



中山大學
SUN YAT-SEN UNIVERSITY



2017级 《数据库原理与应用》 第3周

2020.5.22



表数据修改

- Insert语句
- Delete语句
- Update语句



插入数据

- Insert into 表名 values (<列值集>); 列值集必须按列顺序一一对应
- Insert into 表名 (<列集>) values (<列值集>); 列值集必须与前面的列集顺序对应
- 插入字符串值必须用' ' 括住
- 插入空值用NULL表示

插入时间



```
1* insert into emp (empno,ename,hiredate) values (1110,'Jim','20-11月-10')
SQL> /
```

已创建 1 行。

```
SQL> select * from emp;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1110	Jim			20-11月-10			
7369	SMITH	CLERK	7902	17-12月-80	800		20
7499	ALLEN	SALESMAN	7698	20-2月-81	1600	300	30
7521	WARD	SALESMAN	7698	22-2月-81	1250	500	30
7566	JONES	MANAGER	7839	02-4月-81	2975		20
7654	MARTIN	SALESMAN	7698	28-9月-81	1250	1400	30
7698	BLAKE	MANAGER	7839	01-5月-81	2850		30
7782	CLARK	MANAGER	7839	09-6月-81	2450		10
7839	KING	PRESIDENT		17-11月-81	7000		10
7844	TURNER	SALESMAN	7698	08-9月-81	1500	0	30
7900	JAMES	CLERK	7698	03-12月-81	950		30
7902	FORD	ANALYST	7566	03-12月-81	3000		20
7934	MILLER	CLERK	7782	23-1月-82	1300		10

已选择13行。

```
SQL> rollback;
```

回退已完成。

插入时间



```
1* insert into emp (empno,ename,hiredate) values (1110,'Jim',to_date('19901212100000','yyyymmddhh  
SQL> /
```

已创建 1 行。

```
SQL> select * from emp;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1110	Jim			12-12月-90			
7369	SMITH	CLERK	7902	17-12月-80	800		20
7499	ALLEN	SALESMAN	7698	20-2月-81	1600	300	30
7521	WARD	SALESMAN	7698	22-2月-81	1250	500	30
7566	JONES	MANAGER	7839	02-4月-81	2975		20
7654	MARTIN	SALESMAN	7698	28-9月-81	1250	1400	30
7698	BLAKE	MANAGER	7839	01-5月-81	2850		30
7782	CLARK	MANAGER	7839	09-6月-81	2450		10
7839	KING	PRESIDENT		17-11月-81	7000		10
7844	TURNER	SALESMAN	7698	08-9月-81	1500	0	30
7900	JAMES	CLERK	7698	03-12月-81	950		30
7902	FORD	ANALYST	7566	03-12月-81	3000		20
7934	MILLER	CLERK	7782	23-1月-82	1300		10

已选择13行。



带select子句插入

```
SQL> create table z1  
2 (c1 number(4),c2 varchar2(10));
```

表已创建。

```
SQL> insert into z1 select empno,ename from emp where job='SALESMAN';
```

已创建4行。

```
SQL> select * from z1;
```

C1	C2
7499	ALLEN
7521	WARD
7654	MARTIN
7844	TURNER

删除行



```
SQL> select * from z1;
```

C1	C2
7499	ALLEN
7521	WARD
7654	MARTIN
7844	TURNER

```
SQL> delete from z1 where c1=7521;
```

已删除 1 行。

```
SQL> select * from z1;
```

C1	C2
7499	ALLEN
7654	MARTIN
7844	TURNER

【注意】 如果在delete语句后面忘记加where条件，将删除全部行！


```
SQL> select * from z1;
```

C1	C2
----	----

7499	ALLEN
7654	MARTIN
7844	TURNER

```
SQL> update z1 set c1=8000 where c2='TURNER';
```

已更新 1 行。

```
SQL> select * from z1;
```

C1	C2
----	----

7499	ALLEN
7654	MARTIN
8000	TURNER

DDL和DML

- DDL

Create, drop, alter...

- DML

Insert, update, delete

- 什么是事务?

