

<A>APPENDIX E

The XX\$ Tables

The XX\$ Tables

The actual heart of Oracle is the K and X\$ table structs at the heart of its code. The V\$ dynamic tables are all based on these K and X\$ structs. Normally you cannot see these structures, they cannot be directly selected from except from the SYS or INTERNAL user and cannot be described using the DESCRIBE command. The best possible look a user can get at these structures, without being a part of Oracle Corporation and seeing internal documentation is to look at the definitions of the V\$ tables as shown in the view V\$FIXED_VIEW_DEFINITION by use of a code snippet such as this one:

```
column view_name format a25
column view_definition format a50 word_wrapped
set lines 80 pages 0
start title80 "V$ View Definitions"
spool rep_out\&db\v$view
select * from v$fixed_view_definition;
/
spool off
```

This code snippet produces output (for ORACLE7) similar to:

```
V$ROWCACHE      select
                  kqrstcid,decode(kqrsttyp,1,'PARENT','SUBORDINATE')
                  ,
                  decode(kqrsttyp,2,kqrstsno,null),kqrsttxt,kqrstcsz
                  ,kqrstusg,kqrstfcs,
                  kqrstgrq,kqrstgmi,kqrstsrq,kqrstsmi,kqrstsko,kqrst
                  mrq,kqrstmfl, kqrstilr,kqrstifr,kqrstisr from
                  x$kqrst
```

As you can see, the columns in the K and X\$ tables are very descriptive and their meanings are intuitive to the most casual observer (right.). One of the experts in the field of Oracle Internals is coming out with a book on Oracle internals, perhaps he will cover these elusive structures so that us poor DBAs can understand them! In ORACLE8 instead of the V\$ tables being the lowest on the “V” view hierarchy, the GV\$ tables take this distinction. The GV\$ views

(the G stands for Global) have a column added that corresponds to the instance numbers. This change to the V\$ views was made to allow monitoring of a shared instance environment from a single instance. If you aren't using a shared instance environment this new feature isn't of any importance.

Anyway the data dictionary tables you should be concerned with are actually derived from the K and X\$ tables as well as from user and program input. These are collectively known as the XX\$ tables (at least by me). I call them the XX\$ tables because generally, not always but most times, they end in "\$". For example, COL\$, TAB\$, FET\$ are all XX\$ tables and are the base data dictionary tables that you as a DBA should be concerned with. The tables themselves are built when the database is created by the **sql.bsq** script that is located in the \$ORACLE_HOME/rdbms/admin or \$ORACLE_HOME/dbs directories or their equivalent on your platform. Essentially, any table owned by SYS is an XX\$ table. The **sql.bsq** script has a comment line for virtually every column in every XX\$ table. I suggest it is as another must read for the DBA. Luckily for you the **sql.bsq** script is only about seventy pages long.

If you are an experienced DBA and have monitored Oracle databases for tables with multiple extents you no doubt noted with dispair the data dictionary tables where your biggest culprit, especially in environments with numerous tables and other database objects. When you asked anyone with Oracle support about fixing this "feature" their answer was generally "That is an Oracle internal set of definitions and if you change them we will no longer support you". Not any more. Oracle will now support editing of the sql.bsq script to "improve" the storage values of the base Oracle data dictionary tables. However, **DO NOT** alter any of the table, cluster or column names or any of the datatypes, or specifications. This edit of sql.bsq for sizing must be done **BEFORE** you issue the CREATE DATABASE command. The following objects should probably have their storage initial extent values increased if you have a large environment (my

current environment has around 1000 base tables, with supporting indexes, triggers, views and packages):

Table	Initial Extent
ACCESS\$	180K
ARGUMENT\$	300K
COM\$	100K
CON\$	100K
DEPENDENCY\$	180K
IDL_CHAR\$	450K
IDL_SB4\$	700K
IDL_UB1\$	2M (yes, 2 megabytes)
IDL_UB2\$	3M (yes, 3 megabytes)
OBJ\$	450K
OBJAUTH\$	180K
PROCEDURE\$	32K
SEQ\$	32K
SOURCE\$	5M (yes, 5 megabytes)
TRIGGER\$	300K
TRIGGERCOL\$	60K
VIEW\$	300K

The following Clusters should be adjusted:

Cluster	Initial Extent
C_COBJ#	350K

C_FILE#_BLOCK#	500K
C_OBJ\$	1700K
C_TS\$	180K

The following Indexes should be adjusted

<u>Index</u>	<u>Initial Extent</u>
I_ACCESS1	180K
I_ARGUMENT1	300K
I_CCOL1	57K
I_CDEF1	57K
I_CDEF2	98K
I_COBJ#	57K
I_COL1	475K
I_COL2	300K
I_COM1	57k
I_CON1	180K
I_CON2	100K
I_DEPENDENCY1	300K
I_DEPENDENCY2	180K
I_FILE#_BLOCK#	100K
I_ICOL1	57K
I_IDL_CHAR1	57K
I_IDL_SB41	180K
I_IDL_UB11	100K
I_IDL_UB21	100K
I_IND1	57K

I_OBJ#	100K
I_OBJ1	180K
I_OBJ2	300K
I_OBJAUTH1	180K
I_OBJAUTH2	180K
I_SOURCE1	1500K
I_SYN1	57K
I_TRIGGERCOL	100K
I_VIEW1	32K

Of course if you really want to make your data dictionary efficient you can also move the create index statements into a second script, drop the indexes from the SYSTEM tablespace and recreate them in their own tablespace (I suggest SYSTEM_INDEX) on a separate disk platter or disk array. The sizes stated above are only suggestions. If you have an existing database that is similar to the one you wish to create (i.e. approximately the same number and type of objects) then run the extents report included on the companion disk and adjust your sizes according to its output.

Let's look at a listing of the XX\$ tables and their purposes and descriptions.

1.1 ACCESS\$

The ACCESS\$ table shows objects and dependancies as well as a list of columns involved in the dependencies.

Contents of the ACCESS\$ table

Name	Null?	Type
-----	-----	-----
D_OBJ#	NOT NULL	NUMBER
ORDER#	NOT NULL	NUMBER
COLUMNS		RAW(126)
TYPES	NOT NULL	NUMBER

1.2 ARGUMENT\$

The ARGUMENT\$ table contains information on procedure arguments for the database.

Contents of the ARGUMENT\$ table

Name	Null?	Type
-----	-----	----
OBJ#	NOT NULL	NUMBER
PROCEDURE\$		VARCHAR2(30)
OVERLOAD#	NOT NULL	NUMBER
POSITION#	NOT NULL	NUMBER
SEQUENCE#	NOT NULL	NUMBER
LEVEL#	NOT NULL	NUMBER
ARGUMENT		VARCHAR2(30)
TYPE#	NOT NULL	NUMBER
CHARSETID		NUMBER
CHARSETFORM		NUMBER
DEFAULT#		NUMBER
IN_OUT		NUMBER
LENGTH		NUMBER
PRECISION#		NUMBER
SCALE		NUMBER
RADIX		NUMBER
DEFLLENGTH		NUMBER
DEFAULT\$		LONG

1.3 ATEMPTAB\$

Let's see, how to put this, this table has no discernable function but Oracle needs it now and then to feel secure so leave it alone. (This is an ORACLE8 foblie)

Contents of the ATEMPTAB\$

Name	Null?	Type
-----	-----	----
ID		NUMBER

1.4 ATTRCOL

The ATTRCOL\$ table is used to store data about user defined data types. This is an ORACLE8 only table.

Contents of the ATTRCOL\$

Name	Null?	Type
-----	-----	----
OBJ#	NOT NULL	NUMBER
INTCOL#	NOT NULL	NUMBER
NAME	NOT NULL	VARCHAR2(4000)

1.5 ATTRIBUTE\$

The ATTRIBUTE\$ table contains the detailed data about attributes. This is an ORACLE8 only table.

Contents of ATTRIBUTE\$ table

Name	Null?	Type
-----	-----	-----
TOID	NOT NULL	RAW(16)
VERSION#	NOT NULL	NUMBER
NAME	NOT NULL	VARCHAR2(30)
ATTRIBUTE#	NOT NULL	NUMBER
ATTR_TOID	NOT NULL	RAW(16)
ATTR_VERSION#	NOT NULL	NUMBER
PROPERTIES	NOT NULL	NUMBER
CHARSETID		NUMBER
CHARSETFORM		NUMBER
LENGTH		NUMBER
PRECISION#		NUMBER
SCALE		NUMBER
SPARE1		NUMBER
SPARE2		NUMBER
SPARE3		NUMBER

1.6 AUD\$

The AUD\$ table contains the database audit trail information.

Contents of the AUD\$ table

Name	Null?	Type
-----	-----	-----
SESSIONID	NOT NULL	NUMBER
ENTRYID	NOT NULL	NUMBER
STATEMENT	NOT NULL	NUMBER
TIMESTAMP#	NOT NULL	DATE
USERID		VARCHAR2(30)
USERHOST		VARCHAR2(2000)
TERMINAL		VARCHAR2(2000)
ACTION#	NOT NULL	NUMBER
RETURNCODE	NOT NULL	NUMBER
OBJ\$CREATOR		VARCHAR2(30)

OBJ\$NAME	VARCHAR2(128)
AUTH\$PRIVILEGES	VARCHAR2(16)
AUTH\$GRANTEE	VARCHAR2(30)
NEW\$OWNER	VARCHAR2(30)
NEW\$NAME	VARCHAR2(128)
SES\$ACTIONS	VARCHAR2(19)
SES\$TID	NUMBER
LOGOFF\$LREAD	NUMBER
LOGOFF\$PREAD	NUMBER
LOGOFF\$LWRITE	NUMBER
LOGOFF\$DEAD	NUMBER
LOGOFF\$TIME	DATE
COMMENT\$TEXT	VARCHAR2(4000)
SPARE1	VARCHAR2(255)
SPARE2	NUMBER
OBJ\$LABEL	RAW(255)
SES\$LABEL	RAW(255)
PRIV\$USED	NUMBER

1.7 AUDIT\$

The AUDIT\$ table is created when user level auditing is turned on. If you will be doing user level auditing this table will grow huge. I suggest that it be moved to its own tablespace area.

Contents of the AUDIT\$ table

Name	Null?	Type
-----	-----	----
USER#	NOT NULL	NUMBER
OPTION#	NOT NULL	NUMBER
SUCCESS		NUMBER
FAILURE		NUMBER

1.8 AUDIT_ACTIONS

The table AUDIT_ACTIONS contains the description of auditable actions.

Contents of the AUDIT_ACTIONS table

Name	Null?	Type
-----	-----	----
ACTION	NOT NULL	NUMBER
NAME	NOT NULL	VARCHAR2(27)

1.9 BOOTSTRAP\$

The BOOTSTRAP\$ table contains the initial statements needed to build the rest of database. As its name implies it provides the “bootstrap” by which the rest of the system pulls itself up by during the initial build. The BOOTSTRAP\$ table will usually contain about 45 entries that create basic system structures like the initial rollback segment in system and required base clusters for the data dictionary.

