Content Recap

DBI - Databases and Interfaces
Dr Matthew Pike & Prof Linlin Shen

Relational Algebra

Relational Algebra - Feedback

- “rename aspect with sigma and Pi in relational algebra”
- “I think the one we need to cover the most is relational algebra (symbol to English statement and English statement to symbol) the one in the LAB2”

Pairs of Universities in same County

Step 1

- $\sigma_{c1=c2}$ (ρU1(n1, c1, e1)(University) x ρU2(n2, c2, e2)(University))
- or
- ρU1(n1, c, e1)(University) $\bowtie$ ρU2(n2, c, e2)(University)

Step 2

- $\sigma_{n1 \neq n2}$ (ρU1(n1, c, e1)(University) $\bowtie$ ρU2(n2, c, e2)(University))

Step 3

- $\sigma_{n1>n2}$ (ρU1(n1, c, e1)(University) $\bowtie$ ρU2(n2, c, e2)(University))

University

uName	County	Enrollment
NOTT	Nott/shire	18000
CAM	Cam/shire	22000
UCL	Great/Lon	20000

Stepping back

- Are there any tuples that satisfy this query?

University		
uName	**County**	**Enrollment**
NOTT	Nott/shire	18000
CAM	Cam/shire	22000
UCL	Great/Lon	20000

Add a new one entry

University

uName	County	Enrollment
NOTT	Nott/shire	18000
CAM	Cam/shire	22000
UCL	Great/Lon	20000
TRENT	Nott/shire	10000

Step 1:

$\rho U1(n1, c, e1)(\text{University})$

$\bowtie$

$\rho U2(n2, c, e2)(\text{University})$

Step 1:
ρU1(n1, c, e1)(University)
⋈
ρU2(n2, c, e2)(University)

U1		
n1	**c**	**e1**
NOTT	Nott/shire	18000
CAM	Cam/shire	22000
UCL	Great/Lon	20000
TRENT	Nott/shire	10000

U2		
n2	**c**	**e2**
NOTT	Nott/shire	18000
CAM	Cam/shire	22000
UCL	Great/Lon	20000
TRENT	Nott/shire	10000

Step 1:

$\rho U1(n1, c, e1)(\text{University})$

$\bowtie$

$\rho U2(n2, c, e2)(\text{University})$

n1	c	e1	n2	e2
NOTT	Nott/shire	18000	NOTT	18000
CAM	Cam/shire	22000	CAM	22000
UCL	Great/Lon	20000	UCL	20000
TRENT	Nott/shire	10000	TRENT	10000

Is this correct/complete?

Step 1:
$\rho U1(n1, c, e1)(\text{University})$
$\bowtie$
$\rho U2(n2, c, e2)(\text{University})$

n1	n2	c	e1	e2
NOTT	NOTT	Nott/shire	18000	18000
NOTT	Trent	Nott/shire	18000	10000
Trent	NOTT	Nott/shire	10000	18000
Trent	Trent	Nott/shire	10000	10000
CAM	CAM	Cam/shire	22000	22000
UCL	UCL	Great/Lon	20000	20000

We can reorganize the ordering for clarity.

The values themselves remain unchanged.

Step2: $\sigma_{n1 \neq n2}$ $(\rho U1(n1, c, e1)(University)$ $\bowtie$ $\rho U2(n2, c, e2)(University))$

n1	n2	c	e1	e2
~~NOTT~~	~~NOTT~~	~~Nott/shire~~	~~18000~~	~~18000~~
NOTT	Trent	Nott/shire	18000	10000
Trent	NOTT	Nott/shire	10000	18000
~~Trent~~	~~Trent~~	~~Nott/shire~~	~~10000~~	~~10000~~
~~CAM~~	~~CAM~~	~~Cam/shire~~	~~22000~~	~~22000~~
~~UCL~~	~~UCL~~	~~Great/Lon~~	~~20000~~	~~20000~~

Step2: $\sigma_{n1 \neq n2}$ $(\rho U1(n1, c, e1)(University)$ $\bowtie$ $\rho U2(n2, c, e2)(University))$

n1	n2	c	e1	e2
~~NOTT~~	~~NOTT~~	~~Nott/shire~~	~~18000~~	~~18000~~
NOTT	Trent	Nott/shire	18000	10000
Trent	NOTT	Nott/shire	10000	18000
~~Trent~~	~~Trent~~	~~Nott/shire~~	~~10000~~	~~10000~~
~~CAM~~	~~CAM~~	~~Cam/shire~~	~~22000~~	~~22000~~
~~UCL~~	~~UCL~~	~~Great/Lon~~	~~20000~~	~~20000~~

Same Universities!

