Data Representation and Relational Algebra

January 29, 2019
Data Science CSCI 1951A
Brown University
Instructor: Ellie Pavlick
HTAs: Wennie Zhang, Maulik Dang, Gurnaaz Kaur

But first!

- Waitlist
- We are working our way through.
- We've sent out ~90 overrides so far, and will send out another batch soon
- If you don't get one by EOD today, then the chance of getting one is probably "low"
- If you are registered any way other than enthusiastically, be a mensch and please drop :)

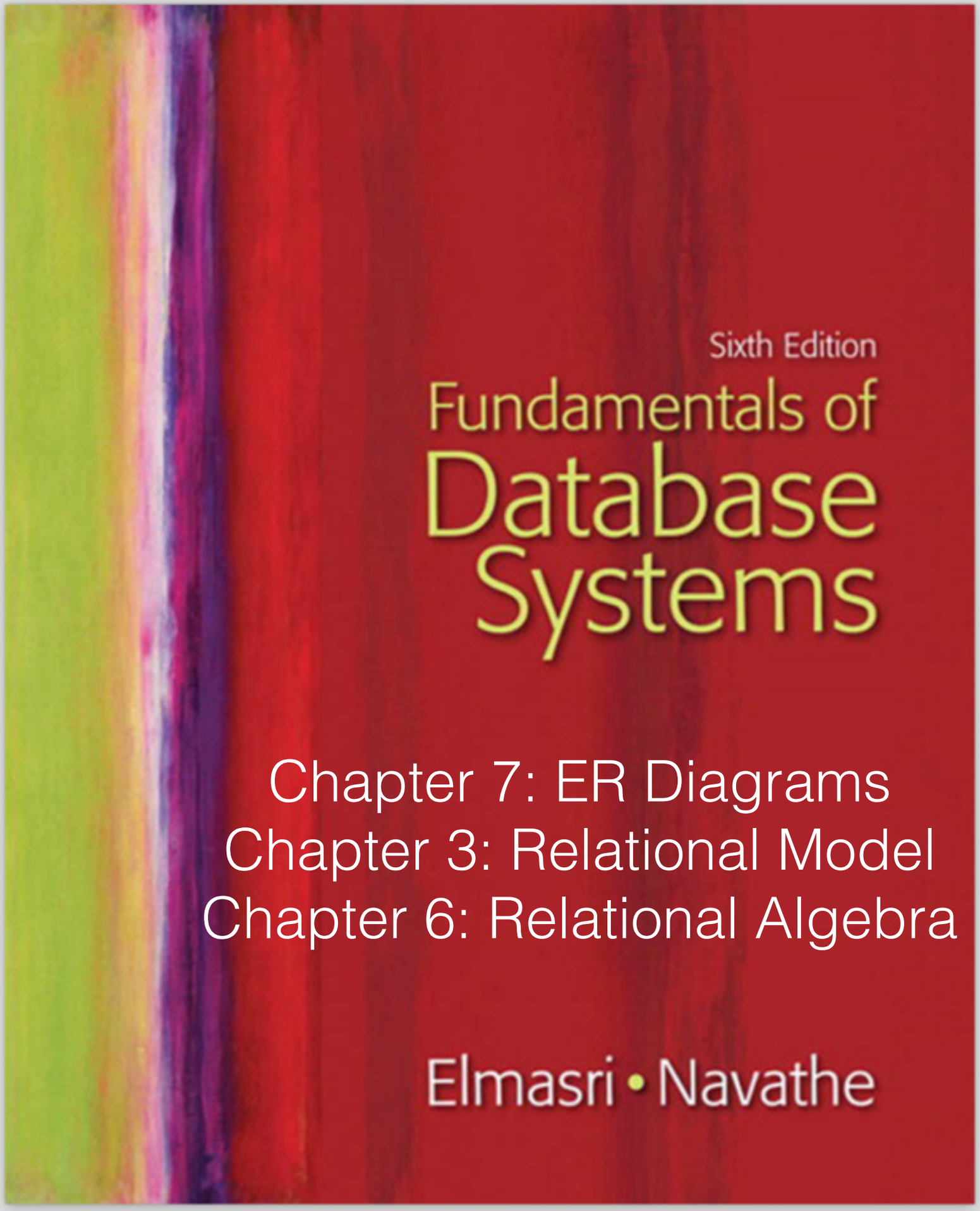
But first!

- There will be no iClicker requirement until after shopping period/empty waitlist
  - iClicker forgiveness for those of you who stayed on Thursday 😇🙏
- You have time to procrastinate on Assignment 1 until end of shopping period
- Sign the collaboration policy!

But first!

Burning Questions?

Data Representation and Relational Algebra


Sixth Edition
Fundamentals of
Database
Systems
Chapter 7: ER Diagrams
Chapter 3: Relational Model
Chapter 6: Relational Algebra
Elmasri • Navathe

DATABASES FOR DATA SCIENTIST

Requirement Engineering	Conceptual Modeling	Logical and Physical Modeling	Asking and Answering Questions (Analysis)
“Book of Duty”	Conceptual Design (ER)	Logical Design (schema, table names, data types), Physical Design (indices, memory layout, optimizations)	Relational Algebra, SQL

https://cs.brown.edu/courses/csci1951-a/res/static/lec02_databases.pdf

DATABASES FOR DATA SCIENTIST

Requirement Engineering	Conceptual Modeling	Logical and Physical Modeling	Asking and Answering Questions (Analysis)
“Book of Duty”	Conceptual Design (ER)	Logical Design (schema, table names, data types), Physical Design (indices, memory layout, optimizations)	Relational Algebra, SQL

https://cs.brown.edu/courses/csci1951-a/res/static/lec02_databases.pdf

“Book of Duty”/“Miniworld”

- Informal description of data domain
- In natural language:
    - What are the objects you care about?
    - What properties/attributes of those objects are you measuring?
    - What are the relationships between them?
    - What assumptions are we making? (E.g. sizes, cardinalities)
    - What is the workload on the database (Read-only? Read/write?)
    - Permissions and privacy concerns?

The most distinctive jargon in U.S. job listings

The most distinctive jargon in U.S. job listings

Description of “Miniworld”

- Project: We want to analyze political trends surrounding 2020 primary candidates
- Plan: Crawl Twitter for posts from or about 2020 primary candidates. We want to analyze the spread of opinions, through through following/follower relationships, share/retweet chains, and language use

Description of “Miniworld”

- Objects used:
- Domains of attributes of objects:
- Identifiers, references / relationships:
- Cardinalities:
- Distributions:
- Workload:
- Priorities and service level agreements:

Description of “Miniworld”

- Objects used: People, Tweets, Candidates
- Domains of attributes of objects:
- Identifiers, references / relationships:
- Cardinalities:
- Distributions:
- Workload:
- Priorities and service level agreements:

Description of “Miniworld”

- Objects used: People, Tweets, Candidates
- Domains of attributes of objects: Tweets have timestamps (date), authors (Person), text (max 140 characters), attachments, hashtags...
- Identifiers, references / relationships:
- Cardinalities:
- Distributions:
- Workload:
- Priorities and service level agreements:

Description of “Miniworld”

- Objects used: People, Tweets, Candidates
- Domains of attributes of objects: Tweets have timestamps (date), authors (Person), text (max 140 characters), attachments, hashtags...
- Identifiers, references / relationships: People have unique IDs, People author Tweets, People retweet Tweets, People like Tweets, People follow People, Tweets mention People...
- Cardinalities:
- Distributions:
- Workload:
- Priorities and service level agreements:

Description of “Miniworld”

- Objects used: People, Tweets, Candidates
- Domains of attributes of objects: Tweets have timestamps (date), authors (Person), text (max 140 characters), attachments, hashtags...
- Identifiers, references / relationships: People have unique IDs, People author Tweets, People retweet Tweets, People like Tweets, People follow People, Tweets mention People...
- Cardinalities: Every tweet has exactly one author, one author can have many tweets, a tweet can mention 0 to many candidates, a candidate can be mentioned in 0 to many tweets
- Distributions:
- Workload:
- Priorities and service level agreements:

Description of “Miniworld”

- Objects used: People, Tweets, Candidates
- Domains of attributes of objects: Tweets have timestamps (date), authors (Person), text (max 140 characters), attachments, hashtags...
- Identifiers, references / relationships: People have unique IDs, People author Tweets, People retweet Tweets, People like Tweets, People follow People, Tweets mention People...
- Cardinalities: Every tweet has exactly one author, one author can have many tweets, a tweet can mention 0 to many candidates, a candidate can be mentioned in 0 to many tweets
- Distributions: Some people tweet a lot, others just follow; some candidates are mentioned a lot; follower/followee asymmetries
- Workload:
- Priorities and service level agreements:

Description of “Miniworld”

- Objects used: People, Tweets, Candidates
- Domains of attributes of objects: Tweets have timestamps (date), authors (Person), text (max 140 characters), attachments, hashtags...
- Identifiers, references / relationships: People have unique IDs, People author Tweets, People retweet Tweets, People like Tweets, People follow People, Tweets mention People...
- Cardinalities: Every tweet has exactly one author, one author can have many tweets, a tweet can mention 0 to many candidates, a candidate can be mentioned in 0 to many tweets
- Distributions: Some people tweet a lot, others just follow; some candidates are mentioned a lot; follower/followee asymmetries
- Workload: scrape and populate ones, read often
- Priorities and service level agreements:

Description of “Miniworld”

- Objects used: People, Tweets, Candidates
- Domains of attributes of objects: Tweets have timestamps (date), authors (Person), text (max 140 characters), attachments, hashtags...
- Identifiers, references / relationships: People have unique IDs, People author Tweets, People retweet Tweets, People like Tweets, People follow People, Tweets mention People...
- Cardinalities: Every tweet has exactly one author, one author can have many tweets, a tweet can mention 0 to many candidates, a candidate can be mentioned in 0 to many tweets
- Distributions: Some people tweet a lot, others just follow; some candidates are mentioned a lot; follower/followee asymmetries
- Workload: scrape and populate ones, read often
- Priorities and service level agreements: “right to be forgotten” rules...

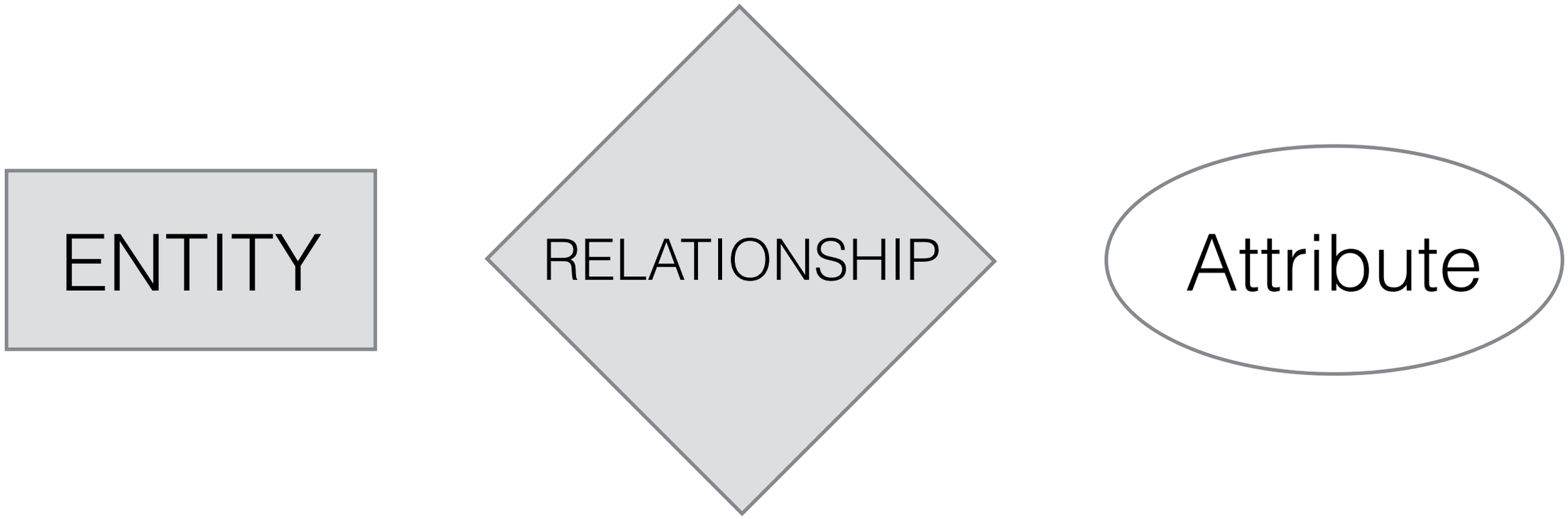
DATABASES FOR DATA SCIENTIST

Requirement Engineering	Conceptual Modeling	Logical and Physical Modeling	Asking and Answering Questions (Analysis)
“Book of Duty”	Conceptual Design (ER)	Logical Design (schema, table names, data types), Physical Design (indices, memory layout, optimizations)	Relational Algebra, SQL

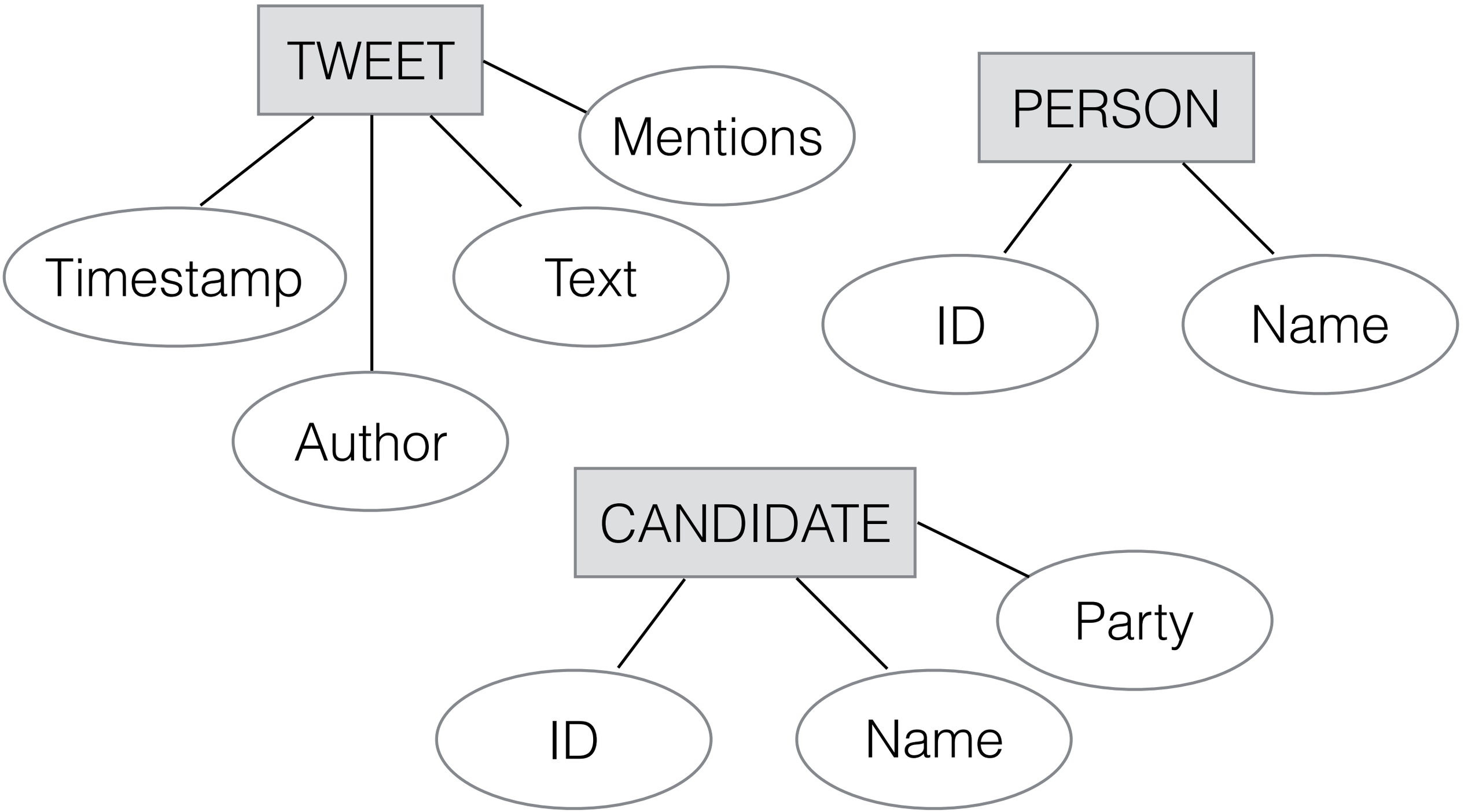
https://cs.brown.edu/courses/csci1951-a/res/static/lec02_databases.pdf

Entity-Relationship (ER) Model

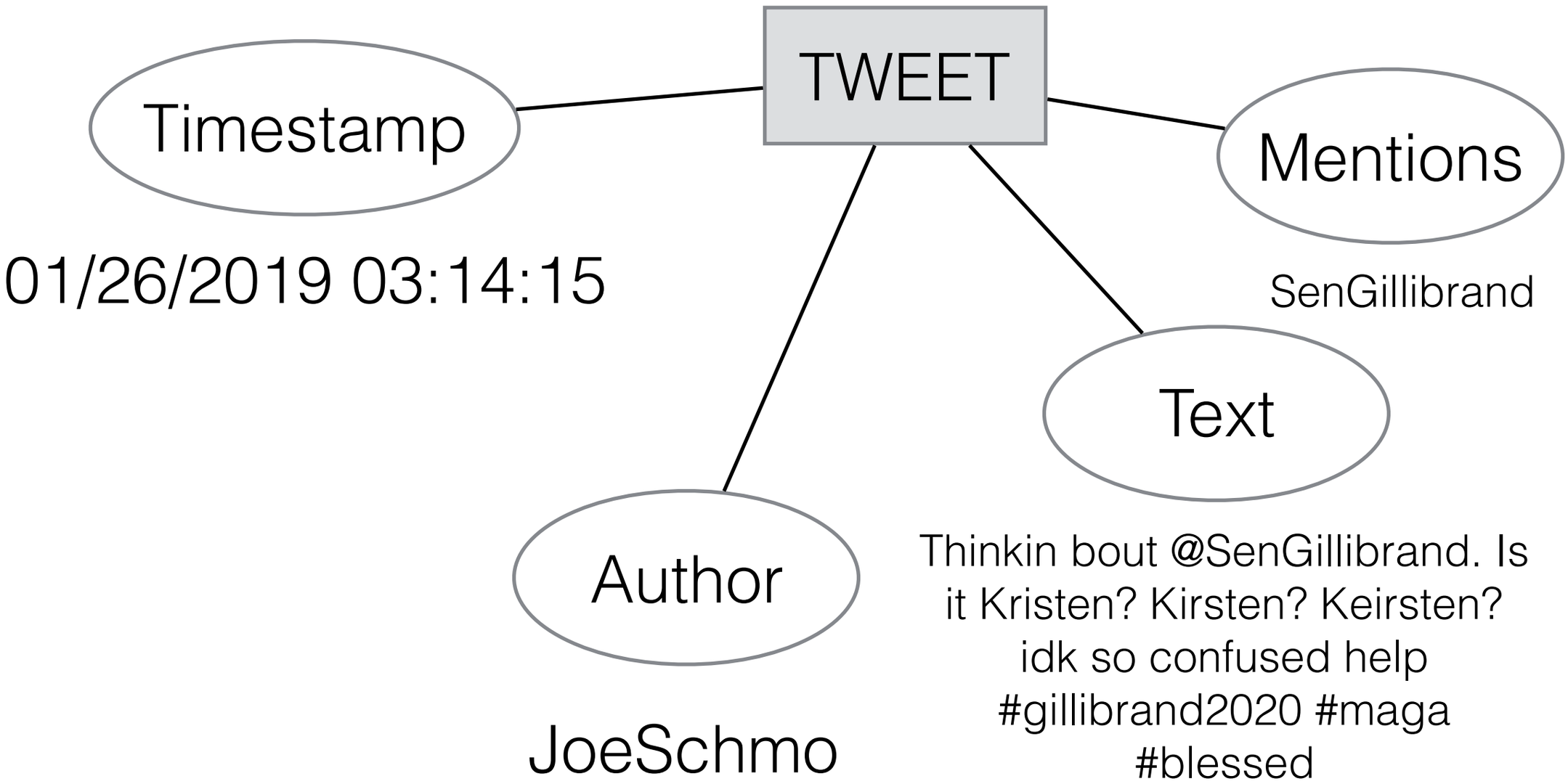
Entity-Relationship (ER) Model



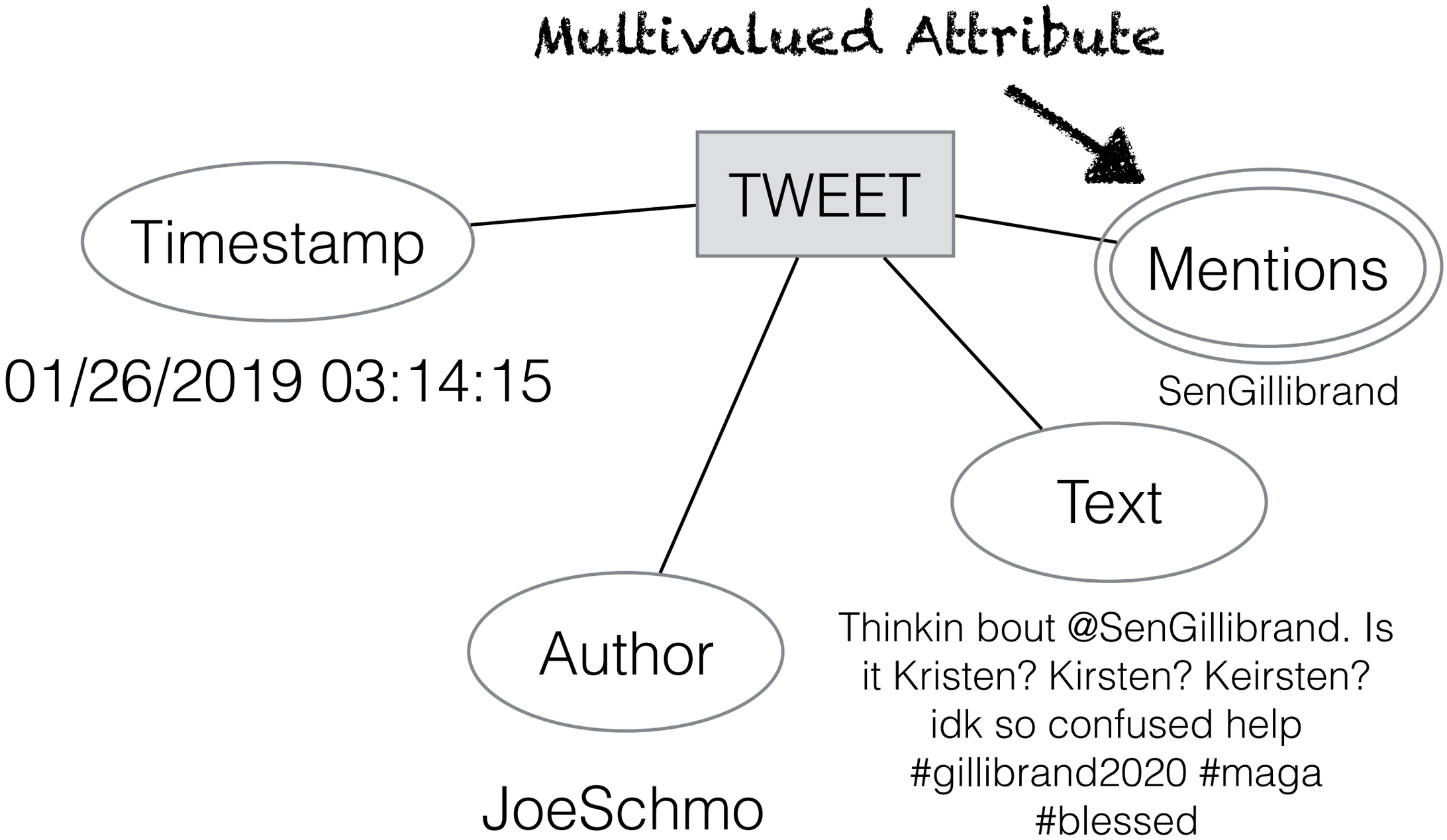
Entity-Relationship (ER) Model



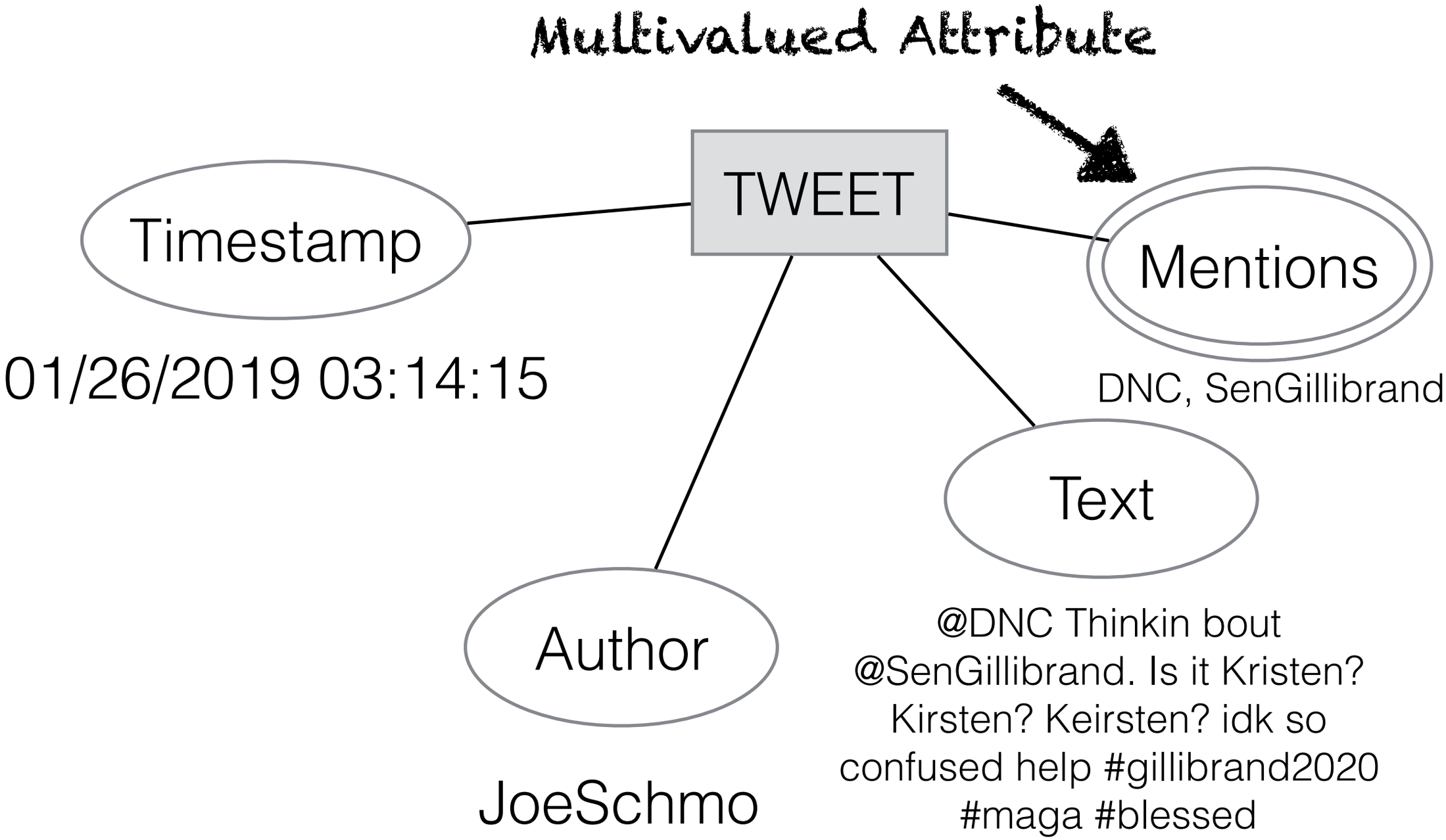
Entity-Relationship (ER) Model



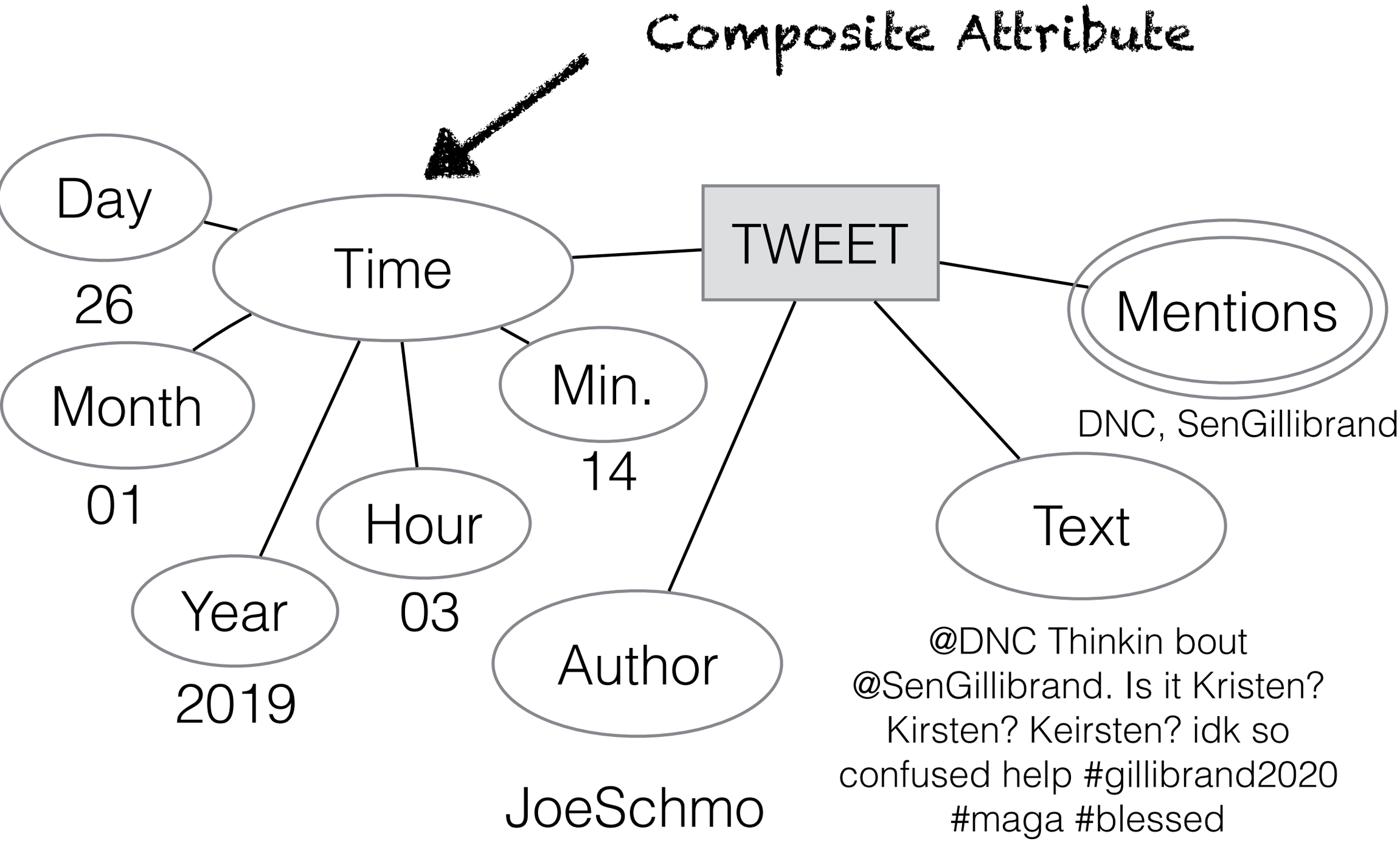
Entity-Relationship (ER) Model



Entity-Relationship (ER) Model

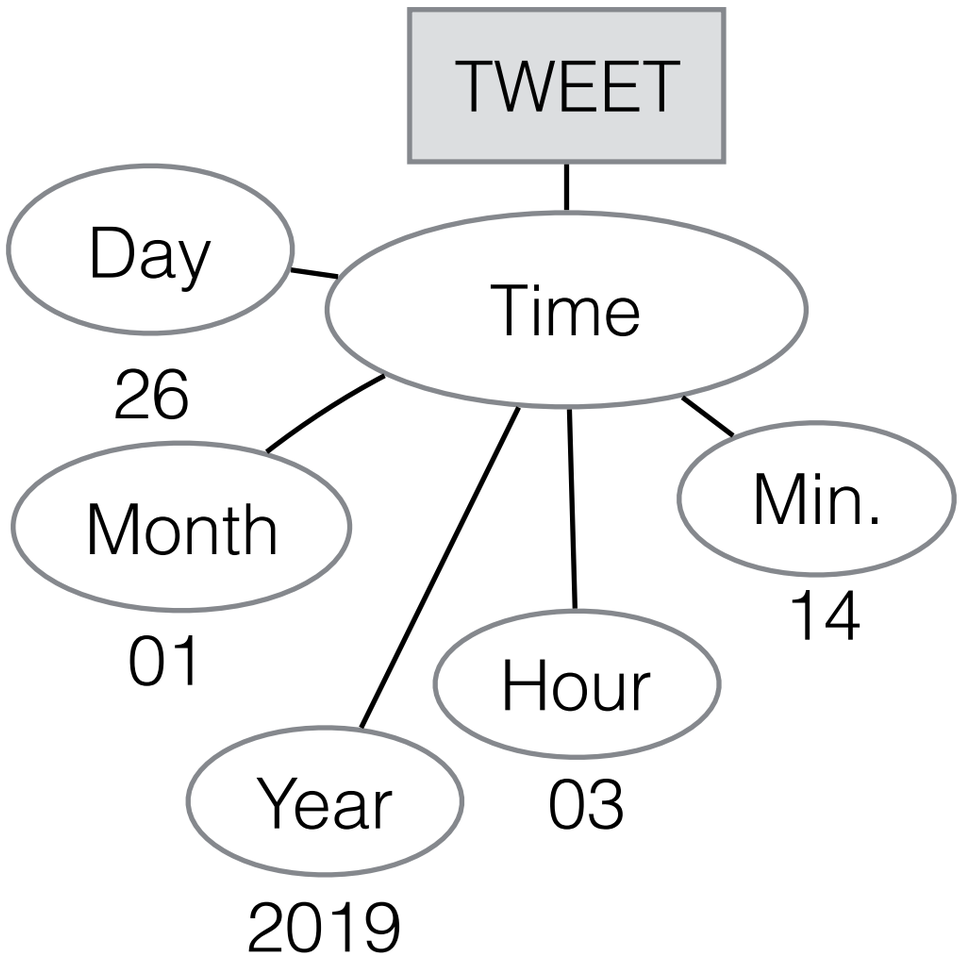


Entity-Relationship (ER) Model

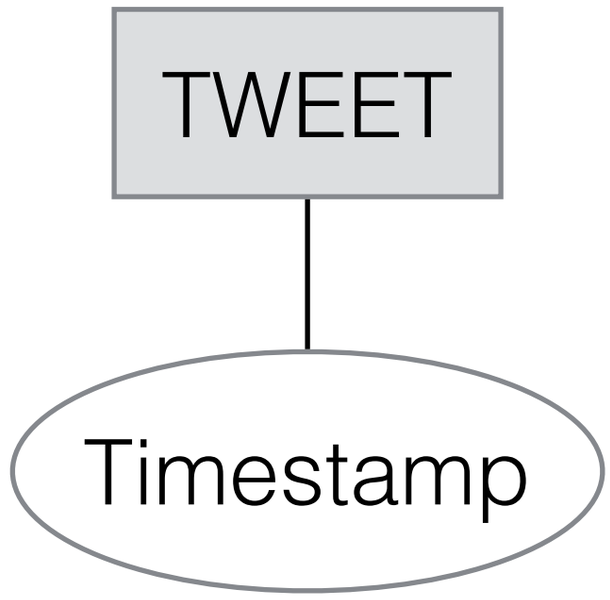


Entity-Relationship (ER) Model

Clicker Question!
Which representation is better?



