Creating Tables

```
INT                     -- Whole Numbers
DECIMAL(M,N)                -- Decimal Numbers - Exact Value
VARCHAR(l)                -- String of text of length l
BLOB                     -- Binary Large Object, Stores large data
DATE                     -- 'YYYY-MM-DD'
TIMESTAMP                  -- 'YYYY-MM-DD HH:MM:SS' - used for recording events
```

```
-- Creating tables
CREATE TABLE student (
student_id INT PRIMARY KEY,
name VARCHAR(40),
major VARCHAR(40)
-- PRIMARY KEY(student_id)
);
```

```
DESCRIBE student;
DROP TABLE student;
ALTER TABLE student ADD gpa DECIMAL;
ALTER TABLE student DROP COLUMN gpa;
```

Inserting Data

```
INSERT INTO student VALUES(1, 'Jack', 'Biology');

INSERT INTO student VALUES(2, 'Kate', 'Sociology');

INSERT INTO student(student_id, name) VALUES(3, 'Claire');

INSERT INTO student VALUES(4, 'Jack', 'Biology');

INSERT INTO student VALUES(5, 'Mike', 'Computer Science');
```

Constraints

```
CREATE TABLE student (

student_id INT PRIMARY KEY AUTO_INCREMENT,

name VARCHAR(40) NOT NULL,

-- name VARCHAR(40) UNIQUE,

major VARCHAR(40) DEFAULT 'undecided',

);
```

Update & Delete

```
DELETE FROM student;

DELETE FROM student
WHERE student_id = 4;

DELETE FROM student
WHERE major = 'Sociology' AND name = 'Kate';

UPDATE student
SET major = 'Undecided';

UPDATE student
SET name = 'Johnny'
WHERE student_id = 4;

UPDATE student
SET major = 'Biological Sciences'
WHERE major = 'Biology';

UPDATE student
SET major = 'Biosociology'
WHERE major = 'Biology' OR major = 'sociology'

UPDATE student
SET major = 'Undecided', name = 'Tom'
WHERE student_id = 4;
```

Basic Queries

```
SELECT *
FROM student;
```

```
SELECT student.name, student.major
FROM student;
```

```
SELECT *
FROM student
WHERE name = 'Jack';
```

```
SELECT *
FROM student
WHERE student_id > 2;
```

```
SELECT *
FROM student
WHERE major = 'Biology' AND student_id > 1;
```