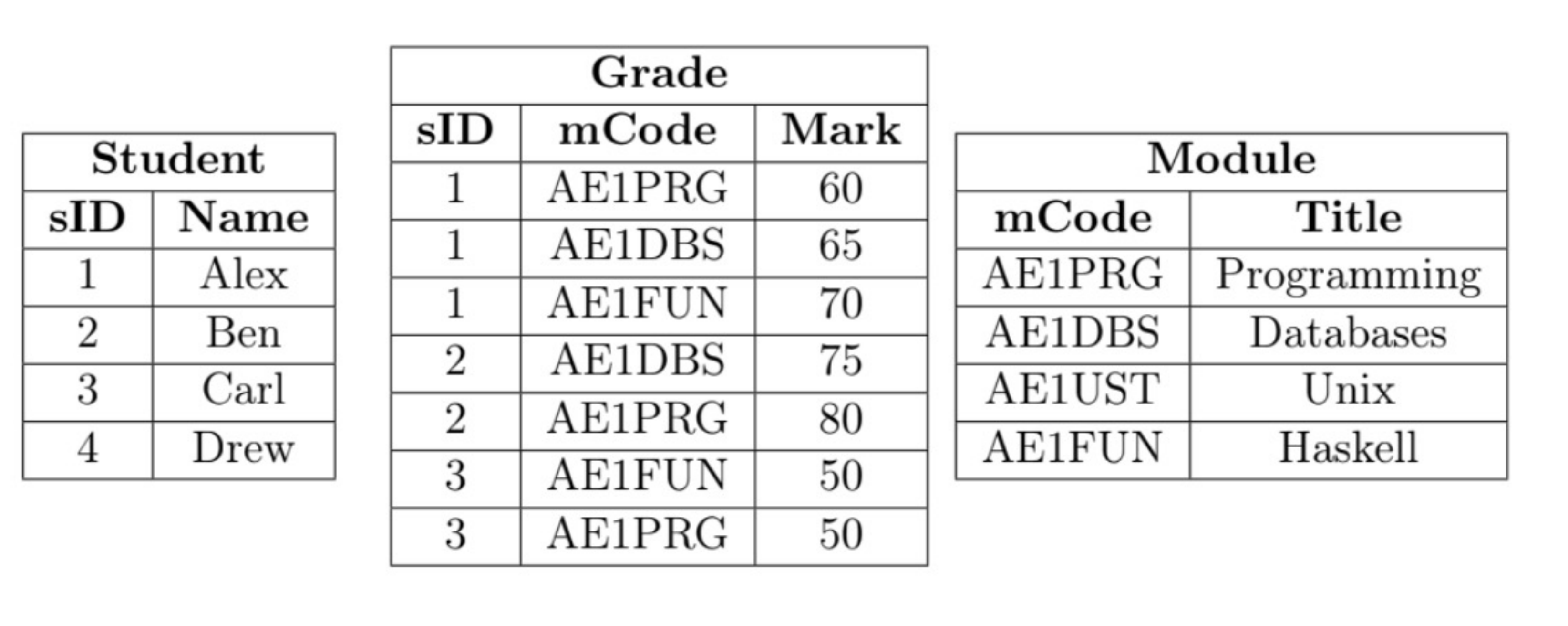
Step 3: $\sigma_{n1>n2}$ ($\rho U1(n1, c, e1)(University)$ $\bowtie$ $\rho U2(n2, c, e2)(University)$)

n1	n2	c	e1	e2
~~NOTT~~	~~Trent~~	~~Nott/shire~~	~~18000~~	~~10000~~
Trent	NOTT	Nott/shire	10000	18000

Translating between Forms

- Solutions to lab 002 are available on Moodle
- Instead, we'll focus on translating between different forms.

