**一、分析说明题**

**信息系统的数据安全和个人隐私保护是社会与用户非常关心的热点问题，请举例说明作为数据库开发人员该如何遵从工程伦理、职业道德、国家法规开发实现数据库应用系统？**

作为数据库开发人员，遵从工程伦理、职业道德、国家法规开发实现数据库应用系统是非常重要的。以下是一些例子，说明如何遵从这些准则：

遵守数据隐私法规

作为数据库开发人员，需要遵守国家的数据隐私法规，例如《个人信息保护法》等。在设计和实现数据库应用系统时，需要考虑如何保护用户的个人信息，并采取相应的措施，例如加密、访问控制等。

保障数据安全

数据库开发人员需要确保数据库应用系统的数据安全，防止被未授权访问、篡改、删除等。为了保障数据安全，需要对数据进行备份、加密、监控等操作，以及限制用户的访问权限。

不滥用数据

数据库开发人员需要遵守职业道德，不得滥用数据库中的数据。例如，不得将用户的个人信息用于商业目的，不得将敏感数据泄露给未经授权的第三方等。

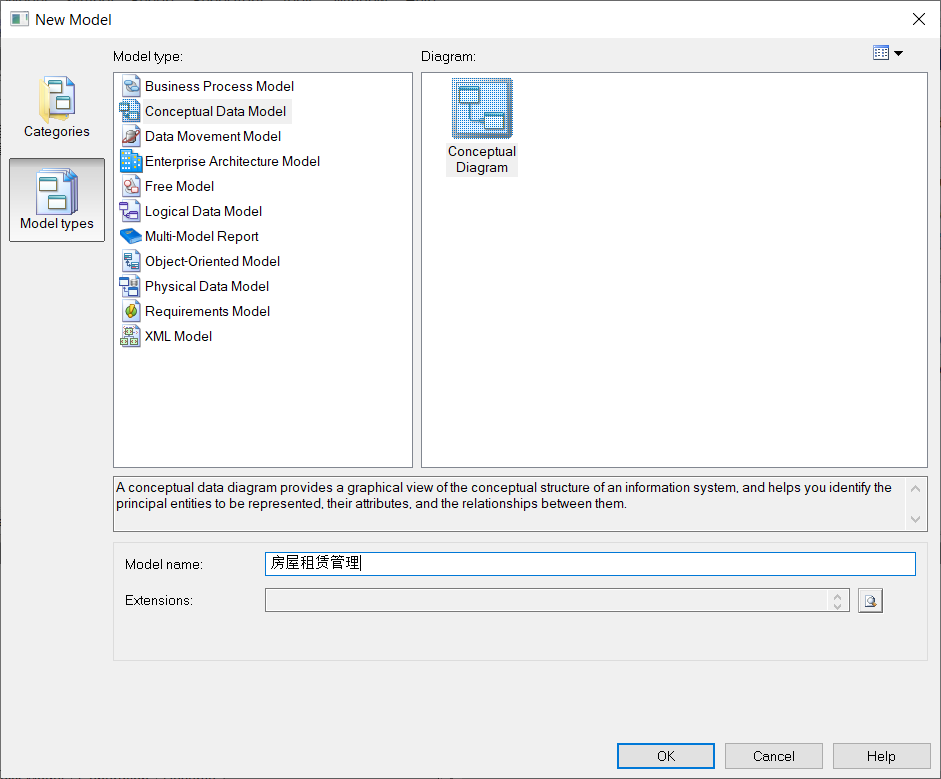
提高技术水平

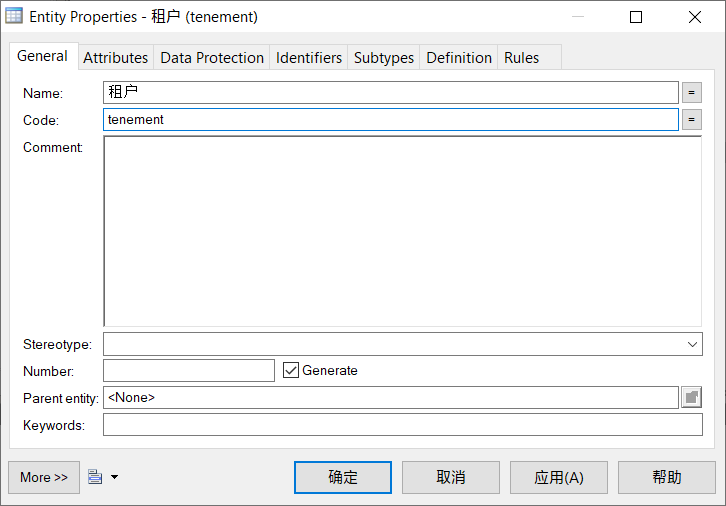
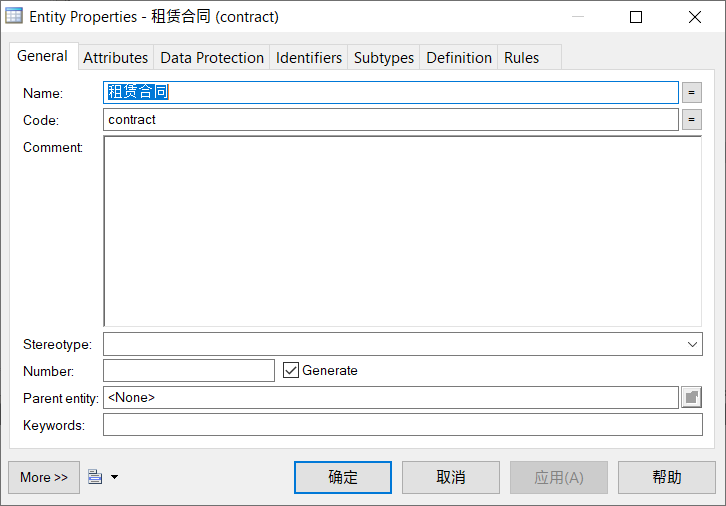
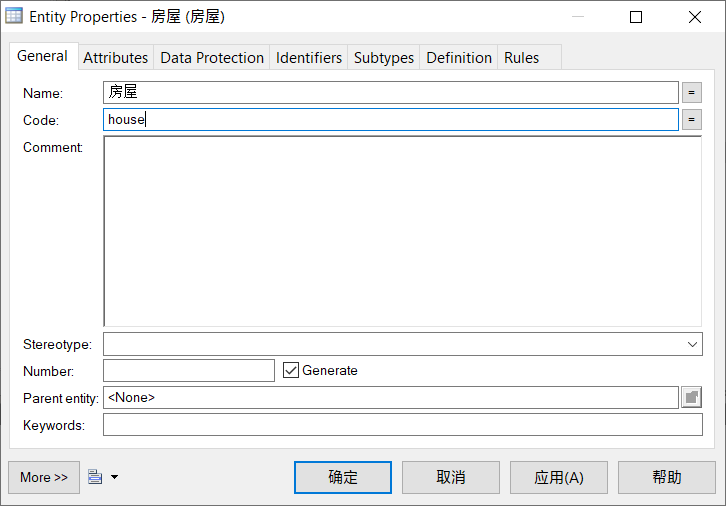
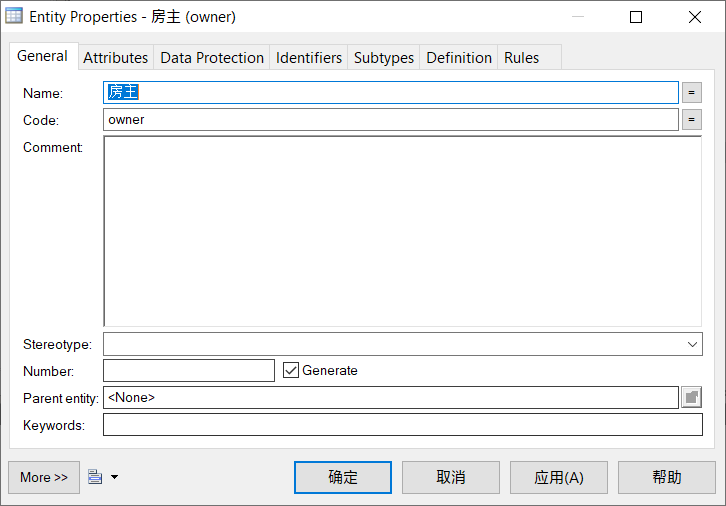
作为数据库开发人员，需要不断学习和提高技术水平，以便更好地设计和实现数据库应用系统。同时，也需要熟悉相关的法规和标准，以便在开发过程中遵从相应的准则。

