PAPER COMPANY DATABASES

Vipul Gami

1. INTRODUCTION

Our project is to computerize paper company data to manage data, so that all the transaction of buying and selling become fast and there should not be any error in transaction like where the goods are going and coming from and who handles who. It keeps records of all employees, client, suppliers, branch. To ensure 100% successful implementation of day to day business. Our database has four modules. First module consists of all employee working for company. Second modules consist of all branches that company have. Third modules consist of all the suppliers from where the company order their products. And last modules consist of the client to which the company sell their products to.

The company is organized into branches. Each branch has a unique number, a name, and a particular employee who manages it.

The company makes its money by selling to clients. Each client has a name and a unique number to identify it.

The foundation of the company is its employees. Each employee has a name, birthday, gender, salary and a unique number.

An employee can work for one branch at a time, and each branch will be managed by one of the employees that work there. We'll also want to keep track of when the current manager started as manager.

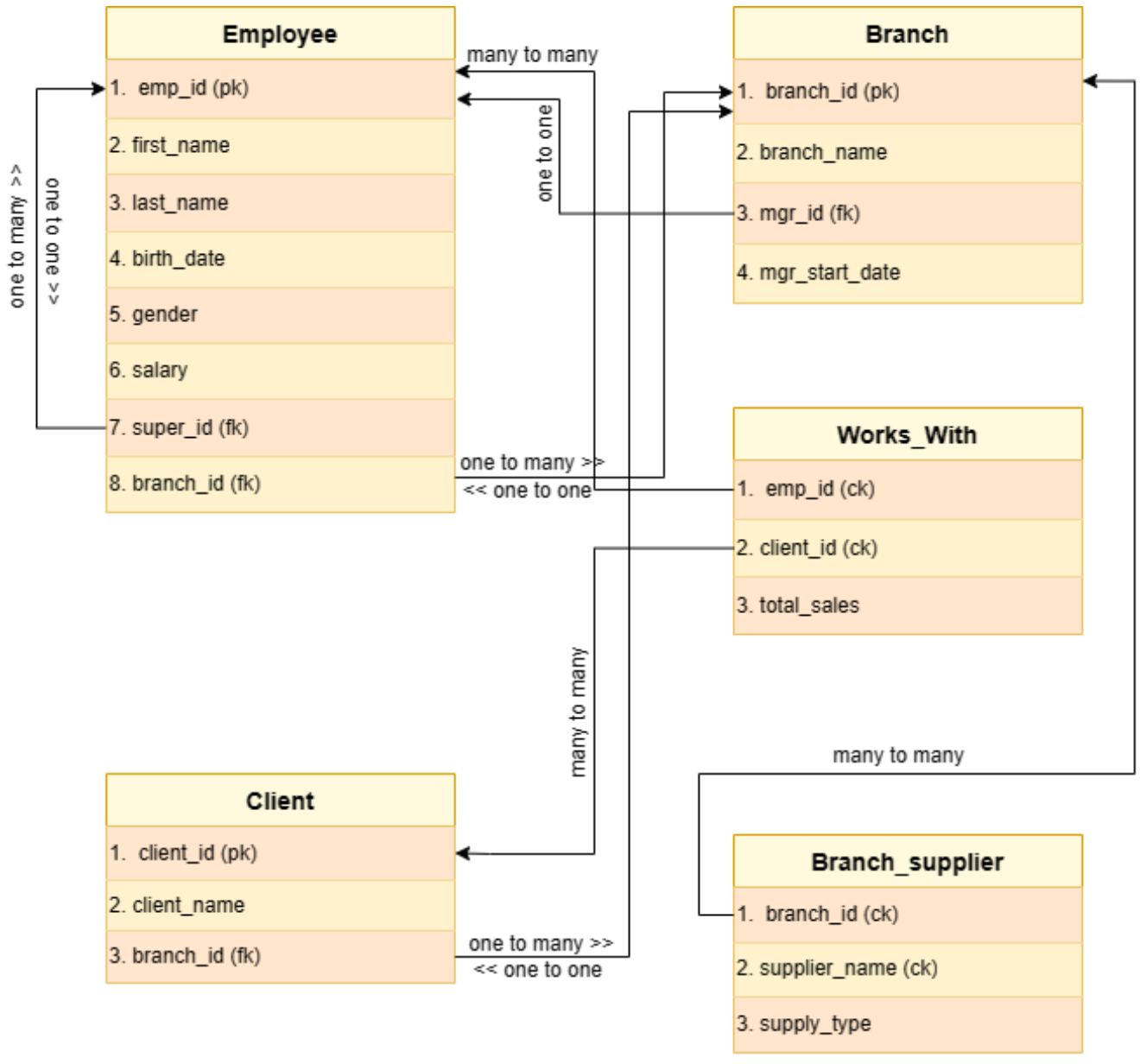
An employee can act as a supervisor for other employees at the branch, an employee may also act as the supervisor for employees at other branches. An employee can have at most one supervisor.

