add attributes

for each set box in turn
ask: if just a primitive value, will app work well?
if not, enrich by adding attributes

pick primitive types

for each attribute & unenriched set
pick a primitive type to represent it

reverse relations

check each relation
if points from one to many, reverse it

group into tables

break graph into little trees
just one relation deep
write out tables

for each table
root becomes the key of the table
relations to other tables become references
sets without primitive types become ids

```
CREATE TABLE users (
  id int primary key,
  name varchar(20)
)
CREATE TABLE shortcuts (shortcut varchar(10) primary key, owner int references users(id), expandsTo int references urls(id), expires date)
CREATE TABLE urls (id int primary key, domain varchar(50), path varchar(100))
```

tricky case: many to many

what to do about users relation?
turn relation into a table of its own

```
CREATE TABLE accountUsers (account int references account(id), user int references user(id))
```

tricky case: subsets

options
one table for the superclass
one for each subclass
one table for each subclass + one for superclass

```
CREATE TABLE shortcuts (shortcut, owner, expires, expandsTo)
CREATE TABLE permShortcuts (shortcut, owner)
CREATE TABLE expShortcuts (shortcut, expires)
```

tricky case: subsets (ctd)

```
CREATE TABLE shortcuts (shortcut, owner, expires, expandsTo)
CREATE TABLE permShortcuts (shortcut, owner)
CREATE TABLE expShortcuts (shortcut, expires)
```

one table for superclass and each subclass
one table for superclass
one table for each subclass