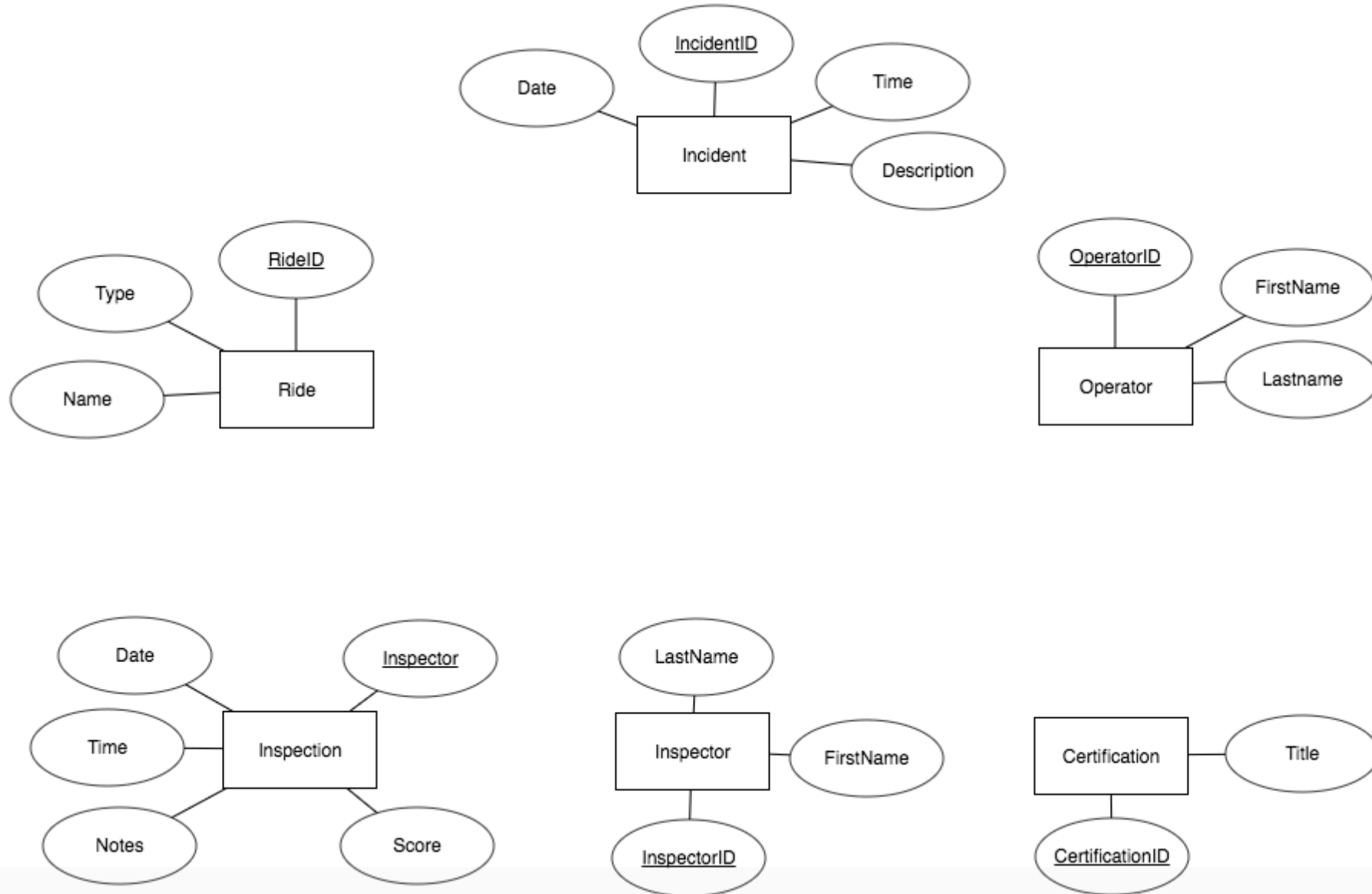
**P3** Operators must be trained and certified on the rides they operate. An operator can be certified for multiple rides, and the certification is valid for two years from the date they complete their training. The title of the certification indicates the ride for which they are certified. When an incident occurs, it is important that the park can produce a list of current certifications for the operator.

# Scenario 1: Park (4)

- Inspection
  - Date, Time, Score, Notes, Inspection ID
- Inspector
  - First Name, Last Name, Inspector ID

**P4** **Inspection** records are also recorded in the database. An inspection occurs at a particular date and time, with a numeric score from 1 to 5, and notes about the outcome of the inspection. **There is only one ride per inspection, even if multiple inspections are done by the same inspector** on the same day and time. The information that needs to be recorded about the **inspector** is their first and last name.

