CSE 199: UB Seminar (Fall 2026) - Data management

Lecture 2: Network Model & DBTG

Zhuoyue Zhao
zzhao35 [at] buffalo [dot] edu

University at Buffalo
Department of Computer Science and Engineering
School of Engineering and Applied Sciences

Data model

- Data modeling is about
  - How do we logically represent the data and their relationships

- Example: the Internet Movie Database (IMDb)
  - Stores data about entities such as
    titles (movies/tv series/…), (production) companies, casts, etc.
    - Each type of entity is associated with similar types of data
      - title: name, production year, genre, …
      - companies: name, country,…
  - What kind of relationships can there be among the entities?
    - Example: Movie “Titanic” was produced by 20th Century Fox, Paramount Pictures, Lightstorem Entertainment, LCI Seguros, Baja Studios [link]
  - **What else? Let’s brain storm.**

Common relationship between entities

- Many-to-many, e.g.,
  - [title] produced by [company]
- One-to-many, e.g.,
  - [cast_member] casted in [title]
- One-to-one, e.g.,
  - [student] owns [UB email account]
- “is a”, e.g.,
  - [actor/actress] is a [cast_member]
  - [director] is a [cast_member]
- And more…

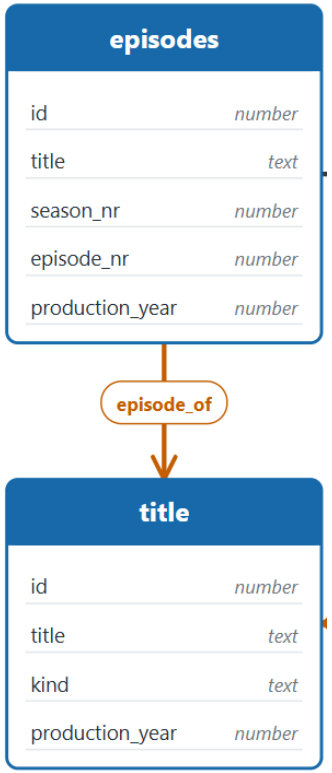
DBTG model

- Entity => data structure (main topics in CSE 250)
  - In network model, each type of entity is represented by a flat record structure

33460	"6Teen"	"tv series"	2004
3645059	"Elena et les hommes"	"movie"	1956
4615374	"Vanishing Son IV"	"tv movie"	1994

DBTG model (cont'd)

- DBTG only models one kind of relationship: one-to-many
  - "DBTG set" (not to be confused with set as in mathematics)
  - models an owner-member relationship
    - you can give each DBTG set a name



episode_of

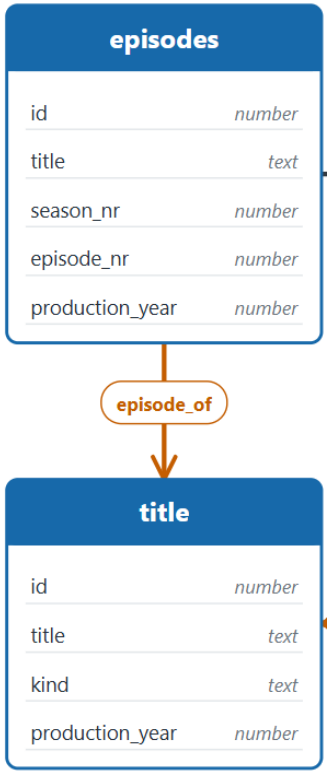
33461	"2-4-1"	3	9	2007
33463	"A Crime of Fashion"	2	24	2006
33467	"Another Day at the Office"	3	10	2007
214701	"Citizens' Assembly"	1	4	2011
214703	"Date Night"	1	1	2011

33460	"6Teen"	"tv series"	2004
214700	"At Home with Julia"	"tv series"	2011
3645059	"Elena et les hommes"	"movie"	1956

DBTG model (cont'd)

- DBTG only models one kind of relationship: one-to-many
  - "DBTG set" (not to be confused with set as in mathematics)
  - models an owner-member relationship
    - you can give each DBTG set a name