Contents of the BOOTSTRAP\$ table

Name	Null?	Type
-----	-----	-----
LINE#	NOT NULL	NUMBER
OBJ#	NOT NULL	NUMBER
SQL_TEXT	NOT NULL	VARCHAR2(4000)

1.10 CCOL\$

The CCOL\$ table contains the names of the constraint columns and links back to the CON\$, OBJ\$, COL\$ tables.

Contents of the CCOL\$ table

Name	Null?	Type
-----	-----	-----
CON#	NOT NULL	NUMBER
OBJ#	NOT NULL	NUMBER
COL#	NOT NULL	NUMBER
POS#		NUMBER
INTCOL#	NOT NULL	NUMBER
SPARE1		NUMBER
SPARE2		NUMBER
SPARE3		NUMBER
SPARE4		VARCHAR2(1000)
SPARE5		VARCHAR2(1000)
SPARE6		DATE

1.11 CDEF\$

The CDEF\$ table stores the constraint definitions for the database.

Contents of the CDEF\$ table

Name	Null?	Type
-----	-----	----
CON#	NOT NULL	NUMBER
OBJ#	NOT NULL	NUMBER
COLS		NUMBER
TYPE#	NOT NULL	NUMBER
ROBJ#		NUMBER
RCON#		NUMBER
RRULES		VARCHAR2(3)
MATCH#		NUMBER
REFACT		NUMBER
ENABLED		NUMBER
CONDLLENGTH		NUMBER
CONDITION		LONG
INTCOLS		NUMBER
MTIME		DATE
DEFER		NUMBER
SPARE1		NUMBER
SPARE2		NUMBER
SPARE3		NUMBER
SPARE4		VARCHAR2(1000)
SPARE5		VARCHAR2(1000)
SPARE6		DATE

1.12 CLU\$

The CLU\$ table is used to store cluster definitions.

Contents of the CLU\$ table

Name	Null?	Type
-----	-----	----
OBJ#	NOT NULL	NUMBER
DATAOBJ#		NUMBER
TS#	NOT NULL	NUMBER
FILE#	NOT NULL	NUMBER
BLOCK#	NOT NULL	NUMBER
COLS	NOT NULL	NUMBER
PCTFREE\$	NOT NULL	NUMBER
PCTUSED\$	NOT NULL	NUMBER
INITRANS	NOT NULL	NUMBER
MAXTRANS	NOT NULL	NUMBER
SIZE\$		NUMBER
HASHFUNC		VARCHAR2(30)
HASHKEYS		NUMBER

FUNC	NUMBER
EXTIND	NUMBER
FLAGS	NUMBER
DEGREE	NUMBER
INSTANCES	NUMBER
AVGCHN	NUMBER
SPARE1	NUMBER
SPARE2	NUMBER
SPARE3	NUMBER
SPARE4	NUMBER
SPARE5	VARCHAR2(1000)
SPARE6	VARCHAR2(1000)
SPARE7	DATE

1.13 COL\$

The COL\$ table contains informaiton on all columns defined in the database.

Contents of the COL\$ table

Name	Null?	Type
-----	-----	-----
OBJ#	NOT NULL	NUMBER
COL#	NOT NULL	NUMBER
SEGCOL#	NOT NULL	NUMBER
SEGCOLLENGTH	NOT NULL	NUMBER
OFFSET	NOT NULL	NUMBER
NAME	NOT NULL	VARCHAR2(30)
TYPE#	NOT NULL	NUMBER
LENGTH	NOT NULL	NUMBER
FIXEDSTORAGE	NOT NULL	NUMBER
PRECISION#		NUMBER
SCALE		NUMBER
NULL\$	NOT NULL	NUMBER
DEFLLENGTH		NUMBER
DEFAULT\$		LONG
INTCOL#	NOT NULL	NUMBER
PROPERTY	NOT NULL	NUMBER
CHARSETID		NUMBER
CHARSETFORM		NUMBER
SPARE1		NUMBER
SPARE2		NUMBER
SPARE3		NUMBER
SPARE4		VARCHAR2(1000)
SPARE5		VARCHAR2(1000)
SPARE6		DATE

1.14 COLLECTION\$

The COLLECTION\$ table contains the information on all “collections” (types) in the database.

This is an ORACLE8 only table.

Contents of the COLLECTION\$ table

Name	Null?	Type
-----	-----	-----
TOID	NOT NULL	RAW(16)
VERSION#	NOT NULL	NUMBER
COLL_TOID	NOT NULL	RAW(16)
COLL_VERSION#	NOT NULL	NUMBER
ELEM_TOID	NOT NULL	RAW(16)
ELEM_VERSION#	NOT NULL	NUMBER
PROPERTIES	NOT NULL	NUMBER
CHARSETID		NUMBER
CHARSETFORM		NUMBER
LENGTH		NUMBER
PRECISION		NUMBER
SCALE		NUMBER
UPPER_BOUND		NUMBER
SPARE1		NUMBER
SPARE2		NUMBER
SPARE3		NUMBER

1.15 COLTYPE\$

The COLTYPE\$ table contains data on columns that contain types rather than standard datatypes

(types are like VARRAYs, TYPE declarations, etc.) This is an ORACLE8 only table.

Contents of the COLTYPE\$ table

Name	Null?	Type
-----	-----	-----
OBJ#	NOT NULL	NUMBER
COL#	NOT NULL	NUMBER
INTCOL#	NOT NULL	NUMBER
TOID	NOT NULL	RAW(16)
VERSION#	NOT NULL	NUMBER
PACKED	NOT NULL	NUMBER
INTCOLS		NUMBER
INTCOL#S		RAW(2000)
FLAGS		NUMBER

1.16 COM\$

The COM\$ table contains the comments for all objects in the database. The

DBA_TAB_COMMENTS and DBA_COL_COMMENTS views are built off of this base table.

Contents of the COM\$ table

Name	Null?	Type
-----	-----	-----
OBJ#	NOT NULL	NUMBER
COL#		NUMBER
COMMENT\$		VARCHAR2(4000)

1.17 CON\$

The CON\$ table contains the information on all constraints in the database.

Contents of the CON\$ table

Name	Null?	Type
-----	-----	-----
OWNER#	NOT NULL	NUMBER
NAME	NOT NULL	VARCHAR2(30)
CON#	NOT NULL	NUMBER
SPARE1		NUMBER
SPARE2		NUMBER
SPARE3		NUMBER
SPARE4		VARCHAR2(1000)
SPARE5		VARCHAR2(1000)
SPARE6		DATE

1.18 DBMS_ALERT_INFO

This is one of the tables that doesn't follow the XX\$ format. This is because it is created not by migrate.bsq or sql.bsq but by DBMSALRT.SQL or PRVTALRT.SQL in support of the DBMS_ALERT package. This table stores the data on all user created alerts for the database.

Contents of the DBMS_ALERT_INFO table

Name	Null?	Type
-----	-----	-----
NAME	NOT NULL	VARCHAR2(30)
SID	NOT NULL	VARCHAR2(30)
CHANGED		VARCHAR2(1)
MESSAGE		VARCHAR2(1800)

1.19 DBMS_LOCK_ALLOCATED

Like the previous table DBMS_ALERT_INFO, this table also deviates from the XX\$ format. This table is created by the DBMSLOCK.SQL or PRVTLOCL.PLB script and is used to store information about user generated locks. This table may require periodic cleanout so the DBA should monitor it for growth if the DBMS_LOCK package is in use.

Contents of the DBMS_LOCK_ALLOCATED table

Name	Null?	Type
-----	-----	-----
NAME	NOT NULL	VARCHAR2(128)
LOCKID		NUMBER(38)
EXPIRATION		DATE

1.20 DEFROLE\$

The DEFROLE\$ table stores the default role value for each user in the database. The table acts as an intersection table between the USER\$ and ROLE\$ tables.

Contents of the DEFROLE\$ table

Name	Null?	Type
-----	-----	-----
USER#	NOT NULL	NUMBER
ROLE#	NOT NULL	NUMBER

1.21 DEPENDENCY\$

The DEPENDENCY\$ table provides the dependency tree information for the database objects.

Any objects beneath a deleted or changed object in the dependency tree are automatically invalidated. The utldtree.sql script can help with interpreting this table.

Contents of the DEPENDENCY\$ table

Name	Null?	Type
-----	-----	-----
D_OBJ#	NOT NULL	NUMBER
D_TIMESTAMP	NOT NULL	DATE
ORDER#	NOT NULL	NUMBER
P_OBJ#	NOT NULL	NUMBER
P_TIMESTAMP	NOT NULL	DATE
D_OWNER#	NOT NULL	NUMBER
PROPERTY	NOT NULL	NUMBER

1.22 DIR\$

The DIR\$ table holds directory definitions. The CREATE DIRECTORY command doesn't create a physical entity, it just adds an entry in this table. This is an ORACLE8 only table.

Contents of the DIR\$ table

Name	Null?	Type
-----	-----	-----
OBJ#	NOT NULL	NUMBER
AUDIT\$	NOT NULL	VARCHAR2(38)
OS_PATH		VARCHAR2(4000)

1.23 DUAL

The DUAL table has been around virtually forever (or at least since version 6). DUAL is just a place holder table. DUAL is the table you select from when there is no table to select from. For example, when you want the SYSDATE, USER, or CURVAL or NEXTVAL from a sequence you can select it from DUAL. One thing to watch for in DUAL is if some rouge process or user inserts a second (or more) value into the table. The DUAL table should be a single column, single

value table whose sole purpose is to provide a single return value to a select. DUAL must exist since many of the Oracle built in functions and procedures use DUAL.

Contents of the DUAL table

Name	Null?	Type
-----	-----	-----
DUMMY		VARCHAR2 (1)

1.24 DUC\$

The DUC\$ table contains information about packages, procedures and any comments entered about them in the database. In 7.2 and 7.3 it has three entries, one for DBMS_IJOB.DROP_USER_JOBS, one for DBMS_IREFRESH.DROP_USER_GROUPS and one for DBMS_ISNAPSHOT.DROP_USER_SNAPSHOTS. This seems to be used to stage procedures used during the dropping of a user since it includes a sequence number in which to run the procedures listed and all of the procedures deal with dropping part of a user's environment.

Contents of the DUC\$ table

Name	Null?	Type
-----	-----	-----
OWNER	NOT NULL	VARCHAR2 (30)
PACK	NOT NULL	VARCHAR2 (30)
PROC	NOT NULL	VARCHAR2 (30)
FIELD1		NUMBER
OPERATION#	NOT NULL	NUMBER
SEQ	NOT NULL	NUMBER
COM		VARCHAR2 (80)

1.25 ERROR\$

The ERROR\$ table will contain the errors associated with the object currently being created in any session. The objects covered are stored objects such as functions, procedures or packages. The data

in this table is not persistent and is periodically automatically cleaned out at Oracle internal intervals. The SHOW ERROR command automatically queries this table for the current object.

