



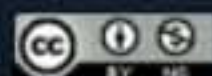
Unit 03.02

## **Data Models and Relational SQL: Implementing a Data Model in Tables**

### **USING DATABASES WITH PYTHON**

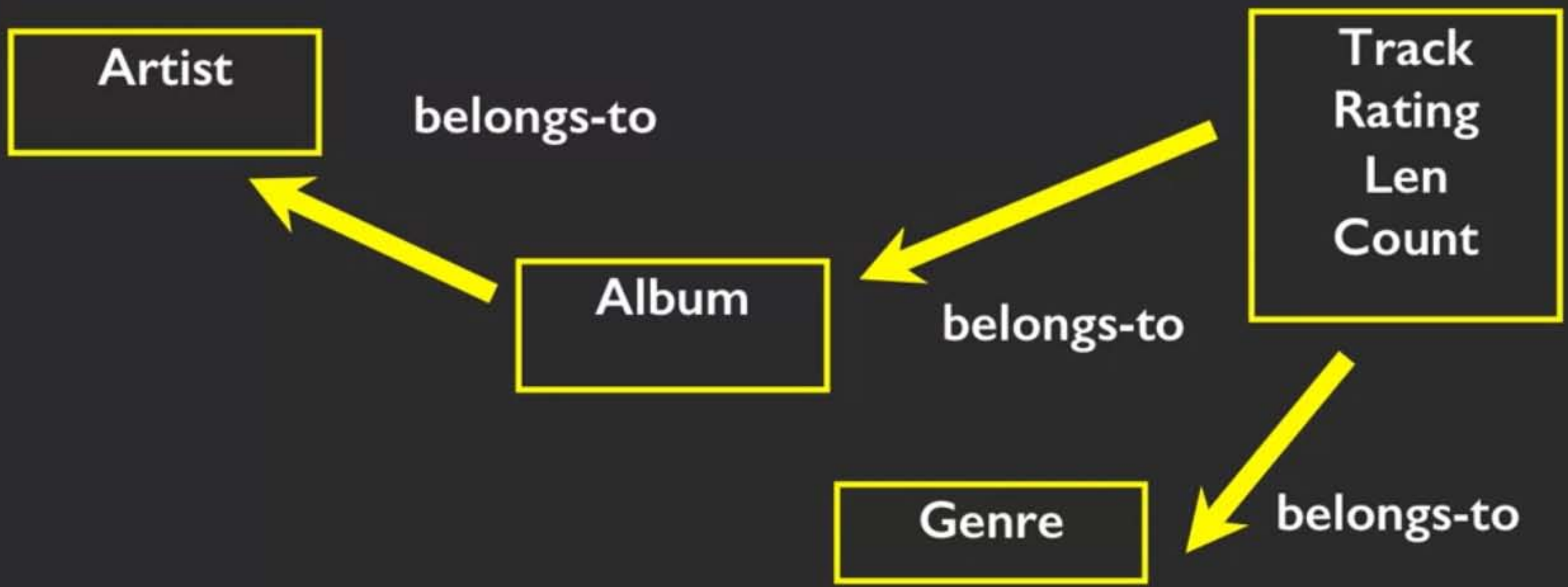
Charles Severance Ph.D.

Clinical Associate Professor of Information  
School of Information



© 2015 Charles Severance and The Regents of the University of Michigan  
Except where otherwise noted, this work is licensed under  
<http://creativecommons.org/licenses/by-nc/3.0/>