- A record cannot have >=2 owners within the same DBTG set
- A record can have any number of members within the same DBTG set
- A record can appear in multiple different DBTG sets



episode_of

33461 "2-4-1" 3 9 2007

33463 "A Crime of Fashion" 2 24 2006

33467 "Another Day at the Office" 3 10 2007

214701 "Citizens' Assembly" 1 4 2011

214703 "Date Night" 1 1 2011

33460 "6Teen" "tv series" 2004

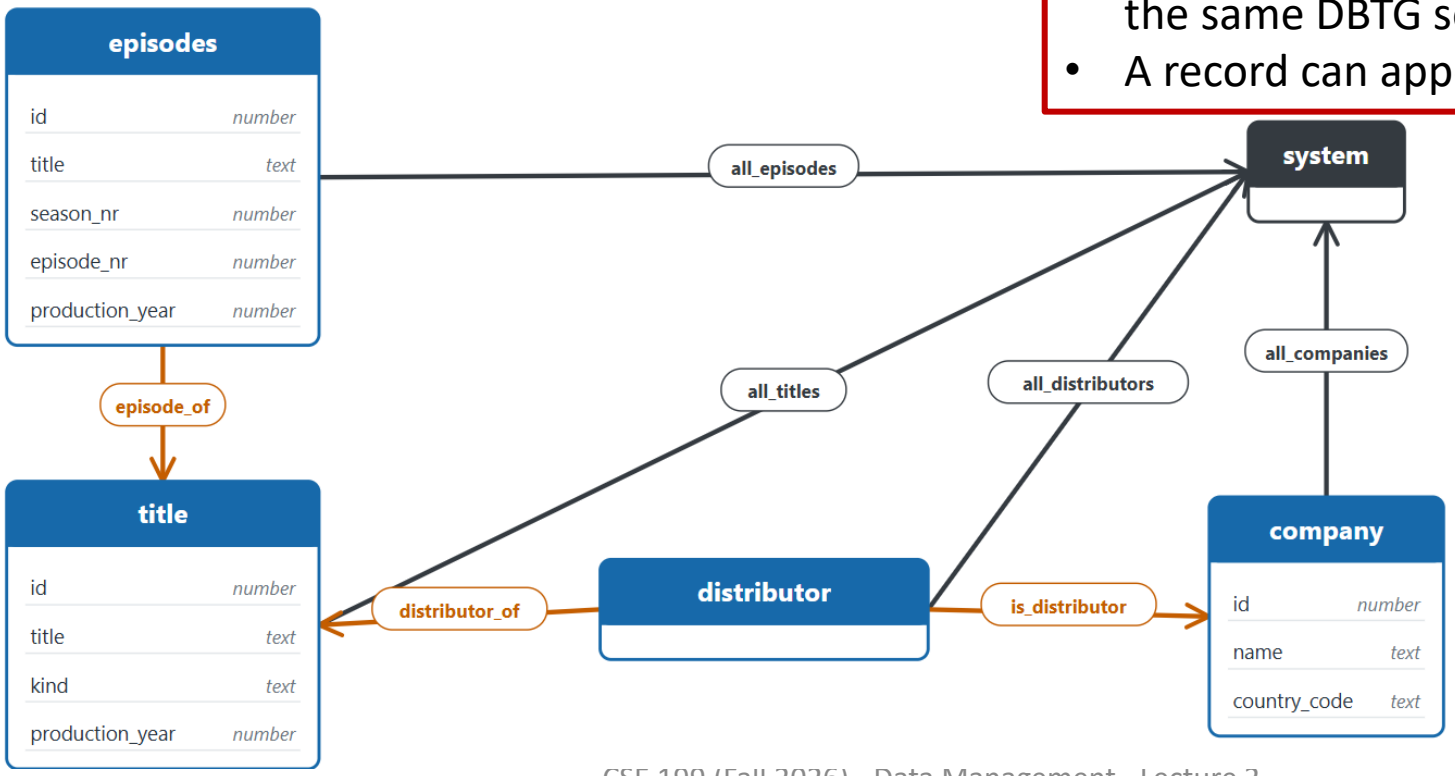
214700 "At Home with Julia" "tv series" 2011

3645059 "Elena et les hommes" "movie" 1956

DBTG model (cont'd)

