Natural Join

* + **The NATURAL JOIN clause is based on all columns in the two tables that have the same name.**
  + **It selects rows from the two tables that have equal values in all matched columns.**
  + **If the columns having the same names have different data types, an error is returned.**

SELECT department\_id, department\_name,

location\_id, city

FROM departments

NATURAL JOIN locations ;

**Using Clause**

**SELECT employees.employee\_id, employees.last\_name,**

**departments.location\_id, department\_id**

**FROM employees JOIN departments**

**USING (department\_id) ;**

**SELECT e.employee\_id, e.last\_name,**

**d.location\_id, department\_id**

**FROM employees e JOIN departments d**

**USING (department\_id) ;**

On Clause

* + **The join condition for the natural join is basically an equijoin of all columns with the same name.**
  + **Use the ON clause to specify arbitrary conditions or specify columns to join.**
  + **The join condition is separated from other search conditions.**
  + **The ON clause makes code easy to understand.**

SELECT e.employee\_id, e.last\_name, e.department\_id,

d.department\_id, d.location\_id

FROM employees e JOIN departments d

ON (e.department\_id = d.department\_id);

Self join

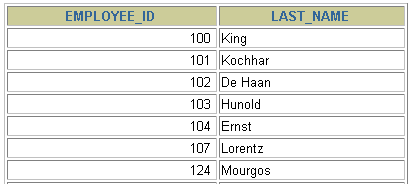
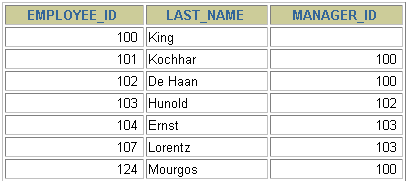
**SELECT e.last\_name emp, m.last\_name mgr**

**FROM employees e JOIN employees m**

**ON (e.manager\_id = m.employee\_id);**

**EMPLOYEES (WORKER)**

**EMPLOYEES (MANAGER)**



**…**

**…**

3 tables join

**SELECT employee\_id, city, department\_name**

**FROM employees e**

**JOIN departments d**

**ON d.department\_id = e.department\_id**

**JOIN locations l**

**ON d.location\_id = l.location\_id;**

**Or**

mysql> SELECT employee\_id, city, department\_name

-> FROM employees e

-> JOIN departments d

-> using ( department\_id)

-> JOIN locations l

-> using ( location\_id);

Left Outer Join

**SELECT e.last\_name, e.department\_id, d.department\_name**

**FROM employees e LEFT OUTER JOIN departments d**

**ON (e.department\_id = d.department\_id) ;**

Right Outer Join

**SELECT e.last\_name, e.department\_id, d.department\_name**

**FROM employees e RIGHT OUTER JOIN departments d**

**ON (e.department\_id = d.department\_id) ;**

Outer Join

**SELECT e.last\_name, d.department\_id, d.department\_name**

**FROM employees e FULL OUTER JOIN departments d**

**ON (e.department\_id = d.department\_id) ;**

Cross Join

* + **The CROSS JOIN clause produces the cross-product of two tables.**
  + **This is also called a Cartesian product between the two tables.**
* **SELECT last\_name, department\_name**
* **FROM employees**
* **CROSS JOIN departments ;**