

Handout - SQLAlchemy

What is SQLAlchemy? It's...

- ... an Object Relational Mapper (ORM)
- ... a Python module
- ... divided into **SQLAlchemy ORM** and **SQLAlchemy Core**

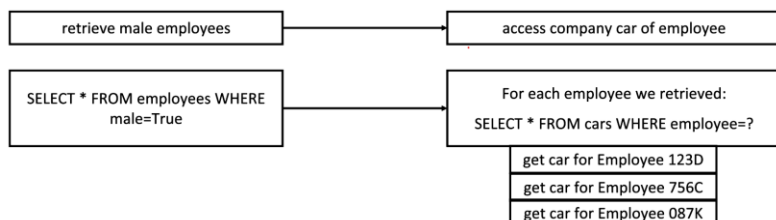
There are several relationship types:

- Type 1 - **One to One** **Employee** <¹-----¹> **Company Car**
- Type 2 - **Many to One | One to Many** **Employee** <¹-----ⁿ> **Company Phone**
- Type 3 - **Many to Many** **Employee** <ⁿ-----^m> **Project**

SQLAlchemy provides several loading strategies for relationships:

Lazy Loading (Basic Setting)

Select a relationship only when it is accessed

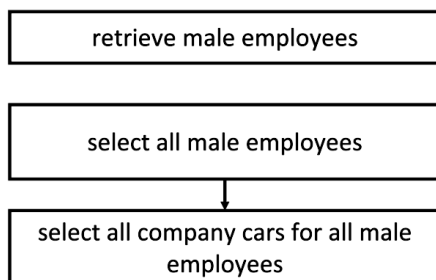


Code Example:

```
cars = relationship('Cars')
```

Subquery Eager Loading

Load a relationship instantly and select it via an additional query.

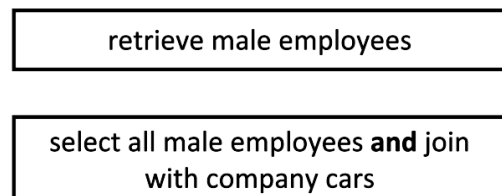


Code Example:

```
cars = relationship('Cars', lazy='subquery')
```

Joined Eager Loading

Load a relationship instantly and select it via a join to the original query.



Code Example:

```
cars = relationship('Cars', lazy='joined')
```

Basic transactional commands:

- Insert Object - `session.add(object)` # not committed yet
- Commit Session - `session.commit()`
- Select a table - `session.query(object).all()`
- Filter a table - `session.query(object).filter(condition).all()`