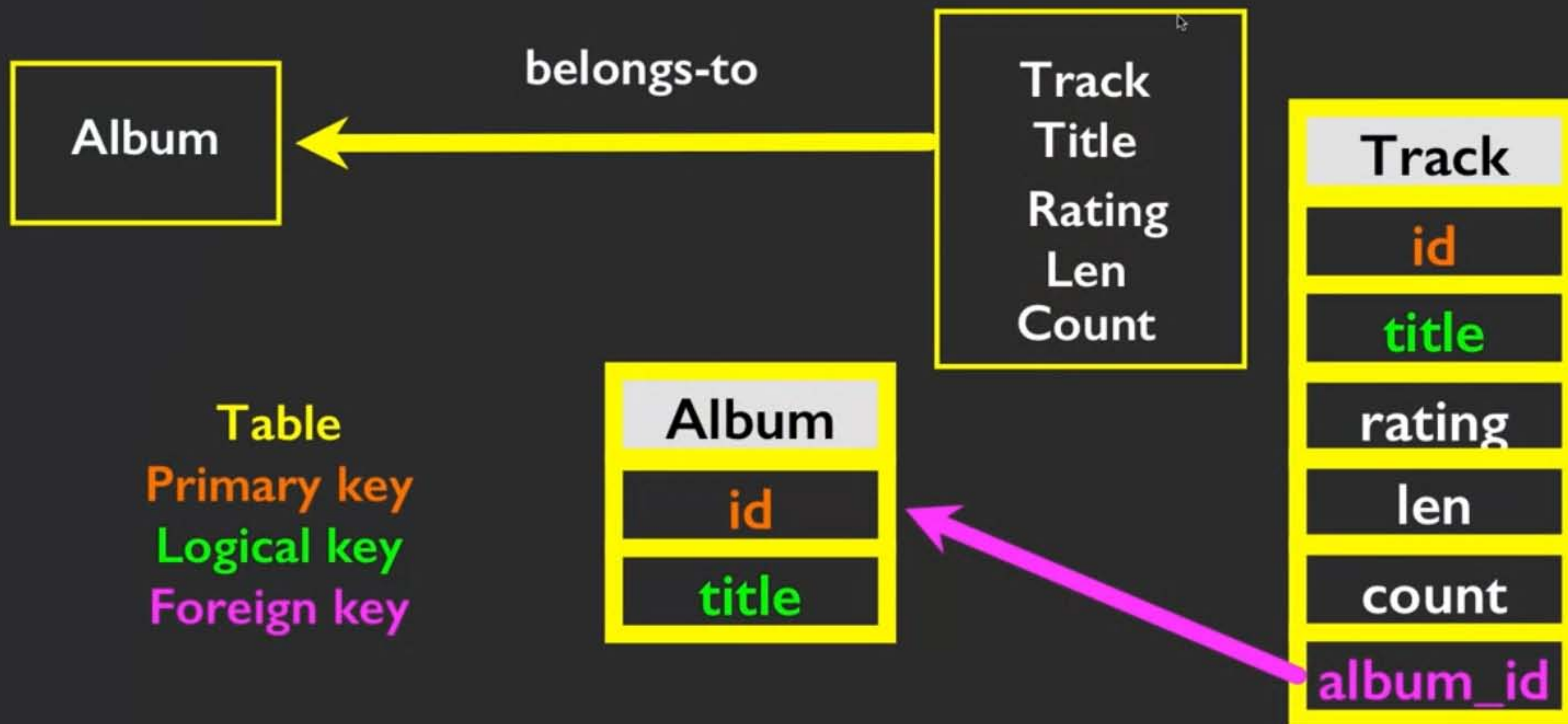


<input checked="" type="checkbox"/> Hells Bells	5:13	AC/DC	Who Made Who	Rock	★★★★★	61
<input checked="" type="checkbox"/> Shake Your Foundations	3:54	AC/DC	Who Made Who	Rock	★★★★★	70
<input checked="" type="checkbox"/> Chase the Ace	3:01	AC/DC	Who Made Who	Rock		56
<input checked="" type="checkbox"/> For Those About To Rock (We ...	5:54	AC/DC	Who Made Who	Rock	★★★★★	61
<input checked="" type="checkbox"/> Dúlamán	3:43	Altan	Natural Wonders M...	New Age		31
<input checked="" type="checkbox"/> Rode Across the Desert	4:10	America	Greatest Hits	Easy Listen...	★★★★★	23
<input checked="" type="checkbox"/> Now You Are Gone	3:08	America	Greatest Hits	Easy Listen...	★★★★★	18
<input checked="" type="checkbox"/> Tin Man					★★★★★	22

This is now how databases represent data. We have to be a little more explicit. There's

