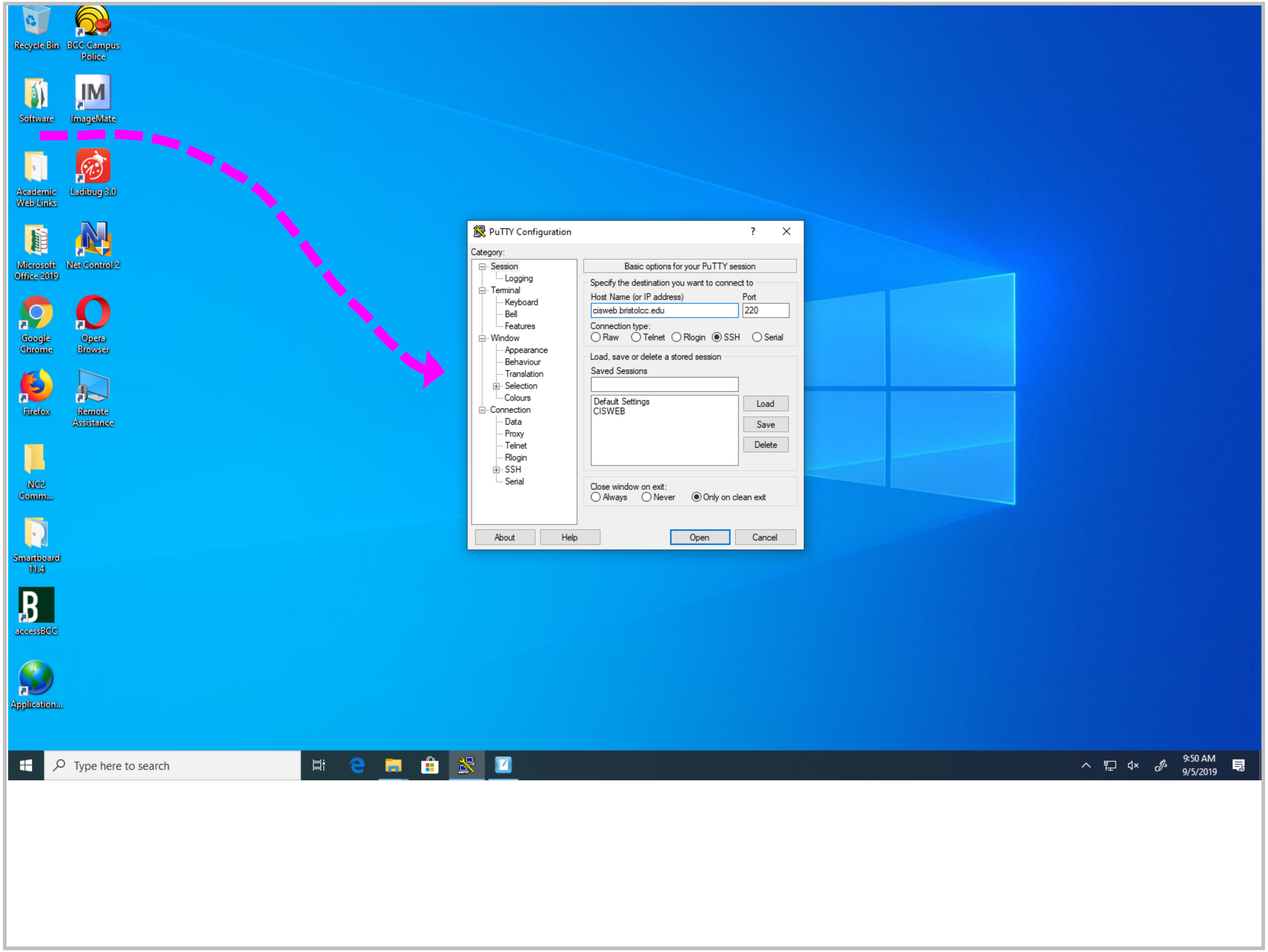


Title: Sep 5-9:48 AM (1 of 20)