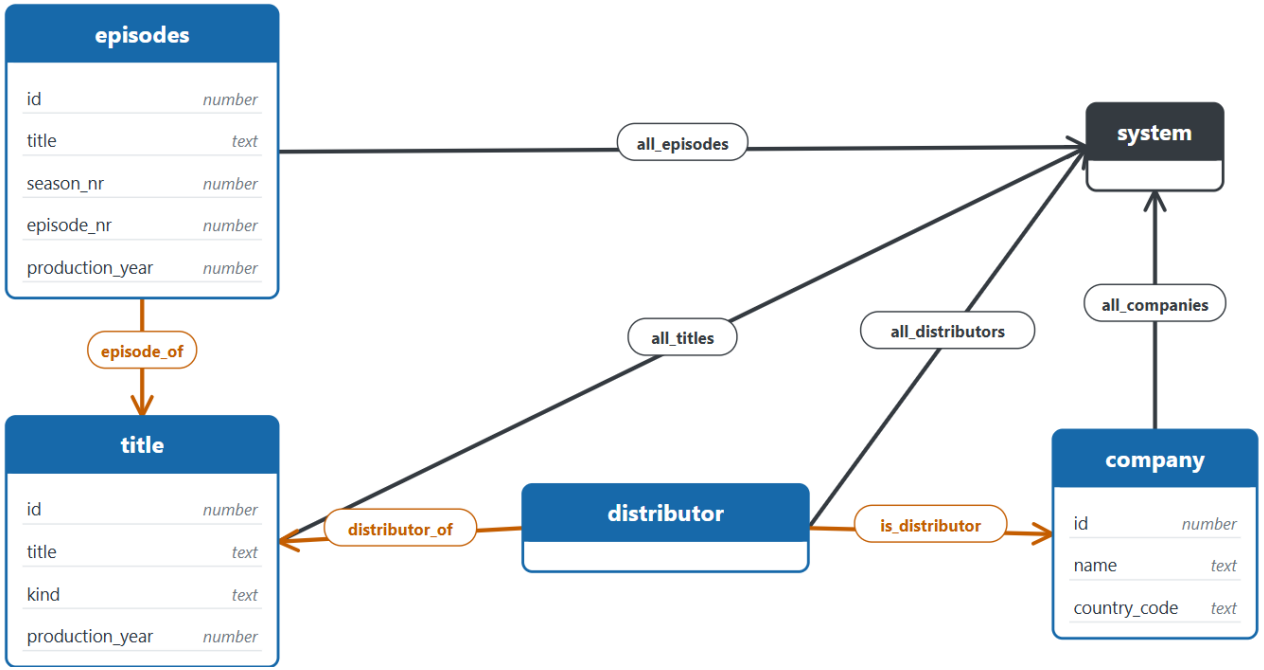
- DBTG only models one kind of relationship: one-to-many
  - "DBTG set" (not to be confused with set as in mathematics)
  - models an owner-member relationship
    - you can give each DBTG set a name

- A record cannot have >=2 owners within the same DBTG set
- A record can have any number of members within the same DBTG set
- A record can appear in multiple different DBTG sets



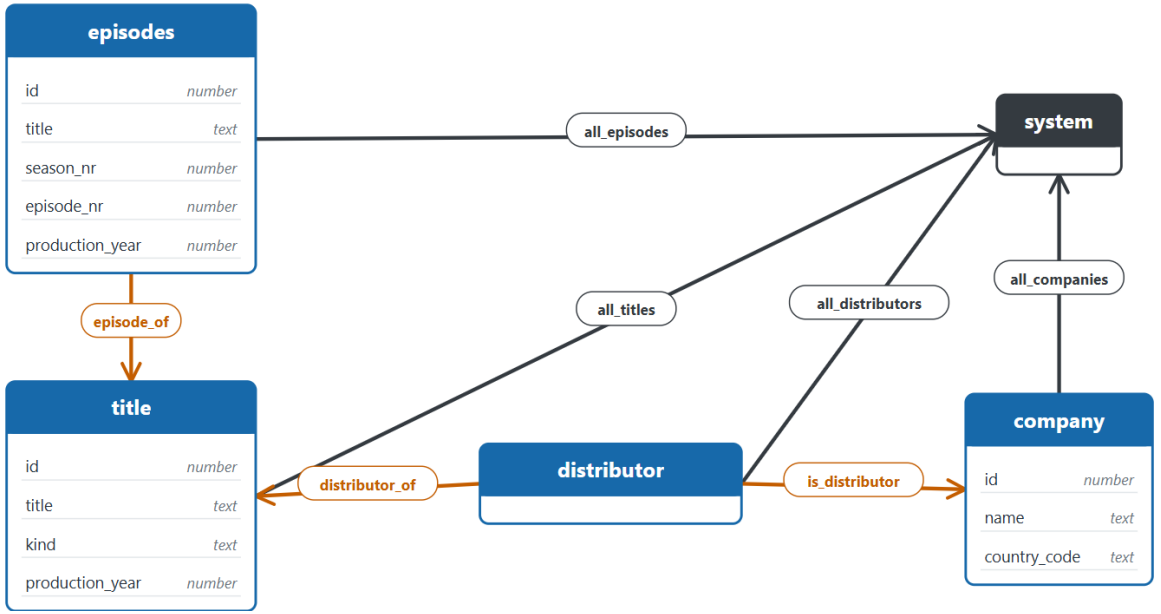
Navigating DBTG database using DBTG commands