# Draw all entities first

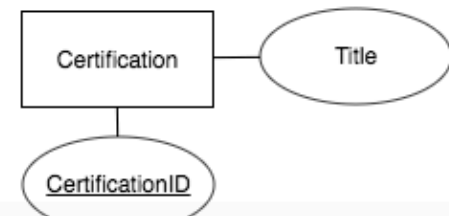
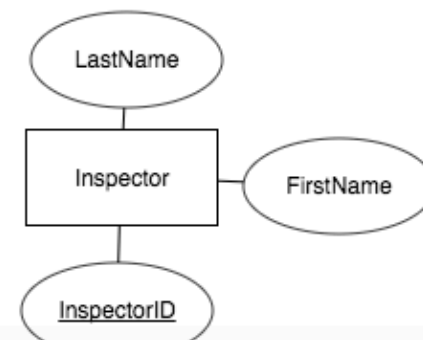
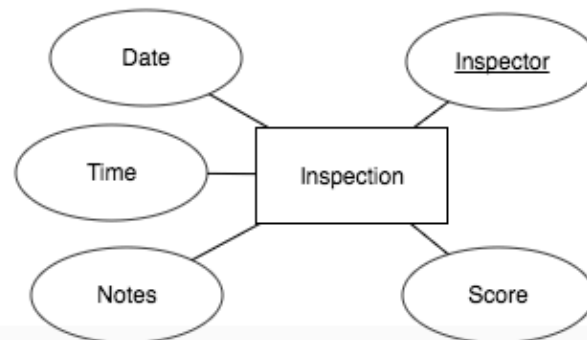
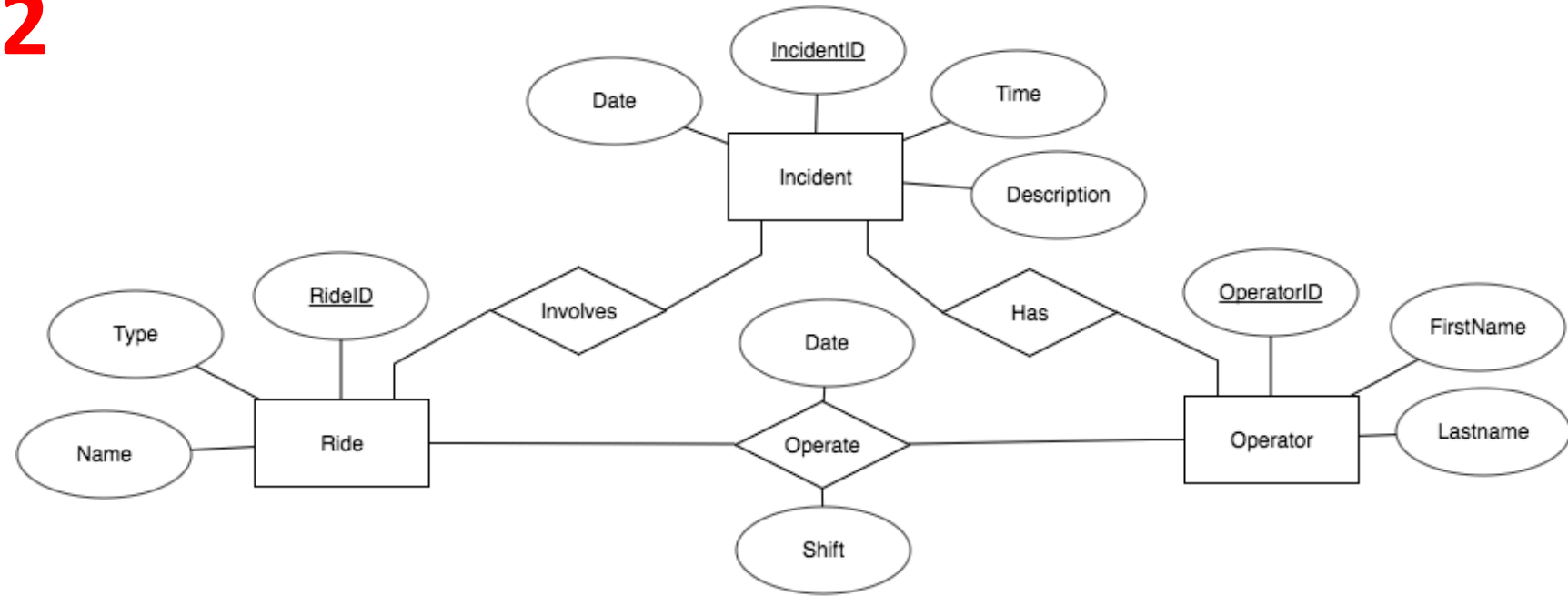




# Implement the relationships (1)

An incident involves both a ride and an operator. All operators and rides are included in the database whether or not they've been involved in a safety incident, and the date and shift are recorded each time an operator runs a ride at the park. A ride is described by a name and type. An operator is described by their first and last name. When an incident occurs, its date and time are recorded along with a description of the incident.

