CZ2007 Introduction to Database Systems (Week 6)

Topic 5: Relational Algebra (2)



Dr. Ng Wee Keong Associate Professor

This presentation is copyright property of NTU. It is intended for students of CZ2007 only.



Last Lecture: Relational Algebra

- Given: Two relations $R_1(A, B, C)$, $R_2(A, B, C)$
- Selection: $\sigma_{A>100} R_1$
- Projection: $\Pi_{A, B} R_1$
- Union: $R_1 \cup R_2$
- Intersection: $R_1 \cap R_2$
- Difference: $R_1 R_2$
- Natural Join: $R_1 \bowtie R_2$
- Theta Join: $R_1 \bowtie_{R1.A=R2.A \text{ AND } R1.B < R2.B} R_2$



This Lecture

- Assignment: $T_1 := \sigma_{A > 100} R_1$
- Rename: $\rho_{\text{test}(A', B', C')} R_1$
- Duplicate Elimination δ
- Extended Projection **II**
- \blacksquare Grouping and Aggregation γ

Assignment :=

Quiz1

<u>Name</u>	Score
Alice	70
Bob	90
Cathy	80
David	100

Evaluation1

Name	Score
Alice	70
Bob	90
Cathy	80
David	100

Over85

Name	Score
Bob	90
David	100

- Conceptually: Make another copy of the table and give it a new name
- Example
 - Evaluation1 := Quiz1
 - Over85 := $\sigma_{\text{Score} > 85}$ Quiz1
- Note: All attribute names are retained

Assignment :=

- Useful for breaking down steps
- Example:
 - \square (Π_{Name} Students) \cup (Π_{Name} Volunteer)
- Equivalent Representation
 - \blacksquare R1 := Π_{Name} Students
 - \blacksquare R2 := Π_{Name} Volunteer
 - □ R1 ∪ R2
- This makes your solution easier to write and easier for others to understand



Rename p

Quiz1

<u>Name</u>	Score
Alice	70
Bob	90
Cathy	80
David	100

Evaluation1

Name	Score
Alice	70
Bob	90
Cathy	80
David	100

Eval1

SName	QScore
Alice	70
Bob	90
Cathy	80
David	100

- Similar to assignment, but allows change of attribute names
- Example
 - ρ_{Evaluation1} Quiz1
 - PEval1(SName, QScore) Quiz1

R2

\mathbf{O}	u	i 7	1 2
K	ч	12	

Name Score
Alice 70
Bob 90
Cathy 80
David 100

R1

Name	Score
Cathy	80

Name1	Score1	Name2	Score2
Bob	90	Cathy	80
David	100	Cathy	80

R3

Name1	Score1
Bob	90
David	100

- Find the students who score higher than Cathy in Quiz1
- R1 := $\sigma_{Name='Cathy'}$ Quiz1
- $\rho_{R2(Name1, Score1, Name2, Score2)}$ (Quiz1 $\bowtie_{Quiz1.Score} > R1.Score R1$)
- R3 := $\Pi_{\text{Name1, Score1}}$ R2

Quiz1

Name Score
Alice 70
Bob 90
Cathy 80

R1

<u>Name</u>	Score
Alice	70
Bob	90
Cathy	80

R2

Name1	Score1	Name2	Score2
Alice	70	Bob	90
Alice	70	Cathy	80
Cathy	80	Bob	90



- Find the students who score the highest in Quiz1
- R1 := Quiz1
- P_{R2(Name1, Score1, Name2, Score2)}(Quiz1 ⋈_{Quiz1.Name} <> R1.Name AND Quiz1.Score</sub> < R1.Score R1)</p>
- R3 := Π_{Name1} R2 (students who scored lower than someone else)
- R4 := Π_{Name} Quiz1 R3 (students who did not score lower than someone else)

Quiz1

<u>Name</u>	Score
Alice	70
Bob	90
Cathy	80
David	100

Quiz2

<u>Name</u>	Score
Alice	80
Bob	90
Cathy	90
David	70

<u>Name</u>	Score
Alice	90
Bob	90
Cathy	80
David	70

- Query: "Find the students whose scores in Quizzes 1, 2, and 3 are increasing"
- Quiz1 ⋈_{Quiz1.Name} = Quiz2.Name AND Quiz1.Score < Quiz2.Score Quiz2)
 Quiz2.Name = Quiz3.Name AND Quiz2.Score < Quiz3.Score Quiz3



Duplicate Elimination δ

Purchase

Name	Product	Date
Alice	iPhone	2017.01.01
Bob	Xbox	2017.01.01
Cathy	iPhone	2017.01.01
David	Xbox	2017.02.17

R1

Product
iPhone
Xbox
iPhone

R2

Product iPhone Xbox

- Effect: Eliminate duplicate tuples
- Query: Find the list of products sold on 2017.01.01
- R1 := $\Pi_{Product}$ ($\sigma_{Date='2017.01.01'}$ Purchase)
- $R2 := \delta(R1)$



