

```

Last login: Mon Jul 3 12:29:17 on ttys000
→ ~ cd vagrant
→ vagrant vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Machine already provisioned. Run `vagrant provision` or use the `--provision`
==> default: flag to force provisioning. Provisioners marked to run always will still run.
→ vagrant vagrant ssh
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.4.0-75-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

```

```

18 packages can be updated.
5 updates are security updates.

```

```

Last login: Mon Jul 3 19:30:20 2017 from 10.0.2.2
vagrant@vagrant:~$ cd src/whalz
vagrant@vagrant:~/src/whalz$ source env/bin/activate
(env) vagrant@vagrant:~/src/whalz$ iConnection to 127.0.0.1 closed by remote host.
Connection to 127.0.0.1 closed.
→ vagrant ssh
usage: ssh [-l246AaCfGgKkMnQsTtVvXxYy] [-b bind_address] [-c cipher_spec]
          [-D [bind_address:]port] [-E log_file] [-e escape_char]
          [-F configfile] [-I pkcs11] [-i identity_file]
          [-L [bind_address:]port:host:hostport] [-l login_name] [-m mac_spec]
          [-O ctl_cmd] [-o option] [-p port]
          [-Q cipher | cipher-auth | mac | kex | key]
          [-R [bind_address:]port:host:hostport] [-S ctl_path] [-W host:port]
          [-w local_tun[:remote_tun]] [user@]hostname [command]
→ vagrant cd src/whalz
cd: no such file or directory: src/whalz
→ vagrant vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Resuming suspended VM...
==> default: Booting VM...
==> default: Waiting for machine to boot. This may take a few minutes...
default: SSH address: 127.0.0.1:2222
default: SSH username: vagrant
default: SSH auth method: private key
==> default: Machine booted and ready!
==> default: Machine already provisioned. Run `vagrant provision` or use the `--provision`
==> default: flag to force provisioning. Provisioners marked to run always will still run.
→ vagrant vagrant ssh
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.4.0-75-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

```

```

18 packages can be updated.
5 updates are security updates.

```

```

Last login: Mon Jul 3 19:54:59 2017 from 10.0.2.2
vagrant@vagrant:~$ cd src/whalz
vagrant@vagrant:~/src/whalz$ source env/bin/activate
(env) vagrant@vagrant:~/src/whalz$ python -i model.py
Connected to DB.
>>> trade_possibilites = db.session.query(Inventory.user.name
...                                     Inventory.inventory_id,
...                                     File "<stdin>", line 2
...                                     ^
...                                     Inventory.inventory_id,
SyntaxError: invalid syntax
>>>                                     Inventory.beer_code,
...                                     File "<stdin>", line 1
...                                     ^
...                                     Inventory.beer_code,
IndentationError: unexpected indent
>>>                                     ISO.user.name
...                                     File "<stdin>", line 1
...                                     ^
...                                     ISO.user.name
IndentationError: unexpected indent
>>>                                     ISO.iso_id,
...                                     File "<stdin>", line 1
...                                     ^
...                                     ISO.iso_id,
IndentationError: unexpected indent
>>>                                     ISO.beer_code).join(ISO).using(beer_code)
...                                     File "<stdin>", line 1
...                                     ^
...                                     ISO.beer_code).join(ISO).using(beer_code)
IndentationError: unexpected indent
>>> trade_possibilites = db.session.query(Inventory.user.name,Inventory.inventory_id,Inventory.beer_code,ISO.user.name,ISO.iso_id,ISO.beer_code).join(ISO).using(beer_c
ode)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
    File "/home/vagrant/src/whalz/env/local/lib/python2.7/site-packages/sqlalchemy/orm/attributes.py", line 193, in __getattr__
    key)
AttributeError: Neither 'InstrumentedAttribute' object nor 'Comparator' object associated with Inventory.user has an attribute 'name'
>>> trade_possibilites = db.session.query(Inventory.name,Inventory.inventory_id,Inventory.beer_code,ISO.name,ISO.iso_id,ISO.beer_code).join(ISO).using(beer_code)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: type object 'Inventory' has no attribute 'name'
>>> dir(Inventory)
['_class_', '_delattr_', '_dict_', '_doc_', '_format_', '_getattr_', '_hash_', '_init_', '_mapper_', '_module_', '_new_', '_reduce_', '_r
educer_ex_', '_repr_', '_setattr_', '_sizeof_', '_str_', '_subclasshook_', '_table_', '_tablename_', '_weakref_', '_cached_table', '_decl_class_reg
istry_', '_sa_class_manager', 'beer', 'beer_code', 'inventory_id', 'inventory_number', 'metadata', 'quantity', 'query', 'query_class', 'user', 'user_id']
>>> Inventory.user.name
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
    File "/home/vagrant/src/whalz/env/local/lib/python2.7/site-packages/sqlalchemy/orm/attributes.py", line 193, in __getattr__
    key)
AttributeError: Neither 'InstrumentedAttribute' object nor 'Comparator' object associated with Inventory.user has an attribute 'name'
>>> Inventory.user
<sqlalchemy.orm.attributes.InstrumentedAttribute object at 0x7f5febca8e90>
>>> Inventory.user(name)

```

```

Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'name' is not defined
>>> dir(Inventory.user)
['_add_', '_and_', '_class_', '_clause_element_', '_contains_', '_delattr_', '_delete_', '_dict_', '_div_', '_doc_', '_eq_', '_format_', '_ge_', '_get_', '_getattr_', '_getattribute_', '_getitem_', '_gt_', '_hash_', '_init_', '_invert_', '_le_', '_lshift_', '_lt_', '_mod_', '_module_', '_mul_', '_ne_', '_neg_', '_new_', '_or_', '_radd_', '_rddiv_', '_reduce_', '_reduce_ex_', '_repr_', '_rmod_', '_rmul_', '_rshift_', '_rsub_', '_rtruediv_', '_selectable_', '_set_', '_setattr_', '_sizeof_', '_slots_', '_str_', '_sub_', '_subclasshook_', '_truediv_', '_weakref_', '_adapt_to_entity_', '_of_type_', '_parententity_', '_parentmapper_', '_query_clause_element_', '_supports_population_', '_adapt_to_entity_', '_adapter_', '_all_', '_any_', '_any_', '_any_op_', '_asc_', '_between_', '_class_', '_collate_', '_comparator_', '_concat_', '_contains_', '_desc_', '_dispatch_', '_distinct_', '_endswith_', '_expression_', '_extension_type_', '_get_history_', '_has_', '_has_op_', '_hasparent_', '_ilike_', '_impl_', '_in_', '_info_', '_is_', '_is_alias_class_', '_is_attribute_', '_is_clause_element_', '_is_distinct_from_', '_is_instance_', '_is_mapper_', '_is_property_', '_is_selectable_', '_isnot_', '_isnot_distinct_from_', '_key_', '_label_', '_like_', '_match_', '_notilike_', '_notin_', '_notlike_', '_nullsfirst_', '_nulls_last_', '_of_type_', '_of_type_op_', '_op_', '_operate_', '_parent_', '_prop_', '_property_', '_reverse_operate_', '_startswith_', '_timetuple_']
>>> Inventory.User.name
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: type object 'Inventory' has no attribute 'User'
>>> Inventory.query.get(1)
2017-07-03 20:10:10,838 INFO sqlalchemy.engine.base.Engine select version()
2017-07-03 20:10:10,839 INFO sqlalchemy.engine.base.Engine {}
2017-07-03 20:10:10,840 INFO sqlalchemy.engine.base.Engine select current_schema()
2017-07-03 20:10:10,841 INFO sqlalchemy.engine.base.Engine {}
2017-07-03 20:10:10,842 INFO sqlalchemy.engine.base.Engine SELECT CAST('test plain returns' AS VARCHAR(60)) AS anon_1
2017-07-03 20:10:10,842 INFO sqlalchemy.engine.base.Engine {}
2017-07-03 20:10:10,843 INFO sqlalchemy.engine.base.Engine SELECT CAST('test unicode returns' AS VARCHAR(60)) AS anon_1
2017-07-03 20:10:10,843 INFO sqlalchemy.engine.base.Engine {}
2017-07-03 20:10:10,844 INFO sqlalchemy.engine.base.Engine show standard_conforming_strings
2017-07-03 20:10:10,844 INFO sqlalchemy.engine.base.Engine {}
2017-07-03 20:10:10,845 INFO sqlalchemy.engine.base.Engine BEGIN (implicit)
2017-07-03 20:10:10,846 INFO sqlalchemy.engine.base.Engine SELECT inventories.inventory_id AS inventories_inventory_id, inventories.beer_code AS inventories_beer_code, inventories.user_id AS inventories_user_id, inventories.quantity AS inventories_quantity
FROM inventories
WHERE inventories.inventory_id = %(param_1)s
2017-07-03 20:10:10,847 INFO sqlalchemy.engine.base.Engine {'param_1': 1}
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "model.py", line 93, in __repr__
    self.inventory_id, self.beer_code, self.user_id, self.inventory)
AttributeError: 'Inventory' object has no attribute 'inventory'
>>> Inventory.query.all()
2017-07-03 20:10:57,285 INFO sqlalchemy.engine.base.Engine SELECT inventories.inventory_id AS inventories_inventory_id, inventories.beer_code AS inventories_beer_code, inventories.user_id AS inventories_user_id, inventories.quantity AS inventories_quantity
FROM inventories
2017-07-03 20:10:57,285 INFO sqlalchemy.engine.base.Engine {}
[Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "model.py", line 93, in __repr__
    self.inventory_id, self.beer_code, self.user_id, self.inventory)
AttributeError: 'Inventory' object has no attribute 'inventory'
>>> trial = User.query.get(1)
2017-07-03 20:12:25,846 INFO sqlalchemy.engine.base.Engine SELECT users.user_id AS users_user_id, users.name AS users_name, users.email AS users_email, users.password AS users_password
FROM users
WHERE users.user_id = %(param_1)s
2017-07-03 20:12:25,846 INFO sqlalchemy.engine.base.Engine {'param_1': 1}
>>> User.name
<sqlalchemy.orm.attributes.InstrumentedAttribute object at 0x7f5febcb98f0>
>>> name = User.name
>>> name
<sqlalchemy.orm.attributes.InstrumentedAttribute object at 0x7f5febcb98f0>
>>> trial.name
u'Robert Sims'
>>> trial = Inventory.query.get(1)
>>> trial.user.name
2017-07-03 20:16:08,181 INFO sqlalchemy.engine.base.Engine SELECT users.user_id AS users_user_id, users.name AS users_name, users.email AS users_email, users.password AS users_password
FROM users
WHERE users.user_id = %(param_1)s
2017-07-03 20:16:08,181 INFO sqlalchemy.engine.base.Engine {'param_1': 50}
u'Claire Cruz'
>>> trade_possibilities = db.session.query(Inventory.inventory_id, Inventory.beer_code, ISO.iso_id, ISO.beer_code).join(ISO).using(beer_code)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "/home/vagrant/src/whalز/env/local/lib/python2.7/site-packages/sqlalchemy/orm/query.py", line 2010, in join
    from_joinpoint=from_joinpoint)
  File "<string>", line 2, in _join
  File "/home/vagrant/src/whalز/env/local/lib/python2.7/site-packages/sqlalchemy/orm/base.py", line 201, in generate
    fn(self, *args[1:], **kw)
  File "/home/vagrant/src/whalز/env/local/lib/python2.7/site-packages/sqlalchemy/orm/query.py", line 2154, in _join
    outerjoin, full, create_aliases, prop)
  File "/home/vagrant/src/whalز/env/local/lib/python2.7/site-packages/sqlalchemy/orm/query.py", line 2227, in _join_left_to_right
    self._join_to_left(l_info, left, right, onclause, outerjoin, full)
  File "/home/vagrant/src/whalز/env/local/lib/python2.7/site-packages/sqlalchemy/orm/query.py", line 2356, in _join_to_left
    "Tried joining to %s, but got: %s" % (right, ae))
sqlalchemy.exc.InvalidRequestError: Could not find a FROM clause to join from. Tried joining to <class '__main__.ISO'>, but got: Can't find any foreign key relationships between 'inventories' and 'iso'.
>>> trade_possibilities = db.session.query(Inventory.inventory_id, Inventory.beer_code, ISO.iso_id, ISO.beer_code).join(ISO).all()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "/home/vagrant/src/whalز/env/local/lib/python2.7/site-packages/sqlalchemy/orm/query.py", line 2010, in join
    from_joinpoint=from_joinpoint)
  File "<string>", line 2, in _join
  File "/home/vagrant/src/whalز/env/local/lib/python2.7/site-packages/sqlalchemy/orm/base.py", line 201, in generate
    fn(self, *args[1:], **kw)
  File "/home/vagrant/src/whalز/env/local/lib/python2.7/site-packages/sqlalchemy/orm/query.py", line 2154, in _join
    outerjoin, full, create_aliases, prop)
  File "/home/vagrant/src/whalز/env/local/lib/python2.7/site-packages/sqlalchemy/orm/query.py", line 2227, in _join_left_to_right
    self._join_to_left(l_info, left, right, onclause, outerjoin, full)
  File "/home/vagrant/src/whalز/env/local/lib/python2.7/site-packages/sqlalchemy/orm/query.py", line 2356, in _join_to_left
    "Tried joining to %s, but got: %s" % (right, ae))
sqlalchemy.exc.InvalidRequestError: Could not find a FROM clause to join from. Tried joining to <class '__main__.ISO'>, but got: Can't find any foreign key relationships between 'inventories' and 'iso'.
>>> trade_possibilities = Inventory.query.options(db.joinedload('ISO')).all()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "/home/vagrant/src/whalز/env/local/lib/python2.7/site-packages/sqlalchemy/orm/query.py", line 1324, in options
    return self._options(False, *args)
  File "<string>", line 2, in _options
  File "/home/vagrant/src/whalز/env/local/lib/python2.7/site-packages/sqlalchemy/orm/base.py", line 201, in generate
    fn(self, *args[1:], **kw)

