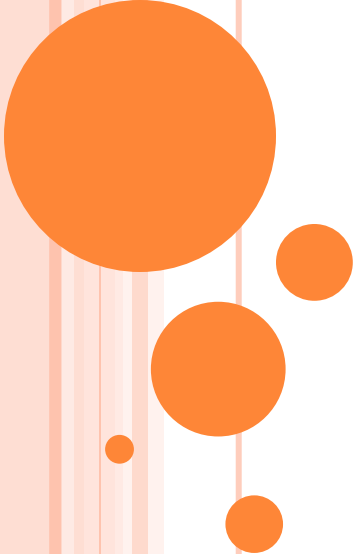


INTRODUCTION TO SQL

DISPLAYING DATA FROM MULTIPLE TABLES (OUTER JOIN)



By Kanokwan Atchariyachanvanich
Faculty of Information Technology
KMITL
Database System Concepts

EXAMPLE: JOIN WITH ON CLAUSE

Query: Find all orders with their product names and price.

ORDER

ORDER_NO	CUSTOMER_NO	P_CODE
1	C001	111110
2	C002	222220

PRODUCT

P_CODE	P_NAME	PRICE
222220	คอมพิวเตอร์	30000
111110	สมุด	120
333330	ปากกา	500

SELECT *

FROM ORDER

Join PRODUCT

On (Order.P_CODE = Product.P_CODE) ;

RECALL : JOIN ORDER

Select *
From PRODUCT
JOIN ORDER
USING (P_CODE) ;

PRODUCT

PRODUCT.P_CODE	P_NAME	PRICE
222220	คอมพิวเตอร์	30000
111110	สมุด	120
333330	ปากกา	500

ORDER_NO	CUSTOMER_NO	ORDER.P_CODE
1	C001	111110
2	C002	222220

Automatically
rename

Step 1: Cartesian Product

ORDER_NO	CUSTOMER_NO	ORDER.P_CODE	PRODUCT.P_CODE	P_NAME	PRICE
1	C001	111110	222220	คอมพิวเตอร์	30000
1	C001	111110	111110	สมุด	120
1	C001	111110	333330	ปากกา	500
2	C002	222220	222220	คอมพิวเตอร์	30000
2	C002	222220	111110	สมุด	120
2	C002	222220	333330	ปากกา	500

RECALL : JOIN

Step 2 : Select Join attribute P_CODE values are equal

ORDER_NO	CUSTOMER_NO	ORDER.P_CODE	PRODUCT.P_CODE	P_NAME	PRICE
1	C001	111110	222220	คอมพิวเตอร์	30000
1	C001	111110	111110	สมุด	120
1	C001	111110	333330	ปากกา	500
2	C002	222220	222220	คอมพิวเตอร์	30000
2	C002	222220	111110	สมุด	120
2	C002	222220	333330	ปากกา	500



ORDER_NO	CUSTOMER_NO	ORDER.P_CODE	PRODUCT.P_CODE	P_NAME	PRICE
1	C001	111110	111110	สมุด	120
2	C002	222220	222220	คอมพิวเตอร์	30000

RECALL : JOIN

Step 3 : use a Projection to eliminate the duplicate attributes

ORDER_NO	CUSTOMER_NO	ORDER.P_CODE	PRODUCT.P_CODE	P_NAME	PRICE
1	C001	111110	111110	สมุด	120
2	C002	222220	222220	คอมพิวเตอร์	30000

Select *
From PRODUCT
JOIN ORDER
USING (P_CODE) ;

ORDER_NO	CUSTOMER_NO	P_CODE	P_NAME	PRICE
1	C001	111110	สมุด	120
2	C002	222220	คอมพิวเตอร์	30000

7. OUTER JOINS

- สามารถแสดงข้อมูลแถวที่ไม่สามารถจับคู่เงื่อนไขเท่ากันกับอีกตาราง ซึ่งปกติจะไม่แสดงเวลาใช้ join

