

Chapter 13

Data Dictionary and Dynamic SQL Scripts

1



Tables in the Oracle Database

- User Tables
 - Collection of tables to store data

- Data Dictionary Tables
 - Collection of tables created and maintained by Oracle server
 - Not directly accessible
 - users typically access the data dictionary via **views**
 - Owned by the user SYS

2



Data Dictionary Views

- Dynamic Views
 - Begin with v\$
 - Are continuously updated to monitor system performance
 - Generally for DBA usage
 - eg: SELECT username FROM v\$session
- Static Views
 - Information including users, privileges granted to users, database object names, table constraints, and auditing information
 - 3 prefixes
 - **USER_** view information regarding user's own schema objects
 - **ALL_** view information on user's own objects and those GRANTED
 - **DBA_** view information regarding all objects (must be DBA)

3



Querying the Data Dictionary

- **USER_OBJECTS**
 - describes all objects owned by the current user
 - tables, views, sequences, indexes, synonyms, functions, procedures, triggers, packages, others

```
SQL> desc USER_OBJECTS

SQL> SELECT object_name, object_type, created, status
       FROM USER_OBJECTS
       ORDER BY object_type, object_name;
```

- Note
 - data dictionary views display object names in **UPPER** case

4



Confirming Views

- **USER_VIEWS** (we used in Ch12)

- contains details of each view owned by the current user

```
SQL> desc USER_VIEWS
```

```
SQL> SELECT view_name, text
        FROM USER_VIEWS;
VIEW_NAME
-----
TEXT
-----
ARTICLEWRITER
SELECT articlenum, title, type, issue, length,
       ln || ', ' || fn AS "Author"
```

```
SQL> set long 5000
SQL>/
```

5



Confirming Tables

- **USER_TABLES**

- describes **base tables** owned by the current user

```
SQL> desc USER_TABLES
SQL> SELECT table_name, blocks, num_rows, avg_row_len,
        to_char(last_analyzed, 'DD-MON-RR HH24:MI')
        FROM USER_TABLES
        ORDER BY table_name;
```

TABLE_NAME	BLOCKS	NUM_ROWS	AVG_ROW_LEN	TO_CHAR(LAST_AN
ARTICLE	5	30	52	13-APR-07 22:00
COURSE	5	30	65	09-APR-07 22:00
ENROLLMENT	5	226	49	09-APR-07 22:00
GRADE	20	2004	49	09-APR-07 22:00
GRADE_CONVERSION	5	15	46	09-APR-07 22:00
GRADE_TYPE	5	6	45	09-APR-07 22:00
GRADE_TYPE_WEIGHT	5	300	48	09-APR-07 22:00
INSTRUCTOR	5	10	85	09-APR-07 22:00
RECEIPT	1756	452899	22	30-NOV-06 22:00

6



Gathering Statistics (lib, pg 817+)

- The Oracle Optimizer uses object statistics to decide how to efficiently execute statements
- Oracle does not continuously update object statistics
 - why not?
- Oracle 10g schedules GATHER_STATS_JOB
 - collects statistics on database objects when the object has no previously gathered statistics or the existing statistics are stale because the underlying object has been modified significantly (eg: >10% of the rows)
- For how long are these statistics accurate?
- Can execute DBMS_STATS to gather statistics on the spot

```
SQL> EXEC DBMS_STATS.gather_table_stats('SCOTT', 'EMPLOYEES');
PL/SQL procedure successfully completed.

SQL> EXEC DBMS_STATS.gather_schema_stats(ownname => 'SCOTT');
PL/SQL procedure successfully completed.
```

7



Confirming Table Columns

- USER_TAB_COLUMNS
 - describes the columns of current user's tables and views

```
SQL> desc USER_TAB_COLUMNS

SQL> SELECT column_name, data_type, data_precision,
           data_scale, avg_col_len
       FROM USER_TAB_COLUMNS
       WHERE table_name = 'ARTICLE'
```

COLUMN_NAME	DATA_TYPE	DATA_PRECISION	DATA_SCALE	AVG_COL_LEN
ARTICLENUM	NUMBER	4	0	2
TITLE	VARCHAR2			28
TYPE	CHAR			3
ISSUE	DATE			7
...				

8



What Data Dictionary Views Are Available?