```

```

File "/home/vagrant/src/whalz/env/local/lib/python2.7/site-packages/sqlalchemy/orm/query.py", line 1341, in _options
    opt.process_query(self)
File "/home/vagrant/src/whalz/env/local/lib/python2.7/site-packages/sqlalchemy/orm/strategy_options.py", line 106, in process_query
    self._process(query, True)
File "/home/vagrant/src/whalz/env/local/lib/python2.7/site-packages/sqlalchemy/orm/strategy_options.py", line 337, in _process
    val._bind_loader(query, query._attributes, raiseerr)
File "/home/vagrant/src/whalz/env/local/lib/python2.7/site-packages/sqlalchemy/orm/strategy_options.py", line 430, in _bind_loader
    loader.path, token, None, raiseerr)
File "/home/vagrant/src/whalz/env/local/lib/python2.7/site-packages/sqlalchemy/orm/strategy_options.py", line 151, in _generate_path
    attr, path.entity)
sqlalchemy.exc.ArgumentError: Can't find property named 'ISO' on the mapped entity Mapper[Inventory|inventories in this Query.
>>> trade_possibilites = db.session.query(Inventory.inventory_id, ISO.iso_id).join(ISO).all()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "/home/vagrant/src/whalz/env/local/lib/python2.7/site-packages/sqlalchemy/orm/query.py", line 2010, in join
    from_joingroup=from_joingroup)
  File "<string>", line 2, in _join
  File "/home/vagrant/src/whalz/env/local/lib/python2.7/site-packages/sqlalchemy/orm/base.py", line 201, in generate
    fn(self, *args[1:], **kw)
  File "/home/vagrant/src/whalz/env/local/lib/python2.7/site-packages/sqlalchemy/orm/query.py", line 2154, in _join
    outerjoin, full, create_aliases, prop)
  File "/home/vagrant/src/whalz/env/local/lib/python2.7/site-packages/sqlalchemy/orm/query.py", line 2227, in _join_left_to_right
    self._join_to_left(l_info, left, right, onclause, outerjoin, full)
  File "/home/vagrant/src/whalz/env/local/lib/python2.7/site-packages/sqlalchemy/orm/query.py", line 2356, in _join_to_left
    "Tried joining to %s, but got: %s" % (right, ae))
sqlalchemy.exc.InvalidRequestError: Could not find a FROM clause to join from. Tried joining to <class '__main__.ISO'>, but got: Can't find any foreign key relationships between 'inventories' and 'iso'.
>>> trade_possibilites = db.session.query(Inventory.inventory_id, ISO.iso_id).join(ISO).using(beer_code).all()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "/home/vagrant/src/whalz/env/local/lib/python2.7/site-packages/sqlalchemy/orm/query.py", line 2010, in join
    from_joingroup=from_joingroup)
  File "<string>", line 2, in _join
  File "/home/vagrant/src/whalz/env/local/lib/python2.7/site-packages/sqlalchemy/orm/base.py", line 201, in generate
    fn(self, *args[1:], **kw)
  File "/home/vagrant/src/whalz/env/local/lib/python2.7/site-packages/sqlalchemy/orm/query.py", line 2154, in _join
    outerjoin, full, create_aliases, prop)
  File "/home/vagrant/src/whalz/env/local/lib/python2.7/site-packages/sqlalchemy/orm/query.py", line 2227, in _join_left_to_right
    self._join_to_left(l_info, left, right, onclause, outerjoin, full)
  File "/home/vagrant/src/whalz/env/local/lib/python2.7/site-packages/sqlalchemy/orm/query.py", line 2356, in _join_to_left
    "Tried joining to %s, but got: %s" % (right, ae))
sqlalchemy.exc.InvalidRequestError: Could not find a FROM clause to join from. Tried joining to <class '__main__.ISO'>, but got: Can't find any foreign key relationships between 'inventories' and 'iso'.
>>> trade_possibilites = db.session.query(Inventory.inventory_id, ISO.iso_id).join(ISO).using(beer_code)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "/home/vagrant/src/whalz/env/local/lib/python2.7/site-packages/sqlalchemy/orm/query.py", line 2010, in join
    from_joingroup=from_joingroup)
  File "<string>", line 2, in _join
  File "/home/vagrant/src/whalz/env/local/lib/python2.7/site-packages/sqlalchemy/orm/base.py", line 201, in generate
    fn(self, *args[1:], **kw)
  File "/home/vagrant/src/whalz/env/local/lib/python2.7/site-packages/sqlalchemy/orm/query.py", line 2154, in _join
    outerjoin, full, create_aliases, prop)
  File "/home/vagrant/src/whalz/env/local/lib/python2.7/site-packages/sqlalchemy/orm/query.py", line 2227, in _join_left_to_right
    self._join_to_left(l_info, left, right, onclause, outerjoin, full)
  File "/home/vagrant/src/whalz/env/local/lib/python2.7/site-packages/sqlalchemy/orm/query.py", line 2356, in _join_to_left
    "Tried joining to %s, but got: %s" % (right, ae))
sqlalchemy.exc.InvalidRequestError: Could not find a FROM clause to join from. Tried joining to <class '__main__.ISO'>, but got: Can't find any foreign key relationships between 'inventories' and 'iso'.
>>> trade_possibilites = db.session.query(Inventory.inventory_id, ISO.iso_id).join(ISO).using(beer_code)
...
...
... exit()
... print
File "<stdin>", line 5
print
^
SyntaxError: invalid syntax
>>> exit()
(env) vagrant@vagrant:~/src/whalz$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   seed.py

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        .DS_Store
        api.pyc
        charts.py
        correlation copy.py
        dashboard.py
        day2.py
        docsthem/
        easy.py
        env/
        fetch.py
        model.pyc
        old_server.py
        secrets.sh
        server.pyc
        templates/trade_details.html
        theme-dashboard 2/
        theme-dashboard.zip
        theme-dashboard/
        tryrandom.py

no changes added to commit (use "git add" and/or "git commit -a")
(env) vagrant@vagrant:~/src/whalz$ git commit -am"Added SQL view for possible trades."
[master 94518f2] Added SQL view for possible trades.
Committer: vagrant <vagrant@vagrant.v>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

```

```
git config --global --edit
```

After doing this, you may fix the identity used for this commit with:

```
git commit --amend --reset-author
```

```
1 file changed, 77 insertions(+), 17 deletions(-)
(env) vagrant@vagrant:~/src/whalz$ python -i model.py
Connected to DB.
>>> possible_trades
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'possible_trades' is not defined
>>> possible_trades = possible_trades.query.all()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'possible_trades' is not defined
>>> possible_trades = possible_trades.query.all()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'possible_trades' is not defined
>>> exit()
(env) vagrant@vagrant:~/src/whalz$ python -i model.py
Connected to DB.
>>> possible_trades = possible_trades.query.all()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'possible_trades' is not defined
>>> exit()
(env) vagrant@vagrant:~/src/whalz$ pg_dump whalz > dump.sql
(env) vagrant@vagrant:~/src/whalz$ dropdb whalz
(env) vagrant@vagrant:~/src/whalz$ createdb whalz
(env) vagrant@vagrant:~/src/whalz$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   model.py
        modified:   seed.py
```

Untracked files:
(use "git add <file>..." to include in what will be committed)

```
.DS_Store
api.pyc
charts.py
correlation copy.py
dashboard.py
day2.py
docstheme/
dump.sql
easy.py
env/
fetch.py
load_possible_trades.py
model.pyc
old_server.py
secrets.sh
server.pyc
templates/trade_details.html
theme-dashboard 2/
theme-dashboard.zip
theme-dashboard/
tryrandom.py
```

```
no changes added to commit (use "git add" and/or "git commit -a")
(env) vagrant@vagrant:~/src/whalz$ git commit -am "Added 'Possible Trades' table to model."
[master 8d64b36] Added 'Possible Trades' table to model.
Committer: vagrant <vagrant@vagrant.v>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:
```

```
git config --global --edit
```

After doing this, you may fix the identity used for this commit with:

```
git commit --amend --reset-author
```

```
2 files changed, 27 insertions(+), 98 deletions(-)
(env) vagrant@vagrant:~/src/whalz$ psql whalz
psql (9.5.6)
Type "help" for help.

whalz=# INSERT INTO possible_trades (inventory_id, iso_id)
whalz=# SELECT inventory_id, iso_id
whalz=# FROM inventories
whalz=# JOIN iso
whalz=# USING (beer_code);
INSERT 0 52
whalz=# COMMIT;
WARNING:  there is no transaction in progress
COMMIT
whalz=# SELECT * FROM possible_trades;
whalz=# \q
(env) vagrant@vagrant:~/src/whalz$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   model.py
        modified:   seed.py
```

Untracked files:
(use "git add <file>..." to include in what will be committed)

```
.DS_Store
api.pyc
charts.py
correlation copy.py
dashboard.py
day2.py
docstheme/
dump.sql
easy.py
env/
fetch.py
load_possible_trades.py
model.pyc
old_server.py
secrets.sh
server.pyc
templates/trade_details.html
theme-dashboard 2/
theme-dashboard.zip
theme-dashboard/
tryrandom.py
```

```
no changes added to commit (use "git add" and/or "git commit -a")
(env) vagrant@vagrant:~/src/whalz$ git commit -am "Safe before modifying trade table with composite primary key."
[master bb53292] Safe before modifying trade table with composite primary key.
Committer: vagrant <vagrant@vagrant.vm>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:
```

```
git config --global --edit
```

