Part 14: Data Dictionaries

Elmasri/Navathe:

- Chapter 17, “Database System Architectures and the System Catalog”

I have also used the following sources:

- Date/Darwen: A Guide to the SQL Standard, Fourth Edition, Addison-Wesley, 1997.
- Sunderraman: Oracle Programming, A Primer. Addison-Wesley, 1999.
- Oracle8 Reference, Oracle Corporation, 1997, Part No. A58242-01.
- Don Chamberlin: A Complete Guide to DB2 Universal Database. Morgan Kaufmann, 1998.
- Michael Gertz: Oracle/SQL Tutorial, 1999. `http://www.db.cs.ucdavis.edu/teaching/sqltutorial/`
- Microsoft SQL Server Books Online: Accessing and Changing Data.

Objectives

- You should know what kind of information you can expect to find in a data dictionary.
- You should feel comfortable with the idea that meta-data (schema information) can be represented as data in the system catalog.

  This is in the beginning difficult for many students.
- You should know at least three tables/views from the Oracle data dictionary.
- You should know where to look for more information.
- You should be able to write SQL queries which refer to data dictionary tables/views.

  Given the possibility to look up table and column names.

Overview

1. General Remarks
2. Oracle Data Dictionary
3. DB2 Data Dictionary
4. SQL Server / SQL-92 Information Schema

Data Dictionaries (1)

- Database systems usually store the “meta data” (data about data, e.g. the database schema) in the database itself.

  All data Oracle has is available in system tables.
- This part of the database is called the “data dictionary” or “system catalog”.

  A “repository” is more general, it also contains data of the schema design process and for application development (usually used by CASE tools).
- In this way, queries to data and meta-data can be formalized in the same language.

  A general query language like SQL is much more powerful than a specialized set of commands like “`describe`”.

Data Dictionaries (2)

- Of course, many of the data dictionary tables are only available to users with DBA rights (database administrators).
- Also, the data dictionary tables are read-only to ensure their consistency.

  E.g. `INSERT` cannot be used on system tables. They can only be changed with commands like `CREATE TABLE`.
- The data dictionary is very system-dependent.

  The tables Oracle uses are completely different from those used in DB2 or SQL Server. The data dictionary even changed substantially between different versions of Oracle. But the SQL-92 standard proposes an information schema with some standard views (currently only in SQL Server).

Data Dictionaries (3)

- The “real” system tables have a rather unreadable format for performance reasons.

  In fact, a system can use any data structure for the system data, as long as it offers a relational interface to these data. It does not necessarily have to be the same data structure as used for normal user tables.
- Oracle has defined views to give a more user-friendly interface.
- There are also graphical tools for browsing the Oracle data dictionary. Try the free tool ODDIS!

  `http://www-db.informatik.uni-hannover.de/software/oddis/`
- The Oracle Enterprise Manager also contains a graphical user interface for some data dictionary information.

Data Dictionaries (4)

Information in the Data Dictionary:

- Accessible tables, views, etc.
- Users of the database system.
- Access rights of the users.
- Information about file space usage.
- Comments about tables and columns.
- A log of executed commands (if “auditing” is switched on).

When you understand all information in the data dictionary, you understand the system. The system has no other information. All parameters are reflected in the data dictionary.

Overview

1. General Remarks
2. Oracle Data Dictionary
3. DB2 Data Dictionary
4. SQL Server / SQL-92 Information Schema

Oracle Data Dictionary (1)

List of Data Dictionary Tables/Views:

- DICT(TABLE_NAME, COMMENTS)
  has an entry for every table (or view) of the data dictionary.
  Currently, it has 613 rows, at least for a DBA.
- DICT_COLUMNS(TABLE_NAME, COLUMN_NAME, COMMENTS)
  contains information about the single columns.

If you want to see some complicated SQL queries:

$ORACLE_HOME/rdbms/admin/catalog.sql

This file contains the definitions of the data dictionary views.

Oracle Data Dictionary (2)

Naming Conventions:

- USER_*: Objects for which the current user is owner.
  There are shorter synonyms declared for the most important of these tables, e.g. CAT for USER_CATALOG.
- ALL_*: Objects accessible by the current user.
- DBA_*: All objects of the DB, only accessible by a DBA.
  Depending on the installation, it might be necessary to write, e.g. "SYS.DBA_OBJECTS".
- V$*: Dynamic performance tables (only for DBA).
  These "tables" give a relational interface to data structures of the server. They are not actually stored tables.

