Share

Unit 04.01

Many-to-Many Relationships in SQL: Many-to-Many

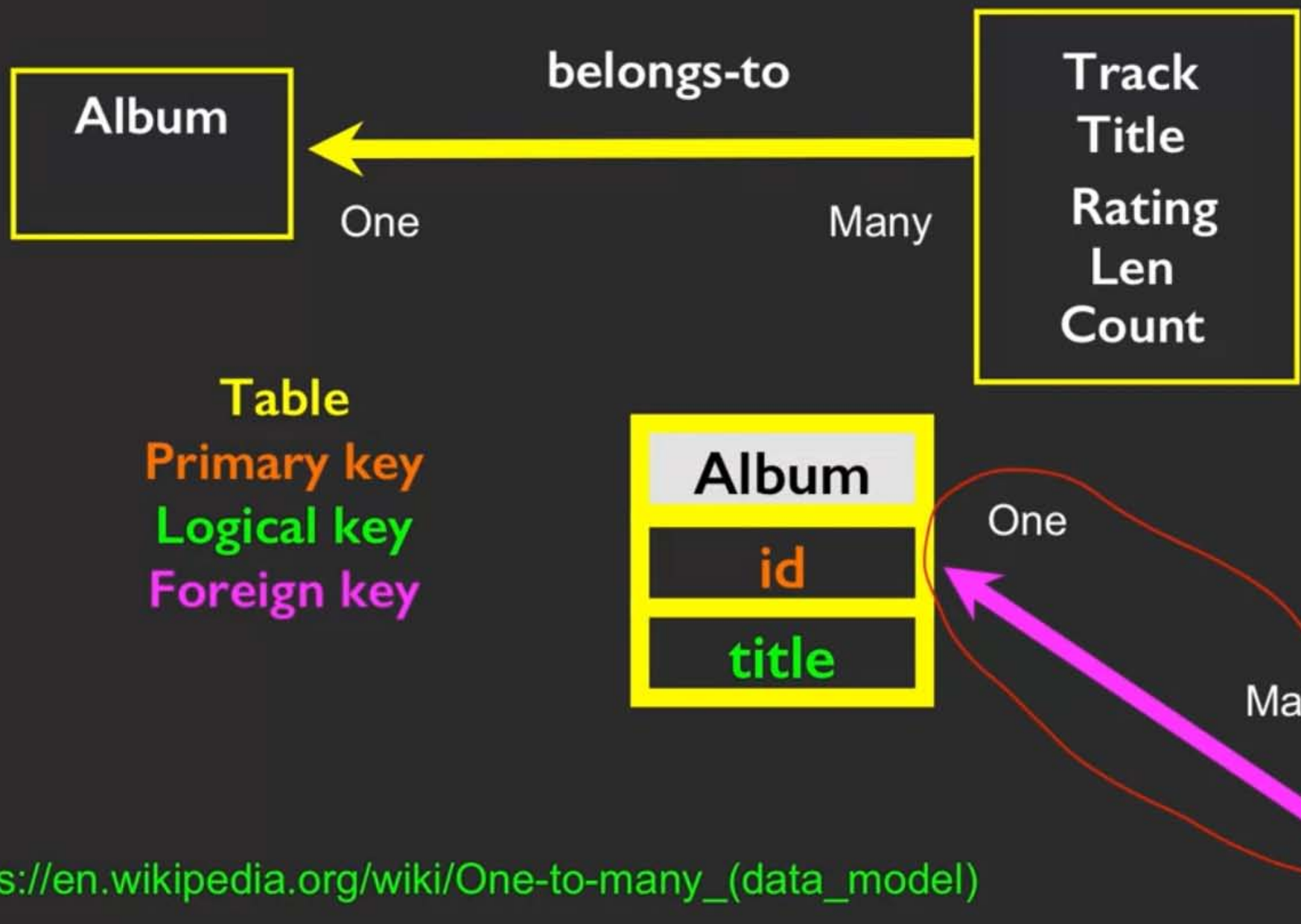
USING DATABASES WITH PYTHON

Charles Severance Ph.D.