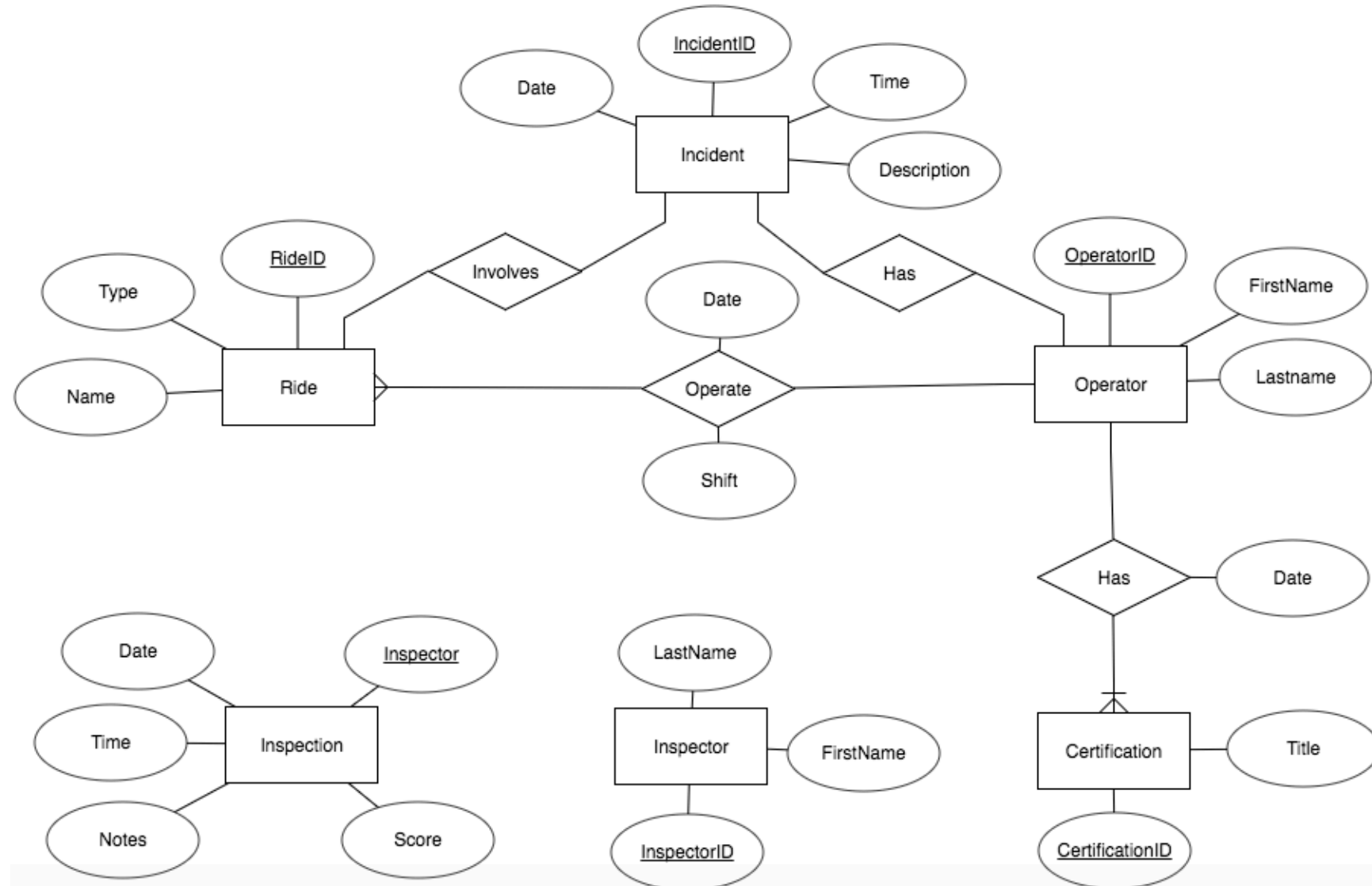
P2



# Implement the relationships (2)

**Operators must be trained and certified** on the rides they operate. An **operator can be certified for multiple rides**, and the certification is valid for two years from the **date** they complete their training. The **title of the certification** indicates the ride for which they are certified. When an incident occurs, it is important that the park can produce a list of current certifications for the operator.

