Lecture 12: SQL Join continue

BADM/ACCY 352

Spring 2017

Instructor: Yi Yang, PhD

Previous lecture

- SQL JOIN
- Natural JOIN, Inner JOIN

```
SELECT columns FROM table1 NATURAL JOIN table2;
```

```
SELECT columns FROM table1
INNER JOIN table2 ON join_condition;
```

MySQL question in test1

List the gallery whose total painting price is greater than 5000.

```
SELECT GAL_NUM, SUM(PTNG_PRICE) FROM PAINTING
GROUP BY GAL_NUM WHERE SUM(PTNG_PRICE)>5000;
```

```
#method 1
SELECT GAL_NUM, SUM(PTNG_PRICE) FROM PAINTING
GROUP BY GAL_NUM
HAVING SUM(PTNG_PRICE) > 5000;

#method 2
SELECT GAL_NUM, SUM(PTNG_PRICE) FROM PAINTING
WHERE SUM(PTNG_PRICE) > 5000
GROUP BY GAL_NUM ;
```

Motivation

Gallery

GAL_NUM	GAL_OWNER	GAL_PHONE
5	Alice	123-4456
6	Waters	353-2243

Painter

PTR_NUM	PTR_NAME	PTR_AREACODE	PTR_PHONE
123	Ross	901	885-4567
126	Itero	901	346-1112
127	Geoff	615	221-4456

