

What is a database?



What is a Database (DB)?

- Any collection of related information
 - Phone Book
 - Shopping list
 - Todo list
 - Your 5 best friends
 - Facebook's User Base
- Databases can be stored in different ways
 - On paper
 - In your mind
 - On a computer
 - This powerpoint
 - Comments Section



Computers + Databases = <3

Amazon.com

vs

Shopping List

- Keeps track of Products, Reviews, Purchase Orders, Credit Cards, Users, Media, etc
 - Trillions of pieces of information need to be stored and readily available
 - Information is extremely valuable and critical to Amazon.com's functioning
 - Security is essential, Amazon stores peoples personal information
 - Credit card #, SSN, Address, phone
 - Information is stored on a computer
- Keeps track of consumer products that need to be purchased
 - 10-20 pieces of information need to be stored and readily available
 - Information is for convenience sake only and not necessary for shopping
 - Security is not important
 - Information is stored on a piece of paper, or even just in someone's memory



Giraffe Academy

Computers are great at keeping track of large amounts of information



Database Management Systems (DBMS)

- A special software program that helps users create and maintain a database
 - Makes it easy to manage large amounts of information
 - Handles Security
 - Backups
 - Importing/exporting data
 - Concurrency
 - Interacts with software applications

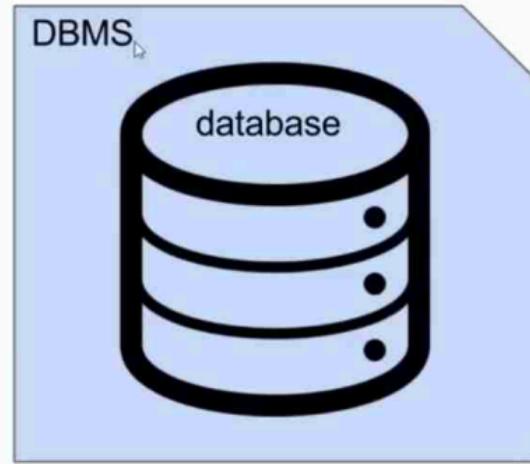
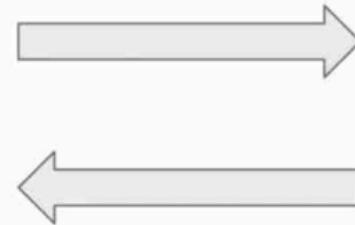


Giraffe Academy

■ Programming Languages



Amazon.com Database Diagram



Amazon.com will interact with the DBMS in order to create, read, update and delete information



Giraffe Academy

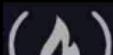


C.R.U.D

Create Read Update Delete



Giraffe Academy



SUBSCRIBE

Two Types of Databases

Relational Databases (SQL)

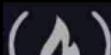
- Organize data into one or more tables
 - Each table has columns and rows
 - A unique key identifies each row

Non-Relational (noSQL / not just SQL)

- Organize data is anything but a traditional table
 - Key-value stores
 - Documents (JSON, XML, etc)
 - Graphs
 - Flexible Tables



Giraffe Academy



SUBSCRIBE

Relational Databases (SQL)



Student Table

*ID #	Name	Major
1	Jack	Biology
2	Kate	Sociology
3	Claire	English
4	John	Chemistry

Users Table

*Username	Password	Email
jsmith22	wordpass	...
catlover45	apple223	...
gamerkid
giraffe



Giraffe Academy



Relational Databases (SQL)

- Relational Database Management Systems (RDBMS)
 - Help users create and maintain a relational database
 - mySQL, Oracle, postgreSQL, mariaDB, etc.
- Structured Query Language (SQL)
 - Standardized language for interacting with RDBMS
 - Used to perform C.R.U.D operations, as well as other administrative tasks (user management, security, backup, etc).
 - Used to define tables and structures
 - SQL code used on one RDBMS is not always portable to another without modification.



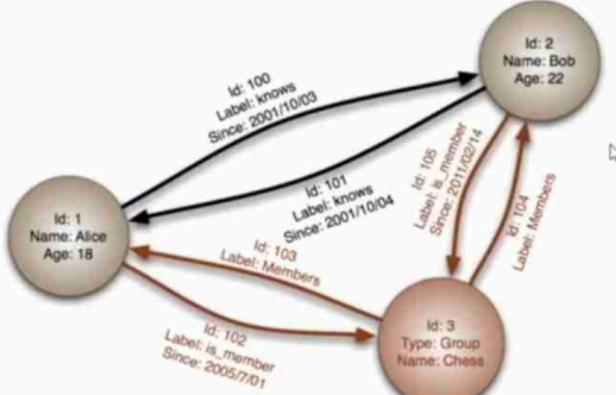
Non-Relational Databases (noSQL / not just SQL)

```
[{  
    "_id": 1345,  
    "name": "Jack",  
    "major": "Biology"  
}, {  
    "_id": 2267,  
    "name": "Kate",  
    "major": "Sociology"  
}, {  
    "_id": 2453,  
    "name": "Claire",  
    "major": "English"  
}, {  
    "_id": 1957,  
    "name": "John",  
    "major": "Chemistry"  
}]
```

Document



JSON, BLOB, XML, etc..)
Giraffe Academy



Graph

Relational nodes

Key	Value
"xyz"	string
"abc"	JSON
"pqr"	BLOB
"lmno"	etc...

Key-Value Hash

Keys are mapped to values
(strings, json, blob, etc..)



SUBSCRIBE

Database Queries

Queries are requests made to the database management system for specific information

As the database's structure become more and more complex, it becomes more difficult to get the specific pieces of information we want.

A google search is a query



Giraffe Academy



Wrap Up

- Database is any collection of related information
- Computers are great for storing databases
- Database Management Systems (DBMS) make it easy to create, maintain and secure a database.
- DBMS allow you to perform the C.R.U.D operations and other administrative tasks
- Two types of Databases, Relational & Non-Relational
- Relational databases use SQL and store data in tables with rows and columns
- Non-Relational data store data using other data structures

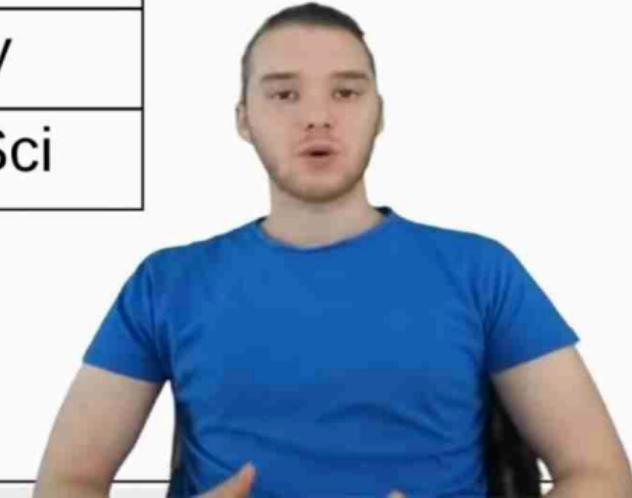


Tables and Keys



Student

<u>student id</u>	name	major
1	Jack	Biology
2	Kate	Sociology
3	Claire	English
4	Jack	Biology
5	Mike	Comp. Sci



Giraffe Academy



Student

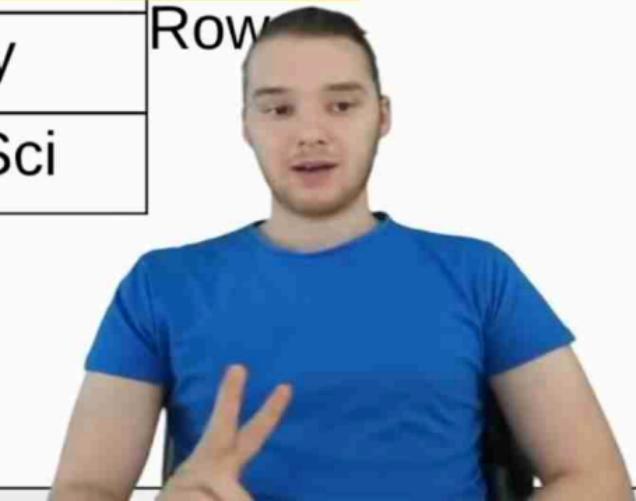
Column

<u>student id</u>	name	major
1	Jack	Biology
2	Kate	Sociology
3	Claire	English
4	Jack	Biology
5	Mike	Comp. Sci



Giraffe Academy

Row



SUBSCRIBE

Student

<u>student_id</u>	name	major
1	Kate	Sociology
2	Jack	Biology
3	Claire	English
4	Jack	Biology
5	Mike	Comp. Sci



Giraffe Academy



User

<u>email</u>	password	date_created	Type
fakemail@fake.co	shivers1	1999-05-11	Admin
fakemail112@fake.co	wordpass	2001-03-15	Free
rsmith@fake.co	redRoad23	2010-09-05	Free
jdoe@fake.co	passw0rd	2008-06-25	Premium
jhalpert@fake.co	557df32d	2003-07-22	Free



Giraffe Academy



Employee

<u>emp_id</u>	first_name	last_name	birth_date	sex	salary
100	Jan	Levinson	1961-05-11	F	110,000
101	Michael	Scott	1964-03-15	M	75,000
102	Josh	Porter	1969-09-05	M	78,000
103	Angela	Martin	1971-06-25	F	63,000
104	Andy	Bernard	1973-07-22	M	65,000



Employee

<u>emp_ssn</u>	first_name	last_name	birth_date	sex	salary
123456789	Jan	Levinson	1961-05-11	F	110,000
555667777	Michael	Scott	1964-03-15	M	75,000
8886665555	Josh	Porter	1969-09-05	M	78,000
111332467	Angela	Martin	1971-06-25	F	63,000
99857463	Andy	Bernard	1973-07-22	M	65,000



Giraffe Academy



SUBSCRIBE

Employee

emp_id	first_name	last_name	birth_date	sex	salary	branch_id
100	Jan	Levinson	1961-05-11	F	110,000	1
101	Michael	Scott	1964-03-15	M	75,000	2
102	Josh	Porter	1969-09-05	M	78,000	3
103	Angela	Martin	1971-06-25	F	63,000	2
104	Andy	Bernard	1973-07-22	M	65,000	3

Employee

emp_id	first_name	last_name	birth_date	sex	salary	branch_id
100	Jan	Levinson	1961-05-11	F	110,000	1
101	Michael	Scott	1964-03-15	M	75,000	2
102	Josh	Porter	1969-09-05	M	78,000	3
103	Angela	Martin	1971-06-25	F	63,000	2
104	Andy	Bernard	1973-07-22	M	65,000	3

Branch

branch_id	branch_name	mgr_id
2	Scranton	101
3	Stamford	102
1	Corporate	108



Giraffe Academy



Employee

emp_id	first_name	last_name	birth_date	sex	salary	branch_id	super_id
100	Jan	Levinson	1961-05-11	F	110,000	1	NULL
101	Michael	Scott	1964-03-15	M	75,000	2	100
102	Josh	Porter	1969-09-05	M	78,000	3	100
103	Angela	Martin	1971-06-25	F	63,000	2	101
104	Andy	Bernard	1973-07-22	M	65,000	3	101

Branch

branch_id	branch_name	mgr_id
2	Scranton	101
3	Stamford	102
1	Corporate	108



Giraffe Academy



SUBSCRIBE

Employee

emp_id	first_name	last_name	birth_date	sex	salary	branch_id	super_id
100	Jan	Levinson	1961-05-11	F	110,000	1	NULL
101	Michael	Scott	1964-03-15	M	75,000	2	100
102	Josh	Porter	1969-09-05	M	78,000	3	100
103	Angela	Martin	1971-06-25	F	63,000	2	101
104	Andy	Bernard	1973-07-22	M	65,000	3	101

Branch

branch_id	branch_name	mgr_id
2	Scranton	101
3	Stamford	102
1	Corporate	108

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 Fo



Giraffe Academy



SUBSCRIBE

Employee

emp_id	first_name	last_name	birth_date	sex	salary	branch_id	super_id
100	Jan	Levinson	1961-05-11	F	110,000	1	NULL
101	Michael	Scott	1964-03-15	M	75,000	2	100
102	Josh	Porter	1969-09-05	M	78,000	3	100
103	Angela	Martin	1971-06-25	F	63,000	2	101
104	Andy	Bernard	1973-07-22	M	65,000	3	101

Branch

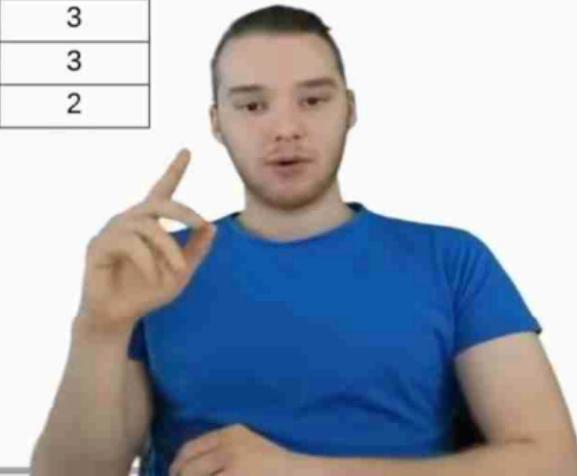
branch_id	branch_name	mgr_id
2	Scranton	101
3	Stamford	102
1	Corporate	108

Works_With

emp_id	client_id	total_sales
107	400	55,000
101	401	267,000
105	402	22,500
104	403	5,000
105	403	12,000
107	404	33,000

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



SQL Basics



Structured Query Language (SQL)

- SQL is a language used for interacting with Relational Database Management Systems (RDBMS)
 - You can use SQL to get the RDBMS to do things for you
 - Create, retrieve, update & delete data
 - Create & manage databases
 - Design & create database tables
 - Perform administration tasks (security, user management, import/export, etc)



Giraffe Academy



Structured Query Language (SQL)

- SQL implementations vary between systems
 - Not all RDBMS' follow the SQL standard to a 'T'
 - The concepts are the same but the implementation may vary



Giraffe Academy



Structured Query Language (SQL)

- SQL is actually a hybrid language, it's basically 4 types of languages in one
 - **Data Query Language (DQL)**
 - Used to query the database for information.
 - Get information that is already stored there
 - **Data Definition Language (DDL)**
 - Used for defining database schemas.
 - **Data Control Language (DCL)**
 - Used for controlling access to the data in the database.
 - User & permissions management
 - **Data Manipulation Language (DML)**
 - Used for inserting, updating and deleting data from the database



Queries

- A query is a set of instructions given to the RDBMS (written in SQL) that tell the RDBMS what information you want it to retrieve for you
 - TONS of data in a DB
 - Often hidden in a complex schema
 - Goal is to only get the data you need

```
1  SELECT employee.name, employee.age  
2  FROM employee  
3  WHERE employee.salary > 3000
```



Creating Tables



PopSQL

File Edit View Query Window

Giraffe

Feedback Q

Queries Ctrl+Q Schema Ctrl+S

Untitled query +

Search... New Query

Private queries Untitled query

Untitled query

Run

```
1 INT
2 DECIMAL(M,N)
3 VARCHAR(1)
4 BLOB
5 DATE
6 TIMESTAMP
7 |
```

-- Whole Numbers
-- Decimal Numbers - Exact Value
-- String of text of length 1
-- Binary Large Object, Stores large data
-- 'YYYY-MM-DD'
-- 'YYYY-MM-DD HH:MM:SS' - used for recording



The results of your query will appear here.