- Navigating through DBTG database requires you to follow the links
  - As a programmer, imagine you're pointing a few "lasers" to some records
    - "Program state" - in memory and you can use immediately
  - Records are in a "pile" -- you have to find it and fetch it to know what it is about
    - "database instance" - on disk and you must "get" it to find the data inside



DBTG query example

- Find all episodes of At Home with Julia
  1. Find the title (record) with a title (name) of “At Home with Julia”
  2. Find the first episode (record) that is an episode of the title in 1.
     a. Find the next episode (record) that is an episode of the title in 1.
     b. Get and print the episode (record).
     c. Repeat 2(a), 2(b) until there is no next.



DBTG query example

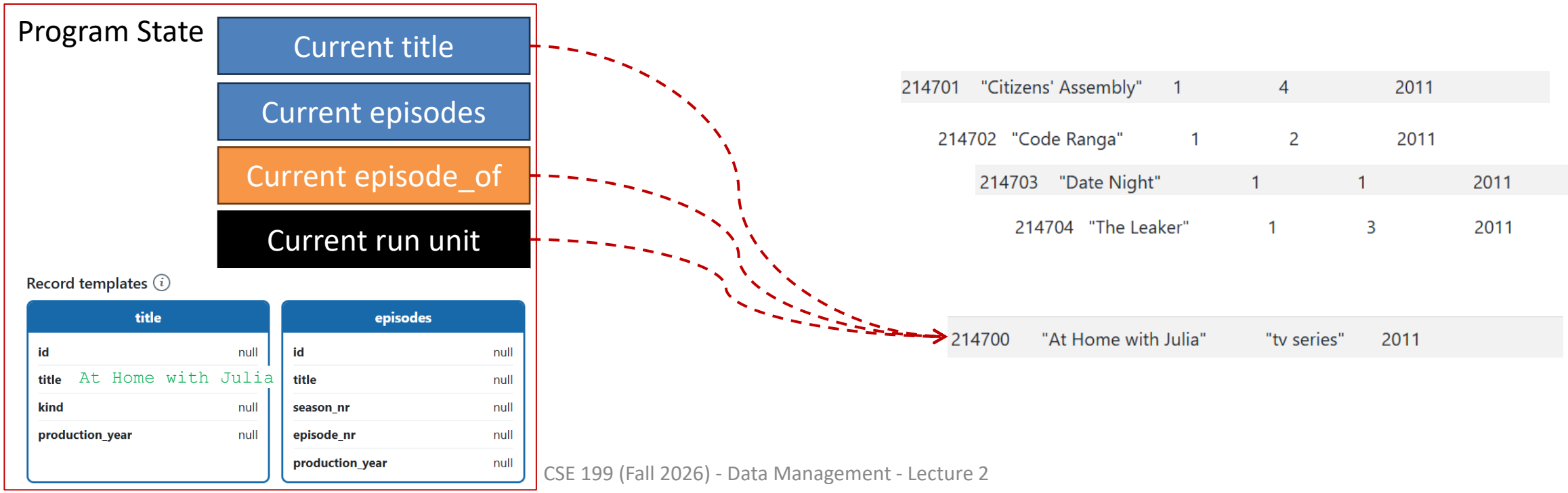
- Find all episodes of At Home with Julia
  1. Find the title (record) with a title (name) of “At Home with Julia”
  2. Find the first episode (record) that is an episode of the title in 1.
     a. Find the next episode (record) that is an episode of the title in 1.
     b. Get and print the episode (record).
     c. Repeat from 2(a) unless there 2(b) didn’t find any

