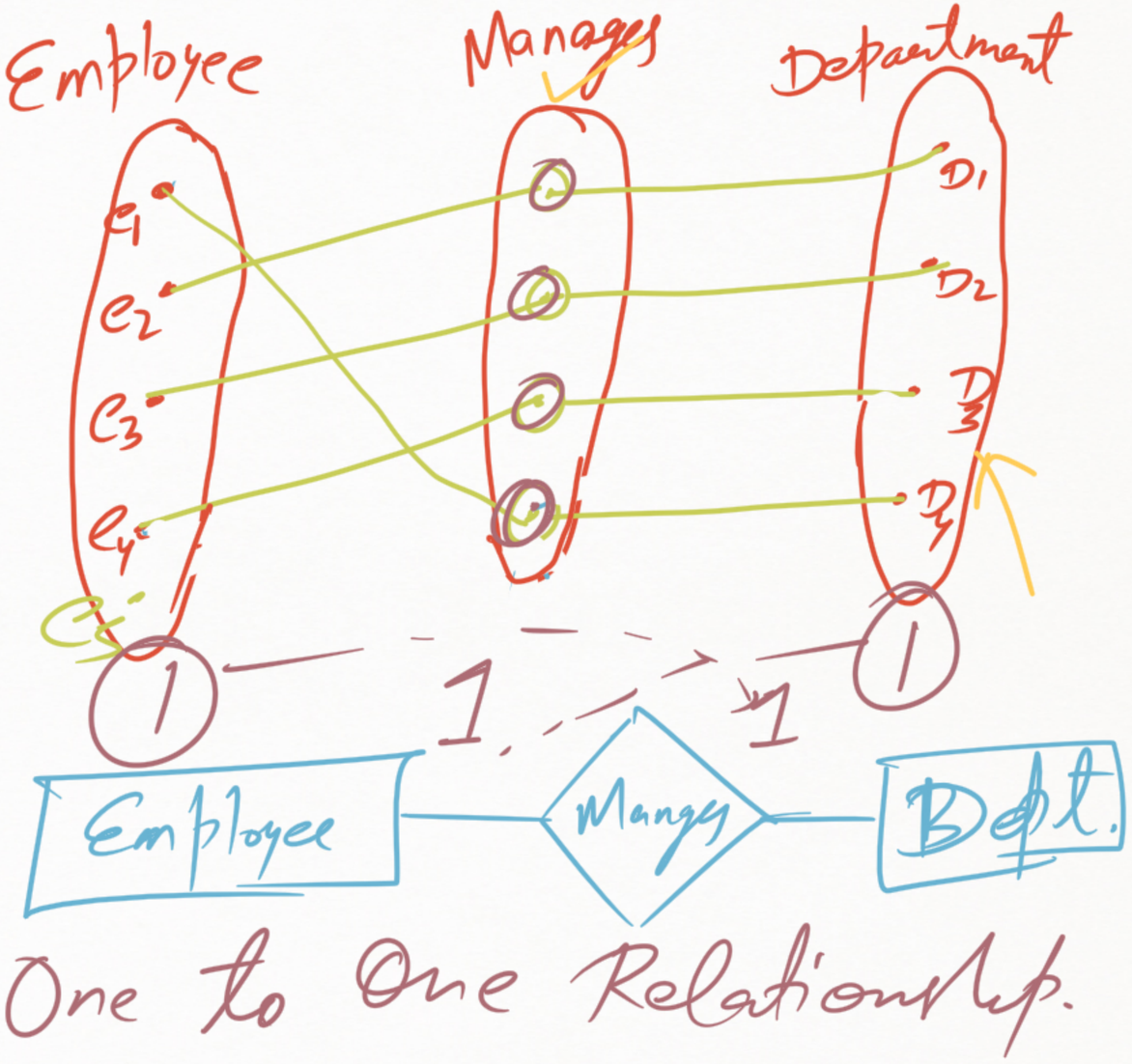
# Cust Req.