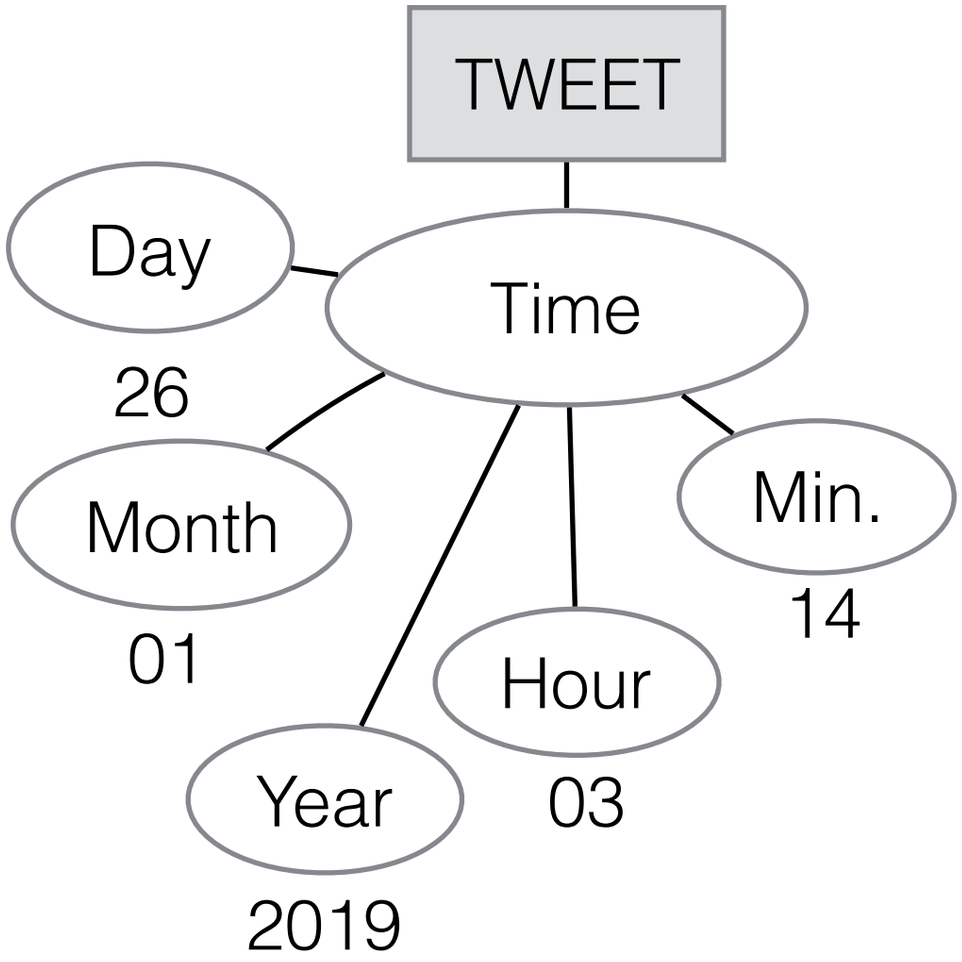
(a) Composite

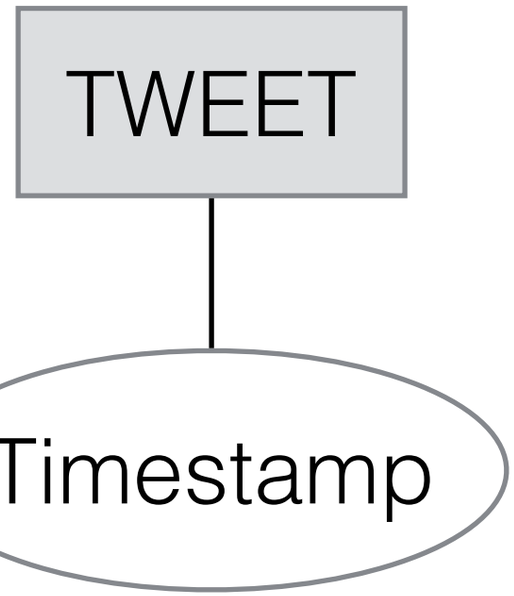


(b) Normal

Entity-Relationship (ER) Model

Clicker Question!
Which representation is better?

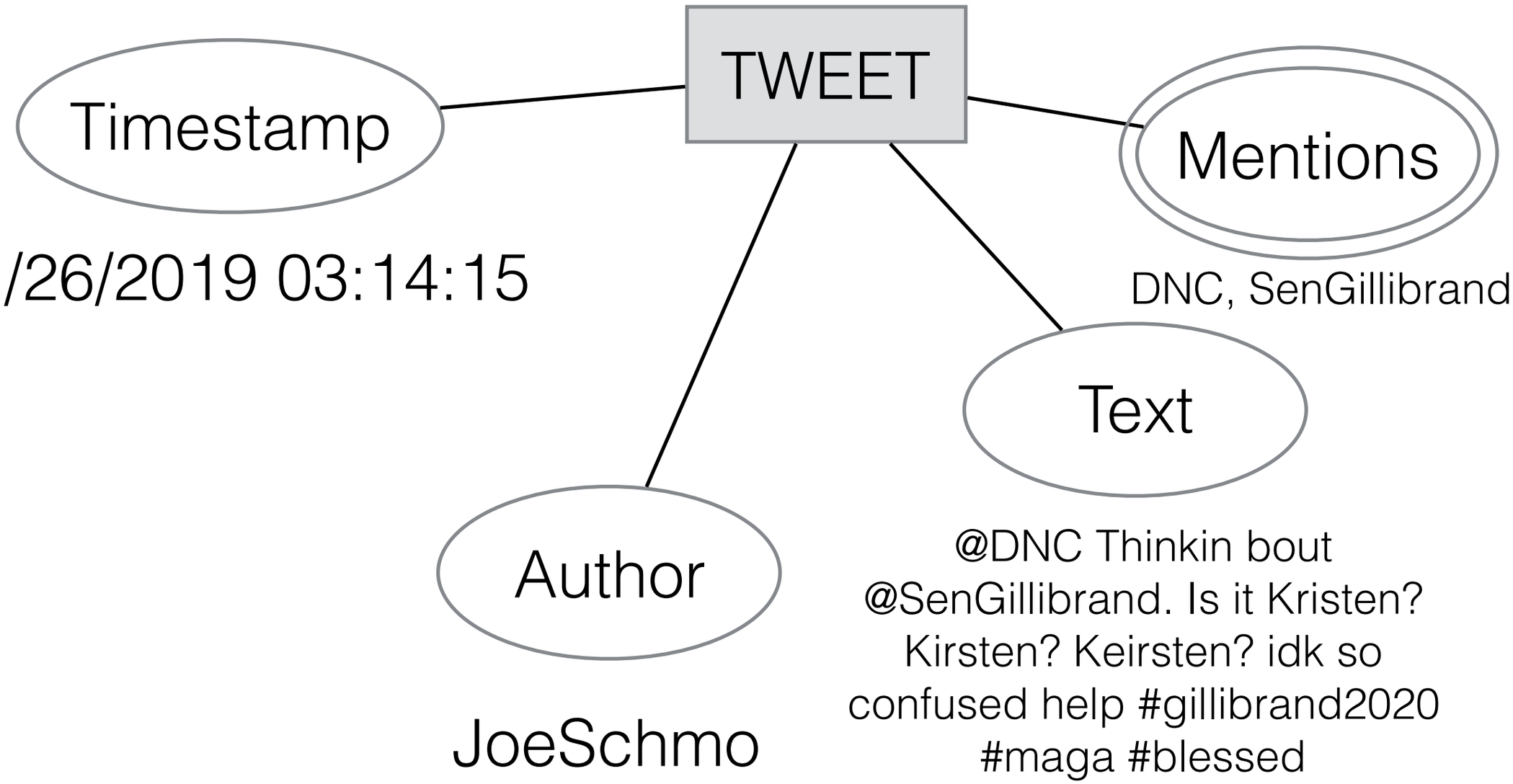




(a) Composite **(b) Normal**

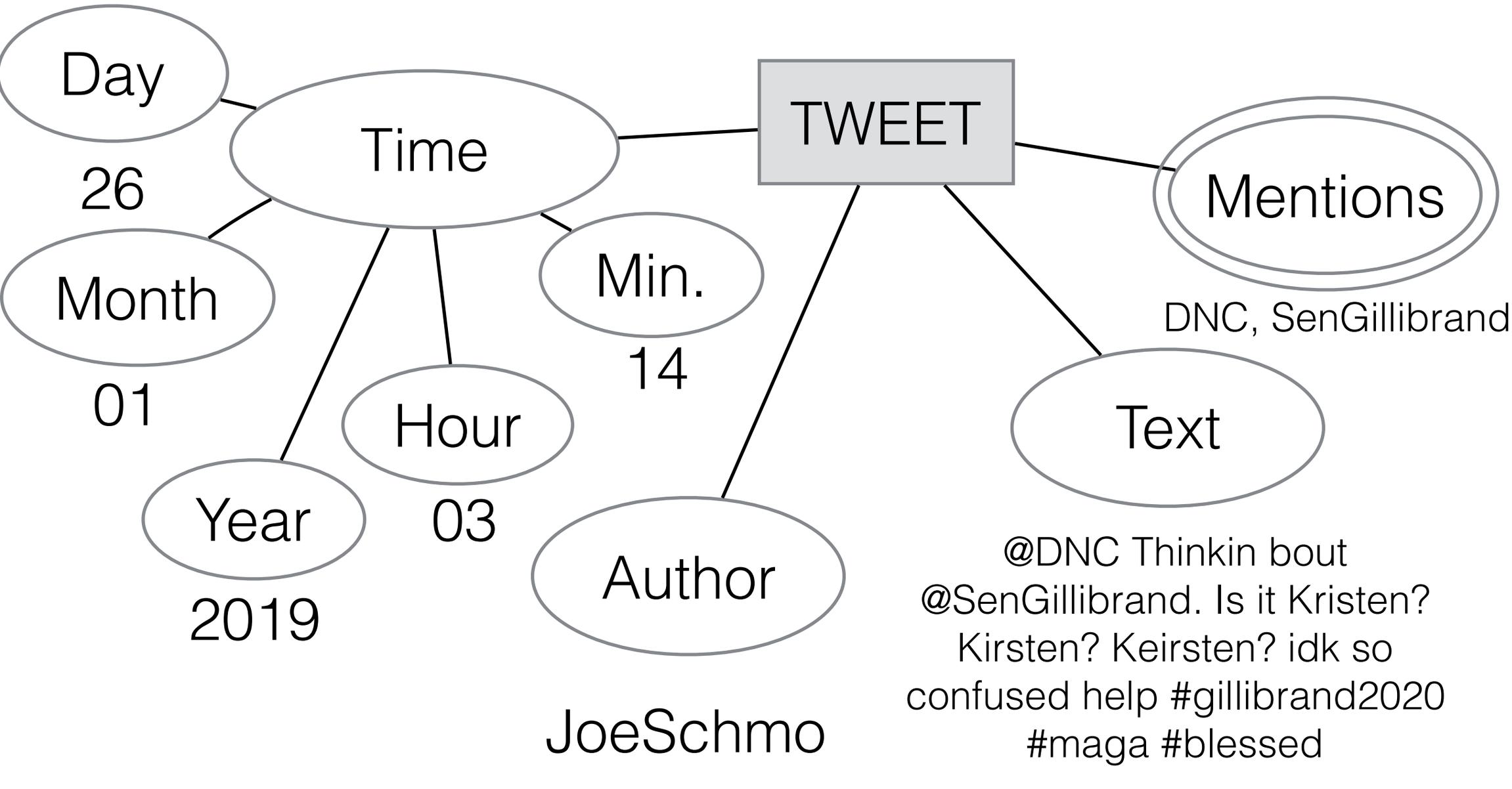
Entity-Relationship (ER) Model

Find all tweets sent between 2am and 4am that mention democratic primary candidates ❌❌❌



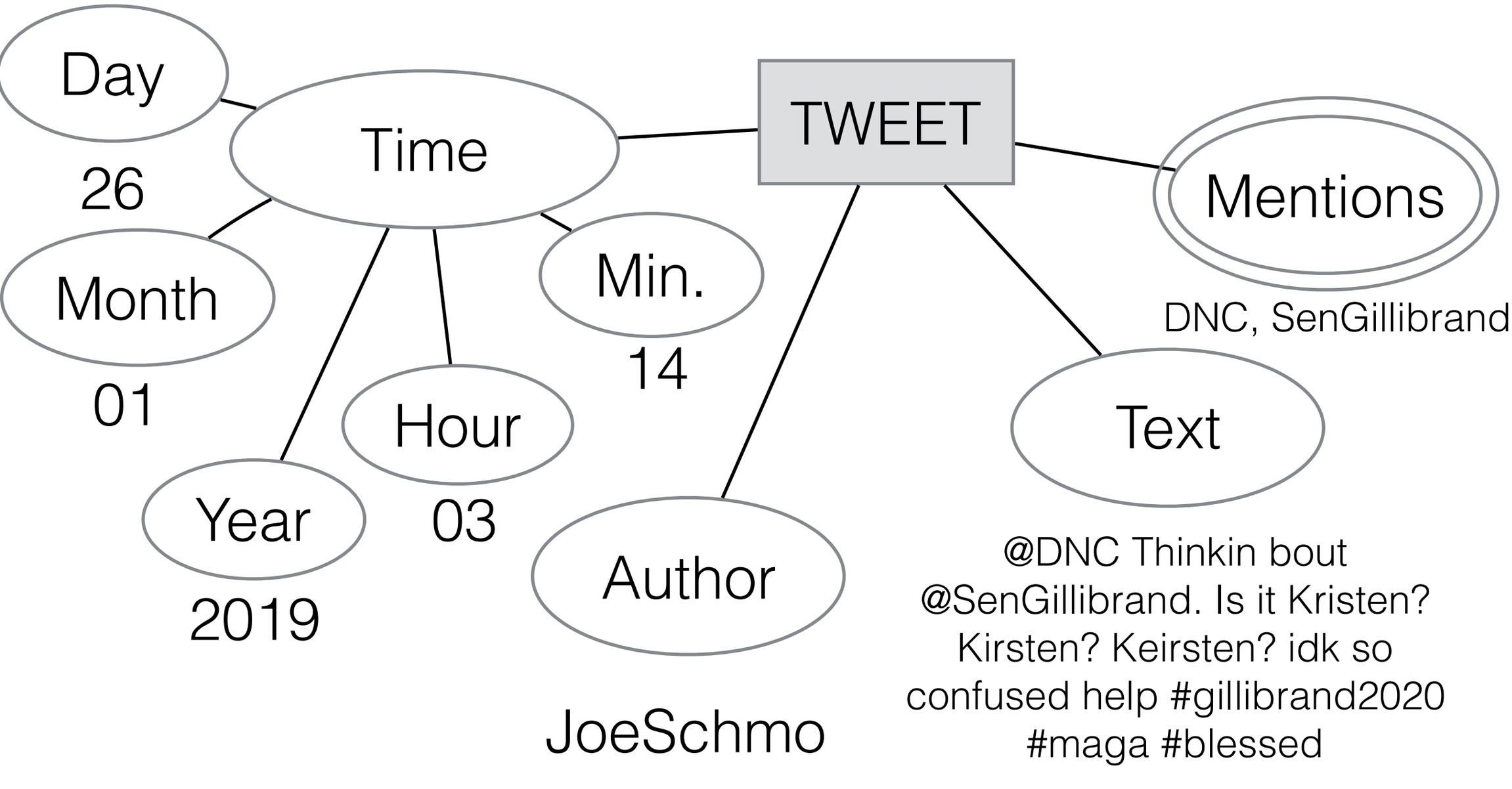
Entity-Relationship (ER) Model

Find all tweets sent between 2am and 4am that mention democratic primary candidates ✔✔✔



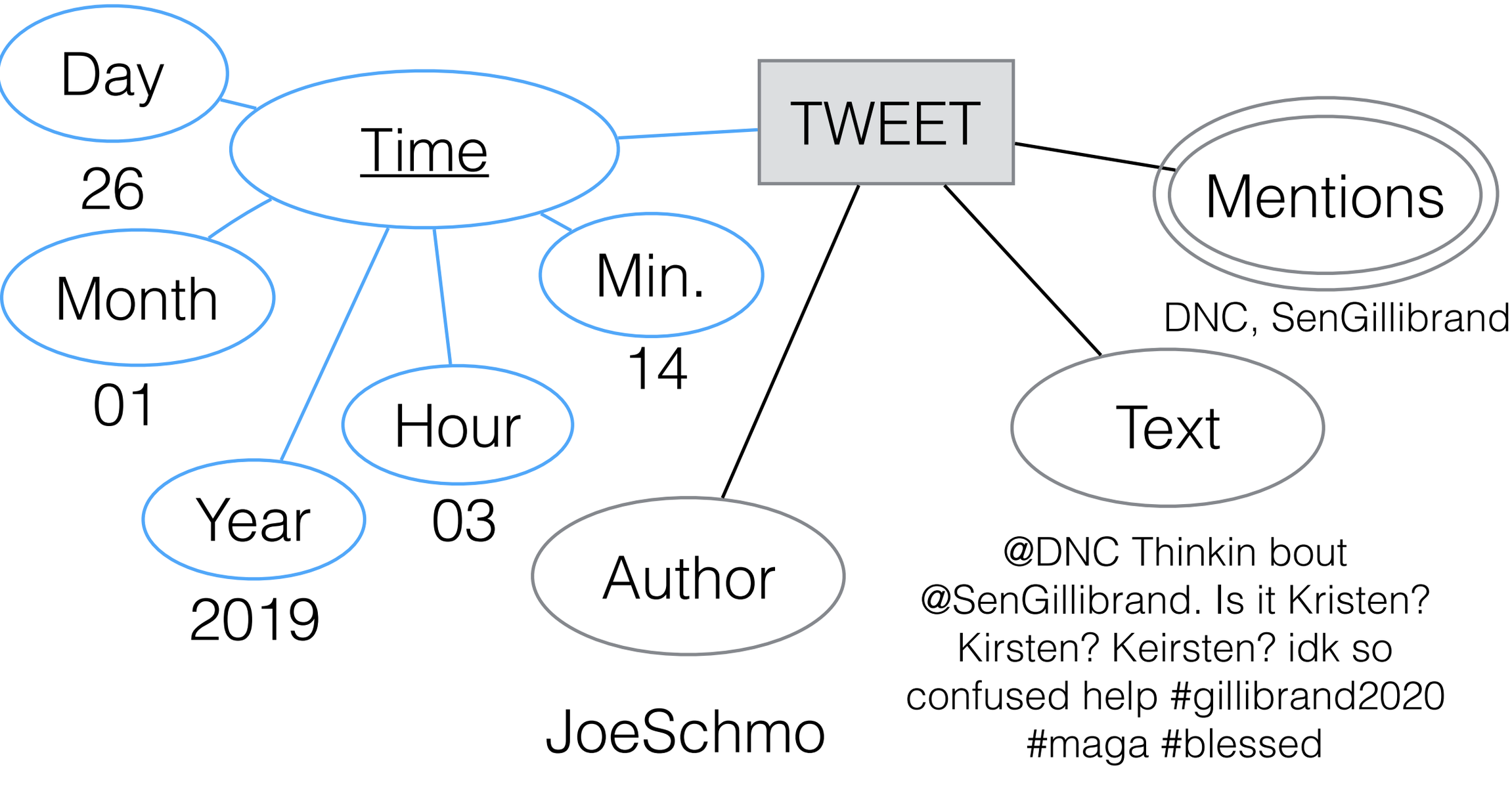
Entity-Relationship (ER) Model

Key Attribute...?



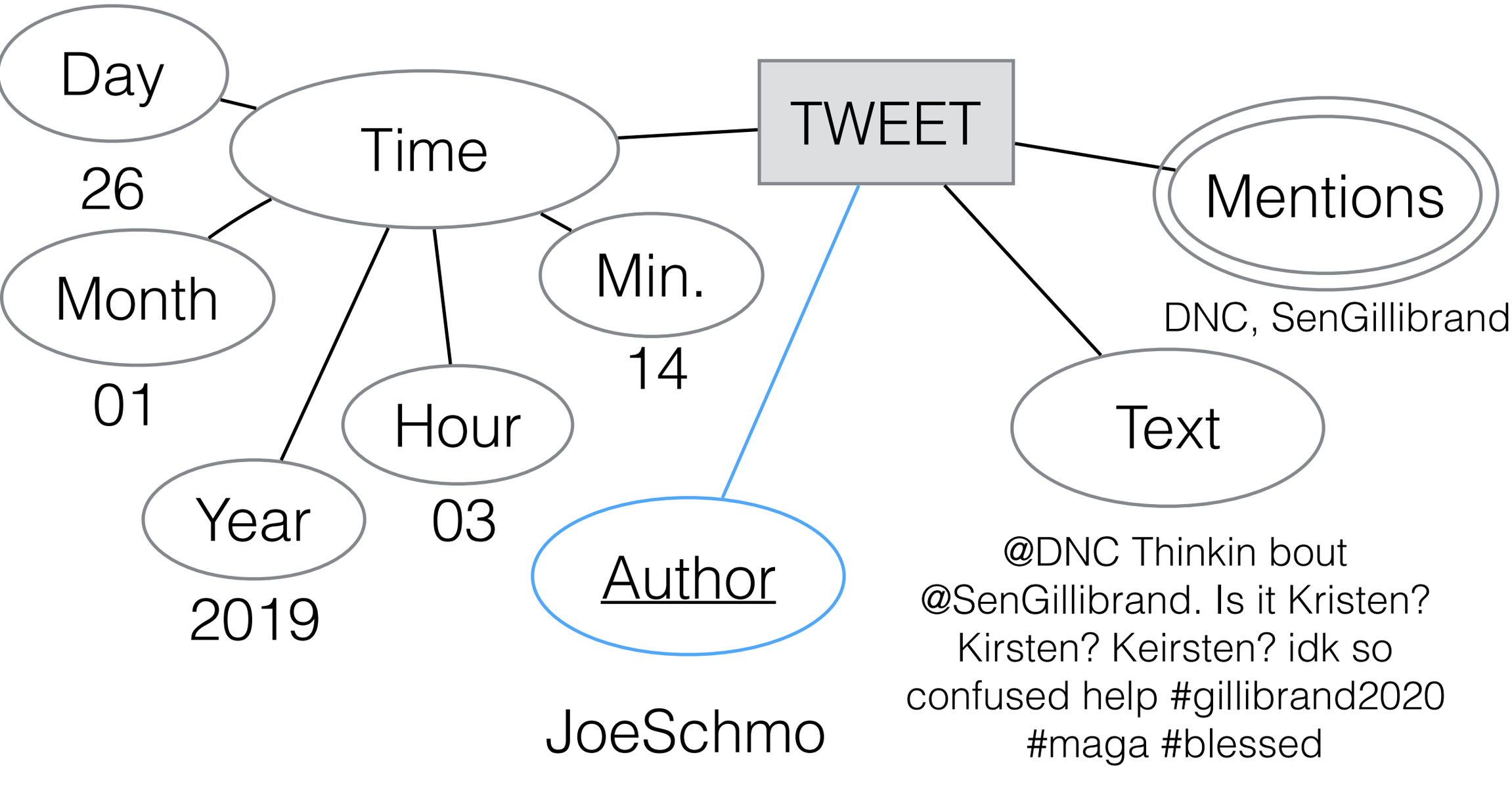
Entity-Relationship (ER) Model

Key Attribute...?



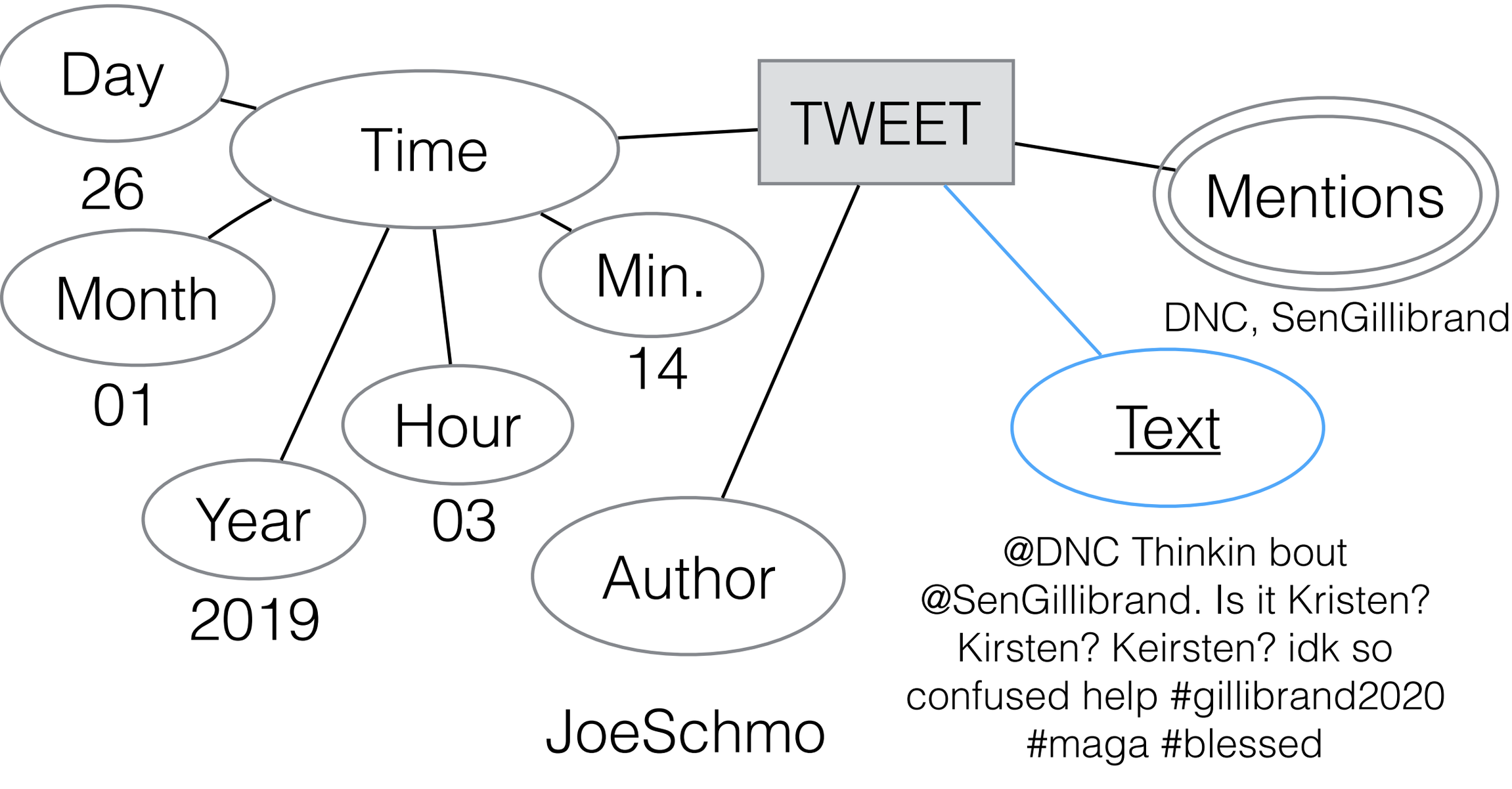
Entity-Relationship (ER) Model

Key Attribute...?



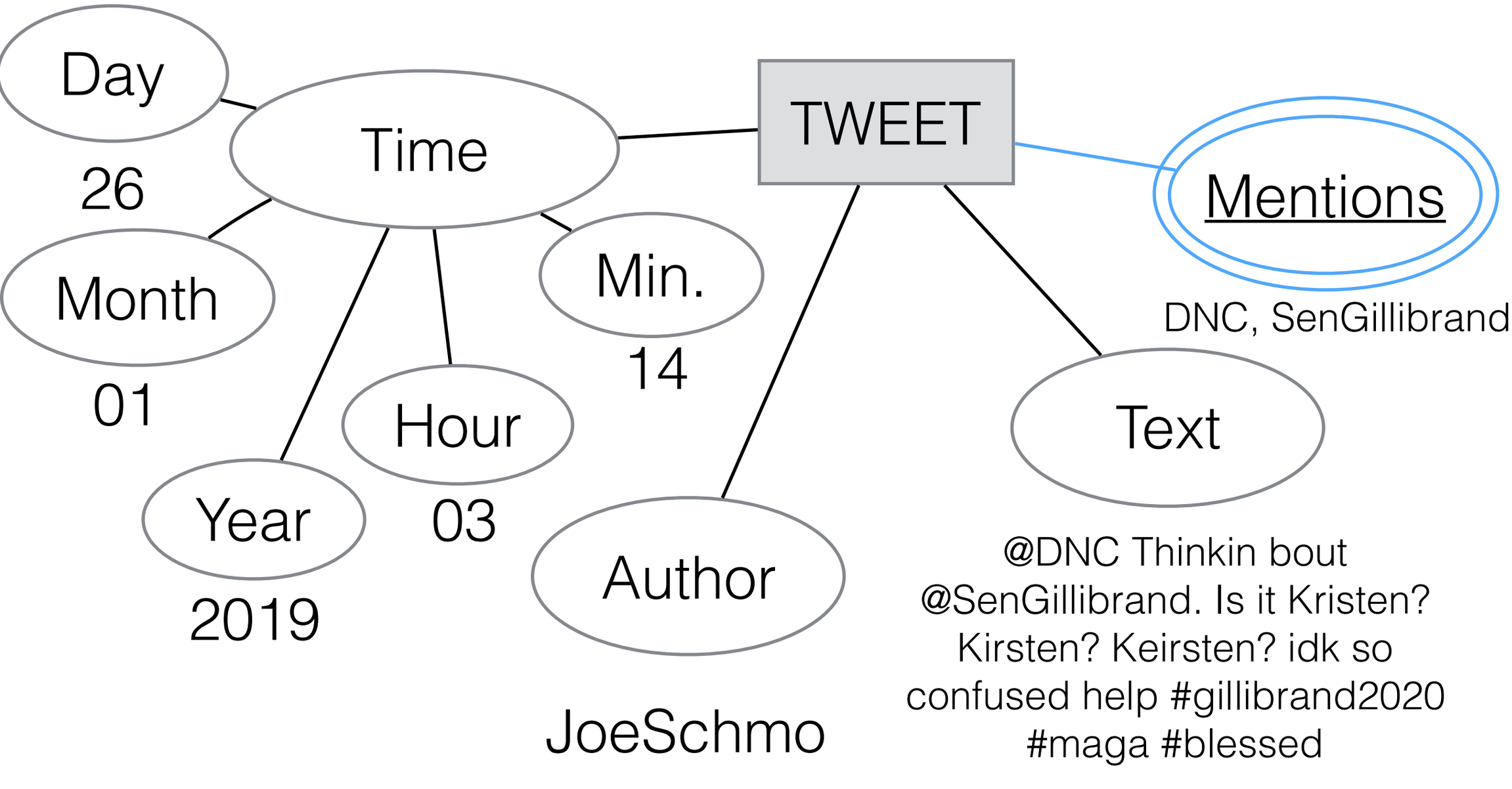
Entity-Relationship (ER) Model

Key Attribute...?



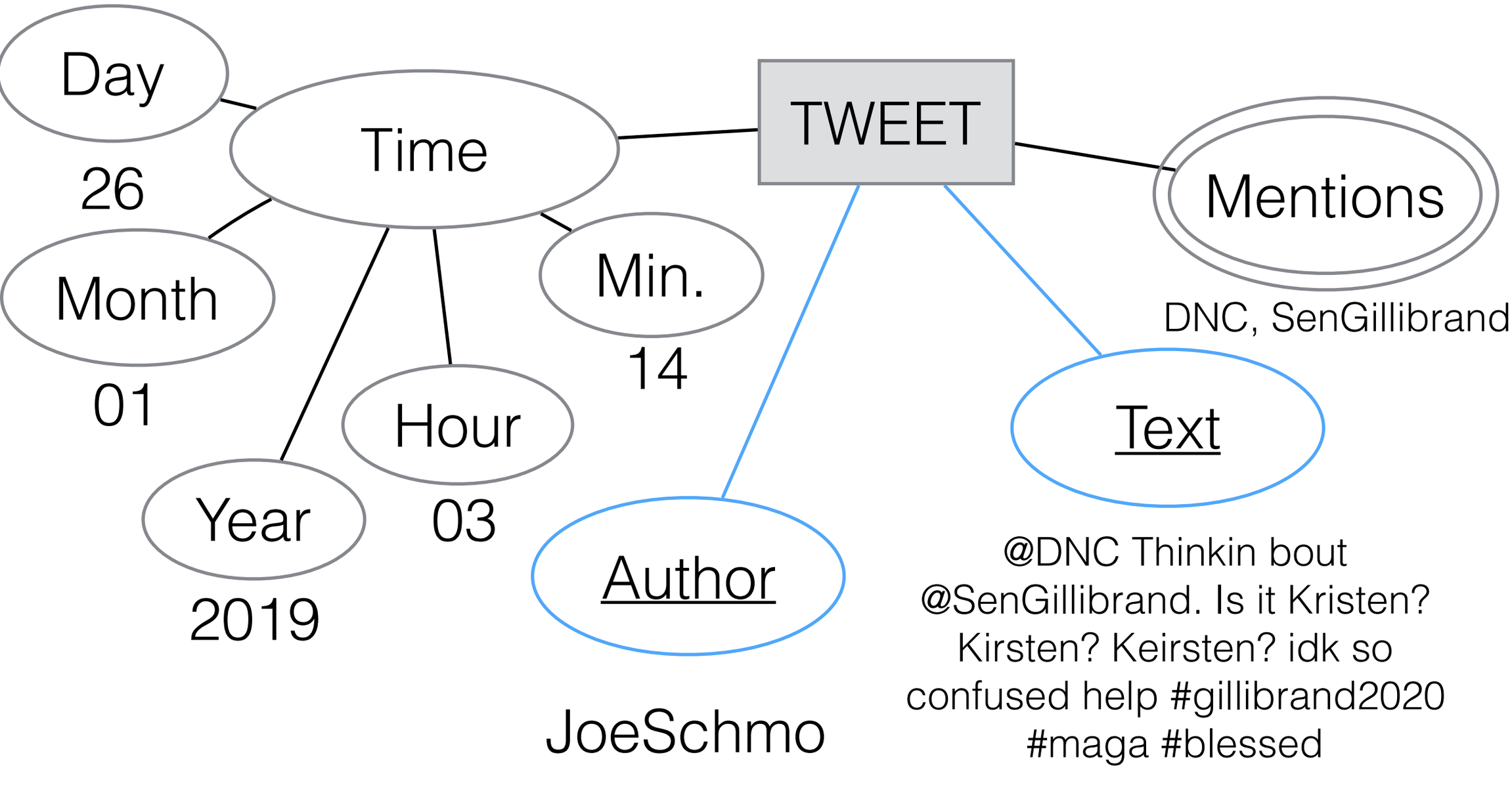
Entity-Relationship (ER) Model

Key Attribute...?



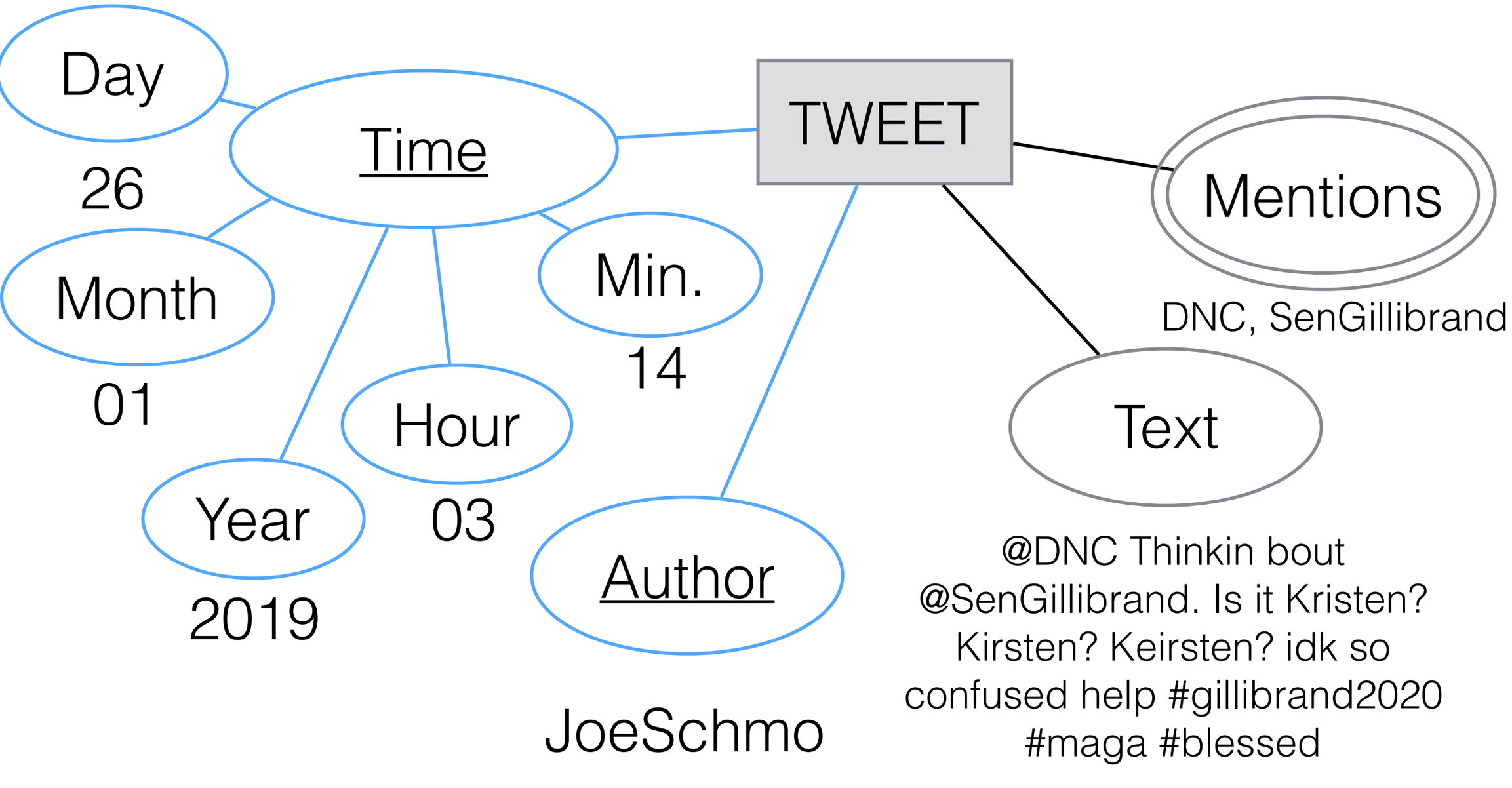
Entity-Relationship (ER) Model

Key Attribute...?