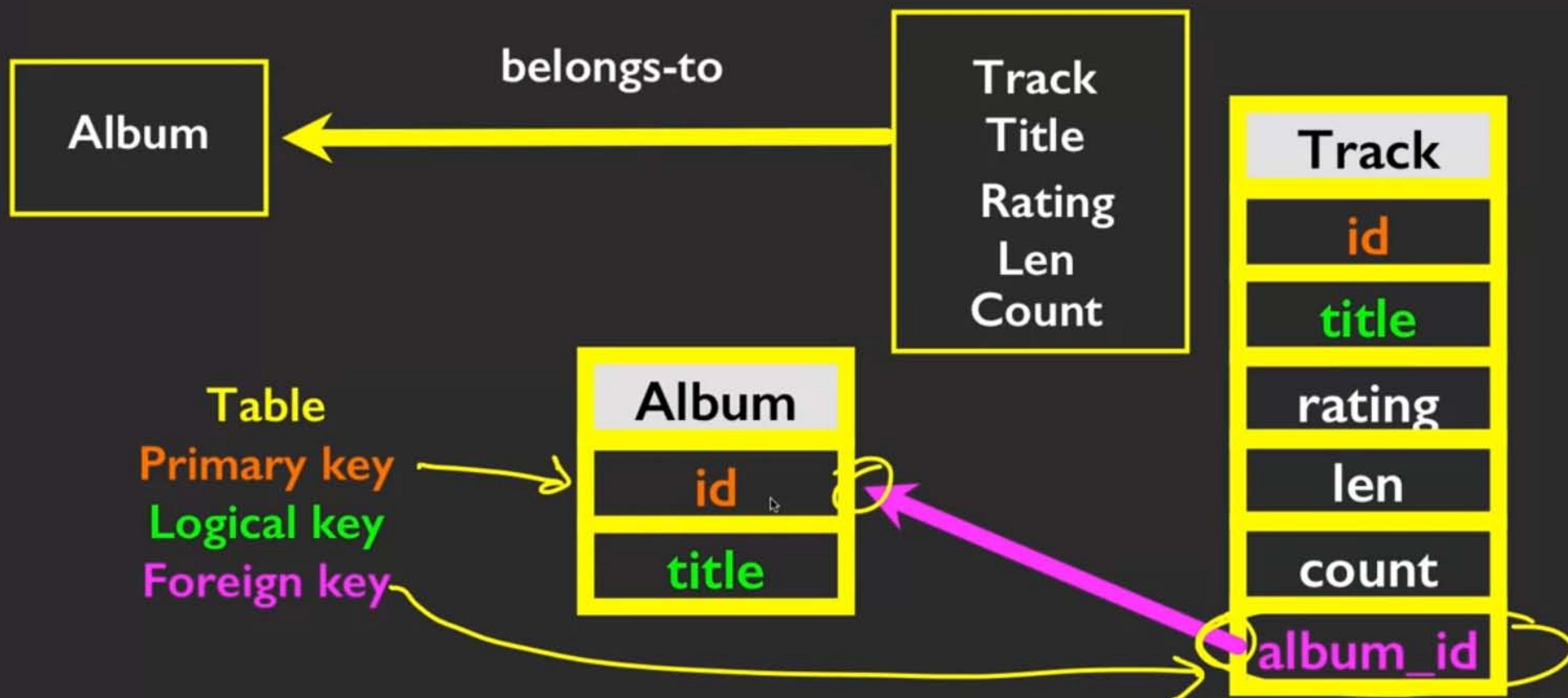


## 03.02 Data Models and Relational SQL: Implementing a Data Model in Tables

USING DATABASES  
WITH PYTHON

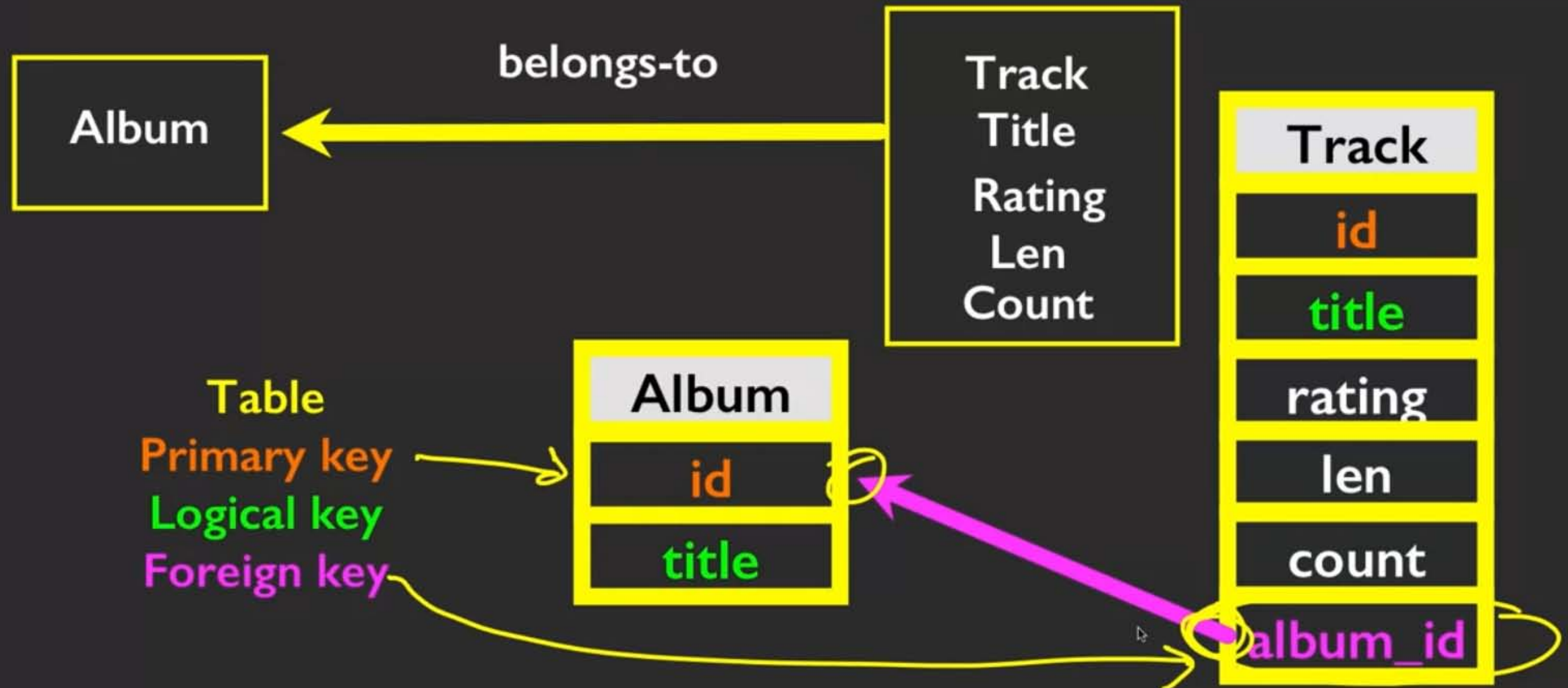
And so, here's the track table, and so we'll make a table called Track.