After doing this, you may fix the identity used for this commit with:

```
git commit --amend --reset-author
```

```
2 files changed, 17 insertions(+)
(env) vagrant@vagrant:~/src/whalz$ dropdb whalz
(env) vagrant@vagrant:~/src/whalz$ createdb whalz
(env) vagrant@vagrant:~/src/whalz$ python -i seed.py
Traceback (most recent call last):
  File "seed.py", line 8, in <module>
    from model import User, Beer, Inventory, ISO, Trade, connect_to_db, db
  File "/home/vagrant/src/whalz/model.py", line 134, in <module>
    class Trade(db.Model):
  File "/home/vagrant/src/whalz/model.py", line 139, in Trade
    __table_args__ = ( db.UniqueConstrain('initial_trade', 'reciprocal_trade'), { } )
AttributeError: 'SQLAlchemy' object has no attribute 'UniqueConstrain'
>>> exit()
(env) vagrant@vagrant:~/src/whalz$ dropdb whalz
(env) vagrant@vagrant:~/src/whalz$ createdb whalz
(env) vagrant@vagrant:~/src/whalz$ python -i seed.py
2017-07-04 01:03:19,193 INFO sqlalchemy.engine.base.Engine select version()
2017-07-04 01:03:19,193 INFO sqlalchemy.engine.base.Engine {}
2017-07-04 01:03:19,195 INFO sqlalchemy.engine.base.Engine select current_schema()
2017-07-04 01:03:19,195 INFO sqlalchemy.engine.base.Engine {}
2017-07-04 01:03:19,196 INFO sqlalchemy.engine.base.Engine SELECT CAST('test plain returns' AS VARCHAR(60)) AS anon_1
2017-07-04 01:03:19,197 INFO sqlalchemy.engine.base.Engine {}
2017-07-04 01:03:19,197 INFO sqlalchemy.engine.base.Engine SELECT CAST('test unicode returns' AS VARCHAR(60)) AS anon_1
2017-07-04 01:03:19,198 INFO sqlalchemy.engine.base.Engine {}
2017-07-04 01:03:19,198 INFO sqlalchemy.engine.base.Engine show standard_conforming_strings
2017-07-04 01:03:19,199 INFO sqlalchemy.engine.base.Engine {}
2017-07-04 01:03:19,200 INFO sqlalchemy.engine.base.Engine select relname from pg_class c join pg_namespace n on n.oid=c.relnamespace where pg_catalog.pg_table_is_visi
ble(c.oid) and relname=(name)s
2017-07-04 01:03:19,201 INFO sqlalchemy.engine.base.Engine {'name': 'u'users'}
2017-07-04 01:03:19,202 INFO sqlalchemy.engine.base.Engine select relname from pg_class c join pg_namespace n on n.oid=c.relnamespace where pg_catalog.pg_table_is_visi
ble(c.oid) and relname=(name)s
2017-07-04 01:03:19,202 INFO sqlalchemy.engine.base.Engine {'name': 'u'trades'}
2017-07-04 01:03:19,203 INFO sqlalchemy.engine.base.Engine select relname from pg_class c join pg_namespace n on n.oid=c.relnamespace where pg_catalog.pg_table_is_visi
ble(c.oid) and relname=(name)s
2017-07-04 01:03:19,203 INFO sqlalchemy.engine.base.Engine {'name': 'u'possible_trades'}
2017-07-04 01:03:19,204 INFO sqlalchemy.engine.base.Engine select relname from pg_class c join pg_namespace n on n.oid=c.relnamespace where pg_catalog.pg_table_is_visi
ble(c.oid) and relname=(name)s
2017-07-04 01:03:19,204 INFO sqlalchemy.engine.base.Engine {'name': 'u'iso'}
2017-07-04 01:03:19,205 INFO sqlalchemy.engine.base.Engine select relname from pg_class c join pg_namespace n on n.oid=c.relnamespace where pg_catalog.pg_table_is_visi
ble(c.oid) and relname=(name)s
2017-07-04 01:03:19,205 INFO sqlalchemy.engine.base.Engine {'name': 'u'beers'}
2017-07-04 01:03:19,206 INFO sqlalchemy.engine.base.Engine select relname from pg_class c join pg_namespace n on n.oid=c.relnamespace where pg_catalog.pg_table_is_visi
ble(c.oid) and relname=(name)s
2017-07-04 01:03:19,206 INFO sqlalchemy.engine.base.Engine {'name': 'u'inventories'}
2017-07-04 01:03:19,207 INFO sqlalchemy.engine.base.Engine
CREATE TABLE users (
    user_id SERIAL NOT NULL,
    name VARCHAR(64) NOT NULL,
    email VARCHAR(64) NOT NULL,
    password VARCHAR(64) NOT NULL,
    PRIMARY KEY (user_id)
)
2017-07-04 01:03:19,209 INFO sqlalchemy.engine.base.Engine {}
2017-07-04 01:03:19,213 INFO sqlalchemy.engine.base.Engine COMMIT
2017-07-04 01:03:19,215 INFO sqlalchemy.engine.base.Engine
CREATE TABLE beers (
    beer_code VARCHAR(10) NOT NULL,
    name VARCHAR(100) NOT NULL,
    style_id INTEGER,
    year INTEGER,
    abv FLOAT,
    beer_variation VARCHAR(100),
    beer_variation_id VARCHAR(100),
    ibu FLOAT,
    PRIMARY KEY (beer_code)
)
2017-07-04 01:03:19,218 INFO sqlalchemy.engine.base.Engine {}
2017-07-04 01:03:19,220 INFO sqlalchemy.engine.base.Engine COMMIT
```

```

2017-07-04 01:03:19,221 INFO sqlalchemy.engine.base.Engine
CREATE TABLE iso (
    iso_id SERIAL NOT NULL,
    beer_code VARCHAR(10),
    user_id INTEGER,
    quantity INTEGER,
    active BOOLEAN NOT NULL,
    PRIMARY KEY (iso_id),
    FOREIGN KEY(beer_code) REFERENCES beers (beer_code),
    FOREIGN KEY(user_id) REFERENCES users (user_id)
)

2017-07-04 01:03:19,222 INFO sqlalchemy.engine.base.Engine {}
2017-07-04 01:03:19,225 INFO sqlalchemy.engine.base.Engine COMMIT
2017-07-04 01:03:19,226 INFO sqlalchemy.engine.base.Engine CREATE INDEX ix_iso_user_id ON iso (user_id)
2017-07-04 01:03:19,226 INFO sqlalchemy.engine.base.Engine {}
2017-07-04 01:03:19,228 INFO sqlalchemy.engine.base.Engine COMMIT
2017-07-04 01:03:19,229 INFO sqlalchemy.engine.base.Engine CREATE INDEX ix_iso_beer_code ON iso (beer_code)
2017-07-04 01:03:19,229 INFO sqlalchemy.engine.base.Engine {}
2017-07-04 01:03:19,231 INFO sqlalchemy.engine.base.Engine COMMIT
2017-07-04 01:03:19,232 INFO sqlalchemy.engine.base.Engine
CREATE TABLE inventories (
    inventory_id SERIAL NOT NULL,
    beer_code VARCHAR(10),
    user_id INTEGER,
    quantity INTEGER,
    PRIMARY KEY (inventory_id),
    FOREIGN KEY(beer_code) REFERENCES beers (beer_code),
    FOREIGN KEY(user_id) REFERENCES users (user_id)
)

2017-07-04 01:03:19,233 INFO sqlalchemy.engine.base.Engine {}
2017-07-04 01:03:19,236 INFO sqlalchemy.engine.base.Engine COMMIT
2017-07-04 01:03:19,237 INFO sqlalchemy.engine.base.Engine CREATE INDEX ix_inventories_beer_code ON inventories (beer_code)
2017-07-04 01:03:19,238 INFO sqlalchemy.engine.base.Engine {}
2017-07-04 01:03:19,239 INFO sqlalchemy.engine.base.Engine COMMIT
2017-07-04 01:03:19,240 INFO sqlalchemy.engine.base.Engine CREATE INDEX ix_inventories_user_id ON inventories (user_id)
2017-07-04 01:03:19,240 INFO sqlalchemy.engine.base.Engine {}
2017-07-04 01:03:19,241 INFO sqlalchemy.engine.base.Engine COMMIT
2017-07-04 01:03:19,242 INFO sqlalchemy.engine.base.Engine
CREATE TABLE possible_trades (
    possible_id SERIAL NOT NULL,
    inventory_id INTEGER,
    iso_id INTEGER,
    PRIMARY KEY (possible_id),
    FOREIGN KEY(inventory_id) REFERENCES inventories (inventory_id),
    FOREIGN KEY(iso_id) REFERENCES iso (iso_id)
)

2017-07-04 01:03:19,242 INFO sqlalchemy.engine.base.Engine {}
2017-07-04 01:03:19,246 INFO sqlalchemy.engine.base.Engine COMMIT
2017-07-04 01:03:19,247 INFO sqlalchemy.engine.base.Engine CREATE INDEX ix_possible_trades_inventory_id ON possible_trades (inventory_id)
2017-07-04 01:03:19,247 INFO sqlalchemy.engine.base.Engine {}
2017-07-04 01:03:19,249 INFO sqlalchemy.engine.base.Engine COMMIT
2017-07-04 01:03:19,250 INFO sqlalchemy.engine.base.Engine CREATE INDEX ix_possible_trades_iso_id ON possible_trades (iso_id)
2017-07-04 01:03:19,251 INFO sqlalchemy.engine.base.Engine {}
2017-07-04 01:03:19,252 INFO sqlalchemy.engine.base.Engine COMMIT
2017-07-04 01:03:19,253 INFO sqlalchemy.engine.base.Engine
CREATE TABLE trades (
    trade_id SERIAL NOT NULL,
    traded_at TIMESTAMP WITHOUT TIME ZONE,
    quantity INTEGER NOT NULL,
    initial_trade INTEGER,
    reciprocal_trade INTEGER,
    PRIMARY KEY (trade_id),
    UNIQUE (initial_trade, reciprocal_trade),
    FOREIGN KEY(initial_trade) REFERENCES possible_trades (possible_id),
    FOREIGN KEY(reciprocal_trade) REFERENCES possible_trades (possible_id)
)

2017-07-04 01:03:19,253 INFO sqlalchemy.engine.base.Engine {}
2017-07-04 01:03:19,257 INFO sqlalchemy.engine.base.Engine COMMIT
2017-07-04 01:03:19,258 INFO sqlalchemy.engine.base.Engine CREATE INDEX ix_trades_initial_trade ON trades (initial_trade)
2017-07-04 01:03:19,258 INFO sqlalchemy.engine.base.Engine {}
2017-07-04 01:03:19,259 INFO sqlalchemy.engine.base.Engine COMMIT
2017-07-04 01:03:19,260 INFO sqlalchemy.engine.base.Engine CREATE INDEX ix_trades_reciprocal_trade ON trades (reciprocal_trade)
2017-07-04 01:03:19,260 INFO sqlalchemy.engine.base.Engine {}
2017-07-04 01:03:19,261 INFO sqlalchemy.engine.base.Engine COMMIT
Fill er up! Beers are being loaded.
Traceback (most recent call last):
  File "seed.py", line 160, in <module>
    load_beer()
  File "seed.py", line 54, in load_beer
    name=name,)
  File "<string>", line 2, in __init__
  File "/home/vagrant/src/whalz/env/local/lib/python2.7/site-packages/sqlalchemy/orm/instrumentation.py", line 347, in _new_state_if_none
    state = self._state_constructor(instance, self)
  File "/home/vagrant/src/whalz/env/local/lib/python2.7/site-packages/sqlalchemy/util/langhelpers.py", line 764, in _get__
    obj.__dict__[self.__name__] = result = self.fget(obj)
  File "/home/vagrant/src/whalz/env/local/lib/python2.7/site-packages/sqlalchemy/orm/instrumentation.py", line 177, in _state_constructor
    self.dispatch.first_init(self, self.class_)
  File "/home/vagrant/src/whalz/env/local/lib/python2.7/site-packages/sqlalchemy/event/attr.py", line 256, in __call__
    fn(*args, **kw)
  File "/home/vagrant/src/whalz/env/local/lib/python2.7/site-packages/sqlalchemy/orm/mapper.py", line 2970, in _event_on_first_init
    configure_mappers()
  File "/home/vagrant/src/whalz/env/local/lib/python2.7/site-packages/sqlalchemy/orm/mapper.py", line 2866, in configure_mappers
    mapper._post_configure_properties()
  File "/home/vagrant/src/whalz/env/local/lib/python2.7/site-packages/sqlalchemy/orm/mapper.py", line 1765, in _post_configure_properties
    prop.init()
  File "/home/vagrant/src/whalz/env/local/lib/python2.7/site-packages/sqlalchemy/orm/interfaces.py", line 184, in init
    self.do_init()
  File "/home/vagrant/src/whalz/env/local/lib/python2.7/site-packages/sqlalchemy/orm/relationships.py", line 1653, in do_init
    self._setup_join_conditions()
  File "/home/vagrant/src/whalz/env/local/lib/python2.7/site-packages/sqlalchemy/orm/relationships.py", line 1728, in _setup_join_conditions
    can_be_synced_fn=self._columns_are_mapped
  File "/home/vagrant/src/whalz/env/local/lib/python2.7/site-packages/sqlalchemy/orm/relationships.py", line 1986, in __init__