1. Every Department should have a manager.
2. Only 1 employee manages a department.
3. An employee can manage only 1 Department.

Cardinality = Max, Rel can entity participate

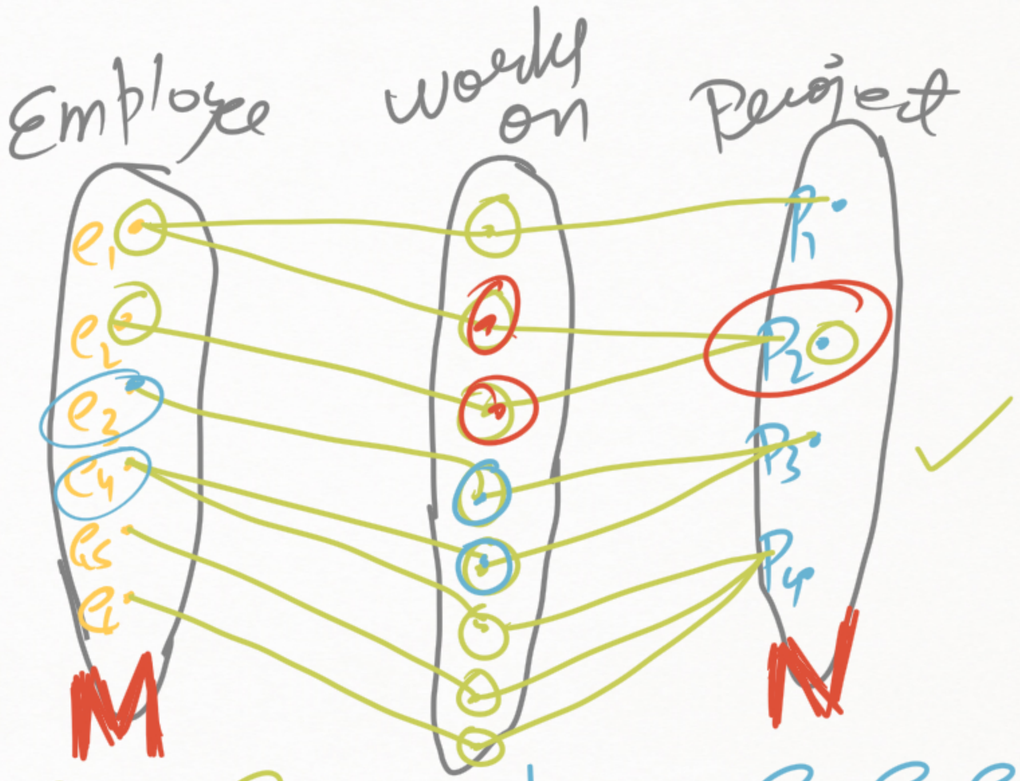



One to One Relationship.

# Req.

1. Every emp. is supposed to work atleast on one project.
2. emp. can work on many projects.
3. Every proj should have atleast 1 employee, it can have many

# Cardinality




$P_1 = e_1$

$P_2 = e_1, e_2$

$P_3 = e_3, e_4$

$P_4 = e_4, e_5, e_6$


Many to Many

# Partial Participation Vs Total Participation

Entity set may not have a relation


* Cust Req =) New Dept. may not have an employee.
* Every emp should work a Dept.


Total Participation

Partial Participation

Q1. Total Vs Partial?


