

# Oracle

姓名：王顺顺

学号:201810414218

班级：18级软件工程二班

## 实验目的

分析SQL执行计划，执行SQL语句的优化指导。理解分析SQL语句的执行计划的重要作用

## 实验内容

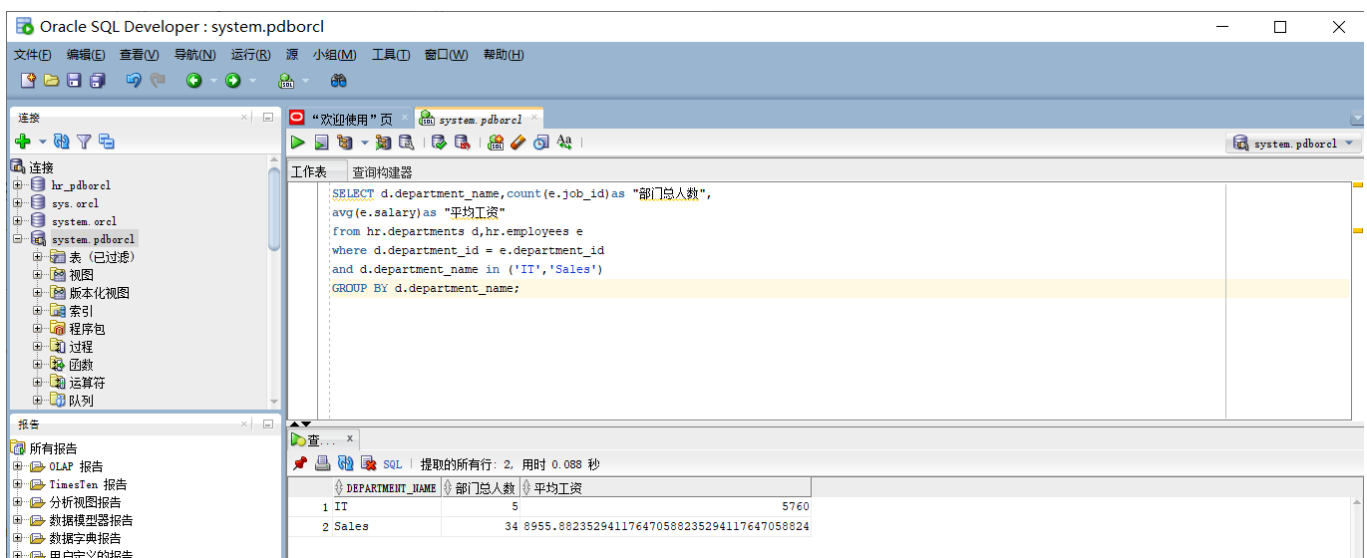
- (1) 对Oracle12c中的HR人力资源管理系统中的表进行查询与分析
- (2) 首先运行和分析教材中的样例：本训练任务目的是查询两个部门('IT'和'Sales')的部门总人数和平均工资，以下两个查询的结果是一样的。但效率不相同
- (3) 设计自己的查询语句，并作相应的分析，查询语句不能太简单

## 查询教材中的内容

### 查询1

```
SELECT d.department_name,count(e.job_id)as "部门总人数", avg(e.salary)as "平均工资" from hr.departments d,hr.employees e where d.department_id = e.department_id and d.department_name in ('IT','Sales') GROUP BY d.department_name;
```

### 代码一运行结果如图



The screenshot shows the Oracle SQL Developer interface. The main window displays the following SQL query:

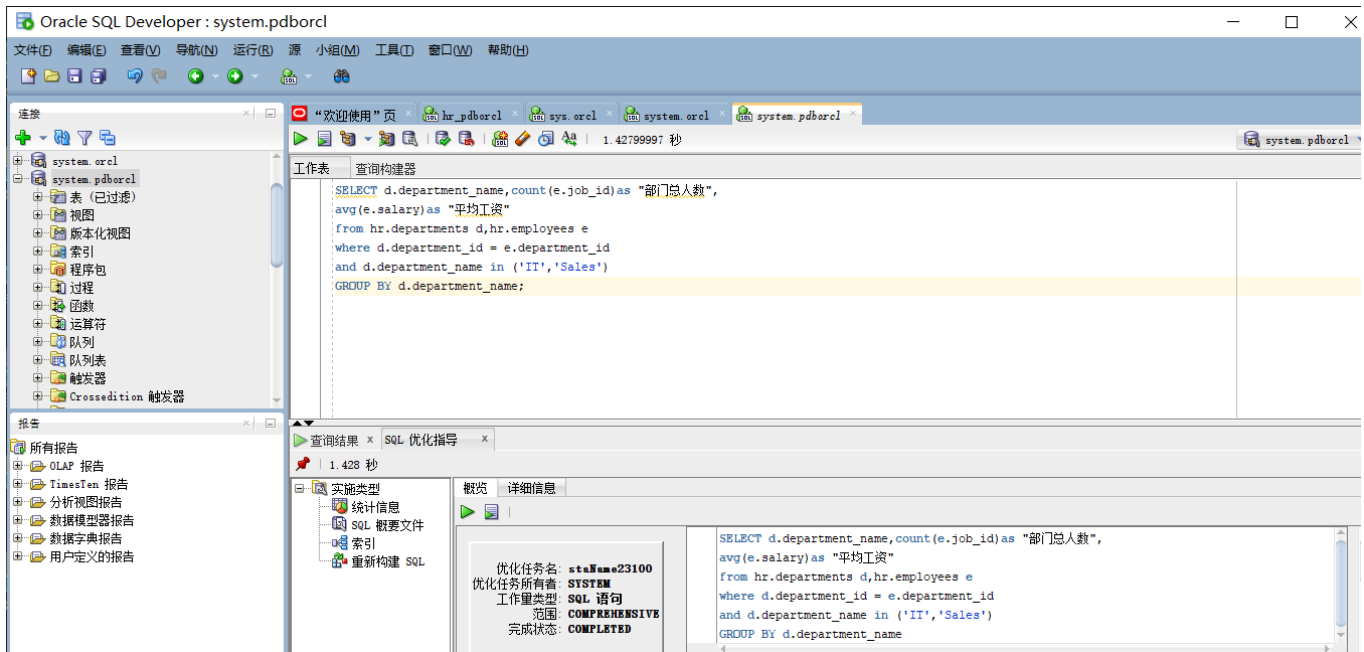
```
SELECT d.department_name,count(e.job_id)as "部门总人数",  
avg(e.salary)as "平均工资"  
from hr.departments d,hr.employees e  
where d.department_id = e.department_id  
and d.department_name in ('IT','Sales')  
GROUP BY d.department_name;
```

The results are shown in a table at the bottom of the window:

DEPARTMENT_NAME	部门总人数	平均工资
1 IT	5	5760
2 Sales	34 8955.862352941176470588235294117647058824	

**分析：**该查询语句通过从两个部门来查询部门的总人数和平均工资，通过创建一个或多个索引可以改进此语句的执行计划。建议考虑运行可以改进物理方案设计的访问指导或者创建推荐的索引。原理是创建推荐的索引可

以显著地改进此语句的执行计划。但是，使用典型的SQL工作量运行“访问指导”可能比单个语句更可取。通过这种方法可以获得全面的索引建议案，包括计算索引维护的开销和附加的空间消耗



## 查询2

```
SELECT d.department_name,count(e.job_id)as "部门总人数", avg(e.salary)as "平均工资" FROM hr.departments
d,hr.employees e WHERE d.department_id = e.department_id GROUP BY d.department_name HAVING
d.department_name in ('IT','Sales');
```

代码二运行结果如图

The screenshot displays the Oracle SQL Developer application window. The title bar reads "Oracle SQL Developer : system.pdborcl". The menu bar includes "文件(F)", "编辑(E)", "查看(V)", "导航(N)", "运行(R)", "源", "小组(M)", "工具(T)", "窗口(W)", and "帮助(H)". The toolbar contains icons for file operations, editing, and execution.

