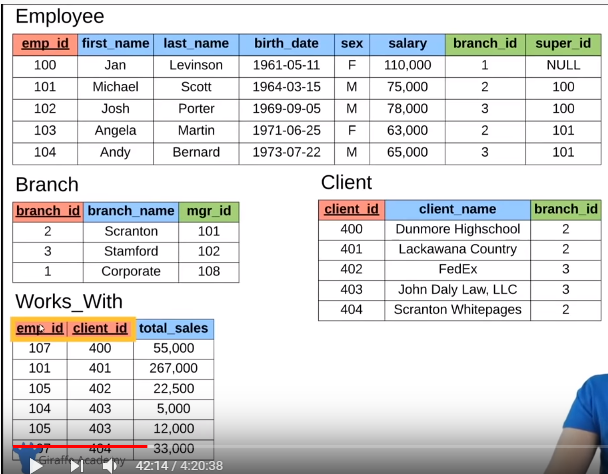
PrimaryKey Foreingn Keys

**branch\_id** is PrimaryKey in **Branch-table**, **super\_id** refers to another employee **inside the Employee-table**



branch\_id is a PrimaryKey in Branch-table, mgr\_id is a ForeignKey and refers to Employee-table

client\_id is a PrimaryKey for Client-table, branch\_id is a ForeignKey and refers to Branch-table

emp\_id + client\_id is a Composite PrimaryKey in Works\_With-table and emp\_id is a ForeignKey referring to Employee-table and client\_id is a ForeignKey referring to Client-table

CREATE TABLE student

DROP TABLE student

ALTER TABLE student ADD gpa DECIMAL(3,2);

ALTER TABLE student DROP COLUMN gpa;

INSERT INTO student(student\_id, name) VALUES(3, "Claire");

\*\*\*

CREATE TABLE student(

student\_id INT PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(20),

major VARCHAR(20) DEFAULT "undecided");

INSERT INTO student(name, major) VALUES("Jack", "Biology");

INSERT INTO student(name, major) VALUES("Kate", "Sociology");

INSERT INTO student(name, major) VALUES("Claire", "Chemistry");

INSERT INTO student(name, major) VALUES("Jack", "Biology");

INSERT INTO student(name) VALUES("Mike");

| **student\_id** | **name** | **major** |
| --- | --- | --- |
| 1 | Jack | Biology |
| 2 | Kate | Sociology |
| 3 | Claire | Chemistry |
| 4 | Jack | Biology |
| 5 | Mike | undecided |

UPDATE student

SET major=”Bio”

WHERE major=”Biology” OR major=”Chemistry”;

SET major=”Bio”, name=”Tom”

WHERE student\_id=2

DELETE FROM student

WHERE student\_id=5;

SELECT \*

FROM table

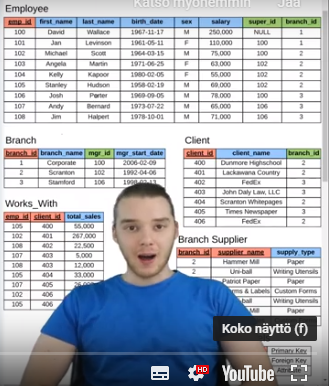
WHERE major=”Bio” OR major=”Chemistry”

ORDER BY column, column DESC

LIMIT 3;

**CREATING A COMPANY DATABASE,**

**relational db with few tables and suitable/necessary keys**



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);

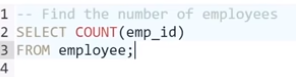
INSERT INTO works\_with VALUES(102, 406, 15000);

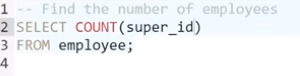
INSERT INTO works\_with VALUES(105, 406, 130000);





**FUNCTIONS**

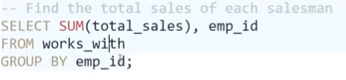
9 employees all together, everyone has emp:id

8 empl. have supervisors, as for 1 super\_id is null



SELECT SUM(salary)

