Introduction to Oracle

Oracle is one powerful Relational Database Management System. The server we are using in the department is Oracle 8.0.5 Server.
The following URLs provide two useful on-line reference for Oracle 8.0.5 and Oracle 8i respectively:

http://oracle8.cse.cuhk.edu.hk
http://oracle8i.cse.cuhk.edu.hk

There are three kinds of commands in Oracle:

SQL commands:
For working with information in the database.

PL/SQL blocks:
For working with information in the database with the use of the block-structured procedural language (flow-of-control statements, variable declaration, procedure definition, …).

*SQL*Plus commands:*
For formatting query results, setting options, and editing and storing SQL commands and PL/SQL blocks.

Getting into Oracle Server

Before getting into the server, we have to setup some environment variables. Try to include the following statement in the file *~/.cshrc* :

source /usr/local/oracle8/setup for CSE, SEM
source /home2/oracle8/setup for IE

Re-login and type the following to get into the server :
sqlplus

Then the system will respond with the following display:

```
SQL*Plus: Release 8.0.5.0.0 - Production on Fri Sep 18 15:47:1
1999
```



```
(c) Copyright 1998 Oracle Corporation.  All rights reserved.

Enter user-name:
```

You are now required to key in your username (e.g. db91234@csedb) and password. After you key in a valid username and password, the system will response with:

```
Connected to:
Oracle8 Enterprise Edition Release 8.0.5.0.0 - Production
PL/SQL Release 8.0.5.0.0 - Production

SQL>
```

Now you have successfully get in the Oracle Server. You can post any SQL query or any system procedure provided by the server.

General Environment

You can span a SQL statement into several lines. You have to insert ";" to indicate the end of the SQL statement, but no need to insert semicolon at the end of SQL*Plus commands.
e.g.

```
SQL> select stu_id from students
  2  where stu_name = 'Chan Tai Man'
  3  ;
```

Where *SQL>, 2, 3* are the prompts for you to insert statement or clauses.

Getting Help

Use *Help command_name* to get a detail description on the syntax and the nature of the command being inquired.
e.g.

```
Help select
```

Learning the Basics

Creating tables
Suppose you want to create a new table *deposit* in Oracle Server with the following scheme:

Deposite-scheme = (
branch_name : string[50],
account_number : integer with max 12 digits,
customer_name : string[50],
balance : fixed point number).

Use *Create Table* to create the table.
e.g.

```
SQL> create table deposit (
  2     branch_name varchar2(50),
  3     account_number number(12),
  4     customer_name varchar2(50),
  5     balance number(12,2),
  6     primary key (account_number)
  7     );
```

Note that the last line in the above example is use to indicate the column "account_number" would be the primary key of the table. The type number(12,2) is the type referring to numbers having precision 12 and scale 2.

Show table structure
To show the structure of table *deposit*, you can:

```
Describe deposit
```

Show tables in your Oracle account
To show all the tables created in your account, you can:

```
Select table_name from sys.user_tables;
```


Deleting tables
To delete the table generated in the previous example, you can:

```
Drop Table deposit;
```

Once you throw away the table, you cannot retrieve all the records inside the table and the table definition will be lost.

Changing definition of tables
Use *Alter Table* for adding columns or adding, changing or dropping constraints on an existing table. This command keeps the existing records in the table without discarding them.

e.g.

```
Alter Table deposit
Add customer_address varchar2(100) null;
```

Inserting tuples in table
To insert a tuple ('CU',10230231,'Chan Tai Man', 5000.50) into the table *deposit*, you can:

```
Insert Into deposit Values ('CU',10230231,
'Chan Tai Man', 5000.50);
```

Removing tuples in a table
To delete tuple with account number equal to 10230231, you can :

```
Delete deposit where account_number = 10230231;
```

To remove all tuples in the table *deposit*, you can:

```
Truncate Table deposit;
```

Updating tuples in a table

To change the name of tuple with account number 10230231, you can:

```
Update deposit Set customer_name = 'Chan Siu Ming'
where account_number = 10230231;
```

Making query

e.g.

```
Select * from deposit;
Select D.account_number from deposit D;
Select * from deposit where customer_name = 'Chan Man';
```

Remote table access and authorization

Use the *Grant* option to grant an object privilege to specified users.

e.g. To grant the Select, Insert, and Delete object privileges of table deposit to all users, you can:

```
Grant select, insert, delete On deposit To Public
```

e.g. To grant all object privileges of table *deposit* to user *smith*, you can:

```
Grant all On deposit To smith
```

To remote access a granted object.

e.g. To access the table *deposit* belonging to user *ass1*, you can:

```
Select * from ass1.deposit where balance > 5000;
```

Using command file

Use a command file for storing a sequence of SQL statements and then use *Start* or @ to execute the file.

e.g. Content of the command file *infile.sql* is:

```
Create Table temp (
        id number(9);
        name varchar2(50);
);
Select * from temp;
Drop Table temp;
```

Execute the command file *infile.sql* by:

```
Start infile
```

Generating results to file

Use *Spool* to generate query results to a file.

e.g. To output result to *outfile.lst*, you can embed the query by *Spool File_name* and *Spool Off*:

```
Spool outfile
Select * from deposit;
Spool Off
```

Adding comment in command file

Use *Remark* or /*…*/ for adding comments

e.g.

```
Remark this is a comment line
/* This comment can span multiple lines */
```

Changing column format

Use *Column* to change column format for query result.

e.g. This is a query:

```
Column customer_name Heading 'customer name'
Column balance Format $99,990
Select customer_name, balance
from deposit
where balance > 3000;
```

Corresponding result:

customer name	balance
Chan Tai Man	$14, 560

Executing shell command

Use *Host Command_name* for execution of shell command.

e.g. Host vi temp.sql
e.g. Host cat result.lst

Exit Oracle server

Use *Exit* to leave from the server.