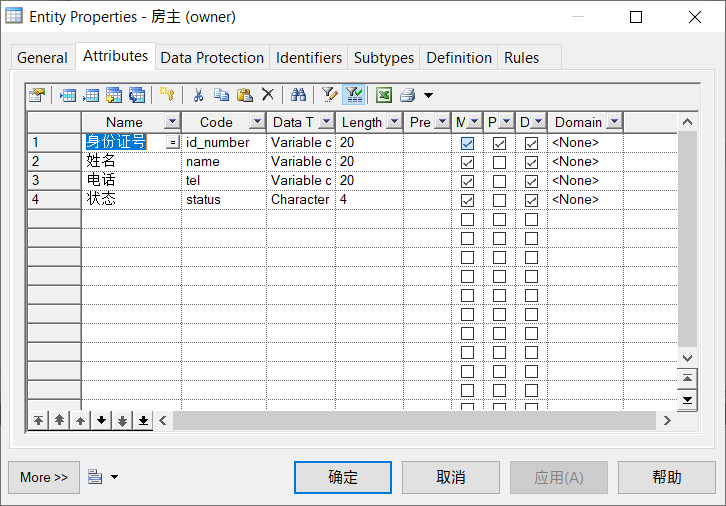
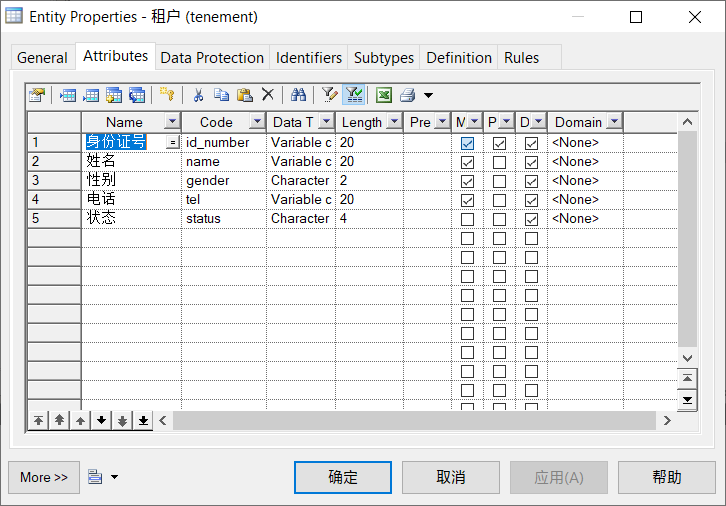
**二、数据库设计与编程实现动手实践题**

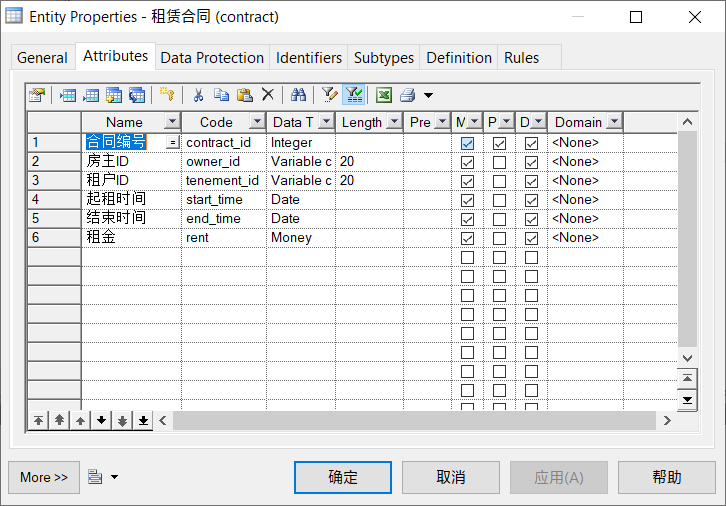
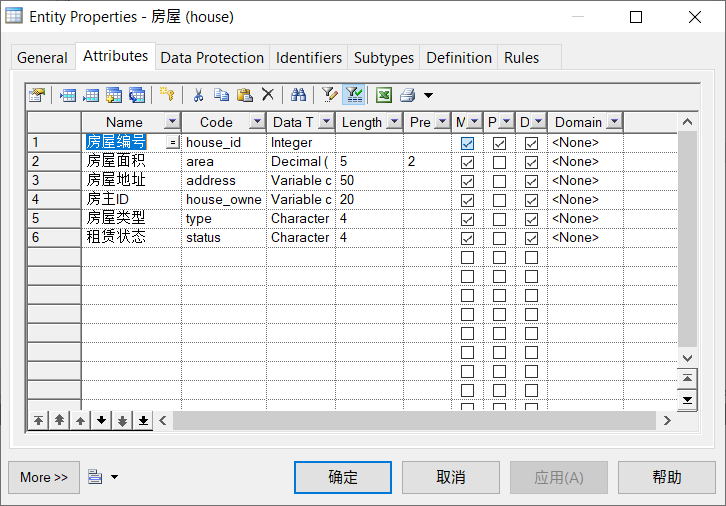
**分析房屋租赁管理系统数据需求，设计该系统数据库，并在PostgreSQL数据库服务器中实现该数据库。**

