1**. lazy='select'** (по умолчанию):

Этот режим загружает связанные объекты только при обращении к ним.

class Parent(Base):

\_\_tablename\_\_ = 'parent'

id = Column(Integer, primary\_key=True)

name = Column(String(50))

children = relationship("Child", lazy='select')

class Child(Base):

\_\_tablename\_\_ = 'child'

id = Column(Integer, primary\_key=True)

name = Column(String(50))

parent\_id = Column(Integer, ForeignKey('parent.id'))

# Использование

parent = session.query(Parent).first()

print(parent.name) # Не загружает детей

print(parent.children) # Теперь загружает детей отдельным запросом

Результат: При обращении к parent.children будет выполнен отдельный SQL-запрос для загрузки детей.

2. lazy='joined':

Загружает связанные объекты сразу вместе с родительским объектом, используя JOIN.

class Parent(Base):

\_\_tablename\_\_ = 'parent'

id = Column(Integer, primary\_key=True)

name = Column(String(50))

children = relationship("Child", lazy='joined')

# Использование

parent = session.query(Parent).first()

print(parent.name)

print(parent.children) # Дети уже загружены

Результат: Один SQL-запрос загрузит и родителя, и всех его детей.

3. lazy='subquery':

Загружает связанные объекты отдельным подзапросом.

class Parent(Base):

\_\_tablename\_\_ = 'parent'

id = Column(Integer, primary\_key=True)

name = Column(String(50))

children = relationship("Child", lazy='subquery')

# Использование

parents = session.query(Parent).all()

for parent in parents:

print(parent.children) # Загружает всех детей одним подзапросом

Результат: Будет выполнен один запрос для загрузки родителей, а затем один подзапрос для загрузки всех детей.

4. lazy='dynamic':

Возвращает объект запроса вместо списка объектов, позволяя дополнительно фильтровать связанные объекты.

class Parent(Base):

\_\_tablename\_\_ = 'parent'

id = Column(Integer, primary\_key=True)

name = Column(String(50))

children = relationship("Child", lazy='dynamic')

# Использование

parent = session.query(Parent).first()

young\_children = parent.children.filter(Child.age < 10).all()

Результат: parent.children возвращает объект запроса, который можно дополнительно фильтровать перед выполнением.

5. lazy='immediate':

Загружает все связанные объекты сразу при первом обращении к любому из них.

class Parent(Base):

\_\_tablename\_\_ = 'parent'

id = Column(Integer, primary\_key=True)

name = Column(String(50))

children = relationship("Child", lazy='immediate')

# Использование

parent = session.query(Parent).first()

print(parent.children[0]) # Загружает всех детей

print(parent.children[1]) # Использует уже загруженные данные

Результат: При первом обращении к любому ребенку будут загружены все дети, последующие обращения используют кэшированные данные.