The primary key's the ending point of the arrow and





the foreign key is the starting point of the arrow.

The logical key is that



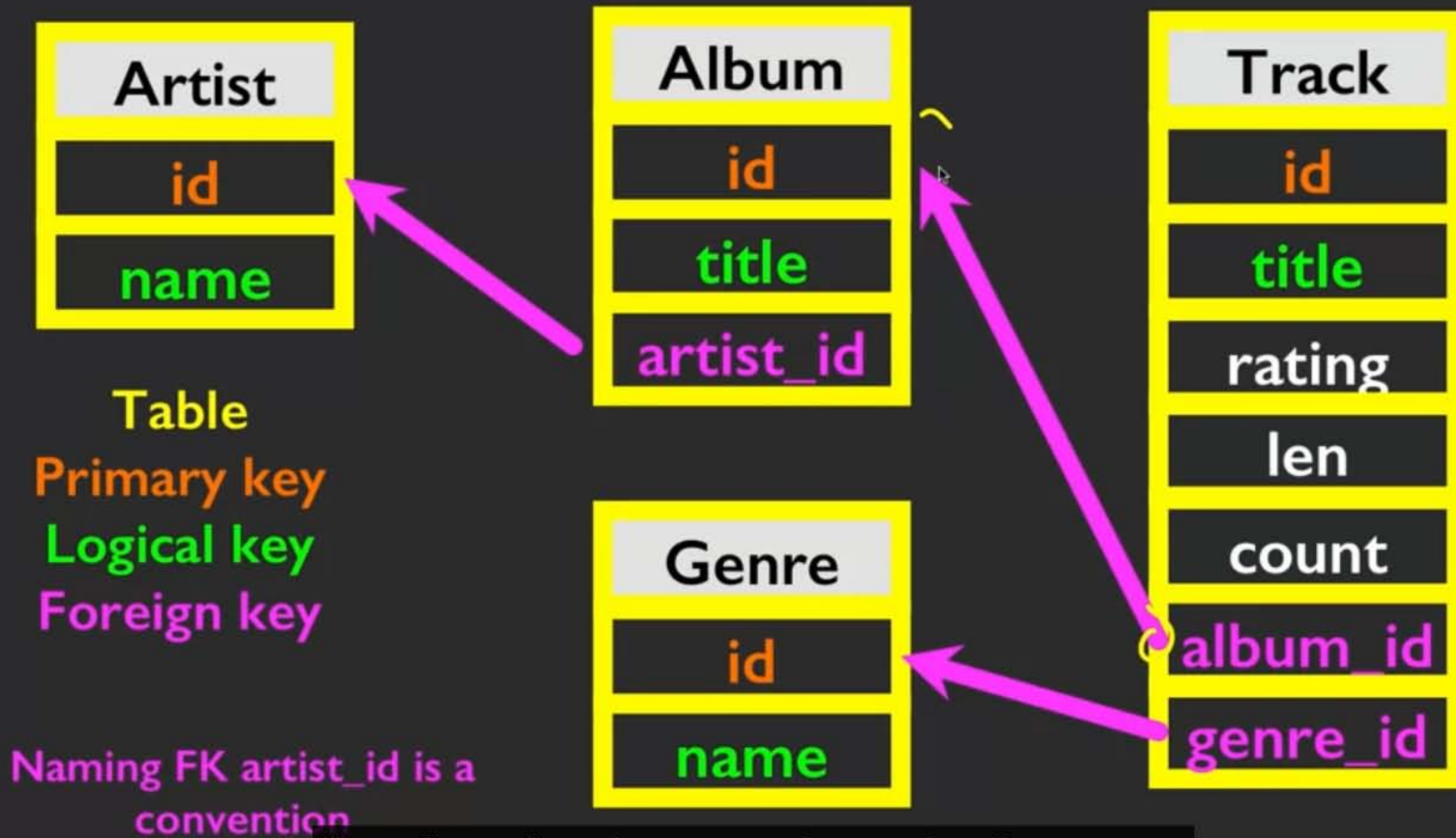




unique thing that we might use to look  
up this row from the outside world.

