

Oracle 第二天

一、多表查询

- 什么是笛卡尔积？

EMPLOYEES (20行)

EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID
100	King	90
101	Kochhar	90
...		
202	Fay	20
205	Higgins	110
206	Gietz	110

20 rows selected.

DEPARTMENTS (8行)

DEPARTMENT_ID	DEPARTMENT_NAME	LOCATION_ID
10	Administration	1700
20	Marketing	1800
50	Shipping	1500
60	IT	1400
80	Sales	2500
90	Executive	1700
110	Accounting	1700
190	Contracting	1700

8 rows selected.

笛卡尔集:
20x8=160行

EMPLOYEE_ID	DEPARTMENT_ID	LOCATION_ID
100	90	1700
101	90	1700
102	90	1700
103	60	1700
104	60	1700
107	60	1700

- Oracle 的连接条件的类型**

- 等值连接
- 不等值连接
- 外连接
- 自连接

- Oracle 多表连接示例**

1.多表连接基本查询

使用一张以上的表做查询就是多表查询

语法： `SELECT {DISTINCT} *|列名.. FROM 表名 别名, 表名1 别名
{WHERE 限制条件 ORDER BY 排序字段 ASC|DESC...}`

范例：查询员工表和部门表



```
select * from emp, dept
```

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	DEPTNO	DNAME	LOC
1	7369	SMITH	CLERK	7902	1980/12/17	800.00		20	10	ACCOUNTING	NEW YORK
2	7499	ALLEN	SALESMAN	7698	1981/2/20	1600.00	300.00	30	10	ACCOUNTING	NEW YORK
3	7521	WARD	SALESMAN	7698	1981/2/22	1250.00	500.00	30	10	ACCOUNTING	NEW YORK
4	7566	JONES	MANAGER	7839	1981/4/2	2975.00		20	10	ACCOUNTING	NEW YORK
5	7654	MARTIN	SALESMAN	7698	1981/9/28	1250.00	1400.00	30	10	ACCOUNTING	NEW YORK
6	7698	BLAKE	MANAGER	7839	1981/5/1	2850.00		30	10	ACCOUNTING	NEW YORK
7	7782	CLARK	MANAGER	7839	1981/6/9	2450.00		10	10	ACCOUNTING	NEW YORK
8	7788	SCOTT	ANALYST	7566	1987/4/19	3000.00		20	10	ACCOUNTING	NEW YORK
9	7839	KING	PRESIDENT		1981/11/17	5000.00		10	10	ACCOUNTING	NEW YORK
10	7844	TURNER	SALESMAN	7698	1981/9/8	1500.00	0.00	30	10	ACCOUNTING	NEW YORK
11	7876	ADAMS	CLERK	7788	1987/5/23	1100.00		20	10	ACCOUNTING	NEW YORK
12	7900	JAMES	CLERK	7698	1981/12/3	950.00		30	10	ACCOUNTING	NEW YORK
13	7902	FORD	ANALYST	7566	1981/12/3	3000.00		20	10	ACCOUNTING	NEW YORK
14	7924	MILLER	CLERK	7782	1982/1/23	1300.00		10	10	ACCOUNTING	NEW YORK
15	7369	SMITH	CLERK	7902	1980/12/17	800.00		20	20	RESEARCH	DALLAS
16	7499	ALLEN	SALESMAN	7698	1981/2/20	1600.00	300.00	30	20	RESEARCH	DALLAS
17	7521	WARD	SALESMAN	7698	1981/2/22	1250.00	500.00	30	20	RESEARCH	DALLAS
18	7566	JONES	MANAGER	7839	1981/4/2	2975.00		20	20	RESEARCH	DALLAS
19	7654	MARTIN	SALESMAN	7698	1981/9/28	1250.00	1400.00	30	20	RESEARCH	DALLAS
20	7698	BLAKE	MANAGER	7839	1981/5/1	2850.00		30	20	RESEARCH	DALLAS
21	7782	CLARK	MANAGER	7839	1981/6/9	2450.00		10	20	RESEARCH	DALLAS
22	7788	SCOTT	ANALYST	7566	1987/4/19	3000.00		20	20	RESEARCH	DALLAS

