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

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

a) NATURAL JOIN (not preferred)

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 (not preferred)

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 (Preferred option for Inner – Equi join)

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

**Entire employee list**

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

**Selected employee list**

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')

show me 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

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

**2) SELF JOIN**

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

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

Sub options of Outher join ( LEFT OUTER JOIN , RIGHT OUTER JOIN , FULL OUTER JOIN)

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

**Employees**

|  |  |  |  |
| --- | --- | --- | --- |
| **Employee\_id** | **Name** | **Salary** | **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 |

If you use regular inner (Equi join) then there will be only 4 mathing records

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

FROM **employees** e JOIN **departments** d

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

But I f you want to include an employee even though employee doesn’t belong to any dept.

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

FROM **employees** e LEFT OUTER 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 there 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 OUTER JOIN   
Matched records + missing records from left table + missing records right table**

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)

1. Non Equi (Range Join)

**Employees**

|  |  |  |  |
| --- | --- | --- | --- |
| **Employee\_id** | **Name** | **Salary** | **Department\_id** |
| 121 | E | 4553.50 | 10 |
| 122 | K | 3467.81 | 10 |
| 123 | C | 5000 | 80 |
| 124 | M | 5341.60 | 90 |
| 125 | CB | 7000 | null |

There is no Level information among employees based on their salary

And it is not wise to create a column in employee table and store this info because constantly it changes

**Employees**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Employee\_id** | **Name** | **Salary** | **Department\_id** | **Level** |
| 121 | E | 4553.50 | 10 | C |
| 122 | K | 3467.81 | 10 | B |
| 123 | C | 5000 | 80 | D |
| 124 | M | 5341.60 | 90 | D |
| 125 | CB | 7000 | null | E |

Or another unsuccesfull option would be to use equi join

**Employees**

|  |  |  |  |
| --- | --- | --- | --- |
| **Employee\_id** | **Name** | **Salary** | **Department\_id** |
| 121 | E | 4553.50 | 10 |
| 122 | K | 3467.81 | 10 |
| 123 | C | 5000 | 80 |
| 124 | M | 5341.60 | 90 |
| 125 | CB | 7000 | null |

Salary\_Level

|  |  |
| --- | --- |
| Level | Salary |
| A | 1000 |
|  | 1001.01 |
| A | 1001.02 |
|  | …  1999.99 |

Successful version would be to use Range join

**Employees**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Employee\_id** | **Name** | **Salary** | **Department\_id** |  | **Make up column** |
| 121 | E | 4553.50 | 10 |  | D |
| 122 | K | 3467.81 | 10 |  | C |
| 123 | C | 5000 | 80 |  | D |
| 124 | M | 5341.60 | 90 |  |  |
| 125 | CB | 7000 | null |  |  |

Salary\_Level

|  |  |  |
| --- | --- | --- |
| Grade\_level | Lowest\_sal | Highest\_al |
| A | 1000 | 2000 |
| B | 2001 | 3000 |
| C | 3001 | 4000 |
| D | 4001 | 5000 |

**SELECT e.last\_name, e.first\_name, e.salary, j.grade\_level**

**FROM employees e JOIN job\_grades j**

**ON e.salary BETWEEN j.lowest\_sal AND j.highest\_sal**

**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

Q3) Determine which orders haven't yet shipped and the name of the customer who placed the order.Sort the result by the date on which the order was placed

Er

Q4) Produce a list of all customers who live in state Florida (FL) and have ordered books about computers (category : COMPUTERS)