A branch may handle a number of clients, with each client having a name and a unique number to identify it. A single client may only be handled by one branch at a time.

Employees can work with clients controlled by their branch to sell them stuff. If necessary multiple employees can work with the same client. We'll want to keep track of how many dollars' worth of stuff each employee sells to each client they work with.

Many branches will need to work with suppliers to buy inventory. For each supplier we'll keep track of their name and the type of product they're selling the branch. A single supplier may supply products to multiple branches.

2. ER Diagram


Employee
1. emp_id (pk)
2. first_name
3. last_name
4. birth_date
5. gender
6. salary
7. super_id (fk)
8. branch_id (fk)
Branch
1. branch_id (pk)
2. branch_name
3. mgr_id (fk)
4. mgr_start_date
Works_With
1. emp_id (ck)
2. client_id (ck)
3. total_sales
Client
1. client_id (pk)
2. client_name
3. branch_id (fk)
Branch_supplier
1. branch_id (ck)
2. supplier_name (ck)
3. supply_type
many to many
one to one
one to many >>
<< one to one
one to many >>
<< one to one
many to many
one to many >>
<< one to one
many to many

3. Database Design

Databases: Paper Company

Tables:

a) Employee
b) Branch
c) Client
d) Works_with
e) Branch_supplier

4. Creating Table

a) Employee

```
mysql> CREATE TABLE employee (
    ->   emp_id INT PRIMARY KEY,
    ->   first_name VARCHAR(40),
    ->   last_name VARCHAR(40),
    ->   birth_day DATE,
    ->   gender VARCHAR(1),
    ->   salary INT,
    ->   super_id INT,
    ->   branch_id INT
    -> );
Query OK, 0 rows affected (0.05 sec)
```

b) Branch

```
mysql> CREATE TABLE branch (
    ->   branch_id INT PRIMARY KEY,
    ->   branch_name VARCHAR(40),
    ->   mgr_id INT,
    ->   mgr_start_date DATE,
    ->   FOREIGN KEY(mgr_id) REFERENCES employee(emp_id) ON DELETE SET NULL
    -> );
Query OK, 0 rows affected (0.03 sec)
```

c) Client

```
mysql> CREATE TABLE client (
    ->   client_id INT PRIMARY KEY,
    ->   client_name VARCHAR(40),
    ->   branch_id INT,
    ->   FOREIGN KEY(branch_id) REFERENCES branch(branch_id) ON DELETE SET NULL
    -> );
Query OK, 0 rows affected (0.03 sec)
```

d) Works_with

```
mysql> 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
    -> );
Query OK, 0 rows affected (0.07 sec)
```

e) Branch_supplier

```
mysql> CREATE TABLE branch_supplier (
    ->   branch_id INT,
    ->   supplier_name VARCHAR(40),
    ->   supply_type VARCHAR(40),
    ->   PRIMARY KEY(branch_id, supplier_name),
    ->   FOREIGN KEY(branch_id) REFERENCES branch(branch_id) ON DELETE CASCADE
    -> );
Query OK, 0 rows affected (0.10 sec)
```

5. Tables in databases

```
mysql> show tables;
+------------------+
| Tables_in_project |
+------------------+
| branch           |
| branch_supplier  |
| client           |
| employee         |
| works_with       |
+------------------+
5 rows in set (0.00 sec)
```

6. Data Definition language (DDL)

a) Creating Tables:

1) Employee

```
mysql> desc employee;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| emp_id     | int         | NO   | PRI | NULL    |       |
| first_name | varchar(40) | YES  |     | NULL    |       |
| last_name  | varchar(40) | YES  |     | NULL    |       |
| birth_day  | date        | YES  |     | NULL    |       |
| gender     | varchar(1)  | YES  |     | NULL    |       |
| salary     | int         | YES  |     | NULL    |       |
| super_id   | int         | YES  | MUL | NULL    |       |
| branch_id  | int         | YES  | MUL | NULL    |       |
+------------+-------------+------+-----+---------+-------+
8 rows in set (0.12 sec)
```