Contents of the ERROR\$ table

Name	Null?	Type
-----	-----	----
OBJ#	NOT NULL	NUMBER
SEQUENCE#	NOT NULL	NUMBER
LINE	NOT NULL	NUMBER
POSITION#	NOT NULL	NUMBER
TEXTLENGTH	NOT NULL	NUMBER
TEXT	NOT NULL	VARCHAR2(4000)

1.26 EXPACT\$

The EXPACT\$ table contains the pre- and post-export actions. These actions are covered in detail in the chapter 5 DBMS_EXPORT_ACTIONS section.

Contents of the EXPACT\$ table

Name	Null?	Type
-----	-----	----
OWNER	NOT NULL	VARCHAR2(30)
NAME	NOT NULL	VARCHAR2(30)
FUNC_SCHEMA	NOT NULL	VARCHAR2(30)
FUNC_PACKAGE	NOT NULL	VARCHAR2(30)
FUNC_PROC	NOT NULL	VARCHAR2(30)
CODE	NOT NULL	NUMBER
CALLORDER		NUMBER
CALLARG		VARCHAR2(1)
OBJ_TYPE	NOT NULL	NUMBER
USER_ARG		VARCHAR2(2000)

1.27 FET\$

The FET\$ table (the Free Extent Table) contains data about all of the extents in the database that are currently “free” (have no physical objects assigned to them).

Contents of the FET\$ table

Name	Null?	Type
------	-------	------

TS#	NOT NULL	NUMBER
FILE#	NOT NULL	NUMBER
BLOCK#	NOT NULL	NUMBER
LENGTH	NOT NULL	NUMBER

1.28 FILE\$

The FILE\$ table contains information on all database files.

Contents of the FILE\$ table

Name	Null?	Type
FILE#	NOT NULL	NUMBER
STATUS\$	NOT NULL	NUMBER
BLOCKS	NOT NULL	NUMBER
TS#		NUMBER
RELFIL#		NUMBER
MAXEXTEND		NUMBER
INC		NUMBER
CRSCNWRP		NUMBER
CRSCNBAS		NUMBER
OWNERINSTANCE		VARCHAR2 (30)
SPARE1		NUMBER
SPARE2		NUMBER
SPARE3		VARCHAR2 (1000)
SPARE4		DATE

1.29 HISTGRM\$

The table HISTGRM\$ contains data about all of the histograms specified for the database. This table contains the “data” used for the histogram calculations and displays. This table is in a many to one relation with the HIST_HEAD\$ table. These tables are present as of 7.3.

Contents of the HISTGRM\$ table

Name	Null?	Type
OBJ#	NOT NULL	NUMBER
COL#	NOT NULL	NUMBER
ROW#		NUMBER
BUCKET	NOT NULL	NUMBER
ENDPOINT	NOT NULL	NUMBER
INTCOL#	NOT NULL	NUMBER
SPARE1		NUMBER
SPARE2		NUMBER

1.30 HIST_HEAD\$

The HIST_HEAD\$ table describes the histograms defined for the database, it is the Histogram Header file. The HIST_HEAD\$ table has a one to many relation with the HISTGRM\$ table. These tables are present as of 7.3.

Contents of the HIST_HEAD\$ table

Name	Null?	Type
-----	-----	-----
OBJ#	NOT NULL	NUMBER
COL#	NOT NULL	NUMBER
BUCKET_CNT	NOT NULL	NUMBER
ROW_CNT	NOT NULL	NUMBER
CACHE_CNT		NUMBER
NULL_CNT		NUMBER
TIMESTAMP#		DATE
SAMPLE_SIZE		NUMBER
MINIMUM		NUMBER
MAXIMUM		NUMBER
DISTCNT		NUMBER
LOWVAL		RAW (32)
HIVAL		RAW (32)
DENSITY		NUMBER
INTCOL#	NOT NULL	NUMBER
SPARE1		NUMBER
SPARE2		NUMBER

1.31 ICOL\$

The ICOL\$ table contains information on all database index columns.

Contents of the ICOL\$ table

Name	Null?	Type
-----	-----	-----
OBJ#	NOT NULL	NUMBER
BO#	NOT NULL	NUMBER
COL#	NOT NULL	NUMBER
POS#	NOT NULL	NUMBER
SEGCOL#	NOT NULL	NUMBER
SEGCOLLENGTH	NOT NULL	NUMBER
OFFSET	NOT NULL	NUMBER
INTCOL#	NOT NULL	NUMBER

SPARE1	NUMBER
SPARE2	NUMBER
SPARE3	NUMBER
SPARE4	VARCHAR2(1000)
SPARE5	VARCHAR2(1000)
SPARE6	DATE

1.32 IDL_CHAR\$

The IDL_CHAR\$ table is used strictly by Oracle internals. It and its related tables: IDL_SB4\$, IDL_UB1\$ and IDL_UB2\$ all are Oracle use only and store code sections in various formats. Only IDL_CHAR\$ can be queried, the others will return either a “ORA-00932: inconsistent datatypes” or rudely tosses you out of the database with a “ORA-03113: end-of-file on communication channel” error. I don’t suggest attempting to even look at these tables since they contain no client usable data.

Contents of the IDL_CHAR\$ table

Name	Null?	Type
-----	-----	-----
OBJ#	NOT NULL	NUMBER
PART	NOT NULL	NUMBER
VERSION		NUMBER
PIECE#	NOT NULL	NUMBER
LENGTH	NOT NULL	NUMBER
PIECE	NOT NULL	LONG

Contents of the IDL_SB4\$ Table

Name	Null?	Type
-----	-----	-----
OBJ#	NOT NULL	NUMBER
PART	NOT NULL	NUMBER
VERSION		NUMBER
PIECE#	NOT NULL	NUMBER
LENGTH	NOT NULL	NUMBER
PIECE	NOT NULL	UNDEFINED

Contents of the IDL_UB1\$ Table

Name	Null?	Type
-----	-----	-----
OBJ#	NOT NULL	NUMBER
PART	NOT NULL	NUMBER
VERSION		NUMBER
PIECE#	NOT NULL	NUMBER
LENGTH	NOT NULL	NUMBER

PIECE NOT NULL LONG RAW

Contents of the IDL_UB2\$ table

Name	Null?	Type
OBJ#	NOT NULL	NUMBER
PART	NOT NULL	NUMBER
VERSION		NUMBER
PIECE#	NOT NULL	NUMBER
LENGTH	NOT NULL	NUMBER
PIECE	NOT NULL	UNDEFINED

1.33 ID_GEN\$

The ID_GEN\$ table contains a number that corresponds to the number of identification generators for the database. This is an ORACLE8 only table.

Contents of the ID_GEN\$ table

Name	Null?	Type
TOTAL	NOT NULL	NUMBER

1.34 INCEXP

INCEXP and its related tables INCFIL and INCVID are more of the tables that are exceptions to the XX\$ format. This table is used to support incremental exports. Useful information about the time of last cumulative (ctime) and the last incremental export (itime) are stored on a per item (name) and owner (owner#) basis. The object type is also stored as a numeric identifier:

0 -- Next Object
1 -- Index
2 -- Table (excludes index only or nested tables)

3 -- Cluster
 4 -- View
 5 -- Synonym
 6 -- Sequence
 Other -- Undefined

Contents of INCEXP table

Name	Null?	Type
OWNER#	NOT NULL	NUMBER
NAME	NOT NULL	VARCHAR2(30)
TYPE#	NOT NULL	NUMBER
CTIME		DATE
ITIME	NOT NULL	DATE
EXPID	NOT NULL	NUMBER(3)

1.35 INCFIL

The INCFIL table contains the file name(s) for the various incremental exports that have active tables in the database. The INCFIL table relates to the INCEXP table through the EXPID field.

Contents of the INCFIL table

Name	Null?	Type
EXPID	NOT NULL	NUMBER(3)
EXPTYPE	NOT NULL	VARCHAR2(1)
EXPFILE	NOT NULL	VARCHAR2(100)
EXPDATE	NOT NULL	DATE
EXPUSER	NOT NULL	VARCHAR2(30)

1.36 INCVID

The INCVID table contains the identifier(s) of the last valid incremental export(s).

Contents of the INCVID table

Name	Null?	Type
-----	-----	-----
EXPID	NOT NULL	NUMBER (3)

1.37 IND\$

The IND\$ table contains data on all database indexes. The table has been reformatted for ORACLE8. The type# field tells what type of index the entry is and has the following possible values as of 8.0.2:

- 1 -- Normal
- 2 -- Bitmap (O8 only)
- 3 -- Cluster (O8 only)
- 4 -- Index Only Table (Top level - O8 only)
- 5 -- Index Only Table (Nested - O8 only)
- 6 -- Secondary (O8 only)
- 7 -- ANSI (O8 only)
- 8 -- LOB (O8 only)
- 9 -- Cooperative Index Method (O8 only - future)

The flags field contains a status indicator (replaces unique\$) in hex format:

- 0x0001 -- unusable
- 0x0002 -- analyzed
- 0x0004 -- nologging

The property field contains non-changeable attribute data in hex format:

- 0x01 -- Unique Index
- 0x02 -- Partitioned Index (O8 only)

0x04 -- Reverse Key Index (O8 only)

0x08 -- Compressed Index

0x10 -- Functional Index

Contents of the IND\$ table

Name	Null?	Type
-----	-----	-----
OBJ#	NOT NULL	NUMBER
DATAOBJ#		NUMBER
TS#	NOT NULL	NUMBER
FILE#	NOT NULL	NUMBER
BLOCK#	NOT NULL	NUMBER
BO#	NOT NULL	NUMBER
INDMETHOD#	NOT NULL	NUMBER
COLS	NOT NULL	NUMBER
PCTFREE\$	NOT NULL	NUMBER
INITTRANS	NOT NULL	NUMBER
MAXTRANS	NOT NULL	NUMBER
PCTTHRES\$		NUMBER
TYPE#	NOT NULL	NUMBER
FLAGS	NOT NULL	NUMBER
PROPERTY	NOT NULL	NUMBER
BLEVEL		NUMBER
LEAFCNT		NUMBER
DISTKEY		NUMBER
LBLKKEY		NUMBER
DBLKKEY		NUMBER
CLUFAC		NUMBER
ANALYZETIME		DATE
SAMPLESIZE		NUMBER
ROWCNT		NUMBER
INTCOLS	NOT NULL	NUMBER
DEGREE		NUMBER
INSTANCES		NUMBER
TRUNC CNT		NUMBER
SPARE1		NUMBER
SPARE2		NUMBER
SPARE3		NUMBER
SPARE4		VARCHAR2(1000)
SPARE5		VARCHAR2(1000)
SPARE6		DATE

1.38 INDPART\$

The INDPART\$ table contains information about partitioned indexes for the database. This is an ORACLE8 only table.