214701	"Citizens' Assembly"	1	4	2011
214702	"Code Ranga"	1	2	2011
214703	"Date Night"	1	1	2011
214704	"The Leaker"	1	3	2011

214700	"At Home with Julia"	"tv series"	2011

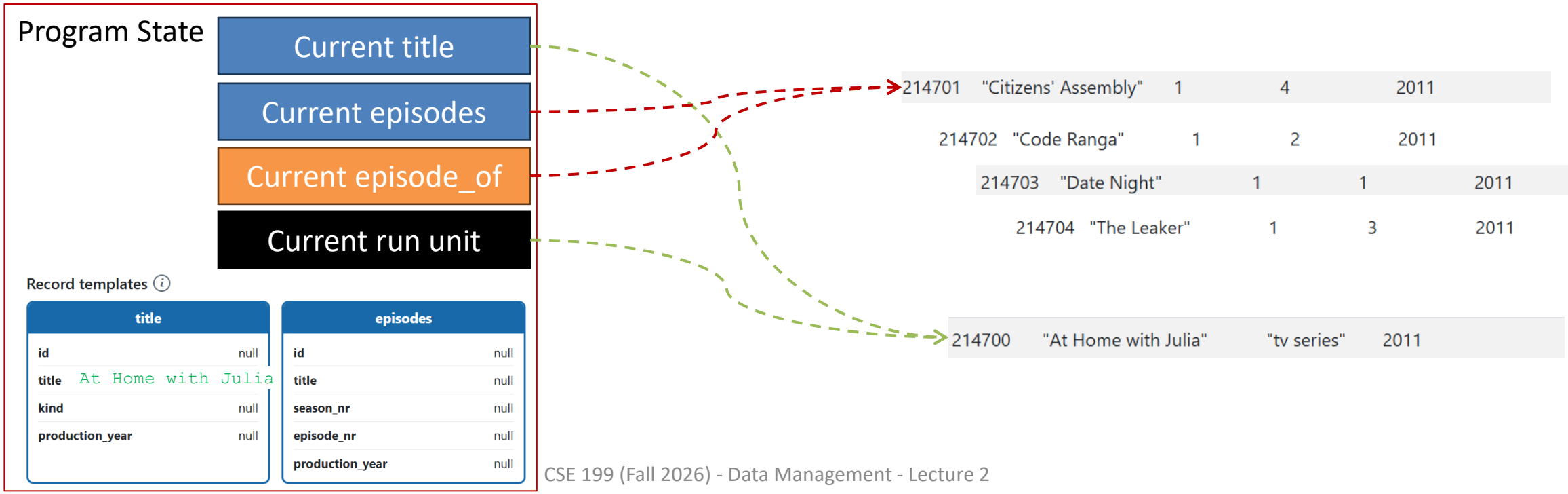
DBTG query example

- Find all episodes of At Home with Julia
  1. `title.title := "At Home with Julia"; find any title using title;`
  2. `find first episodes within episode_of;`
     a. Get and print the episode (record).
     b. Find the next episode (record) that is an episode of the title in 1.
     c. Repeat from 2(a) unless there 2(b) didn't find any