1:24 scott@ORCL 56 rows selected in 0.281 seconds

我们发现产生的记录数是 56 条，我们还会发现 emp 表是 14 条，dept 表是 4 条，56 正是 emp 表和 dept 表的记录数的乘积，我们称其为笛卡尔积。

如果多张表进行一起查询而且每张表的数据很大的话笛卡尔积就会变得非常大，对性能造成影响，想要去掉笛卡尔积我们需要关联查询。

在两张表中我们发现有一个共同的字段是 deptno，deptno 就是两张表的关联的字段，我们可以使用这个字段来做限制条件，两张表的关联查询字段一般是其中一张表的主键，另一张表的外键。

```
select * from emp, dept where emp.deptno = dept.deptno
```

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	DEPTNO	DNAME	LOC
1	7369	SMITH	CLERK	7902	1980/12/17	800.00		20	20	RESEARCH	DALLAS
2	7499	ALLEN	SALESMAN	7698	1981/2/20	1600.00	300.00	30	30	SALES	CHICAGO
3	7521	WARD	SALESMAN	7698	1981/2/22	1250.00	500.00	30	30	SALES	CHICAGO
4	7566	JONES	MANAGER	7839	1981/4/2	2975.00		20	20	RESEARCH	DALLAS
5	7654	MARTIN	SALESMAN	7698	1981/9/28	1250.00	1400.00	30	30	SALES	CHICAGO
6	7698	BLAKE	MANAGER	7839	1981/5/1	2850.00		30	30	SALES	CHICAGO
7	7782	CLARK	MANAGER	7839	1981/6/9	2450.00		10	10	ACCOUNTING	NEW YORK
8	7788	SCOTT	ANALYST	7566	1987/4/19	3000.00		20	20	RESEARCH	DALLAS
9	7839	KING	PRESIDENT		1981/11/17	5000.00		10	10	ACCOUNTING	NEW YORK
10	7844	TURNER	SALESMAN	7698	1981/9/8	1500.00	0.00	30	30	SALES	CHICAGO
11	7876	ADAMS	CLERK	7788	1987/5/23	1100.00		20	20	RESEARCH	DALLAS
12	7900	JAMES	CLERK	7698	1981/12/3	950.00		30	30	SALES	CHICAGO
13	7902	FORD	ANALYST	7566	1981/12/3	3000.00		20	20	RESEARCH	DALLAS
14	7924	MILLER	CLERK	7782	1982/1/23	1300.00		10	10	ACCOUNTING	NEW YORK

关联之后我们发现数据条数是 14 条，不在是 56 条。

多表查询我们可以为每一张表起一个别名



```
select * from emp e, dept d where e.deptno = d.deptno
```

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	DEPTNO	DNAME	LOC
1	7369	SMITH	CLERK	7902	1980/12/17	800.00		20	20	RESEARCH	DALLAS
2	7499	ALLEN	SALESMAN	7698	1981/2/20	1600.00	300.00	30	30	SALES	CHICAGO
3	7521	WARD	SALESMAN	7698	1981/2/22	1250.00	500.00	30	30	SALES	CHICAGO
4	7566	JONES	MANAGER	7839	1981/4/2	2975.00		20	20	RESEARCH	DALLAS
5	7654	MARTIN	SALESMAN	7698	1981/9/28	1250.00	1400.00	30	30	SALES	CHICAGO
6	7698	BLAKE	MANAGER	7839	1981/5/1	2850.00		30	30	SALES	CHICAGO
7	7782	CLARK	MANAGER	7839	1981/6/9	2450.00		10	10	ACCOUNTING	NEW YORK
8	7788	SCOTT	ANALYST	7566	1987/4/19	3000.00		20	20	RESEARCH	DALLAS
9	7839	KING	PRESIDENT		1981/11/17	5000.00		10	10	ACCOUNTING	NEW YORK
10	7844	TURNER	SALESMAN	7698	1981/9/8	1500.00	0.00	30	30	SALES	CHICAGO
11	7876	ADAMS	CLERK	7788	1987/5/23	1100.00		20	20	RESEARCH	DALLAS
12	7900	JAMES	CLERK	7698	1981/12/3	950.00		30	30	SALES	CHICAGO
13	7902	FORD	ANALYST	7566	1981/12/3	3000.00		20	20	RESEARCH	DALLAS
14	7934	MILLER	CLERK	7782	1982/1/23	1300.00		10	10	ACCOUNTING	NEW YORK

