Relational Algebra: Derived Operators

Computing on Data

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Kevin C.C. Chang, Professor

Computer Science @ Illinois

Learning Objectives

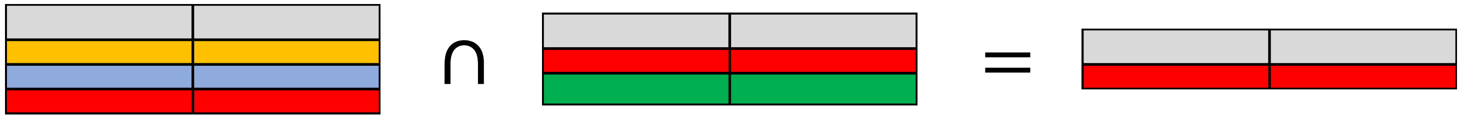
By the end of this video, you will be able to:

- Identify the derived operators of relational algebra.
- Perform computation on relations using basic/derived operators.

Derived Operations

- Some operations are very often used.
- But they can be expressed in terms of the basic operations.
- We thus define ”shorthand” for these operations.
- We will introduce:
  - Set Intersection $R_1 \cap R_2$
  - Theta Join $R_1 \bowtie_\theta R_2$
  - Natural Join $R_1 \bowtie R_2$

Set Intersection



Intersection of two relations

- Notation: $R_1 \cap R_2$
- Input: R_1 and R_2, which have the same schema.
- Output:
  - A relation with all tuples common in R_1 and R_2
  - Schema: same as R_1 and R_2

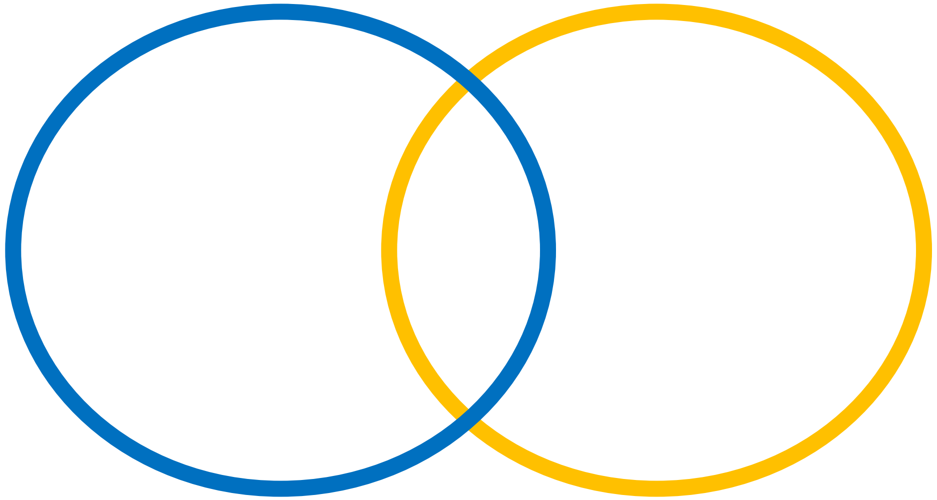
id	number	term	grade
1	411	Fall 2017	A+
4	411	Fall 2017	B
1	426	Fall 2017	A
2	426	Spring 2017	A-
2	225	Spring 2017	B

Example Enrolls relation

- Which students take both 411 and 426?
  - $\pi_{\text{id}}(\sigma_{\text{number}=\text{"411"}}\text{Enrolls}) \cap \pi_{\text{id}}\left(\sigma_{\text{number}=\text{"426"}}\text{Enrolls}\right)$
    $= \{(1),(4)\} \cap \{(1),(2)\} = \{(1)\}.$

Intersection: Derivable from the Basics

- Intersection can be derived from Difference.
- $\boldsymbol{R_1} \cap R_2 = \boldsymbol{R_1} - (\boldsymbol{R_1} - R_2)$



Intersection of two relations

Intersection Examples

- *Q1: Find beers that are drunk and favorited by some drinkers.*
- *Q2: Find ”really-happy drinkers” who have bars on the same streets they live and the bars sell beers that they drink.*