Giraffe Academy



Search... New Query

Untitled query

Private queries Untitled query

1 CREAT

Database Course - Tables & Keys.jpg - Photos

Run

student_id name major

student_id	name	major
1	Jack	Biology
2	Kate	Sociology
3	Claire	English
4	Jack	Biology
5	Mike	Comp. Sci

This results of your query will appear here



New Query Untitled query Run

Untitled query

Private query Untitled query

```
1 CREATE TABLE student (
2     student_id INT PRIMARY KEY,
3     name VARCHAR(20),
4     major VARCHAR(20)
5 );
```

I

Success

Rows affected: 0

Explore SQL Data Chart Export ▾ 12:09 PM 0.035 seconds



New Query + Untitled query Run

Untitled query

Private quer Untitled q

```
1 CREATE TABLE student (
2     student_id INT,
3     name VARCHAR(20),
4     major VARCHAR(20),
5     PRIMARY KEY(student_id)
6 );
```

Success

Rows affected: 0

12:09 PM 0.035 seconds



Inserting Data



C



```
5     PRIMARY KEY(student_id)
6 );
7 SELECT * FROM student;
8
9 INSERT INTO student VALUES(1, 'Jack', 'Biology');
10 INSERT INTO student VALUES(2, 'Kate', 'Sociology');
11 INSERT INTO student(student_id, name) VALUES(3, 'Claire');
12 INSERT INTO student VALUES(4, 'Jack', 'Biology');
13 INSERT INTO student VALUES(5, 'Mike', 'Computer Science');
14
15
16
```

Success

5 rows

Explore SQL Data Chart Export

12:57 PM

0.065 seconds

student_id	name	major
1	Jack	Biology
2	Kate	Sociology
3	Claire	NULL
4	Jack	Biology
5	Mike	Computer Science



Untitled query

N

Untitled query

Run

```
2
3 CREATE TABLE student (
4     student_id INT,
5     name VARCHAR(20) NOT NULL,
6     major VARCHAR(20) UNIQUE,
7     PRIMARY KEY(student_id)
8 );
9 SELECT * FROM student;
10
11 INSERT INTO student VALUES(1, 'Jack', 'Biology');
```

Success

Rows affected: 0

[Explore](#) [SQL](#) [Data](#) [Check](#) [Export](#) [%](#)

2:03 PM

0.101 seconds



Untitled query

Untitled query

Run

```
1     PRIMARY KEY(student_id)
2 );
3 SELECT * FROM student;
4
5 INSERT INTO student(student_id, name) VALUES(1, 'Jack');
6 INSERT INTO student VALUES(2, 'Kate', 'Sociology');
7 INSERT INTO student VALUES(3, NULL, "Chemistry");
8 INSERT INTO student VALUES(4, 'Jack', 'Biology');
9 INSERT INTO student VALUES(5, 'Mike', 'Computer Science');
```

Success

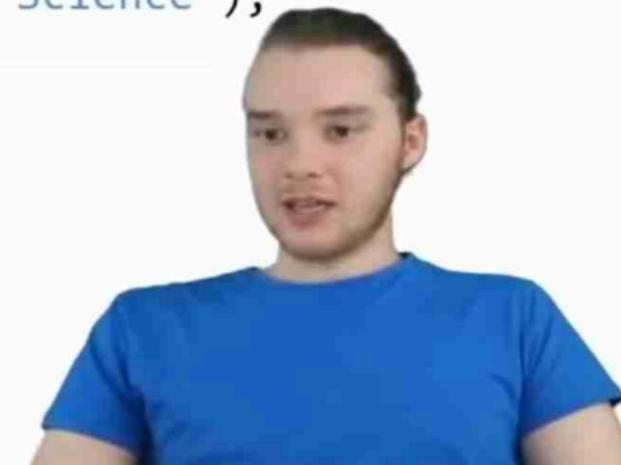


Rows affected: 1

Execute SQL Data Chart Export ?

2:09 PM

0.045 seconds



Update & Delete



other comparision ops:

= : equals

<> : not equals

3> UPDATE student
4 SET major = 'Bio' > : greater than

5 WHERE major = 'Biology'; < : less than

>= : greater than or equal

<= : less than or equal

```
1 SELECT * FROM student;  
2  
3> UPDATE student  
4 SET major = 'Bio'  
5 WHERE major = 'Biology';  
6  
7  
8  
9  
10
```

Success
5 rows

student_id	name	major
1	Jack	Biology
2	Kate	Sociology
3	Claire	Chemistry
4	Jack	Biology
5	Mike	Computer Science



Giraffe Academy



Untitled query

Run

```
1 SELECT * FROM student;
2
3 UPDATE student
4 SET major = 'Biochemistry'
5 WHERE major = 'Bio' OR major = 'Chemistry';
```

6

7

8

9

10

Success

5 rows

Explore SQL Data Chart Export %

1.27 MB

0.05 seconds

student_id	name	major
1	Jack	Bio
2	Kate	Sociology
3	Claire	Chemistry
4	Jack	Comp Sci
5	Mike	Comp Sci



Giraffe Academy



Basic Queries



PopSQL

File Edit View Query Window

Giraffe

Untitled query

Run

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

Success
5 rows

name	major
Jack	Biology
Kate	Sociology
Claire	Chemistry
Jack	Biology
Mike	Computer Science

Explore SQL Data Chart Export % 2:22 PM 0.038 seconds

Giraffe Academy

(🔥)
SUBSCRIBE

Company Database Into



Database Course - Page 6.pdf

1 / 2

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

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

Works_With

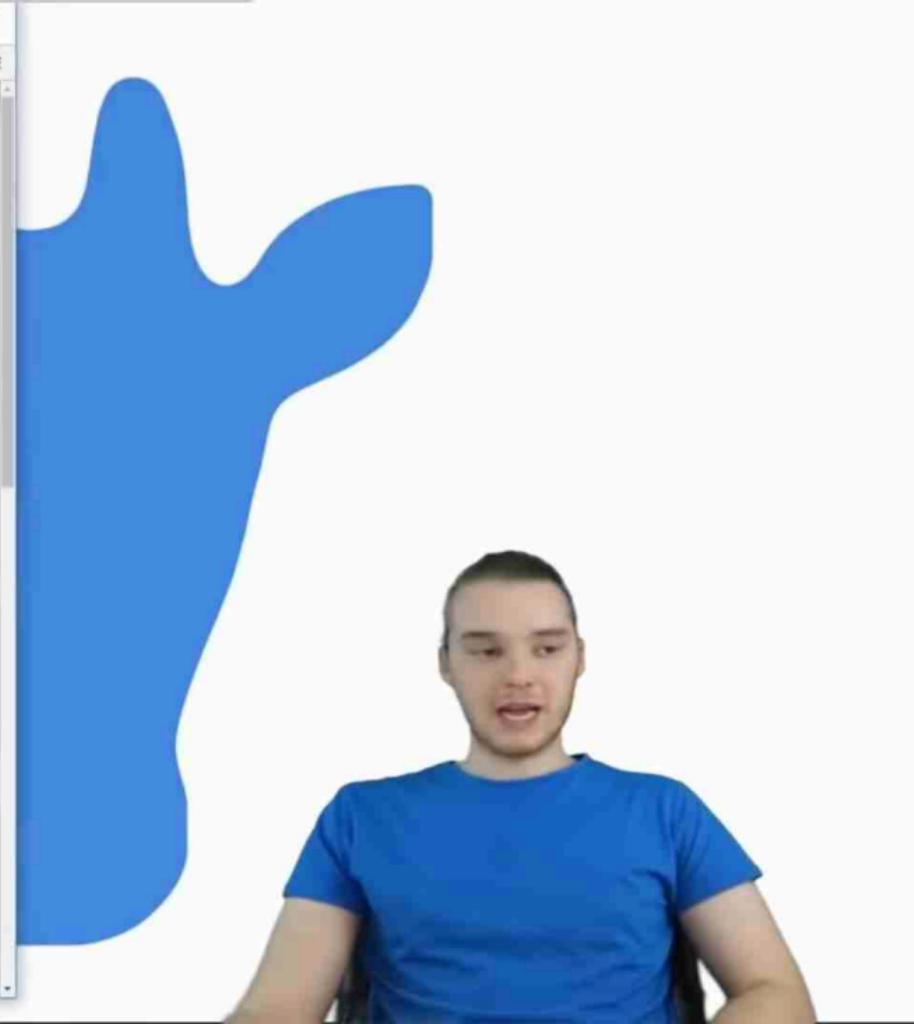
emp_id	client_id	total_sales
105	400	55,000
102	401	267,000
108	402	22,500
107	403	5,000
108	403	12,000
105	404	33,000
107	405	26,000
102	406	15,000
105	406	130,000

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

Giraffe Academy

I labels



Wildcards





— Congratulations, mdakram09! —

This certificate is awarded to

Md Akram Khan

for your accomplishments in Hash Code 2021

Qualifications



7548*

hash code

Google

Coding Competitions

On Delete



PopSQL

File Edit View Query Window

Giraffe

Feedback: Q

Untitled query + Run

```
1 CREATE TABLE branch (
2     branch_id INT PRIMARY KEY,
3     branch_name VARCHAR(40),
4     mgr_id INT,
5     mgr_start_date DATE,
6     FOREIGN KEY(mgr_id) REFERENCES employee(emp_id) ON DELETE SET NULL
7 );
```

Success 9 rows

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	22500
108	403	12000

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

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

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
105	406	12,000

Branch Supplier

branch_id	supplier_name	supply_type
2	Hammer Mill	Paper
2	Uni-ball	Writing Utensils
2	Patriot Paper	Paper
2	Custom Forms & Labels	Custom Forms
2	Uni-ball	Writing Utensils
2	Patriot Paper	Paper
2	Custom Forms & Labels	Custom Forms

Primary Key
Foreign Key
Composite

(🔥)
SUBSCRIBE

Triggers



PopSQL

File Edit View Query Window

Giraffe * Feedback

Untitled query +

1 mysql -u root -p

Run

Success

Rows affected: 0

Output SQL Data Clear Export %

99.99% 0.157 seconds

```
mysql>
mysql> |
```



Giraffe Academy

Windows



SUBSCRIBE

PopSQL

File Edit View Query Window

Giraffe

Untitled query

Untitled query

Run

```
1 DELIMITER $$  
2 CREATE  
3 TRIGGER my_trigger BEFORE INSERT  
4 ON employee  
5 FOR EACH ROW BEGIN  
6     INSERT INTO trigger_test VALUES('added new employee');  
7 END$$  
8 DELIMITER ;
```

SUCCESS

Rows affected: 0

Explore SQL Data Client Export %

9:45 PM
0.157 seconds



Giraffe Academy



SUBSCRIBE

PopSQL

File Edit View Query Window

Giraffe

Untitled query

Untitled query

Run

```
1 DELIMITER $$  
2 CREATE  
3     TRIGGER my_trigger BEFORE INSERT  
4     ON employee  
5     FOR EACH ROW BEGIN  
6         INSERT INTO trigger_test VALUES('added new employee');  
7     END$$  
8 DELIMITER ;  
9  
10  
11  
12  
13
```

SUCCESS

Rows affected: 1

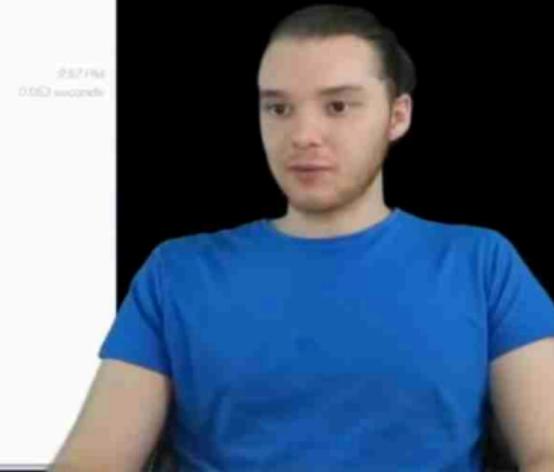
0.000 seconds

mysql> DELIMITER \$\$
mysql> 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 (0.01 sec)

mysql>



Giraffe Academy



Windows



SUBSCRIBE

PopSQL

File Edit View Query Window

Giraffe *

Untitled query +

Untitled query

Run

```
1 DELIMITER $$  
2 CREATE  
3     TRIGGER my_trigger BEFORE INSERT  
4     ON employee  
5     FOR EACH ROW BEGIN  
6         INSERT INTO trigger_test VALUES('added new employee');  
7     END$$  
8 DELIMITER ;  
9  
10  
11  
12  
13
```