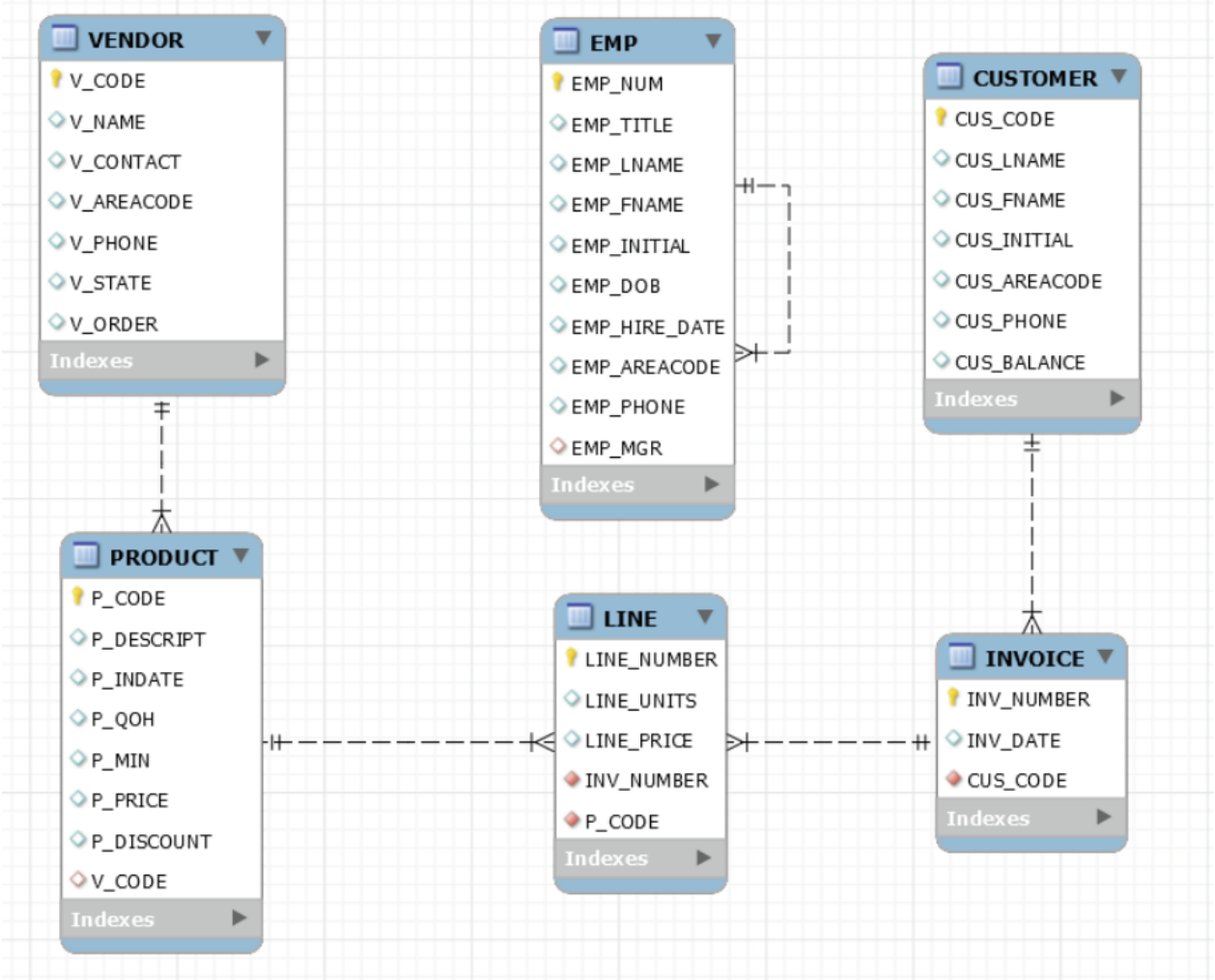
Painting

PTNG_NUM	PTNG_TITLE	PTNG_PRICE	PTR_NUM	GAL_NUM
1338	Dawn Thunder	245.50	123	5
1339	A Faded Rose	6723.00	123	NULL
1340	The Founders	567.99	126	6
1341	Hasty Pudding Exit	145.50	123	NULL
1342	Plastic Paradise	8328.99	126	6
1343	Roamin'	785.00	127	6
1344	Wild Waters	999.00	127	5
1345	Stuff 'n Such 'n Some	9800.00	123	5

PTNG_NUM	PTNG_TITLE	PTNG_PRICE	PTR_NUM	GAL_NUM
1338	Dawn Thunder	245.50	123	5
1339	A Faded Rose	6723.00	123	NULL
1340	The Founders	567.99	126	6
1341	Hasty Pudding Exit	145.50	123	NULL
1342	Plastic Paradise	8328.99	126	6
1343	Roamin'	785.00	127	6
1344	Wild Waters	999.00	127	5
1345	Stuff 'n Such 'n Some	9800.00	123	5

PAINTING

PTNG_NUM	PTNG_TITLE	PTNG_PRICE	PTR_NUM	PTR_NAME	PTR_AREACODE	PTR_PHONE	GAL_NUM	GAL_OWNER	GAL_PHONE
1338	Dawn Thunder	245.50	123	Ross	901	885-4567	5	Alice	123-4456
1339	A Faded Rose	6723.00	123	Ross	901	885-4567	NULL	NULL	NULL
1340	The Founders	567.99	126	Itero	901	346-1112	6	Waters	353-2243
1341	Hasty Pudding Exit	145.50	123	Ross	901	885-4567	NULL	NULL	NULL
1342	Plastic Paradise	8328.99	126	Itero	901	346-1112	6	Waters	353-2243
1343	Roamin'	785.00	127	Geoff	615	221-4456	6	Waters	353-2243
1344	Wild Waters	999.00	127	Geoff	615	221-4456	5	Alice	123-4456
1345	Stuff 'n Such 'n Some	9800.00	123	Ross	901	885-4567	5	Alice	123-4456



- The manager wants to know customers' product purchase history.
- Note that customer table does not contain anything about product, and product table does not contain anything about customer.

CUSTOMER

CUS_CODE	CUS_LNAME	CUS_FNAME
10010	Ramas	Alfred
10011	Dunne	Leona
10012	Smith	Kathy
10013	Olowski	Paul
10014	Orlando	Myron
10015	O'Brian	Amy
10016	Brown	James
10017	Williams	George
10018	Farriss	Anne
10019	Smith	Olette

INVOICE

INV_NUMBER	CUS_CODE	INV_DATE
1001	10014	2014-01-16 00:00:00
1002	10011	2014-01-16 01:00:00
1003	10012	2014-01-16 01:00:00
1004	10011	2014-01-17 00:00:00
1005	10018	2014-01-17 01:00:00
1006	10014	2014-01-17 02:00:00
1007	10015	2014-01-17 02:00:00
1008	10011	2014-01-17 02:00:00

LINE

INV_NUMBER	LINE_NUMBER	P_CODE	LINE_UNITS	LINE_PRICE
1001	1	13-Q2/P2	1	14.99
1001	2	23109-HB	1	9.95
1002	1	54778-2T	2	4.99
1003	1	2238/QPD	1	38.95
1003	2	1546-QQ2	1	39.95
1003	3	13-Q2/P2	5	14.99
1004	1	54778-2T	3	4.99
1004	2	23109-HB	2	9.95
1005	1	PVC23DRT	12	5.87
1006	1	SM-18277	3	6.99
1006	2	2232/QTY	1	109.92
1006	3	23109-HB	1	9.95