Entity-Relationship (ER) Model

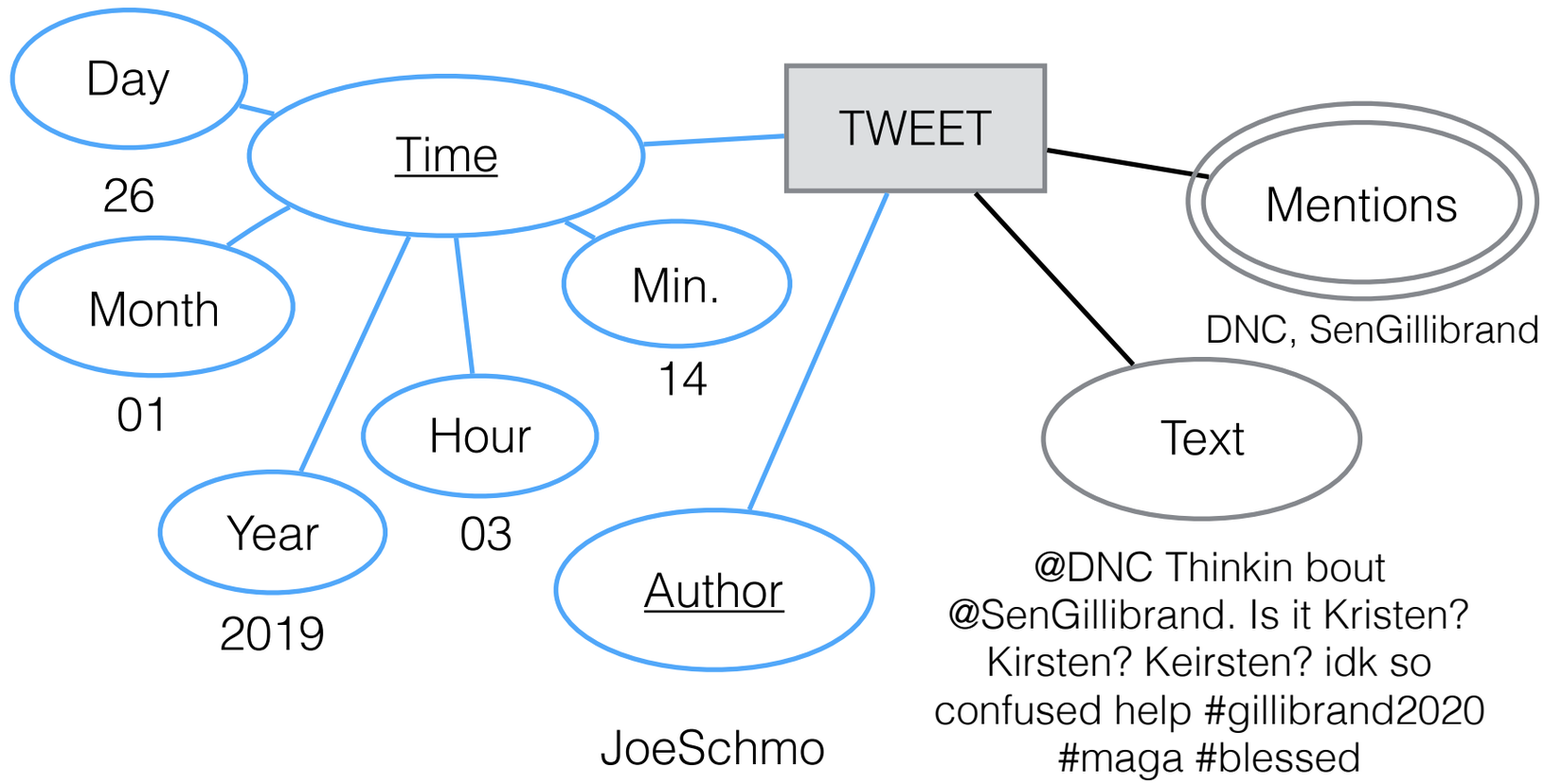
Key Attribute...?



Entity-Relationship (ER) Model

Clicker Question!

Is it a good idea to use author +timestamp as a key?



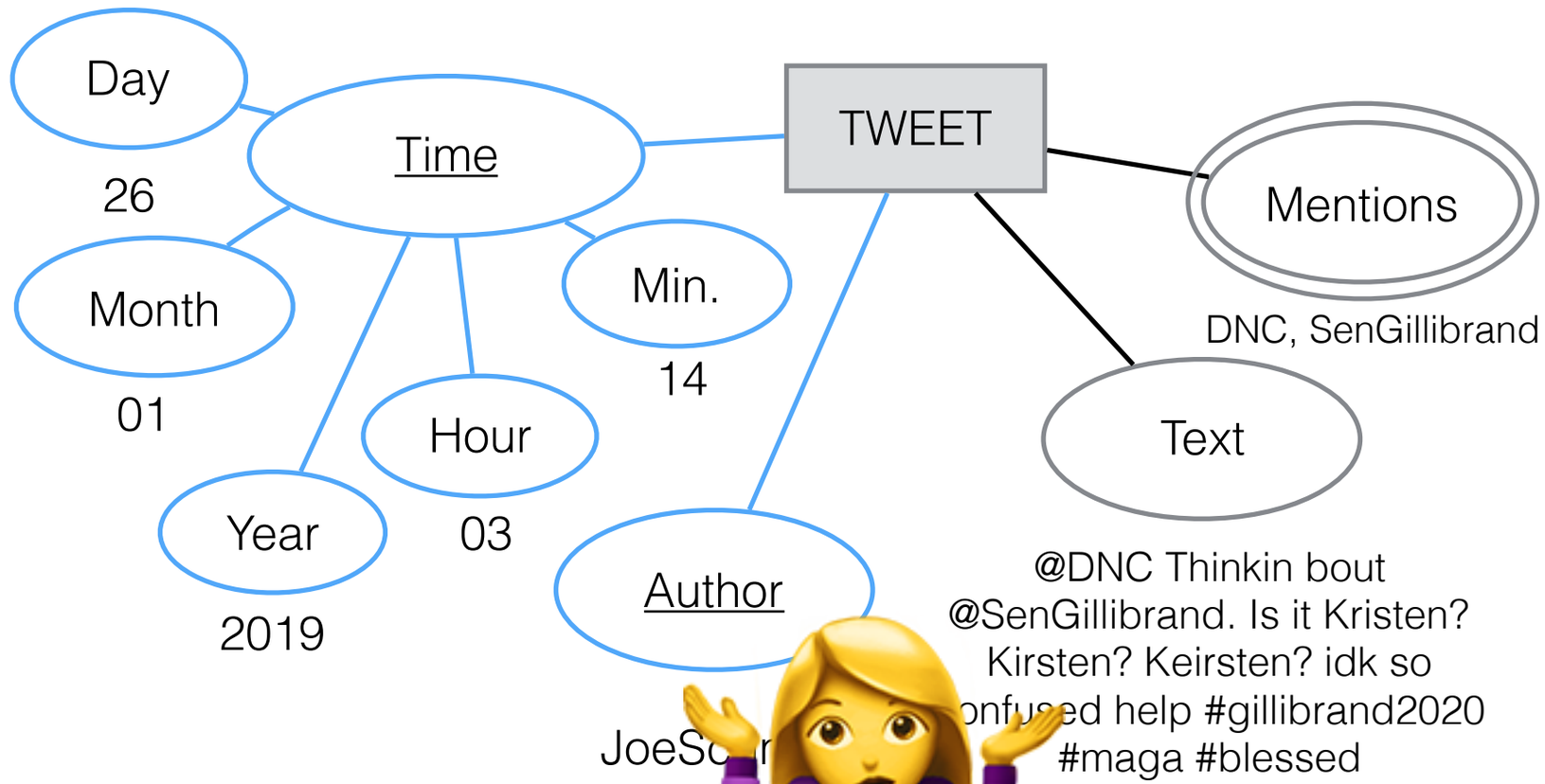
(a) Yes!

(b) No!

Entity-Relationship (ER) Model

Clicker Question!

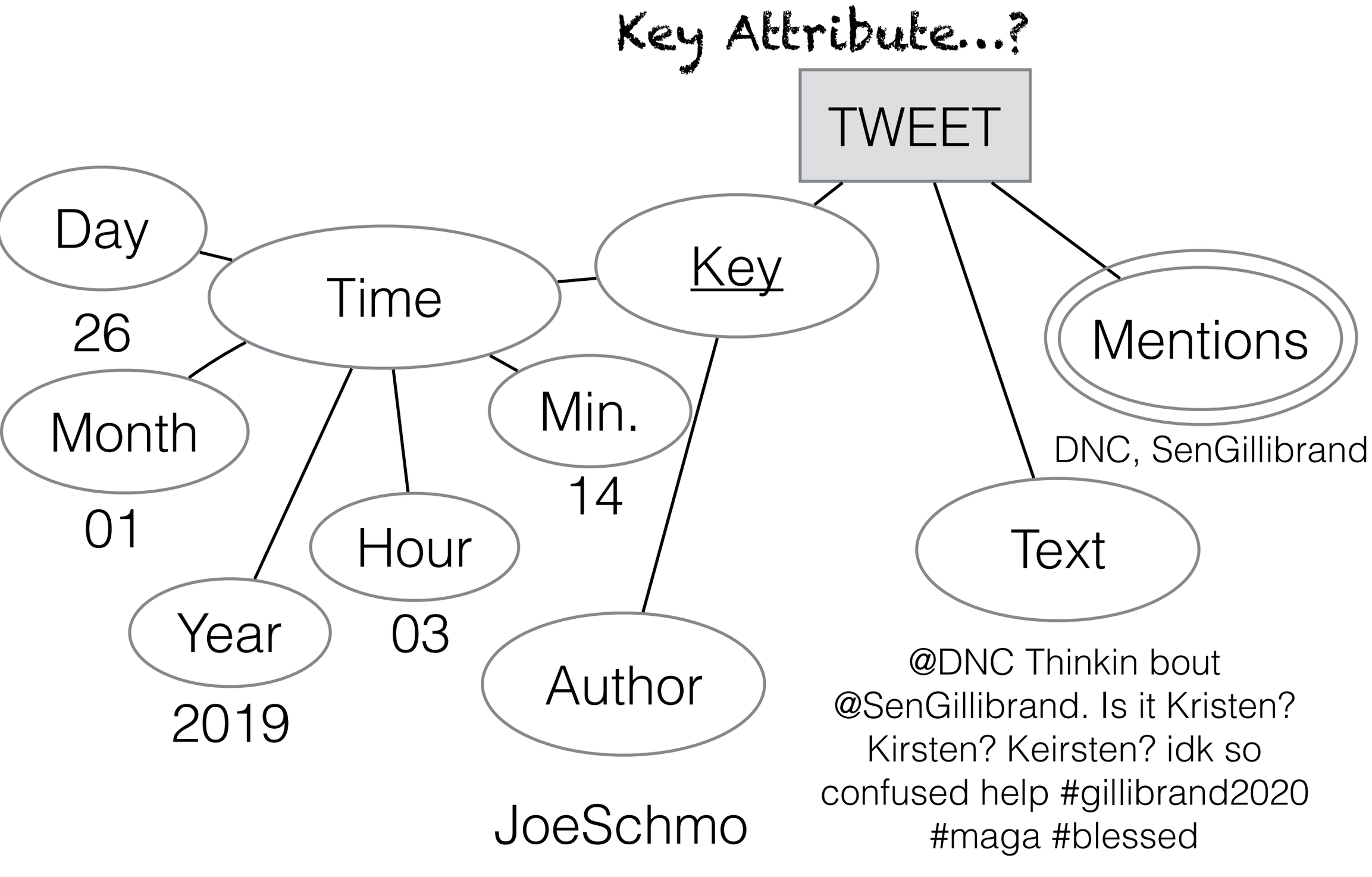
Is it a good idea to use author +timestamp as a key?



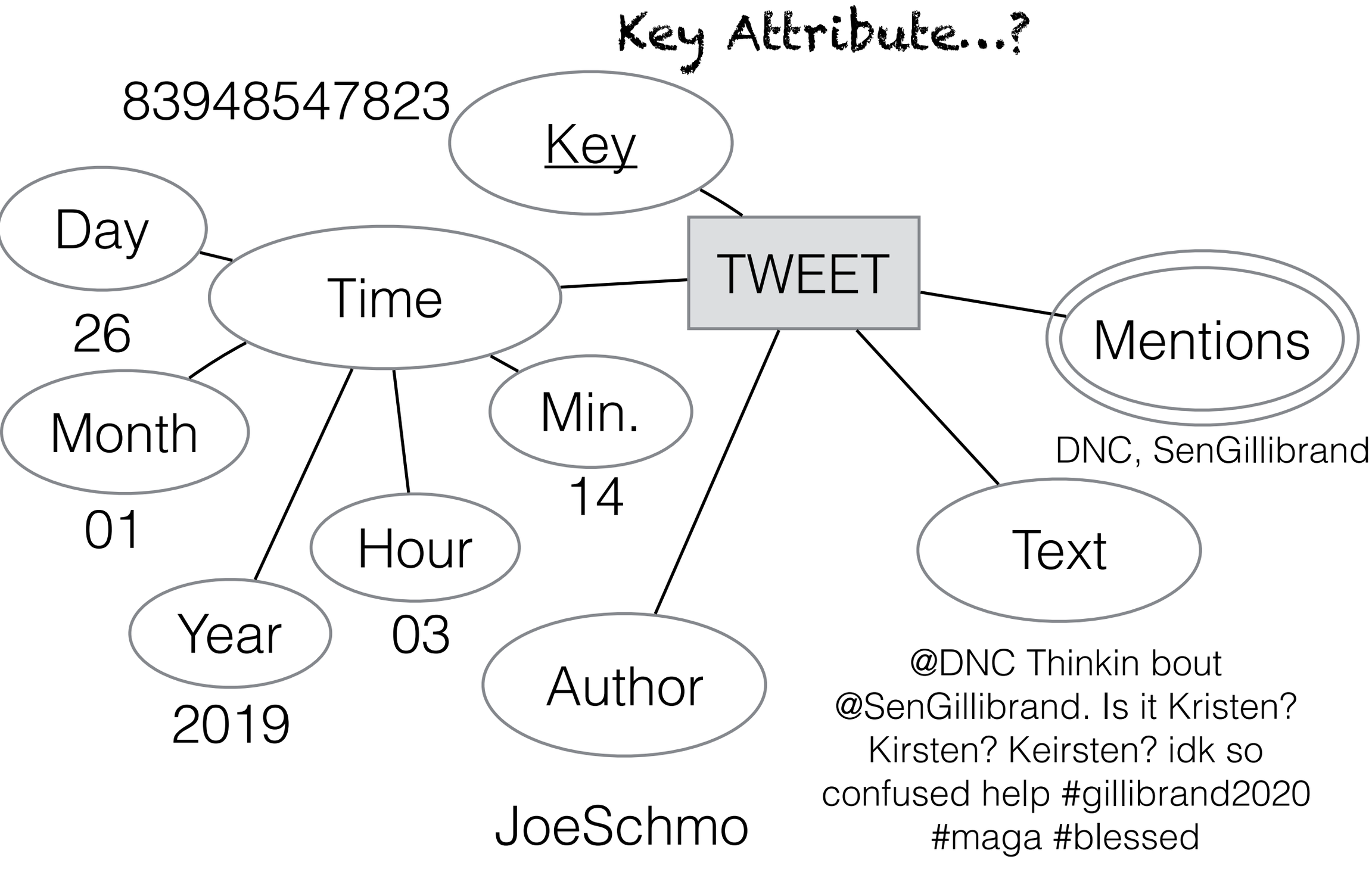
(a) Yes!

(b) No!

