

# Relational Modeling

Big Data Systems

Dr. Rubi Boim

# Motivation (for this course)

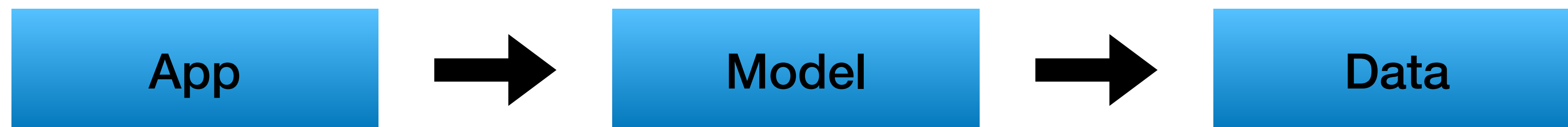
- Data modeling is an important process when creating a relational database
- Data modeling is **the most important** process when creating a **big data database**
- Modeling for NoSQL is “different” than relational  
understanding relational modeling is crucial for wide column modeling

# Relational vs NoSQL - design

- Relational  
focus on entities



- NoSQL  
focus on queries



# Relational data modeling

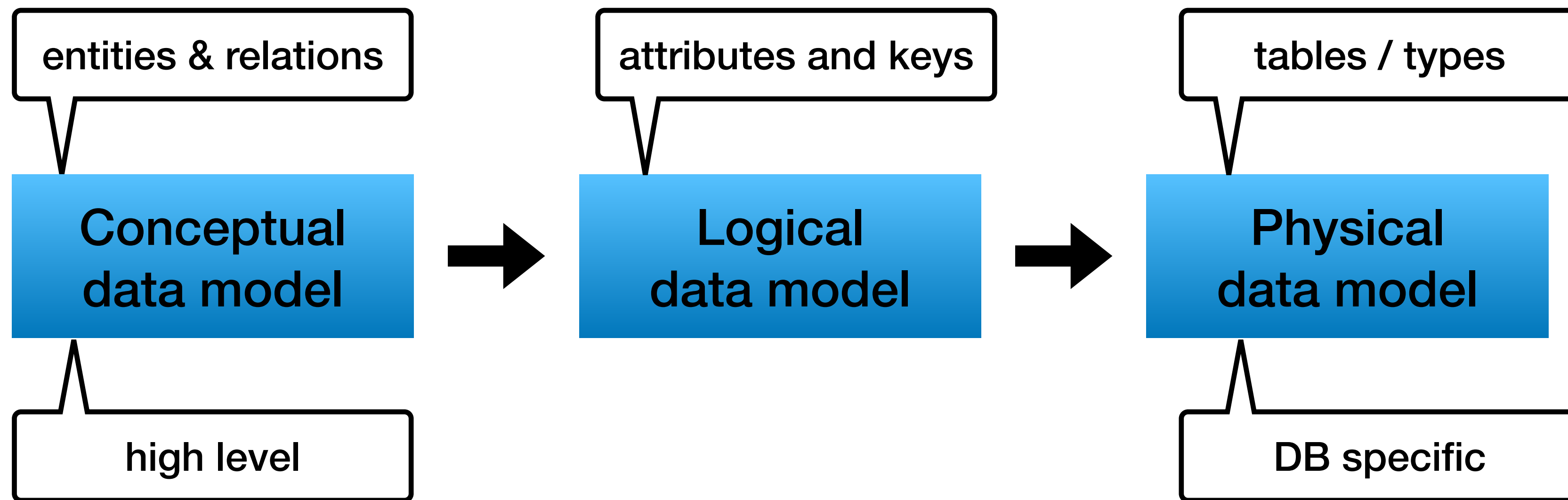
# Modeling is an Art

- Multiple ways to solve design problems
- Uncommon use case —> think out of the box

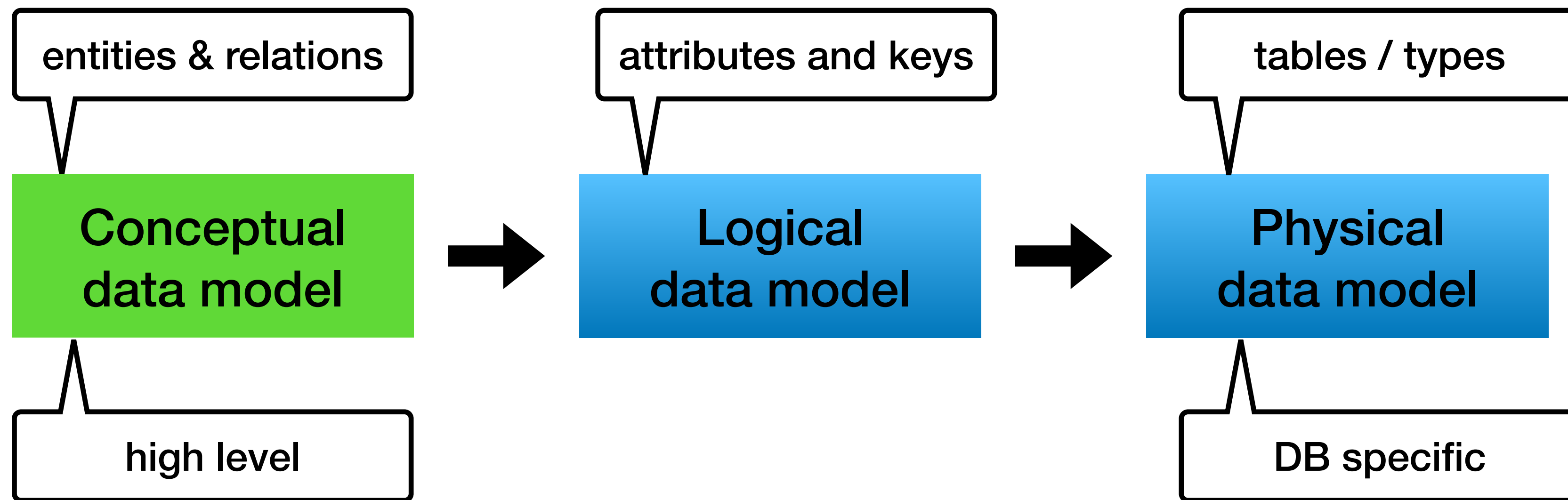
# Relational Modeling - general steps

- Map conceptual entities, attributes and their relations
- Map primary and foreign keys
- Define data types
- Create tables

# Relational Modeling - 10,000 foot view



# Relational Modeling - 10,000 foot view



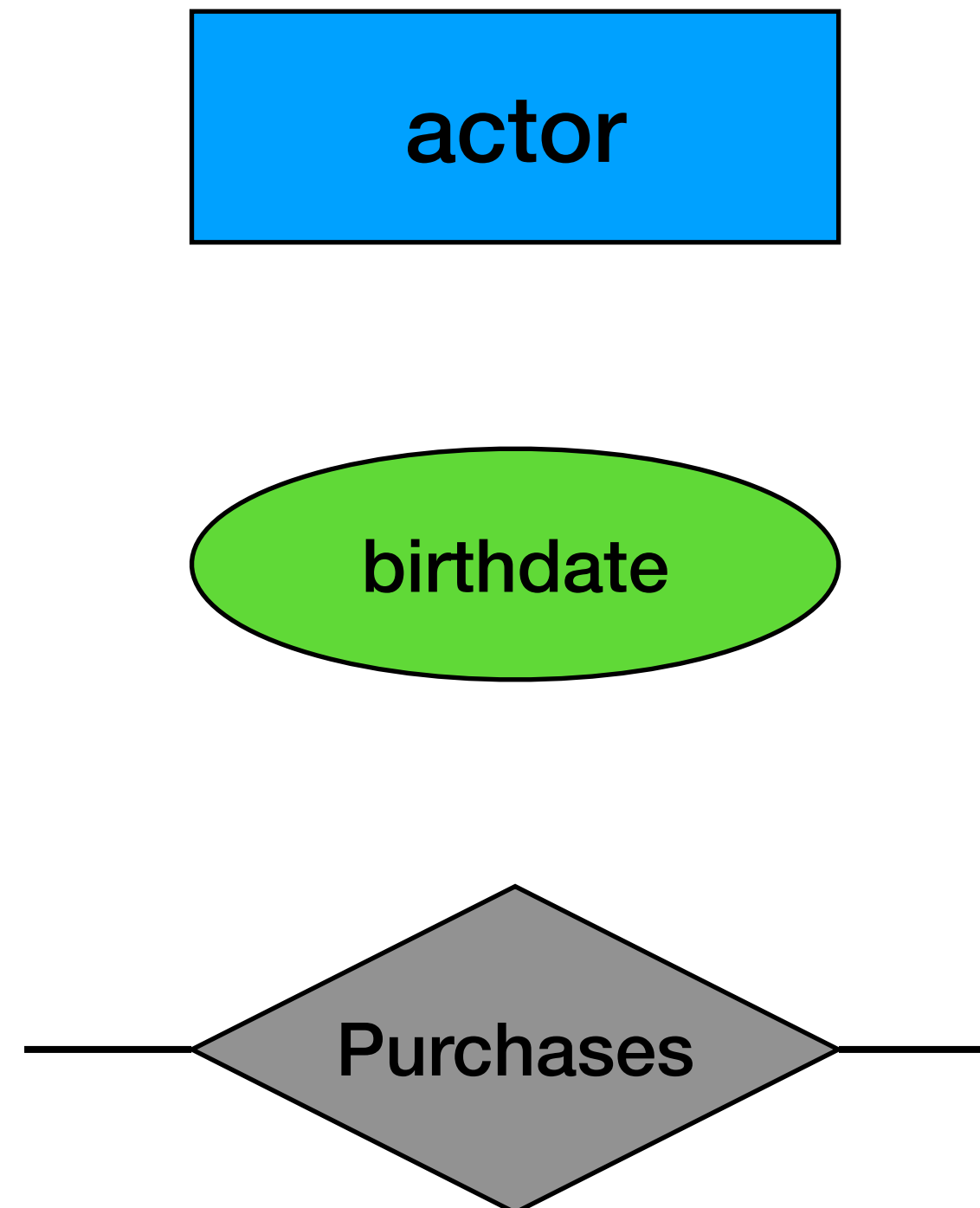


# Conceptual data model

- **Abstract view of the world**  
server and database types are irrelevant
- **Can be defined by non technical teams**  
not really in reality...
- **Entity / Relationship model (ER)**

# ER Model

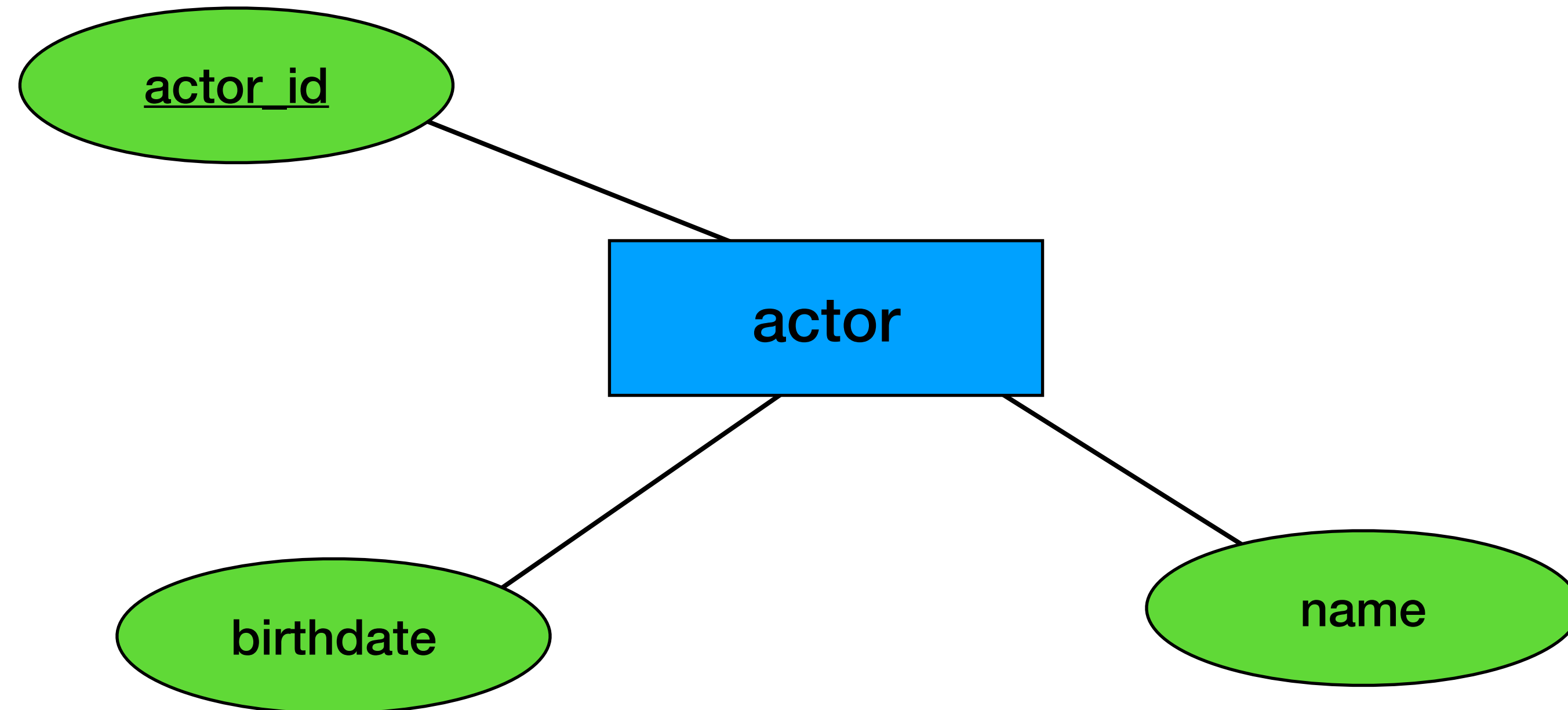
- Entities
- Attributes
- Relations  
between entities



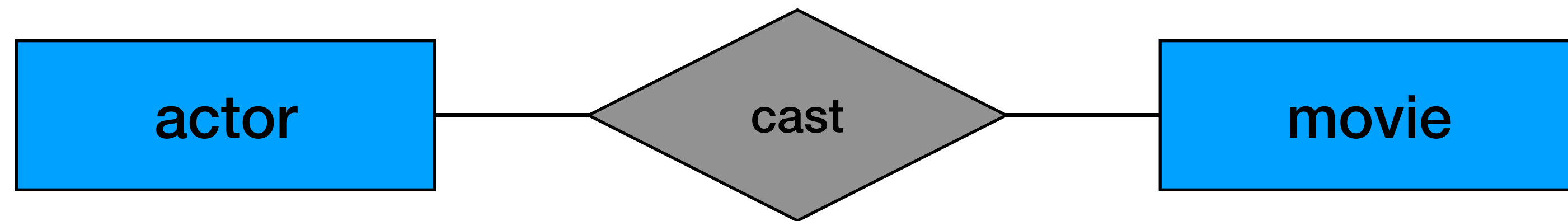
\* There are more types like ISA (is a)

# Entity

- Each entity must have a key

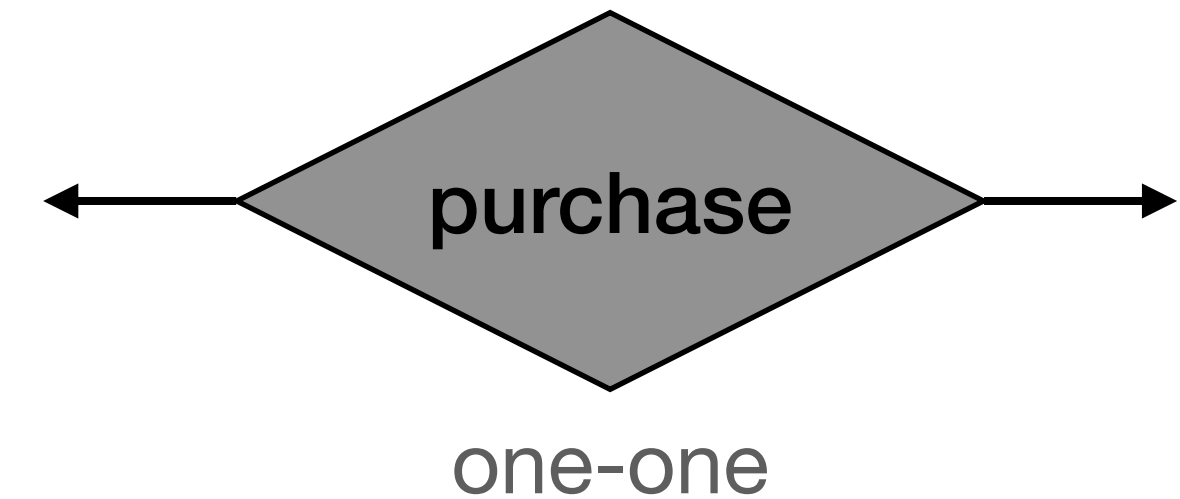
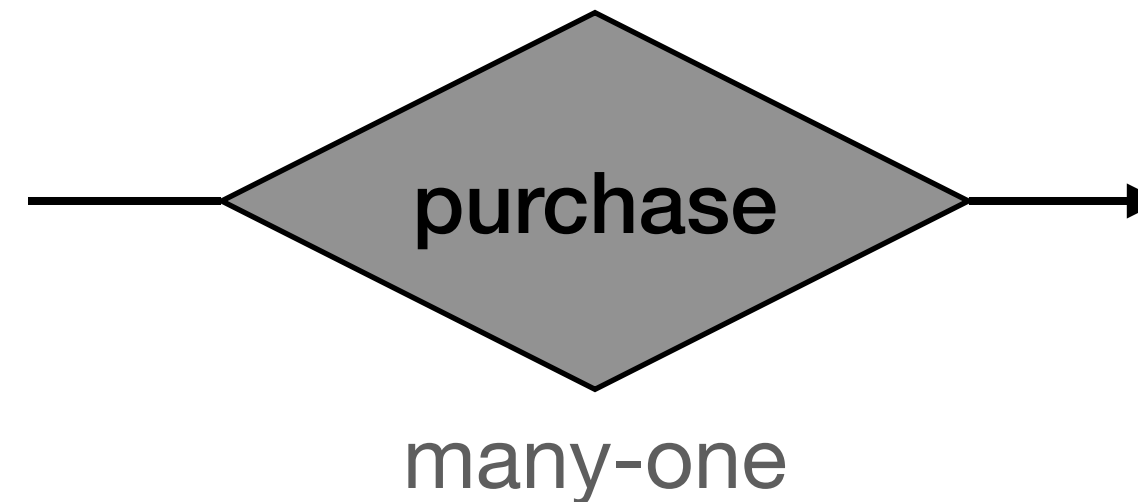
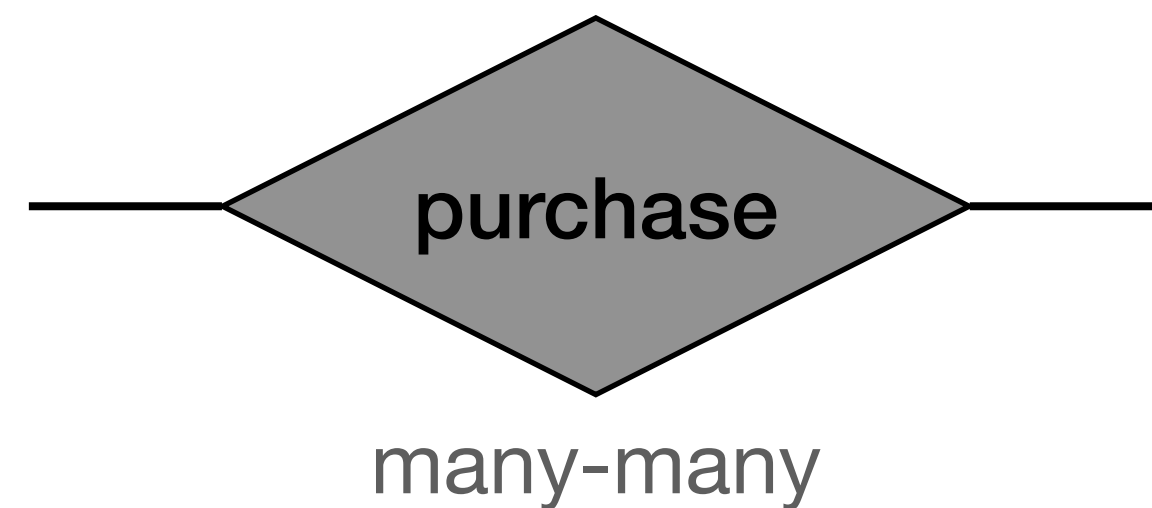


# Relation (between entities)

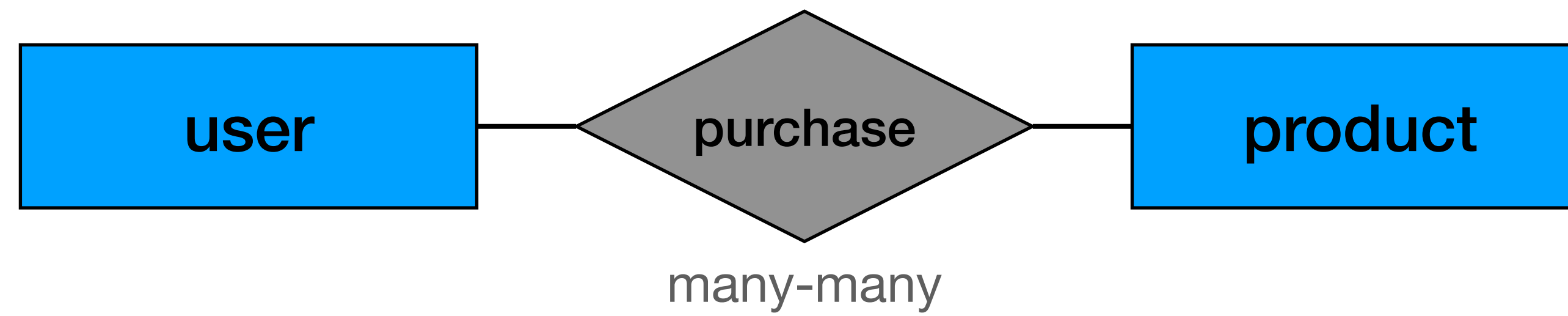


# Cardinality (of relation)

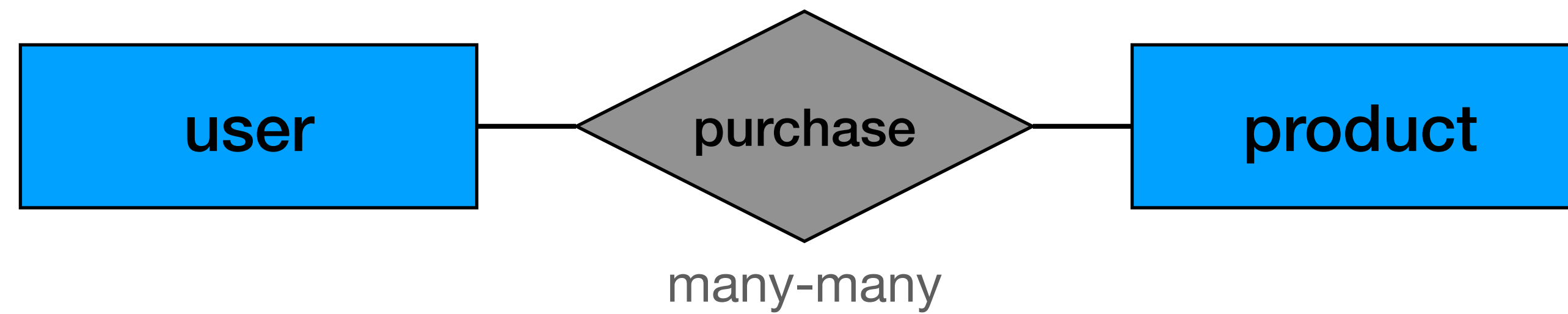
- cardinality is the number of occurrences in one entity which are associated to the number of occurrences in another



# Many to Many

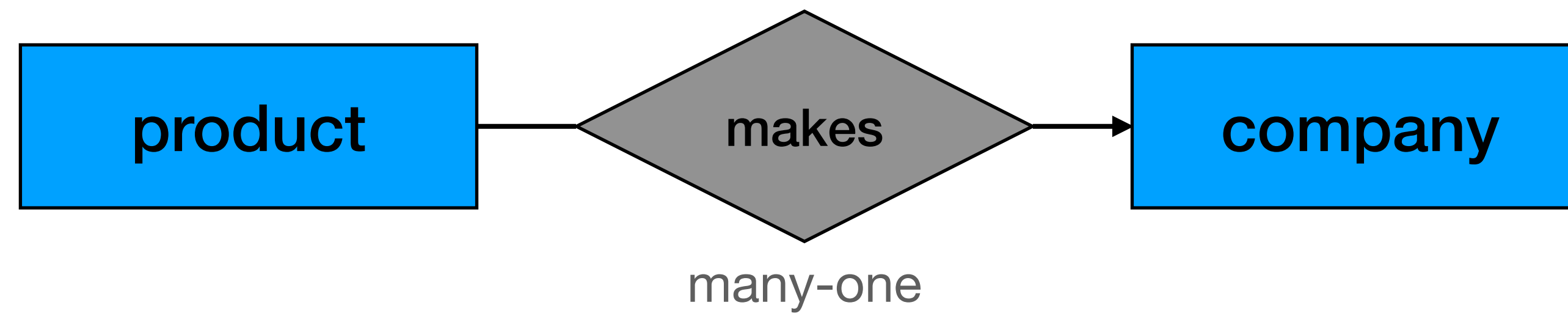


# Many to Many



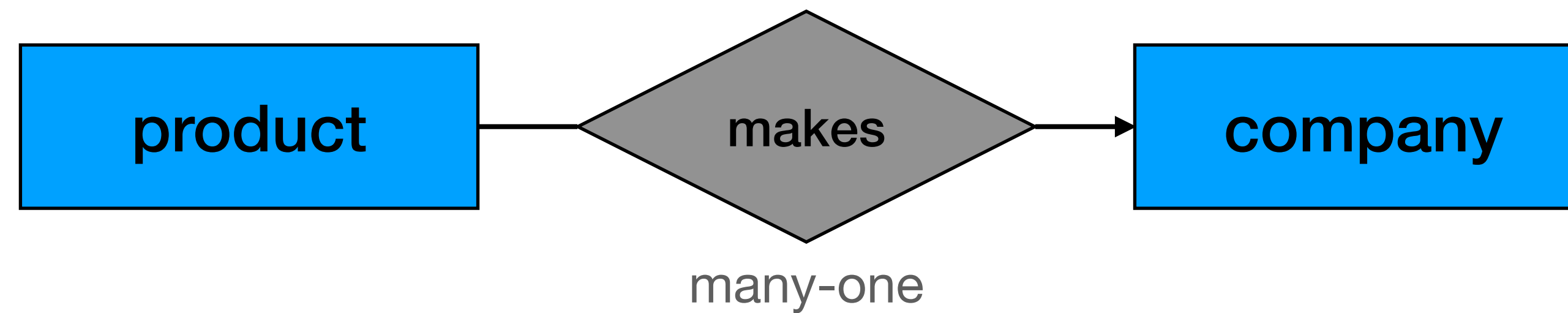
Each user can buy many products  
(but each product only once)