PRODUCT

P_CODE	P_DESCRIPT
11QER/31	Power painter, 15 psi., 3-nozzle
13-Q2/P2	7.25-in. pwr. saw blade
14-Q1/L3	9.00-in. pwr. saw blade
1546-QQ2	Hrd. cloth, 1/4-in., 2x50
1558-QW1	Hrd. cloth, 1/2-in., 3x50
2232/QTY	B&D jigsaw, 12-in. blade
2232/QWE	B&D jigsaw, 8-in. blade
2238/QPD	B&D cordless drill, 1/2-in.
23109-HB	Claw hammer
23114-AA	Sledge hammer, 12 lb.
54778-2T	Rat-tail file, 1/8-in. fine
89-WRE-Q	Hicut chain saw, 16 in.
PVC23DRT	PVC pipe, 3.5-in., 8-ft

```
SELECT CUS_CODE,CUS_LNAME, CUS_FNAME, P_DESCRIPT
FROM CUSTOMER NATURAL JOIN INVOICE
NATURAL JOIN LINE NATURAL JOIN PRODUCT;
```

CUS_CODE	CUS_LNAME	CUS_FNAME	P_DESCRIPT
10014	Orlando	Myron	7.25-in. pwr. saw blade
10014	Orlando	Myron	Claw hammer
10011	Dunne	Leona	Rat-tail file, 1/8-in. fine
10012	Smith	Kathy	B&D cordless drill, 1/2-in.
10012	Smith	Kathy	Hrd. cloth, 1/4-in., 2x50
10012	Smith	Kathy	7.25-in. pwr. saw blade
10011	Dunne	Leona	Rat-tail file, 1/8-in. fine
10011	Dunne	Leona	Claw hammer
10018	Farriss	Anne	PVC pipe, 3.5-in., 8-ft
10014	Orlando	Myron	1.25-in. metal screw, 25
10014	Orlando	Myron	B&D jigsaw, 12-in. blade

Facebook

- When you log in your FB account, your profile, friends, posts, comments, likes are retrieved from FB database in a second.

- BTW, how does FB retrieve your information from 2billion user database so fast? And why does database JOIN operation join multiple tables so fast?

- Using index card.

N8670
.A65 **The Annual art sales index.** 1971/72-
[Weybridge, Eng.]
v. illus. 22 cm.