The left sidebar shows the "连接" (Connections) tree with the following structure:

- 连接
  - hr\_pdborcl
  - sys.oracl
  - system.oracl
  - system.pdborcl (selected)
    - 表 (已过滤)
    - 视图
    - 版本化视图
    - 索引
    - 程序包
    - 过程
    - 函数
    - 运算符
    - 队列

The main workspace shows the "查询构建器" (Query Builder) tab. The SQL query is as follows:

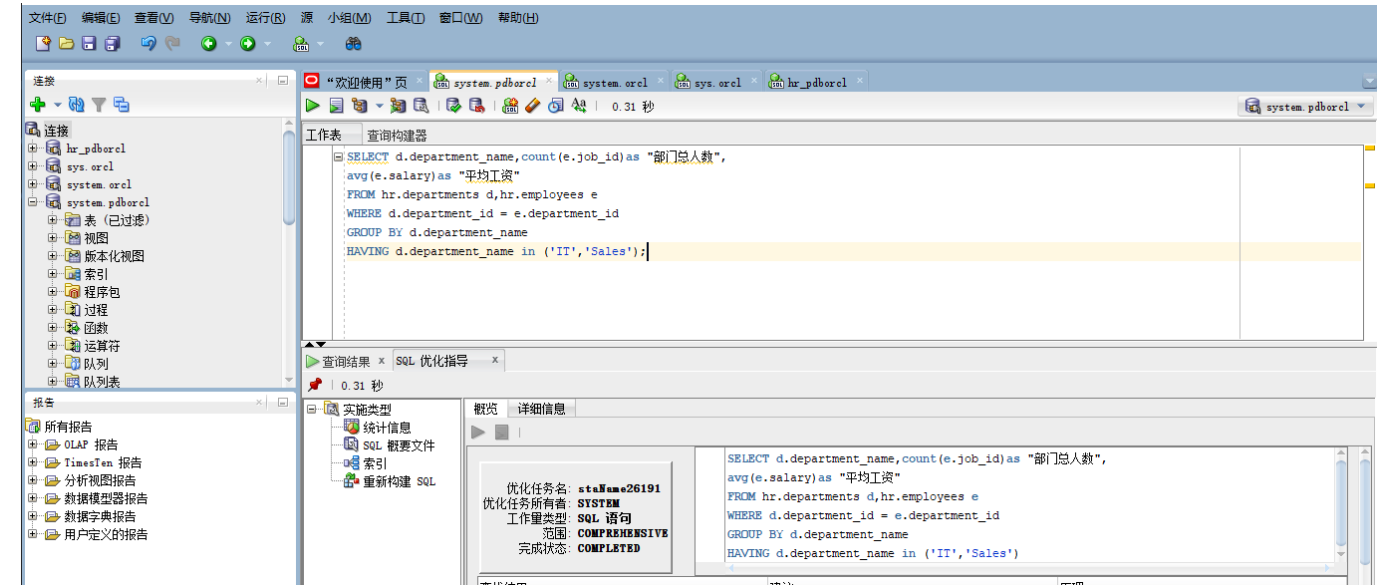
```
SELECT d.department_name, count(e.job_id) as "部门总人数",
       avg(e.salary) as "平均工资"
FROM hr.departments d, hr.employees e
WHERE d.department_id = e.department_id
GROUP BY d.department_name
HAVING d.department_name in ('IT', 'Sales');
```

Below the query editor, the "SQL 优化指导" (SQL Optimizer) and "查询结果" (Query Results) tabs are visible. The "查询结果" tab shows the execution results:

SQL | 提取的所有行: 2, 用时 0.004 秒

DEPARTMENT_NAME	部门总人数	平均工资
1 IT	5	5760
2 Sales	34 8955.882352941176470588235294117647058824	

**分析：**该查询语句通过从HR的部门表和员工表按照部门分组，判断部门ID和员工ID是否对应，由having确认部门名字是IT和sales来查询部门总人数和平均工资。该查询语句比第一条查询语句要好一点，没有给出优化建议



优化后的代码

SELECT d.department\_name,count(e.job\_id)as "部门总人数",avg(e.salary)as "平均工资" FROM hr.departments d,hr.employees e WHERE e.department\_id=d.department\_id and (d.department\_id=60 or d.department\_id=80) group by d.department\_name

分析：该查询语句通过从部门表和员工表判断部门ID和员工ID相等，以部门名分组来达到查询部门总人数和平均工资的目的

运行结果如图

