1.1 Using SSH Secure Shell 3.2.9 to login to CS Systems

1. Launch the ssh secure shell
2. Click on Quick Connect
3. Enter Host Name eg: luna.cs.uwindsor.ca
4. Enter User Name eg: nkulu
5. Port Number: 22 (default)
6. Authentication Method: <profile Setting> (default)
7. Then press ENTER
8. Enter Password eg: XXXXXX

When connected to the computer science system luna.cs.uwindsor.ca or sol.cs.uwindsor.ca you can use SQL*PLUS Oracle’s interactive Interface to the database server. SQL statements can be issued at the SQL> prompt and file containing SQL statements can be executed from within SQL*PLUS.

2.1 Oracle SQL*PLUS

Oracle’s SQL*PLUS program provides a convenient interactive environment with the Oracle Database Server. The user may type the commands directly at the SQL> prompt or have SQL*PLUS execute commands residing in operating system files.

Entering and Exiting Oracle SQL*PLUS

To enter the SQL*PLUS environment, the sqlplus program should be executed in one of the following two ways, where

user id > is the oracle user identification and
password > is the associated password:
- sqlplus <userid>@<connection string>
- sqlplus
The Oracle **userid** and **password** are different from the **userid** and **password** to get access to the operating system (luna.cs.uwindsor.ca/sol.cs.uwindsor.ca)

If the **sqlplus** program is invoked with only `<**userid**>`, the program prompts the user for the password; if it is invoked without any parameters, the program prompts for the `<**userid**>` and `<**password**>`

To Exit the SQL*PLUS environment, the **exit** or **quit** command must be entered at the SQL> prompt.

**SQL> quit**

**Executing Commands in SQL*PLUS**

Once the user is within the SQL*PLUS environment, the system will usually display the prompt SQL> and wait for the user commands. The user may enter three kinds of commands:

- SQL statements, to access the database
- PL/SQL blocks, also to access the database
- SQL*PLUS commands, for editing and storing SQL statements and PL/SQL blocks, setting options, and formatting query results.

SQL statements can be entered at the SQL> prompt. A statement may be broken into multiple lines. SQL*PLUS displays a line number (starting at 2) after the user presses the RETURN key to go to the next line. The SQL statement may be terminated in one of the three ways:

- With a semicolon (;), indicating to SQL*PLUS that it should execute the statement immediately.
- With a slash (/) on a line by itself, also indicating to SQL*PLUS that it should execute the statement immediately.
- With a blank line, indicating to SQL*PLUS that it should not do anything with the statement. The statement is stored in a buffer and can be executed at a later stage

The following is a screen capture of an SQL statement executed in SQL*PLUS from the CEZEIFE ACCOUNT

SQL Statement:
```sql
SQL> select course_no AS course#, c_title "course title", c_credit " course credit"
    2  from uw_courses;
```

Figure 2.2
Figure 2.3
You can also enter PL/SQL anonymous blocks at the SQL> prompt for execution and issue statements such as **create function** and **create procedure** at the SQL> prompt to create PL/SQL stored objects.

Eg: SQL Statement

```sql
/*
**This is a PL/SQL anonymous block
*/
DECLARE
    v_date DATE; -- variable v_date that will store the today date
BEGIN
    -- get the date from the system date and store it into the v_date
    SELECT SYSDATE
    INTO v_date
    FROM DUAL;
    -- print the today’s date on the screen
    DBMS_OUTPUT.PUT_LINE('Today''s date is ' || v_date);
END;
/*
Figure 2.4
```
The above PL/SQL anonymous block is executed by typing the “run” command at the SQL> prompt.

Besides SQL and PL/SQL, users can also enter SQL*PLUS commands at the SQL> prompt. These commands can manipulate SQL commands and PL/SQL blocks, format and print query results, and set various options for SQL*PLUS. SQL*PLUS must be entered in one line. If the command is long, it may be continued to the next line by typing the hyphen symbol (-) at the end of the line before pressing the RETURN key. Here is an example of an SQL*PLUS command that formats a column of the SQL query.

SQL Statement
SQL> select course_no AS course#, c_title "course title", c_credit " course credit"
   2 from uw_courses;

SQL> column c_credit format –
         99.99 heading “Course Credit”
SQL> run
1 select course_no, c_title, c_credit
2 from uw_courses
The **column** command formats a particular column in the current query (in this case the column is formatted and given a different name for display purposes). SQL*PLUS commands need not be terminated with semicolon.