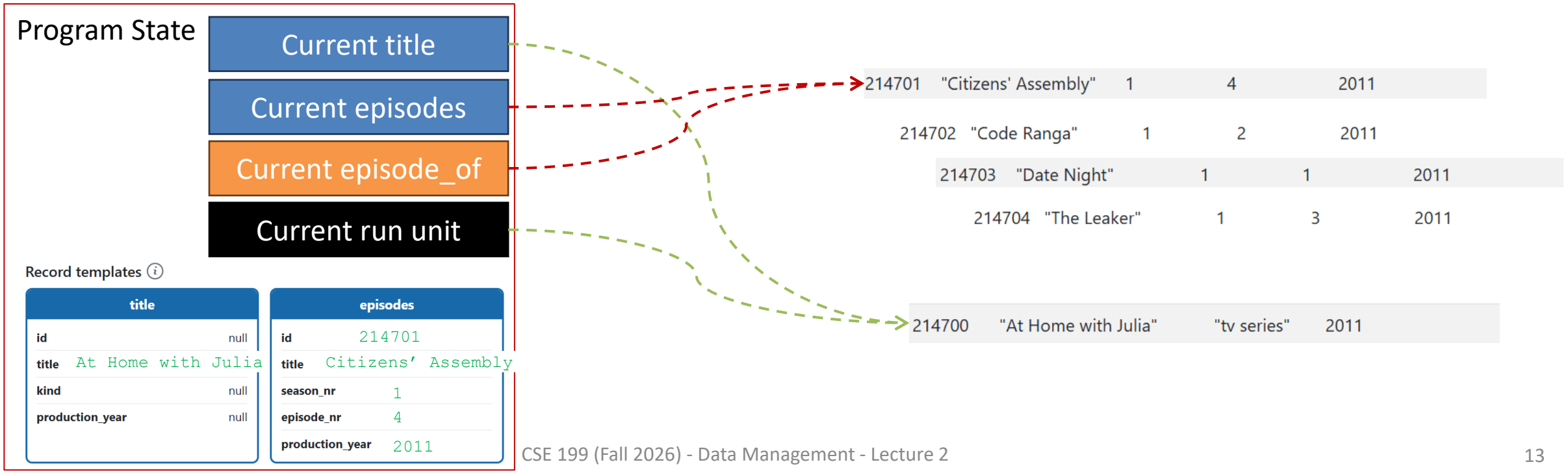
DBTG query example

- Find all episodes of At Home with Julia
  1. `title.title := "At Home with Julia"; find any title using title;`
  2. `find first episodes within episode_of;`
     a. `get episodes;`
     b. Find the next episode (record) that is an episode of the title in 1.
     c. Repeat from 2(a) unless there 2(b) didn't find any



DBTG query example

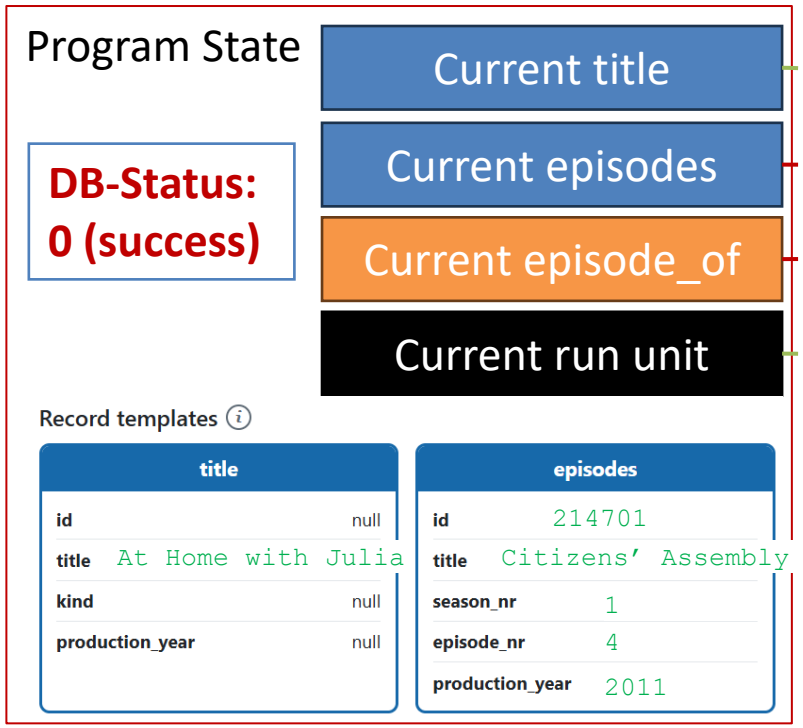
- Find all episodes of At Home with Julia
  1. `title.title := "At Home with Julia"; find any title using title;`
  2. `find first episodes within episode_of;`
     - a. `get episodes;`
     - b. `find next episodes within episode_of;`
     - c. Repeat from 2(a) unless there 2(b) didn't find any



DBTG query example

- Find all episodes of At Home with Julia
  1. `title.title := "At Home with Julia"; find any title using title;`
  2. `find first episodes within episode_of;`
     a. `get episodes;`
     b. `find next episodes within episode_of;`
     c. Repeat from 2(a) unless there 2(b) didn't find any

How do we know we found something?



