Installing MYSQL in ubuntu:

firstly I remove all the packages:

1. sudo apt-get remove --purge mysql*
2. sudo apt-get purge mysql*
3. sudo apt-get autoremove.
4. sudo apt-get autoclean.
5. sudo apt-get remove dbconfig-mysql.
6. sudo apt-get dist-upgrade.
7. sudo apt-get install mysql-server.

Install mysql:

1. sudo apt-get install mysql-server
2. sudo apt-get install mysql-client
3. sudo mysql_secure_installation
4. mysql --version

CREATING DATABASE:

CREATING DATABASE:

CREATE DATABASE bank;

VERIFY DATABASE IS CREATED OR NOT:

SHOW DATABASES;

DESCRIBE WHICH DATABASE YOU USED:

USE bank;

CREATING TABLES:

- ***Employee Table***

CREATING TABLE "Employee":

CREATE TABLE Employee(
emp_id INT,
first_name varchar(20),
last_name VARCHAR(20),
birth_date DATE,
sex varchar(1),

```
salary INT,
super_id INT,
branch_id INT,
PRIMARY KEY(emp_id));
```

VERIFY TABLE CREATED OR NOT:

```
SHOW TABLES;
```

SEE PROPERTIES OF "Employee" table:

```
DESCRIBE Employee;
```

- ***branch Table***

CREATE TABLE "branch" :

```
CREATE TABLE branch(
branch_id INT,
branch_name VARCHAR(20),
mgr_id INT,
mgr_start_date DATE,
PRIMARY KEY(branch_id),
FOREIGN KEY (mgr_id) REFERENCES Employee (emp_id) ON DELETE SET NULL
);
```

SEE PROPERTIES OF "branch" TABLE:

```
DESCRIBE branch;
```

OUTPUT:

```
+----------------+-------------+------+-----+---------+-------+
| Field          | Type        | Null | Key | Default | Extra |
+----------------+-------------+------+-----+---------+-------+
| branch_id      | int         | NO   | PRI | NULL    |       |
| branch_name    | varchar(20) | YES  |     | NULL    |       |
| mgr_id         | int         | YES  | MUL | NULL    |       |
| mgr_start_date | date        | YES  |     | NULL    |       |
+----------------+-------------+------+-----+---------+-------+
4 rows in set (0.01 sec)
```

MAKE CHANGES OF "Employee" table (making "super_id" and "branch_id" Foreign key):

ALTER TABLE Employee
ADD FOREIGN KEY (super_id) **REFERENCES** Employee (emp_id)
ON DELETE SET NULL;

ALTER TABLE Employee
ADD FOREIGN KEY (branch_id) **REFERENCES** branch(branch_id)
ON DELETE SET NULL;

SEE PROPERTIES OF "Employee" TABLE:

DESCRIBE Employee;

OUTPUT:

```
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| emp_id     | int         | NO   | PRI | NULL    |       |
| first_name | varchar(20) | YES  |     | NULL    |       |
| last_name  | varchar(20) | YES  |     | NULL    |       |
| birth_date | date        | YES  |     | NULL    |       |
| sex        | varchar(1)  | YES  |     | NULL    |       |
| salary     | int         | YES  |     | NULL    |       |
| super_id   | int         | YES  | MUL | NULL    |       |
| branch_id  | int         | YES  | MUL | NULL    |       |
+------------+-------------+------+-----+---------+-------+
8 rows in set (0.00 sec)
```

- ***client Table***

MAKE TABLE "client" :

CREATE TABLE client(
-> client_id INT,
-> client_name VARCHAR(20),
-> branch_id INT,
-> PRIMARY KEY(client_id),
-> FOREIGN KEY (branch_id) **REFERENCES** branch(branch_id)
ON DELETE SET NULL);

VERIFY TABLE ARE CREATED OR NOT:

SHOW TABLES;

OUTPUT:

```
+----------------+
| Tables_in_bank |
+----------------+
| Employee       |
| branch         |
| client         |
+----------------+
```

SEE PROPERTIES OF "client" TABLE:

DESCRIBE client;

OUTPUT:

```
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| client_id   | int         | NO   | PRI | NULL    |       |
| client_name | varchar(20) | YES  |     | NULL    |       |
| branch_id   | int         | YES  | MUL | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
3 rows in set (0.01 sec)
```

- ***works_with Table***

CREATE TABLE "works_with" :

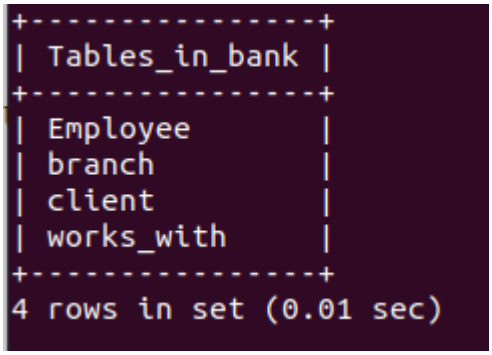
CREATE TABLE works_with (
emp_id INT,

client_id INT,
total_sales INT,
PRIMARY KEY(emp_id,client_id),
FOREIGN KEY (emp_id) **REFERENCES** Employee(emp_id) **ON DELETE** CASCADE,
FOREIGN KEY (client_id) **REFERENCES** client(client_id) **ON DELETE** CASCADE
);

VERIFY TABLE ARE CREATED OR NOT:

SHOW TABLES;

OUTPUT:

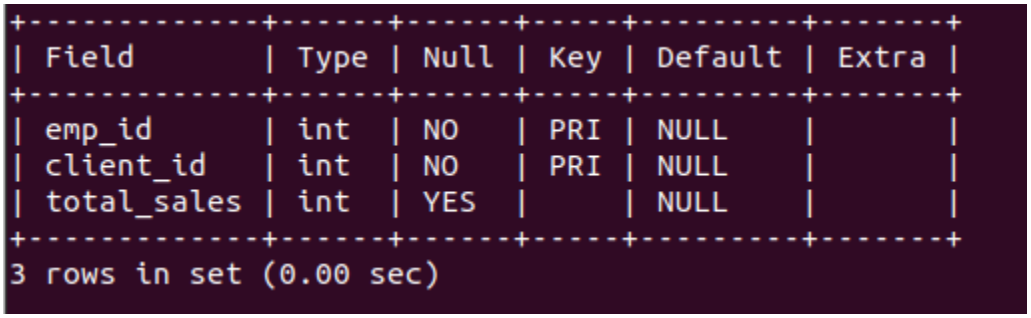


```
+----------------+
| Tables_in_bank |
+----------------+
| Employee       |
| branch         |
| client         |
| works_with     |
+----------------+
4 rows in set (0.01 sec)
```