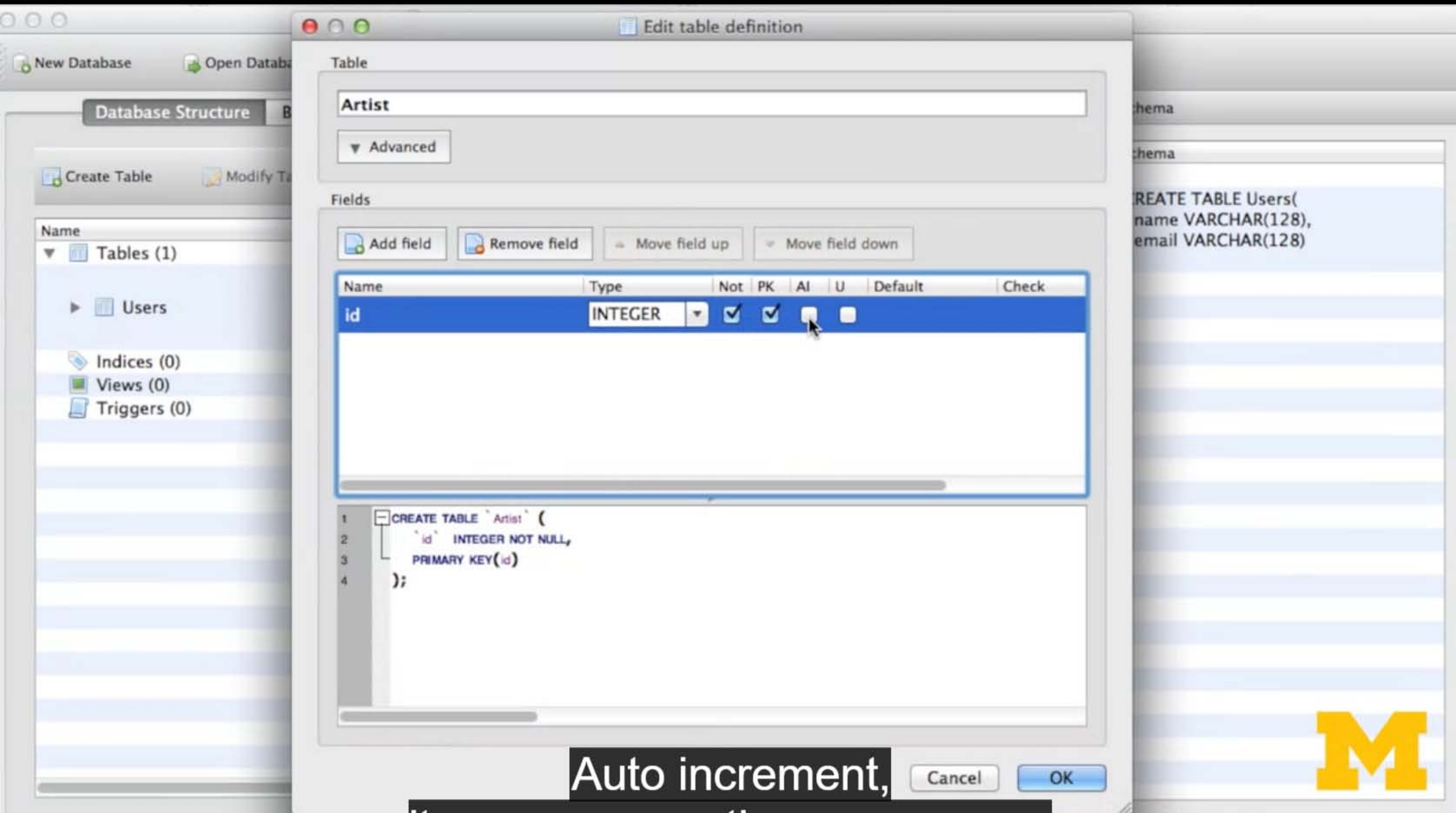






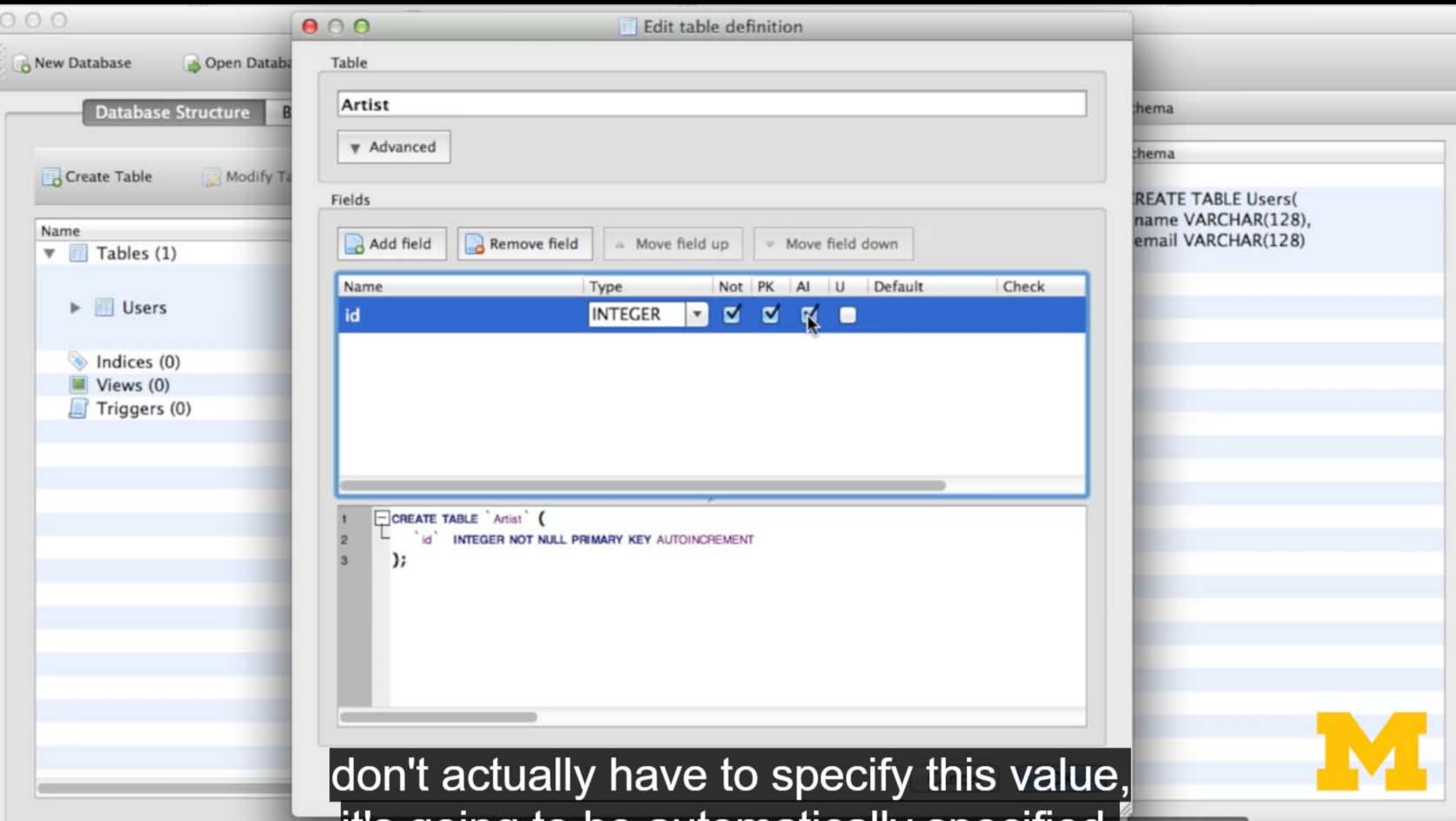
the beginning and end of arrow,  
beginning and end of arrows, beginning