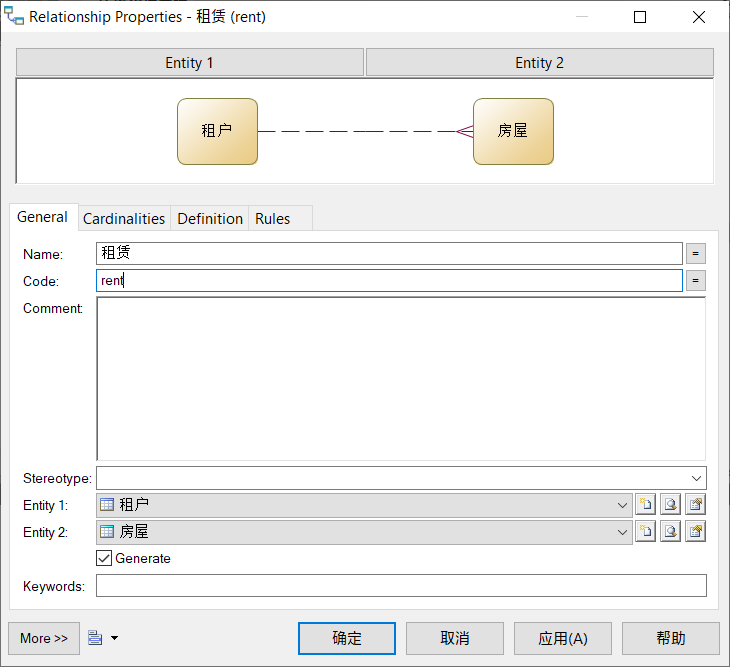
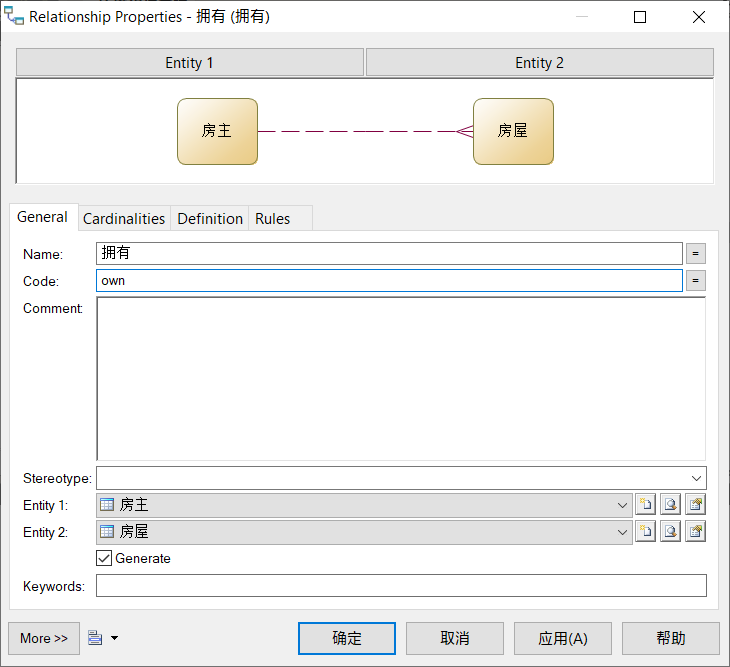
1）针对房屋租赁管理系统数据需求，使用Power Designer建模工具，设计房屋租赁管理系统概念数据模型。

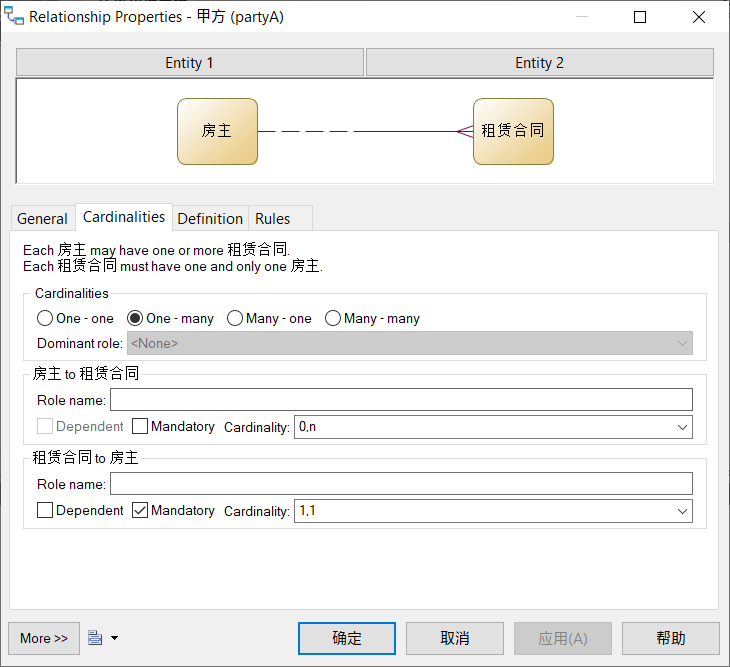
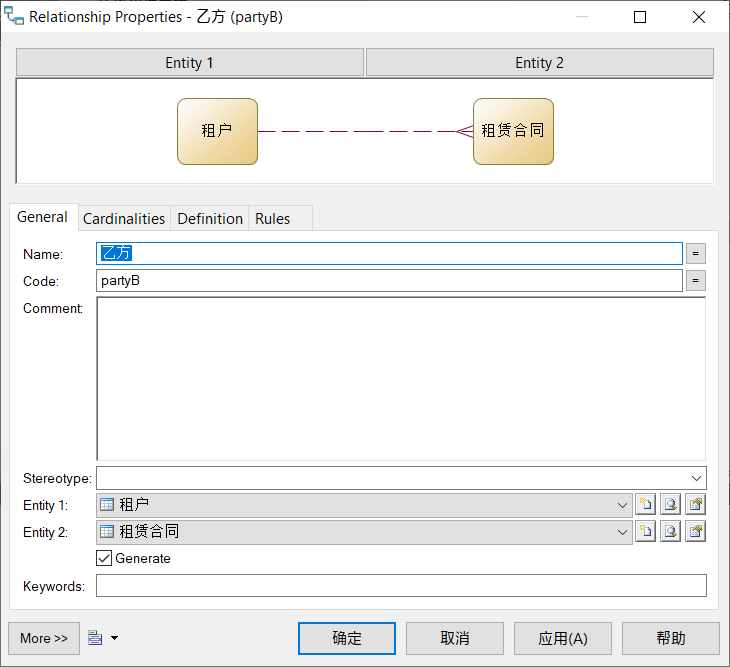
 在PowerDesigner中新建概念数据模型：