File Edit View History Bookmarks Tools Help

CIS150/50 Course Page

www.pgrocer.net/PFGCIS150.html

Oracle and SQL
CIS150/50

About This Course:
- **Need help or support?**
- **Introduction to CIS150**
- **Syllabus**
- **Withdrawal Policy as of Fall 2017**

Site Resources:
- **Notes and Handouts**
- **Presentations**
- **SmartBoard Presentations**
- **Audio**

Weekly Schedule:
- **Weekly schedule - CIS150/50**

Using Oracle:
- **Installation and login**

Links:
- **Links & Tutorials**

	Thursday 2-5 in K101 Jayson Rios from the tutoring center is also available on Monday 11-3 and Wednesday 11-1 in New Bedford.

Week #1
Week of September 2nd
	I recommend one of the text books in the syllabus, but a textbook is not required. There is enough material on the web and in my notes that I decided to make the book a recommendation. Please note that many students have found it useful to have a copy of the text. I do recommend that you get one of these books or another book dealing with Oracle SQL (we are using version 11g or 12c this semester) as a reference. We will not do chapter by chapter coverage, the text gives you additional information on topics and a reference if you have questions.
	ional database information
	LOGIN information:
	You need to decide if you want to download Oracle, use the BCC labs or login from home using Putty.
	Note that there is a reference to help you download and instal under Using Oracle: Installation and login.
	Downloading Be very careful when you install, Oracle leaves a lot of footprints that makes reinstalling very difficult.
	You should have received the login information you need for this course. If not, let me know.
	PuTTY is available in the labs. Two sites that can be used to download PuTTY are:
	PuTTY
	PuTTY
	Using Putty (copy and paste)
	Basic steps to login to putty and get to Oracle:
	1) Start putty and put in the hostname as cisweb.bristolcc.edu and the port as 220
	2) At the login screen enter your username and your Linux password.
	3) When you see that you are at cisweb, enter sqlplus12c.sh
	4) You then connect to Oracle and need to enter your username and Oracle password
	5) You should see the SQL prompt
	Now you can create a table. Note that after you create tables, to see existing tables select table_name from user_tables.
	Accompanying step-by-step notes
	First we will focus on getting into Oracle and then these are the notes that we will start to cover this week:
	Introduction to SQL in Oracle
	Create a table in SQL
	SQL Conditions
	These are the accompanying presentations:
	Presentations:
	Introduction to SQL in Oracle
	    Separate speaker notes for Introduction to SQL in Oracle
	Create a table in SQL
	    Separate speaker notes for Create a table in SQL
	SQL Conditions
	    Separate speaker notes for SQL Conditions
	NOTES: Note: You can use the committ command to make sure your work gets saved.
	Copy from SQLPlus so you can paste it onto the document you are submitting as your assignent
	Assignments:
	I strongly urge you to do the COMMIT; after the table is created.
	Oracle assignment on introductory material (html)
	Oracle assignment on introductory material (doc)
	Comment on how I want the assignment: You should take the commands you do in SQL and the results and paste them into a word document. It should be similiar to the way I prepared the notes. You then send me the word document. I do not want you to show the screen itself or pictures of the screen - they are really hard to read!

Send e-mail to Mrs. Grocer:

Type here to search

9:53 AM
9/5/2019

Title: Sep 5-9:53 AM (2 of 20)

File Edit View History Bookmarks Tools Help

CIS150/50 Course Page

pgrocer@cisweb:~

```
| *****WARNING*****
| This is a private computer system.  Only authorized users are allowed
| access to this system. Use of this system is subject to the terms of the
| Bristol Community College Policy for the Responsible Use of Information
| Technology, available at:
| http://www.bristolcc.edu/responsibleusepolicy/
| *****WARNING*****
|
| *****IMPORTANT*****
| The college is not responsible for any user data on this system including
| but not limited to student work and assignments.  Users of this system
| are fully responsible for backing up their data.
| *****IMPORTANT*****
  End of banner message from server
  pgrocer@cisweb.bristolcc.edu's password:
Last failed login: Wed Sep  4 17:26:07 EDT 2019 from pool-96-230-44-57.bstnma.fi
os.verizon.net on ssh:notty
There was 1 failed login attempt since the last successful login.
Last login: Wed Sep  4 14:36:33 2019 from pool-173-76-233-11.bstnma.fios.verizon
.net
[pgrocer@cisweb ~]$ sqlplus12c.sh
Connecting to Oracle 12c server
Enter Oracle username: pgrocer
Enter password:
```

