01-ORM相关模版代码

安装SQLAlchemy

```python
pip install "sqlalchemy[asyncio]" aiomysql
```

ORM 配置

```python
from sqlalchemy.ext.asyncio import async_sessionmaker, AsyncSession, create_async_engine

# 数据库URL
ASYNC_DATABASE_URL = "mysql+aiomysql://root:123456@localhost:3306/news_app?charset=utf8mb4"

# 创建异步引擎
async_engine = create_async_engine(
    ASYNC_DATABASE_URL,
    echo=True,  # 可选：输出SQL日志
    pool_size=10,  # 设置连接池中保持的持久连接数
    max_overflow=20  # 设置连接池允许创建的额外连接数
)

# 创建异步会话工厂
AsyncSessionLocal = async_sessionmaker(
    bind=async_engine,
    class_=AsyncSession,
    expire_on_commit=False
)

# 依赖项，用于获取数据库会话
async def get_db():
    async with AsyncSessionLocal() as session:
        try:
            yield session
            await session.commit()
        except Exception:
            await session.rollback()
            raise
        finally:
            await session.close()
```

News 模型类

新闻模型类   Python

```python
class News(Base):
    __tablename__ = "news"

    # 创建索引: 提升查询速度
    __table_args__ = (
        Index('fk_news_category_idx', 'category_id'),
        Index('idx_publish_time', 'publish_time')
    )

    id: Mapped[int] = mapped_column(Integer, primary_key=True, autoincrement=True, comment="新闻ID")
    title: Mapped[str] = mapped_column(String(255), nullable=False, comment="新闻标题")
    description: Mapped[Optional[str]] = mapped_column(String(500), comment="新闻简介")
    content: Mapped[str] = mapped_column(Text, nullable=False, comment="新闻内容")
    image: Mapped[Optional[str]] = mapped_column(String(255), comment="封面图片URL")
    author: Mapped[Optional[str]] = mapped_column(String(50), comment="作者")
    category_id: Mapped[int] = mapped_column(Integer, ForeignKey('news_category.id'), nullable=False, comment="分类ID")
    views: Mapped[int] = mapped_column(Integer, default=0, nullable=False, comment="浏览量")
    publish_time: Mapped[datetime] = mapped_column(DateTime, default=datetime.now, comment="发布时间")

    def __repr__(self):
        return f"<News(id={self.id}, title='{self.title}', views={self.views})>"
```