DB Schema: Tables and Columns (1)

Catalog (Table Names):

- List of all tables and table-like objects owned by you:
  CAT(TABLE_NAME, TABLE_TYPE).
  The type can be TABLE, VIEW, SYNONYM, SEQUENCE.
  CAT is actually a synonym for USER_CATALOG.
- All accessible table-like objects:
  ALL_CATALOG(OWNER, TABLE_NAME, TABLE_TYPE)
- E.g. which tables of "SBRASS" are accessible?
  SELECT * FROM ALL_CATALOG WHERE OWNER = 'SBRASS'
- CAT and ALL_CATALOG are views and show every user a different result (tables owned or accessible by this user).

DB Schema: Tables and Columns (2)

DB Objects (includes Tables):

- All own database objects (tables, indexes, etc.):

```
OBJ(OBJECT_NAME, SUBOBJECT_NAME, OBJECT_ID,
    DATA_OBJECT_ID, OBJECT_TYPE, CREATED,
    LAST_DDL_TIME, TIMESTAMP, STATUS, TEMPORARY,
    GENERATED).
```

- OBJECT_TYPE can be, e.g.,:
  CLUSTER, FUNCTION, INDEX, LIBRARY, PACKAGE, PACKAGE BODY, PROCEDURE, SEQUENCE, SYNONYM, TABLE, TRIGGER, TYPE, UNDEFINED, VIEW.
- OBJ is a synonym for USER_OBJECTS.

DB Schema: Tables and Columns (3)

- CREATED is the date on which the object was created, LAST_DDL_TIME is the date of the last schema modification.
  - LAST_DDL_TIME is set when the access rights are changed.
- GENERATED means that the object name is system-generated.
- TEMPORARY means that no multi-user synchronization and no backup/recovery has to be done for this object.
- E.g. when was the table "EMP" created?

```
SELECT CREATED FROM OBJ WHERE OBJECT_NAME='EMP'
```

- Data about all accessible DB objects:

```
ALL_OBJECTS(OWNER, OBJECT_NAME, ...).
```


DB Schema: Tables and Columns (4)

Table Columns:

- Information about columns of own tables:

```
COLS(TABLE_NAME, COLUMN_NAME, DATA_TYPE, ...,
     DATA_LENGTH, DATA_PRECISION, DATA_SCALE,
     NULLABLE, COLUMN_ID, ...)
```

- COLS is a synonym for USER_TAB_COLUMNS.
- TABLE_NAME and COLUMN_NAME identify the column.
- COLUMN_ID: Position (1,2,...) of the column in the table.
- DATA_TYPE: E.g., CHAR or NUMBER.
- DATA_PRECISION, DATA_SCALE: Size of numeric types.

DB Schema: Tables and Columns (5)

- CHAR_COL_DECL_LENGTH: Declared length for string types.
- DATA_LENGTH: Maximum length of the column in bytes.
- NULLABLE: "N" if "NOT NULL", "Y" otherwise.
- In columns not shown, the table contains statistical information used by the optimizer (after "ANALYZE TABLE").
- E.g., list all columns of the table "PRESIDENT":

```
SELECT COLUMN_ID, COLUMN_NAME
FROM   COLS
WHERE  TABLE_NAME = 'PRESIDENT'
ORDER  BY COLUMN_ID
```


Constraints (1)

- List of all constraints on tables owned by current user:

```
USER_CONSTRAINTS(OWNER, CONSTRAINT_NAME,
  CONSTRAINT_TYPE, TABLE_NAME, SEARCH_CONDITION,
  R_OWNER, R_CONSTRAINT_NAME, DELETE_RULE, ...)
```

- CONSTRAINT_TYPE is:
  - C for a check constraint
  - P for primary key
  - U for unique constraint (alternate key)
  - R for a foreign key (referential integrity).
  - V for "with check option" clause in a view definition.

Constraints (2)

- `R_OWNER` is the owner of the table used in a referential constraint, `R_CONSTRAINT_NAME` is the name of the key constraint for the referenced table. `DELETE_RULE` can be `CASCADE` or `NO ACTION`.
- `SEARCH_CONDITION` is the text of the `CHECK`-condition.
  `NOT NULL` constraints have "`A IS NOT NULL`".
- E.g.: Print all `CHECK` constraints on the table "`COURSE`":

```
SELECT SEARCH_CONDITION
FROM   USER_CONSTRAINTS
WHERE  TABLE_NAME = 'COURSE'
AND    CONSTRAINT_TYPE = 'C'
```


Constraints (3)

- Columns in constraint definitions (e.g. keys):

```
USER_CONS_COLUMNS(OWNER, CONSTRAINT_NAME,
        TABLE_NAME, COLUMN_NAME, POSITION)
```

- `OWNER` and `CONSTRAINT_NAME` identify the constraint.
  Every database user has in Oracle his/her own schema (i.e. namespace for identifiers). Thus, a DB-wide identification always consists of user and name.
- `TABLE_NAME` is the same as in the corresponding entry in the constraint table.
- `POSITION` can e.g. help to print a primary key consisting of more than one attribute in the sequence it was defined.