SUCCESS

Rows affected: 1

MySQL 5.7 Command Line Client

```
mysql> DELIMITER $$  
mysql> CREATE TRIGGER my_trigger BEFORE INSERT ON employee FOR EACH ROW BEGIN  
    INSERT INTO trigger_test VALUES('added new employee');  
END$$  
Query OK, 0 rows affected (0.01 sec)  
  
mysql> DELIMITER ;  
mysql>
```



Giraffe Academy



Windows

PopSQL

File Edit View Query Window

Giraffe

Untitled query +

Untitled query

Run

```
1 DELIMITER $$  
2 CREATE  
3     TRIGGER my_trigger BEFORE INSERT  
4     ON employee  
5     FOR EACH ROW BEGIN  
6         INSERT INTO trigger_test VALUES('added new employee');  
7     END$$  
8 DELIMITER ;  
9  
10  
11  
12  
13
```

SUCCESS

Rows affected: 1

MySQL 5.7 Command Line Client

```
mysql> DELIMITER $$  
mysql> CREATE TRIGGER my_trigger BEFORE INSERT ON employee FOR EACH ROW BEGIN  
    INSERT INTO trigger_test VALUES('added new employee');  
END$$  
Query OK, 0 rows affected (0.01 sec)  
  
mysql> DELIMITER ;  
mysql>
```



Giraffe Academy



PepSQL

File Edit View Query Window

Giraffe * Feedback

Untitled query +

Untitled query

8 DELIMITER ;
9
10 INSERT INTO employee
11 VALUES(109, 'Oscar', 'Martinez', '1968-02-19', 'M', 69000, 106, 3);
12
13 SELECT * FROM trigger_test;

14
15
16

▶ Run

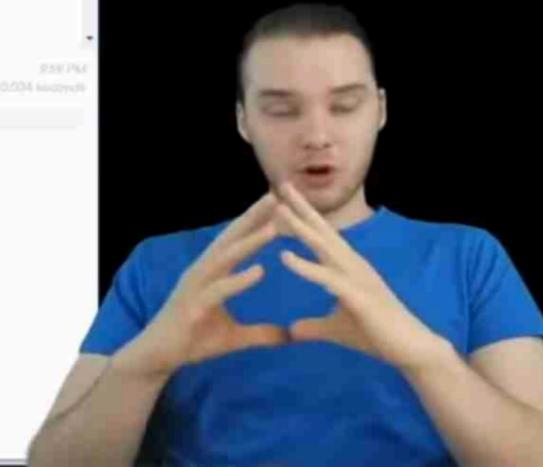
Success
1 rows

Explore SQL Data User Export %

message
added new employee

mysql> DELIMITER \$\$
mysql> 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 (0.01 sec)

mysql> DELIMITER ;
mysql>



Giraffe Academy



SUBSCRIBE

PopSQL

File Edit View Query Window

Giraffe

Untitled query

Untitled query

Run

```
1 DELIMITER $$  
2 CREATE  
3     TRIGGER my_trigger BEFORE INSERT  
4     ON employee  
5     FOR EACH ROW BEGIN  
6         INSERT INTO trigger_test VALUES(NEW.first_name);  
7     END$$  
8 DELIMITER ;  
9  
10  
11  
12 SELECT * FROM trigger_test;  
13
```

Success

1 rows

Explore SQL Data Close Export %

message
added new employee

MySQL 5.7 Command Line Client

```
mysql> DELIMITER $$  
mysql> CREATE TRIGGER my_trigger BEFORE INSERT ON employee FOR EACH ROW BEGIN  
    INSERT INTO trigger_test VALUES('added new employee');  
END$$  
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> DELIMITER ;  
mysql>
```



Giraffe Academy



SUBSCRIBE

PopSQL

File Edit View Query Window

Giraffe *

Untitled query

```
1      INSERT INTO trigger_test VALUES(NEW.first_name);
2
3      END$$
4 DELIMITER ;
5
6
7
8
9
10 INSERT INTO employee
11 VALUES(110, 'Kevin', 'Malone', '1978-02-19', 'M', 69000, 106, 3);
12
13
14
15 SELECT * FROM trigger_test;
16
17
18
```

Success
3 rows

Explore SQL Data Client Export %

message
added new employee
added new employee
Kevin

This screenshot shows the PopSQL application interface. On the left, there's a code editor window titled 'Untitled query' containing MySQL code. The code includes a trigger definition, an insertion into the 'employee' table, and a select statement from 'trigger_test'. Below the code editor is a success message indicating 3 rows affected. At the bottom, there are tabs for Explore, SQL, Data, Client, and Export, along with a percentage completion indicator (0.043 seconds).

MySQL 5.7 Command Line Client

```
mysql> DELIMITER ;
mysql> DELIMITER $$
mysql> CREATE TRIGGER my_trigger BEFORE INSERT ON employee FOR EACH ROW BEGIN INSERT INTO trigger_test VALUES(NEW.first_name); END$$
ERROR 1359 (HY000): Trigger already exists
mysql> CREATE TRIGGER my_trigger1 BEFORE INSERT ON employee FOR EACH ROW BEGIN INSERT INTO trigger_test VALUES(NEW.first_name); END$$
Query OK, 0 rows affected (0.02 sec)

mysql> DELIMITER ;
mysql>
```

This screenshot shows a MySQL command-line client window. It displays a command to create a trigger named 'my_trigger' on the 'employee' table, which inserts into the 'trigger_test' table. An error message 'Trigger already exists' is shown. A second attempt to create a trigger with the same name ('my_trigger1') is successful, with a 'Query OK' message and 0 rows affected. The client interface includes a video feed of a person speaking and standard command-line client controls.



Giraffe Academy



SUBSCRIBE

PopSQL

File Edit View Query Window

Giraffe * Feedback

Untitled query +

Untitled query

Run

```
1 DELIMITER $$  
2 CREATE  
3 TRIGGER my_trigger BEFORE INSERT  
4 ON employee  
5 FOR EACH ROW BEGIN  
6     IF NEW.sex = 'M' THEN  
7         INSERT INTO trigger_test VALUES('added male employee');  
8     ELSEIF NEW.sex = 'F' THEN  
9         INSERT INTO trigger_test VALUES('added female');  
10    ELSE  
11        INSERT INTO trigger_test VALUES('added other employee');  
12    END IF;  
13 END$$  
14 DELIMITER ;  
15  
16  
17  
18 SELECT * FROM trigger_test;  
19  
20
```

Success
3 rows

Explore SQL Data Client Export %

10:09 PM 0.043 seconds

message
added new employee
added new employee

Giraffe Academy

MySQL 5.7 Command Line Client

```
mysql> DELIMITER ;  
mysql> DELIMITER $$  
mysql> CREATE TRIGGER my_trigger BEFORE INSERT ON employee FOR EACH ROW BEGIN INSERT INTO trigger_test VALUES(NEW.first_name); END$$  
ERROR 1359 (HY000): Trigger already exists  
mysql> CREATE TRIGGER my_trigger1 BEFORE INSERT ON employee FOR EACH ROW BEGIN INSERT INTO trigger_test VALUES(NEW.first_name); END$$  
Query OK, 0 rows affected (0.02 sec)  
  
mysql> DELIMITER ;  
mysql>
```



(🔥)
SUBSCRIBE

A screenshot of a video call interface. On the right, a developer wearing a blue t-shirt is speaking. On the left, there is a code editor window titled "Untitled query" and a MySQL command-line client window.

Code Editor (Untitled query):

```
12     END IF;
13 END$$
14 DELIMITER ;
15
16 INSERT INTO employee
17 VALUES(111, 'Pam', 'Beesly', '1988-02-19', 'F', 69000, 106, 3);
18
19
20 SELECT * FROM trigger_test;
21
22
23
```

MySQL Command Line Client:

```
Query OK, 0 rows affected (0.01 sec)

mysql> DELIMITER ;
mysql>
```

Success:

6 rows

message
added new employee
added new employee
Kevin
added new employee
Pam
added female



Giraffe Academy



SUBSCRIBE

PopSQL

File Edit View Query Window

Giraffe

Untitled query

Untitled query

Run

```
3  TRIGGER my_trigger2 AFTER DELETE
4  ON employee
5  FOR EACH ROW BEGIN
6      IF NEW.sex = 'M' THEN
7          INSERT INTO trigger_test VALUES('added male employee');
8      ELSEIF NEW.sex = 'F' THEN
9          INSERT INTO trigger_test VALUES('added female');
10     ELSE
11         INSERT INTO trigger_test VALUES('added other employee');
12     END IF;
13 END$$
14 DELIMITER ;
15
```

Success
6 rows

Explore SQL Data Count Export %

message
added new employee
added new employee
Kevin
added new employee
Pam
added female

MySQL 5.7 Command Line Client

```
mysql> DROP TRIGGER my_trigger;
Query OK, 0 rows affected (0.01 sec)
```

mysql>



Windows.



Giraffe Academy



SUBSCRIBE

ER Diagrams Intro



Entity - An object we want to model & store information about

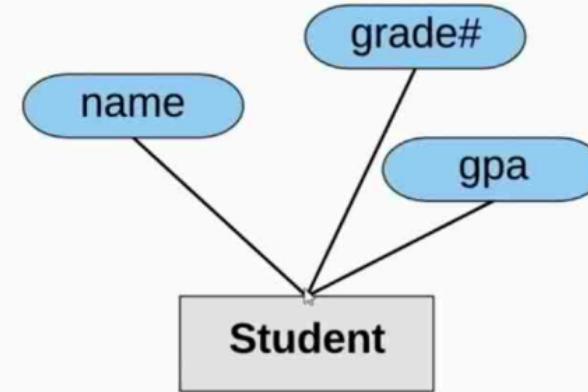
Student



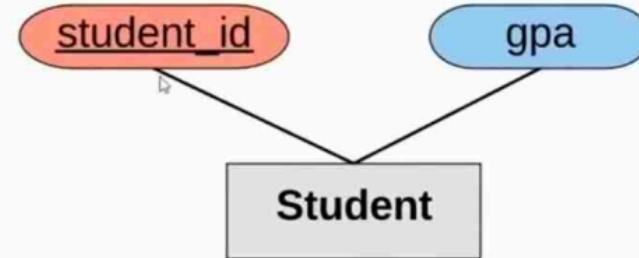
Giraffe Academy

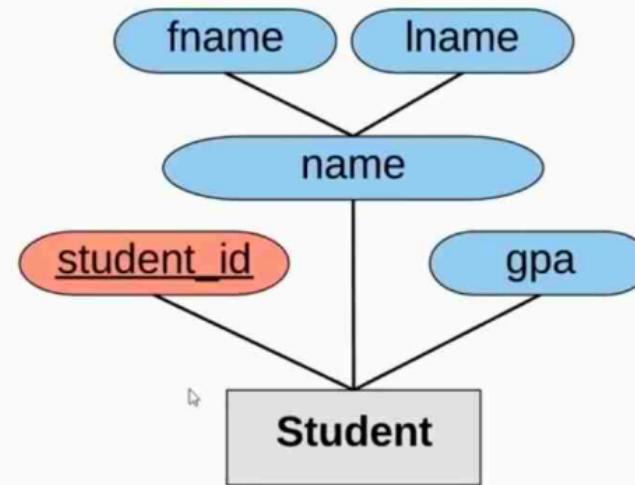


Attributes - Specific pieces of information about an entity



Primary Key - An attribute(s) that uniquely identify an entry in the database table

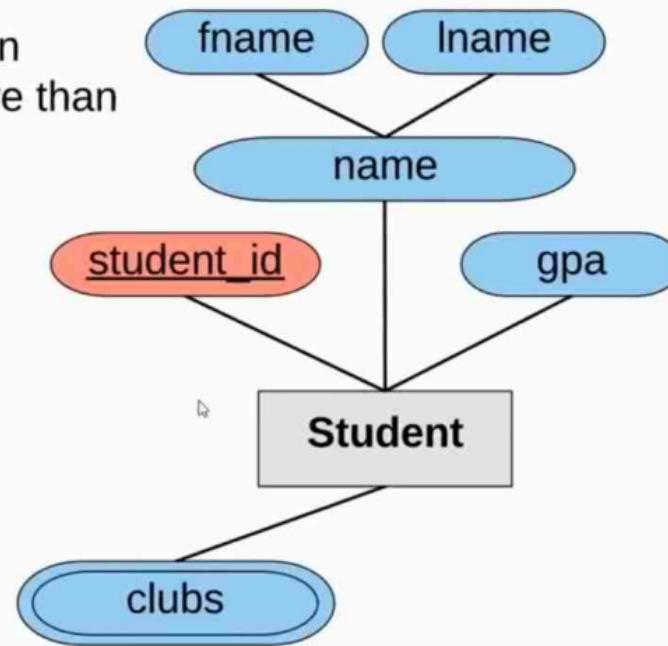




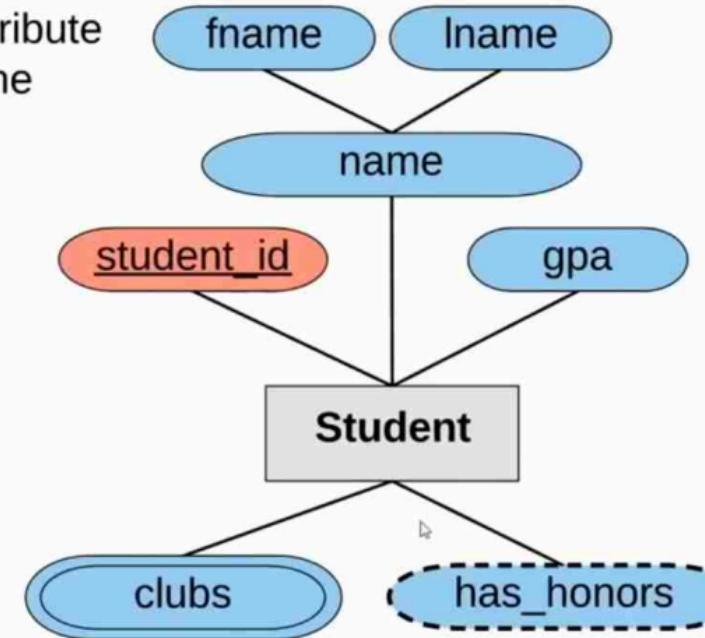
Composite Attribute - An attribute that can be broken up into sub-attributes