Company Database

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

Labels

🟥	Primary Key
🟩	Foreign Key
🟦	Attribute

Creating Company Database

```
CREATE TABLE employee (
emp_id INT PRIMARY KEY,
first_name VARCHAR(40),
last_name VARCHAR(40),
birth_day DATE,
sex VARCHAR(1),
salary INT,
super_id INT,
branch_id INT
);

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
);

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

ALTER TABLE employee
ADD FOREIGN KEY(super_id)
REFERENCES employee(emp_id)
```

```
ON DELETE SET NULL;

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
);

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
);

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
);

-- -----------------------------------------------------------------------------


-- Corporate
INSERT INTO employee VALUES(100, 'David', 'Wallace', '1967-11-17', 'M', 250000, NULL, NULL);
INSERT INTO branch VALUES(1, 'Corporate', 100, '2006-02-09');

UPDATE employee
SET branch_id = 1
WHERE emp_id = 100;

INSERT INTO employee VALUES(101, 'Jan', 'Levinson', '1961-05-11', 'F', 110000, 100, 1);

-- Scranton
INSERT INTO employee VALUES(102, 'Michael', 'Scott', '1964-03-15', 'M', 75000, 100, NULL);

INSERT INTO branch VALUES(2, 'Scranton', 102, '1992-04-06');

UPDATE employee
SET branch_id = 2
WHERE emp_id = 102;

INSERT INTO employee VALUES(103, 'Angela', 'Martin', '1971-06-25', 'F', 63000, 102, 2);
INSERT INTO employee VALUES(104, 'Kelly', 'Kapoor', '1980-02-05', 'F', 55000, 102, 2);
INSERT INTO employee VALUES(105, 'Stanley', 'Hudson', '1958-02-19', 'M', 69000, 102, 2);

-- Stamford
INSERT INTO employee VALUES(106, 'Josh', 'Porter', '1969-09-05', 'M', 78000, 100, NULL);

INSERT INTO branch VALUES(3, 'Stamford', 106, '1998-02-13');

UPDATE employee
SET branch_id = 3
WHERE emp_id = 106;

INSERT INTO employee VALUES(107, 'Andy', 'Bernard', '1973-07-22', 'M', 65000, 106, 3);
```

```sql
INSERT INTO employee VALUES(108, 'Jim', 'Halpert', '1978-10-01', 'M', 71000, 106, 3);


-- BRANCH SUPPLIER
INSERT INTO branch_supplier VALUES(2, 'Hammer Mill', 'Paper');
INSERT INTO branch_supplier VALUES(2, 'Uni-ball', 'Writing Utensils');
INSERT INTO branch_supplier VALUES(3, 'Patriot Paper', 'Paper');
INSERT INTO branch_supplier VALUES(2, 'J.T. Forms & Labels', 'Custom Forms');
INSERT INTO branch_supplier VALUES(3, 'Uni-ball', 'Writing Utensils');
INSERT INTO branch_supplier VALUES(3, 'Hammer Mill', 'Paper');
INSERT INTO branch_supplier VALUES(3, 'Stamford Lables', 'Custom Forms');

-- CLIENT
INSERT INTO client VALUES(400, 'Dunmore Highschool', 2);
INSERT INTO client VALUES(401, 'Lackawana Country', 2);
INSERT INTO client VALUES(402, 'FedEx', 3);
INSERT INTO client VALUES(403, 'John Daly Law, LLC', 3);
INSERT INTO client VALUES(404, 'Scranton Whitepages', 2);
INSERT INTO client VALUES(405, 'Times Newspaper', 3);
INSERT INTO client VALUES(406, 'FedEx', 2);

-- WORKS_WITH
INSERT INTO works_with VALUES(105, 400, 55000);
INSERT INTO works_with VALUES(102, 401, 267000);
INSERT INTO works_with VALUES(108, 402, 22500);
INSERT INTO works_with VALUES(107, 403, 5000);
INSERT INTO works_with VALUES(108, 403, 12000);
INSERT INTO works_with VALUES(105, 404, 33000);
INSERT INTO works_with VALUES(107, 405, 26000);
INSERT INTO works_with VALUES(102, 406, 15000);
INSERT INTO works_with VALUES(105, 406, 130000);
```

More Basic Queries

```
-- Find all employees
SELECT *
FROM employee;

-- Find all clients
SELECT *
FROM clients;

-- Find all employees ordered by salary
SELECT *
from employee
ORDER BY salary ASC/DESC;

-- Find all employees ordered by sex then name
SELECT *
from employee
ORDER BY sex, name;

-- Find the first 5 employees in the table
SELECT *
from employee
LIMIT 5;

-- Find the first and last names of all employees
SELECT first_name, employee.last_name
FROM employee;
```

```sql
-- Find the forename and surnames names of all employees
SELECT first_name AS forename, employee.last_name AS surname
FROM employee;

-- Find out all the different genders
SELECT DISCINCT sex
FROM employee;

-- Find all male employees
SELECT *
FROM employee
WHERE sex = 'M';

-- Find all employees at branch 2
SELECT *
FROM employee
WHERE branch_id = 2;

-- Find all employee's id's and names who were born after 1969
SELECT emp_id, first_name, last_name
FROM employee
WHERE birth_day >= 1970-01-01;

-- Find all female employees at branch 2
SELECT *
FROM employee
WHERE branch_id = 2 AND sex = 'F';

-- Find all employees who are female & born after 1969 or who make over 80000
SELECT *
FROM employee
```

```sql
WHERE (birth_day >= '1970-01-01' AND sex = 'F') OR salary > 80000;

-- Find all employees born between 1970 and 1975
SELECT *
FROM employee
WHERE birth_day BETWEEN '1970-01-01' AND '1975-01-01';

-- Find all employees named Jim, Michael, Johnny or David
SELECT *
FROM employee
WHERE first_name IN ('Jim', 'Michael', 'Johnny', 'David');
```