# Many to One



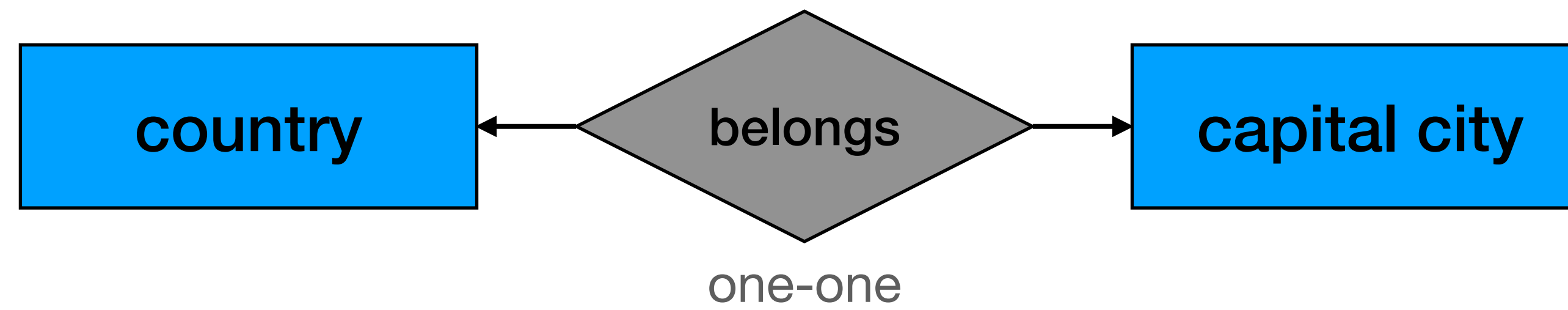


# Many to One

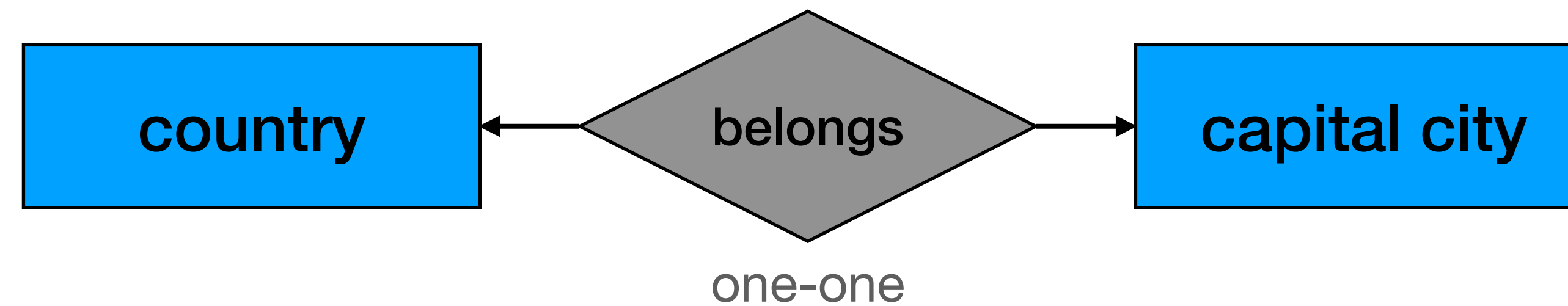


Each product is made by one company

# One to One

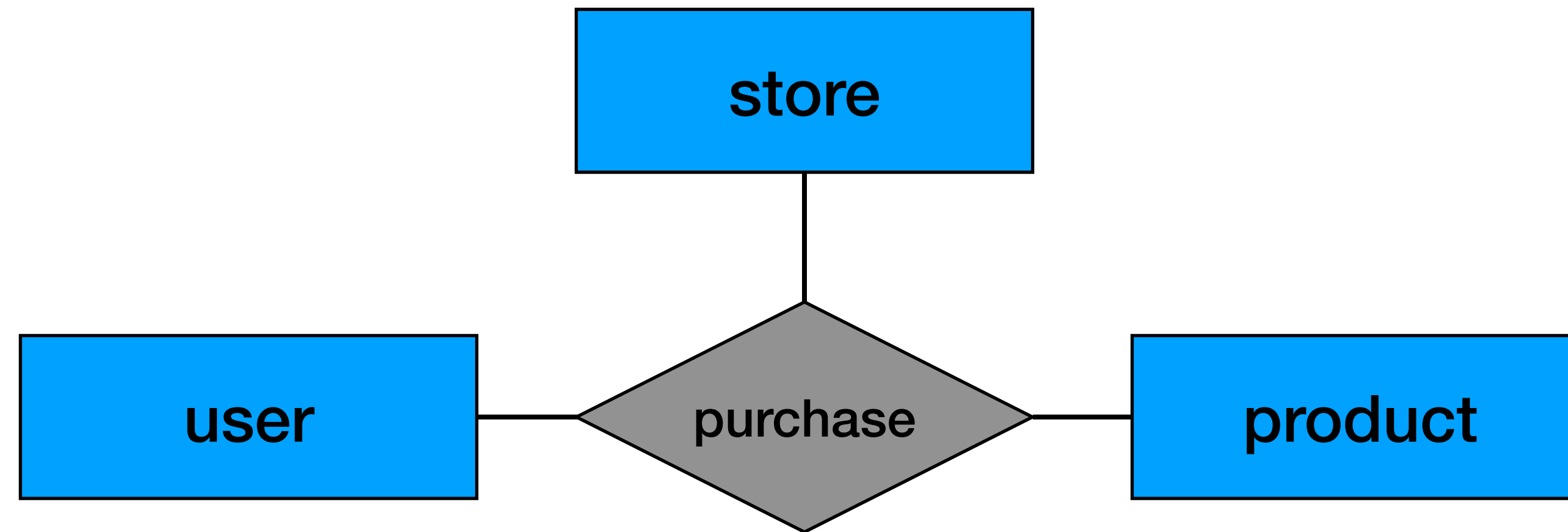


# One to One

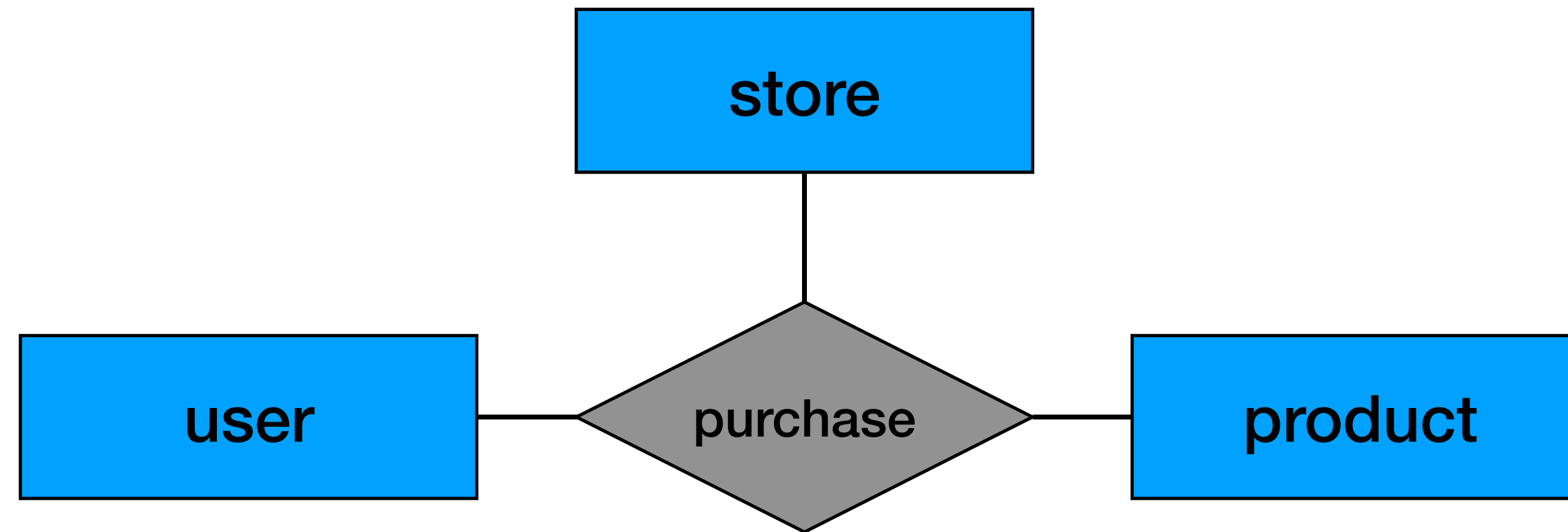


Each country has one capital city, and each capital city belongs to one country

# Multi way relations

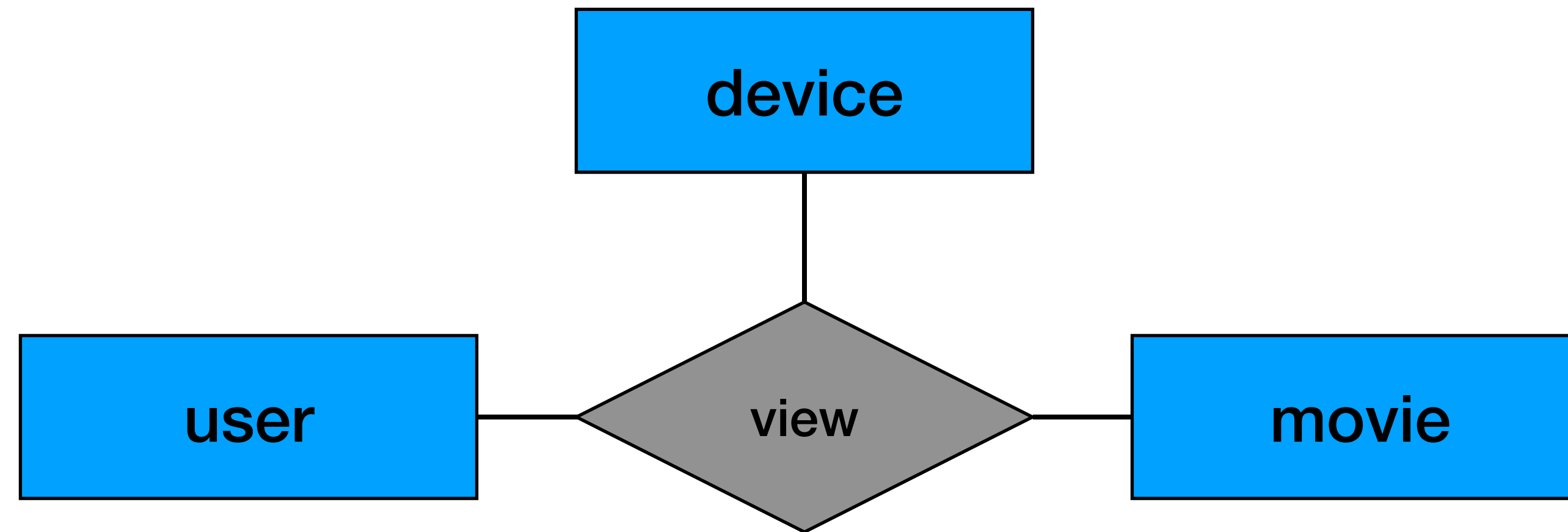


# Multi way relations

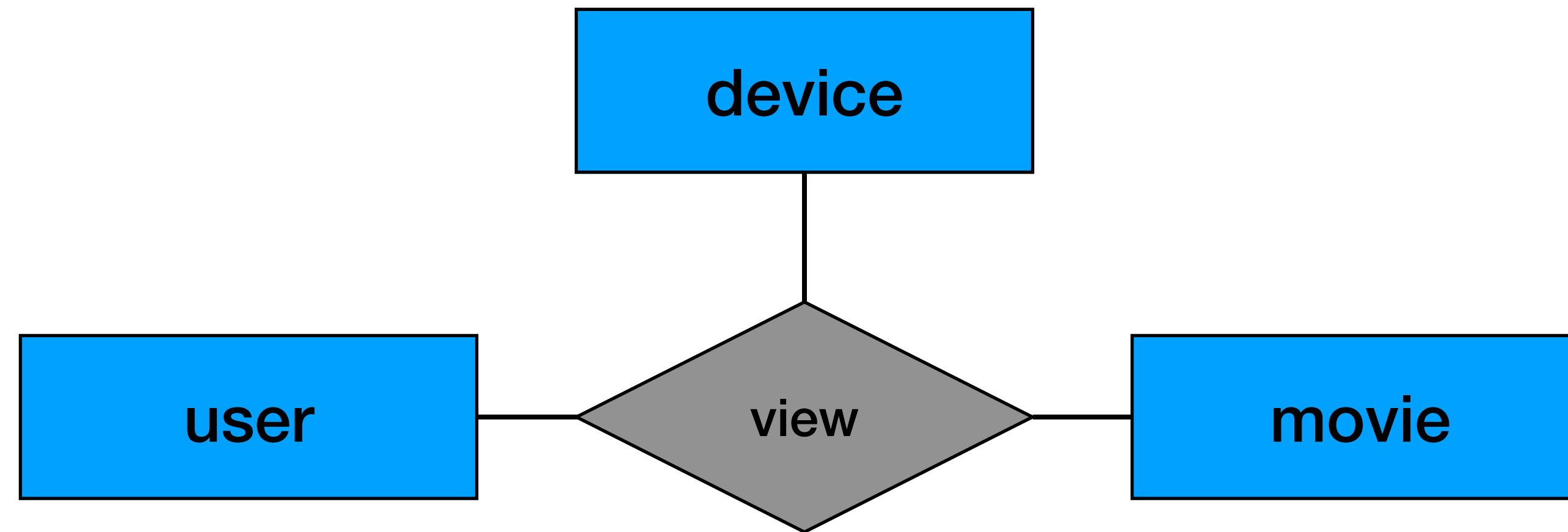


Each user can buy many products in different stores  
(but user-store-product combination only once)

# Multi way relations (another example)

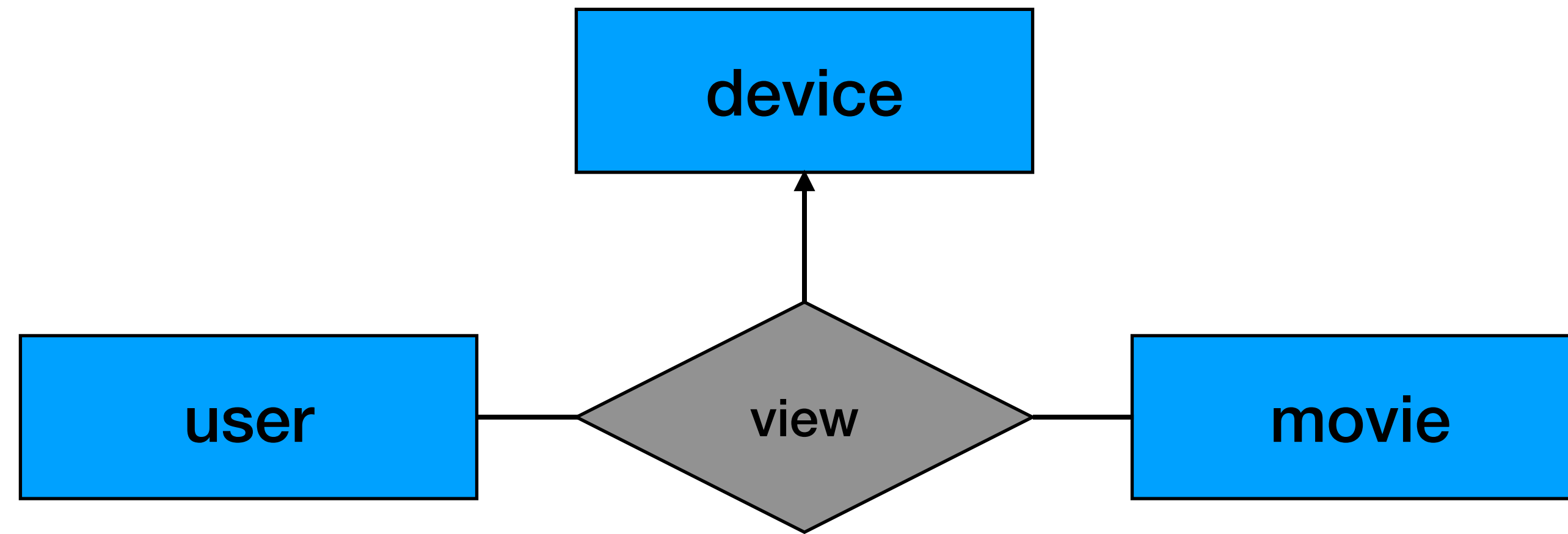


# Multi way relations (another example)



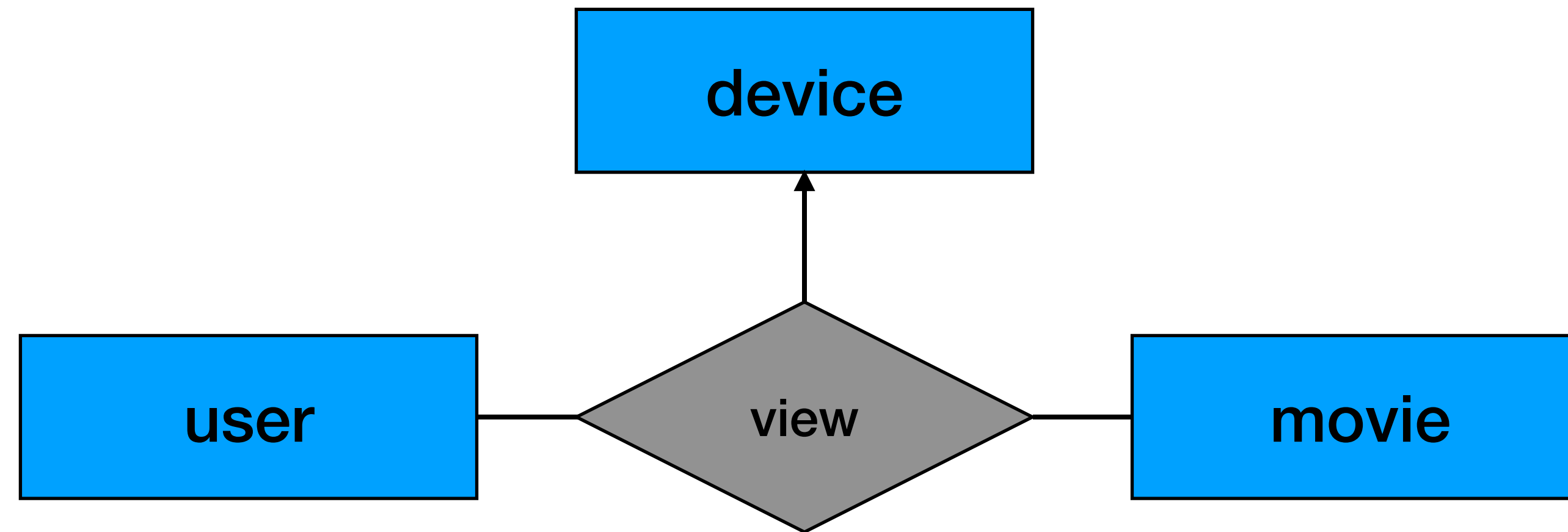
Each user can view many movies on different devices  
(but user-movie-decive combination only once)

# Multi way relations + cardinality



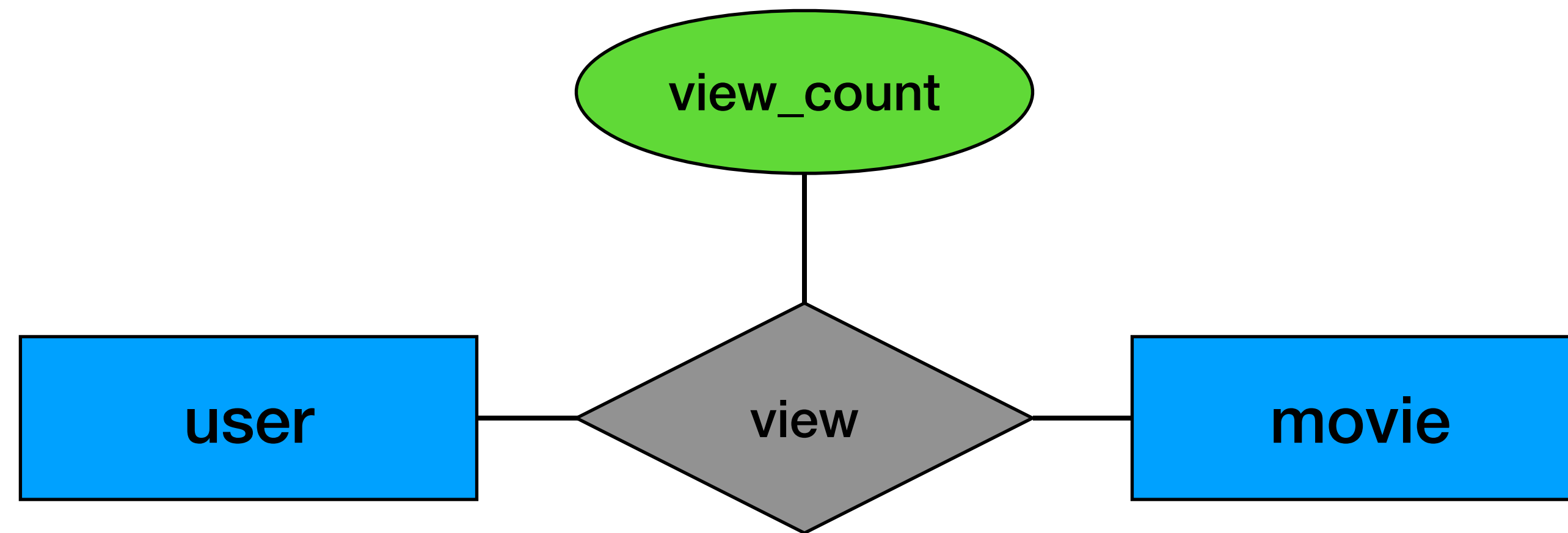


# Multi way relations + cardinality

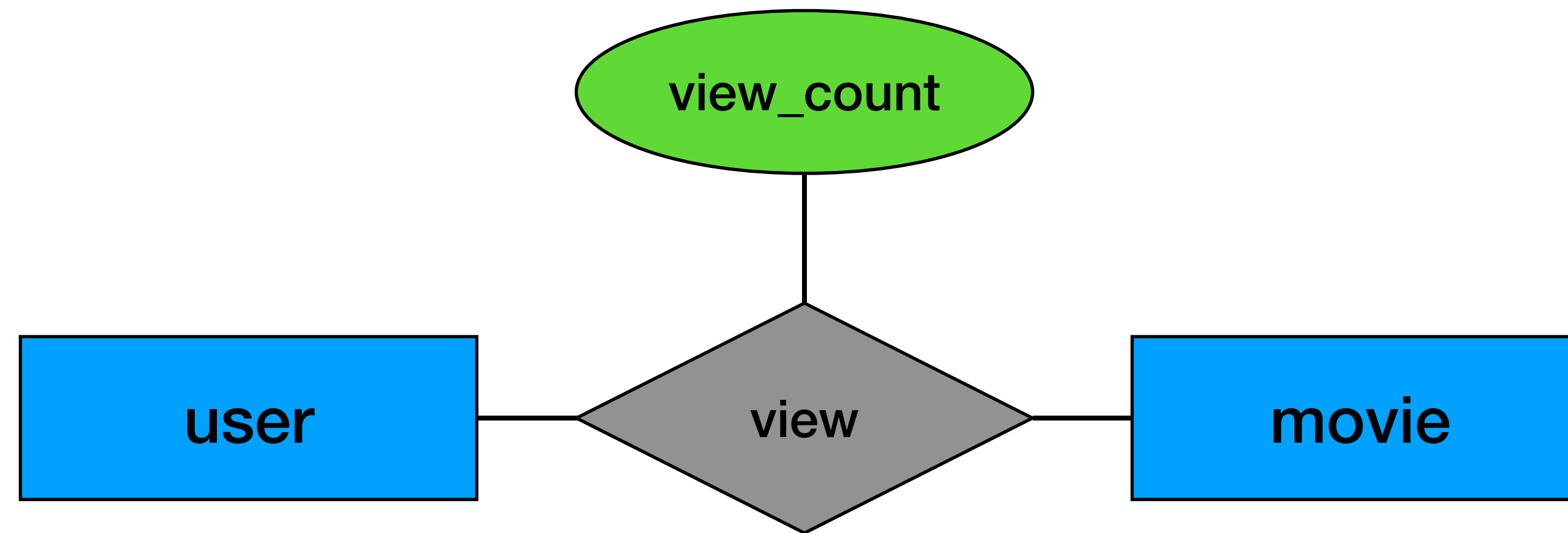


Each user can view many movies.  
If we know the user and the movie, we know the device

# Attributes for relations

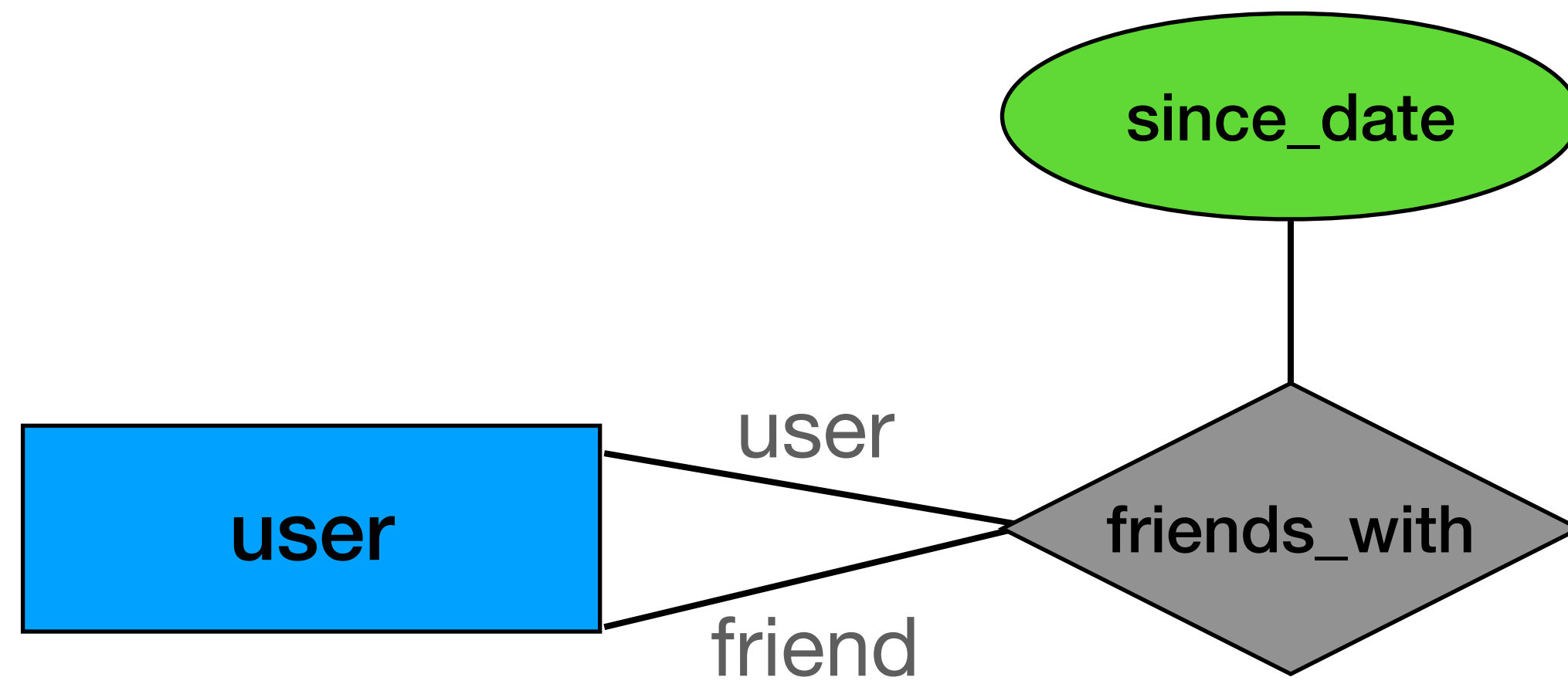


# Attributes for relations

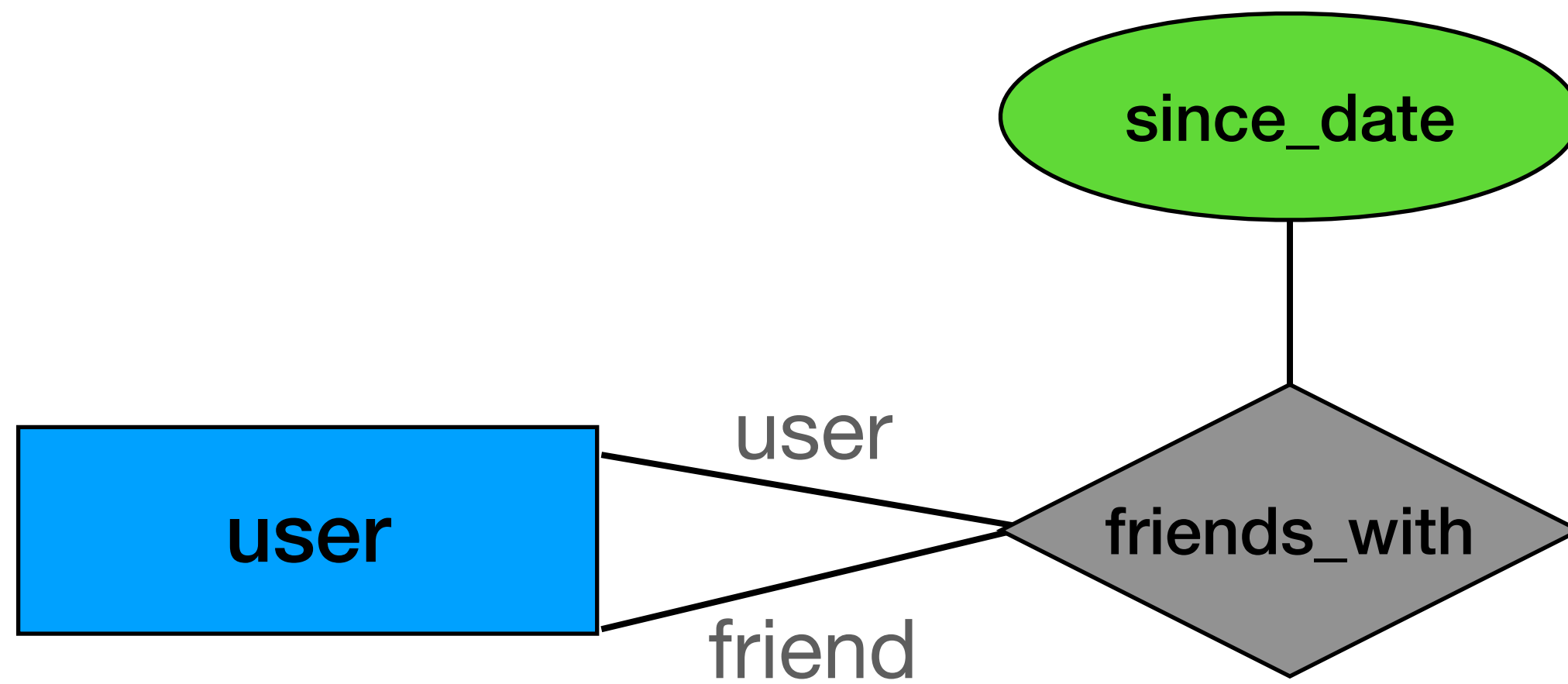


Each user can view many movies.  
For each “view” we also save the view\_count

# Roles in relations



# Roles in relations



A user can be friends a different user

From previous class:

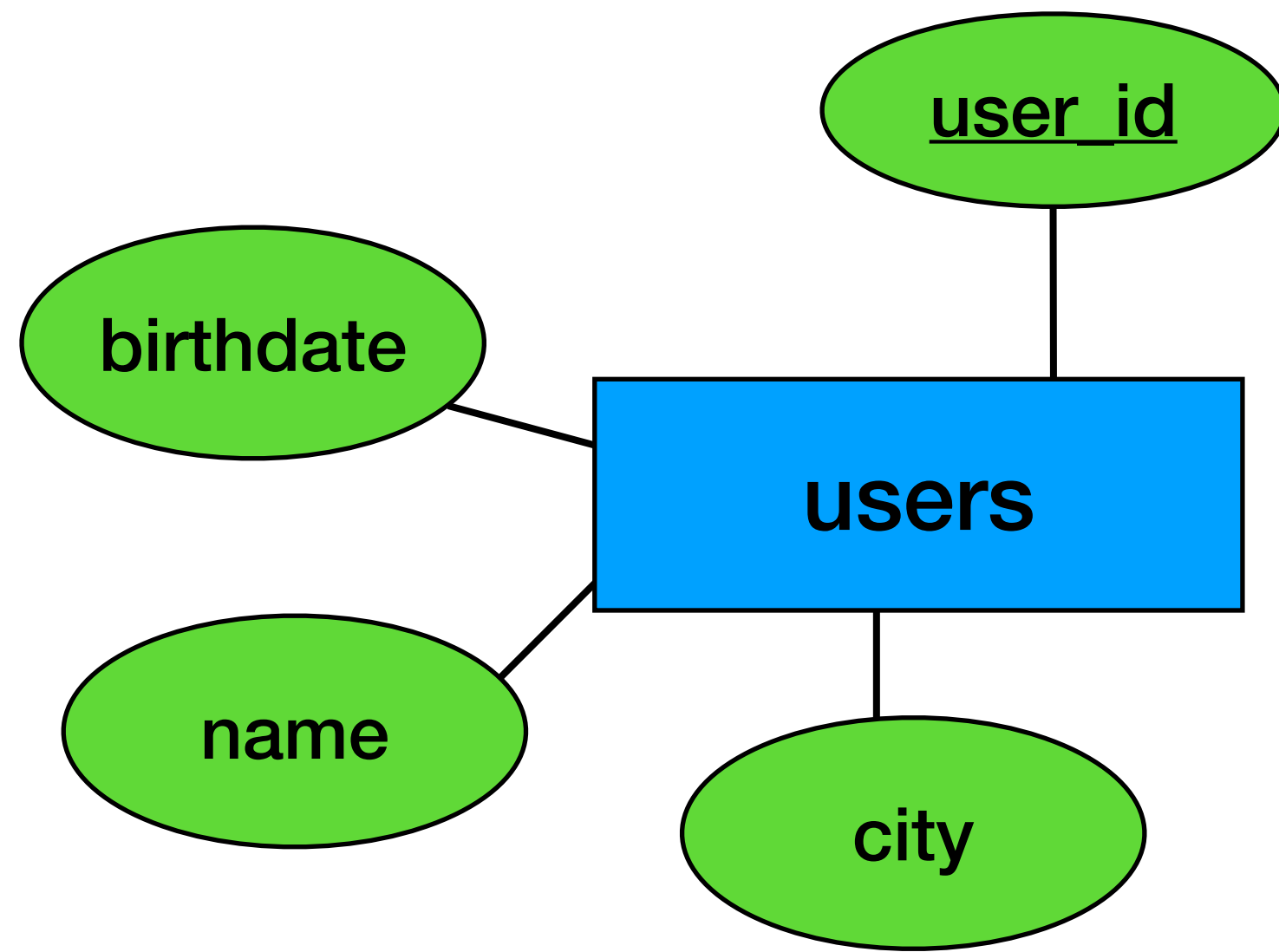
```
friends(user_id, friend user_id, since_date)
```

# Example

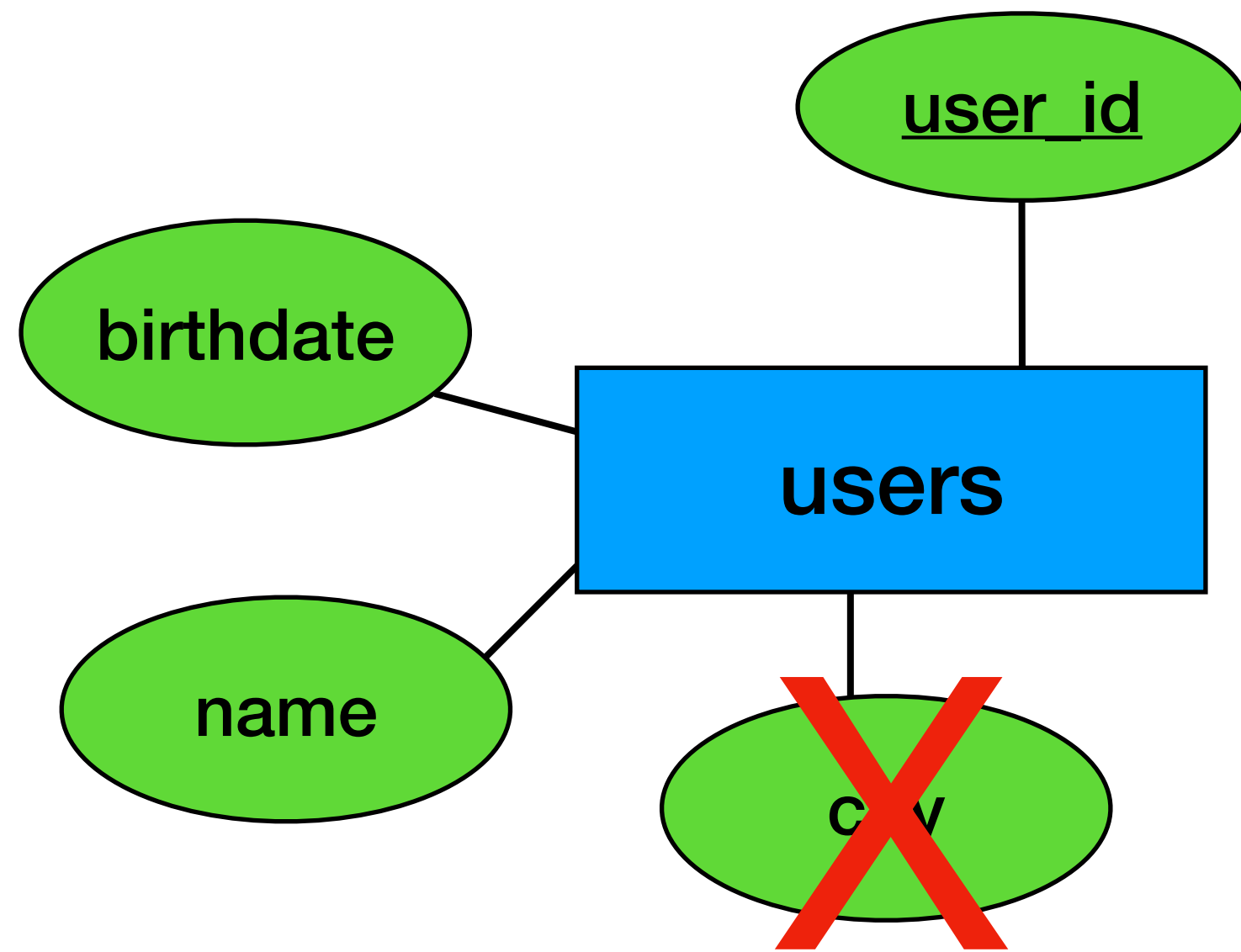
# Story time

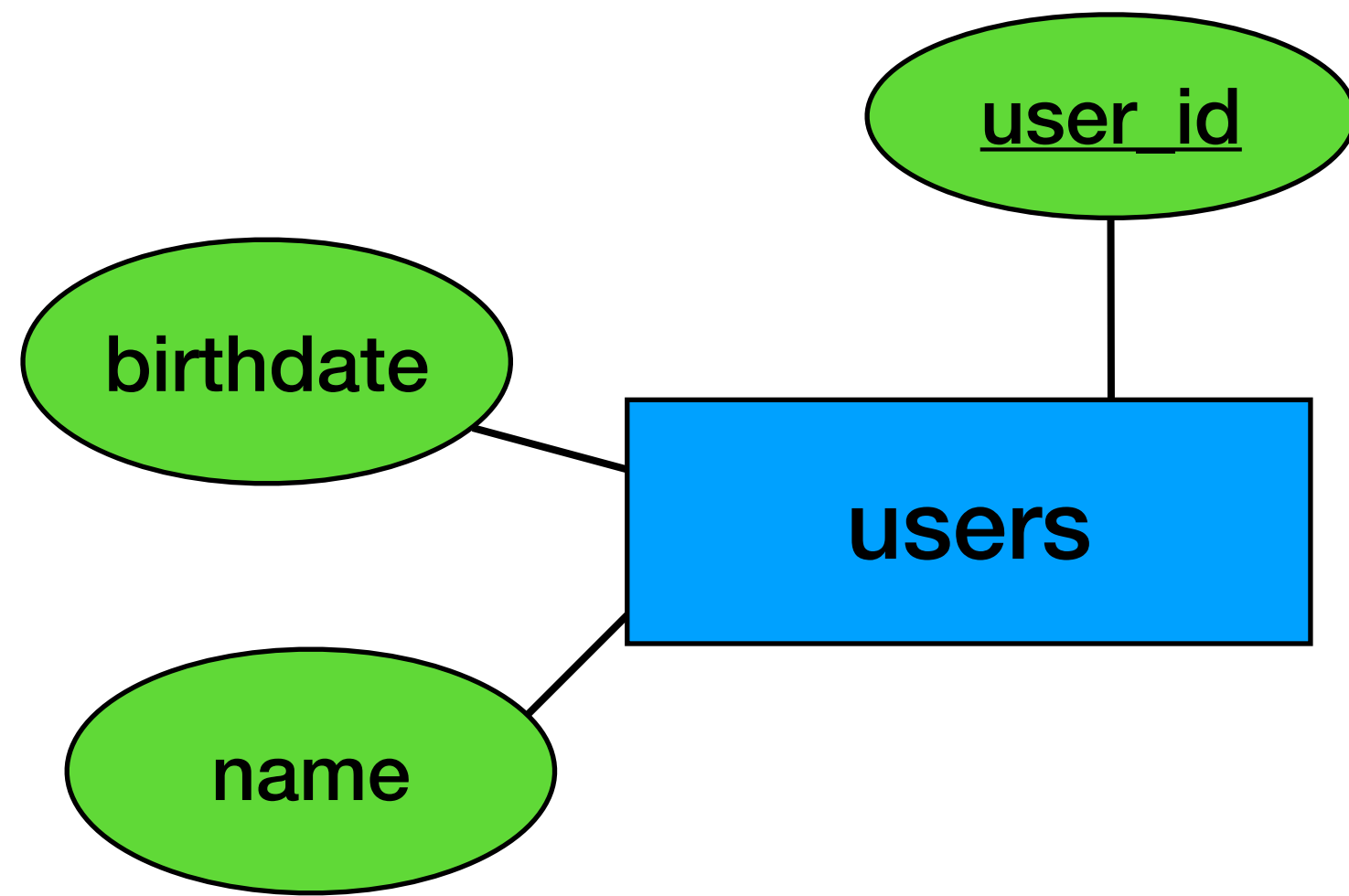
Design an ER diagram for a video platform:

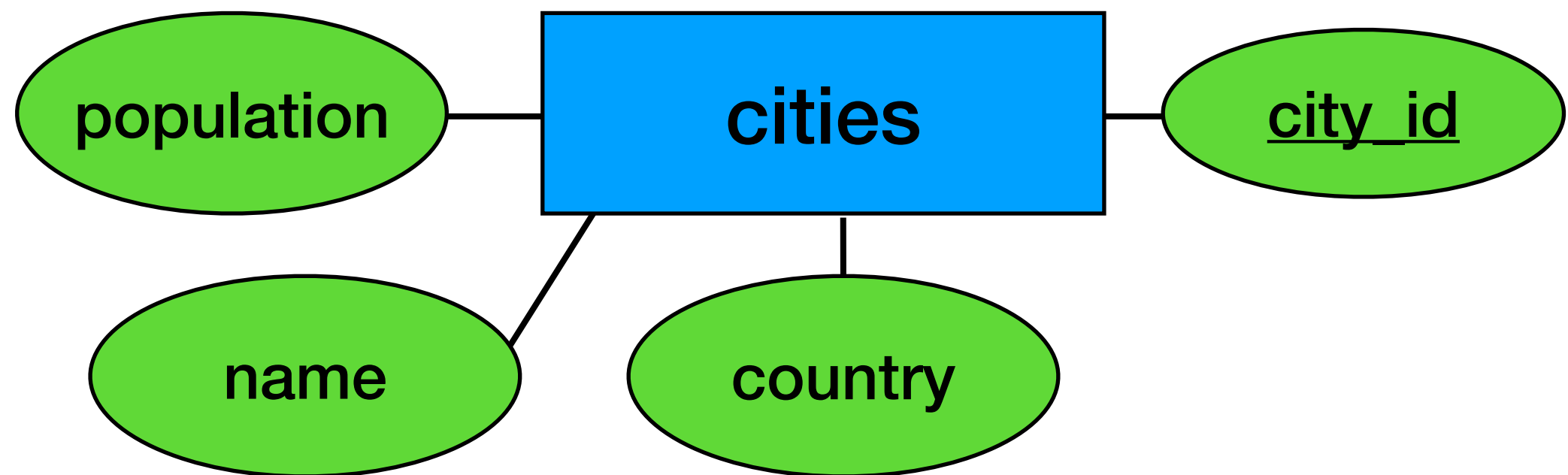
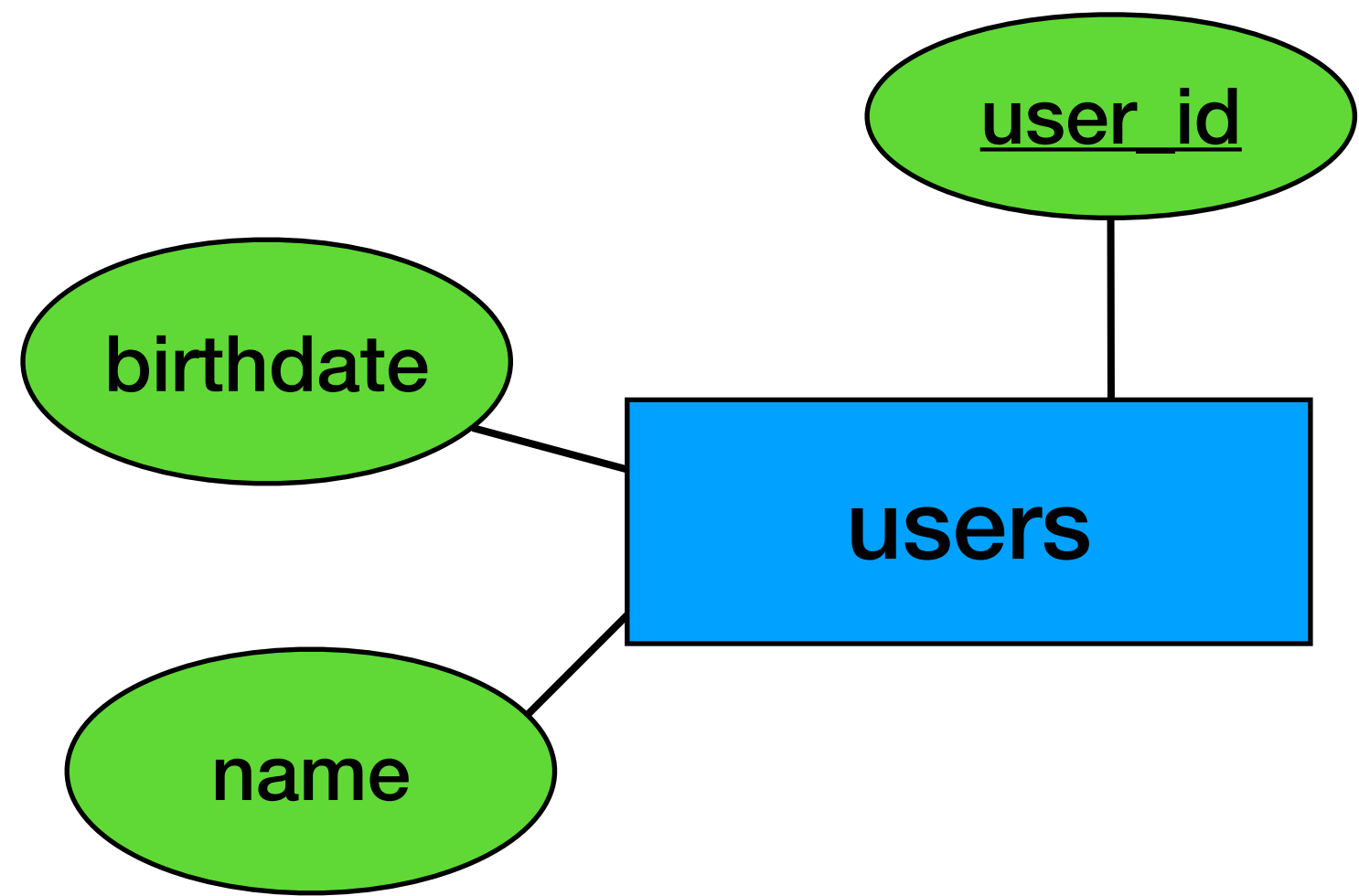
- A user is defined by `user_id`. We also save her name, birthdate and city. For each city we save the `city_id`, name, population and country
- A video is defined by a `video_id` and we store its genre, release date and title
- For each video we keep the actors that appears in it along with their character name.
- The actors are defined by an `actor_id` along with their name
- For analytics, if a user views a video we save the most recent viewing timestamp

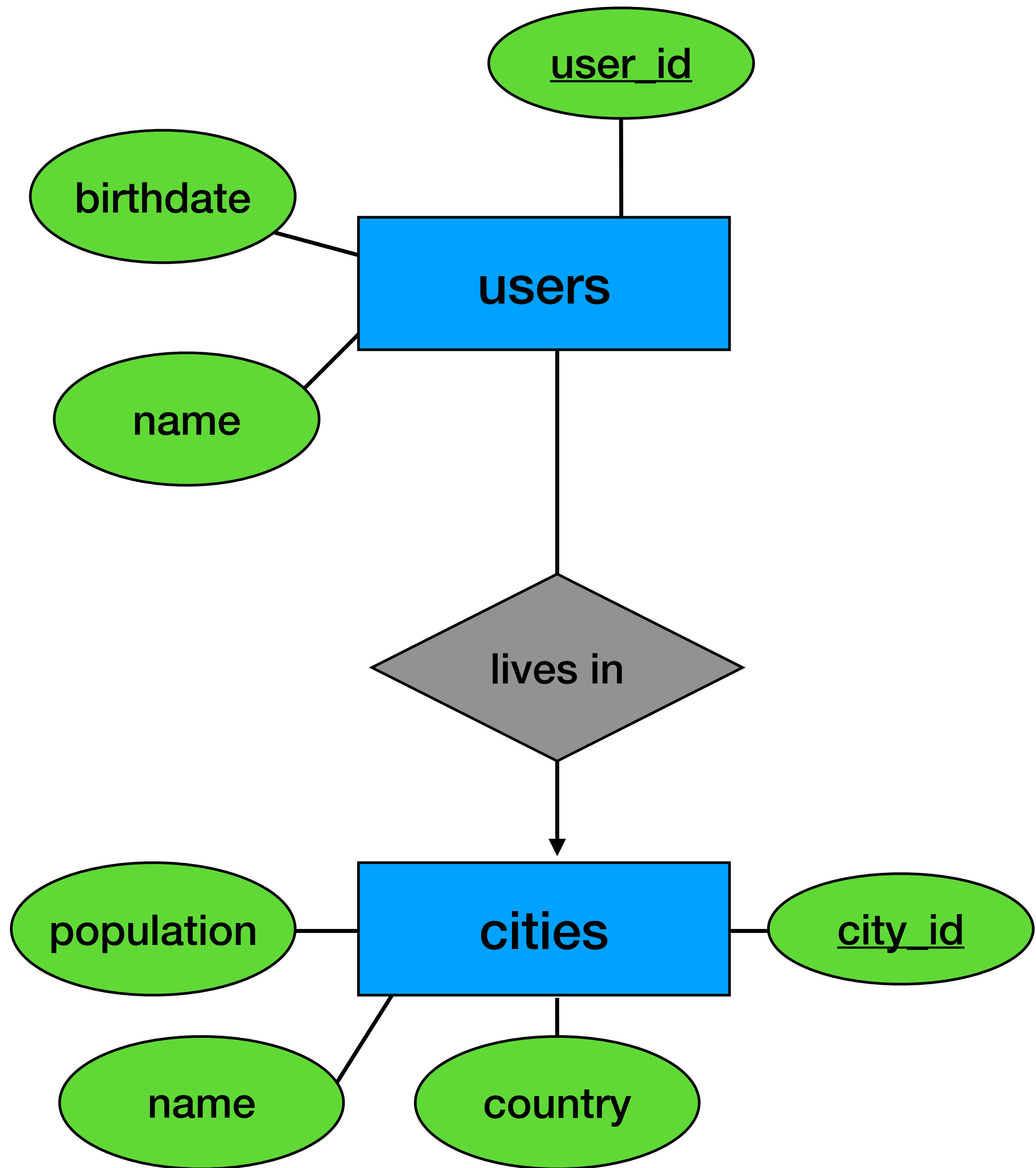


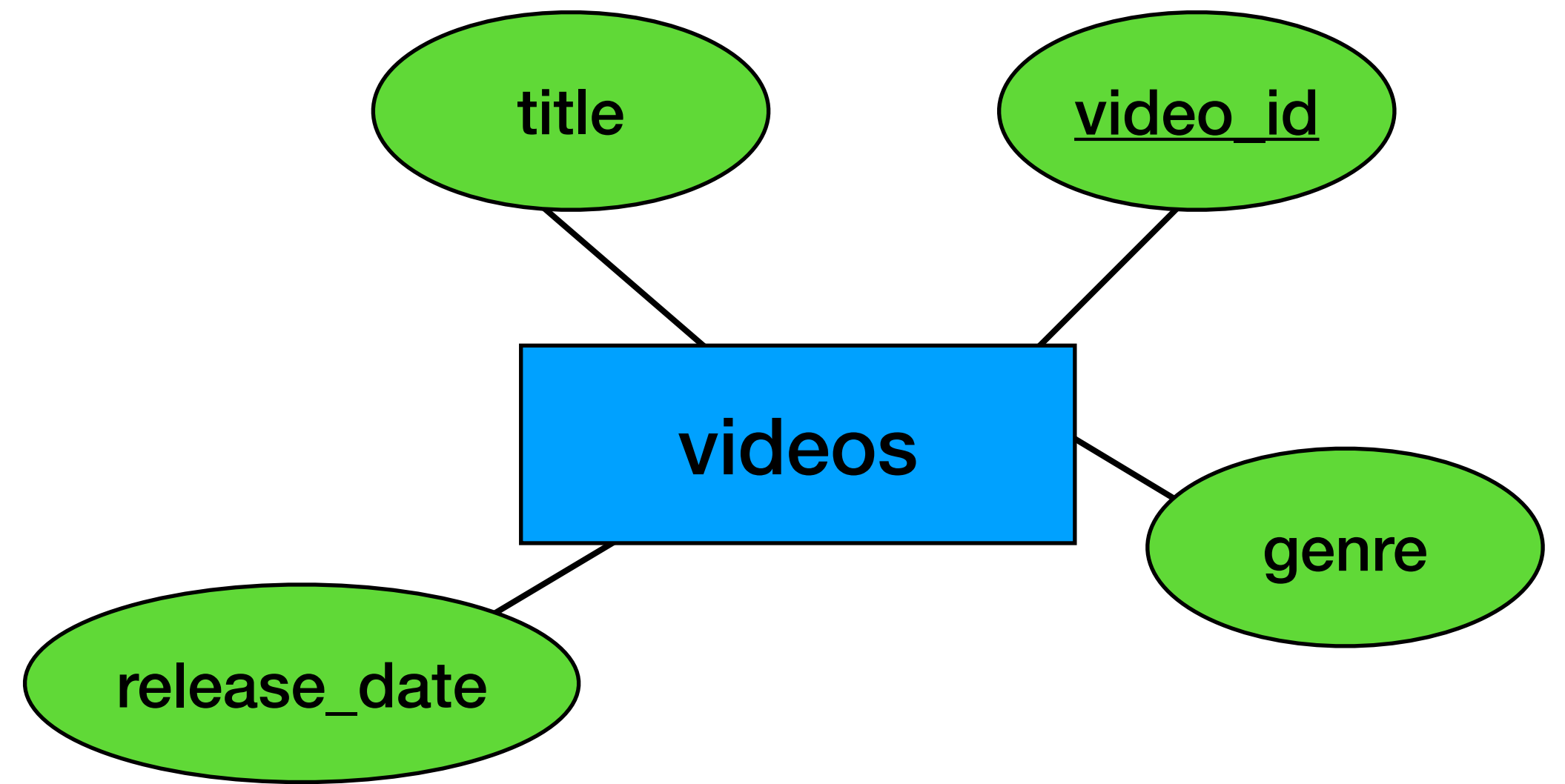
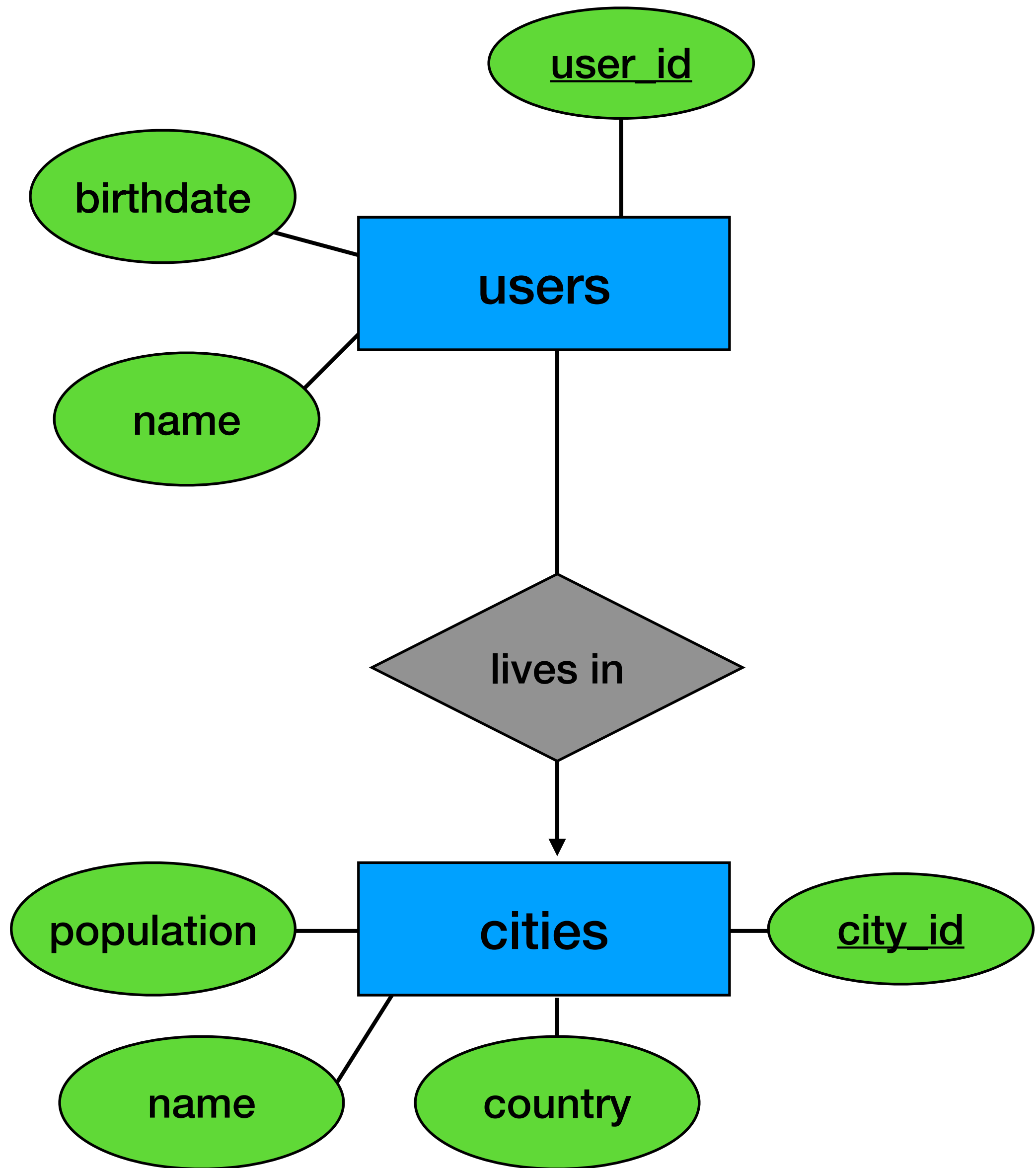


