Course   Progress   Dates   Discussion   Instructional Team   Office Hours

Course / Lesson 3: Relational Databases / Lesson 3 Content

Previous   Next
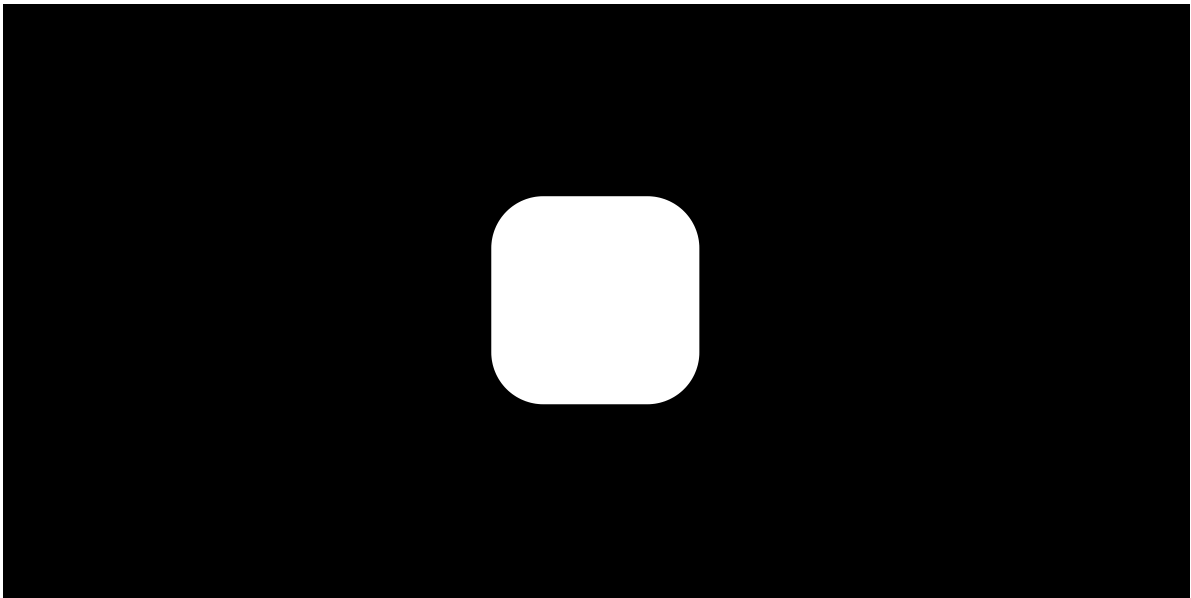
Part 4: Data Design

Bookmark this page

Part 4: Data Design

Now we are going to learn about the best way to design the data in our database. We will use the diagram in the "stuff" directory, which is also included below, along with the SQL code:

Behind the Scenes

Database Technologies

We've seen that the really cool thing with peewee

is that you almost don't know you're working with a database.

It's all about working in Python, and things are very straightforward.

But with a database, you can always look at the data used in the database
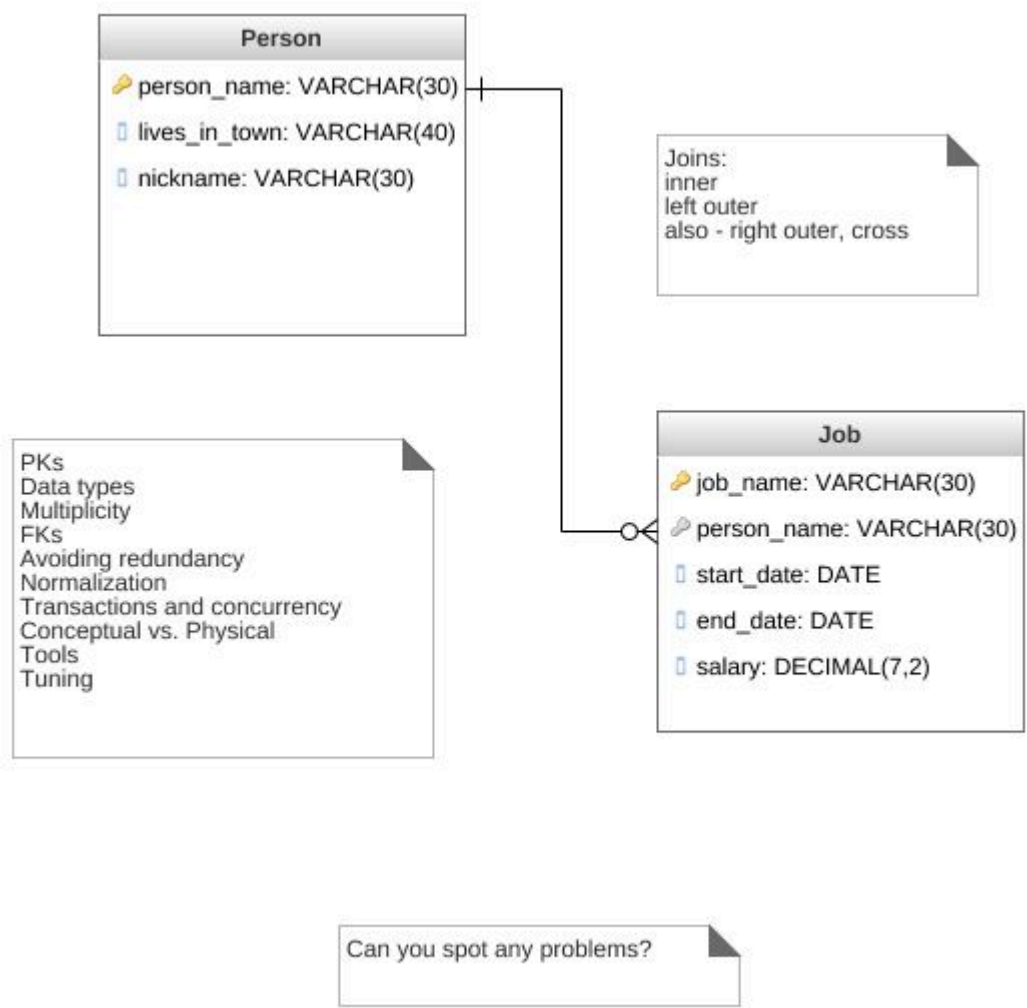
just itself.

