PERPETUAL TECHNOLOGIES, INC. www.PTI.net 800-538-0453 *ADVANCING THE DATA-DRIVEN WORLD*

ORACLE SQL QUICK GUIDE
FOR THE BEGINNING USER

LANGUAGE FUNDAMENTALS

Structured Query Language (SQL) is the method by which we are able to work with objects and their data inside our database. The following list gives an overview of the commands and their classification in SQL.

DDL

Data Definition Language: Commands that we use to create and alter object structures in the database. These commands do not change the actual data. Each change is committed immediately and ends the transaction including all DML issued up to that point.

CREATE	Create a new object n the database.
ALTER	Change the structure of an existing object.
DROP	Remove an object from the database.
ADD	Add a column, constraint, etc to an existing object.
MODIFY	Change an attribute such as a column datatype.
RENAME	Change the name of a column or object.

DML

Data Manipulation Language: Allows us to retrieve and make changes to the data in the database. Changes may be explicitly committed or rolled back.

SELECT	Query data in the database.
INSERT	Insert a new row into an existing table.
UPDATE	Change the value of existing row data in a table.
DELETE	Remove a row of data from an existing table.
MERGE	Merges one or more tables by either updating the existing row in the target table or inserting a new one depending upon whether it exists already or not.

DCL

Data Control Language: Allows us to control which users have privileges to access objects or carry out certain actions in the database.

GRANT	Give a role or privilege to a user.
REVOKE	Take a role or privilege away from a user.

TCL

Transaction Control Language: Allows us to make changes permanent, undo them, or create periodic rollback-to points.

COMMIT	Makes a DML change permanent.
ROLLBACK	Un-does an uncommitted change rather than COMMITing it. Brings back the before image.
SAVEPOINT	Creates a marker in a series of statements within a transaction so that we can ROLLBACK part of a transaction rather than the entire transaction.

BASIC SYNTAX AND EXAMPLES

Note: There are many variations and extensions to the syntax and examples provided below. See Oracle's documentation for complete syntax diagrams and usage examples.

http://tahiti.oracle.com

DDL (Data Definition) WITH TABLES…

```
CREATE TABLE table_name (
col1_name DATATYPE [NOT NULL] [DEFAULT expr]
column_constraint spec…],
col2…, col3…, …colx,
[table_constraint_spec] );
```

```
CREATE TABLE names (
fname VARCHAR2(20), lname VARCHAR2(20),
ssn INTEGER(9) PRIMARY KEY, ph_num INTEGER(10) NOT
NULL );
```

```
CREATE TABLE names_cpy AS
SELECT * FROM names;
```

```
ALTER TABLE names ADD (bday VARCHAR2(10));
```

```
ALTER TABLE names MODIFY (bday DATE);
```

```
ALTER TABLE names DROP COLUMN bday;
```

```
ALTER TABLE names RENAME COLUMN ph_num TO phone;
```

```
ALTER TABLE names RENAME TO staff
```

```
DROP TABLE names_cpy;
```

DML (Data Manipulation) WITH TABLES…

SELECT is different than other DML statements in that it does not actually change/manipulate the data by itself. It is sometimes used with other commands to carry out DML using data retrieved from elsewhere in the database. Following is the basic query-only syntax.

```
SELECT col1, col2, …colx | * FROM table_name
[WHERE colx = expr] [ORDER BY colx];
```

```
SELECT fname, lname FROM staff
WHERE phone IS NOT NULL
ORDER BY lname;
```

True Data Manipulation…

```
INSERT INTO staff (fname, lname, ssn)
VALUES (‘Chris’, ‘Plum’, 318675309);
```

```
INSERT INTO staff SELECT * FROM new_hires;
```

```
DELETE FROM staff WHERE lname = ‘Smith’;
```

```
UPDATE staff SET lname = ’Plum’
WHERE fname=’Anna’
```

```
MERGE INTO staff s USING my_staff m
ON (s.ssn = m.ss#)
WHEN MATCHED THEN
  UPDATE staff SET
  s.fname = m.fname,
  s.lname = m.lname,
  s.phone = m.ph#
WHEN NOT MATCHED THEN
  INSERT (fname, lname, ssn, phone)
  VALUES (m.fname, m.lname, m.ss#, m.ph#);
```

DCL (Data Control) …and PRIVILEGES

Privileges are required in order for a user to be able to do anything in the database. OBJECT PRIVILEGEs enable a user to make a change to the data or, in other words, use DML against specific objects in the database. SYSTEM PRIVILEGEs allow a user to carry out a particular action in the database. For a complete list and descriptions, see Oracle documentation. Some examples follow.