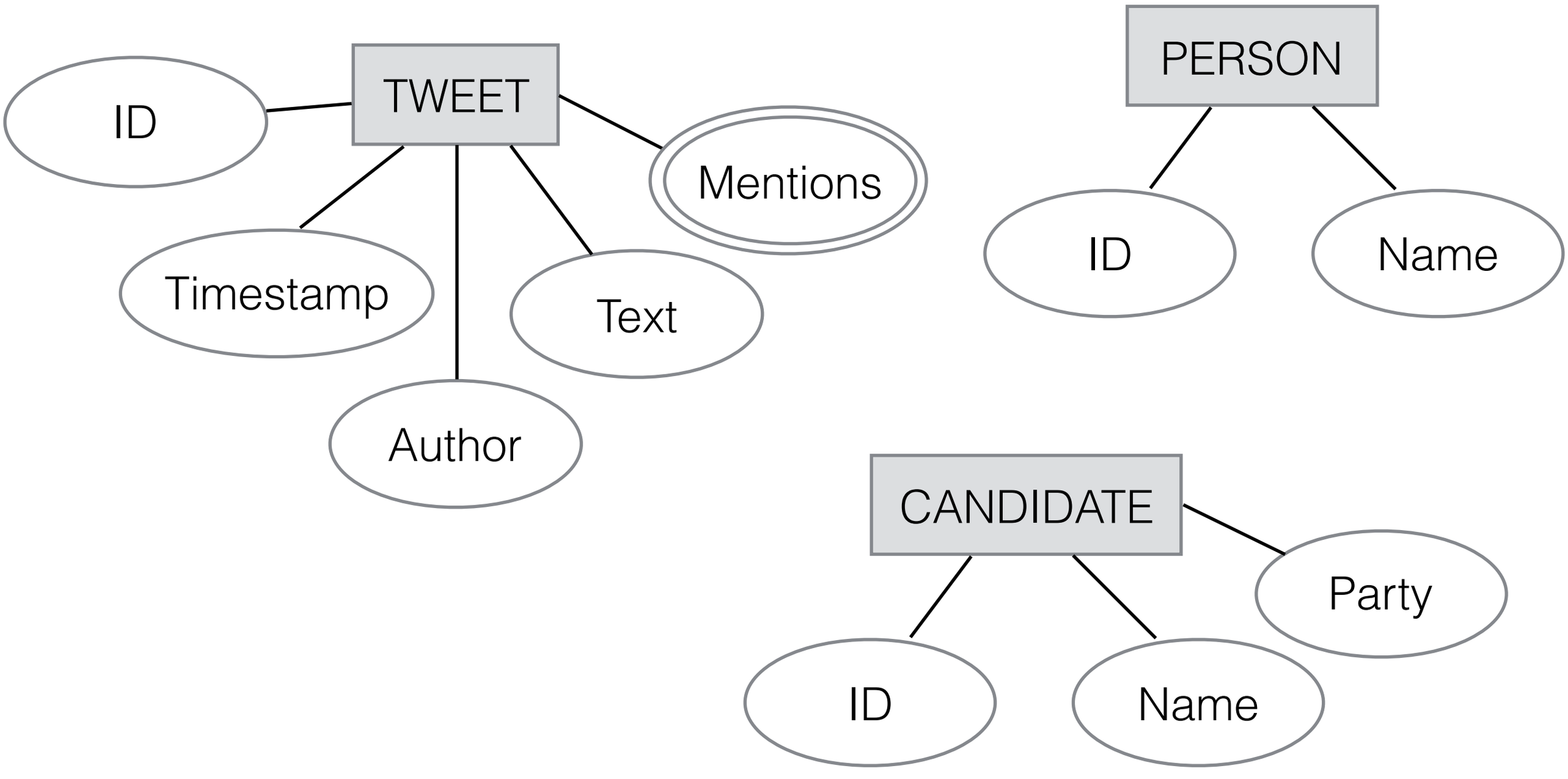
Entity-Relationship (ER) Model



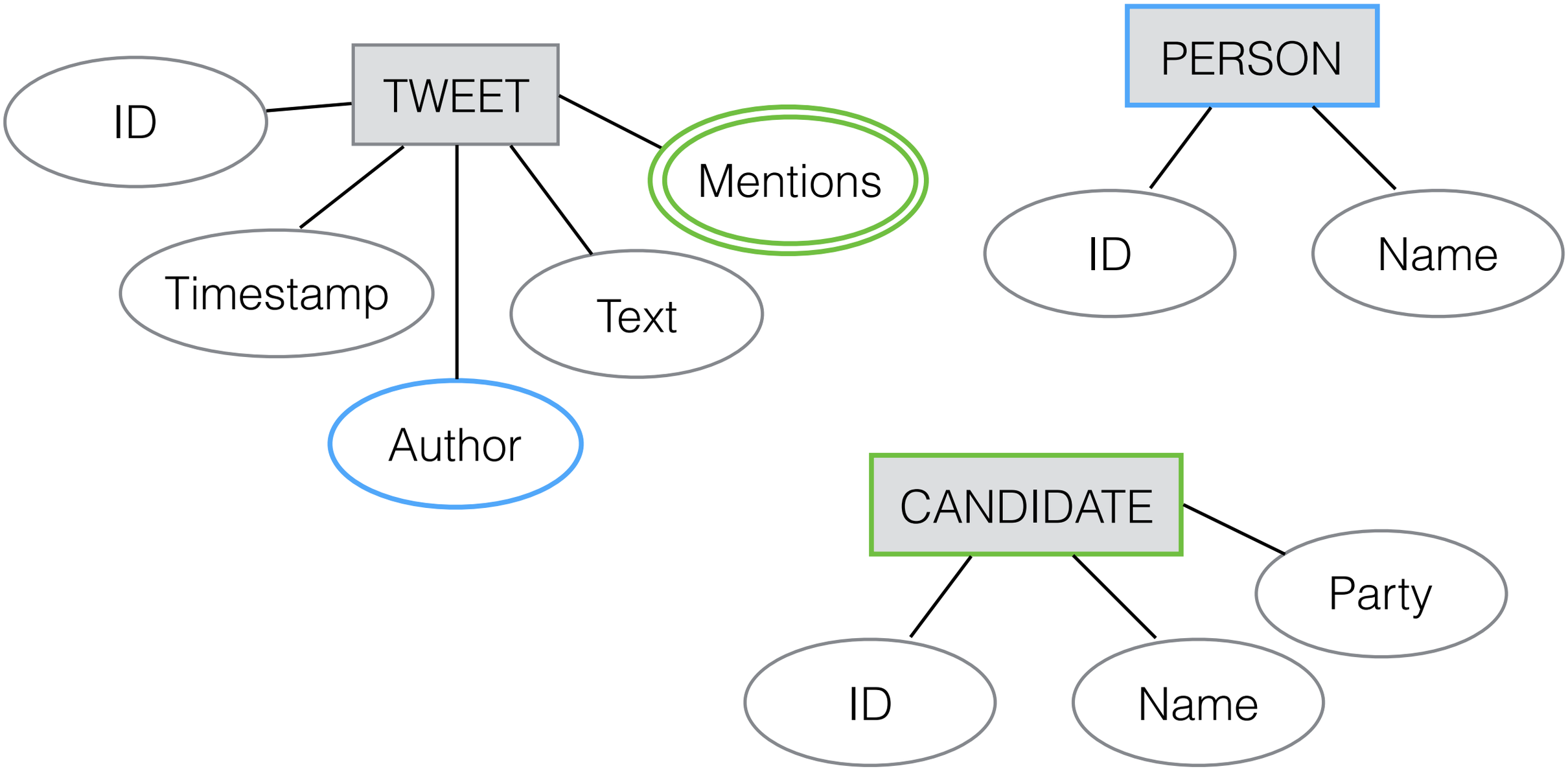
Entity-Relationship (ER) Model



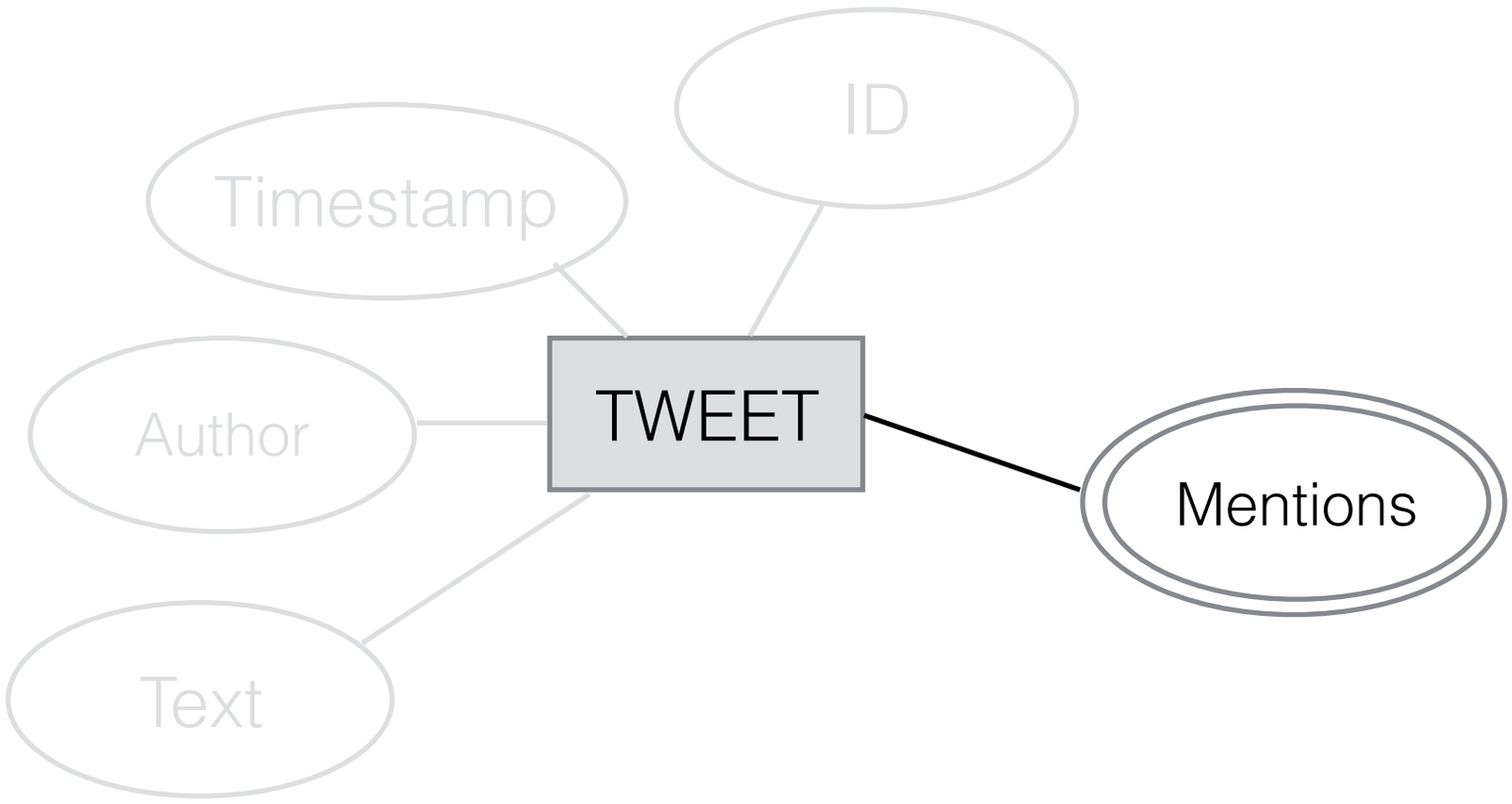
Entity-Relationship (ER) Model

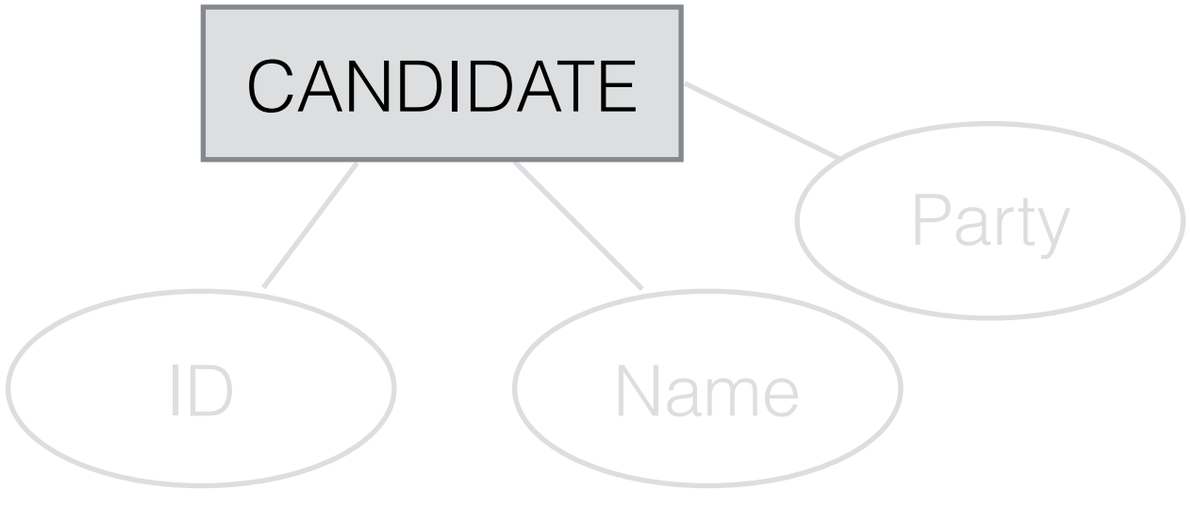


Entity-Relationship (ER) Model

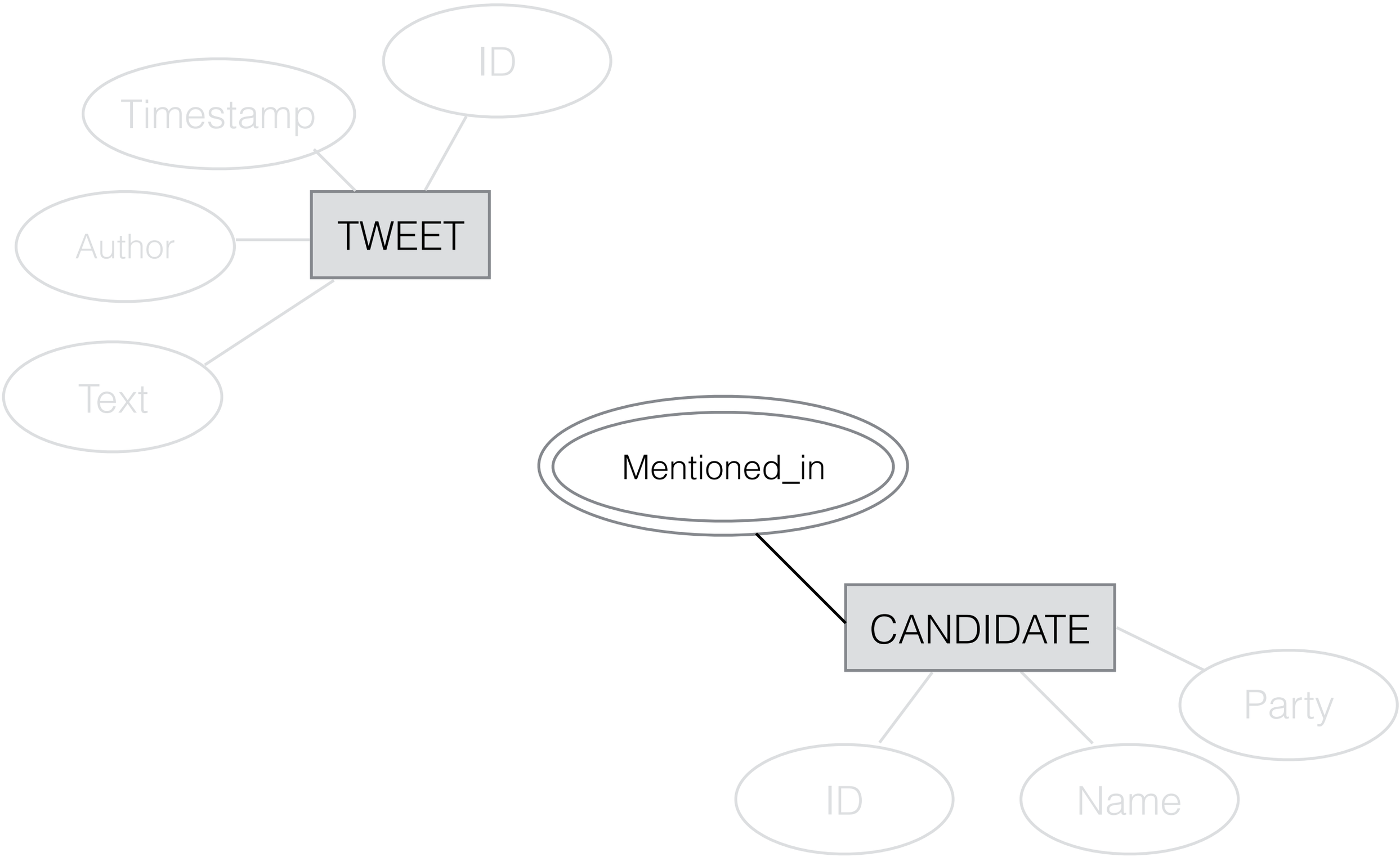


Entity-Relationship (ER) Model

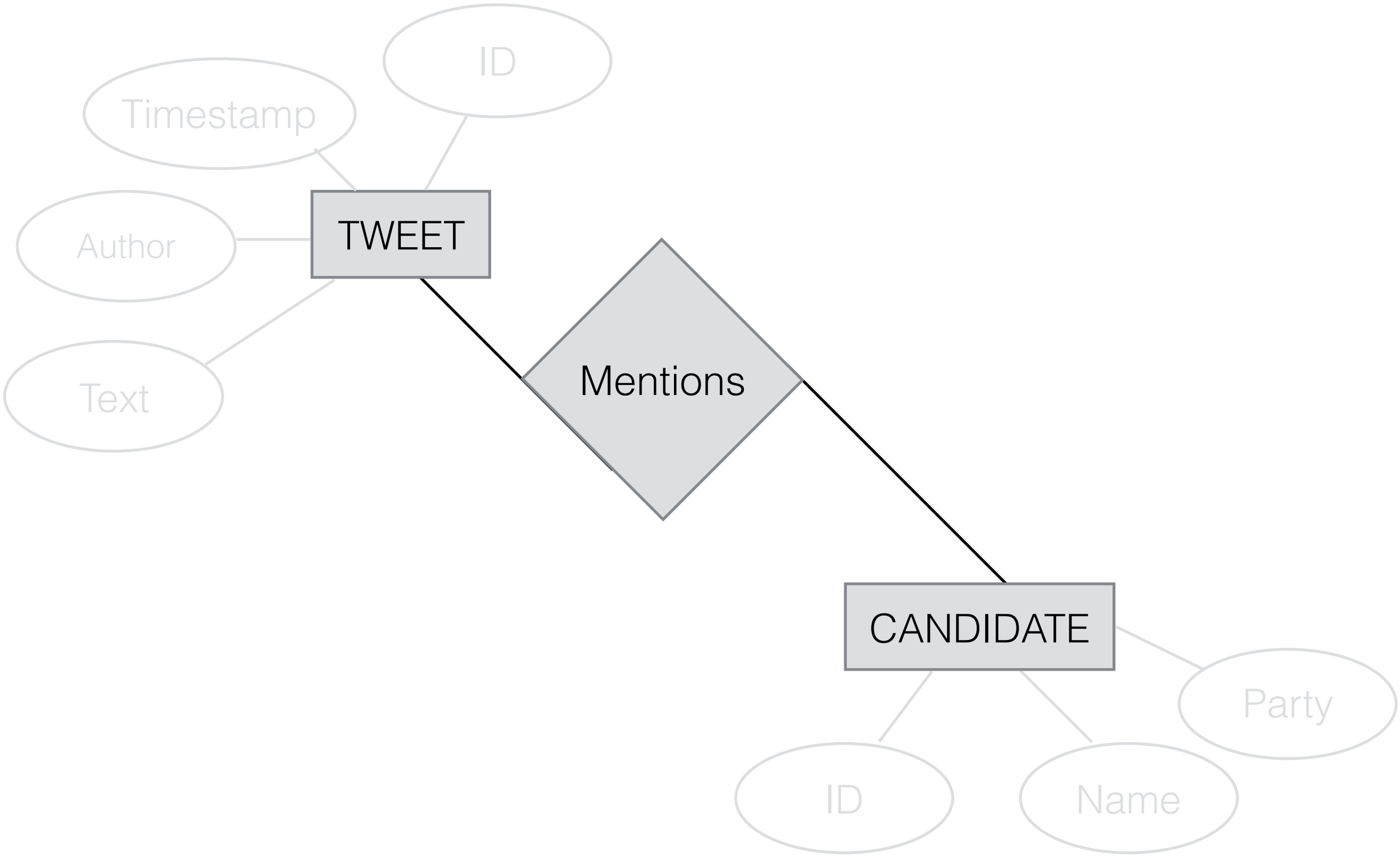




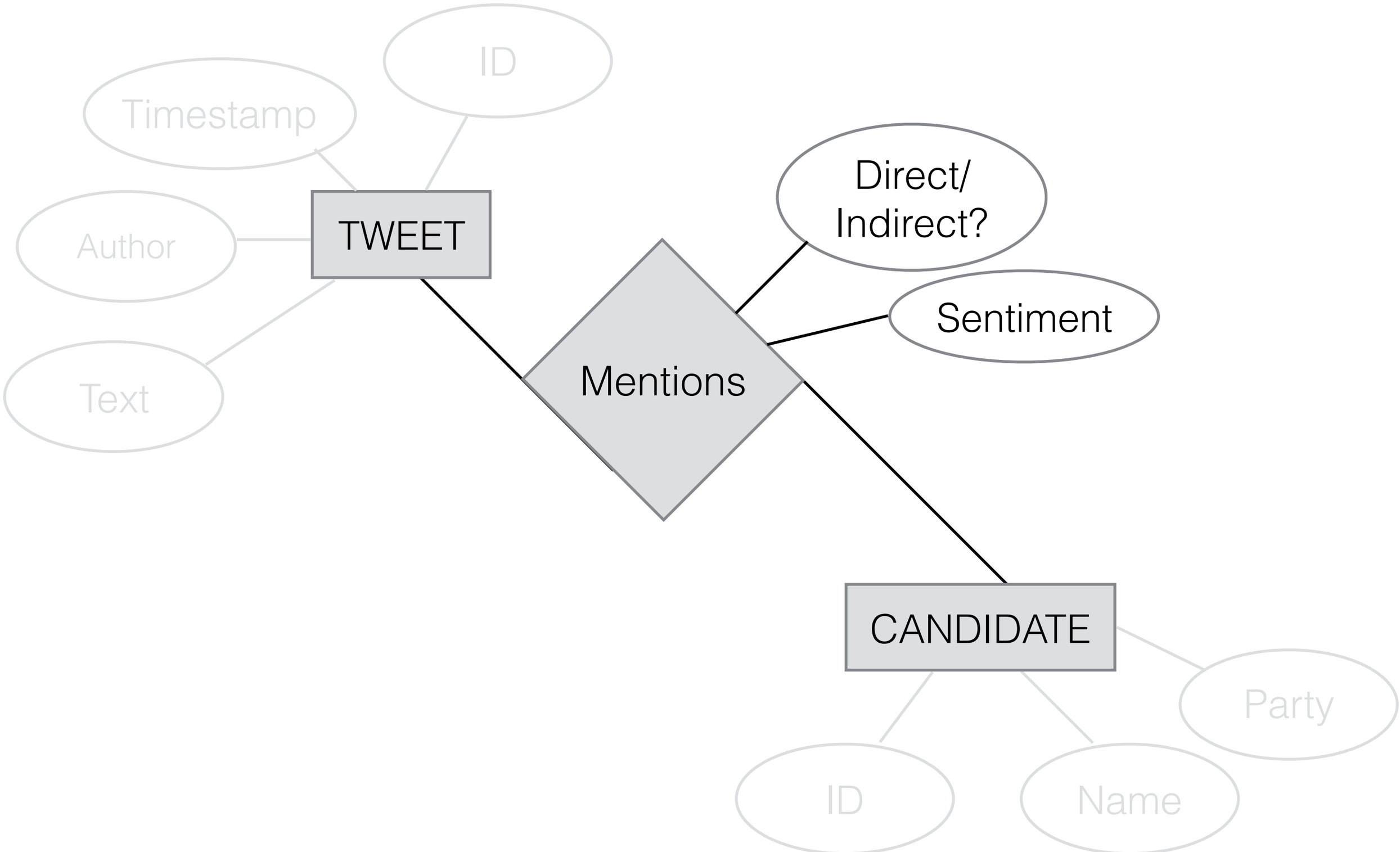
Entity-Relationship (ER) Model



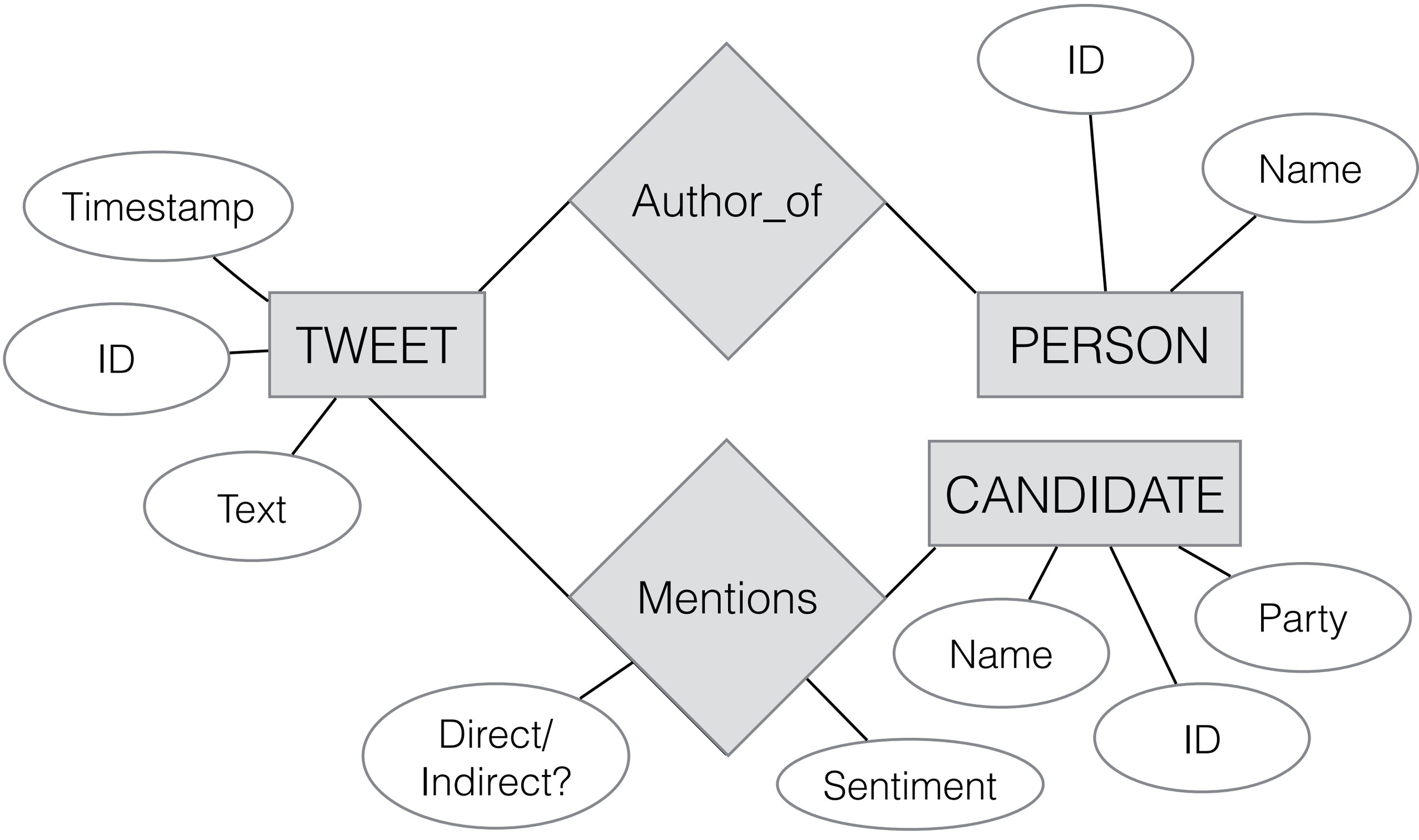
Entity-Relationship (ER) Model



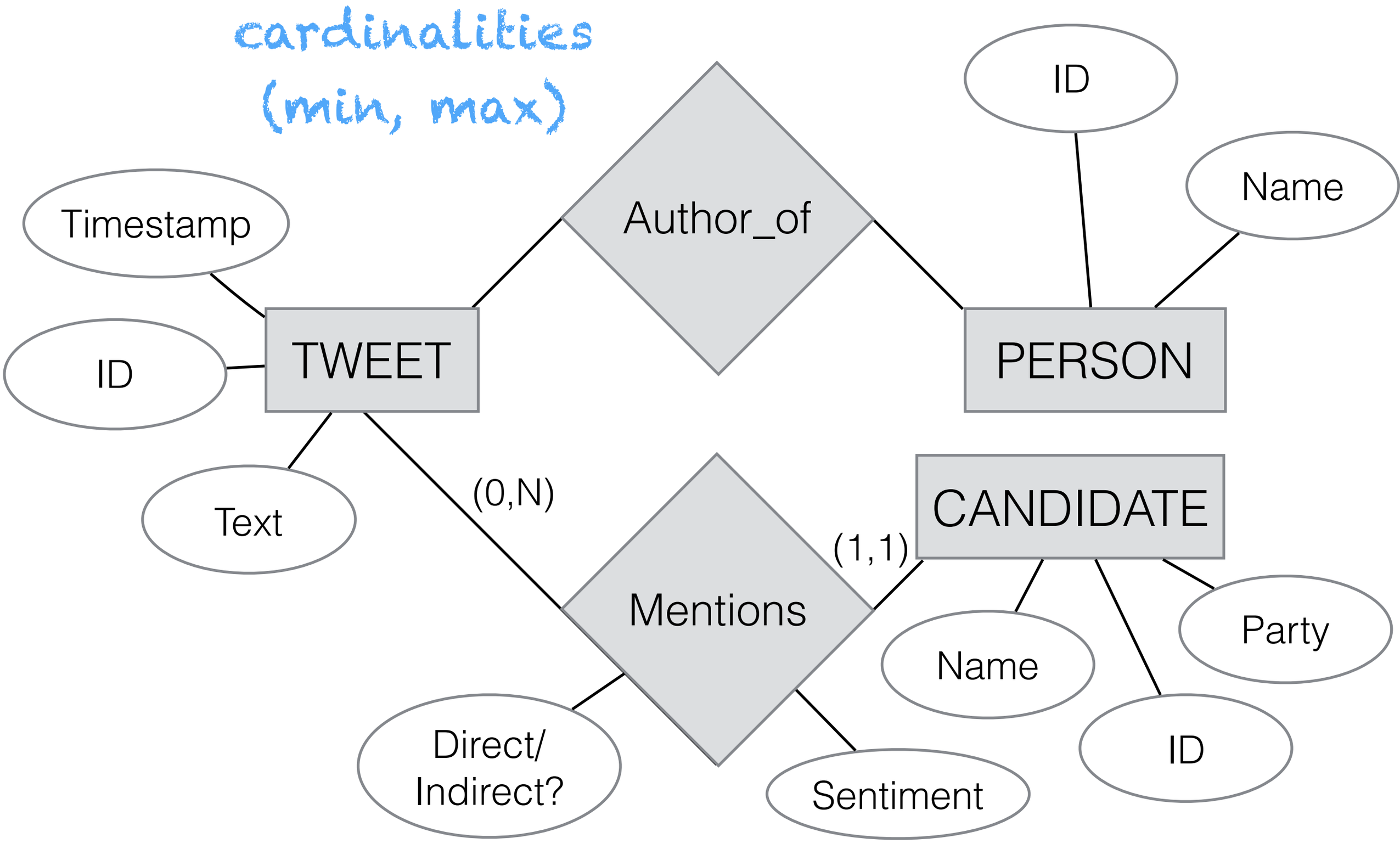
Entity-Relationship (ER) Model



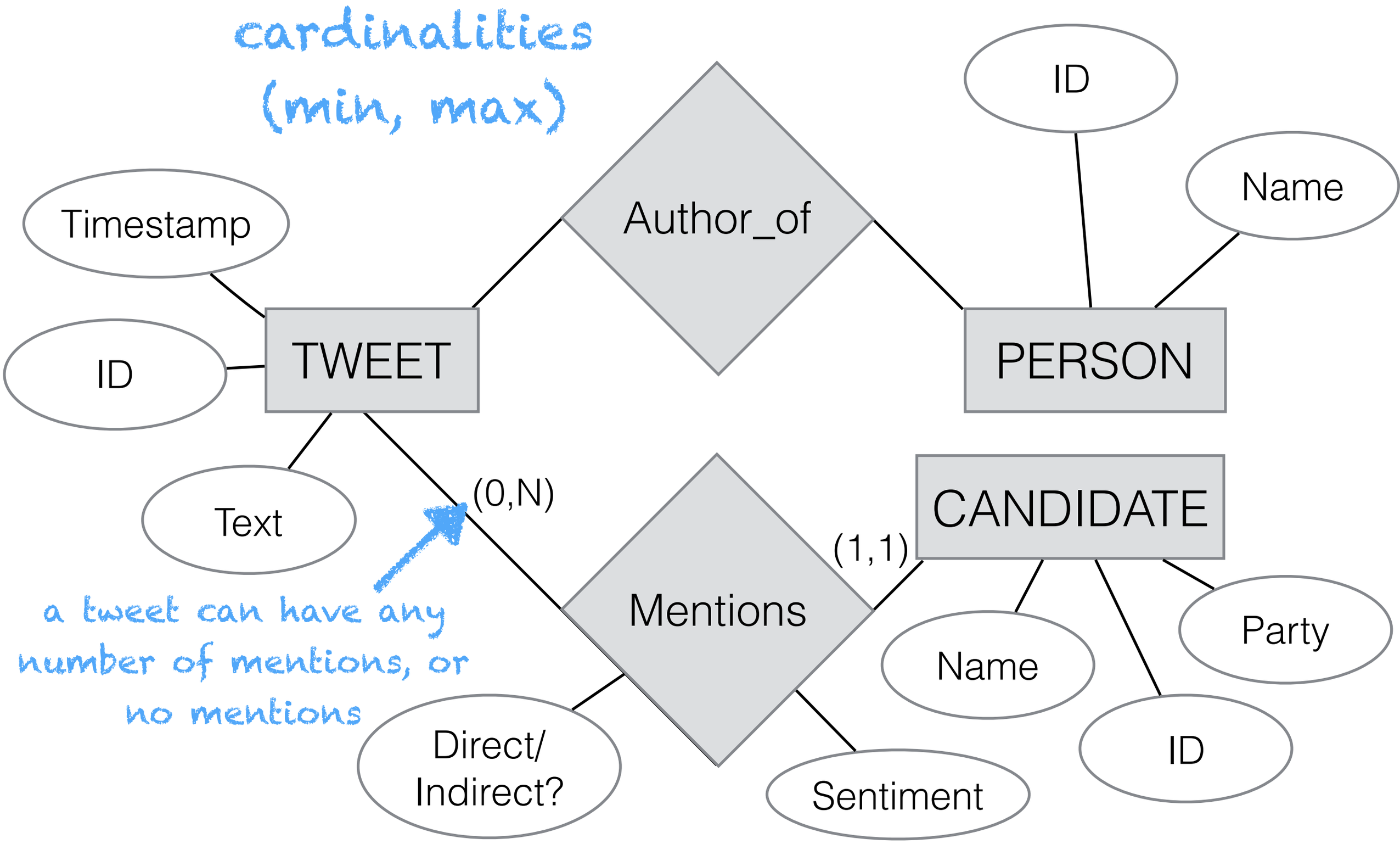
Entity-Relationship (ER) Model



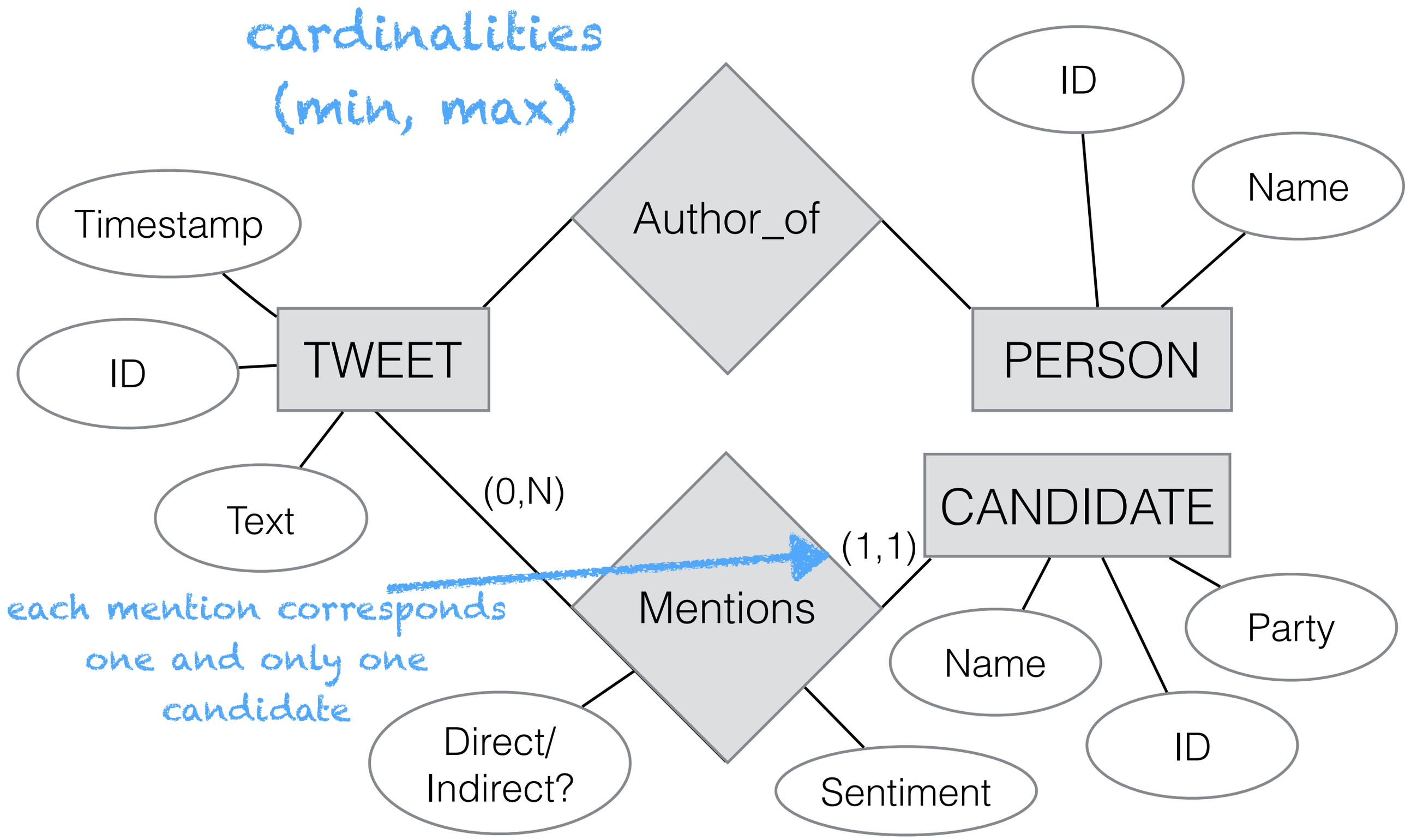
Entity-Relationship (ER) Model



Entity-Relationship (ER) Model



Entity-Relationship (ER) Model



Design Decisions

Design Decisions

- You can talk about things that don't exist! Make sure the representation supports the analysis you want to do.

Design Decisions

- You can talk about things that don't exist! Make sure the representation supports the analysis you want to do.
- Should this concept be an entity? Attribute? Relation?

Design Decisions

- You can talk about things that don't exist! Make sure the representation supports the analysis you want to do.
- Should this concept be an entity? Attribute? Relation?
    - As with most things, there is no good answer

Design Decisions

- You can talk about things that don't exist! Make sure the representation supports the analysis you want to do.
- Should this concept be an entity? Attribute? Relation?
  - As with most things, there is no good answer
  - Draft, refine, document, iterate…

Before we proceed…

Burning Questions?

DATABASES FOR DATA SCIENTIST

Requirement Engineering	Conceptual Modeling	Logical and Physical Modeling	Asking and Answering Questions (Analysis)
“Book of Duty”	Conceptual Design (ER)	Logical Design (schema, table names, data types), Physical Design (indices, memory layout, optimizations)	Relational Algebra, SQL

https://cs.brown.edu/courses/csci1951-a/res/static/lec02_databases.pdf

Relational Model

Relational Model

TWEET

ID	Timestamp	Author	Text	Mentions
389472	1/1/19 12:34	Bob	hey	NULL
123794	1/1/19 12:32	Maria	lol	{Bob}
596208	1/2/19 1:04	Yu	:-D	NULL

Relational Model

Relation

TWEET

ID	Timestamp	Author	Text	Mentions
389472	1/1/19 12:34	Bob	hey	NULL
123794	1/1/19 12:32	Maria	lol	{Bob}
596208	1/2/19 1:04	Yu	:-D	NULL

Relational Model

TWEET ← Relation Name

ID	Timestamp	Author	Text	Mentions
389472	1/1/19 12:34	Bob	hey	NULL
123794	1/1/19 12:32	Maria	lol	{Bob}
596208	1/2/19 1:04	Yu	:-D	NULL

Relational Model

Attribute

TWEET

ID	Timestamp	Author	Text	Mentions
389472	1/1/19 12:34	Bob	hey	NULL
123794	1/1/19 12:32	Maria	lol	{Bob}
596208	1/2/19 1:04	Yu	:-D	NULL

Domain: D = dom(Timestamp) =

Valid time strings = ##/##/#### ##:##

Relational Model

TWEET

ID	Timestamp	Author	Text	Mentions
389472	1/1/19 12:34	Bob	hey	NULL
123794	1/1/19 12:32	Maria	lol	{Bob}
596208	1/2/19 1:04	Yu	:-D	NULL

Tuple

Relational Model

TWEET:
<ID:int, Timestamp:time, Author:string,Text:string, Mentions:set(string)>

<389472, 1/1/19 12:34, Bob, hey, {}>

<123794, 1/1/19 12:32, Maria, lol, {Bob}>

<596208, 1/2/19 1:04, Yu, :-D, {}>

Relational Model

Relation Name

TWEET:

<ID:int, Timestamp:time, Author:string, Text:string, Mentions:set(string)>

Attribute

Domain

<389472, 1/1/19 12:34, Bob, hey, {}>

<123794, 1/1/19 12:32, Maria, lol, {Bob}>

<596208, 1/2/19 1:04, Yu, :-D, {}>

Tuple

Relational Model

Order Matters!!

TWEET:
<ID:int, Timestamp:time, Author:string,Text:string, Mentions:set(string)>

<389472, 1/1/19 12:34, Bob, hey, {}>

<123794, 1/1/19 12:32, Maria, lol, {Bob}>

<596208, 1/2/19 1:04, Yu, :-D, {}>

Relational Model

Order Matters!!

TWEET:
<ID:int, Timestamp:time, Author:string,Text:string, Mentions:set(string)>

<389472, 1/1/19 12:34, Bob, hey, {}>

<123794, 1/1/19 12:32, Maria, lol, {Bob}>

<596208, 1/2/19 1:04, Yu, :-D, {}>

Relational Model

Order Matters!!

TWEET:

<ID:int, Timestamp:time, Author:string,Text:string, Mentions:set(string)>

<389472, 1/1/19 12:34, hey, Bob, {}>

<123794, 1/1/19 12:32, Maria, lol, {Bob}>

<596208, 1/2/19 1:04, Yu, :-D, {}>

Relational Model

Relation Schema (R)

TWEET:

<ID:int, Timestamp:time, Author:string,Text:string, Mentions:set(string)>

<389472, 1/1/19 12:34, Bob, hey, {}>

<123794, 1/1/19 12:32, Maria, lol, {Bob}>

<596208, 1/2/19 1:04, Yu, :-D, {}>

Relation State r(R)

Relational Model

Intension

TWEET:

<ID:int, Timestamp:time, Author:string,Text:string, Mentions:set(string)>

<389472, 1/1/19 12:34, Bob, hey, {}>

<123794, 1/1/19 12:32, Maria, lol, {Bob}>

<596208, 1/2/19 1:04, Yu, :-D, {}>

Extension

Relational Model

Sense

TWEET:

<ID:int, Timestamp:time, Author:string,Text:string, Mentions:set(string)>

<389472, 1/1/19 12:34, Bob, hey, {}>

<123794, 1/1/19 12:32, Maria, lol, {Bob}>

<596208, 1/2/19 1:04, Yu, :-D, {}>

Reference

Find all the tweets by authors named Maulik.

TWEET:
<ID:int, Timestamp:time, Author:string,Text:string, Mentions:set(string)>

<389472, 1/1/19 12:34, Bob, hey, {}>

<123794, 1/1/19 12:32, Maria, lol, {Bob}>

<596208, 1/2/19 1:04, Yu, :-D, {}>

Find all the tweets by authors named Maulik.

```
SELECT * FROM TWEET WHERE Name is “Maulik”
```

TWEET:
<ID:int, Timestamp:time, Author:string,Text:string, Mentions:set(string)>

<389472, 1/1/19 12:34, Bob, hey, {}>

<123794, 1/1/19 12:32, Maria, lol, {Bob}>

<596208, 1/2/19 1:04, Yu, :-D, {}>

Find all the tweets by authors named Maulik.

```
SELECT * FROM TWEET WHERE Name is “Maulik”
```

∅

TWEET:
<ID:int, Timestamp:time, Author:string,Text:string, Mentions:set(string)>

<389472, 1/1/19 12:34, Bob, hey, {}>

<123794, 1/1/19 12:32, Maria, lol, {Bob}>

<596208, 1/2/19 1:04, Yu, :-D, {}>

Find all the tweets by authors named Maulik.

```
SELECT * FROM TWEET WHERE Name is “Maulik”
```

∅ ✔

TWEET:
<ID:int, Timestamp:time, Author:string,Text:string, Mentions:set(string)>

<389472, 1/1/19 12:34, Bob, hey, {}>

<123794, 1/1/19 12:32, Maria, lol, {Bob}>

<596208, 1/2/19 1:04, Yu, :-D, {}>

Find all the tweets by authors named Maulik.

```
SELECT * FROM TWEET WHERE Name is “Maulik”
```

∅ ✔

“Closed world assumption”

TWEET:
<ID:int, Timestamp:time, Author:string,Text:string, Mentions:set(string)>

<389472, 1/1/19 12:34, Bob, hey, {}>

<123794, 1/1/19 12:32, Maria, lol, {Bob}>

<596208, 1/2/19 1:04, Yu, :-D, {}>

Find all the tweets which weight less than 45lbs

TWEET:
<ID:int, Timestamp:time, Author:string,Text:string, Mentions:set(string)>

<389472, 1/1/19 12:34, Bob, hey, {}>

<123794, 1/1/19 12:32, Maria, lol, {Bob}>

<596208, 1/2/19 1:04, Yu, :-D, {}>

Find all the tweets which weight less than 45lbs

```
SELECT * FROM TWEET WHERE Weight < 45
```

TWEET:
<ID:int, Timestamp:time, Author:string,Text:string, Mentions:set(string)>

<389472, 1/1/19 12:34, Bob, hey, {}>

<123794, 1/1/19 12:32, Maria, lol, {Bob}>

<596208, 1/2/19 1:04, Yu, :-D, {}>

Find all the tweets which weight less than 45lbs

```
SELECT * FROM TWEET WHERE Weight < 45
```

????

TWEET:
<ID:int, Timestamp:time, Author:string,Text:string, Mentions:set(string)>

<389472, 1/1/19 12:34, Bob, hey, {}>

<123794, 1/1/19 12:32, Maria, lol, {Bob}>

<596208, 1/2/19 1:04, Yu, :-D, {}>

Relational Model

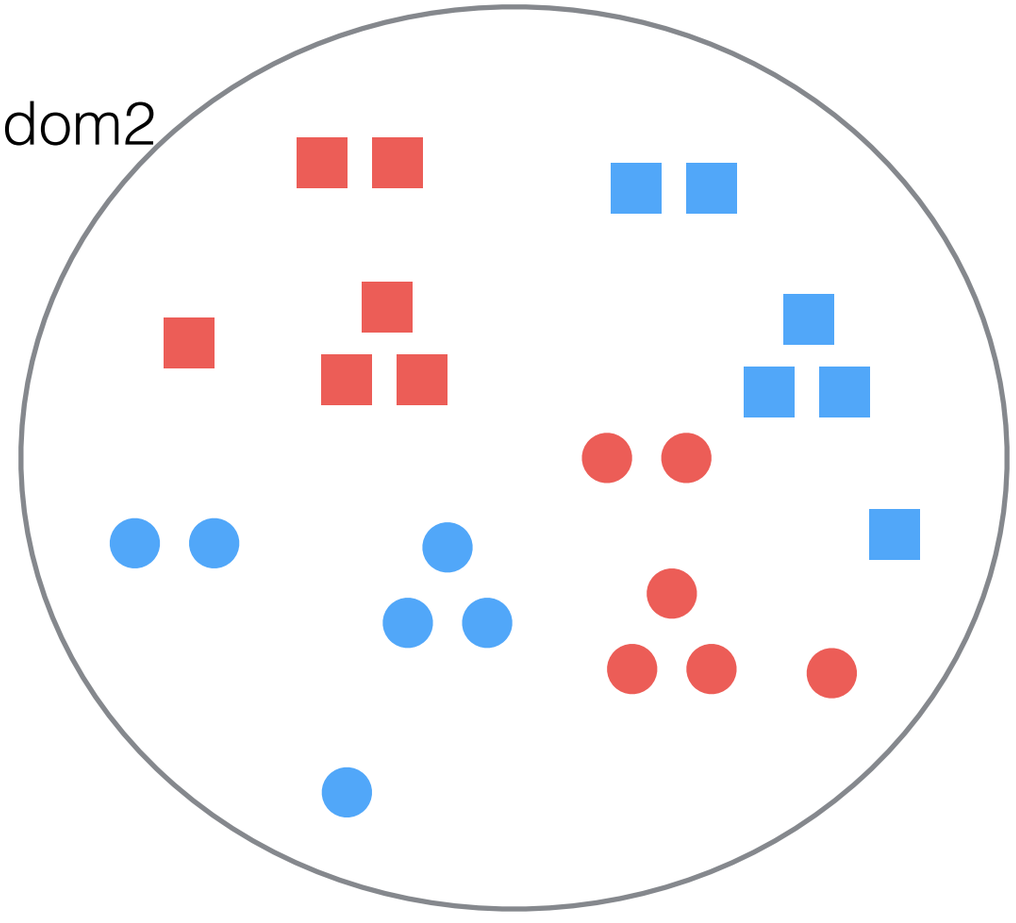
R: (A0:dom(A0), A1:dom(A1), … An:dom(An))

$r(R) \subseteq dom(A0) \times dom(A1) \times \ldots \times dom(An)$

Relational Model

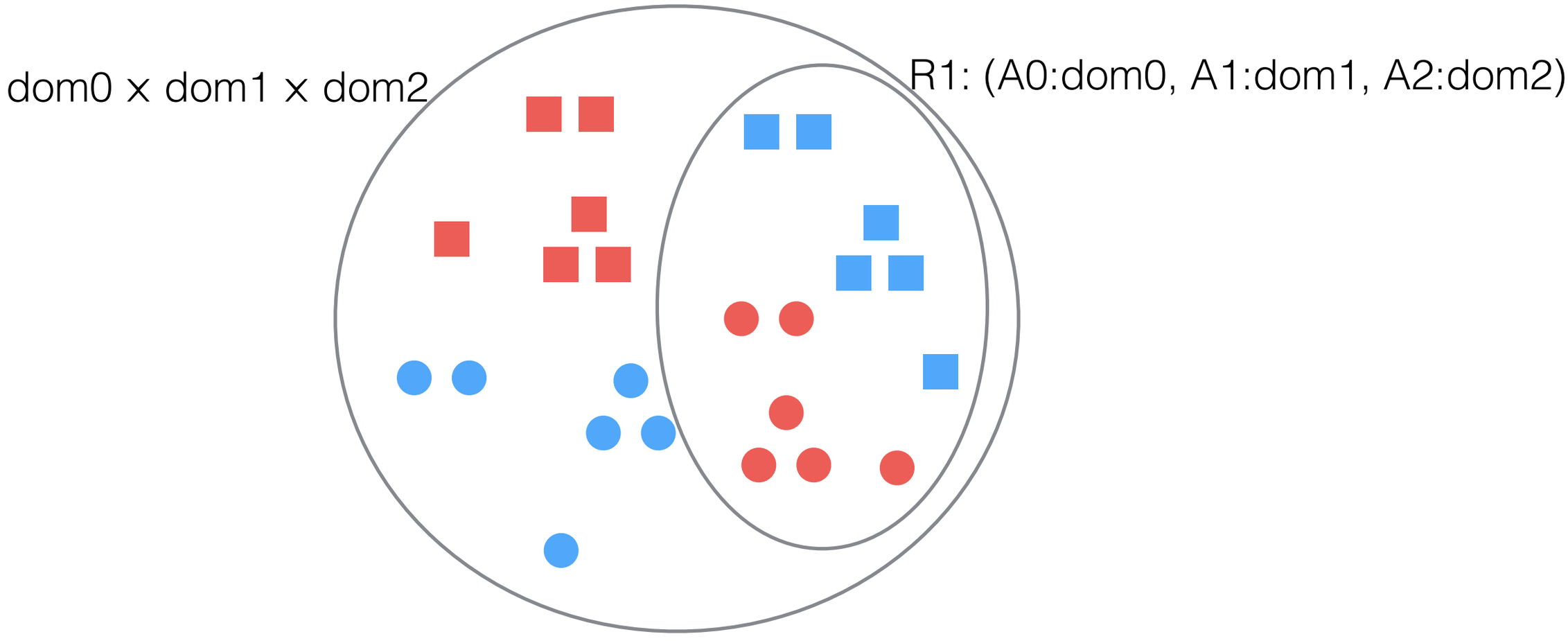
dom0 = {  ,  }
dom1 = {  ,  }
dom2 = {✔, ✔✔, ✔✔✔}