Video
Download video file

Transcripts
Download SubRip (.srt) file
Download Text (.txt) file

Database diagram:

Person
- 🔑 person_name: VARCHAR(30)
- ▯ lives_in_town: VARCHAR(40)
- ▯ nickname: VARCHAR(30)

Joins:
inner
left outer
also - right outer, cross

PKs
Data types
Multiplicity
FKs
Avoiding redundancy
Normalization
Transactions and concurrency
Conceptual vs. Physical
Tools
Tuning

Job
- 🔑 job_name: VARCHAR(30)
- 🔑 person_name: VARCHAR(30)
- ▯ start_date: DATE
- ▯ end_date: DATE
- ▯ salary: DECIMAL(7,2)

Can you spot any problems?

Code samples from the video:

SQL statement

```
select * from person ;
```

Start sqlite3 database (from the command line):

```
sqlite3 personjob.db
```

The sqlite> prompt indicates we are ready to enter sqlite commands.

```
sqlite> .tables
job person personnumkey
```

Here is how sqlite sees the schema:

```
sqlite> .schema

CREATE TABLE IF NOT EXISTS "person" ("person_name" VARCHAR(30) NOT NULL PRIMARY KEY,
"lives_in_town" VARCHAR(40) NOT NULL, "nickname" VARCHAR(20));

CREATE TABLE IF NOT EXISTS "job" ("job_name" VARCHAR(30) NOT NULL PRIMARY KEY, "start_date" DATE
NOT NULL, "end_date" DATE NOT NULL, "salary" DECIMAL(7, 2) NOT NULL, "person_employed_id"
VARCHAR(30) NOT NULL, FOREIGN KEY ("person_employed_id") REFERENCES "person" ("person_name"));

CREATE INDEX "job_person_employed_id" ON "job" ("person_employed_id");

CREATE TABLE IF NOT EXISTS "personnumkey" ("id" INTEGER NOT NULL PRIMARY KEY, "person_name"
VARCHAR(30) NOT NULL, "lives_in_town" VARCHAR(40) NOT NULL, "nickname" VARCHAR(20));
```

```
sqlite> .mode column
sqlite> .width 15 15 15 15 15
sqlite> .headers on
```

```
sqlite> select * from person;
sqlite> select * from job;
```
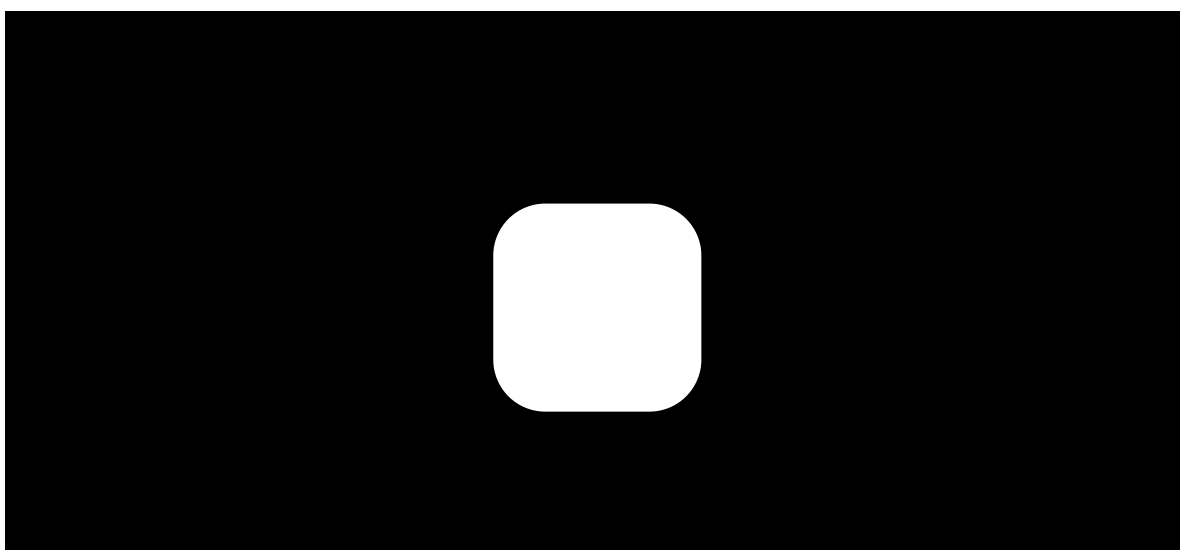
Enter .quit to leave sqlite.

Lesson Summary

In this lesson we have learned about how we define, store and retrieve data in a relational database using Python, Peewee and sqlite.

Conclusion

Summary

So you've seen Peewee in action.

It's actually very easy to be able to save your data

and retrieve it from a relational

and retrieve it from a relational database.

And using classes, it makes everything seem simple and very Pythonic.

Now you'll have a chance to apply those techniques in your assignment.

Video
Download video file

Transcripts
Download SubRip (.srt) file
Download Text (.txt) file

Help Center   Contact Us   Privacy   Terms

Built on OPENedX by RACCOONGANG