2) Branch

```
mysql> desc branch;
+----------------+-------------+------+-----+---------+-------+
| Field          | Type        | Null | Key | Default | Extra |
+----------------+-------------+------+-----+---------+-------+
| branch_id      | int         | NO   | PRI | NULL    |       |
| branch_name    | varchar(40) | YES  |     | NULL    |       |
| mgr_id         | int         | YES  | MUL | NULL    |       |
| mgr_start_date | date        | YES  |     | NULL    |       |
+----------------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

3) Client

```
mysql> desc client;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| client_id   | int         | NO   | PRI | NULL    |       |
| client_name | varchar(40) | YES  |     | NULL    |       |
| branch_id   | int         | YES  | MUL | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

4) Works_with

```
mysql> desc works_with;
+-------------+------+------+-----+---------+-------+
| 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)
```

5) Branch_supplier

```
mysql> desc branch_supplier;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| branch_id     | int         | NO   | PRI | NULL    |       |
| supplier_name | varchar(40) | NO   | PRI | NULL    |       |
| supply_type   | varchar(40) | YES  |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

b) Alter table

1) Alter table add column:

```
mysql> ALTER TABLE employee
    -> ADD FOREIGN KEY(branch_id)
    -> REFERENCES branch(branch_id)
    -> ON DELETE SET NULL;
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

2) Alter table modify column:

```
mysql> ALTER TABLE client
    -> MODIFY client_name varchar(60);
Query OK, 0 rows affected (0.08 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

3) Alter table rename column:

```
mysql> ALTER TABLE client
    -> CHANGE phone client_phone int;
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

4) Alter table drop column:

```
mysql> ALTER TABLE client
    -> DROP client_phone;
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

c) Rename table

```
mysql> ALTER TABLE client_day
    -> RENAME client;
Query OK, 0 rows affected (0.02 sec)
```

d) Truncate table

```
mysql> TRUNCATE test;
Query OK, 0 rows affected (0.04 sec)
```

e) Drop table

```
mysql> DROP TABLE test;
Query OK, 0 rows affected (0.01 sec)
```

7. Data Manipulation language (DML)

a. Insert into table

```
mysql> INSERT INTO employee VALUES(100, 'David', 'Wallace', '1967-11-17', 'M', 250000, NULL, NULL);
Query OK, 1 row affected (0.01 sec)
```

b. Update into table

```
mysql> UPDATE employee
    -> SET branch_id = 1
    -> WHERE emp_id = 100;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

c. Delete into table

```
mysql> DELETE FROM employee WHERE emp_id=1012;
Query OK, 1 row affected (0.01 sec)
```

8. Data Query Language (DQL)

a. Select query

```
mysql> -- Find all employees
mysql> SELECT *
    -> FROM employee;
+--------+------------+-----------+------------+--------+--------+----------+-----------+
| emp_id | first_name | last_name | birth_day  | gender | salary | super_id | branch_id |
+--------+------------+-----------+------------+--------+--------+----------+-----------+
|    100 | David      | Wallace   | 1967-11-17 | M      | 250000 |     NULL |         1 |
|    101 | Jan        | Levinson  | 1961-05-11 | F      | 110000 |      100 |         1 |
|    102 | Michael    | Scott     | 1964-03-15 | 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    | Hudson    | 1958-02-19 | M      |  69000 |      102 |         2 |
|    106 | Josh       | Porter    | 1969-09-05 | M      |  78000 |      100 |         3 |
|    107 | Andy       | Bernard   | 1973-07-22 | M      |  65000 |      106 |         3 |
|    108 | Jim        | Halpert   | 1978-10-01 | M      |  71000 |      106 |         3 |
+--------+------------+-----------+------------+--------+--------+----------+-----------+
9 rows in set (0.00 sec)
```

```
mysql> -- Find all clients
mysql> SELECT *
    -> FROM 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 |
+-----------+----------------------+-----------+
7 rows in set (0.00 sec)
```