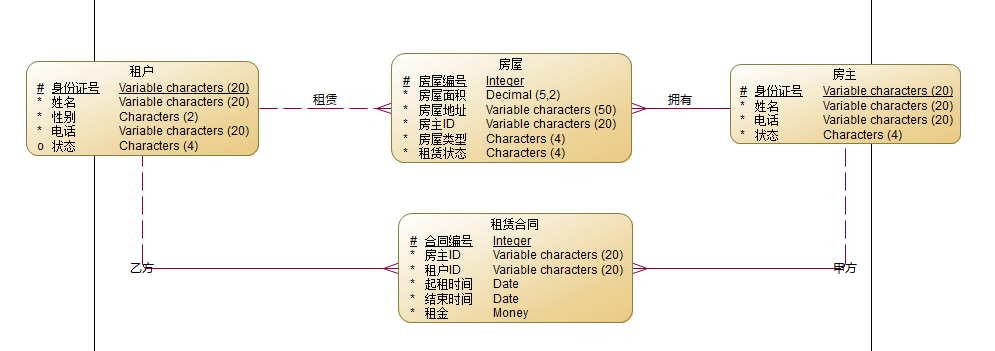
分别建立租户、房主、房屋、租赁合同四个实体：

为以上四个实体设置属性：

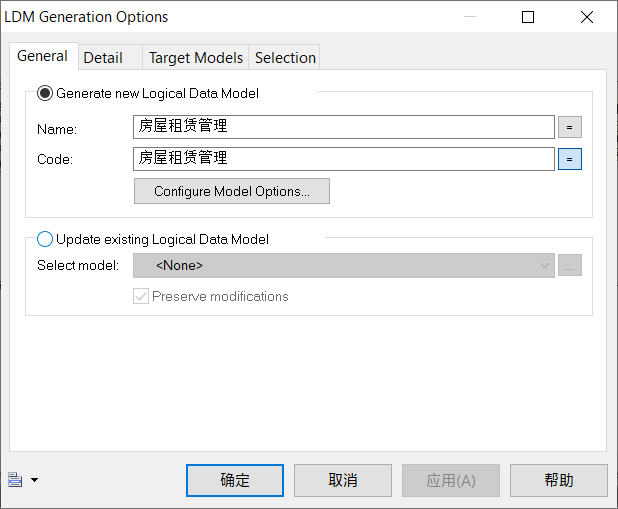


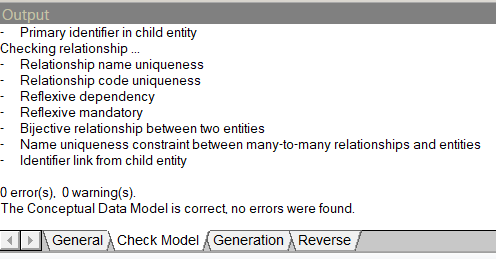
设置实体间联系：



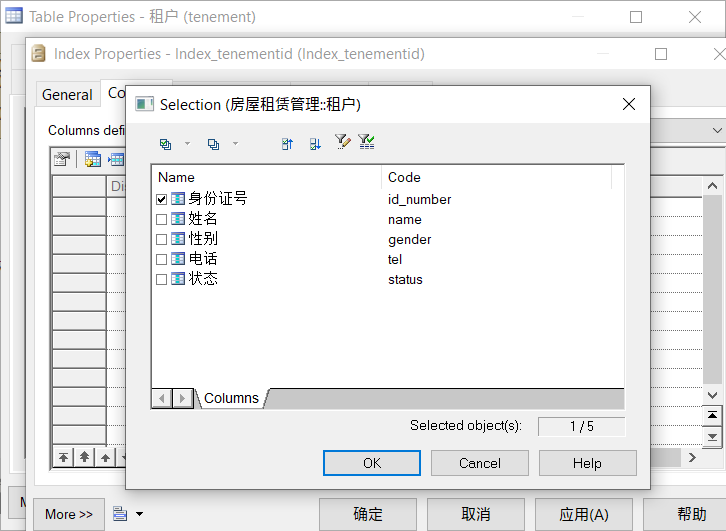