About 7
Need I
Introd
Syllab
Withd
Site Re
Notes
Preser
Smart
Audio
Weekly S
Weekl
Using (
Install
Links:
Links

The information at this site is in draft form and is the property of Priscilla Grocer. The information may not be duplicated or distributed without her permission. ©Copyright 2019

101
he tutoring center is also available on Monday 11-3 and Wednesday 11-1 in New Bedford.
of the text books in the syllabus, but a textbook is not required. There is enough material on the web and in my notes that I decided to make
endation. Please note that many students have found it useful to have a copy of the text. I do recommend that you get one of these books or
ng with Oracle SQL (we are using version 11g or 12c this semester) as a reference. We will not do chapter by chapter coverage, the text
l information on topics and a reference if you have questions.
rmation
on:
if you want to download Oracle, use the BCC labs or login from home using Putty.
reference to help you download and instal under Using Oracle: Installation and login.
ery careful when you install, Oracle leaves a lot of footprints that makes reinstalling very difficult.
ceived the login information you need for this course. If not, let me know.
in the labs. Two sites that can be used to download PuTTY are:

and paste)
to putty and get to Oracle:
ut in the hostname as cisweb.bristolcc.edu and the port as 220
en enter your username and your Linux password.
at you are at cisweb, enter sqlplus12c.sh
t to Oracle and need to enter your username and Oracle password
5) You should see the SQL prompt

Week #1 Week of September 2nd	Now you can create a table. Note that after you create tables, to see existing tables select table_name from user_tables. **Accompanying step-by-step notes** First we will focus on getting into Oracle and then these are the notes that we will start to cover this week: **Introduction to SQL in Oracle** **Create a table in SQL** **SQL Conditions** These are the accompanying presentations: **Presentations:** **Introduction to SQL in Oracle** **Separate speaker notes for Introduction to SQL in Oracle** **Create a table in SQL** **Separate speaker notes for Create a table in SQL** **SQL Conditions** **Separate speaker notes for SQL Conditions** **NOTES:** Note: You can use the committ command to make sure your work gets saved. **Copy from SQLPlus so you can paste it onto the document you are submitting as your assignent** ***Assignments:*** I strongly urge you to do the COMMIT; after the table is created. **Oracle assignment on introductory material (html)** **Oracle assignment on introductory material (doc)** Comment on how I want the assignment: You should take the commands you do in SQL and the results and paste them into a word document. It should be similiar to the way I prepared the notes. You then send me the word document. I do not want you to show the screen itself or pictures of the screen - they are really hard to read!

Send e-mail to Mrs. Grocer:

Type here to search

9:54 AM
9/5/2019

Title: Sep 5-9:54 AM (3 of 20)

File Edit View History Bookmarks Tools Help

Create a table

```
pgrocer@cisweb:~
Copyright (c) 1982, 2013, Oracle.  All rights reserved.

Last Successful login time: Wed Sep 04 2019 14:37:01 -04:00

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing opt
ions

SQL> create table whatever
  2  (idno varchar2(4),
  3  name varchar2(15),
  4  amt number(5));

Table created.

SQL> desc whatever;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 IDNO                                               VARCHAR2(4)
 NAME                                               VARCHAR2(15)
 AMT                                                NUMBER(5)

SQL>
```

Creat

This handou ...icated elements at that time.