b. Order by query ASC.

```
mysql>
mysql> -- Find all employees ordered by salary
mysql> SELECT *
    -> from employee
    -> ORDER BY salary ASC;
+--------+------------+-----------+------------+--------+--------+----------+-----------+
| emp_id | first_name | last_name | birth_day  | gender | 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   | 1973-07-22 | M      |  65000 |      106 |         3 |
|    105 | Stanley    | Hudson    | 1958-02-19 | M      |  69000 |      102 |         2 |
|    108 | Jim        | Halpert   | 1978-10-01 | M      |  71000 |      106 |         3 |
|    102 | Michael    | Scott     | 1964-03-15 | M      |  75000 |      100 |         2 |
|    106 | Josh       | Porter    | 1969-09-05 | M      |  78000 |      100 |         3 |
|    101 | Jan        | Levinson  | 1961-05-11 | F      | 110000 |      100 |         1 |
|    100 | David      | Wallace   | 1967-11-17 | M      | 250000 |     NULL |         1 |
+--------+------------+-----------+------------+--------+--------+----------+-----------+
9 rows in set (0.00 sec)
```

c. Order by query DESC

```
mysql>
mysql> SELECT *
    -> from employee
    -> ORDER BY salary DESC;
+--------+------------+-----------+------------+--------+--------+----------+-----------+
| emp_id | first_name | last_name | birth_day  | gender | salary | super_id | branch_id |
+--------+------------+-----------+------------+--------+--------+----------+-----------+
|    100 | David      | Wallace   | 1967-11-17 | M      | 250000 |     NULL |         1 |
|    101 | Jan        | Levinson  | 1961-05-11 | F      | 110000 |      100 |         1 |
|    106 | Josh       | Porter    | 1969-09-05 | M      |  78000 |      100 |         3 |
|    102 | Michael    | Scott     | 1964-03-15 | M      |  75000 |      100 |         2 |
|    108 | Jim        | Halpert   | 1978-10-01 | M      |  71000 |      106 |         3 |
|    105 | Stanley    | Hudson    | 1958-02-19 | M      |  69000 |      102 |         2 |
|    107 | Andy       | Bernard   | 1973-07-22 | 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)
```

d. Order by column

```
mysql>
mysql> -- Find all employees ordered by gender then name
mysql> SELECT *
    -> from employee
    -> ORDER BY gender, first_name;
+--------+------------+-----------+------------+--------+--------+----------+-----------+
| emp_id | first_name | last_name | birth_day  | gender | salary | super_id | branch_id |
+--------+------------+-----------+------------+--------+--------+----------+-----------+
|    103 | Angela     | Martin    | 1971-06-25 | F      |  63000 |      102 |         2 |
|    101 | Jan        | Levinson  | 1961-05-11 | F      | 110000 |      100 |         1 |
|    104 | Kelly      | Kapoor    | 1980-02-05 | F      |  55000 |      102 |         2 |
|    107 | Andy       | Bernard   | 1973-07-22 | M      |  65000 |      106 |         3 |
|    100 | David      | Wallace   | 1967-11-17 | M      | 250000 |     NULL |         1 |
|    108 | Jim        | Halpert   | 1978-10-01 | M      |  71000 |      106 |         3 |
|    106 | Josh       | Porter    | 1969-09-05 | M      |  78000 |      100 |         3 |
|    102 | Michael    | Scott     | 1964-03-15 | M      |  75000 |      100 |         2 |
|    105 | Stanley    | Hudson    | 1958-02-19 | M      |  69000 |      102 |         2 |
+--------+------------+-----------+------------+--------+--------+----------+-----------+
9 rows in set (0.00 sec)
```

e. Limit query

```
mysql>
mysql> -- Find the first 5 employees in the table
mysql> SELECT *
    -> from employee
    -> LIMIT 5;
+--------+------------+-----------+------------+--------+--------+----------+-----------+
| emp_id | first_name | last_name | birth_day  | gender | salary | super_id | branch_id |
+--------+------------+-----------+------------+--------+--------+----------+-----------+
|    100 | David      | Wallace   | 1967-11-17 | M      | 250000 |     NULL |         1 |
|    101 | Jan        | Levinson  | 1961-05-11 | F      | 110000 |      100 |         1 |
|    102 | Michael    | Scott     | 1964-03-15 | 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)
```