Consider the following relations



Student	
sID	**Name**
1	Alex
2	Ben
3	Carl
4	Drew

Grade		
sID	**mCode**	**Mark**
1	AE1PRG	60
1	AE1DBS	65
1	AE1FUN	70
2	AE1DBS	75
2	AE1PRG	80
3	AE1FUN	50
3	AE1PRG	50

Module	
mCode	**Title**
AE1PRG	Programming
AE1DBS	Databases
AE1UST	Unix
AE1FUN	Haskell

SQL -> Relational

Translate the following SQL query into relational algebra (using only σ, π, × operators):

```
SELECT Name, Title, Mark
FROM Student
NATURAL JOIN Grade NATURAL JOIN Module;
```

- Divide and Conquer!

SQL -> Relational (1)

1. Begin with SELECT

SELECT Name, Title, Mark

=>

$\pi_{\text{Name, Title, Mark}}$

SQL -> Relational (2)

FROM Student

=>

$\pi_{\text{Name, Title, Mark}}{}^{\text{(Student)}}$

SQL -> Relational (3)

```
NATURAL JOIN Grade NATURAL JOIN Module;
```

=>

$\pi_{Name, Title, Mark}(Student \bowtie Grade \bowtie Module)$

=>

```
SELECT Name, Title, Mark
FROM Student
NATURAL JOIN Grade NATURAL JOIN Module;
```

SQL -> Relational (4)

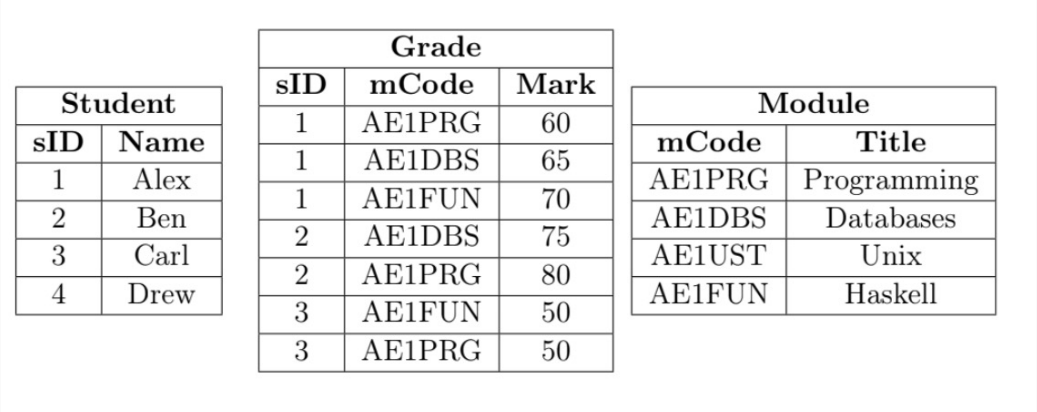
- Remember ->
  - ‘(using only σ, π, × operators)’:
- Also remember
  - “Natural join does not add expressive power to Relational Algebra, just facilitates the writing of complex queries”
- How can we translate Natural Join to one of σ, π, × operators?

SQL -> Relational (5)

Student ⋈ Grade

=>

$$\sigma_{\text{Student.sID = Grade.sID}}(\text{Student x Grade})$$



Student	
sID	**Name**
1	Alex
2	Ben
3	Carl
4	Drew

Grade		
sID	**mCode**	**Mark**
1	AE1PRG	60
1	AE1DBS	65
1	AE1FUN	70
2	AE1DBS	75
2	AE1PRG	80
3	AE1FUN	50
3	AE1PRG	50

Module	
mCode	**Title**
AE1PRG	Programming
AE1DBS	Databases
AE1UST	Unix
AE1FUN	Haskell

SQL -> Relational (6)

Finally....