```

```

self._determine_joins()
File "/home/vagrant/src/whalzh/env/local/lib/python2.7/site-packages/sqlalchemy/orm/relationships.py", line 2113, in _determine_joins
% self.prop)
sqlalchemy.exc.AmbiguousForeignKeysError: Could not determine join condition between parent/child tables on relationship Trade.possible - there are multiple foreign key
y paths linking the tables. Specify the 'foreign_keys' argument, providing a list of those columns which should be counted as containing a foreign key reference to th
e parent table.
>>> exit()
(env) vagrant@vagrant:~/src/whalzh$ dropdb whalzh
(env) vagrant@vagrant:~/src/whalzh$ createdb whalzh
(env) vagrant@vagrant:~/src/whalzh$ python -i seed.py
2017-07-04 01:05:30,533 INFO sqlalchemy.engine.base.Engine select version()
2017-07-04 01:05:30,534 INFO sqlalchemy.engine.base.Engine {}
2017-07-04 01:05:30,535 INFO sqlalchemy.engine.base.Engine select current_schema()
2017-07-04 01:05:30,535 INFO sqlalchemy.engine.base.Engine {}
2017-07-04 01:05:30,536 INFO sqlalchemy.engine.base.Engine SELECT CAST('test plain returns' AS VARCHAR(60)) AS anon_1
2017-07-04 01:05:30,536 INFO sqlalchemy.engine.base.Engine {}
2017-07-04 01:05:30,537 INFO sqlalchemy.engine.base.Engine SELECT CAST('test unicode returns' AS VARCHAR(60)) AS anon_1
2017-07-04 01:05:30,537 INFO sqlalchemy.engine.base.Engine {}
2017-07-04 01:05:30,538 INFO sqlalchemy.engine.base.Engine show standard_conforming_strings
2017-07-04 01:05:30,538 INFO sqlalchemy.engine.base.Engine {}
2017-07-04 01:05:30,540 INFO sqlalchemy.engine.base.Engine select relname from pg_class c join pg_namespace n on n.oid=c.relnamespace where pg_catalog.pg_table_is_visi
ble(c.oid) and relname=(name)s
2017-07-04 01:05:30,540 INFO sqlalchemy.engine.base.Engine {'name': 'u'users'}
2017-07-04 01:05:30,541 INFO sqlalchemy.engine.base.Engine select relname from pg_class c join pg_namespace n on n.oid=c.relnamespace where pg_catalog.pg_table_is_visi
ble(c.oid) and relname=(name)s
2017-07-04 01:05:30,541 INFO sqlalchemy.engine.base.Engine {'name': 'u'trades'}
2017-07-04 01:05:30,542 INFO sqlalchemy.engine.base.Engine select relname from pg_class c join pg_namespace n on n.oid=c.relnamespace where pg_catalog.pg_table_is_visi
ble(c.oid) and relname=(name)s
2017-07-04 01:05:30,542 INFO sqlalchemy.engine.base.Engine {'name': 'u'possible_trades'}
2017-07-04 01:05:30,543 INFO sqlalchemy.engine.base.Engine select relname from pg_class c join pg_namespace n on n.oid=c.relnamespace where pg_catalog.pg_table_is_visi
ble(c.oid) and relname=(name)s
2017-07-04 01:05:30,543 INFO sqlalchemy.engine.base.Engine {'name': 'u'iso'}
2017-07-04 01:05:30,544 INFO sqlalchemy.engine.base.Engine select relname from pg_class c join pg_namespace n on n.oid=c.relnamespace where pg_catalog.pg_table_is_visi
ble(c.oid) and relname=(name)s
2017-07-04 01:05:30,544 INFO sqlalchemy.engine.base.Engine {'name': 'u'beers'}
2017-07-04 01:05:30,545 INFO sqlalchemy.engine.base.Engine select relname from pg_class c join pg_namespace n on n.oid=c.relnamespace where pg_catalog.pg_table_is_visi
ble(c.oid) and relname=(name)s
2017-07-04 01:05:30,545 INFO sqlalchemy.engine.base.Engine {'name': 'u'inventories'}
2017-07-04 01:05:30,546 INFO sqlalchemy.engine.base.Engine
CREATE TABLE users (
    user_id SERIAL NOT NULL,
    name VARCHAR(64) NOT NULL,
    email VARCHAR(64) NOT NULL,
    password VARCHAR(64) NOT NULL,
    PRIMARY KEY (user_id)
)

2017-07-04 01:05:30,548 INFO sqlalchemy.engine.base.Engine {}
2017-07-04 01:05:30,553 INFO sqlalchemy.engine.base.Engine COMMIT
2017-07-04 01:05:30,554 INFO sqlalchemy.engine.base.Engine
CREATE TABLE beers (
    beer_code VARCHAR(10) NOT NULL,
    name VARCHAR(100) NOT NULL,
    style_id INTEGER,
    year INTEGER,
    abv FLOAT,
    beer_variation VARCHAR(100),
    beer_variation_id VARCHAR(100),
    ibu FLOAT,
    PRIMARY KEY (beer_code)
)

2017-07-04 01:05:30,556 INFO sqlalchemy.engine.base.Engine {}
2017-07-04 01:05:30,558 INFO sqlalchemy.engine.base.Engine COMMIT
2017-07-04 01:05:30,560 INFO sqlalchemy.engine.base.Engine
CREATE TABLE iso (
    iso_id SERIAL NOT NULL,
    beer_code VARCHAR(10),
    user_id INTEGER,
    quantity INTEGER,
    active BOOLEAN NOT NULL,
    PRIMARY KEY (iso_id),
    FOREIGN KEY(beer_code) REFERENCES beers (beer_code),
    FOREIGN KEY(user_id) REFERENCES users (user_id)
)

2017-07-04 01:05:30,561 INFO sqlalchemy.engine.base.Engine {}
2017-07-04 01:05:30,564 INFO sqlalchemy.engine.base.Engine COMMIT
2017-07-04 01:05:30,565 INFO sqlalchemy.engine.base.Engine CREATE INDEX ix_iso_user_id ON iso (user_id)
2017-07-04 01:05:30,565 INFO sqlalchemy.engine.base.Engine {}
2017-07-04 01:05:30,567 INFO sqlalchemy.engine.base.Engine COMMIT
2017-07-04 01:05:30,567 INFO sqlalchemy.engine.base.Engine CREATE INDEX ix_iso_beer_code ON iso (beer_code)
2017-07-04 01:05:30,568 INFO sqlalchemy.engine.base.Engine {}
2017-07-04 01:05:30,570 INFO sqlalchemy.engine.base.Engine COMMIT
2017-07-04 01:05:30,571 INFO sqlalchemy.engine.base.Engine
CREATE TABLE inventories (
    inventory_id SERIAL NOT NULL,
    beer_code VARCHAR(10),
    user_id INTEGER,
    quantity INTEGER,
    PRIMARY KEY (inventory_id),
    FOREIGN KEY(beer_code) REFERENCES beers (beer_code),
    FOREIGN KEY(user_id) REFERENCES users (user_id)
)

2017-07-04 01:05:30,573 INFO sqlalchemy.engine.base.Engine {}
2017-07-04 01:05:30,576 INFO sqlalchemy.engine.base.Engine COMMIT
2017-07-04 01:05:30,577 INFO sqlalchemy.engine.base.Engine CREATE INDEX ix_inventories_beer_code ON inventories (beer_code)
2017-07-04 01:05:30,577 INFO sqlalchemy.engine.base.Engine {}
2017-07-04 01:05:30,579 INFO sqlalchemy.engine.base.Engine COMMIT
2017-07-04 01:05:30,579 INFO sqlalchemy.engine.base.Engine CREATE INDEX ix_inventories_user_id ON inventories (user_id)
2017-07-04 01:05:30,580 INFO sqlalchemy.engine.base.Engine {}
2017-07-04 01:05:30,581 INFO sqlalchemy.engine.base.Engine COMMIT
2017-07-04 01:05:30,582 INFO sqlalchemy.engine.base.Engine
CREATE TABLE possible_trades (
    possible_id SERIAL NOT NULL,

