

数据库第十五周作业

19336035 陈梓乐

1. 建一个有外键和值约束的表（例如之前的 SC 表），再通过数据字典视图查出这些约束的信息。

```
select substr(owner, 1, 5) owner,
       substr(constraint_name, 1, 12) constraint_name,
       constraint_type,
       substr(r_owner, 1, 5) r_owner,
       substr(r_constraint_name,1, 12) r_constraint_name
from   user_constraints
where  table_name = 'SCORE';
```

终端 问题 输出 调试控制台

SQL =

```
select substr(owner, 1, 5) owner,
2      substr(constraint_name, 1, 12) constraint_name,
3      constraint_type,
4      substr(r_owner, 1, 5) r_owner,
5      substr(r_constraint_name,1, 12) r_constraint_name
6 from   user_constraints
7 where  table_name = 'SCORE';
```

OWNER	CONSTRAINT_NAME	C	R_OWNER	R_CONSTRAINT_NAME
SCOTT	SYS_C0011439	C		
SCOTT	SYS_C0011440	C		
SCOTT	SYS_C0011441	C		
SCOTT	SYS_C0011442	P		
SCOTT	SYS_C0011443	R	SCOTT	SYS_C0011435
SCOTT	SYS_C0011444	R	SCOTT	SYS_C0011438

已选择6行。

SQL =

2. 查询 emp 表上的所有触发器及其对应代码

```
select substr(trigger_name,1, 10),
       substr(trigger_type,1, 30),
       trigger_body
from   user_triggers
where  table_name = 'EMP';
```

SQL =

```
select substr(trigger_name,1, 10),
2      substr(trigger_type,1, 30),
3      trigger_body
4 from   user_triggers
5 where  table_name = 'EMP';
```

SUBSTR(TRIGGER_NAME, SUBSTR(TRIGGER_TYPE,1,30)	TRIGGER_BODY
EMP_TIME	BEFORE STATEMENT

begin
if (to_char(sysdate,'DY') in ('星期六','星期日')) or
(to_char(

SQL =

3. 查看当前 public 用户有哪些系统特权和角色？

sys 下：

```
select grantee,
       privilege
from   dba_sys_privs
where  grantee = 'PUBLIC';
```

终端 问题 输出 调试控制台

SQL = select grantee,
2 privilege
3 from dba_sys_privs
4 where grantee = 'PUBLIC';

未选定行

SQL = grant UNLIMITED TABLESPACE to public;

授权成功。

SQL = select grantee,
2 privilege
3 from dba_sys_privs
4 where grantee = 'PUBLIC';

GRANTEE	PRIVILEGE
PUBLIC	UNLIMITED TABLESPACE

SQL =

4. 查询 emp 表上索引的信息

```
select * from user_indexes where table_name = 'EMP';
```

SQL = select * from user_indexes where table_name = 'EMP';

INDEX_NAME	TABLE_NAME	INDEX_TYPE	TABLE_TYPE	TABLE_OWNER
UNIQUENESS	COMPRESS	PREFIX_LENGTH	TABLESPACE_NAME	
INI_TRANS	MAX_TRANS	INITIAL_EXTENT		
NEXT_EXTENT	MIN_EXTENTS	MAX_EXTENTS	PCT_INCREASE	PCT_THRESHOLD
INCLUDE_COLUMN	FREELISTS	FREELIST_GROUPS	PCT_FREE	LOGGIN
BLEVEL	LEAF_BLOCKS	DISTINCT_KEYS	AVG_LEAF_BLOCKS_PER_KEY	AVG_DATA_BLOCKS_PER_KEY
CLUSTERING_FACTOR	STATUS	NUM_ROWS	SAMPLE_SIZE	LAST_ANALYZED
DEGREE	PAR	T	G	S
BUFFER_	FLASH_C	CELL_FL	USE	
DURATION	ITYP_NAME	PCT_DIRECT_ACCESS	ITYP_OWNER	
PARAMETERS				
GLO	DOMIDX_STATU	DOMIDX_FUNCIDX	JOI	IOT
DRO	VISIBILI	DOMIDX_MANAGEM	SEG	
PK_EMP	EMP	NORMAL	TABLE	SCOTT
UNIQUE	DISABLED	USERS	TABLE	SCOTT
1048576	1	2147483645	255	65536
1	1	14	10 YES	0
1	1	1 VALID	14	1 14 25-3月 -22
NO	N	N	DEFAULT	DEFAULT
NO	NO	NO	VISIBLE	

SQL =

5. 有些数据字典视图称为“动态性能视图”，记录系统物理底层信息，以 V\$ 开头，统计下总共多少个这样的视图

sys 下:

```
select count(*) from dba_views where view_name like 'V$%';
```

```
SQL = select count(*) from dba_views where view_name like 'V$%';
```

```
COUNT(*)  
-----  
4
```

```
SQL =
```

2. 用实验比较一下 with admin option 和 with grant option 在操作逻辑上的区别?