OBJECT PRIVILEGES	SYSTEM PRIVILEGES
SELECT ON table, sequence, view	CREATE (ANY) object
INSERT ON table, view	ALTER (ANY) object
UPDATE ON table, view	DROP (ANY) object
DELETE ON table, view	CREATE SESSION
ALTER ON table	UNLIMITED TABLESPACE
FLASHBACK ON table	FLASHBACK (ANY) table
EXECUTE ON procedure, function	EXECUTE (ANY) object
INDEX ON table	RESUMABLE
REFERENCES ON table	EXPORT/IMPORT FULL DATABASE

```
GRANT privilege ON object
TO user [WITH GRANT
OPTION];
```

```
GRANT privilege TO user
[WITH ADMIN OPTION];
```

Notice the WITH GRANT/ADMIN OPTION for each type of privilege. These options give the grantee the ability to further grant the same privileges to other users. Also note that OBJECT PRIVILEGES are always granted “ON” a specific object. The owner of an object automatically has all object privileges on the objects they own. To grant all object privileges to a non-owner:

```
GRANT ALL ON object TO user;
```

Transaction Control

Transaction Control is a very important aspect of SQL statement execution to understand. When you request a change be made in the database, it is not visible to other users or made permanent until you “COMMIT” your transaction. Consider the following which deletes all rows from the “staff” table.

```
DELETE FROM staff;
```

This being the first change you execute inside your session, begins your first transaction. Any users who SELECT from staff will still see all rows in the table because you haven't yet made the change permanent. You may issue additional DML statements to change the data as your transaction continues. The following will end your transaction:

```
COMMIT;
```

Be aware, in addition to an explicit COMMIT, ANY DDL issued inside this session will implicitly commit everything up until that point in your session.

Prior to issuing a COMMIT or DDL statement, you may decide that you were in error and want to undo the changes. As in the above example, maybe you forgot your WHERE clause. If this is the case, issue a ROLLBACK to completely undo the change.

```
ROLLBACK;
```

Unfortunately, had you also made many other changes that were not yet committed in your session, they would also be rolled back. To control this ROLLBACK behavior, you should consider using SAVEPOINTs as follows.

```
UPDATE staff SET lname = ‘Greer’
WHERE ssn = 315984545;
--many more updates…
SAVEPOINT 1_after_upd;
DELETE FROM staff WHERE lname = ‘Plum’;
DELETE FROM staff;
--forgot WHERE clause…
SAVEPOINT 2_after_del;
INSERT INTO staff SELECT * from new_hires;
--realize your mistake…
ROLLBACK to SAVEPOINT 1_after_upd
```

In this case, your changes are only rolled back to the SAVEPOINT that you created just before you did the large DELETE operation, so your updates are preserved.

LOGGING INTO AND OUT OF SQLPLUS

```
Sqlplus username/ @database_name:  logging into
sqlplus
```

```
C:\app\oracle>sqlplus system@oradb
SQL*Plus: Release 11.1.0.7.0 - Production on Wed
Aug 26 14:04:37 2009
Copyright (c) 1982, 2008, Oracle.
All rights reserved.
Enter password:
Connected to:
Oracle Database 11g Enterprise Edition Release
11.1.0.7.0 - Production
With the Partitioning, OLAP, Data Mining and Real
Application Testing options
SQL>
```

Exit: logs out of sqlplus, commits or rolls back any pending changes and returns control to the operating system.

```
SQL> exit
Disconnected from Oracle Database 11g Enterprise
Edition Release 11.1.0.7.0 - Production
With the Partitioning, OLAP, Data Mining and Real
Application Testing options
C:\app\oracle>
```