Content of the INDPART\$ table

Name	Null?	Type
OBJ#	NOT NULL	NUMBER
DATAOBJ#	NOT NULL	NUMBER
BO#	NOT NULL	NUMBER
PART#	NOT NULL	NUMBER
HIBOUNDLEN	NOT NULL	NUMBER
HIBOUNDVAL		LONG
FLAGS	NOT NULL	NUMBER
TS#	NOT NULL	NUMBER
FILE#	NOT NULL	NUMBER
BLOCK#	NOT NULL	NUMBER
PCTFREE\$	NOT NULL	NUMBER
PCTTHRES\$		NUMBER
INITTRANS	NOT NULL	NUMBER
MAXTRANS	NOT NULL	NUMBER
ANALYZETIME		DATE
SAMPLESIZE		NUMBER
ROWCNT		NUMBER
BLEVEL		NUMBER
LEAFCNT		NUMBER
DISTKEY		NUMBER
LBLKKEY		NUMBER
DBLKKEY		NUMBER
CLUFAC		NUMBER
SPARE1		NUMBER
SPARE2		NUMBER
SPARE3		NUMBER
INCLCOL		NUMBER

1.39 JOB\$

The JOB\$ table contains data about all jobs in the database.

Contents of the JOB\$ table

Name	Null?	Type
JOB	NOT NULL	NUMBER
LOWNER	NOT NULL	VARCHAR2(30)
POWNER	NOT NULL	VARCHAR2(30)
COWNER	NOT NULL	VARCHAR2(30)
LAST_DATE		DATE
THIS_DATE		DATE
NEXT_DATE	NOT NULL	DATE
TOTAL	NOT NULL	NUMBER
INTERVAL#	NOT NULL	VARCHAR2(200)
FAILURES		NUMBER
FLAG	NOT NULL	NUMBER

WHAT	VARCHAR2(4000)
NLSENV	VARCHAR2(4000)
ENV	RAW(32)
CUR_SES_LABEL	RAW MLSLABEL
CLEARANCE_HI	RAW MLSLABEL
CLEARANCE_LO	RAW MLSLABEL
CHARENV	VARCHAR2(4000)
FIELD1	NUMBER

1.40 KOPM\$

The KOPM\$ table contains the database metadata entries. This is an ORACLE8 only table.

Contents of the KOPM\$ table

Name	Null?	Type
NAME	NOT NULL	VARCHAR2(30)
LENGTH	NOT NULL	NUMBER
METADATA		RAW(255)

1.41 LAB\$

The LAB\$ table contains information on labels used by secure Oracle for security tracking.

Contents of the LAB\$ table

Name	Null?	Type
LAB#	NOT NULL	RAW MLSLABEL
OLAB		RAW(255)
ALIAS#		VARCHAR2(30)

1.42 LIBRARY\$

The LIBRARY\$ table contains the entries for all libraries specified for external procedure and function calls. These entries are the only physical implementations of the library in the database, the actual library is a sharable system library external to the database. This is an ORACLE8 table.

Contents of the LIBRARY\$ table

Name	Null?	Type
-----	-----	-----
OBJ#	NOT NULL	NUMBER
FILESPEC		VARCHAR2(2000)
PROPERTY		NUMBER
AUDIT\$	NOT NULL	VARCHAR2(38)

1.43 LINK\$

The LINK\$ table contains information about all of the remote database links in the database. For Trusted Oracle the table TRUSTED_LIST\$ gives all of the usernames that can use privileged database links.

Contents of the LINK\$ table

Name	Null?	Type
-----	-----	-----
OWNER#	NOT NULL	NUMBER
NAME	NOT NULL	VARCHAR2(128)
CTIME	NOT NULL	DATE
HOST		VARCHAR2(2000)
USERID		VARCHAR2(30)
PASSWORD		VARCHAR2(30)
FLAG		NUMBER
AUTHUSR		VARCHAR2(30)
AUTHPWD		VARCHAR2(30)

1.44 LOB\$

The LOB\$ table contains information on all LOB (Large Objects) in the database. This is an ORACLE8 only table. The flags field contains hex format values that tell if various LOB attributes are set:

0x0000 -- CACHE

0x0001 -- NOCACHE LOGGING

0x0002 -- NOCACHE NOLOGGING

The property field is defined as:

0x00 -- User defined LOB column

0x01 -- Kernel column(s) stored as a LOB

0x02 -- User LOB column with row data

Contents of the LOB\$ table

Name	Null?	Type
-----	-----	-----
OBJ#	NOT NULL	NUMBER
COL#	NOT NULL	NUMBER
INTCOL#	NOT NULL	NUMBER
LOBJ#	NOT NULL	NUMBER
PART#	NOT NULL	NUMBER
IND#	NOT NULL	NUMBER
TS#	NOT NULL	NUMBER
FILE#	NOT NULL	NUMBER
BLOCK#	NOT NULL	NUMBER
CHUNK	NOT NULL	NUMBER
PCTVERSION\$	NOT NULL	NUMBER
FLAGS	NOT NULL	NUMBER
PROPERTY	NOT NULL	NUMBER
SPARE1		NUMBER
SPARE2		NUMBER
SPARE3		VARCHAR2 (255)

1.45 METHOD\$

The METHOD\$ table is the method signature table and contains the properties for each method stored against objects in the database. This is an ORACLE8 only table. The properties field has the following possible values:

- 1 - PRIVATE method
- 2 - PUBLIC method (default)
- 4 - INLINE method

8	-	VIRTUAL method
16	-	CONSTANT method
32	-	Constructor Method
64	-	Destructor Method
128	-	Operator Method
256	-	Selfish Method
512	-	MAP Method
2048	-	ORDER Method
4096	-	Read No Data State Method (default)
8192	-	Write No Data State Method
16384	-	Read No Process State Method
32768	-	Write No Process State Method

Contents of the METHOD\$ table

Name	Null?	Type
TOID	NOT NULL	RAW(16)
VERSION#	NOT NULL	NUMBER
METHOD#	NOT NULL	NUMBER
NAME	NOT NULL	VARCHAR2(30)
PROPERTIES	NOT NULL	NUMBER
PARAMETERS#	NOT NULL	NUMBER
RESULTS	NOT NULL	NUMBER
SPARE1		NUMBER
SPARE2		NUMBER
SPARE3		NUMBER

1.46 MIGRATE\$

The MIGRATE\$ table tracks all migrated tables in an ORACLE8 database. If the MIG80 utility wasn't used to migrate from a version 7 to version 8 database then this table is not used.

Contents of the MIGRATE\$ table

Name	Null?	Type
------	-------	------

-----	-----	----
VERSION#		VARCHAR2(30)
MIGDATE		DATE
MIGRATED	NOT NULL	NUMBER

1.47 MLOG\$

The MLOG\$ table is used to track local master tables used by snapshots and also tracks snapshot log related information that pertains to that master table.

Contents of the MLOG\$ table

Name	Null?	Type
-----	-----	----
MOWNER	NOT NULL	VARCHAR2(30)
MASTER	NOT NULL	VARCHAR2(30)
OLDEST		DATE
OLDEST_PK		DATE
OSCN		NUMBER
YOUNGEST		DATE
YSCN		NUMBER
LOG	NOT NULL	VARCHAR2(30)
TRIG		VARCHAR2(30)
FLAG		NUMBER
MTIME	NOT NULL	DATE

1.48 MLOG_REFCOL\$

The MLOG_REFCOL\$ table tracks the columns used as filter columns for snapshot logs. This is an ORACLE8 only table.

Contents of the MLOG_REFCOL\$ table

Name	Null?	Type
-----	-----	----
MOWNER	NOT NULL	VARCHAR2(30)
MASTER	NOT NULL	VARCHAR2(30)
COLNAME	NOT NULL	VARCHAR2(30)
OLDEST		DATE
FLAG		NUMBER(38)

1.49 NOEXP\$

The NOEXP\$ table contains the name of the tables that will not be exported by the export process.

This is an ORACLE8 only table.

Contents of the NOEXP\$

Name	Null?	Type
OWNER	NOT NULL	VARCHAR2(30)
NAME	NOT NULL	VARCHAR2(30)
OBJ_TYPE	NOT NULL	NUMBER

1.50 NTAB\$

The NTAB\$ table contains information on all Nested Tables in the database. This is an ORACLE8 only table.

Contents of the NTAB\$ table

Name	Null?	Type
OBJ#	NOT NULL	NUMBER
COL#	NOT NULL	NUMBER
INTCOL#	NOT NULL	NUMBER
NTAB#	NOT NULL	NUMBER
NAME	NOT NULL	VARCHAR2(4000)

1.51 OBJ\$

The OBJ\$ table contains information about database objects. Information on the following types of database objects is stored here:

- 1 - Index
- 2 - Table
- 3 - Cluster
- 4 - View

5	-	Synonym
6	-	Sequence
7	-	Procedure
8	-	Function
9	-	Package
10	-	Non-existent
11	-	Package Body
12	-	Trigger
13	-	Type
14	-	Type Body
19	-	Table Partition
20	-	Index Partition
21	-	LOB
22	-	LIBRARY
23	-	DIRECTORY

The objects are considered part of a namespace. The namespace values are:

1	-	Table/Procedure/Type
2	-	Body
3	-	Trigger
4	-	Index
5	-	Cluster
8	-	LOB
9	-	Directory

Each object can have one of the following status codes:

- 1 - Valid or authorized with no errors
- 2 - Valid or authorized with authorization errors
- 3 - Valid or authorized with compilation errors
- 4 - Valid but unauthorized
- 5 - Invalid and unauthorized

The objects can have one of the following flags set:

- 0x01 - Extent map checking required
- 0x02 - Temporary object
- 0x04 - System generated object

Contents of the OBJ\$ table

Name	Null?	Type
-----	-----	----
OBJ#	NOT NULL	NUMBER
DATAOBJ#		NUMBER
OWNER#	NOT NULL	NUMBER
NAME	NOT NULL	VARCHAR2(30)
NAMESPACE	NOT NULL	NUMBER
SUBNAME		VARCHAR2(30)
TYPE#	NOT NULL	NUMBER
CTIME	NOT NULL	DATE
MTIME	NOT NULL	DATE
STIME	NOT NULL	DATE
STATUS	NOT NULL	NUMBER
REMOTEOWNER		VARCHAR2(30)
LINKNAME		VARCHAR2(128)
FLAGS		NUMBER
OID\$		RAW(16)
SPARE1		NUMBER
SPARE2		NUMBER
SPARE3		NUMBER
SPARE4		VARCHAR2(1000)
SPARE5		VARCHAR2(1000)
SPARE6		DATE

1.52 OBJAUTH\$

The OBJAUTH\$ table contains data on table and column level authorizations for the database. If col# is null the entry is null the entry is a table grant, otherwise it will contain the column number to tie to the col\$ table.

Contents of the OBJAUTH\$ table

Name	Null?	Type
-----	-----	----
OBJ#	NOT NULL	NUMBER
GRANTOR#	NOT NULL	NUMBER
GRANTEE#	NOT NULL	NUMBER
PRIVILEGE#	NOT NULL	NUMBER
SEQUENCE#	NOT NULL	NUMBER
PARENT		ROWID
OPTION\$		NUMBER
COL#		NUMBER

1.53 OBJPRIV\$

The OBJPRIV\$ table supposedly contains privileges granted to objects, however on all systems I have checked the table is empty. I am aware of no command to grant privileges to objects other than to roles which are handled via the USER\$ and SYSAUTH\$ tables.