SEE PROPERTIES OF "works_with" TABLE:

DESCRIBE works_with;

OUTPUT:



```
+-------------+------+------+-----+---------+-------+
| Field       | Type | Null | Key | Default | Extra |
+-------------+------+------+-----+---------+-------+
| emp_id      | int  | NO   | PRI | NULL    |       |
| client_id   | int  | NO   | PRI | NULL    |       |
| total_sales | int  | YES  |     | NULL    |       |
+-------------+------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

- ***branch_supplier Table***

CREATE TABLE "branch_supplier" :

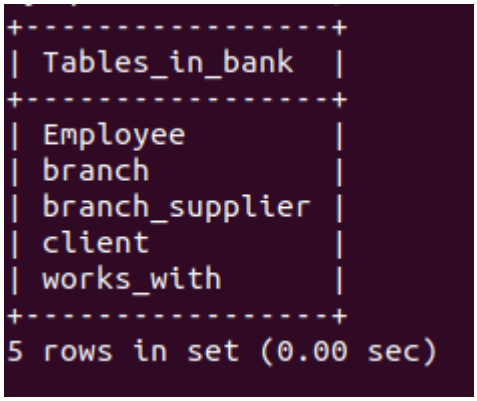
CREATE TABLE branch_supplier(
branch_id INT,

```
supplier_name VARCHAR(20),
supply_type VARCHAR(20),
PRIMARY KEY(branch_id,supplier_name),
FOREIGN KEY (branch_id) REFERENCES branch(branch_id)
);
```

VERIFY TABLE ARE CREATED OR NOT:

```
SHOW TABLES;
```

OUTPUT:

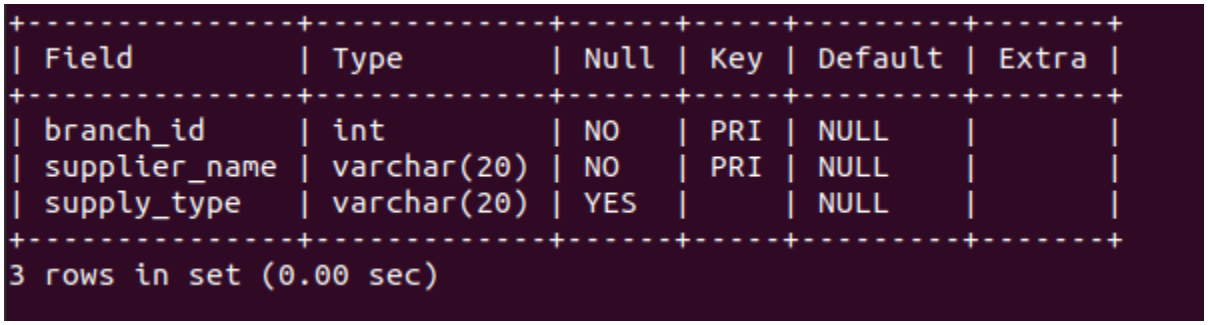


```
+-----------------+
| Tables_in_bank  |
+-----------------+
| Employee        |
| branch          |
| branch_supplier |
| client          |
| works_with      |
+-----------------+
5 rows in set (0.00 sec)
```

SEE PROPERTIES OF "works_with" TABLE:

```
DESCRIBE branch_supplier;
```

OUTPUT:



```
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| branch_id     | int         | NO   | PRI | NULL    |       |
| supplier_name | varchar(20) | NO   | PRI | NULL    |       |
| supply_type   | varchar(20) | YES  |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

INSERTING VALUES IN TABLES:

- ***Employee Table***

INSERT INTO Employee **VALUES** (100,'david','wallace','1967-11-17','M',250000,NULL,NULL);

NOTE:

I try to put foreign key values but that key have no values in his table. So it create error so I used NULL instead of its actual values.

SELECT *
FROM Employee;

```
+--------+------------+-----------+------------+------+--------+----------+-----------+
| emp_id | first_name | last_name | birth_date | sex  | salary | super_id | branch_id |
+--------+------------+-----------+------------+------+--------+----------+-----------+
|    100 | david      | wallace   | 1967-11-17 | M    | 250000 |     NULL |      NULL |
+--------+------------+-----------+------------+------+--------+----------+-----------+
1 row in set (0.00 sec)
```

I**NSERT INTO** Employee **VALUES**
-> (101,'jan','jevinson','1961-05-11','F',110000,NULL,NULL),
-> (102,'micheal','scott','1961-06-25','M',75000,NULL,NULL),
-> (103,'angela','martin','1971-06-25','F',63000,NULL,NULL),
-> (104,'kelly','kapoor','1980-02-05','F',55000,NULL,NULL),
-> (105,'stanley','hudsen','1956-2-19','M',69000,NULL,NULL),
-> (106,'josh','porter','1969-08-05','M',78000,NULL,NULL),
-> (107,'andy','bernard','1973-10-01','M',65000,NULL,NULL),
-> (108,'jim','hairpert','1978-10-01','M',71000,NULL,NULL);

SELECT *
FROM Employee;

emp_id	first_name	last_name	birth_date	sex	salary	super_id	branch_id
100	david	wallace	1967-11-17	M	250000	NULL	NULL
101	jan	jevinson	1961-05-11	F	110000	NULL	NULL
102	micheal	scott	1961-06-25	M	75000	NULL	NULL
103	angela	martin	1971-06-25	F	63000	NULL	NULL
104	kelly	kapoor	1980-02-05	F	55000	NULL	NULL
105	stanley	hudsen	1956-02-19	M	69000	NULL	NULL
106	josh	porter	1969-08-05	M	78000	NULL	NULL
107	andy	bernard	1973-10-01	M	65000	NULL	NULL
108	jim	hairpert	1978-10-01	M	71000	NULL	NULL

9 rows in set (0.00 sec)

- ***branch Table***

INSERT INTO branch **VALUES**
(1,'corporate',100,'2006-02-09'),
(2,'scranton',102,'1992-04-06'),
(3,'stanford',106,'1998-02-13');

SELECT *
FROM branch;

branch_id	branch_name	mgr_id	mgr_start_date
1	corporate	100	2006-02-09
2	scranton	102	1992-04-06
3	stanford	106	1998-02-13