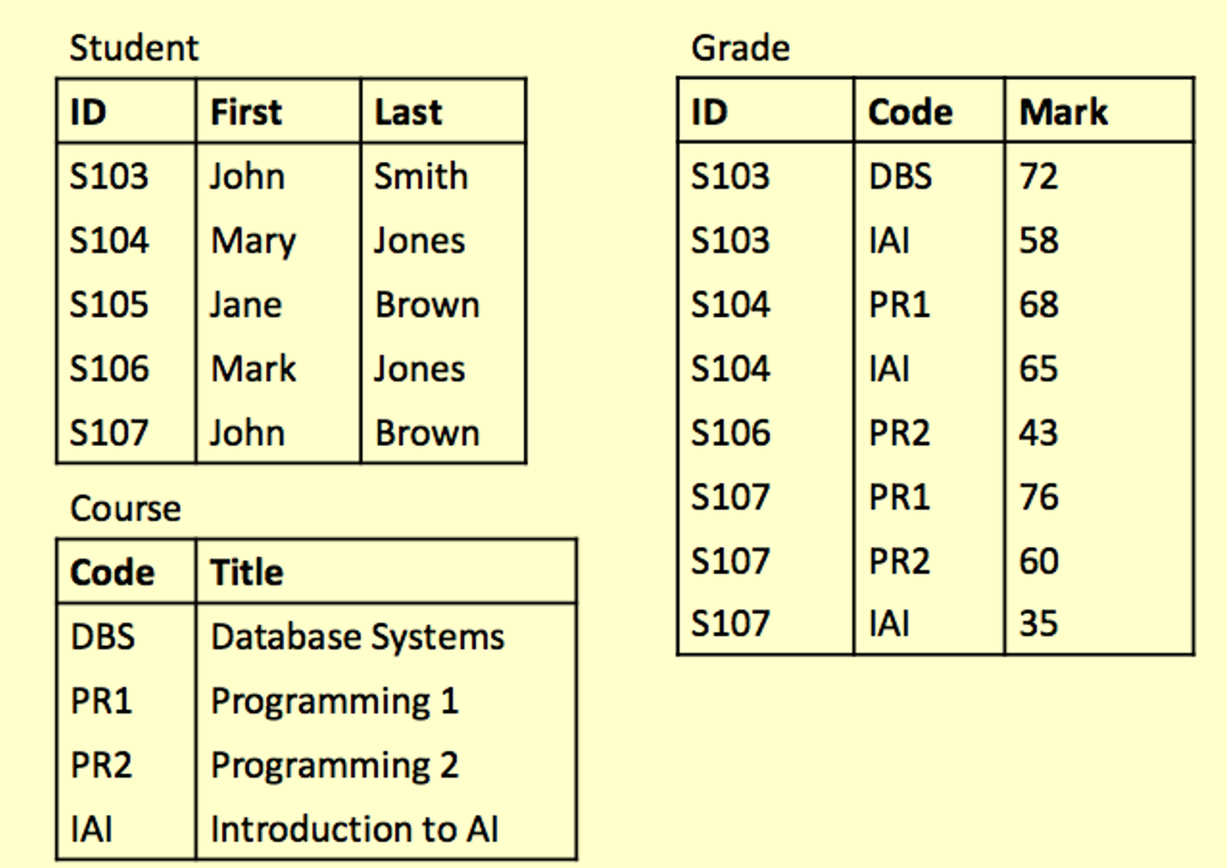
$\pi_{\text{Name, Title, Mark(}}{}^{\sigma}\ \text{((Student.sID = Grade.sID) AND (Grade.mCode = Module.mCode)}{}^{\text{(Student x Grade x Module)}}$

SQL

SQL - Feedback

- “I hope more examples on the SQL commands (SELECT, INSERT, CREATE) can be given.” and “ I'd prefer some coding practice or sth like SQL code advice”
- “Database normalization”, “Normalization” and “Lecture - 009, Database Normalization plz!”
- “Select”
- “exists”

Interactive Walkthrough (from scratch)



Student

ID	First	Last
S103	John	Smith
S104	Mary	Jones
S105	Jane	Brown
S106	Mark	Jones
S107	John	Brown

Course

Code	Title
DBS	Database Systems
PR1	Programming 1
PR2	Programming 2
IAI	Introduction to AI

Grade

ID	Code	Mark
S103	DBS	72
S103	IAI	58
S104	PR1	68
S104	IAI	65
S106	PR2	43
S107	PR1	76
S107	PR2	60
S107	IAI	35

TODO

- Check if table already exists
- CREATE Tables
- INSERT DATA
- Write Queries to Answer our questions

Questions

1. List the titles of all course
2. Find the average grade for IAI
3. Find information on students who **do not** have any registered grades
4. What is the average grade for each course? Include the course’s title and average grade.
5. Find the second highest grade

Questions?