MySQL任务3作业参考答案

项目十 各部门工资最高的员工

这题思路比较简单，就是先查询出每个部门最高的工资，然后再通过JOIN匹配部门信息

```sql
-- 创建表及插入数据
CREATE TABLE employee(
Id           INT         NOT NULL PRIMARY KEY,
Name         VARCHAR(20) NOT NULL,
Salary       INT,
DepartmentId INT
);

INSERT INTO employee (Id,Name,Salary,DepartmentId)
VALUES
(1,'Joe',  70000,1),
(2,'Henry',80000,2),
(3,'Sam',  60000,2),
(4,'Max',  90000,2);

CREATE TABLE Department(
Id   INT         NOT NULL,
Name VARCHAR(20)
);

INSERT INTO Department (Id,Name)
VALUES
(1,'IT'),
(2,'Sales');

-- 参考答案
SELECT d.Name as Department, e.Name as Employee, e.Salary
  FROM Department d
  JOIN employee e
    ON d.Id = e.DepartmentId
   AND e.Salary = (SELECT MAX(Salary)
                     FROM employee
                    WHERE DepartmentId = d.Id)
```

项目十一 换座位

从最后结果来看就是 ：
①id为偶数的需要往前挪
②id为奇数的需要往后挪
③再考虑最后一位是奇数还是偶数，奇数不变（发现偶数的情况已经包含在前面了）

通过 id%2=1 来判断是奇数。
这里需要注意的地方是，最后一位，我们在判断奇数往后挪的时候，是不包含最后一位的。所以在②的时候要限定 id 小于总个数。

```sql
SELECT *
  FROM(
        -- 遇到偶数就往上移一个位置
        SELECT id-1 AS id
              ,student
          FROM seat WHERE id%2=0

         UNION
        -- 遇到奇数就往下移一个位置，这里不包含最后一个位置
        SELECT id+1 AS id
              ,student
          FROM seat
         WHERE id%2=1
           AND (id+1) <= (SELECT COUNT(*) FROM seat)

         UNION
        -- 处理最后一个位置，这里只考虑奇数情况，保持不变（偶数已经在第一步里处理了）
        SELECT id AS id
              ,student
          FROM seat
         WHERE id%2=1
           AND (id+1) > (SELECT COUNT(*) FROM seat)
          ) AS rank
ORDER BY id ASC;
```

项目十二 分数排名

这里的排名是连续排名。将分数去重后的清单排名就是最终排名。然后将这个排名JOIN原始数据。
问题在于，怎么在去重的分数清单增加一列ID，这个ID会随着数据自动增加。这个可以通过参数化查询搞定，但是我们还没学，所以pass。
然后就有个取巧的办法。我们还是需要获取去重的分数清单，但是不需要排名ID了：
① 原始表里的最大值。 分数清单只取最大值，然后COUNT(*)，这样就只有1
② 原始表里的第二大值。 分数清单取最大值和第二大值，然后COUNT(*), 这样就是2
```
以此内推。
在code里就是 分数清单里的score要大于等于原始表的某个具体的值。

这个写法的缺点是，每一个原始表socre列的值，都需要重新扫描去重后的分数清单，在数据表大的情况下，性能会很差。

```
-- 创建表及插入数据
CREATE TABLE score(
id   INT,
score DECIMAL(3,2)
);

INSERT INTO score
VALUES
(1,3.50),
(2,3.65),
(3,4.00),
(4,3.85),
(5,4.00),
(6,3.65);

-- 参考答案
SELECT score
      ,(SELECT COUNT(DISTINCT score)
          FROM score
         WHERE score >= s.score
) AS Rank
  FROM score s
 ORDER BY score DESC;

```

## 项目十三：连续出现的数字

全连接三张**Logs**表，分别用来搜索连续三个数中的第一、二、三个数，检查是否相同。并且最后还要排除重复。

```
SELECT DISTINCT l1.Num AS
ConsecutiveNums FROM
Logs AS l1, Logs AS l2, Logs AS l3
WHERE l1.Num = l2.Num AND l2.Num =
l3.Num AND
l1.Id = l2.Id - 1 AND l2.Id = l3.Id - 1;
```
```

项目十四：树节点

对于一个节点，先判断是否为根节点。如果不是，再判断是内部节点还是叶子节点。
如果一个节点的父节点id为空（null），则为根节点：

```
SELECT id, IF(ISNULL(p_id), 'Root', IF()) AS Type
FROM tree;
```

对于一个不是根节点的节点，如果它是某些节点的父节点，则为内部节点，否则为叶子节点：

```
SELECT id, IF(ISNULL(p_id), 'Root', IF(id IN (SELECT p_id FROM tree), 'Inner',
'Leaf')) AS Type
FROM tree;
```

最后，根据id排序：

```
SELECT id, IF(ISNULL(p_id), 'Root', IF(id IN (SELECT p_id FROM tree), 'Inner',
'Leaf')) AS Type
FROM tree ORDER BY id;
```

项目十五：至少有五名直接下属的经理

可以根据ManagerId分组，找出有5个下属的主管的Id：

```
SELECT ManagerId FROM Employee GROUP BY ManagerId HAVING
COUNT(*) > 4;
```

然后输出对应的姓名：

```
SELECT Name FROM Employee WHERE Id IN
(SELECT ManagerId FROM Employee GROUP BY ManagerId HAVING COUNT(*)
> 4);
```

或者，把两张**Employee**表连接（join）起来，一张作为员工，一张作为主管。然后根据主管分组，并筛选出符合条件的组：

```
SELECT m.Name FROM Employee AS e
JOIN Employee AS m ON e.ManagerId =
m.Id
GROUP BY m.Name HAVING
COUNT(e.Name) >= 5;
```