

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：牛伟明

学号：1170300810

班级：1703108

葛润禅

1170300808

1703107

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。

数据库完整性：包括实体完整性、参照完整性和用户定义完整性。对每个表中都应设置主键约束，并检查主键是否唯一非空，如果不满足就拒绝插入或修改；表的定义中应存在外键约束，以便将其与另一个表联系起来；对表的某些属性设置非空、唯一或满足某一表达式，符合客观逻辑性(例如性别只能是男或女，售价必须大于 0 等)。

描述该应用对数据进行哪些处理，有何性能要求：

数据处理：

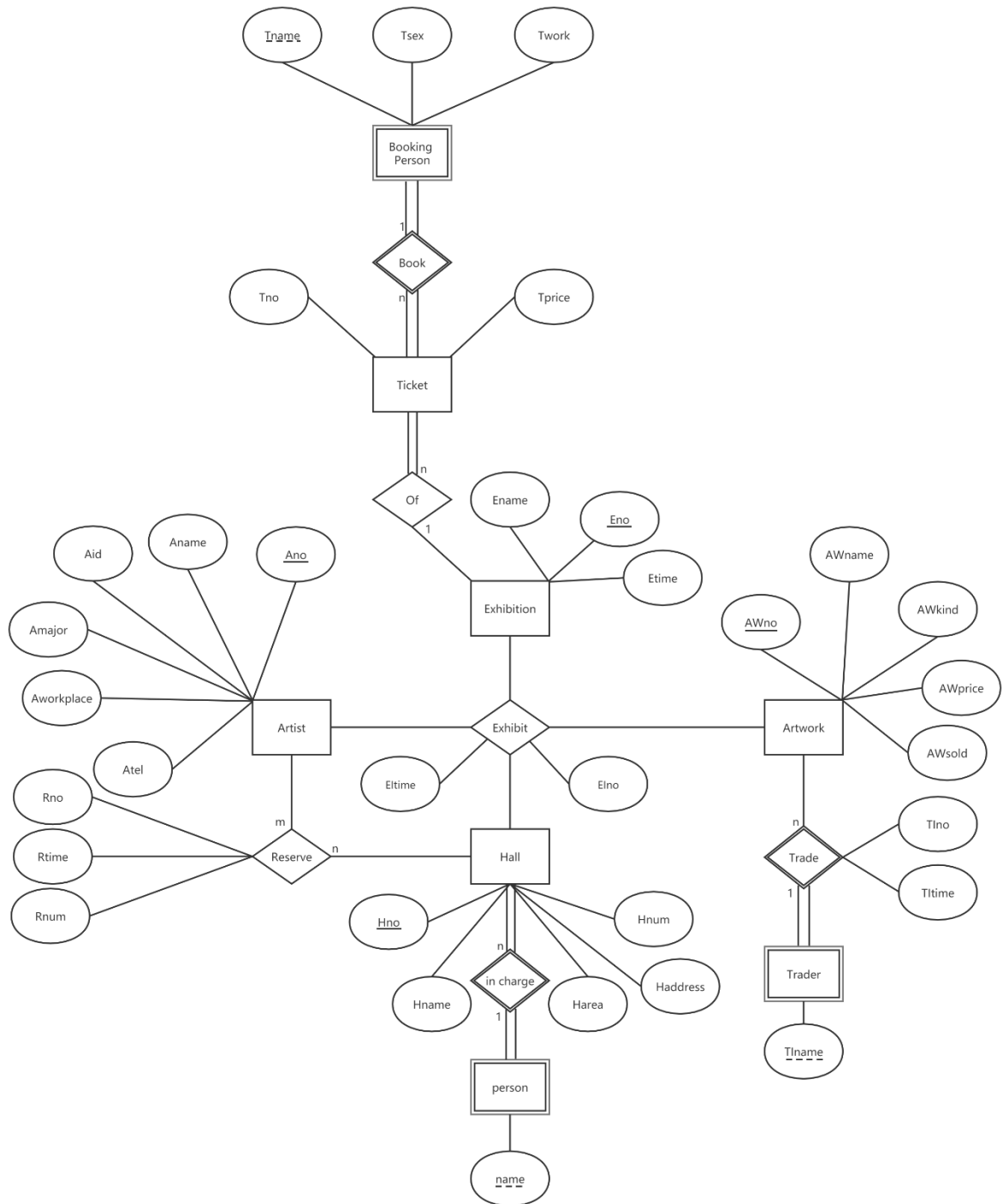
管理员 Root 应具有所有数据的权限，即对艺术家、艺术品、展厅、展会以及展览信息的增删改查。User 具有的权限相对较少，有预约展厅、购买门票、购买艺术品以及对面向客户的信息的查询等。