- DICT

- shows the name & a comment for each data dictionary view the current user can use

```
SQL> desc DICT
```

```
SQL> SELECT *  
      FROM DICT  
      WHERE UPPER(COMMENTS) LIKE '%TABLE%'
```

```
TABLE_NAME  
-----  
COMMENTS  
-----  
USER_CATALOG  
Tables, Views, Synonyms and Sequences owned by the user  
ALL_CATALOG  
All tables, views, synonyms, sequences accessible to the user  
...
```

9



Data Dictionary Practice

- Which data dictionary views seem to relate to constraints?

10

What Do The Columns in a Data Dictionary View Mean?

■ DICT_COLUMNS

- shows the name & a comment for each column in a data dictionary view

```
SQL> desc DICT_COLUMNS
SQL> SELECT column_name, comments
       FROM DICT_COLUMNS
       WHERE table_name = 'USER_TABLES'
          AND column_name IN ('BLOCKS', 'AVG_ROW_LEN',
                              'NUM_ROWS', 'LAST_ANALYZED')
```

COLUMN_NAME	COMMENTS
NUM_ROWS	The number of rows in the table
BLOCKS	The number of used blocks in the table
AVG_ROW_LEN	The average row length, including row overhead
LAST_ANALYZED	The date of the most recent time this table was analyzed

11

Confirming Constraints

■ USER_CONSTRAINTS

- describes the constraints on tables owned by the current user
- constraint type codes: C P R U

```
SQL> SELECT constraint_name, constraint_type,
           search_condition
       FROM USER_CONSTRAINTS
       WHERE table_name = 'WRITER';
```

CONSTRAINT_NAME	C	SEARCH_CONDITION
SYS_C008004	C	"WRITERID" IS NOT NULL
SYS_C008005	C	"LN" IS NOT NULL
SYS_C008006	C	"FN" IS NOT NULL
WRITER_FREELANCER_CK	C	freelancer IN ('Y', 'N')
WRITER_WRITERID_PK	P	
WRITER_CONTACT_FK	R	

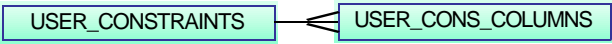
12

Viewing the Columns Associated with Constraints

- **USER_CONS_COLUMNS**
 - shows which column(s) a constraint operates on
 - especially useful for constraints with a system-assigned name

```
SQL> SELECT constraint_name, column_name
      FROM USER_CONS_COLUMNS
      WHERE table_name = 'WRITER'
```

CONSTRAINT_NAME	COLUMN_NAME
SYS_C008004	WRITERID
SYS_C008005	LN
SYS_C008006	FN
WRITER_CONTACT_FK	CONTACT
WRITER_FREELANCER_CK	FREELANCER
WRITER_WRITERID_PK	WRITERID



Joining Dictionary Views

- Related views can be joined via a common column
 - CONSTRAINT_NAME common to user_constraints and user_cons_columns
 - eg: show details for WRITER table's constraints

```
SQL> SELECT ucc.constraint_name, column_name,
      search_condition, constraint_type
      FROM user_constraints uc, user_cons_columns ucc
      WHERE UCC.CONSTRAINT_NAME = UC.CONSTRAINT_NAME
      AND ucc.table_name = 'WRITER'
      ORDER BY 1, 2;
```

CONSTRAINT_NAME	COLUMN_NAME	SEARCH_CONDITION	C
SYS_C008004	WRITERID	"WRITERID" IS NOT NULL	C
SYS_C008005	LN	"LN" IS NOT NULL	C
SYS_C008006	FN	"FN" IS NOT NULL	C
WRITER_CONTACT_FK	CONTACT		R
WRITER_FREELANCER_CK	FREELANCER	freelancer IN ('Y', 'N')	C
WRITER_WRITERID_PK	WRITERID		P



Retrieving an Object's DDL

- The Data Dictionary contains each object's definition
- Can use **DBMS_METADATA.GET_DDL** to extract the SQL statement that defines the object

```
SQL> SELECT DBMS_METADATA.GET_DDL('TABLE', 'ARTICLE', 'TTROLLEN')  
FROM DUAL
```