* Every branch should have a manager.
* Employee may or may not be a manager

# Participation is the property of Relationship and not entities

# JOINS

Left

| ID | NAME |
|---|---|
| 1 | A |
| 2 | B |
| 3 | C |
| 4 | D |

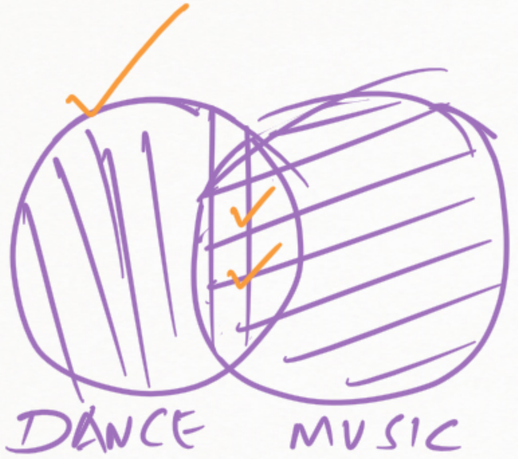
DANCE

INNER JOIN


Right

| SID | FULLNAME |
|---|---|
| 3 | C |
| 4 | D |
| 5 | E |
| 6 | F |

MUSIC




LEFT JOIN

JOIN CONDITION
ID = SID


RIGHT JOIN

# INNER JOIN

| ID | NAME | SID | FULLNAME |
|---|---|---|---|
| 3 | C | 3 | C |
| 4 | D | 4 | D |

# LEFT JOIN

| ID | NAME | SID | FULLNAME |
|---|---|---|---|
| 1 | A | NULL | NULL |
| 2 | B | NULL | NULL |
| 3 | C | 3 | C |
| 4 | D | 4 | D |

# Right JOIN

| ID | NAME | SID | FullNAme |
|---|---|---|---|
| 3 | C | 3 | C |
| 4 | D | 4 | D |
| NULL | NULL | 5 | E |
| NULL | NULL | 6 | F |

INNER ✓
LEFT ✓
RIGHT ✓

JOIN CONDITION & Left & Right Table are decided by US whenever we write query.

# ALIAS => Rename column OR Table NAME on the fly.

=> AS

```
Select first_name as name
from employee.
```

```
mysql> SELECT * FROM DANCE;
+------+--------+
| ID   | NAME   |
+------+--------+
|    1 | ARNOLD |
|    2 | BOB    |
|    3 | CAT    |
|    4 | DOG    |
+------+--------+
```

Left

| ID | NAME |
|---|---|
| 3 | CAT |
| 4 | Doh |

```
mysql> SELECT * FROM MUSIC;
+------+---------+
| ID   | NAME    |
+------+---------+
|    3 | CAT     |
|    4 | DOG     |
|    5 | LION    |
|    6 | SUVEDHA |
+------+---------+
```

