Introduction to the Oracle DBMS

Kristian Torp

Department of Computer Science
Aalborg University
www.cs.aau.dk/~torp
torp@cs.aau.dk

December 2, 2011

daisy.aau.dk

Center for Data-intensive Systems
Learning Goals

Prerequisites
- Knowledge about basic database concepts, e.g., table and index
- Basic knowledge of SQL

Learning Goals
- Use the Oracle DBMS
- Know where Oracle different from standard or other DBMSs
- Know how to find database metadata
Outline

1. Users
2. The Data Dictionary
3. General Stuff
For Testing Only!

- Examples in the following are only for test machines!
- Should not be used in a production environment!

For Production

- Contact your local DBA
- Read the manual on how to make a secure environment
Supplied Users

- SYS, the “root” user of the Oracle DBMS
- SYSTEM, has database administration rights

Note

- It is a good idea to create a separate user for your database
- How to create users is different on different DBMSs
- There are nice GUI tools for creating users
New DBA User

User Requirements

- Create a new super user on the Oracle DBMS
- The new user must not be one of the supplied users.

Example (Create a Super User)

```sql
-- login as system (predefined super user)
sqlplus system@<system identifier>

-- Create a new super user, make nicer password
create user dbadm identified by dbadm;

-- grant rights
grant dba to dbadm;

-- logout
exit;

-- login as new dba
sqlplus dbadm/dbadm@<system identifier>

-- Now change your password
```
New User

User Requirements

- Create a new user on the Oracle DBMS no administration rights
- Should be able to create tables and query them

Example (Create a Super User)

```sql
-- login as system
sqlplus dbadm@<system identifier>
-- Create a new user
create user dbuser identified by dbuser;
-- grant rights to new user
grant connect, resource to dbuser;
-- connect as new user
connect dbuser/dbuser
-- Now change your password
```
Example (Drop New Super User and Plain Users)

```sql
-- login as system
sqlplus system@<system_identifier>
-- drop the users created above
drop user dbuser;
```

Note

- All schema object owned by the dropped users are also dropped!
- You cannot drop the user currently used
Outline

1. Users
2. The Data Dictionary
3. General Stuff
Overview of Data Dictionary

The idea in the naming

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>all_[rest of name]</td>
<td>All the stuff that the current user can see</td>
</tr>
<tr>
<td>dba_[rest of name]</td>
<td>All the stuff in the database</td>
</tr>
<tr>
<td>user_[rest of name]</td>
<td>All the stuff that the current user owns</td>
</tr>
</tbody>
</table>

\( \text{user} \subseteq \text{all} \subseteq \text{dba} \)

Access Rights

It is possible that you do not have access to the 'dba' views.

Example (Info. on Indexes)

```sql
select *
from all_indexes
```

Example (Info. on Views)

```sql
select *
from user_views
```
What Stuff is in the DBMS

Example (Objects users have in the database)

```sql
select ao.owner, ao.object_type, count(*) as cnt
from all_objects ao
group by ao.owner, ao.object_type
order by cnt desc
```

Note

- The `all_objects` lists all objects stored in the database
- The `object_type` is, e.g., tables, indexes, java classes, and synonyms

Example (Object Types)

```sql
select distinct object_type
from all_objects
```

Result

<table>
<thead>
<tr>
<th>OBJECT_TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLE</td>
</tr>
<tr>
<td>INDEX</td>
</tr>
<tr>
<td>SYNONYM</td>
</tr>
<tr>
<td>..</td>
</tr>
</tbody>
</table>
Outline

1. Users
2. The Data Dictionary
3. General Stuff
The Unix `ls` command

Example (Show me the stuff)

```sql
select * from cat;
```

Note

- cat is short for catalog
- A view with two columns `table_name` and `table_type`
- Contains tables, views, sequences (but not itself!)
Limiting the Number of Rows Returned

Example (Give me the first ten rows)

```sql
select * 
from cat 
where rownum <= 10
```

Note

- The `rownum` pseudo column is Oracle specific on other DBMSs it is called `limit`
Outline

4 Tables

5 Columns

6 Constraints

7 Views
Available Tables

Example (Look at all my tables)
```
select *  
from user_tables;
```

Example (Look at all tables available to me)
```
select *  
from all_tables;
```

Example (Find tables that are non-local)
```
select *  
from all_tables  
except  
select *  
from user_tables;
```