SELECT COUNT(sex), sex FROM employee GROUP BY sex; AGGREGATION



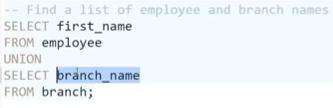
**WILDCARDS**





4 hyphens for any year and – and 10 and for October

**UNIONS**

gives 1 column list of all firstnames and branch names. UNION must select the same amount of columns from each table + they must have the same data type

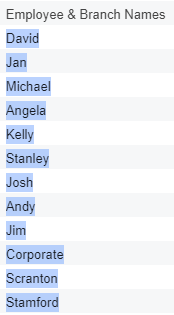
SELECT first\_name AS "Employee & Branch Names"

FROM employee

UNION

SELECT branch\_name AS "Employee & Branch Names"

FROM branch;



**JOINS**

 INNER JOIN gives the branches WHERE MGR\_ID IS NOT NULL!!! **HOW DO I GET ALL BRANCHES WHETHER they have manager filled in or not**

This is equal to the above:

SELECT a.branch\_id AS "BranchID", a.branch\_name AS "Branch Name",

b.emp\_id AS "Manager's EmplID", b.first\_name AS "Manager's Name"

FROM branch a, employee b

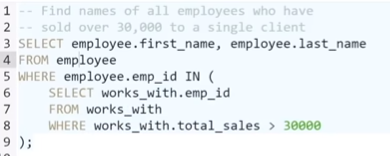
WHERE a.mgr\_id = b.emp\_id;

LEFT JOIN gives ALL the employees

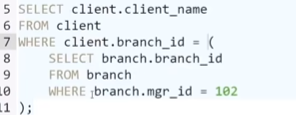
RIGHT JOIN gives ALL branches

FULL OUTER JOIN can’t be used in MySQL but it would give all employees and branches regardless whether the emp\_id or mgr\_id is NULL

**NESTED QUERIES**



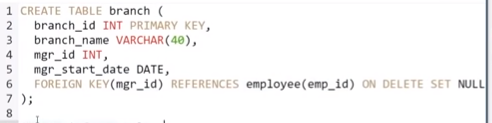
Assume you know Michael’s emp\_id and find out the clients that belong to his branch (he has only 1 branch! if he can have several use IN)



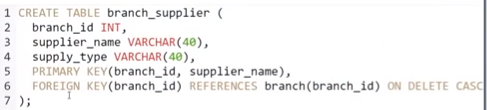
**ON DELETE**

When creating a table referencing to another “main” table think about what happens in this related table when you DELETE a record from the “main” table!

With ON DELETE SET NULL in the related table, the rows/records stay there, only the value referring to the “main” table is deleted meaning SET TO NULL. This IS COOL when the deleted value is only acting as FOREIGN KEY in this related table!



With ON DELETE CASCADE the entire rows/records are deleted from this related table. This NEEDS TO BE because the deleted value is part of the PRIMARY KEY in this related table, primary key can not be set to null!!



**TRIGGERS**

Can’t be done in POPSQL so we’ll go to a cmd

s1900570@BC-H8FV7S2:/var/www/html$ cd giraffe

s1900570@BC-H8FV7S2:/var/www/html/giraffe$ mysql -u root -p mysql

Enter password:

MariaDB [mysql]> use giraffe

MariaDB [giraffe]> CREATE TABLE trigger\_test (

-> message VARCHAR(100)

-> );

Query OK, 0 rows affected (0.039 sec)

MariaDB [giraffe]> DELIMITER $$ \*changing the delimiter from ; to $$

MariaDB [giraffe]> CREATE

-> TRIGGER my\_trigger BEFORE INSERT

-> ON employee

-> FOR EACH ROW BEGIN

-> INSERT INTO trigger\_test VALUES("added new employee");