- ***MODIFY Employee Table***

emp_id	first_name	last_name	birth_date	sex	salary	super_id	branch_id
100	david	wallace	1967-11-17	M	250000	NULL	NULL
101	jan	jevinson	1961-05-11	F	110000	NULL	NULL
102	micheal	scott	1961-06-25	M	75000	NULL	NULL
103	angela	martin	1971-06-25	F	63000	NULL	NULL
104	kelly	kapoor	1980-02-05	F	55000	NULL	NULL
105	stanley	hudsen	1956-02-19	M	69000	NULL	NULL
106	josh	porter	1969-08-05	M	78000	NULL	NULL
107	andy	bernard	1973-10-01	M	65000	NULL	NULL
108	jim	hairpert	1978-10-01	M	71000	NULL	NULL

9 rows in set (0.00 sec)

UPDATE Employee
SET branch_id = 1
WHERE emp_id **BETWEEN** 100 **AND** 101;

NOTE:
It can update value (100 to 101)

UPDATE Employee
SET branch_id = 2
WHERE emp_id= 102;

SELECT *
FROM Employee;

emp_id	first_name	last_name	birth_date	sex	salary	super_id	branch_id
100	david	wallace	1967-11-17	M	250000	NULL	1
101	jan	jevinson	1961-05-11	F	110000	NULL	1
102	micheal	scott	1961-06-25	M	75000	NULL	2
103	angela	martin	1971-06-25	F	63000	NULL	NULL
104	kelly	kapoor	1980-02-05	F	55000	NULL	NULL
105	stanley	hudsen	1956-02-19	M	69000	NULL	NULL
106	josh	porter	1969-08-05	M	78000	NULL	NULL
107	andy	bernard	1973-10-01	M	65000	NULL	NULL
108	jim	hairpert	1978-10-01	M	71000	NULL	NULL

9 rows in set (0.00 sec)

UPDATE Employee
SET super_id = 102, branch_id = 2
WHERE emp_id BETWEEN 103 AND 105;

SELECT *
FROM Employee;

emp_id	first_name	last_name	birth_date	sex	salary	super_id	branch_id
100	david	wallace	1967-11-17	M	250000	NULL	1
101	jan	jevinson	1961-05-11	F	110000	NULL	1
102	micheal	scott	1961-06-25	M	75000	NULL	2
103	angela	martin	1971-06-25	F	63000	102	2
104	kelly	kapoor	1980-02-05	F	55000	102	2
105	stanley	hudsen	1956-02-19	M	69000	102	2
106	josh	porter	1969-08-05	M	78000	NULL	NULL
107	andy	bernard	1973-10-01	M	65000	NULL	NULL
108	jim	hairpert	1978-10-01	M	71000	NULL	NULL

9 rows in set (0.00 sec)

UPDATE Employee
SET super_id = 100, branch_id = 3
WHERE emp_id=106;

SELECT *
FROM Employee;

emp_id	first_name	last_name	birth_date	sex	salary	super_id	branch_id
100	david	wallace	1967-11-17	M	250000	NULL	1
101	jan	jevinson	1961-05-11	F	110000	NULL	1
102	micheal	scott	1961-06-25	M	75000	NULL	2
103	angela	martin	1971-06-25	F	63000	102	2
104	kelly	kapoor	1980-02-05	F	55000	102	2
105	stanley	hudsen	1956-02-19	M	69000	102	2
106	josh	porter	1969-08-05	M	78000	100	3
107	andy	bernard	1973-10-01	M	65000	NULL	NULL
108	jim	hairpert	1978-10-01	M	71000	NULL	NULL

9 rows in set (0.00 sec)

UPDATE Employee
SET super_id=106 , branch_id=3
WHERE emp_id **IN** (107,108);

SELECT *
FROM Employee;

emp_id	first_name	last_name	birth_date	sex	salary	super_id	branch_id
100	david	wallace	1967-11-17	M	250000	NULL	1
101	jan	jevinson	1961-05-11	F	110000	NULL	1
102	micheal	scott	1961-06-25	M	75000	NULL	2
103	angela	martin	1971-06-25	F	63000	102	2
104	kelly	kapoor	1980-02-05	F	55000	102	2
105	stanley	hudsen	1956-02-19	M	69000	102	2
106	josh	porter	1969-08-05	M	78000	100	3
107	andy	bernard	1973-10-01	M	65000	106	3
108	jim	hairpert	1978-10-01	M	71000	106	3

9 rows in set (0.00 sec)

UPDATE Employee
SET super_id = 100
WHERE emp_id **IN** (101,102);

SELECT *
FROM Employee;

emp_id	first_name	last_name	birth_date	sex	salary	super_id	branch_id
100	david	wallace	1967-11-17	M	250000	NULL	1
101	jan	jevinson	1961-05-11	F	110000	100	1
102	micheal	scott	1961-06-25	M	75000	100	2
103	angela	martin	1971-06-25	F	63000	102	2
104	kelly	kapoor	1980-02-05	F	55000	102	2
105	stanley	hudsen	1956-02-19	M	69000	102	2
106	josh	porter	1969-08-05	M	78000	100	3
107	andy	bernard	1973-10-01	M	65000	106	3
108	jim	hairpert	1978-10-01	M	71000	106	3

9 rows in set (0.00 sec)

- ***client Table***

INSERT INTO client **VALUES**
(400,'dunmore high school',2),
(401,'lackawana country',2),
(402,'fedex',3),
(403,'john daly law,LLC',3),
(404,'scranton whitepages',3),
(405,'times newspaper',3),
(406,'fedex',2);

SELECT *
FROM client;

```
+-----------+---------------------+-----------+
| client_id | client_name         | branch_id |
+-----------+---------------------+-----------+
|       400 | dunmore high school |         2 |
|       401 | lackawana country   |         2 |
|       402 | fedex               |         3 |
|       403 | john daly law,LLC   |         3 |
|       404 | scranton whitepages |         3 |
|       405 | times newspaper     |         3 |
|       406 | fedex               |         2 |
+-----------+---------------------+-----------+
7 rows in set (0.00 sec)
```

- ***works_with Table***

INSERT INTO works_with **VALUES**
(105,400,55000),
(102,401,267000),
(108,402,255000),
(107,403,5000),
(108,403,12000),
(105,404,33000),
(107,405,26000),
(102,406,15000),
(105,406,130000);