Multi-valued Attribute - An attribute that can have more than one value

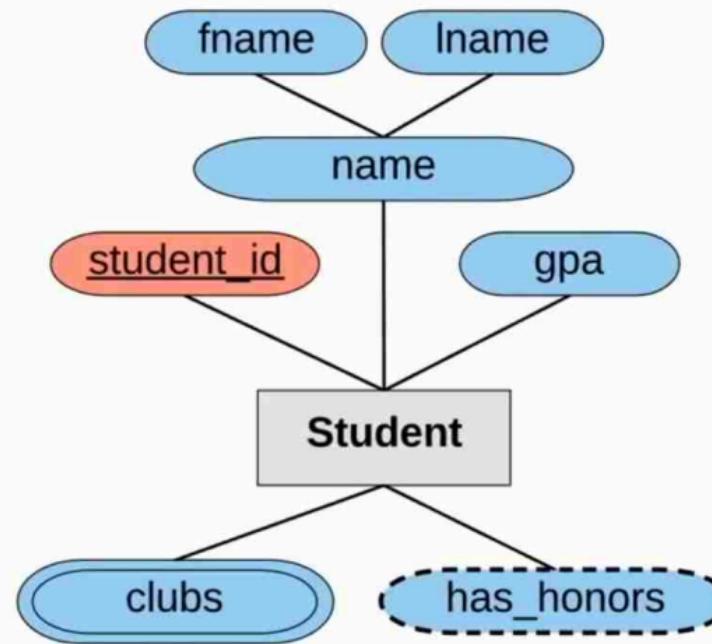


Derived Attribute - An attribute that can be derived from the other attributes



Giraffe Academy



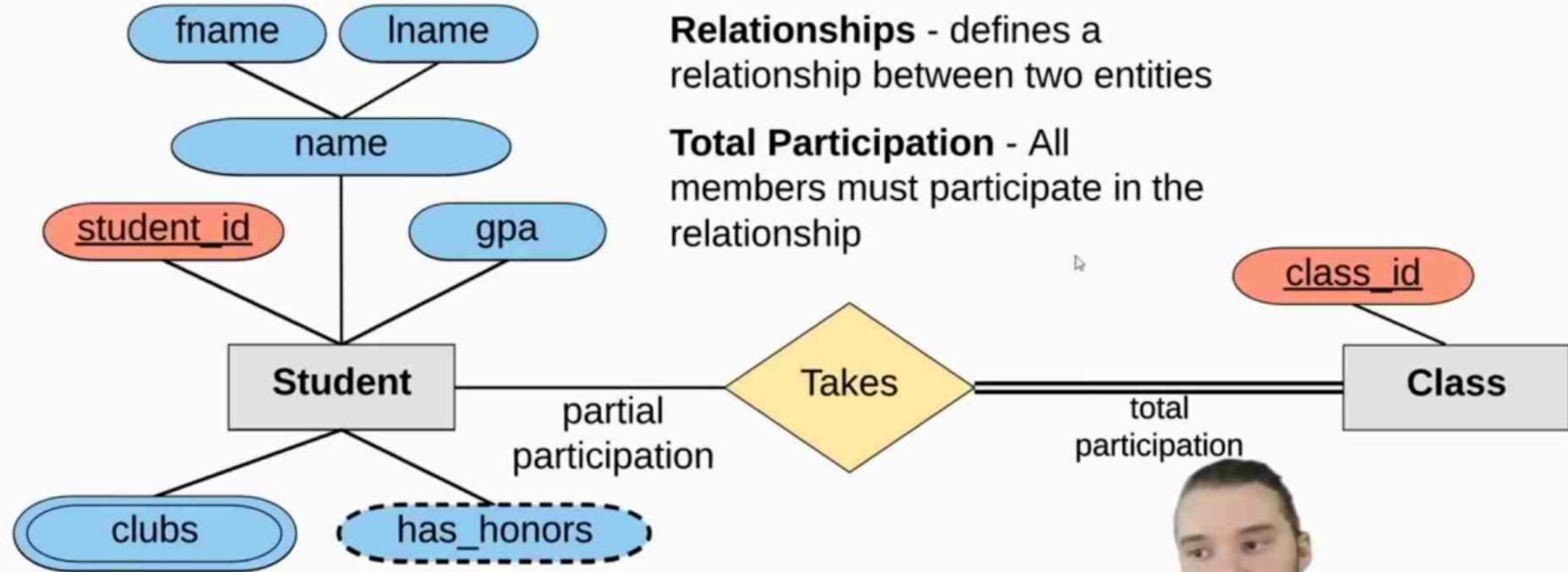


Multiple Entities - You can define more than one entity in the diagram



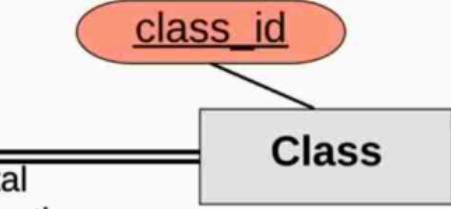
Giraffe Academy

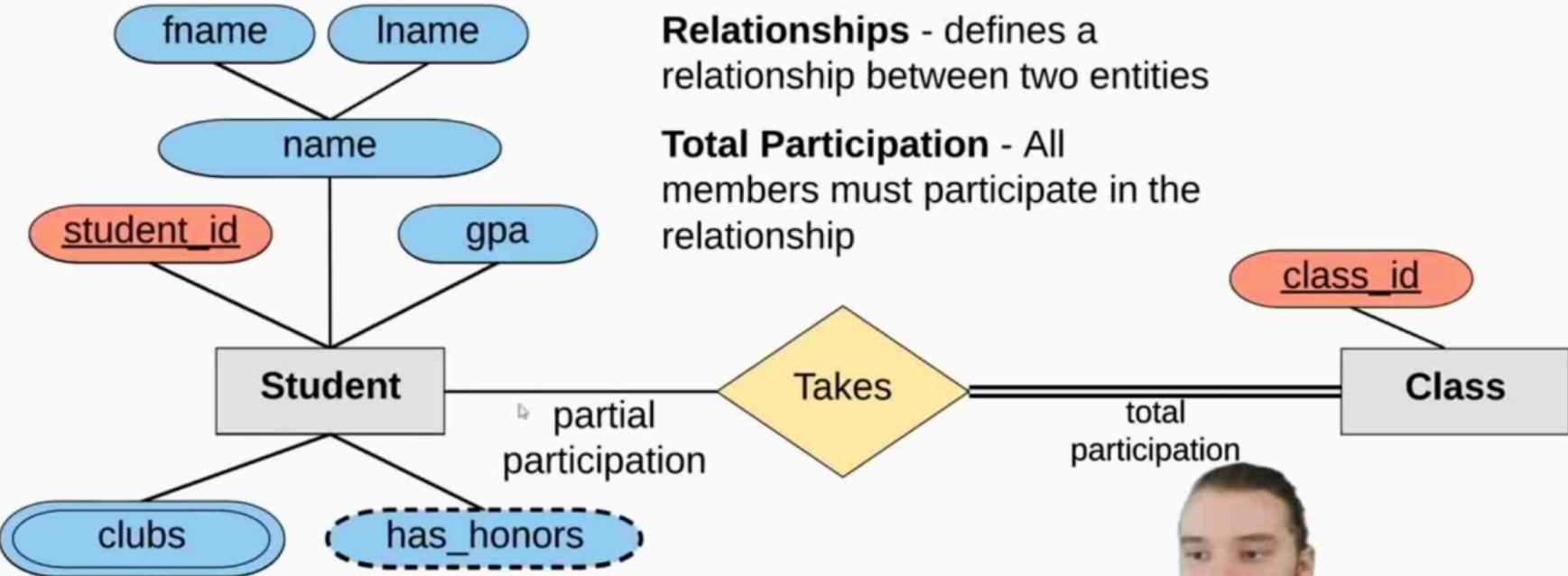




Relationships - defines a relationship between two entities

Total Participation - All members must participate in the relationship





Relationships - defines a relationship between two entities

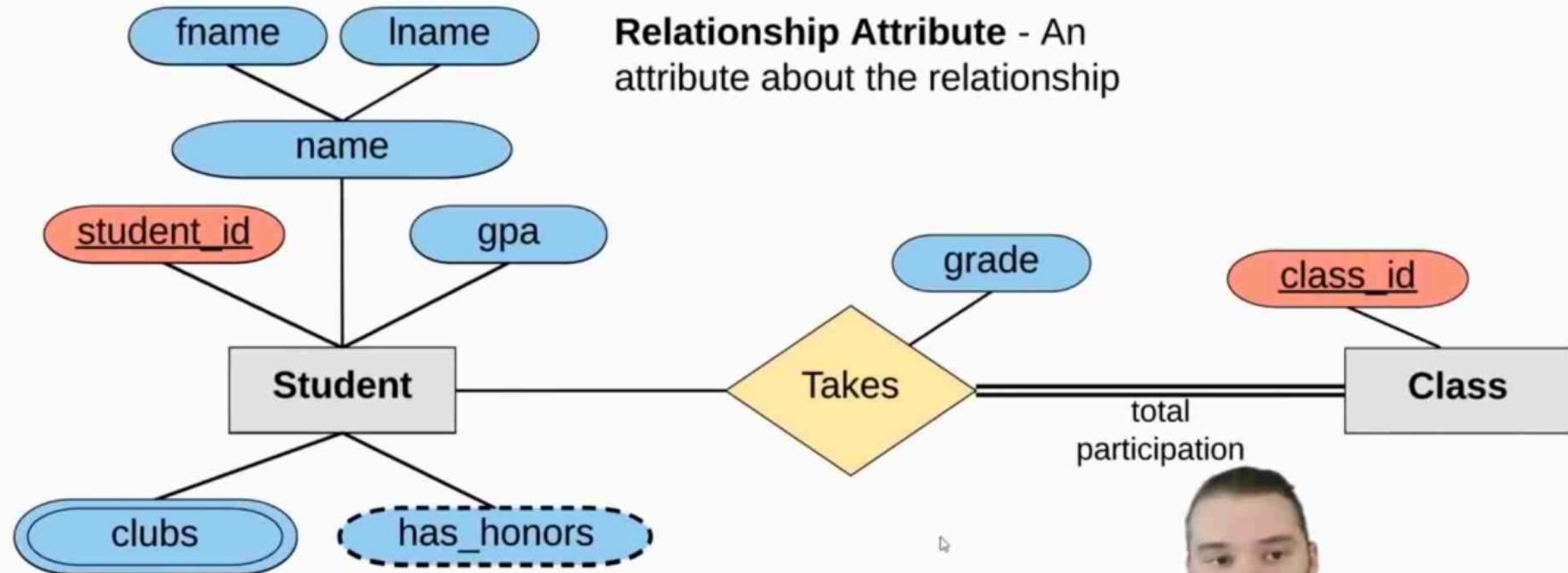
Total Participation - All members must participate in the relationship