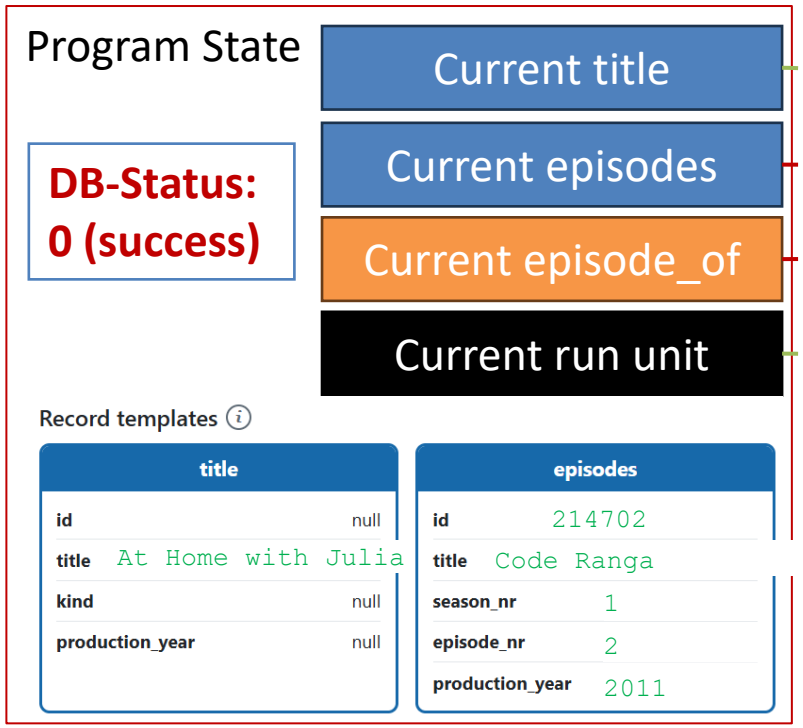
214701	"Citizens' Assembly"	1	4	2011
214702	"Code Ranga"	1	2	2011
214703	"Date Night"	1	1	2011
214704	"The Leaker"	1	3	2011

214700	"At Home with Julia"	"tv series"	2011

DBTG query example

- Find all episodes of At Home with Julia
  1. `title.title := "At Home with Julia"; find any title using title;`
  2. `find first episodes within episode_of;`
     - a. `get episodes;`
     - b. `find next episodes within episode_of;`
     - c. Repeat from 2(a) unless there 2(b) didn't find any

How do we know we found something?



214701	"Citizens' Assembly"	1	4	2011
214702	"Code Ranga"	1	2	2011
214703	"Date Night"	1	1	2011
214704	"The Leaker"	1	3	2011

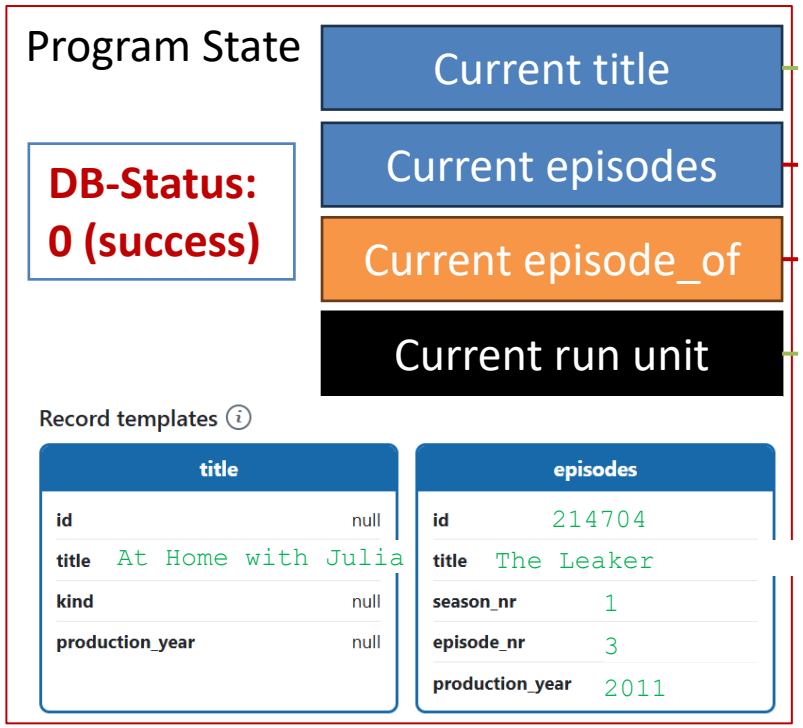
214700	"At Home with Julia"	"tv series"	2011