f. Select query with specific column

```
mysql>
mysql> -- Find the first and last names of all employees
mysql> SELECT first_name, employee.last_name
    -> FROM employee;
+------------+-----------+
| first_name | last_name |
+------------+-----------+
| David      | Wallace   |
| Jan        | Levinson  |
| Michael    | Scott     |
| Angela     | Martin    |
| Kelly      | Kapoor    |
| Stanley    | Hudson    |
| Josh       | Porter    |
| Andy       | Bernard   |
| Jim        | Halpert   |
+------------+-----------+
9 rows in set (0.00 sec)
```

g. Select query with column name change

```
mysql>
mysql> -- Find the forename and surnames names of all employees
mysql> SELECT first_name AS forename, employee.last_name AS surname
    -> FROM employee;
+----------+----------+
| forename | surname  |
+----------+----------+
| David    | Wallace  |
| Jan      | Levinson |
| Michael  | Scott    |
| Angela   | Martin   |
| Kelly    | Kapoor   |
| Stanley  | Hudson   |
| Josh     | Porter   |
| Andy     | Bernard  |
| Jim      | Halpert  |
+----------+----------+
9 rows in set (0.00 sec)
```

h. Distinct query

```
mysql> -- Find out all the different genders
mysql> SELECT DISTINCT(gender)
    -> FROM employee;
+--------+
| gender |
+--------+
| M      |
| F      |
+--------+
2 rows in set (0.00 sec)
```

9. Using where clause

a. With Comparision Operator

```
mysql>
mysql> -- Find all male employees
mysql> SELECT *
    -> FROM employee
    -> WHERE gender = 'M';
+--------+------------+-----------+------------+--------+--------+----------+-----------+
| emp_id | first_name | last_name | birth_day  | gender | salary | super_id | branch_id |
+--------+------------+-----------+------------+--------+--------+----------+-----------+
|    100 | David      | Wallace   | 1967-11-17 | M      | 250000 |     NULL |         1 |
|    102 | Michael    | Scott     | 1964-03-15 | M      |  75000 |      100 |         2 |
|    105 | Stanley    | Hudson    | 1958-02-19 | M      |  69000 |      102 |         2 |
|    106 | Josh       | Porter    | 1969-09-05 | M      |  78000 |      100 |         3 |
|    107 | Andy       | Bernard   | 1973-07-22 | M      |  65000 |      106 |         3 |
|    108 | Jim        | Halpert   | 1978-10-01 | M      |  71000 |      106 |         3 |
+--------+------------+-----------+------------+--------+--------+----------+-----------+
6 rows in set (0.00 sec)
```

```
mysql>
mysql> -- Find all employees at branch 2
mysql> SELECT *
    -> FROM employee
    -> WHERE branch_id = 2;
+--------+------------+-----------+------------+--------+--------+----------+-----------+
| emp_id | first_name | last_name | birth_day  | gender | salary | super_id | branch_id |
+--------+------------+-----------+------------+--------+--------+----------+-----------+
|    102 | Michael    | Scott     | 1964-03-15 | 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    | Hudson    | 1958-02-19 | M      |  69000 |      102 |         2 |
+--------+------------+-----------+------------+--------+--------+----------+-----------+
4 rows in set (0.00 sec)
```

```
mysql>
mysql> -- Find all employee's id's and names who were born after 1969
mysql> SELECT emp_id, first_name, last_name
    -> FROM employee
    -> WHERE birth_day >= 1970-01-01;
+--------+------------+-----------+
| emp_id | first_name | last_name |
+--------+------------+-----------+
|    100 | David      | Wallace   |
|    101 | Jan        | Levinson  |
|    102 | Michael    | Scott     |
|    103 | Angela     | Martin    |
|    104 | Kelly      | Kapoor    |
|    105 | Stanley    | Hudson    |
|    106 | Josh       | Porter    |
|    107 | Andy       | Bernard   |
|    108 | Jim        | Halpert   |
+--------+------------+-----------+
9 rows in set, 1 warning (0.00 sec)
```

