

华东师范大学计算机科学与技术实验报告

实验课程：数据库系统实践	年级：2018	实验成绩：
实验名称：实验9	姓名：李泽浩	实验日期：2021/5/24
实验编号：9	学号：10185102142	实验时间：15:00-16:40
指导教师：窦亮	组号：	

一、设计题

- (1) ER图如下
- (2) 关系模式如下

二、上机题

- (1) 写出下面的SQL查询语句。
 - (a)查询会员信息中姓“赵”的会员信息，输出会员姓名、会员等级、会员折扣

```
SELECT memberName, memberShip, discount
FROM memberinfo
WHERE memberName LIKE "赵%";
#1-1
```

memberName	memberShip	discount	
赵一	A	0.5	
赵二	B	0.7	
赵三	C	0.9	
赵四	D	1.0	

- (b)查询“这个杀手不太冷”这部电影的会员票价最高的出票详细信息，输出座位编号、放映编号、会员票价、会员评分

```

SELECT t.seatID, t.playID, t.finalPrice, t.grade
FROM ticketinfo as t,(
    SELECT playID
    FROM filminfo as f INNER JOIN playinfo as p ON f.filmID = p.filmID
    WHERE f.filmName LIKE "这个杀手不太冷") as P
WHERE t.playID = P.playID
ORDER BY finalPrice DESC
LIMIT 1;
#1-2

```

seatID	playID	finalPrice	grade
06-18	12	178.00	8.0

(c)查询放映信息中票价在100到200之间的放映信息，输出放映编号、电影编号、电影名称、电影类别名称、原始票价

```

SELECT p.playID, p.filmID, f.filmName, f.typeID, p.price
FROM playinfo as p INNER JOIN filminfo as f ON p.filmID = f.filmID
WHERE p.price > 100 AND p.price < 200;
#1-3

```

信息 结果 1 剖析					
playID	filmID	filmName	typeID	price	
7	GH19540107900001	百万英镑	XJ	168.00	
8	GH19540107900001	百万英镑	XJ	168.00	
9	GH19540107900001	百万英镑	XJ	168.00	
10	GH19540107900001	百万英镑	XJ	168.00	
11	GH19540107900001	百万英镑	XJ	168.00	
12	GH19940914600001	这个杀手不太冷	GF	178.00	
13	GH19940914600001	这个杀手不太冷	GF	178.00	
14	GH19940914600001	这个杀手不太冷	GF	178.00	
15	GH19940914600001	这个杀手不太冷	GF	178.00	
16	GH19940914600001	这个杀手不太冷	GF	178.00	
17	GH19940914600001	这个杀手不太冷	GF	178.00	
3	GH20080618400001	倚天屠龙记	GF	198.00	

(d)查询各门店放映厅排片信息，输出放映编号、电影编号、门店地址、电影名称、电影类别编号、演员、导演和原始票价（如果该门店放映厅没有排片信息，也要求输出相关信息）

#有排场信息的影院

```
(SELECT p.playID, p.filmID, c.address, f.filmName, f.typeID, f.director, p.price
FROM cinema_hallinfo as c, filminfo as f, playinfo as p
WHERE c.cinema_hallID = p.cinema_hallID
      AND f.filmID = p.filmID)
```

UNION

#没有任何排片的影院

```
(SELECT 'NULL', 'NULL', c.address, 'NULL', 'NULL', 'NULL', 'NULL'
FROM cinema_hallinfo as c
WHERE c.cinema_hallID NOT IN(SELECT cinema_hallID FROM playinfo) = 1)
#1-4
```

```
4  AND f.filmID = p.filmID)
```