Extended Projection II

Scores

<u>Name</u>	Quiz1	Quiz2
Alice	70	90
Bob	90	80
Cathy	80	100
David	100	90

Results

<u>Name</u>	Total
Alice	160
Bob	170
Cathy	180
David	190

- Similar to ordinary projection, but allows the creation of new attributes via arithmetic
- Query: "For each student, find his/her total score in Quiz 1 and 2"
- $\Pi_{\text{Name, Quiz1} + \text{Quiz2}} \rightarrow \text{Total}$ Scores
- The left hand side of "→" gives the arithmetic performed
- The right hand side gives an attribute name to the result

Extended Projection II

Scores

<u>Name</u>	Quiz1	Quiz2
Alice	70	90
Bob	90	80
Cathy	80	100
David	100	90

Results

<u>Name</u>	Average
Alice	80
Bob	85
Cathy	90
David	95

- Similar to ordinary projection, but allows the creation of new attributes via arithmetic
- Query: "For each student, find his/her average score in Quiz 1 and 2"
- $\Pi_{\text{Name, (Quiz1 + Quiz2)/2}}$ Average Scores

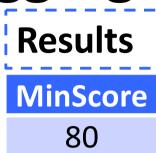


<u>Name</u>	School	Score
Alice	SCSE	90
Bob	EEE	80
Cathy	EEE	100
David	SCSE	90



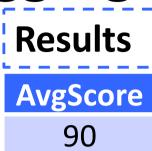
- Query: "Find the highest score in Quiz1"
- $\gamma_{\text{MAX(Score)}} \rightarrow \text{MaxScore} \ \text{Quiz1}$
- The attribute name on right hand side of "→" can be arbitrary

<u>Name</u>	School	Score
Alice	SCSE	90
Bob	EEE	80
Cathy	EEE	100
David	SCSE	90



- Query: "Find the lowest score in Quiz1"
- $\gamma_{MIN(Score)} \rightarrow MinScore Quiz1$

<u>Name</u>	School	Score
Alice	SCSE	90
Bob	EEE	80
Cathy	EEE	100
David	SCSE	90



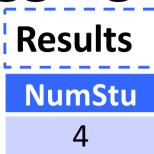
- Query: "Find the average score in Quiz1"
- $\gamma_{\text{AVG(Score)}} \rightarrow \text{AvgScore} \ \text{Quiz1}$

<u>Name</u>	School	Score
Alice	SCSE	90
Bob	EEE	80
Cathy	EEE	100
David	SCSE	90



- Query: "Find the sum of scores in Quiz1"
- $\gamma_{\text{SUM(Score)}} \rightarrow \text{SumScore} \ \text{Quiz1}$

<u>Name</u>	School	Score
Alice	SCSE	90
Bob	EEE	80
Cathy	EEE	100
David	SCSE	90



- Query: "Find the number of students in Quiz1"
- $\gamma_{\text{COUNT(Name)}} \rightarrow \text{NumStu} \text{Quiz1}$
- $\gamma_{\text{COUNT(School)}} \rightarrow \text{NumStu} \ \text{Quiz1}$
- $\gamma_{\text{COUNT(Score)}} \rightarrow \text{NumStu} \ \text{Quiz1}$
- All three queries above give the number of tuples in Quiz1

Aggregate Functions

```
MAX( ... )
```

- MIN(...)
- AVG(...)
- **SUM(...)**
- COUNT(...)

Grouping and Aggregation γ

<u>Name</u>	School	GPA
Alice	SCSE	4
Bob	EEE	3
Cathy	EEE	3.4
David	SCSE	3.6

Results	
School	AvgGPA
SCSE	3.8
EEE	3.2

- Query: "Find the average GPA in each school"
- $\gamma_{\text{School, AVG(GPA)}} \rightarrow \text{AvgGPA} \text{ Quiz } 1$
- Effect: Divide tuples into separate groups based on their "School" value, and then compute the average GPA in each group

Quiz1

<u>Name</u>	School	GPA
Alice	SCSE	4
Bob	EEE	3
Cathy	EEE	3.4
David	SCSE	3.6

School	AvgGPA	MaxGPA
SCSE	3.8	4
EEE	3.2	3.4

Results

- Query: "Find the average GPA and highest GPA in each school"
- $^{\circ}$ Yschool, AVG(GPA) \rightarrow AvgGPA, MAX(GPA) \rightarrow MaxGPA Quiz1



Grouping and Aggregation γ

Quiz1

<u>Name</u>	School	Year	GPA
Alice	SCSE	3	4
Bob	EEE	1	3
Cathy	EEE	2	3.4
David	SCSE	3	3.6

Results

School	Year	GPA
SCSE	3	3.8
EEE	1	3
EEE	2	3.4

- $\gamma_{\text{School, Year, AVG(GPA)}} \rightarrow AvgGPA Quiz1$
- Effect: Divide tuples into separate groups based on their "School, year" value combination, and then compute the average GPA in each group