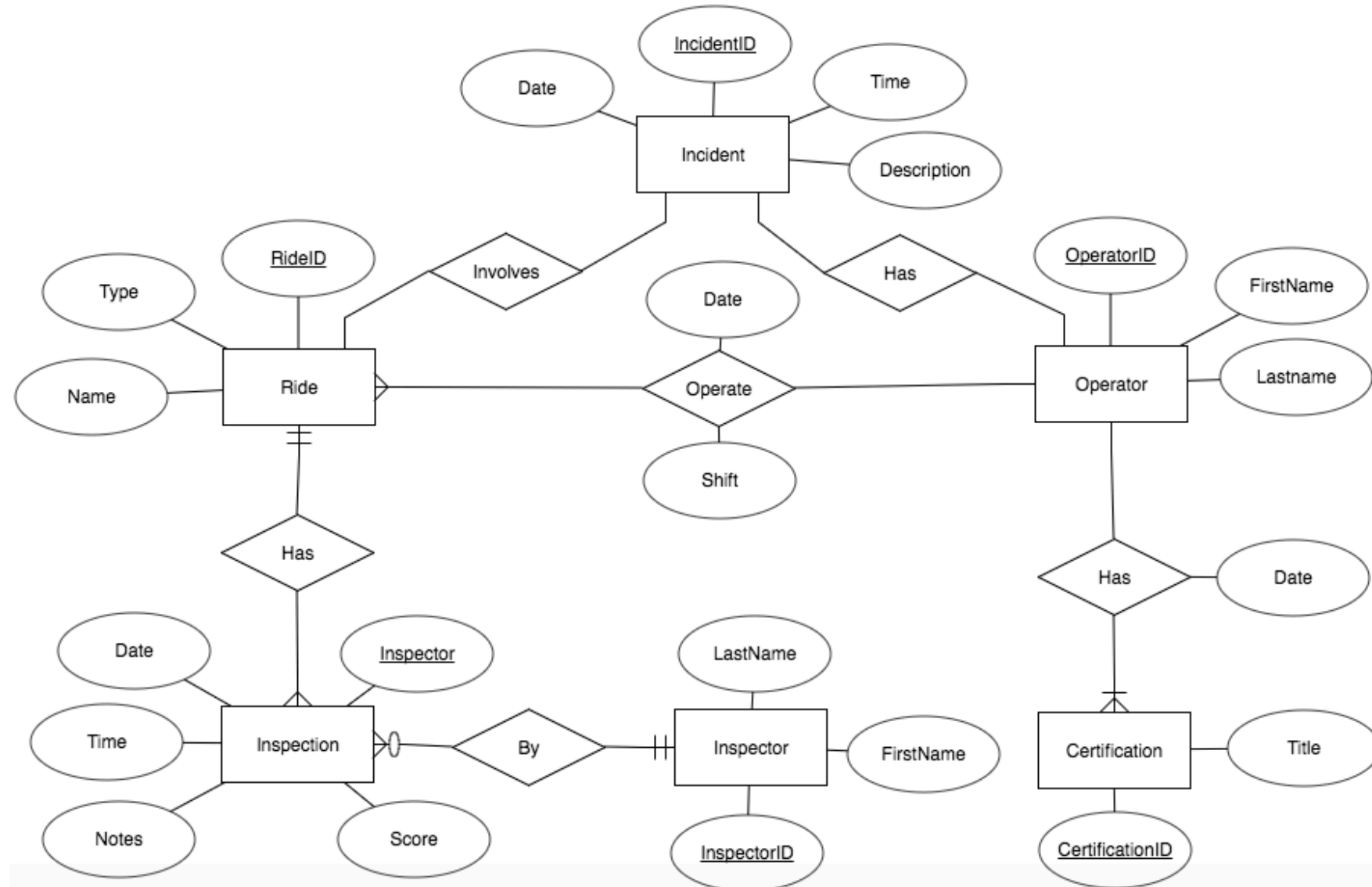
**P3**



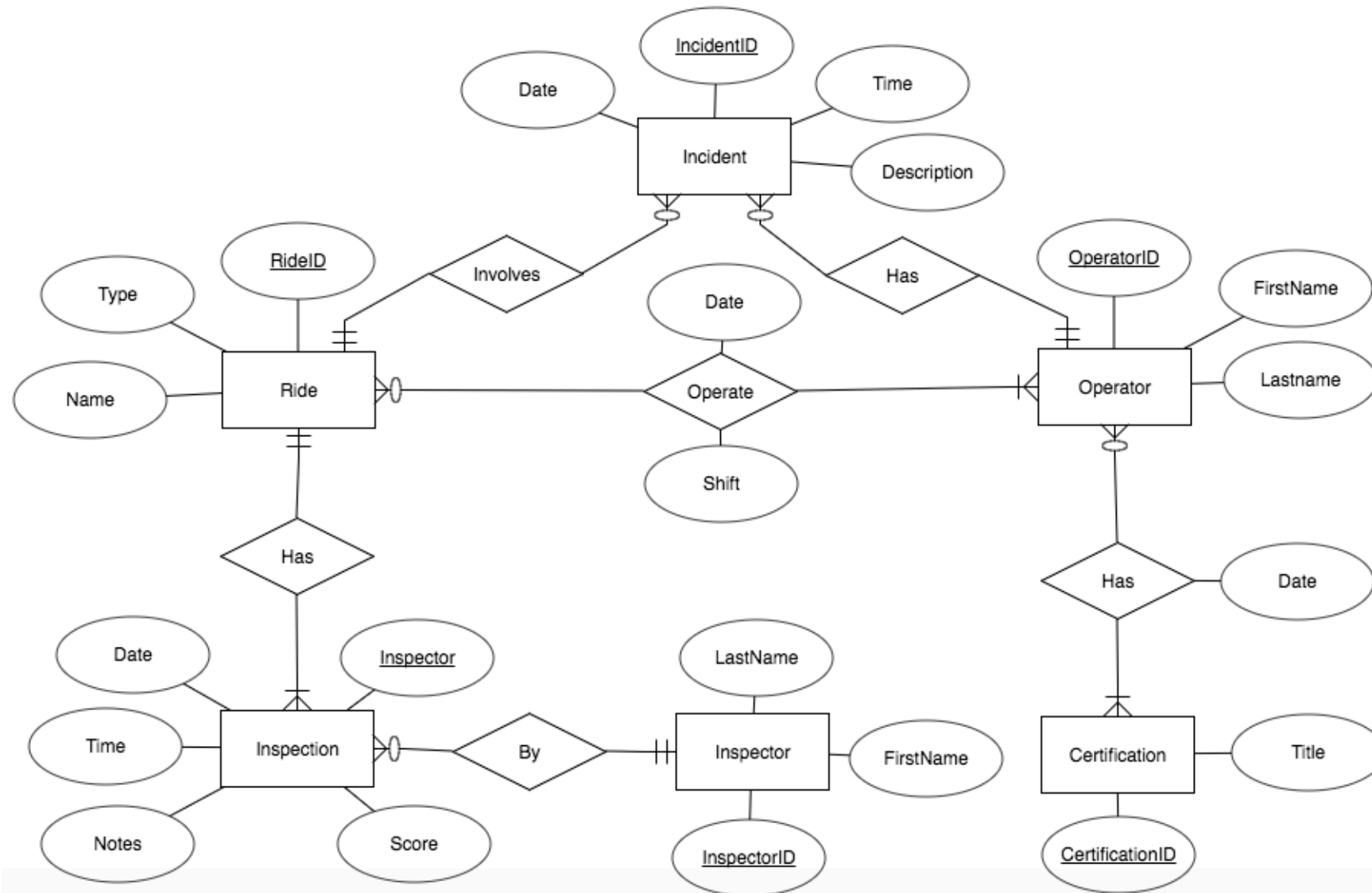
# Implement the relationships (3)

**Inspection** records are also recorded in the database. An inspection occurs at a particular date and time, with a numeric score from 1 to 5, and notes about the outcome of the inspection. **There is only one ride per inspection, even if multiple inspections are done by the same inspector** on the same day and time. The information that needs to be recorded about the **inspector** is their first and last name.

P4



We need to figure out the cardinality of the Rest



# Scenario 2: SchUber (1)

- What's the purpose of this database?
  - Track trips and customers
  - Drivers and passengers rate each other
- What entities do we have?
  - Driver, Passenger (Customer), Trip, Rating

**P1**

A new Philadelphia startup called SchUber is a matching service between freelance taxi **drivers and passengers**. The database to support the service tracks **trips**, **customer** accounts, and includes a **rating** system where drivers and passengers can rate each other.

# Scenario 2: SchUber (2)

- Trip
  - Date, Time, Fare, Trip ID
- Driver
  - First Name, Last Name, SSN, Email, Driver ID
- Passenger
  - First Name, Last Name, Email, Passenger ID

P2

The basic transaction for SchUber is the trip. A **trip** is described by the date and time the trip occurs and the fare. **A trip involves one driver and at least one passenger.** A **driver** has a first name, a last name, a social security number, and an email address. A **passenger** has a first name, last name, and an email address.