Clinical Associate Professor of Information
School of Information

UNIVERSITY OF MICHIGAN

0:02 / 17:16


Album
belongs-to
Track
Title
Rating
Len
Count
One
Many
Review:
One to Many
Table
Primary key
Logical key
Foreign key
Album
id
title
One
Many
Track
id
title
rating
len
count
album_id
https://en.wikipedia.org/wiki/One-to-many_(data_model)

id	name
Filter	Filter
1	Rock
2	Metal

One

Many

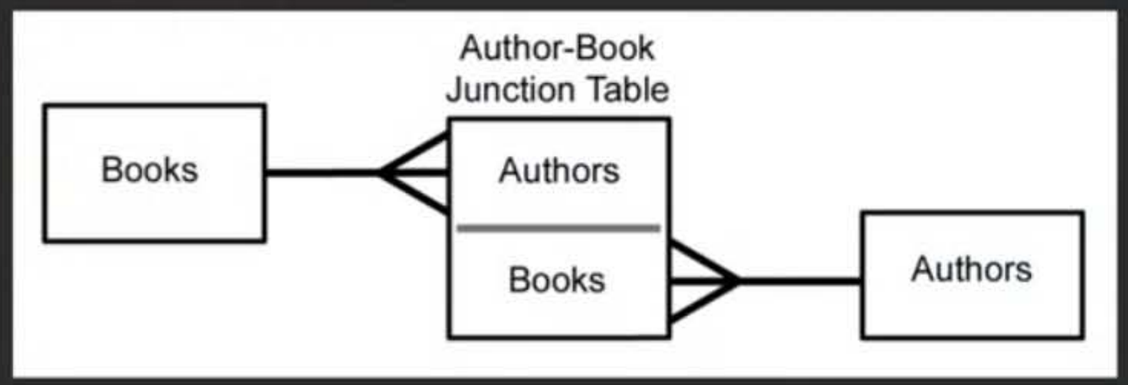
id	title	album_id	genre_id	len	rating	count
Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	Black Dog	2	1	297	5	0
2	Stairway	2	1	482	5	0
3	About to Rock	1	2	313	5	0
4	Who Made Who	1	2	207	5	0

