

答案

题目

SQL和关系代数练习题

某关系模式如下：

代表队（队号，队名，领队，积分，名次）

运动员（运动员号，姓名，性别，年龄，代表队号）

比赛项目（项目号，项目类别，项目名称，成绩单位），注：成绩单位表示成绩的计算方式（比如时间，长度等）

比赛详情（运动员号，项目号，成绩，积分，名次）

1. 请使用关系代数表达式实现以下操作

- （1）查询参加了所有比赛项目的运动员号；
- （2）查询参加了“男子100米”但是没有参加“男子110跨栏”的运动员号和姓名。

2. 请使用SQL语句实现以下操作

- （1）实现表“比赛详情”的创建，创建时请注意同时实现主码和外码；
- （2）查找“桂小花”同学参加的比赛项目的项目名称以及成绩和名次；
- （3）查询参加了“男子100米”也没有参加“男子110跨栏”的运动员的运动员号和姓名；
- （4）创建一个视图，里面存储有队号为dpt3的所有运动员的运动员号、姓名、性别、年龄、项目名称、成绩、积分；
- （5）查询代表队号为 dpt3 参赛的各个运动员的编号、姓名和积分情况，并按照积分由大到小排序；
- （6）将运动员号为 GD220005 的运动员的代表队号更新为 dpt3。
- （7）查询参加了运动员王二参加的所有项目的运动员号

参考答案

1. 关系代数表达式

- （1）查询参加了所有比赛项目的运动员号：

$$\pi \text{ 运动员号 (比赛详情)} \div \pi \text{ 项目号 (比赛项目)}$$

- （2）查询参加了“男子100米”但是没有参加“男子110跨栏”的运动员号和姓名：

$$\pi \text{ 运动员号, 姓名 } ((\sigma \text{ 项目名称='男子100米'} (\text{运动员} \bowtie \text{比赛详情} \bowtie \text{比赛项目})) - (\sigma \text{ 项目名称='男子110跨栏'} (\text{运动员} \bowtie \text{比赛详情} \bowtie \text{比赛项目})))$$

2. SQL语句

- （1）实现表“比赛详情”的创建，创建时请注意同时实现主码和外码；

```
CREATE TABLE 比赛详情 (
    运动员号 VARCHAR(20),
    项目号 VARCHAR(20),
    成绩 FLOAT,
    积分 INT,
    名次 INT,
    PRIMARY KEY (运动员号, 项目号),
    FOREIGN KEY (运动员号) REFERENCES 运动员 (运动员号),
    FOREIGN KEY (项目号) REFERENCES 比赛项目 (项目号)
);
```

(2) 查找“桂小花”同学参加的比赛项目的项目名称以及成绩和名次;

```
SELECT 比赛项目.项目名称, 比赛详情.成绩, 比赛详情.名次
FROM 比赛详情, 运动员, 比赛项目
WHERE 比赛详情.运动员号 = 运动员.运动员号
AND 比赛详情.项目号 = 比赛项目.项目号
```

(3) 查询参加了“男子100米”也没有参加“男子110跨栏”的运动员的运动员号和姓名;

```
SELECT 运动员号, 姓名
FROM 运动员
WHERE 运动员号 IN (
    SELECT 运动员号
    FROM 比赛详情
    WHERE 项目号 = '男子100米'
    AND 运动员号 NOT IN (
        SELECT 运动员号
        FROM 比赛详情
        WHERE 项目号 = '男子110跨栏'
    )
);
```

或

```
SELECT 运动员.运动员号, 运动员.姓名
FROM 运动员, 比赛详情, 比赛项目
WHERE 运动员.运动员号 = 比赛详情.运动员号
AND 比赛详情.项目号 = 比赛项目.项目号
AND 项目名称 = '男子100米'
EXCEPT
SELECT 运动员.运动员号, 运动员.姓名
FROM 运动员, 比赛详情, 比赛项目
WHERE 运动员.运动员号 = 比赛详情.运动员号
AND 比赛详情.项目号 = 比赛项目.项目号
AND 项目名称 = '男子110跨栏';
```

(4) 创建一个视图，里面存储有队号为dpt3的所有运动员的运动员号、姓名、性别、年龄、项目名称、成绩、积分;

```
CREATE VIEW Dpt3_运动员 AS
```

```
    SELECT 运动员.运动员号, 运动员.姓名, 运动员.性别, 运动员.年龄, 比赛项目.项目名称, 比赛详情.成绩, 比赛详情.积分
    FROM 运动员, 比赛详情, 比赛项目
    WHERE 运动员.运动员号 = 比赛详情.运动员号
    AND 比赛详情.项目号 = 比赛项目.项目号
    AND 运动员.代表队号 = 'dpt3';
```

(5) 查询代表队号为 dpt3 参赛的各个运动员的编号、姓名和积分情况，并按照积分由大到小排序；

```
SELECT 运动员.运动员号, 运动员.姓名, SUM(比赛详情.积分) AS 总积分
FROM 运动员
INNER JOIN 比赛详情 ON 运动员.运动员号 = 比赛详情.运动员号
WHERE 运动员.代表队号 = 'dpt3'
GROUP BY 运动员.运动员号
ORDER BY 总积分 DESC;
```

或使用 (4) 的视图继续查询：

```
SELECT 运动员号, 姓名, SUM(积分) as 总积分
FROM Dpt3_运动员
GROUP BY 运动员号
ORDER BY 总积分 DESC;
```

(6) 将运动员号为 GD220005 的运动员的代表队号更新为 dpt3。

```
UPDATE 运动员
SET 代表队号 = 'dpt3'
WHERE 运动员号 = 'GD220005';
```

(7) 查询参加了运动员王二参加的所有项目的运动员号

```
SELECT DISTINCT 运动员号
FROM 比赛详情
WHERE 运动员号 IN (
    SELECT 运动员号
    FROM 运动员
    WHERE 姓名 = '王二'
)
GROUP BY 运动员号
HAVING COUNT(DISTINCT 项目号) = (
    SELECT COUNT(DISTINCT 项目号)
    FROM 比赛详情
    WHERE 运动员号 IN (
        SELECT 运动员号
        FROM 运动员
        WHERE 姓名 = '王二'
    )
);
```

首先使用子查询查找到姓名为'王二'的运动员号，然后在比赛详情表中筛选出这些运动员号所参加的所有比赛项目，最后根据运动员号进行分组，并通过HAVING子句筛选出参加的项目数等于'王二'参加的全部项目数的运动员号。