

Mini User's Guide for SQL*Plus

T. J. Teorey

Table of Contents

Oracle/logging-in	1	Nested subqueries	5
SQL create table/naming rules	2	Complex functions	6
Update commands	3	Save a query/perm table	6
Select commands	3	Special commands	6
Set operations	4	SQL views	9
Built-in functions	4	Index	10

Oracle

Oracle 8.1.6 resides on a Sun Sparc10 node. However, you do not need to login directly to this machine, but think of it as a database server to be accessed from another machine, for example, canoe.engin.umich.edu.

Logging in to UNIX and Oracle SQL*Plus

```
login: <your Unix login-id>
password: <your Unix login-password>
your-unix-node% source /usr/caen/oracle/local/muscle
your-unix-node% sqlplus
Enter user-name: <your-oracle-id>
Enter password: <your-oracle-password>
```

/*You are now in SQL*Plus and can start issuing SQL*Plus commands*/

```
SQL> select * from employee; /*example of an sqlplus command*/
SQL> grant connect to your-oracle-id identified by your-new-password; /*change password*/
```

Input and Output Files for Oracle

To save the transcript of an sqlplus session or part of a session:

```
SQL> set echo on; /*displays SQL code with results of the query*/
SQL> spool <filename>; /*filename will be appended with .1st if a postfix is not provided*/
SQL> spool off; /*will turn off the transcript spooling. You can now print the file with the
instructions provided below*/
```

To read in a file:

```
SQL> start <filename>;
or SQL> @<filename>;
where filename must be a OpSys file ending with .sql (the .sql may or may not be supplied to
Oracle, but must be part of the name of the file).
```

Print Oracle Results

To print on a Sun or IBM the users can say: lpr -P<printer_name> <filename>
printer_names (2340eecss2,2341eecsh1,4327eecss1,4327eecss2,4440eecss1,4440eecsm1)
HP: lp -d<printer_name> <filename>

Access from any computer:

1. To any UNIX machine, by typing "telnet <node_name>"
2. To any UNIX machine from a Merit terminal (where you see the "Which Host?" prompt), by doing a "telnet <host>@engin.umich.edu"

Note: use "telnet hostinfo" to see a selection of free nodes.

Oracle SQL*Plus Programming Examples

Naming rules:

1-30 characters long (a-z, 0-9,_,,,\$,#), begin with a letter
 No quotation marks
 No duplicates of Oracle reserved words, no duplicate of another Oracle object of the same type

Basic example: suppliers, parts, and shipments.

```
create table supplier
(snum      number not null,
 sname     char(12),           /*max is 240 characters*/
 city      char(12),
 status    number,
 primary key (snum)); /*must be defined as a primary
                                     key before it is defined as a foreign key*/

create table part
(pnum      number not null,
 pname     char(10),
 length    number,
 weight    number,
 primary key (pnum)); /*cannot have both "unique" and
                                     primary key*/

create table shipment
(snum      number not null,
 pnum      number not null,
 qty       number,
 shipdate  date not null,
 primary key (snum, pnum,shipdate),
 foreign key (snum) references supplier,
 foreign key (pnum) references part);
```

Syntax Rules

1. Semicolon needed, no continuation character needed.
2. not => !, ~(hat), not(.....)
3. Constraints on create table
 - not null - null not allowed for this column (attribute)
 - unique - attribute may not have duplicate values
 - primary key - explicitly designates simple or composite primary key
 - foreign key - explicitly specifies referential integrity
 - check - specifies range constraints or specific values (see 5-39)
4. Logical operators used with "where" clause: and, or, not, !=, ()
5. Comparison operators: (),=,!=,~=,<,<=,>,>=
 - in - equal to any member of, same as "=any"
 - not in - same as "!= all" => false of any member of set (select...) is null
 - any - same as "in"
 - all - compares a value to every value returned by a list
6. Set operators
 - union - combines queries to display any row in each subquery
 - intersect - combines queries to display distinct rows common to all subqueries
 - minus - combines queries to return all distinct rows returned by the first query but not the second
7. Order by: asc, desc
8. Basic definitions
 - create table - defines a table
 - alter table - add a new column, lengthen the width of a column /*enlargements only*/.
 - drop table - destroys an existing base table
 - create view, drop view
 - create index, drop index.....create index x on t (p,q desc, r);
 - create integrity, drop integrity
9. Data types:

number (integer, 31 bits), smallint (15 bits)
 (p[q]), p digits total, q to the right of the decimal point
 float, real (not in Oracle), double precision (not in Oracle)
 char(n), character(n), varchar(n), date

Update Commands

alter table: alter table supplier modify (sname varchar(12));
 alter table supplier add (address char(20) not null);
insert (a single row): insert into supplier values (1,'Smith','Detroit',10);
delete (one or more rows): delete from supplier where status > 1;
update: update shipment
 set qty = 450
 where qty = 500 and snum =3 and pnum = 31;

Select commands

/*display the entire supplier table*/ select * from supplier;
/*display supplier number, status for all suppliers in London

Note: case sensitive within the quotes*/

```
select snum, status
from supplier
where city = 'London';
```

/*display all supplier and shipped part information, but omitting suppliers with status of 40*/

```
select s.*, sh.*
from supplier s, shipment sh
where s.snum = sh.snum
and s.status != 40;
```

/*use multiple table invocations to determine manager's name, one level above employee*/

```
select f.ename
from emp e, emp f
where e.ename = 'Smith'
and f.empno = e.mgrno;
```

Set operations

/*which parts (part numbers) are shipped by supplier 1 or supplier 2?*/

```
select pnum, snum from shipment -OR- select pnum, snum from shipment
where snum = 1                    where snum = 1 or snum = 2;
union
select pnum, snum from shipment -OR- select pnum, snum from shipment
where snum =2;                   where (snum = 1 or snum =2);
```

/*which parts (part numbers) are shipped by both suppliers 1 and 3?*/

```
select pnum from shipment
where snum =1
intersect
select pnum from shipment
CANNOT DO THIS:
select pnum from shipment
where snum = 1 and snum = 3;
```

where snum = 3;

Built-in functions

/*display the total number of suppliers*/

```
select count(*) from supplier;
```

/*display the total number of suppliers actually shipping parts*/

```
select count (distinct snum)
from shipment;
```

/*display the average quantity of a shipment of part number 31*/

```
select avg (qty)
from shipment
where pnum = 31;
```

/*order by attribute names -- note that asc is the default*/

```
select pnum, snum, qty
from shipment
order by pnum asc, snum desc;
```

/*order by column number as specified in the select line*/

```
select pnum, snum, qty
from shipment
order by 1, 2;
```

/*for each part shipped, display the part number, the total shipment quantity, and the count of orders for each part; note that pnum in the select line must be in a "group by" command*/

```
select pnum, sum(qty), count(qty)
from shipment
group by pnum;    /*note: the group by orders items ascending by default*/
```

/*display part numbers for all parts supplied by more than one supplier*/

```
select pnum
from shipment
group by pnum
having count(distinct snum) >1;
```

/*group by primary, secondary columns*/

```
select pnum, snum, max(qty)
from shipment
group by pnum,snum;
```

/*find greatest and least values among attributes within each row, for all rows*/

```
select greatest(empno,mgrno), ename
from emp;
```

Nested subqueries

/*display supplier names who supply part 32.....and equivalent query*/

<pre>select s.sname from supplier s where s.snum in (select snum from shipment sh where sh.pnum = 32);</pre>	<pre>select s.sname from supplier s, shipment sh where s.snum = sh.snum and sh.pnum = 32;</pre>
---	---