Functions

```sql
-- Find the number of employees
SELECT COUNT(super_id)
FROM employee;

-- Find the average of all employee's salaries
SELECT AVG(salary)
FROM employee;

-- Find the sum of all employee's salaries
SELECT SUM(salary)
FROM employee;

-- Find out how many males and females there are
SELECT COUNT(sex), sex
FROM employee
```

```sql
GROUP BY sex

-- Find the total sales of each salesman
SELECT SUM(total_sales), emp_id
FROM works_with
GROUP BY client_id;

-- Find the total amount of money spent by each client
SELECT SUM(total_sales), client_id
FROM works_with
GROUP BY client_id;
```

Wildcards

```sql
-- % = any # characters, _ = one character

-- Find any client's who are an LLC
SELECT *
FROM client
WHERE client_name LIKE '%LLC';

-- Find any branch suppliers who are in the label business
SELECT *
FROM branch_supplier
WHERE supplier_name LIKE '% Label%';

-- Find any employee born on the 10th day of the month
SELECT *
FROM employee
```

```
WHERE birth_day LIKE '_____10%';
```

```
-- Find any clients who are schools
SELECT *
FROM client
WHERE client_name LIKE '%Highschool%';
```

Union

```
-- Find a list of employee and branch names
SELECT employee.first_name AS Employee_Branch_Names
FROM employee
UNION
SELECT branch.branch_name
FROM branch;
```

```
-- Find a list of all clients & branch suppliers' names
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;
```

Joins

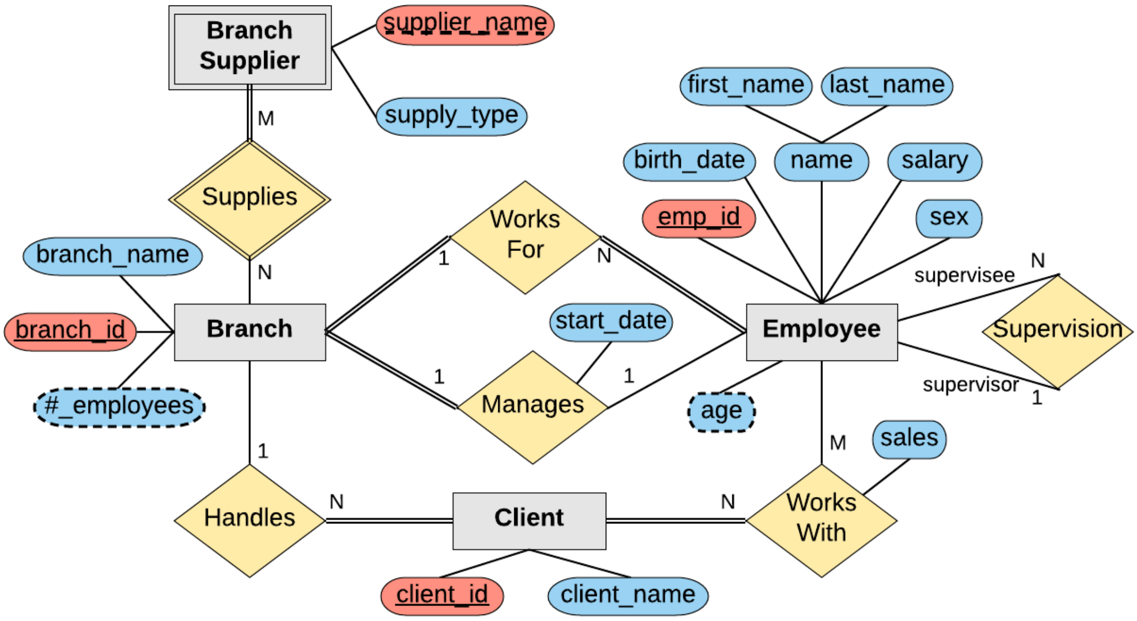
```
-- Add the extra branch

INSERT INTO branch VALUES(4, "Buffalo", NULL, NULL);


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;
```