Auto increment,  
it means we as the programmer

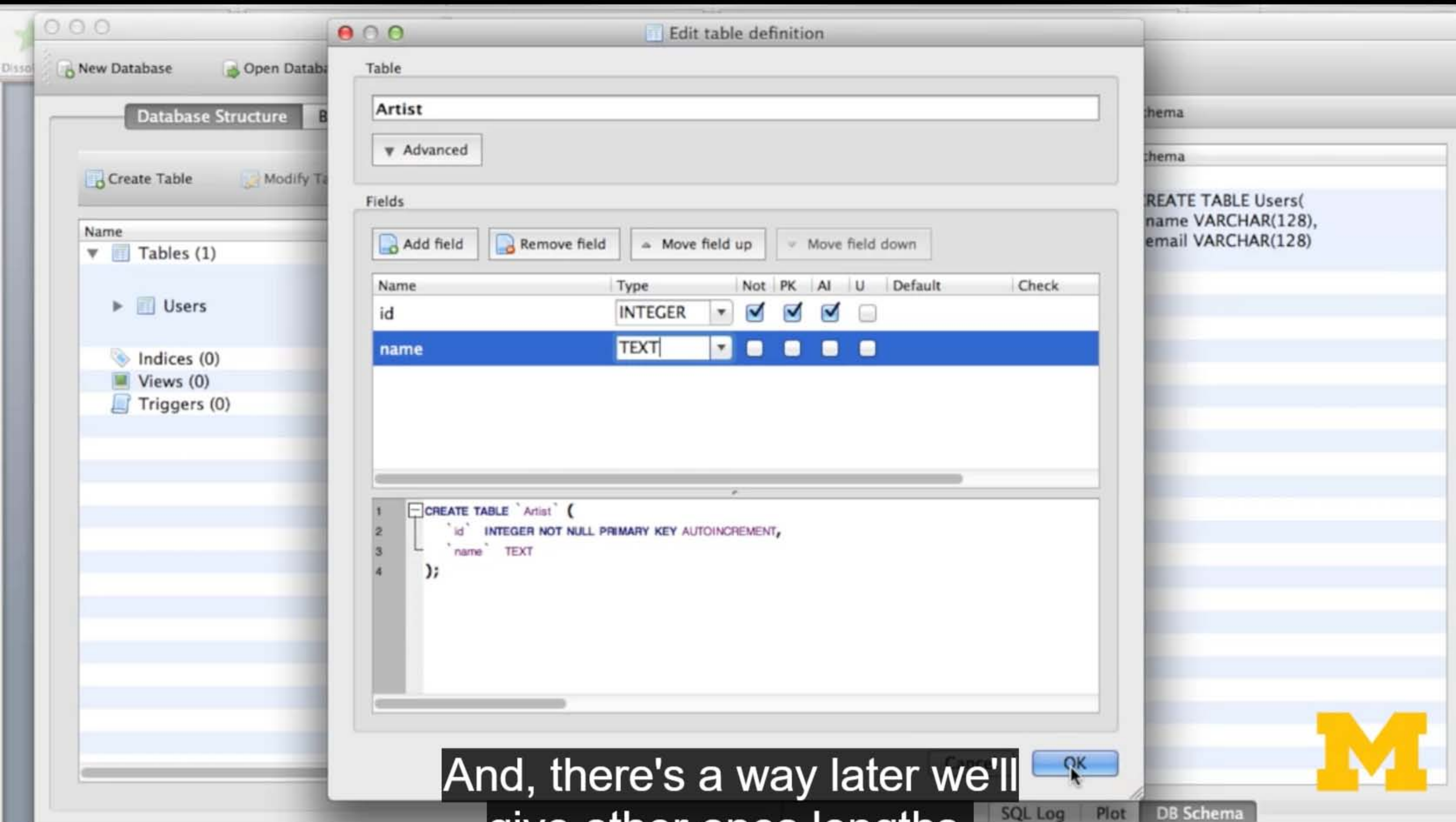




don't actually have to specify this value,  
