

# Describing a Database with Metadata

---



**Chris Venturini**

@ChrisMVenturini [www.chrisv.me](http://www.chrisv.me)



# Overview



Overview of SQLAlchemy foundational concepts

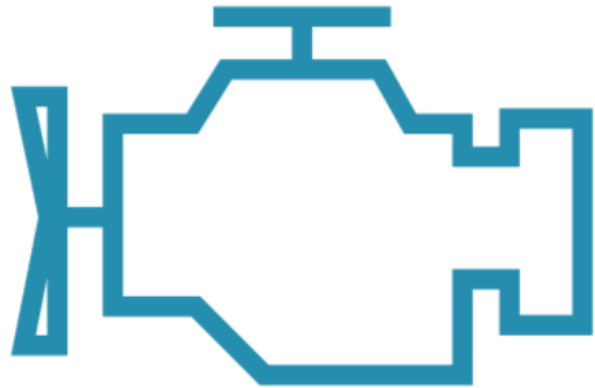
Connect to a SQLite Database with code!

What is metadata?

Define a table in code

Return data back from the defined table





## Engine

**Connects a Pool and Dialect together to provide a source of database connectivity and behavior.**

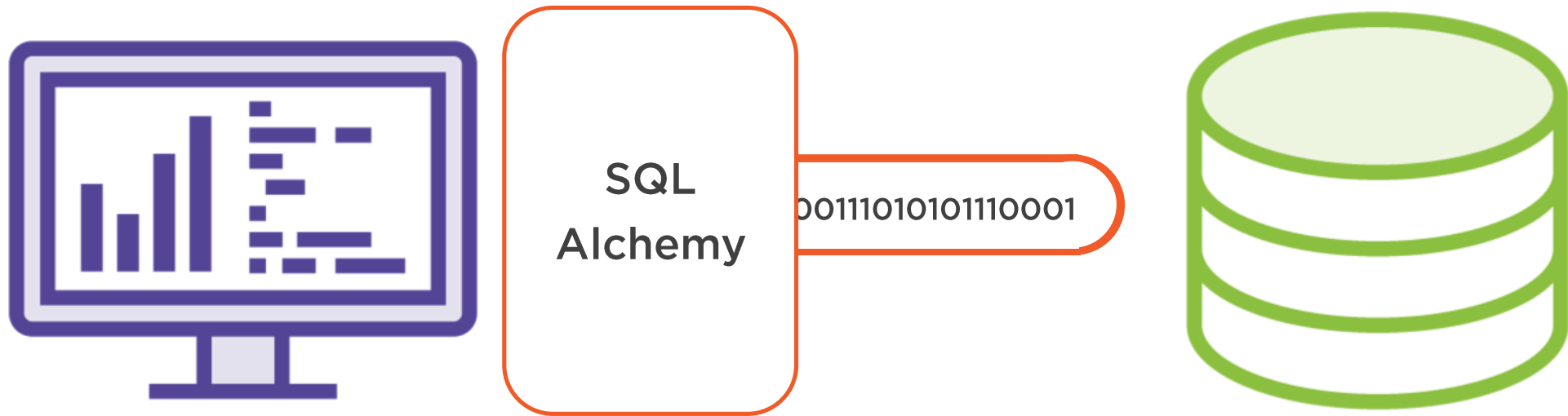
What is a Pool?

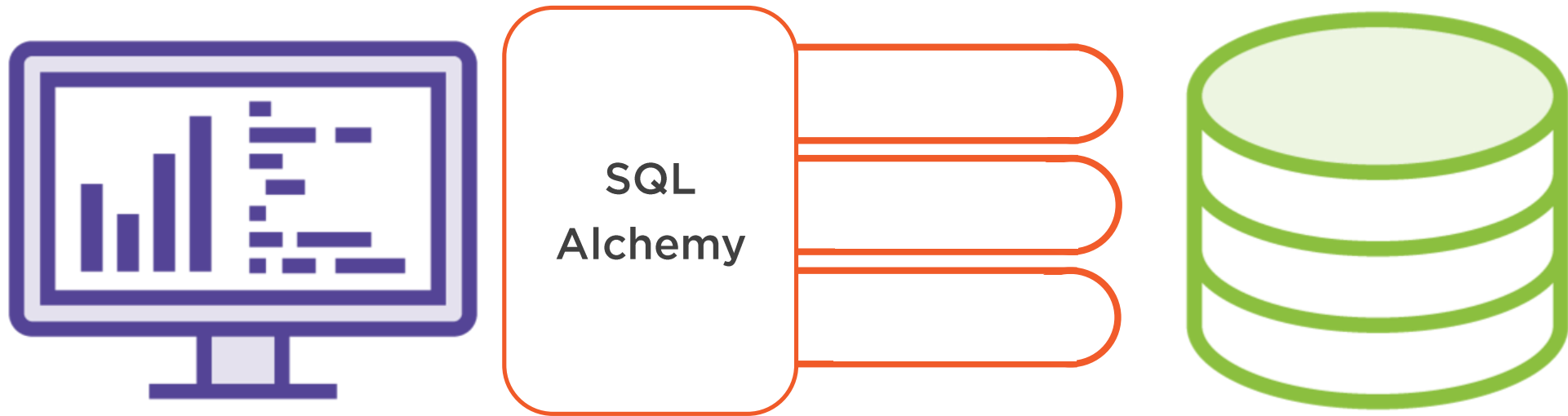




0111010100111010101110001







What is a Dialect?







```
CREATE TABLE user  
(  
  user_id INT NOT NULL PRIMARY KEY,  
);
```





