

Introduction to Database II



204202 IT II

Foreign key



- **Foreign key** คือ **attribute** หนึ่งในตารางที่อ้างอิงถึง **attribute** ของตารางอื่น
- **Attribute** ของตารางอื่นที่ว่า ส่วนใหญ่จะเป็น **primary key** ของตารางนั้น
- **foreign key** มักใช้ในกรณีที่ต้องการแยกตารางใหญ่ๆ ออกเป็นตารางย่อยๆ โดยที่ยังสามารถเชื่อมตารางสองตารางเข้าไว้ด้วยกันด้วย **foreign key**
- โดย **foreign key** จะมีหน้าที่รักษาความสอดคล้องของข้อมูลที่อ้างอิงกันกัน (referential integrity)

Example



<u>Student ID</u>	Name	Advisor
1000159	Jamie Bulldog	Philip Patek
1000259	Caroline White	Philip Patek
1000359	Sam Smith	Philip Patek
1000459	John Burrad	Wesley Stone
1000559	Steven King	Philip Patek
1000659	Roger Cunningham	Wesley Stone

ตารางข้างบน เป็นตารางของนศ และชื่อของที่ปรึกษา ซึ่งเก็บในรูปแบบของตาราง
เดียว

Example (cont.)



- สมมุติว่า **Phillip** ลาออกจากการเป็นที่ปรึกษา
- เราต้องแก้ไขตารางดังกล่าว 4 จุด โดยต้องไม่ลืมที่จะแก้แถวหนึ่งแถวใด
- ทว่า การแก้ไขหลายๆจุด อาจเกิดการผิดพลาดขึ้นได้
- วิธีที่จะลดการผิดพลาดคือการแยกตารางออกเป็นสองตาราง ดังต่อไปนี้

Using 2 tables design



<u>Student ID</u>	Name	Advisor
1000159	Jamie Bulldog	20011
1000259	Caroline White	20011
1000359	Sam Smith	20011
1000459	John Burrad	15029
1000559	Steven King	20011
1000659	Roger Cunningham	15029

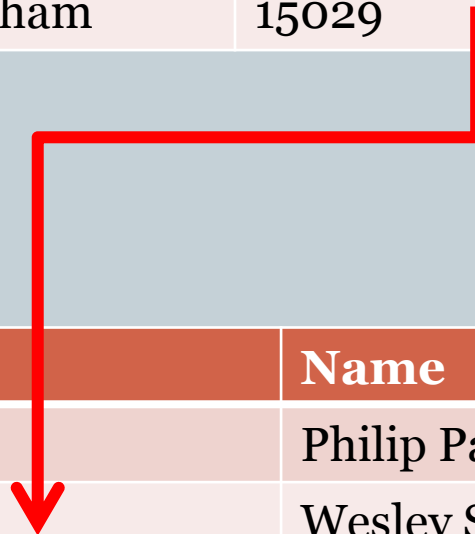
ตัวที่ขีดเส้นใต้คือ

Primary key

ส่วน **Advisor** คือ

Foreign key

<u>Staff ID</u>	Name
20011	Philip Patek
15029	Wesley Stone



JOINS



- **JOINS** เป็น operation ที่ใช้ในการรวมข้อมูลของตารางเข้าด้วยกัน

- **JOIN** มีหลายแบบ
 - ✦ **CROSS JOIN**
 - ✦ **INNER JOIN**
 - ✦ **NATURAL JOIN**
 - ✦ **OUTER JOIN**

A CROSS JOIN B

- returns **all** pairs of rows from A and B

A NATURAL JOIN B

- returns pairs of rows with common values for identically named columns and without duplicating columns

CROSS JOIN

Student

ID	Name
123	John
124	Mary
125	Mark
126	Jane

Enrolment

ID	Code
123	DBS
124	PRG
124	DBS
126	PRG

```
SELECT * FROM  
Student CROSS JOIN  
Enrolment
```

ID	Name	ID	Code
123	John	123	DBS
124	Mary	123	DBS
125	Mark	123	DBS
126	Jane	123	DBS
123	John	124	PRG
124	Mary	124	PRG
125	Mark	124	PRG
126	Jane	124	PRG
123	John	124	DBS
124	Mary	124	DBS

NATURAL JOIN



Student

ID	Name
123	John
124	Mary
125	Mark
126	Jane

Enrolment

ID	Code
123	DBS
124	PRG
124	DBS
126	PRG

```
SELECT * FROM
```

```
Student NATURAL JOIN  
Enrolment
```

ID	Name	Code
123	John	DBS
124	Mary	PRG
124	Mary	DBS
126	Jane	PRG

CROSS and NATURAL JOIN

```
SELECT * FROM  
  A CROSS JOIN B
```

- เหมือนกับ

```
SELECT * FROM A, B
```

```
SELECT * FROM  
  A NATURAL JOIN B
```

- เหมือนกับ

```
SELECT A.col1,... A.coln,  
FROM A, B  
WHERE A.col1 = B.col1  
      AND A.col2 = B.col2  
      ...AND A.coln = B.col.n
```

(สมมุติว่า **col1... coln** คือ **attribute** ที่เหมือนกันใน **A** และ **B**)

SQL functions: Count()



```
SQL> SELECT * FROM employee_tbl;
```

id	name	work_date	daily_typing_pages
1	John	2007-01-24	250
2	Ram	2007-05-27	220
3	Jack	2007-05-06	170
4	Jack	2007-04-06	100
5	Jill	2007-04-06	220
6	Zara	2007-06-06	300

```
SQL> SELECT COUNT(*) FROM employee_tbl ;
```

COUNT(*)
7

SUM(), AVG() functions



```
SQL> SELECT SUM(daily_typing_pages) FROM  
employee_tbl;
```

```
+-----+  
| SUM(daily_typing_pages) |  
+-----+  
|          1610          |  
+-----+
```

```
SQL> SELECT AVG(daily_typing_pages) FROM  
employee_tbl;
```

```
+-----+  
| AVG(daily_typing_pages) |  
+-----+  
|          230.0000        |  
+-----+
```

Other useful functions



- **MAX()** หาค่ามากที่สุด
- **MIN()** หาค่าน้อยสุด
- **FLOOR()** ฟังก์ชันปัดลง
- **CEIL()** ฟังก์ชันปัดขึ้น
- **LCASE()** เปลี่ยนผลลัพธ์ให้เป็นตัวเล็ก
- **UCASE()** เปลี่ยนผลลัพธ์ให้เป็นตัวใหญ่