

Advanced Query Formulation

with SQL

Outline

- Outer join problems
- *Type I* nested queries
- *Type II* nested queries and difference problems
- *Nested queries* in the FROM clause
- Division problems
- Null value effects

Type I Nested Queries

IN / NOT IN CLAUSE

Type I Nested Queries

- Query inside a query (คิวรีช้อนคิวรี)
- Use in WHERE and HAVING conditions
- Similar to a nested procedure
- Executes one time
- No reference to outer query
- Also known as non-correlated or independent nested query

IN Clause (Check membership)

```
SELECT *  
FROM Faculty  
WHERE FacDept IN ('MS', 'FIN');
```

```
SELECT *  
FROM Faculty  
WHERE FacDept = 'MS' OR FacDept = 'FIN'
```

IN ดีกว่าอย่างไร ?

SQL สืบค้นตามเงื่อนไข where

โดย ตรวจสอบทีละແລ້ວ*

Faculty อาจารย์

	FacSSN	Fac FirstName	Fac LastName	Fac HireDate	Fac Salary	Fac Supervisor	Fac Dept
1	5432	LEONARD	VINCE	1995-04-10	35000	654-32-1098	MS
2	0987	VICTORIA	EMMANUEL	1996-04-15	120000		MS
3	1098	TOM	FIBON	1994-05-01	70000	543-21-0987	MS

*การณีปกติ ไม่มีคิววิชช่อนกัน

Type I Nested Query Examples I

Example 6 (Access): แสดงชื่ออาจารย์ภาคการเงินที่สอนวิชา IS

```
SELECT FacSSN, FacLastName, FacDept  
FROM Faculty
```

```
WHERE FacDept = 'FIN' AND FacSSN IN
```

```
( SELECT FacSSN FROM Offering  
WHERE CourseNo LIKE 'IS*' )
```

1

2

↑ *Nested query executes one time*

Execute ขั้นที่ 1 เสร็จ ทำขั้นที่ 2 ต่อแบบปกติ*

Type I Nested Query Examples I

2

แสดงข้อมูลอาจารย์ภาควิชาเงิน*

```
SELECT FacSSN, FacLastName, FacDept  
FROM Faculty  
WHERE FacDept = 'FIN' AND  
      FacSSN IN
```

1

(รหัสอาจารย์ที่สอนวิชา IS)

* ขั้นที่ 2 สืบค้นตามเงื่อนไข where โดย ตรวจสอบทีละแล้ว

Offering

1. อ.ที่สอนวิชา IS มี 5432,4321,3210

OfferNo	CourseNo	OffTerm	OffYear	OffLocation	OffTime	FacSSN	OffDays
1	IS320	SUMMER	2016	BLM302	10:30		MW
2	IS320 ✓	FALL	2015	BLM302	10:30	5432✓	MW
6	FIN300	WINTER	2016	BLM207	08:30	2109	MW
7	IS480 ✓	WINTER	2016	BLM302	10:30	4321✓	MW
8	IS480 ✓	SPRING	2016	BLM412	15:30	3210✓	TTH
9	FIN450	WINTER	2016	BLM212	10:30	4321	TTH
10	FIN480	SPRING	2016	BLM305	13:30	2109	MW