EMPLOYEES

LAST_NAME	LAST_NAME	FIRST_NAME	DEPARTMENT_ID
King	King	Steven	90
Kochhar	Kochhar	Neena	90
De Haan	De Haan	Lex	90
Hunold	Hunold	Alexander	60
Ernst	Ernst	Bruce	60
Taylor	Taylor	Jonathon	80
Livingston	Livingston	Jack	80
Grant	Grant	Kimberely	
Johnson	Johnson	Charles	80
Taylor	Taylor	Winston	50
Fleaur	Fleaur	Jean	50
Sullivan	Sullivan	Martha	50
Geoni	Geoni	Girard	50

107 row selected

DEPARTMENTS

LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
Whalen	10	Administration
Hartstein	20	Marketing
Fay	20	Marketing
Raphaely	30	Purchasing
Khoo	30	Purchasing
Baida	30	Purchasing
Colmenares	30	Purchasing

106 row selected

UNMATCHED ROWS VERSUS MATCHED ROWS

- To return the unmatched rows, you can use an outer join
 - There are three types of outer joins:
 - FULL OUTER
 - LEFT OUTER
 - RIGHT OUTER
- Reminder: Joining tables with the NATURAL JOIN, USING or ON clauses results in matched rows displayed in the output.

7. OUTER JOINS

- A join between two tables that returns the results of the inner join as well as
- Full outer join: เป็นการ join ที่ขยายมาจาก join-using คือ สามารถแสดงข้อมูลแถวที่สามารถและไม่สามารถจับคู่เงื่อนไขเท่ากันกับอีกตาราง ซึ่งปกติจะไม่แสดงเวลาใช้ join-using
- Left outer join: แสดงแถวในตารางที่อยู่ทางซ้ายมือออกมาทั้งหมด ถึงแม้จะมีแถวในตารางซ้ายมือมีค่าของฟิลด์ที่ไม่ตรงกับฟิลด์เชื่อมของตารางทางขวามือ ส่วนตารางทางขวามือ ก็จะ return ค่าเป็น NULL สำหรับแถวที่เงื่อนไขไม่ match กับตารางหลัก
- Right outer join: แสดงแถวในตารางที่อยู่ทางขวามือออกมาทั้งหมด ถึงแม้จะมีแถวในตารางขวามือมีค่าของฟิลด์ที่ไม่ตรงกับฟิลด์เชื่อมของตารางทางซ้ายมือ ส่วนตารางทางซ้ายมือ ก็จะ return ค่าเป็น NULL สำหรับแถวที่เงื่อนไขไม่ match กับตารางหลัก

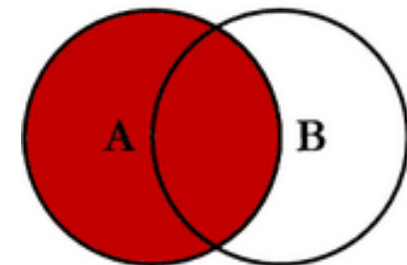
LEFT OUTER JOIN

SELECT	attribute
FROM	A
LEFT OUTER JOIN	B
USING	(attribute) ;

- LEFT OUTER JOIN ใช้สำหรับการแสดงแถวในตารางที่อยู่ทางซ้ายมือออกมา ทั้งหมด ถึงแม้จะมีแถวในตารางซ้ายมือมีค่าของฟิลด์ที่ไม่ตรงกับฟิลด์เชื่อมของ ตารางทางขวามือ ส่วนตารางทางขวามือ ก็จะ return ค่าเป็น NULL สำหรับ แถวที่เงื่อนไขไม่ match กับตารางหลัก
- จะรวมข้อมูลที่ตรงกัน โดยจะเน้นตารางฝั่งซ้ายเป็นหลัก
- Matched pairs are retained and any unmatched values in A table are retained too, but *any unmatched values in B table are eliminated.*

Result set

1. Matched values from result of Join-using
2. Unmatched values of attributes in A



```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```

EXAMPLE: LEFT OUTER JOIN

Query: Find all orders with their product names and price.

ORDER

ORDER_NO	CUSTOMER_NO	P_CODE
1	C001	111110
2	C002	222220

PRODUCT

P_CODE	P_NAME	PRICE
222220	คอมพิวเตอร์	30000
111110	สมุด	120
333330	ปากกา	500

SELECT *

FROM ***ORDER***

Left Outer Join PRODUCT

On (Order.P_CODE = Product.P_CODE) ;

EXAMPLE : LEFT OUTER JOIN

1. Yields **result from ORDER** Join **PRODUCT**
2. Plus **all rows in ORDER** table, including those that do not have a matching value in the **PRODUCT** table