范例：查询出雇员的编号，姓名，部门的编号和名称，地址

```
select e.empno, e.ename, d.deptno, d.dname, d.loc
from emp e, dept d
where e.deptno = d.deptno
```

	EMPNO	ENAME	DEPTNO	DNAME	LOC
1	7369	SMITH	20	RESEARCH	DALLAS
2	7499	ALLEN	30	SALES	CHICAGO
3	7521	WARD	30	SALES	CHICAGO
4	7566	JONES	20	RESEARCH	DALLAS
5	7654	MARTIN	30	SALES	CHICAGO
6	7698	BLAKE	30	SALES	CHICAGO
7	7782	CLARK	10	ACCOUNTING	NEW YORK
8	7788	SCOTT	20	RESEARCH	DALLAS
9	7839	KING	10	ACCOUNTING	NEW YORK
10	7844	TURNER	30	SALES	CHICAGO
11	7876	ADAMS	20	RESEARCH	DALLAS
12	7900	JAMES	30	SALES	CHICAGO
13	7902	FORD	20	RESEARCH	DALLAS
14	7934	MILLER	10	ACCOUNTING	NEW YORK

范例：查询出每个员工的上级领导

分析：emp 表中的 mgr 字段是当前雇员的上级领导的编号，所以该字段对 emp 表产生了自身关联，可以使用 mgr 字段和 empno 来关联



```
select e.empno,e.ename,e2.empno,e2.ename
from emp e ,emp e2
where e.mgr = e2.empno;
```

	EMPNO	ENAME	EMPNO	ENAME
1	7369	SMITH	7902	FORD
2	7499	ALLEN	7698	BLAKE
3	7521	WARD	7698	BLAKE
4	7566	JONES	7839	KING
5	7654	MARTIN	7698	BLAKE
6	7698	BLAKE	7839	KING
7	7782	CLARK	7839	KING
8	7788	SCOTT	7566	JONES
9	7844	TURNER	7698	BLAKE
10	7876	ADAMS	7788	SCOTT
11	7900	JAMES	7698	BLAKE
12	7902	FORD	7566	JONES
13	7934	MILLER	7782	CLARK

范例:在上一个例子的基础上查询该员工的部门名称

分析:只要在上一个例子基础上再加一张表的关联,使用 deptno 来做关联字段即可

```
select e.empno, e.ename, e1.empno, e1.ename, d.dname
from emp e, emp e1, dept d
where e.mgr = e1.empno
and e.deptno = d.deptno
```

	EMPNO	ENAME	EMPNO	ENAME	DNAME
1	7369	SMITH	7902	FORD	RESEARCH
2	7499	ALLEN	7698	BLAKE	SALES
3	7521	WARD	7698	BLAKE	SALES
4	7566	JONES	7839	KING	RESEARCH
5	7654	MARTIN	7698	BLAKE	SALES
6	7698	BLAKE	7839	KING	SALES
7	7782	CLARK	7839	KING	ACCOUNTING
8	7788	SCOTT	7566	JONES	RESEARCH
9	7844	TURNER	7698	BLAKE	SALES
10	7876	ADAMS	7788	SCOTT	RESEARCH
11	7900	JAMES	7698	BLAKE	SALES
12	7902	FORD	7566	JONES	RESEARCH
13	7934	MILLER	7782	CLARK	ACCOUNTING

范例:查询出每个员工编号,姓名,部门名称,工资等级和他的上级领导的姓名,工资等级

```
select e.empno,
       e.ename,
       decode(s.grade,
              1,'一级',
              2,'二级',
```



```

        3, '三级',
        4, '四级',
        5, '五级') grade,
d. dname,
e1. empno,
e1. ename,
decode(s1. grade,
        1, '一级',
        2, '二级',
        3, '三级',
        4, '四级',
        5, '五级') grade
from emp e, emp e1, dept d, salgrade s, salgrade s1
where e.mgr = e1.empno
and e.deptno = d.deptno
and e.sal between s.losal and s.hisal
and e1.sal between s1.losal and s1.hisal