Faculty

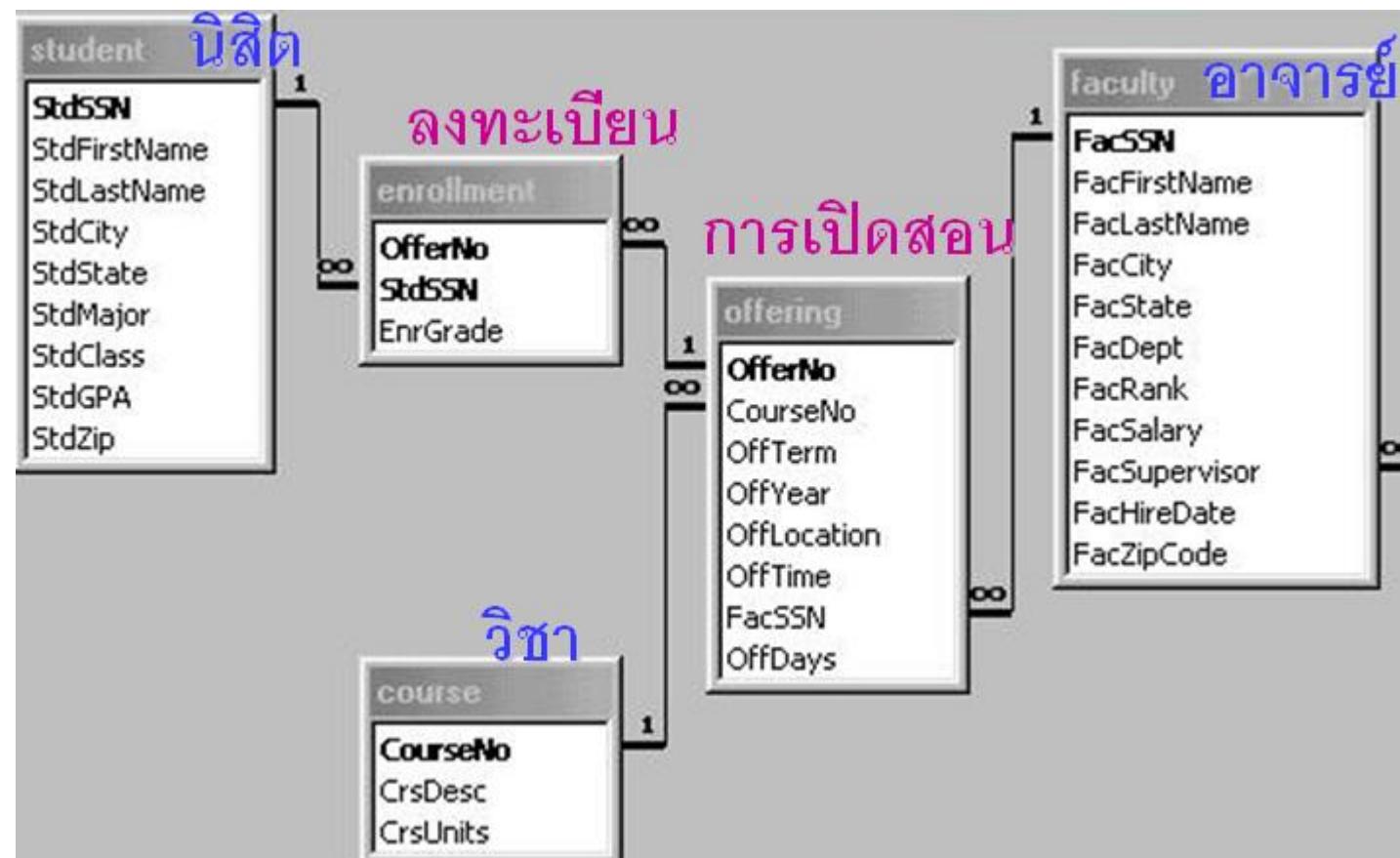
2. อาจารย์ในภาควิชาระบบทั้งหมด 5432,4321,3210

FacSSN	Fac FirstName	Fac LastName	Fac HireDate	Fac Salary	Fac Supervisor	Fac Dept
5432	LEONARD	VINCE	1995-04-10	35000	1098	MS
2109	NICKI	MACON	1997-04-11	65000		FIN
3210	CRISTOPHER	COLAN	1999-03-01	40000	1098	MS
4321 ✓	JULIA	MILLS	2000-03-15	75000	2109	FIN

Type I Nested Query Examples II

Example 7 ค้นหาอาจารย์ภาคการเงินที่สอนวิชา IS ที่มี 4 หน่วยกิต

ต้องสืบค้นจากตารางใดบ้าง ?



Type I Nested Query Examples II

Example 7 (Oracle) อาจารย์ภาคการเงินที่สอนวิชา IS ที่มี 4 หน่วยกิต

```
SELECT FacSSN, FacLastName, FacDept  
FROM Faculty
```

```
WHERE FacDept = 'FIN' AND FacSSN IN
```



2

```
( SELECT FacSSN FROM offering วิชา IS ที่มี 4 หน่วยกิต
```

```
WHERE CourseNo LIKE 'IS%' AND CourseNo IN
```

1

```
( SELECT CourseNo FROM Course  
WHERE CrsUnits = 4 )
```

วิชา 4 หน่วยกิต

)