Constraints (4)

- E.g. print the primary key of "`COURSE`":

```
SELECT COLUMN_NAME
FROM   USER_CONSTRAINTS con,
       USER_CONS_COLUMNS col
WHERE  con.TABLE_NAME = 'COURSE'
AND    CONSTRAINT_TYPE = 'P'
AND    con.OWNER = col.OWNER
AND    con.CONSTRAINT_NAME = col.CONSTRAINT_NAME
```

- Exercise: Print referencing table and column and referenced table for all foreign key constraints.
  You can assume that they consist only of one attribute.

Storage Information (1)

- Information about base tables (i.e. not views):

```
TABS(TABLE_NAME, TABLESPACE_NAME, ...,
     NUM_ROWS, BLOCKS, EMPTY_BLOCKS, AVG_SPACE,
     CHAIN_CNT, AVG_ROW_LEN, ...)
```

- A tablespace is something like a "logical disk" consisting of one or more database files.
  See also: `USER_TS_QUOTAS(TABLESPACE_NAME, ...)`.
- `TABS` contains the storage parameters for every base table.
  These columns are not shown above. They correspond to parameters of the `CREATE TABLE` command which we have not yet discussed (and which are Oracle-specific).

Storage Information (2)

- In addition, TABS contains information about the table size (number of rows etc.), to be used by the optimizer.
  But this information is not kept current for performance reasons: Every transaction would need to lock these data.
- You need to execute the following Oracle SQL command:
  ANALYZE TABLE ⟨Table⟩ COMPUTE STATISTICS
- NUM_ROWS is the number of rows in the table.
- BLOCKS is the number of used data blocks in the table.
  db_block_size is set for our database to 2048 bytes.
- AVG_ROW_LEN: Average length of a row in bytes.

Comments

- Comments about all accessible tables:
  ALL_TAB_COMMENTS(OWNER, TABLE_NAME, TABLE_TYPE, COMMENTS)
  Of course, there is a corresponding table “USER_TAB_COMMENTS”.
- Comments about columns of accessible tables:
  ALL_COL_COMMENTS(OWNER, TABLE_NAME, COLUMN_NAME, COMMENTS)
- You can define comments with the commands (not SQL-92):
  COMMENT ON TABLE ⟨Table⟩ IS '⟨Text⟩'
  COMMENT ON COLUMN ⟨Table⟩.⟨Column⟩ IS '⟨Text⟩'

Views in the Data Dictionary

- Views are represented in CAT (with TABLE_TYPE='VIEW').
  Of course, they are also contained in OBJ.
- The defining query of a view is stored in
  USER_VIEWS(VIEW_NAME, TEXT_LENGTH, TEXT, ...)
- Dependencies of procedures and views on tables etc.:
  USER_DEPENDENCIES(NAME, TYPE, REFERENCED_OWNER, REFERENCED_NAME, REFERENCED_TYPE, ...).
- Which columns of join views are updatable:
  USER_UPDATABLE_COLUMNS(OWNER, TABLE_NAME, COLUMN_NAME, UPDATABLE, INSERTABLE, DELETABLE)

Users (1)

- List of all DB users (visible to all users):
  ALL_USERS(USERNAME, USER_ID, CREATED)
- Information for DBA (with encrypted password):
  DBA_USERS(USERNAME, USER_ID, PASSWORD, DEFAULT_TABLESPACE, TEMPORARY_TABLESPACE, CREATED, PROFILE, ACCOUNT_STATUS, LOCK_DATE, EXPIRY_DATE, EXTERNAL_NAME)
- Some information stored about yourself:
  USER_USERS(USERNAME, USER_ID, ACCOUNT_STATUS, LOCK_DATE, EXPIRY_DATE, DEFAULT_TABLESPACE, CREATED, EXTERNAL_NAME)

Users (2)

- How many bytes/blocks on which tablespace are charged to which user, and what is the allowable maximum (quota):

```
DBA_TS_QUOTAS(TABLESPACE_NAME, USERNAME,
              BYTES, MAX_BYTES,
              BLOCKS, MAX_BLOCKS)
```

- Your current and maximal file space usage: USER_TS_QUOTAS.
- All table data is charged to the table owner (even if other users actually inserted the rows).
- Information from Profile: USER_PASSWORD_LIMITS, USER_RESOURCE_LIMITS.