- 事务的作用

- 1事务的原子性

- 2 事务的隔离性

- 3 读一致性

Oracle通过回滚段实现事务



- 显式提交
- 隐式提交
- 自动提交

回滚



中山大學
SUN YAT-SEN UNIVERSITY

- 显式回滚
- 隐式回滚

2022.5.22

聚组



中山大學
SUN YAT-SEN UNIVERSITY

- Group by 子句
- 聚组函数
- Having子句

2022.5.22



聚组函数

- Avg()
- Sum()
- Count()
- Max()
- Min()
- Stddev()
- Variance()



Group by子句

- 聚组表达式的意义
- Select后面可以出现什么?
- 例子: 求每个部门的平均工资

```
SQL> select deptno,avg(sal) from emp group by deptno;
```

DEPTNO	AVG(SAL)
30	1566.66667
20	2258.33333
10	3583.33333



Count和count distinct

```
SQL> select deptno,count(*) from emp group by deptno;
```

DEPTNO	COUNT(*)
10	3
20	3
30	6
40	1

```
SQL> select count(distinct job) from emp;
```

COUNT(DISTINCTJOB)
5

```
SQL> select distinct job from emp;
```

JOB
CLERK
SALESMAN
PRESIDENT
MANAGER
ANALYST

已选择6行。



Having子句

```
SQL> select deptno,count(*) from emp group by deptno having count(*)>5;
```

DEPTNO	COUNT(*)
30	6

Select语句各种子句的执行顺序

- Where->group by->having->order by



连接 (join)

- 无条件连接 (笛卡尔积)
- 有条件连接
- 自连接
- 外连接



```
SQL> select ename,dname from emp,dept;
```

ENAME	DNAME
Jim	Math
SMITH	Math
ALLEN	Math
WARD	Math
JONES	Math
MARTIN	Math
BLAKE	Math
CLARK	Math
KING	Math
TURNER	Math
JAMES	Math
FORD	Math
MILLER	Math
Jim	ACCOUNTING
SMITH	ACCOUNTING
ALLEN	ACCOUNTING
WARD	ACCOUNTING



通过别名消除歧义

```
SQL> select empno,deptno from emp,dept;  
select empno,deptno from emp,dept
```

*

第 1 行出现错误:

ORA-00918: 未明确定义列

```
SQL> select a.empno,b.deptno from emp a,dept b;
```

EMPNO	DEPTNO
1110	10
7369	10
7499	10
7521	10
7566	10
7654	10
7608	10

有条件连接



- 列出员工的姓名及其所在部门名称

```
SQL> select ename,dname from emp,dept where emp.deptno=dept.deptno;
```

ENAME	DNAME
SMITH	RESEARCH
ALLEN	SALES
WARD	SALES
JONES	RESEARCH
MARTIN	SALES
BLAKE	SALES
CLARK	ACCOUNTING
KING	ACCOUNTING
TURNER	SALES
JAMES	SALES
FORD	RESEARCH
MILLER	ACCOUNTING

已选择12行。



自连接

- 一个表与它自己进行连接
- 例子：列出员工的姓名及其上司的姓名

```
SQL> select a.ename,b.ename from emp a,emp b  
2  where a.mgr=b.empno;
```

ENAME	ENAME
SMITH	FORD
ALLEN	BLAKE
WARD	BLAKE
JONES	KING
MARTIN	BLAKE
BLAKE	KING
CLARK	KING
TURNER	BLAKE
JAMES	BLAKE
FORD	JONES
MILLER	CLARK

已选择11行。



外连接

```
1 select a.ename,b.ename from emp a,emp b
2* where a.mgr=b.empno(+)
SQL>
SQL>
SQL> /
```

ENAME	ENAME
-----	-----
SMITH	FORD
ALLEN	BLAKE
WARD	BLAKE
JONES	KING
MARTIN	BLAKE
BLAKE	KING
CLARK	KING
KING	
TURNER	BLAKE
JAMES	BLAKE
FORD	JONES
MILLER	CLARK

已选择12行。



各种连接

- 内连接
- 外连接
- 左外连接
- 右外连接



子查询

- 在select语句中嵌套select子句
- 例子：找出比Jones工资高的员工

```
SQL> select * from emp  
2 where sal > (select sal from emp where ename='JONES');
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT		17-11月-81	7000		10
7902	FORD	ANALYST	7566	03-12月-81	3000		20



列出工资比上司高的员工