*Single line
single line = partial participation



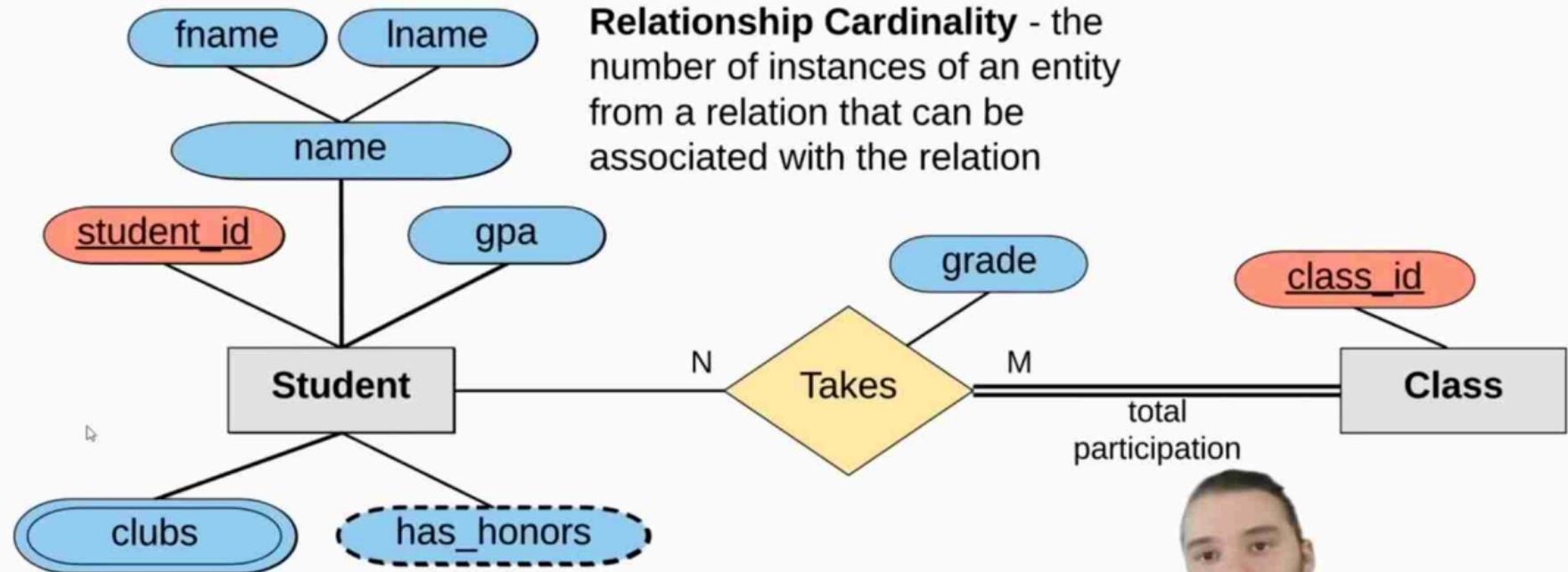
Giraffe Academy





Giraffe Academy

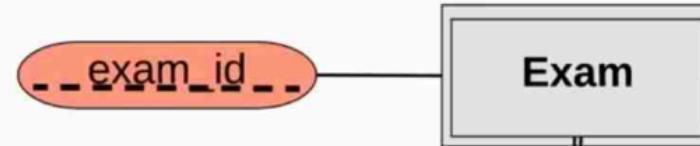




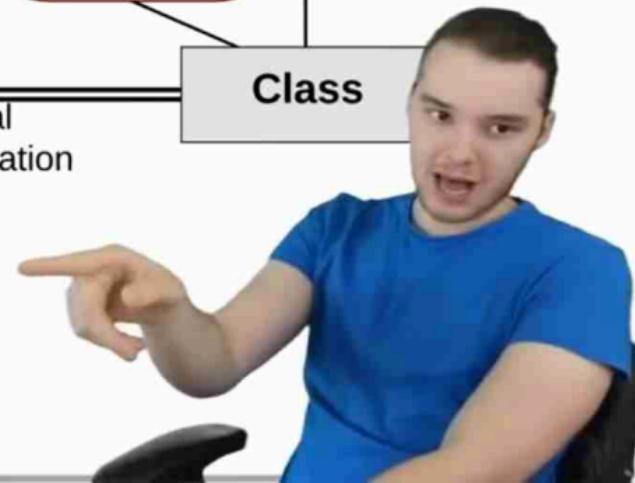
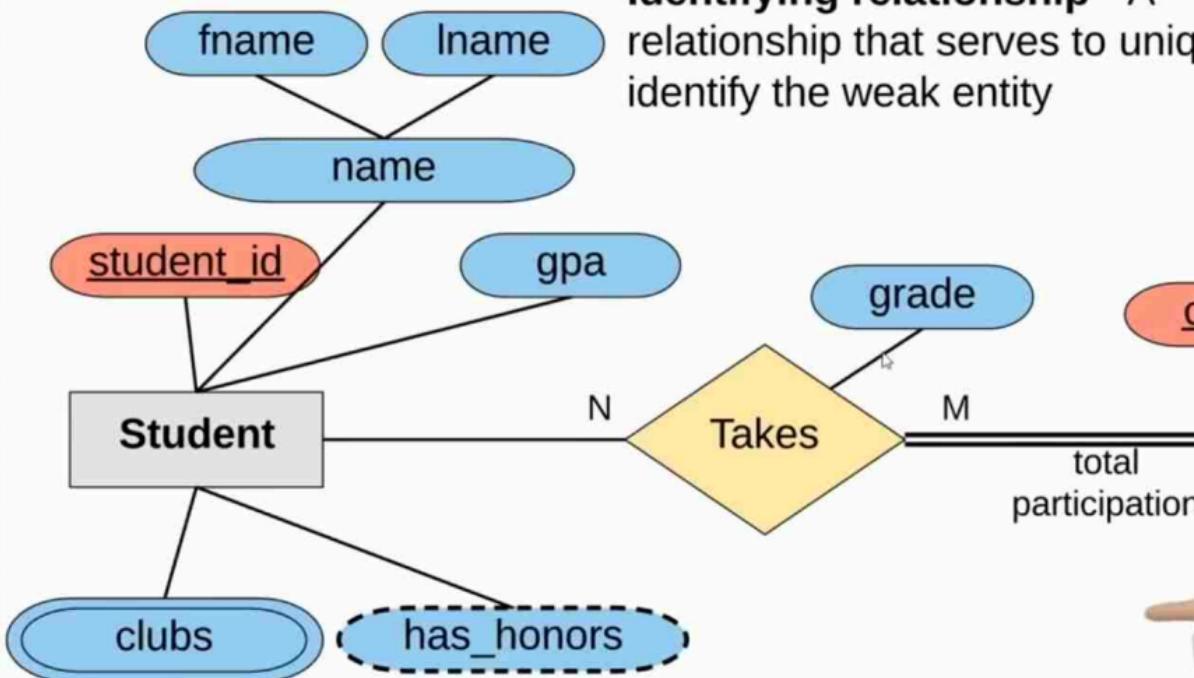
1 : 1
1 : N
N : M



Weak Entity - An entity that cannot be uniquely identified by its attributes alone



Identifying relationship - A relationship that serves to uniquely identify the weak entity



Giraffe Academy



Designing an ER Diagram



Company Data Requirements

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, sex, 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, or they can also act as the supervisor for employees at other branches. An employee can have one supervisor.



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. It's 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 can supply products to multiple branches.



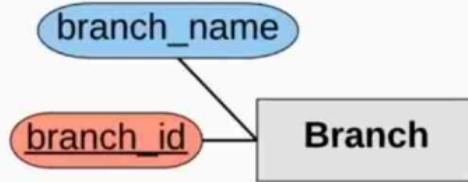
number to identify it. A single client may only be handled by one branch at a time.

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.

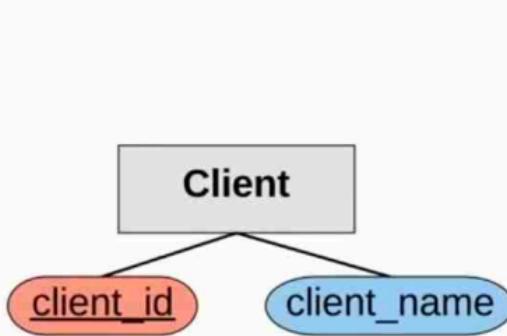
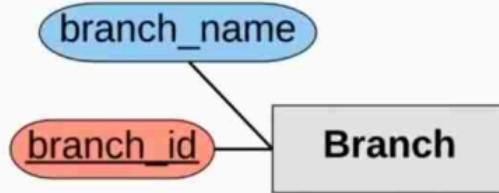
Many branches will need to work with suppliers to buy inventory. For each supplier we'll keep track of their name and the type of product they're selling the branch. A single supplier may supply products to multiple branches.



The company is organized into **branches**.
Each branch has a **unique number**, and a
name



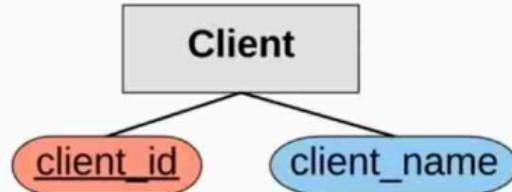
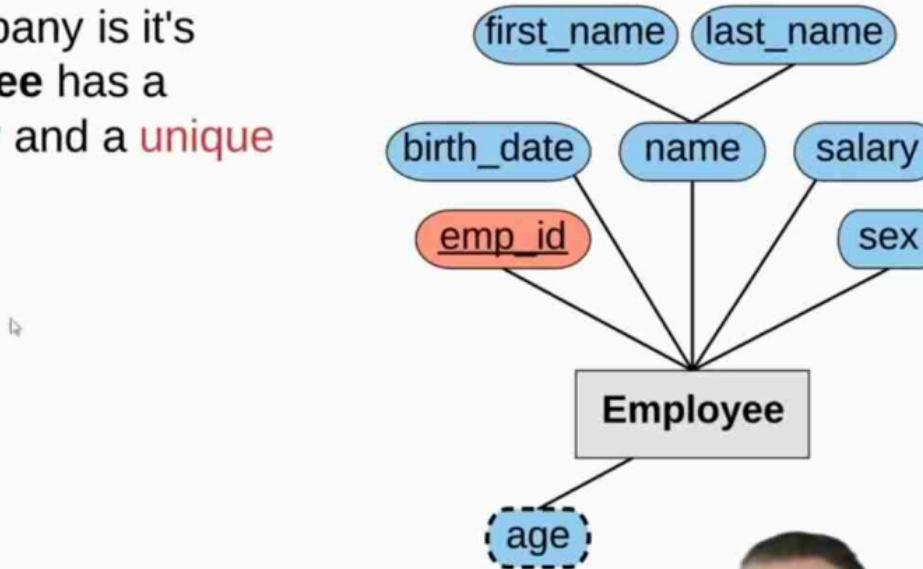
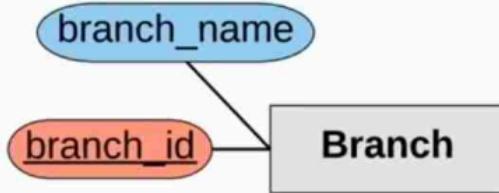
The company makes it's money by selling to **clients**. Each **client** has a **name** and a **unique number** to identify it.