it's going to be automatically specified.





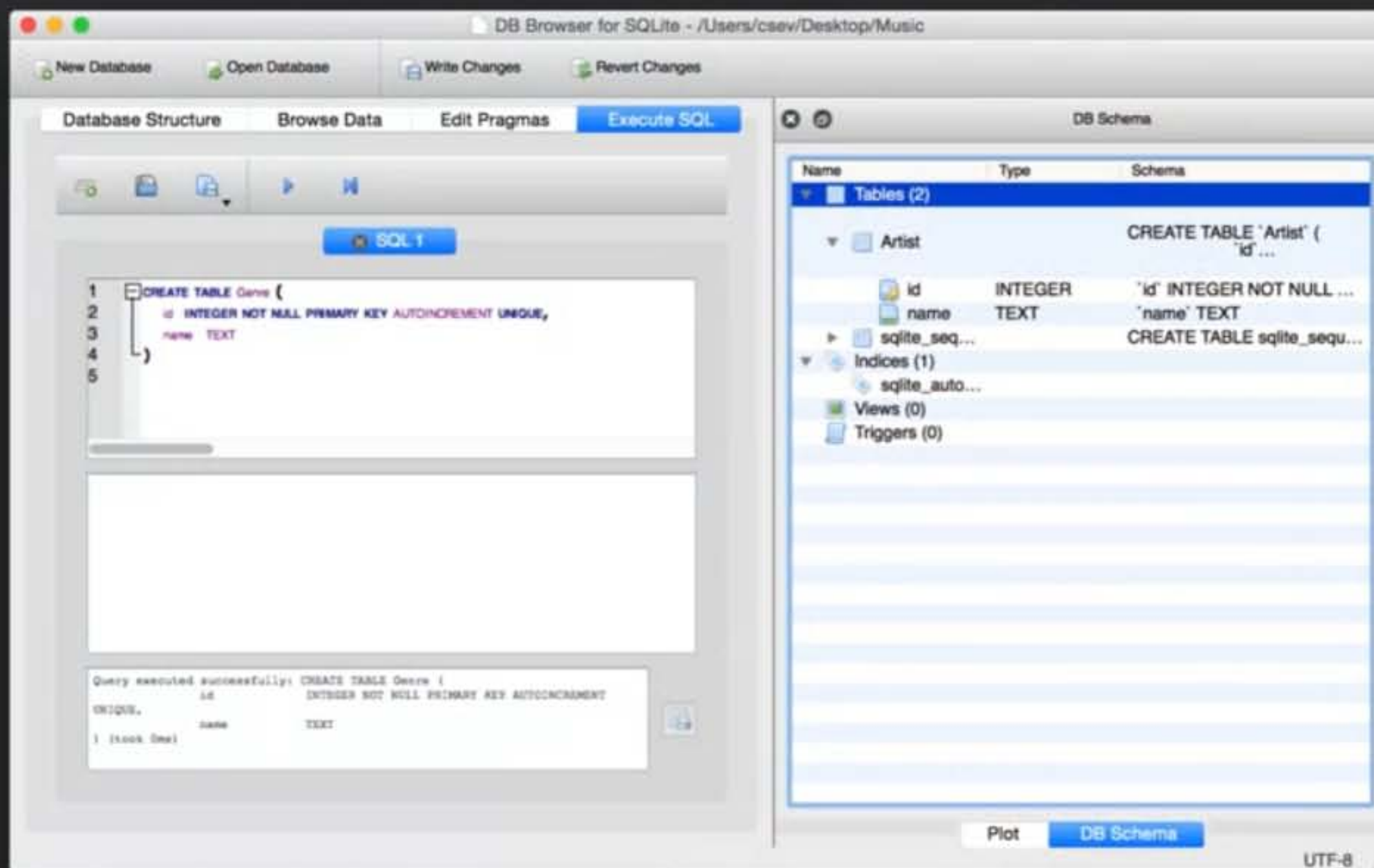


And, there's a way later we'll  
give other ones lengths.





