# CZ2007 Tutorial 5: Relational Algebra

Week 7

# Tutorial 5

## Relational Algebra

- Selection:
  - Query: "Find the students who are either in SCSE or under 21"
  - $\sigma_{\text{School = 'SCSE' OR Age < 21}}$ Students
- Projection:
  - Query: "Find the IDs and Names of all students"
  - $\Pi_{\text{ID, Name}}$ Students
- Union:
  - Query: "Find the persons who are either students or volunteers"
  - Students $\cup$ Volunteer
- Intersection:
  - Query: "Find the persons who are both students and volunteers"
  - Students $\cap$ Volunteer
- Difference:
  - Query: "Find the persons who are students but not volunteers"
  - Students – Volunteer
- Natural Join:
  - Students $\bowtie$ Donations
  - Meaning: "For those students who have made donation, find their names, schools, and amounts of their donations"
- Theta Join:
  - Students $\bowtie_{\text{Sname=Name}}$ Donations
  - Difference from natural join: Duplicate attributes will NOT be removed from the results
  - Query: "Find the students who score higher in quiz 2 than quiz 1"
  - Quiz1 $\bowtie_{\text{Quiz1.Name = Quiz2.Name AND Quiz1.Score < Quiz2.Score}}$ Quiz2

- Assignment $T_1 := \sigma_{A > 100} R_1$

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

- Rename: $\rho_{test(A', B', C')} R_1$

- Similar to assignment, but allows change of attribute names
- Example
  - $\rho_{Evaluation1}$ Quiz1
  - $\rho_{Eval1(SName, QScore)}$ Quiz1

- Duplicate Elimination $\delta$

- 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 $\Pi$

- 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"
- $\Pi_{Name,\ Quiz1 + Quiz2 \rightarrow Total}$ Scores
- The left hand side of "→" gives the arithmetic performed
- The right hand side gives an attribute name to the result

- Grouping and Aggregation $\gamma$
  - MAX( … )
  - MIN( … )
  - AVG( … )
  - SUM( … )
  - COUNT( … )

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

- Division: $\div$
  - Query: "Find each person that owns all Apple products"
  - Owns $\div$ AppleP
  - In general, $R_1(A, B) \div R_2(B)$ returns a table that contains only A

# Question 1

Consider a database with three tables as follows:

Shopper(**shopperName**, street, ageGroup)
Mall(**mallName**, street)
ShopAt(**shopperName, mallName, date, time**, dayOfWeek)

The three tables record information about shoppers, shopping malls, as well as "which shoppers shop at which malls". Primary Keys are in **bold**.

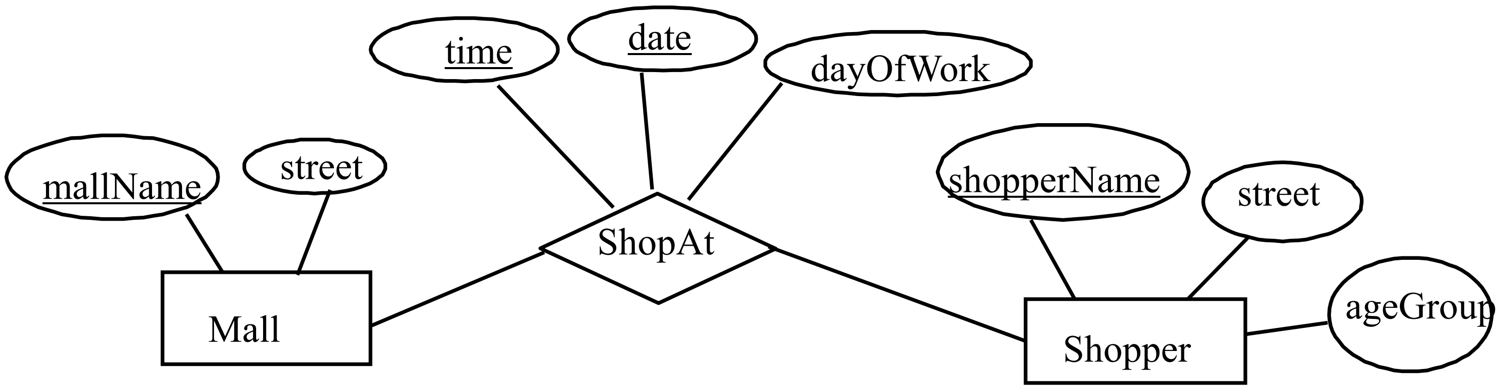
Write the following queries in relational algebra. When answering relational algebra queries, breakdown your answers into intermediate steps. When answering relational algebra queries, each answer should be in the form of one single table containing only relevant output attributes. If you think a question's solution cannot be expressed in relational algebra, explain why.

