

# 数据模型设计说明

## 一、编写目的

在开发一个信息系统时，一个贯穿于整个开发过程的问题就是数据以及对数据的加工。数据通常存放在数据库中，因此，数据库设计是信息系统设计的主要工作。数据库设计的核心是确定一个合适的数据模型，这个数据模型应当满足以下三个要求：

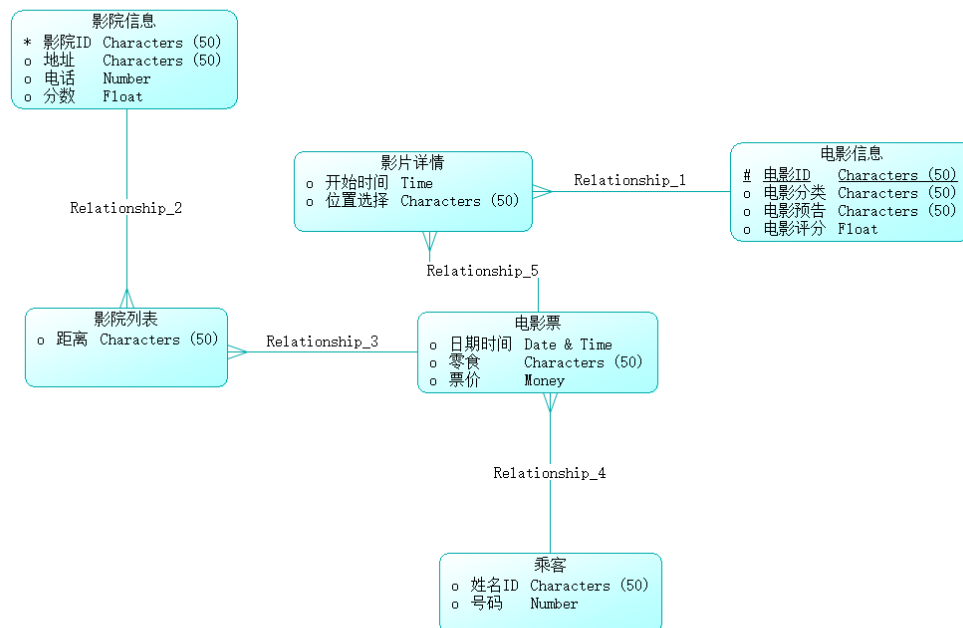
（1）符合用户的要求。既能包含用户需要处理的所有数据，又能支持用户提出的所有处理功能的实现；

（2）能被某个现有的数据库系统（DBMS）所接受，如 SQL Server、ORACLE、INFORMIX 等；

（3）具有较高的质量，如易于理解、便于维护、没有数据冲突、完整性好、效率高等。此外，在数据库设计中还要考虑数据库的安全问题，使各类用户合理操作数据库。

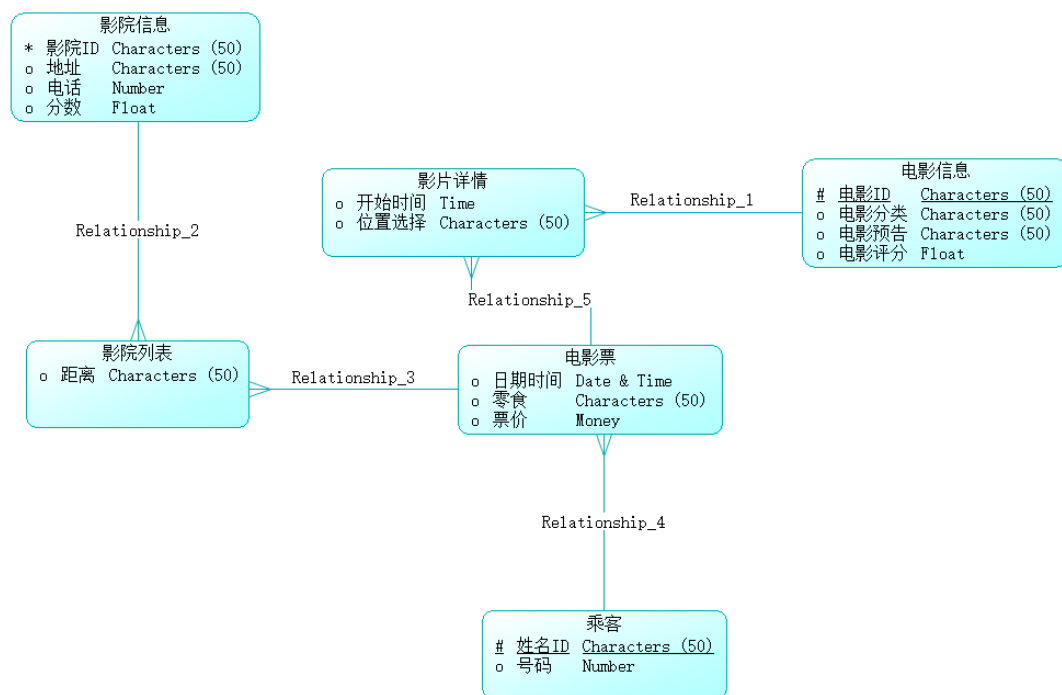
## 二、CDM

CDM 完成可以完成以下工作：数据图像化，形象化；生成 LDM，PDM。我们采用 E-R 图来表示。如影院信息（实体）与影院列表（实体）的关系，其中每个实体都有其各自的属性。



