|  |  |  |
| --- | --- | --- |
| 2020春《数据库系统》实验报告 | | |
| 实验1：数据库设计与应用开发 | | |
| 姓名： | 学号： | 班级： |

1. 需求分析

**描述该应用中数据的结构和约束：**

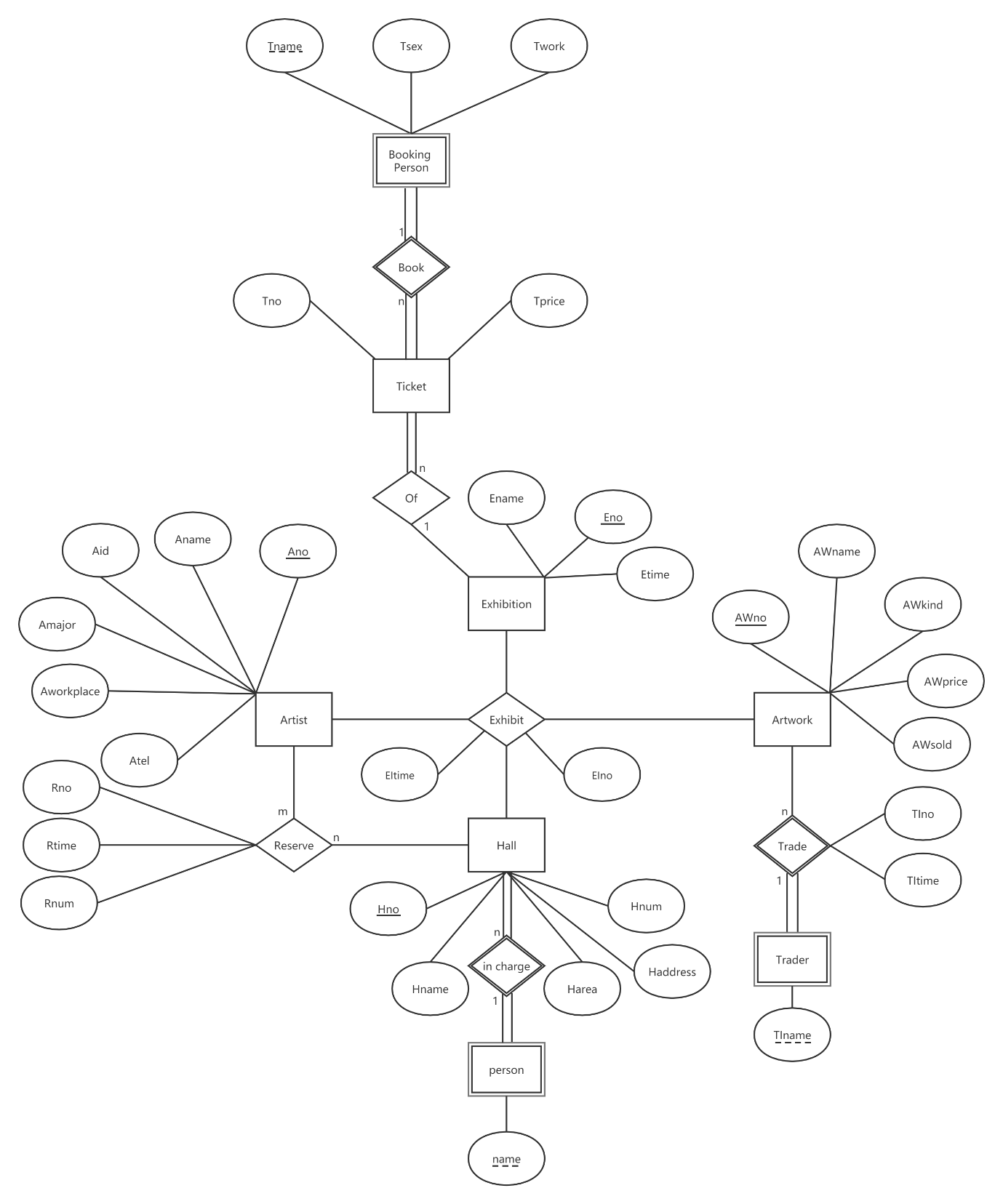
该数据库模型的设计是为了满足某一艺术宫举办艺术品展览(如绘画书法等)的日常需求，因此数据的结构和约束大体如下：