```
SELECT  ORDER_NO, CUSTOMER_NO,
        P_CODE, P_NAME, PRICE
FROM    ORDER
LEFT OUTER JOIN  PRODUCT
On (Order.P_CODE = Product.P_CODE) ;
```

ORDER_NO	CUSTOMER_NO	P_CODE	P_NAME	PRICE
1	C001	111110	สมุด	120
2	C002	222220	คอมพิวเตอร์	30000

1. Matched values
from result of
ORDER ⋈ PRODUCT

ORDER

ORDER_NO	CUSTOMER_NO	P_CODE
1	C001	111110
2	C002	222220

2. Unmatched values from
ORDER (NOT FOUND)

EXAMPLE

1. จงเขียน SQL Query ที่แสดงข้อมูลพนักงานที่มีชื่อแผนกและที่ยังไม่มี
แผนก

```
SELECT      *  
  
FROM        employees  
  
LEFT OUTER JOIN departments  
  
ON (employees.department_id= departments. department_id);
```

```
SELECT      *  
  
FROM        departments  
  
RIGHT OUTER JOIN employees  
  
ON (employees.department_id= departments. department_id);
```

DEPARTMENTS
department_id
department_name
manager_id
location_id

EMPLOYEES
employee_id
first_name
last_name
email
phone_number
hire_date
job_id
salary
commission_pct
manager_id
department_id

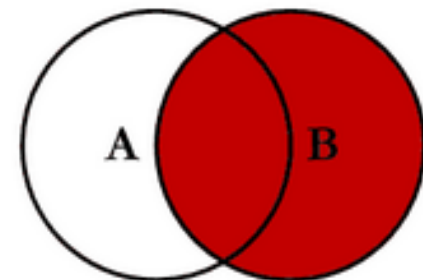
RIGHT OUTER JOIN

SELECT	attribute
FROM	A
RIGHT OUTER JOIN	B
USING	(attribute) ;

- RIGHT OUTER JOIN ใช้สำหรับการแสดงแถวในตารางที่อยู่ทางขวามือออกมาทั้งหมด ถึงแม้จะมีแถวในตารางขวามือมีค่าของฟิลด์ที่ไม่ตรงกับฟิลด์เชื่อมของตารางทางซ้ายมือ ส่วนตารางทางซ้ายมือ ก็จะ return ค่าเป็น NULL สำหรับแถวที่เงื่อนไขไม่ match กับตารางหลัก
- จะรวมข้อมูลที่ตรงกัน โดยจะเน้นตารางฝั่งขวาเป็นหลัก
- Matched pairs are retained and any unmatched values in B table are retained too, but *any unmatched values in A table are eliminated.*

Result set

- Matched values from result of Join-using
- Unmatched values of attributes in B



```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key
```

RIGHT OUTER JOIN

นำไปประยุกต์ตอบโจทย์ >> แสดงรหัส ชื่อสินค้าและราคาที่ไม่มีการสั่งซื้อจากลูกค้า

1. Yields **result from ORDER** join **PRODUCT**
2. Plus **all rows in PRODUCT** table, including those that do not have a matching value in the ORDER table