Contents of the OBJPRIV\$ table

Name	Null?	Type
-----	-----	----
OBJ#	NOT NULL	NUMBER
PRIVILEGE#	NOT NULL	NUMBER

1.54 OID\$

The OID\$ table contains all of the Object Identifiers for objects in the database. Object identifiers apply to object tables, views and types. This is an ORACLE8 only table.

Contents of the OID\$ table

Name	Null?	Type
-----	-----	-----
USER#	NOT NULL	NUMBER
OID\$	NOT NULL	RAW(16)
OBJ#	NOT NULL	NUMBER

1.55 PARAMETER\$

The PARAMETER\$ table contains the parameters for methods. This is an ORACLE8 only table.

The properties column has the following possible values:

256	-	IN parameter
512	-	OUT parameter
1024	-	Pass by reference parameter
2048	-	Required parameter (default)
16384	-	POINTER parameter
37768	-	REF parameter

The CHARSETFORM column has the following possible values:

1	-	Implicit for CHAR, VARCHAR2, CLOB (w/o a specifier set)
2	-	national character for NCHAR, NCHAR VARRYING, NLOB
3	-	explicit for CHAR, etc. with CHARACTER SET clause specified
4	-	Flexible for PL/SQL “flexible” parameters

Contents of the PARAMETER\$ table

Name	Null?	Type
-----	-----	-----
TOID	NOT NULL	RAW(16)
VERSION#	NOT NULL	NUMBER
METHOD#	NOT NULL	NUMBER
NAME	NOT NULL	VARCHAR2(30)
PARAMETER#	NOT NULL	NUMBER

PARAM_TOID	NOT NULL	RAW(16)
PARAM_VERSION#	NOT NULL	NUMBER
PROPERTIES	NOT NULL	NUMBER
CHARSETID		NUMBER
CHARSETFORM		NUMBER
DEFAULT\$		VARCHAR2(4000)
SPARE1		NUMBER
SPARE2		NUMBER
SPARE3		NUMBER

1.56 PARTCOL\$

The PARTCOL\$ table has one row per partitioning key column for a partitioned table. This is an ORACLE8 only table.

Contents of the PARTCOL\$ table

Name	Null?	Type
-----	-----	----
OBJ#	NOT NULL	NUMBER
INTCOL#	NOT NULL	NUMBER
COL#	NOT NULL	NUMBER
POS#	NOT NULL	NUMBER
SPARE1		NUMBER

1.57 PARTOBJ\$

The PARTOBJ\$ table has one row per partitioned table or index. This table has a one-to-many relationship to the PARTCOL\$ table. This is an ORACLE8 only table. The flags column has the following possible values:

0	-	Table
1	-	local index
2	-	prefixed index

The deflogging parameter has the following possible values:

- 0 - Unspecified
- 1 - logging
- 2 - nologging

Contents of the PARTOBJ\$ table

Name	Null?	Type
-----	-----	----
OBJ#	NOT NULL	NUMBER
PARTTYPE	NOT NULL	NUMBER
PARTCNT	NOT NULL	NUMBER
PARTKEYCOLS	NOT NULL	NUMBER
FLAGS		NUMBER
DEFTS#	NOT NULL	NUMBER
DEFPCTFREE	NOT NULL	NUMBER
DEFPCTUSED	NOT NULL	NUMBER
DEFPCTTHRES		NUMBER
DEFINITRANS	NOT NULL	NUMBER
DEFMAXTRANS	NOT NULL	NUMBER
DEFTINIEXTS	NOT NULL	NUMBER
DEFEXTSIZE	NOT NULL	NUMBER
DEFMINEXTS	NOT NULL	NUMBER
DEFMAXEXTS	NOT NULL	NUMBER
DEFEXTPCT	NOT NULL	NUMBER
DEFLISTS	NOT NULL	NUMBER
DEFGROUPS	NOT NULL	NUMBER
DEFLOGGING	NOT NULL	NUMBER
SPARE1		NUMBER
SPARE2		NUMBER
SPARE3		NUMBER
DEFINCLCOL		NUMBER

1.58 PENDING_SESSIONS\$

The PENDING_SESSIONS\$ table is the child table of the PENDING_TRANS\$ table. Both of these tables are used to track distributed transactions. This table has a child table called PENDING_SUB_SESSIONS\$. All of these tables help track the status of transaction branching for distributed two phase commit transactions.

Contents of the PENDING_SESSIONS\$ table

Name	Null?	Type
------	-------	------

LOCAL_TRAN_ID	NOT NULL	VARCHAR2(22)
SESSION_ID	NOT NULL	NUMBER(38)
BRANCH_ID	NOT NULL	RAW(64)
INTERFACE	NOT NULL	VARCHAR2(1)
TYPE#		NUMBER
PARENT_DBID		VARCHAR2(16)
PARENT_DB		VARCHAR2(128)
DB_USERID	NOT NULL	NUMBER(38)

Contents of the PENDING_SUB_SESSIONS\$ table

Name	Null?	Type
LOCAL_TRAN_ID	NOT NULL	VARCHAR2(22)
SESSION_ID	NOT NULL	NUMBER(38)
SUB_SESSION_ID	NOT NULL	NUMBER(38)
INTERFACE	NOT NULL	VARCHAR2(1)
DBID	NOT NULL	VARCHAR2(16)
LINK_OWNER	NOT NULL	NUMBER(38)
DBLINK	NOT NULL	VARCHAR2(128)
BRANCH_ID		RAW(64)
SPARE		RAW(64)

The PENDING_TRANS\$ table is the parent to the PENDING_SESSIONS\$ and its dependent table PENDING_SUB_SESSIONS\$. The PENDING_TRANS\$ table keeps track of pending or indoubt two-phase-commit distributed transactions.

Contents of the PENDING_TRANS\$ table

Name	Null?	Type
LOCAL_TRAN_ID	NOT NULL	VARCHAR2(22)
GLOBAL_TRAN_FMT	NOT NULL	NUMBER(38)
GLOBAL_ORACLE_ID		VARCHAR2(64)
GLOBAL_FOREIGN_ID		RAW(64)
TRAN_COMMENT		VARCHAR2(2000)
STATE	NOT NULL	VARCHAR2(16)
STATUS	NOT NULL	VARCHAR2(1)
HEURISTIC_DFLT		VARCHAR2(1)
SESSION_VECTOR	NOT NULL	RAW(4)
RECO_VECTOR	NOT NULL	RAW(4)
TYPE#		NUMBER
FAIL_TIME	NOT NULL	DATE
HEURISTIC_TIME		DATE
RECO_TIME	NOT NULL	DATE
TOP_DB_USER		VARCHAR2(30)
TOP_OS_USER		VARCHAR2(2000)
TOP_OS_HOST		VARCHAR2(2000)
TOP_OS_TERMINAL		VARCHAR2(2000)
GLOBAL_COMMIT#		VARCHAR2(16)
SPARE1		NUMBER
SPARE2		VARCHAR2(30)

SPARE3	NUMBER
SPARE4	VARCHAR2 (30)

1.59 PROCEDURE\$

The PROCEDURE\$ table stores basic information such as auditing options and size for each stored procedure in the database.

Contents of the PROCEDURE\$ table

Name	Null?	Type
-----	-----	-----
OBJ#	NOT NULL	NUMBER
AUDIT\$	NOT NULL	VARCHAR2 (38)
STORAGESIZE		NUMBER
OPTIONS		NUMBER

1.60 PROFILE\$

The PROFILE\$ table stores the resource type and resource limit settings for all of the profiles in the database.

Contents of the PROFILE\$ table

Name	Null?	Type
-----	-----	-----
PROFILE#	NOT NULL	NUMBER
RESOURCE#	NOT NULL	NUMBER
TYPE#	NOT NULL	NUMBER
LIMIT#	NOT NULL	NUMBER

1.61 PROFNAME\$

The PROFNAME\$ table contains the profile names for all of the profiles in the database. The PROFNAME\$ table has a one-to-many relationship with the PROFILE\$ table.

Contents of the PROFNAME\$ table

Name	Null?	Type
-----	-----	-----
PROFILE#	NOT NULL	NUMBER
NAME	NOT NULL	VARCHAR2(30)

1.62 PROPS\$

The PROPS\$ table contains database property entries. Properties such as NLS parameters, export table versions and the global database name are stored in this table. When selecting out of it be sure to set arraysize to one.

Contents of the PROPS\$ table

Name	Null?	Type
-----	-----	-----
NAME	NOT NULL	VARCHAR2(30)
VALUE\$		VARCHAR2(4000)
COMMENT\$		VARCHAR2(4000)

1.63 PSTUBTBL

The PSTUBTBL is create by the diutl.sql routine which is called by the catproc.sql routine during normal database builds. The table is used to generate stubs for procedures and functions as well as packages using the diana set of subroutines. You will never use this table directly.

Contents of the PSTUBTBL table

Name	Null?	Type
-----	-----	-----
USERNAME		VARCHAR2(30)
DBNAME		VARCHAR2(128)
LUN		VARCHAR2(30)
LUTYPE		VARCHAR2(3)
LINENO		NUMBER
LINE		VARCHAR2(1800)

1.64 REFCON\$

The REFCON\$ table stores information about the constraints used to support REFing between children and parent tables in ORACLE8. The REFTYPE column has two possible values:

0x01 = REF is scoped

0x02 = REF is stored with ROWID

If the REF is scoped then STABID will contain the OID to the REF scope table.

Contents of the REFCON\$ table

Name	Null?	Type
-----	-----	----
OBJ#	NOT NULL	NUMBER
COL#	NOT NULL	NUMBER
INTCOL#	NOT NULL	NUMBER
REFTYP	NOT NULL	NUMBER
STABID		RAW(16)

1.65 REG_SNAP\$

The REG_SNAP\$ (registered snapshots) table contains the data on snapshots that have master tables on this instance. It is the inverse of the MLOG\$ table that contains information on tables that are masters on this instance.

Contents of the REG_SNAP\$ table

Name	Null?	Type
-----	-----	----
SOWNER	NOT NULL	VARCHAR2(30)
SNAPNAME	NOT NULL	VARCHAR2(30)
SNAPSITE	NOT NULL	VARCHAR2(128)
SNAPSHOT_ID		NUMBER(38)
FLAG		NUMBER
REP_TYPE		NUMBER
COMMENT\$		VARCHAR2(4000)
QUERY_TXT		LONG

1.66 RESOURCE_COST\$

The RESOURCE_COST\$ table contains the data entered by the DBA concerning the costs to place on profile resource to compute the composite limit. The RESOURCE# column maps into the following profile parameters:

1	-	sessions_per_user
2	-	cpu_per_session
3	-	not used
4	-	logical_reads_per_session
5	-	not used
6	-	not used
7	-	connect_time
8	-	private_sga
9	-	not used

All values in the table are initially set to zero (0).