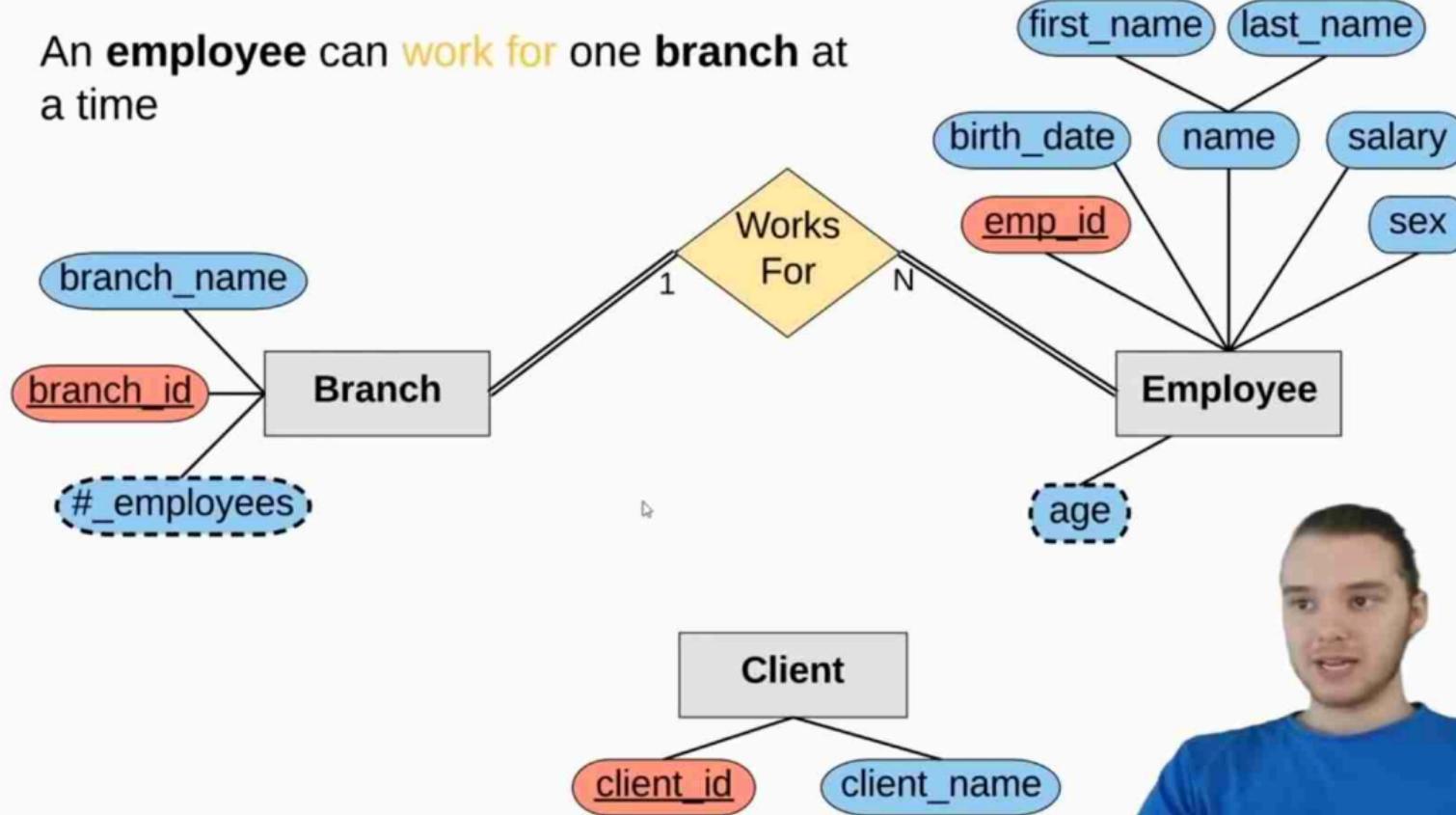
Giraffe Academy



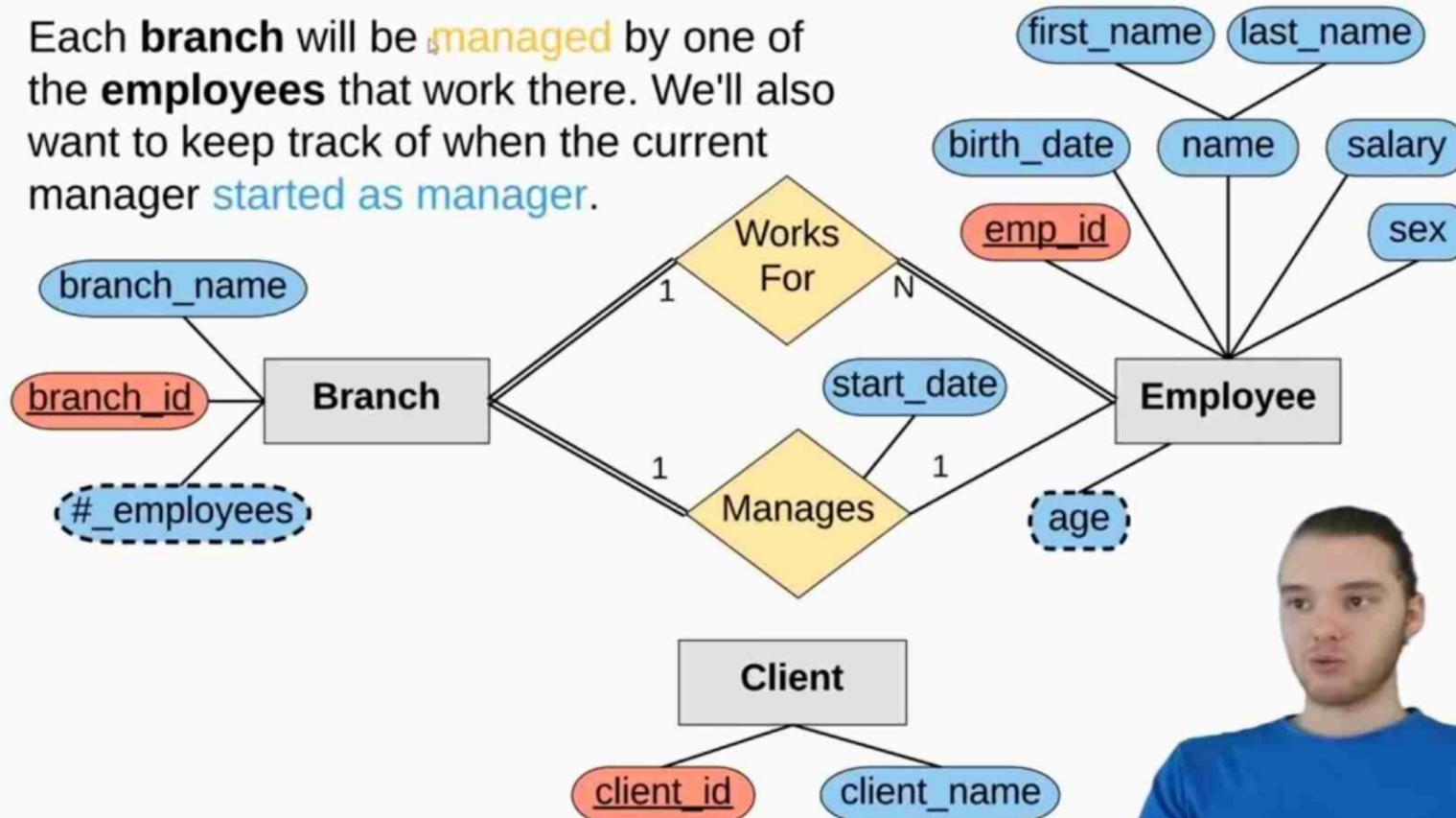
The foundation of the company is it's **employees**. Each **employee** has a **name**, **birthday**, **sex**, **salary** and a **unique number** to identify it.



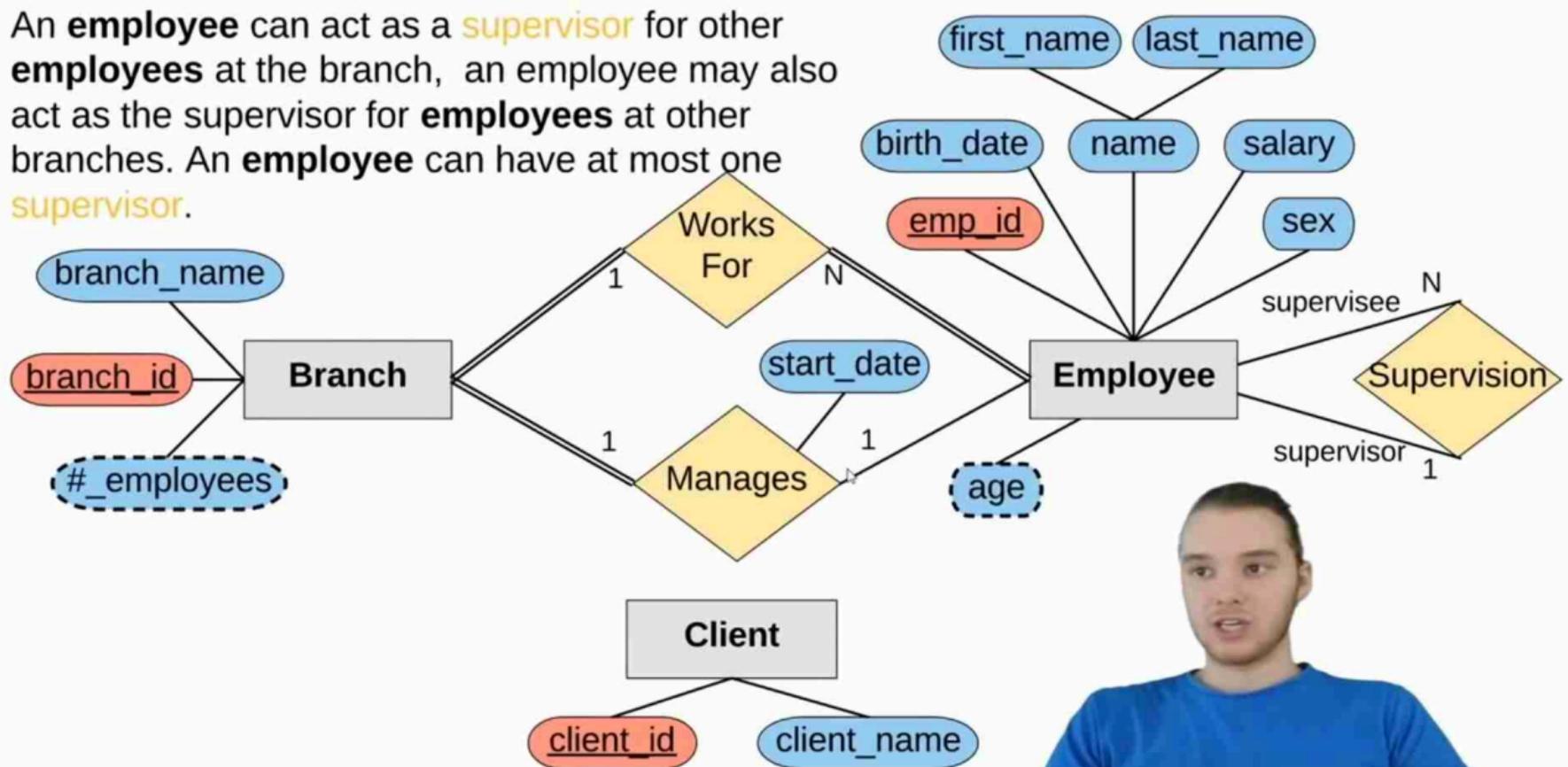
An **employee** can **work for** one **branch** at a time



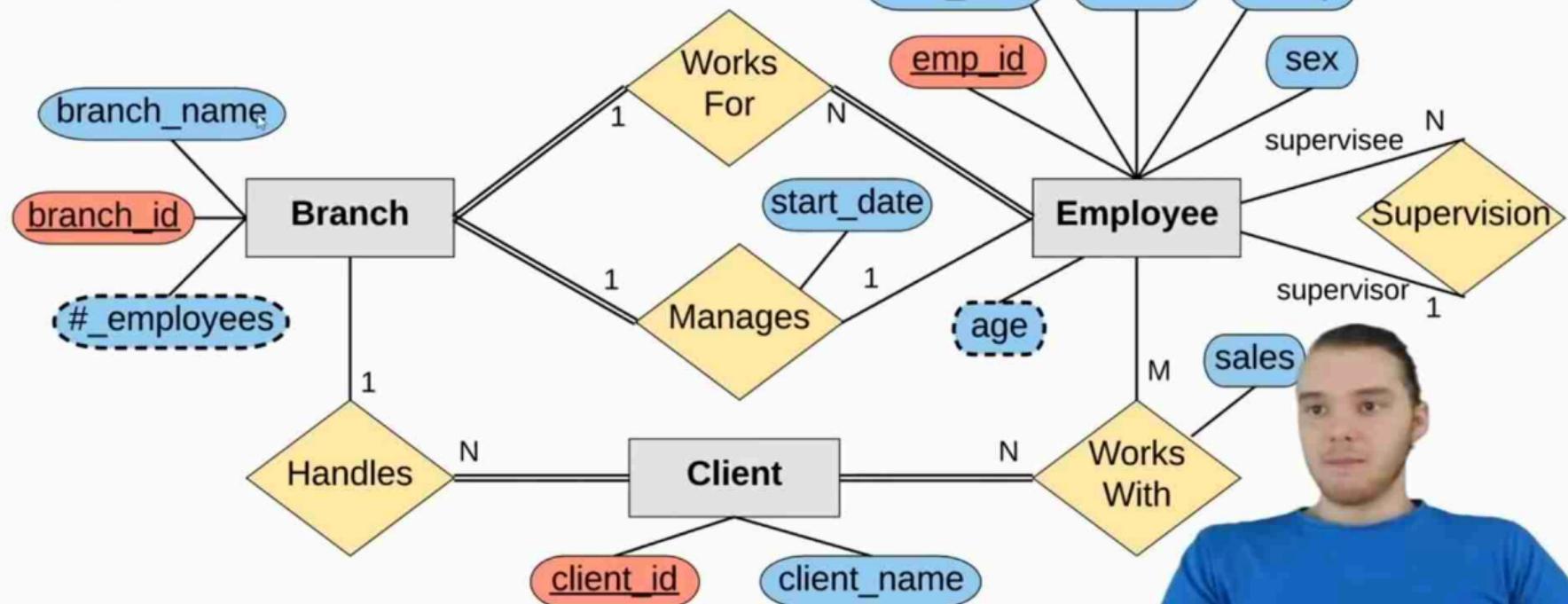
Each **branch** will be **managed** by one of the **employees** that work there. We'll also want to keep track of when the current manager **started as manager**.



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**.

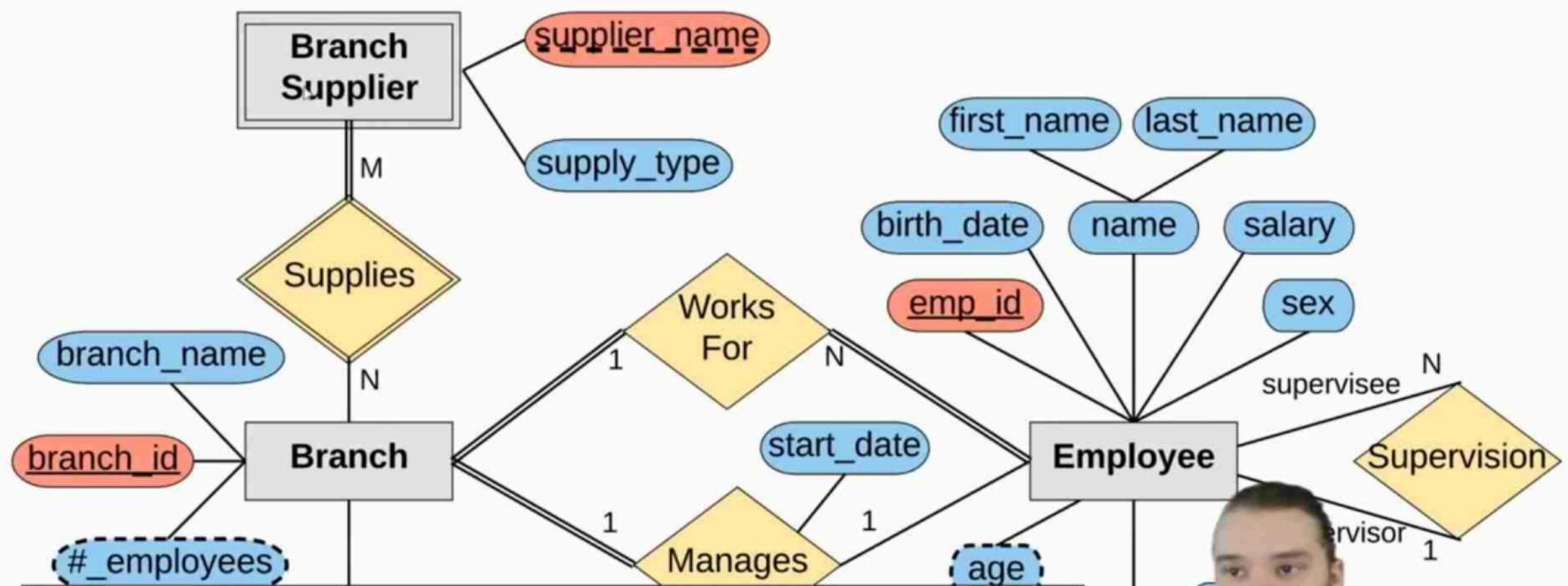


Employees can **work with** **clients** controlled by their `first_name` `last_name` branch to sell them stuff. If necessary multiple employees can work with the same client.