最终所得概念数据模型:

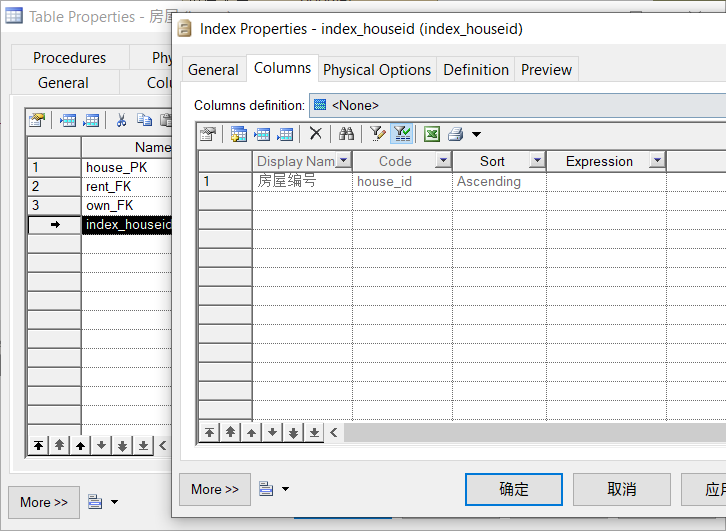
2）在概念数据模型基础上，采用关系数据库设计方式，将房屋租赁管理系统概念数据模型转换为系统逻辑数据模型设计，并进行规范化完善设计，至少满足3NF规范。

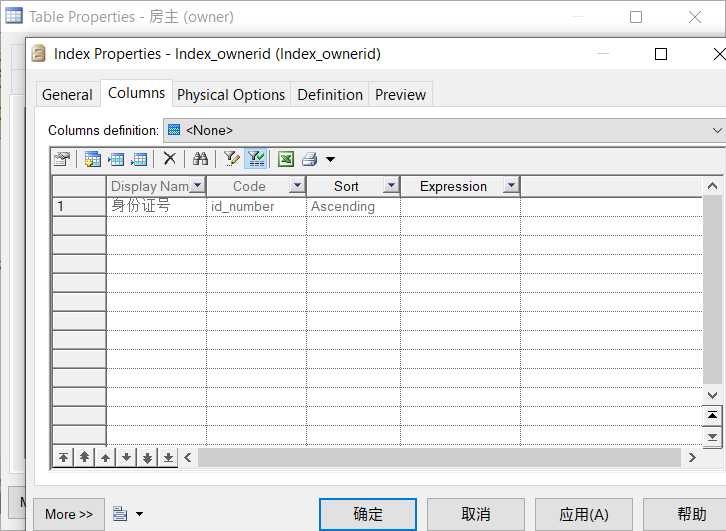
在概念数据模型基础上进行设计，本模型已满足3NF规范，使用PowerDesigner工具转换为逻辑数据模型：