-> END$$

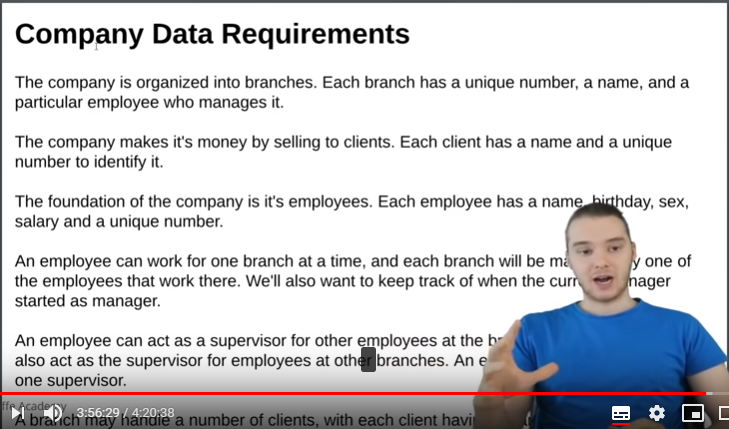
Query OK, 0 rows affected (0.063 sec)

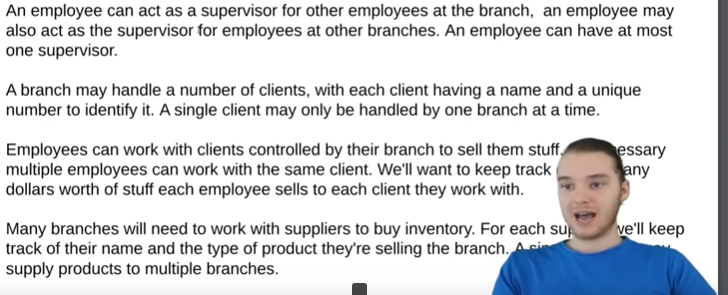
MariaDB [giraffe]> DELIMITER ; \*changing the delimiter back to ;

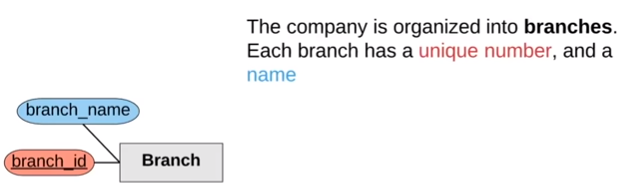
MariaDB [giraffe]>

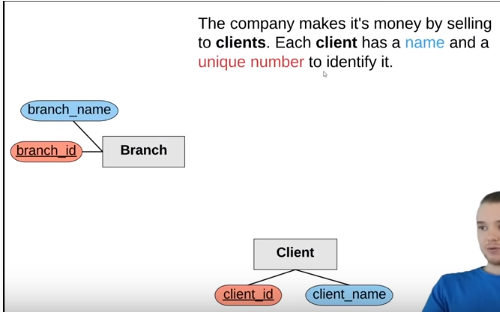
Tästä välistä hävis muistiinpanot ☹ en muista mitä kaikkea ☹

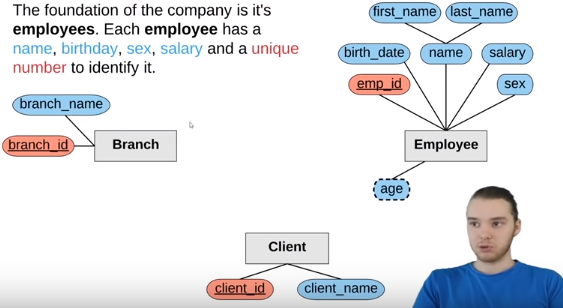
**Requirements -> ER Diagram -> Database Schema**