Note
- Not union compatible!
The dual Table

Example (Find the current time)

```sql
select current_timestamp
from dual
```

Note

- A utility table in Oracle because the `from` clause is required
- Has one column `dummy`
- Has one row (with the value X)
- Uses the built-in function `current_timestamp`
  - Cultural format, e.g., 2012-02-28 12:34:56 vs. 02-28-2012 12:34:56
- Uses the dual table
Example (How much space are my tables taking up?)

```
select us.segment_name as "table name",
       bytes/1024 as "KB"
from user_segments us
where us.segment_type = 'TABLE'
order by KB desc;
```

Note

- Note the word 'TABLE' has to be in upper case
Example (How much space are my tables taking up (take two)?)


```
select segment_name as table_name,
       sum(bytes)/(1024) as table_size_kb
from user_extents
where segment_type = 'TABLE'
group by segment_name;
```
Space Consumption III

Example (How much space are my tables taking up (take three)?)

- Size of table based on rows
  

  ```sql
  SELECT table_name, (AVG(row_len*num_rows)/(1024))
  FROM user_tables;
  ```

Example (Show more information on table size calculation)

  ```sql
  SELECT table_name, avg_row_len, num_rows,
          (AVG(row_len*num_rows)/(1024)) "size kb"
  FROM user_tables;
  ```
Create a SQL Script to drop all Tables

Example (Drop Single Table)

```
drop table tab;
```

Note

- Time consuming when many tables

Example (A script that drops all tables)

```
select 'drop table ' || ua.table_name || ';
from user_tables ua;
```

Note

- `||` is the string concatenation operator in Oracle
  - `'hello,' || ' world!' = 'hello, world!'`
- Recall keywords case insensitive in SQL
User Requirements

- All names in lower case in scripts

Example (A script that drops all tables (all lower case))

```
select 'drop table ' || lower(ua.table_name) || ' ; '
from user_tables ua;
```

Note

- The function `lower` converts a character string to lower case
Create a SQL Script to drop all Tables III

Example (Store the drop script in a file)

```sql
set heading off  -- do not show the column names
set feedback off  -- do not write n rows returned at the end
set pagesize 50000  -- make the page size very large
-- send output to a file
spool c:\drop_tables.sql
-- The actual query to execute
select 'drop table ' || lower(ua.table_name) || ' ;' from user_tables ua;
spool off  -- stop writing to a file
```

Example (Execute the commands in the file from within SQL*Plus)

```
SQL> @c:\drop_tables.sql
```

Note

- Assume file is stored in c: drop_files.sql
Outline

4 Tables

5 Columns

6 Constraints

7 Views
Example (The columns on the student table)

```sql
select *
from user_tab_cols utc
where utc.table_name = 'STUDENT'
```

Note

- Internally all identifiers are in upper case
Finding Specific Columns Based on Name

Example (Columns names that include the string ID)

```
select utc.*
from user_tab_cols utc
where utc.column_name like '%ID%'
```

Note

- `like` is simple string matching and `%` is multi-character wildcard
- Recall that internally in Oracle all identifiers are in upper case
Space Consumption Column Values

Example (Find the number of bytes needed to store a date)

```sql
select vsize(date) from dual;
```

Example (Find the number of bytes needed to store an interval)

```sql
select vsize(interval '37 12' day to hour) from dual;
```

Example (Find the number of bytes needed to store a float)

```sql
select vsize(123456789012345.34567890) from dual;
```
Data Types for Columns

Example (Find the most used data type for columns)

```
select utc.data_type, count(*) as no_cols
from user_tab_cols utc
group by utc.data_type
order by no_cols desc
```
Finding Primary Keys

Example (Name of tables that have a primary key)

```sql
select uc.table_name
from user_constraints uc
where uc.constraint_type = 'P'
```

<table>
<thead>
<tr>
<th>Constraint Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>Primary key</td>
</tr>
<tr>
<td>R</td>
<td>Referential Integrity (foreign key)</td>
</tr>
<tr>
<td>C</td>
<td>Check constraint</td>
</tr>
</tbody>
</table>
Missing Primary Keys

Example (Tables with no primary key)

```sql
select table_name
from user_tables
where table_name not in (select table_name
                          from user_constraints
                          where constraint_type = 'P');
```

Action to Take

Now you should add the missing primary keys or face the consequences!
Constraints on Which Columns

Example (Columns associated with a constraint)

```sql
select *
from user_cons_columns ucc
where ucc.table_name = 'STUDENT'
```

Note

- Only for the Student table
- All types of constraints are used
Example (Find tables with a composite primary key)

```sql
select uc.table_name, count(ucc.column_name)
from user_cons_columns ucc, user_constraints uc
where uc.constraint_type = 'P' -- primary key
and uc.owner = ucc.owner
and uc.constraint_name = ucc.constraint_name
group by uc.table_name
having count(ucc.column_name) > 1
```
Outline