We'll want to keep track of how many dollars worth of stuff each employee **sells** to each client they work with.





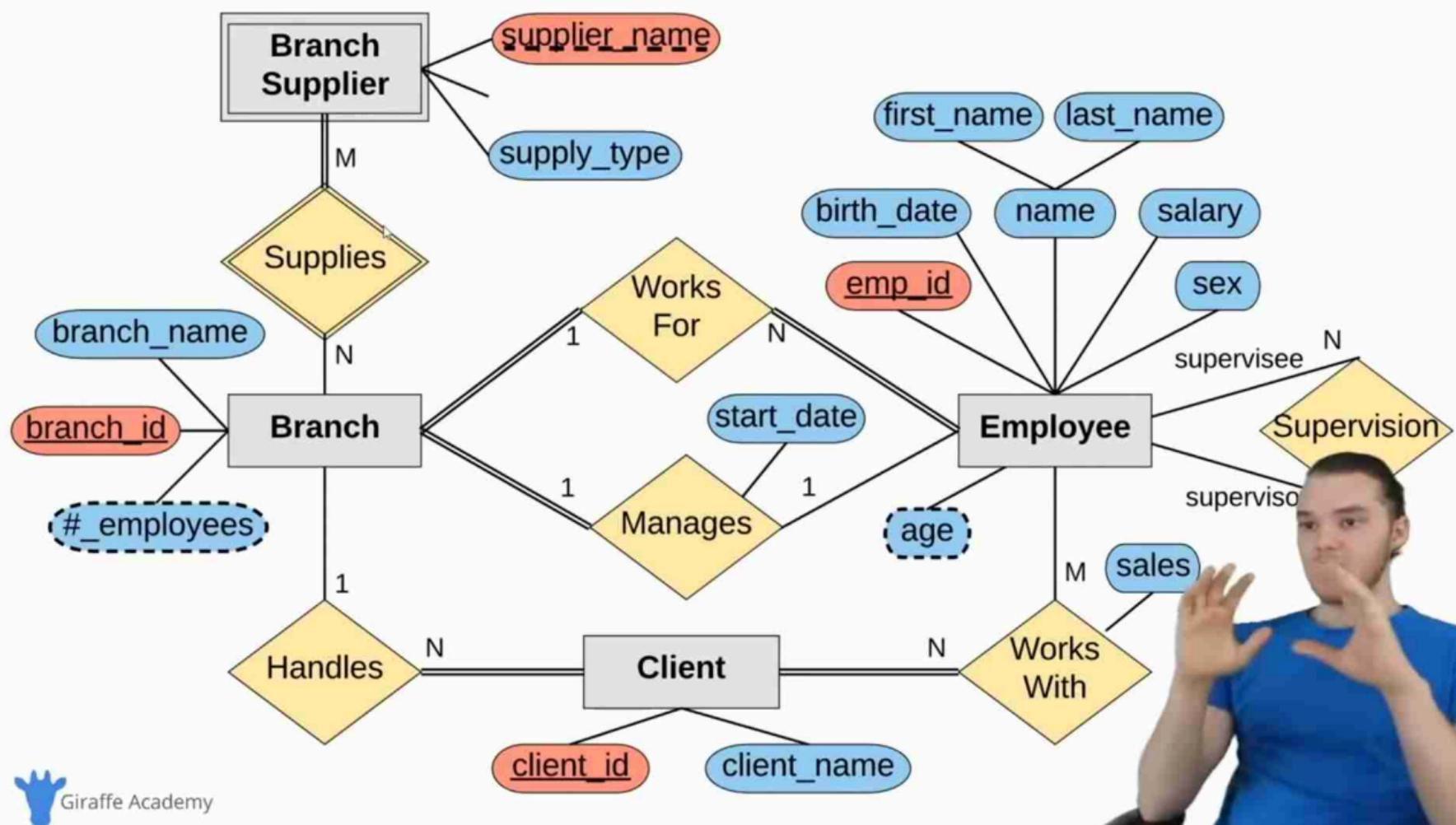
Many **branches** will need to work with **suppliers** to buy inventory. For each supplier we'll keep track of their **name** and the **type of product** they're selling the **branch**. A single supplier may supply products to multiple branches.

`client_id` `client_name`



Giraffe Academy





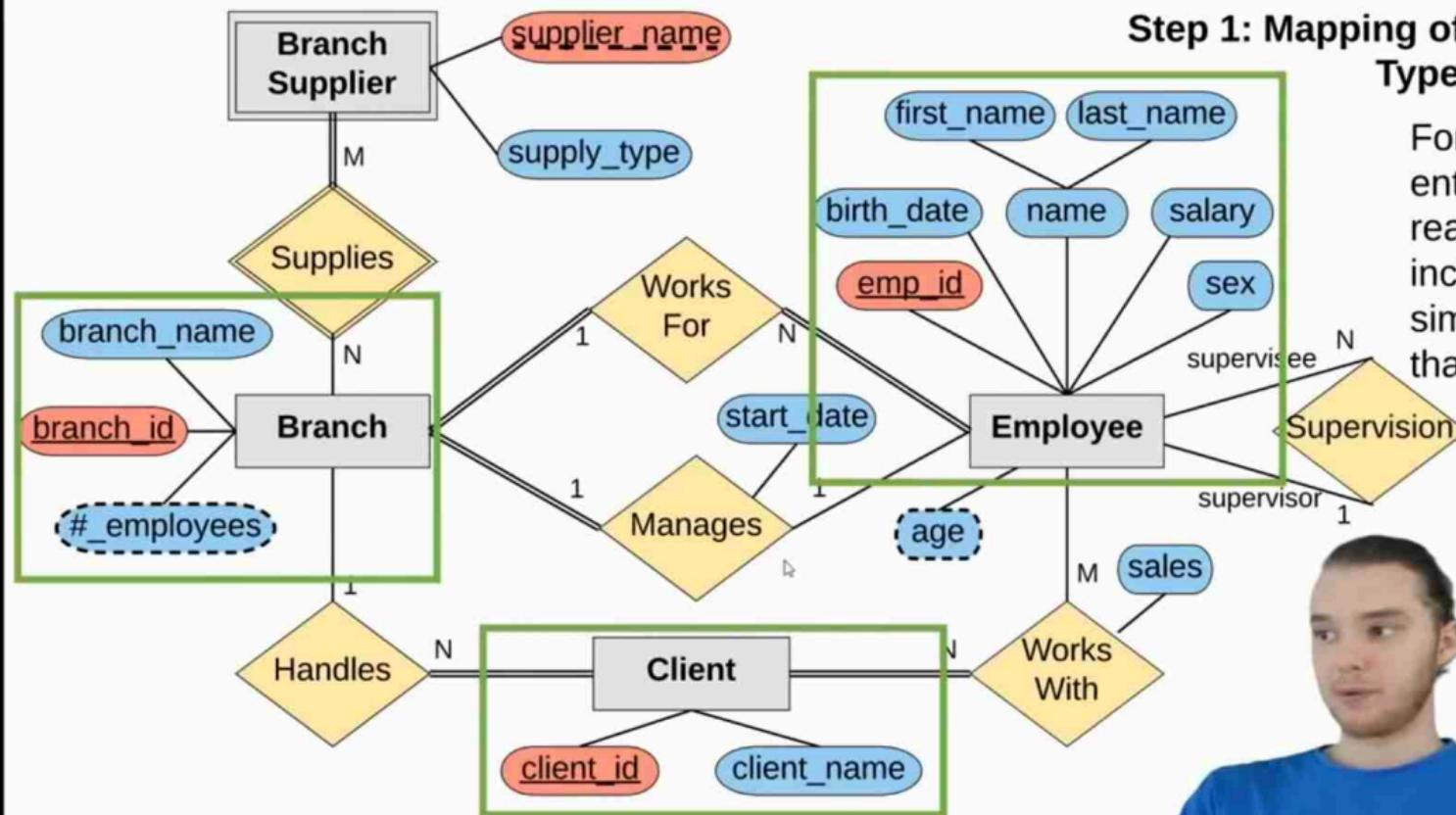
Giraffe Academy



SUBSCRIBE

Converting ER Diagrams to Schemas





Step 1: Mapping of Regular Entity Types

For each regular entity type create a relation (table) that includes all the simple attributes of that entity



Employee

<u>emp_id</u>	first_name	last_name	birth_date	sex	salary
---------------	------------	-----------	------------	-----	--------

Branch

<u>branch_id</u>	branch_name
------------------	-------------

Client

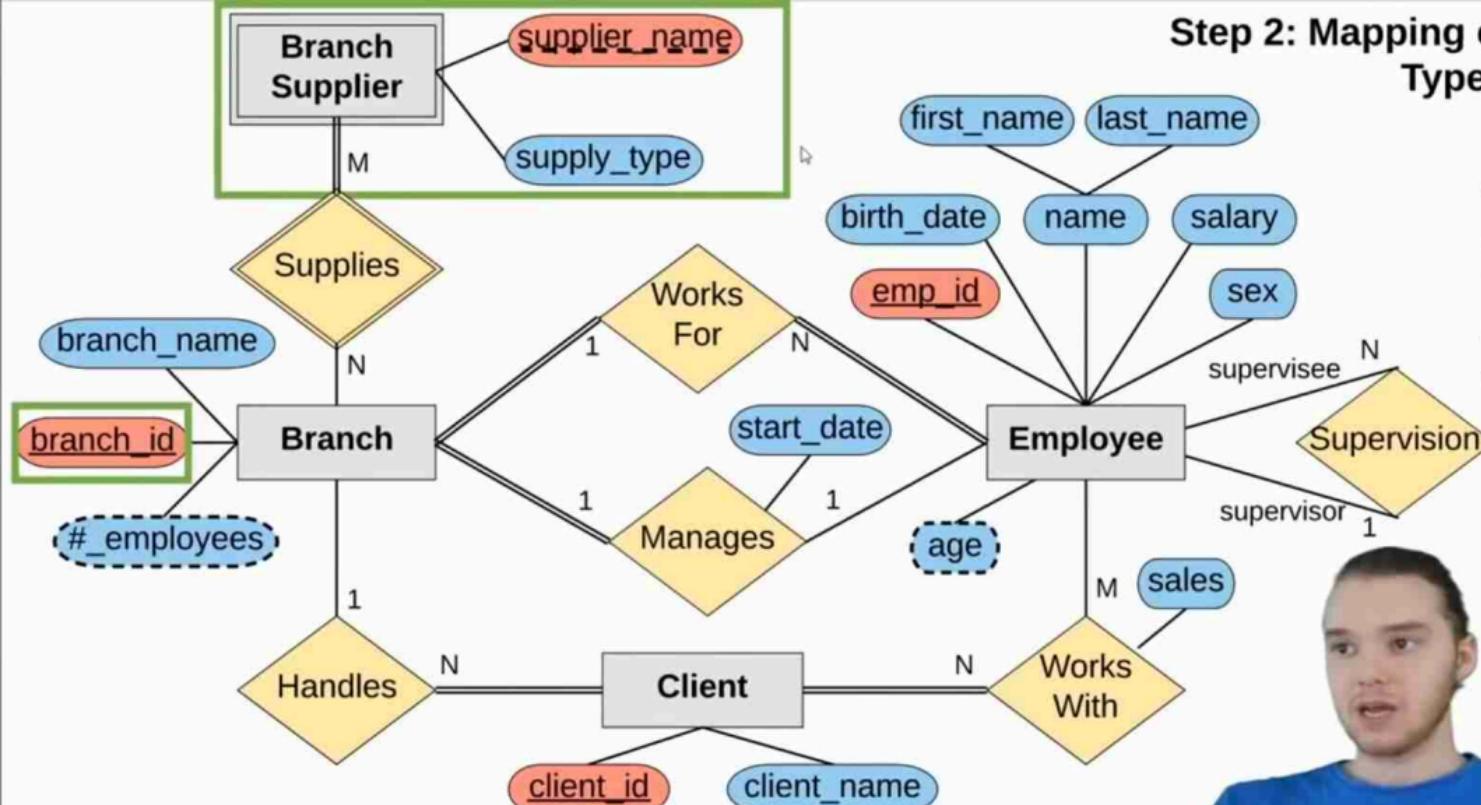
<u>client_id</u>	client_name
------------------	-------------



Giraffe Academy



SUBSCRIBE



Step 2: Mapping of Weak Entity Types

For each weak entity type create a relation (table) that includes all simple attributes of the weak entity.