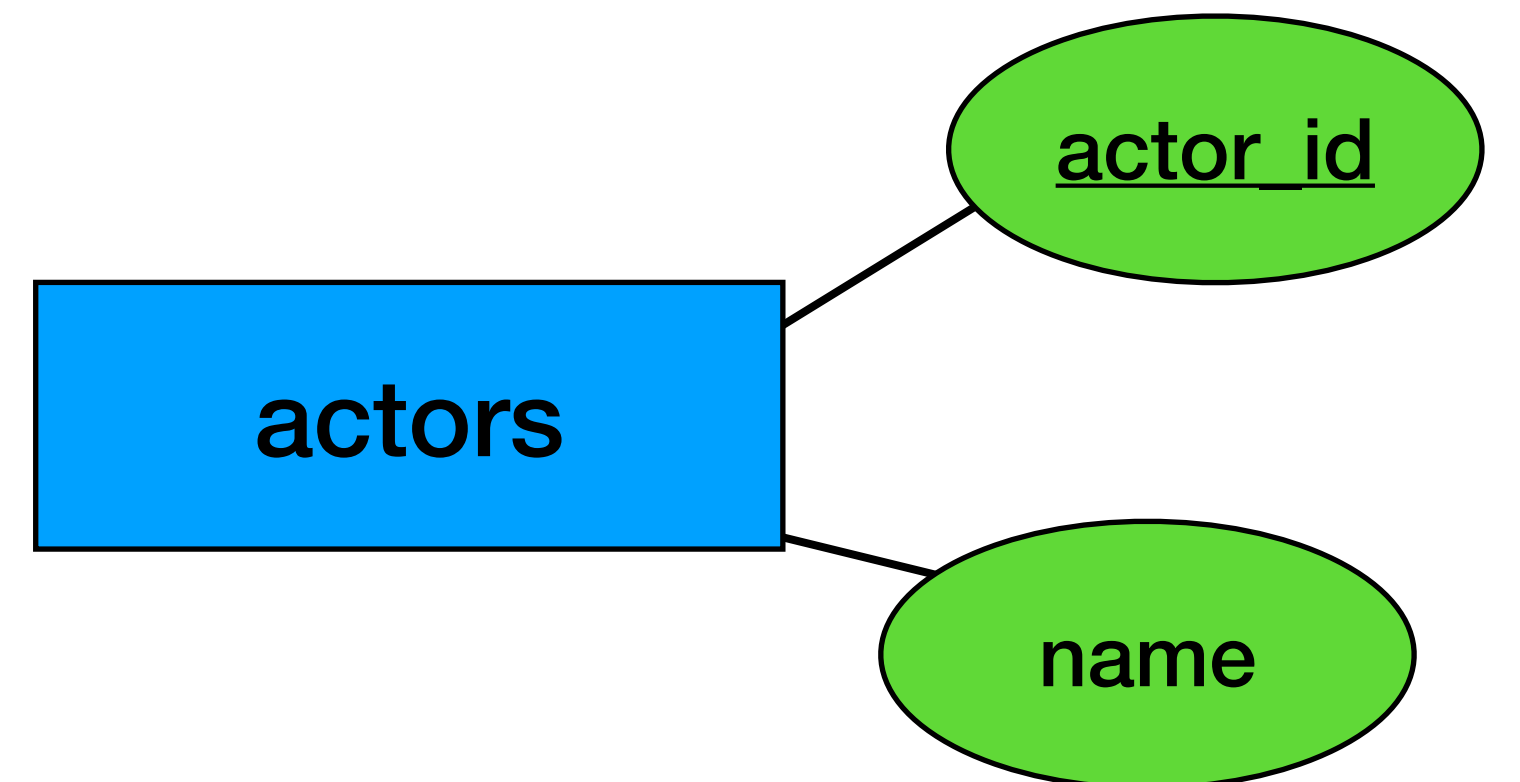
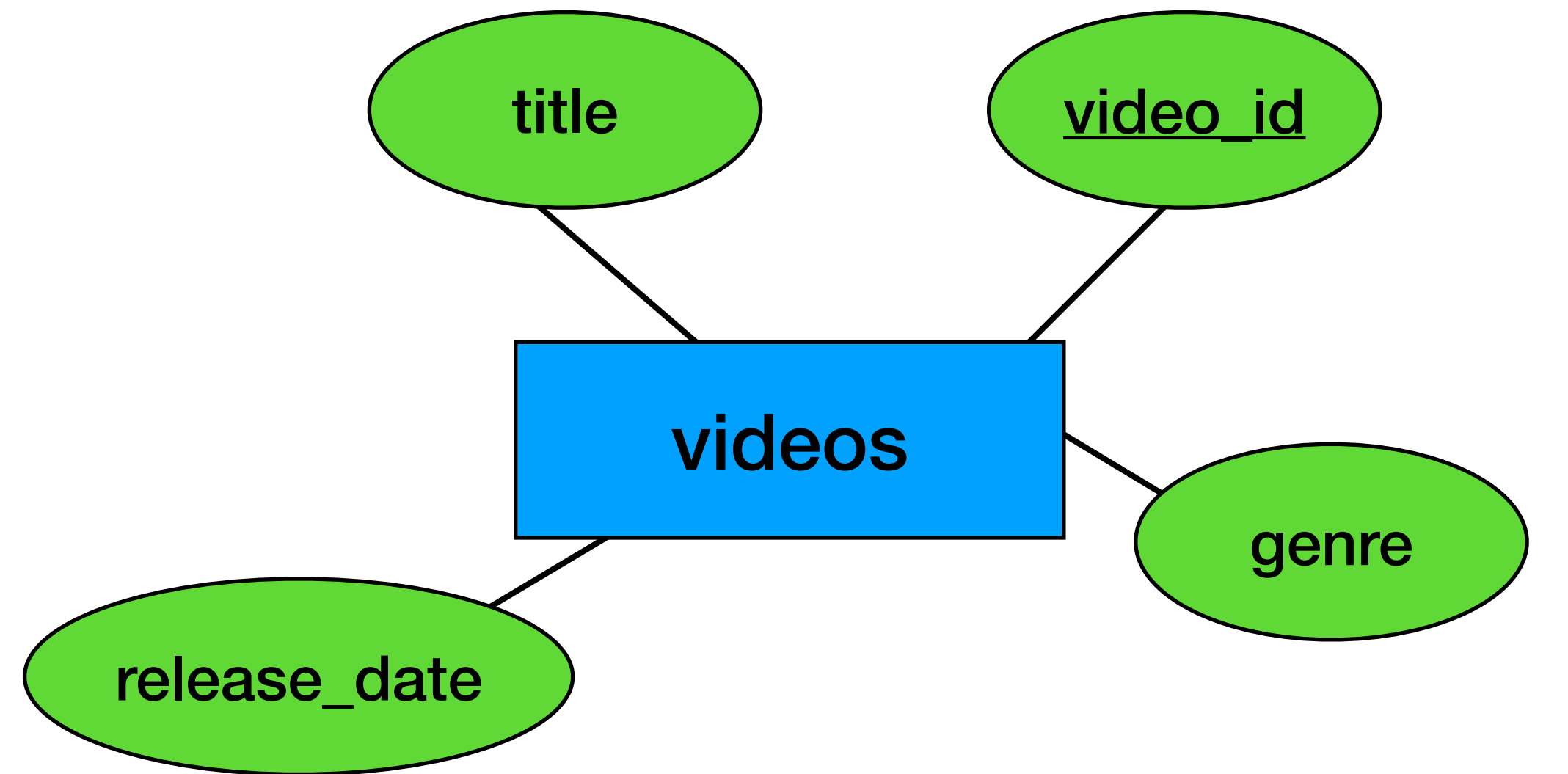
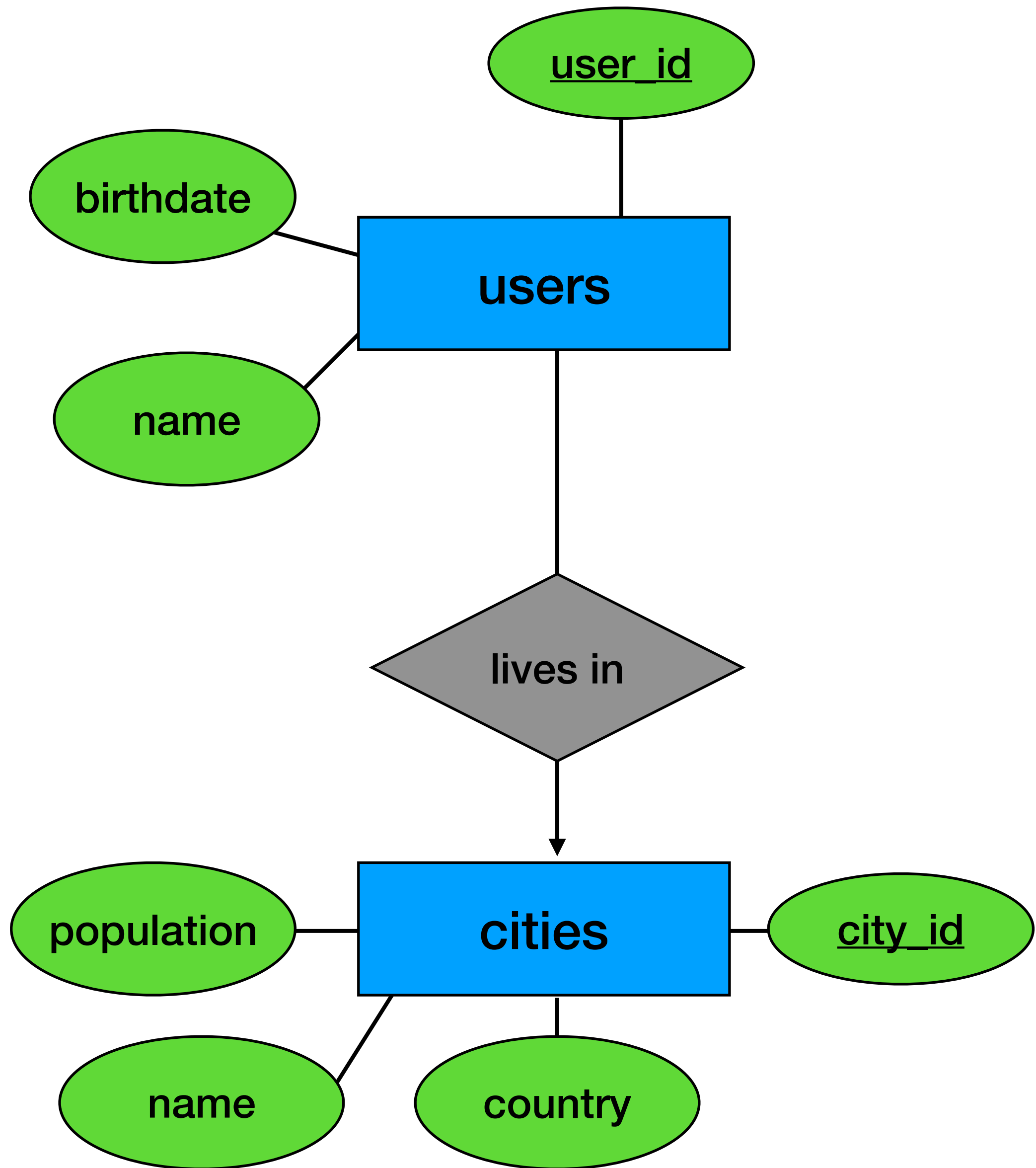


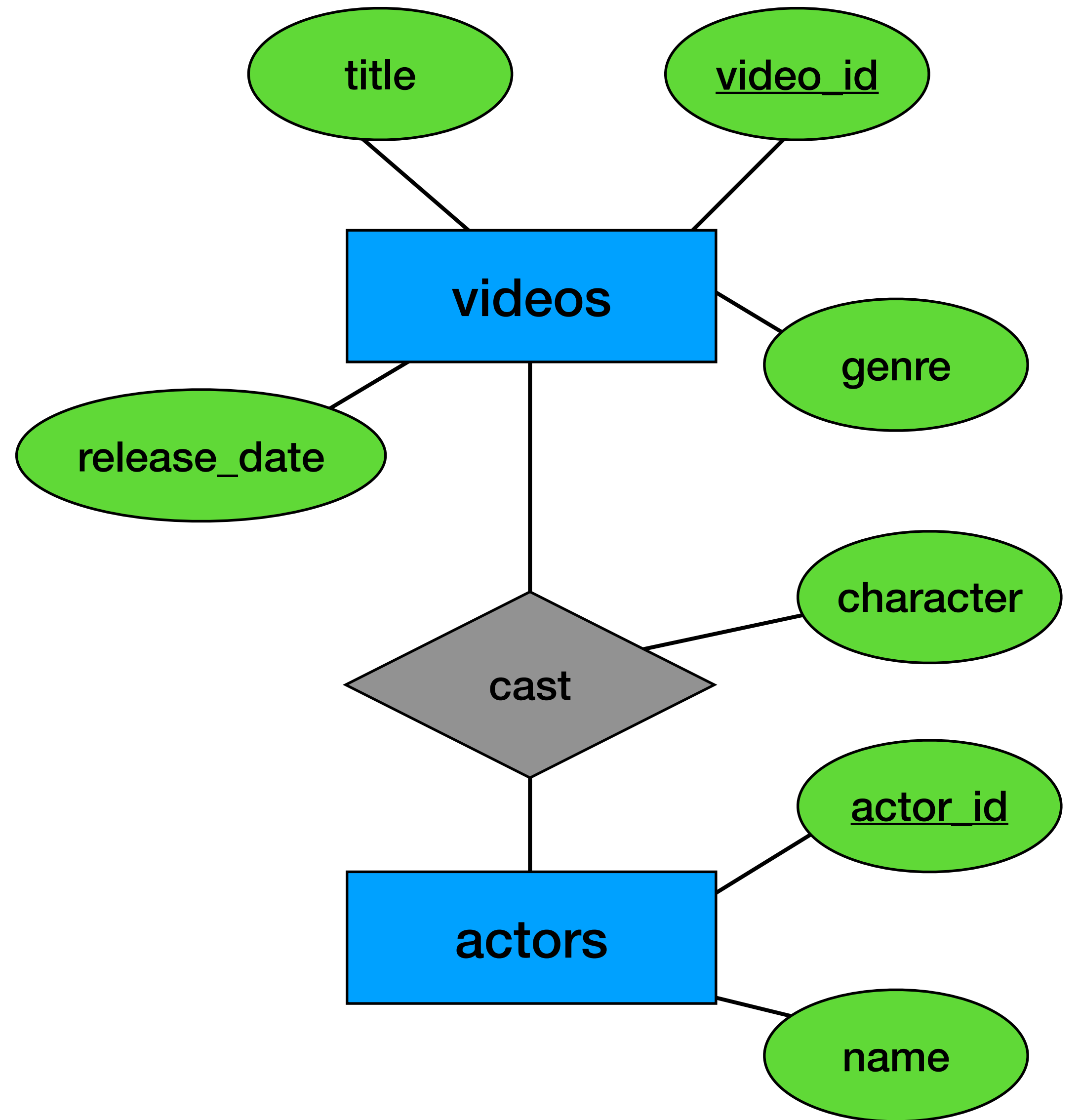
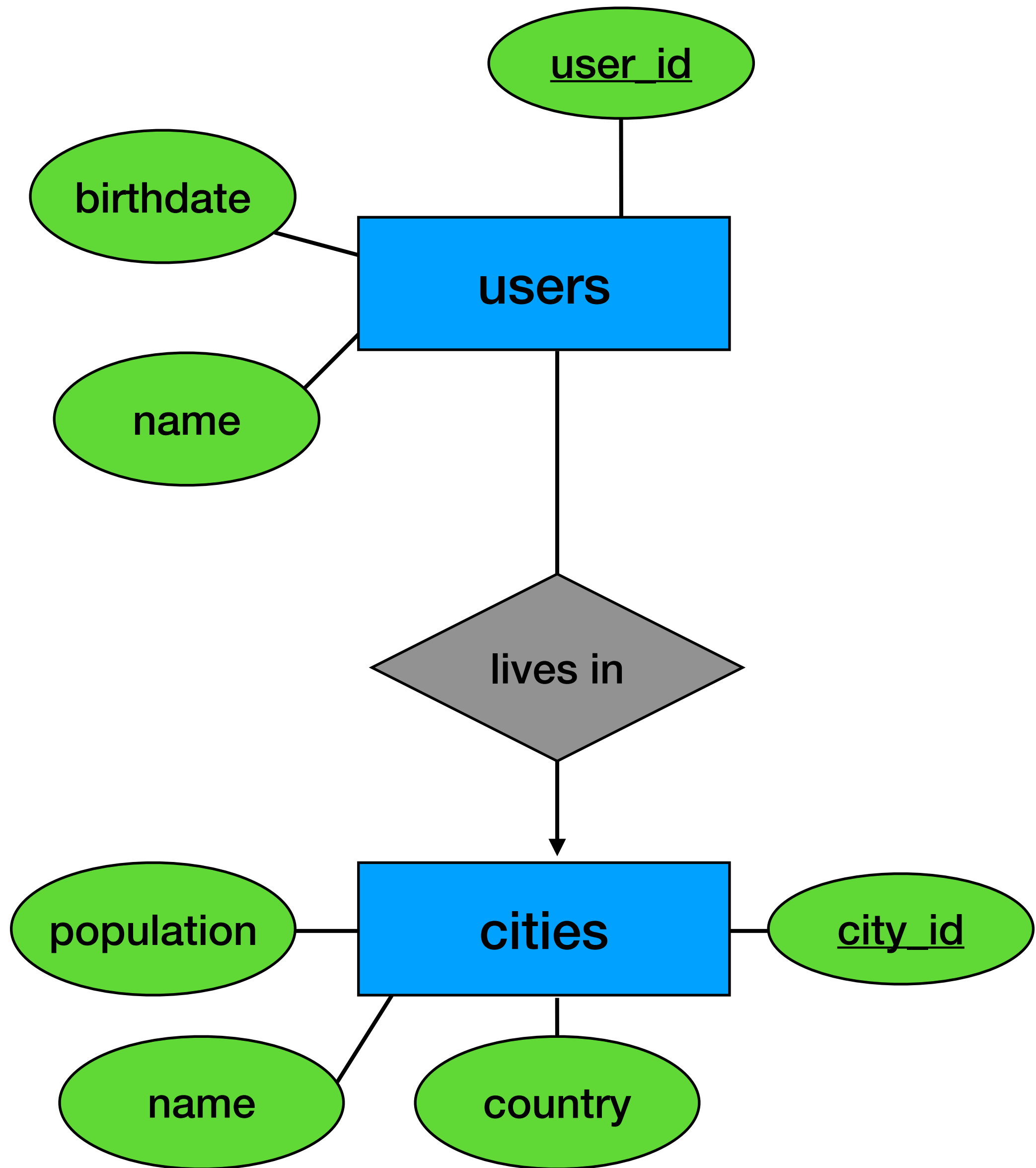


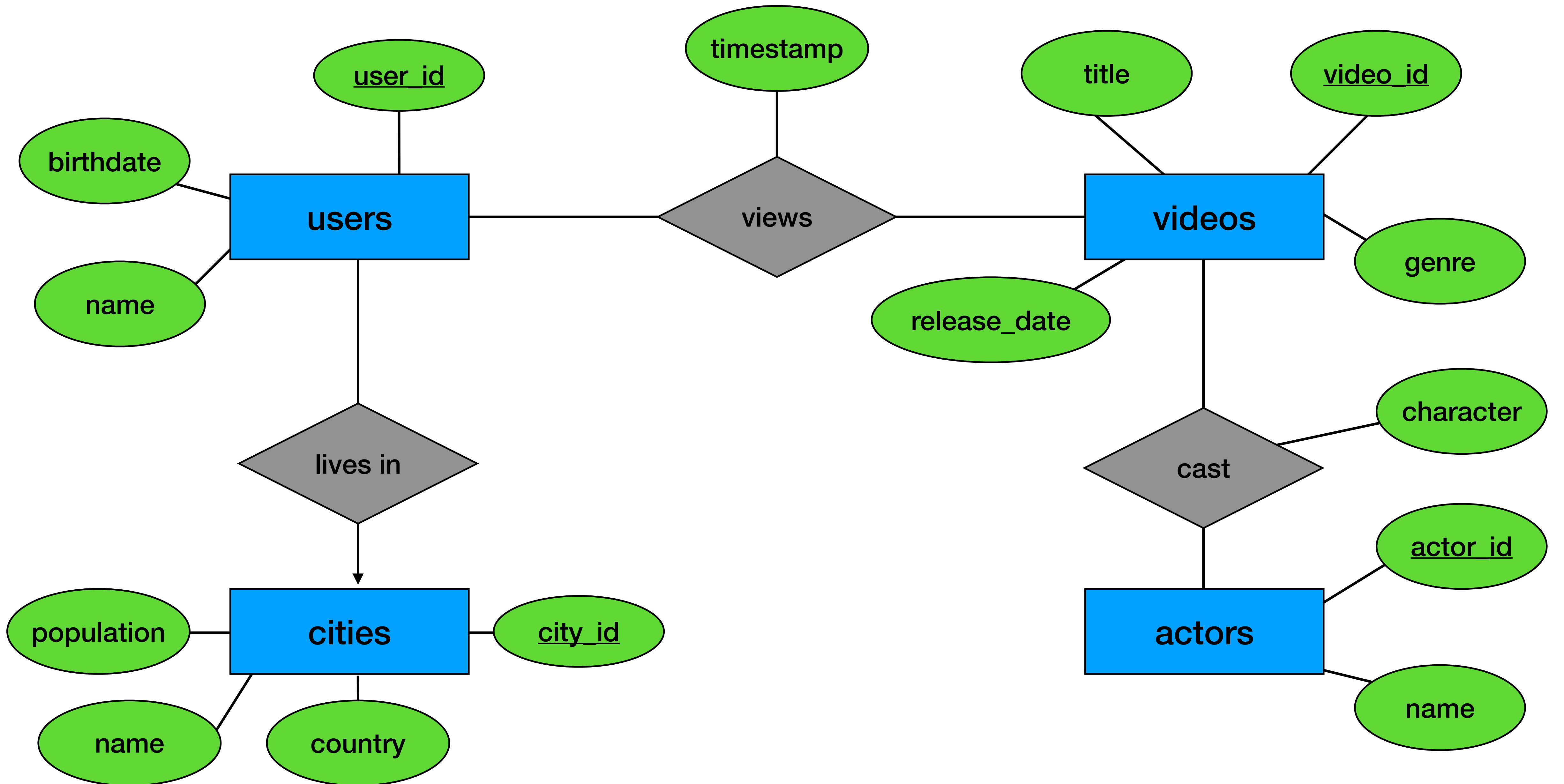






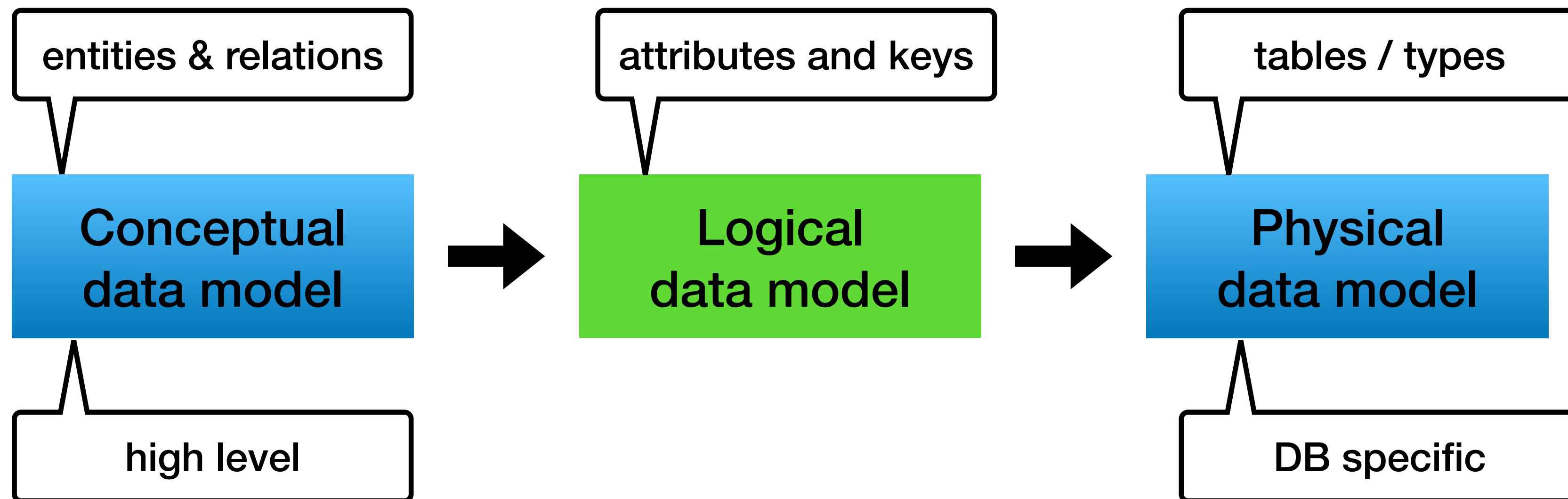








# Relational Modeling - 10,000 foot view

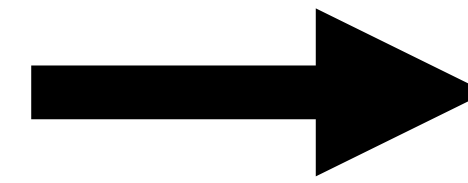


# Logical data model

- From concept the “schema”
- Keys, foreign keys
- Data types are not yet defined

