DATAGURU 炼数成金

数据库引航 第8课—复杂一些的SQL语句

2012.10.15

法律声明

【声明】本视频和幻灯片为炼数成金网络课程的教学资料，所有资料只能在课程内使用，不得在课程以外范围散播，违者将可能被追究法律和经济责任。

课程详情访问炼数成金培训网站

http://edu.dataguru.cn

SQL语言本质---集合的运算

- 不要认为SQL语句操作的一定是一个表里的数据，实际上SQL操作的是一个集合，整个SQL的处理过程，实际上就是集合的运算过程。

SQL=集合的运算

子查询---结果集的操作

- 单表
  - select count(*) from (select * from emp);
  - select ename,sum(sal) sum_sal from (select * from emp) group by enmae;
  - select ename,sum_sal from select ename,sum(sal) sum_sal from (select * from emp);
- 多表

```
select b.dname,a.ename
  from
     (select deptno,ename from emp) a,
     (select deptno,dname from dept) b
  where a.deptno=b.deptno  order by 1,2;

select b.dname,a.sal
  from
     (select deptno,sum(sal) sal from emp group by deptno) a,
     (select deptno,dname from dept) b
  where a.deptno=b.deptno order by 1,2 ;
```

表连接

- 内连接 innner join
  - 两个结果集的交集

Employee

Name	EmpId	DeptName
Harry	3415	Finance
Sally	2241	Sales
George	3401	Finance
Harriet	2202	Sales

Dept

DeptName	Manager
Finance	George
Sales	Harriet
Production	Charles

Employee ⋈ *Dept*

Name	EmpId	DeptName	Manager
Harry	3415	Finance	George
Sally	2241	Sales	Harriet
George	3401	Finance	George
Harriet	2202	Sales	Harriet

```
SQL> select a.name,a.empid,a.deptname,b.manager from employee a,dept b where a.deptname=b.deptname;

NAME            EMPID DEPTNAME   MANAGER
---------- ---------- ---------- ----------
Harry            3415 Finance    George
Sally            2241 Sales      Harriet
George           3401 Finance    George
Harriet          2202 Sales      Harriet
```

表连接

- 外连接 outer join
  - 左连接 左边集合的全集。

Employee

Name	EmpId	DeptName
Harry	3415	Finance
Sally	2241	Sales
George	3401	Finance
Harriet	2202	Sales
Tim	1123	Executive

Dept

DeptName	Manager
Sales	Harriet
Production	Charles

Employee *Dept*

Name	EmpId	DeptName	Manager
Harry	3415	Finance	ω
Sally	2241	Sales	Harriet
George	3401	Finance	ω
Harriet	2202	Sales	Harriet
Tim	1123	Executive	ω

```
SQL> select a.name,a.empid,a.deptname,b.manager from employee a,dept b where a.deptname=b.deptname(+);

NAME            EMPID DEPTNAME   MANAGER
---------- ---------- ---------- ----------
Harriet           2202 Sales      Harriet
Sally             2241 Sales      Harriet
Tim               1123 Executive
George            3401 Finance
Harry             3415 Finance
```

表连接

- 外连接 outer join
  - 右连接 右边集合的全集

Employee

Name	EmpId	DeptName
Harry	3415	Finance
Sally	2241	Sales
George	3401	Finance
Harriet	2202	Sales
Tim	1123	Executive

Dept

DeptName	Manager
Sales	Harriet
Production	Charles

Employee *Dept*

Name	EmpId	DeptName	Manager
Sally	2241	Sales	Harriet
Harriet	2202	Sales	Harriet
ω	ω	Production	Charles

```
SQL> select a.name,a.empid,a.deptname,b.manager from employee a,dept b where a.deptname(+)=b.deptname;

NAME            EMPID DEPTNAME   MANAGER
---------- ---------- ---------- ----------
Sally            2241 Sales      Harriet
Harriet          2202 Sales      Harriet
                                 Charles
```

标量子查询---scalar subquery

- select (select dname from dept b where b.deptno=a.deptno),ename from emp a order by 1,2;

with-- 重用结果集

- 找出工资大于平均工资的员工

```
select ename,sum(sal) from emp  group by ename having sum(sal)>= (select sum(sal)/14
    from emp) ;
```

- 使用with

```
with t as (select ename,sum(sal) sal from emp  group by ename )
select ename,sal from t where sal>=(select sum(sal)/14 from t)
```

列字段的处理SQL

case

```
select
    case
    when deptno=10 then 'ACCOUNTING'
    when deptno=20 then 'RESERCH'
    when deptno=30 then 'SALES'
    end,
    sum(sal) from emp
    group by deptno
```

列字段的处理SQL

◆ decode

```
select
    decode(deptno,
            10, 'ACCOUNTING',
            20, 'RESERCH',
            30, 'SALES'
            ),
    sum(sal) from emp
    group by deptno
```