SEE CONT. CARD

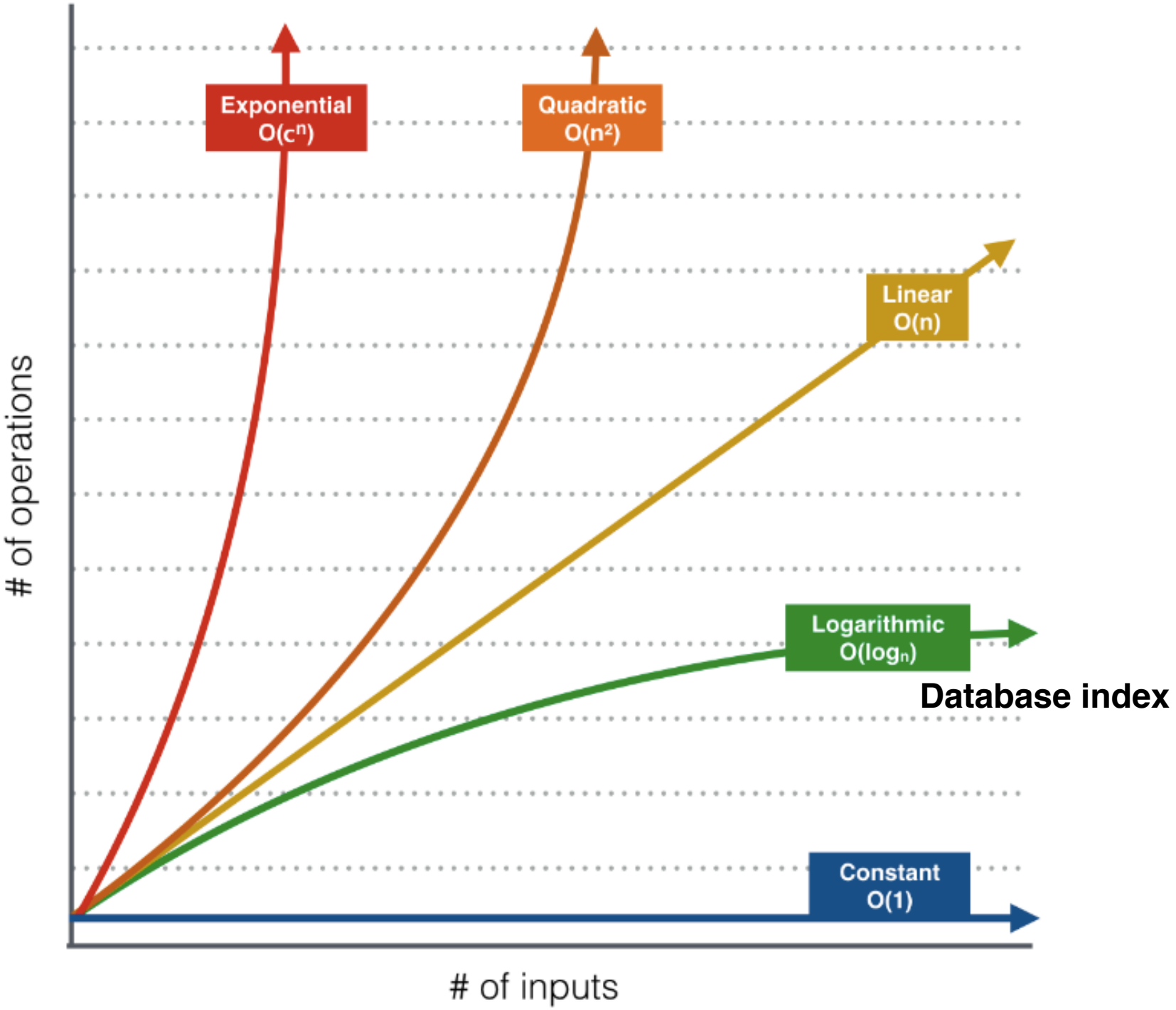
1. Art—Prices—Indexes—Yearbooks.

N8670.A65 700'.75 MB 73-645118
MARC-S

Library of Congress 73 [2]

Database index

- An index is created on a column of a table, which improves the data retrieval speed.
- Database index helps you to find data quickly. It improves the efficiency of searches.
- RDBMS automatically creates an index for primary key.
- Creating index is a trade-off decision.
- A common practice is to create an index on column that is used as a search key



Database index

Cardinality

- In SQL, the term cardinality refers to the uniqueness of data values contained in a particular column of a database table. The lower the cardinality, the more duplicated elements in a column.
- Gender column: low or high cardinality?
- Another common practice is to NOT create an index on columns with low-cardinality.

What is inner/natural join missing

gallery

GAL_NUM	GAL_OWNER	GAL_PHONE
5	Alice	123-4456
6	Waters	353-2243

painting

PTNG_NUM	PTNG_TITLE	PTNG_PRICE	PTR_NUM	GAL_NUM
1338	Dawn Thunder	245.50	123	5
1339	A Faded Rose	6723.00	123	NULL
1340	The Founders	567.99	126	6
1341	Hasty Pudding Exit	145.50	123	NULL
1342	Plastic Paradise	8328.99	126	6
1343	Roamin'	785.00	127	6
1344	Wild Waters	999.00	127	5
1345	Stuff 'n Such 'n Some	9800.00	123	5

What if you natural join or inner join two tables?

```
SELECT * FROM GALLERY INNER JOIN PAINTING
ON GALLERY.GAL_NUM = PAINTING.GAL_NUM;
```

or

```
SELECT * FROM GALLERY NATURAL JOIN PAINTING;
```

GAL_NUM	GAL_OWNER	GAL_PHONE	PTNG_NUM	PTNG_TITLE	PTNG_PRICE	PTR_NUM	GAL_NUM
5	Alice	123-4456	1338	Dawn Thunder	245.50	123	5
6	Waters	353-2243	1340	The Founders	567.99	126	6
6	Waters	353-2243	1342	Plastic Paradise	8328.99	126	6
6	Waters	353-2243	1343	Roamin'	785.00	127	6
5	Alice	123-4456	1344	Wild Waters	999.00	127	5
5	Alice	123-4456	1345	Stuff 'n Such 'n Some	9800.00	123	5

- Paintings with NULL GAL_NUM are not in the joined table.
- Can we modify the inner join query to:

```
SELECT * FROM GALLERY INNER JOIN PAINTING
ON PAINTING.GAL_NUM = GALLERY.GAL_NUM
OR PAINTING.GAL_NUM IS NULL;
```

- No, because the inner join without join-condition results in a relational product.