# Question 1

Shopper(**shopperName**, street, ageGroup)

Mall(**mallName**, street)

ShopAt(<u>**shopperName, mallName, date, time**</u>, dayOfWeek)



# Question 1

Shopper(**shopperName**, street, ageGroup)
Mall(**mallName**, street)
ShopAt(**shopperName, mallName, date, time**, dayOfWeek)

**Find those shopper(s) who shopped at all the malls on "Nanyang Ave" every Thursday between 10am to 5pm, and find the streets that these shoppers live in.**

- R1 := $\Pi_{\text{mallName}}$ ($\sigma_{\text{street = 'Nanyang Ave'}}$ **Mall**)
- R2 := $\sigma_{\text{dayOfWeek = 'Thursday' and time >= 10am and time <= 5pm}}$ (**ShopAt**)
- R3 := $\Pi_{\text{shopperName, mallName}}$ (**R2**)
- R4 := **R3** ÷ **R1**
- R5 := R4 ⋈ Shopper
- Answer: $\Pi_{\text{shopperName, street}}$ (R5)

# Question 2

Shopper(**shopperName**, street, ageGroup)

Mall(**mallName**, street)

ShopAt(**shopperName, mallName, date, time**, dayOfWeek)

**Find the age groups of those shopper(s) who only shop at malls that are located on the street where he/she lives.**

- $R1 := \Pi_{\text{shopperName, mallName, Shopper.street}}(\text{Shopper} \bowtie_{\text{Shopper.shopperName=ShopAt.shopperName}} \text{ShopAt})$
- $R2 := \Pi_{\text{shopperName, mallName, Mall.street}}(\text{Mall} \bowtie_{\text{Mall.mallName=ShopAt.mallName}} \text{ShopAt})$
- $\rho_{\text{R3(shopperName, mName, sStreet)}}(\text{R1})$
- $\rho_{\text{R4(shopperName, mName, mStreet)}}(\text{R2})$
- $R5 := \Pi_{\text{shopperName}}(\text{R3} \bowtie_{\text{R3.shopperName=R4.shopperName and R3.mName=R4.mName and sStreet<>mStreet}} \text{R4})$
- $R6: \Pi_{\text{shopperName}}(\text{Shopper}) - R5$
- $R7 := R6 \bowtie \text{Shopper}$
- Answer: $\Pi_{\text{shopperName, ageGroup}}(\text{R7})$

# Question 3

Shopper(**shopperName**, street, ageGroup)

Mall(**mallName**, street)

ShopAt(**shopperName, mallName, date, time**, dayOfWeek)

**Consider Jurong Point Mall, the shopping mall that is 3.5km south of NTU. Find those shoppers who have shopped there more times than anyone else does. Also find out these shoppers' age groups.**

- R1 := $\sigma_{\text{mallName = 'Jurong Point'}}$ ShopAt
- R2 := $\gamma_{\text{shopperName, COUNT(date)} \rightarrow \text{VisitCount}}$ R1
- R3 := $\gamma_{\text{MAX(VisitCount)} \rightarrow \text{MaxVisitCount}}$ R2
- R4 := $\Pi_{\text{shopperName}}$ (R2 $\bowtie_{\text{VisitCount = MaxVisitCount}}$ R3)
- Result := $\Pi_{\text{shopperName, ageGroup}}$ (Shopper $\bowtie$ R4)