SELECT *
FROM works_with;

```
+--------+-----------+-------------+
| emp_id | client_id | total_sales |
+--------+-----------+-------------+
|    102 |       401 |      267000 |
|    102 |       406 |       15000 |
|    105 |       400 |       55000 |
|    105 |       404 |       33000 |
|    105 |       406 |      130000 |
|    107 |       403 |        5000 |
|    107 |       405 |       26000 |
|    108 |       402 |      255000 |
|    108 |       403 |       12000 |
+--------+-----------+-------------+
9 rows in set (0.00 sec)
```

- ***branch supplier Table***

I**NSERT INTO** branch_supplier **VALUES**
(2,'hammer_mills','paper'),
(2,'uni_ball','writing literature'),
(3,'patriot paper','paper'),
(3,'stamord labels','writing utensils'),
(3,'uni_ball','paper'),
(3,'hammer mall','custom forms'),
(2,'jt forms & labels','custom forms');

SELECT *
FROM branch_supplier;

branch_id	supplier_name	supply_type
2	hammer_mills	paper
2	jt forms & labels	custom forms
2	uni_ball	writing literature
3	hammer mall	custom forms
3	patriot paper	paper
3	stamord labels	writing utensils
3	uni_ball	paper

7 rows in set (0.01 sec)

QUERIES:

Find all Employees:

SELECT *
FROM Employee;

emp_id	first_name	last_name	birth_date	sex	salary	super_id	branch_id
100	david	wallace	1967-11-17	M	250000	NULL	1
101	jan	jevinson	1961-05-11	F	110000	100	1
102	micheal	scott	1961-06-25	M	75000	100	2
103	angela	martin	1971-06-25	F	63000	102	2
104	kelly	kapoor	1980-02-05	F	55000	102	2
105	stanley	hudsen	1956-02-19	M	69000	102	2
106	josh	porter	1969-08-05	M	78000	100	3
107	andy	bernard	1973-10-01	M	65000	106	3
108	jim	hairpert	1978-10-01	M	71000	106	3

9 rows in set (0.00 sec)

Find all clients:

SELECT *
FROM client;

```
+-----------+----------------------+-----------+
| client_id | client_name          | branch_id |
+-----------+----------------------+-----------+
|       400 | dunmore high school  |         2 |
|       401 | lackawana country    |         2 |
|       402 | fedex                |         3 |
|       403 | john daly law,LLC    |         3 |
|       404 | scranton whitepages  |         3 |
|       405 | times newspaper      |         3 |
|       406 | fedex                |         2 |
+-----------+----------------------+-----------+
7 rows in set (0.00 sec)
```

Find all employees ordered by salary (ascending order):

SELECT *
FROM Employee
ORDER BY salary **ASC**;

```
+--------+------------+-----------+------------+------+--------+----------+-----------+
| emp_id | first_name | last_name | birth_date | sex  | salary | super_id | branch_id |
+--------+------------+-----------+------------+------+--------+----------+-----------+
|    104 | kelly      | kapoor    | 1980-02-05 | F    |  55000 |      102 |         2 |
|    103 | angela     | martin    | 1971-06-25 | F    |  63000 |      102 |         2 |
|    107 | andy       | bernard   | 1973-10-01 | M    |  65000 |      106 |         3 |
|    105 | stanley    | hudsen    | 1956-02-19 | M    |  69000 |      102 |         2 |
|    108 | jim        | hairpert  | 1978-10-01 | M    |  71000 |      106 |         3 |
|    102 | micheal    | scott     | 1961-06-25 | M    |  75000 |      100 |         2 |
|    106 | josh       | porter    | 1969-08-05 | M    |  78000 |      100 |         3 |
|    101 | jan        | jevinson  | 1961-05-11 | F    | 110000 |      100 |         1 |
|    100 | david      | wallace   | 1967-11-17 | M    | 250000 |     NULL |         1 |
+--------+------------+-----------+------------+------+--------+----------+-----------+
9 rows in set (0.00 sec)
```

Find all employees ordered by salary (descending order):

SELECT *
FROM Employee
ORDER BY salary **DESC**;

emp_id	first_name	last_name	birth_date	sex	salary	super_id	branch_id
100	david	wallace	1967-11-17	M	250000	NULL	1
101	jan	jevinson	1961-05-11	F	110000	100	1
106	josh	porter	1969-08-05	M	78000	100	3
102	micheal	scott	1961-06-25	M	75000	100	2
108	jim	hairpert	1978-10-01	M	71000	106	3
105	stanley	hudsen	1956-02-19	M	69000	102	2
107	andy	bernard	1973-10-01	M	65000	106	3
103	angela	martin	1971-06-25	F	63000	102	2
104	kelly	kapoor	1980-02-05	F	55000	102	2

9 rows in set (0.00 sec)

Find all employees ordered by sex than name:

SELECT *
FROM Employee
ORDER BY sex,first_name ASC;

OR

SELECT *
FROM Employee
ORDER BY sex,first_name;

emp_id	first_name	last_name	birth_date	sex	salary	super_id	branch_id
103	angela	martin	1971-06-25	F	63000	102	2
101	jan	jevinson	1961-05-11	F	110000	100	1
104	kelly	kapoor	1980-02-05	F	55000	102	2
107	andy	bernard	1973-10-01	M	65000	106	3
100	david	wallace	1967-11-17	M	250000	NULL	1
108	jim	hairpert	1978-10-01	M	71000	106	3
106	josh	porter	1969-08-05	M	78000	100	3
102	micheal	scott	1961-06-25	M	75000	100	2
105	stanley	hudsen	1956-02-19	M	69000	102	2

Find all employees ordered by name than sex:

SELECT *
FROM Employee

ORDER BY first_name,sex;

```
+--------+------------+-----------+------------+------+--------+----------+-----------+
| emp_id | first_name | last_name | birth_date | sex  | salary | super_id | branch_id |
+--------+------------+-----------+------------+------+--------+----------+-----------+
|    107 | andy       | bernard   | 1973-10-01 | M    |  65000 |      106 |         3 |
|    103 | angela     | martin    | 1971-06-25 | F    |  63000 |      102 |         2 |
|    100 | david      | wallace   | 1967-11-17 | M    | 250000 |     NULL |         1 |
|    101 | jan        | jevinson  | 1961-05-11 | F    | 110000 |      100 |         1 |
|    108 | jim        | hairpert  | 1978-10-01 | M    |  71000 |      106 |         3 |
|    106 | josh       | porter    | 1969-08-05 | M    |  78000 |      100 |         3 |
|    104 | kelly      | kapoor    | 1980-02-05 | F    |  55000 |      102 |         2 |
|    102 | micheal    | scott     | 1961-06-25 | M    |  75000 |      100 |         2 |
|    105 | stanley    | hudsen    | 1956-02-19 | M    |  69000 |      102 |         2 |
+--------+------------+-----------+------------+------+--------+----------+-----------+
9 rows in set (0.00 sec)
```