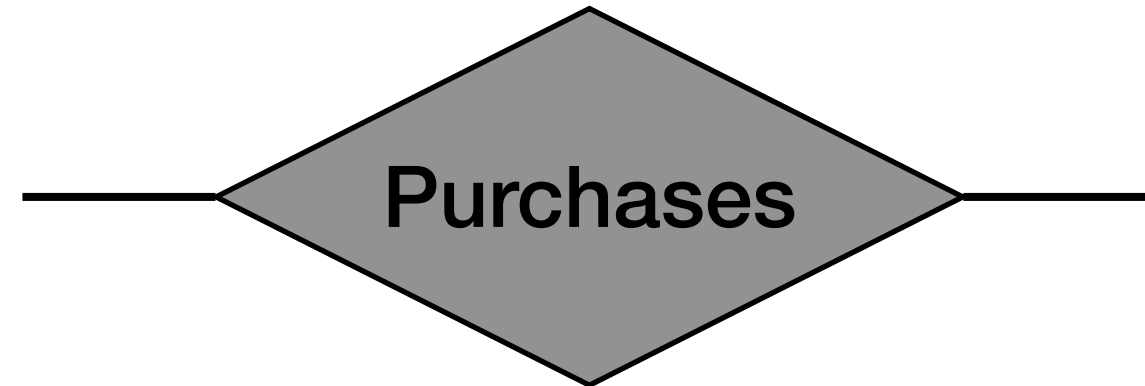
# ER to Relational schema

- Entities



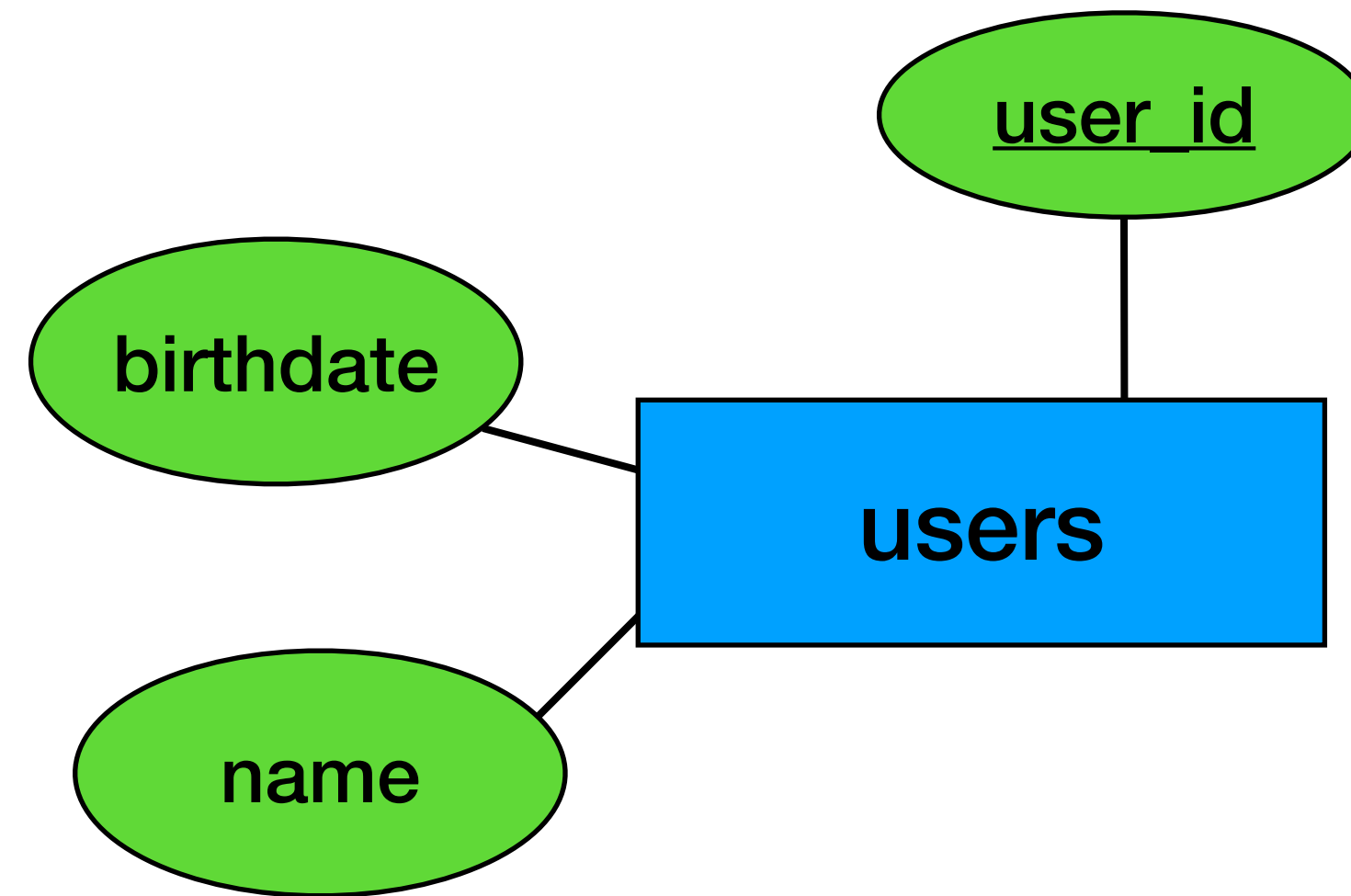
relation

- Relations

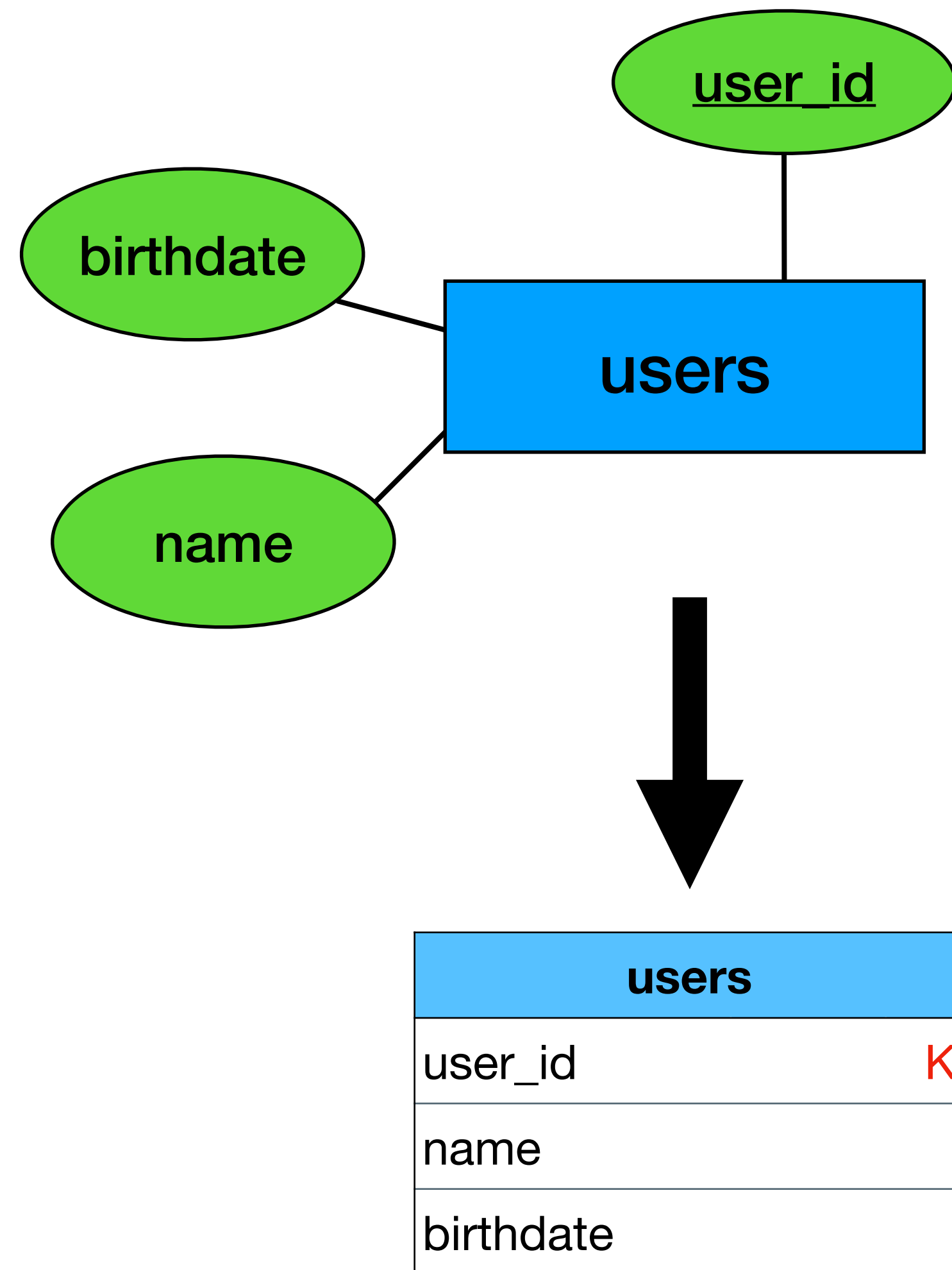


relation

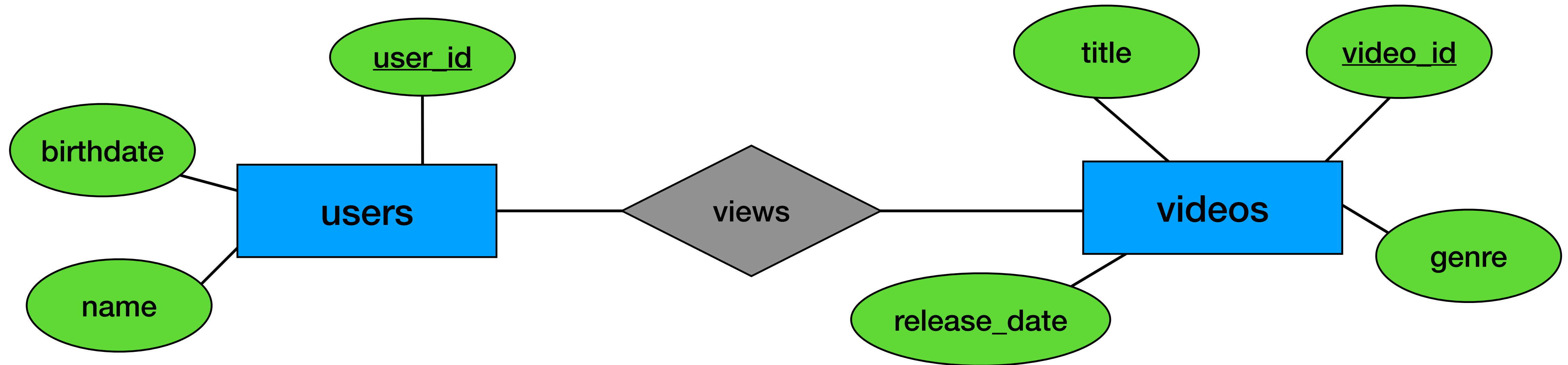
# Entity to Relation



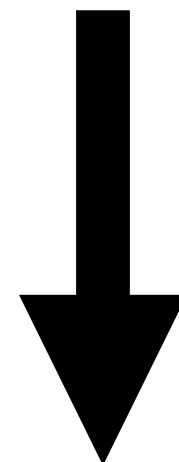
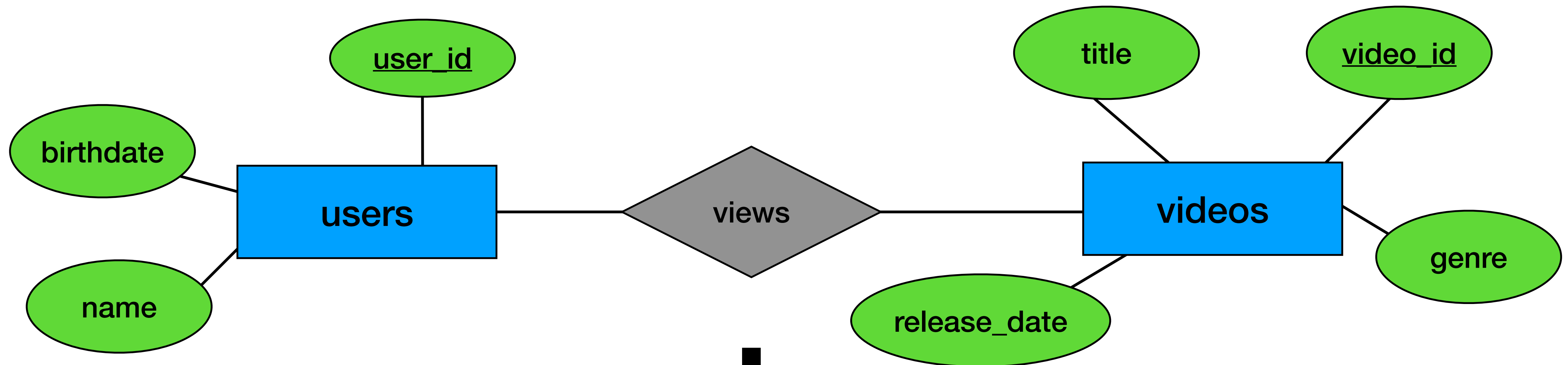
# Entity to Relation



# Relation to Relation (many-to-many)



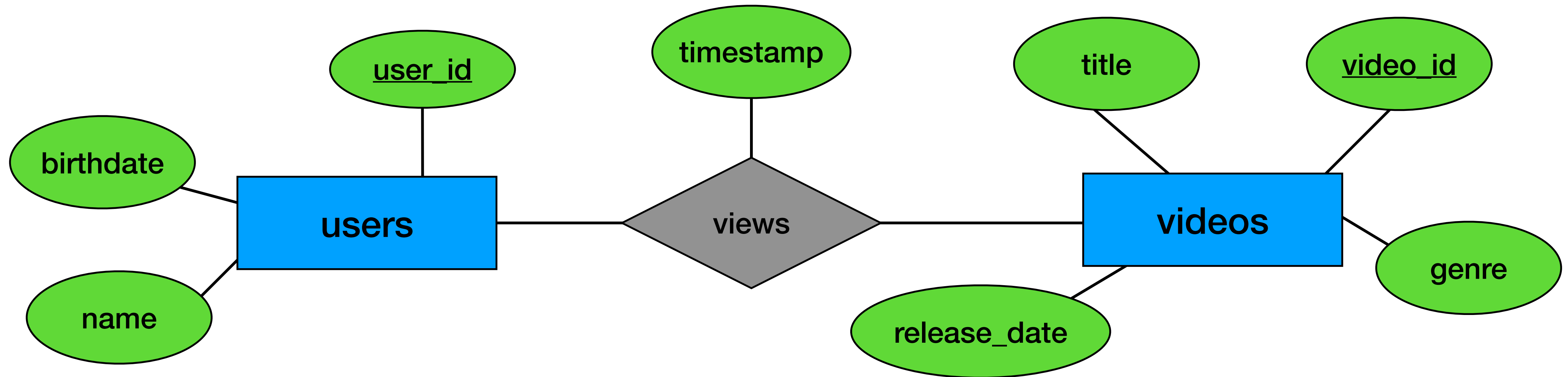
# Relation to Relation (many-to-many)



Keys are derived from entities

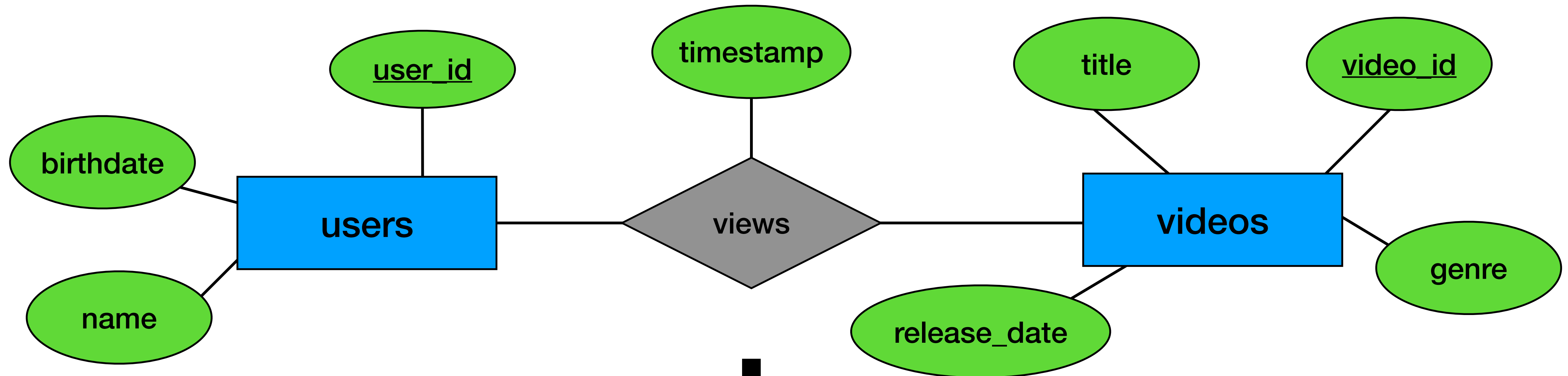
views	
user_id	K
video_id	K

# Relation to Relation (+attributes)





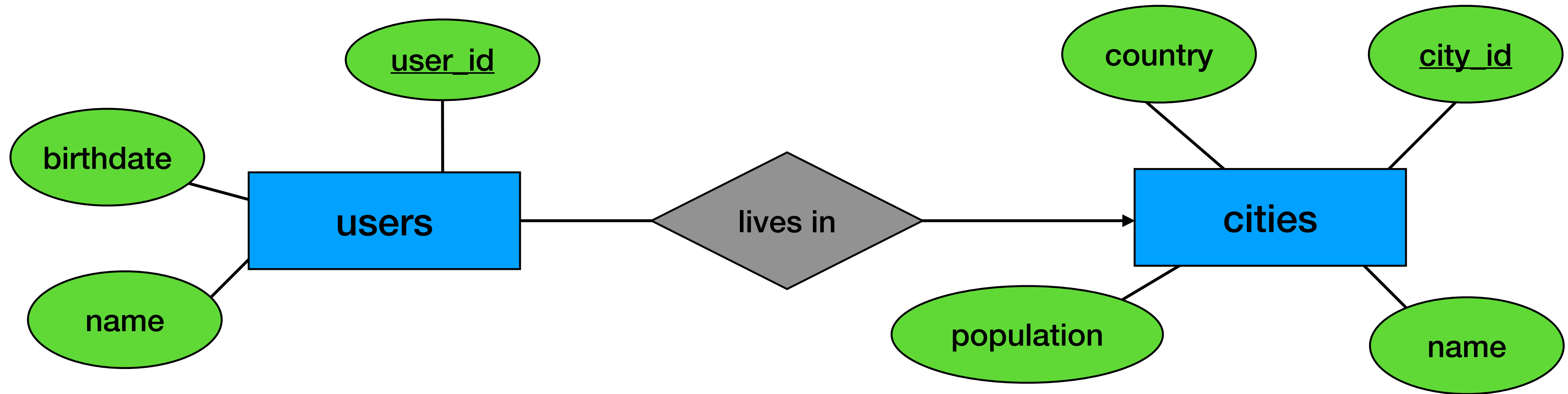
# Relation to Relation (+attributes)



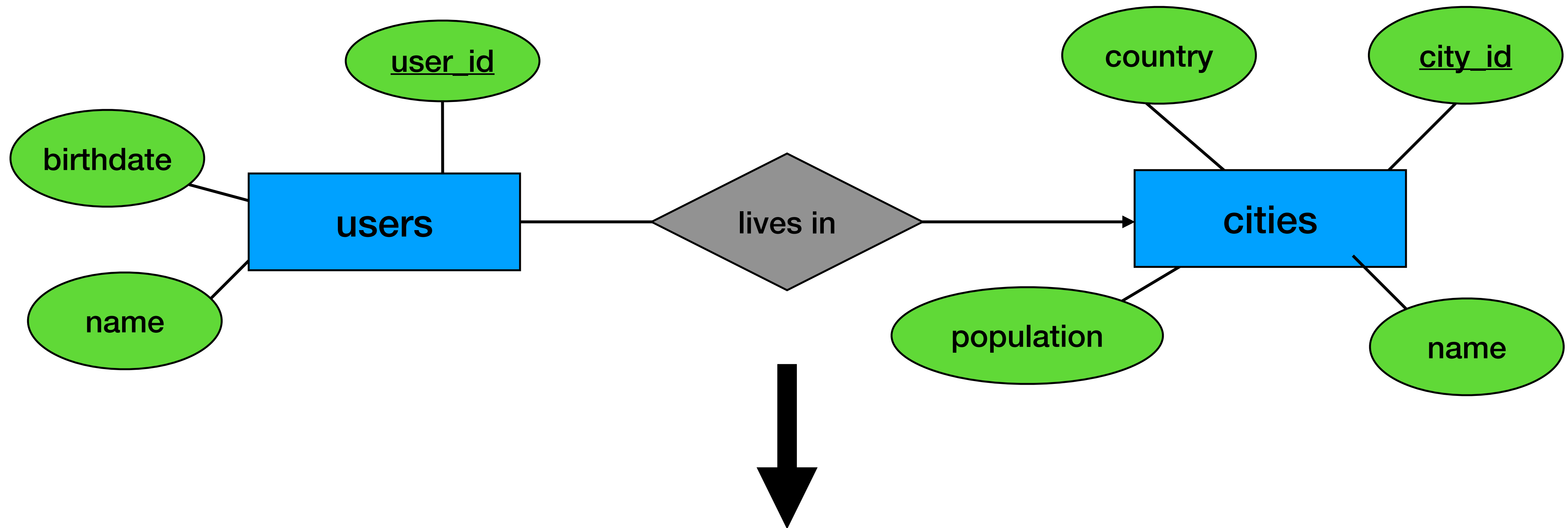
Additional attributes

views	
user_id	K
video_id	K
timestamp	

# Relation to Relation (many-to-one)



# Relation to Relation (many-to-one)

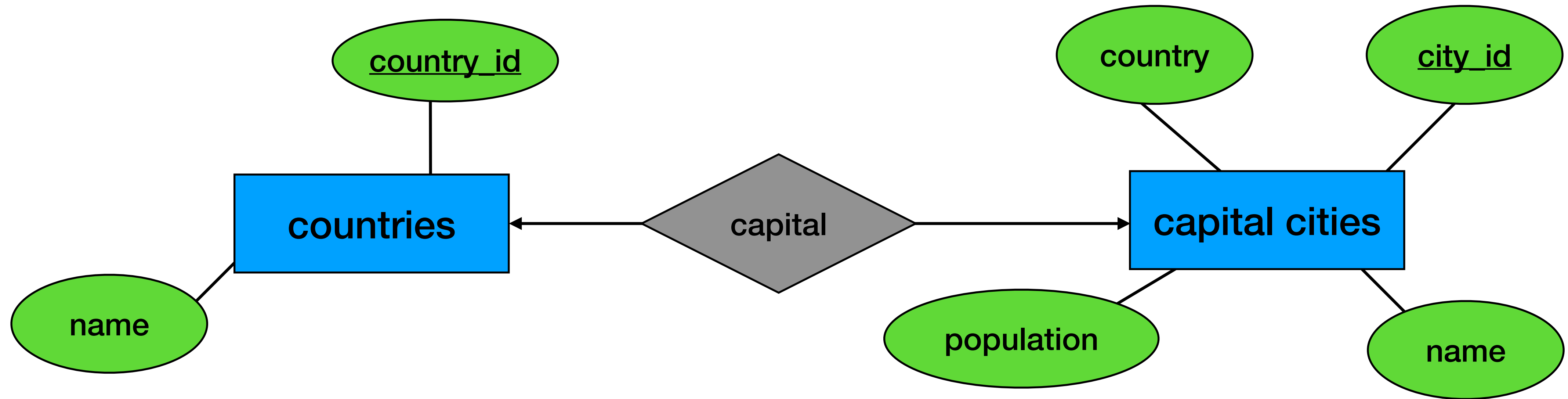


No additional table is required. We add to users the key(s) of cities

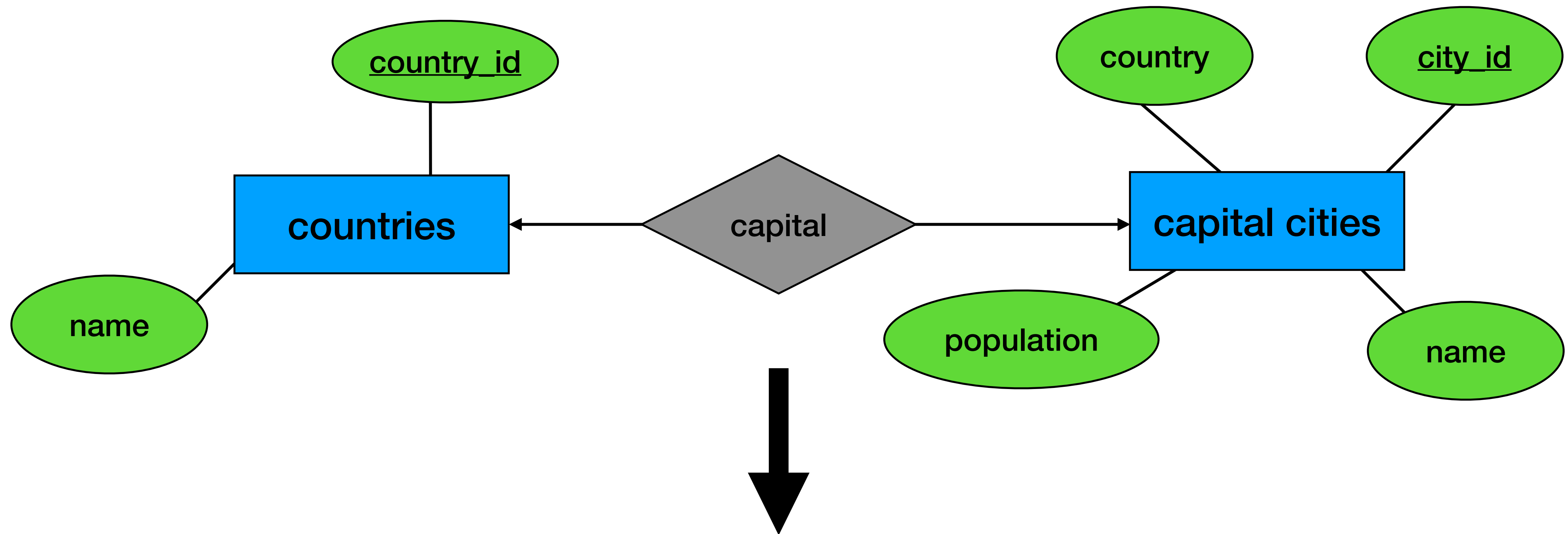
users	
user_id	K
name	
birthdate	
city_id	FK

Sometimes FKs are omitted and will be represented by arrows (see next slides)

# Relation to Relation (one-to-one)



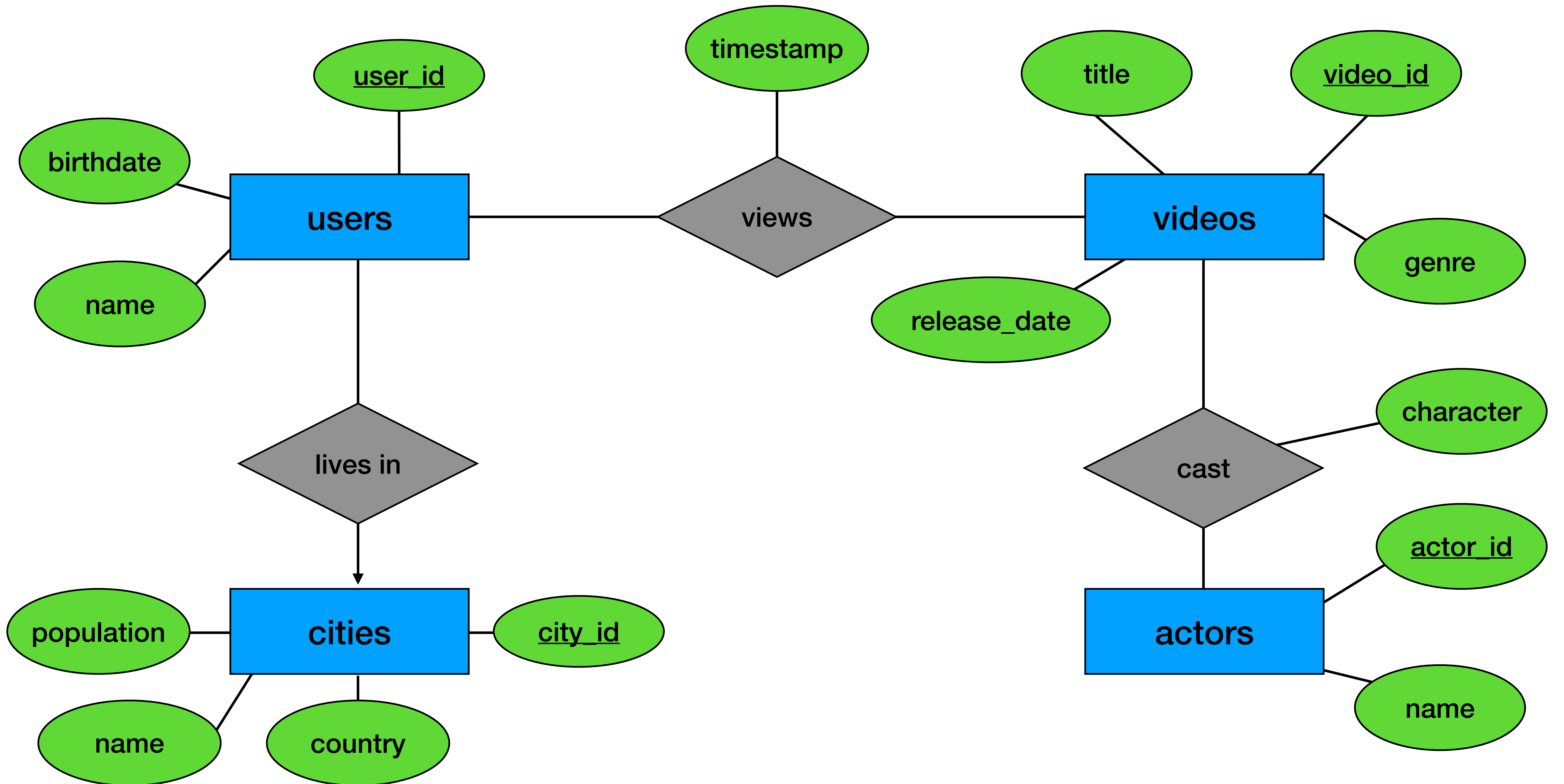
# Relation to Relation (one-to-one)