10. Using Logical Operator

a. Using AND operator

```
mysql>
mysql> -- Find all female employees at branch 2
mysql> SELECT *
    -> FROM employee
    -> WHERE branch_id = 2 AND gender = 'F';
+--------+------------+-----------+------------+--------+--------+----------+-----------+
| emp_id | first_name | last_name | birth_day  | gender | 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)
```

b. Using AND/ OR operator

```
mysql>
mysql> -- Find all employees who are female & born after 1969 or who make over 80000
mysql> SELECT *
    -> FROM employee
    -> WHERE (birth_day >= '1970-01-01' AND gender = 'F') OR salary > 80000;
+--------+------------+-----------+------------+--------+--------+----------+-----------+
| emp_id | first_name | last_name | birth_day  | gender | salary | super_id | branch_id |
+--------+------------+-----------+------------+--------+--------+----------+-----------+
|    100 | David      | Wallace   | 1967-11-17 | M      | 250000 |     NULL |         1 |
|    101 | Jan        | Levinson  | 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)
```

c. Using between clause

```
mysql>
mysql> -- Find all employees born between 1970 and 1975
mysql> SELECT *
    -> FROM employee
    -> WHERE birth_day BETWEEN '1970-01-01' AND '1975-01-01';
+--------+------------+-----------+------------+--------+--------+----------+-----------+
| emp_id | first_name | last_name | birth_day  | gender | salary | super_id | branch_id |
+--------+------------+-----------+------------+--------+--------+----------+-----------+
|    103 | Angela     | Martin    | 1971-06-25 | F      |  63000 |      102 |         2 |
|    107 | Andy       | Bernard   | 1973-07-22 | M      |  65000 |      106 |         3 |
+--------+------------+-----------+------------+--------+--------+----------+-----------+
2 rows in set (0.00 sec)
```

d. Using IN clause

```
mysql>
mysql> -- Find all employees named Jim, Michael, Johnny or David
mysql> SELECT *
    -> FROM employee
    -> WHERE first_name IN ('Jim', 'Michael', 'Johnny', 'David');
+--------+------------+-----------+------------+--------+--------+----------+-----------+
| emp_id | first_name | last_name | birth_day  | gender | salary | super_id | branch_id |
+--------+------------+-----------+------------+--------+--------+----------+-----------+
|    100 | David      | Wallace   | 1967-11-17 | M      | 250000 |     NULL |         1 |
|    102 | Michael    | Scott     | 1964-03-15 | M      |  75000 |      100 |         2 |
|    108 | Jim        | Halpert   | 1978-10-01 | M      |  71000 |      106 |         3 |
+--------+------------+-----------+------------+--------+--------+----------+-----------+
3 rows in set (0.00 sec)
```

11. Aggregate function

a. Count function

```
mysql>
mysql> -- Find the number of employees
mysql> SELECT COUNT(super_id)
    -> FROM employee;
+-----------------+
| COUNT(super_id) |
+-----------------+
|               8 |
+-----------------+
1 row in set (0.00 sec)
```

b. Average function

```
mysql>
mysql> -- Find the average of all employee's salaries
mysql> SELECT AVG(salary)
    -> FROM employee;
+-------------+
| AVG(salary) |
+-------------+
|  92888.8889 |
+-------------+
1 row in set (0.00 sec)
```

c. Sum function

```
mysql>
mysql> -- Find the sum of all employee's salaries
mysql> SELECT SUM(salary)
    -> FROM employee;
+-------------+
| SUM(salary) |
+-------------+
|      836000 |
+-------------+
1 row in set (0.00 sec)
```

12. Group By clause

```
mysql>
mysql> -- Find out how many males and females there are
mysql> SELECT COUNT(gender), gender
    -> FROM employee
    -> GROUP BY gender;
+---------------+--------+
| COUNT(gender) | gender |
+---------------+--------+
|             6 | M      |
|             3 | F      |
+---------------+--------+
2 rows in set (0.00 sec)
```

```
mysql>
mysql> -- Find the total sales of each salesman
mysql> SELECT SUM(total_sales), emp_id
    -> FROM works_with
    -> GROUP BY client_id;
+------------------+--------+
| SUM(total_sales) | emp_id |
+------------------+--------+
|            55000 |    105 |
|           267000 |    102 |
|            22500 |    108 |
|            17000 |    107 |
|            33000 |    105 |
|            26000 |    107 |
|           145000 |    102 |
+------------------+--------+
7 rows in set (0.00 sec)
```

```
mysql>
mysql> -- Find the total amount of money spent by each client
mysql> SELECT SUM(total_sales), client_id
    -> FROM works_with
    -> GROUP BY client_id;
+------------------+-----------+
| SUM(total_sales) | client_id |
+------------------+-----------+
|            55000 |       400 |
|           267000 |       401 |
|            22500 |       402 |
|            17000 |       403 |
|            33000 |       404 |
|            26000 |       405 |
|           145000 |       406 |
+------------------+-----------+
7 rows in set (0.00 sec)
```