BASIC EXAMPLES FOR SQLPLUS EDITING AND FORMATTING

SQLPLUS Command	Description
`DESCRIBE <schema>.<table>`	describes table columns
`EDIT`	edits current sql statement in buffer
`START, RUN or @`	executes stored statement on O/S filesystem
`HOST <O/S command>`	executes host command on O/S while in SQLPLUS
`SPOOL <filename.ext>`	saves sql statement output to filename on O/S filesystem
`CHANGE or c/name/name`	changes first input with second input for line in buffer
`GET </path/filename.ext>`	loads the file into the sqlplus buffer
`SAVE </path/filename.ext>`	saves the contents in the buffer to a file
`SHOW <parameter>`	shows the current value for parameter ex. (user, spool, linesize)
`LIST`	show current buffer contents

```
SQL> describe hr.departments
 Name                                          Null?
Type
 ------------------- -------- ---------------------
-- DEPARTMENT_ID                   NOT NULL NUM-
BER(4)
DEPARTMENT_NAME                 NOT NULL VAR-
CHAR2(30)
MANAGER_ID                                NUMBER(6)
LOCATION_ID                               NUMBER(4)
```

```
SQL> host dir
 Volume in drive C has no label.
 Volume Serial Number is 94F3-F664
Directory of C:\Users\mpyle
08/24/2009  09:51 PM    <DIR>          .
08/24/2009  09:51 PM    <DIR>          ..
08/12/2009  12:46 PM    <DIR>          AppData
3 File(s)       272,445 bytes
3 Dir(s)  55,841,816,576 bytes free
```

```
SQL> select department_nmae from hr.departments;
select department_nmae from hr.departments
      *
ERROR at line 1:
ORA-00904: “DEPARTMENT_NMAE”: invalid identifier
SQL> c/department_nmae/department_name
  1* select department_name from hr.departments
SQL> /
DEPARTMENT_NAME
------------------------------
Administration
Marketing
Purchasing
```

```
SQL> show linesize
linesize 80
```

```
SQL> show user
USER is “SYS”
```

FORMATTING YOUR OUTPUT

Using commands such as the following, helps to display your output in a more readable presentation and can be used to format reports for printing.

`col <column_name> format a30`	sets the column to display alphabetic character up to specified number of characters until wrapping occurs
`col <column_name> format  999,999`	sets the column to display specified character at specified intervals on number columns
`set pages 200`	sets the page length to specified number of rows until heading is displayed again
`set lines 150`	sets the line length to specified number of characters until wrapping occurs
`set head off`	sets the heading off
`set trim on`	trims the blank spaces off ends of lines
`set underline <character> off/on`	sets underline of column heading to specified character, on/off
`set pause on/off`	sets pause for pages on/off (pauses are based on what pagesize is set to)
`set feedback on/off`	Sets the informational return of rows on/off

Using this alter session statement allows you to change the date format for your session only.

```
SQL> alter session set nls_date_format=”DD-MON-YYYY
hh24:MI:SS”;
```

CONCLUSION

There are many many good online references for SQL statement execution. This document only bullet-points some of the very basic commands. It is a very powerful language that, when exploited, will enable you to produce very complex reports.

Perpetual Technologies, Inc. (PTI) provides mission-critical database and information systems support to commercial and government enterprises worldwide. Focused on improving performance and lowering costs, our subject-matter experts plan, design, develop, deploy, and manage Oracle database environments running on UNIX and Windows platforms. Perpetual Technologies strives to create tailored, flexible IT solutions in the areas of Oracle database 8i, 9i, 10g, 11g, Oracle RAC, capacity planning, disaster recovery planning, performance tuning, Oracle Application Server , Oracle content manager, Oracle database design, complete or supplemental remote Oracle database administration, after-hours DBA coverage, and Oracle database vacation support.

PERPETUAL TECHNOLOGIES, INC.

5649 Lee Road
Indianapolis, IN 46216
800-538-0453
www.PTI.net

PERPETUAL TECHNOLOGIES, INC.

ORACLE SQL
Quick Reference Guide
FOR THE BEGINNING USER

ADVANCING THE DATA-DRIVEN WORLD