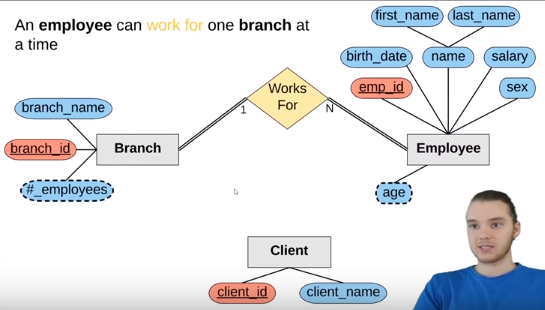




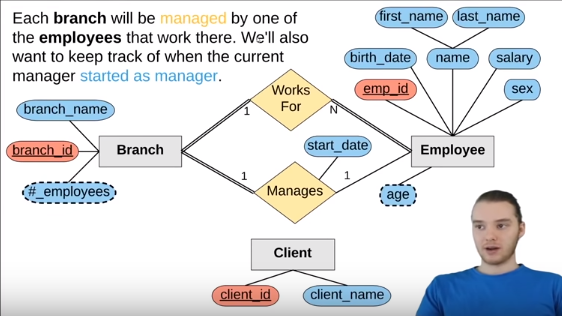




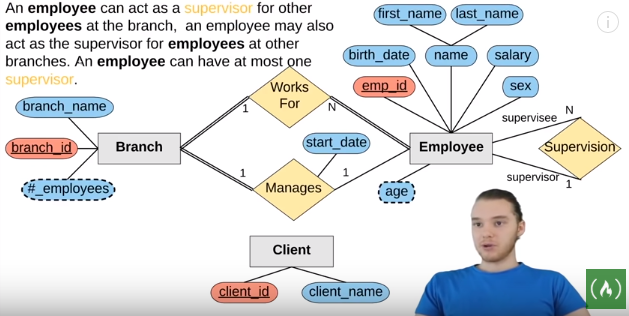
age is a DERIVED value



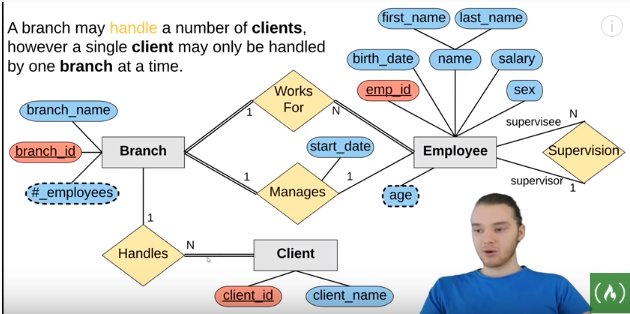
Full participation : every branch must have an employee and every employee must have a branch (double line both ways.Branch can have ANY number of employees (N), employee can work for only 1 branch (1)



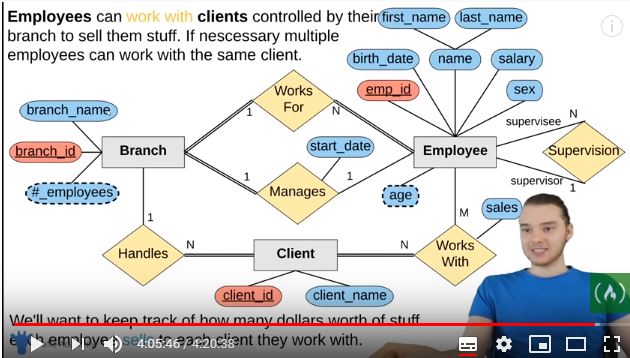
Each branch MUST be managed by an employee (full participation, double line), an employee CAN manage a branch



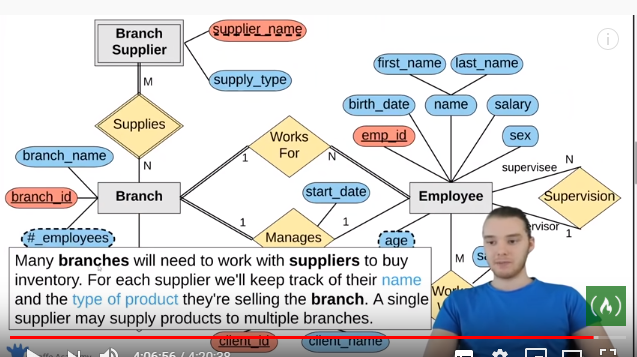
An employee can be a supervisor for another employee, an employee can be a supervisee for another employer, an employee can supervise any number of supervisees, but a supervisee can be supervised only by 1 supervisor