*/*note indentation for nested query*/*

/*display supplier names who supply at least one part with weight over 20.....&...equiv query*/

```

select s.sname
from supplier s
where s.snum in
  (select sh.snum
   from shipment sh
   where sh.pnum = any
     (select p.pnum
      from part p
      where p.weight > 20));

```

```

select s.sname
from supplier s, shipment sh, part p
where s.snum = sh.snum
and sh.pnum = p.pnum
and p.weight > 20;

```

/*which suppliers are currently not shipping any parts with weight over 20?

```

select s.sname
from supplier s
where s.snum not in
  (select sh.snum
   from shipment sh, part p
   where sh.pnum = p.pnum and p.weight > 20);

```


SQL> run; /*for && - redoes query with same value as before*/

Note: & means prompt with value, not saved; && means prompt with value, value is saved.

8. Setting up a report title, suppress the title, and set up special column headings:

SQL> tttitle [right|left] 'This is a Title of a Report' ; /*default is center – persists until you execute "tttitle off"*/

SQL> column schema_column_name heading "new_column_name";

/*This will produce special column names as specified in quotes, but does not

persist. Quote

SQL> select ename "Employee Name", sal "Employee Salary" from emp;

SQL> select ename empname, sal empsalary from emp;

9. Report formatting:

SQL> break on deptno skip 1;

/*do not repeat deptno, skip a line between deptno's*/

SQL> break on deptno on mgr skip 1; /* do not repeat deptno or mgr, skip a line*/

SQL> run; /*execute the previous select with the new breaks*/

/*Note: you need to leave a blank line between the end of the query and the "run". This will cause echoing of the query. */

SQL> clear break;

SQL> clear column; /*resets break and column settings*/

SQL> set pagesize 54; /*overrides default of 14 lines/page*/

10. Data Formats: set column settings to override defaults, in-line format specifications

SQL> column avg(sal.monsal) format \$99,999.99;

SQL> column deptname format A6;

SQL> column deptno format 99999;

SQL>where to_char(shipdate, 'yy') = 94.....etc.

SQL> select to_char(monsalary, '\$99,999.99').....etc.

SQL> alter session set NLS_DATE_FORMAT = 'DD-MON-YYYY'; /*display 4-digit year*/

11. Recursive hierarchy access in SQL*Plus (top-down hierarchy)

**SQL> select lpad(' ',2*level)||ename organization_chart
from emp
connect by prior empno = mgrno
start with ename = 'King';**

Note: a bottom-up hierarchy can be obtained by reversing the attributes after "connect by

prior" and spe

12. Partial matching (see also SOUNDEx for words that sound like something else)

/* Look for names with "a" as the second letter and any string afterwards */

SQL> select ename from emp where ename like '_a%';

13. Size check -- determine the count of rows satisfying the query before displaying the results

**select count(*) from shipment
where pnum = 31;**

Size check -- limit display of rows before displaying the results of the whole query.

**select snum, pnum from shipment
where pnum = 31
and rownum <= 15;**

14. Create sequence command to set up artificial primary key, max. number is 10*e**27 -1.

```
SQL> create sequence myseq increment by 1 start with 1; /*defaults incr to 1 start with 1*/
SQL> create table mytable (myseq, attr1....., attr2 ....., attr3);
SQL> insert into mytable values (myseq.nextval, value for attr1, .....);
SQL> alter sequence mseq increment by 5;
```

15. Create index commands for B⁺-tree

```
SQL> create [unique] index indexname on supplier(snum [asc|desc]); /*unique=> hashing*/
SQL> create unique index indexname on supplier(snum); /*unique index on primary key*/
SQL> create index indexname on shipment (shipdate); /*non-unique index on non-key*/
SQL> create index index2 on shipment (pnum, shipdate); /*non-unique concatenated index*/
```

16. Check clause in create table commands.

```
check (status>10),
check (status between 10 and 40),
check (city in ('Athens','London')),
check (city != 'Paris' or status = 20);
```