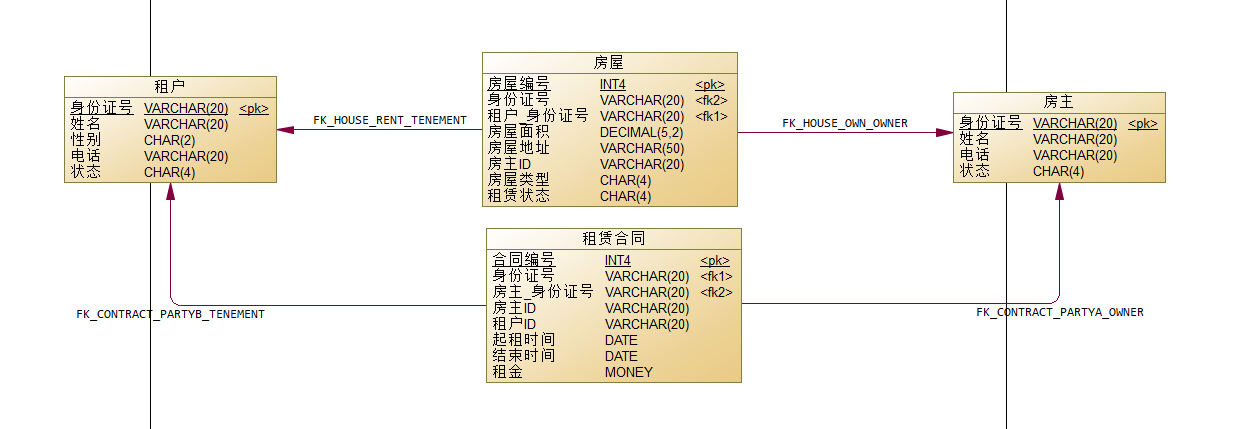
转换结果：

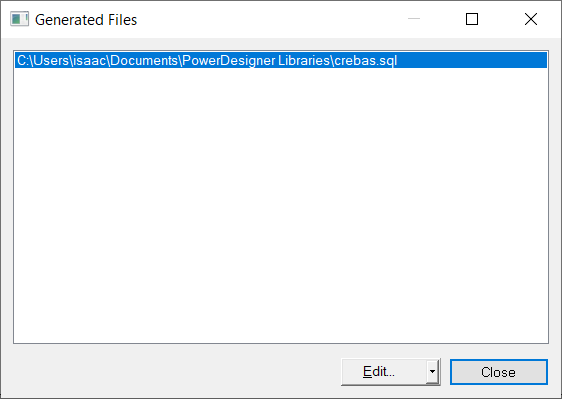
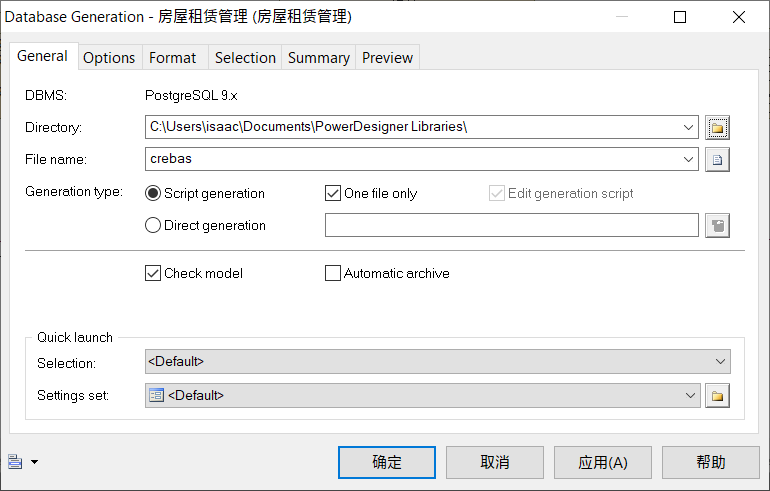
3）针对PostgreSQL数据库实现，将房屋租赁管理系统逻辑数据模型转换为系统物理数据模型设计，并进行有关索引设计、视图设计。