dom0 x dom1 x dom2



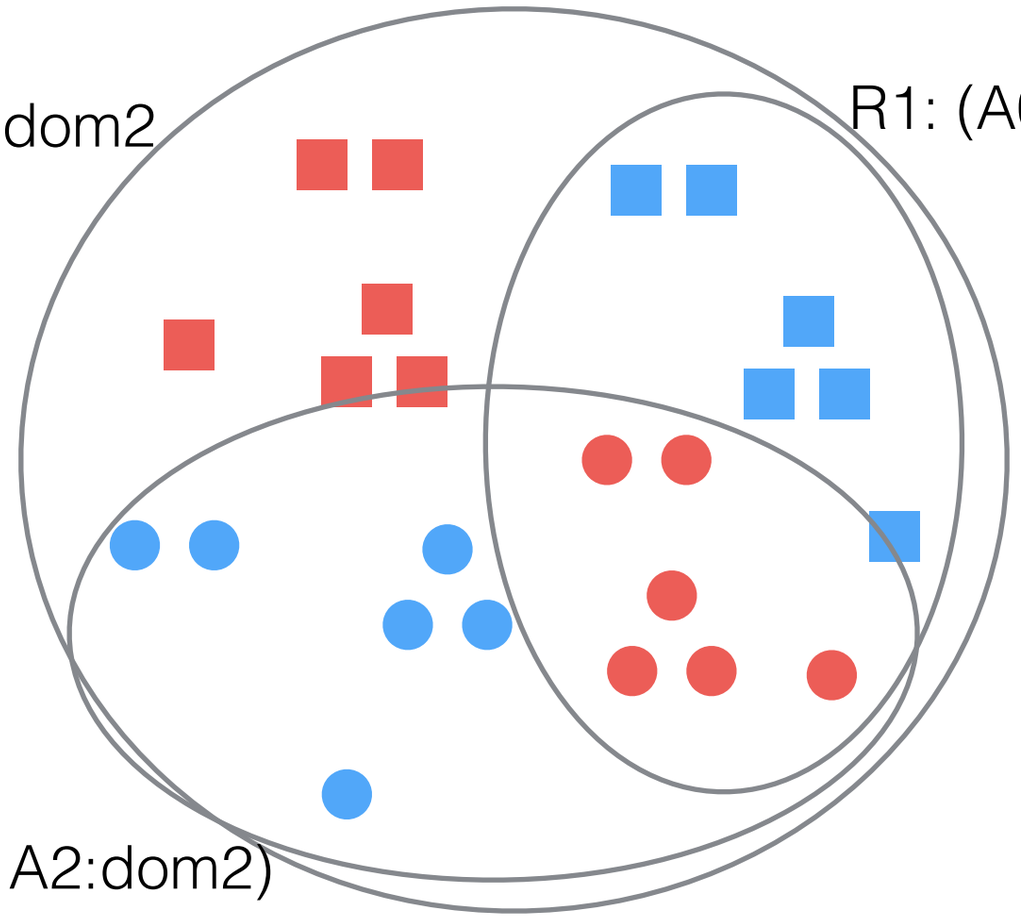
Relational Model

dom0 = { , }
dom1 = { , }
dom2 = {✔, ✔✔, ✔✔✔}



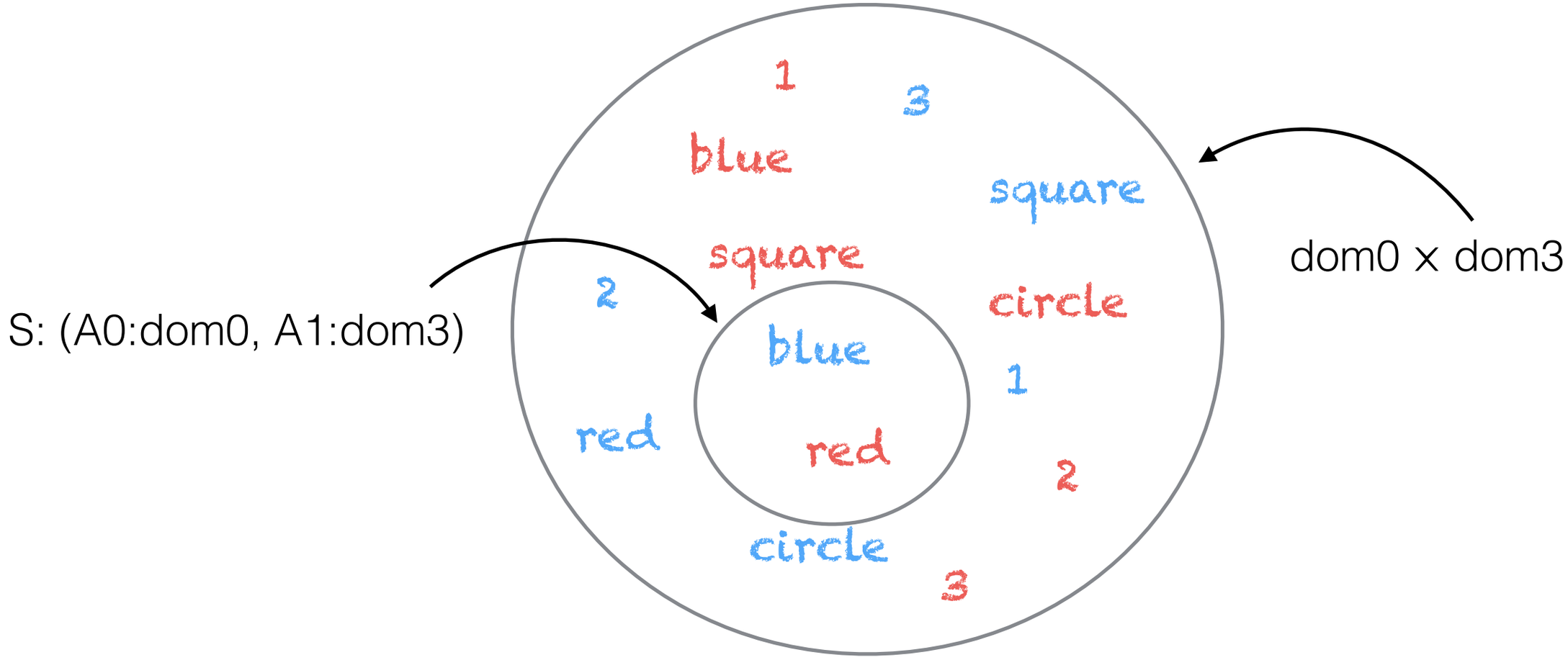
Relational Model

dom0 = { , }
dom1 = { , }
dom2 = {✔, ✔✔, ✔✔✔}



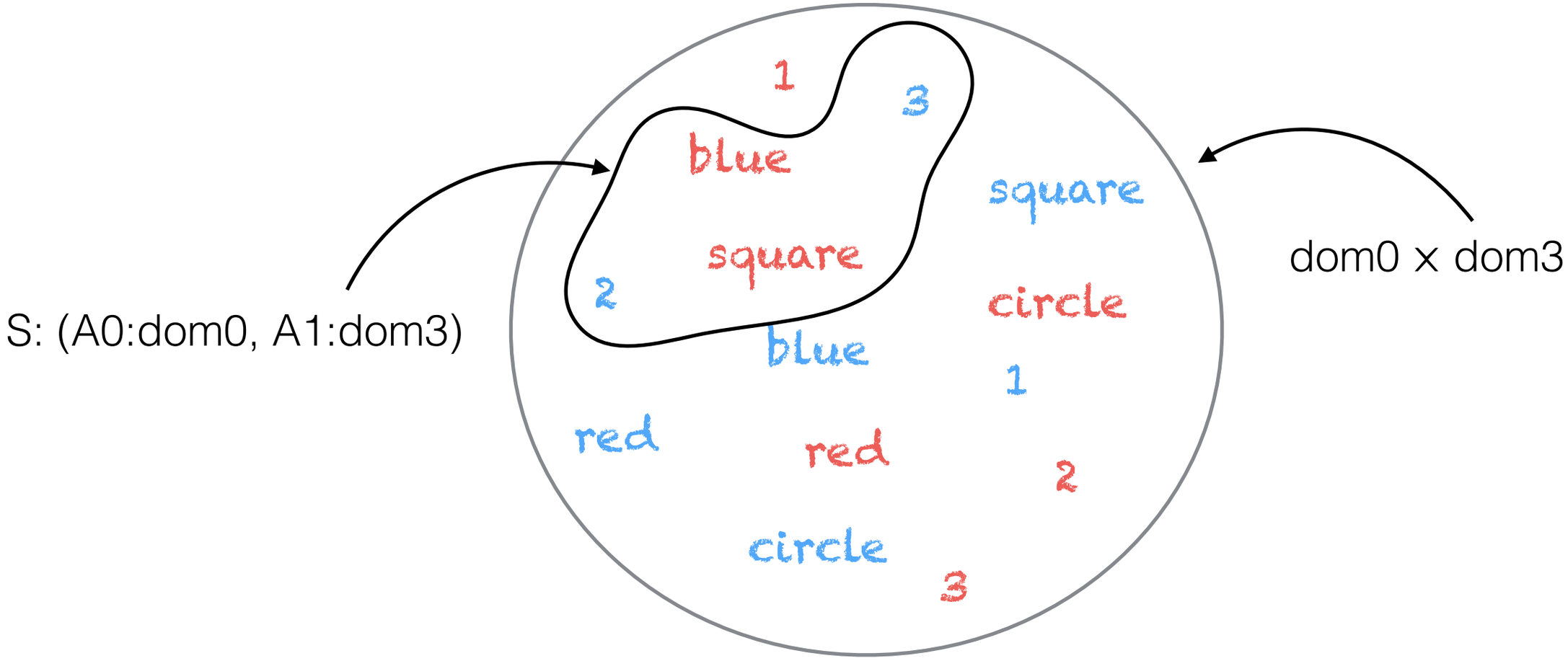
Relational Model

dom0 = {  ,  }
dom1 = {  ,  }
dom2 = {✔, ✔✔, ✔✔✔}
dom3 = {blue, red, circle, square, 1, 2, 3}



Relational Model

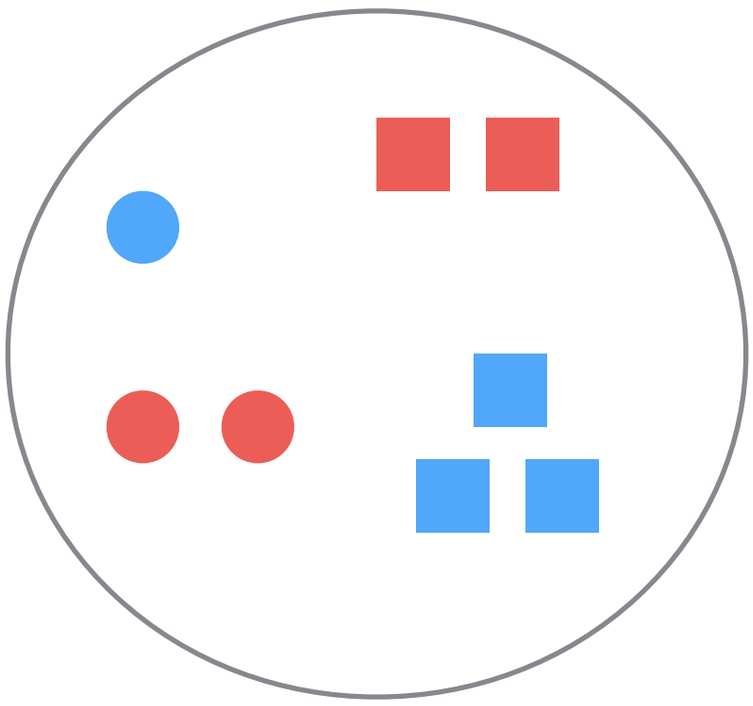
dom0 = {  ,  }
dom1 = {  ,  }
dom2 = {✔, ✔✔, ✔✔✔}
dom3 = {blue, red, circle, square, 1, 2, 3}



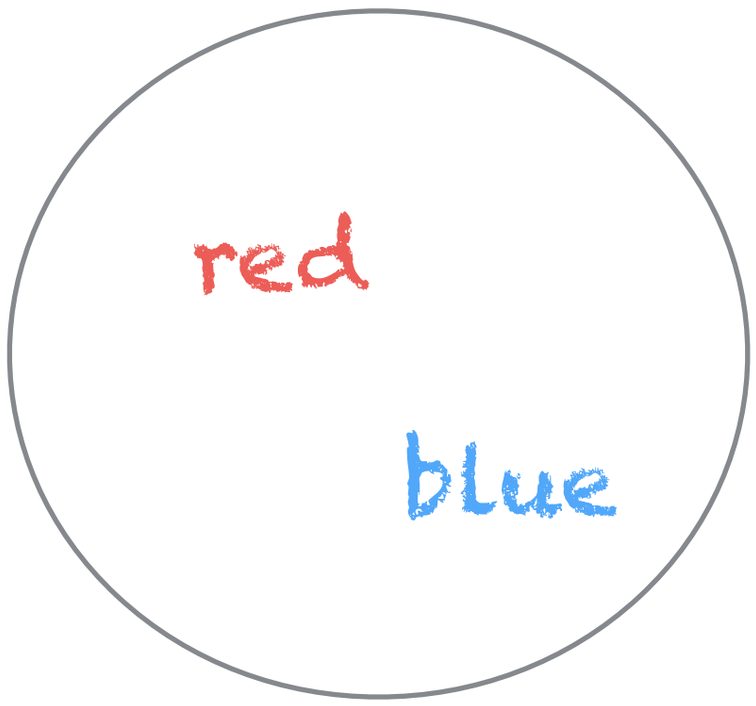
Relational Model

dom0 = {  ,  }
dom1 = {  ,  }
dom2 = {✔, ✔✔, ✔✔✔}
dom3 = {blue, red, circle, square, 1, 2, 3}

R: (A0:dom0, A1:dom1, A2:dom2)
S: (A0:dom0, A1:dom3)

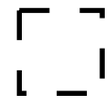


r(R)



s(S)

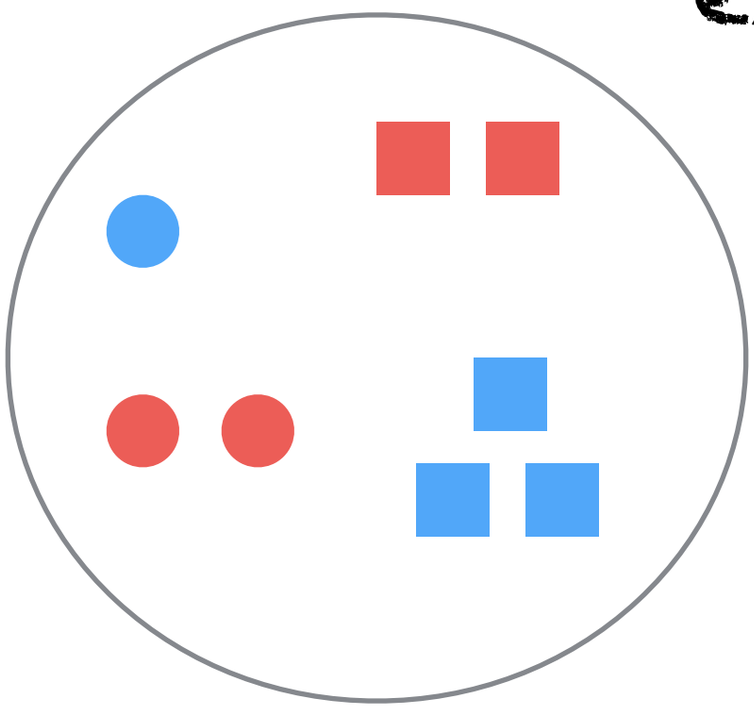
Relational Model

dom0 = {  ,  }
dom1 = {  ,  }
dom2 = {✔, ✔✔, ✔✔✔}
dom3 = {blue, red, circle, square, 1, 2, 3}

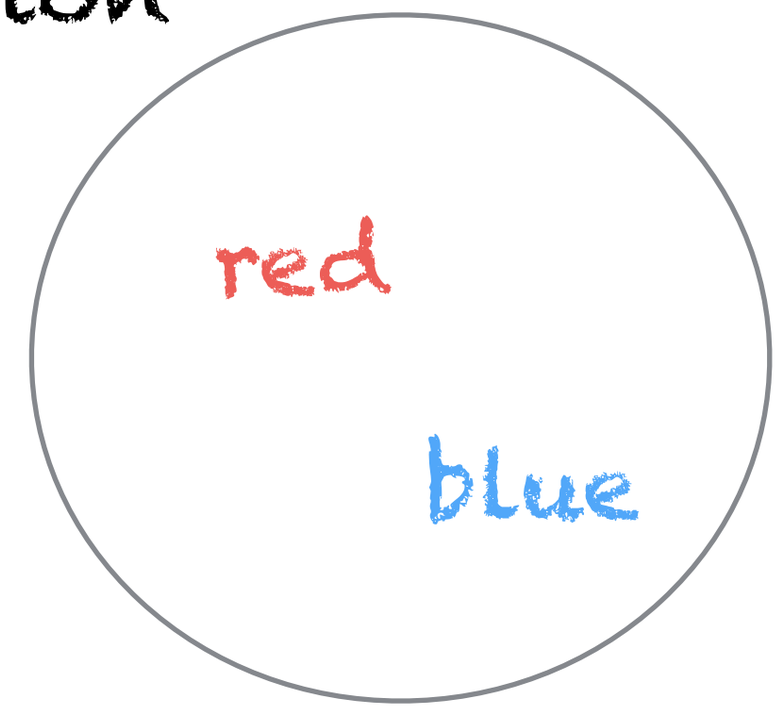
R: (A0:dom0, A1:dom1, A2:dom2)
S: (A0:dom0, A1:dom3)

intension

extension



r(R)

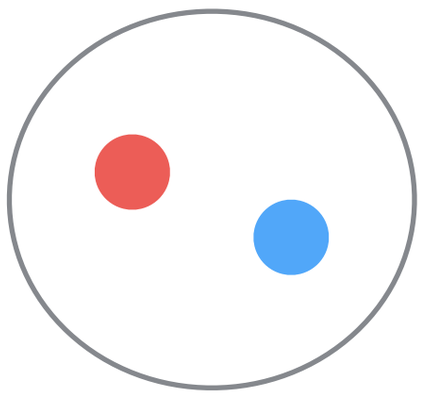


s(S)

dom0 = {  ,  }
dom1 = {  ,  }
dom2 = {✔, ✔✔, ✔✔✔}
dom3 = {blue, red, circle, square, 1, 2, 3}

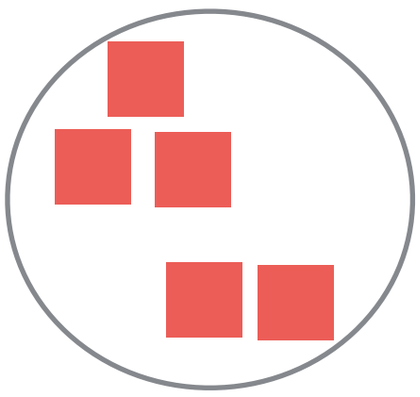
Clicker Question!
In which of the following intension/ extension pairs is NOT valid?

R: (A0:dom0, A1:dom1)



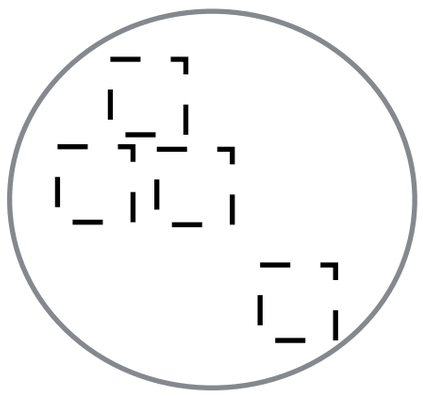
(a)

R: (A0:dom0, A1:dom1)



(b)

R: (A0:dom1, A1:dom2)

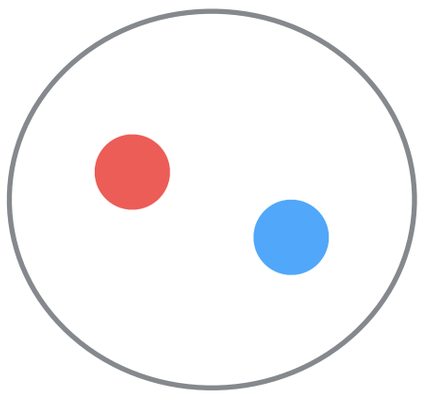


(c)

dom0 = {  ,  }
dom1 = {  ,  }
dom2 = {✔, ✔✔, ✔✔✔}
dom3 = {blue, red, circle, square, 1, 2, 3}

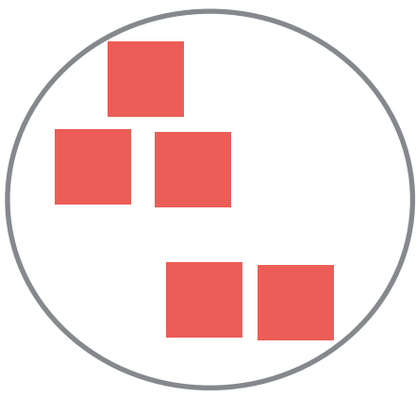
Clicker Question!
In which of the following intension/ extension pairs is NOT valid?

R: (A0:dom0, A1:dom1)



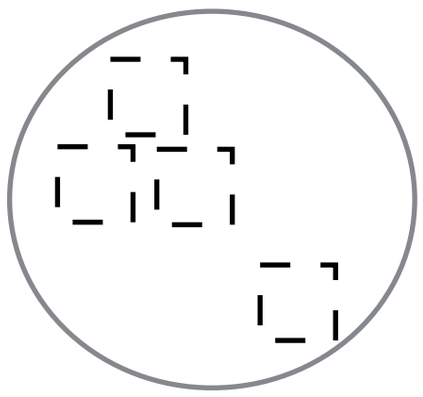
(a)

R: (A0:dom0, A1:dom1)



(b)

R: (A0:dom1, A1:dom2)

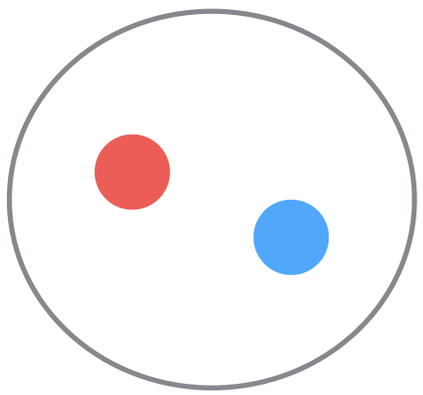


(c)

dom0 = {  ,  }
dom1 = {  ,  }
dom2 = {✔, ✔✔, ✔✔✔}
dom3 = {blue, red, circle, square, 1, 2, 3}

Clicker Question!
In which of the following intension/ extension pairs is NOT valid?

R: (A0:dom0, A1:dom1)

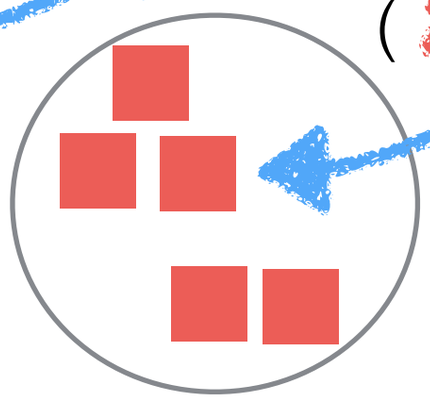


(a)

R: (A0:dom0, A1:dom1)

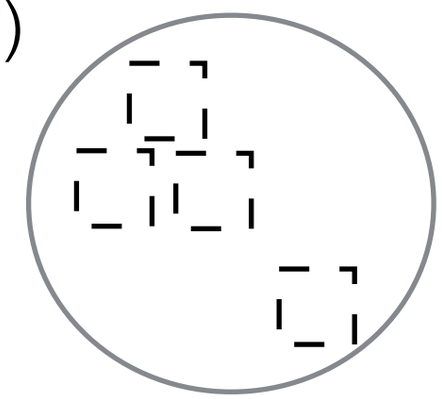
dom2 not part of relation schema

( ,  , ✔✔✔)



(b)

R: (A0:dom1, A1:dom2)

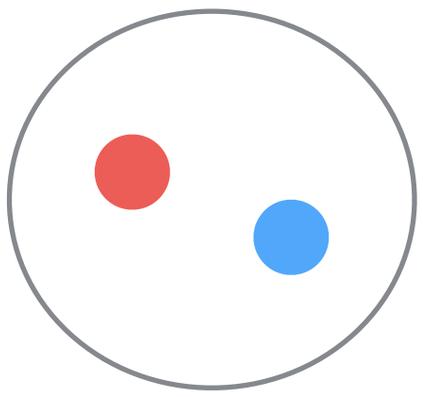


(c)

dom0 = {  ,  }
dom1 = {  ,  }
dom2 = {✔, ✔✔, ✔✔✔}
dom3 = {blue, red, circle, square, 1, 2, 3}

Clicker Question!
In which of the following intension/ extension pairs is NOT valid?

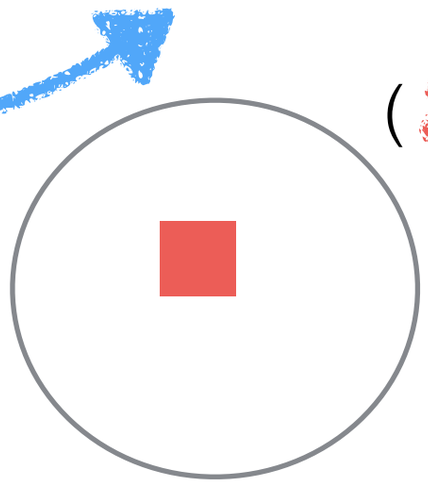
R: (A0:dom0, A1:dom1)



(a)

R: (A0:dom0, A1:dom1)

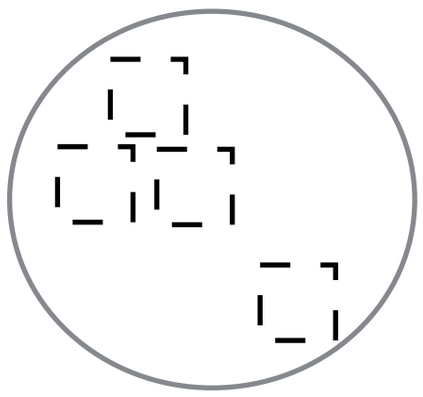
dom2 not part of relation schema



( , )

(b)

R: (A0:dom1, A1:dom2)



(c)

Relational Algebra

- σ selection
- π project
- ∪ union
- – minus
- × cross
- ρ rename

Relational Algebra

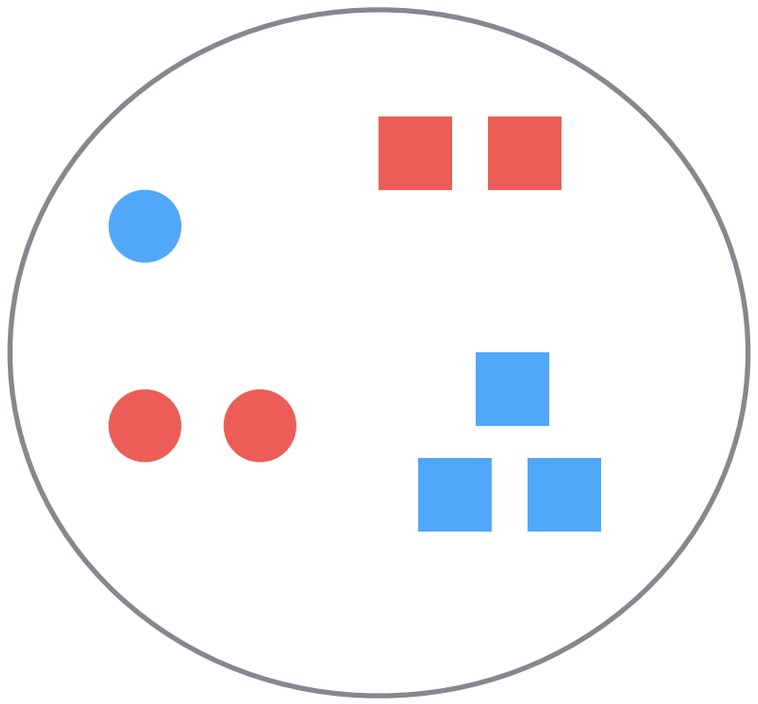
- σ selection
- π project
- ∪ union
- – minus
- × cross
- ρ rename

Relational Algebra
Select

dom0 = {  }
dom1 = {  }
dom2 = {✔, ✔✔, ✔✔✔}
dom3 = {blue, red, circle, square, 1, 2, 3}

R: (A0:dom0, A1:dom1, A2:dom2)
S: (A0:dom0, A1:dom3)

R

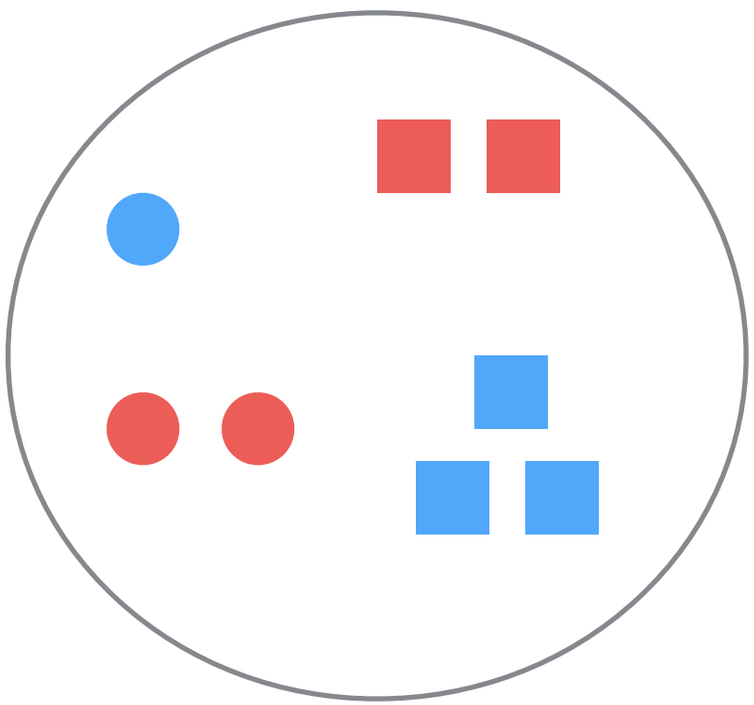


Relational Algebra
Select

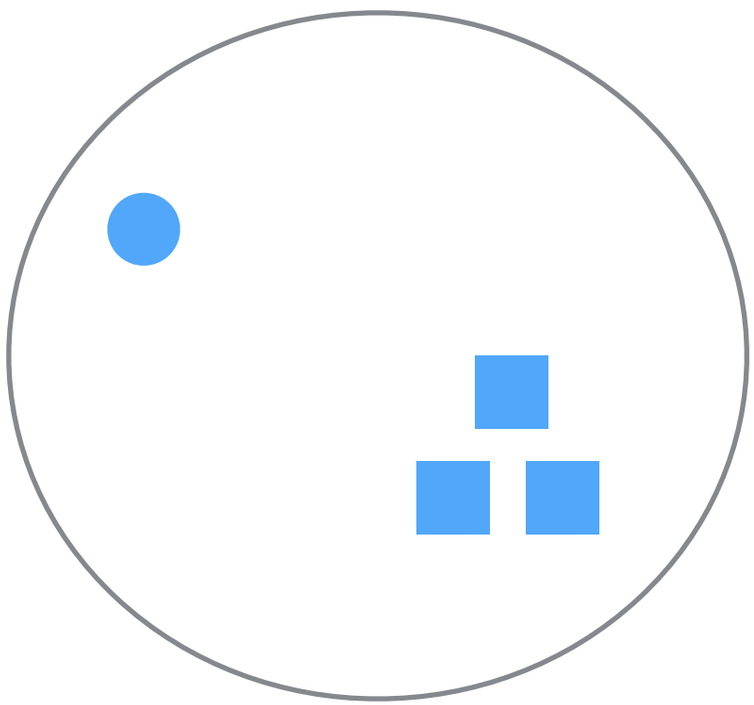
dom0 = { 𝔞 , 𝔞 }
dom1 = { ◌ , ⌐¬ }
dom2 = {✔, ✔✔, ✔✔✔}
dom3 = {blue, red, circle, square, 1, 2, 3}

R: (A0:dom0, A1:dom1, A2:dom2)
S: (A0:dom0, A1:dom3)

R



$\sigma_{A0=\mathfrak{a}}(R)$

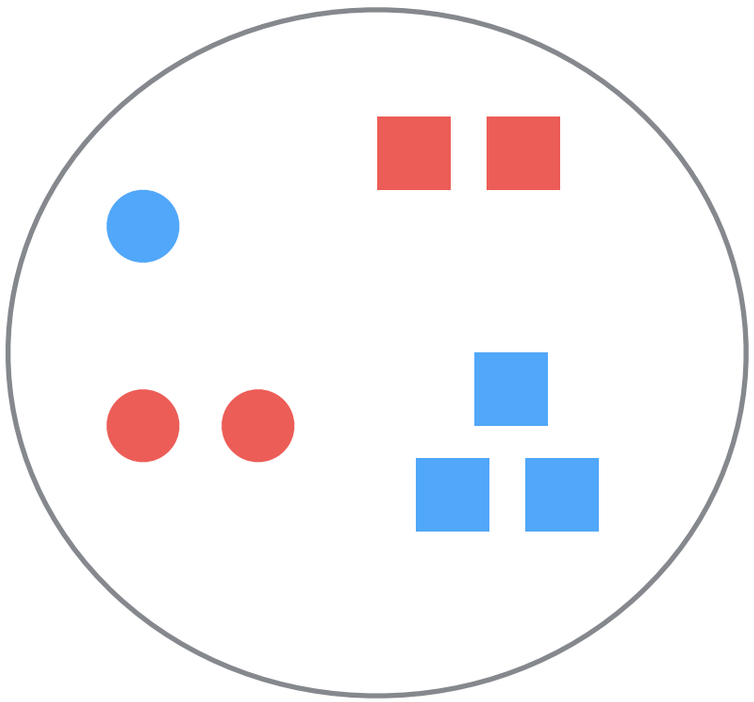


Relational Algebra
Select

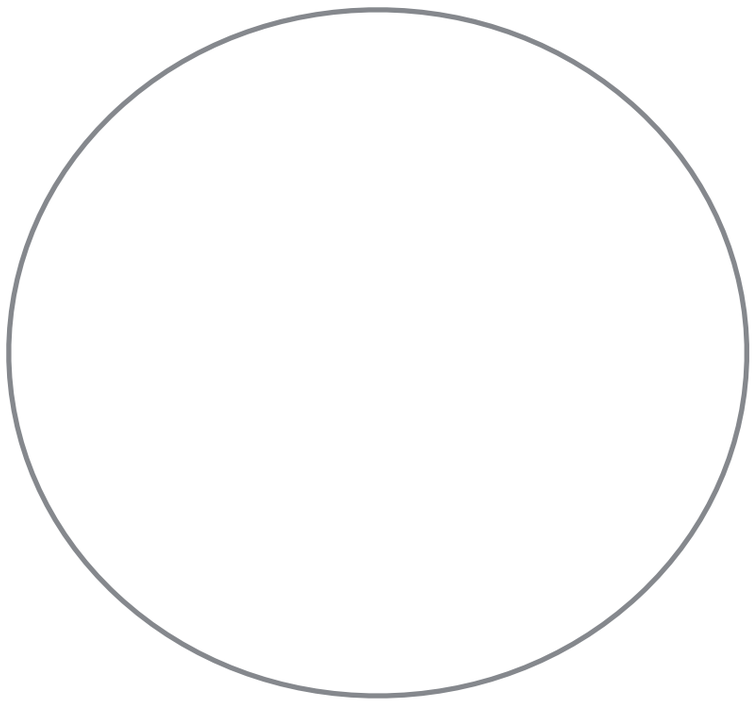
dom0 = { 🟦 , 🟥 }
dom1 = { ◌ , ⌐¬ }
dom2 = {✔, ✔✔, ✔✔✔}
dom3 = {blue, red, circle, square, 1, 2, 3}

R: (A0:dom0, A1:dom1, A2:dom2)
S: (A0:dom0, A1:dom3)

R



$\sigma_{A1=◌ \text{ or } A2=✔✔}(R)$

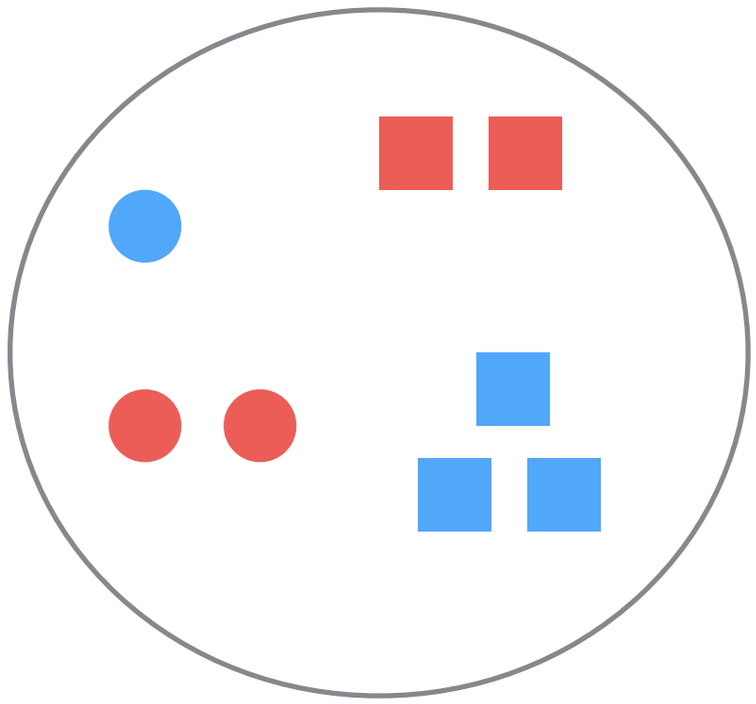


Relational Algebra
Select

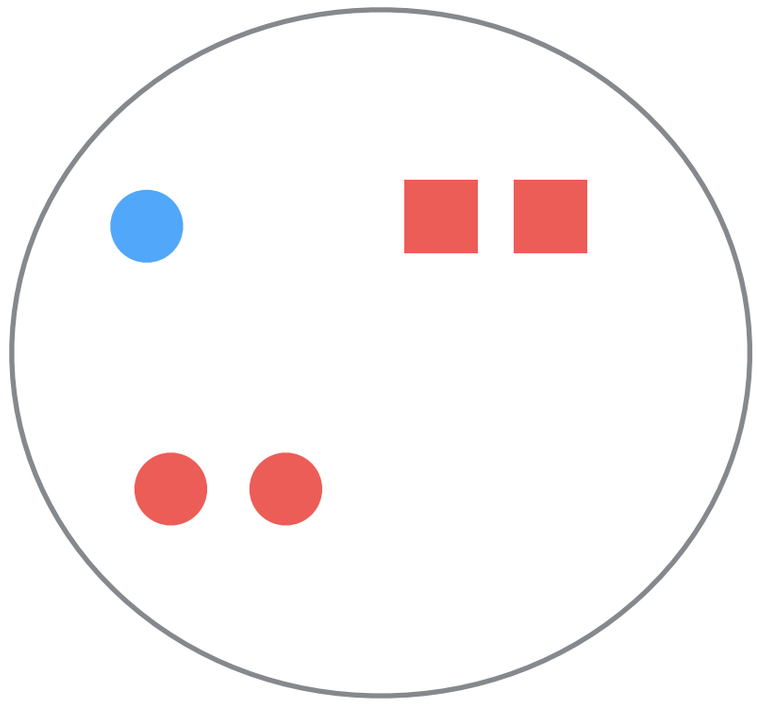
dom0 = {  ,  }
dom1 = {  ,  }
dom2 = {✔, ✔✔, ✔✔✔}
dom3 = {blue, red, circle, square, 1, 2, 3}

R: (A0:dom0, A1:dom1, A2:dom2)
S: (A0:dom0, A1:dom3)