The CREAT ... The table will contain an identification number, a name, a code, a date and an amount without decimals and an amount with decimals. The data types t

Data T
VARCHAR
CHAR(size
NUMBER(
DATE

The CREAT ... listed inside a set of parenthesis with the column name and type. The columns are separated by commas. The example described is illustrated b

SQL COD

```
SQL> CREATE TABLE first_table
  2  (idno varchar2(4),
  3  name varchar2(15),
  4  first_code char(2),
  5  first_date date,
  6  first_wholeamt number(3),
  7  first_decimal number(6,2));

Table created.
```

To see the layout of the table that was created, you can use the DESCRIBE command as shown below:

SQL CODE:

```
SQL> DESCRIBE first_table;
 Name                            Null?    Type
 ------------------------------- -------- ----
 IDNO                                     VARCHAR2(4)
 NAME                                     VARCHAR2(15)
 FIRST_CODE                               CHAR(2)
 FIRST_DATE                               DATE
 FIRST_WHOLEAMT                           NUMBER(3)
 FIRST_DECIMAL                            NUMBER(6,2)
```

To put data into the table, you use the INSERT command. The format is:

INSERT INTO table
values(list the fields that are going into each column for this row - varchar2, char and date fields must be enclosed in single quotes - numeric columns are not enclosed in quotes)

Type here to search

9:58 AM
9/5/2019

Title: Sep 5-9:58 AM (4 of 20)

File Edit View History Bookmarks Tools Help

Create a table

pgrocer@cisweb:~

```
 ---------------------------------------- -------- ----------------------------
 IDNO                                              VARCHAR2(4)
 NAME                                              VARCHAR2(15)
 AMT                                               NUMBER(5)

SQL> desc whatever;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 IDNO                                               VARCHAR2(4)
 NAME                                               VARCHAR2(15)
 AMT                                                NUMBER(5)

SQL> insert into whatever
  2  values ('1111','Susan Ash',500);

1 row created.

SQL> select * from whatever;

IDNO NAME                   AMT
---- --------------- ----------
1111 Susan Ash              500

SQL>
```

SQL CODE

```
SQL> CREAT
  2  (idno
  3  name
  4  first
  5  first
  6  first
  7  first

Table crea
```

To see the la

SQL CODE

```
SQL> DESCR
 Name
 ---------
 IDNO
 NAME
 FIRST_COD
 FIRST_DAT
 FIRST_WHOLEAMT                            NUMBER(3)
 FIRST_DECIMAL                             NUMBER(6,2)
```

To put data into the table, you use the INSERT command. The format is:

INSERT INTO table
values(list the fields that are going into each column for this row - varchar2, char and date fields must be enclosed in single quotes - numeric columns are not enclosed in quotes)

SQL CODE:

```
SQL> INSERT INTO first_table
  2  values('1111','Susan Johnson','CI','12-JAN-96',123,12.45);

1 row created.

SQL> INSERT INTO first_table
  2  values('2222','John Costa','CI','15-FEB-97',654,765.23);

1 row created.

SQL> INSERT INTO first_table
  2  values('3333','Linda Higgins','CF','23-MAR-95',12,1.48);

1 row created.

SQL> INSERT INTO first_table
  2  values('4444','Stephen Woods','CI','30-APR-98',4,4.59);

1 row created.
```

These four records have now been inserted in the table. We can now list the table.

Type here to search

10:02 AM
9/5/2019

File Edit View History Bookmarks Tools Help

Create a table

pgrocer@cisweb:~

```
  GNU nano 2.3.1              File: afiedt.buf                        Modified

select * from whatever
/

                                [ Read 2 lines ]
^G Get Help   ^O WriteOut   ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit       ^J Justify    ^W Where Is   ^V Next Page  ^U UnCut Text ^T To Spell
```

^O writeout puts your correction in the buffer
You then use ^X to exit and back at the prompt
you use / to execute.

SQL CODE

```
SQL> CREAT
  2  (idno
  3  name
  4  first
  5  first
  6  first
  7  first

Table crea
```

To see the la

SQL CODE

```
SQL> DESCR
 Name
 ---------
 IDNO
 NAME
 FIRST_COD
 FIRST_DAT
 FIRST_WHOLEAMT                          NUMBER(3)
 FIRST_DECIMAL                           NUMBER(6,2)
```

To put data into the table, you use the INSERT command. The format is:

INSERT INTO table
values(list the fields that are going into each column for this row - varchar2, char and date fields must be enclosed in single quotes - numeric columns are not enclosed in quotes)