租户表索引设计：

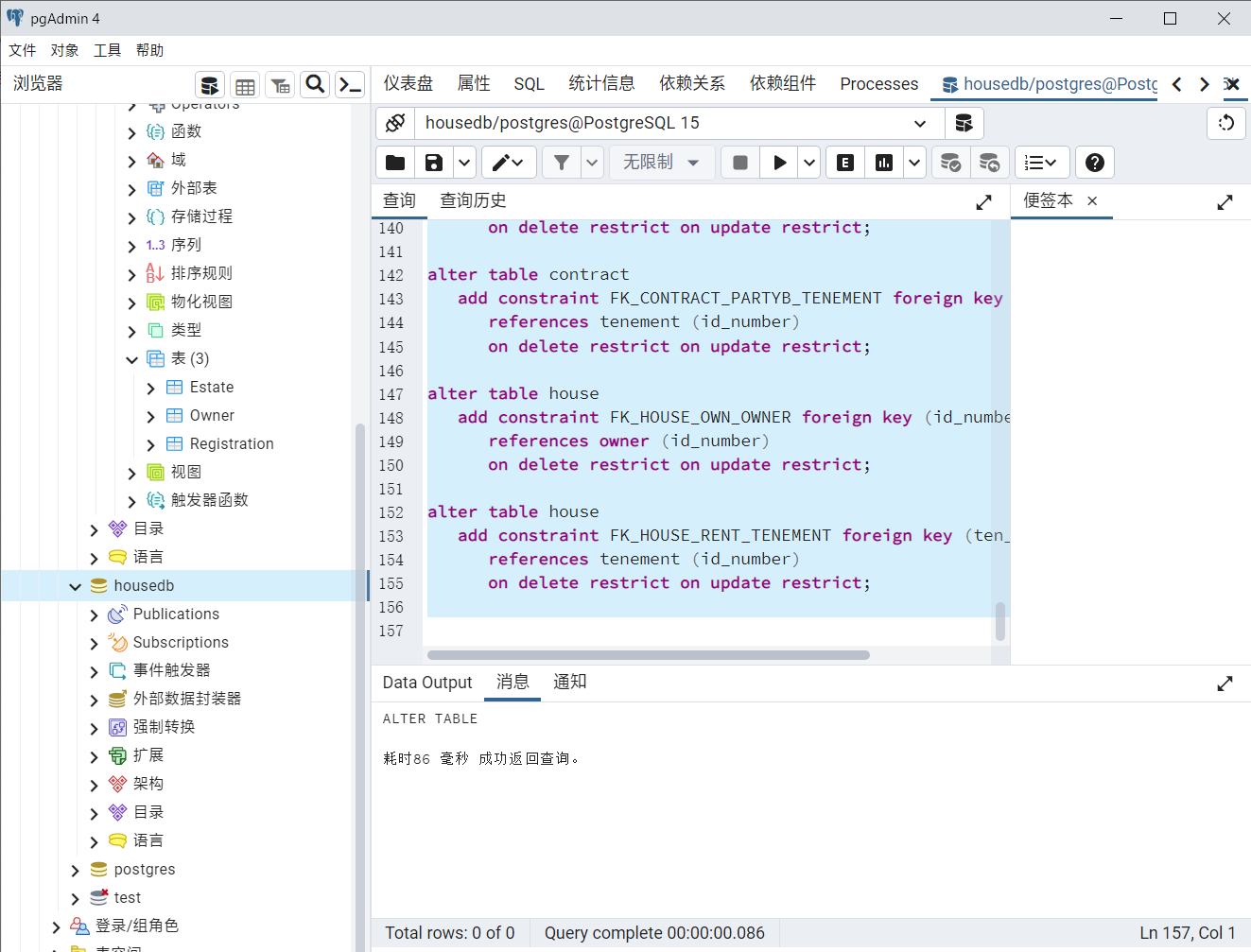
房屋表索引设计：

房主表索引设计：

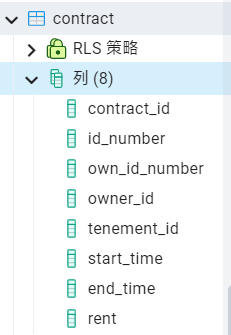
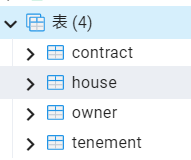
物理数据模型：

4）将房屋租赁管理系统物理数据模型转换为SQL脚本程序。

5）在PostgreSQL数据库服务器中，执行该SQL脚本程序，实现房屋租赁管理系统数据库对象创建实现。

将生成的SQL代码在pgAdmin4中运行，可见运行成功

检查数据库中表，可见创建成功：

****

**三、挑战性问题研究**

**针对电商系统的高并发访问、高性能存取、可扩展应用需求，在商品销售数据库设计与实现中，该如何来应对？请给出具体解决方案。**

针对电商系统的高并发访问、高性能存取、可扩展应用需求，在商品销售数据库设计与实现中，可以采取以下具体解决方案：

1. 数据库架构设计

采用分布式数据库架构，将数据库分成多个部分，每个部分可以单独扩展和部署。同时，可以采用主从复制和读写分离等技术，提高数据库的读写性能。

2. 数据库表设计

在设计数据库表时，需要考虑到高并发访问的需求，采用合适的表结构和索引，以提高查询性能。例如，可以将常用的查询条件作为索引字段，避免全表扫描。

3. 数据库缓存

采用缓存技术，将热点数据缓存到内存中，以提高访问速度。同时，可以使用分布式缓存技术，将缓存数据分散到多个节点上，提高缓存的可扩展性和稳定性。

4. 数据库分区

采用数据库分区技术，将数据分散到多个物理存储设备上，以提高访问速度和可扩展性。同时，可以采用动态分区技术，根据数据的访问情况自动调整分区策略。

5. 数据库事务设计

在设计数据库事务时，需要考虑到并发访问的需求，采用合适的隔离级别和锁机制，避免数据冲突和死锁。同时，可以采用分布式事务技术，保证多个节点上的事务的一致性和可靠性。

6. 数据库监控和优化

需要对数据库进行监控和优化，及时发现和解决潜在的性能问题。可以采用性能分析工具，对数据库的查询和操作进行分析，找出性能瓶颈并进行优化。