Find first five employee in table:

SELECT *
FROM Employee
limit 5;

```
+--------+------------+-----------+------------+------+--------+----------+-----------+
| emp_id | first_name | last_name | birth_date | sex  | salary | super_id | branch_id |
+--------+------------+-----------+------------+------+--------+----------+-----------+
|    100 | david      | wallace   | 1967-11-17 | M    | 250000 |     NULL |         1 |
|    101 | jan        | jevinson  | 1961-05-11 | F    | 110000 |      100 |         1 |
|    102 | micheal    | scott     | 1961-06-25 | M    |  75000 |      100 |         2 |
|    103 | angela     | martin    | 1971-06-25 | F    |  63000 |      102 |         2 |
|    104 | kelly      | kapoor    | 1980-02-05 | F    |  55000 |      102 |         2 |
+--------+------------+-----------+------------+------+--------+----------+-----------+
5 rows in set (0.00 sec)
```

Find first five employee in table ordered by name:

SELECT *
FROM Employee
ORDER BY first_name
LIMIT 5;

```
+--------+------------+-----------+------------+------+--------+----------+-----------+
| emp_id | first_name | last_name | birth_date | sex  | salary | super_id | branch_id |
+--------+------------+-----------+------------+------+--------+----------+-----------+
|    107 | andy       | bernard   | 1973-10-01 | M    |  65000 |      106 |         3 |
|    103 | angela     | martin    | 1971-06-25 | F    |  63000 |      102 |         2 |
|    100 | david      | wallace   | 1967-11-17 | M    | 250000 |     NULL |         1 |
|    101 | jan        | jevinson  | 1961-05-11 | F    | 110000 |      100 |         1 |
|    108 | jim        | hairpert  | 1978-10-01 | M    |  71000 |      106 |         3 |
+--------+------------+-----------+------------+------+--------+----------+-----------+
5 rows in set (0.00 sec)
```

Find first and last name of all employees:

SELECT first_name,last_name
FROM Employee;

```
+------------+-----------+
| first_name | last_name |
+------------+-----------+
| david      | wallace   |
| jan        | jevinson  |
| micheal    | scott     |
| angela     | martin    |
| kelly      | kapoor    |
| stanley    | hudsen    |
| josh       | porter    |
| andy       | bernard   |
| jim        | hairpert  |
+------------+-----------+
9 rows in set (0.00 sec)
```

Find fore name and sur name of all employees:

SELECT first_name **AS** forename,last_name **AS** surname
FROM Employee;

```
+----------+----------+
| forename | surname  |
+----------+----------+
| david    | wallace  |
| jan      | jevinson |
| micheal  | scott    |
| angela   | martin   |
| kelly    | kapoor   |
| stanley  | hudsen   |
| josh     | porter   |
| andy     | bernard  |
| jim      | hairpert |
+----------+----------+
9 rows in set (0.00 sec)
```

Find all male employees:

SELECT *
FROM Employee
WHERE sex = 'M';

```
+--------+------------+-----------+------------+------+--------+----------+-----------+
| emp_id | first_name | last_name | birth_date | sex  | salary | super_id | branch_id |
+--------+------------+-----------+------------+------+--------+----------+-----------+
|    100 | david      | wallace   | 1967-11-17 | M    | 250000 |     NULL |         1 |
|    102 | micheal    | scott     | 1961-06-25 | M    |  75000 |      100 |         2 |
|    105 | stanley    | hudsen    | 1956-02-19 | M    |  69000 |      102 |         2 |
|    106 | josh       | porter    | 1969-08-05 | M    |  78000 |      100 |         3 |
|    107 | andy       | bernard   | 1973-10-01 | M    |  65000 |      106 |         3 |
|    108 | jim        | hairpert  | 1978-10-01 | M    |  71000 |      106 |         3 |
+--------+------------+-----------+------------+------+--------+----------+-----------+
6 rows in set (0.00 sec)
```

Find all employees from branch 2:

SELECT *
FROM Employee
WHERE branch_id = 2;

```
+--------+------------+-----------+------------+------+--------+----------+-----------+
| emp_id | first_name | last_name | birth_date | sex  | salary | super_id | branch_id |
+--------+------------+-----------+------------+------+--------+----------+-----------+
|    102 | micheal    | scott     | 1961-06-25 | M    |  75000 |      100 |         2 |
|    103 | angela     | martin    | 1971-06-25 | F    |  63000 |      102 |         2 |
|    104 | kelly      | kapoor    | 1980-02-05 | F    |  55000 |      102 |         2 |
|    105 | stanley    | hudsen    | 1956-02-19 | M    |  69000 |      102 |         2 |
+--------+------------+-----------+------------+------+--------+----------+-----------+
4 rows in set (0.00 sec)
```

Find all employees from branch 2:

SELECT *
FROM Employee
WHERE branch_id =2 **AND** sex = 'F';

```
+--------+------------+-----------+------------+------+--------+----------+-----------+
| emp_id | first_name | last_name | birth_date | sex  | salary | super_id | branch_id |
+--------+------------+-----------+------------+------+--------+----------+-----------+
|    103 | angela     | martin    | 1971-06-25 | F    |  63000 |      102 |         2 |
|    104 | kelly      | kapoor    | 1980-02-05 | F    |  55000 |      102 |         2 |
+--------+------------+-----------+------------+------+--------+----------+-----------+
2 rows in set (0.00 sec)
```

Find out all dintinct gender:

SELECT DISTINCT sex
FROM Employee;

```
+------+
| sex  |
+------+
| M    |
| F    |
+------+
2 rows in set (0.00 sec)
```

Find all employees id's and name who born after 1969:

SELECT emp_id **AS** id , first_name **AS** name
FROM Employee
WHERE birth_date > '1969-01-01';

```
+-----+--------+
| id  | name   |
+-----+--------+
| 103 | angela |
| 104 | kelly  |
| 106 | josh   |
| 107 | andy   |
| 108 | jim    |
+-----+--------+
5 rows in set (0.00 sec)
```

SELECT emp_id **AS** id , first_name **AS** name
FROM Employee

WHERE birth_date >= '1970-01-01';

OR

SELECT emp_id **AS** id , first_name **AS** name
FROM Employee
WHERE birth_date > '1970-01-01';

```
+-----+--------+
| id  | name   |
+-----+--------+
| 103 | angela |
| 104 | kelly  |
| 107 | andy   |
| 108 | jim    |
+-----+--------+
4 rows in set (0.00 sec)
```

NOTE:

here you see that if you write ‘1969-01-01’ it include year 1969 also but I want greater than 1969 that are possible when I write >= ‘1970-01-01’ or > ‘1970-01-01’ .