https://en.wikipedia.org/w

we see this in our tables where we have genre_id or album_id and we have

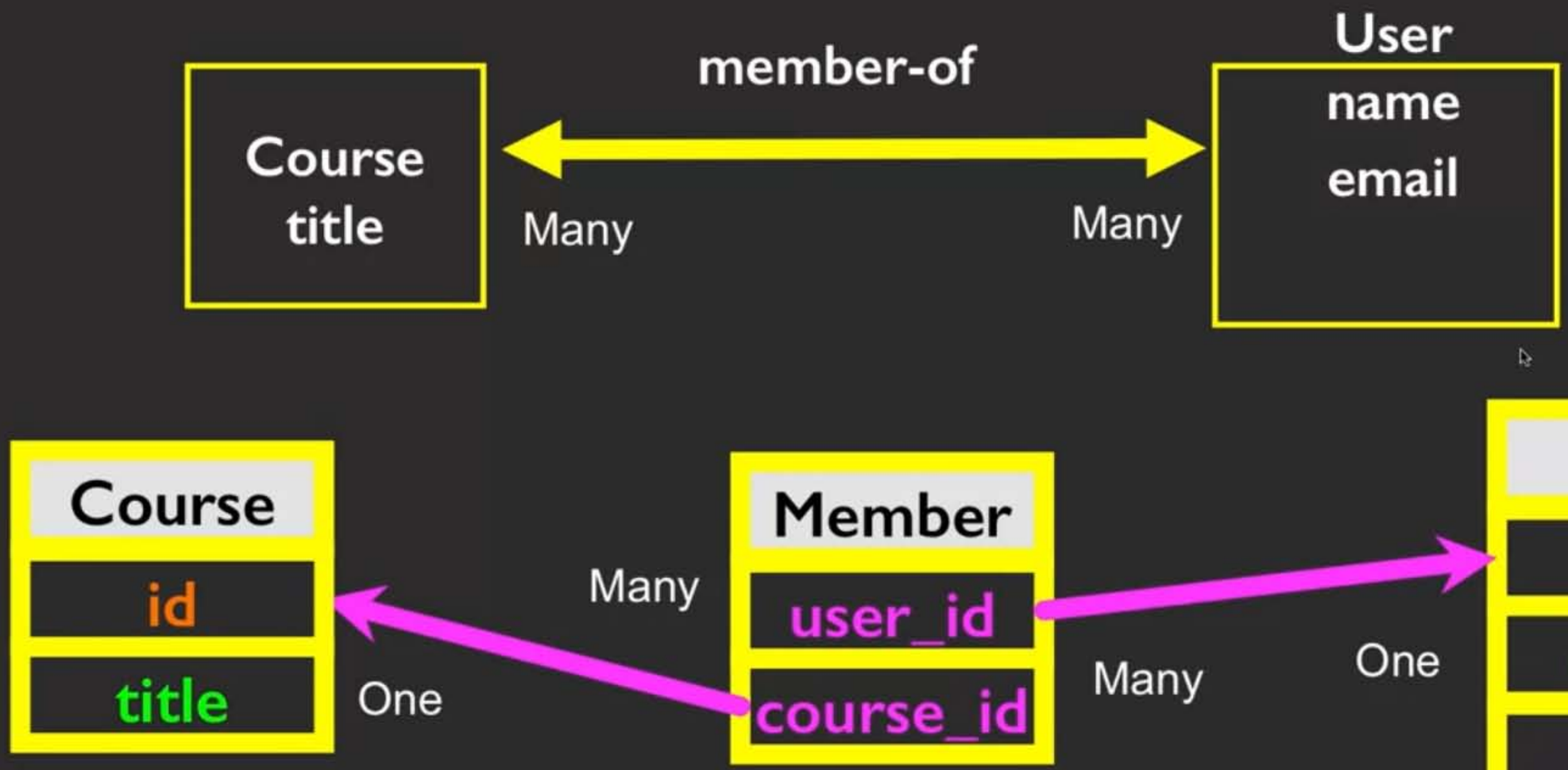
Many to Many

- Sometimes we need to model a relationship that is many-to-many
- We need to add a "connection" table with two foreign keys
- There is usually no separate primary key