SQL CODE:

```
SQL> INSERT INTO first_table
  2  values('1111','Susan Johnson','CI','12-JAN-96',123,12.45);

1 row created.

SQL> INSERT INTO first_table
  2  values('2222','John Costa','CI','15-FEB-97',654,765.23);

1 row created.

SQL> INSERT INTO first_table
  2  values('3333','Linda Higgins','CF','23-MAR-95',12,1.48);

1 row created.

SQL> INSERT INTO first_table
  2  values('4444','Stephen Woods','CI','30-APR-98',4,4.59);

1 row created.
```

These four records have now been inserted in the table. We can now list the table.

Type here to search

10:04 AM
9/5/2019

File Edit View History Bookmarks Tools Help

Create a table

pgrocer@cisweb:~

```
 GNU nano 2.3.1              File: afiedt.buf                    Modified

select * from whatever
/





File Name to Write: afiedt.buf
^G Get Help          M-D DOS Format       M-A Append          M-B Backup File
^C Cancel            M-M Mac Format       M-P Prepend
```

SQL COD

```
SQL> CREAT
  2  (idnc
  3  name
  4  first
  5  first
  6  first
  7  first

Table crea
```

To see the la

SQL COD

```
SQL> DESCR
 Name
 ---------
 IDNO
 NAME
 FIRST_COI
 FIRST_DAT
 FIRST_WHOLEAMT                            NUMBER(3)
 FIRST_DECIMAL                             NUMBER(6,2)
```

To put data into the table, you use the INSERT command. The format is:

INSERT INTO table
values(list the fields that are going into each column for this row - varchar2, char and date fields must be enclosed in single quotes - numeric columns are not enclosed in quotes)

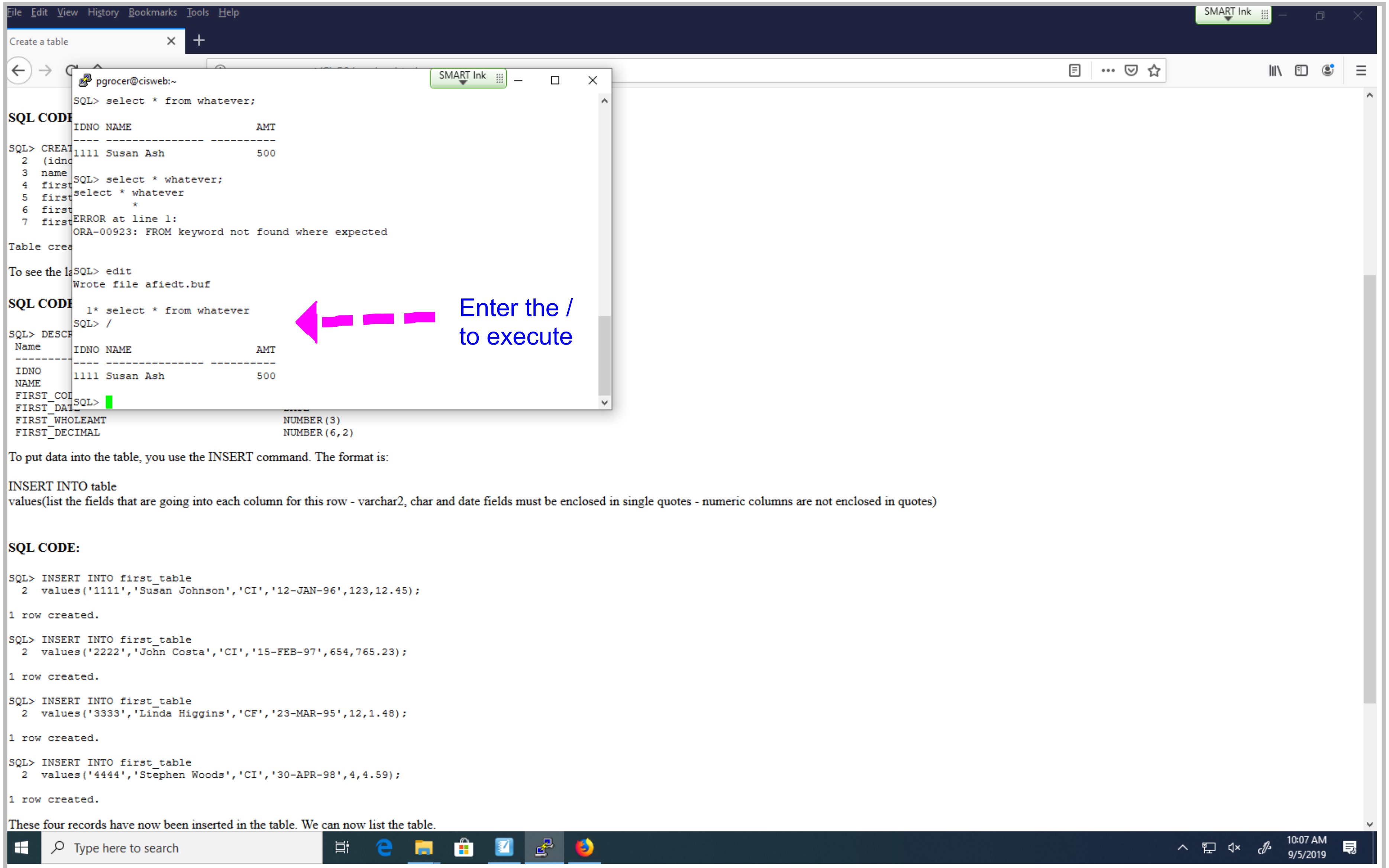
SQL CODE:

```
SQL> INSERT INTO first_table
  2  values('1111','Susan Johnson','CI','12-JAN-96',123,12.45);

1 row created.

SQL> INSERT INTO first_table
  2  values('2222','John Costa','CI','15-FEB-97',654,765.23);

1 row created.

SQL> INSERT INTO first_table
  2  values('3333','Linda Higgins','CF','23-MAR-95',12,1.48);

1 row created.

SQL> INSERT INTO first_table
  2  values('4444','Stephen Woods','CI','30-APR-98',4,4.59);

1 row created.
```