13. Like operator

```
mysql> -- % = any # characters, _ = one character
mysql>
mysql> -- Find any client's who are an LLC
mysql> SELECT *
    -> FROM client
    -> WHERE client_name LIKE '%LLC';
+-----------+--------------------+-----------+
| client_id | client_name        | branch_id |
+-----------+--------------------+-----------+
|       403 | John Daly Law, LLC |         3 |
+-----------+--------------------+-----------+
1 row in set (0.01 sec)
```

```
mysql>
mysql> -- Find any branch suppliers who are in the label business
mysql> SELECT *
    -> FROM branch_supplier
    -> WHERE supplier_name LIKE '% Label%';
+-----------+---------------------+--------------+
| branch_id | supplier_name       | supply_type  |
+-----------+---------------------+--------------+
|         2 | J.T. Forms & Labels | Custom Forms |
+-----------+---------------------+--------------+
1 row in set (0.00 sec)
```

```
mysql>
mysql> -- Find any employee born on the 10th day of the month
mysql> SELECT *
    -> FROM employee
    -> WHERE birth_day LIKE '_____10%';
+--------+------------+-----------+------------+--------+--------+----------+-----------+
| emp_id | first_name | last_name | birth_day  | gender | salary | super_id | branch_id |
+--------+------------+-----------+------------+--------+--------+----------+-----------+
|    108 | Jim        | Halpert   | 1978-10-01 | M      |  71000 |      106 |         3 |
+--------+------------+-----------+------------+--------+--------+----------+-----------+
1 row in set (0.00 sec)
```

```
mysql>
mysql> -- Find any clients who are schools
mysql> SELECT *
    -> FROM client
    -> WHERE client_name LIKE '%Highschool%';
+-----------+--------------------+-----------+
| client_id | client_name        | branch_id |
+-----------+--------------------+-----------+
|       400 | Dunmore Highschool |         2 |
+-----------+--------------------+-----------+
1 row in set (0.00 sec)
```

14. Union

```
mysql>
mysql> -- Find a list of employee and branch names
mysql> SELECT employee.first_name AS Employee_Branch_Names
    -> FROM employee
    -> UNION
    -> SELECT branch.branch_name
    -> FROM branch;
+-----------------------+
| Employee_Branch_Names |
+-----------------------+
| David                 |
| Jan                   |
| Michael               |
| Angela                |
| Kelly                 |
| Stanley               |
| Josh                  |
| Andy                  |
| Jim                   |
| Corporate             |
| Scranton              |
| Stamford              |
| Buffalo               |
+-----------------------+
13 rows in set (0.00 sec)
```

```
mysql>
mysql> -- Find a list of all clients & branch suppliers' names
mysql> SELECT client.client_name AS Non_Employee_Entities, client.branch_id AS Branch_ID
    -> FROM client
    -> UNION
    -> SELECT branch_supplier.supplier_name, branch_supplier.branch_id
    -> FROM branch_supplier;
+----------------------+-----------+
| Non_Employee_Entities | Branch_ID |
+----------------------+-----------+
| Dunmore Highschool    |         2 |
| Lackawana Country     |         2 |
| FedEx                 |         3 |
| John Daly Law, LLC    |         3 |
| Scranton Whitepages   |         2 |
| Times Newspaper       |         3 |
| FedEx                 |         2 |
| Hammer Mill           |         2 |
| J.T. Forms & Labels   |         2 |
| Uni-ball              |         2 |
| Hammer Mill           |         3 |
| Patriot Paper         |         3 |
| Stamford Lables       |         3 |
| Uni-ball              |         3 |
+----------------------+-----------+
14 rows in set (0.00 sec)
```

15. Joins

```
mysql>
mysql> SELECT employee.emp_id, employee.first_name, branch.branch_name
    -> FROM employee
    -> JOIN branch    -- LEFT JOIN, RIGHT JOIN
    -> ON employee.emp_id = branch.mgr_id;
+--------+------------+-------------+
| emp_id | first_name | branch_name |
+--------+------------+-------------+
|    100 | David      | Corporate   |
|    102 | Michael    | Scranton    |
|    106 | Josh       | Stamford    |
+--------+------------+-------------+
3 rows in set (0.00 sec)
```