R



$\sigma_{A1=\text{◌}\ \text{or}\ A2=✔✔}(R)$



Relational Algebra

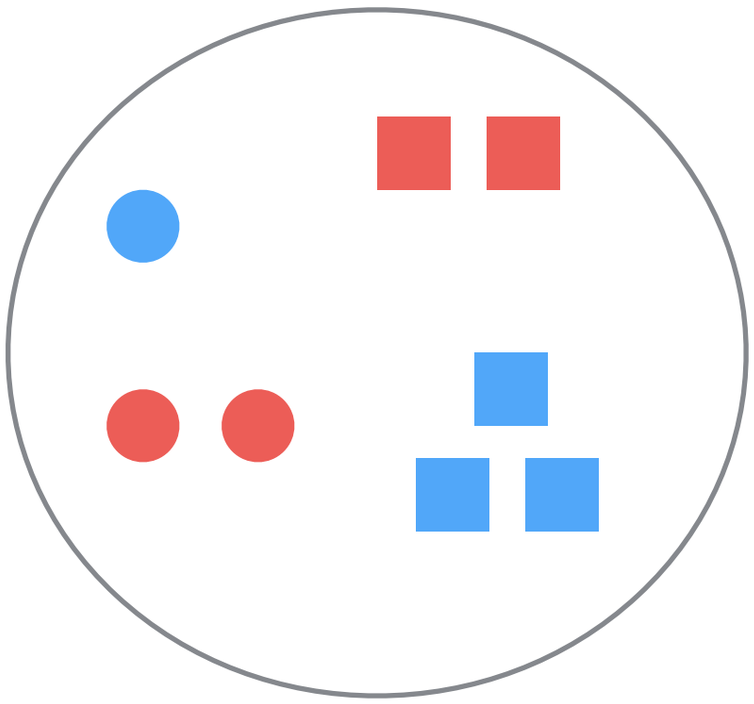
- σ selection
- π project
- ∪ union
- – minus
- × cross
- ρ rename

Relational Algebra
Project

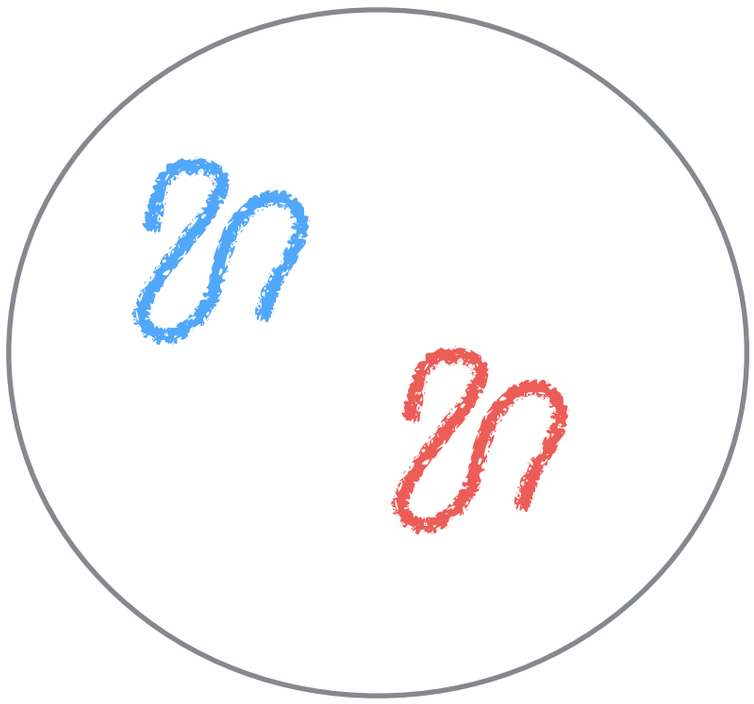
dom0 = {  ,  }
dom1 = { ◯ , ⌐¬ }
dom2 = {✔, ✔✔, ✔✔✔}
dom3 = {blue, red, circle, square, 1, 2, 3}

R: (A0:dom0, A1:dom1, A2:dom2)
S: (A0:dom0, A1:dom3)

R



$\pi_{A0}(R)$

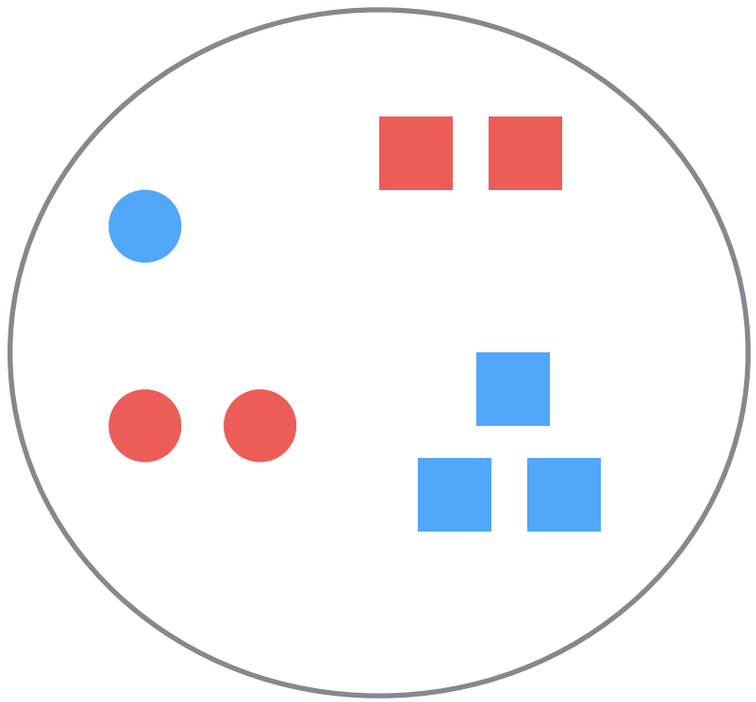


Relational Algebra
Project

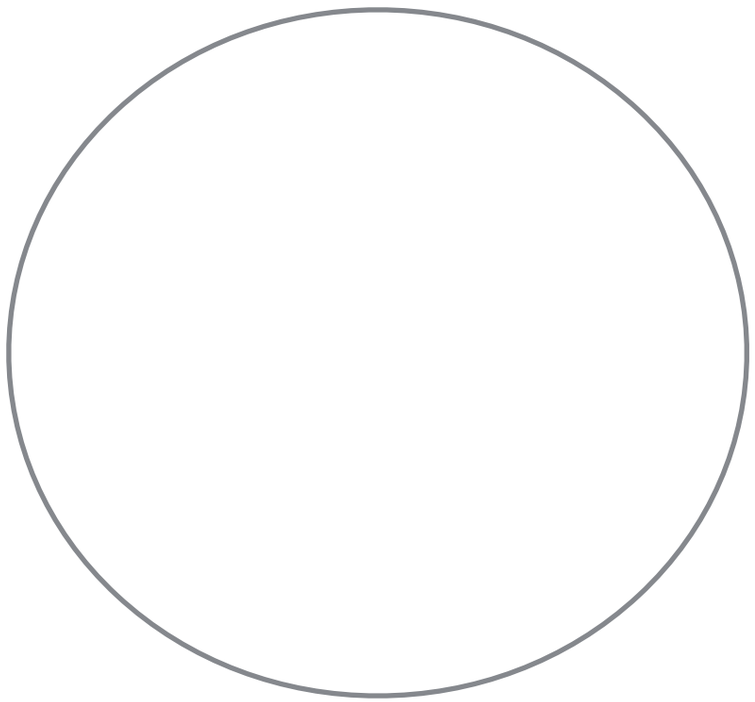
dom0 = {  ,  }
dom1 = {  ,  }
dom2 = {✔, ✔✔, ✔✔✔}
dom3 = {blue, red, circle, square, 1, 2, 3}

R: (A0:dom0, A1:dom1, A2:dom2)
S: (A0:dom0, A1:dom3)

R



$\pi_{A0,A3}(R)$

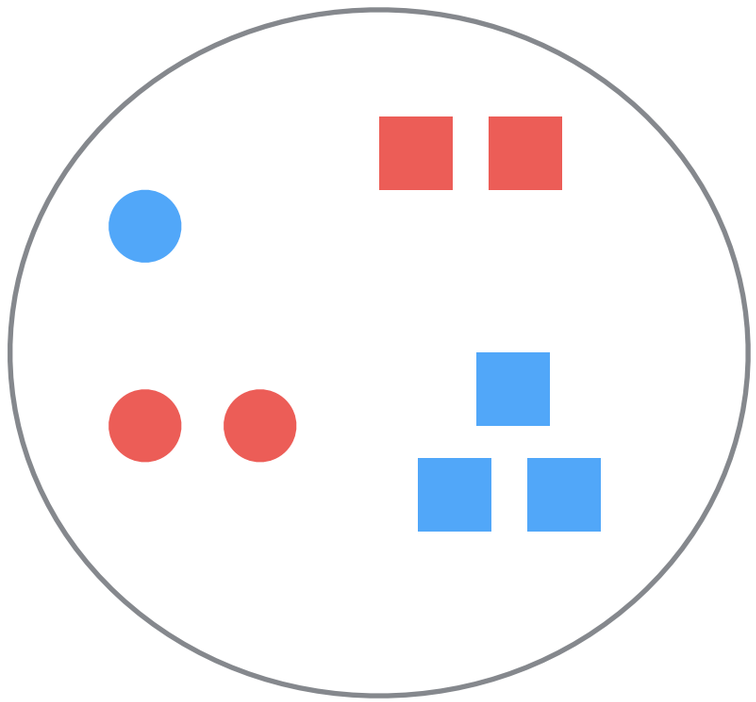


Relational Algebra
Project

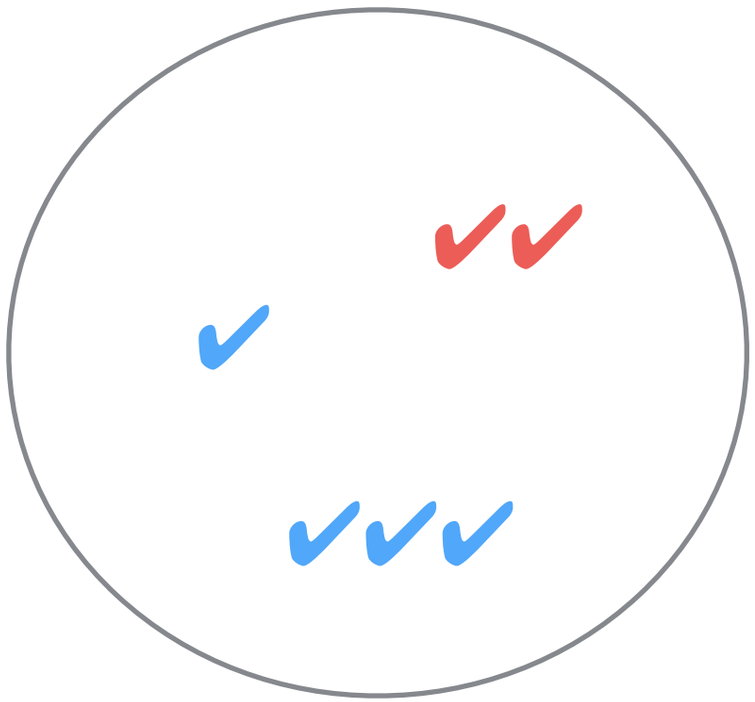
dom0 = {  ,  }
dom1 = {  ,  }
dom2 = {✔, ✔✔, ✔✔✔}
dom3 = {blue, red, circle, square, 1, 2, 3}

R: (A0:dom0, A1:dom1, A2:dom2)
S: (A0:dom0, A1:dom3)

R



$\pi_{A0,A3}(R)$



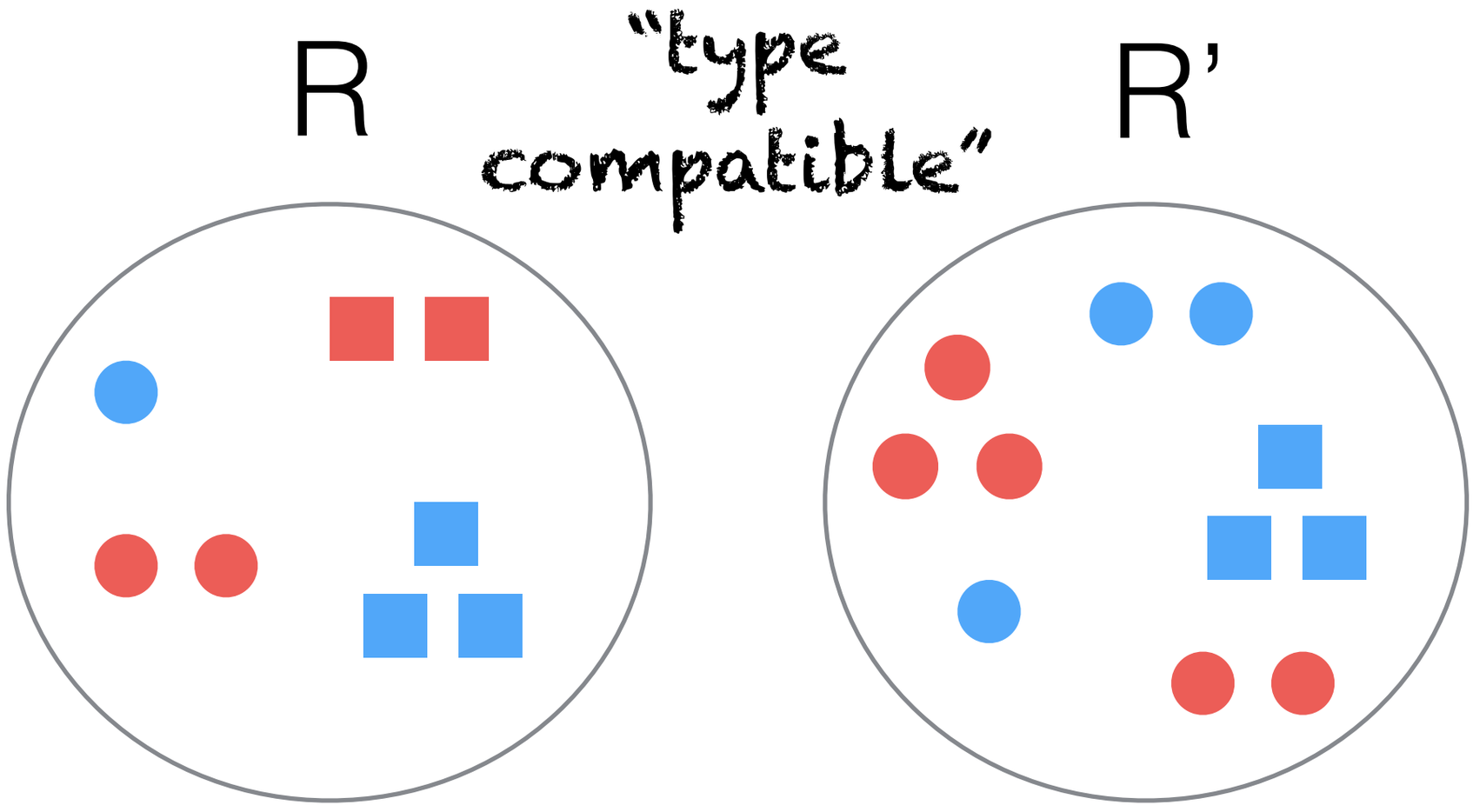
Relational Algebra

- σ selection
- π project
- ∪ union
- – minus
- × cross
- ρ rename

Relational Algebra
Union

dom0 = {  ,  }
dom1 = {  ,  }
dom2 = {✔, ✔✔, ✔✔✔}
dom3 = {blue, red, circle, square, 1, 2, 3}

R: (A0:dom0, A1:dom1, A2:dom2)
S: (A0:dom0, A1:dom3)



Relational Algebra
Union

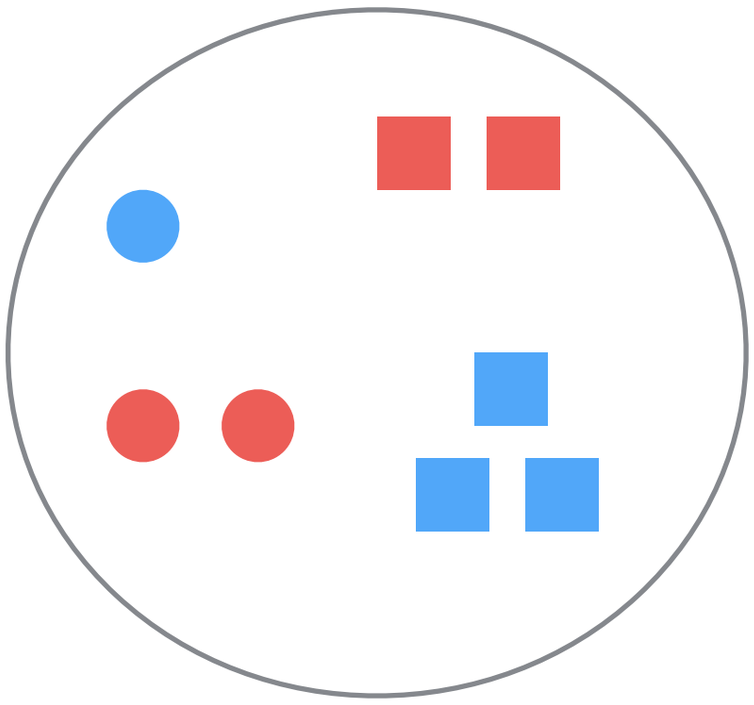
dom0 = {  ,  }
dom1 = {  ,  }
dom2 = {✔, ✔✔, ✔✔✔}
dom3 = {blue, red, circle, square, 1, 2, 3}

R: (A0:dom0, A1:dom1, A2:dom2)
S: (A0:dom0, A1:dom3)

R

NOT "type compatible"

S

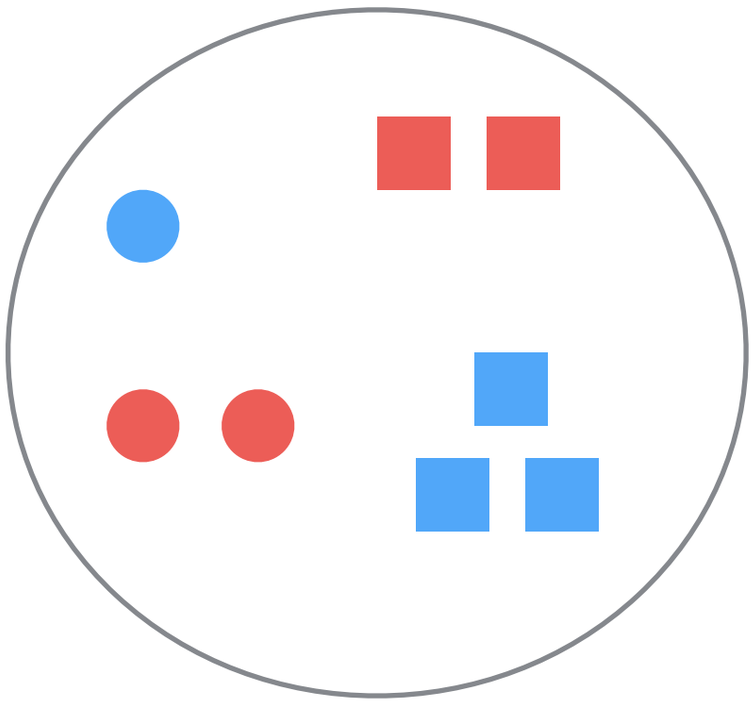


Relational Algebra
Union

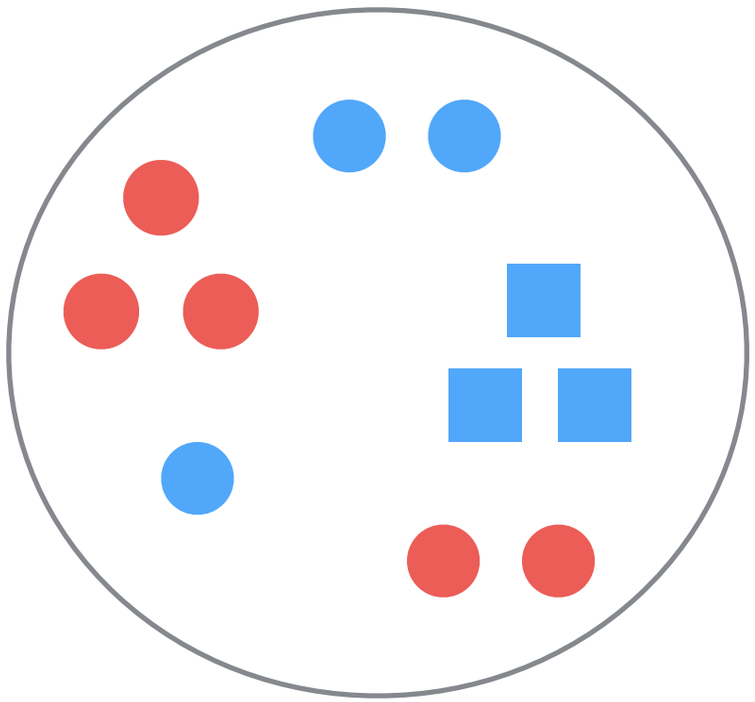
dom0 = {  ,  }
dom1 = {  ,  }
dom2 = {✔, ✔✔, ✔✔✔}
dom3 = {blue, red, circle, square, 1, 2, 3}

R: (A0:dom0, A1:dom1, A2:dom2)
S: (A0:dom0, A1:dom3)

R



R’

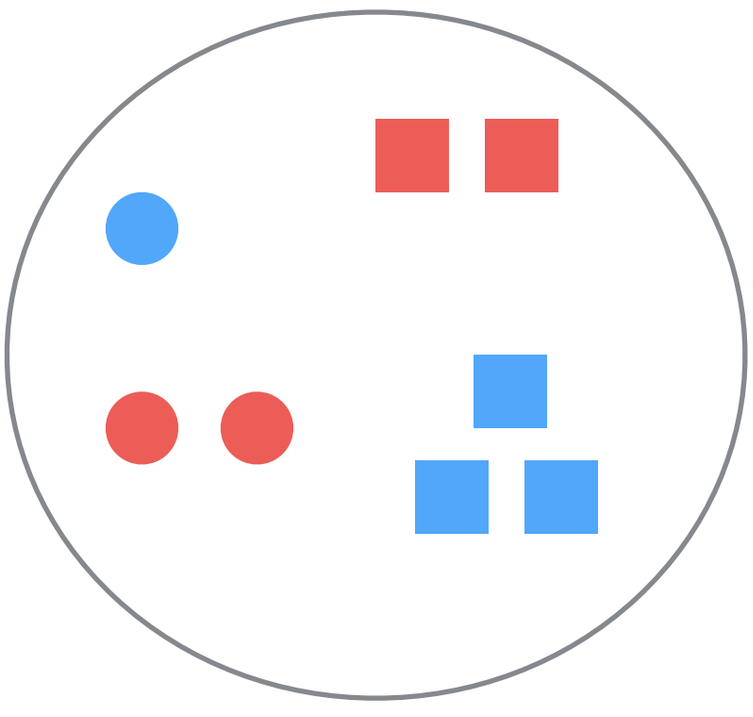


Relational Algebra
Union

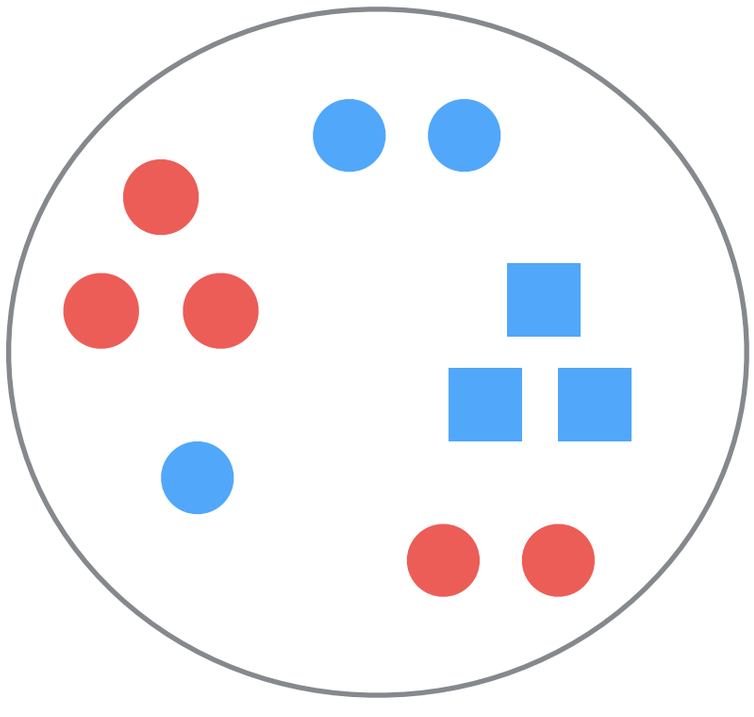
dom0 = {  ,  }
dom1 = {  ,  }
dom2 = {✔, ✔✔, ✔✔✔}
dom3 = {blue, red, circle, square, 1, 2, 3}

R: (A0:dom0, A1:dom1, A2:dom2)
S: (A0:dom0, A1:dom3)

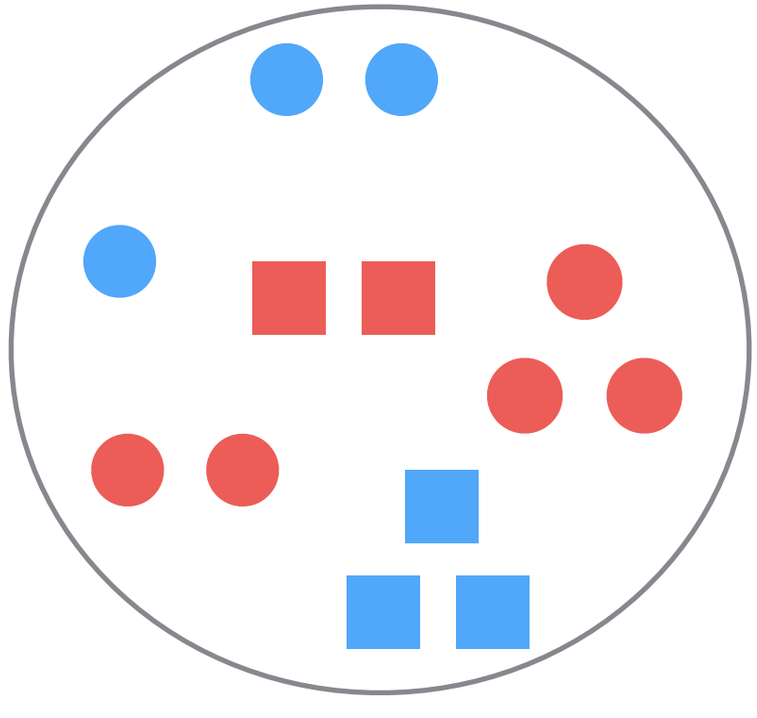
R



R'



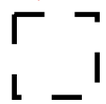
R ∪ R'



Relational Algebra

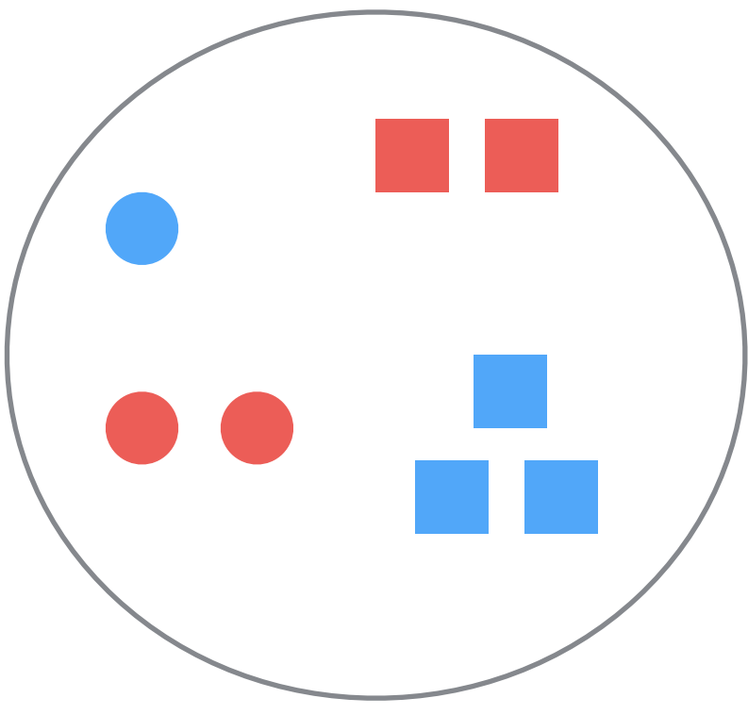
- σ selection
- π project
- ∪ union
- – minus
- × cross
- ρ rename

Relational Algebra
Minus

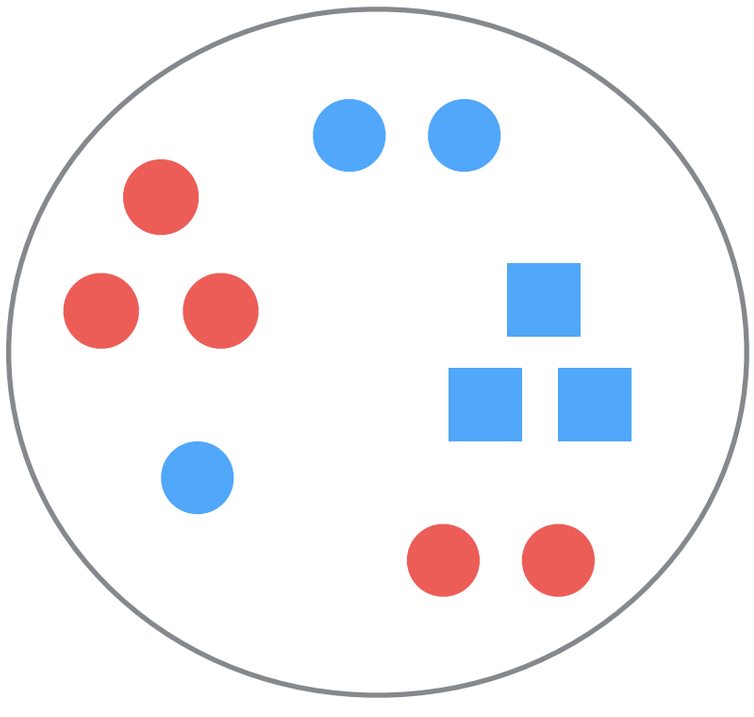
dom0 = {  ,  }
dom1 = {  ,  }
dom2 = {✔, ✔✔, ✔✔✔}
dom3 = {blue, red, circle, square, 1, 2, 3}

R: (A0:dom0, A1:dom1, A2:dom2)
S: (A0:dom0, A1:dom3)

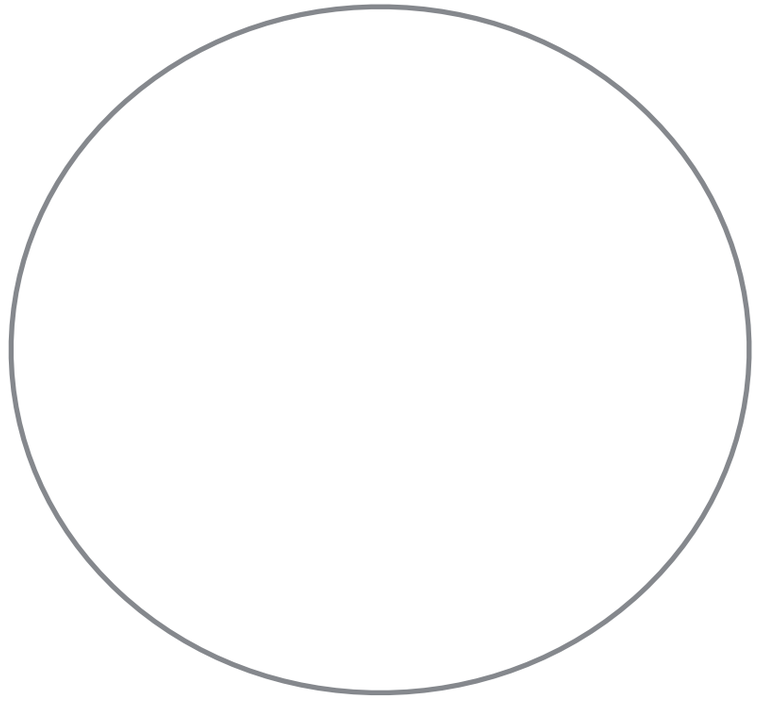
R



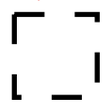
R’



R - R’

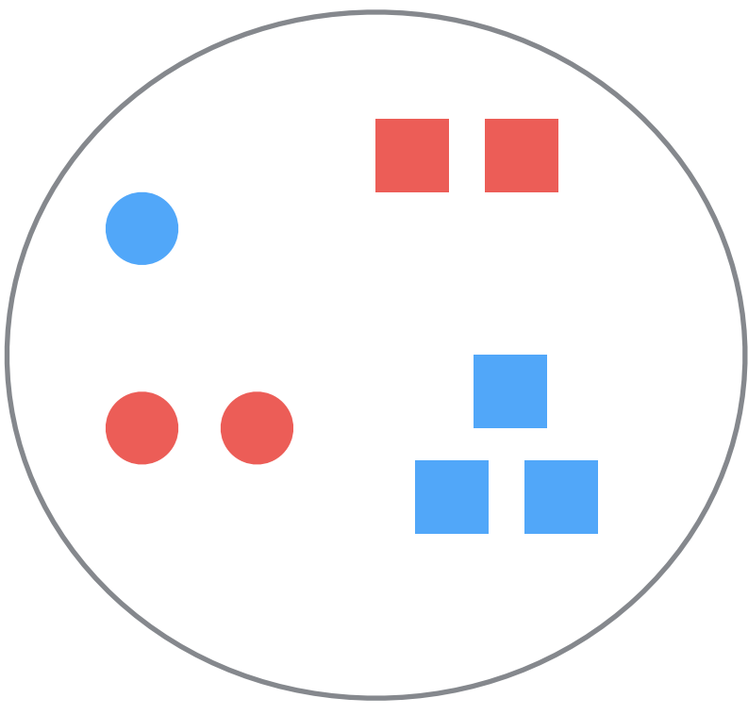


Relational Algebra
Minus

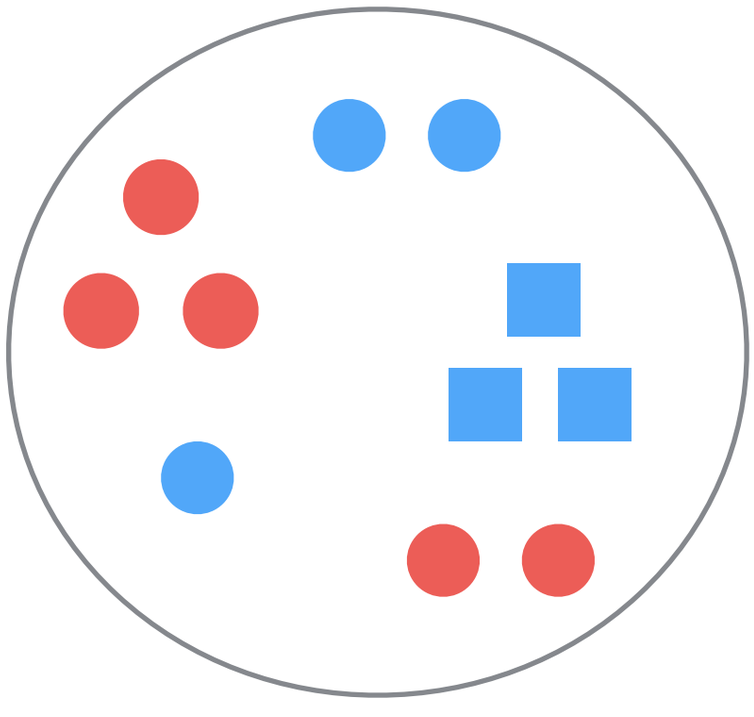
dom0 = {  ,  }
dom1 = {  ,  }
dom2 = {✔, ✔✔, ✔✔✔}
dom3 = {blue, red, circle, square, 1, 2, 3}

R: (A0:dom0, A1:dom1, A2:dom2)
S: (A0:dom0, A1:dom3)

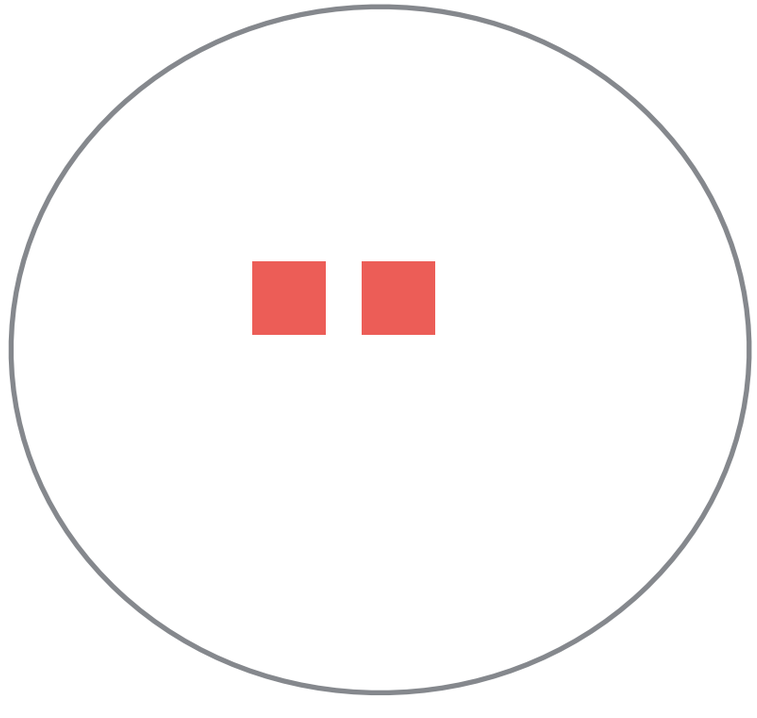
R



R’