The following are a few of the more commonly used SQL*PLUS commands:

- **describe** ([desc]) List the column definitions for a database table. The following is an example of the **describe** command.

### Figure 2.5

```sql
SQL> describe uw_courses
Name                      Null?  Type
--------------------------------------
COURSE_NO                  NOT NULL NUMBER(1)
C_TITLE                    NOT NULL VARCHAR2(20)
C_CREDIT                   NOT NULL NUMBER(1)
```

### Figure 2.6

```sql
SQL> describe uw_courses
Name                      Null?  Type
--------------------------------------
COURSE_NO                  NOT NULL NUMBER(1)
C_TITLE                    NOT NULL VARCHAR2(20)
C_CREDIT                   NOT NULL NUMBER(1)
```
Figure 2.7

```
SQL> help column

COLUMN
-----

Specifies display attributes for a given column, such as:
- column heading text
- column heading alignment
- NUMBER data format
- column data wrapping

Also lists the current display attributes for a single column or all columns.

COL([UUID] [{column | expr} [option...] ])
```

where option is one of the following clauses:
- ALI[AS] alias
- CLE[AR]
- FOLD_A[FTER]
- FOLD_B[ BEFORE]
- FOR[MAT] format
- HEA[DING] text
- JUS[TIFY] {L[eft] | C[ENTER] | R[ight]}
- LIKE {expr | alias}
- NEWL[INE]
- NEW_V[ALUE] variable
- NOPE[NT] | PRI[NT]
- NUL[L] text
- OLD_V[ALUE] variable
- UNI[CASE]
- WRA[PPED] | WOR[D_WRAPED] | TRU[NCATED]

SQL> help column

- **execute**. Execute a single PL/SQL statement. The syntax is

```
SQL> execute statement
```

- **help**. Gets online help for SQL*PLUS commands. For example,

```
SQL> help column
```

Will list the description of the column command. To get a list of all commands use the following command:

```
SQL> help <command name>
```
• **host.** Execute a host operating system command without leaving SQL*PLUS. For example,

```sql
SQL> host ls *.sql
```

Will list all the files in the current directory with a .sql extension. The exclamation key (!) may be used instead of the host command to achieve the same effect.

Figure 2.7

```sql
SQL> help host

HOST
----

Executes a host operating system command without leaving SQL*Plus.

HOST [command]
```

```sql
SQL> host ls *.sql
wsproject.sql
```

```
```

• **remark.** Used for comments. Any line beginning with keyword remark or rem or two hyphens (--) is treated as a comment and is ignored by SQL*PLUS.

• **Run.** Executes the SQL statement present in the buffer. The run command works the same as the slash command, except that it also displays the buffer contents before executing the statement in the buffer.

• **Set.** Sets SQL*PLUS system variables. Some of the more useful system variables include

```sql
SQL> set pause on;
SQL> set autoCommit on;
SQL> set echo on;
```

Setting **pause** to on causes SQL*PLUS to pause at the beginning of each page. The user must press RETURN key to see the next page.
Setting **autoCommit** to **on** informs Oracle to commit any changes to the database immediately after the SQL statement that has caused the changes, is executed.

Setting **echo** to **on** causes SQL*PLUS to list each commands in a file when the file is run with the start command. The names of other system variables, along with explanations, can be obtained by using **help** on the **set** command.

**Figure 2.8**

```sql
SQL> help spool

spool
-----
Stores query results in an operating system file, or sends the
file to a printer.
spool [file_name[.ext]] | OFF | OUT

SQL> help start

start
-----
Executes the contents of a command file.
start [arg ...]

start
-----
Starts an Oracle instance with several options, including mounting,
and opening a database.
start [force] [restrict] [pfile=filename] [exclusive]
[parallel [retry]] [shared [retry]]
[mount [dbname] | open [open_options] [dbname] | nomount]

where open_options has the following syntax:
READ {ONLY|WRITE [RECOVER]} | RECOVER

SQL>  
```
3.1 Buffer Manipulation Commands

The most recent command that is entered on the SQL prompt is stored in the SQL*PLUS buffer. It is possible to access, change, append to, and save the contents of the buffer. The SQL*PLUS buffer editing commands are listed below. All the editing commands (except for the list command) affect only one line, the current line. To make a particular line the current line, simply list that line by typing the line number the following SQL*PLUS session illustrates some of the editing commands.