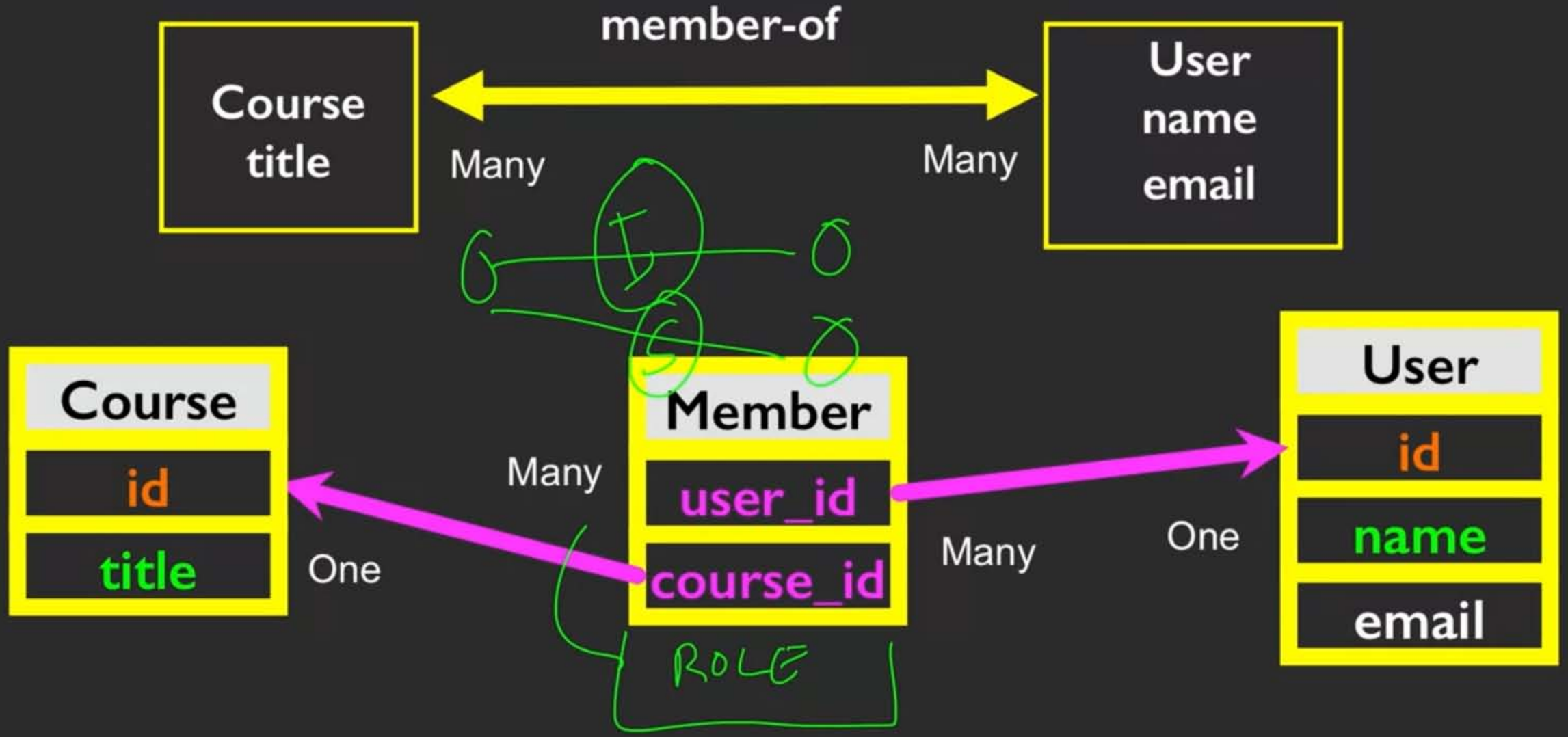


https://en.wikipedia.org/wiki/Many-to-many_(data_model)

04.01 Many-to-Many Relationships in SQL: Many-to-Many



https://en.wikipedia.org/wiki/One-to-many_(data_model)



https://en.wikipedia.org/wiki/Or

we have a little bit of data that we add down there.

Start with a Fresh Database

```
CREATE TABLE User (
    id      INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT UNIQUE,
    name    TEXT,
    email   TEXT
)

CREATE TABLE Course (
    id      INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT UNIQUE,
    title   TEXT
)

CREATE TABLE Member (
    user_id     INTEGER,
    course_id   INTEGER,
        role        INTEGER,
    PRIMARY KEY (user_id, course_id)
)
```

So we make this little table in the middle, and so here we're going to run

Insert Users and Courses

```
INSERT INTO User (name, email) VALUES ('Jane',
'jane@tsugi.org');
INSERT INTO User (name, email) VALUES ('Ed', 'ed@tsugi.org');
INSERT INTO User (name, email) VALUES ('Sue', 'sue@tsugi.org');

INSERT INTO Course (title) VALUES ('Python');
INSERT INTO Course (title) VALUES ('SQL');
INSERT INTO Course (title) VALUES ('PHP');
```

Insert Memberships

id	name	email
Filter	Filter	Filter
1	Jane	jane@tsugi.org
2	Ed	ed@tsugi.org
3	Sue	sue@tsugi.org

id	title
Filter	Filter
1	Python
2	SQL
3	PHP

```
INSERT INTO Member (user_id, course_id, role) VALUES (1, 1, 1);
INSERT INTO Member (user_id, course_id, role) VALUES (2, 1, 0);
INSERT INTO Member (user_id, course_id, role) VALUES (3, 1, 0);

INSERT INTO Member (user_id, course_id, role) VALUES (1, 2, 0);
INSERT INTO Member (user_id, course_id, role) VALUES (2, 2, 1);

INSERT INTO Member (user_id, course_id, role) VALUES (2, 3, 1);
INSERT INTO Member (user_id, course_id, role) VALUES (3, 3, 0);
```

id	name	email
Filter	Filter	Filter
1	Jane	jane@tsugi.org
2	Ed	ed@tsugi.org
3	Sue	sue@tsugi.org