Contents of the RESOURCE_COST\$ table

Name	Null?	Type
-----	-----	----
RESOURCE#	NOT NULL	NUMBER
COST	NOT NULL	NUMBER

1.67 RESOURCE_MAP

The RESOURCE_MAP table contains the mapping of resource number to name. The mapping is similar to that shown for the RESOURCE_COST\$ except non-costed values as well as password control values (for ORACLE8 only) are mapped.

RESOURCE#	Description	Type Code
0	composit_limit	0
1	sessions_per_user	0
2	cpu_per_session	0
3	cpu_per_call	0
4	logical_reads_per_session	0
5	logical_reads_per_call	0
6	idle_time	0
7	connect_time	0
8	private_sga	0
0	failed_login_attempts	1
1	password_life_time	1
2	password_reuse_time	1
3	password_reuse_max	1
4	password_verify_function	1
5	password_lock_time	1
6	password_grace_time	1

Contents of the RESOURCE_MAP table

Name	Null?	Type
-----	-----	----
RESOURCE#	NOT NULL	NUMBER
TYPE#	NOT NULL	NUMBER
NAME	NOT NULL	VARCHAR2(32)

1.68 RESULT\$

The RESULT\$ table is used to catalog the metadata on results returned from methods in ORACLE8. The properties column has the following possible values:

0x4000 (16384)- POINTER result

0x8000 (32768)- REF result

Columns from the properties column to the end of the table aren't currently used, but may be in future releases

Contents of the RESULT\$ table

Name	Null?	Type
-----	-----	----
TOID	NOT NULL	RAW(16)
VERSION#	NOT NULL	NUMBER
METHOD#	NOT NULL	NUMBER
RESULT#	NOT NULL	NUMBER
RESULT_TOID	NOT NULL	RAW(16)
RESULT_VERSION#	NOT NULL	NUMBER
PROPERTIES	NOT NULL	NUMBER
CHARSETID		NUMBER
CHARSETFORM		NUMBER
SPARE1		NUMBER
SPARE2		NUMBER
SPARE3		NUMBER

1.69 RGCHILD\$

The RGCHILD\$ table contains the data about refresh group children. Refresh groups are sets of snapshots that are refreshed as a group, the children are the individual snapshots.

Contents of the RGCHILD\$ table

Name	Null?	Type
-----	-----	----
OWNER	NOT NULL	VARCHAR2(30)
NAME	NOT NULL	VARCHAR2(30)
TYPE#		VARCHAR2(30)
FIELD1		NUMBER
REFGROUP		NUMBER

1.70 RGROUP\$

The RGROUP\$ table contains information about the refresh groups used for snapshot refresh. The flag field controls group behavior.

FLAG VALUE	BEHAVIOR
0x01	Destroy group when empty
0x02	Do not push queues
0x04	Refresh after errors

Contents of the RGROUP\$ table

Name	Null?	Type
REFGROUP		NUMBER
OWNER	NOT NULL	VARCHAR2(30)
NAME	NOT NULL	VARCHAR2(30)
FLAG		NUMBER
ROLLBACK_SEG		VARCHAR2(30)
FIELD1		NUMBER
JOB	NOT NULL	NUMBER

1.71 SEG\$

The SEG\$ table is used to store information about all “segments” in the database. Segments are of several basic types, these are delimited by the values allowed for the TYPE# field.

TYPE#	Meaning
1	UNDO (Rollback Segment)
2	SAVE UNDO (Used by tablespaces when placed in offline immediate)
3	Temporary
4	Cache
5	DATA

Contents of the SEG\$ table

Name	Null?	Type
-----	-----	----
FILE#	NOT NULL	NUMBER
BLOCK#	NOT NULL	NUMBER
TYPE#	NOT NULL	NUMBER
TS#	NOT NULL	NUMBER
BLOCKS	NOT NULL	NUMBER
EXTENTS	NOT NULL	NUMBER
INIEXTS	NOT NULL	NUMBER
MINEXTS	NOT NULL	NUMBER
MAXEXTS	NOT NULL	NUMBER
EXTSIZE	NOT NULL	NUMBER
EXTPCT	NOT NULL	NUMBER
USER#	NOT NULL	NUMBER
LISTS		NUMBER
GROUPS		NUMBER
BITMAPRANGES	NOT NULL	NUMBER
CACHEHINT	NOT NULL	NUMBER
SCANHINT	NOT NULL	NUMBER
HWMINCR	NOT NULL	NUMBER
SPARE1		NUMBER
SPARE2		NUMBER

1.72 SEQ\$

The SEQ\$ table contains information about database sequences. The “HIGHWATER” column corresponds to the LAST_NUMBER value from the DBA_SEQUENCES view and is the last value selected out of the sequence.

Contents of the SEQ\$ table

Name	Null?	Type
-----	-----	----
OBJ#	NOT NULL	NUMBER
INCREMENT\$	NOT NULL	NUMBER
MINVALUE		NUMBER
MAXVALUE		NUMBER
CYCLE#	NOT NULL	NUMBER
ORDER\$	NOT NULL	NUMBER
CACHE	NOT NULL	NUMBER
HIGHWATER	NOT NULL	NUMBER
AUDIT\$	NOT NULL	VARCHAR2(38)

1.73 SLOG\$

The SLOG\$ table contains information about snapshots on local masters

Contents of the SLOG\$ table

Name	Null?	Type
-----	-----	-----
MOWNER	NOT NULL	VARCHAR2(30)
MASTER	NOT NULL	VARCHAR2(30)
SNAPSHOT		DATE
SNAPID		NUMBER(38)
SSCN		NUMBER
SNAPTIME	NOT NULL	DATE
TSCN		NUMBER
USER#		NUMBER

1.74 SNAP\$

The SNAP\$ table is a listing of local snapshots for the database. The field MaSTER_VERSION I used to identify the Oracle version of the master site for this snapshot, 1 means V7.x and 2 V8.x.

The FLAG field has the following possible values:

- 0x0001 - Can use master log
- 0x0002 - Snapshot is updatable
- 0x0010 - ROWID snapshot
- 0x0020 - PRIMARY Key snapshot

Contents of the SNAP\$ table

Name	Null?	Type
-----	-----	-----
SOWNER	NOT NULL	VARCHAR2(30)
VNAME	NOT NULL	VARCHAR2(30)
TNAME	NOT NULL	VARCHAR2(30)
MVIEW		VARCHAR2(30)
MOWNER		VARCHAR2(30)
MASTER		VARCHAR2(30)
MLINK		VARCHAR2(128)
CAN_USE_LOG		VARCHAR2(1)
SNAPSHOT		DATE
SNAPID		NUMBER(38)
SSCN		NUMBER
SNAPTIME		DATE

TSCN		NUMBER
ERROR#		NUMBER
AUTO_FAST		VARCHAR2(1)
AUTO_FUN		VARCHAR2(200)
AUTO_DATE		DATE
REFGROUP		NUMBER
USTRG		VARCHAR2(30)
USLOG		VARCHAR2(30)
STATUS		NUMBER(38)
MASTER_VERSION		NUMBER(38)
TABLES		NUMBER(38)
FIELD1		NUMBER
FIELD2		VARCHAR2(30)
FLAG		NUMBER
QUERY_TXT		LONG
LOBMASKVEC		RAW(255)
MTIME	NOT NULL	DATE
MAS_ROLL_SEG		VARCHAR2(30)

1.75 SNAP_COLMAP\$

The SNAP_COLMAP\$ table contains snapshot column aliasing information.

Contents of the SNAP_COLMAP\$ table

Name	Null?	Type
-----	-----	----
SOWNER	NOT NULL	VARCHAR2(30)
VNAME	NOT NULL	VARCHAR2(30)
SNACOL	NOT NULL	VARCHAR2(30)
TABNUM	NOT NULL	NUMBER(38)
MASCOL		VARCHAR2(30)
MASPOS		NUMBER(38)
COLROLE		NUMBER

1.76 SNAP_REFOP\$

The SNAP_REFOP\$ contains data on fast refresh operations for snapshots in the database. These operations are different depending on whether an inner or outer table is being accessed. The field OPERATION# contains the data concerning the operation to be performed.

Operation#	Operation to be performed	Inner or Outer table
0	SELECT for delete phase	Outer

1	DELETE statement	Outer
2	SELECT for upd/ins phase	Outer
3	UPDATE statement	Outer
4	INSERT statement	Outer
5	DELETE local inserts (if upd)	Outer
0	SELECT for delete phase	Inner
1	DELETE statement	Inner
2	SELECT for insert phase	Inner
3	INSERT statement	Inner
6	Insert as select	Complete refresh operations

Contents of the SNAP_REFOP\$ table

Name	Null?	Type
-----	-----	----
SOWNER	NOT NULL	VARCHAR2(30)
VNAME	NOT NULL	VARCHAR2(30)
TABNUM	NOT NULL	NUMBER(38)
OPERATION#	NOT NULL	NUMBER(38)
COLS		NUMBER
FCMASKVEC		RAW(255)
EJMASKVEC		RAW(255)
SQL_TXT		LONG

1.77 SNAP_REFTIME\$

The table SNAP_REFTIME\$ contains the last refresh time for each snapshot in the database.

Contents of the SNAP_REFTIME\$ table

Name	Null?	Type
-----	-----	----
SOWNER	NOT NULL	VARCHAR2(30)
VNAME	NOT NULL	VARCHAR2(30)
TABLENUM	NOT NULL	NUMBER(38)
SNAPTIME		DATE
MOWNER		VARCHAR2(30)
MASTER		VARCHAR2(30)

1.78 SOURCE\$

The SOURCE\$ table contains the source code for all stored objects in the database (procedures, functions and packages. Triggers are stored in TRIGGER\$.

Contents of the SOURCE\$ table

Name	Null?	Type
-----	-----	-----
OBJ#	NOT NULL	NUMBER
LINE	NOT NULL	NUMBER
SOURCE		VARCHAR2(4000)

1.79 STMT_AUDIT_OPTION_MAP

The STMT_AUDIT_OPTION_MAP contains the mapping data from an auditing number to an audit name. The map table contains nearly 200 entries, too many to list here. I suggest looking in the sql.bsq file for your system if you want to see them all.

Contents of the STMT_AUDIT_OPTION_MAP table

Name	Null?	Type
-----	-----	-----
OPTION#	NOT NULL	NUMBER
NAME	NOT NULL	VARCHAR2(40)

1.80 SYN\$

The SYN\$ table contains the information on all synonyms in the database.

Contents of the SYN\$ table

Name	Null?	Type
-----	-----	-----
OBJ#	NOT NULL	NUMBER
NODE		VARCHAR2(128)

OWNER	VARCHAR2(30)
NAME	NOT NULL VARCHAR2(30)

1.81 SYSAUTH\$

The SYSAUTH\$ table contains the data on system level authorizations. The GRANTEE# column maps back in a many to one relationship to either the ROLES\$ or USERS\$ tables.