GAL_NUM	GAL_OWNER	GAL_PHONE	PTNG_NUM	PTNG_TITLE	PTNG_PRICE	PTR_NUM	GAL_NUM
5	Alice	123-4456	1341	Hasty Pudding Exit	145.50	123	NULL
5	Alice	123-4456	1339	A Faded Rose	6723.00	123	NULL
5	Alice	123-4456	1345	Stuff 'n Such 'n Some	9800.00	123	5
5	Alice	123-4456	1338	Dawn Thunder	245.50	123	5
5	Alice	123-4456	1344	Wild Waters	999.00	127	5
6	Waters	353-2243	1339	A Faded Rose	6723.00	123	NULL
6	Waters	353-2243	1341	Hasty Pudding Exit	145.50	123	NULL
6	Waters	353-2243	1340	The Founders	567.99	126	6
6	Waters	353-2243	1342	Plastic Paradise	8328.99	126	6
6	Waters	353-2243	1343	Roamin'	785.00	127	6

Outer JOIN

- An outer join returns not only the rows matching the join condition, it returns the rows with unmatched values.
- It includes: left join, right join, full join.

```
SELECT columns FROM table1 JOIN table2 on....
```

Left JOIN

- It returns the rows matching the join condition, plus the rows in the left table with unmatched values.
- Syntax:

```
SELECT columns FROM table1
LEFT JOIN table2
ON join_condition
```

Left Join example

EMP

ID	NAME	CITY_ID
11	Alice	1
22	Bob	1
33	Charlie	3
44	David	NULL

LEFT JOIN

CITY

CITY_ID	CITY
1	Chicago
2	New York
3	Huston
4	Seattle

```
SELECT * FROM EMP LEFT JOIN CITY
ON EMP.CITY_ID = CITY.CITY_ID;
```

ID	NAME	EMP.CITY_ID	CITY.CITY_ID	CITY
11	Alice	1	1	Chicago
22	Bob	1	1	Chicago
33	Charlie	3	3	Huston
44	David	NULL	NULL	NULL

RIGHT JOIN

- It returns the rows matching the join condition, plus the rows in the right table with unmatched values.
- Syntax:

```
SELECT columns FROM table1
RIGHT JOIN table2
ON join_condition
```

Right Join example

EMP

ID	NAME	CITY_ID
11	Alice	1
22	Bob	1
33	Charlie	3
44	David	NULL

RIGHT JOIN

CITY

CITY_ID	CITY
1	Chicago
2	New York
3	Huston
4	Seattle

```
SELECT * FROM EMP RIGHT JOIN CITY
ON EMP.CITY_ID = CITY.CITY_ID;
```

ID	NAME	EMP.CITY_ID	CITY.CITY_ID	CITY
11	Alice	1	1	Chicago
22	Bob	1	1	Chicago
33	Charlie	3	3	Huston
NULL	NULL	NULL	2	New York
NULL	NULL	NULL	4	Seattle

Should I use a left join or a right join?

- It really doesn't matter.
- The only thing that matters is the order of tables in the **`from`** clause.

FULL JOIN

- It returns the rows matching the join condition, plus the rows with unmatched values in the left and right table.
- However, MySQL does not support FULL JOIN syntax. If you use other RDBMS, the full join syntax is

```
SELECT columns FROM table1
FULL JOIN table2
ON join_condition
```

- We will learn FULL JOIN equivalent in MySQL later.

Full Join example

EMP

ID	NAME	CITY_ID
11	Alice	1
22	Bob	1
33	Charlie	3
44	David	NULL

FULL JOIN

CITY

CITY_ID	CITY
1	Chicago
2	New York
3	Huston
4	Seattle

ID	NAME	EMP.CITY_ID	CITY.CITY_ID	CITY
11	Alice	1	1	Chicago
22	Bob	1	1	Chicago
33	Charlie	3	3	Huston
44	David	NULL	NULL	NULL
NULL	NULL	NULL	2	New York
NULL	NULL	NULL	4	Seattle