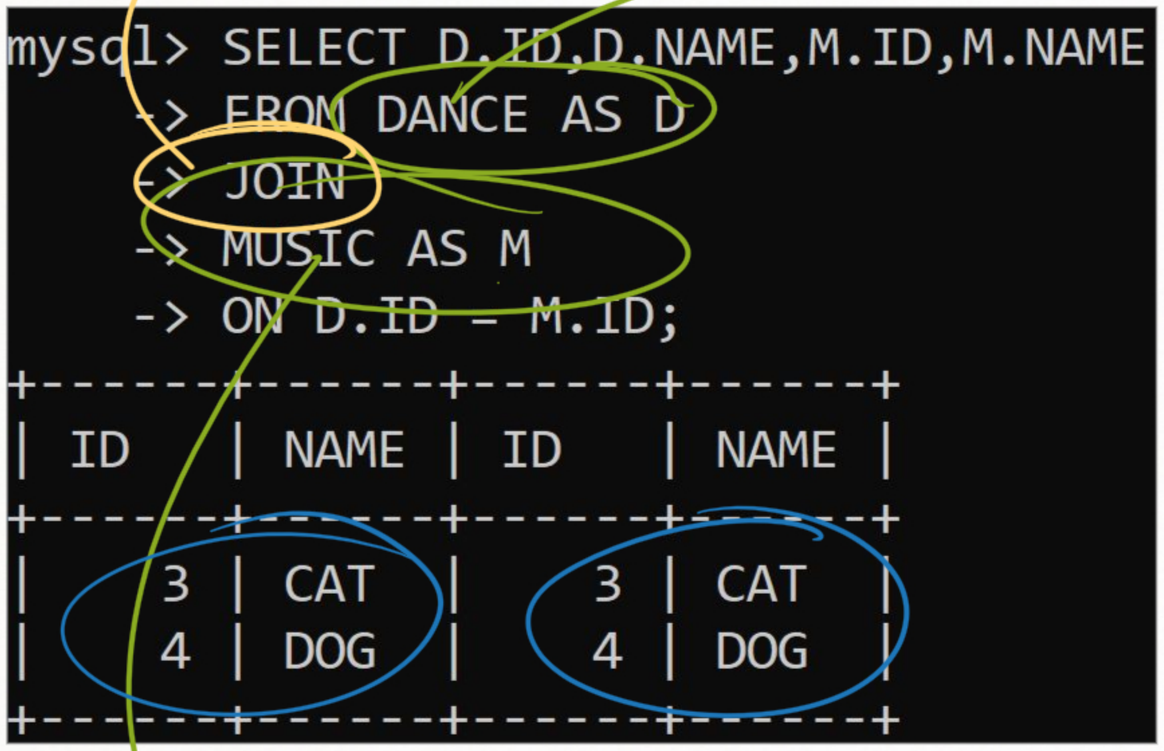
Right

Left Join

| ID | NAME | ID | NAME |
|---|---|---|---|
| 1 | ARNOLD | NULL | NULL |
| 2 | BOB | NULL | NULL |
| 3 | CAT | 3 | CAT |
| 4 | DOG | 4 | DOG |

Right Join.

| ID | NAME | ID | NAME |
|---|---|---|---|
| 3 | CAT | 3 | CAT |
| 4 | DOG | 4 | DOG |
| 5 | LION | NULL | NULL |
| 6 | SUVEDHA | NULL | NULL |




INNER JOIN

DANCE is a LEFT table because it arrived first when writing the Query. ✓

ON ? specifies the condition on which JOIN should be applied

MUSIC is RIGHT table ✓

#

LEFT JOIN

```
mysql> SELECT D.ID,D.NAME,M.ID,M.NAME
    -> FROM DANCE AS D
    -> LEFT JOIN
    -> MUSIC AS M
    -> ON D.ID = M.ID;
+------+--------+------+------+
| ID   | NAME   | ID   | NAME |
+------+--------+------+------+
|    1 | ARNOLD | NULL | NULL |
|    2 | BOB    | NULL | NULL |
|    3 | CAT    |    3 | CAT  |
|    4 | DOG    |    4 | DOG  |
+------+--------+------+------+
```

LEFT

RIGHT

Whatever is present in left Table will be there. along with common values of Right table

```
mysql> SELECT * FROM DANCE;
+------+--------+
| ID   | NAME   |
+------+--------+
|    1 | ARNOLD |
|    2 | BOB    |
|    3 | CAT    |
|    4 | DOG    |
+------+--------+
```

```
mysql> SELECT * FROM MUSIC;
+------+---------+
| ID   | NAME    |
+------+---------+
|    3 | CAT     |
|    4 | DOG     |
|    5 | LION    |
|    6 | SUVEDHA |
+------+---------+
```

Red will be NULL.

# RIGHT JOIN

```
mysql> SELECT D.ID,D.NAME,M.ID,M.NAME
    -> FROM DANCE AS D
    -> RIGHT JOIN
    -> MUSIC AS M
    -> ON D.ID = M.ID;
+------+------+------+---------+
| ID   | NAME | ID   | NAME    |
+------+------+------+---------+
|    3 | CAT  |    3 | CAT     |
|    4 | DOG  |    4 | DOG     |
| NULL | NULL |    5 | LION    |
| NULL | NULL |    6 | SUVEDHA |
+------+------+------+---------+
```

LEFT

RIGHT

Because it was written first ✓

Everything will be there from the RIGHT common along with from LEFT

```
mysql> SELECT * FROM DANCE;
+------+--------+
| ID   | NAME   |
+------+--------+
|    1 | ARNOLD |
|    2 | BOB    |
|    3 | CAT    |
|    4 | DOG    |
+------+--------+
```

LEFT

```
mysql> SELECT * FROM MUSIC;
+------+---------+
| ID   | NAME    |
+------+---------+
|    3 | CAT     |
|    4 | DOG     |
|    5 | LION    |
|    6 | SUVEDHA |
+------+---------+
```

RIGHT

Rest will be NULL.