R - R’

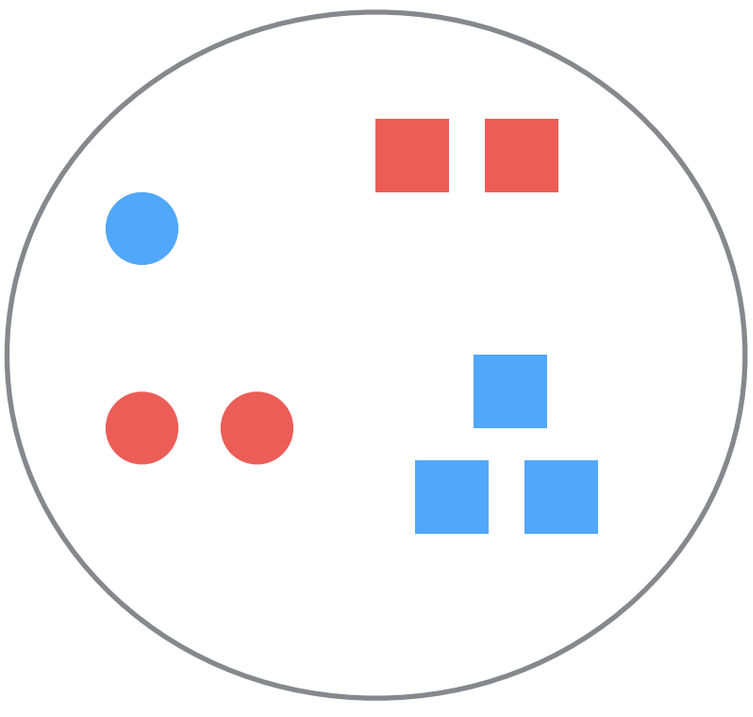


Relational Algebra
Minus

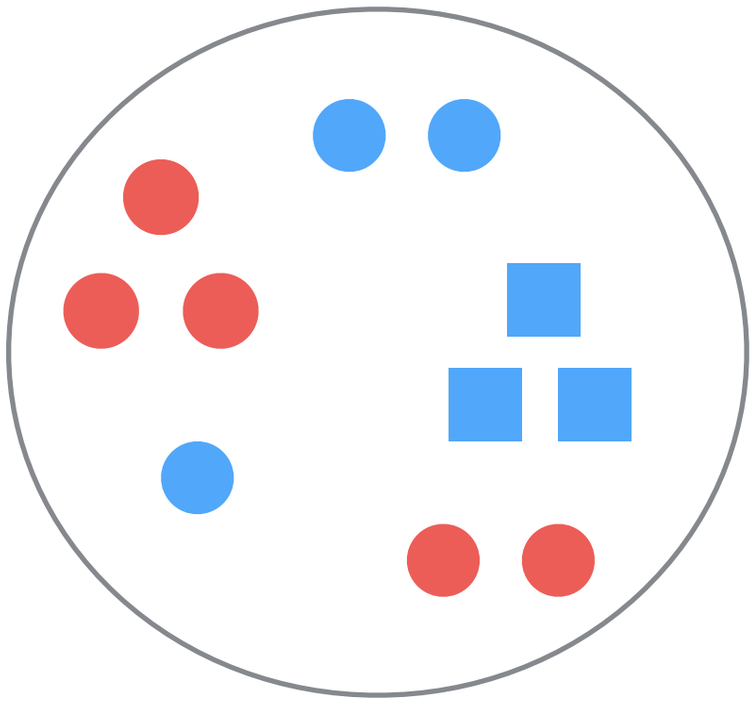
dom0 = {  ,  }
dom1 = {  ,  }
dom2 = {✔, ✔✔, ✔✔✔}
dom3 = {blue, red, circle, square, 1, 2, 3}

R: (A0:dom0, A1:dom1, A2:dom2)
S: (A0:dom0, A1:dom3)

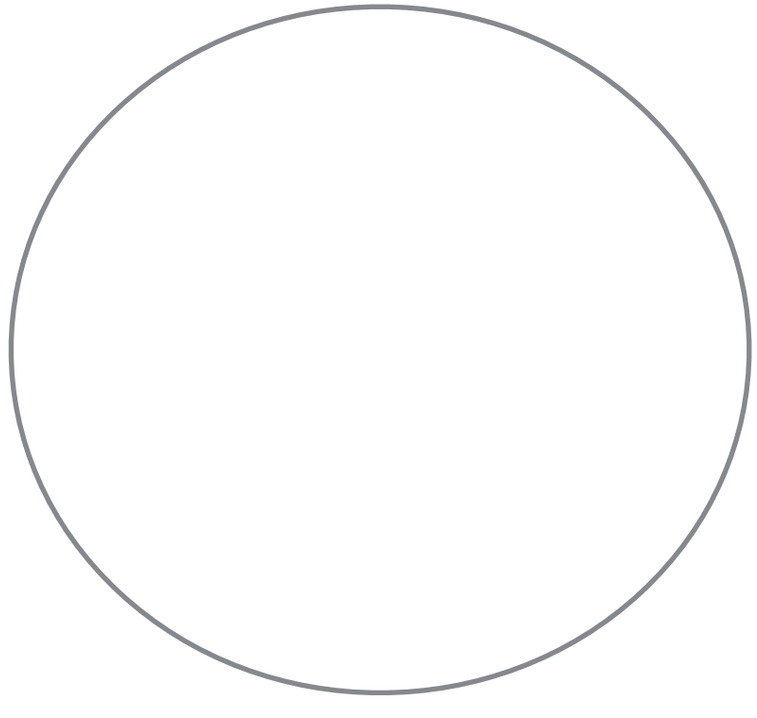
R



R’



R’ – R

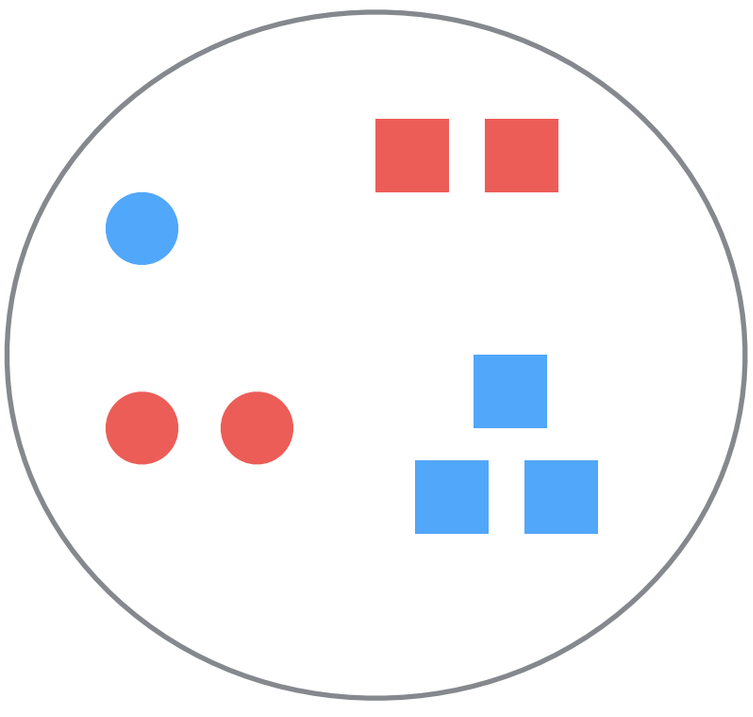


Relational Algebra
Minus

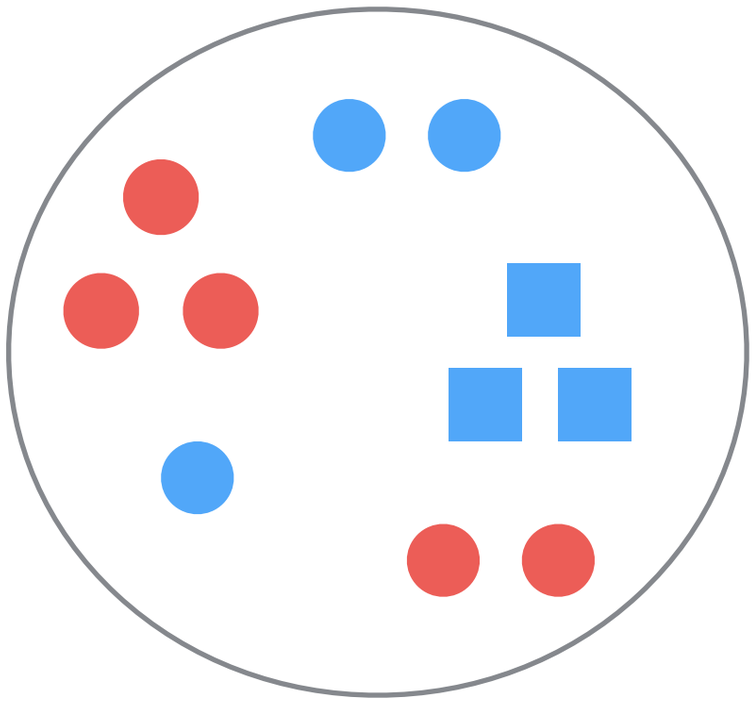
dom0 = {  ,  }
dom1 = {  ,  }
dom2 = {✔, ✔✔, ✔✔✔}
dom3 = {blue, red, circle, square, 1, 2, 3}

R: (A0:dom0, A1:dom1, A2:dom2)
S: (A0:dom0, A1:dom3)

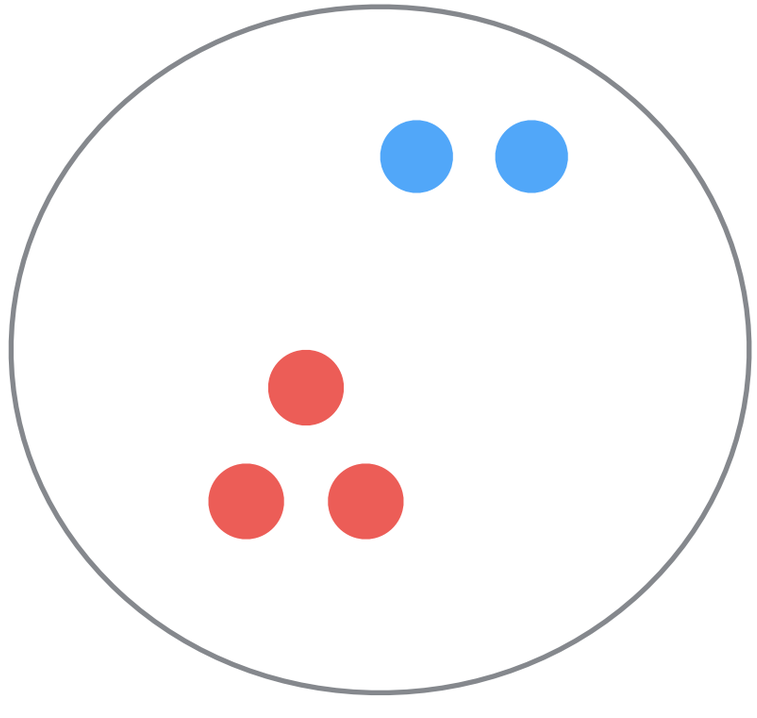
R



R’



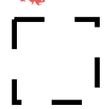
R’ – R



Relational Algebra

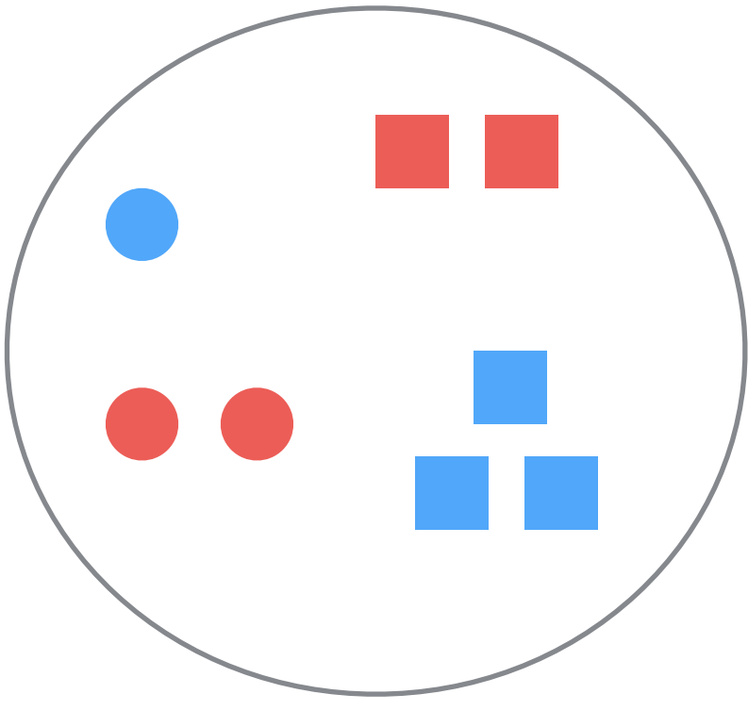
- σ selection
- π project
- ∪ union
- – minus
- × cross
- ρ rename

Relational Algebra
Cartesian Product

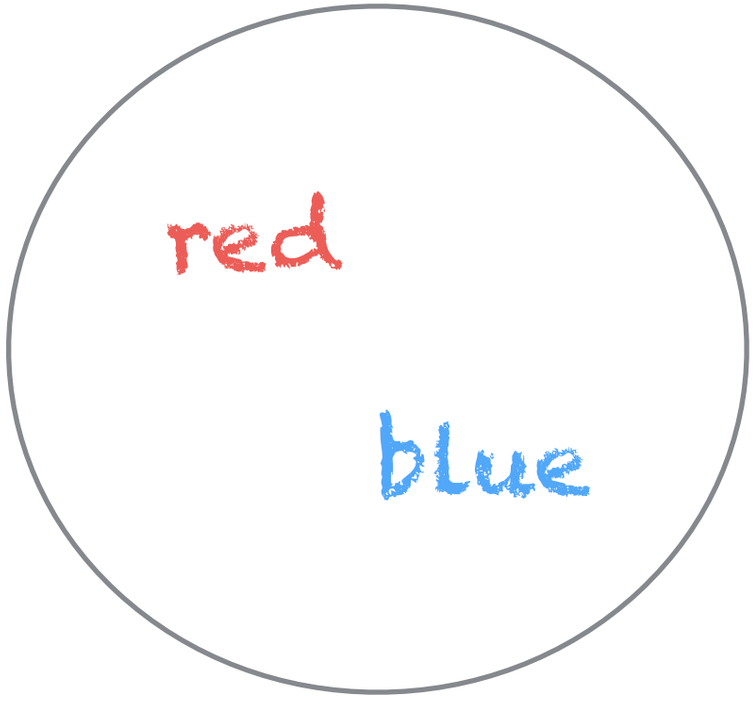
dom0 = {  ,  }
dom1 = {  ,  }
dom2 = {✔, ✔✔, ✔✔✔}
dom3 = {blue, red, circle, square, 1, 2, 3}

R: (A0:dom0, A1:dom1, A2:dom2)
S: (A0:dom0, A1:dom3)

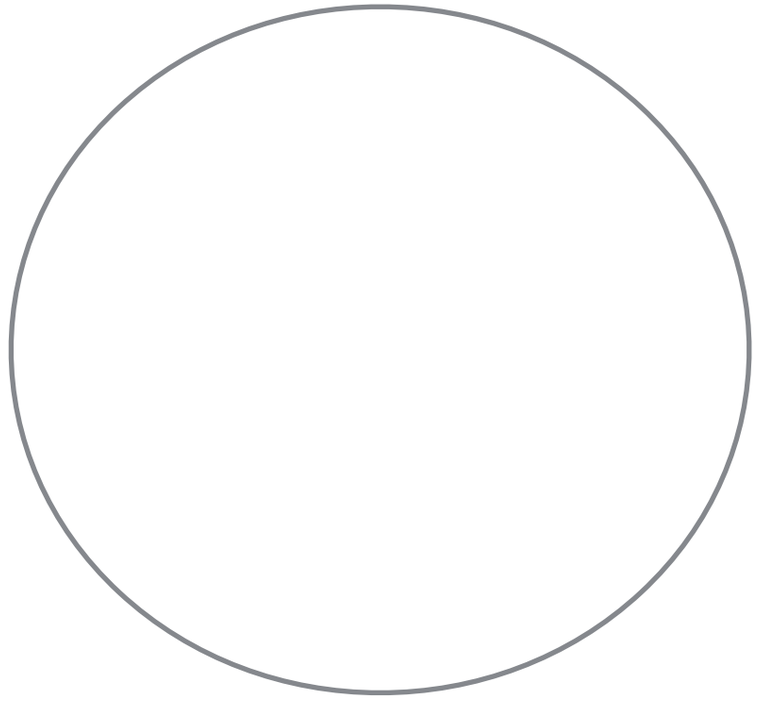
R



S



R × S

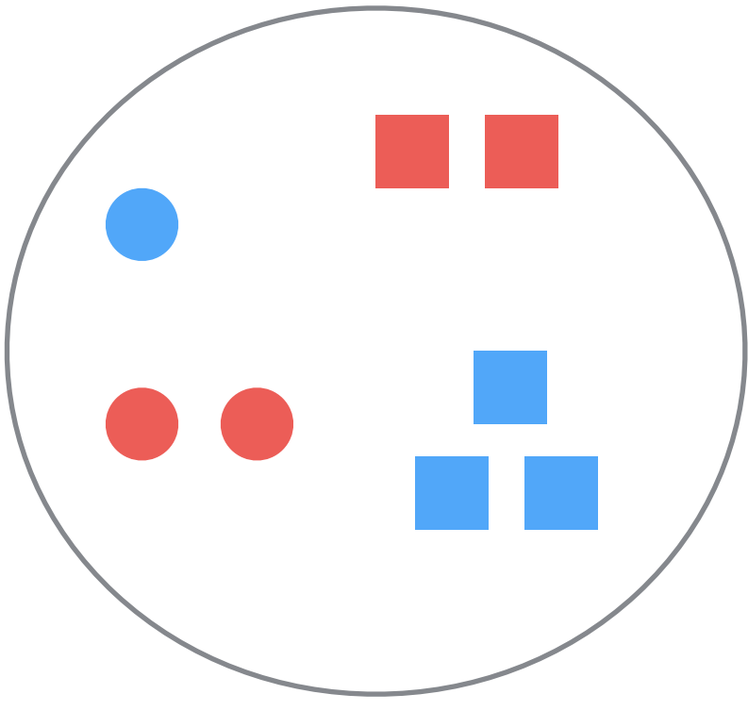


Relational Algebra
Cartesian Product

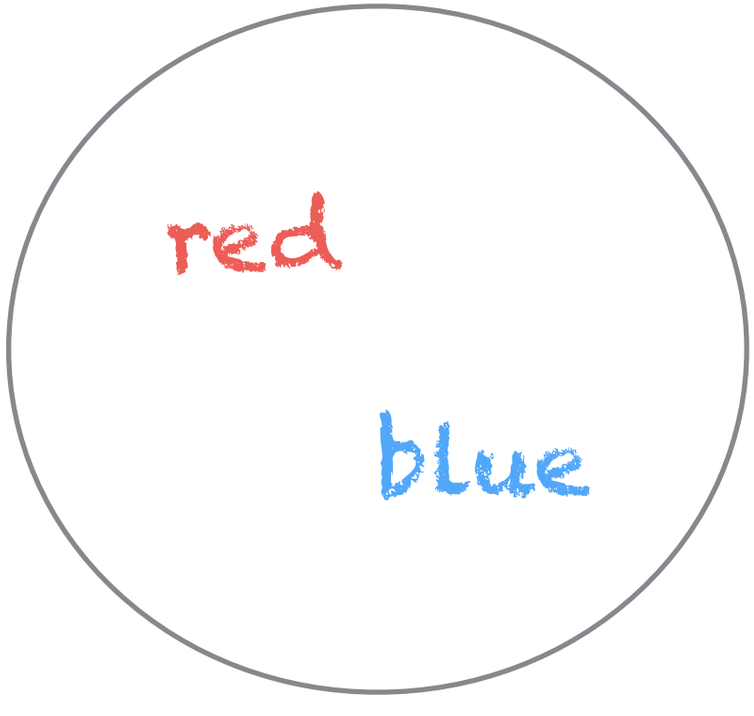
dom0 = {  ,  }
dom1 = {  ,  }
dom2 = {✔, ✔✔, ✔✔✔}
dom3 = {blue, red, circle, square, 1, 2, 3}

R: (A0:dom0, A1:dom1, A2:dom2)
S: (A0:dom0, A1:dom3)

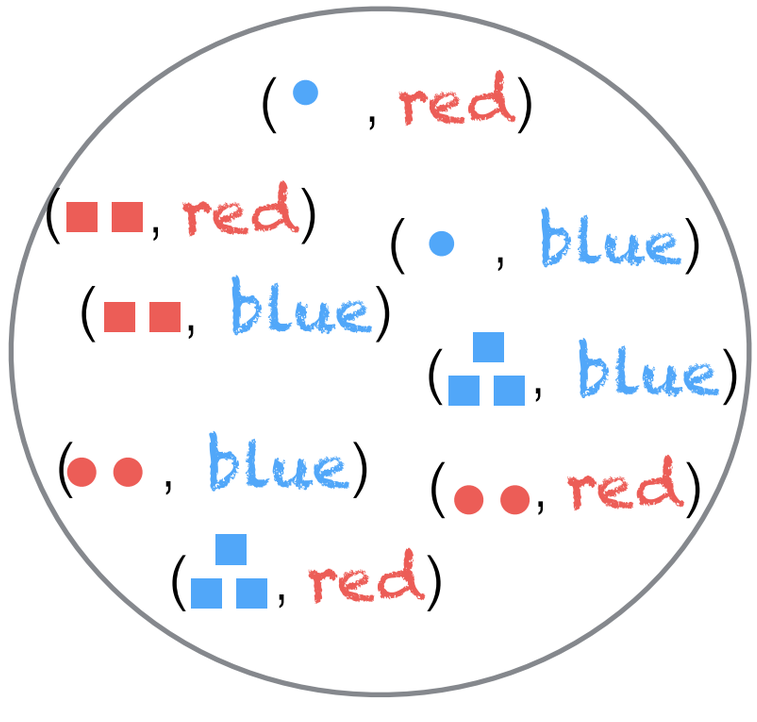
R



S



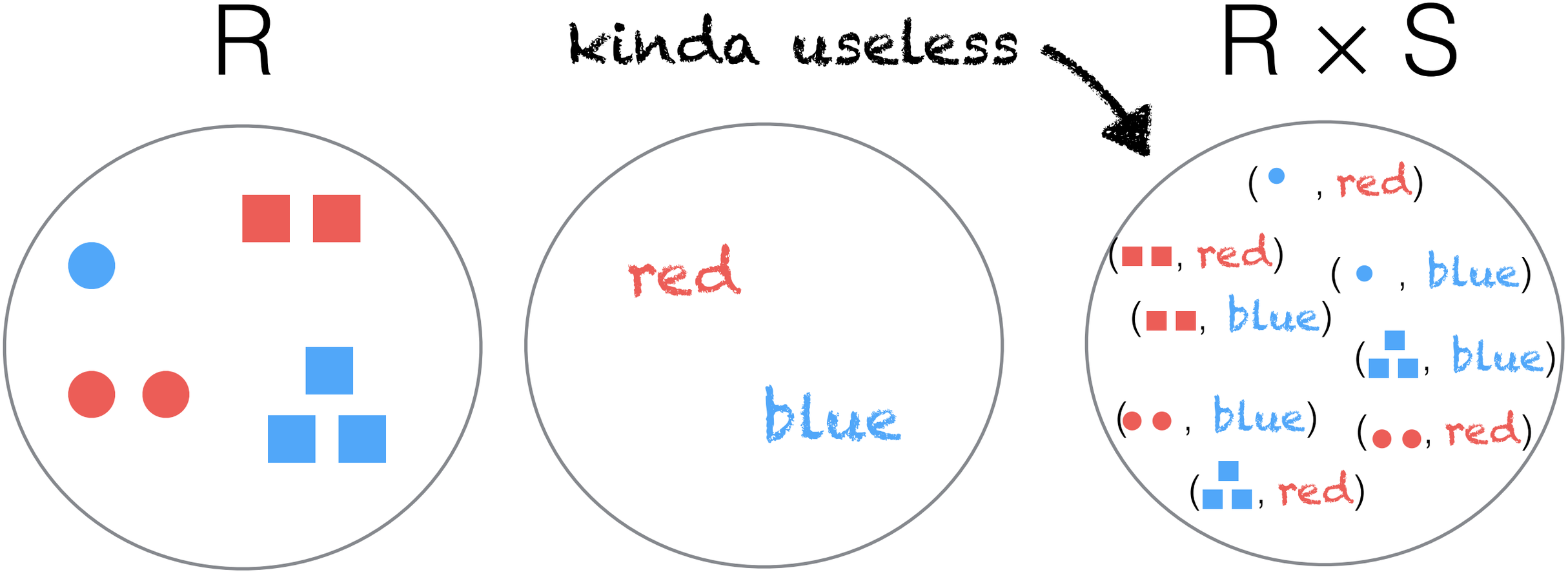
R × S



Relational Algebra
Cartesian Product

dom0 = { , }
dom1 = { , }
dom2 = {✔, ✔✔, ✔✔✔}
dom3 = {blue, red, circle, square, 1, 2, 3}

R: (A0:dom0, A1:dom1, A2:dom2)
S: (A0:dom0, A1:dom3)

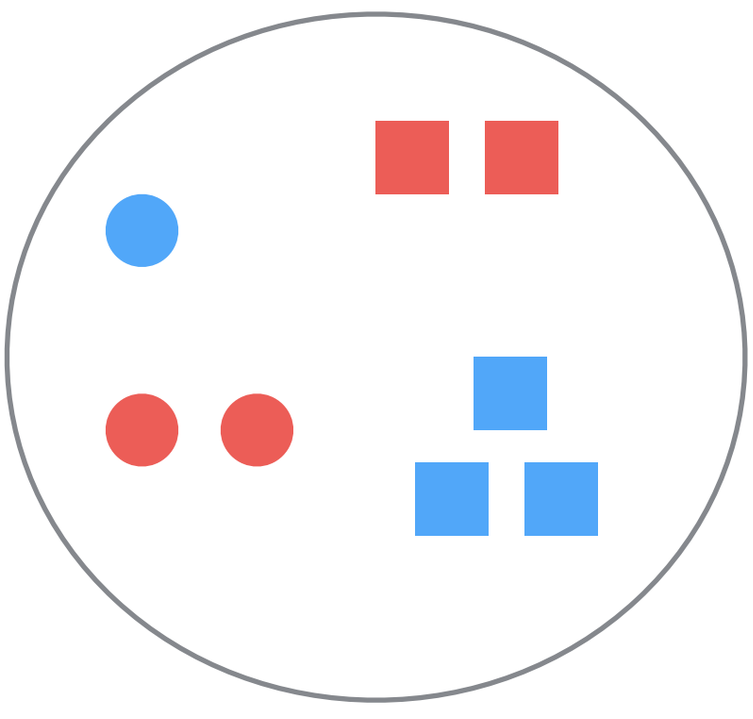


Relational Algebra
Cartesian Product

dom0 = { , }
dom1 = { , }
dom2 = {✔, ✔✔, ✔✔✔}
dom3 = {blue, red, circle, square, 1, 2, 3}

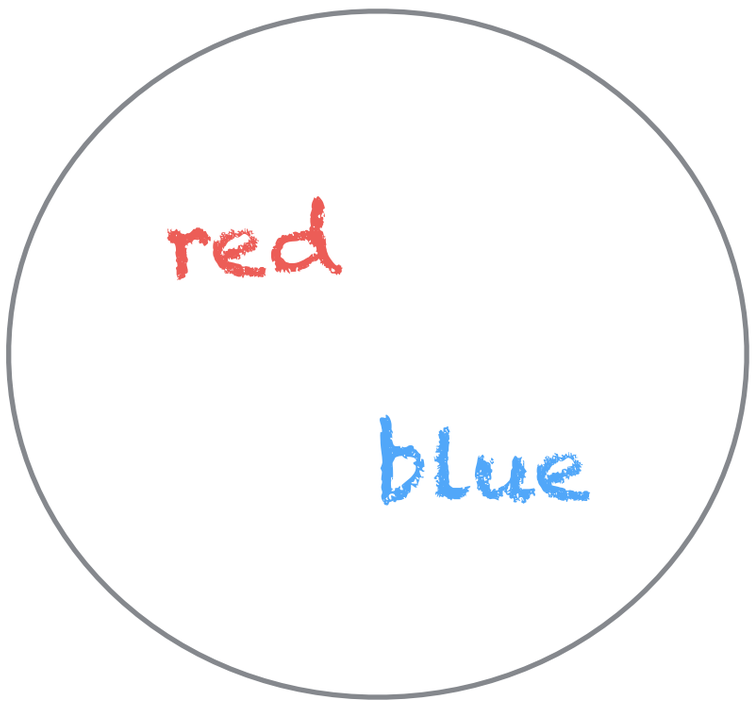
R: (A0:dom0, A1:dom1, A2:dom2)
S: (A0:dom0, A1:dom3)

R

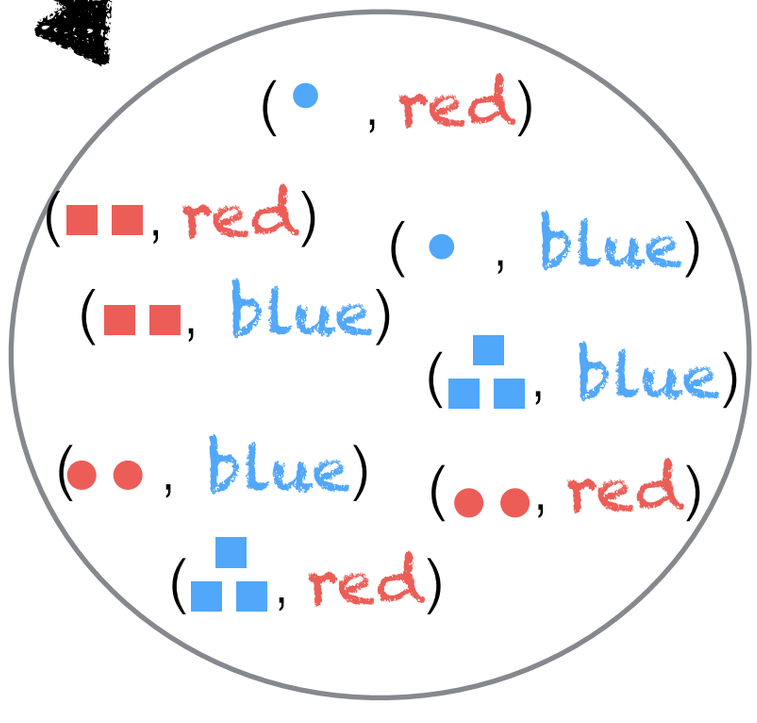


...more later...

S



R x S



Relational Algebra

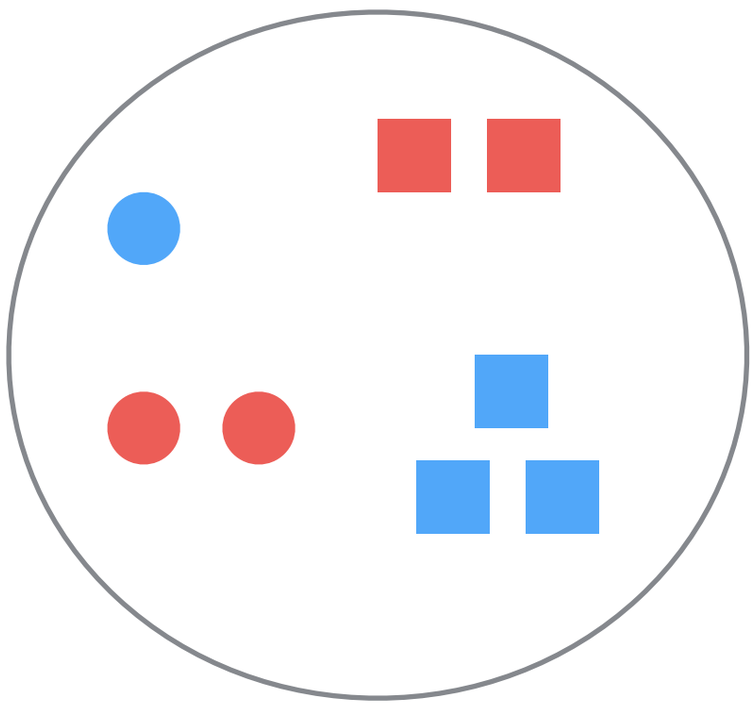
- σ selection
- π project
- ∪ union
- – minus
- × cross
- ρ rename

Relational Algebra
Rename

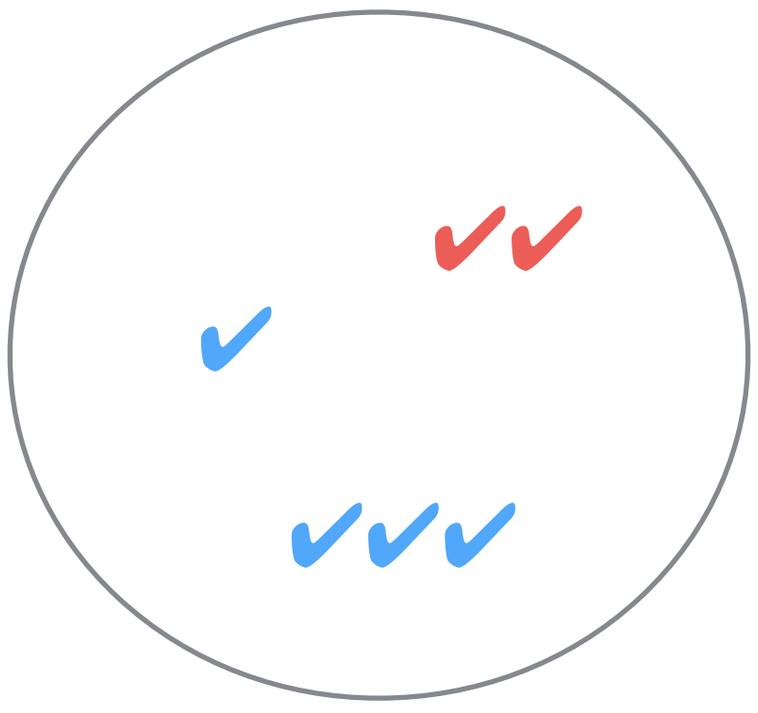
dom0 = {  ,  }
dom1 = {  ,  }
dom2 = {✔, ✔✔, ✔✔✔}
dom3 = {blue, red, circle, square, 1, 2, 3}

R: (A0:dom0, A1:dom1, A2:dom2)
S: (A0:dom0, A1:dom3)