```
SELECT ORDER_NO, CUSTOMER_NO,
       P_CODE, P_NAME, PRICE
FROM   ORDER
RIGHT OUTER JOIN PRODUCT
On (Order.P_CODE = Product.P_CODE) ;
```

ORDER_NO	CUSTOMER_NO	P_CODE	P_NAME	PRICE
1	C001	111110	สมุด	120
2	C002	222220	คอมพิวเตอร์	30000
NULL	NULL	333330	ปากกา	500

1. Matched values
from result of
ORDER \bowtie PRODUCT

PRODUCT

P_CODE	P_NAME	PRICE
222220	คอมพิวเตอร์	30000
111110	สมุด	120
333330	ปากกา	500

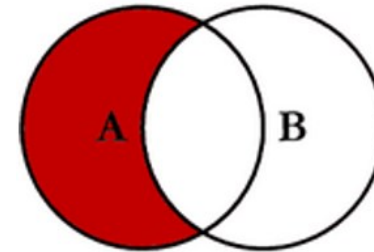
2. Unmatched values from
PRODUCT

จะทราบสินค้าที่ไม่ถูก
สั่งซื้อ

EXERCISE

1. จงเขียน SQL Query ที่แสดงข้อมูลแผนกที่ยังไม่มีพนักงาน

```
SELECT *
FROM departments
LEFT OUTER JOIN employees
ON (employees.department_id = departments.department_id)
Where employees.department_id IS NULL;
```

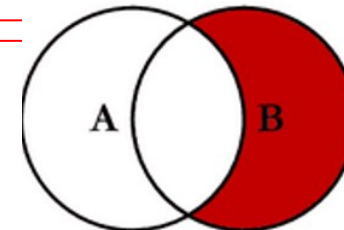


SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL

DEPARTMENTS				
department_id				
department_name				
manager_id				
location_id				

EMPLOYEES				
employee_id				
first_name				
last_name				
email				
phone_number				
hire_date				
job_id				
salary				
commission_pct				
manager_id				
department_id				

```
SELECT *
FROM employees
RIGHT OUTER JOIN departments
ON (employees.department_id = departments.department_id)
Where employees.department_id IS NULL;
```



SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL

FULL OUTER JOIN

```
SELECT      attribute
FROM        A
LEFT OUTER JOIN  B
USING      (attribute)
UNION
SELECT      attribute
FROM        A
RIGHT OUTER JOIN B
USING      (attribute) ;
```

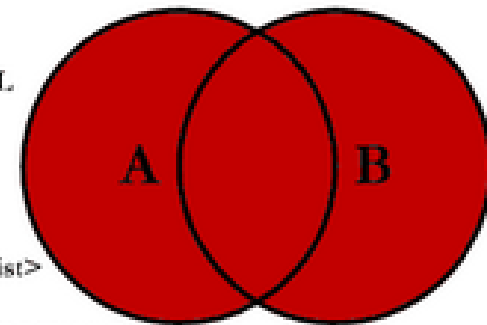
- เป็นการ join ที่ขยายมาจาก join-using คือ สามารถแสดงข้อมูลแถวที่ไม่สามารถจับคู่เงื่อนไขเท่ากันกับอีกตาราง ซึ่งปกติจะไม่แสดงเวลาใช้ join-using
- จะแสดงแถวในตารางที่ อยู่ทั้งทางซ้ายและขวา ของเงื่อนไขออกมาทั้งหมด ตารางใดไม่ match กับเงื่อนไข ก็จะมี return ค่าออกมาเป็น NULL
- Process of Outer Join is the same as that of Join-using, but the result of Outer Join is

Result-set

1. Matched values from result of Join-using
2. Unmatched values of attributes in A
3. Unmatched values of attributes in B

```
-- JOIN TableB B
-- L.Key = B.Key
-- RE B.Key IS NULL
```

```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
```



EXAMPLE: FULL OUTER JOIN

Full Outer Join ใช้กับ Oracle
ถ้า MySQL ใช้ LEFT OUTER JOIN
UNION RIGHT OUTER JOIN

ORDER

ORDER_NO	CUSTOMER_NO	P_CODE
1	C001	111110
2	C002	222220

PRODUCT

P_CODE	P_NAME	PRICE
222220	คอมพิวเตอร์	30000
111110	สมุด	120
333330	ปากกา	500

```
SELECT  ORDER_NO, CUSTOMER_NO, P_CODE,
        P_NAME, PRICE
FROM    ORDER
FULL OUTER JOIN  PRODUCT
On      (Order.P_CODE = Product.P_CODE) ;
```

EXAMPLE: FULL OUTER JOIN