Nested Queries

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

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);


-- Find all clients who are handles by the branch that Michael Scott manages

-- Assume you know Michael's ID

SELECT client.client_id, client.client_name

FROM client

WHERE client.branch_id = (SELECT branch.branch_id

FROM branch

WHERE branch.mgr_id = 102);


-- Find all clients who are handles by the branch that Michael Scott manages

-- Assume you DONT'T know Michael's ID
```

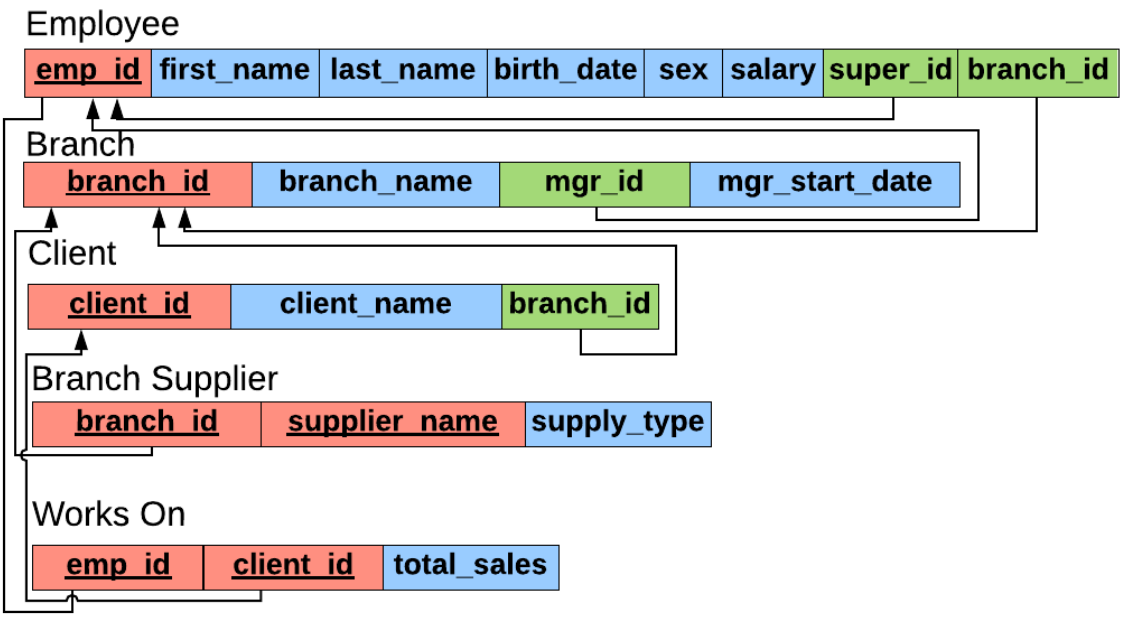
```
SELECT client.client_id, client.client_name
FROM client
WHERE client.branch_id = (SELECT branch.branch_id
FROM branch
WHERE branch.mgr_id = (SELECT employee.emp_id
FROM employee
WHERE employee.first_name = 'Michael' AND employee.last_name ='Scott'
LIMIT 1));