Access Rights (1)

- Information about grants on objects for which the current user is owner, grantor, or grantee:

```
USER_TAB_PRIVS(GRANTEE, OWNER, TABLE_NAME,
               GRANTOR, PRIVILEGE, GRANTABLE).
```

- OWNER and TABLE_NAME identify the table.
- GRANTEE is the user who got the privilege.
- GRANTOR is the user who gave the privilege.
  Not only the owner can be grantor (grant option).
- PRIVILEGE is the right, e.g. 'SELECT'.
- GRANTABLE is 'YES' if the right was given with grant option.

Access Rights (2)

- Information about grants on tables owned by the current user:

```
USER_TAB_PRIVS_MADE(GRANTEE, TABLE_NAME,
                    GRANTOR, PRIVILEGE,
                    GRANTABLE).
```

- Information about grants for which the current user is grantee:

```
USER_TAB_PRIVS_RECD(OWNER, TABLE_NAME,
                    GRANTOR, PRIVILEGE,
                    GRANTABLE).
```


Access Rights (3)

- Grants which refer to single table columns are stored in another table:

```
USER_COL_PRIVS(GRANTEE, OWNER, TABLE_NAME,
               COLUMN_NAME, GRANTOR, PRIVILEGE,
               GRANTABLE).
```

  There are also USER_COL_PRIVS_MADE, USER_COL_PRIVS_RECD.
- Synonyms created by the current user:

```
USER_SYNONYMS(SYNONYM_NAME, TABLE_OWNER,
              TABLE_NAME, DB_LINK)
```

  USER_SYNONYMS may be abbreviated to SYN. ALL_SYNONYMS lists all accessible synonyms. PUBLICSYN lists all public synonyms.

System Privileges

- Which user has which system privileges (only for DBA):

  `DBA_SYS_PRIVS(GRANTEE, PRIVILEGE, ADMIN_OPTION)`
- System privileges granted to the user or to `PUBLIC`:

  `USER_SYS_PRIVS(GRANTEE, PRIVILEGE, ADMIN_OPTION)`
- Only directly granted privileges are listed. You may have more system privileges granted via roles (see below).
- List of all existing system privileges (code and name):

  `SYSTEM_PRIVILEGE_MAP(PRIVILEGE, NAME)`

Roles (1)

- List of all roles defined in the system:

  `DBA_ROLES(ROLE, PASSWORD_REQUIRED)`
- Which roles are granted to which user (or which other role)?

```
DBA_ROLE_PRIVS(GRANTEE, GRANTED_ROLE,
               ADMIN_OPTION, DEFAULT_ROLE)
```
- Roles granted to the current user (or to `PUBLIC`):

```
USER_ROLE_PRIVS(USERNAME, GRANTED_ROLE,
                ADMIN_OPTION, DEFAULT_ROLE,
                OS_GRANTED)
```


Roles (2)

- Role-to-role grants (only roles of the current user are shown):

```
ROLE_ROLE_PRIVS(ROLE, GRANTED_ROLE,
                ADMIN_OPTION)
```
- System privileges granted to roles (accessible by user):

```
ROLE_SYS_PRIVS(ROLE, GRANTED_ROLE,
               ADMIN_OPTION)
```
- Table privileges granted to roles (accessible by user):

```
ROLE_TAB_PRIVS(ROLE, OWNER, TABLE_NAME,
               COLUMN_NAME, PRIVILEGE,
               GRANTABLE)
```


Overview

1. General Remarks
2. Oracle Data Dictionary
3. DB2 Data Dictionary
4. SQL Server / SQL-92 Information Schema

General Remarks

- An older/internal version of the data dictionary is stored in the tables of the schema SYSIBM.
- Views for the user are made available in the schema SYSCAT.

  It consists of 38 views (i.e. much smaller than Oracle).
- In addition, there are views containing statistical information about table sizes etc. in the schema SYSSTAT.

  It consists of 5 views.
- The statistical information is updated by running the RUNSTATS utility.

  However, in order to influence the optimizer in specific ways, the views in SYSSTAT are actually updatable.

Schemas

- Information about schemas is made available in the view SYSCAT.SCHEMATA.
- It has the following columns:
  - SCHEMANAME: Name of the schema.
  - OWNER: Name of the user who owns the schema.
  - DEFINER: Name of the user who created the schema.
  - CREATE_TIME: Timestamp indicating when the schema was created.
  - REMARKS: Text stored with the COMMENT ON SCHEMA command.
- The underlying base table is SYSIBM.SYSSCHEMATA.