```
SELECT  ORDER_NO, CUSTOMER_NO, P_CODE,
        P_NAME, PRICE
FROM    ORDER
FULL OUTER JOIN  PRODUCT
On (Order.P_CODE = Product.P_CODE) ;
```

ORDER_NO	CUSTOMER_NO	P_CODE	P_NAME	PRICE
1	C001	111110	สมุด	120
2	C002	222220	คอมพิวเตอร์	30000
NULL	NULL	333330	ปากกา	500

1. Matched values from result of ORDER \bowtie PRODUCT

2. Unmatched values from PRODUCT

3. NOT FOUND
Unmatched values from ORDER

PRODUCT

P_CODE	P_NAME	PRICE
222220	คอมพิวเตอร์	30000
111110	สมุด	120
333330	ปากกา	500

ORDER

ORDER_NO	CUSTOMER_NO	P_CODE
1	C001	111110
2	C002	222220

RECALL EXAMPLE: JOIN-USING

Note: EMPLOYEES and DEPARTMENTS tables have more than one column matches (department_id, manager_id). So, USING clause was used to match only one column

```
SELECT    last_name, department_id, department_name
FROM      employees
JOIN      departments
ON        (employees.department_id = departments.department_id)
```

LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
Whalen	10	Administration
Hartstein	20	Marketing
Fay	20	Marketing
Raphaely	30	Purchasing
Khoo	30	Purchasing
Baida	30	Purchasing
Colmenares	30	Purchasing
Himuro	30	Purchasing
Tobias	30	Purchasing
.....		
Sciarra	100	Finance
Urman	100	Finance
Popp	100	Finance
Higgins	110	Accounting
Gietz	110	Accounting

7.2 LEFT OUTER JOIN

```
SELECT employees.last_name, employees.department_id, department_name
FROM           employees
LEFT OUTER JOIN departments
ON             (employees.department_id = departments.department_id)
```

LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
Whalen	10	Administration
Fay	20	Marketing
Hartstein	20	Marketing
Colmenares	30	Purchasing
Himuro	30	Purchasing
....		
Faviet	100	Finance
Greenberg	100	Finance
Gietz	110	Accounting
Higgins	110	Accounting
Grant		

107 rows selected.

7.3 RIGHT OUTER JOIN

```
SELECT employees.last_name, employees.department_id, department_name
FROM          employees
RIGHT OUTER JOIN departments
ON            (employees.department_id = departments.department_id)
```

LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
Whalen	10	Administration
....		
Gietz	110	Accounting
		Treasury
		Corporate Tax
		Control And Credit
		Shareholder Services
		Benefits
		Manufacturing
		Construction
		Contracting
		Operations
		IT Support
		NOC
		IT Helpdesk
		Government Sales
		Retail Sales
		Recruiting
		Payroll

122 rows selected.

7.1 FULL OUTER JOIN (ORACLE)

```
SELECT      last_name, department_id, department_name
FROM        employees
FULL OUTER JOIN departments
USING      (department_id);
```

LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
Whalen	10	Administration
....		
Higgins	110	Accounting
Grant		
		Treasury
		Corporate Tax
		Control And Credit
		Shareholder Services
		Benefits
		Manufacturing
		Construction
		Contracting
		Operations
		IT Support
		NOC
		IT Helpdesk
		Government Sales
		Retail Sales
		Recruiting
		Payroll

123 rows selected.

7.1 FULL OUTER JOIN