# Scenario 2: SchUber (3)

- Address
  - Street, City, State, Zip Code, Home, Address ID
- Sentences in green do not talk about entities but the relationships.

P3

A trip is also associated with at least two addresses. An address is also described by the street address, city, state, and zip code. Each address must be designated as either an origin or destination address for the trip. Each address also has an attribute which designates the address as a home address for a passenger – this is useful for billing and for making a request for a home pick-up. However, a passenger can only have one address (their home address) associated with their account.

# Scenario 2: SchUber (4)

- Payment
  - Methods, Expiration Date, Payment ID
  - What should we do with PayPal, Credit Card and Google Wallet?
- Again, sentences in green do not talk about entities but the relationship attributes.

P4

A passenger pays for the trip using any one of the **payment** methods linked to their account. Valid payment method types are *PayPal, Credit Card, or Google Wallet*. Credit cards also have an expiration date, but this information is not needed for PayPal or Google Wallet. **Passengers cannot share a payment method, but if there are multiple passengers on a trip they can each pay their share, and each of their payments can be a different amount (which should be recorded in the database) to the fare.**

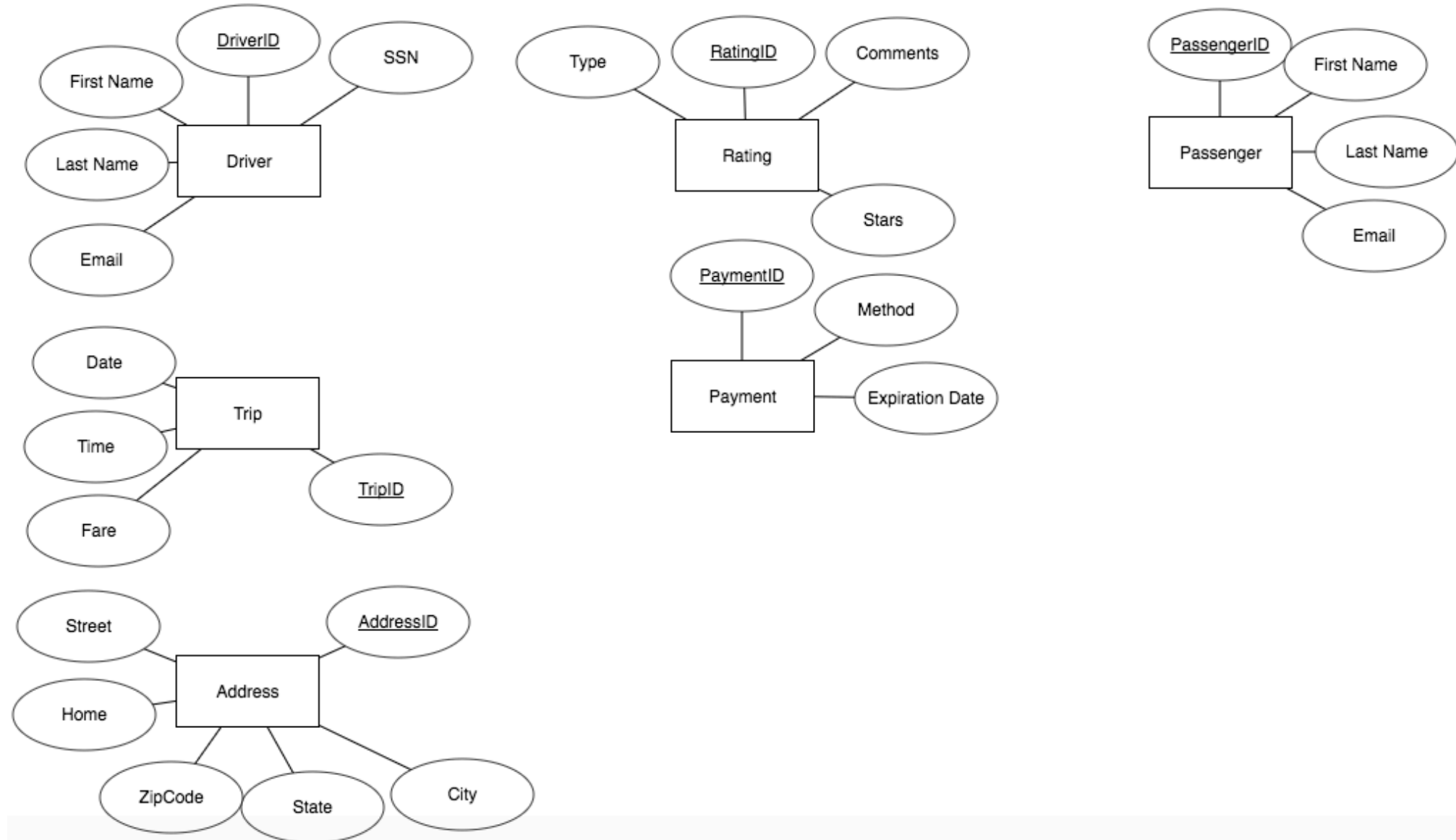


# Scenario 2: SchUber (5)

- Rating
  - Star, Comments, Type, Rating ID
- Now we have identified all entities

**P5** The rating system is one of the key features of SchUber. Both drivers and passengers can have ratings, although they may not have any ratings, especially if they just started with the service. When a driver or a passenger submits the rating, they give a star rating (1 to 5), brief comments, and the type of review (i.e., for driver or for passenger). SchUber users can look up ratings by driver, passenger, or by trip (a trip could have more than two ratings if there were multiple passengers on the trip).