```

	EMPNO	ENAME	GRADE	DNAME	EMPNO	ENAME	GRADE
▶ 1	7369	SMITH	一级	RESEARCH	7902	FORD	四级
2	7900	JAMES	一级	SALES	7698	BLAKE	四级
3	7876	ADAMS	一级	RESEARCH	7788	SCOTT	四级
4	7521	WARD	二级	SALES	7698	BLAKE	四级
5	7654	MARTIN	二级	SALES	7698	BLAKE	四级
6	7934	MILLER	二级	ACCOUNTING	7782	CLARK	四级
7	7844	TURNER	三级	SALES	7698	BLAKE	四级
8	7499	ALLEN	三级	SALES	7698	BLAKE	四级
9	7782	CLARK	四级	ACCOUNTING	7839	KING	五级
10	7698	BLAKE	四级	SALES	7839	KING	五级
11	7566	JONES	四级	RESEARCH	7839	KING	五级
12	7902	FORD	四级	RESEARCH	7566	JONES	四级
13	7788	SCOTT	四级	RESEARCH	7566	JONES	四级

2. 外连接（左右连接）

1. 右连接

当我们在做基本连接查询的时候，查询出所有的部门下的员工，我们发现编号为 40 的部门下没有员工，但是要求把该部门也展示出来，我们发现上面的基本查询是办不到的



```
select e.empno, e.ename, d.deptno, d.dname
  from emp e, dept d
 where e.deptno(+) = d.deptno;
```

Select dept

Select emp

Select emp

使用(+)表示左连接或者右连接。

范例：查询出所有员工的上级领导

分析：我们发现使用我们以前的做法发现 KING 的上级领导没有被展示，我们需要使用外连接把他查询出来

```
select e.empno, e.ename, m.empno, m.ename
  from emp e, emp m
 where e.mgr = m.empno(+)
```

EMPNO	ENAME	EMPNO	ENAME
1	7369 SMITH	7902	FORD
2	7499 ALLEN	7698	BLAKE
3	7521 WARD	7698	BLAKE
4	7566 JONES	7839	KING
5	7654 MARTIN	7698	BLAKE
6	7698 BLAKE	7839	KING
7	7782 CLARK	7839	KING
8	7788 SCOTT	7566	JONES
9	7839 KING		
10	7844 TURNER	7698	BLAKE
11	7876 ADAMS	7788	SCOTT
12	7900 JAMES	7698	BLAKE
13	7902 FORD	7566	JONES
14	7934 MILLER	7782	CLARK

如果用 left join 实现：



```
select e.deptno, e.ename, m.deptno, m.ename
from emp e left join emp m
on e.mgr = m.empno
```

	DEPTNO	ENAME	DEPTNO	ENAME
1	20	SMITH	20	FORD
2	30	ALLEN	30	BLAKE
3	30	WARD	30	BLAKE
4	20	JONES	10	KING
5	30	MARTIN	30	BLAKE
6	30	BLAKE	10	KING
7	10	CLARK	10	KING
8	20	SCOTT	20	JONES
9	10	KING		
10	30	TURNER	30	BLAKE
11	20	ADAMS	20	SCOTT
12	30	JAMES	30	BLAKE
13	20	FORD	20	JONES
14	10	MILLER	10	CLARK

因为 (+) 这种形式是 oracle 数据库独有的，所以要求大家一定要掌握 left join 或 right join 方式的写法。

二、子查询

● 什么是子查询？

使用子查询解决问题：谁的工资比 SCOTT 高？





● 子查询的语法

```
SELECT  select_list
FROM    table
WHERE   expr operator
        (SELECT      select_list
         FROM        table);
```

- 子查询 (内查询) 在主查询之前一次执行完成。
- 子查询的结果被主查询使用 (外查询)。

● 子查询的类型

· 单行子查询



· 多行子查询



● 单行子查询

- 只返回一条记录
- 单行操作符

操作符	含义
=	Equal to
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to
<>	Not equal to