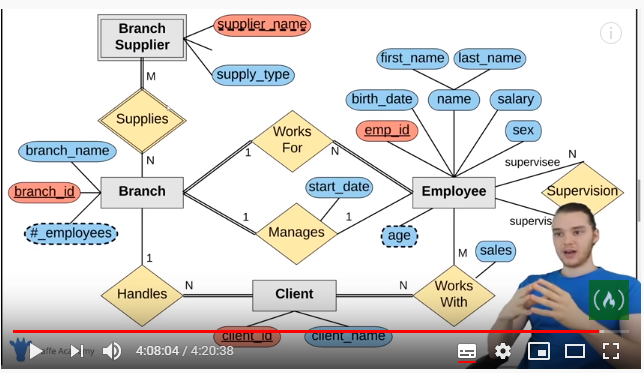
every client must be handled by a branch, every branch does not have to have clients (sisäiset palvelut), and branch can have any number of clients, client can be handled by 1 branch only



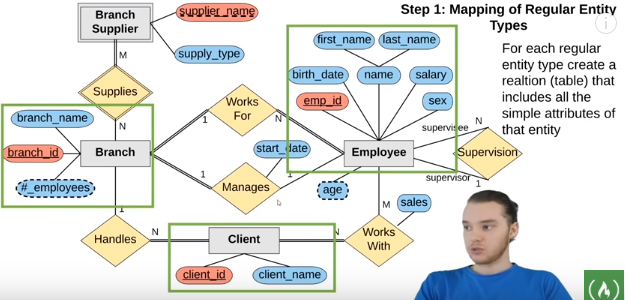
all clients must work with an employee but all employess do not have to work with clients, a client can work with any number of employees and an employee can work with any number of clients

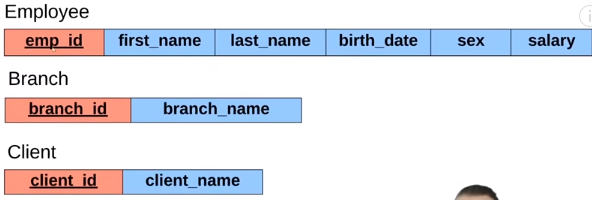


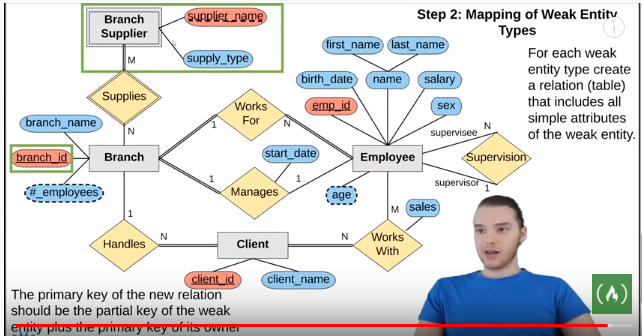
Branch Supplier – weak entity?!