```
SQL> select * from emp a
      2 where sal > (select sal from emp where empno=a.mgr);
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7902	FORD	ANALYST	7566	03-12月-81	3000		20

```
SQL> select * from emp;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-12月-80	800		20
7499	ALLEN	SALESMAN	7698	20-2月-81	1600	300	30
7521	WARD	SALESMAN	7698	22-2月-81	1250	500	30
7566	JONES	MANAGER	7839	02-4月-81	2975		20
7654	MARTIN	SALESMAN	7698	28-9月-81	1250	1400	30
7698	BLAKE	MANAGER	7839	01-5月-81	2850		30
7782	CLARK	MANAGER	7839	09-6月-81	2450		10
7839	KING	PRESIDENT		17-11月-81	7000		10
7844	TURNER	SALESMAN	7698	08-9月-81	1500	0	30
7900	JAMES	CLERK	7698	03-12月-81	950		30
7902	FORD	ANALYST	7566	03-12月-81	3000		20
7934	MILLER	CLERK	7782	23-1月-82	1300		10

已选择12行。

综合练习



中山大學
SUN YAT-SEN UNIVERSITY

- 列出工资比最高工资的一半要高的员工
- 列出平均工资超过公司平均工资的部门名称
- 列出人数最多的两个工种
- 列出人数最多的部门名称
- 列出工资差（最高-最低）最大的部门名称

```
SQL> create user y1 identified by abc;
```

用户已创建。

```
SQL> create user y2 identified by 123;  
create user y2 identified by 123
```

*

第 1 行出现错误:

ORA-00988: 口令缺失或无效

```
SQL> create user y2 identified by "123";
```

用户已创建。

```
SQL> |
```



```
SQL> alter user y1 identified by xyz  
2 ;
```

用户已更改。

```
SQL>
```

删除用户



- Drop user 用户名;
- Drop user 用户名 **cascade**;

- 系统特权：支配系统中一般性资源的能力，一般由DBA授予
- 对象特权：支配某一具体数据库对象的能力，一般由对象拥有者授予



查看所有系统特权

```
SQL> connect / as sysdba
```

已连接。

```
SQL> desc dba_sys_privs
```

名称

是否为空? 类型

GRANTEE

NOT NULL VARCHAR2(30)

PRIVILEGE

NOT NULL VARCHAR2(40)

ADMIN_OPTION

VARCHAR2(3)

```
SQL> select privilege from dba_sys_privs where grantee='DBA';
```

PRIVILEGE

CHANGE NOTIFICATION

ADMINISTER ANY SQL TUNING SET

ALTER ANY SQL PROFILE

CREATE RULE

EXPORT FULL DATABASE

EXECUTE ANY EVALUATION CONTEXT

DEQUEUE ANY QUEUE

DROP ANY INDEXTYPE

ALTER ANY INDEXTYPE

EXECUTE ANY LIBRARY

CREATE ANY LIBRARY



几个有趣的系统特权

- SELECT ANY TABLE
- DROP USER, CREATE USER
- CREATE SESSION
- UNLIMITED TABLESPACE



向用户授予系统特权

```
SQL>
```

```
SQL> grant create user to y1;
```

授权成功。

```
SQL> grant create user to y2 with admin option;
```

授权成功。



撤销用户的系统特权

```
SQL> revoke create user from y2;
```

撤销成功。



```
SQL>
```

```
SQL>
```

```
SQL> grant connect to y1;
```

授权成功。

```
SQL> connect scott/tiger
```

已连接。

```
SQL> grant select on emp to y1;
```

授权成功。

```
SQL> grant select on dept to y1 with grant option;
```

授权成功。

```
SQL>
```

■ With admin option和with grant option的区别
?



有哪些对象特权?

- Select
- Update
- Insert
- Delete
- Alter
- Index
- Execute

撤销对象特权



```
SQL>
```

```
SQL> revoke select on dept from y1;
```

撤销成功。

```
SQL>
```

```
SQL> |
```

角色 (role)

- 什么是角色——角色是权限和角色的集合
- 角色作用——简化授权管理

```
SQL> create role r1;
```

角色已创建。

```
SQL> grant select on emp to r1;
```

授权成功。

```
SQL> connect / as sysdba
```

已连接。

```
SQL> grant create user to r1;
```

授权成功。

```
SQL> grant r1 to y1;
```

授权成功。

- Drop role 角色名;
- 权限/角色树形结构

公众用户public



中山大學
SUN YAT-SEN UNIVERSITY

- 什么是公众用户
- 对公众用户授权的作用

2017.10.10



中山大學
SUN YAT-SEN UNIVERSITY

Thanks

FAQ时间