```
SELECT      last_name, department_id, department_name
FROM        employees
FULL OUTER JOIN departments
USING      (department_id);
```

(ORACLE)

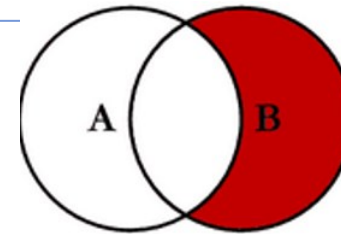
```
SELECT employees.last_name, employees.department_id, department_name
FROM      employees
LEFT OUTER JOIN departments
ON        (employees.department_id = departments.department_id)
UNION
SELECT employees.last_name, departments.department_id, department_name
FROM      employees
RIGHT OUTER JOIN departments
ON        (employees.department_id = departments.department_id);
```

(MySQL)

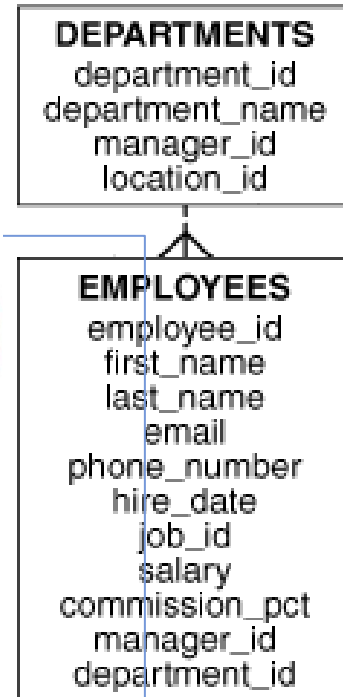
EXERCISE # 1

1. จงเขียน SQL Query ที่แสดงชื่อแผนกที่ยังไม่มีพนักงาน

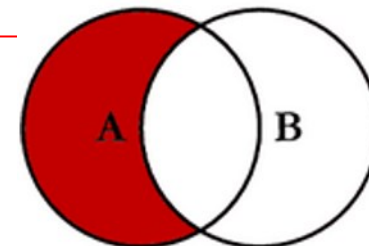
```
SELECT    department_name
FROM      employees
RIGHT OUTER JOIN  departments
ON  (employees.department_id = departments.department_id)
WHERE     employees.department_id IS NULL;
```



SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL



```
SELECT    department_name
FROM      departments
LEFT OUTER JOIN  employees
ON  (departments.department_id = employees.department_id)
WHERE     employees.department_id IS NULL;
```



SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL

PRACTICE

- Assume we have two relations:

จงเขียน SQL query

- แสดงข้อมูลเมนูต่างๆที่ไม่ได้รับการสั่งจากลูกค้าคนใดเลย
- แสดงข้อมูลลูกค้าที่สั่งอาหารแต่ไม่มีอาหารนั้นในตารางเมนู
- แสดงข้อมูลลูกค้าพร้อมทั้งเมนูต่างๆที่ลูกค้าสั่งในเมนู
- แสดงข้อมูลลูกค้าพร้อมทั้งเมนูต่างๆที่ลูกค้าสั่งและข้อมูลอื่นๆ ที่ไม่สามารถจับคู่ตามเงื่อนไขได้

Menu

Food	Day
Pizza	Monday
Hamburger	Tuesday
Chicken	Wednesday
Pasta	Thursday
Tacos	Friday

Customers

Name	Age	Food
Alice	21	Hamburger
Bill	24	Pizza
Carl	23	Beer
Dina	19	Shrimp



PRACTICE 2

2. แสดงข้อมูลของเมนูทั้งหมดที่ไม่ได้รับการสั่งจากลูกค้าคนใดเลย

```
SELECT      Menu.food,    day
FROM        menu
LEFT OUTER JOIN  customers
ON          (menu.food = customers.food)
WHERE       customers.food IS NULL ;
```

Menu

Food	Day
Pizza	Monday
Hamburger	Tuesday
Chicken	Wednesday
Pasta	Thursday
Tacos	Friday

Customers

```
SELECT      Menu.food,    day
FROM        customers
RIGHT OUTER JOIN  menu
ON          (menu.food = customers.food)
WHERE       customers.food IS NULL ;
```

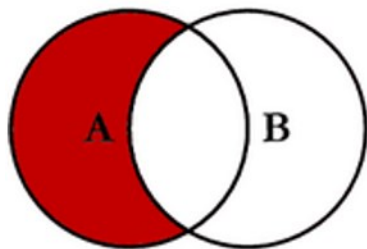
Name	Age	Food
Alice	21	Hamburger
Bill	24	Pizza
Carl	23	Beer
Dina	19	Shrimp



PRACTICE 3

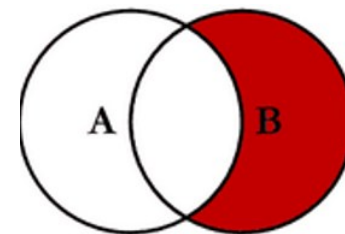
3. แสดงชื่อ และอายุของลูกค้า ที่สั่งอาหารแต่ไม่มีอาหารนั้นในตารางเมนู