行转列

```
SQL> select job,ename,sal from emp where job='MANAGER';

JOB       ENAME      SAL
--------- ---------- ----------
MANAGER   JONES            2975
MANAGER   BLAKE            2850
MANAGER   CLARK            2450
```

```
SQL>  select job,
  2             sum(decode(ename,'BLAKE',SAL)) BLAKE,
  3             sum(decode(ename,'JONES',SAL)) JONES,
  4             sum(decode(ename,'CLARK',SAL)) CLARK
  5   from emp
  6     where job='MANAGER' group by job;

JOB            BLAKE      JONES      CLARK
--------- ---------- ---------- ----------
MANAGER         2850       2975       2450

SQL>
```

行转列

```
  1   select job,
  2             decode(ename,'BLAKE',SAL) BLAKE,
  3             decode(ename,'JONES',SAL) JONES,
  4             decode(ename,'CLARK',SAL) CLARK
  5   from emp
  6*    where job='MANAGER'
SQL> /

JOB            BLAKE      JONES      CLARK
--------- ---------- ---------- ----------
MANAGER                   2975
MANAGER         2850
MANAGER                              2450

SQL>
```

```
  1   select job,
  2             avg(decode(ename,'BLAKE',SAL)) BLAKE,
  3             max(decode(ename,'JONES',SAL)) JONES,
  4             min(decode(ename,'CLARK',SAL)) CLARK
  5   from emp
  6     where job='MANAGER'
  7*    group by job
SQL>
SQL> /

JOB            BLAKE      JONES      CLARK
--------- ---------- ---------- ----------
MANAGER         2850       2975       2450

SQL>
```

DBA常用的SQL语句

- 数据库的大小

```
  1* select ((select sum(bytes) from dba_data_files)+(select sum(bytes) from dba_temp_files)) total_size from dual
SQL> /

TOTAL_SIZE
----------
2193096704
```

- 查询某个段对象（表，索引）的大小

```
SQL> select sum(bytes) from user_segments where segment_Name='T';

SUM(BYTES)
----------
     65536
```

- 查询某个表空间的大小

```
SQL> select sum(bytes) from dba_data_files where tablespace_Name='USERS';

SUM(BYTES)
----------
  70778880
```

DBA常用的SQL语句

- 查看日志文件的状态

```
SQL> select group#,status from v$log;

    GROUP# STATUS
---------- ----------------
         1 CURRENT
         2 INACTIVE
         3 INACTIVE
```

- 查询事务阻塞

```
SQL> select sid,type,lmode,request from v$lock where TYPE in ('TM','TX') ORDER BY 1,2;

       SID TY      LMODE    REQUEST
---------- -- ---------- ----------
       167 TM          3          0
       167 TX          6          0
       198 TM          3          0
       198 TX          0          6
```

SQL和执行计划

- 执行计划是SQL获取和处理数据的途径和方法。

执行计划和性能

```
SQL> select * from t where object_id<100;

执行计划
----------------------------------------------------------
Plan hash value: 1601196873

--------------------------------------------------------------------------------
| Id  | Operation          | Name | Rows  | Bytes | Cost (%CPU)| Time     |
--------------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |      |    93 |  2697 |   100   (1)| 00:00:02 |
|*  1 |  TABLE ACCESS FULL| T    |    93 |  2697 |   100   (1)| 00:00:02 |
--------------------------------------------------------------------------------
```

```
SQL> select * from t where object_id<100;

执行计划
----------------------------------------------------------
Plan hash value: 1594971208

--------------------------------------------------------------------------------
| Id  | Operation                    | Name   | Rows  | Bytes | Cost (%CPU)| Time     |
--------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |        |    93 |  2697 |     3   (0)| 00:00:01 |
|   1 |  TABLE ACCESS BY INDEX ROWID| T      |    93 |  2697 |     3   (0)| 00:00:01 |
|*  2 |   INDEX RANGE SCAN           | IDX_T |    93 |       |     2   (0)| 00:00:01 |
--------------------------------------------------------------------------------
```

SQL --数据库性能的始作俑者

- 所有的数据库性能，几乎全部来自SQL。
- 优秀的SQL是数据库最大的福祉。
- 一条很烂的SQL，可以搞瘫一台性能极好的服务器。

为什么高效的SQL这么难？

- 语言的效率，是SQL语言的最难的地方
  - tablesan
  - index range scan
  - index fast scan
  - nested loop join
  - merge join
  - hash join

  ……
- 优化器机制开发者无法掌控

炼数成金逆向收费式网络课程

- Dataguru（炼数成金）是专业数据分析网站，提供教育，媒体，内容，社区，出版，数据分析业务等服务。我们的课程采用新兴的互联网教育形式，独创地发展了逆向收费式网络培训课程模式。既继承传统教育重学习氛围，重竞争压力的特点，同时又发挥互联网的威力打破时空限制，把天南地北志同道合的朋友组织在一起交流学习，使到原先孤立的学习个体组合成有组织的探索力量。并且把原先动辄成千上万的学习成本，直线下降至百元范围，造福大众。我们的目标是：低成本传播高价值知识，构架中国第一的网上知识流转阵地。
- 关于逆向收费式网络的详情，请看我们的培训网站 http://edu.dataguru.cn

Thanks

FAQ时间