```

```

inventory_id INTEGER,
iso_id INTEGER,
PRIMARY KEY (possible_id),
FOREIGN KEY(inventory_id) REFERENCES inventories (inventory_id),
FOREIGN KEY(iso_id) REFERENCES iso (iso_id)
)

2017-07-04 01:05:30,583 INFO sqlalchemy.engine.base.Engine {}
2017-07-04 01:05:30,585 INFO sqlalchemy.engine.base.Engine COMMIT
2017-07-04 01:05:30,586 INFO sqlalchemy.engine.base.Engine CREATE INDEX ix_possible_trades_inventory_id ON possible_trades (inventory_id)
2017-07-04 01:05:30,586 INFO sqlalchemy.engine.base.Engine {}
2017-07-04 01:05:30,587 INFO sqlalchemy.engine.base.Engine COMMIT
2017-07-04 01:05:30,588 INFO sqlalchemy.engine.base.Engine CREATE INDEX ix_possible_trades_iso_id ON possible_trades (iso_id)
2017-07-04 01:05:30,588 INFO sqlalchemy.engine.base.Engine {}
2017-07-04 01:05:30,589 INFO sqlalchemy.engine.base.Engine COMMIT
2017-07-04 01:05:30,590 INFO sqlalchemy.engine.base.Engine
CREATE TABLE trades (
    trade_id SERIAL NOT NULL,
    traded_at TIMESTAMP WITHOUT TIME ZONE,
    quantity INTEGER NOT NULL,
    initial_trade INTEGER,
    reciprocal_trade INTEGER,
    PRIMARY KEY (trade_id),
    UNIQUE (initial_trade, reciprocal_trade),
    FOREIGN KEY(initial_trade) REFERENCES possible_trades (possible_id),
    FOREIGN KEY(reciprocal_trade) REFERENCES possible_trades (possible_id)
)

2017-07-04 01:05:30,591 INFO sqlalchemy.engine.base.Engine {}
2017-07-04 01:05:30,594 INFO sqlalchemy.engine.base.Engine COMMIT
2017-07-04 01:05:30,595 INFO sqlalchemy.engine.base.Engine CREATE INDEX ix_trades_initial_trade ON trades (initial_trade)
2017-07-04 01:05:30,595 INFO sqlalchemy.engine.base.Engine {}
2017-07-04 01:05:30,597 INFO sqlalchemy.engine.base.Engine COMMIT
2017-07-04 01:05:30,597 INFO sqlalchemy.engine.base.Engine CREATE INDEX ix_trades_reciprocal_trade ON trades (reciprocal_trade)
2017-07-04 01:05:30,598 INFO sqlalchemy.engine.base.Engine {}
2017-07-04 01:05:30,599 INFO sqlalchemy.engine.base.Engine COMMIT
Fill er up! Beers are being loaded.
2017-07-04 01:05:30,686 INFO sqlalchemy.engine.base.Engine BEGIN (implicit)
2017-07-04 01:05:30,690 INFO sqlalchemy.engine.base.Engine INSERT INTO beers (beer_code, name, style_id, year, abv, beer_variation, beer_variation_id, ibu) VALUES (%b
eer_code)s, %(name)s, %(style_id)s, %(year)s, %(abv)s, %(beer_variation)s, %(beer_variation_id)s, %(ibu)s)
2017-07-04 01:05:30,691 INFO sqlalchemy.engine.base.Engine ({'name': u'1912', 'style_id': 132, 'beer_variation_id': None, 'beer_variation': None, 'abv': u'7', 'year': N
one, 'beer_code': u'4vmwih', 'ibu': u'50'}, {'name': u'1916 Irish Stout', 'style_id': 42, 'beer_variation_id': None, 'beer_variation': None, 'abv': None, 'year': None,
'beer_code': u'0QR5YM', 'ibu': None}, {'name': u'1916 Irish Stout', 'style_id': 42, 'beer_variation_id': None, 'beer_variation': None, 'abv': u'3.5', 'year': None, 'b
eer_code': u'Qgdpgp', 'ibu': None}, {'name': u'1916 Shore Shiver', 'style_id': 30, 'beer_variation_id': None, 'beer_variation': None, 'abv': u'6.9', 'year': None, 'bee
r_code': u'9WNkio', 'ibu': None}, {'name': u'1919 Root Beer', 'style_id': 139, 'beer_variation_id': None, 'beer_variation': None, 'abv': None, 'year': None, 'beer_code
': u'0DVz81', 'ibu': None}, {'name': u'1920 Cognac Stout Imperiale', 'style_id': 132, 'beer_variation_id': None, 'beer_variation': None, 'abv': u'11.64', 'year': None,
'beer_code': u'YP4dCI', 'ibu': u'82'}, {'name': u'1926', 'style_id': 25, 'beer_variation_id': None, 'beer_variation': None, 'abv': u'5', 'year': None, 'beer_code': u'
Eyr9Kg', 'ibu': u'45'}, {'name': u'1927', 'style_id': 29, 'beer_variation_id': None, 'beer_variation': None, 'abv': u'7.5', 'year': None, 'beer_code': u'jT5vQ0', 'ibu'
: None} ... displaying 10 of 49 total bound parameter sets ... {'name': u'2 Cilindri', 'style_id': 18, 'beer_variation_id': None, 'beer_variation': None, 'abv': u'5'
, 'year': None, 'beer_code': u'U405po', 'ibu': None}, {'name': u'2 HOP PALE u2013 CENTENNIAL / APOLLO', 'style_id': 25, 'beer_variation_id': None, 'beer_variation': N
one, 'abv': u'5.8', 'year': None, 'beer_code': u'09Yz5Y', 'ibu': u'40'})
2017-07-04 01:05:30,697 INFO sqlalchemy.engine.base.Engine COMMIT
All of your great fake users are being loaded
2017-07-04 01:05:30,713 INFO sqlalchemy.engine.base.Engine BEGIN (implicit)
2017-07-04 01:05:30,714 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,714 INFO sqlalchemy.engine.base.Engine {'password': 'R!4wMY_Vct', 'name': 'Robert Sims', 'email': 'taylorjames@yahoo.com'}
2017-07-04 01:05:30,716 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,716 INFO sqlalchemy.engine.base.Engine {'password': '@Xmo5v414', 'name': 'Alexis Fitzpatrick MD', 'email': 'darrylylons@green.net'}
2017-07-04 01:05:30,717 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,717 INFO sqlalchemy.engine.base.Engine {'password': 'n5P0a9#e#K', 'name': 'Robert Brewer', 'email': 'pbrown@douglas-holmes.com'}
2017-07-04 01:05:30,718 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,718 INFO sqlalchemy.engine.base.Engine {'password': 'R#f1MkpK6b', 'name': 'Isabel Anderson', 'email': 'evansrebecca@young.com'}
2017-07-04 01:05:30,719 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,719 INFO sqlalchemy.engine.base.Engine {'password': 'Cns!71Hxly', 'name': 'Emily Price', 'email': 'alexandria54@stephens-wilson.info'}
2017-07-04 01:05:30,720 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,720 INFO sqlalchemy.engine.base.Engine {'password': '5yN&Z3t6#5', 'name': 'Laura Ford', 'email': 'brian26@yahoo.com'}
2017-07-04 01:05:30,721 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,721 INFO sqlalchemy.engine.base.Engine {'password': 'i2G65_kd'h', 'name': 'Lauren Ramirez', 'email': 'taylorlkimberly@gmail.com'}
2017-07-04 01:05:30,721 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,722 INFO sqlalchemy.engine.base.Engine {'password': 'fclDpu1_6', 'name': 'Adrian Gonzalez', 'email': 'jdickson@lewis.com'}
2017-07-04 01:05:30,722 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,723 INFO sqlalchemy.engine.base.Engine {'password': 'sV6X5Msvs', 'name': 'Dawn Smith', 'email': 'westjohn@rivera.org'}
2017-07-04 01:05:30,723 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,723 INFO sqlalchemy.engine.base.Engine {'password': 'q!XyYHwK9', 'name': 'Robert Banks', 'email': 'qpattton@hotmail.com'}
2017-07-04 01:05:30,724 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,724 INFO sqlalchemy.engine.base.Engine {'password': 'K!k6MV+y**', 'name': 'Clinton Pierce', 'email': 'hillsharon@hotmail.com'}
2017-07-04 01:05:30,725 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,725 INFO sqlalchemy.engine.base.Engine {'password': 'pVq4+Zogw', 'name': 'Valerie Tapia', 'email': 'christinerivera@mcDaniel.org'}
2017-07-04 01:05:30,726 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,726 INFO sqlalchemy.engine.base.Engine {'password': '!j3dMkmRKV', 'name': 'Brent Chavez', 'email': 'dhouston@gibson-allen.com'}
2017-07-04 01:05:30,727 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,727 INFO sqlalchemy.engine.base.Engine {'password': 'Zo!4MorNn6', 'name': 'Steven Valencia', 'email': 'nina62@walsh.com'}
2017-07-04 01:05:30,728 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,728 INFO sqlalchemy.engine.base.Engine {'password': 'VVoX5Yw5', 'name': 'Sarah Carter', 'email': 'wongnicole@cooke-jackson.com'}
2017-07-04 01:05:30,728 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,729 INFO sqlalchemy.engine.base.Engine {'password': 'Ddw9bbMY+3', 'name': 'Kelly Fuller', 'email': 'dperez@porter.org'}
2017-07-04 01:05:30,729 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,729 INFO sqlalchemy.engine.base.Engine {'password': 'xKk0ZRR57B', 'name': 'Jennifer Lewis', 'email': 'mhorton@hendricks.org'}
2017-07-04 01:05:30,730 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,730 INFO sqlalchemy.engine.base.Engine {'password': 'b0s5FYMg#C', 'name': 'John Reilly', 'email': 'sarah10@jenkins.com'}
2017-07-04 01:05:30,731 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,731 INFO sqlalchemy.engine.base.Engine {'password': 'R&2xWalcw', 'name': 'Dr. Jennifer Wilson', 'email': 'craig25@gmail.com'}
2017-07-04 01:05:30,732 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,732 INFO sqlalchemy.engine.base.Engine {'password': '2zSD(s2zi2', 'name': 'Destiny Oconnor', 'email': 'paul64@powers.com'}
2017-07-04 01:05:30,732 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,733 INFO sqlalchemy.engine.base.Engine {'password': 'A20nf0Que', 'name': 'Kathy Cummings', 'email': 'muellerkevin@le.com'}
2017-07-04 01:05:30,733 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,734 INFO sqlalchemy.engine.base.Engine {'password': 'CDAnvjh0!4', 'name': 'Nathan Gray', 'email': 'christopher71@gmail.com'}
2017-07-04 01:05:30,734 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,735 INFO sqlalchemy.engine.base.Engine {'password': '8DwWAcQ!_P', 'name': 'Elizabeth Thomas', 'email': 'owright@hotmail.com'}
2017-07-04 01:05:30,735 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,736 INFO sqlalchemy.engine.base.Engine {'password': 'a!6XAso+*', 'name': 'Samantha White', 'email': 'brand10@yahoo.com'}
2017-07-04 01:05:30,736 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,737 INFO sqlalchemy.engine.base.Engine {'password': '#h00m0bU%3', 'name': 'Robert Dean', 'email': 'clements@vickie@gmail.com'}
2017-07-04 01:05:30,737 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,738 INFO sqlalchemy.engine.base.Engine {'password': 'y+dw1PxNm', 'name': 'Ryan Dominguez', 'email': 'huangrandy@martinez-smith.com'}
2017-07-04 01:05:30,738 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,738 INFO sqlalchemy.engine.base.Engine {'password': 'LPNsPzr7!0', 'name': 'Elizabeth Hudson', 'email': 'davidibarra@buckley.org'}