Tables

- Information about tables is available in SYSCAT.TABLES.
- This view has e.g. the following columns:
  - TABSCHEMA: Schema in which the table is defined.
  - TABNAME: Name of the table.
  - DEFINER: User who created the table.
  - TYPE: 'T' for tables, 'V' for views, 'A' for aliases.
  - CREATE_TIME: Timestamp when the table was created.
  - COLCOUNT: Number of columns.
  - REMARKS: Explanation, stored with COMMENT command.

Columns

- Information about columns of tables and views is available in SYSCAT.COLUMNS. This table has e.g. the following columns:
  - TABSCHEMA/TABNAME: Schema and name of the table.
  - COLNAME: Name of the column.
  - COLNO: Position of the column, the first has COLNO=0.
  - TYPESCHEMA/TYPENAME: Name of the data type.
  - LENGTH: Maximum length of the column.
  - SCALE: Number of digits after decimal point.
  - DEFAULT: Default value for this column.
  - NULLS: 'Y' if coumn allows null values, 'N' otherwise.
  - REMARKS: Text stored with COMMENT ON COLUMN command.

Access Rights (Table Privileges)

- Information about granted privileges for tables and views is available in SYSCAT.TABAUTH It has the following columns:
  - GRANTOR: User who granted the privilege.
  - GRANTEE: User who received the privilege.
  - GRANTEETYPE: 'U' if GRANTEE is user, 'G' if group.
  - TABSCHEMA/TABNAME: Schema and name of the table/view to which the privilege applies.
  - CONTROLAUTH, ALTERAUTH, DELETEAUTH, INDEXAUTH, INSERTAUTH, SELECTAUTH, REFAUTH, UPDATEAUTH: 'Y' if privilege was granted without grant option, 'G' if granted with grant option, 'N' if not granted.

Overview

General Remarks

- SQL Server has a “native” data dictionary and implements part of the SQL-92 “information schema” standard.
- The native data dictionary consists of two parts:
  - The “System Catalog” which is stored in the database “master” and contains settings for the server.
  - Each database managed by the server contains a “Database Catalog” with information for that DB only.
- The names of these system tables/views start with “sys”. They are owned by “dbo”, so no prefix is needed (except DB).
- The views corresponding to the SQL-92 standard are owned by the user “INFORMATION_SCHEMA”.

  SQL-92 has 17 such tables, the standard mentions 24.

Tables

- Information about tables and views in the current database for which the current user has permissions are available in INFORMATION_SCHEMA.TABLES.
- It has the following columns:
  - TABLE_CATALOG: Name of the database.
  - TABLE_SCHEMA: Owner of the table.
  - TABLE_NAME: Name of the table.
  - TABLE_TYPE: 'VIEW' or 'BASE TABLE'.
- This view is based on the table SYSOBJECTS from the database catalog.

Columns (1)

- INFORMATION_SCHEMA.COLUMNS contains information about columns accessible by the current user in the current DB.
- It has e.g. the following columns (23 in total):
  - TABLE_CATALOG/TABLE_SCHEMA/TABLE_NAME: Database, owner and name of the table.
  - COLUMN_NAME: Name of the column.
  - ORDINAL_POSITION: Column position.
    E.g. 1 for first/leftmost column.
  - COLUMN_DEFAULT: Default value.
  - IS_NULLABLE: 'YES' if null values are allowed.
    Otherwise 'No' is printed (the manual says 'NO').

Columns (2)

- Columns of INFORMATION_SCHEMA.COLUMNS, continued:
  - DATA_TYPE: Data type name.
  - CHARACTER_MAXIMUM_LENGTH: Max. number of characters.
    CHARACTER_OCTET_LENGTH: maximum length in bytes.
  - NUMERIC_PRECISION: Maximal number of stored digits.
  - NUMERIC_PRECISION_RADIX: E.g. 10 if decimal digits.
  - NUMERIC_SCALE: Number of digits after decimal point.
- This view is based on the tables SYSCOLUMNS and SYSTYPES from the database catalog (plus some other tables).

Table Privileges

- INFORMATION_SCHEMA.TABLE_PRIVILEGES contains information about access rights granted to or by the current user in the current database. It has the following columns:
  - GRANTOR: User who granted the privilege.
  - GRANTEE: User who received the privilege.
  - TABLE_CATALOG/TABLE_SCHEMA/TABLE_NAME:
    Table to which the access right applies.
  - PRIVILEGE_TYPE: E.g. 'SELECT', 'INSERT'.
  - IS_GRANTABLE: 'YES' if with grant option, 'NO' otherwise.
- This is based on the following database catalog tables: SYSPROTECTS, SYSOBJECTS, SYSUSERS.