Find all employees who are female and born after 1960 or who make over 80000:

SELECT *
FROM Employee
WHERE (sex='F' **AND** birth_date > '1970-01-01') **OR** salary > 80000;

```
+--------+------------+-----------+------------+------+--------+----------+-----------+
| emp_id | first_name | last_name | birth_date | sex  | salary | super_id | branch_id |
+--------+------------+-----------+------------+------+--------+----------+-----------+
|    100 | david      | wallace   | 1967-11-17 | M    | 250000 |     NULL |         1 |
|    101 | jan        | jevinson  | 1961-05-11 | F    | 110000 |      100 |         1 |
|    103 | angela     | martin    | 1971-06-25 | F    |  63000 |      102 |         2 |
|    104 | kelly      | kapoor    | 1980-02-05 | F    |  55000 |      102 |         2 |
+--------+------------+-----------+------------+------+--------+----------+-----------+
4 rows in set (0.00 sec)
```

Find all employees born between 1970 and 1975:

SELECT *
FROM Employee
WHERE birth_date **BETWEEN** '1970-01-01' **AND** '1975-01-01';

```
+--------+------------+-----------+------------+------+--------+----------+-----------+
| emp_id | first_name | last_name | birth_date | sex  | salary | super_id | branch_id |
+--------+------------+-----------+------------+------+--------+----------+-----------+
|    103 | angela     | martin    | 1971-06-25 | F    |  63000 |      102 |         2 |
|    107 | andy       | bernard   | 1973-10-01 | M    |  65000 |      106 |         3 |
+--------+------------+-----------+------------+------+--------+----------+-----------+
2 rows in set (0.00 sec)
```

Find all employees named “jim , micheal ,johnny ,david”:

SELECT *
FROM Employee
WHERE first_name **IN** ('jim','micheal','johnny','david');

```
+--------+------------+-----------+------------+------+--------+----------+-----------+
| emp_id | first_name | last_name | birth_date | sex  | salary | super_id | branch_id |
+--------+------------+-----------+------------+------+--------+----------+-----------+
|    100 | david      | wallace   | 1967-11-17 | M    | 250000 |     NULL |         1 |
|    102 | micheal    | scott     | 1961-06-25 | M    |  75000 |      100 |         2 |
|    108 | jim        | hairpert  | 1978-10-01 | M    |  71000 |      106 |         3 |
+--------+------------+-----------+------------+------+--------+----------+-----------+
3 rows in set (0.00 sec)
```

Find number of employee:

SELECT COUNT(emp_id)
FROM Employee;

```
+---------------+
| COUNT(emp_id) |
+---------------+
|             9 |
+---------------+
1 row in set (0.00 sec)
```

Find average of all employee salaries:

SELECT AVG(salary)
FROM Employee;

```
+-------------+
| AVG(salary) |
+-------------+
|  92888.8889 |
+-------------+
1 row in set (0.00 sec)
```

Find sum of all employee salaries:

SELECT SUM(salary)
FROM Employee;

```
+-------------+
| SUM(salary) |
+-------------+
|      836000 |
+-------------+
1 row in set (0.00 sec)
```

Find number of distinct sex in employee table:

SELECT COUNT(DISTINCT sex)
FROM Employee;

```
+----------------------+
| COUNT(DISTINCT sex) |
+----------------------+
|                    2 |
+----------------------+
1 row in set (0.00 sec)
```

Find number of sex in employee table:

SELECT COUNT(sex)
FROM Employee;

```
+------------+
| COUNT(sex) |
+------------+
|          9 |
+------------+
1 row in set (0.00 sec)
```

Find how many sex in employee table:

SELECT COUNT(sex)
FROM Employee
GROUP BY sex;

```
+------------+
| COUNT(sex) |
+------------+
|          6 |
|          3 |
+------------+
2 rows in set (0.00 sec)
```

SELECT COUNT(sex),sex
FROM Employee
GROUP BY sex;

```
+------------+------+
| COUNT(sex) | sex  |
+------------+------+
|          6 | M    |
|          3 | F    |
+------------+------+
2 rows in set (0.00 sec)
```

SELECT sex,**COUNT**(sex)
FROM Employee
GROUP BY sex;

```
+------+------------+
| sex  | COUNT(sex) |
+------+------------+
| M    |          6 |
| F    |          3 |
+------+------------+
2 rows in set (0.00 sec)
```

Find how many sales of each salesman:

SELECT emp_id,**COUNT**(total_sales)
FROM works_with

GROUP BY (emp_id);

```
+--------+--------------------+
| emp_id | COUNT(total_sales) |
+--------+--------------------+
|    102 |                  2 |
|    105 |                  3 |
|    107 |                  2 |
|    108 |                  2 |
+--------+--------------------+
4 rows in set (0.00 sec)
```

Find total sales of each salesman:

SELECT emp_id,**SUM**(total_sales)
FROM works_with
GROUP BY (emp_id);

```
+--------+------------------+
| emp_id | SUM(total_sales) |
+--------+------------------+
|    102 |           282000 |
|    105 |           218000 |
|    107 |            31000 |
|    108 |           267000 |
+--------+------------------+
4 rows in set (0.00 sec)
```

Find total amount of money spend by each client:

SELECT client_id,**SUM**(total_sales)
FROM works_with
GROUP BY client_id;

```
+-----------+------------------+
| client_id | SUM(total_sales) |
+-----------+------------------+
|       400 |            55000 |
|       401 |           267000 |
|       402 |           255000 |
|       403 |            17000 |
|       404 |            33000 |
|       405 |            26000 |
|       406 |           145000 |
+-----------+------------------+
7 rows in set (0.00 sec)
```

Find which employee did minimum amount of sales:

NOTE:
having can't work with "max", "min" . It work on count when you find max and min value with group by used "order by and limit "
and also "where " can't used with "group by".

```
SELECT emp_id,SUM(total_sales) AS total_sales
FROM works_with
GROUP BY (emp_id)
ORDER BY (total_sales)
LIMIT 1;
```

```
+--------+-------------+
| emp_id | total_sales |
+--------+-------------+
|    107 |       31000 |
+--------+-------------+
1 row in set (0.00 sec)
```