# We draw all entities first

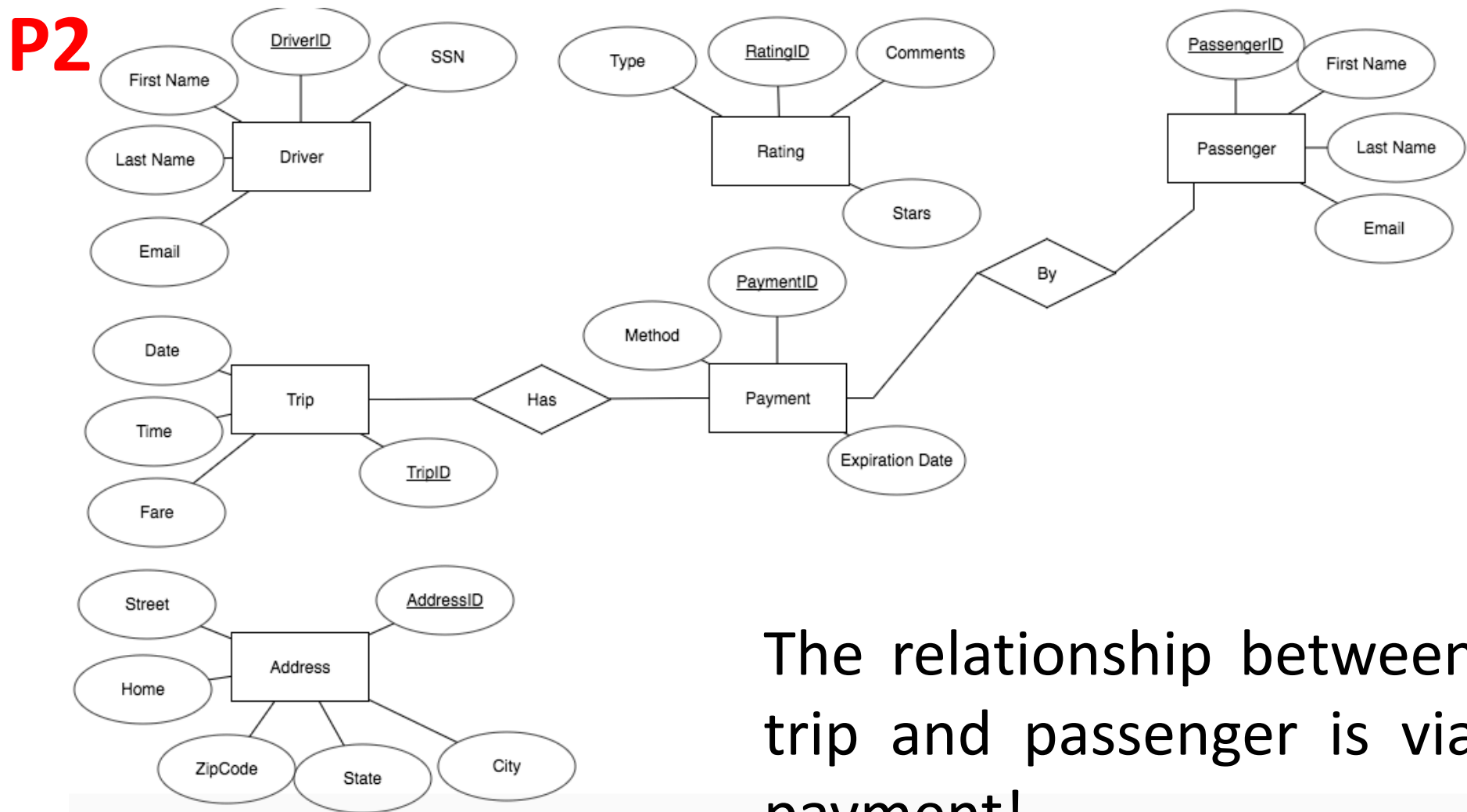


# Now we implement relationships (1)

The basic transaction for SchUber is the trip. A **trip** is described by the date and time the trip occurs and the fare.

A trip involves one driver and at least one passenger. A **driver** has a first name, a last name, a social security number, and an email address.

A **passenger** has a first name, last name, and an email address.