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

-- Find the names of all clients who have spent more than 100,000 dollars
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
);
```

Designing An Er Diagram:



Er Diagram Mapping

Company Database Schema



Employee

emp_id	first_name	last_name	birth_date	sex	salary	super_id	branch_id

Branch

branch_id	branch_name	mgr_id	mgr_start_date

Client

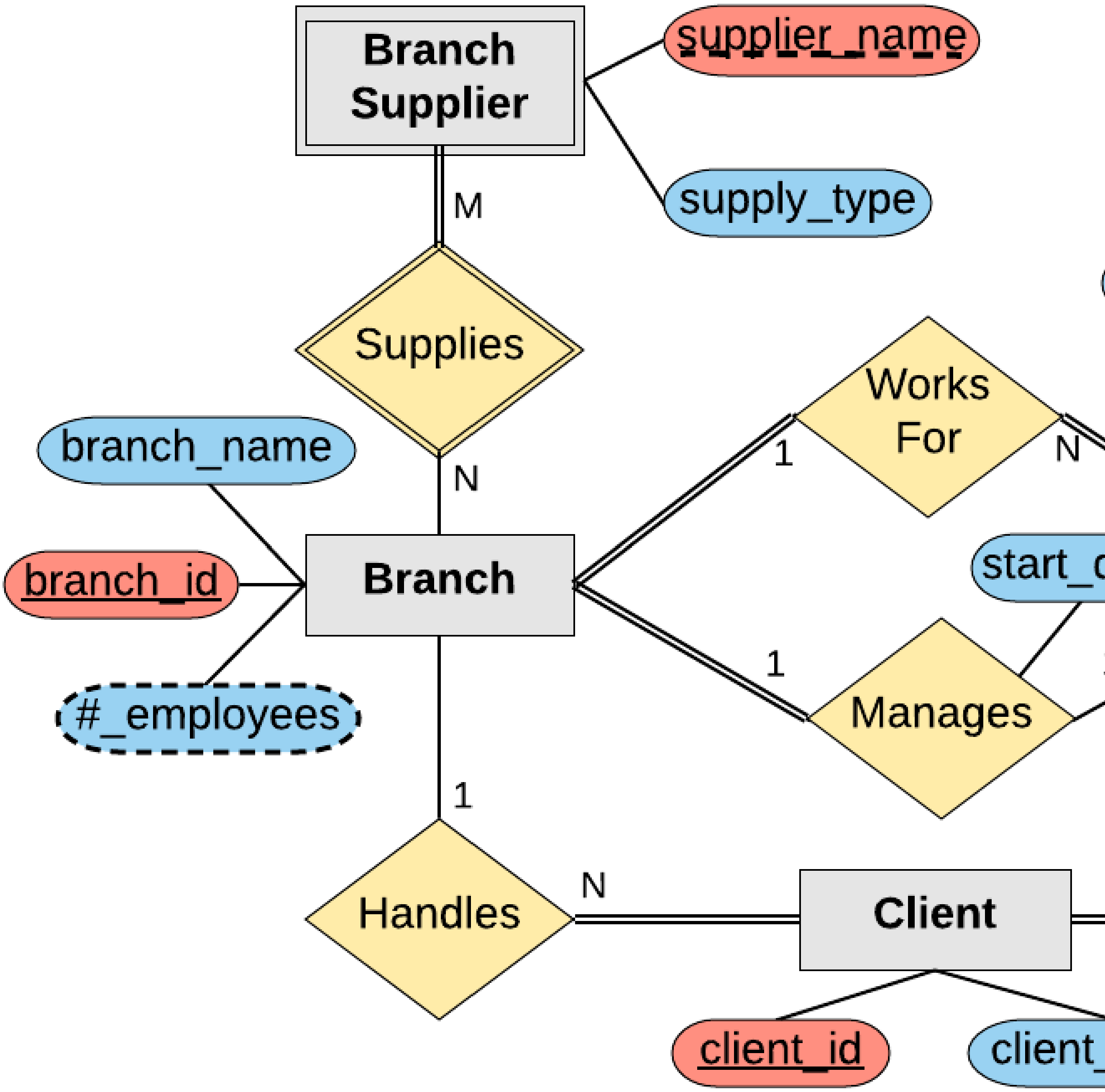
client_id	client_name	branch_id