- with admin option 只能在赋予 system privilege 的时使用
- with grant option 只能在赋予 object privilege 的时使用
- **撤消带有admin option 的system privileges 时，连带的权限将保留**

例如：

1. DBA 给了CREATE TABLE 系统权限给JEFF WITH ADMIN OPTION
2. JEFF CREATES TABLE
3. JEFF grants the CREATE TABLE 系统权限给EMI
4. EMI CREATES A table
5. DBA 撤消CREATE TABLE 系统权限从JEFF

结果：

JEFF'S TABLE 依然存在，但不能创建新的TABLE 了

EMI'S TABLE 依然存在，他还保留着CREATE TABLE 系统权限。

- **撤消带有grant option 的object privileges 时，连带的权限也将撤消**

例如：

1. JEFF 给了SELECT object privileges 在EMP 上 WITH GRANT OPTION
2. JEFF 给了SELECT 权限在EMP 上 TO EMI
3. 后来，撤消JEFF的SELECT 权限

结果：

EMI 的权限也被撤消了

3. 阅读《Introduction to PL/SQL》第 15 章，试使用适当的 LOB 字段，将一个 pdf 文件写入到 Oracle 数据库的表中储存，然后完成反向操作，把保存了 pdf 文件的字段内容还原为原文件

将第15周作业列表置于 data 文件夹下并更名为 15.pdf，我们将以该文件为例展示二进制数据的读写。

读入数据并保存到数据库中

```
Czile D: > Work > 数据库 17:11:26  
pwsh > cp "D:\download\第15周作业.pdf" data  
  
Czile D: > Work > 数据库 17:12:40  
pwsh >
```

先来创建记录数据的表格：

```
create table filedata (  
    fname varchar2(100),  
    data blob  
);
```

然后创建 data 目录，并执行如下代码：

```

declare
    fname    varchar2(100);
    fdata    blob;
    content  blob;
    f        bfile;
begin
    fname := '15.pdf';
    f := bfilename('DATA', fname);

    insert into scott.filedata
    values      (fname, empty_blob())
    returning   data
    into        content;

    dbms_lob.fileopen(f, dbms_lob.file_readonly);
    dbms_lob.loadfromfile(content, f, dbms_lob.getlength(f));
    dbms_lob.fileclose(f);
    commit;
end;
/

```

效果如下：

```

SQL = declare
2      fname    varchar2(100);
3      fdata    blob;
4      content  blob;
5      f        bfile;
6  begin
7      fname := '15.pdf';
8      f := bfilename('DATA', fname);
9
10     insert into scott.filedata
11     values      (fname, empty_blob())
12     returning   data
13     into        content;
14
15     dbms_lob.fileopen(f, dbms_lob.file_readonly);
16     dbms_lob.loadfromfile(content, f, dbms_lob.getlength(f));
17     dbms_lob.fileclose(f);
18     commit;
19 end;
20 /

PL/SQL 过程已成功完成。

SQL = select * from scott.filedata
2  ;

FNAME
-----
DATA
-----
15.pdf
255044462D312E360D25E2E3CFD30D0A36322030206F626A0D3C3C2F4C696E656172697A656420312F4C203135303531302F4F2036342F45203134343634342F4E20312F54
203135303230322F48205B

```

取出数据并保存到文件中

```

declare
    fname    varchar2(100);
    content  blob;
    buffer   raw(32767);
    f        UTL_FILE.FILE_TYPE;
    csize    integer;
    bsize    integer;
    amount   integer;
    offset   number(38) := 1;
begin
    select data
    into   content
    from   scott.filedata;

    select fname
    into   fname
    from   scott.filedata;

    csize := dbms_lob.getchunksize(content);
    if (csize <= 32767) then
        bsize := csize;
    else
        bsize := 32767;
    end if;
    amount := bsize;
    dbms_lob.open(content, dbms_lob.lob_readonly);

    f := UTL_FILE.FOPEN(
        location    => 'DATA',
        filename    => '[db] ' || fname,
        open_mode   => 'wb',
        max_linesize => 32767
    );

    while amount >= bsize loop
        dbms_lob.read(
            lob_loc => content,
            amount  => amount,
            offset  => offset,
            buffer  => buffer
        );
        offset := offset + amount;
        utl_file.put_raw(
            file    => f,
            buffer  => buffer
        );
        utl_file.fflush(file => f);
    end loop;
    utl_file.fflush(file => f);
    utl_file.fclose(f);
    dbms_lob.close(content);
end;
/

```

下图将说明以上代码可以实现将数据库中的数据保存到文件：

15.md15-2 write-file.sql[db] 15.pdf x15-1 read-file.sql设置扩展: PowerShelldata > [db] 15.pdf1 / 1自动缩放

第 15 周作业

- 看完第 15 周课程幻灯片，通过数据字典视图查询以下信息
- 1) 建立一个有外键和值约束的表（例如之前的 SC 表），再通过数据字典视图查出这些约束的信息
- 2) 查询 emp 表上的所有触发器及其对应代码
- 3) 查看当前 public 用户有哪些系统特权和角色？
- 4) 查询 emp 表上索引的信息

终端问题输出调试控制台pwsh + - ^ x

```
45         file  => f,
46         buffer => buffer
47     );
48     utl_file.fflush(file => f);
49 end loop;
50 utl_file.fflush(file => f);
51 utl_file.fclose(f);
52 dbms_lob.close(content);
53 end;
54 /
```

PL/SQL 过程已成功完成。

SQL =

Czile D: > Work > 数据库 19:09:00

pwsh > ls data

Directory: D:\Work\数据库\data

Mode	LastWriteTime	Length	Name
-a---	2022/5/27 19:08	150510	[db] 15.pdf
-a---	2022/5/20 17:19	467356	000001.SS.csv
-a---	2022/5/27 15:28	150510	15.pdf
-a---	2022/2/25 15:37	114999998	2.txt
-a---	2022/3/11 14:59	55	class_data.csv
-a---	2022/3/11 16:04	2873	EMP.CSV