# Question 4

Shopper(**shopperName**, street, ageGroup)

Mall(**mallName**, street)

ShopAt(**shopperName, mallName, date, time**, dayOfWeek)

**Consider Jurong Point Mall, the shopping mall that is 3.5km south of NTU. Find those shoppers in the 20s-30s age group who have never shopped at Jurong Point Mall on Friday evenings between 7pm to 10pm. Also find out which streets these shoppers live in.**

- R1 := $\sigma_{\text{mallName = 'Jurong Point' and dayOfWeek = 'Friday' and time >= 7pm and time <= 10pm}}$ (ShopAt)
- R2 := $\Pi_{\text{shopperName}}$ (R1)
- R3 := $\sigma_{\text{ageGroup = '20s-30s'}}$ (Shopper)
- R4 := $\Pi_{\text{shopperName}}$ (R2 ⋈ R3)
- R5 := ($\Pi_{\text{shopperName}}$ Shopper) – R4
- Result := $\Pi_{\text{shopperName, street}}$ (Shopper ⋈ R5)

# Question 5

Shopper(**shopperName**, street, ageGroup)

Mall(**mallName**, street)

ShopAt(**shopperName, mallName, date, time**, dayOfWeek)

**Find shopping malls that have never been visited by shoppers in the 40s-50s age group on Wednesday mornings between 9am to 11am. Also find out which streets these malls are located.**

- R1 := $\sigma_{\text{dayOfWeek = 'Wednesday' and time >= 9am and time <= 11am}}$ (ShopAt)
- R2 := $\sigma_{\text{ageGroup = '40s-50s'}}$ (Shopper)
- R3 := $\Pi_{\text{mallName}}$ (R1 ⋈ R2)
- R4 := ($\Pi_{\text{mallName}}$ Mall) – R3
- Result := $\Pi_{\text{mallName, street}}$ (Mall ⋈ R4)

# Question 6

Shopper(**shopperName**, street, ageGroup)

Mall(**mallName**, street)

ShopAt(**shopperName, mallName, date, time**, dayOfWeek)

**For each shopper, find how many other shoppers shopped at the same malls as him/her on the same date.**

- $\rho_{R1(s1,\ mall,\ date)}\ (\Pi_{shopperName,\ mallName,\ date}\ ShopAt)$
- $\rho_{R2(s2,\ mall,\ date)}\ (\Pi_{shopperName,\ mallName,\ date}\ ShopAt)$
- $R3 := R1 \bowtie_{s1 <> s2 \text{ and } R1.mall=R2.mall \text{ and } R1.date=R2.date} R2$
- $R4 := \Pi_{s1,\ s2}\ (R3)$
- $Result := \gamma_{s1,\ COUNT(s2) \rightarrow numS2}\ R4$

# Question 7

Shopper(**shopperName**, street, ageGroup)

Mall(**mallName**, street)

ShopAt(**shopperName, mallName, date, time**, dayOfWeek)

**Find the mall(s) that is/are shopped by the largest number of repeat shoppers in the 20s-30s age group. Repeat shoppers of a mall are shoppers who have shopped at least once in the mall.**

- R1 := $\gamma_{\text{mallName, shopperName, COUNT(date)} \rightarrow \text{numTimes}}$(ShopAt)
- R2 := $\sigma_{\text{numTimes} > 1}$(R1)
- $\rho_{\text{RepeatShoppers}(\textbf{mallName}, \textbf{shopperName}, \text{numTimes})}$ R2
- R3 := $\sigma_{\text{ageGroup} = \text{"20s-30s"}}$ (Shopper ⋈ RepeatShoppers)
- R4 := $\gamma_{\text{mallName, COUNT(shopperName)} \rightarrow \text{NumShoppers}}$(R3)
- R5 := $\gamma_{\text{MAX(NumShoppers)} \rightarrow \text{MaxNum}}$(R4)
- Result: $\Pi_{\text{mallName}}$ (R4 $\bowtie_{\text{NumShoppers} = \text{MaxNum}}$ R5)