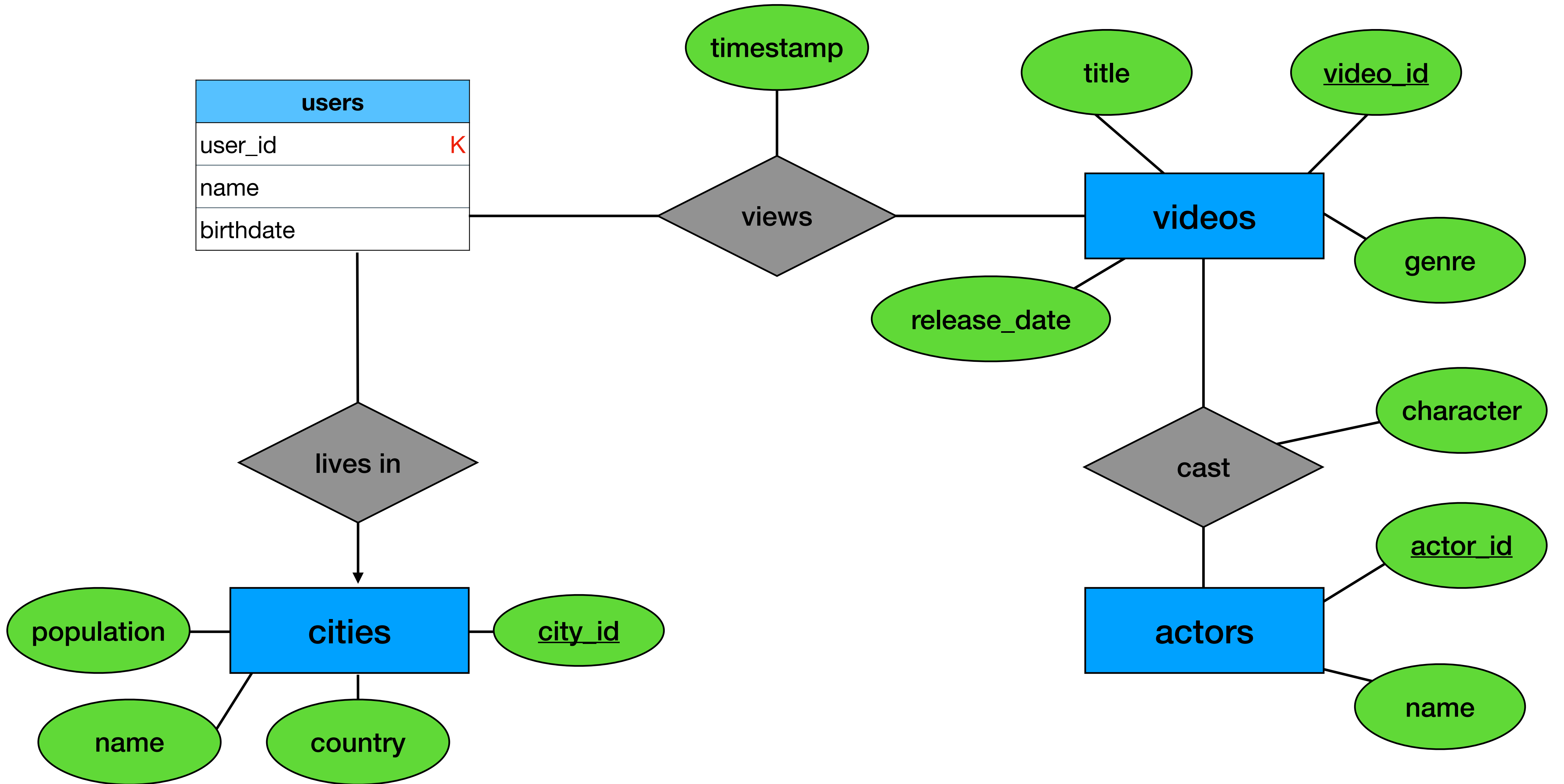


countries	
country_id	K
name	
city_id	FK, U

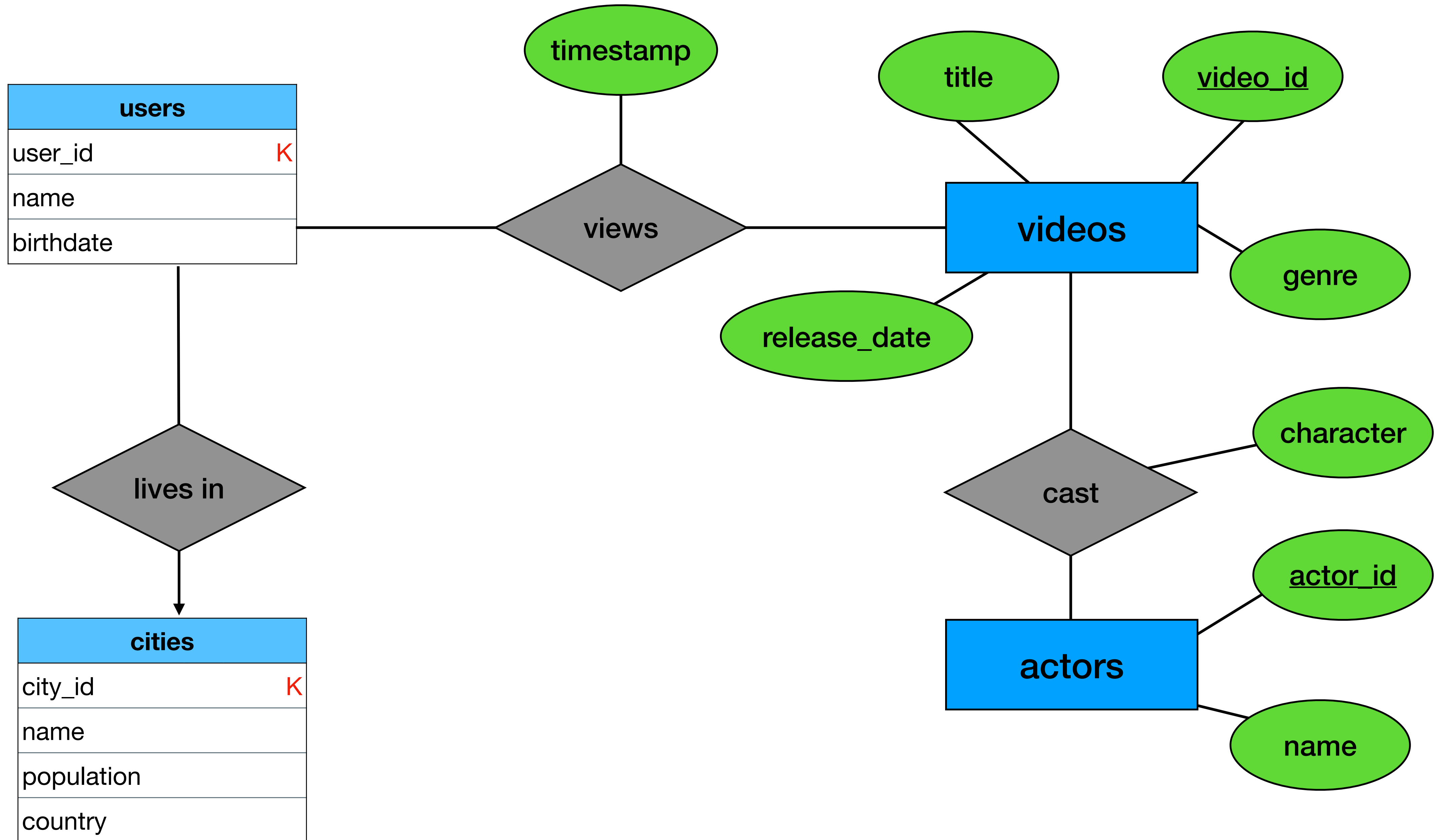
FK + Unique index

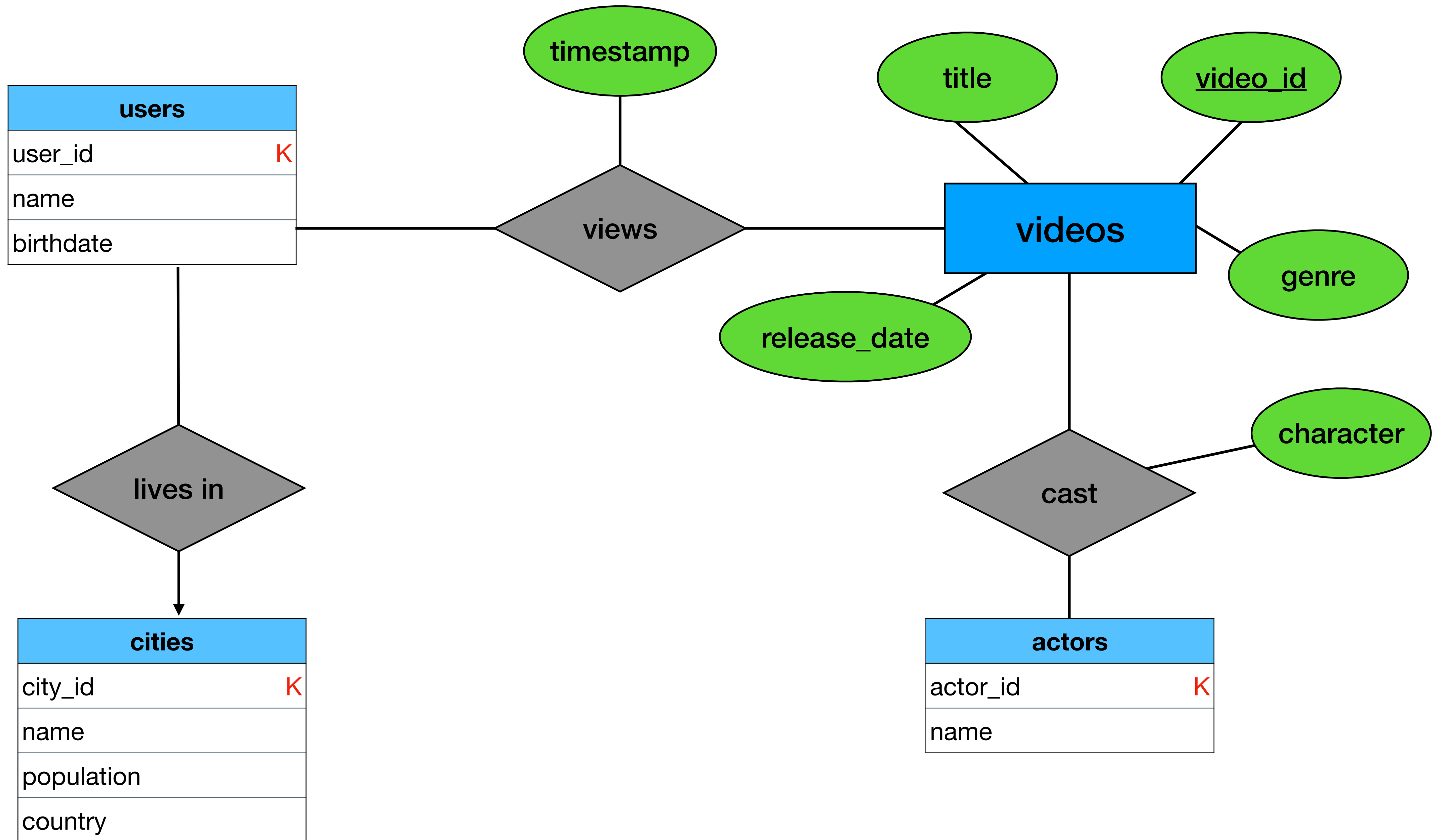
# Example

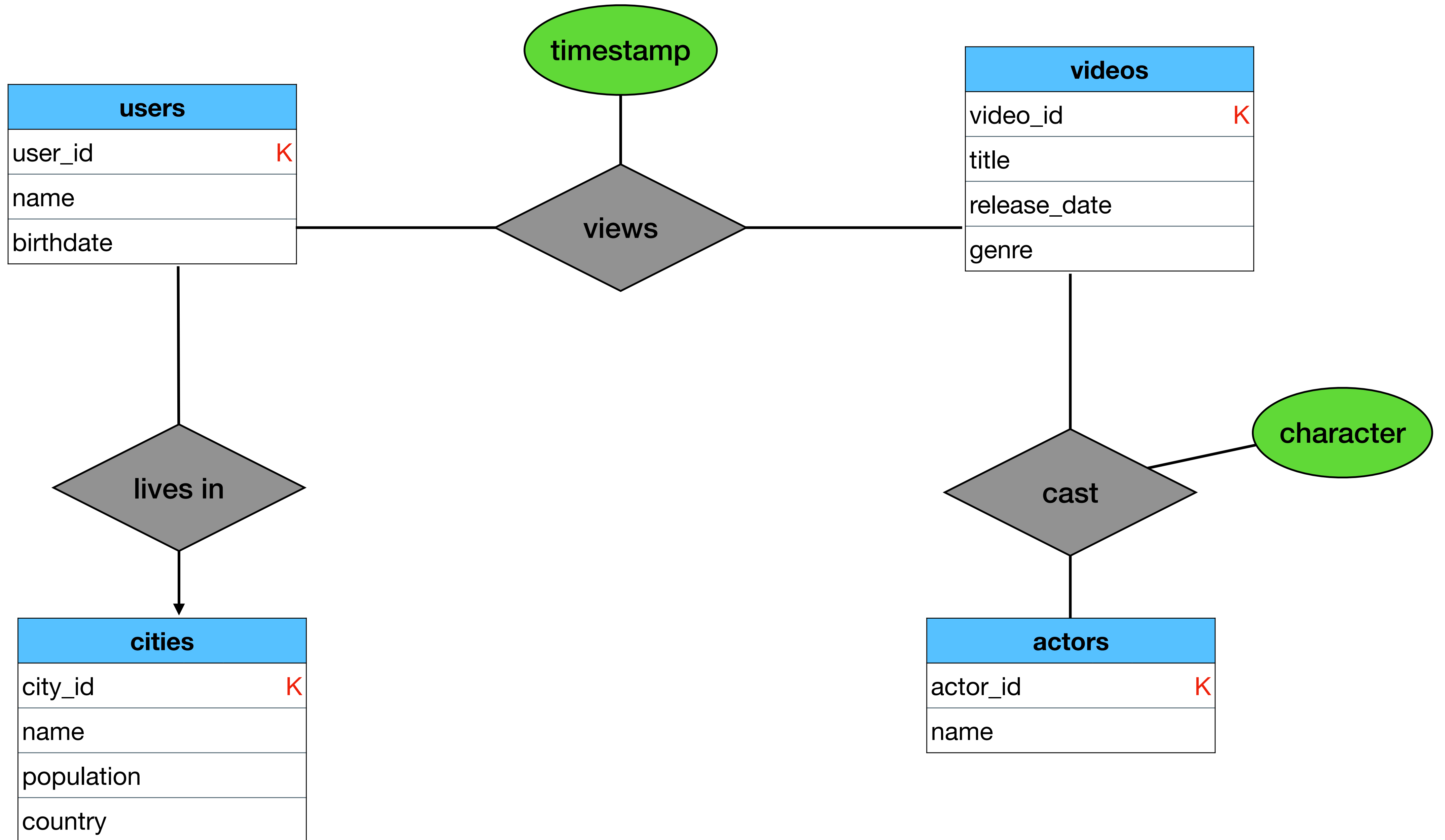


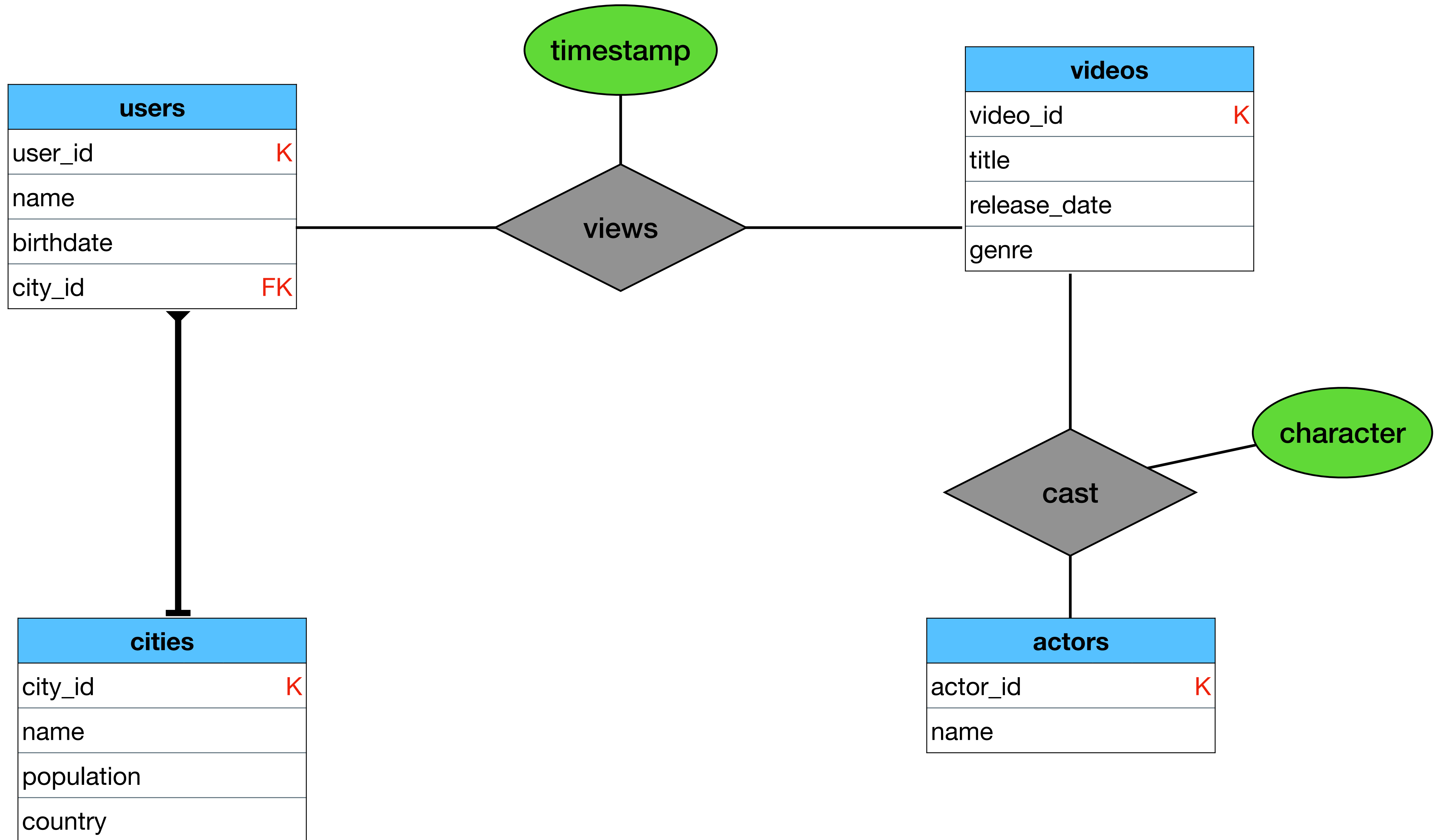


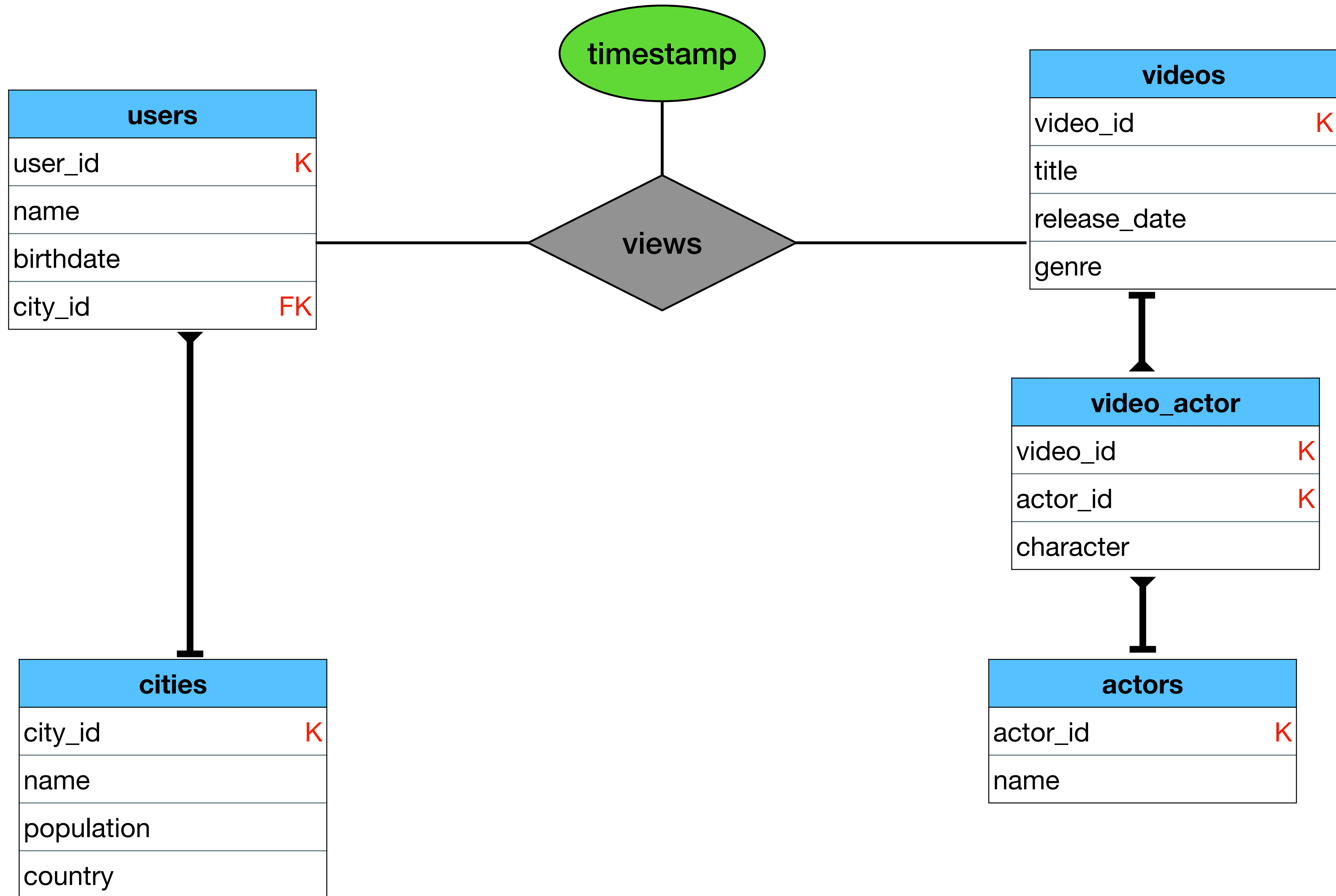


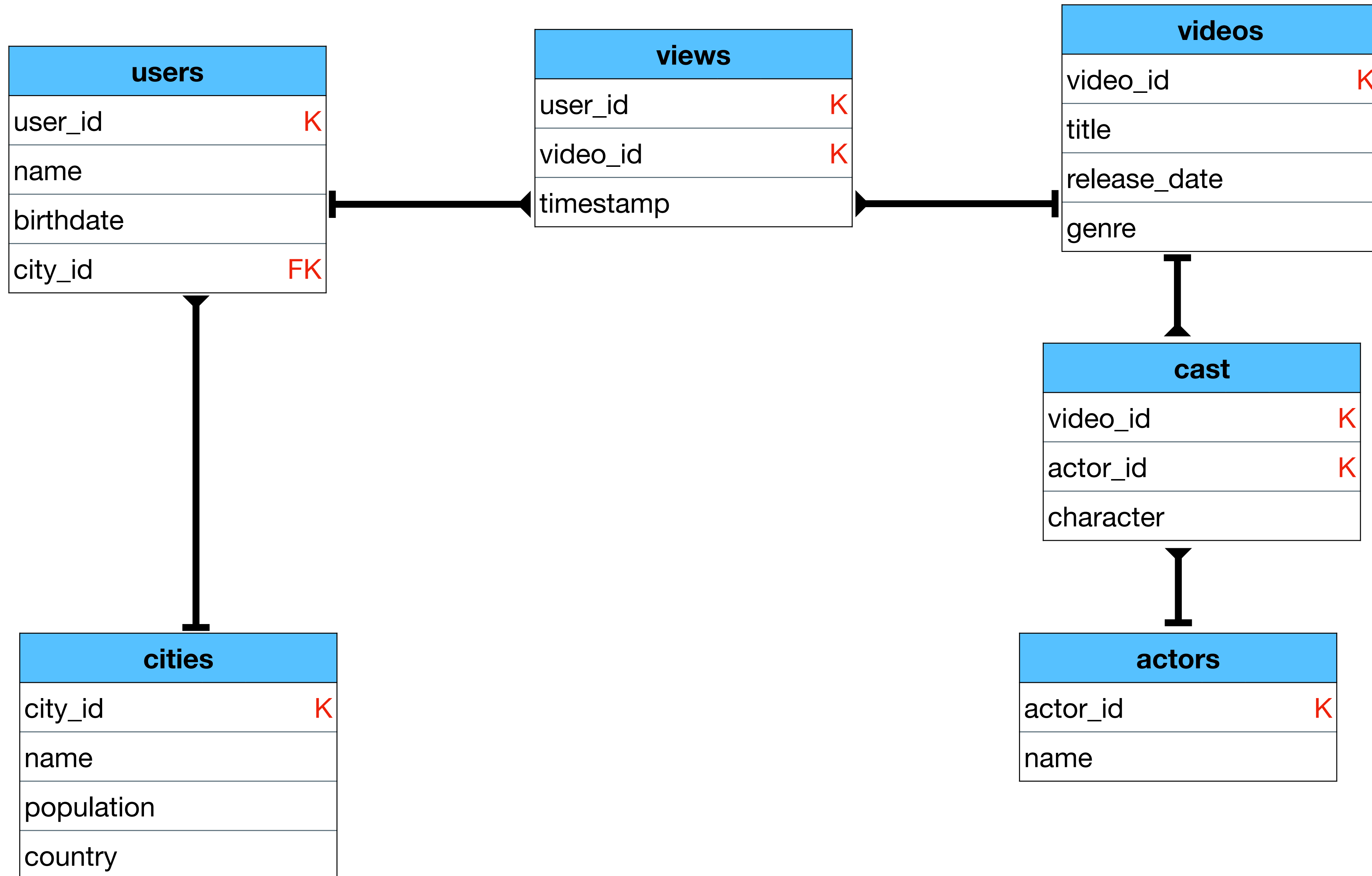


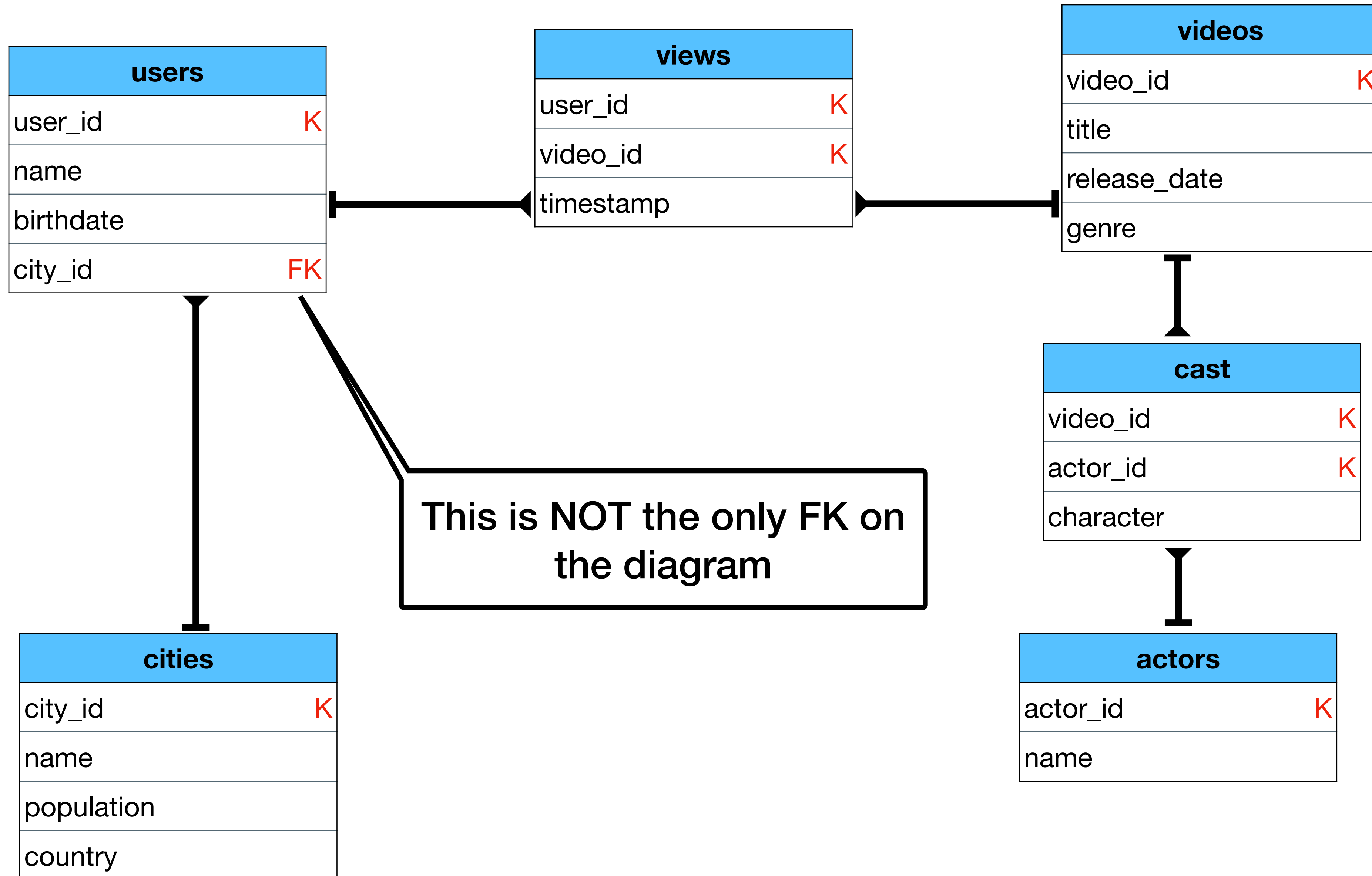




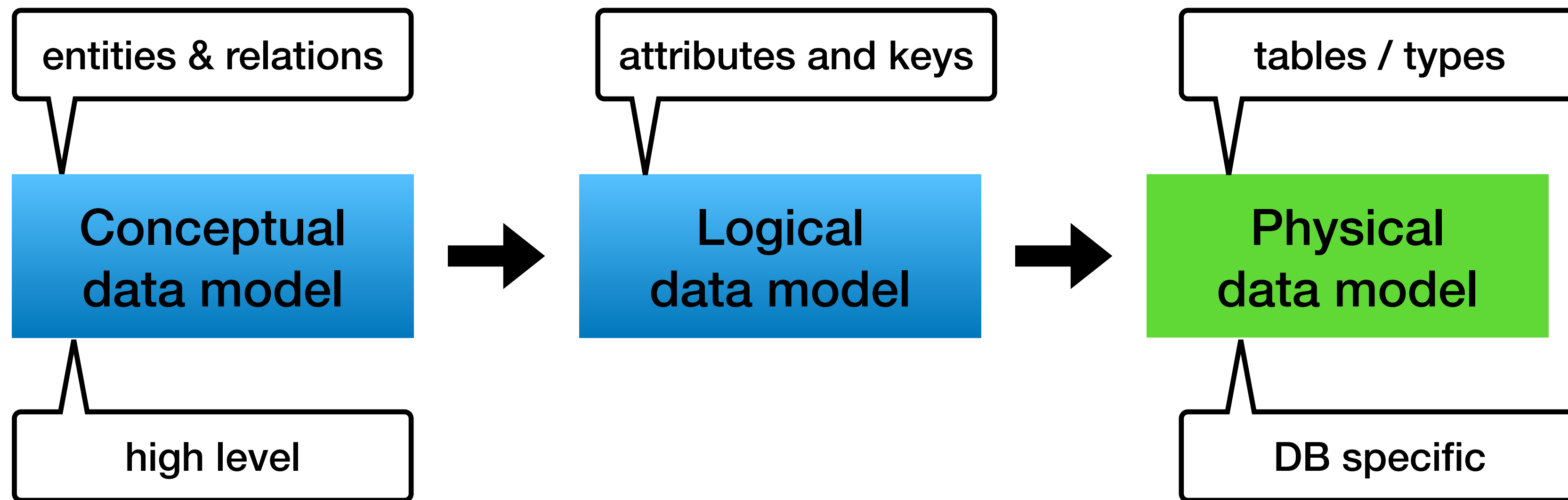








# Relational Modeling - 10,000 foot view

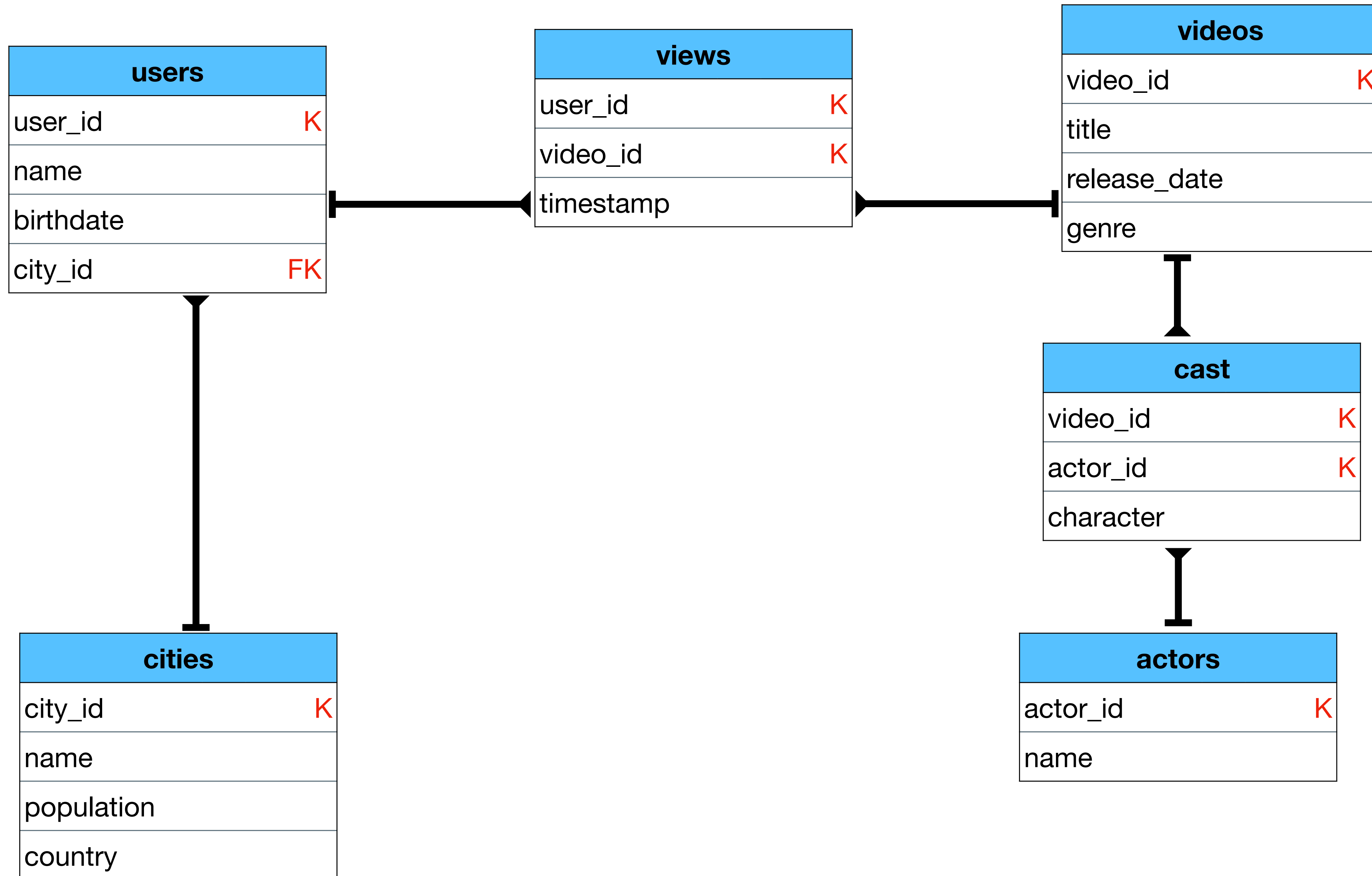


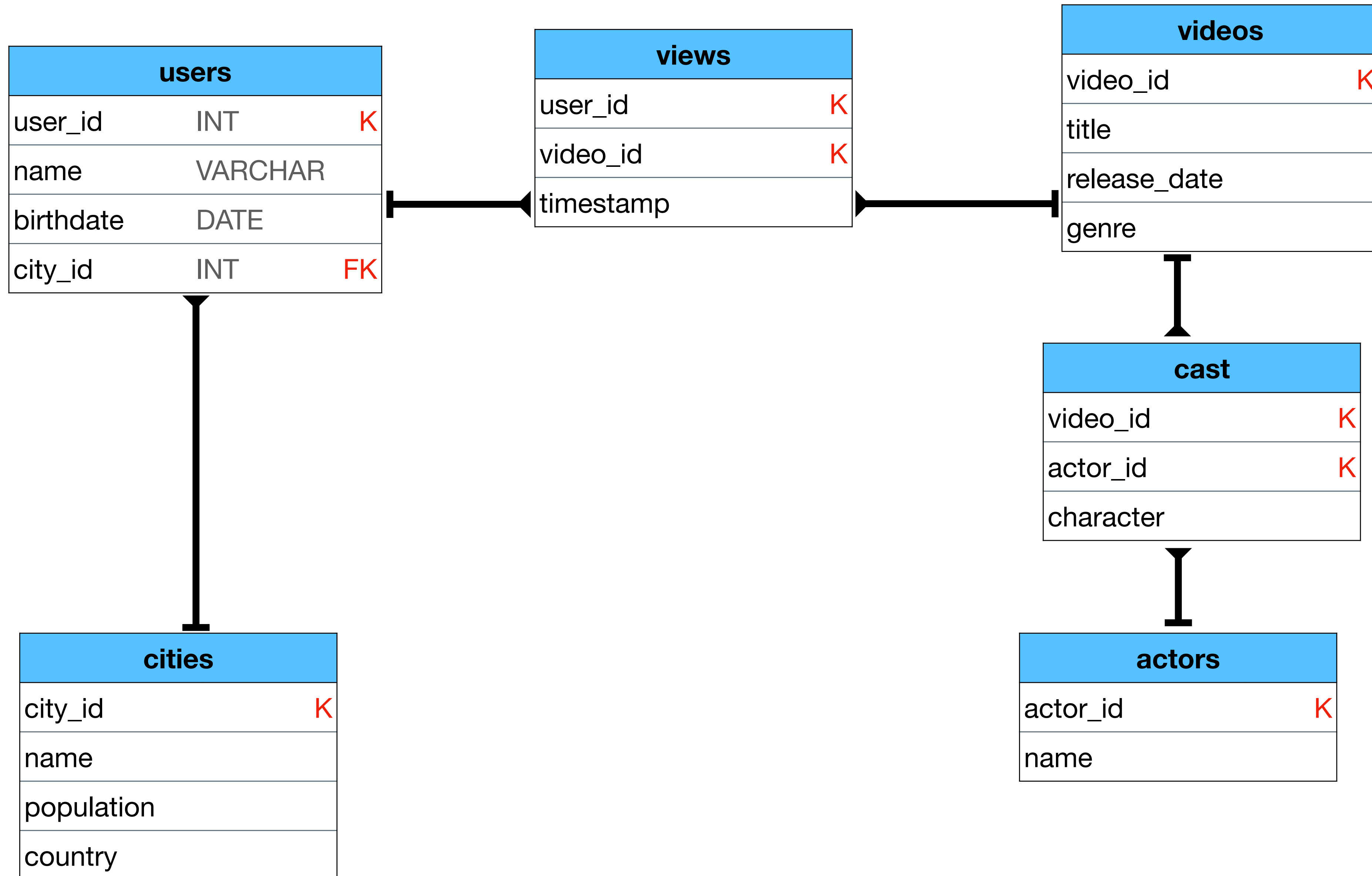