```
SELECT      name, age
FROM        customers
LEFT OUTER JOIN  menu
ON          (menu.food = customers.food)
WHERE       Menu.food IS NULL ;
```



```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL.
```

```
SELECT      name, age
FROM        menu
RIGHT OUTER JOIN customers
ON          (menu.food = customers.food)
WHERE       Menu.food IS NULL ;
```



```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL.
```



PRACTICE 4

4. แสดง ชื่อลูกค้า อายุ ชื่อเมนูและวันที่ลูกค้าสั่งในเมนู

```
SELECT          name, age, food, day
FROM            customers
NATURAL JOIN    menu ;
```

```
SELECT          name, age, customers.food,day
FROM            customers
JOIN            menu
ON              (menu.food = customers.food);
```



PRACTICE 5

5. แสดงชื่อลูกค้า อายุ ชื่อเมนู และวันที่ลูกค้าสั่งเมนูต่างๆ และข้อมูลอื่นๆ ที่ไม่สามารถจับคู่ตามเงื่อนไขได้

ORACLE

```
SELECT      name,  age, food, day
FROM        customers

FULL OUTER JOIN  menu

ON          (menu.food =
customers.food);
```

MySQL

```
SELECT      name, age,customers.food,  day
FROM        customers

LEFT OUTER JOIN      menu


ON  (menu.food = customers.food)