Theta Join

- Notation: $R_1 \bowtie_\theta R_2$, where θ is a condition.
- Input: $R_1(A_1, \ldots, A_n), R_2(B_1, \ldots, B_m)$
- Parameters: θ is a condition over R_1 and/or R_2's attributes.
- Output:
  - A relation of the *product* of R_1 and R_2 *filtered* by condition θ.
  - I.e., can be expressed by $\sigma_\theta(R_1 \times R_2)$.
  - Schema: $(A_1, \ldots, A_n, B_1, \ldots, B_m)$
    - Assume no attribute name clash. Or, use $R_1.A$ vs. $R_2.A$ to distinguish.
- *What courses did Bugs Bunny take?*
  - $\pi_{\text{number}}\ \sigma_{\text{S.id=E.id AND name="Bugs Bunny"}}(\text{Students} \times \text{Enrolls})$
  - $\pi_{\text{number}}\ (\text{Students}\ \bowtie_{\textbf{S.id=E.id AND name="Bugs Bunny"}}\ \text{Enrolls})$

Theta-Join Examples

- *Q1: Find ”really-happy drinkers” who have bars on the same streets they live and the bars sell beer that they drink. Now use theta-joins instead of Cartesian products.*

Natural Join

- Notation: $R_1 \bowtie R_2$
- Input: $R_1(A_1, \ldots, A_n), R_2(B_1, \ldots, B_m)$
- Output:
  - A relation combining all pairs of tuples in R_1 and R_2 that agree on their "common" attributes $\{A_1, \ldots, A_n\} \cap \{B_1, \ldots, B_m\}$, the *join attributes*.
  - I.e., can be expressed by $\sigma_\theta(R_1 \times R_2)$.
  - Schema: $\{A_1, \ldots, A_n\} \cup \{B_1, \ldots, B_m\}$, i.e., merging the two schemas and removing duplicates.
- *What courses did Bugs Bunny take?*
  - $\pi_{\text{number}}\ (\text{Students}\ \bowtie_{\text{S.id=E.id AND name="Bugs Bunny"}}\ \text{Enrolls})$
  - $\pi_{\text{number}}\ (\sigma_{\text{names="Bugs Bunny"}}\ \textbf{Students} \bowtie \textbf{Enrolls})$

id	name	major	birthday
1	Bugs Bunny	CS	2004-11-06
2	Donald Duck	ECE	1997-02-01
3	Peter Pan	SS	1998-10-01
4	Mickey Mouse	Music	1995-04-01

X

id	number	term	grade
1	411	Fall 2017	A+
4	411	Fall 2017	B
1	426	Fall 2017	A

=

S.id	name	major	birthday	E.id	number	term	grade
1	Bugs Bunny	CS	2014-11-06	1	411	Fall 2017	A+
1	Bugs Bunny	CS	2014-11-06	4	411	Fall 2017	B
1	Bugs Bunny	CS	2014-11-06	1	426	Fall 2017	A
2	Donald Duck	ECE	1997-02-01	1	411	Fall 2017	A+
2	Donald Duck	ECE	1997-02-01	4	411	Fall 2017	B
2	Donald Duck	ECE	1997-02-01	1	426	Fall 2017	A
…	…	…	…	…	…	…	…

Cartesian product of two tables

Natural-Join Examples

- *Q1: Find ”really-happy drinkers” who have bars on the same streets they live and the bars sell beer that they drink. Now use natural joins if possible.*

Food for Thought

Joins are expensive. One major advantage that document-model databases claim is -- they do not need joins. Can you explain why?

mongoDB

CouchDB
relax

```
{
  "_id": "<ObjectId1>",
  "name": "Samuel Adams",
  "brewer": {
    "name": "Boston Beer Company",
    "location": "Boston, Massachusetts"
  },
  "alcohol": 4.9,
  "type": "larger",
  "year introduced": 1984,
  "variants": [
    "<ObjectId2>",
    "<ObjectId3>"
  ]
}
```

Document model data