

# SeSQL and SQLAlchemy guide

## Contents

<b>1 Introduction</b>	<b>1</b>
<b>2 Design</b>	<b>1</b>
<b>3 How to</b>	<b>1</b>
<b>4 Limitations</b>	<b>2</b>

## 1 Introduction

Since version 0.10, SeSQL can be used with other ORM than Django, for example SQLAlchemy. Not all features are supported, and this considered beta status.

## 2 Design

The SQLAlchemy version is designed to work with different databases for core data and SeSQL data. It can be two PostgreSQL database, or PostgreSQL for SeSQL and something else (was tested with Sqlite and MySQL) for the core data.

## 3 How to

To use SQLAlchemy with SeSQL you need to do three things :

1. Configure SeSQL to use SQLAlchemy, rhat will look like :

```
#  
# Select the ORM to use  
#  
from sesql.orm.alchemy import AlchemyOrmAdapter  
orm = AlchemyOrmAdapter()
```

2. Bind your `AlchemyOrmAdapter` with your program bootstrap code (to have it access its database, your sessionmaker for “normal data”, and to create SeSQL tables if required), with something like :

```
Session = sessionmaker(bind=engine)
sesql_config.orm.bind('postgresql:///sesql', Session)
sesql_config.orm.sync_db()
```

3. Express your queries with the Q object from `sesql.orm.alchemy`.

## 4 Limitations

1. You **need** to manually call `sync_db` at a point.
2. The management commands are not supported.
3. The SearchHistory feature is not supported.
4. SQLAlchemy bulk operations are not supported.