Contents of the SYSAUTH\$ table

Name	Null?	Type
-----	-----	----
GRANTEE#	NOT NULL	NUMBER
PRIVILEGE#	NOT NULL	NUMBER
SEQUENCE#	NOT NULL	NUMBER
OPTION\$		NUMBER

1.82 SYSTEM_PRIVILEGE_MAP

The SYSTEM_PRIVILEGE_MAP table maps a system privilege number to the privilege name. There are nearly 200 entries in this table, for a detailed listing query the table or look in sql.bsq on your system.

Contents of the SYSTEM_PRIVILEGE_MAP table

Name	Null?	Type
-----	-----	----
PRIVILEGE	NOT NULL	NUMBER
NAME	NOT NULL	VARCHAR2(40)

1.83 TAB\$

The TAB\$ table contains information on all tables in the database. The FLAG column contains table status information. The possible entries for FLAG are:

0x00	-	table unmodified since last backup
0x01	-	table modified since last backup
0x02	-	DML locks restricted to <= SX
0x04	-	DML locks <= SX not acquired
0x08	-	Table should be cached
0x10	-	table has been analyzed
0x20	-	table has no logging
0x40	-	7.x -> 8.0 data object, migration required

The column INTCOL tells the number of columns that have metadata associated with them. The column KERNELCOLS has the number of REAL columns (columns that actually contain data). The PROPERTY column defines the type(s) of table and has the following values:

Property	Meaning
0x01	Typed Table
0x02	Has ADT columns
0x04	Has Nested-table columns
0x08	Has REF columns
0x10	Has VAARRAY columns
0x20	partitioned table
0x40	Index only table (IOT)
0x80	IOT with row overflow
0x100	IOT with row clustering
0x200	IOT overflow segment
0x400	Clustered Table
0x800	Has internal LOB columns
0x1000	Has primary key based OID\$ column

0x2000	Is a nested-table
0x4000	Is a read only view
0x8000	Has FILE columns
0x10000	Object Views OID is system generated
0x20000	Used as an Advanced Queueing table
0x40000	Has user defined LOB columns.

Note that the value can be additive for tables with multiple possible values.

Contents of the TAB\$ table

Name	Null?	Type
OBJ#	NOT NULL	NUMBER
DATAOBJ#		NUMBER
TS#	NOT NULL	NUMBER
FILE#	NOT NULL	NUMBER
BLOCK#	NOT NULL	NUMBER
BOBJ#		NUMBER
TAB#		NUMBER
COLS	NOT NULL	NUMBER
CLUCOLS		NUMBER
PCTFREE\$	NOT NULL	NUMBER
PCTUSED\$	NOT NULL	NUMBER
INITTRANS	NOT NULL	NUMBER
MAXTRANS	NOT NULL	NUMBER
FLAGS	NOT NULL	NUMBER
AUDIT\$	NOT NULL	VARCHAR2(38)
ROWCNT		NUMBER
BLKCNT		NUMBER
EMPCNT		NUMBER
AVGSPC		NUMBER
CHNCNT		NUMBER
AVGRLN		NUMBER
AVGSPC_FLB		NUMBER
FLBCNT		NUMBER
ANALYZETIME		DATE
SAMPLESIZE		NUMBER
DEGREE		NUMBER
INSTANCES		NUMBER
INTCOLS	NOT NULL	NUMBER
KERNELCOLS	NOT NULL	NUMBER
PROPERTY	NOT NULL	NUMBER
TRIGFLAG		NUMBER
SPARE1		NUMBER
SPARE2		NUMBER
SPARE3		NUMBER
SPARE4		VARCHAR2(1000)
SPARE5		VARCHAR2(1000)
SPARE6		DATE

1.84 TABLE_PRIVILEGE_MAP

The TABLE_PRIVILEGE_MAP contains the mapping from a table privilege number (auditing option) to the privilege name. The tables contents are:

0	ALTER
1	AUDIT
2	COMMENT
3	DELETE
4	GRANT
5	INDEX
6	INSERT
7	LOCK
8	RENAME
9	SELECT
10	UPDATE
11	REFERENCES
12	EXECUTE
16	CREATE
17	READ
18	WRITE

Contents of the TABLE_PRIVILEGE_MAP table

Name	Null?	Type
-----	-----	----
PRIVILEGE	NOT NULL	NUMBER
NAME	NOT NULL	VARCHAR2(40)

1.85 TABPART\$

The TABPART\$ table contains information on table partitions (ORACLE8 only table). The table has one row per partition per table. The FLAGS columns has the following possible values:

- 0x01 7.3 to 8.0 data object migration (partitioned view type migration)
- 0x02 Partition has been analyzed
- 0x04 No logging on this partition

Contents of the TABPART\$ table

Name	Null?	Type
-----	-----	----
OBJ#	NOT NULL	NUMBER
DATAOBJ#	NOT NULL	NUMBER
BO#	NOT NULL	NUMBER
PART#	NOT NULL	NUMBER
HIBOUNDLEN	NOT NULL	NUMBER
HIBOUNDVAL		LONG
TS#	NOT NULL	NUMBER
FILE#	NOT NULL	NUMBER
BLOCK#	NOT NULL	NUMBER
PCTFREE\$	NOT NULL	NUMBER
PCTUSED\$	NOT NULL	NUMBER
INITTRANS	NOT NULL	NUMBER
MAXTRANS	NOT NULL	NUMBER
FLAGS	NOT NULL	NUMBER
ANALYZETIME		DATE
SAMPLESIZE		NUMBER
ROWCNT		NUMBER
BLKCNT		NUMBER
EMPCNT		NUMBER
AVGSPC		NUMBER
CHNCNT		NUMBER
AVGRLN		NUMBER
SPARE1		NUMBER
SPARE2		NUMBER
SPARE3		NUMBER

1.86 TRIGGER\$

The TRIGGER\$ table is to triggers what SOURCE\$ is to packages, procedures and functions. The TRIGGER\$ table contains the source code lines for each trigger in the database. The TYPE# column contains the following possible values:

- 0 before table
- 1 Before row
- 2 After table
- 3 After row
- 4 Instead of

The PROPERTY column has two possible values NULL means this trigger is for a table, 0x01 indicates this trigger is for a view (INSTEAD OF trigger only).

Contents of the TRIGGER\$ table

Name	Null?	Type
-----	-----	----
OBJ#	NOT NULL	NUMBER
TYPE#	NOT NULL	NUMBER
UPDATE\$	NOT NULL	NUMBER
INSERT\$	NOT NULL	NUMBER
DELETE\$	NOT NULL	NUMBER
BASEOBJECT	NOT NULL	NUMBER
REFOLDNAME		VARCHAR2(30)
REFNEWNAME		VARCHAR2(30)
DEFINITION		VARCHAR2(4000)
WHENCLAUSE		VARCHAR2(4000)
ACTION#		LONG
ACTIONSIZ		NUMBER
ENABLED		NUMBER
PROPERTY	NOT NULL	NUMBER

1.87 TRIGGERCOL\$

The TRIGGERCOL\$ table contains information on the OLD/NEW and column status for triggers.

The TYPE# column has the following possible values:

- 5 NEW in argument
- 6 OLD in argument
- 9 NEW out argument

Contents of the TRIGGERCOL\$ table

Name	Null?	Type
-----	-----	----
OBJ#	NOT NULL	NUMBER
COL#	NOT NULL	NUMBER
TYPE#	NOT NULL	NUMBER
POSITION#		NUMBER
INTCOL#	NOT NULL	NUMBER

1.88 TRUSTED_LIST\$

The TRUSTED_LIST\$ table provides a list of trusted users for the trusted database links in the database (Trusted Oracle only).

Contents of the TRUSTED_LIST\$ table

Name	Null?	Type
-----	-----	----
DBNAME	NOT NULL	VARCHAR2(128)
USERNAME	NOT NULL	VARCHAR2(4000)

1.89 TS\$

The TS\$ table contains data on all tablespaces used by the database. The ONLINE\$ column has three possible values: 1 - Online, 2 - Offline, 3 - Invalid. The DFLOGGING column has two possible values: 0 - Nologging, 1 - Logging. The FLAGS column is only used to indicate tablespace point in time recovery status.

Contents of the TS\$

Name	Null?	Type
-----	-----	----
TS#	NOT NULL	NUMBER
NAME	NOT NULL	VARCHAR2(30)
OWNER#	NOT NULL	NUMBER
ONLINE\$	NOT NULL	NUMBER
CONTENT\$	NOT NULL	NUMBER
UNDOFILE#		NUMBER

UNDOBLOCK#		NUMBER
BLOCKSIZE	NOT NULL	NUMBER
INC#	NOT NULL	NUMBER
SCNWRP		NUMBER
SCNBAS		NUMBER
DFLMINEXT	NOT NULL	NUMBER
DFLMAXEXT	NOT NULL	NUMBER
DFLINIT	NOT NULL	NUMBER
DFLINCR	NOT NULL	NUMBER
DFLMINLEN	NOT NULL	NUMBER
DFLEXTPCT	NOT NULL	NUMBER
DFLOGGING	NOT NULL	NUMBER
AFFSTRENGTH	NOT NULL	NUMBER
BITMAPPED	NOT NULL	NUMBER
PLUGGED	NOT NULL	NUMBER
DIRECTALLOWED	NOT NULL	NUMBER
FLAGS	NOT NULL	NUMBER
PITRSCNWRP		NUMBER
PITRSCNBAS		NUMBER
OWNERINSTANCE		VARCHAR2(30)
BACKUPOWNER		VARCHAR2(30)
GROUPNAME		VARCHAR2(30)
SPARE1		NUMBER
SPARE2		NUMBER
SPARE3		VARCHAR2(1000)
SPARE4		DATE

1.90 TSQ\$

The TSQ\$ table is used to hold data on tablespace quotas for database users. Essentially if you have an entry in this table you have been granted the ability to add objects to the indicated tablespace. If the value for MAXBLOCKS is set to a positive integer, that is the number of blocks (database) that the user can utilize. If the number is negative it indicates the user has unrestricted quota on the tablespace and if the value is zero indicates the user has unlimited tablespace and has placed an object in the tablespace.

Contents of the TSQ\$ table

Name	Null?	Type
-----	-----	----
TS#	NOT NULL	NUMBER
USER#	NOT NULL	NUMBER
GRANTOR#	NOT NULL	NUMBER
BLOCKS	NOT NULL	NUMBER
MAXBLOCKS		NUMBER
PRIV1	NOT NULL	NUMBER
PRIV2	NOT NULL	NUMBER

PRIV3

NOT NULL NUMBER

1.91 TYPE\$

The TYPE\$ table contains data on all TYPEs defined in the database. The PROPERTIES column tells details about the TYPE.