ต่างจาก join 3 ตาราง (Faculty, Offering, Course) อย่างไร

**SELECT CourseNo FROM Course
WHERE CrsUnits = 4)**



Course		
CourseNo	crsDesc	CrsUnits
<i>FIN300</i>	<i>FUNDAMENTALS OF FINANCE</i>	4
FIN450	PRINCIPLES OF INVESTMENTS	3
<i>FIN480</i>	<i>CORPORATE FINANCE</i>	4
<i>IS320</i>	<i>FUNDAMENTALS OF BUSINESS PROGRAMMING</i>	4
IS460	SYSTEMS ANALYSIS	3
IS470	BUSINESS DATA COMMUNICATIONS	3
IS480	FUNDAMENTALS OF DATABASE MANAGEMENT	3

ໄດ້ ('FIN300', 'FIN480', 'IS320')

Type I Nested Query Examples II (Cont.)

Example 7 (Oracle): List finance faculty who teach 4 unit IS courses.

SELECT FacSSN, FacLastName, FacDept
FROM **Faculty**

WHERE FacDept = '**FIN**' AND FacSSN **IN**

(**SELECT FacSSN FROM Offering**

WHERE CourseNo LIKE 'IS%' AND CourseNo IN

('FIN300', 'FIN480', 'IS320')

3

2

1

**SELECT FacSSN FROM Offering
 WHERE CourseNo LIKE 'IS%' AND CourseNo IN
 ('FIN300', 'FIN480', 'IS320')**

OfferNo	CourseNo	OffTerm	OffYear	OffLocation	OffTime	FacSSN	OffDays
2	IS320 ✓	FALL	2015	BLM302	10:30	5432 ✓	MW
3	IS460	SUMMER	2015	BLM412	13:30		TTH
4	IS320 ✓	SPRING	2016	BLM214	08:30	5432 ✓	MW
5	FIN300	WINTER	2016	BLM207	08:30	2109	MW
6	IS480	SPRING	2016	BLM412	15:30	3210	TTH
7	FIN450	WINTER	2016	BLM212	10:30	4321	TTH
8	FIN480	SPRING	2016	BLM305	13:30	2109	MW
9	IS320 ✓	SUMMER	2016	BLM405	13:30	1098 ✓	MW

ได้อ.รหัส 5432, 1098

Type I Nested Query Examples II (Cont.)



```
SELECT FacSSN, FacLastName, FacDept  
FROM Faculty  
WHERE FacDept = 'FIN' AND FacSSN IN  
      (5432,1098)
```

FacDept = 'FIN' AND FacSSN **IN**



Final result

FacSSN	FacLastName	FacDept
1098	FIBON	FIN

DELETE Example

- Use Type I nested queries to test conditions on other tables
- Use for **UPDATE** statements also

Example 8: Delete offerings taught by Leonard Vince.

```
DELETE FROM Offering  
WHERE Offering.FacSSN IN  
  (SELECT FacSSN FROM Faculty  
   WHERE FacFirstName = 'Leonard'  
   AND FacLastName = 'Vince' )
```

Faculty อาจารย์

FacSSN	Fac FirstName	Fac LastName
5432	LEONARD	VINCE
2109	NICKI	MACON
3210	CRISTOPHER	COLAN
4321	JULIA	MILLS

```

SELECT FROM Faculty
WHERE FacSSN IN
(SELECT FacSSN FROM Offering
WHERE OffYear = "2017" )
    
```

```

SELECT FROM Faculty
WHERE FacSSN IN
(5432, 2109)
    
```

Offering การเปิดสอน

IN

OfferNo	CourseNo	OffTerm	OffYear	FacSSN
1	IS320	SUMMER	2016	
2	IS320	FALL	2017	5432
6	FIN300	WINTER	2017	2109
7	IS480	WINTER	2016	4321
8	IS480	SPRING	2016	3210
9	FIN450	WINTER	2016	4321
10	FIN480	SPRING	2016	2109

IN vs Exists

- IN ตรวจสอบ membership
 - ◆ $x \text{ in } (1,2,3)$
 - ◆ เปรียบเทียบ $x = 1 \text{ or } x=2 \text{ or } x=3$
- Exists ตรวจสอบได้มากกว่า การเท่ากัน ($=$)
 - ◆ เช่น เปรียบเทียบ $x > 1 \text{ or } x > 2 \text{ or } x > 3$
 - ◆ และเปรียบเทียบได้ทีละหลาย attributes
 - ◆ $>, <, >=, <=, =, \text{ like, not}$
- มีรูปแบบคำสั่งต่างกัน และ การ execute ต่างกัน

