**1) Natural (Inner ) Join = (Equi Join)**

**Matching records from one table to another. And Only list the matched ones**

a) NATURAL JOIN caluse

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

FROM **employees** NATURAL JOIN **departments**

**Assuming FK definition is there**

**Assuming they use same column data type and column name for PK and FK**

b) USING keyword (clause)

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

FROM employees JOIN departments

USING (department\_id) ;

**Assuming FK definition is there**

**Assuming they use same column data type and column name for PK and FK**

c) ON keyword

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

FROM employees JOIN departments

ON employees.department\_id = departments.department\_id

**Formula:**

emplooyes.department\_id = Departments.department\_id

**child's FK = Parent's PK**

**This is how you use Equi Join with ON with Table name alias**

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

FROM employees e JOIN departments d

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

**1) INNER (Equi Join)**

Question: bring employee's first name and last name and department name they work

**Solution #1 - Natural JOIN keyword**

SELECT first\_name, last\_name , department\_name

FROM employees

NATURAL JOIN departments

**Solution #2 : USING keyword**

SELECT first\_name, last\_name , department\_name

FROM employees JOIN departments

USING (department\_id)

**Solution #3 : use ON clause**

SELECT employees.first\_name, employees.last\_name , departments.department\_name

FROM employees JOIN departments

ON ( employees.department\_id = departments.department\_id )

**Question: List employees first name, last name, job if, salary ,department name and manager\_id of that department**

SELECT e.first\_name, e.last\_name, e.job\_id, e.salary, d.department\_name, d.manager\_id

FROM employees e JOIN departments d

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

**Question: List employees first name, last name, job if, salary ,department name and manager\_id of that department for those who work as SA\_REP or SA\_MAN**

SELECT e.first\_name, e.last\_name, e.salary, e.job\_id, d.department\_name, d.manager\_id

FROM employees e JOIN departments d

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

WHERE Job\_id IN ('SA\_REP' ,'SA\_MAN')

**List of all employees who is working as SA\_REP or SA\_MAN**

SELECT first\_name, last\_name, job\_id, salary

FROM employees

WHERE job\_id IN ('SA\_REP', 'SA\_MAN')

Rule 1:

Formula:

emplooyes.department\_id = Departments.department\_id

child's FK = Parent's PK

Rule 2:

If there are two tables joining you have to have at least (n-1 -> 2-1=)1 ON join condition

if there are three tables joining you have to have at least (3-1=2) 2 ON condition

**Question:**

Get a list of employees who work for department 50 or 80, 90, 110 and also at the

same time who earn salary > 4500

List should include e.fname, e.last\_name, e.salary, d.department\_name,d.manager\_id, l.city, c.country\_name

employees e departments d locations l countries c

SELECT e.first\_name, e.last\_name, e.salary, d.department\_name,d.manager\_id, l.city, c.country\_name

FROM employees e JOIN departments d

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

JOIN locations l

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

JOIN countries c

ON l.country\_id=c.country\_id

WHERE e.department\_id IN (50 , 80, 90, 110) AND e.salary >4500

**July 27th notes**

A) INNER Join (Equi Join) -- Bringing List of records for only the matched ones

Question #1: bring employee's first name and last name and department name they work

Solution #1 - Natural JOIN keyword

SELECT first\_name, last\_name , department\_name

FROM employees

NATURAL JOIN departments

Solution #2 : USING keyword

SELECT first\_name, last\_name , department\_name

FROM employees JOIN departments

USING (department\_id)

Solution #3 : use ON clause

SELECT employees.first\_name, employees.last\_name , departments.department\_name

FROM employees JOIN departments

ON ( employees.department\_id = departments.department\_id )

Solution #4 )enchanced version (using table short cut alias)

SELECT e.first\_name, e.last\_name , d.department\_name

FROM employees e JOIN departments d

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

employees departments

Ersan Cam 40 10 Accounting

James Snow 20 20 Sales

Jimmy Carter null 40 IT

Ersan Cam IT

James Snow Sales

=================

A) INNER JOIN

Question #2: bring some employee's first name and last name and department name they work for

the employees whose working as SA\_REP and IT\_PROG (job\_id) and make salary greater than 5000