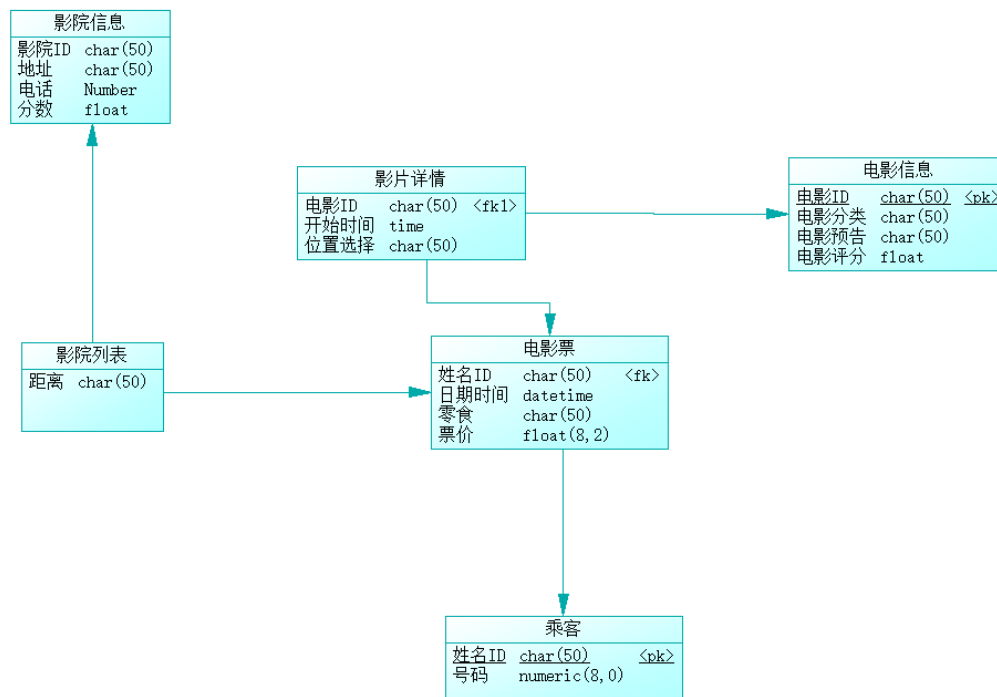
## 三、LDM

LDM 用于完成信息系统的逻辑设计，它独立于 DBMS 但却可以进行外部键，索引，视图等对象的设计工作。



#### 四、PDM

CDM 完成的是信息系统的概要设计，但在数据库的物理设计阶段必须在此基础上进行详细的后台设计，只有将 CDM 转换成 PDM 才能完成该阶段的设计工作。在 CDM 生成 PDM 之前必须要选择一种 DBMS 作为目标数据库，CDM 中定义的实体属性的数据类型将转换为目标数据库对的数据类型。CDM 生成 PDM 时，其中的对象要转换成 PDM 中的对象，即物理数据库中的对象。



## 五、代码

drop table if exists 乘客;

drop table if exists 影片详情;

drop table if exists 影院信息;

drop table if exists 影院列表;

drop table if exists 电影信息;

drop table if exists 电影票;

```

/*=====*/
/* Table: 乘客                                     */
/*=====*/
create table 乘客
(
    nameID          char(50) not null,
    number          numeric(8,0),
    primary key (nameID)
);
  
```

```

/*=====*/
/* Table: 影片详情 */
/*=====*/
create table 影片详情
(
    movieID          char(50) not null,
    starttime        time,
    chosenseat       char(50)
);

/*=====*/
/* Table: 影院信息 */
/*=====*/
create table 影院信息
(
    CinemaID         char(50) not null,
    address           char(50),
    telephoneNumber  Number,
    score             float
);

/*=====*/
/* Table: 影院列表 */
/*=====*/
create table 影院列表
(
    dsitance         char(50)
);

/*=====*/
/* Table: 电影信息 */
/*=====*/
create table 电影信息
(
    movieID          char(50) not null,
    movielevel       char(50),
    filmreview       char(50),
    moviescore       float,
    primary key (movieID)
);

/*=====*/
/* Table: 电影票 */
/*=====*/

```

```
create table 电影票
```

```
(  
    nameID          char(50) not null,  
    datetime        datetime,  
    onlineSnackSALE char(50),  
    price            float(8,2)  
);
```

```
alter table 影片详情 add constraint FK_Relationship_1 foreign key (movieID)  
    references 电影信息 (movieID) on delete restrict on update restrict;
```

```
alter table 影片详情 add constraint FK_Relationship_5 foreign key ()  
    references 电影票 on delete restrict on update restrict;
```

```
alter table 影院列表 add constraint FK_Relationship_2 foreign key ()  
    references 影院信息 on delete restrict on update restrict;
```

```
alter table 影院列表 add constraint FK_Relationship_3 foreign key ()  
    references 电影票 on delete restrict on update restrict;
```

```
alter table 电影票 add constraint FK_Relationship_4 foreign key (nameID)  
    references 乘客 (nameID) on delete restrict on update restrict;
```