Faculty อาจารย์

FacSSN	Fac FirstName	Fac LastName
5432	LEONARD	VINCE
2109	NICKI	MACON
3210	CRISTOPHER	COLAN
4321	JULIA	MILLS

```

SELECT * FROM Faculty F
WHERE Exists
  (SELECT * FROM Offering O
  WHERE (OffYear = '2017') and
  O.FacSSN = F.FacSSN)
  
```



ตรวจสอบทุกແຄวใน Faculty กับทุกແຄวใน Offering

Offering การเปิดสอน

OfferNo	CourseNo	OffTerm	OffYear	FacSSN
1				
2	IS320	FALL	2017	5432
6	FIN300	WINTER	2017	2109
7				
8				
9				
10				

Exists

Type II Nested Query

Exists, Not Exists,
Set Difference Problems

Type II Nested Queries

- Similar to nested loops
- *Executes one time for each row of outer query*
- Reference to outer query
- Also known as *correlated* or variably *nested query*
- Use for difference problems not joins

EXISTS Clause

```
SELECT *  
FROM Faculty  
WHERE EXISTS ( sub-query)
```

Outer
Query

- Sub-query จะถูก execute ทุกครั้งที่มีการตรวจสอบข้อมูลแต่ละแถวใน faculty
- Sub-query สามารถอ้างอิงข้อมูลใน FACULTY ได้
- Exists (**sub-query** ให้ผลลัพธ์ ≥ 1 และ) = true
 - ◆ *stop the scanning process as soon as it found a match.*
- Exists (NULL) = false

EXISTS Clause

แสดงรายชื่ออาจารย์ที่เคยสอนอย่างน้อย 1 วิชา

SELECT *

FROM Faculty

WHERE EXISTS

(**SELECT * FROM Offering**

WHERE Faculty.FacSSN = Offering.FacSSN)

วิธีทำ

- ตรวจสอบแต่ละแถวใน **Faculty** ว่า

- ◆ มีผลลัพธ์จาก **sub-query** ไหม

- ◆ ถ้ามีคือ **Exists (select)** เป็นจริง ก็แสดงผลข้อมูลแทนนั้น

Exists(มีผลลัพธ์) = true

Exists(null) = false

ແຕ່ລະແກໃນ Faculty

ตรวจสอบ If sub-query is Not NULL

FacSSN	FacFirstName	FacLastName	FacDept
--------	--------------	-------------	---------

5432	LEONARD	VINCE	MS
------	---------	-------	----

0987	VICTORIA	EMMANUEL	MS
------	----------	----------	----

1098	LEONARD	FIBON	MS
------	---------	-------	----

SELECT * FROM Offering
WHERE OffTerm = 'WINTER'
AND OffYear = 2016
AND **0987** = Offering.FacSSN

1

```
SELECT * FROM Offering  
WHERE OffTerm = 'WINTER'  
AND OffYear = 2016  
AND 5432 = Offering.FacSSN
```

2

Sub-query...

3

NOT EXISTS Clause

แสดงรายชื่ออาจารย์ที่ยัง ไม่เคยสอนวิชาใดเลย

SELECT *

FROM Faculty

WHERE NOT EXISTS

(SELECT * FROM Offering

WHERE Faculty.FacSSN = Offering.FacSSN)

Not Exists(มีผลลัพธ์) = false

Not Exists(null) = true

วิธีทำ

- ตรวจสอบแต่ละแถวใน Faculty ว่า
 - ◆ มีผลลัพธ์จาก sub-select clause ไหม
 - ◆ ถ้าไม่มีคือ Not Exists (select) เป็นจริง ก็แสดงผล

ແຕລະແກວໃນ faculty

ຕរាងតួន if sub-query is NULL

FacSSN	FacFirstName	FacLastName	FacDept
--------	--------------	-------------	---------

5432	LEONARD	VINCE	MS
0987	VICTORIA	EMMANUEL	MS
1098	LEONARD	FIBON	MS

```
SELECT * FROM Offering  
WHERE OffTerm = 'WINTER'  
AND OffYear = 2016  
AND 0987 = Offering.FacSSN
```

```
1  
SELECT * FROM Offering  
WHERE OffTerm = 'WINTER'  
AND OffYear = 2016  
AND 5432 = Offering.FacSSN
```

2

Sub-query...