# Additional Exercises

# Question 1

A library database schema contains the following tables:

LIB-MEMBER(<u>ID</u>, name, age)

BOOK(<u>serial#</u>, title, author, year-of-publication)

LOAN(<u>ID, serial#, date-due</u>)

State what each of the following relational algebra queries is looking for:

a) $\pi_{name}((\sigma_{year\text{-}of\text{-}publication<1960}\ BOOK \bowtie LOAN) \bowtie LIB\text{-}MEMBER)$

b) $\pi_{ID}(\sigma_{age<21}\ LIB\text{-}MEMBER) - \pi_{ID}(\sigma_{author=\text{“J.K.Rowling”}}\ BOOK \bowtie LOAN)$

c) $\pi_{name}((\pi_{ID,serial\#}\ LOAN \div \pi_{serial\#}(\sigma_{title\ like\ \text{‘C Progamming’}}\ BOOK)) \bowtie LIB\text{-}MEMBER)$

# Question 1(a)

LIB-MEMBER(<u>ID</u>, name, age)

BOOK(<u>serial#</u>, title, author, year-of-publication)

LOAN(<u>ID, serial#, date-due</u>)

a) $\pi_{name}((\sigma_{year\text{-}of\text{-}publication<1960}\ BOOK \bowtie LOAN) \bowtie LIB\text{-}MEMBER)$

Find names of members who have loaned books published before 1960.

# Question 1(b)

LIB-MEMBER(<u>ID</u>, name, age)

BOOK(<u>serial#</u>, title, author, year-of-publication)

LOAN(<u>ID, serial#, date-due</u>)

a) $\pi_{ID}(\sigma_{age<21}\ \text{LIB-MEMBER}) - \pi_{ID}(\sigma_{author=\text{“J.K.Rowling”}}\ \text{BOOK} \bowtie \text{LOAN})$

Find IDs of members under the age of 21 who have not loaned a book by author “J.K.Rowling”.

# Question 1(c)

LIB-MEMBER(<u>ID</u>, name, age)

BOOK(<u>serial#</u>, title, author, year-of-publication)

LOAN(<u>ID, serial#, date-due</u>)

a) $\pi_{name}((\pi_{ID,serial\#}\ LOAN \div \pi_{serial\#}(\sigma_{title\ like\ \text{'C Progamming'}}\ BOOK)) \bowtie LIB\text{-}MEMBER)$

Find names of members who have loaned all books with the title 'C Programming'

# Question 2

The schema of a database containing university-type data is given below. Primary key is underlined for each relation.

STUDENT(<u>Sid</u>, Sname, Sex, Age, Year, GPA)

DEPT(<u>Dname</u>, NumPhds)

PROF(<u>Pname</u>, Dname)

MAJOR(<u>Dname, Sid</u>)

COURSE(<u>Dname, C#</u>, Cname)

SECTION(<u>Dname, C#, Sect#</u>, Pname)

ENROLL(<u>Sid, Dname, C#, Sect#</u>, Grade)

# Question 2

Write the following queries in relational algebra.

a) Find the names of professors who work in departments that have fewer than 50 PhD students.

b) Find the name(s) of student(s) with the lowest GPA.

c) Find the names and majors of students who have taken the ‘Database System’ course.

d) Find the ids, names, and GPAs of the students who have taken all courses from the ‘Civil Engineering’ department.