Properties Entry	Meaning
0x0001 (1)	Contains at least one embedded ADT attribute
0x0002 (2)	Contains at least one REF value
0x0004 (4)	Contains at least one complex attribute: multiset, table, array, LOB, file, etc.
0x0010 (16)	Predefined system or built-in type which is unalterable
0x0080 (128)	Contains at least one multiset attribute
0x0100 (256)	Is an incomplete type
0x0200 (512)	Contains at least one LOB attribute
0x0400 (1024)	Contains at least one file attribute (BFILE)
0x0800 (2048)	This is a system generated type

Contents of the TYPE\$ table

Name	Null?	Type
-----	-----	-----
TOID	NOT NULL	RAW(16)
VERSION#	NOT NULL	NUMBER
VERSION	NOT NULL	VARCHAR2(30)
TVOID	NOT NULL	RAW(16)
TYPECODE	NOT NULL	NUMBER
PROPERTIES	NOT NULL	NUMBER
ATTRIBUTES		NUMBER
METHODS		NUMBER
SUPERTYPES		NUMBER
SUBTYPES		NUMBER
SPARE1		NUMBER
SPARE2		NUMBER
SPARE3		NUMBER

1.92 TYPED_VIEW\$

The TYPED_VIEW\$ table contains additional data for a typed view. This is an ORACLE* only table.

Contents of the TYPED_VIEW\$ table

Name	Null?	Type
-----	-----	----
OBJ#	NOT NULL	NUMBER
TYPEOWNER		VARCHAR2(30)
TYPENAME		VARCHAR2(30)
TYPETEXTLENGTH		NUMBER
TYPETEXT		VARCHAR2(4000)
OIDTEXTLENGTH		NUMBER
OIDTEXT		VARCHAR2(4000)
TRANSTEXTLENGTH		NUMBER
TRANSTEXT		LONG

1.93 TYPE_MISC\$

The TYPE_MISC\$ table contains miscellaneous type data. The table tracks data about audit options and the PROPERTIES column has a single possible value (as of 8.0.2): 0x01 which indicates that the TYPE is a potential REF-dependency parent.

Contents of the TYPE_MISC\$ table

Name	Null?	Type
-----	-----	----
OBJ#	NOT NULL	NUMBER
AUDIT\$	NOT NULL	VARCHAR2(38)
PROPERTIES	NOT NULL	NUMBER

1.94 UET\$

The UET\$ table is the companion table to the FET\$ table. The FET\$ table contains free extent information, the UET\$ table contains used extent information.

Contents of the UET\$ table

Name	Null?	Type
-----	-----	-----
SEGFILE#	NOT NULL	NUMBER
SEGBLOCK#	NOT NULL	NUMBER
EXT#	NOT NULL	NUMBER
TS#	NOT NULL	NUMBER
FILE#	NOT NULL	NUMBER
BLOCK#	NOT NULL	NUMBER
LENGTH	NOT NULL	NUMBER

1.95 UGROUP\$

The UGROUP\$ table contains the information on undo (rollback segment) groups. The concept of rollback segment groups has not been implemented in 8.0.2 but may be available in future releases. I assume this is designed for use with parallel server and that a group of rollback segments will be assignable to a specific instance, however, this is just conjecture on my part.

Contents of the UGROUP\$ table

Name	Null?	Type
-----	-----	-----
UGRP#	NOT NULL	NUMBER
NAME	NOT NULL	VARCHAR2(30)
SEQ#		NUMBER
SPARE1		NUMBER
SPARE2		VARCHAR2(30)
SPARE3		NUMBER

1.96 UNDO\$

The UNDO\$ table contains data on rollback segments (called UNDO segments). The USER# column contains two possible values: 0 for a SYS or PRIVATE rollback segment or 1 for a private rollback segment. In ORACLE8 unless you are running parallel servedr these are treated identically. The INST# column contains the number of the parallel instance that either was assigned the rollback segment via a ROLLBACK_SEGMENTS assignment in the initialization file, or,

acquired the public rollback on startup or as it was needed. The SATUS\$ column has the following possible values:

Status\$	Meaning
1	Invalid
2	Available
3	In use
4	Offline
5	Needs recovery
6	Partly available (has some in doubt transactions)

The UGRP# doesn't seem to be used but will contain a reference to the UGROUP\$ table when rollback segment groups are implemented.

Contents of the UNDO\$ table

Name	Null?	Type
US#	NOT NULL	NUMBER
NAME	NOT NULL	VARCHAR2(30)
USER#	NOT NULL	NUMBER
FILE#	NOT NULL	NUMBER
BLOCK#	NOT NULL	NUMBER
SCNBAS		NUMBER
SCNWRP		NUMBER
XACTSQN		NUMBER
UNDOSQN		NUMBER
INST#		NUMBER
STATUS\$	NOT NULL	NUMBER
TS#		NUMBER
UGRP#		NUMBER
KEEP		NUMBER
OPTIMAL		NUMBER
FLAGS		NUMBER
SPARE1		NUMBER
SPARE2		NUMBER
SPARE3		NUMBER
SPARE4		VARCHAR2(1000)
SPARE5		VARCHAR2(1000)
SPARE6		DATE

1.97 USER\$

The USER\$ table contains data about all database users and roles. The TYPE# column has a value of 0 if the entry is a role and 1 if the entry is a regular user. The DEFROLE column has three possible values:

- 0 User has no roles
- 1 All roles granted
- 2 Users roles are in the DEFROLE\$ table.

The ASTATUS column is new with ORACLE8 and indicates the users automatically set status based on password management algorithms:

- 0 Open
- 1 Locked
- 2 Expired(Grace)
- 3 Locked and expired.
- 4 Locked(timed)
- 5 Expired and locked (timed)
- 6 Expired(Grace) and Locked (timed)
- 8 Locked
- 9 Expired and locked
- 10 Expired(grace) and locked

Contents of the USER\$ table

Name	Null?	Type
-----	-----	----
USER#	NOT NULL	NUMBER
NAME	NOT NULL	VARCHAR2 (30)
TYPE#	NOT NULL	NUMBER
PASSWORD		VARCHAR2 (30)
DATATS#	NOT NULL	NUMBER
TEMPTS#	NOT NULL	NUMBER

CTIME	NOT NULL	DATE
PTIME		DATE
EXPTIME		DATE
LTIME		DATE
RESOURCE\$	NOT NULL	NUMBER
AUDIT\$		VARCHAR2(38)
DEFROLE	NOT NULL	NUMBER
DEFGRP#		NUMBER
DEFGRP_SEQ#		NUMBER
ASTATUS	NOT NULL	NUMBER
LCOUNT	NOT NULL	NUMBER
DEFSCHCLASS		VARCHAR2(30)
EXT_USERNAME		VARCHAR2(4000)
SPARE1		NUMBER
SPARE2		NUMBER
SPARE3		NUMBER
SPARE4		VARCHAR2(1000)
SPARE5		VARCHAR2(1000)
SPARE6		DATE

1.98 USER_ASTATUS_MAP

The USER_ASTATUS_MAP contains the values that map into the ASTATUS in the USER\$ table. This is an ORACLE8 only table that is used with the password management facility. The values for this table are listed in section 1.85.

Contents of the USER_ASTATUS_MAP table

Name	Null?	Type
STATUS#	NOT NULL	NUMBER
STATUS	NOT NULL	VARCHAR2(32)

1.99 USER_HISTORY\$

The USER_HISTORY\$ table tracks user password usage for the password management utilities.

This is an ORACLE8 only table.

Contents of the USER_HISTORY\$

Name	Null?	Type
USER#	NOT NULL	NUMBER
PASSWORD		VARCHAR2(30)

PASSWORD_DATE

DATE

1.100 VIEW\$

The VIEW\$ table contains information on all views in the database. The table has been expanded under ORACLE8 to include all possible ORACLE8 view types. The table column PROPERTY contains the following possible values:

Property	Meaning
0x001(1)	This is a typed view
0x002(2)	This has ADT columns
0x004(4)	This has nested tables
0x008(8)	This has REF columns
0x010(16)	This has VARRAY columns
0x1000(4096)	This view has primary key-based OID
0x4000(16384)	This view is read-only.

The column FLAGS column has the following possible values:

0x1000(4096)	The view is insertable via trigger
0x2000(8192)	The view is updatable via trigger
0x4000(16384)	The view is deletable via trigger.

The table contains a length indicator (TEXTLENGTH) and the LONG column TEXT to store the entire view definition.

Contents of the VIEW\$ table

Name	Null?	Type
------	-------	------

-----	-----	-----
OBJ#	NOT NULL	NUMBER
AUDIT\$	NOT NULL	VARCHAR2 (38)
COLS	NOT NULL	NUMBER
INTCOLS	NOT NULL	NUMBER
PROPERTY	NOT NULL	NUMBER
FLAGS	NOT NULL	NUMBER
TEXTLENGTH		NUMBER
TEXT		LONG

1.101 PROCEDURE\$

The PROCEDURE\$ table contains the object number audit options, storage size and any compile options for all procedure in the database

Contents of the procedure\$ table:

Name	Null?	Type
-----	-----	-----
OBJ#	NOT NULL	NUMBER
AUDIT\$	NOT NULL	VARCHAR2 (38)
STORAGESIZE		NUMBER
OPTIONS		NUMBER

1.102 PENDING_SUB_SESSIONS\$

The PENDING_SUB_SEESIONSS\$ table is the child table to the PENDING_SESSIONS\$ table.

The interface column can have the values C (hold commit) or N (No hold). The local_tran_id ties back to session.

Contents of the pending_sub_sessions\$ table:

Name	Null?	Type
-----	-----	-----
LOCAL_TRAN_ID	NOT NULL	VARCHAR2 (22)
SESSION_ID	NOT NULL	NUMBER (38)
SUB_SESSION_ID	NOT NULL	NUMBER (38)
INTERFACE	NOT NULL	VARCHAR2 (1)
DBID	NOT NULL	VARCHAR2 (16)
LINK_OWNER	NOT NULL	NUMBER (38)
DBLINK	NOT NULL	VARCHAR2 (128)
BRANCH_ID		RAW (64)
SPARE		RAW (64)

1.103 EXT_TO_OBJ

The EXT_TO_OBJ table is built by the catparr.sql script. The table maps extents to objects. The v\$ping and v\$cache views use this table. The table is not dynamic so it must be periodically rebuilt. This table is based on a SELECT * FROM EXT_TO_OBJ_VIEW. The EXT_TO_OBJ_VIEW is also built by the catparr.sql script (the view doesn't have to be rebuilt, just the table). Therefore, before depending on any information from the v\$ping or v\$cache views, issue the following commands:

```
SQL> CONNECT SYS/SYS_PASSWORD
```

```
SQL> TRUNCATE TABLE EXT_TO_OBJ;
```

```
SQL> INSERT INTO EXT_TO_OBJ SELECT * FROM EXT_TO_OBJ_VIEW;
```

Contents of the ext_to_obj static table:

Name	Null?	Type
-----	-----	----
FILE#		NUMBER
LOWB		NUMBER
HIGHB		NUMBER
NAME		VARCHAR2 (30)
PARTITION_NAME		VARCHAR2 (30)
KIND		VARCHAR2 (15)
OWNER#		NUMBER