15



Object Dependencies

- Some types of objects are dependent on other objects in order to function
 - eg: a view depends on base tables and their columns
- **USER_DEPENDENCIES**
 - describes dependencies between views, procedures, packages, functions, and triggers owned by the current user

```
SQL> desc USER_DEPENDENCIES  
SQL> SELECT name, referenced_name,  
           referenced_type, referenced_owner  
FROM USER_DEPENDENCIES  
WHERE name = 'ARTICLE_WRITER'
```

NAME	REFERENCED_NAME	REFERENCED_TYPE	REFERENCED_OWNER
ARTICLE_WRITER	WRITER	TABLE	TTROLLEN
ARTICLE_WRITER	ARTICLE	TABLE	TTROLLEN



SQL*PLUS Scripts

- What is a script?
 - a text file containing SQL statements and SQL*Plus commands
 - eg: page 640
- How do I run a script?
 - SQL> **start** *scriptname*
 - SQL> **@***scriptname*
 - SQL> **@s:**\cis119\scripts\createstudent.sql
- What is the default script location?
 - c:\oracle\product\10.2.0\client_1\bin
 - c:\oracle\ora92\bin
- What is the default script filename extension?
 - .sql

17



Common SQL*Plus Commands

- SPOOL filename
- SET PAGESIZE [n|off]
- SET LINESIZE n
- SET FEEDBACK [numlines|off]
 - see/suppress number of rows processed
- SET VERIFY [on|off]
 - see/suppress substitution variable details
- comments
 - single-line -- or REM
 - multi-line /* */
- line continuation
 - use a - at end of line in a SQL*Plus command
- SET TERMOUT [on|off]
 - see/suppress on-screen output generated by commands executed from a *script*
- HOST [command]
 - issue operating system commands
- ACCEPT
- DEFINE
- UNDEFINE
- PROMPT
- PAUSE


18



SQL*Plus Substitution Variables

- An SQL*Plus feature
- What is a variable?
- Substitution variables make SQL statements more **versatile**
- When a **substitution variable** is embedded in a SQL statement, SQL*Plus substitutes the variable's value into the statement, prompting for its value if not previously created, then sends the command to the oracle server for processing
- Can be created **explicitly**
 - DEFINE
 - ACCEPT
- Will be created **implicitly** if not previously created
 - &

19



SQL*Plus Substitution Variables: Example

```
SQL> SELECT constraint_name, constraint_type,
        search_condition
        FROM USER_CONSTRAINTS
        WHERE table_name = '&v_table';
```

```
Enter value for v_table: WRITER
old  4: WHERE table_name = '&v_table'
new  4: WHERE table_name = 'WRITER'
CONSTRAINT_NAME          COLUMN_NAME
-----
SYS_C008004              WRITERID
SYS_C008005              LN
SYS_C008006              FN
WRITER_CONTACT_FK       CONTACT
WRITER_FREELANCER_CK    FREELANCER
WRITER_WRITERID_PK      WRITERID
```

20

ACCEPT

```
ACCEPT variable [NUMBER|CHAR|DATE] [FORMAT format]
[PROMPT text] [HIDE]
```

- A SQL*Plus command
- Defines a **user variable**, its data type and a prompt message
- Can refer to the variable throughout the script
 - won't get prompted for each occurrence
 - the variable retains its value until exit SQL*Plus or the variable is UNDEFINED

```
ACCEPT v_table CHAR PROMPT 'Which table:'
ACCEPT hired DATE FORMAT 'mm/dd/yyyy' -
    DEFAULT '01/01/2001' PROMPT 'Enter hire date:'
ACCEPT v_length NUMBER PROMPT 'What size?'
ACCEPT pw CHAR PROMPT 'Password' HIDE
```

21

ACCEPT Example

```
-- tabledoc.sql
set verify off
ACCEPT v_table CHAR PROMPT 'Which table:'

Prompt
Prompt &v_table table's structure:
desc &v_table
select '&v_table' has ' || count(*) || ' records.' from &v_table;

Prompt Table's Primary Key is:
select column_name, search_condition
  from user_constraints uc, user_cons_columns ucc
 where ucc.constraint_name = uc.constraint_name
       and ucc.table_name = upper('&v_table')
       and constraint_type = 'P';

undefine v_table
set verify on
```