# Practical App. of JOINS

Employee

| emp_id | first_name | last_name | birth_date | sex | salary | super_id | branch_id |
|---|---|---|---|---|---|---|---|
| 100 | David | Wallace | 1967-11-17 | M | 250,000 | NULL | 1 |
| 101 | Jan | Levinson | 1961-05-11 | F | 110,000 | 100 | 1 |
| 102 | Michael | Scott | 1964-03-15 | M | 75,000 | 100 | 2 |
| 103 | Angela | Martin | 1971-06-25 | F | 63,000 | 102 | 2 |
| 104 | Kelly | Kapoor | 1980-02-05 | F | 55,000 | 102 | 2 |
| 105 | Stanley | Hudson | 1958-02-19 | M | 69,000 | 102 | 2 |
| 106 | Josh | Porter | 1969-09-05 | M | 78,000 | 100 | 3 |
| 107 | Andy | Bernard | 1973-07-22 | M | 65,000 | 106 | 3 |
| 108 | Jim | Halpert | 1978-10-01 | M | 71,000 | 106 | 3 |

Branch

| branch_id | branch_name | mgr_id | mgr_start_date |
|---|---|---|---|
| 1 | Corporate | 100 | 2006-02-09 |
| 2 | Scranton | 102 | 1992-04-06 |
| 3 | Stamford | 106 | 1998-02-13 |

Client

| client_id | client_name | branch_id |
|---|---|---|
| 400 | Dunmore Highschool | 2 |
| 401 | Lackawana Country | 2 |
| 402 | FedEx | 3 |
| 403 | John Daly Law, LLC | 3 |
| 404 | Scranton Whitepages | 2 |
| 405 | Times Newspaper | 3 |
| 406 | FedEx | 2 |

Works_With

| emp_id | client_id | total_sales |
|---|---|---|
| 105 | 400 | 55,000 |
| 102 | 401 | 267,000 |
| 108 | 402 | 22,500 |
| 107 | 403 | 5,000 |
| 108 | 403 | 12,000 |
| 105 | 404 | 33,000 |
| 107 | 405 | 26,000 |
| 102 | 406 | 15,000 |
| 105 | 406 | 130,000 |