ORACLE®



```
CREATE TABLE user
(  
  user_id numeric(10) not null,  
  CONSTRAINT user_pk PRIMARY KEY (user_id)  
);
```





SQL  
Alchemy

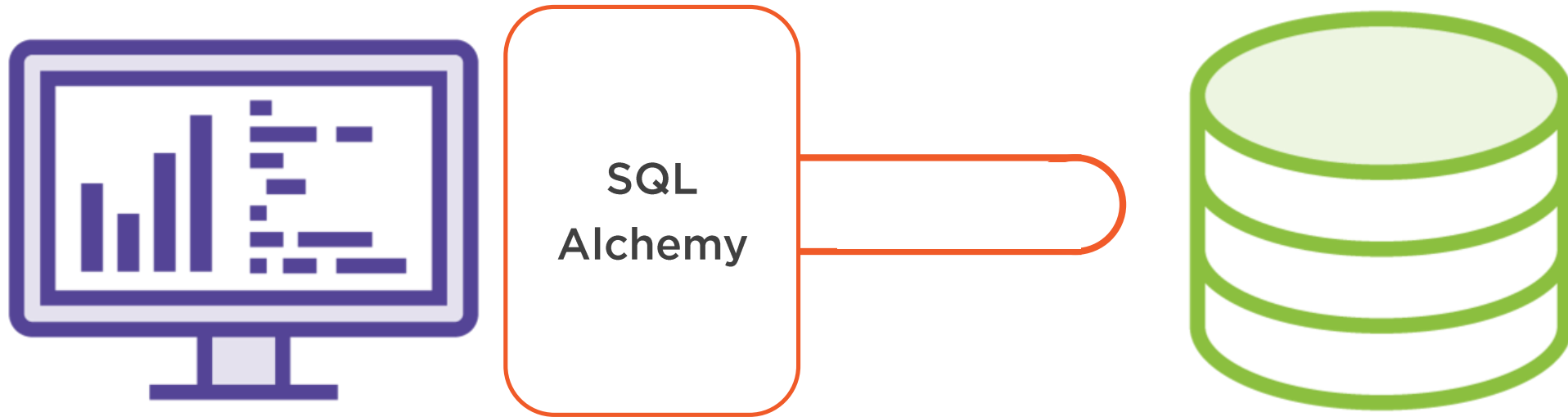


```
Table('user', metadata,  
      Column('user_id', Integer, primary_key=True)  
)
```

```
CREATE TABLE user  
(  
  user_id INT NOT NULL PRIMARY KEY,  
)
```



ORACLE®



```
Table('user', metadata,  
      Column('user_id', Integer, primary_key=True)  
)
```

```
CREATE TABLE user  
(  
  user_id numeric(10) not null,  
  CONSTRAINT user_pk PRIMARY KEY (user_id)  
);
```



# Database URLs

mysql://localhost/mydatabase



# Database URLs

dialect://sever/database



# Database URLs

sqlite://relative/path/to/db





## Metadata

**Metadata is a container object that keeps together many different features of a database being described.**





heroes



hero\_id

INTEGER

"hero\_id" INTEGER NOT NULL



name

VARCHAR(20)

"name" VARCHAR(20) NOT NULL UNIQUE



	hero_id	name
	Filter	Filter
1	1	Crareek
2	2	Tan
3	3	Huzzt
4	4	Pamble
5	5	Grimm

```
CREATE TABLE heroes (  
    hero_id INTEGER NOT NULL,  
    name VARCHAR(20) NOT NULL,  
    PRIMARY KEY (hero_id),  
    UNIQUE (name)  
)
```



```
heroes_table = Table('heroes', metadata,  
    Column('hero_id', Integer, nullable=False, primary_key=True),  
    Column('name', String(20), unique=True, nullable=False)  
)
```



```
Table('heroes', metadata,  
      Column('hero_id', Integer, nullable=False, primary_key=True),  
      Column('name', String(20), unique=True, nullable=False)  
)
```



Table(

)



```
Table('heroes',
```

```
)
```



```
Table('heroes', metadata,  
)
```





```
Table('heroes', metadata,  
      Column('hero_id', Integer, nullable=False, primary_key=True),  
      Column('name', String(20), unique=True, nullable=False)  
)
```



Column( )



```
Column('hero_id', )
```



```
Column('hero_id', Integer, )
```



```
Column('hero_id', Integer, nullable=False, primary_key=True)
```



There be dragons.....



`sqlalchemy.types.DECIMAL` **!=** `Decimal`



## Generic Column Data Types

`Column('[Column name]', [Column type], [Kwargs column attributes...])`

- BigInteger
- Binary
- Blob
- Boolean
- Char
- Date
- DateTime
- Decimal
- Enum
- Integer
- String

But wait! There's more! There is special support for vendor-specific types!

MySQL's BIGINT

Postgres' INET





## Keyword Argument Column Attributes

`Column('[Column name]', [Column type], [Kwargs column attributes...])`

- `autoincrement`
- `default`
- `index`
- `primary_key`



# HEROES of the Endless War



# Summary



## Overview of SQLAlchemy foundational concepts

- Engine
- Connection Pool
- Dialect

## Connected to a SQLite Database

## Overview of metadata

- Table
- Column

## Defined a table in code using metadata

## Returned data back from the defined table