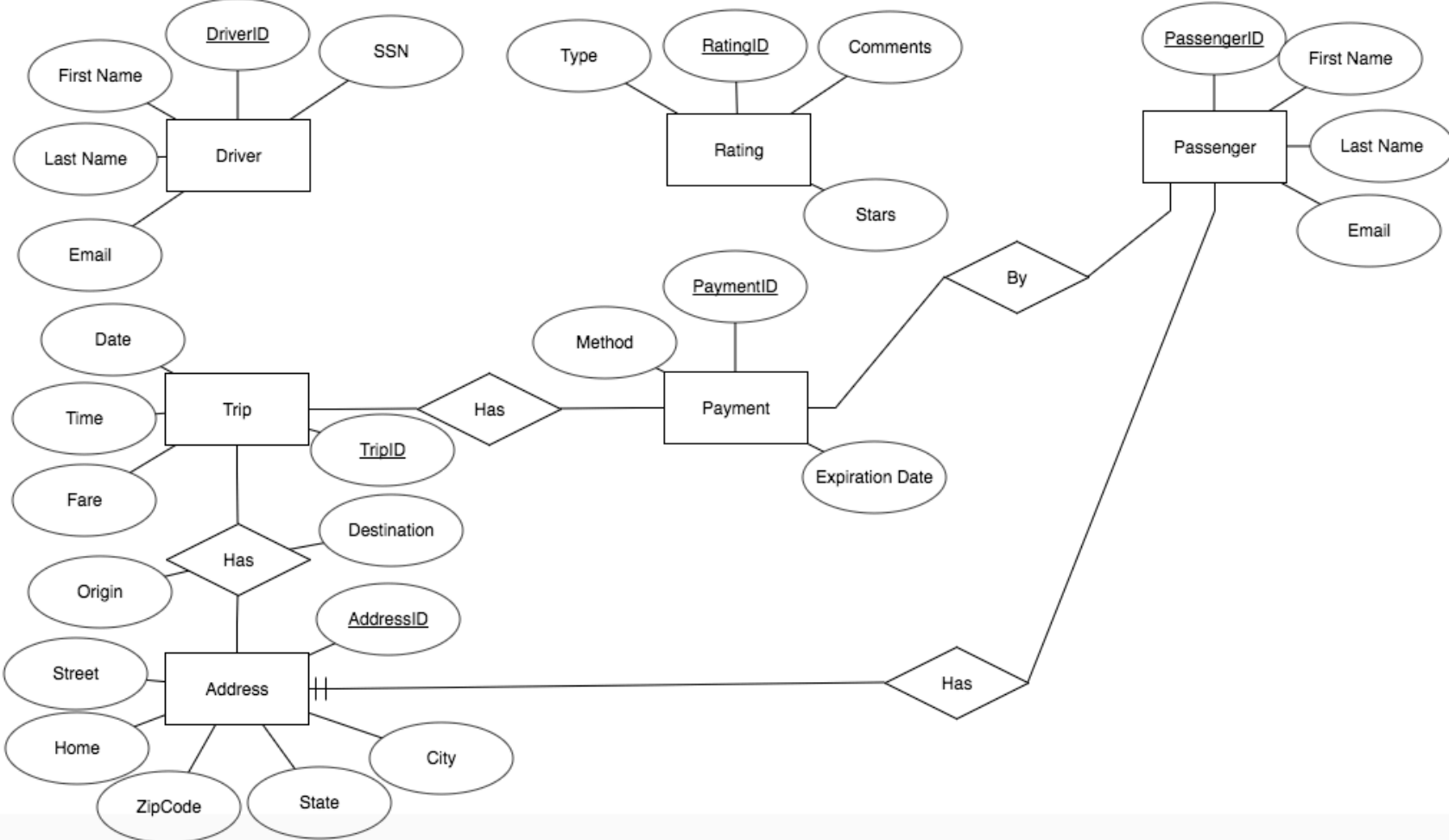


The relationship between trip and passenger is via payment!

# Now we implement relationships (2)

A trip is also associated with at least two addresses. An address is also described by the street address, city, state, and zip code. Each address must be designated as either an origin or destination address for the trip. Each address also has an attribute which designates the address as a home address for a passenger – this is useful for billing and for making a request for a home pick-up. However, a passenger can only have one address (their home address) associated with their account.