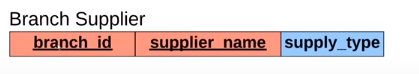


**ER Diagram -> Database Schema**

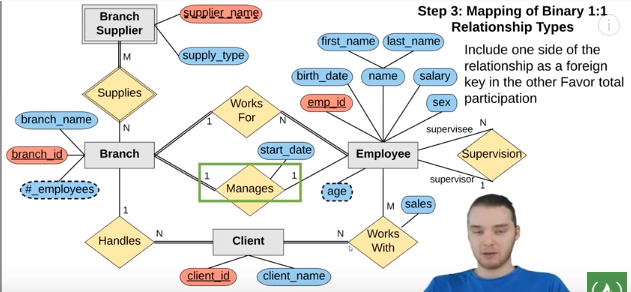


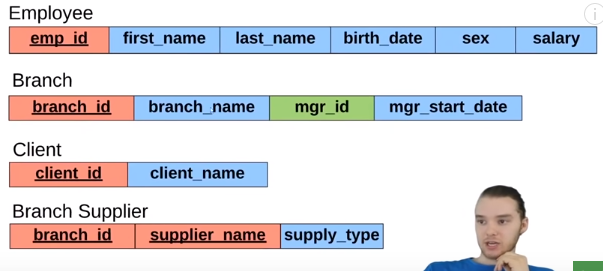




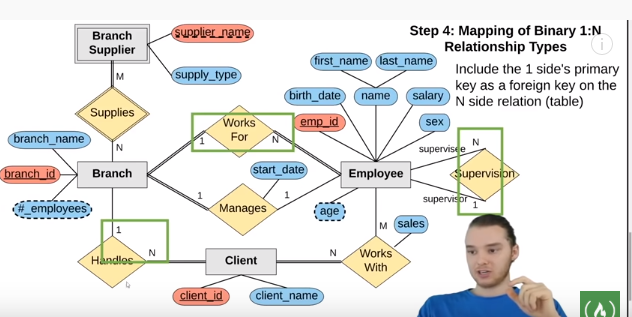


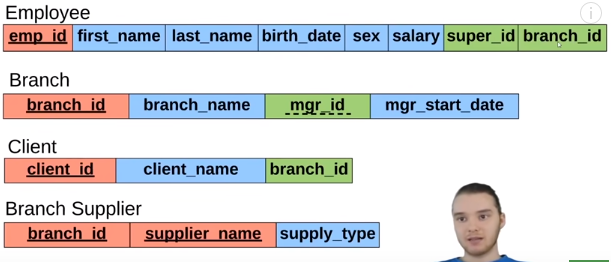
Composite key



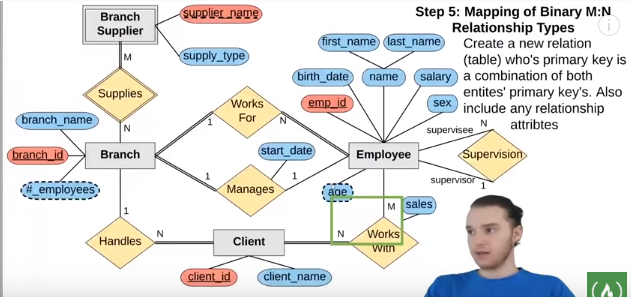


Foreign key : Branch.mgr\_id = Employee.emp\_id

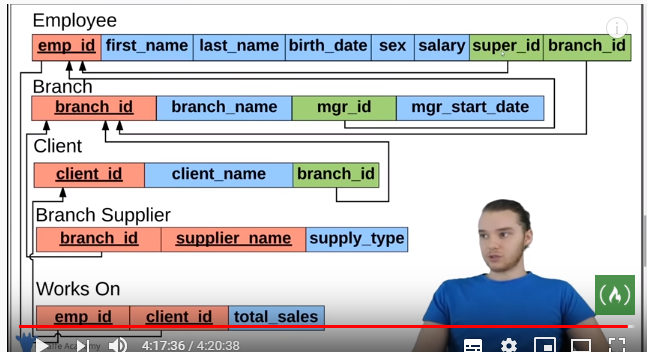


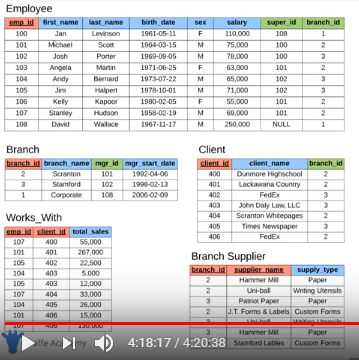


Foreign key is added to the 1 side of the relationship : employee can have only 1 Super and only 1 branch where he works. Client is handled by only 1 branch



New table : Works On with compsoite key Employee.emp\_id + Client.client\_id





**Code for Creating the 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);

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);