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);

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);

-- 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;

-- 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

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');

-- 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

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;

-- % = 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%';

-- 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;

-- 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;

-- 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

);

-- CREATE

-- TRIGGER `event\_name` BEFORE/AFTER INSERT/UPDATE/DELETE

-- ON `database`.`table`

-- FOR EACH ROW BEGIN

-- -- trigger body

-- -- this code is applied to every

-- -- inserted/updated/deleted row

-- END;

CREATE TABLE trigger\_test (

message VARCHAR(100)

);

DELIMITER $$

CREATE

TRIGGER my\_trigger BEFORE INSERT

ON employee

FOR EACH ROW BEGIN

INSERT INTO trigger\_test VALUES('added new employee');

END$$

DELIMITER ;

INSERT INTO employee

VALUES(109, 'Oscar', 'Martinez', '1968-02-19', 'M', 69000, 106, 3);

DELIMITER $$

CREATE

TRIGGER my\_trigger BEFORE INSERT

ON employee

FOR EACH ROW BEGIN

INSERT INTO trigger\_test VALUES(NEW.first\_name);

END$$

DELIMITER ;

INSERT INTO employee

VALUES(110, 'Kevin', 'Malone', '1978-02-19', 'M', 69000, 106, 3);

DELIMITER $$

CREATE

TRIGGER my\_trigger BEFORE INSERT

ON employee

FOR EACH ROW BEGIN

IF NEW.sex = 'M' THEN

INSERT INTO trigger\_test VALUES('added male employee');

ELSEIF NEW.sex = 'F' THEN

INSERT INTO trigger\_test VALUES('added female');

ELSE

INSERT INTO trigger\_test VALUES('added other employee');

END IF;

END$$

DELIMITER ;

INSERT INTO employee

VALUES(111, 'Pam', 'Beesly', '1988-02-19', 'F', 69000, 106, 3);

DROP TRIGGER my\_trigger;