UNION

SELECT      name, age, Menu.food, day
FROM        customers

RIGHT OUTER JOIN  menu

ON  (menu.food = customers.food) ;
```



SUMMARY

○ Cartesian Products

- A Cartesian product results in a display of all combinations of rows.
- This is done by either omitting the WHERE clause or specifying the CROSS JOIN clause.

○ Table Aliases

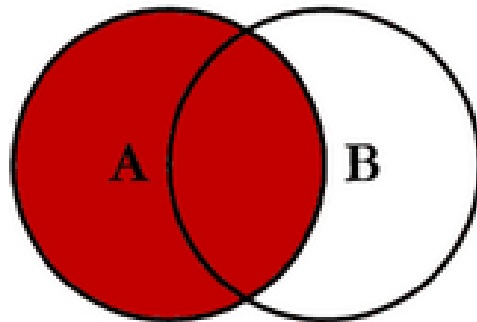
- Table aliases speed up database access.
- Table aliases can help to keep SQL code smaller by conserving memory.

○ There are multiple ways to join tables. **Types of Joins**

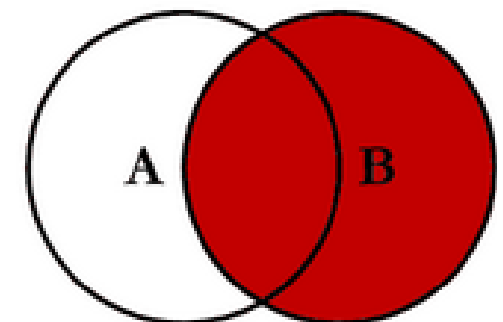
- Cross joins
- Equijoins
- Outer joins
- Self-joins
- Natural joins
- Full (or two-sided) outer joins

JOIN CLASSIFICATION	JOIN TYPE	SQL SYNTAX EXAMPLE	DESCRIPTION
CROSS	CROSS JOIN	SELECT * FROM T1, T2	Returns the Cartesian product of T1 and T2 (old style).
		SELECT * FROM T1 CROSS JOIN T2	Returns the Cartesian product of T1 and T2.
INNER	Old-Style JOIN	SELECT * FROM T1, T2 WHERE T1.C1=T2.C1	Returns only the rows that meet the join condition in the WHERE clause (old style). Only rows with matching values are selected.
	NATURAL JOIN	SELECT * FROM T1 NATURAL JOIN T2	Returns only the rows with matching values in the matching columns. The matching columns must have the same names and similar data types.
	JOIN USING	SELECT * FROM T1 JOIN T2 USING (C1)	Returns only the rows with matching values in the columns indicated in the USING clause.
	JOIN ON	SELECT * FROM T1 JOIN T2 ON T1.C1=T2.C1	Returns only the rows that meet the join condition indicated in the ON clause.
OUTER	LEFT JOIN	SELECT * FROM T1 LEFT OUTER JOIN T2 ON T1.C1=T2.C1	Returns rows with matching values and includes all rows from the left table (T1) with unmatched values.
	RIGHT JOIN	SELECT * FROM T1 RIGHT OUTER JOIN T2 ON T1.C1=T2.C1	Returns rows with matching values and includes all rows from the right table (T2) with unmatched values.
	FULL JOIN	SELECT * FROM T1 FULL OUTER JOIN T2 ON T1.C1=T2.C1	Returns rows with matching values and includes all rows from both tables (T1 and T2) with unmatched values.

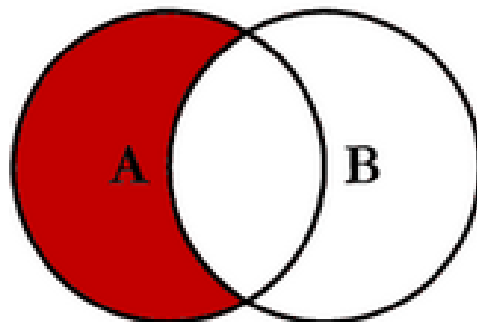
SQL JOINS



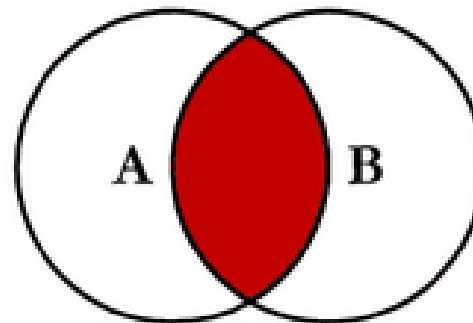
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```



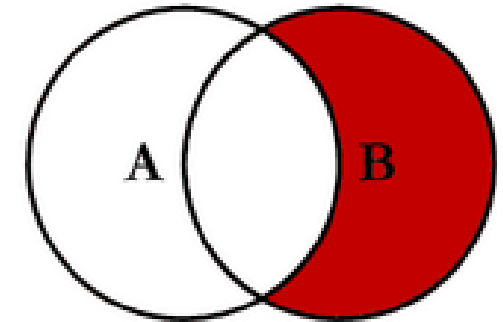
```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```



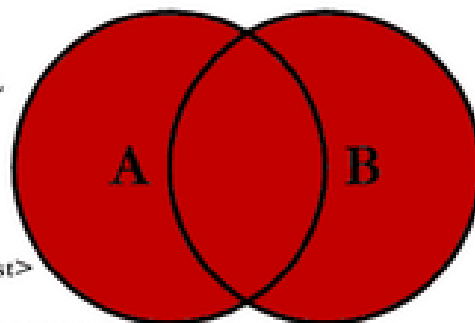
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
```



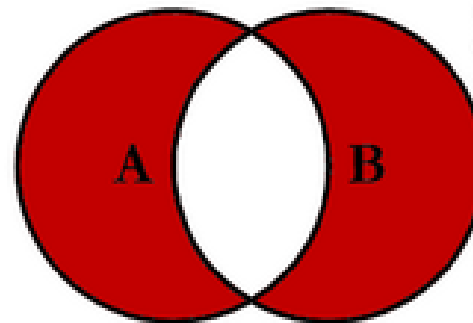
```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```



```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
```



```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
```



```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
```


JOINING TABLES USING SQL:1999 SYNTAX

- Use a join to query data from more than one table:

```
SELECT table1.column, table2.column  
FROM   table1  
  [CROSS JOIN table2] |  
  [NATURAL JOIN table2] |  
  [JOIN table2 USING (column_name) ] |  
  [JOIN table2  
    ON (table1.column_name = table2.column_name) ] |  
  [LEFT | RIGHT | FULL OUTER JOIN table2  
    ON (table1.column_name = table2.column_name) ] ;
```