Oracle/SQL Tutorial[1]

Michael Gertz
Database and Information Systems Group
Department of Computer Science
University of California, Davis
gertz@cs.ucdavis.edu
`http://www.db.cs.ucdavis.edu`

This Oracle/SQL tutorial provides a detailed introduction to the SQL query language and the Oracle Relational Database Management System. Further information about Oracle and SQL can be found on the web site `www.db.cs.ucdavis.edu/dbs`.

Comments, corrections, or additions to these notes are welcome. Many thanks to Christina Chung for comments on the previous version.

Recommended Literature

George Koch and Kevin Loney: ORACLE*8 The Complete Reference* (The Single Most Comprehensive Sourcebook for Oracle Server, Includes CD with electronic version of the book), 1299 pages, McGraw-Hill/Osborne, 1997.

Michael Abbey and Michael Corey: ORACLE*8 : A Beginner's Guide [A Thorough Introduction for First-time Users]*, 767 pages, McGraw-Hill/Osborne, 1997.

Steven Feuerstein, Bill Pribyl, Debby Russell: ORACLE *PL/SQL Programming* (2nd Edition), O'Reilly & Associates, 1028 pages, 1997.

C.J. Date and Hugh Darwen: *A Guide to the SQL Standard* (4th Edition), Addison-Wesley, 1997.

Jim Melton and Alan R. Simon: *Understanding the New SQL: A Complete Guide* (2nd Edition, Dec 2000), The Morgan Kaufmann Series in Data Management Systems, 2000.

[1] revised Version 1.01, January 2000, Michael Gertz, Copyright 2000.

Contents

1 SQL – Structured Query Language

1.1 Tables

In relational database systems (DBS) data are represented using *tables* (*relations*). A query issued against the DBS also results in a table. A table has the following structure:

Column 1	Column 2	. . .	Column n	
				⟵ Tuple (or Record)
. . .	. . .	. . .	. . .	

A table is uniquely identified by its name and consists of *rows* that contain the stored information, each row containing exactly one *tuple* (or *record*). A table can have one or more columns. A *column* is made up of a column name and a data type, and it describes an attribute of the tuples. The structure of a table, also called *relation schema*, thus is defined by its attributes. The type of information to be stored in a table is defined by the data types of the attributes at table creation time.

SQL uses the terms *table, row*, and *column* for *relation, tuple*, and *attribute*, respectively. In this tutorial we will use the terms interchangeably.

A table can have up to 254 columns which may have different or same data types and sets of values (domains), respectively. Possible domains are alphanumeric data (strings), numbers and date formats. Oracle offers the following basic data types:

- **char**(n): Fixed-length character data (string), n characters long. The maximum size for n is 255 bytes (2000 in Oracle8). Note that a string of type **char** is always padded on right with blanks to full length of n. (☞ can be memory consuming).
  Example: **char**(40)

- **varchar2**(n): Variable-length character string. The maximum size for n is 2000 (4000 in Oracle8). Only the bytes used for a string require storage. *Example:* **varchar2**(80)

- **number**(o, d): Numeric data type for integers and reals. o = overall number of digits, d = number of digits to the right of the decimal point.
  Maximum values: o =38, d= −84 to +127. *Examples:* **number**(8), **number**(5,2)
  Note that, e.g., **number**(5,2) cannot contain anything larger than 999.99 without resulting in an error. Data types derived from **number** are **int**[**eger**], **dec**[**imal**], **smallint** and **real**.

- **date**: Date data type for storing date and time.
  The default format for a date is: DD-MMM-YY. *Examples*: '13-OCT-94', '07-JAN-98'

- **long**: Character data up to a length of 2GB. Only one **long** column is allowed per table.

Note: In ORACLE-SQL there is no data type **boolean**. It can, however, be simulated by using either **char**(1) or **number**(1).

As long as no constraint restricts the possible values of an attribute, it may have the special value *null* (for unknown). This value is different from the number 0, and it is also different from the empty string ''.

Further properties of tables are:

- the order in which tuples appear in a table is not relevant (unless a query requires an explicit sorting).
- a table has no duplicate tuples (depending on the query, however, duplicate tuples can appear in the query result).

A *database schema* is a set of relation schemas. The extension of a *database schema* at database run-time is called a *database instance* or *database*, for short.

1.1.1 Example Database

In the following discussions and examples we use an example database to manage information about employees, departments and salary scales. The corresponding tables can be created under the UNIX shell using the command **demobld**. The tables can be dropped by issuing the command **demodrop** under the UNIX shell.

The table EMP is used to store information about employees:

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	800	20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	30
....	....	....	....	....	....	....
7698	BLAKE	MANAGER		01-MAY-81	3850	30
7902	FORD	ANALYST	7566	03-DEC-81	3000	10

For the attributes, the following data types are defined:

EMPNO:**number**(4), ENAME:**varchar2**(30), JOB:**char**(10), MGR:**number**(4),
HIREDATE:**date**, SAL:**number**(7,2), DEPTNO:**number**(2)

Each row (tuple) from the table is interpreted as follows: an employee has a number, a name, a job title and a salary. Furthermore, for each employee the number of his/her manager, the date he/she was hired, and the number of the department where he/she is working are stored.

The table DEPT stores information about departments (number, name, and location):

DEPTNO	DNAME	LOC
10	STORE	CHICAGO
20	RESEARCH	DALLAS
30	SALES	NEW YORK
40	MARKETING	BOSTON

Finally, the table SALGRADE contains all information about the salary scales, more precisely, the maximum and minimum salary of each scale.

GRADE	LOSAL	HISAL
1	700	1200
2	1201	1400
3	1401	2000
4	2001	3000
5	3001	9999

1.2 Queries (Part I)

In order to retrieve the information stored in the database, the SQL query language is used. In the following we restrict our attention to simple SQL queries and defer the discussion of more complex queries to Section 1.5

In SQL a query has the following (simplified) form (components in brackets [ ] are optional):

```
select [distinct] <column(s)>
from <table>
[ where <condition> ]
[ order by <column(s) [asc|desc]> ]
```

1.2.1 Selecting Columns

The columns to be selected from a table are specified after the keyword **select**. This operation is also called *projection*. For example, the query

```
select LOC, DEPTNO from DEPT;
```

lists only the number and the location for each tuple from the relation DEPT. If all columns should be selected, the asterisk symbol "∗" can be used to denote all attributes. The query

```
select * from EMP;
```

retrieves all tuples with all columns from the table EMP. Instead of an attribute name, the **select** clause may also contain arithmetic expressions involving arithmetic operators etc.

```
select ENAME, DEPTNO, SAL * 1.55 from EMP;
```