Example

R2

Q	uiz	1
	••••	

R1

Name	Score	MaxScore
David	100	100

Name Score
Alice 70
Bob 90
Cathy 80
David 100

MaxScore 100

R3

Name David

- Query: "Find the student that scores the highest in Quiz1"
- $\sigma_{Score = MAX(Score)}$ Quiz1?
- Wrong: Aggregate functions can only be used with the aggregation operation γ
- R1 := $\gamma_{MAX(Score)} \rightarrow MaxScore(Quiz1)$
- R2 := Quiz1 ⋈_{Score} = MaxScore R1
- R3 := $\Pi_{Name}(R2)$



Quiz1

Name Score
Alice 70
Bob 90
Cathy 80
David 100

R1

MaxScore 100 **R2**

Name	Score	MaxScore
David	100	100

R3

Name Score
David 100

- Query: "Find the student that scores the second highest in Quiz1"
- R1 := $\gamma_{MAX(Score)} \rightarrow MaxScore(Quiz1)$
- R2 := Quiz1 ⋈_{Score} = MaxScore</sub> R1
- R3 := $\Pi_{\text{Name, Score}}(R2)$
- R4 := Quiz1 R3
- R5 := $\gamma_{MAX(Score)} \rightarrow 2ndMaxScore(R4)$
- R6 := R4 $\bowtie_{Score} = 2ndMaxScore$ R5

Quiz1

Students

R1

Name	Score	School
Alice	70	SCSE
Bob	90	EEE
Cathy	80	EEE
David	100	SCSE

<u>Name</u>	Score
Alice	70
Bob	90
Cathy	80
David	100

<u>Name</u>	School
Alice	SCSE
Bob	EEE
Cathy	EEE
David	SCSE

- Query: "For each school, find the student that scores the highest in Quiz1"
- R1 := Quiz1 ⋈ Student
- R2 := $\gamma_{School, MAX(Score)} \rightarrow MaxScore(R1)$

Quiz1

Students

R2

School	MaxScore
SCSE	100
EEE	90

<u>Name</u>	Score
Alice	70
Bob	90
Cathy	80
David	100

Name School
Alice SCSE
Bob EEE
Cathy EEE
David SCSE

- Query: "For each school, find the student that scores the highest in Quiz1"
- R1 := Quiz1 ⋈ Student
- R2 := $\gamma_{School, MAX(Score)} \rightarrow MaxScore(R1)$
- R3 := R1 ⋈_{R1.School} = R2.School AND Score = MaxScore R2



Quiz1

Students

R3	
----	--

<u>Name</u>	Score	School	MaxScore
Bob	90	EEE	90
David	100	SCSE	100

<u>Name</u>	Score
Alice	70
Bob	90
Cathy	80
David	100

Name School
Alice SCSE
Bob EEE
Cathy EEE
David SCSE

- Query: "For each school, find the student that scores the highest in Quiz1"
- R1 := Quiz1 ⋈ Student
- R2 := $\gamma_{School, MAX(Score)} \rightarrow MaxScore(R1)$
- R3 := R1 ⋈_{R1.School} = R2.School AND Score = MaxScore R2
- R3 := $\Pi_{\text{Name, Score}}(R3)$

Grades

<u>Name</u>	Course	Grade
Alice	DB	Α
Alice	DM	С
Bob	DB	В
Bob	NN	В
Cathy	SP	В
Cathy	NN	Α

CrsSch

Course	School
DB	SCSE
DM	SCSE
NN	EEE
SP	EEE

- Query: "Find the students who have taken all courses from SCSE"
- R1 := $\sigma_{School} = 'SCSE'$ CrsSch
- R2 := Grades ⋈ R1
- R3 := $\gamma_{COUNT(Course) \rightarrow ScseCNT}$ (R1)
- R4 := $\gamma_{\text{Name, COUNT(Course)}} \rightarrow \text{CrsCNT}$ (R2)
- R5 := R4 \bowtie_{CrsCNT} = ScseCNT R3
- R6 := Π_{Name} (R5)



Grades

<u>Name</u>	Course	Grade
Alice	DB	Α
Alice	DM	С
Bob	DB	В
Bob	NN	В
Cathy	SP	В
Cathy	NN	А

CrsSch

Course	School
DB	SCSE
DM	SCSE
NN	EEE
SP	EEE

- Query: "For each school, find the students who have taken all courses in the school"
- R1 := Grades ⋈ CrsSch
- R2 := $\gamma_{School, COUNT(Course) \rightarrow CrsCNT}$ (CrsSch)
- R3 := $\gamma_{\text{Name, School, COUNT(Course)}} \rightarrow \text{CrsCNT}$ (R1)
- R4 := R2 ⋈_{R2.School} = R3.School AND R2.CrsCNT = R3.CrsCNT R3



Next lecture:

Topic 5: Relational Algebra (3)