Practice

gallery

GAL_NUM	GAL_OWNER	GAL_PHONE
5	Alice	123-4456
6	Waters	353-2243

painting

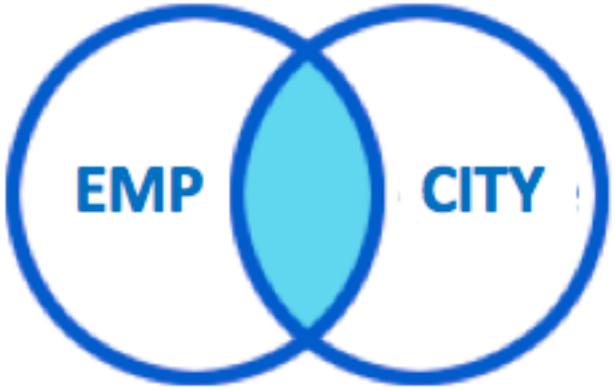
PTNG_NUM	PTNG_TITLE	PTNG_PRICE	PTR_NUM	GAL_NUM
1338	Dawn Thunder	245.50	123	5
1339	A Faded Rose	6723.00	123	NULL
1340	The Founders	567.99	126	6
1341	Hasty Pudding Exit	145.50	123	NULL
1342	Plastic Paradise	8328.99	126	6
1343	Roamin'	785.00	127	6
1344	Wild Waters	999.00	127	5
1345	Stuff 'n Such 'n Some	9800.00	123	5

- List the gallery info of painting ‘Dawn Thunder’ and ‘A Faded Rose’. Can you use subquery?

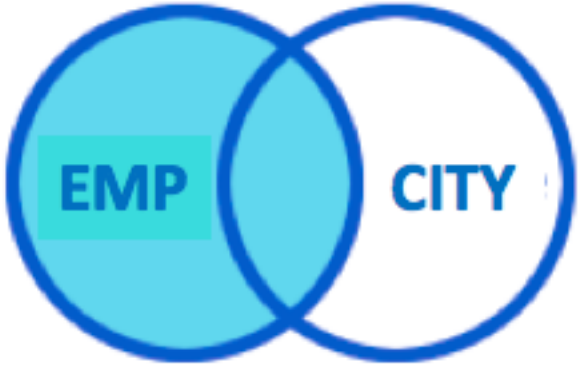
```sql
SELECT PAINTING.PTNG_TITLE, GALLERY.GAL_NUM, GALLERY.GAL_OWNER
FROM GALLERY RIGHT JOIN PAINTING
ON GALLERY.GAL_NUM = PAINTING.GAL_NUM
WHERE PAINTING.PTNG_TITLE = 'Dawn Thunder'
OR PAINTING.PTNG_TITLE = 'A Faded Rose';
```

PTNG_TITLE	GAL_NUM	GAL_OWNER
Dawn Thunder	5	Alice
A Faded Rose	NULL	NULL

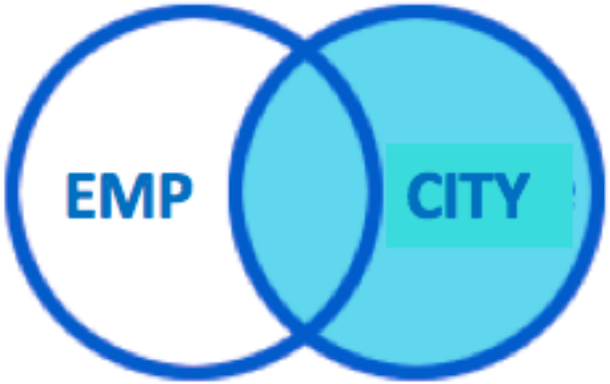
An Intuitive way to understand JOIN (Venn Diagram)



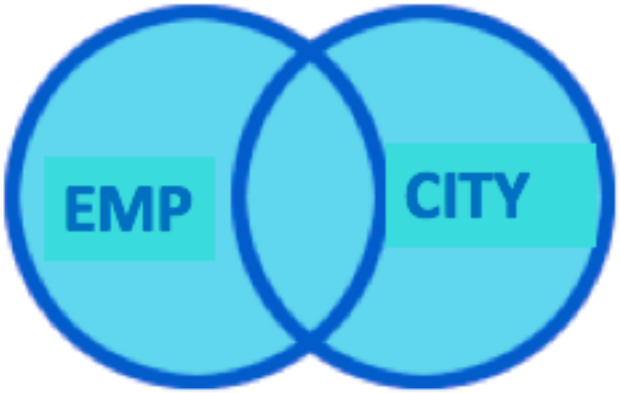
inner/natural join



left join



right join



full join

JOIN summary

- JOIN is used to retrieve data from more than one tables.
- Inner/Natural Join, Left Join, Right Join, Full Join.
- Learning objective:
  - Know the differences between different joins
  - Know how to use inner/natural join, left join, right join.