```



```
2017-07-04 01:05:30,739 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,739 INFO sqlalchemy.engine.base.Engine {'password': 'fM!@6SFJjW', 'name': 'Erin Johnson', 'email': 'claudia99@hotmail.com'}
2017-07-04 01:05:30,740 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,740 INFO sqlalchemy.engine.base.Engine {'password': 'D560Hln1C!', 'name': 'Richard Ali', 'email': 'frange@whitaker.org'}
2017-07-04 01:05:30,741 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,741 INFO sqlalchemy.engine.base.Engine {'password': '805mNgga)a', 'name': 'Philip Peterson', 'email': 'jennifer11@gmail.com'}
2017-07-04 01:05:30,741 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,742 INFO sqlalchemy.engine.base.Engine {'password': '0y856Mq2C%', 'name': 'Kevin Morales', 'email': 'ubradygross.com'}
2017-07-04 01:05:30,742 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,742 INFO sqlalchemy.engine.base.Engine {'password': 'ae00tm$u**', 'name': 'Jeremy Solomon', 'email': 'hatfieldn1cole@gmail.com'}
2017-07-04 01:05:30,743 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,743 INFO sqlalchemy.engine.base.Engine {'password': '80qPGrunqu', 'name': 'Joseph Mcgee', 'email': 'megan46@gmail.com'}
2017-07-04 01:05:30,744 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,744 INFO sqlalchemy.engine.base.Engine {'password': '6R8B&wBes!', 'name': 'Amber Diaz', 'email': 'robert96@hotmail.com'}
2017-07-04 01:05:30,744 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,745 INFO sqlalchemy.engine.base.Engine {'password': '%$#Zg5m@35', 'name': 'Miguel Long', 'email': 'iconner@yahoo.com'}
2017-07-04 01:05:30,745 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,746 INFO sqlalchemy.engine.base.Engine {'password': '#.41l0hsh', 'name': 'Michael Collins', 'email': 'ronald28@graham-mcdonald.net'}
2017-07-04 01:05:30,746 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,746 INFO sqlalchemy.engine.base.Engine {'password': '%hU2J2L%&', 'name': 'Tammy Cuevas', 'email': 'gina87@anderson-carter.com'}
2017-07-04 01:05:30,747 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,747 INFO sqlalchemy.engine.base.Engine {'password': 'Qq75Qdcy@b', 'name': 'Michael Lawson', 'email': 'rbrooks@fisher-bolton.net'}
2017-07-04 01:05:30,748 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,748 INFO sqlalchemy.engine.base.Engine {'password': 're$KEL1r&d', 'name': 'Jessica Scott', 'email': 'michellemiller@francis-lee.info'}
2017-07-04 01:05:30,748 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,749 INFO sqlalchemy.engine.base.Engine {'password': 'XG3ZAFp$T%', 'name': 'Heather Burton', 'email': 'rileymatthew@gmail.com'}
2017-07-04 01:05:30,749 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,749 INFO sqlalchemy.engine.base.Engine {'password': '%6WzZTF8w6', 'name': 'Elizabeth Holmes', 'email': 'gkrause@yahoo.com'}
2017-07-04 01:05:30,750 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,750 INFO sqlalchemy.engine.base.Engine {'password': '%omEGGVe?', 'name': 'Jacob Baker', 'email': 'fergusondebora@gmail.com'}
2017-07-04 01:05:30,751 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,752 INFO sqlalchemy.engine.base.Engine {'password': '%d7Qptqsq3', 'name': 'Garrett Moreno', 'email': 'kiddmariah@gmail.com'}
2017-07-04 01:05:30,752 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,753 INFO sqlalchemy.engine.base.Engine {'password': '%!Q&kQXp8', 'name': 'Joshua Lynch', 'email': 'fisherbrittany@nelson.com'}
2017-07-04 01:05:30,753 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,754 INFO sqlalchemy.engine.base.Engine {'password': '%1x9XF33br', 'name': 'Andrew Ford', 'email': 'kaufmannicholas@gmail.com'}
2017-07-04 01:05:30,754 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,755 INFO sqlalchemy.engine.base.Engine {'password': '%G07+Jwen$', 'name': 'Francisco Garcia', 'email': 'kellyjustin@marsh.net'}
2017-07-04 01:05:30,755 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,756 INFO sqlalchemy.engine.base.Engine {'password': '%TQm0Msb+?', 'name': 'Elizabeth Brown', 'email': 'parkerlisa@collins-castro.com'}
2017-07-04 01:05:30,757 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,757 INFO sqlalchemy.engine.base.Engine {'password': '%r10_8lgaJH', 'name': 'Kimberly Gordon', 'email': 'morgangomez@wheeler.info'}
2017-07-04 01:05:30,758 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,758 INFO sqlalchemy.engine.base.Engine {'password': '%3YD6c&#!b', 'name': 'Erin Velasquez', 'email': 'gonzaleseric@watkins.org'}
2017-07-04 01:05:30,758 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,759 INFO sqlalchemy.engine.base.Engine {'password': '%n@85GjAKc', 'name': 'Claire Cruz', 'email': 'kristakelly@phillips-valencia.org'}
2017-07-04 01:05:30,759 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,760 INFO sqlalchemy.engine.base.Engine {'password': '%8l1X6U5%$', 'name': 'Michael Brown', 'email': 'gmahoney@perez.com'}
2017-07-04 01:05:30,760 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,760 INFO sqlalchemy.engine.base.Engine {'password': '%+50fyK38W', 'name': 'Jacob Le', 'email': 'robertwright@yahoo.com'}
2017-07-04 01:05:30,761 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,761 INFO sqlalchemy.engine.base.Engine {'password': '%TPQfKA$P_#?', 'name': 'Benjamin Klein', 'email': 'welliott@williams-cline.biz'}
2017-07-04 01:05:30,762 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,762 INFO sqlalchemy.engine.base.Engine {'password': '%(J14Es&0&?', 'name': 'Susan Perez', 'email': 'baldwinchristopher@smith.com'}
2017-07-04 01:05:30,763 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,763 INFO sqlalchemy.engine.base.Engine {'password': '%Jb*7T8gV2B', 'name': 'Casey Farrell DVM', 'email': 'ygraham@davis-bennett.com'}
2017-07-04 01:05:30,764 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,764 INFO sqlalchemy.engine.base.Engine {'password': '%(3%VattC!}', 'name': 'Ricardo Carter', 'email': 'edwardssherry@pearson.com'}
2017-07-04 01:05:30,765 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,765 INFO sqlalchemy.engine.base.Engine {'password': '%6suuQCLK13', 'name': 'Thomas Johnson', 'email': 'caroline2@gmail.com'}
2017-07-04 01:05:30,765 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,766 INFO sqlalchemy.engine.base.Engine {'password': '%w*8B)6Plac', 'name': 'Scott Norman', 'email': 'mary16@yahoo.com'}
2017-07-04 01:05:30,766 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,767 INFO sqlalchemy.engine.base.Engine {'password': '%2!b64dZo&%', 'name': 'Erik Morales', 'email': 'rjoseph@calhoun.info'}
2017-07-04 01:05:30,768 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,769 INFO sqlalchemy.engine.base.Engine {'password': '%9lLfKfY0%', 'name': 'Charles Jordan', 'email': 'andrew99@miller-ibarra.com'}
2017-07-04 01:05:30,770 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,770 INFO sqlalchemy.engine.base.Engine {'password': '%II05h3Aa0%', 'name': 'Ronald Wheeler', 'email': 'cgrant@boone.biz'}
2017-07-04 01:05:30,771 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,771 INFO sqlalchemy.engine.base.Engine {'password': '%T!eV2Yk4R', 'name': 'Aaron Smith', 'email': 'mbriggs@bates.org'}
2017-07-04 01:05:30,771 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,772 INFO sqlalchemy.engine.base.Engine {'password': '%R5B6fkvr$%', 'name': 'Christopher Ochoa', 'email': 'richard47@hotmail.com'}
2017-07-04 01:05:30,772 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,772 INFO sqlalchemy.engine.base.Engine {'password': '%mv1F0D1VP$', 'name': 'Nicholas Miranda', 'email': 'dixontimothy@immons.net'}
2017-07-04 01:05:30,773 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,773 INFO sqlalchemy.engine.base.Engine {'password': '%t)4TBMExKb8', 'name': 'Gary Peterson', 'email': 'derrick85@hotmail.com'}
2017-07-04 01:05:30,774 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,774 INFO sqlalchemy.engine.base.Engine {'password': '%H)F8DEd4XD', 'name': 'Stacy Colon', 'email': 'kristina08@loyd.com'}
2017-07-04 01:05:30,774 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,775 INFO sqlalchemy.engine.base.Engine {'password': '%5vBpN)2g+4', 'name': 'Anna Walker', 'email': 'zpatton@yahoo.com'}
2017-07-04 01:05:30,775 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,775 INFO sqlalchemy.engine.base.Engine {'password': '%#7VItGmx0', 'name': 'Lori Sanchez', 'email': 'dpope@hotmail.com'}
2017-07-04 01:05:30,776 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,776 INFO sqlalchemy.engine.base.Engine {'password': '%#8r+RrQD0', 'name': 'Brooke Johnson', 'email': 'richardmendoza@miller.com'}
2017-07-04 01:05:30,776 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,777 INFO sqlalchemy.engine.base.Engine {'password': '%N+y8Cv+pvZ!', 'name': 'Adam Green', 'email': 'christinamartinez@leach.com'}
2017-07-04 01:05:30,777 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,777 INFO sqlalchemy.engine.base.Engine {'password': '%qK&k3YhkK4', 'name': 'Mallory Bernard MD', 'email': 'efloyd@crawford.com'}
2017-07-04 01:05:30,778 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,778 INFO sqlalchemy.engine.base.Engine {'password': '%N)5BLuFRLk', 'name': 'Brandon Gonzales', 'email': 'cbowers@lee.com'}
2017-07-04 01:05:30,778 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,779 INFO sqlalchemy.engine.base.Engine {'password': '%V6^~Lg@6', 'name': 'Joseph Santos', 'email': 'kingmatthew@acosta.com'}
2017-07-04 01:05:30,779 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,780 INFO sqlalchemy.engine.base.Engine {'password': '%(QAq40pc2}', 'name': 'Robert Humphrey', 'email': 'tonya76@yahoo.com'}
2017-07-04 01:05:30,780 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,780 INFO sqlalchemy.engine.base.Engine {'password': '%DG22UbuEw*', 'name': 'Kerry Wood', 'email': 'manuel11@miller.org'}
2017-07-04 01:05:30,781 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,781 INFO sqlalchemy.engine.base.Engine {'password': '%Qq4NCCY$%', 'name': 'Samuel Barber', 'email': 'fhenry@gmail.com'}
2017-07-04 01:05:30,781 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,782 INFO sqlalchemy.engine.base.Engine {'password': '%5L191or00', 'name': 'Jennifer Goodman', 'email': 'kristin42@fields.biz'}
2017-07-04 01:05:30,782 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,782 INFO sqlalchemy.engine.base.Engine {'password': '%0z20QVmp6M', 'name': 'Anthony James', 'email': 'eduardomyers@gmail.com'}
2017-07-04 01:05:30,783 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,783 INFO sqlalchemy.engine.base.Engine {'password': '%1y7qAUl0c', 'name': 'Kenneth Lynch', 'email': 'tammy51@roth.com'}
2017-07-04 01:05:30,784 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,785 INFO sqlalchemy.engine.base.Engine {'password': '%6%N1w%)M&R', 'name': 'Brent Green', 'email': 'claytonadkins@evans.com'}
2017-07-04 01:05:30,785 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,785 INFO sqlalchemy.engine.base.Engine {'password': '%_0LTxbDg1', 'name': 'Thomas Green', 'email': 'reynoldsveronica@gmail.com'}
2017-07-04 01:05:30,786 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,786 INFO sqlalchemy.engine.base.Engine {'password': '%*t7ELAm5_j', 'name': 'Jennifer Hawkins', 'email': 'cfrye@carrillo.com'}
```

```
2017-07-04 01:05:30,787 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,787 INFO sqlalchemy.engine.base.Engine {'password': '915LxjBj%', 'name': 'Keith Ford', 'email': 'dannydelgado@gmail.com'}
2017-07-04 01:05:30,788 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,788 INFO sqlalchemy.engine.base.Engine {'password': 'wPPEfeqes3', 'name': 'Tiffany Flores', 'email': 'pcobb@higgins-wilson.org'}
2017-07-04 01:05:30,788 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,789 INFO sqlalchemy.engine.base.Engine {'password': 'r4%7rNqE9K', 'name': 'Jose Becker', 'email': 'darryl31@danields.com'}
2017-07-04 01:05:30,789 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,790 INFO sqlalchemy.engine.base.Engine {'password': '6_0710AaX', 'name': 'Cathy Thomas', 'email': 'henry88@miller.biz'}
2017-07-04 01:05:30,790 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,790 INFO sqlalchemy.engine.base.Engine {'password': '61#1QdZnw', 'name': 'Michael Hudson', 'email': 'lisa@melton@gmail.com'}
2017-07-04 01:05:30,791 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,791 INFO sqlalchemy.engine.base.Engine {'password': 'V12g0ALVb', 'name': 'Melissa Wu', 'email': 'phillips@nicole@rubio.org'}
2017-07-04 01:05:30,791 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,792 INFO sqlalchemy.engine.base.Engine {'password': '1z*76ZiAc', 'name': 'Julie Simpson', 'email': 'wyatt@yesenia@yahoo.com'}
2017-07-04 01:05:30,792 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,792 INFO sqlalchemy.engine.base.Engine {'password': 'qo2y3pCtF', 'name': 'Sarah Branch', 'email': 'pamelawelch@andrews.biz'}
2017-07-04 01:05:30,793 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,793 INFO sqlalchemy.engine.base.Engine {'password': '6o6QCn8yU', 'name': 'Ricky Rose', 'email': 'cunninghamjessica@hotmail.com'}
2017-07-04 01:05:30,794 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,794 INFO sqlalchemy.engine.base.Engine {'password': '*_b0woHy_9', 'name': 'Billy Floyd', 'email': 'donnaortiz@harper-figueroa.com'}
2017-07-04 01:05:30,794 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,795 INFO sqlalchemy.engine.base.Engine {'password': 'A2%kL2o6%', 'name': 'Lori Ford', 'email': 'carterleah@gmail.com'}
2017-07-04 01:05:30,795 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,796 INFO sqlalchemy.engine.base.Engine {'password': 'ws24XxLKl', 'name': 'Patricia Park', 'email': 'joshuaruiz@hotmail.com'}
2017-07-04 01:05:30,796 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,796 INFO sqlalchemy.engine.base.Engine {'password': '$2GWR_Fp^m', 'name': 'Misty Ray', 'email': 'hjones@carrillo.com'}
2017-07-04 01:05:30,797 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,797 INFO sqlalchemy.engine.base.Engine {'password': 'Q*7_7NxxB', 'name': 'Angel Stevens', 'email': 'buckrandy@gmail.com'}
2017-07-04 01:05:30,798 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,798 INFO sqlalchemy.engine.base.Engine {'password': 'Te0K@aElZ', 'name': 'Brian Johnson', 'email': 'moniquelittle@palmer.com'}
2017-07-04 01:05:30,799 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,799 INFO sqlalchemy.engine.base.Engine {'password': 'sJ3B!54nG', 'name': 'Michael Hanson', 'email': 'abarker@hotmail.com'}
2017-07-04 01:05:30,800 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,800 INFO sqlalchemy.engine.base.Engine {'password': 'B+5P5H3qEc', 'name': 'Bethany Malone', 'email': 'ryanwarren@turner.com'}
2017-07-04 01:05:30,802 INFO sqlalchemy.engine.base.Engine INSERT INTO users (name, email, password) VALUES (%(name)s, %(email)s, %(password)s) RETURNING users.user_id
2017-07-04 01:05:30,802 INFO sqlalchemy.engine.base.Engine {'password': 'H_8s)Xym1', 'name': 'James Bishop', 'email': 'walterbarnett@yahoo.com'}
2017-07-04 01:05:30,805 INFO sqlalchemy.engine.base.Engine COMMIT
Inventories are filling up. Waiting in IPP lines really pays off.
2017-07-04 01:05:32,247 INFO sqlalchemy.engine.base.Engine BEGIN (implicit)
2017-07-04 01:05:32,248 INFO sqlalchemy.engine.base.Engine INSERT INTO inventories (beer_code, user_id, quantity) VALUES (%(beer_code)s, %(user_id)s, %(quantity)s) RET
URNING inventories.inventory_id
2017-07-04 01:05:32,249 INFO sqlalchemy.engine.base.Engine {'beer_code': '9wNKio', 'user_id': 13, 'quantity': 7}
2017-07-04 01:05:32,250 INFO sqlalchemy.engine.base.Engine COMMIT
2017-07-04 01:05:32,252 INFO sqlalchemy.engine.base.Engine BEGIN (implicit)
2017-07-04 01:05:32,252 INFO sqlalchemy.engine.base.Engine INSERT INTO inventories (beer_code, user_id, quantity) VALUES (%(beer_code)s, %(user_id)s, %(quantity)s) RET
URNING inventories.inventory_id
2017-07-04 01:05:32,253 INFO sqlalchemy.engine.base.Engine {'beer_code': 'jT5v00', 'user_id': 100, 'quantity': 7}
2017-07-04 01:05:32,253 INFO sqlalchemy.engine.base.Engine COMMIT
2017-07-04 01:05:32,255 INFO sqlalchemy.engine.base.Engine BEGIN (implicit)
2017-07-04 01:05:32,255 INFO sqlalchemy.engine.base.Engine INSERT INTO inventories (beer_code, user_id, quantity) VALUES (%(beer_code)s, %(user_id)s, %(quantity)s) RET
URNING inventories.inventory_id
2017-07-04 01:05:32,255 INFO sqlalchemy.engine.base.Engine {'beer_code': 'oQR5YM', 'user_id': 2, 'quantity': 7}
2017-07-04 01:05:32,256 INFO sqlalchemy.engine.base.Engine COMMIT
2017-07-04 01:05:32,257 INFO sqlalchemy.engine.base.Engine BEGIN (implicit)
2017-07-04 01:05:32,258 INFO sqlalchemy.engine.base.Engine INSERT INTO inventories (beer_code, user_id, quantity) VALUES (%(beer_code)s, %(user_id)s, %(quantity)s) RET
URNING inventories.inventory_id
2017-07-04 01:05:32,258 INFO sqlalchemy.engine.base.Engine {'beer_code': '0DVz81', 'user_id': 78, 'quantity': 7}
2017-07-04 01:05:32,259 INFO sqlalchemy.engine.base.Engine COMMIT
2017-07-04 01:05:32,260 INFO sqlalchemy.engine.base.Engine BEGIN (implicit)
2017-07-04 01:05:32,260 INFO sqlalchemy.engine.base.Engine INSERT INTO inventories (beer_code, user_id, quantity) VALUES (%(beer_code)s, %(user_id)s, %(quantity)s) RET
URNING inventories.inventory_id
2017-07-04 01:05:32,261 INFO sqlalchemy.engine.base.Engine {'beer_code': 'kQ42vv', 'user_id': 57, 'quantity': 7}
2017-07-04 01:05:32,261 INFO sqlalchemy.engine.base.Engine COMMIT
2017-07-04 01:05:32,263 INFO sqlalchemy.engine.base.Engine BEGIN (implicit)
2017-07-04 01:05:32,263 INFO sqlalchemy.engine.base.Engine INSERT INTO inventories (beer_code, user_id, quantity) VALUES (%(beer_code)s, %(user_id)s, %(quantity)s) RET
URNING inventories.inventory_id
2017-07-04 01:05:32,264 INFO sqlalchemy.engine.base.Engine {'beer_code': '0DVz81', 'user_id': 2, 'quantity': 7}
2017-07-04 01:05:32,264 INFO sqlalchemy.engine.base.Engine COMMIT
2017-07-04 01:05:32,266 INFO sqlalchemy.engine.base.Engine BEGIN (implicit)
2017-07-04 01:05:32,266 INFO sqlalchemy.engine.base.Engine INSERT INTO inventories (beer_code, user_id, quantity) VALUES (%(beer_code)s, %(user_id)s, %(quantity)s) RET
URNING inventories.inventory_id
2017-07-04 01:05:32,266 INFO sqlalchemy.engine.base.Engine {'beer_code': '0DVz81', 'user_id': 44, 'quantity': 7}
2017-07-04 01:05:32,267 INFO sqlalchemy.engine.base.Engine COMMIT
2017-07-04 01:05:32,268 INFO sqlalchemy.engine.base.Engine BEGIN (implicit)
2017-07-04 01:05:32,269 INFO sqlalchemy.engine.base.Engine INSERT INTO inventories (beer_code, user_id, quantity) VALUES (%(beer_code)s, %(user_id)s, %(quantity)s) RET
URNING inventories.inventory_id
2017-07-04 01:05:32,269 INFO sqlalchemy.engine.base.Engine {'beer_code': '0DVz81', 'user_id': 74, 'quantity': 7}
2017-07-04 01:05:32,270 INFO sqlalchemy.engine.base.Engine COMMIT
2017-07-04 01:05:32,271 INFO sqlalchemy.engine.base.Engine BEGIN (implicit)
2017-07-04 01:05:32,271 INFO sqlalchemy.engine.base.Engine INSERT INTO inventories (beer_code, user_id, quantity) VALUES (%(beer_code)s, %(user_id)s, %(quantity)s) RET
URNING inventories.inventory_id
2017-07-04 01:05:32,271 INFO sqlalchemy.engine.base.Engine {'beer_code': 'Qg6dpj', 'user_id': 40, 'quantity': 7}
2017-07-04 01:05:32,272 INFO sqlalchemy.engine.base.Engine COMMIT
2017-07-04 01:05:32,273 INFO sqlalchemy.engine.base.Engine BEGIN (implicit)
2017-07-04 01:05:32,274 INFO sqlalchemy.engine.base.Engine INSERT INTO inventories (beer_code, user_id, quantity) VALUES (%(beer_code)s, %(user_id)s, %(quantity)s) RET
URNING inventories.inventory_id
2017-07-04 01:05:32,274 INFO sqlalchemy.engine.base.Engine {'beer_code': 'Qg6dpj', 'user_id': 58, 'quantity': 7}
2017-07-04 01:05:32,275 INFO sqlalchemy.engine.base.Engine COMMIT
2017-07-04 01:05:32,276 INFO sqlalchemy.engine.base.Engine BEGIN (implicit)
2017-07-04 01:05:32,276 INFO sqlalchemy.engine.base.Engine INSERT INTO inventories (beer_code, user_id, quantity) VALUES (%(beer_code)s, %(user_id)s, %(quantity)s) RET
URNING inventories.inventory_id
2017-07-04 01:05:32,277 INFO sqlalchemy.engine.base.Engine {'beer_code': 'jT5v00', 'user_id': 39, 'quantity': 7}
2017-07-04 01:05:32,277 INFO sqlalchemy.engine.base.Engine COMMIT
2017-07-04 01:05:32,278 INFO sqlalchemy.engine.base.Engine BEGIN (implicit)
2017-07-04 01:05:32,279 INFO sqlalchemy.engine.base.Engine INSERT INTO inventories (beer_code, user_id, quantity) VALUES (%(beer_code)s, %(user_id)s, %(quantity)s) RET
URNING inventories.inventory_id
2017-07-04 01:05:32,279 INFO sqlalchemy.engine.base.Engine {'beer_code': 'kQ42vv', 'user_id': 72, 'quantity': 7}
2017-07-04 01:05:32,280 INFO sqlalchemy.engine.base.Engine COMMIT
2017-07-04 01:05:32,281 INFO sqlalchemy.engine.base.Engine BEGIN (implicit)
2017-07-04 01:05:32,282 INFO sqlalchemy.engine.base.Engine INSERT INTO inventories (beer_code, user_id, quantity) VALUES (%(beer_code)s, %(user_id)s, %(quantity)s) RET
URNING inventories.inventory_id
2017-07-04 01:05:32,282 INFO sqlalchemy.engine.base.Engine {'beer_code': '0DVz81', 'user_id': 6, 'quantity': 7}
2017-07-04 01:05:32,283 INFO sqlalchemy.engine.base.Engine COMMIT
2017-07-04 01:05:32,284 INFO sqlalchemy.engine.base.Engine BEGIN (implicit)
2017-07-04 01:05:32,285 INFO sqlalchemy.engine.base.Engine INSERT INTO inventories (beer_code, user_id, quantity) VALUES (%(beer_code)s, %(user_id)s, %(quantity)s) RET
URNING inventories.inventory_id
2017-07-04 01:05:32,285 INFO sqlalchemy.engine.base.Engine {'beer_code': 'oQR5YM', 'user_id': 55, 'quantity': 7}
2017-07-04 01:05:32,286 INFO sqlalchemy.engine.base.Engine COMMIT
2017-07-04 01:05:32,287 INFO sqlalchemy.engine.base.Engine BEGIN (implicit)
2017-07-04 01:05:32,288 INFO sqlalchemy.engine.base.Engine INSERT INTO inventories (beer_code, user_id, quantity) VALUES (%(beer_code)s, %(user_id)s, %(quantity)s) RET
```

