CSC343: Introduction to Databases

Winter 2019

Questions?

Summary

Recall …

- **Superkey:** a set of one or more attributes whose combined values are unique
  - Does every relation has a superkey?

- **Key:** the minimal set of one or more attributes whose combined values are unique

$$\text{Movies}(\underline{\text{mID}}, \text{title}, \text{director}, \text{year}, \text{length})$$

  - A key vs coincidence
  - A kind of integrity constraint

Recall …

- **Referential Integrity constraints:** relations refer to each other → restricts the values for the referring attribute

$$R_1[X_1, \ldots, X_N] \subseteq \mathrm{R}_2[\mathrm{Y}_1, \ldots, \mathrm{Y}_\mathrm{N}]$$

- **Foreign Key:** iff $< Y_1, \ldots, Y_N >$ is a key in R_2 → refers to a **single** tuple

$$\mathrm{Roles}(\underline{mID, aID, \mathrm{character}})$$

$$\mathrm{Roles[mID]} \subseteq \mathrm{Movies[mID]}$$

  - mID and aID are NOT keys in the relation Roles

Relational Model Exercises: Qs #1 - 3

Relational Algebra

Relational Algebra

- Queries operate on relations and return relations as a result
  - **Operands:** relation
  - **Operators:** filter, slice, join

Movies

mID	title	director	year	length
1	Shining	Kubrick	1980	146
2	Player	Altman	1992	146
3	Chinatown	Polanski	1974	131
4	Repulsion	Polanski	1965	143
5	Star Wars IV	Lucas	1977	126
6	American Graffiti	Lucas	1973	110
7	Full Metal Jacket	Kubrick	1987	156

Artists

aID	aName	nat
1	Nicholson	American
2	Ford	American
3	Stone	British
4	Fisher	American

Roles

mID	aID	character
1	1	Jack Torrance
3	1	Jake ‘J.J.’ Gittes
1	3	Delbert Grady
5	2	Han Solo
6	2	Bob Falfa
5	4	Princess Leia Organa

Select: filter rows

- Select/filter some rows that satisfy a condition i.e. the result is a **relation** that contains tuples that satisfy some condition

- Notation: $\sigma_c(e)$
  - c: a condition (a Boolean expression)
  - e: an expression

- Example: $\sigma_{\text{length}>130}(\text{Movies})$

Select: filter rows

Example: $\sigma_{\text{length}>130}(\text{Movies})$

mID	title	director	year	length
1	Shining	Kubrick	1980	146
2	Player	Altman	1992	146
3	Chinatown	Polanski	1974	131
4	Repulsion	Polanski	1965	143
~~5~~	~~Star Wars IV~~	~~Lucas~~	~~1977~~	~~126~~
~~6~~	~~American Graffiti~~	~~Lucas~~	~~1973~~	~~110~~
7	Full Metal Jacket	Kubrick	1987	156

Project: select columns

- Select some columns i.e. the result is a **relation** that contains the specified attributes

- Notation: $\pi_L(R)$

  R: a relation

  L: a list of one or more attributes

- Example: $\pi_{\text{title}}(\text{Movies})$

Project: select columns

Example: $\pi_{\text{title,year}}(\text{Movies})$

title	year
Shining	1980
Player	1992
Chinatown	1974
Repulsion	1965
Star Wars IV	1977
American Graffiti	1973
Full Metal Jacket	1987

- Can compose larger expressions
  - Example: $\pi_{\text{title,year}}(\sigma_{\text{length}>130}(\text{Movies}))$

title	year
Shining	1980
Player	1992
Chinatown	1974
Repulsion	1965
Full Metal Jacket	1987

Relational Model Exercises: Qs #4

Join: Cartesian Product

- Notation: $R_1 \times R_2$

R_1

A	B
a1	b1
a2	b2

R_2

B	C
b1	c1
b2	c2

A	B	B	C
a1	b1	b1	c1
a1	b1	b2	c2
a2	b2	b1	c1
a2	b2	b2	c2

- If the number of rows in R_1, R_2, R_3 is 100, 20, 3 respectively, how many rows will $R_1 \times R_2 \times R_3$ have?

- Can compose larger expressions
  - Example: Find all characters in movies with length > 130

- Can compose larger expressions
  - Example: Find all characters in movies with length > 130

$$\pi_{\text{character}}\left(\sigma_{(\text{Movies.mID}=\text{Roles.mID})\wedge(\text{length}>130)}(\text{Movies} \times \text{Roles})\right)$$

Join: Natural Join

- Notation: $R_1 \bowtie R_2$

R_1

A	B
a1	b1
a2	b2

R_2

B	C
b1	c1
b2	c2

A	B	B	C
a1	b1	b1	c1
a1	b1	b2	c2
a2	b2	b1	c1
a2	b2	b2	c2

Join: Natural Join

- Notation: $R_1 \bowtie R_2$

R_1

A	B
a1	b1
a2	b2

R_2

B	C
b1	c1
b2	c2

A	B	B	C
a1	b1	b1	c1
~~a1~~	~~b1~~	~~b2~~	~~c2~~
~~a2~~	~~b2~~	~~b1~~	~~c1~~
a2	b2	b2	c2

Join: Natural Join

- Notation: $R_1 \bowtie R_2$

R_1

A	B
a1	b1
a2	b2

R_2

B	C
b1	c1
b2	c2

A	B	B	C
a1	b1	b1	c1
a2	b2	b2	c2

Join: Natural Join

- Notation: $R_1 \bowtie R_2$

R_1

A	B
a1	b1
a2	b2

R_2

B	C
b1	c1
b2	c2

A	B	C
a1	b1	c1
a2	b2	c2

Join: Theta Join

- Notation: $R_1 \bowtie_{condition} R_2$
- This is equivalent to $\sigma_{condition}(R_1 \times R_2)$

Set Operations

- $R_1 \cup R_2$ → union
- $R_1 \cap R_2$ → intersection
- $R_1 - R_2 \neq R_2 - R_1$ → difference
- Only if R_1 and R_2 have the same attributes (name and order)

Composing large expressions

Precedence Rules:

- $\sigma, \pi > \times, \bowtie > \cap > \cup, -$
- Use brackets

Assignment operators:

- $\mathrm{R} := \text{expression}$
- $\mathrm{R}(\mathrm{A}_1, \ldots, \mathrm{A}_n) := \text{expression}$