22



DEFINE

- A SQL*Plus command
- Specifies a user variable and assigns it a **CHAR** value
- Lists the value and data type of existing variables

```
SQL> DEFINE v_type = BUS
SQL> SELECT * FROM article WHERE type = '&v_type'

SQL> DEFINE SAL = 5000
SQL> DEFINE
```

23



UNDEFINE

- A SQL*Plus command
- Deletes user variables that were defined either explicitly (with ACCEPT or DEFINE) or implicitly

```
SQL> UNDEFINE v_type

SQL> UNDEFINE sal

SQL> DEFINE
```

24

Suppressing the Use of Substitution Variables

(Ch 10)

- In SQL*Plus, & indicates that what follows is a substitution variable

```
SQL> SELECT title, issue, length
      FROM article
      WHERE title = 'Rock & Roll'
Enter value for roll:
```

- SET DEFINE OFF

- suspends the use of substitution variables
- from then on, & isn't interpreted as a substitution variable

```
SQL> set define off
SQL> SELECT title, issue, length
      FROM article
      WHERE title = 'Rock & Roll';
SQL> set define on
```

25

PROMPT

```
PROMPT [message]
```

- A SQL*Plus command
- Displays a specified message or a blank line

```
-- prompt_demo.sql

PROMPT Enter a valid employee ID
PROMPT For Example 7566
PROMPT
ACCEPT ENUMBER NUMBER PROMPT 'Please Enter Employee Number:'

SELECT ename, sal, deptno
      FROM scott.emp
      WHERE empno=&ENUMBER;

UNDEFINE ENUMBER
```

26



PAUSE

PAUSE [message]

- A SQL*Plus command that halts execution until user presses the [Enter] key
- Is frequently used after a PROMPT to ensure the user has time to read the prompt before a script continues to execute
- Risk: it could be a long wait... resources held

Pause

27



Generate Dynamic SQL

- What does all this concatenation produce?

```
SQL> SELECT 'ALTER TABLE ' || table_name ||  
          ' DISABLE CONSTRAINT ' || constraint_name || ';'   
        FROM user_constraints  
        WHERE table_name = '&v_table';
```

```
Enter value for v_table: ENROLLMENT  
ALTER TABLE ENROLLMENT DISABLE CONSTRAINT ENR_STU_FK;  
ALTER TABLE ENROLLMENT DISABLE CONSTRAINT ENR_SECT_FK;  
ALTER TABLE ENROLLMENT DISABLE CONSTRAINT ENR_MODIFIED_DATE_NULL;  
ALTER TABLE ENROLLMENT DISABLE CONSTRAINT ENR_STUDENT_ID_NULL;  
ALTER TABLE ENROLLMENT DISABLE CONSTRAINT ENR_ENROLL_DATE_NULL;  
ALTER TABLE ENROLLMENT DISABLE CONSTRAINT ENR_MODIFIED_BY_NULL;  
ALTER TABLE ENROLLMENT DISABLE CONSTRAINT ENR_SECTION_ID_NULL;  
ALTER TABLE ENROLLMENT DISABLE CONSTRAINT ENR_CREATED_BY_NULL;  
ALTER TABLE ENROLLMENT DISABLE CONSTRAINT ENR_CREATED_DATE_NULL;  
ALTER TABLE ENROLLMENT DISABLE CONSTRAINT ENR_PK;
```

28



Generate Dynamic SQL

- Can issue a series of commands whose results are concatenated with literals to produce SQL statements
- Can spool (or copy) these rows to a text file to build a **script**
- Thus we can **use SQL to generate SQL**

```
-- disable.sql
Prompt This script disables the constraints for table ENROLLMENT
set pagesize 0
set feedback off
ALTER TABLE ENROLLMENT DISABLE CONSTRAINT ENR_STU_FK;
ALTER TABLE ENROLLMENT DISABLE CONSTRAINT ENR_SECT_FK;
ALTER TABLE ENROLLMENT DISABLE CONSTRAINT ENR_MODIFIED_DATE_NULL;
ALTER TABLE ENROLLMENT DISABLE CONSTRAINT ENR_STUDENT_ID_NULL;
...
ALTER TABLE ENROLLMENT DISABLE CONSTRAINT ENR_PK;
set pagesize 50
set feedback on
```