3

Type II Nested Query

Example for a Set Difference Problem

Example 9: แสดงรายชื่ออาจารย์ภาค MS ที่ไม่ได้สอนในเทอม winter ปี 2016.

- รายชื่ออาจารย์ มาจากตาราง **Faculty**
- เงื่อนไข **ไม่ได้สอนเทอม Winter 2016**
 - ต้องสืบค้นในตาราง Offering
- ถ้า JOIN Faculty กับ Offering จะได้เฉพาะอ.ที่สอนในเทอม ที่ระบุ
- ทำอย่างไร จะได้คนที่ ไม่ได้สอน ในเทอม winter ปี 2016.

NOT IN

Example 9: ค้นหาอาจารย์ภาค MS ที่ไม่ได้สอนในเทอม winter ปี 2016.

```
SELECT FacSSN, FacLastName, FacDept  
FROM Faculty  
WHERE FacDept = 'MS' AND
```

Faculty.FacSSN NOT IN (อาจารย์ A , B , C)

```
( SELECT Offering.FacSSN FROM Offering  
WHERE OffTerm = 'WINTER' AND OffYear = 2016 )
```

Type II Nested Query

Example for a Set Difference Problem

Example 9: ค้นหาอาจารย์ภาค MS ที่ไม่ได้สอนในเทอม winter ปี 2016.

```
SELECT FacSSN, FacLastName, FacDept  
FROM Faculty  
WHERE FacDept = 'MS' AND NOT EXISTS  
( SELECT * FROM Offering  
    WHERE OffTerm = 'WINTER' AND OffYear = 2016  
    AND Faculty.FacSSN = Offering.FacSSN )
```

corelated

Limited Formulations for Set Difference Problems (A-B)

- Type I nested query with **NOT IN** condition
- One-sided outer join with **IS NULL** condition
- Difference operation using **MINUS (EXCEPT)** operator

Type I Difference Formulation

Example 10: แสดงรายชื่ออาจารย์ภาควิชา MS ที่ ?

```
SELECT FacSSN, FacLastName, FacDept  
FROM Faculty, Offering  
WHERE FacDept = 'MS' AND  
      OffTerm <> 'WINTER' AND  
      OffYear <> 2016 )
```

ได้รายชื่อ. ทุกคนที่สอนในเทอมต่างๆ ซึ่ง
อ.ที่สอนทั้ง summer และ winter ก็จะปรากฏในผลลัพธ์ด้วยซึ่งไม่ใช่ที่เราต้องการ

Type I Difference Formulation

Example 10: แสดงรายชื่ออาจารย์ภาควิชา MS ที่ไม่ได้สอนในเทอม winter ปี 2016.

```
SELECT FacSSN, FacLastName, FacDept  
FROM Faculty  
WHERE FacDept = 'MS' AND FacSSN NOT IN
```

```
( SELECT FacSSN FROM Offering  
WHERE OffTerm = 'WINTER'  
AND OFFYear = 2016 )
```



Faculty อาจารย์ภาค MS ที่ไม่ได้สอนในเทอม winter ปี 2016

2

Yes
No
Yes
Yes

FacSSN	Fac FirstName	Fac LastName	Fac HireDate	Fac Salary	Fac Supervisor	Fac Dept
5432	LEONARD	VINCE	1995-04-10	35000	654-32-1098	MS
0987	VICTORIA	EMMANUEL	1996-04-15	120000		MS
1098	LEONARD	FIBON	1994-05-01	70000	543-21-0987	MS
2109	NICKI	MACON	1997-04-11	65000		FIN
3210	CRISTOPHER	COLAN	1999-03-01	40000	654-32-1098	MS
4321	JULIA	MILLS	2000-03-15	75000	765-43-2109	FIN

Offering อาจารย์ที่สอน in Winter 2016

Check if not in

OfferNo	CourseNo	OffTerm	OffYear	OffLocation	OffTime	FacSSN	OffDays
4444	IS320	WINTER	2016	BLM302	15:30	0987	TTH
5555	FIN300	WINTER	2016	BLM207	08:30	2109	MW
5678	IS480	WINTER	2016	BLM302	10:30	4321	MW
6666	FIN450	WINTER	2016	BLM212	10:30	4321	TTH