设计的实体包括主要包括艺术家Artist、艺术品Artwork、展馆Hall、展会Exhibition、门票Ticket、订购者BookingPerson、展馆负责人Person和艺术品买家Trader。订购者可以订购多张门票，每张门票有唯一的门票序号作为标识，且该门票只能用于参观一场展会；展馆负责人对一个或多个展馆负责；艺术品买家可以购买一个或多个艺术品；艺术家可以申请展馆(用于展示某项艺术品)，但不会直接生成相应的展会信息。

**描述该应用对数据进行哪些处理，有何性能要求：**

1. 因为存在用户想了解艺术展信息的情况，所以需要快速地根据某一艺术家姓名查询到其作品展出情况(考虑到艺术家姓名不能作为主键，即可能会有重名的情况，因此程序中尽量设计为先查出艺术家编号，再去查询其作品信息)
2. 观众应该具有订票、查票、退票的权限，所以需要根据票号快速查询到相应的信息。

2. 概念数据库设计



3. 逻辑数据库设计

**将ER模型转换为关系数据库模式：**

Artist(Ano,Aname,Aid,Amajor,Aworkplace,Atel)

Artwork(AWno,AWname,AWkind,AWprice,AWsold,TIno,TItime,TIname)

Hall(Hno,Hname,Haddress,Hnum,Hperson)

Exhibition(Eno,Ename,Etime)

ExhibitionInfo(EIno,EItime,Ano,AWno,Hno,Eno)其中Ano,AWno,Hno,Eno分别参照Artist.Ano、Artwork.AWno、Hall.Hno、Exhibition.Eno

Ticket(Tno,Tprice,Eno,Tname,Tsex,Twork)其中Eno参照Exhibition.Eno

Reserve(Rno,Rtime,Rnum,Ano,Hno)其中Ano,Hno分别参照Artist.Ano、Hall.Hno

**运用关系数据库规范化理论，对数据库模式进行规范化：**

问题：非主属性TItime,TIname依赖于TIno，而TIno依赖于AWno，因此TItime,TIname传递依赖于AWno。

解决办法：将Artwork分解为下面两个关系模式：

Artwork(AWno,AWname,AWkind,AWprice,AWsold)

TradeInfo(TIno,TItime,TIname,AWno)其中AWno参照Artwork.Awno

其他所有实体的属性均满足规范化要求，因此无其他改动。

4. 物理数据库设计

**根据数据库工作负载，运用反规范化和分表等方法，调整数据库模式：**

上一步骤中规范化后得到的Artwork和TradeInfo不存在反复连接用于查找的情况，并且二者从逻辑上也是较为独立的实体，因此不进行反规范化。

数据库模式设计时比较合理，因此没有适合分表的实体。

**根据数据库工作负载，设计数据库索引：**

考虑到用户需要按照姓名去查找对应的艺术家的编号，进而用于查找其作品的具体展览信息，因此在Artist上设计艺术家姓名Aname的索引。

5. 数据库建立

**选用一种RDBMS，使用SQL定义数据库的概念模式：**

**根据应用需求，设计数据库的外模式，并使用SQL定义数据库视图：**

由于经常需要查询购票信息、艺术品购买信息、展馆预定信息，因此创建相应的三个视图：

v\_Ticket(Tno,Tname,Tsex,Twork,Ename,Tprice)

v\_TradeInfo(TIno,TIname,TItime,AWname,AWkind,AWprice)

v\_Reserve(Rno,Aname,Hname,Rtime,Rnum)

使用SQL定义数据库索引。

6. 总结与体会

总结你的数据库设计方案，突出设计方案的亮点。

说明你在数据库设计过程中的心得体会。