Figure 3.1
SQL> desc uw_courses
Name                      | Null? | Type
-------------------------|-------|-------------------
COURSE_NO                 | NO    | NUMBER(7)
c_TITLE                  | NO    | VARCHAR2(100)
c_CREDIT                 | NO    | NUMBER(1)

SQL> select course_no, c_title
       2   from courses;
from courses
  *
ERROR at line 2:
ORA-00942: table or view does not exist

SQL> 2
     2* from courses
SQL> change /courses/uw_courses/
     2* from uw_courses
SQL> list
     1  select course_no, c_title
     2* from uw_courses
SQL> /

COURSE_NO
---------
c_TITLE
---------

360100
Key Concepts in Computer Science

360104
Computer Concepts for End-Users

360105
Programming in C for Beginners

COURSE_NO
---------

Figure 3.2
SQL> select course_no, c_title
   2  from uw_courses;
select course_no, c_title
   *
ERROR at line 1:
ORA-00904: invalid column name

SQL> l
1* select course_no, c_title
SQL> change /tl/title/
1* select course_no, c_title
SQL> list
1  select course_no, c_title
2* from uw_courses
SQL> /

<table>
<thead>
<tr>
<th>COURSE_NO</th>
<th>C_TITLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>360100</td>
<td>Key Concepts in Computer Science</td>
</tr>
<tr>
<td>360104</td>
<td>Computer Concepts for End-Users</td>
</tr>
<tr>
<td>360106</td>
<td>Programming in C for Beginners</td>
</tr>
<tr>
<td>360140</td>
<td>Introduction to Algorithms and Programming I</td>
</tr>
<tr>
<td>360141</td>
<td>Introduction to Algorithms and Programming II</td>
</tr>
<tr>
<td>360205</td>
<td>Introduction to the Internet</td>
</tr>
<tr>
<td>360207</td>
<td>Problem Solving and Information on the Internet</td>
</tr>
<tr>
<td>360212</td>
<td>Object-Oriented Programming using Java</td>
</tr>
<tr>
<td>360214</td>
<td>Computer Languages- Grammars and Translators</td>
</tr>
<tr>
<td>360221</td>
<td>Theoretical Foundations of Computer Science</td>
</tr>
<tr>
<td>360254</td>
<td>Data Structures and Algorithms</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>COURSE_NO</th>
<th>C_TITLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>360256</td>
<td>Systems Programming</td>
</tr>
<tr>
<td>360265</td>
<td>Computer Architecture I: Digital Design</td>
</tr>
<tr>
<td>360266</td>
<td>Computer Architecture II: Microprocessor Programming</td>
</tr>
<tr>
<td>360270</td>
<td>Advanced Website Design- Construction and Deployment</td>
</tr>
<tr>
<td>360275</td>
<td>Selected Topics I</td>
</tr>
<tr>
<td>360280</td>
<td>Practicum I</td>
</tr>
<tr>
<td>360298</td>
<td>Co-op Work Term I</td>
</tr>
</tbody>
</table>
Table 3.1 SQL*PLUS buffer editing commands.

<table>
<thead>
<tr>
<th>Command</th>
<th>Abbreviation</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Append *text</td>
<td>A text</td>
<td>Add text to the end of a line</td>
</tr>
<tr>
<td>Change *old/new</td>
<td>C /old/new</td>
<td>Change old to new in a line</td>
</tr>
<tr>
<td>Change *text</td>
<td>C /text</td>
<td>Delete text from a line</td>
</tr>
<tr>
<td>Clear buffer</td>
<td>C buff</td>
<td>Delete all lines</td>
</tr>
<tr>
<td>Del</td>
<td></td>
<td>Delete a line</td>
</tr>
<tr>
<td>Get file</td>
<td></td>
<td>Load contents of file named file into buffer</td>
</tr>
<tr>
<td>Input</td>
<td>I</td>
<td>Add one or more lines</td>
</tr>
<tr>
<td>Input *text</td>
<td>I text</td>
<td>Add a line consisting of text</td>
</tr>
<tr>
<td>List</td>
<td>L</td>
<td>List all lines in buffer</td>
</tr>
<tr>
<td>List n</td>
<td>L n or n</td>
<td>List one line and make it the current line</td>
</tr>
<tr>
<td>List *</td>
<td>L *</td>
<td>List the current line</td>
</tr>
<tr>
<td>List last</td>
<td>L last</td>
<td>List the last line</td>
</tr>
<tr>
<td>List m n</td>
<td>L m n</td>
<td>List lines m through n</td>
</tr>
<tr>
<td>Save file</td>
<td>Sav file</td>
<td>Save contents of buffer to file named file</td>
</tr>
</tbody>
</table>