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，所以需要根据票码快速查询到相应的信息。
3. 艺术品买家在考虑购买艺术品时，艺术品的种类和价格会是其考虑的主要因素，因此程序应能快速地根据种类和价格查询到相应的艺术品。
4. 多数查询是基于某一编号进行的，所以以编号作为查询条件时，查询应较快。

2. 概念数据库设计

数据库 ER 模型对应 ER 图如下：



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,Eltime,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 Tltime,Tlname 依赖于 TIno，而 TIno 依赖于 AWno，因此 Tltime,Tlname 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,Tltime,Tlname,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：

这里参考了需求分析中的性能要求，逐一设计对应的索引以满足需求。

1. 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。
2. 票务方面需要将票码设计成索引，然而定义时 Tno 是主键，因此会自动形成主索引
3. 艺术品需要按照种类和价格区间分别索引，因此设计 AWkind 的索引和 AWprice 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：

使用 Mysql。详情见附件 AG.sql。

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：

外模式：共存在两个外模式，分别针对管理员 Root 和普通用户 user。数据库中包含的 Artist、艺术品买家、观众(买票的人)均属于 User(这里主要是考虑了三者的身份有重叠，例如一个观众完全可能购买艺术品，或一个 Artist 也有可能去参观其他人的作品，因此将三者独立不符合逻辑)。管理员 Root 应具有所有数据的权限，即对艺术家、艺术品、展厅、展会以及展览信息的增删改查。User 具有的权限相对较少，有预约展厅、购买门票、购买艺术品以及对几乎所有属性的查询等。

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,Eno,Tprice)
```

```
v_TradeInfo (TIno,TIname,TItime,AWname,AWkind,AWprice)
```

```
v_reserve(Rno,Hname,Aname,Rtime,Rnum)
```

```
v_exhibitionInfo(EIno,Aname,AWname,Hname,Ename,EItime,AWkind,AWprice)
```

具体定义见 AG.sql。

使用 SQL 定义数据库索引：

```
CREATE INDEX i_name ON Artist (Aname);  
CREATE INDEX i_kind ON Artwork (AWkind);  
CREATE INDEX i_price ON Artwork (AWprice);
```

其他主键索引和外键索引皆在定义时声明。

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：

1. 从需求出发，并着重考虑数据性能需求，索引设计得比较合理，充分满足了用户需求。
2. 对数据库的各类概念覆盖得较为全面，尽可能用到多方面的知识，例如 n 元联系、弱实体型等。
3. 尽可能地完成数据库建立的完整流程。完全按照概念、逻辑、物理的流程建立数据库，保证了关系的完整性。例如设计实体时首先考虑实体的主键；设计联系时先考虑外键的参照关系，再考虑联系另一边的实体。
4. 关系数据库设计得比较合理，因而后续需要规范化的实体只有一个，并且此次规范化后的结果是在最初考虑范围内的，因此也无需相应的反规范化。
5. 实体的设计是从其本身表达的含义出发的，因此之间相互独立，数据冗余度很低。

说明你在数据库设计过程中的心得体会：

本次数据库实验设计得非常完善，对知识的考察很全面，我在实验的过程中很多次需要参考 ppt 中的知识内容，因此也很好地巩固了知识。同时真正理解了数据库的含义与作用，体会到了数据库的优点。