Branch Supplier

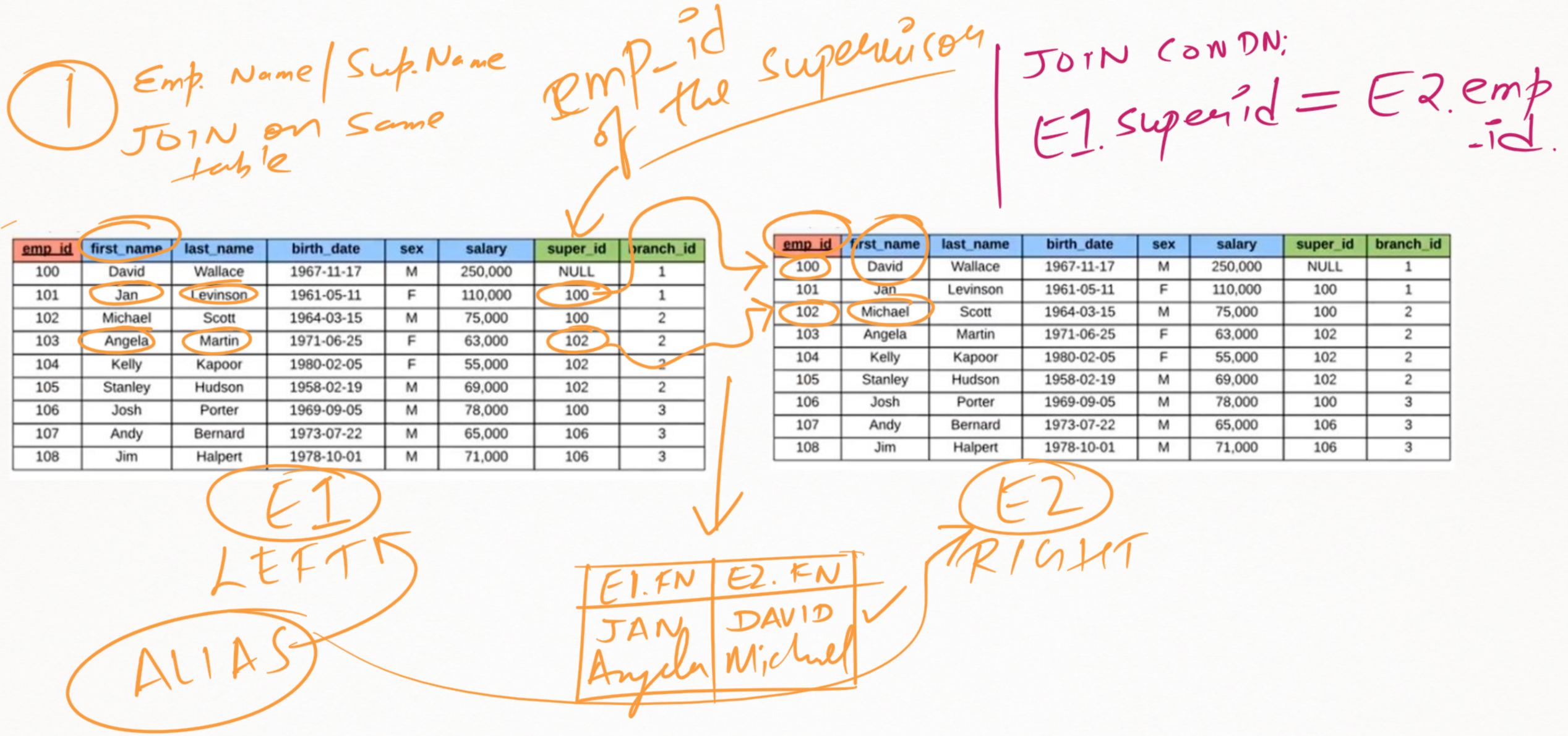
| branch_id | supplier_name | supply_type |
|---|---|---|
| 2 | Hammer Mill | Paper |
| 2 | Uni-ball | Writing Utensils |
| 3 | Patriot Paper | Paper |
| 2 | J.T. Forms & Labels | Custom Forms |
| 3 | Uni-ball | Writing Utensils |
| 3 | Hammer Mill | Paper |
| 3 | Stamford Lables | Custom Forms |

Q1. Print Employee Name and their Supervisor Name

[JOINS can be applied on the same table
Remember ALIAS]

Q2. Print Branch Name and their Manager Name

Q3. Print client_name, emp_id and total sales realated.




| emp_id | first_name | last_name | birth_date | sex | salary | super_id | branch_id |
|---|---|---|---|---|---|---|---|
| 100 | David | Wallace | 1967-11-17 | M | 250,000 | NULL | 1 |
| 101 | Jan | Levinson | 1961-05-11 | F | 110,000 | 100 | 1 |
| 102 | Michael | Scott | 1964-03-15 | M | 75,000 | 100 | 2 |
| 103 | Angela | Martin | 1971-06-25 | F | 63,000 | 102 | 2 |
| 104 | Kelly | Kapoor | 1980-02-05 | F | 55,000 | 102 | 2 |
| 105 | Stanley | Hudson | 1958-02-19 | M | 69,000 | 102 | 2 |
| 106 | Josh | Porter | 1969-09-05 | M | 78,000 | 100 | 3 |
| 107 | Andy | Bernard | 1973-07-22 | M | 65,000 | 106 | 3 |
| 108 | Jim | Halpert | 1978-10-01 | M | 71,000 | 106 | 3 |