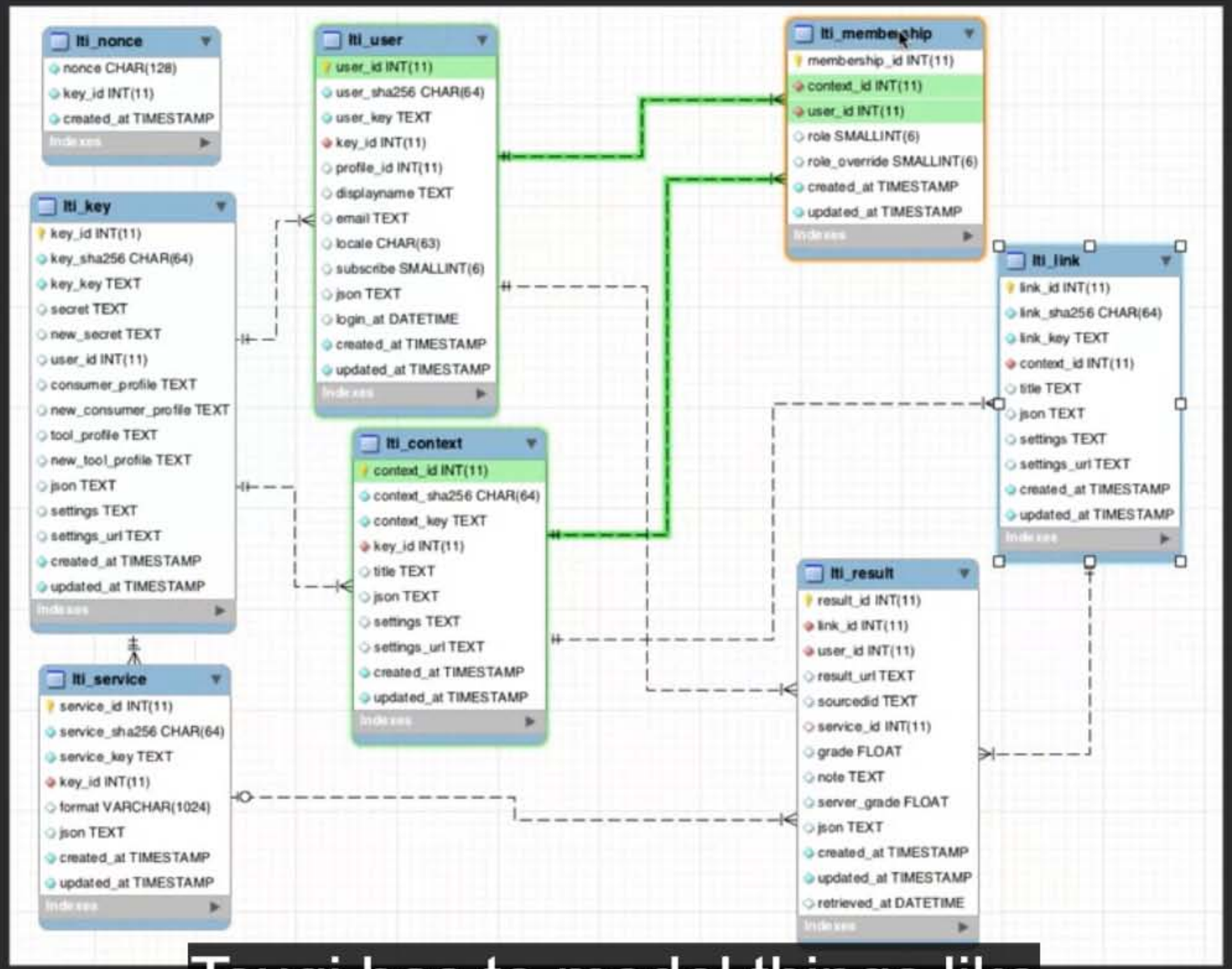
user_id	course_id	role
Filter	Filter	Filter
1	1	1
2	1	0
3	1	0
1	2	0
2	2	1
2	3	1
3	3	0

id	title
Filter	Filter
1	Python
2	SQL
3	PHP

	name	role	title
2	Sue	0	PHP
3	Jane	1	Python
4	Ed	0	Python
5	Sue	0	Python
6	Ed	1	SQL

```
SELECT User.name, Member.role, Course.title
FROM User JOIN Member JOIN Course
ON Member.user_id = User.id AND Member.course_id = Course.id
ORDER BY Course.title, Member.role DESC, User.name
```

in our database, we have now
connected these things together.



Tsugi has to model things like what course you're coming from,