iClicker question

EMP

ID	NAME	CITY_ID
11	Alice	1
22	Bob	1
33	Charlie	3
44	David	NULL
55	Ellen	3

CITY

CITY_ID	NAME
1	Chicago
2	New York
3	Houston
4	Seattle
5	Champaign

```
select * from EMP inner join CITY on EMP.CITY_ID = CITY.CITY_ID;

select * from EMP left join CITY on EMP.CITY_ID = CITY.CITY_ID;

select * from EMP right join CITY on EMP.CITY_ID = CITY.CITY_ID;
```

How many rows will the above queries return?

A: (4, 5, 7) B: (5,5,6) C: (4, 6, 6) D: (5, 6, 7)

Relational Set Operators

- In relational database, you can combine two or more tables (sets) to create new tables (sets).
- UNION, INTERSECT

UNION

- UNION keyword combines rows from two or more SELECT queries without including duplicate rows.
- Syntax:

```
query UNION query;
```

- The two SELECT statements must return the same number of attributes and similar data types.

UNION example

CUST_1

ID	NAME	CITY
11	Alice	Chicago
22	Bob	New York
33	Charlie	Dallas
44	David	Huston

CUST_2

ID	NAME
90001	Emma
90002	Fox
90003	Bob
90004	David

```
SELECT NAME FROM CUS_1
UNION
SELECT NAME FROM CUS_2;
```

NAME
Alice
Bob
Charlie
David
Emma
Fox

UNION example (cont.)

```
SELECT * FROM CUS_1
UNION
SELECT * FROM CUS_2;
```

"Different number of Columns" error!

```
SELECT ID, NAME FROM CUS_1
UNION
SELECT ID, NAME FROM CUS_2;
```

ID	NAME
11	Alice
22	Bob
33	Charlie
44	David
90001	Emma
90002	Fox
90003	Bob
90004	David

UNION ALL

- Like UNION, UNION ALL combines rows from multiple queries, but the result table retains duplicate rows.

- Syntax:

```
query UNION ALL query;
```

UNION ALL example

CUST_1

ID	NAME	CITY
11	Alice	Chicago
22	Bob	New York
33	Charlie	Dallas
44	David	Huston

CUST_2

ID	NAME
90001	Emma
90002	Fox
90003	Bob
90004	David

```
SELECT NAME FROM CUS_1
UNION ALL
SELECT NAME FROM CUS_2;
```

NAME
Alice
Bob
Charlie
David
Emma
Fox
Bob
David

Contains duplicated rows

FULL JOIN Review

EMP

ID	NAME	CITY_ID
11	Alice	1
22	Bob	1
33	Charlie	3
44	David	NULL

FULL JOIN

CITY

CITY_ID	CITY
1	Chicago
2	New York
3	Huston
4	Seattle

ID	NAME	EMP.CITY_ID	CITY.CITY_ID	CITY
11	Alice	1	1	Chicago
22	Bob	1	1	Chicago
33	Charlie	3	3	Huston
44	David	NULL	NULL	NULL
NULL	NULL	NULL	2	New York
NULL	NULL	NULL	4	Seattle

Simulate Full JOIN in MySQL

Use UNION

```
SELECT * from EMP LEFT JOIN CITY
ON EMP.CITY_ID = CITY.city_id
UNION
SELECT * from EMP RIGHT JOIN CITY
ON EMP.CITY_ID = CITY.city_id
```

INTERSECT

- It returns only the rows that appear in both sets.
- MySQL does not support INTERSECT.
- However, we can get the desired results using alternative MySQL statement.

- For example, how many customers are in both CUS_1 and CUS_2 tables.

INTERSECT Alternative

CUST_1

ID	NAME	CITY
11	Alice	Chicago
22	Bob	New York
33	Charlie	Dallas
44	David	Huston

CUST_2

ID	NAME
90001	Emma
90002	Fox
90003	Bob
90004	David

```
SELECT DISTINCT NAME FROM
CUS_1 INNER JOIN CUS_2
ON CUS_1.NAME = CUS_2.NAME;
```

NAME
Bob
David