Solution #2

SELECT e.first\_name,e.last\_name, d.department\_name

FROM employees e JOIN departments d

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

WHERE e.job\_id IN ( 'SA\_REP' , 'IT\_PROG') and e.salary > 5000

A) INNER JOIN ( 3 table join)

Question #3: List all employees' first name and last name and department name they work, city they work

SELECT e.first\_name, e.last\_name, d.department\_name, city

FROM employees e JOIN departments d

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

JOIN locations l

ON (d.location\_id = l.location\_id)

Question #4: (3 table Join with additional WHERE filter)

List all employees' first name and last name and department name they work, city they work

for those ones who make less than 4000

SELECT e.first\_name, e.last\_name, d.department\_name, city

FROM employees e JOIN departments d

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

JOIN locations l

ON (d.location\_id = l.location\_id)

WHERE e.salary < 4000

**2) SELF JOIN**

Sometimes database designers do not create separate table to have Parent Child replationship because that parent child relationship stays in the same table.

For instance in Employees table there is all types of employees, such as clerk, agent, managers , directors (manager’s manager) and Sr.Directors (Manager of Directors) so you can not keep creating separate table for each. What you do is you created Foreign key inside same table.

Employee\_id employee\_name manager\_id manager\_name

105 Ersan 121 Jason

121 Jason 137 Luke

137 Luke

In this example Ersan’s manager is Jason. Jason’s manager is Luke

To find this relationship we use Self join..

Self join means that we join employees table twice and each time we give different name

SELECT

FROM Employees worker JOIN Employees manager

ON

We called same table with alias as worker and also manager because in the same table we have employees and managers togetger..

**Here is the trick. When you build ON condition you have to choose correct formula.**

**Employees’ manager id must be equal (match) to Manager’s his/her employee id**

**ON worker.manager\_id = manager.employee\_id**

**Question:**

show which employee (first\_name, last\_name, salary) reports to which other employees (first\_name, last\_name)

**SELECT wrk.first\_name, wrk.last\_name, wrk.salary , mgr.first\_name, mgr.last\_name**

**FROM employees wrk JOIN employees mgr**

**ON wrk.manager\_id = mgr.employee\_id**

**Practice Questions**

1. Determine which books customer Jake Lucas has purchased. Perform the search using the customer’s name, not the customer number. If he has purchased multiple copies of the same book , unduplicated (DISTINCT)

Er

2 ) Create a list that displays the title of each book and the name and phone number of the contact at the publisher’s office for reordering each book

**OUTHER JOIN :** in between two tables join relation ,this will not only bring match records but also unmatched records as well

( LEFT OUTHER , RIGHT OUTHER JOIN ,, FULL OUTHER JOIN)

**Scenario #1: If you are missing employee records because employee does not have any department defined**

**Employees**

|  |  |  |
| --- | --- | --- |
| **Employee\_id** | **Name** | **Department\_id** |
| 121 | E | 10 |
| 122 | K | 10 |
| 123 | C | 80 |
| 124 | M | 90 |
| 125 | CB | null |

**Departments**

|  |  |
| --- | --- |
| Department\_id | Department\_name |
| 10 | Accounting |
| 80 | Sales |
| 90 | Finance |
| 180 | Marketing |

SELECT e.first\_name, e.last\_name,e.salary, d.department\_name

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

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

121 E Accounting

122 K accounting

123 C Sales

123 M Finance

125 CB null

**Scenario #2 : If there are some departments being created in Departments but no employees working in then INNER (Equi ) join will not list them**

**Employees**

|  |  |  |
| --- | --- | --- |
| **Employee\_id** | **Name** | **Department\_id** |
| 121 | E | 10 |
| 122 | K | 10 |
| 123 | C | 80 |
| 124 | M | 90 |
| 125 | CB | null |

**Departments**

|  |  |
| --- | --- |
| Department\_id | Department\_name |
| 10 | Accounting |
| 80 | Sales |
| 90 | Finance |
| 180 | Marketing |

SELECT e.first\_name, e.last\_name,e.salary, d.department\_name

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

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

121 E Accounting

122 K Accounting

123 C Sales

123 M Finance

Null null Marketing

**Scenario #3 FULL OUTHER JOIN   
Matched records + missing records from left + missing records right**

SELECT e.first\_name, e.last\_name,e.salary, d.department\_name

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

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