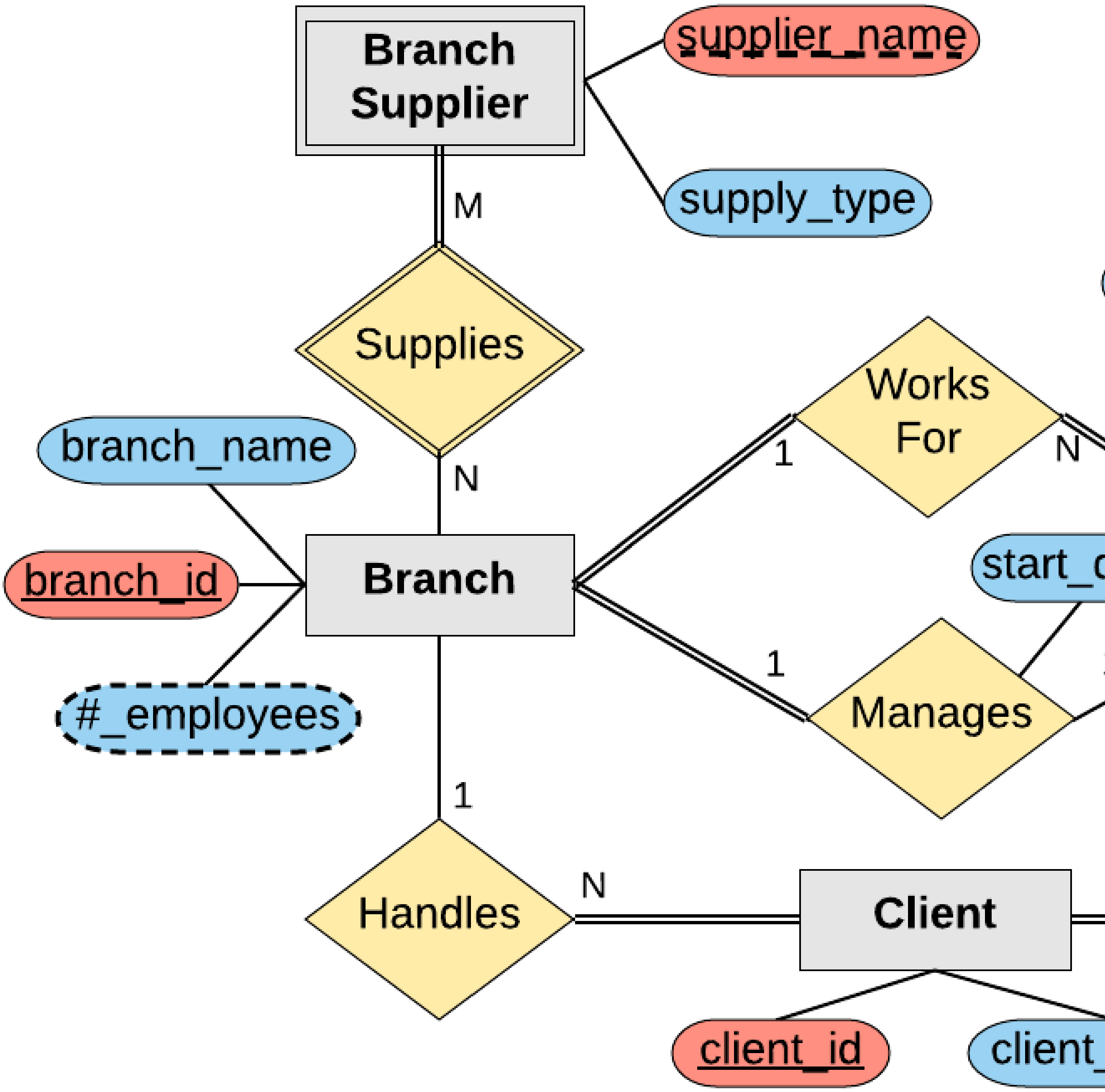
Branch Supplier

branch_id	supplier_name	supply_type

Works On

emp_id	client_id	total_sales


Branch Supplier
supplier_name
supply_type
M
Supplies
N
branch_name
branch_id
#_employees
Branch
Works For
1
N
start_d
1
Manages
1
Handles
N
Client
client_id
client_


Branch Supplier
supplier_name
supply_type
M
Supplies
N
branch_name
branch_id
Branch
#_employees
Works For
1
N
start_d
Manages
1
1
Handles
N
Client
client_id
client_