- 单行子查询示例 1



```
SELECT ename, job, sal
FROM emp
WHERE job =
      (SELECT job
       FROM emp
       WHERE empno = 7566)
AND sal >
      (SELECT sal
       FROM emp
       WHERE empno = 7782);
```

MANAGER
2450

● 单行子查询示例 2

```
SELECT ename, job, sal
FROM emp
WHERE sal =
      (SELECT MIN(sal)
       FROM emp);
```

800

● 单行子查询示例 3

```
SELECT deptno, MIN(sal)
FROM emp
GROUP BY deptno
HAVING MIN(sal) >
      (SELECT MIN(sal)
       FROM emp
       WHERE deptno = 10);
```

800

● 单行子查询示例 4

范例：查询出比雇员 7654 的工资高，同时从事和 7788 的工作一样的员工

```
select *
from emp t1
where t1.sal > (select t.sal from emp t where t.empno = 7654)
and t1.job = (select t2.job from emp t2 where t2.empno = 7788)
```

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7788	SCOTT	ANALYST	7566	1987/4/19	3000.00		20
2	7902	FORD	ANALYST	7566	1981/12/3	3000.00		20

范例：要求查询每个部门的最低工资和最低工资的雇员和部门名称

```
select d.dname, a.minsal, e.ename
from dept d,
     (select deptno, min(sal) minsal from emp group by deptno) a,
     emp e
where d.deptno = a.deptno
and e.sal = a.minsal
```

	DNAME	MINSAL	ENAME
1	RESEARCH	800	SMITH
2	SALES	950	JAMES
3	ACCOUNTING	1300	MILLER

● 非法使用单行子查询示例



```
SELECT empno, ename
FROM emp
WHERE sal =
      (SELECT MIN(sal)
       FROM emp
       GROUP BY deptno);
```

```
ERROR at line 4:
ORA-01427: single-row subquery returns more than
one row
```

● 多行子查询

- 返回了多条记录
- 多行操作符

● 子查询中的 null 值问题

■ 单行子查询中的 null 值问题

```
SELECT ename, job
FROM emp
WHERE job =
      (SELECT job
       FROM emp
       WHERE ename = 'Mike');
```

■ 多行子查询中的 null 值问题

示例：查询不是老板的员工

```
SELECT *
FROM emp
WHERE empno not in( SELECT mgr
                   FROM emp);
```

多行子查询中 null 值需要注意的问题：



- ❖ **Returning Nulls in the Resulting Set of a Subquery**
- ❖ The SQL statement on the slide attempts to display all the employees who do not have any subordinates. Logically, this SQL statement should have returned 12 rows. However, the SQL statement does not return any rows. One of the values returned by the inner query is a null value, and hence the entire query returns no rows. The reason is that all conditions that compare a null value result in a null. So whenever null values are likely to be part of the results set of a subquery, do not use the NOT IN operator. The NOT IN operator is equivalent to <> ALL.
- ❖ **Notice** that the null value as part of the results set of a subquery is not a problem if you use the IN operator. The IN operator is equivalent to =ANY. For example, to display the employees who have subordinates, use the following SQL statement:
- ❖

```
SELECT emp.last_name
FROM employees emp
WHERE emp.employee_id IN
      (SELECT mgr.manager_id
       FROM employees mgr);
```
- ❖ Alternatively, a WHERE clause can be included in the subquery to display all employees who do not have any subordinates:
- ❖

```
SELECT last_name FROM employees
WHERE employee_id NOT IN
      (SELECT manager_id
       FROM employees
       WHERE manager_id IS NOT NULL);
```

● Exists 用法:

语法:

exists(sql 查询语句)

用法一: 判断 exists 后面的 sql 语句是否为真

sql 查询语句为空 返回值是 false

sql 查询语句有值 返回值就是 true

```
select * from emp where exists (select * from dept where deptno=1)
```

等同于:

```
select * from emp where 1=2
```

```
select * from emp where exists (select * from dept where deptno=10)
```

等同于:

```
select * from emp where 1=1
```

用法二: 判断一个表中是否包含另一张表中外键的记录

范例: 查询有员工的部门

```
select * from dept d
where exists (select * from emp e where e.deptno = d.deptno)
```


