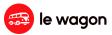# Database

Relational databases

Example 1 - Storing cities and countries in Excel

Example 2 – Teacher & Students
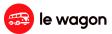
Schema
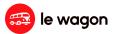
db.lewagon.org

SQL

```
SELECT
INSERT INTO
UPDATE
DELETE FROM
```
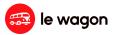
REST & SQL

SQLAlchemy

The most popular python **ORM**

flask-sqlalchemy.pocoo.org/2.3/quickstart/

```python
# wsgi.py
# [...]
app = Flask(__name__)
db = SQLAlchemy(app)
```

```python
# models.py
from wsgi import db

class Product(db.Model):
    __tablename__ = "products"
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(80), unique=True)
```
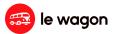
```
SELECT * FROM products
```

```
db.session.query(Product).all()
```

```sql
SELECT * FROM products WHERE id = 1
```

```python
db.session.query(Product).get(1)
```

`INSERT INTO products (name) VALUES ('Skello')`

```
product = Product()
product.name = "Skello"
db.session.add(product)
db.session.commit()
```

`UPDATE products SET name = 'Skello!' WHERE id = 1`

```python
product = db.session.query(Product).get(1)
product.name = "Skello!"
db.session.commit()
```

`DELETE FROM products WHERE id = 1`

```python
product = db.session.query(Product).get(1)
db.session.delete(product)
db.session.commit()
```

Alembic

For SQLAlchemy schema **updates**.

Package: `flask-migrate`
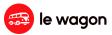
Run once:

```
pipenv run python manage.py db init
```

```
yourproject/
    alembic/
        env.py
        README
        script.py.mako
        versions/
            3512b954651e_add_account.py
            2b1ae634e5cd_add_order_id.py
            3adcc9a56557_rename_username_field.py
```

Change model definition then run:

```
pipenv run python manage.py db migrate -m "..."
```

```python
# migrations/versions/xxxxxxxx_create_products.py
"""

Revision ID: 5274a4a91f8a
Create Date: 2018-07-11 23:04:27.581840
"""


def upgrade():
    # ### commands auto generated by Alembic - please adj
    op.create_table('products',
    sa.Column('id', sa.Integer(), nullable=False),
    sa.Column('name', sa.String(), unique=True, nullable=
    sa.PrimaryKeyConstraint('id')
    )
    # ### end Alembic commands ###


def downgrade():
    # ### commands auto generated by Alembic - please adj
    op.drop_table('products')
    # ### end Alembic commands ###
```

Apply migration files to the actual Database schema:

```
pipenv run python manage.py db upgrade
```

NB: `downgrade` exists for production rollbacks

Deployment

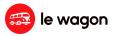**PostgreSQL** on Heroku

www.heroku.com/postgres

Marshmallow

For **serialization** (JSON) (as `flask.jsonify` does not work...)

Happy SQLAlchem-ing!