信息 结果 1 剖析 状态

playID	filmID	address	filmName	typeID	director	price
7	GH19540107900001	杨浦店	百万英镑	XJ	罗纳德·尼姆	168.00
8	GH19540107900001	杨浦店	百万英镑	XJ	罗纳德·尼姆	168.00
9	GH19540107900001	杨浦店	百万英镑	XJ	罗纳德·尼姆	168.00
10	GH19540107900001	杨浦店	百万英镑	XJ	罗纳德·尼姆	168.00
11	GH19540107900001	杨浦店	百万英镑	XJ	罗纳德·尼姆	168.00
18	GH19890602600002	杨浦店	死亡诗社	QT	彼得·威尔	218.00
19	GH19890602600002	杨浦店	死亡诗社	QT	彼得·威尔	218.00
20	GH19890602600002	杨浦店	死亡诗社	QT	彼得·威尔	218.00
21	GH19890602600002	杨浦店	死亡诗社	QT	彼得·威尔	218.00
22	GH19890602600002	杨浦店	死亡诗社	QT	彼得·威尔	218.00
23	GH19890602600002	杨浦店	死亡诗社	QT	彼得·威尔	218.00
24	GH19890602600002	杨浦店	死亡诗社	QT	彼得·威尔	218.00
25	GH19890602600002	杨浦店	死亡诗社	QT	彼得·威尔	218.00
12	GH19940914600001	杨浦店	这个杀手不太冷	GF	吕克·贝松	178.00
13	GH19940914600001	杨浦店	这个杀手不太冷	GF	吕克·贝松	178.00
14	GH19940914600001	杨浦店	这个杀手不太冷	GF	吕克·贝松	178.00
15	GH19940914600001	杨浦店	这个杀手不太冷	GF	吕克·贝松	178.00
16	GH19940914600001	杨浦店	这个杀手不太冷	GF	吕克·贝松	178.00
17	GH19940914600001	杨浦店	这个杀手不太冷	GF	吕克·贝松	178.00
1	GH20070518300005	长宁店	唐伯虎点秋香	XJ	李力持	80.00
2	GH20080318300005	长宁店	猫和老鼠	DH	威廉·汉纳及约瑟夫·巴	80.00
3	GH20080618400001	长宁店	倚天屠龙记	GF	王晶	198.00
4	GH20190510500001	闵行店	大侦探皮卡丘	DH	罗伯·莱特曼	68.00
5	GH20190524800001	闵行店	五月天人生无限公司	QT	陈奕仁	70.00
6	GH20190524800002	闵行店	龙珠超：布罗利	DH	长峰达也	45.00
NULL	NULL	闵行店	NULL	NULL	NULL	NULL

(e) 查询购买了所有排片电影的会员信息，输出会员姓名、会员等级、会员折扣

```

SELECT memberName,memberShip,discount
FROM memberinfo c
WHERE c.memberID=(
SELECT A.memberID
FROM ticketinfo A left outer join playinfo B on A.playID=B.playID
GROUP BY A.memberID
HAVING COUNT(DISTINCT(filmID))
= (SELECT COUNT(DISTINCT(filmID))
FROM playinfo))
#1-5

```

(2) 按照上述说明在数据库中建立各个主键、外键、约束、默认。

(3) 在数据库中建立一个名为member_ticket的视图，该视图查询出每个会员的出票情况，包括放映厅名称、门店地址、电影名称、座位编号和会员票价。

```

DROP VIEW IF EXISTS member_ticket;

CREATE VIEW member_ticket
AS
SELECT c.hallName, c.address, f.filmName, t.seatID, t.finalPrice
FROM cinema_hallinfo as c, ticketinfo as t, filminfo as f,
(SELECT playID, filmID, cinema_hallID
FROM playinfo) as P
WHERE c.cinema_hallID = P.cinema_hallID
AND t.playID = P.playID
AND f.filmID = P.filmID;

```

(4) 为TicketInfo表建立一个插入触发器t_insert_ticketinfo，要求在出票信息表中插入某个出票信息（即会员购票）时，会依据memberID的会员折扣，自动计算出票的会员票价finalPrice。

(5) 为TicketInfo表建立一个更新触发器t_update_ticketinfo，当会员观看完电影之后，在出票信息表中更新某个电影评分（grade）时，依据更新后的会员评分grade自动计算出FilmInfo表中的电影平均评分filmGrade。

```

DROP TRIGGER IF EXISTS t_update_ticketinfo;

CREATE TRIGGER t_update_ticketinfo
AFTER INSERT ON ticketinfo FOR EACH ROW
BEGIN

```

```
DECLARE star DECIMAL(3,1);

SELECT filmGrade INTO star
FROM filminfo as f,
    (SELECT p.filmID
     FROM ticketinfo as t INNER JOIN playinfo as p on t.playID = p.playID
     WHERE t.memberID = new.memberID) as A
WHERE f.filmID = A.filmID;

UPDATE filminfo SET filmGrade = (star + new.grade) / 2
WHERE filminfo.filmID = (SELECT p.filmID
                        FROM playinfo as p
                        WHERE p.playID = new.playID);

END;
```

保存所有操作之后，在Navicat中选中数据库filmdb，右键选择【转储SQL文件...】，文件名设置为filmdb学号，单击【保存】，导出成功后，点击【关闭】。