三、课堂练习

- 找到员工表中工资最高的前三名，如下格式：

ROWNUM	EMPNO	ENAME	SAL
1	7839	KING	5000
2	7788	SCOTT	3000
3	7902	FORD	3000

- 找到员工表中薪水大于本部门平均薪水的员工。

EMPNO	ENAME	SAL	AUGSAL
7499	ALLEN	1600	1566.66667
7566	JONES	2975	2175
7698	BLAKE	2850	1566.66667
7788	SCOTT	3000	2175
7839	KING	5000	2916.66667
7902	FORD	3000	2175

- 统计每年入职的员工个数

Total	1980	1981	1982	1987
14	1	10	1	2

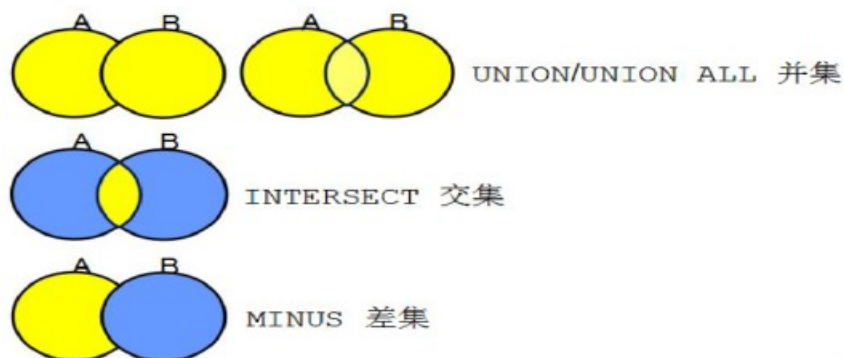
- 补充知识点：Oracle 中的分页查询

ROWNUM:表示行号，实际上此是一个列,但是这个列是一个伪列,此列可以在每张表中出现。

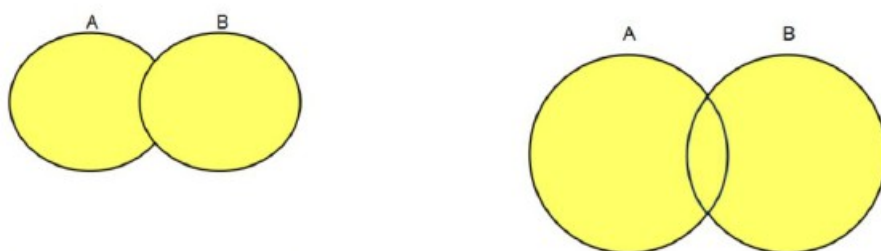
ROWID:表中没行数据指向磁盘上的物理地址。

四、集合运算

● 什么是集合运算？



● 并集



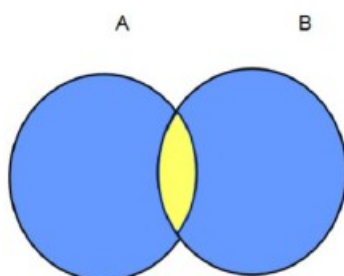
UNION运算符返回两个集合去掉重复元素后的所有记录。

UNION ALL 返回两个集合的所有记录，包括重复的

范例：工资大于 1500，或者是 20 号部门下的员工（并集）

```
select * from emp where sal>1500
union -- 或union all
select * from emp where deptno=20;
```

● 交集



INTERSECT 运算符返回同时属于两个集合的记录

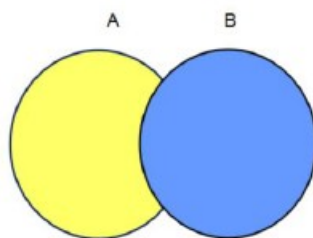
范例：工资大于 1500，并且是 20 号部门下的员工（交集）

```
select * from emp where sal>1500
intersect --交集
```



```
select * from emp where deptno=20;
```

● 差集



MINUS返回属于第一个集合，但不属于第二个集合的记录。

--范例：1981年入职的普通员工（不包括总裁和经理）（差集）

```
select * from emp where to_char(hiredate,'yyyy')='1981'
```

minus

```
select * from emp where job='PRESIDENT' or job='MANAGER'
```

● 集合运算的特征

集合运算两边查询的字段数量、字段类型、顺序必须一致