

```
mysql> select w.emp_id, c.client_name,w.total_sales
->      from works_with as w
->      join client as c
->      ON c.client_id = w.client_id;
+-----+-----+-----+
| emp_id | client_name          | total_sales |
+-----+-----+-----+
|    105 | Dunmore Highschool   |     55000 |
|    102 | Lackawana Country    |    267000 |
|    108 | FedEx                  |     22500 |
|    107 | John Daly Law         |      5000 |
|    108 | John Daly Law         |     12000 |
|    105 | Scranton Whitepages  |     33000 |
|    107 | Times Newspaper        |     26000 |
|    102 | FedEx                  |     15000 |
|    105 | FedEx                  |   130000 |
+-----+-----+-----+
9 rows in set (0.00 sec)
```

Nested Queries

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

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

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

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. Name of all the employees who is earning more than Angela Martin.

Q2. All the people Name who are in same branch as Michael Scott

Q3. Print fname of emp who earns more than everybody in branch=2.

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

Once executed
it will replaced
by that value
63000

The outer query will execute.

```

mysql> SELECT first_name,salary FROM employee
-> WHERE
-> salary > (SELECT salary FROM employee WHERE first_name='Angela' AND last_name='Martin');
+-----+-----+
| first_name | salary |
+-----+-----+
| David      | 250000 |
| Jan        | 110000 |
| Michael    | 75000  |
| Stanley    | 69000  |
| Josh       | 78000  |
| Andy       | 65000  |
| Jim        | 71000  |
+-----+-----+
7 rows in set (0.00 sec)

```

Nested Query This will execute first

```
mysql> SELECT FIRST_NAME,branch_id FROM EMPLOYEE where branch_id=
-> (select branch_id from employee where first_name='Michael' and last_name='Scott');
+-----+-----+
| FIRST_NAME | branch_id |
+-----+-----+
| Michael    |      2 |
| Angela     |      2 |
| Kelly       |      2 |
| Stanley    |      2 |
+-----+-----+
4 rows in set (0.00 sec)
```

```
mysql> SELECT first_name FROM employee
-> WHERE
-> salary > (SELECT MAX(salary) FROM EMPLOYEE where branch_id=2);
+-----+
| first_name |
+-----+
| David      |
| Jan        |
| Josh       |
+-----+
3 rows in set (0.00 sec)
```

ALL, ANY

Salary >

ALL([1, 2, 3, 4])

5, 6, 7, 8 -

Salary >

ANY([0, 1, 2, 3, 4])

2, 3, 4, 5, 1 ✓ - -

ALL

```
mysql> SELECT first_name FROM employee
-> WHERE
-> salary > ALL(SELECT salary FROM EMPLOYEE where branch_id=2);
+-----+
| first_name |
+-----+
| David      |
| Jan        |
| Josh       |
+-----+
3 rows in set (0.00 sec)
```

ANY

```
mysql> SELECT first_name FROM employee
-> WHERE
-> salary > ANY(SELECT salary FROM EMPLOYEE where branch_id=2);
+-----+
| first_name |
+-----+
| David      |
| Jan        |
| Michael    |
| Angela     |
| Stanley    |
| Josh       |
| Andy       |
| Jim        |
+-----+
8 rows in set (0.00 sec)
```

DISTINCT

```
mysql> SELECT DISTINCT sex FROM employee;
+-----+
| sex  |
+-----+
| M   |
| F   |
+-----+
2 rows in set (0.00 sec)
```

```
mysql> SELECT DISTINCT branch_id FROM employee;
+-----+
| branch_id |
+-----+
|      1    |
|      2    |
|      3    |
+-----+
3 rows in set (0.00 sec)
```

Correlated Nested Queries

* Inner query uses data from outer query.

* A correlated Nested Query is evaluated once for each row in outer query.

How
Correlated
Nested
Queries
Work?

GET Candidate now
from outer query.

EXECUTE => Inner query using
candidate now value

USE => Values from inner
query to qualify or
disqualify a candidate now.

SYNTAX

```
Select col1, col2  
FROM table1 outer  
WHERE col1 operator ] Outer  
  
( SELECT col1, col2  
FROM table2  
WHERE  
expr1 = outer.expr2) ] Nester  
/ Inner
```

Q Find out the 2nd Highest Salary?

SELECT salary FROM employee e

WHERE 2 = (SELECT COUNT(DISTINCT salary)

FROM employee P

WHERE

e.salary <= P.salary);

```
mysql> SELECT salary FROM employee e
    -> WHERE 2 = (SELECT COUNT(DISTINCT salary)
    -> FROM employee p
    -> WHERE
    -> e.salary <= p.salary);

+-----+
| salary |
+-----+
| 110000 |
+-----+
1 row in set (0.00 sec)
```