[illegible]

```

2017-07-04 01:05:32,359 INFO sqlalchemy.engine.base.Engine INSERT INTO iso (beer_code, user_id, quantity, active) VALUES (%(beer_code)s, %(user_id)s, %(quantity)s, %(active)s) RETURNING iso.iso_id
2017-07-04 01:05:32,360 INFO sqlalchemy.engine.base.Engine {'active': True, 'beer_code': 'jT5vQ0', 'user_id': 3, 'quantity': 1}
2017-07-04 01:05:32,360 INFO sqlalchemy.engine.base.Engine COMMIT
2017-07-04 01:05:32,361 INFO sqlalchemy.engine.base.Engine BEGIN (implicit)
2017-07-04 01:05:32,362 INFO sqlalchemy.engine.base.Engine INSERT INTO iso (beer_code, user_id, quantity, active) VALUES (%(beer_code)s, %(user_id)s, %(quantity)s, %(active)s) RETURNING iso.iso_id
2017-07-04 01:05:32,363 INFO sqlalchemy.engine.base.Engine {'active': True, 'beer_code': '9wNKio', 'user_id': 6, 'quantity': 1}
2017-07-04 01:05:32,363 INFO sqlalchemy.engine.base.Engine COMMIT
2017-07-04 01:05:32,364 INFO sqlalchemy.engine.base.Engine BEGIN (implicit)
2017-07-04 01:05:32,365 INFO sqlalchemy.engine.base.Engine INSERT INTO iso (beer_code, user_id, quantity, active) VALUES (%(beer_code)s, %(user_id)s, %(quantity)s, %(active)s) RETURNING iso.iso_id
2017-07-04 01:05:32,365 INFO sqlalchemy.engine.base.Engine {'active': True, 'beer_code': 'Qg6dpg', 'user_id': 27, 'quantity': 1}
2017-07-04 01:05:32,366 INFO sqlalchemy.engine.base.Engine COMMIT
2017-07-04 01:05:32,368 INFO sqlalchemy.engine.base.Engine BEGIN (implicit)
2017-07-04 01:05:32,368 INFO sqlalchemy.engine.base.Engine INSERT INTO iso (beer_code, user_id, quantity, active) VALUES (%(beer_code)s, %(user_id)s, %(quantity)s, %(active)s) RETURNING iso.iso_id
2017-07-04 01:05:32,368 INFO sqlalchemy.engine.base.Engine {'active': True, 'beer_code': '9wNKio', 'user_id': 41, 'quantity': 1}
2017-07-04 01:05:32,369 INFO sqlalchemy.engine.base.Engine COMMIT
>>> all_possible = db.session.query(ViewPossibleTrade.inventory_id, ViewPossibleTrade.iso_id).all()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'ViewPossibleTrade' is not defined
>>> exit()
(env) vagrant@vagrant:~/src/whalz$ python -i model.py
Connected to DB.
>>> all_possible = db.session.query(ViewPossibleTrade.inventory_id, ViewPossibleTrade.iso_id).all()
2017-07-04 01:14:56,696 INFO sqlalchemy.engine.base.Engine select version()
2017-07-04 01:14:56,697 INFO sqlalchemy.engine.base.Engine {}
2017-07-04 01:14:56,698 INFO sqlalchemy.engine.base.Engine select current_schema()
2017-07-04 01:14:56,699 INFO sqlalchemy.engine.base.Engine {}
2017-07-04 01:14:56,700 INFO sqlalchemy.engine.base.Engine SELECT CAST('test plain returns' AS VARCHAR(60)) AS anon_1
2017-07-04 01:14:56,700 INFO sqlalchemy.engine.base.Engine {}
2017-07-04 01:14:56,701 INFO sqlalchemy.engine.base.Engine SELECT CAST('test unicode returns' AS VARCHAR(60)) AS anon_1
2017-07-04 01:14:56,701 INFO sqlalchemy.engine.base.Engine {}
2017-07-04 01:14:56,702 INFO sqlalchemy.engine.base.Engine show standard_conforming_strings
2017-07-04 01:14:56,702 INFO sqlalchemy.engine.base.Engine {}
2017-07-04 01:14:56,703 INFO sqlalchemy.engine.base.Engine BEGIN (implicit)
2017-07-04 01:14:56,703 INFO sqlalchemy.engine.base.Engine SELECT possible_trades.inventory_id AS possible_trades_inventory_id, possible_trades.iso_id AS possible_trades_iso_id
FROM possible_trades
2017-07-04 01:14:56,704 INFO sqlalchemy.engine.base.Engine {}
>>> possible_list = []
>>> for each in all_possible:
...     possible_list.append(each)
...
>>> check_the_other_way = [(t[1], t[0]) for t in possible_list]
>>>
>>> its_a_match = set(all_possible) & set(check_the_other_way)
>>>
>>> its_a_match
set([(16, 6), (19, 9), (11, 11), (7, 3), (10, 2), (18, 18), (2, 10), (1, 18), (2, 17), (17, 2), (3, 7), (9, 19), (1, 1), (18, 1), (6, 16)])
>>> hi = db.session.query(ISO).get(16)
2017-07-04 01:23:35,795 INFO sqlalchemy.engine.base.Engine SELECT iso.iso_id AS iso_iso_id, iso.beer_code AS iso_beer_code, iso.user_id AS iso_user_id, iso.quantity AS iso_quantity, iso.active AS iso_active
FROM iso
WHERE iso.iso_id = %(param_1)s
2017-07-04 01:23:35,796 INFO sqlalchemy.engine.base.Engine {'param_1': 16}
>>> dir(hi)
['ViewPossibleTrade', '__class__', '__delattr__', '__dict__', '__doc__', '__format__', '__getattribute__', '__hash__', '__init__', '__mapper__', '__module__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__setattr__', '__sizeof__', '__str__', '__subclasshook__', '__table__', '__tablename__', '__weakref__', '__cached_tablename', '__decl_class_registry', '__sa_class_manager', '__sa_instance_state', 'active', 'beer', 'beer_code', 'iso_id', 'metadata', 'quantity', 'query', 'query_class', 'user', 'user_id']
>>> hi.active
True
>>> hi.active = "False"
>>> hi.active
'False'
>>> hi.user.name
2017-07-04 01:28:55,190 INFO sqlalchemy.engine.base.Engine UPDATE iso SET active=%(active)s WHERE iso.iso_id = %(iso_iso_id)s
2017-07-04 01:28:55,190 INFO sqlalchemy.engine.base.Engine {'active': 'False', 'iso_iso_id': 16}
2017-07-04 01:28:55,192 INFO sqlalchemy.engine.base.Engine SELECT users.user_id AS users_user_id, users.name AS users_name, users.email AS users_email, users.password AS users_password
FROM users
WHERE users.user_id = %(param_1)s
2017-07-04 01:28:55,192 INFO sqlalchemy.engine.base.Engine {'param_1': 13}
>>> u=Brent Chavez'
>>> dir(hi.user)
['__class__', '__delattr__', '__dict__', '__doc__', '__format__', '__getattribute__', '__hash__', '__init__', '__mapper__', '__module__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__setattr__', '__sizeof__', '__str__', '__subclasshook__', '__table__', '__tablename__', '__weakref__', '__cached_tablename', '__decl_class_registry', '__sa_class_manager', '__sa_instance_state', 'email', 'inventories', 'iso', 'metadata', 'name', 'password', 'query', 'query_class', 'user_id']
>>> hi.user.inventories
2017-07-04 01:30:44,045 INFO sqlalchemy.engine.base.Engine SELECT inventories.inventory_id AS inventories_inventory_id, inventories.beer_code AS inventories_beer_code, inventories.user_id AS inventories_user_id, inventories.quantity AS inventories_quantity
FROM inventories
WHERE %(param_1)s = inventories.user_id ORDER BY inventories.inventory_id
2017-07-04 01:30:44,045 INFO sqlalchemy.engine.base.Engine {'param_1': 13}
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "model.py", line 93, in __repr__
    self.inventory_id, self.beer_code, self.user_id, self.inventory)
AttributeError: 'Inventory' object has no attribute 'inventory'
>>> dir(hi.user)
['__class__', '__delattr__', '__dict__', '__doc__', '__format__', '__getattribute__', '__hash__', '__init__', '__mapper__', '__module__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__setattr__', '__sizeof__', '__str__', '__subclasshook__', '__table__', '__tablename__', '__weakref__', '__cached_tablename', '__decl_class_registry', '__sa_class_manager', '__sa_instance_state', 'email', 'inventories', 'iso', 'metadata', 'name', 'password', 'query', 'query_class', 'user_id']
>>> dir(hi.user.user_id)
['_abs_', '_add_', '_and_', '_class_', '_cmp_', '_coerce_', '_delattr_', '_div_', '_divmod_', '_doc_', '_float_', '_floordiv_', '_format_', '_getattribute_', '_getnewargs_', '_hash_', '_hex_', '_index_', '_init_', '_int_', '_invert_', '_long_', '_lshift_', '_mod_', '_mul_', '_neg_', '_new_', '_nonzero_', '_oct_', '_or_', '_pos_', '_pow_', '_radd_', '_rand_', '_rdiv_', '_rdivmod_', '_reduce_', '_reduce_ex_', '_repr_', '_rfloordiv_', '_rlshift_', '_rmod_', '_rmul_', '_ror_', '_rpow_', '_rrshift_', '_rshift_', '_rsub_', '_rtruediv_', '_rxor_', '_setattr_', '_sizeof_', '_str_', '_sub_', '_subclasshook_', '_truediv_', '_trunc_', '_xor_', 'bit_length', 'conjugate', 'denominator', 'imag', 'numerator', 'real']
>>> hi.user.user_id
13
>>> dir(hi.user.user_id)
['_abs_', '_add_', '_and_', '_class_', '_cmp_', '_coerce_', '_delattr_', '_div_', '_divmod_', '_doc_', '_float_', '_floordiv_', '_format_', '_getattribute_', '_getnewargs_', '_hash_', '_hex_', '_index_', '_init_', '_int_', '_invert_', '_long_', '_lshift_', '_mod_', '_mul_', '_neg_', '_new_', '_nonzero_', '_oct_', '_or_', '_pos_', '_pow_', '_radd_', '_rand_', '_rdiv_', '_rdivmod_', '_reduce_', '_reduce_ex_', '_repr_', '_rfloordiv_', '_rlshift_', '_rmod_', '_rmul_', '_ror_', '_rpow_', '_rrshift_', '_rshift_', '_rsub_', '_rtruediv_', '_rxor_', '_setattr_', '_sizeof_', '_str_', '_sub_', '_subclasshook_', '_truediv_', '_trunc_', '_xor_', 'bit_length', 'conjugate', 'denominator', 'imag', 'numerator', 'real']