4. Tables

5. Columns

6. Constraints

7. Views
Which Views are Defined

Example (Show all views)

```sql
select * from user_views
```
Example (The select statements used for creating the view)

```sql
select uv.text
from user_views uv
```

Note

- The text is a `long` column (an old type of `clob`) therefore the text may not be directly visible. Click on the field to see the content in most IDEs.
Outline

8  Sequences

9  Directories and Files

10 Synonyms

11 Comments
Which Sequences are Available

Example (Find all sequences)

```sql
select * from user_sequences
```

Example (Find the last number used from the sequence)

```sql
select us.last_number from user_sequences us
```
Outline

8 Sequences

9 Directories and Files

10 Synonyms

11 Comments
Examples

Example (Create a Directory Alias)

```sql
create directory file_stuff as 'c:\tmp';
```

Note

- Directory alias `file_stuff` is stored and used in upper case `FILE_STUFF`
- Directory name is case sensitive
- No warning if directory does not exist on server side

Example (Create a Stupid Directory Alias)

```sql
create directory dummy as 'c:\DoesNotExist';
```

Note

- Create a directory alias to a non-existing directory
Overview: Directories

Idea

- An alias for a directory on the server file system
- Can be used to read from and write to files
- Must have the create any directory to create directories
- Directory aliases can be use for external tables
- If the directory is drop in the operating system, a run-time error is raised if alias is used.
List Directory Aliases

Example

```
select * 
from all_directories;
```

Note

- Describes all directories aliases accessible to the current user
- View is in the `sys` schema
Example (Grant Read/Write Access)

grant read, write on directory file_stuff to <user name>;

Note

- Provides both read and write rights the directory alias

Example (Grant Read-Only Access)

grant read on directory file_stuff to <user name>;

Note

- Provides only read rights the directory alias
create or replace procedure str2file (dir_name varchar2, file_name varchar2, str varchar2) as
    v_file utl_file.file_type;
begin
    v_file := utl_file.fopen(dir_name, file_name, 'w');
    utl_file.put_line(v_file, str);
    utl_file.fclose(v_file);
exception
    when others then
        if utl_file.is_open(v_file) then
            utl_file.fclose(v_file);
        end if;
end;

Example (Called the Stored Procedure)
execute str2file (‘FILE_STUFF’, ‘new_file.txt’, ‘Hello File!’);
convert a text file to a clob

create or replace function file2clob (dir_name varchar2, file_name varchar2)
return clob is
  v_file utl_file.file_type; v_clob clob;
  v_line varchar2(2000); v_length int;
begin
  dbms_lob.createtemporary (v_clob, false);
  v_file := utl_file.fopen (dir_name, file_name, 'r');
  begin
    loop
      utl_file.get_line (v_file, v_line);
      v_length := dbms_lob.getlength (v_clob) + 1;
      dbms_lob.write (v_clob, length (v_line), v_length, v_line);
    end loop;
  exception when others then
    if utl_file.is_open (v_file) then
      utl_file.fclose (v_file);
    end if;
  end;
  return v_clob;
Example (Use the Clob Conversion)

```sql
-- Create a table with a clob column
create table withAClob(int, l clob);

-- Use the new function to load a file into a clob column
insert into withAClob
    values(1, file2clob('FILE_STUFF', 'test_read_file.txt'));
commit;

-- Check file is 'uploaded' as a clob
select *
from   withAClob;
```
Updating/Moving a Directory

Example (Relocate Directory Alias)

```sql
create or replace directory file_stuff as 'c:\tmp\db_files';
```

Note
- The directory alias has now been relocated

Example (Stupid Relocation)

```sql
create or replace directory file_stuff as '/unix/path/dos/machine';
```

Note
- No warning/error when creating non-existing directory
Example (Drop a Directory)

```sql
drop directory file_stuff;
```

Note

- The directory alias is now removed
Outline

8 Sequences

9 Directories and Files

10 Synonyms

11 Comments
Concept

- A synonym is an alias, such that it is simpler to refer to database objects owned by another user.