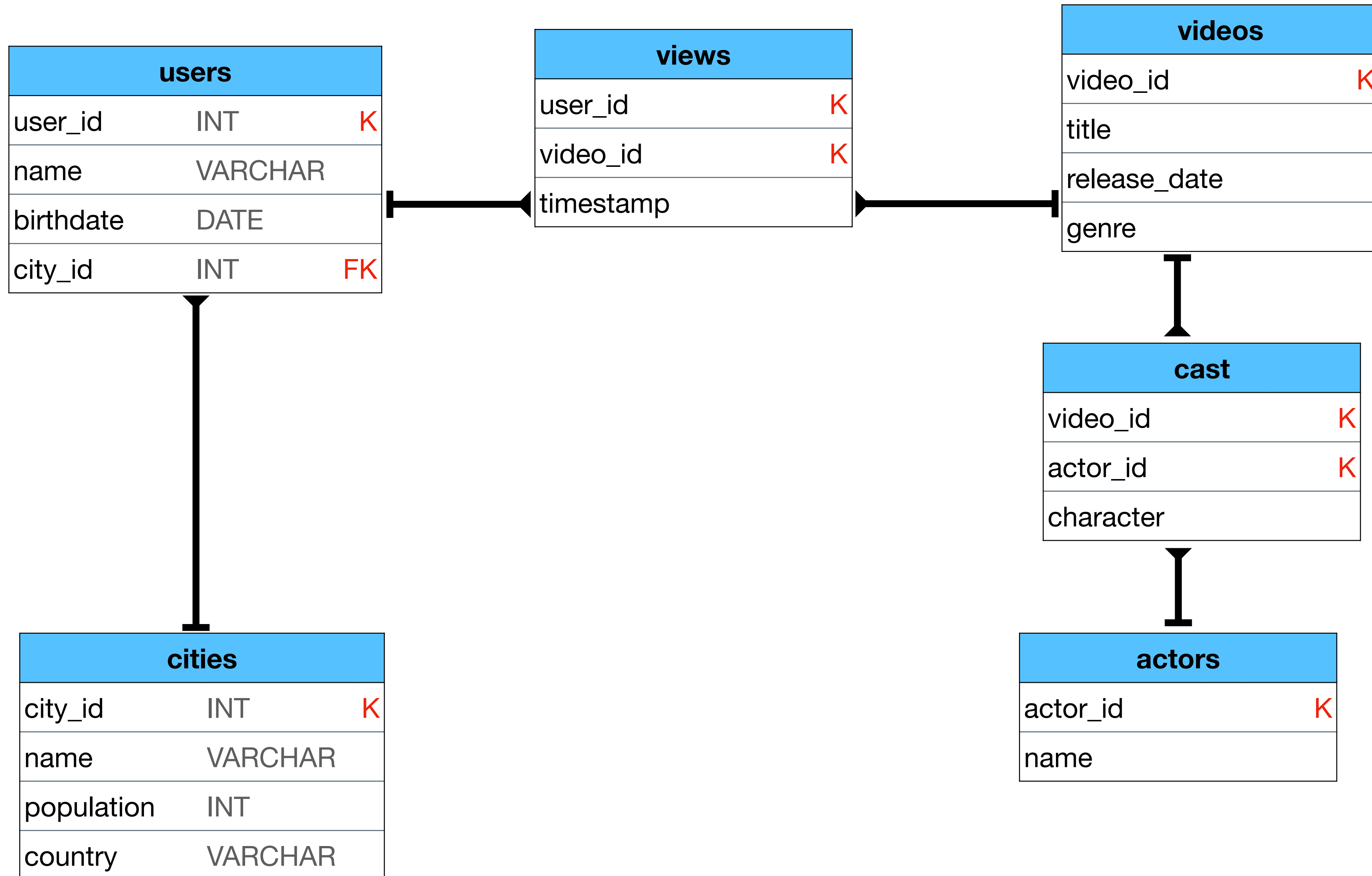
# Physical data model

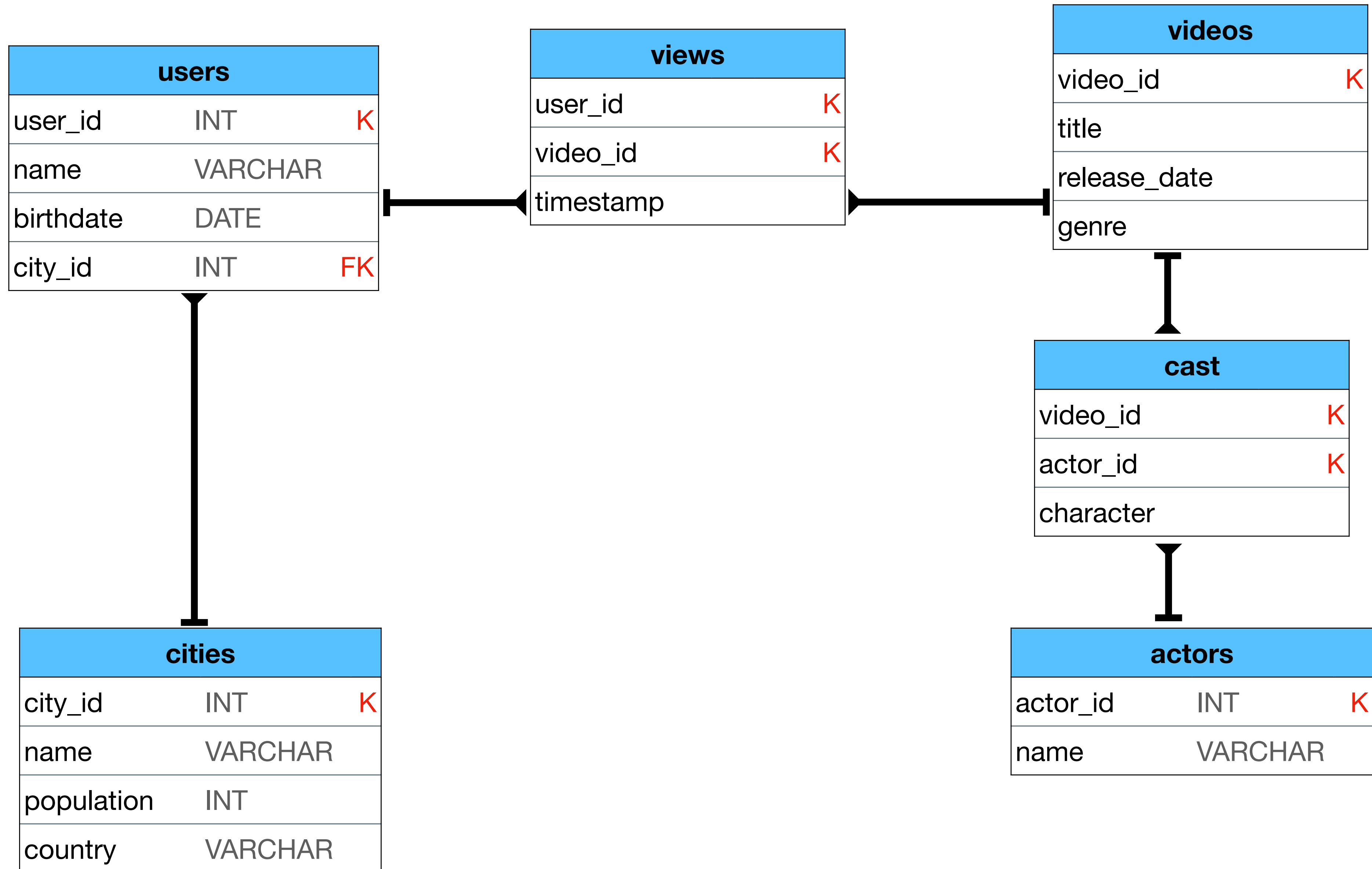
- Finalize the schema
- Add types
- Generate create table statements

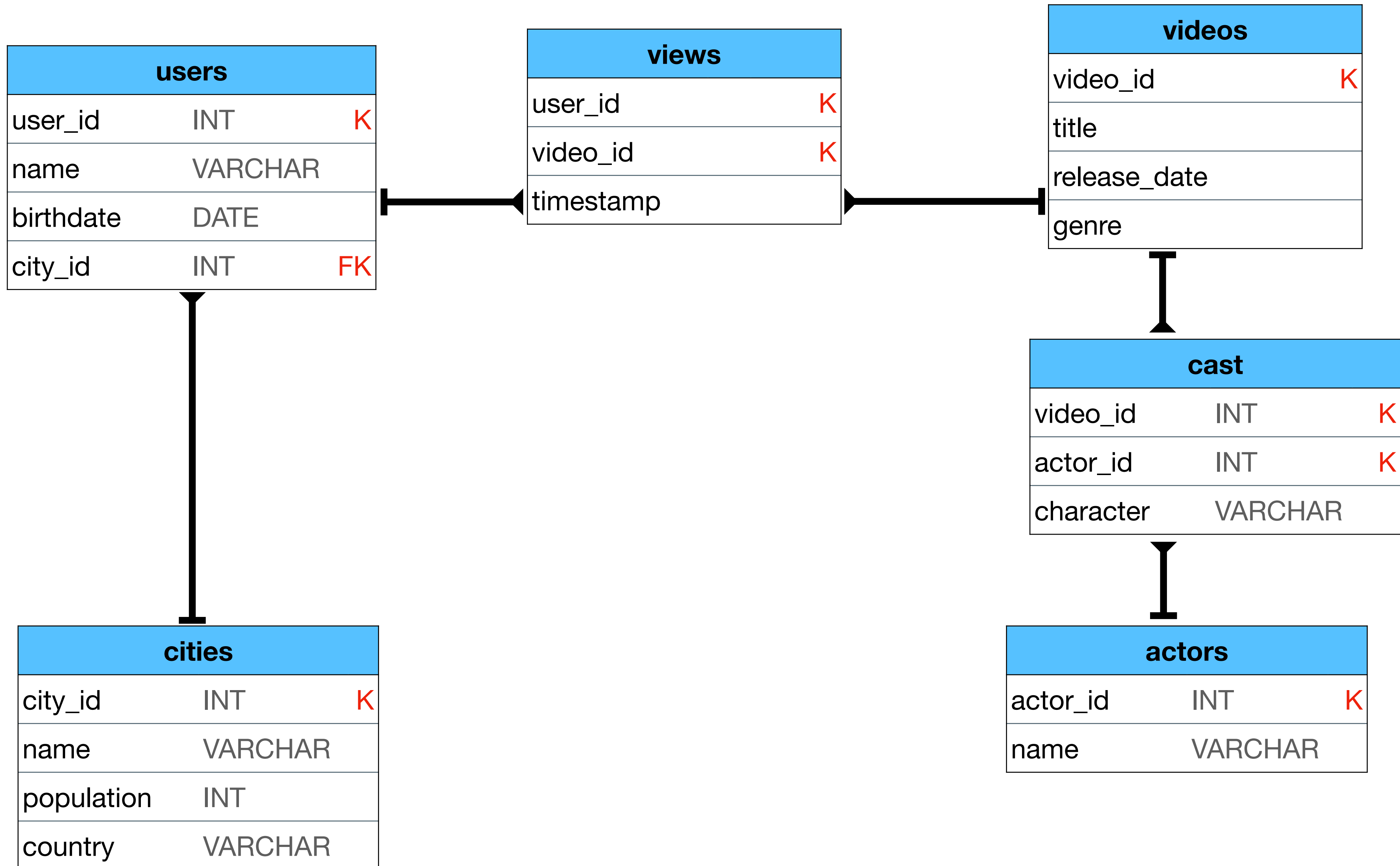
# Example

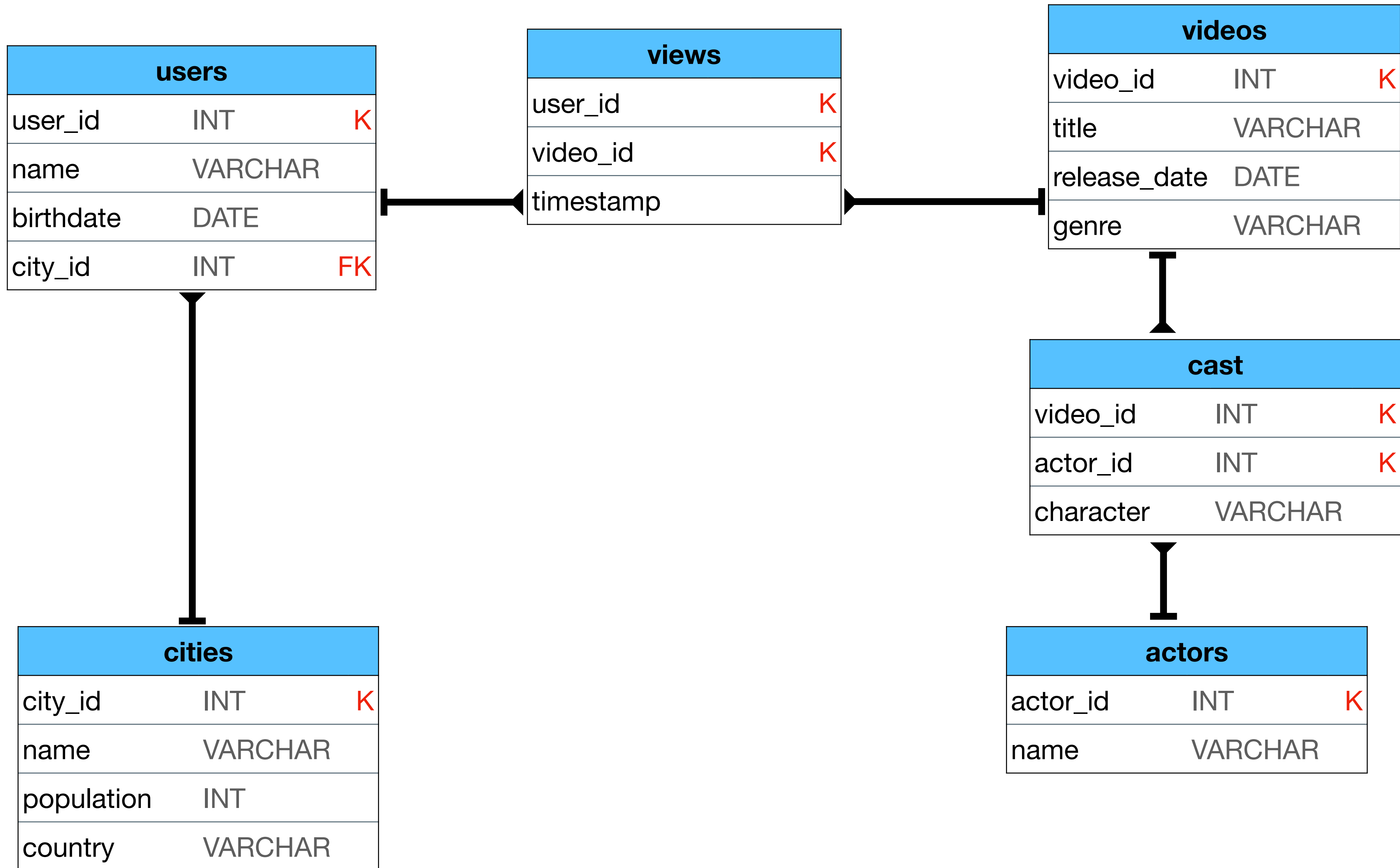




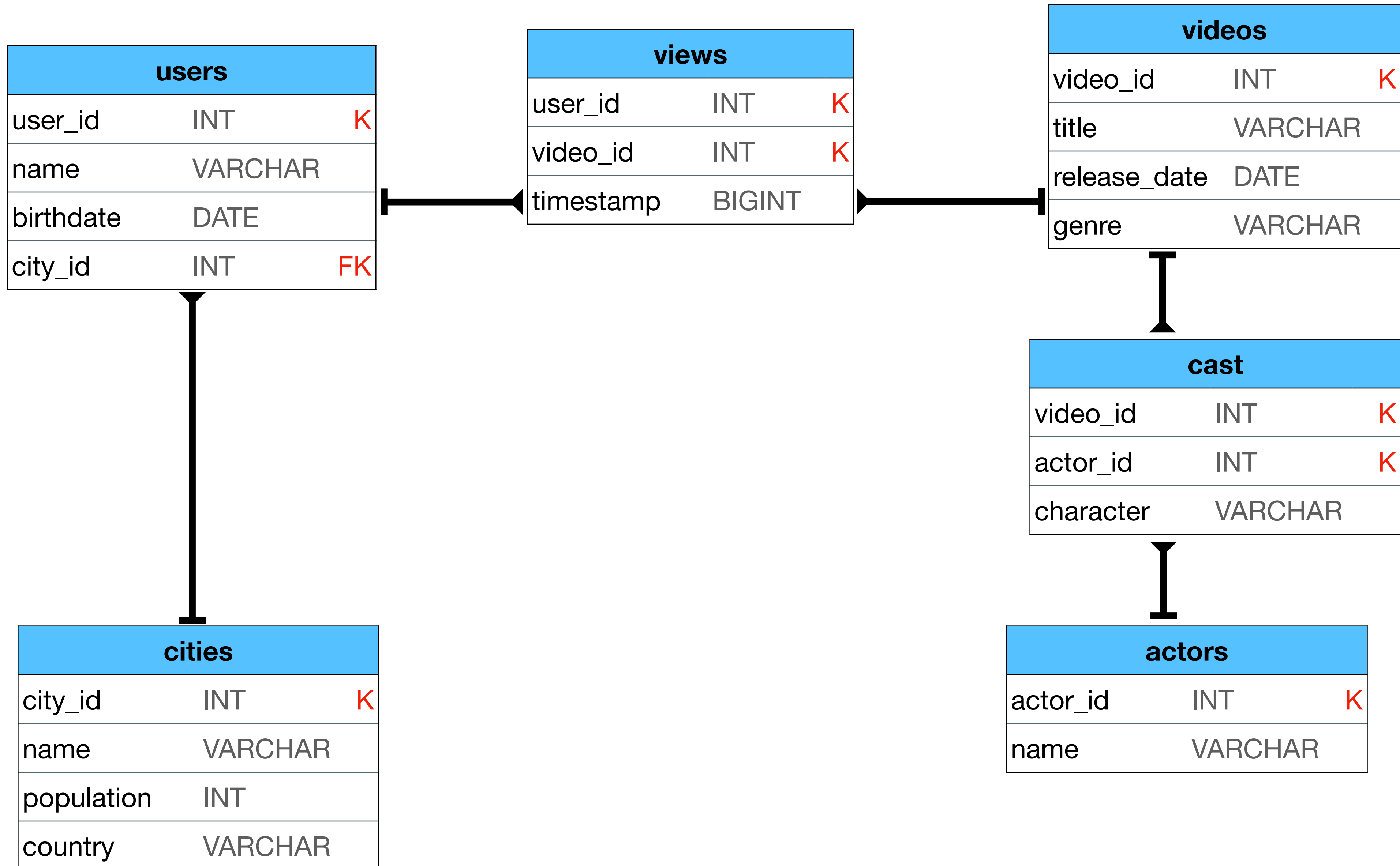


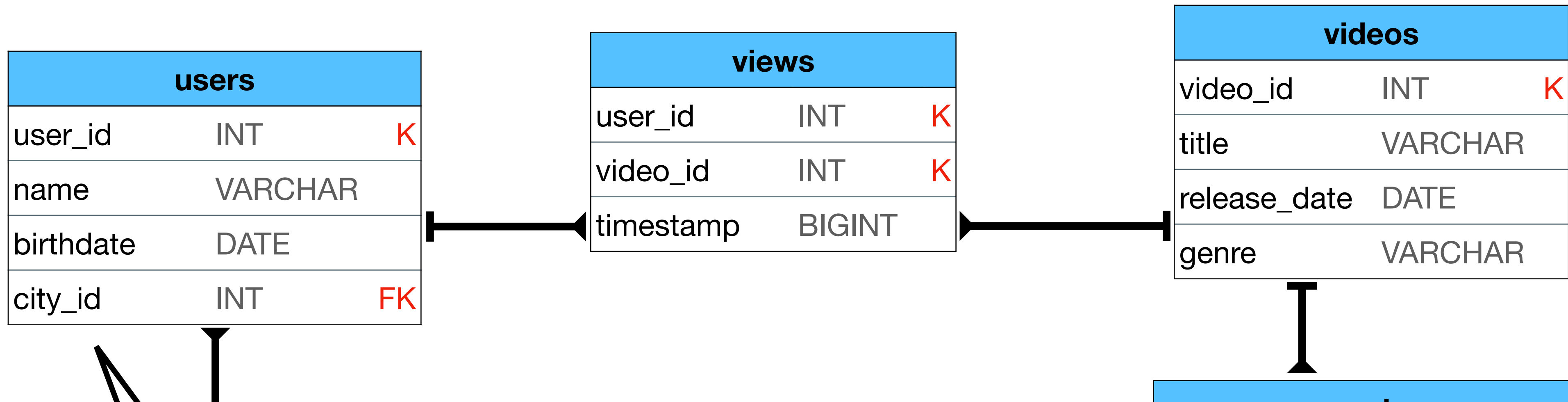








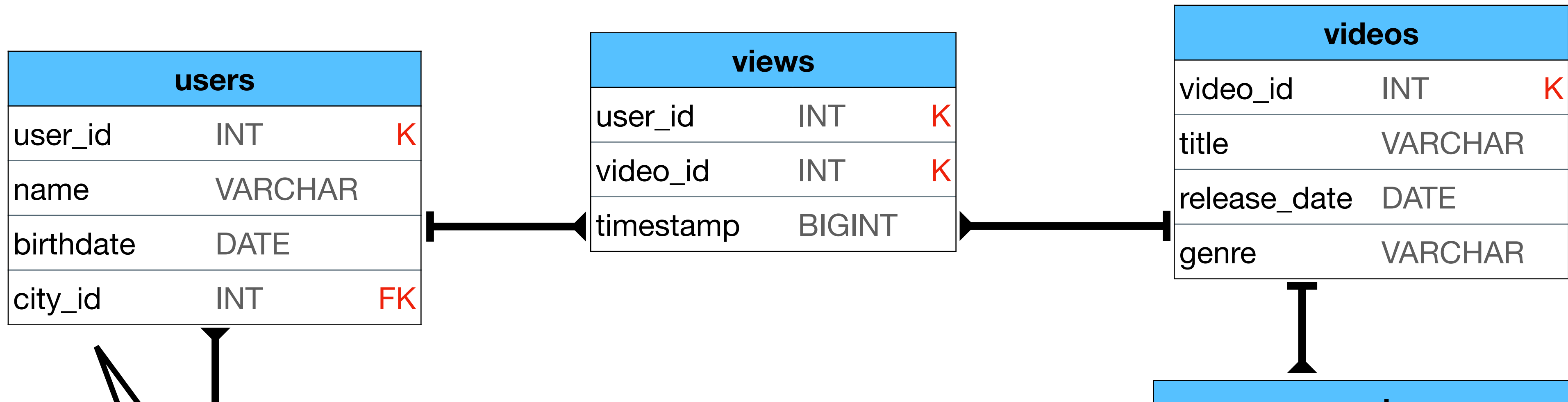




```

CREATE TABLE users (
  user_id INT NOT NULL,
  name VARCHAR(255),
  birthdate DATE,
  city_id INT,
  PRIMARY KEY (user_id),
  FOREIGN KEY (city_id)
  REFERENCES cities(id) ON DELETE REJECT
)
  