Executed one time

1

One-Sided Outer Join Difference Formulation

Example 11: แสดงรายชื่ออาจารย์ภาค MS ที่ไม่เคยสอนวิชาใดเลย
(research faculty)

```
SELECT FacSSN, FacLastName, FacDept
FROM Faculty LEFT JOIN Offering
ON Faculty.FacSSN = Offering.FacSSN
WHERE FacDept = 'MS'
AND Offering.FacSSN IS NULL
```

MINUS Operator Difference Formulation

Example 12 (Oracle): แสดงรายชื่ออาจารย์ที่ไม่ได้เป็นนักเรียน

```
SELECT FacSSN AS SSN, FacFirstName AS FirstName,  
       FacLastName AS LastName, FacCity AS City,  
       FacState AS State  
FROM Faculty  
MINUS  
SELECT StdSSN AS SSN, StdFirstName AS FirstName,  
       StdLastName AS LastName, StdCity AS City,  
       StdState AS State  
FROM Student
```

Nested Queries in the FROM Clause

- More recent introduction than nested queries in the WHERE and HAVING clauses
- Consistency in language design
- *Wherever table appears, table expression can appear*
- Specialized uses
 - ◆ Nested aggregates
 - ◆ Multiple independent aggregate calculations

Nested FROM Query Example

Example 13: แสดงรหัสวิชา คำอธิบายรายวิชา จำนวนครั้งที่เปิดสอน และ ค่าเฉลี่ยของจำนวนนักเรียนที่ลงทะเบียนเรียน
(*course number, course description, the number of offerings, and the average enrollment across offering*)

วิชานึงเปิดได้หลายครั้งในหลายๆ เทอม (นับแคาใน offering)
แต่ละครั้งที่เปิด มีนักเรียนลงทะเบียนได้หลายคน (นับแคาใน enrollment)

ตัวอย่าง ผลลัพธ์

CourseNo	Description	NumOfferings	AvgEnr
204351	Database Systems	27	50
204111	Computer Programming	54	100

(Course X Offering) X Enrollment ???

Nested FROM Query Example

Example 13: แสดงรหัสวิชา คำอธิบายรายวิชา จำนวนครั้งที่เปิดสอน และค่าเฉลี่ยของจำนวนนักเรียนที่ลงทะเบียนเรียน

- สืบค้นจาก ตาราง Offering, Enrollment, Course
 - เชื่อมโยงโดยให้
 $Offering.OfferNo = Enrollment.OfferNo$
 $Course.CourseNo = Offering.CourseNo$
- จัดกลุ่มเพื่อหาค่า AVG, Count
 $GROUP BY Course.CourseNo, Offering.OfferNo$

(Course X Offering) X Enrollment

ตัวอย่าง ผลลัพธ์

OfferNo						
204351	database	1	2016	winter	sect1	st1
204351	database	1	2016	winter	sect1	st2
204351	database	1	2016	winter	sect1	st3
204351	database	2	2016	fall	sect1	st1
204351	database	2	2016	fall	sect1	st2
204351	database	2	2016	fall	sect1	st3
204111	C#	3	2016	fall	sect1	st1
204111	C#	3	2016	fall	sect1	st2
204111	C#	3	2016	fall	sect1	st3
204111	C#	4	2016	fall	sect2	st1
204111	C#	4	2016	fall	sect2	st2

204111 เปิด 2 ครั้ง

1.นับคนลงเรียนแต่ละ
หมู่ จัดกลุ่มโดย ?

2.นับจำนวนครั้งที่เปิดแต่ละวิชา
จัดกลุ่มโดย ?

ตัวอย่าง ผลลัพธ์ของการจัดกลุ่มโดยวิชาและหมู่เรียน

				คนลงเรียน
204351	database	1	2016 winter	sect1 3
204351	database	2	2016 fall	sect1 3
204111	C#	3	2016 fall	sect1 3
204111	C#	4	2016 fall	sect2 2

2. นับจำนวนครั้งที่เปิดแต่ละวิชา จัดกลุ่มโดย ?