DBTG query example

- Find all episodes of At Home with Julia
  1. `title.title := "At Home with Julia"; find any title using title;`
  2. `find first episodes within episode_of;`
     - a. `get episodes;`
     - b. `find next episodes within episode_of;`
     - c. Repeat from 2(a) unless there 2(b) didn't find any

fast forward for a few iterations…

How do we know we found something?



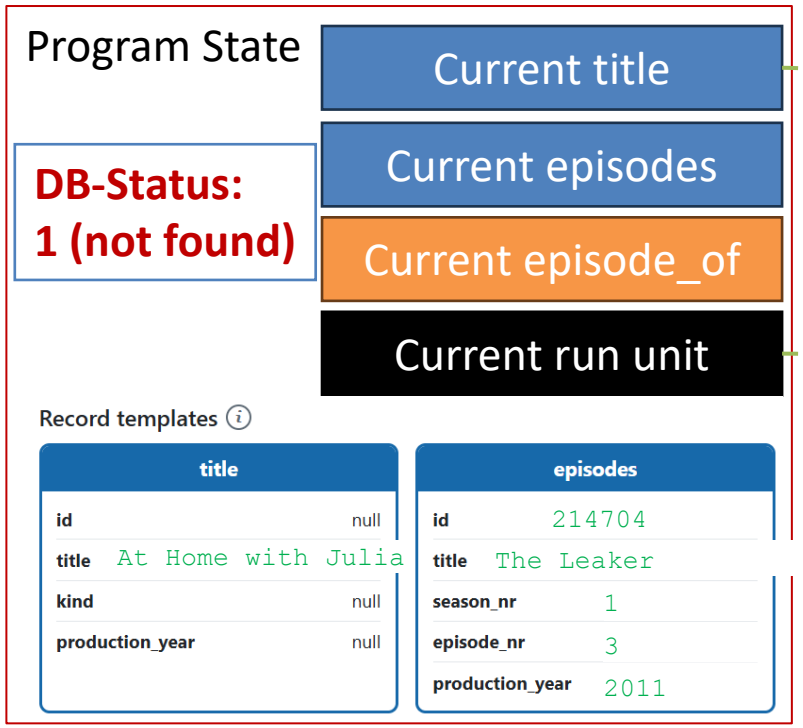
214701	"Citizens' Assembly"	1	4	2011
214702	"Code Ranga"	1	2	2011
214703	"Date Night"	1	1	2011
214704	"The Leaker"	1	3	2011

214700	"At Home with Julia"	"tv series"	2011

DBTG query example

- Find all episodes of At Home with Julia
  1. `title.title := "At Home with Julia"; find any title using title;`
  2. `find first episodes within episode_of;`
     a. `get episodes;`
     b. `find next episodes within episode_of;`
     c. Repeat from 2(a) unless there 2(b) didn't find any

How do we know we found something?



214701	"Citizens' Assembly"	1	4	2011
214702	"Code Ranga"	1	2	2011
214703	"Date Night"	1	1	2011
214704	"The Leaker"	1	3	2011

214700	"At Home with Julia"	"tv series"	2011

The end

DBTG Commands

- A few more commands we have not covered.
  - How do we find all titles that are tv series?

```
title.kind := "tv series"
find any/duplicate title using kind;
```

  - How do we find all titles with no restrictions?

```
find first/duplicate title within all_titles;
```

  - How do we find the tv series that an episode belongs to? i.e., navigating from member to owner

```
find owner within episode_of;
```

Try it yourself in this week's recitation.

Quiz time

- Submit your answer in tophat.