17. SQL editing line-by-line

```
SQL> select * from supplier
  2 where snum = 14
  3 and sname = 'Smith';
```

no rows selected

```
SQL> 2 where snum = 1
SQL> run;
  1 select * from supplier
  2 where snum = 1
  3* and sname = 'Smith'
```

SNUM	SNAME	CITY	STATUS
1	Smith	London	20

SQL Views

view -- a named, derived (virtual) table in SQL

base table -- actual tables used in the original schema definition

motivation for views

(1) simplicity -- simplifies complex queries often used, or accessed by novice SQL users

(2) security -- provides different views of the same data

(3) data independence -- view queries constant even though the base table schemas are changed

query on a view -- treats the view as if it were a real table

recursive definitions -- a view may contain other views

Example: /*create a view that shows which parts are under shipment by each supplier and how many*/

```
create view shipments (suppname, partname, quantity, shipdate) as
  select s.sname, p.pname, sh.qty, sh.shipdate
  from supplier s, part p, shipment sh
  where s.snum = sh.snum
  and sh.pnum = p.pnum;
```

Note: attributes in the select clause must match the attributes in the create view attribute list
select * from shipments;

View Update Example

table1	<u>A</u>	<u>B</u>	<u>C</u>	table2	<u>C</u>	<u>D</u>
	a	b	e		e	k
	a	d	f		e	m
	a	d	e		f	n

```
create view view1 (A,B,C,D) as
  select table1.A, table1.B, table1.C, table2.D
  from table1, table2
  where table1.C = table2.C;
```

view1	<u>A</u>	<u>B</u>	<u>C</u>	<u>D</u>	
	a	b	e	k	/* row 1 */
	a	b	e	m	/* row 2 */
	a	d	f	n	/* row 3 */
	a	d	e	k	/* row 4 */
	a	d	e	m	/* row 5 */

- view update: delete row 1 from view1
- (1) delete table1.row1 => destroys view1.row 2 as a side effect
 - (2) delete table2.row1 => destroys view1.row 4 as a side effect
 - (3) delete both table1.row1 and table2.row1 => destroys both rows 1 and 4 as side effects

Rules on View Updating (Oracle SQL*Plus)

1. View must be based on a single base table
2. View query must not contain group by clause, distinct clause, group functions, rownum
3. If you specify an attribute as "not null" in the original schema, you must include that attribute in the view (otherwise you will get an error message when inserting into the view).
4. If you specify the "with check option" in the view, no attribute constrained by "where" in the view can be updated.

Index

alter table - 3
and - 3
available tables - 6
avg - 4
break on - 7
change password - 6
check - 8
clear breaks - 7
clear column - 7
column format command - 7
complex functions - 6
computations - 6
counseling (Oracle SQL*Plus) - 2
count - 4
create index - 8
create sequence - 8
create table - 2,6
delete - 3
desc table - 6
distinct - 4
editing line-by-line in SQL*Plus - 8
greatest - 5,6
group by - 4, 5
group functions (**max**, **min**, **avg**, **sum**) - 4
having - 5
help commands - 6
input and output files (Oracle) - 1
insert - 3
intersect - 4
least - 5,6
logging in (UNIX, Oracle) - 1
lpad function (hierarchies) - 7
max (min) function - 5
minus - 4 (not listed, but check union for format)
months_between function- 6
multiple invocations of a table - 4
naming rules -2
nested subqueries - 5
nvl function - 6
order by - 4
pagesize - 7
partial matching - 7
permission (read-only) - 6
print results (Oracle) - 1
privileges in tables - 6
prompts - 7
renaming tables - 6
report titles - 7
save a query result in a table - 6
select - 3-6
 select * - 3
set operations (**union**, **intersect**, **minus**) - 4
set pagesize - 7
size check - 8

SOUNDEX - 8
subqueries - 5
sum function - 5
syntax rules for SQL*Plus - 2,3
title - 7
union - 4
update - 3
views - 9
view update - 9
where - 3