16. Subquery

```
mysql>
mysql> -- Find names of all employees who have sold over 50,000
mysql> SELECT employee.first_name, employee.last_name
    -> FROM employee
    -> WHERE employee.emp_id IN (SELECT works_with.emp_id
    ->                           FROM works_with
    ->                           WHERE works_with.total_sales > 50000);
+------------+-----------+
| first_name | last_name |
+------------+-----------+
| Michael    | Scott     |
| Stanley    | Hudson    |
+------------+-----------+
2 rows in set (0.01 sec)
```

```
mysql>
mysql> -- Find all clients who are handles by the branch that Michael Scott manages
mysql> -- Assume you know Michael's ID
mysql> SELECT client.client_id, client.client_name
    -> FROM client
    -> WHERE client.branch_id = (SELECT branch.branch_id
    ->                           FROM branch
    ->                           WHERE branch.mgr_id = 102);
+-----------+--------------------+
| client_id | client_name        |
+-----------+--------------------+
|       400 | Dunmore Highschool |
|       401 | Lackawana Country  |
|       404 | Scranton Whitepages|
|       406 | FedEx              |
+-----------+--------------------+
4 rows in set (0.00 sec)
```

```
mysql>
mysql>  -- Find all clients who are handles by the branch that Michael Scott manages
mysql>  -- Assume you DONT'T know Michael's ID
mysql>  SELECT client.client_id, client.client_name
    ->  FROM client
    ->  WHERE client.branch_id = (SELECT branch.branch_id
    ->                            FROM branch
    ->                            WHERE branch.mgr_id = (SELECT employee.emp_id
    ->                                                   FROM employee
    ->                                                   WHERE employee.first_name = 'Michael' AND employee.last_name ='Scott'
    ->                                                   LIMIT 1));
+-----------+--------------------+
| client_id | client_name        |
+-----------+--------------------+
|       400 | Dunmore Highschool |
|       401 | Lackawana Country  |
|       404 | Scranton Whitepages|
|       406 | FedEx              |
+-----------+--------------------+
4 rows in set (0.00 sec)
```

```
mysql>
mysql>
mysql> -- Find the names of employees who work with clients handled by the scranton branch
mysql> SELECT employee.first_name, employee.last_name
    -> FROM employee
    -> WHERE employee.emp_id IN (
    ->                           SELECT works_with.emp_id
    ->                           FROM works_with
    ->                           )
    -> AND employee.branch_id = 2;
+------------+-----------+
| first_name | last_name |
+------------+-----------+
| Michael    | Scott     |
| Stanley    | Hudson    |
+------------+-----------+
2 rows in set (0.00 sec)
```

```
mysql>
mysql> -- Find the names of all clients who have spent more than 100,000 dollars
mysql> SELECT client.client_name
    -> FROM client
    -> WHERE client.client_id IN (
    ->                           SELECT client_id
    ->                           FROM (
    ->                                 SELECT SUM(works_with.total_sales) AS totals, client_id
    ->                                 FROM works_with
    ->                                 GROUP BY client_id) AS total_client_sales
    ->                           WHERE totals > 100000
    -> );
+--------------------+
| client_name        |
+--------------------+
| Lackawana Country  |
| FedEx              |
+--------------------+
2 rows in set (0.00 sec)
```

17. Views

```
mysql> CREATE VIEW employee_view AS
    -> SELECT emp_id,first_name,last_name,salary
    -> FROM employee;
Query OK, 0 rows affected (0.06 sec)
```

```
mysql> SELECT * FROM employee_view;
+--------+------------+-----------+--------+
| emp_id | first_name | last_name | salary |
+--------+------------+-----------+--------+
|    100 | David      | Wallace   | 250000 |
|    101 | Jan        | Levinson  | 110000 |
|    102 | Michael    | Scott     |  75000 |
|    103 | Angela     | Martin    |  63000 |
|    104 | Kelly      | Kapoor    |  55000 |
|    105 | Stanley    | Hudson    |  69000 |
|    106 | Josh       | Porter    |  78000 |
|    107 | Andy       | Bernard   |  65000 |
|    108 | Jim        | Halpert   |  71000 |
+--------+------------+-----------+--------+
9 rows in set (0.00 sec)
```