P3

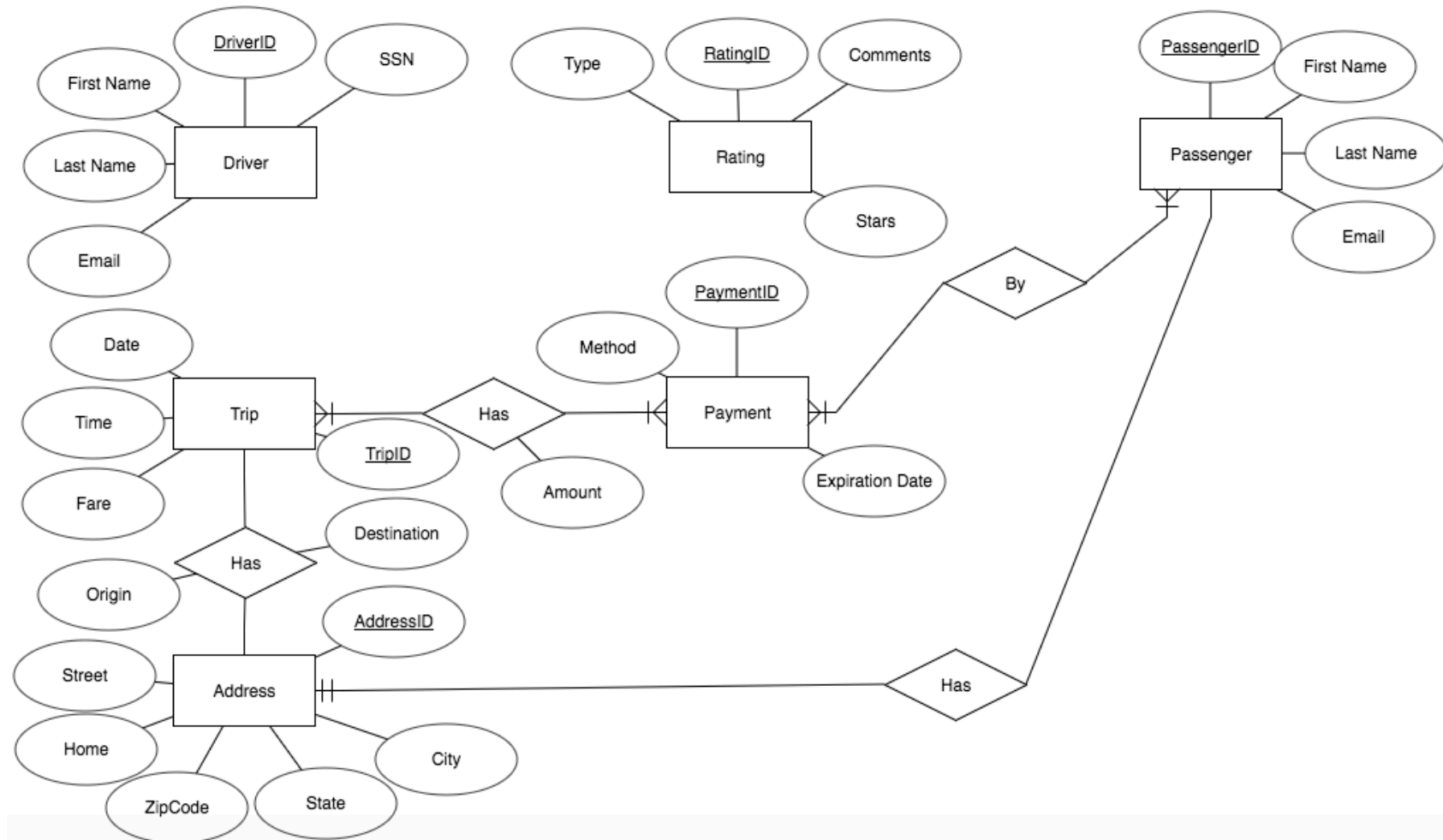


# Now we implement relationships (3)

A passenger pays for the trip using any one of the **payment** methods linked to their account. Valid payment method types are **PayPal, Credit Card, or Google Wallet**. Credit cards also have an expiration date, but this information is not needed for PayPal or Google Wallet.

Passengers cannot share a payment method, but if there are multiple passengers on a trip they can each pay their share, and each of their payments can be a different amount (which should be recorded in the database) to the fare.

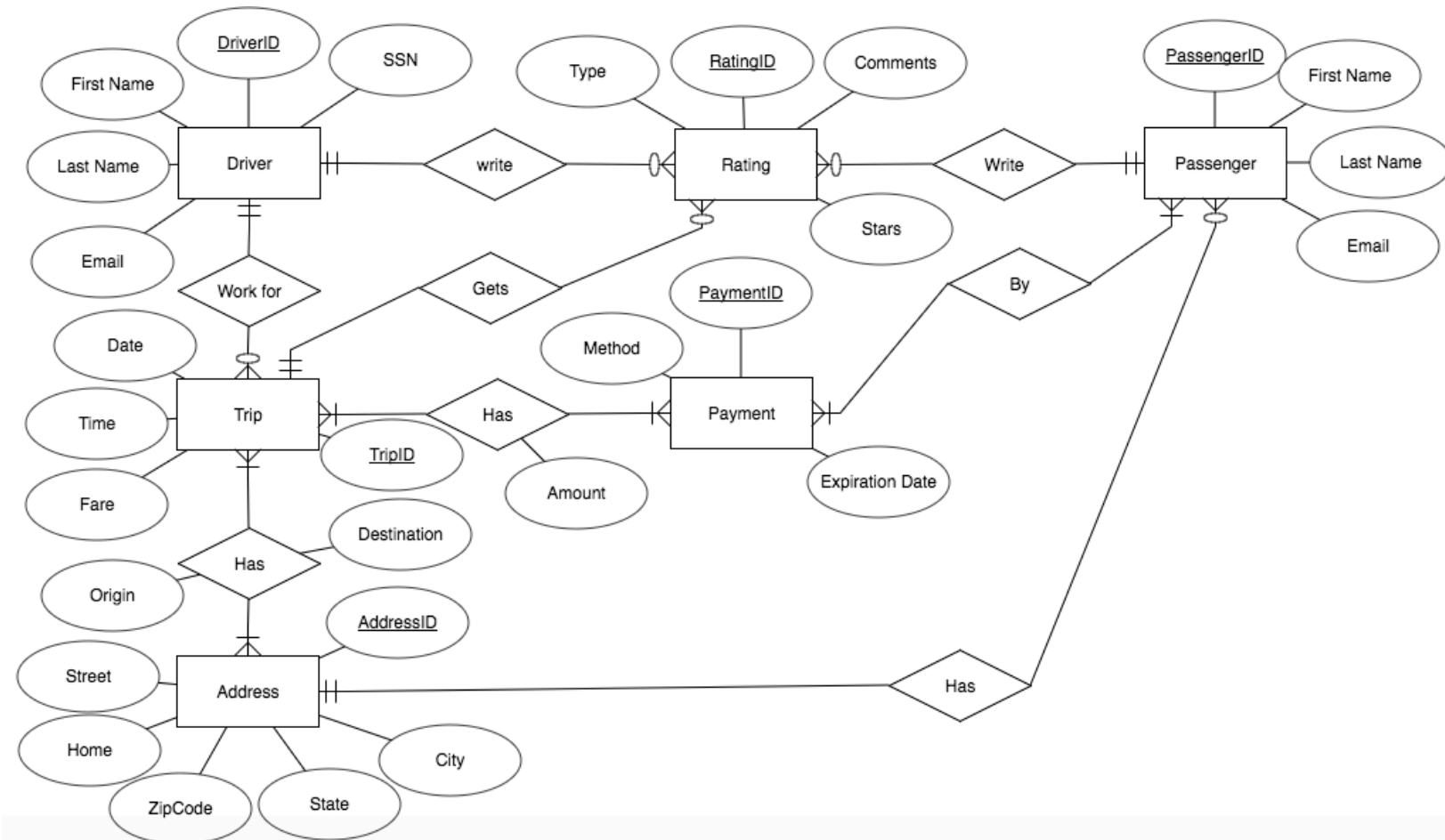
P4



# Now we implement relationships (3)

The rating system is one of the key features of SchUber. Both drivers and passengers can have ratings, although they may not have any ratings, especially if they just started with the service. When a driver or a passenger submits the rating, they give a star rating (1 to 5), brief comments, and the type of review (i.e., for driver or for passenger). SchUber users can look up ratings by driver, passenger, or by trip (a trip could have more than two ratings if there were multiple passengers on the trip).

P5



# How to Draw ERD: a Check List

- Step 1: Identify all entities
  - Primary Key
  - Attributes
- Step 2: Implement all relationships
  - Identify relationship attributes
- Step 3: Cardinality
  - Maximum cardinality
  - Minimum cardinality