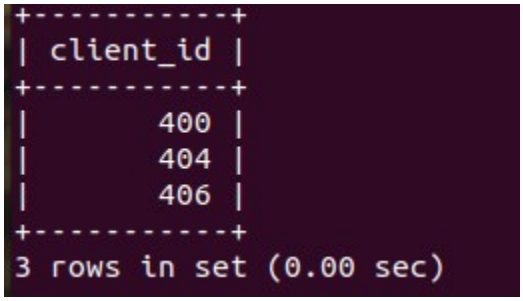
Find which employee did total_sales sales greater than 2:

```
SELECT emp_id,SUM(total_sales) AS total_sales
FROM works_with
GROUP BY (emp_id)
HAVING COUNT(*) > 2;
```

```
+--------+-------------+
| emp_id | total_sales |
+--------+-------------+
|    107 |       31000 |
+--------+-------------+
1 row in set (0.00 sec)
```

Find which employee did total_sales sales greater than 2 and find which client id it works with:

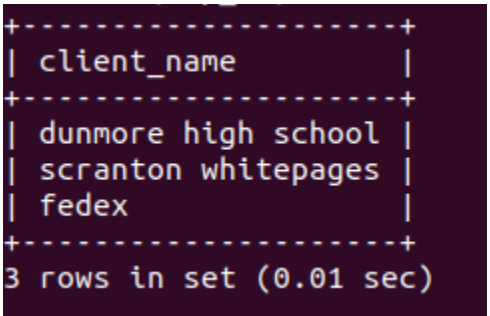
```
SELECT client_id
FROM works_with
WHERE emp_id IN  (

SELECT emp_id
FROM works_with
GROUP BY (emp_id)
HAVING COUNT(*) > 2
);
```



```
+-----------+
| client_id |
+-----------+
|       400 |
|       404 |
|       406 |
+-----------+
3 rows in set (0.00 sec)
```

<u>Find which employee did total_sales sales greater than 2 and find write its client name it works with:</u>

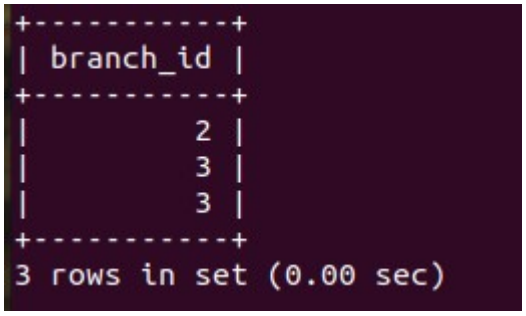
```
SELECT client_name
FROM client
WHERE client_id IN (

SELECT client_id
FROM works_with
WHERE emp_id IN (

SELECT emp_id
FROM works_with
GROUP BY (emp_id)
HAVING COUNT(*) > 2
));
```



```
+---------------------+
| client_name         |
+---------------------+
| dunmore high school |
| scranton whitepages |
| fedex               |
+---------------------+
3 rows in set (0.01 sec)
```

Find branch_id whose supply_type is “paper”:

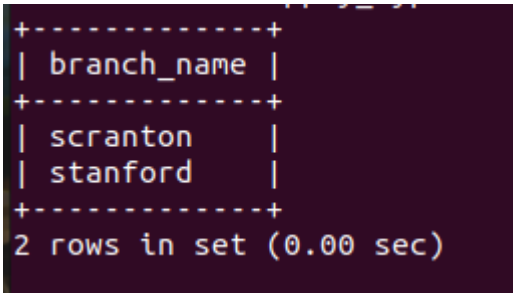
```
SELECT branch_id
FROM branch_supplier
WHERE supply_type = 'paper';
```



```
+-----------+
| branch_id |
+-----------+
|         2 |
|         3 |
|         3 |
+-----------+
3 rows in set (0.00 sec)
```

Find branch_name whose supply_type is “paper”:

```
SELECT branch_name
FROM branch
Where branch_id IN (

SELECT branch_id
FROM branch_supplier
WHERE supply_type = 'paper'
);
```



```
+-------------+
| branch_name |
+-------------+
| scranton    |
| stanford    |
+-------------+
2 rows in set (0.00 sec)
```

WILDCARDS

Find any client's who are an LLC:

SELECT client_name
FROM client
WHERE client_name **LIKE** '%LLC';

```
+-------------------+
| client_name       |
+-------------------+
| john daly law,LLC |
+-------------------+
1 row in set (0.00 sec)
```

Find any branch suppliers who are in the label business:

SELECT supplier_name
FROM branch_supplier
WHERE supplier_name **LIKE** '%labels%';

```
+--------------------+
| supplier_name      |
+--------------------+
| jt forms & labels  |
| stamord labels     |
+--------------------+
2 rows in set (0.00 sec)
```

Find any employee born on the 10th day of the month:

SELECT first_name
FROM Employee
WHERE birth_date **LIKE** '_____10%';

```
+------------+
| first_name |
+------------+
| andy       |
| jim        |
+------------+
2 rows in set (0.00 sec)
```

Find any clients who are high schools:

SELECT client_id,client_name
FROM client
WHERE client_name LIKE '%school%';

```
+-----------+----------------------+
| client_id | client_name          |
+-----------+----------------------+
|       400 | dunmore high school  |
+-----------+----------------------+
1 row in set (0.00 sec)
```

UNIONS

NOTE:
union are used when we same data type. And with “select” statement same data type of variable and same number of variable used.