```

city_id	
name	
populat	
country	VARCHAR

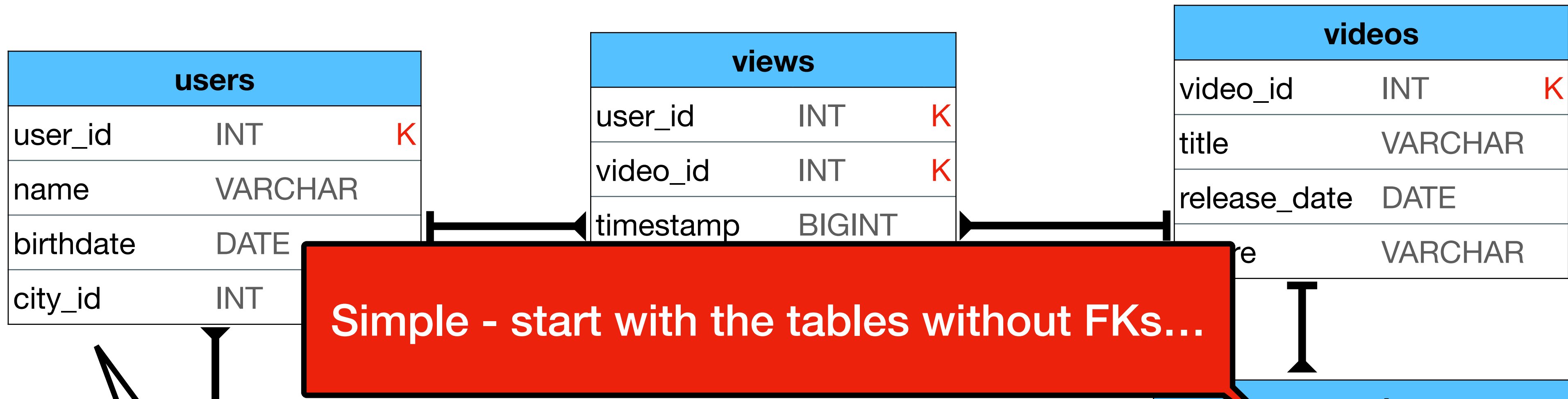


```

CREATE TABLE users (
  user_id INT NOT NULL,
  name VARCHAR(255),
  birthdate DATE,
  city_id INT,
  PRIMARY KEY (user_id),
  FOREIGN KEY (city_id)
  REFERENCES cities(id) ON DELETE REJECT
)
  
```

**BUT cities does not yet exists...  
What do you do?**

city_id	
name	
populat	
country	VARCHAR



```
CREATE TABLE users (
  user_id INT NOT NULL,
  name VARCHAR(255),
  birthdate DATE,
  city_id INT,
  PRIMARY KEY (user_id),
  FOREIGN KEY (city_id)
  REFERENCES cities(id) ON DELETE REJECT
)
```

BUT cities does not yet exists...  
What do you do?

city_id	
name	
populat	
country	VARCHAR

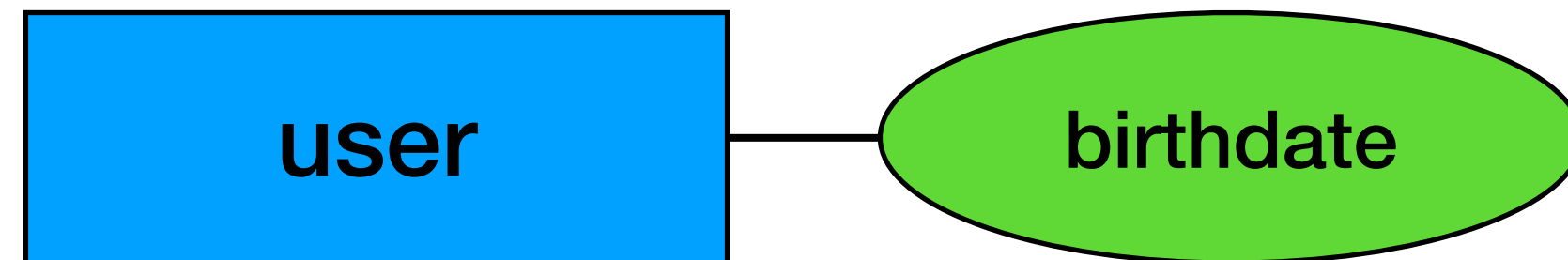
# Design examples

# Example (1)

- What is the problem here? Solution?

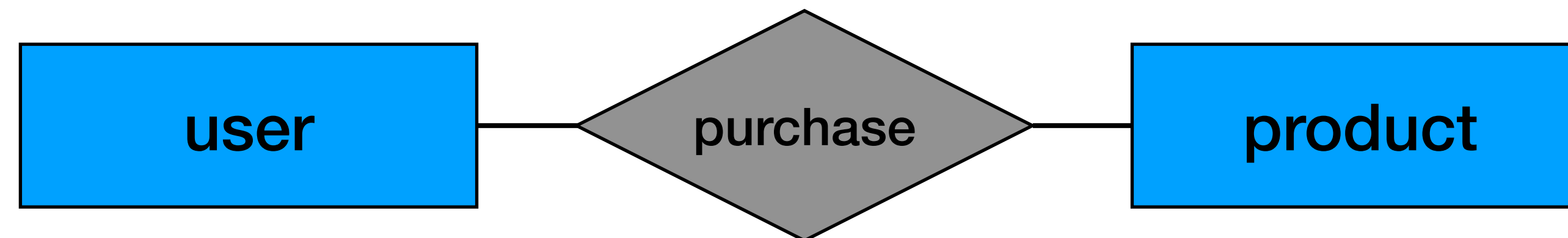


# Example (1)



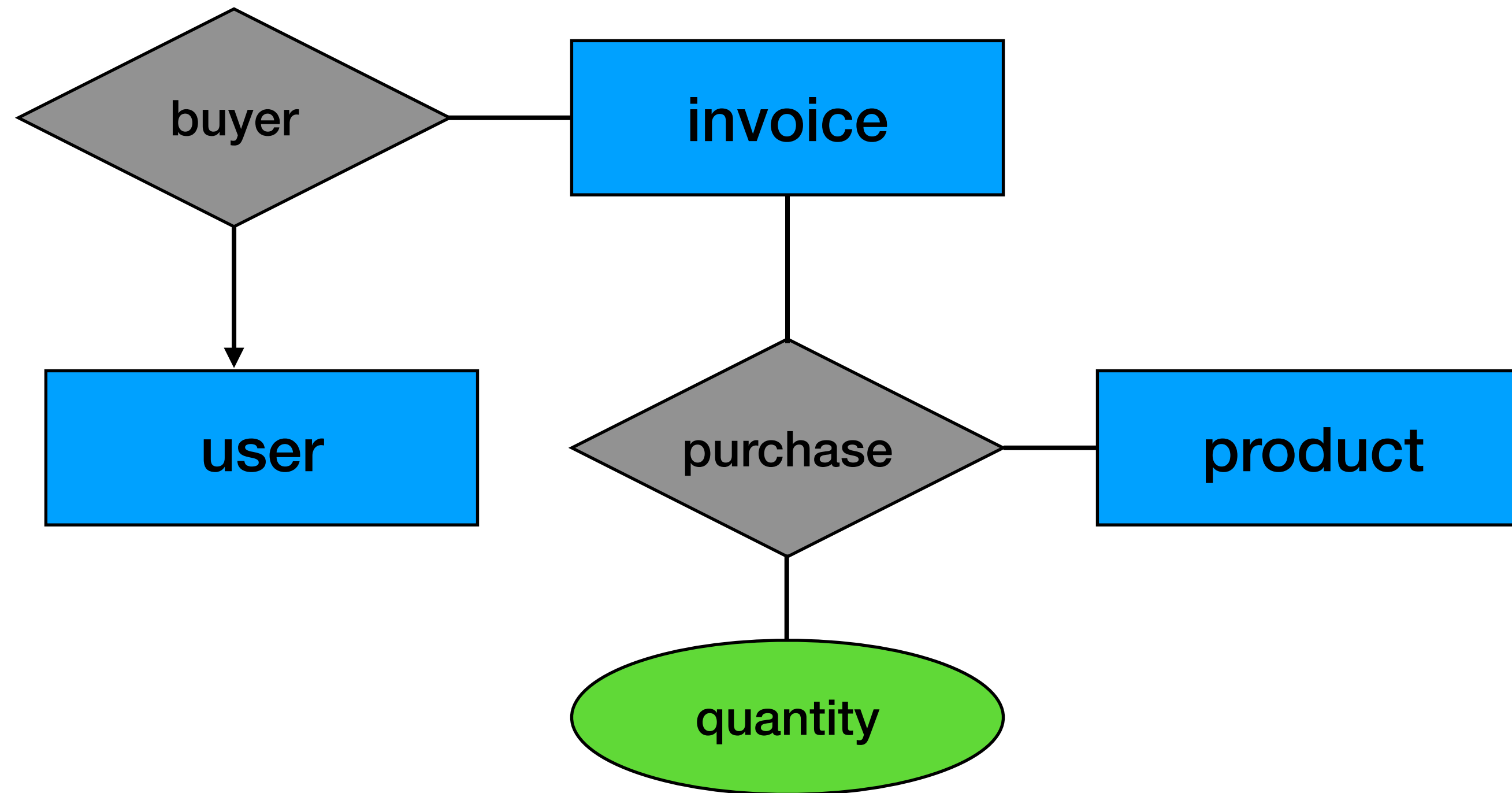
# Example (2)

- What is the problem here? Solution?



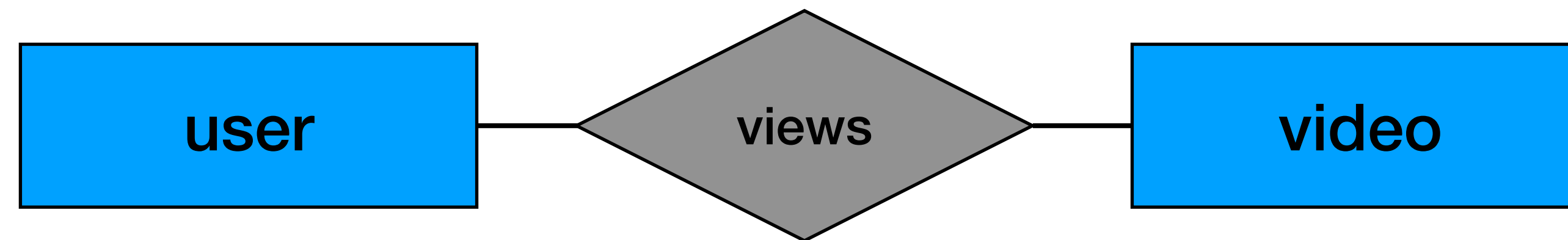


# Example (2)



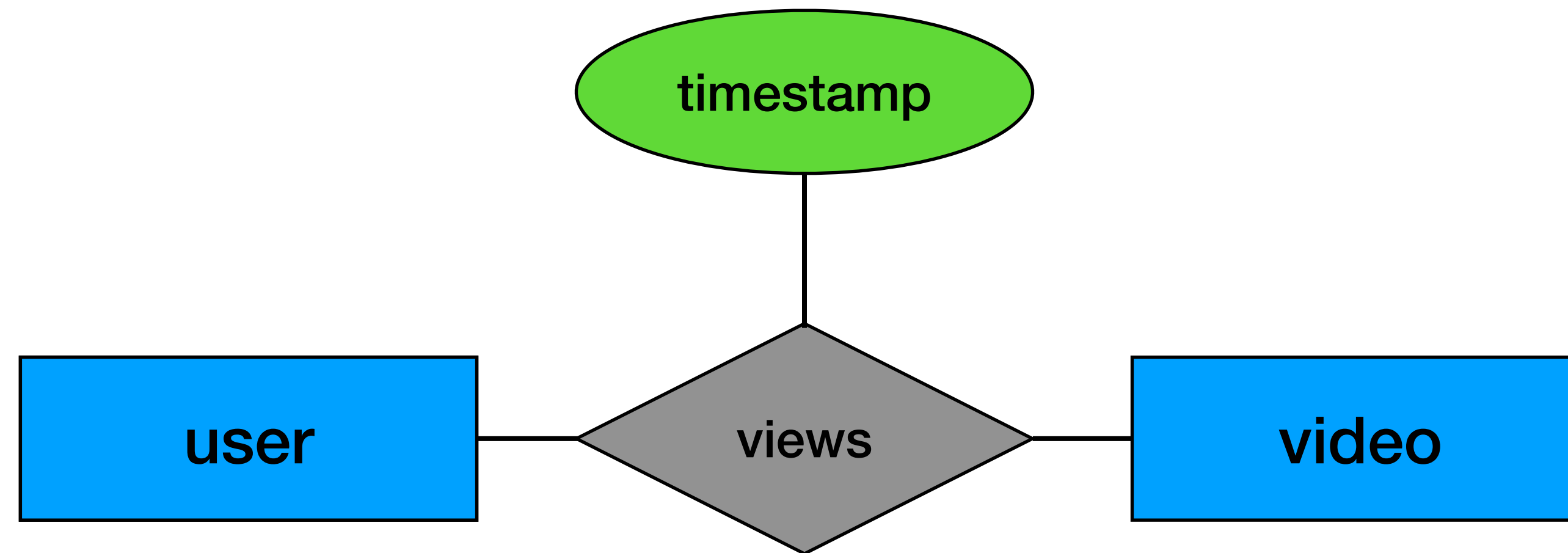
# Example (3)

- What is the problem here? Solution?



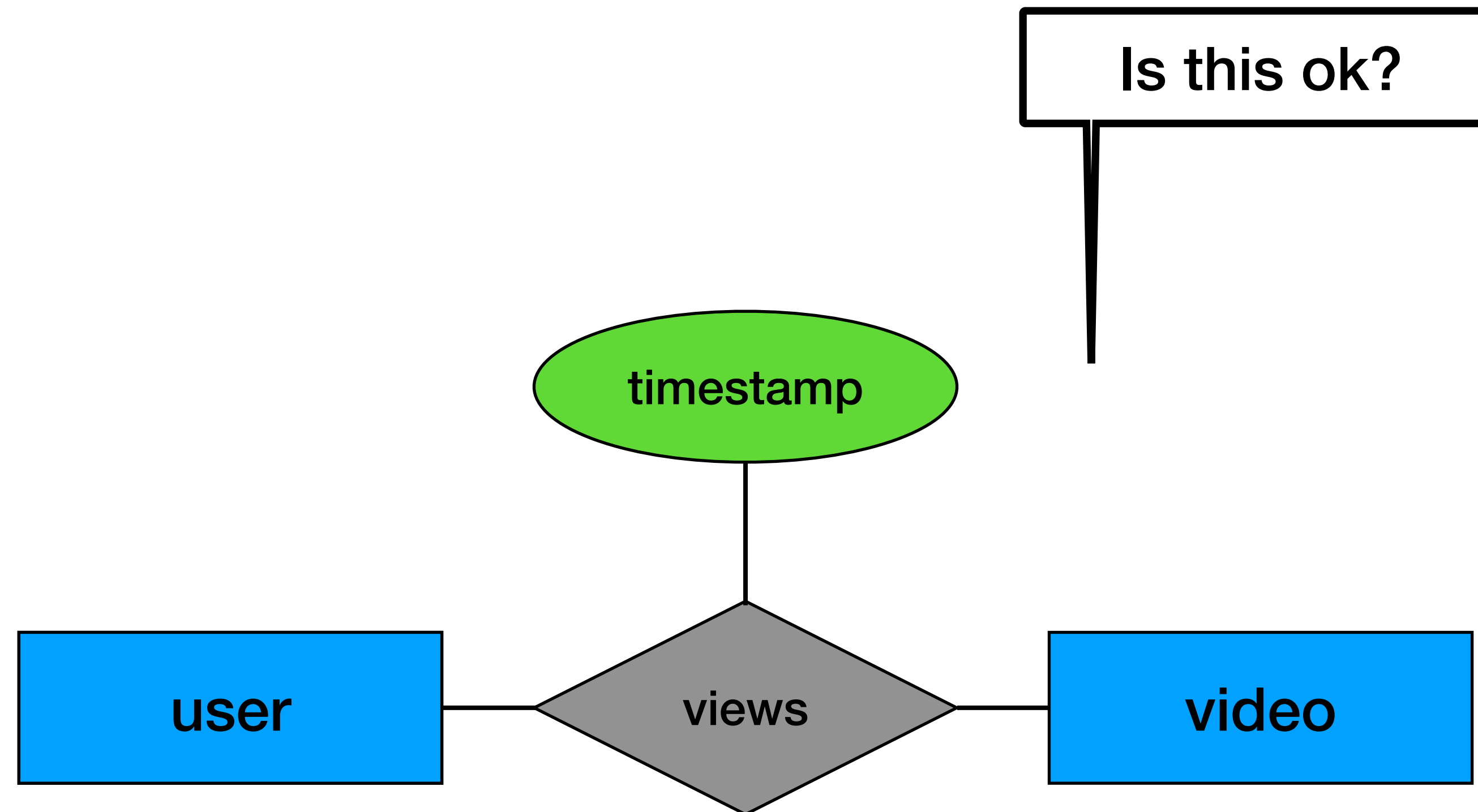
# Example (3)

- Option 1



# Example (3)

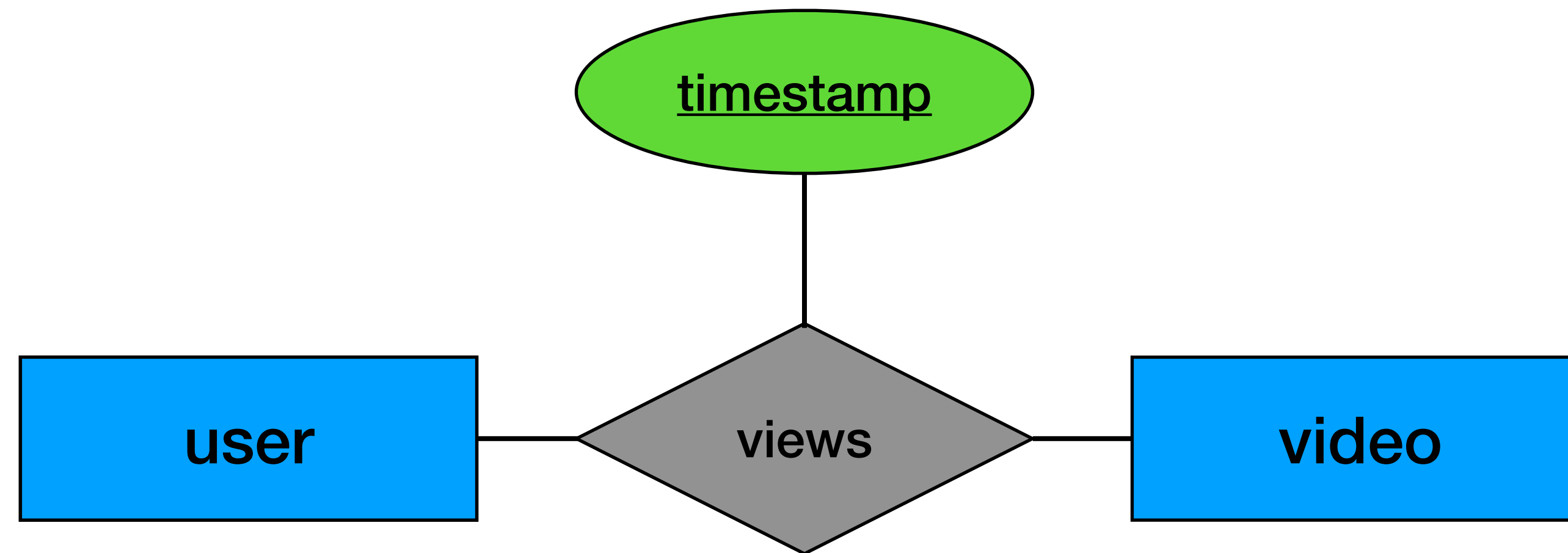
- Option 1



views_option_1		
user_id	INT	K
video_id	INT	K
timestamp	BIGINT	

# Example (3)

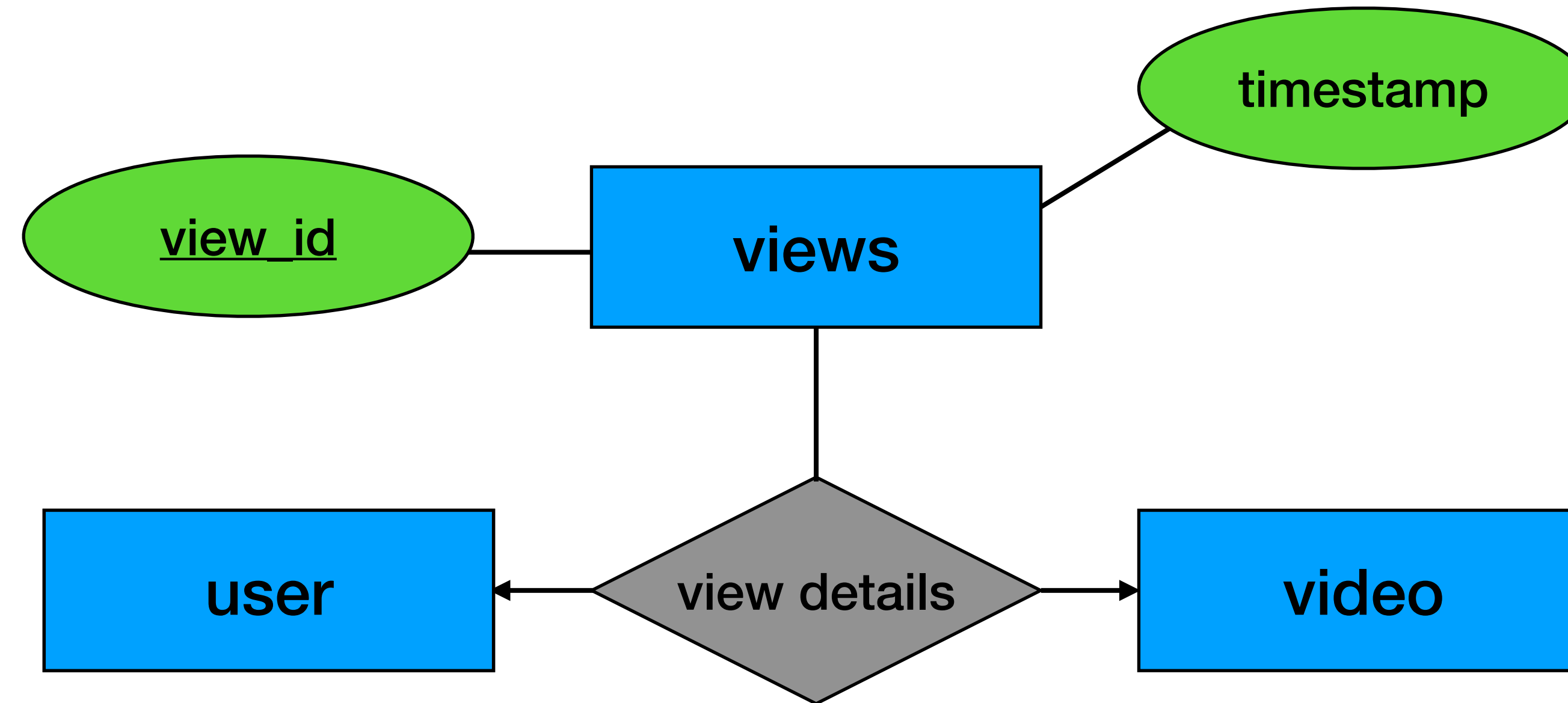
- Option 1



views_option_1		
user_id	INT	K
video_id	INT	K
timestamp	BIGINT	K

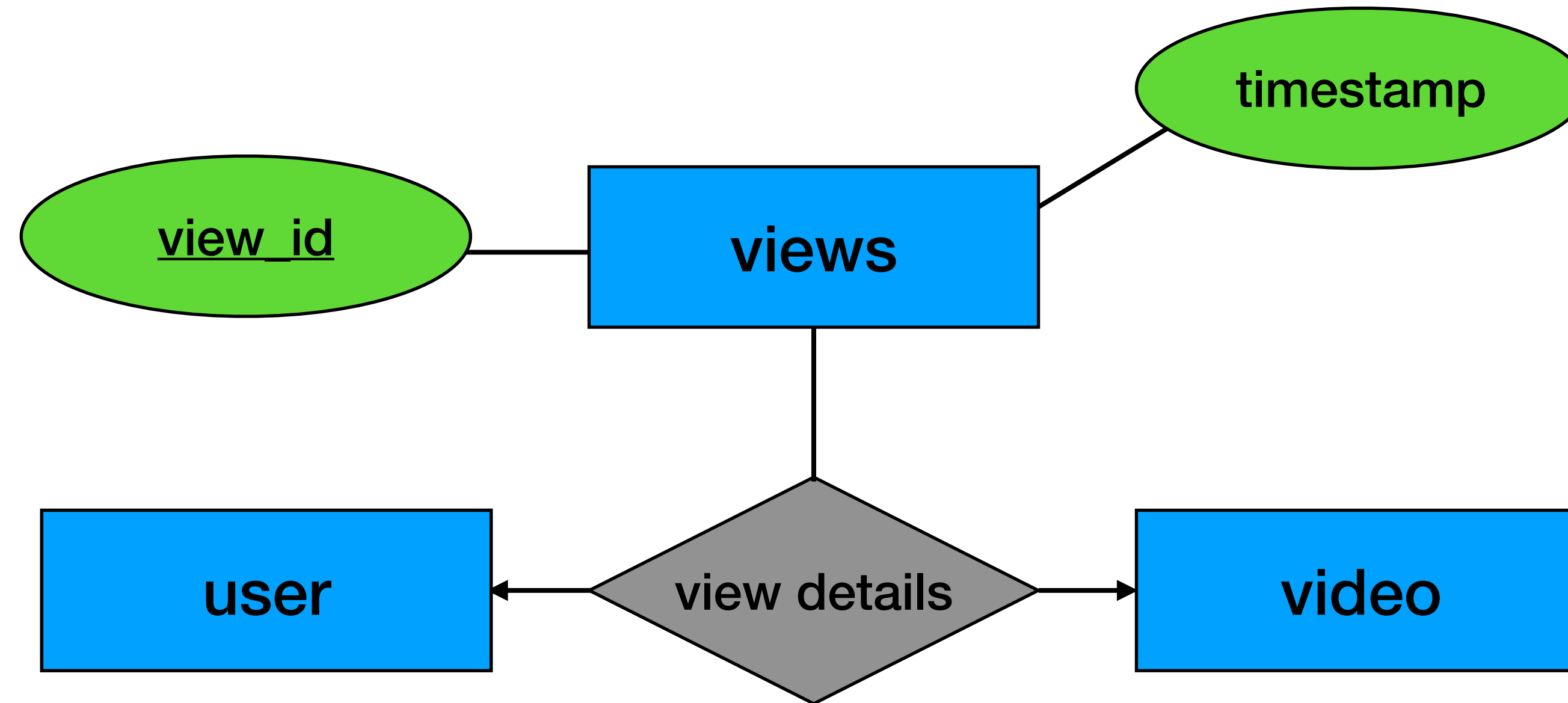
# Example (3)

- Option 2



# Example (3)

- Option 2



views_option_2		
view_id	INT	K
user_id	INT	FK
video_id	INT	FK
timestamp	BIGINT	

# Example (3)

- Option 1 vs Option 2

views_option_1		
user_id	INT	K
video_id	INT	K
timestamp	BIGINT	K

Classic relational modeling -  
"By the book"

views_option_2		
view_id	INT	K
user_id	INT	FK
video_id	INT	FK
timestamp	BIGINT	

"NoSQL style" -  
Can improve performance on  
large scale



# Example (3)

- Option 1 vs Option 2

views_option_1		
user_id	INT	K
video_id	INT	K
timestamp	BIGINT	K

views_option_2		
view_id	INT	K
user_id	INT	FK
video_id	INT	FK
timestamp	BIGINT	

Modeling is an art...  
There is not always a clear right / wrong answer

Classic relational modeling -  
"By the book"

"NoSQL style" -  
Can improve performance on  
large scale