The primary key of the new relation should be the partial key of the weak entity plus the primary key of its owner



Employee

<u>emp_id</u>	first_name	last_name	birth_date	sex	salary
---------------	------------	-----------	------------	-----	--------

Branch

<u>branch_id</u>	branch_name
------------------	-------------

Client

<u>client_id</u>	client_name
------------------	-------------

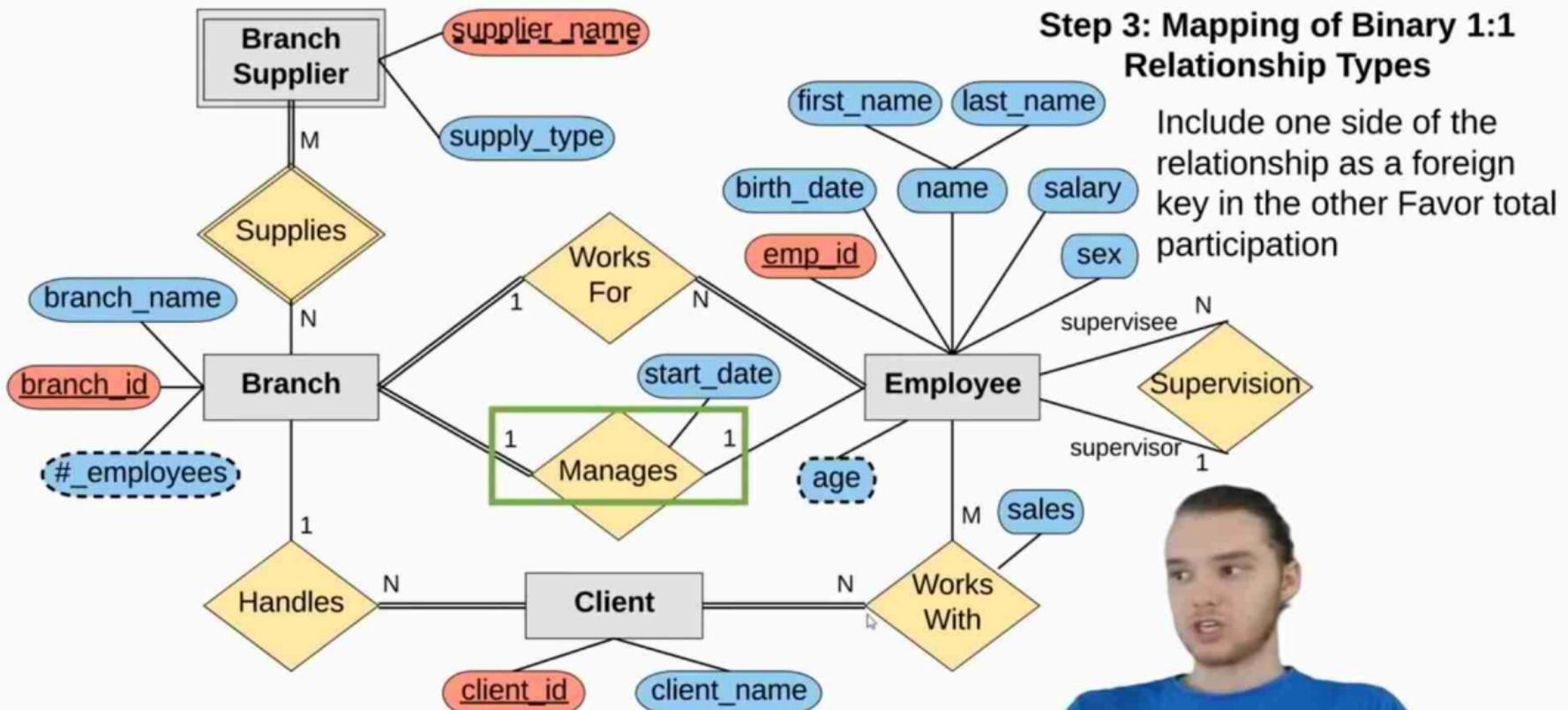
Branch Supplier

<u>branch_id</u>	<u>supplier_name</u>	supply_type
------------------	----------------------	-------------



Giraffe Academy



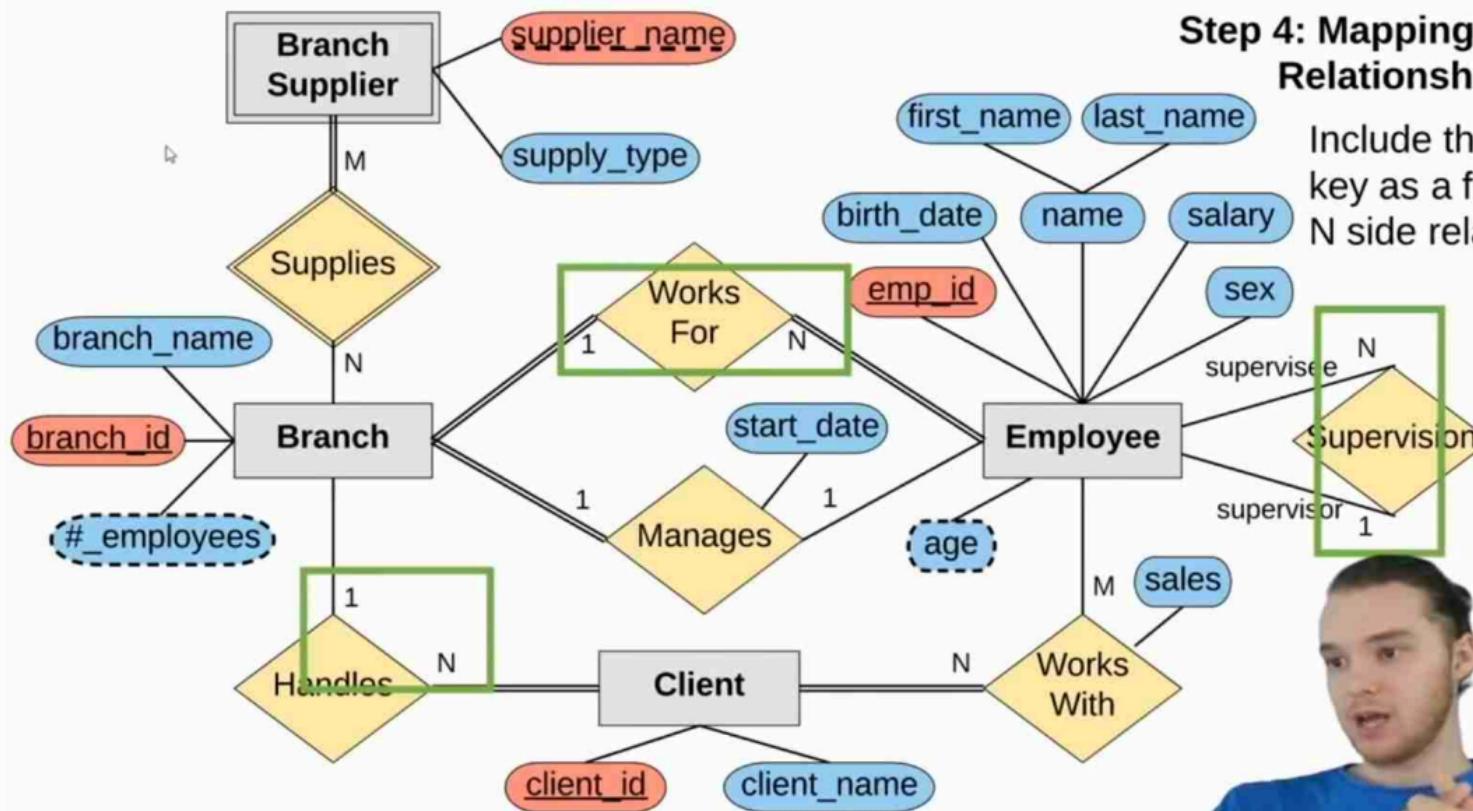


Giraffe Academy



Step 4: Mapping of Binary 1:N Relationship Types

Include the 1 side's primary key as a foreign key on the N side relation (table)



Employee

<u>emp_id</u>	first_name	last_name	birth_date	sex	salary	super_id	branch_id
---------------	------------	-----------	------------	-----	--------	----------	-----------

Branch

<u>branch_id</u>	branch_name	<u>mgr_id</u>	mgr_start_date
------------------	-------------	---------------	----------------

Client

<u>client_id</u>	client_name	branch_id
------------------	-------------	-----------

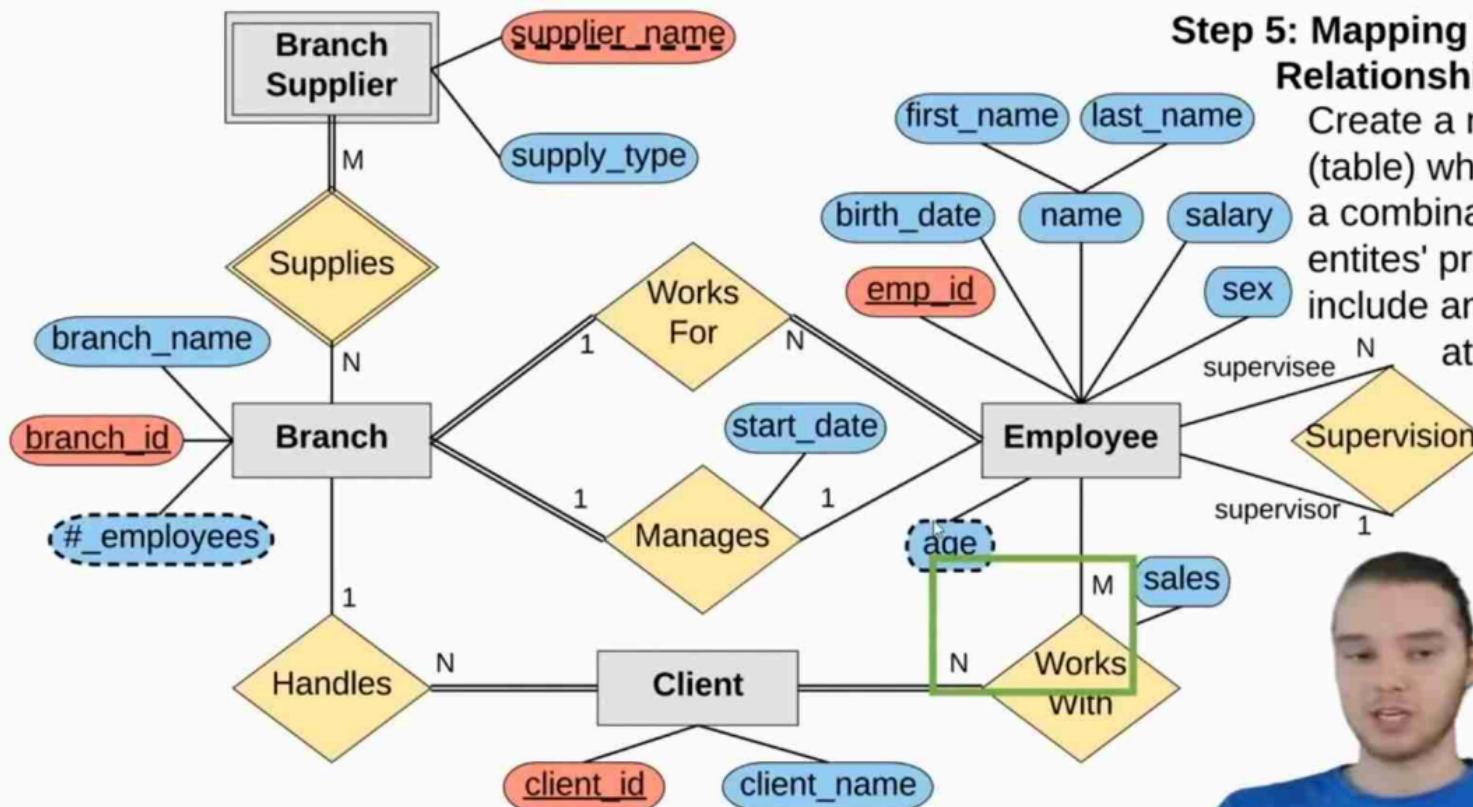
Branch Supplier

<u>branch_id</u>	<u>supplier_name</u>	supply_type
------------------	----------------------	-------------



Giraffe Academy





Step 5: Mapping of Binary M:N Relationship Types

Create a new relation (table) who's primary key is a combination of both entities' primary key's. Also include any relationship attributes



Employee

<u>emp_id</u>	first_name	last_name	birth_date	sex	salary	super_id	branch_id
---------------	------------	-----------	------------	-----	--------	----------	-----------

Branch

<u>branch_id</u>	branch_name	mgr_id	mgr_start_date
------------------	-------------	--------	----------------

Client

<u>client_id</u>	client_name	branch_id
------------------	-------------	-----------

Branch Supplier

<u>branch_id</u>	<u>supplier_name</u>	supply_type
------------------	----------------------	-------------

Works On

<u>emp_id</u>	<u>client_id</u>	total_sales
---------------	------------------	-------------



Employee

<u>emp_id</u>	first_name	last_name	birth_date	sex	salary	super_id	branch_id
---------------	------------	-----------	------------	-----	--------	----------	-----------

Branch

<u>branch_id</u>	branch_name	mgr_id	mgr_start_date
------------------	-------------	--------	----------------

Client

<u>client_id</u>	client_name	branch_id
------------------	-------------	-----------

Branch Supplier

<u>branch_id</u>	<u>supplier_name</u>	supply_type
------------------	----------------------	-------------

Works On

<u>emp_id</u>	<u>client_id</u>	total_sales
---------------	------------------	-------------

Employee

emp_id	first_name	last_name	birth_date	sex	salary	super_id	branch_id
100	Jan	Levinson	1961-05-11	F	110,000	108	1
101	Michael	Scott	1964-03-15	M	75,000	100	2
102	Josh	Porter	1969-09-05	M	78,000	100	3
103	Angela	Martin	1971-06-25	F	63,000	101	2
104	Andy	Bernard	1973-07-22	M	65,000	102	3
105	Jim	Halpert	1978-10-01	M	71,000	102	3
106	Kelly	Kapoor	1980-02-05	F	55,000	101	2
107	Stanley	Hudson	1958-02-19	M	69,000	101	2
108	David	Wallace	1967-11-17	M	250,000	NULL	1

Branch

branch_id	branch_name	mgr_id	mgr_start_date
2	Scranton	101	1992-04-06
3	Stamford	102	1998-02-13
1	Corporate	108	2006-02-09

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

Works_With

emp_id	client_id	total_sales
107	400	55,000
101	401	267,000
105	402	22,500
104	403	5,000
105	403	12,000
107	404	33,000
104	405	26,000
101	406	15,000
107	406	130,000

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



SUBSCRIBE