ตัวอย่าง ผลลัพธ์สุดท้าย

	จน. หมู่ที่เปิด	AVG(นร.)ต่อหมู่
204351 database	2	3
204111 C#	2	2.5

Nested FROM Query Example

Example 13: แสดง course number, course description, the number of offerings, and the average enrollment across offering.

```
SELECT T.CourseNo, T.CrsDesc,  
       COUNT(*) AS NumOfferings,  
       AVG(T.EnrollCount) AS AvgEnroll
```

FROM

(SELECT Course.CourseNo, CrsDesc,

นับจำนวนหมู่ที่เปิด
เฉลี่ยคนลงรายวิชา

Offering.OfferNo,

COUNT(*) AS EnrollCount

table expression

FROM Offering, Enrollment, Course

นับคนลงเรียนแต่ละหมู่

WHERE Offering.OfferNo = Enrollment.OfferNo

AND Course.CourseNo = Offering.CourseNo

GROUP BY Course.CourseNo, CrsDesc,

จัดกลุ่มรายหมู่

Offering.OfferNo) T

จัดกลุ่มรายวิชา

GROUP BY T.CourseNo, T.CrsDesc

Divide Operator

- เป็นการ **match** ข้อมูล บางส่วน ในตาราง
 - ◆ **Suppliers who supply all parts**
 - ◆ **Faculty who teach every IS course**
- ไม่มี operator โดยตรงต้อง apply ตัวอื่น
- ส่วนมากใช้กับตารางเชื่อมโยงความสัมพันธ์แบบ **M-N (associative tables)**

Division Example

ตัวตั้ง

ตัวหาร

ผลลัพธ์

SuppPart

SuppNo	PartNo
s3	p1
s3	p2
s3	p3
s0	p1
s1	p2

Part

PartNo
p1
p2
p3

SuppPart DIVIDEBY Part

SuppNo
s3

s3 {p1, p2, p3}
contains {p1, p2, p3}

COUNT Method for Division Problems

- ใช้วิธีนับจำนวนข้อมูลที่สนใจในตารางที่สัมพันธ์กัน เพื่อเปรียบเทียบกัน

Example 14: รายชื่อนักเรียนที่เป็นสมาชิกของทุกชมรม (all clubs)

- คำนวณ
 - ? จำนวนชมรมที่นร. แต่ละคนเป็นสมาชิก (X_i)
 - ? จำนวนชมรมทั้งหมด (Y)
- จำนวน X_i & Y ต้องเท่ากันถ้าเป็นสมาชิกทุกชมรม

COUNT Method for Division Problems

- Type I nested query in the HAVING clause

Example 14: รายชื่อนักเรียนที่เป็นสมาชิกของทุกชมรม (all clubs)

```
SELECT StdNo  
FROM StdClub  
GROUP BY StdNo  
HAVING COUNT(*) =
```

(*SELECT COUNT(*) FROM Club*)

Nested query จะ execute ครั้งเดียวเท่านั้น: ไม่มีการอ้างอิง
ถึงข้อมูลในตาราง ใน query ด้านบน

เปรียบเทียบจน. ชมรม

Typical Division Problems

- เปรียบเทียบบางส่วนของตาราง ไม่ทั้งหมด

Ex15: รายชื่อนักเรียนที่เป็นสมาชิกของทุกชั้นรมสังคม (*all social clubs*)

```
SELECT Student1.StdNo, SName  
FROM StdClub, Club, Student1  
WHERE StdClub.ClubNo = Club.ClubNo  
AND Student1.StdNo = StdClub.StdNo  
AND CPurpose = 'SOCIAL'  
GROUP BY Student1.StdNo, SName  
HAVING COUNT(*) =  
( SELECT COUNT(*) FROM Club  
WHERE CPurpose = 'SOCIAL' )
```

เปรียบเทียบด้วย

เงื่อนไขคล้ายกัน

Advanced Division Problems

- Count distinct values rather than rows
 - ◆ อาจารย์ที่สอนอย่างน้อย 1 หมู่ของทุกวิชาที่เปิดสอน
 - ◆ ตาราง Offering วิชาหนึ่งอาจเปิดได้หลายครั้งและหลายหมู่
- Use COUNT (DISTINCT column)
- Use stored query or nested FROM query in Access

Advanced Division Problem Example

Example 16: ค้นหารหัสอาจารย์ที่สอนอย่างน้อย 1 หน่วยของทุกวิชา IS ในเทอมนี้ (winter 2016, IS courses)

วิธีทำ

- ? อาจารย์ที่สอนวิชา IS ในเทอม winter 2016
- ? จำนวนวิชาที่แต่ละคนสอน (X)
- ? จำนวนวิชา IS ที่เปิดในเทอม winter 2016 (Y)
- จำนวน X & Y ต้องเท่ากันถ้าสอนทุกวิชา