# Question 2(a)

STUDENT(Sid, Sname, Sex, Age, Year, GPA)

DEPT(Dname, NumPhds)

PROF(Pname, Dname)

MAJOR(Dname, Sid)

COURSE(Dname, C#, Cname)

SECTION(Dname, C#, Sect#, Pname)

ENROLL(Sid, Dname, C#, Sect#, Grade)

Find the names of professors who work in departments that have fewer than 50 PhD students.

$$R1 := \sigma_{NumPhds < 50} (DEPT)$$

$$Answer := \pi_{Pname} (PROF \bowtie R1)$$

# Question 2(b)

STUDENT(<u>Sid</u>, Sname, Sex, Age, Year, GPA)

DEPT(<u>Dname</u>, NumPhds)

PROF(<u>Pname</u>, Dname)

MAJOR(<u>Dname, Sid</u>)

COURSE(<u>Dname, C#</u>, Cname)

SECTION(<u>Dname, C#, Sect#</u>, Pname)

ENROLL(<u>Sid, Dname, C#, Sect#</u>, Grade)

Find the name(s) of student(s) with the lowest GPA.

$$R1 := \gamma_{MIN(GPA) \rightarrow MinGPA} (STUDENT)$$

$$R2 := STUDENT \bowtie_{STUDENT.GPA=R1.MinGPA} R1$$

$$Answer := \pi_{Sname} (R2)$$

# Question 2(c)

STUDENT(<u>Sid</u>, Sname, Sex, Age, Year, GPA)

DEPT(<u>Dname</u>, NumPhds)

PROF(<u>Pname</u>, Dname)

MAJOR(<u>Dname, Sid</u>)

COURSE(<u>Dname, C#</u>, Cname)

SECTION(<u>Dname, C#, Sect#</u>, Pname)

ENROLL(<u>Sid, Dname, C#, Sect#</u>, Grade)

Find the names and majors of students who have taken the ‘Database Systems’ course.

$$R1 := \pi_{\text{Dname, Cno}} (\sigma_{\text{Cname='Database Systems'}}\ \text{COURSE})$$

$$R2 := \pi_{\text{Sid}} (R1 \bowtie \text{ENROLL})$$

$$\text{Answer} := \pi_{\text{Sname, Dname}} (R2 \bowtie \text{MAJOR} \bowtie \text{STUDENT})$$

# Question 2(d)

STUDENT(<u>Sid</u>, Sname, Sex, Age, Year, GPA)

DEPT(<u>Dname</u>, NumPhds)

PROF(<u>Pname</u>, Dname)

MAJOR(<u>Dname, Sid</u>)

COURSE(<u>Dname, C#</u>, Cname)

SECTION(<u>Dname, C#, Sect#</u>, Pname)

ENROLL(<u>Sid, Dname, C#, Sect#</u>, Grade)

Find the ids, names, and GPAs of the students who have taken all courses from the ‘Civil Engineering’ department.

$$R1 := \pi_{\text{Dname, Cno}} (\sigma_{\text{Dname = ‘Civil Engineering’}} \text{COURSE})$$

$$R2 := \pi_{\text{Sid, Dname, Cno}} \text{ENROLL}$$

$$R3 := R2 \div R1$$

$$\text{Answer} := \pi_{\text{Sid, Sname, GPA}} (R3 \bowtie \text{STUDENT})$$

# Question 3(a)

Find the names of the players who won at least one gold and one silver.

PLAYERS(<u>player-id</u>, name, countryname, age)

EVENTS(<u>event-id</u>, name, eventtype)

RESULTS(<u>player-id</u>, <u>event-id</u>, medal)

$$R1 := \pi_{\text{player-id}} (\sigma_{\text{medal = 'gold'}} \text{ RESULTS})$$

$$R2 := \pi_{\text{player-id}} (\sigma_{\text{medal = 'silver'}} \text{ RESULTS})$$

$$R3 := R2 \cap R1$$

$$\text{Answer} := \pi_{\text{name}} (R3 \bowtie \text{PLAYERS})$$

# Question 3(b)

Find the players who did not win a medal.

PLAYERS(<u>player-id</u>, name, countryname, age)

EVENTS(<u>event-id</u>, name, eventtype)

RESULTS(<u>player-id</u>, <u>event-id</u>, medal)

$$R1 := \pi_{\text{player-id}}\ \text{PLAYERS}$$

$$R2 := \pi_{\text{player-id}}\ \text{RESULTS}$$

$$R3 := R1 - R2$$

$$\text{Answer} := \pi_{\text{name}}\ (R3 \bowtie \text{PLAYERS})$$