Find a list of employee and branch names:

```
SELECT first_name AS company_name
FROM Employee
UNION
SELECT branch_name
FROM branch
UNION
SELECT client_name
FROM client;
```

```
+---------------------+
| company_name        |
+---------------------+
| david               |
| jan                 |
| micheal             |
| angela              |
| kelly               |
| stanley             |
| josh                |
| andy                |
| jim                 |
| corporate           |
| scranton            |
| stanford            |
| dunmore high school |
| lackawana country   |
| fedex               |
| john daly law,LLC   |
| scranton whitepages |
| times newspaper     |
+---------------------+
18 rows in set (0.00 sec)
```

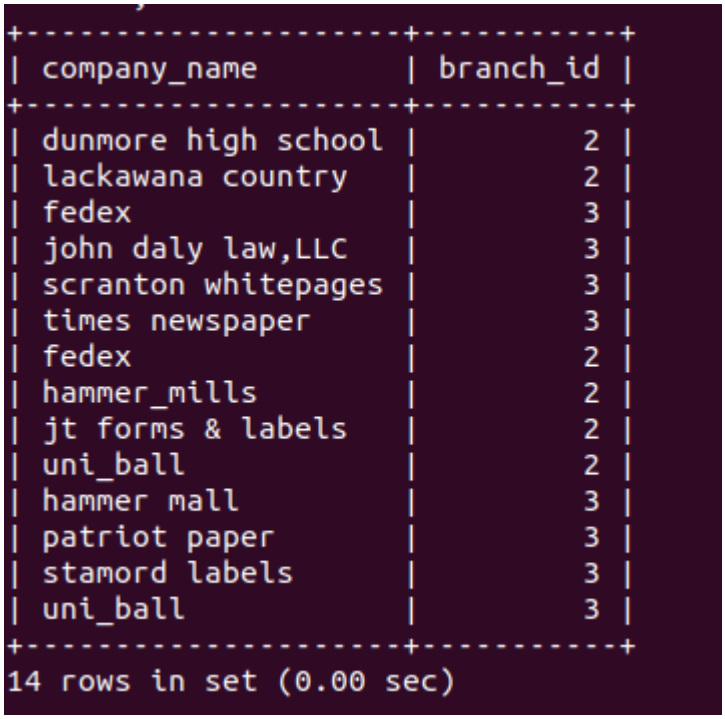
Find a list of all clients & branch suppliers' names:

SELECT client_name **AS** company_name
FROM client
UNION
SELECT supplier_name
FROM branch_supplier;

```
+---------------------+
| company_name        |
+---------------------+
| dunmore high school |
| lackawana country   |
| fedex               |
| john daly law,LLC   |
| scranton whitepages |
| times newspaper     |
| hammer_mills        |
| jt forms & labels   |
| uni_ball            |
| hammer mall         |
| patriot paper       |
| stamord labels      |
+---------------------+
12 rows in set (0.00 sec)
```

Find a list of all clients & branch suppliers' names and its branch_id:

```
SELECT client_name AS company_name, branch_id
FROM client
UNION
SELECT supplier_name,branch_id
FROM branch_supplier;
```



company_name	branch_id
dunmore high school	2
lackawana country	2
fedex	3
john daly law,LLC	3
scranton whitepages	3
times newspaper	3
fedex	2
hammer_mills	2
jt forms & labels	2
uni_ball	2
hammer mall	3
patriot paper	3
stamord labels	3
uni_ball	3

14 rows in set (0.00 sec)

JOINS

Find out branch manager name:

SELECT Employee.first_name , branch.branch_name

FROM Employee
JOIN branch
ON Employee.emp_id = branch.mgr_id;

```
+------------+-------------+
| first_name | branch_name |
+------------+-------------+
| david      | corporate   |
| micheal    | scranton    |
| josh       | stanford    |
+------------+-------------+
3 rows in set (0.00 sec)
```

Find out branch manager name and its name:

SELECT Employee.first_name , branch.branch_name
FROM Employee
JOIN branch
ON Employee.branch_id = branch.branch_id;

```
+------------+-------------+
| first_name | branch_name |
+------------+-------------+
| david      | corporate   |
| jan        | corporate   |
| micheal    | scranton    |
| angela     | scranton    |
| kelly      | scranton    |
| stanley    | scranton    |
| josh       | stanford    |
| andy       | stanford    |
| jim        | stanford    |
+------------+-------------+
9 rows in set (0.00 sec)
```

Find out branch name of each suppliers:

SELECT branch_supplier.supplier_name , branch.branch_name
FROM branch_supplier
JOIN branch
ON branch_supplier.branch_id = branch.branch_id;

supplier_name	branch_name
hammer_mills	scranton
jt forms & labels	scranton
uni_ball	scranton
hammer mall	stanford
patriot paper	stanford
stamord labels	stanford
uni_ball	stanford

7 rows in set (0.00 sec)

JOINS with GROUP BY

Find out total sales of each client name:

SELECT client.client_name, SUM(works_with.total_sales) AS sales
FROM client

JOIN works_with
ON client.client_id = works_with.client_id
GROUP BY works_with.client_id;

```
+---------------------+--------+
| client_name         | sales  |
+---------------------+--------+
| dunmore high school |  55000 |
| lackawana country   | 267000 |
| fedex               | 255000 |
| john daly law,LLC   |  17000 |
| scranton whitepages |  33000 |
| times newspaper     |  26000 |
| fedex               | 145000 |
+---------------------+--------+
7 rows in set (0.00 sec)
```

Find out total sales of each client name and its client_id:

SELECT client. client_name, SUM(works_with.total_sales) AS sales
FROM client
JOIN works_with
ON client.client_id = works_with.client_id
GROUP BY works_with.client_id;

```
+---------------------+-----------+--------+
| client_name         | client_id | sales  |
+---------------------+-----------+--------+
| dunmore high school |       400 |  55000 |
| lackawana country   |       401 | 267000 |
| fedex               |       402 | 255000 |
| john daly law,LLC   |       403 |  17000 |
| scranton whitepages |       404 |  33000 |
| times newspaper     |       405 |  26000 |
| fedex               |       406 | 145000 |
+---------------------+-----------+--------+
7 rows in set (0.00 sec)
```

Find names of all employees who have sold over 50,000

SELECT Employee.first_name, SUM(works_with.total_sales) AS total_sales
FROM Employee
JOIN works_with
ON Employee.emp_id = works_with.emp_id

```
GROUP BY (works_with.emp_id)
HAVING total_sales > 50000;
```

```
+------------+-------------+
| first_name | total_sales |
+------------+-------------+
| micheal    |      282000 |
| stanley    |      218000 |
| jim        |      267000 |
+------------+-------------+
3 rows in set (0.00 sec)
```

```
SELECT Employee.first_name
FROM Employee
JOIN works_with
ON Employee.emp_id = works_with.emp_id
GROUP BY (works_with.emp_id)
HAVING SUM(works_with.total_sales) > 50000;
```

```
+------------+
| first_name |
+------------+
| micheal    |
| stanley    |
| jim        |
+------------+
3 rows in set (0.00 sec)
```

Find all clients who are handles by the branch that Michael Scott manages
-- Assume you know Michael's ID

Find all clients who are handles by the branch that Michael Scott manages
-- Assume you DONT'T know Michael's ID

Find the names of employees who work with clients handled by the scranton branch

Find the names of all clients who have spent more than 100,000 dollars