Example (List all non-public synonyms)

```sql
select *
from all_synonyms als
where als.owner <> 'PUBLIC';
```

Note

- A public synonym is visible to all database users
- Public synonyms are used heavily by the Oracle DBMS to provide access to meta data.
Example

Find the synonym name plus the base table for a public synonym that is a table and starts with a 'D'

```sql
SELECT als.synonym_name, als.table_owner, als.table_name
FROM all_synonyms als
WHERE als.owner = 'PUBLIC'
AND als.synonym_name LIKE 'D%'
AND EXISTS (SELECT 'x'
            FROM all_tables att
            WHERE att.owner = als.table_owner
            AND att.table_name = als.table_name);
```

Note

- The dual is return. This means that dual is real table and not a view.
Comments on Tables

Example (Find all tables with comments)

```sql
select utc.*
from user_tab_comments utc
where utc.comments is not null
```
Example (Find column comments on the Student table)

```
select ucc.column_name, ucc.comments
from user_col_comments ucc
where ucc.table_name = 'STUDENT'
```
Outline

12 Stored Procedures

13 Triggers

14 Timing
Example (Create the ls command as a stored procedure)

```sql
create or replace procedure ls is
  cursor c_tables is
    select * from cat;
  v_table_name cat.table_name%type;
  v_type cat.table_type%type;
begin
  open c_tables;
  loop
    fetch c_tables into v_table_name, v_type;
    exit when c_tables%notfound;
    p(v_table_name);
  end loop;
  close c_tables;
end;
/
```

Note

- The slash '/' at the end
Example (Executed the stored procedure)

-- to enable output from the server
set serveroutput on
-- exec is short for execute
exec ls;
User Requirements

- Create a stored procedure that returns the space usage for a table
- Must take the table name as input and return the size in KB
Example

```sql
CREATE OR REPLACE FUNCTION space_usage
     (p_table_name IN USER_TABLES TABLE_NAME%TYPE)
RETURN NUMBER IS
    v_size_kb NUMBER;
BEGIN
    SELECT (UT.AVG_ROW_LEN*UT.NUM_ROWS)/(1024)
    INTO v_size_kb
    FROM USER_TABLES UT
    WHERE UT.TABLE_NAME = UPPER(p_table_name);
    RETURN v_size_kb;
END;
/
```
The `into` keyword

The `upper` keyword

The `return` keyword

It is very important that parameter names, e.g., `p_table_name` are different from any column name

- The code convention is to prefix all parameter names with `'p_'`

There is no error handling here at all

- Should be added in a production version
Example (Executed the function)

```sql
-- to enable output from the server
set serveroutput on
declare
  v_rv number;
begin
  v_rv := space_usage('student');
  dbms_output.put_line('Size student table is ' || v_rv || ' KB');
end;
/
```

Note

- Note allowed to use lower case table names, e.g., student
- The return value has to be defined
- The return is printed to the screen
- The trailing slash
Example (Show source code for the `ls` stored procedure)

```
select us.text
from user_source us
where us.name = 'LS'
order by us.line
```

Note

- The text comes out in the corrected order due to the `order by` clause.
- Note the bad name convention there is a column name `name`. This names overlaps with a reserved word.
Reserved Words

- There is a huge number of reserved words in PL/SQL.
- They can (naturally) be queried

Example (List the Reserved Words)

```sql
select *
from v$reserved_words
```

Note

- All v$ views are also called `dynamic views`

Access Rights

It is possible that you do not have access to the view v$reserved_words
Outline

12 Stored Procedures

13 Triggers

14 Timing
Example (List all tables with triggers)

```sql
select distinct ut.table_name
from user_triggers ut
where ut.base_object_type = 'TABLE'
```
Outline

12 Stored Procedures

13 Triggers

14 Timing
Timing a Query

Example (From SQL*Plus)

set heading off
set pagesize 50000
set termout off
spool c:\test.lst
timing start
-- Start query to time
select c.car_id, c.road_id, c.speed
from car_gps_array c
order by car_id desc;
-- Stop query to time
timing stop
spool off
set termout on

Note

- Output is **spooled** to a file
- The result of the query is not displayed on screen
15 Tablespace Usage

16 Additional Information
### Example (Space consumption for tablespaces)

```sql
SELECT segment_name, tablespace_name, SUM(bytes/(1024*1024)) "MB"
FROM dba_segments
WHERE owner='TORP' AND segment_name NOT LIKE 'BIN'
GROUP BY segment_name, tablespace_name
ORDER BY segment_name, tablespace_name;
```

### Note
- Dropped tablespaces are not included
- Dropped database objects are in the ’BIN’ segment
- The size is in MB
Outline

15 Tablespace Usage

16 Additional Information
Sites

- Oracle technical network otn.oracle.com
  - The official homepage good information but no critique
- Morgan’s Library www.psooug.org/library.html
  - Many concrete examples, little or no explanation
- dbazine.com’s Oracle home page www.dbazine.com/oracle
  - Good examples
- dbasupport.com’s www.dbasupport.com/oracle/
  - Overall good site with information on several DBMSs