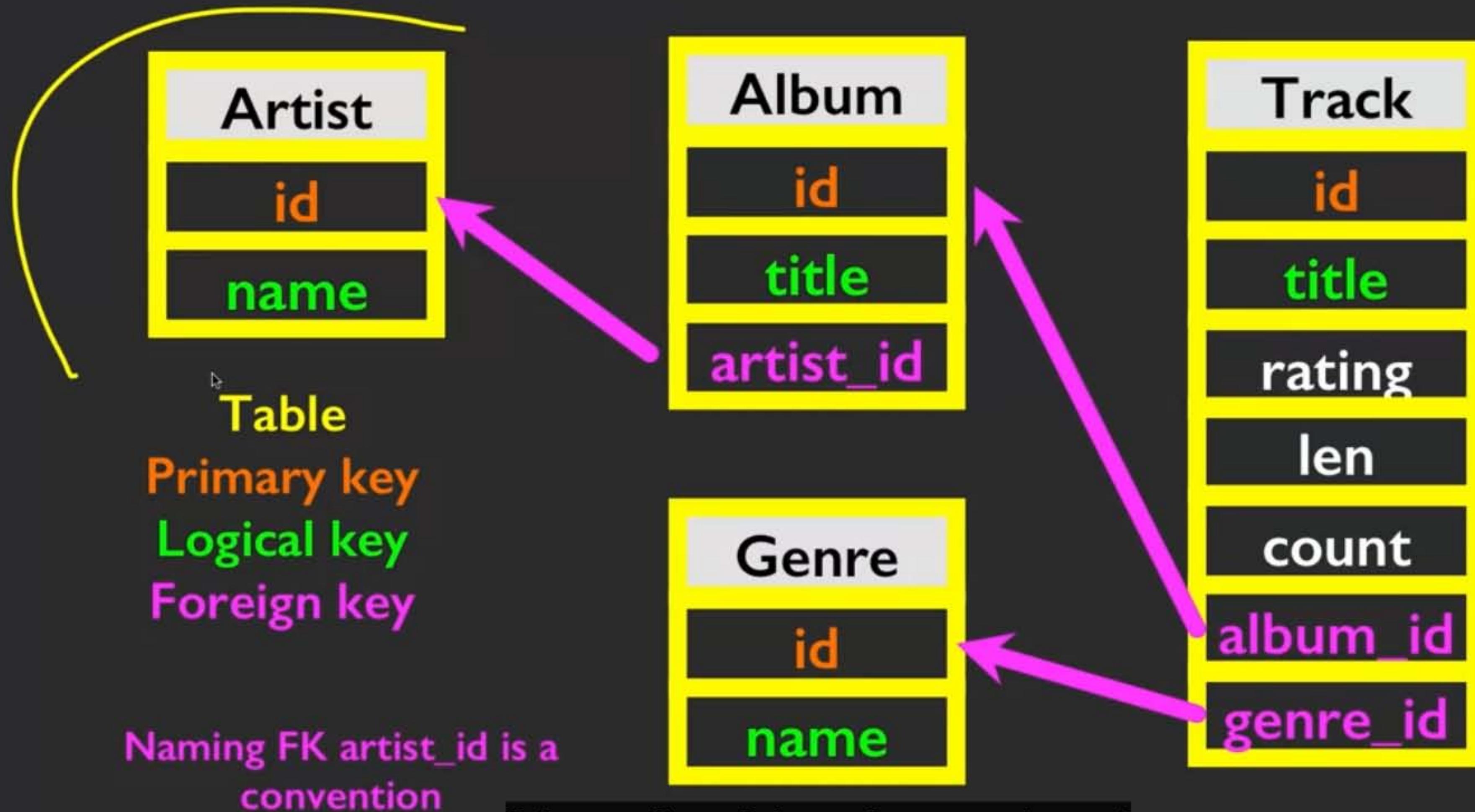
## 03.02 Data Models and Relational SQL: Implementing a Data Model in Tables



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

The next table we're going to create is the Genre table.





Now the idea is we tend to work from outward in.



```
CREATE TABLE Album (  
    id          INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT UNIQUE,  
    artist_id   INTEGER,  
    title       TEXT  
)
```

```
CREATE TABLE Track (  
    id          INTEGER NOT NULL PRIMARY KEY  
                AUTOINCREMENT UNIQUE,  
    title       TEXT,  
    album_id   INTEGER,  
    genre_id   INTEGER,  
    len        INTEGER, rating INTEGER, count INTEGER  
)
```

is that the Album table is our  
first example of a foreign key.



New Database Open Database Write Changes Revert Changes

Database Structure Browse Data Edit Pragmas Execute SQL

Table: Artist New Record Delete Record

id	name
Filter	Filter

0 - 0 of 0 Go to: 1

DB Schema

Name	Type	Schema
▶ Album	CREATE TABLE Album (	id INTEGER ...
▶ Artist	CREATE TABLE `Artist` (	`id` INTEG...
▶ Genre	CREATE TABLE Genre (	id INTEGER ...
▶ Track	CREATE TABLE Track (	id INTEGER ...
▶ sqlite_sequ...	CREATE TABLE sqlite_sequence(nam...	
▼ Indices (3)		
▶ sqlite_autoi...		
▶ sqlite_autoi...		
▶ sqlite_autoi...		
Views (0)		
Triggers (0)		

SQL Log Plot DB Schema

UTF-8

we got all of them done.  
Album is now complex.