```

```

>>> trade = Trade(initial_trade=4, reciprocal_trade=22, quantity=1)
>>> db.session.add(trade)
>>> successful_match_one = db.session.query(ISO).get(4)
2017-07-04 01:41:39,614 INFO sqlalchemy.engine.base.Engine INSERT INTO trades (traded_at, quantity, initial_trade, reciprocal_trade) VALUES (%(traded_at)s, %(quantity)s, %(initial_trade)s, %(reciprocal_trade)s) RETURNING trades.trade_id
2017-07-04 01:41:39,614 INFO sqlalchemy.engine.base.Engine {'initial_trade': 4, 'traded_at': datetime.datetime(2017, 7, 4, 1, 41, 39, 614057), 'reciprocal_trade': 22, 'quantity': 1}
2017-07-04 01:41:39,619 INFO sqlalchemy.engine.base.Engine SELECT iso.iso_id AS iso_iso_id, iso.beer_code AS iso_beer_code, iso.user_id AS iso_user_id, iso.quantity AS iso_quantity, iso.active AS iso_active
FROM iso
WHERE iso.iso_id = %(param_1)s
2017-07-04 01:41:39,620 INFO sqlalchemy.engine.base.Engine {'param_1': 4}
>>> successful_match_one = db.session.query(ISO).get(22)
2017-07-04 01:41:50,029 INFO sqlalchemy.engine.base.Engine SELECT iso.iso_id AS iso_iso_id, iso.beer_code AS iso_beer_code, iso.user_id AS iso_user_id, iso.quantity AS iso_quantity, iso.active AS iso_active
FROM iso
WHERE iso.iso_id = %(param_1)s
2017-07-04 01:41:50,029 INFO sqlalchemy.engine.base.Engine {'param_1': 22}
>>> successful_match_one.active = "False"
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: 'NoneType' object has no attribute 'active'
>>> dir(successful_match_one)
['_class_', '__delattr__', '__doc__', '__format__', '__getattribute__', '__hash__', '__init__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__setattr__', '__sizeof__', '__str__', '__subclasshook__']
>>> successful_match_one
>>> db.session.commit()
2017-07-04 01:44:41,772 INFO sqlalchemy.engine.base.Engine COMMIT
>>> trade = Trade(initial_trade=19, reciprocal_trade=9, quantity=1)
>>> db.session.add(trade)
>>> db.session.commit()
2017-07-04 01:49:53,372 INFO sqlalchemy.engine.base.Engine BEGIN (implicit)
2017-07-04 01:49:53,372 INFO sqlalchemy.engine.base.Engine INSERT INTO trades (traded_at, quantity, initial_trade, reciprocal_trade) VALUES (%(traded_at)s, %(quantity)s, %(initial_trade)s, %(reciprocal_trade)s) RETURNING trades.trade_id
2017-07-04 01:49:53,372 INFO sqlalchemy.engine.base.Engine {'initial_trade': 19, 'traded_at': datetime.datetime(2017, 7, 4, 1, 49, 53, 372789), 'reciprocal_trade': 9, 'quantity': 1}
2017-07-04 01:49:53,374 INFO sqlalchemy.engine.base.Engine COMMIT
>>> trade = Trade(initial_trade=7, reciprocal_trade=3, quantity=1)
>>> db.session.add(trade)
>>> db.session.commit()
2017-07-04 01:50:19,348 INFO sqlalchemy.engine.base.Engine BEGIN (implicit)
2017-07-04 01:50:19,349 INFO sqlalchemy.engine.base.Engine INSERT INTO trades (traded_at, quantity, initial_trade, reciprocal_trade) VALUES (%(traded_at)s, %(quantity)s, %(initial_trade)s, %(reciprocal_trade)s) RETURNING trades.trade_id
2017-07-04 01:50:19,349 INFO sqlalchemy.engine.base.Engine {'initial_trade': 7, 'traded_at': datetime.datetime(2017, 7, 4, 1, 50, 19, 349085), 'reciprocal_trade': 3, 'quantity': 1}
2017-07-04 01:50:19,350 INFO sqlalchemy.engine.base.Engine COMMIT
>>> trade = Trade(initial_trade=10, reciprocal_trade=2, quantity=1)
>>> db.session.add(trade)
>>> db.session.commit()
2017-07-04 01:51:08,327 INFO sqlalchemy.engine.base.Engine BEGIN (implicit)
2017-07-04 01:51:08,327 INFO sqlalchemy.engine.base.Engine INSERT INTO trades (traded_at, quantity, initial_trade, reciprocal_trade) VALUES (%(traded_at)s, %(quantity)s, %(initial_trade)s, %(reciprocal_trade)s) RETURNING trades.trade_id
2017-07-04 01:51:08,327 INFO sqlalchemy.engine.base.Engine {'initial_trade': 10, 'traded_at': datetime.datetime(2017, 7, 4, 1, 51, 8, 327493), 'reciprocal_trade': 2, 'quantity': 1}
2017-07-04 01:51:08,328 INFO sqlalchemy.engine.base.Engine COMMIT
>>> trade = Trade(initial_trade=1, reciprocal_trade=18, quantity=1)
>>> db.session.add(trade)
>>> db.session.commit()
2017-07-04 01:52:16,949 INFO sqlalchemy.engine.base.Engine BEGIN (implicit)
2017-07-04 01:52:16,949 INFO sqlalchemy.engine.base.Engine INSERT INTO trades (traded_at, quantity, initial_trade, reciprocal_trade) VALUES (%(traded_at)s, %(quantity)s, %(initial_trade)s, %(reciprocal_trade)s) RETURNING trades.trade_id
2017-07-04 01:52:16,949 INFO sqlalchemy.engine.base.Engine {'initial_trade': 1, 'traded_at': datetime.datetime(2017, 7, 4, 1, 52, 16, 949810), 'reciprocal_trade': 18, 'quantity': 1}
2017-07-04 01:52:16,951 INFO sqlalchemy.engine.base.Engine COMMIT
>>> trade = Trade(initial_trade=17, reciprocal_trade=2, quantity=1)
>>> db.session.add(trade)
>>> db.session.commit()
2017-07-04 01:53:01,741 INFO sqlalchemy.engine.base.Engine BEGIN (implicit)
2017-07-04 01:53:01,741 INFO sqlalchemy.engine.base.Engine INSERT INTO trades (traded_at, quantity, initial_trade, reciprocal_trade) VALUES (%(traded_at)s, %(quantity)s, %(initial_trade)s, %(reciprocal_trade)s) RETURNING trades.trade_id
2017-07-04 01:53:01,742 INFO sqlalchemy.engine.base.Engine {'initial_trade': 17, 'traded_at': datetime.datetime(2017, 7, 4, 1, 53, 1, 741840), 'reciprocal_trade': 2, 'quantity': 1}
2017-07-04 01:53:01,743 INFO sqlalchemy.engine.base.Engine COMMIT
>>>

```