| emp_id | first_name | last_name | birth_date | sex | salary | super_id | branch_id |
|---|---|---|---|---|---|---|---|
| 100 | David | Wallace | 1967-11-17 | M | 250,000 | NULL | 1 |
| 101 | Jan | Levinson | 1961-05-11 | F | 110,000 | 100 | 1 |
| 102 | Michael | Scott | 1964-03-15 | M | 75,000 | 100 | 2 |
| 103 | Angela | Martin | 1971-06-25 | F | 63,000 | 102 | 2 |
| 104 | Kelly | Kapoor | 1980-02-05 | F | 55,000 | 102 | 2 |
| 105 | Stanley | Hudson | 1958-02-19 | M | 69,000 | 102 | 2 |
| 106 | Josh | Porter | 1969-09-05 | M | 78,000 | 100 | 3 |
| 107 | Andy | Bernard | 1973-07-22 | M | 65,000 | 106 | 3 |
| 108 | Jim | Halpert | 1978-10-01 | M | 71,000 | 106 | 3 |

```
mysql> SELECT E1.FIRST_NAME,E2.FIRST_NAME
    -> FROM
    -> EMPLOYEE AS E1
    -> JOIN
    -> EMPLOYEE AS E2
    -> ON E1.SUPER_ID=E2.EMP_ID;
+------------+------------+
| FIRST_NAME | FIRST_NAME |
+------------+------------+
| Jan        | David      |
| Michael    | David      |
| Angela     | Michael    |
| Kelly      | Michael    |
| Stanley    | Michael    |
| Josh       | David      |
| Andy       | Josh       |
| Jim        | Josh       |
+------------+------------+
```

# Q2. Print Branch Name and their Manager Name.

emp_id on employee table

B1.mgr_id = E1.emp_id

Common Data

INNER JOIN

Branch

| branch_id | branch_name | mgr_id | mgr_start_date |
|---|---|---|---|
| 1 | Corporate | 100 | 2006-02-09 |
| 2 | Scranton | 102 | 1992-04-06 |
| 3 | Stamford | 106 | 1998-02-13 |

B1

| emp_id | first_name | last_name | birth_date | sex | salary | super_id | branch_id |
|---|---|---|---|---|---|---|---|
| 100 | David | Wallace | 1967-11-17 | M | 250,000 | NULL | 1 |
| 101 | Jan | Levinson | 1961-05-11 | F | 110,000 | 100 | 1 |
| 102 | Michael | Scott | 1964-03-15 | M | 75,000 | 100 | 2 |
| 103 | Angela | Martin | 1971-06-25 | F | 63,000 | 102 | 2 |
| 104 | Kelly | Kapoor | 1980-02-05 | F | 55,000 | 102 | 2 |
| 105 | Stanley | Hudson | 1958-02-19 | M | 69,000 | 102 | 2 |
| 106 | Josh | Porter | 1969-09-05 | M | 78,000 | 100 | 3 |
| 107 | Andy | Bernard | 1973-07-22 | M | 65,000 | 106 | 3 |
| 108 | Jim | Halpert | 1978-10-01 | M | 71,000 | 106 | 3 |

employee E1

```
mysql> SELECT B1.branch_name,E1.first_name
    -> FROM
    -> EMPLOYEE AS E1
    -> JOIN
    -> BRANCH AS B1
    -> ON B1.mgr_id = E1.emp_id;
+-------------+------------+
| branch_name | first_name |
+-------------+------------+
| Corporate   | David      |
| Scranton    | Michael    |
| Stamford    | Josh       |
+-------------+------------+
```

```
SELECT B1.branch_name
FROM
Branch as B1
JOIN
Employee as E1
ON B1.mgr_id = E1.emp_id.
```