R



$\pi_{A0,A2}(R)$



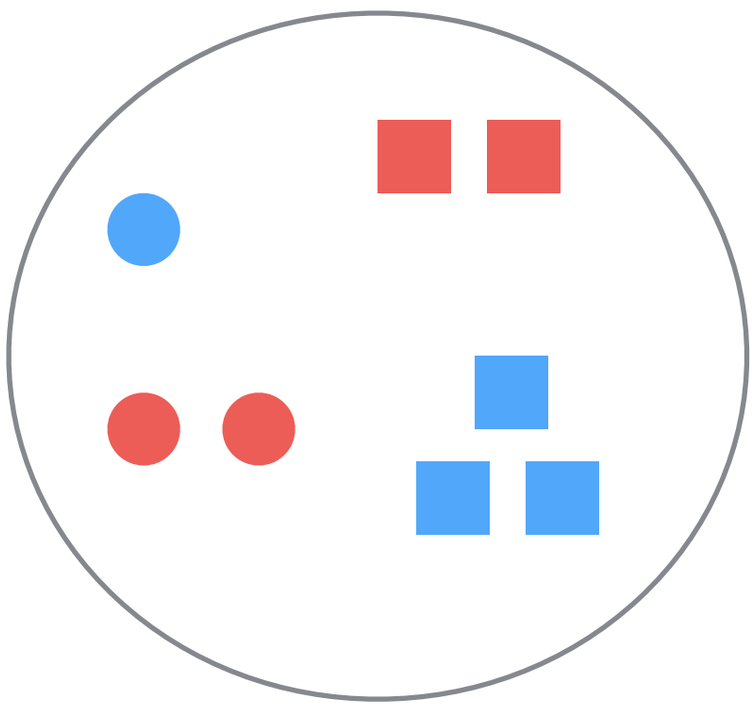
Relational Algebra
Rename

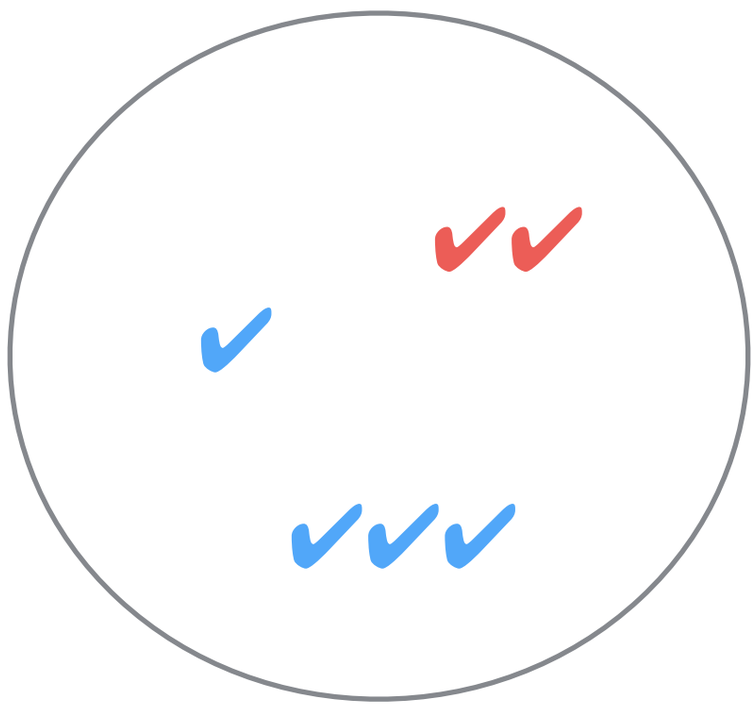
dom0 = {  ,  }
dom1 = {  ,  }
dom2 = {✔, ✔✔, ✔✔✔}
dom3 = {blue, red, circle, square, 1, 2, 3}

R: (A0:dom0, A1:dom1, A2:dom2)
S: (A0:dom0, A1:dom3)

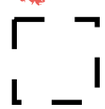
R

$new\_R = \pi_{A0,A2}(R)$



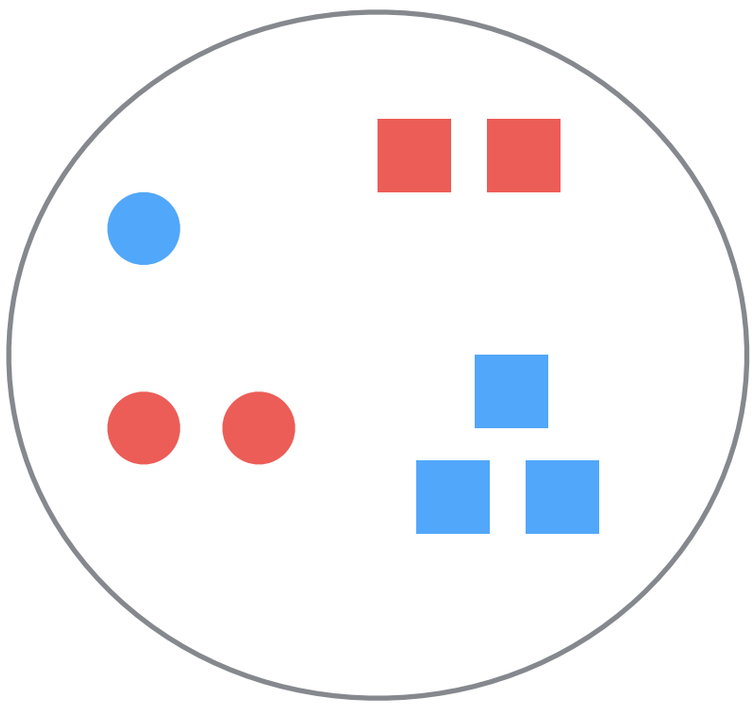


Relational Algebra
Rename

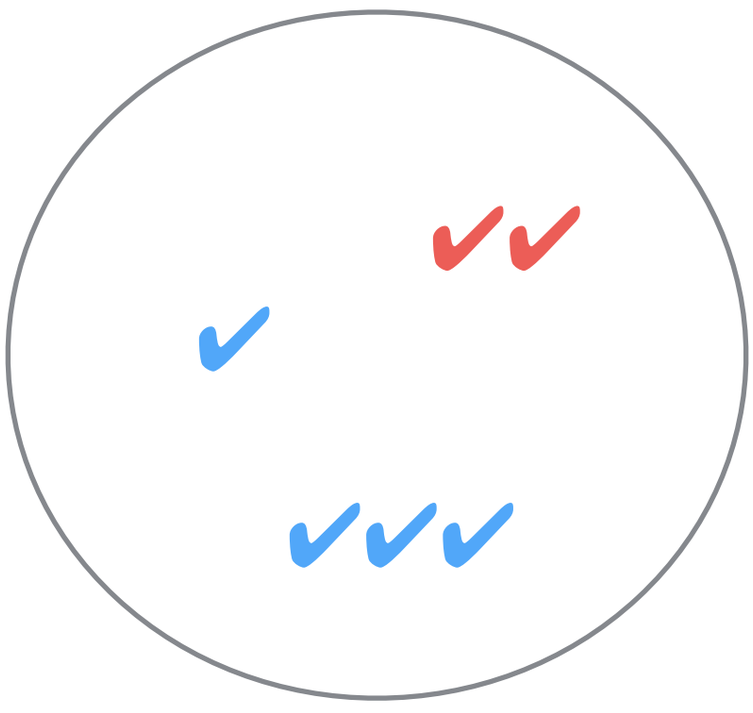
dom0 = {  ,  }
dom1 = {  ,  }
dom2 = {✔, ✔✔, ✔✔✔}
dom3 = {blue, red, circle, square, 1, 2, 3}

R: (A0:dom0, A1:dom1, A2:dom2)
S: (A0:dom0, A1:dom3)

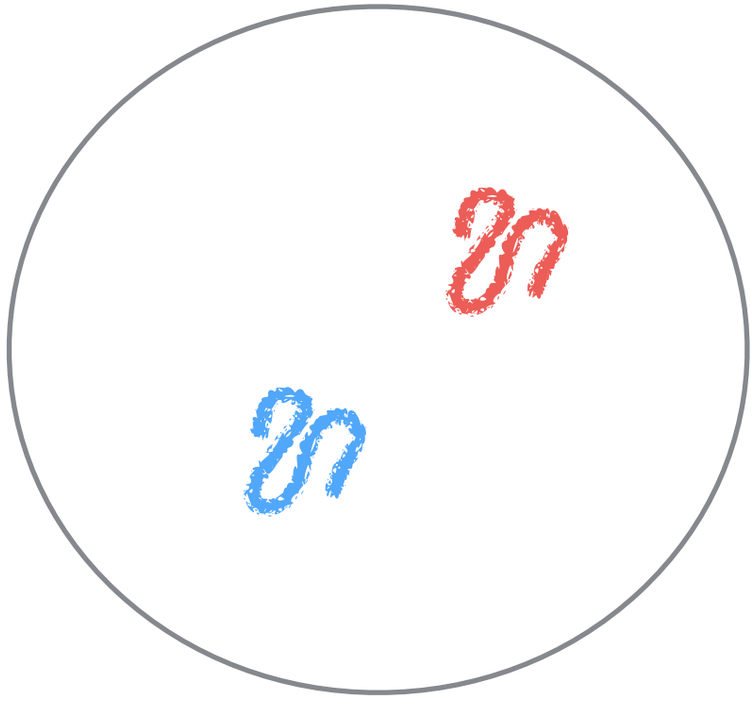
R



$new\_R = \pi_{A0,A2}(R)$

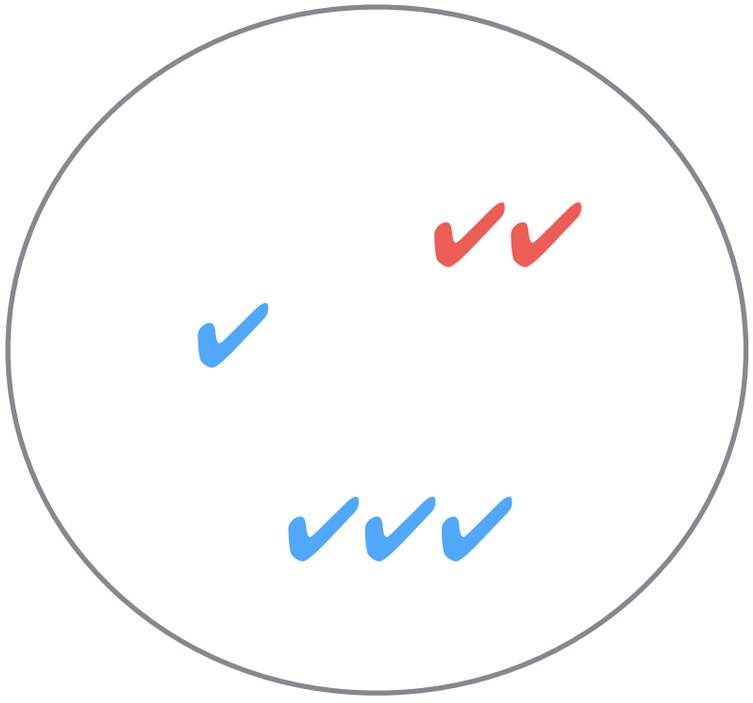


$\pi_{A0}(new\_R)$

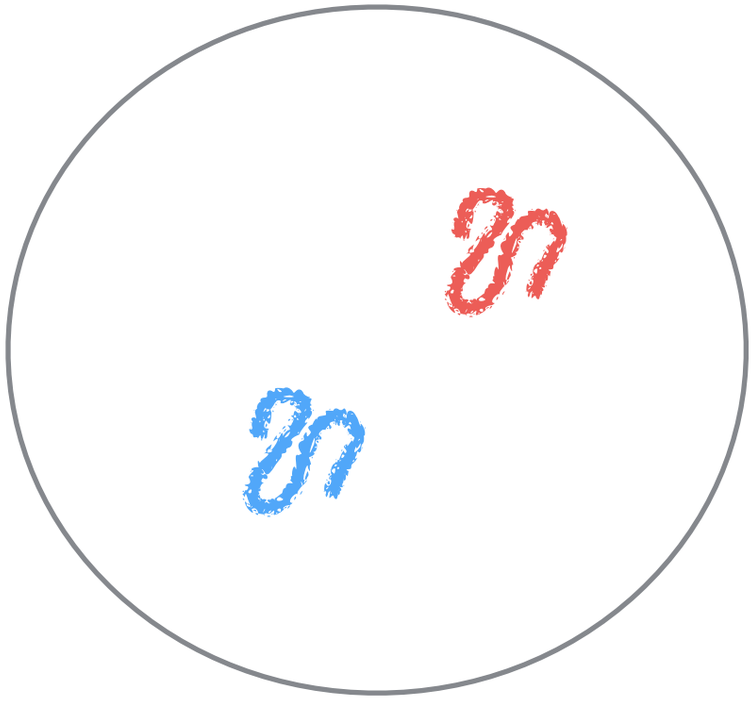


Relational Algebra
Rename

$\rho_{new\_R\_final}(new\_R)$



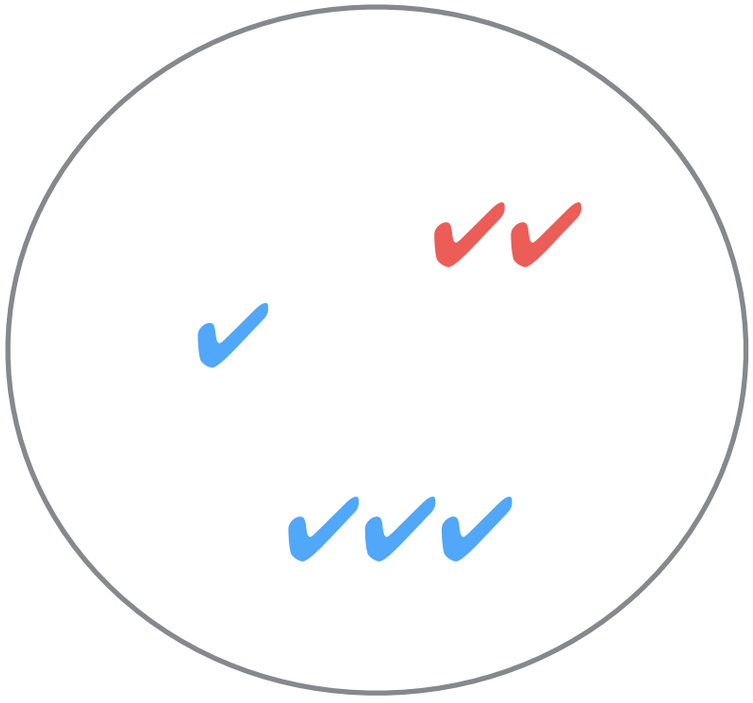
$new\_R = \pi_{A0,A2}(R)$



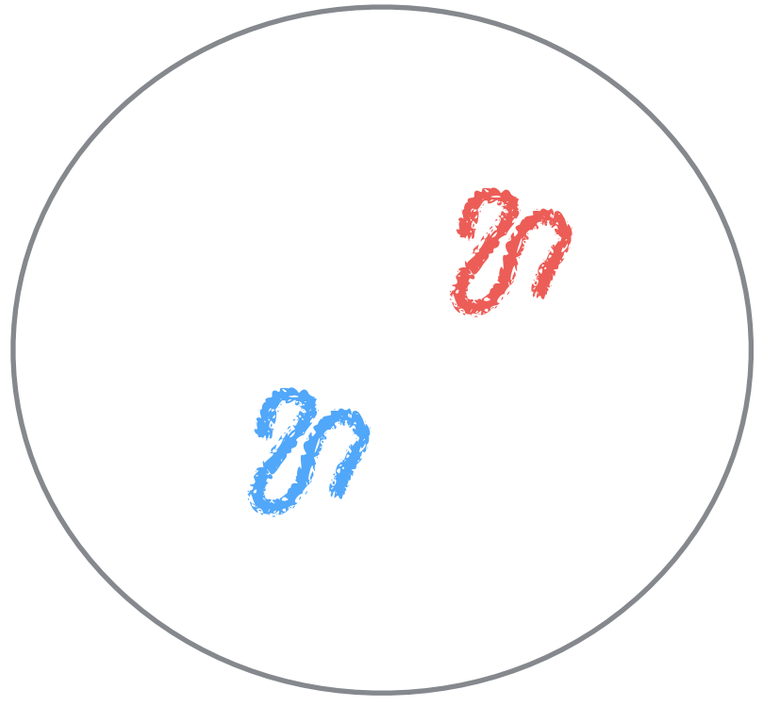
$\pi_{A0}(new\_R\_final)$

Relational Algebra
Rename

$\rho_{new\_R\_final(A0\_working\_jan28\_do\_not\_delete\_this\_one,\ A1)}(new\_R)$



$new\_R = \pi_{A0,A2}(R)$



$\pi_{A0\_working\_jan28\_do\_not\_delete\_this\_one}(new\_R\_final)$

Relational Algebra

- σ selection
- π project
- ∪ union
- – minus
- × cross
- ρ rename

<u>"complete set"</u>
i.e. any of the core relations we want to express can be expressed as a combination of these operations.

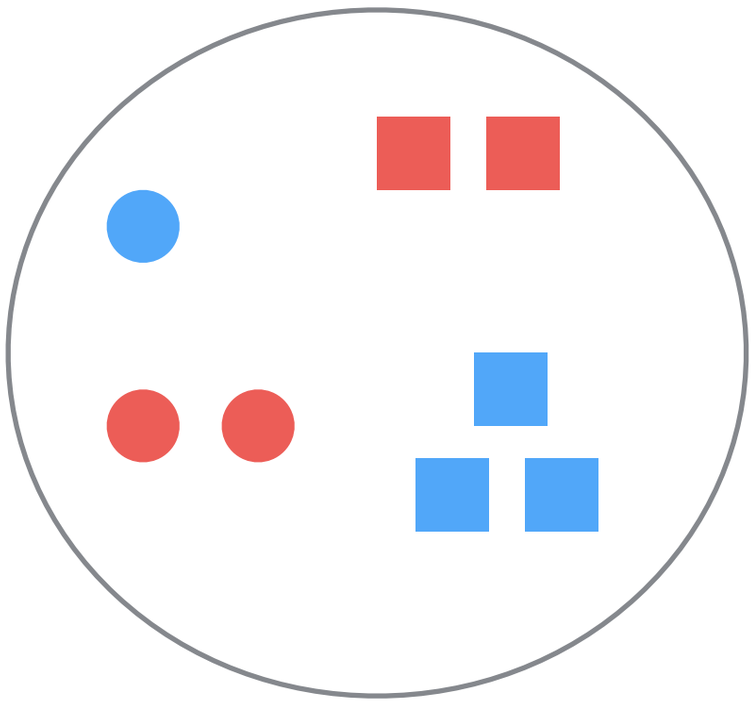
Relational Algebra

Join

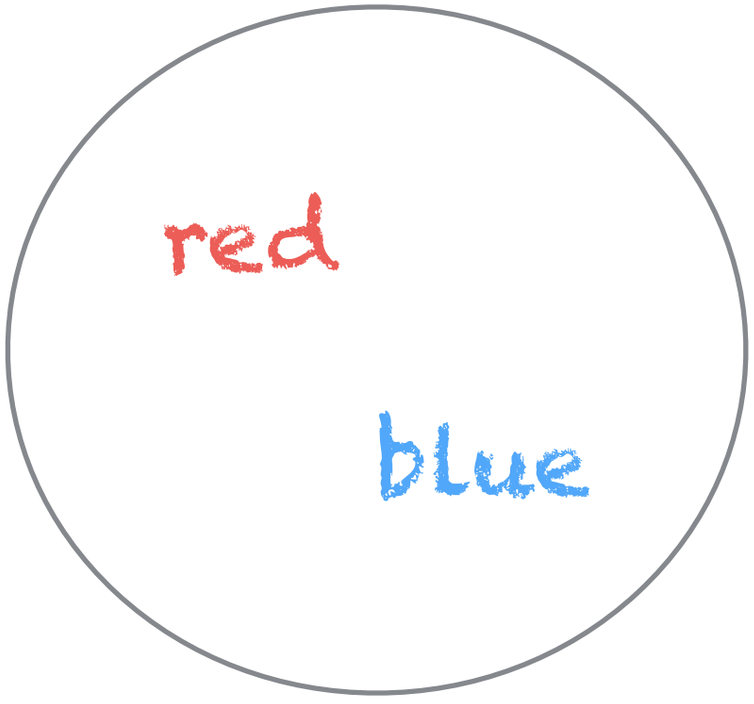
dom0 = {  ,  }
dom1 = {  ,  }
dom2 = {✔, ✔✔, ✔✔✔}
dom3 = {blue, red, circle, square, 1, 2, 3}

R: (A0:dom0, A1:dom1, A2:dom2)
S: (A0:dom0, A1:dom3)

R



S



Find the names of the colors of each element of R.

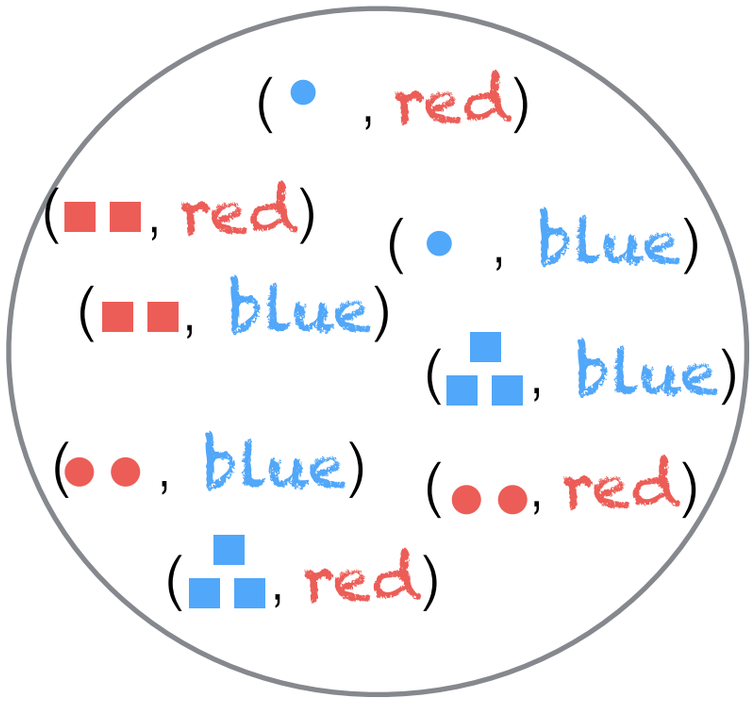
Relational Algebra

Join

dom0 = {  }
dom1 = {  }
dom2 = {✔, ✔✔, ✔✔✔}
dom3 = {blue, red, circle, square, 1, 2, 3}

R: (A0:dom0, A1:dom1, A2:dom2)
S: (A0:dom0, A1:dom3)

R × S



Find the names of the colors of each element of R.

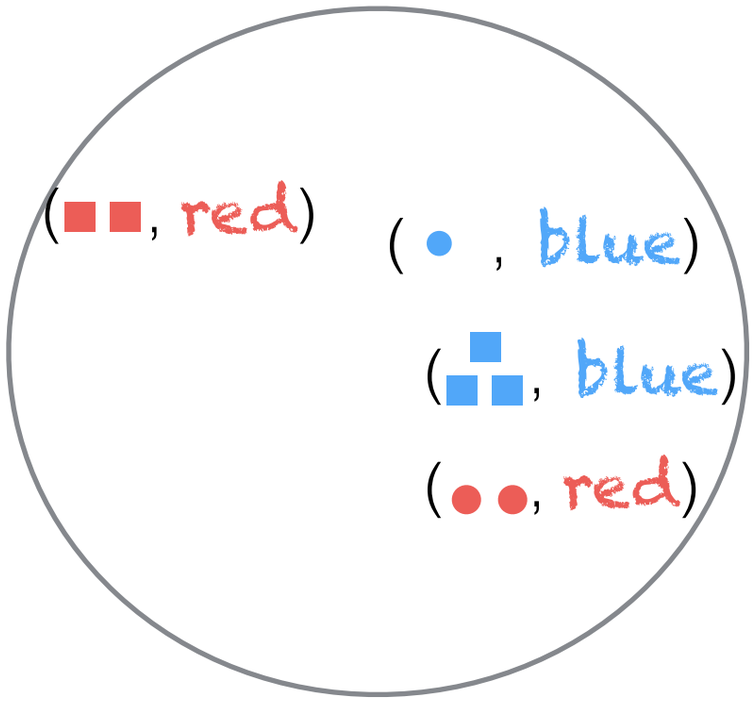
Relational Algebra

Join

dom0 = {  ,  }
dom1 = {  ,  }
dom2 = {✔, ✔✔, ✔✔✔}
dom3 = {blue, red, circle, square, 1, 2, 3}

R: (A0:dom0, A1:dom1, A2:dom2)
S: (A0:dom0, A1:dom3)

$$\sigma_{R.A0 = S.A0}(R \times S)$$



Find the names of the colors of each element of R.

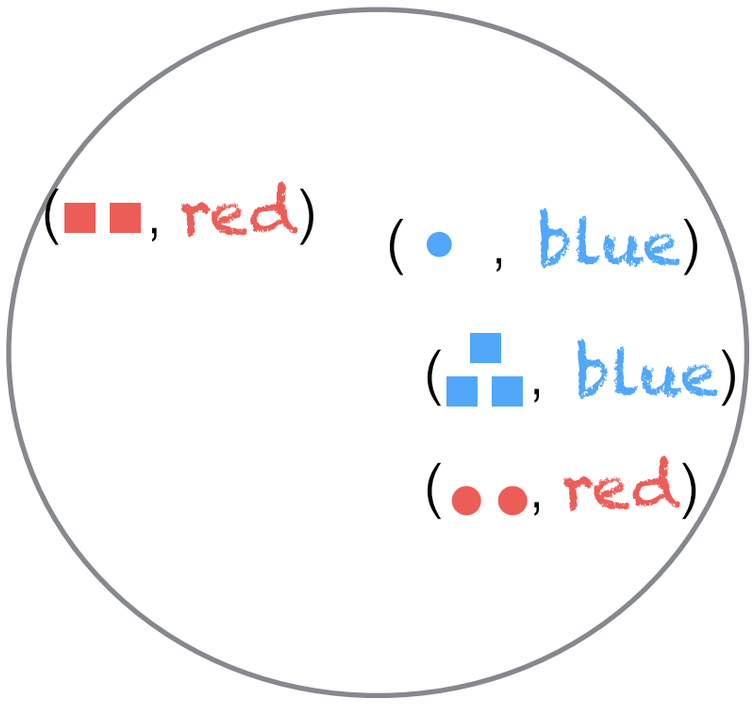
Relational Algebra

Join

dom0 = {  ,  }
dom1 = {  ,  }
dom2 = {✔, ✔✔, ✔✔✔}
dom3 = {blue, red, circle, square, 1, 2, 3}

R: (A0:dom0, A1:dom1, A2:dom2)
S: (A0:dom0, A1:dom3)

$$R \bowtie_{R.A0 = S.A0} S$$



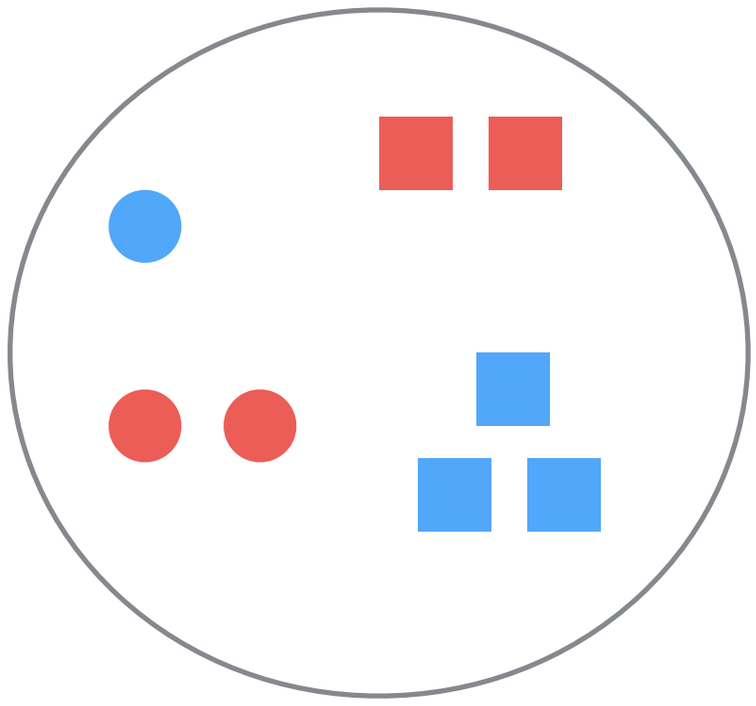
Find the names of the colors of each element of R.

Relational Algebra
Intersection

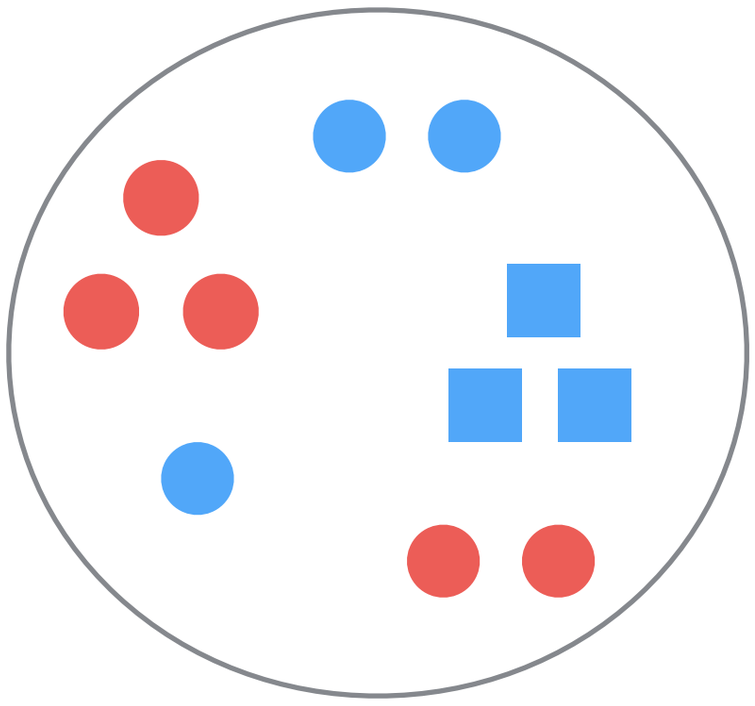
dom0 = {  ,  }
dom1 = {  ,  }
dom2 = {✔, ✔✔, ✔✔✔}
dom3 = {blue, red, circle, square, 1, 2, 3}

R: (A0:dom0, A1:dom1, A2:dom2)
R': (A0:dom0, A1:dom1, A2:dom2)

R



R'



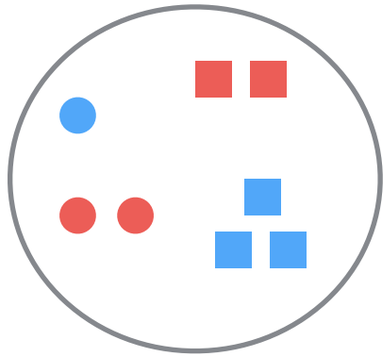
Find the elements appearing in both R and R'.

Relational Algebra Intersection

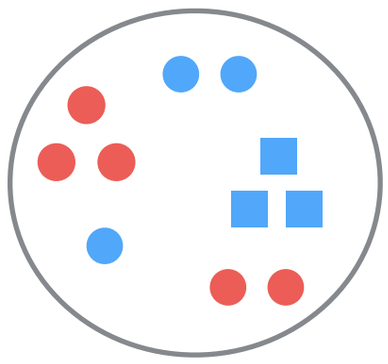
dom0 = {  ,  }
dom1 = {  ,  }
dom2 = {✔, ✔✔, ✔✔✔}
dom3 = {blue, red, circle, square, 1, 2, 3}

R: (A0:dom0, A1:dom1, A2:dom2)
R': (A0:dom0, A1:dom1, A2:dom2)

R



R'



- σ selection
- – minus
- π project
- × cross
- ∪ union
- ρ rename

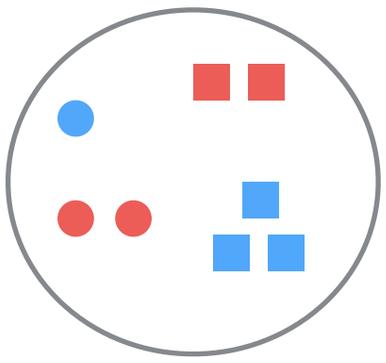
Find the elements appearing in both R and R'.

Relational Algebra Intersection

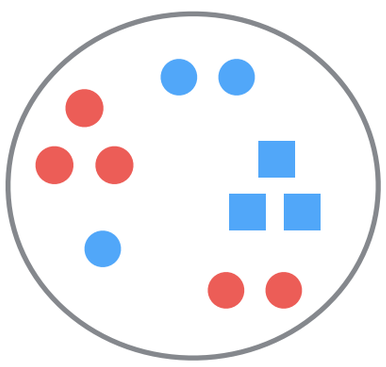
dom0 = {  ,  }
dom1 = {  ,  }
dom2 = {✔, ✔✔, ✔✔✔}
dom3 = {blue, red, circle, square, 1, 2, 3}

R: (A0:dom0, A1:dom1, A2:dom2)
R': (A0:dom0, A1:dom1, A2:dom2)

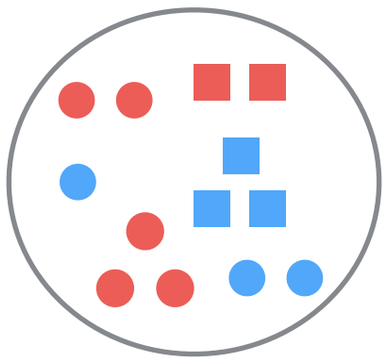
R



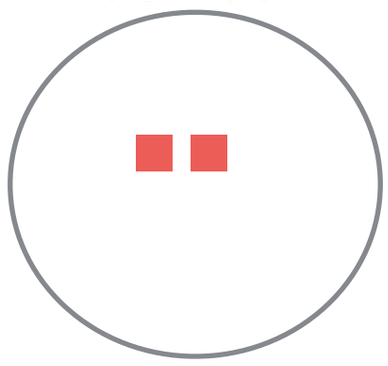
R'



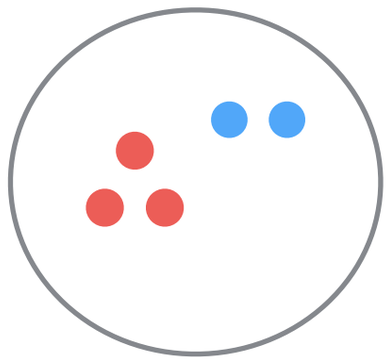
R ∪ R'



R - R'



R' - R



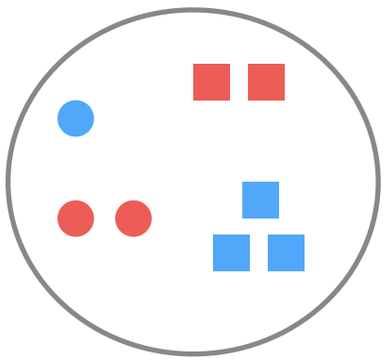
Find the elements appearing in both R and R'.

Relational Algebra Intersection

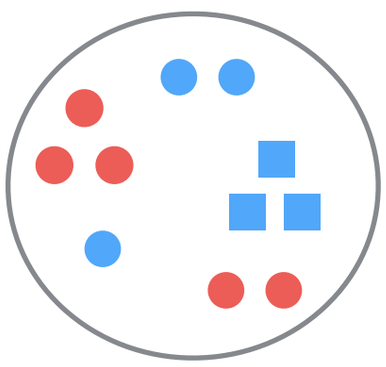
dom0 = {  ,  }
dom1 = {  ,  }
dom2 = {✔, ✔✔, ✔✔✔}
dom3 = {blue, red, circle, square, 1, 2, 3}

R: (A0:dom0, A1:dom1, A2:dom2)
R': (A0:dom0, A1:dom1, A2:dom2)

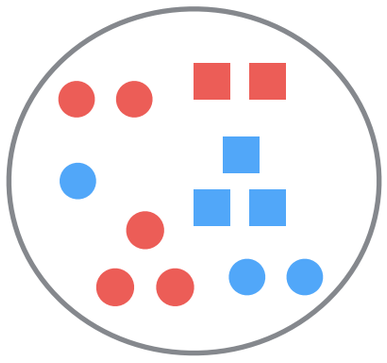
R



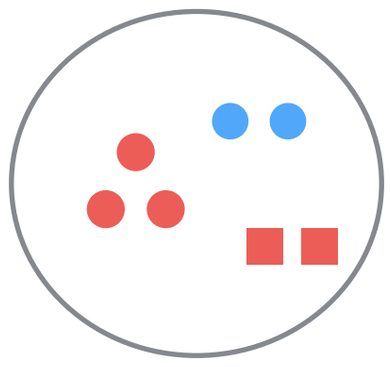
R'



R ∪ R'



(R - R') ∪ (R' - R)



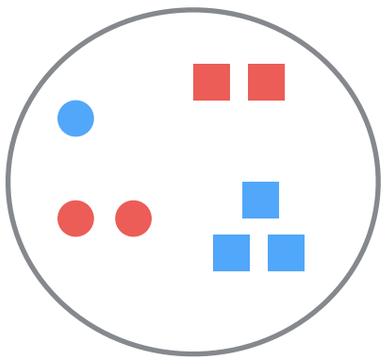
Find the elements appearing in both R and R'.

Relational Algebra Intersection

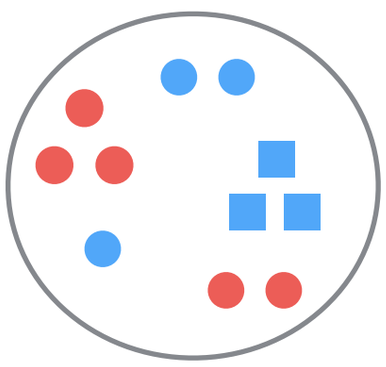
dom0 = {  ,  }
dom1 = {  ,  }
dom2 = {✔, ✔✔, ✔✔✔}
dom3 = {blue, red, circle, square, 1, 2, 3}

R: (A0:dom0, A1:dom1, A2:dom2)
R': (A0:dom0, A1:dom1, A2:dom2)

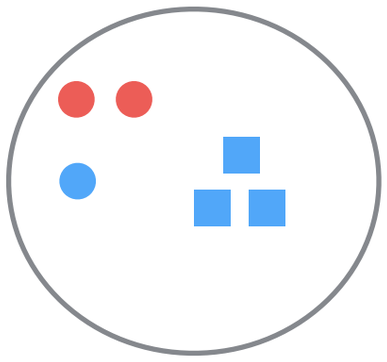
R



R'



R ∪ R' – (R - R') ∪ (R' - R)



Find the elements appearing in both R and R'.

k bye