Example : ค้นหาลูกค้าที่ใช้บริการทุกชนิดของ **TRUE MOVE** ลูกค้าที่ซื้อขัมเปงและนมในครัวเดียวกัน

Advanced Division Problem Example

Example 16: Step 1

```
SELECT Faculty.FacSSN, Faculty.FirstName,  
       Faculty.LastName  
FROM Faculty, Offering  
WHERE Faculty.FacSSN = Offering.FacSSN  
      AND OffTerm = 'WINTER'  
      AND CourseNo LIKE 'IS%'  
      AND OffYear = 2016
```

อาจารย์ที่สอนวิชา **IS** ในSEMESTERนี้ (*winter 2016, IS courses*)

Advanced Division Problem Example

Example 16: Step 2 จัดกลุ่มนับจำนวนวิชาที่แต่ละคนสอน

```
SELECT Faculty.FacSSN, FacFirstName,  
       FacLastName  
FROM Faculty, Offering  
WHERE Faculty.FacSSN = Offering.FacSSN  
      AND OffTerm = 'WINTER'  
      AND CourseNo LIKE 'IS%'  
      AND OffYear = 2016  
GROUP BY Faculty.FacSSN, FacFirstName, FacLastName  
HAVING COUNT(DISTINCT CourseNo) =  
          จำนวนวิชา IS ทั้งหมดที่เปิดในเทอม winter 2016
```

Advanced Division Problem Example

Example 16: List the SSN and the name of faculty who teach at least one section of all of the winter 2016, IS courses.

```
SELECT Faculty.FacSSN, FacFirstName,  
       FacLastName  
FROM Faculty, Offering  
WHERE Faculty.FacSSN = Offering.FacSSN  
      AND OffTerm = 'WINTER'  
      AND CourseNo LIKE 'IS%'  
      AND OffYear = 2016  
GROUP BY Faculty.FacSSN, FacFirstName,  
         FacLastName  
HAVING COUNT(DISTINCT CourseNo) = ( SELECT COUNT(DISTINCT CourseNo)  
          FROM Offering  
         WHERE OffTerm = 'WINTER' AND  
               OffYear = 2016 AND  
               CourseNo LIKE 'IS%' )
```

faculty who teach
the winter 2016,
IS courses.

→ Num of courses
taught

Num of all of
the winter 2016,
IS courses

Null Value Effects

- Simple conditions
- Compound conditions
- Grouping and aggregate functions
- SQL:2003 standard but implementation may vary

Simple Conditions

- Simple condition *is null if* either left-hand or right-hand side is null. ($30 > \text{null}$)
- *Discard rows* evaluating to false or null
- Retain rows evaluating to true
- Rows evaluating to null will *not appear* in the result of the simple condition or its negation

Aggregate Functions

- Null values ignored
- Effects can be subtle
 - ◆ $COUNT(*)$ may differ from $Count(Column)$
 - $COUNT(column)$ returns the number of non-null values in the column
 - ◆ $SUM(Column1) + SUM(Column2)$ may differ from $SUM(Column1 + Column2)$
 - Since $1+NULL = Null$

A	B
10	10
Null	20
30	30
Sum(A)=40	Sum(B)=60
$SUM(A)+SUM(B) = 100$	
$SUM(A+B) =$	
$(10+10)+(Null+20)+(30+30)$	
$=80$	

Grouping Effects

- Rows with **null** values are grouped together
 - ◆ เช่น จัดกลุ่ม นร. ตามจังหวัดที่อาศัย แต่ นร. บางคนไม่มีข้อมูลคือเป็น **NULL**
- Grouping column contains null values
 - ◆ Null group can be placed at beginning or end of the non-null groups
 - ◆ ถ้าข้อมูลที่มีคอลัมน์ที่ใช้จัดกลุ่มเป็น **null** จะถูกจัดในกลุ่มเดียวกัน และแสดงไว้ตอนต้นหรือท้ายในผลลัพธ์ของคิวรี
 - ◆ นร. ที่ไม่มีข้อมูลจังหวัด ก็จะถูกจัดกลุ่มเดียวกัน

Summary

- Advanced matching problems not common but important when necessary
- Understand *outer join, difference, and division* operators
- *Nested queries* important for advanced matching problems
- *Lots of practice* to master query formulation and SQL