These four records have now been inserted in the table. We can now list the table.

Type here to search

10:05 AM
9/5/2019


```
SQL> select * from whatever;

IDNO NAME                   AMT
---- --------------- ----------
1111 Susan Ash              500

SQL> select * whatever;
select * whatever
         *
ERROR at line 1:
ORA-00923: FROM keyword not found where expected


SQL> edit
Wrote file afiedt.buf

  1* select * from whatever
SQL> /

IDNO NAME                   AMT
---- --------------- ----------
1111 Susan Ash              500

SQL>
```


SQL CODE:

```
SQL> DESCR
 Name
 ---------
 IDNO
 NAME
 FIRST_COD
 FIRST_DAT
 FIRST_WHOLEAMT                            NUMBER(3)
 FIRST_DECIMAL                             NUMBER(6,2)
```

To put data into the table, you use the INSERT command. The format is:

INSERT INTO table
values(list the fields that are going into each column for this row - varchar2, char and date fields must be enclosed in single quotes - numeric columns are not enclosed in quotes)

SQL CODE:

```
SQL> INSERT INTO first_table
  2  values('1111','Susan Johnson','CI','12-JAN-96',123,12.45);

1 row created.

SQL> INSERT INTO first_table
  2  values('2222','John Costa','CI','15-FEB-97',654,765.23);

1 row created.

SQL> INSERT INTO first_table
  2  values('3333','Linda Higgins','CF','23-MAR-95',12,1.48);

1 row created.

SQL> INSERT INTO first_table
  2  values('4444','Stephen Woods','CI','30-APR-98',4,4.59);

1 row created.
```

These four records have now been inserted in the table. We can now list the table.

Title: Sep 5-10:07 AM (8 of 20)

File Edit View History Bookmarks Tools Help

Notes for CIS50

www.pgrocer.net/Cis50/cis50notes.html

Notes for CIS50 - Oracle and SQL

Notes and handouts	Link to retrieve
Introduction to Relational Databases	Student Payroll
Introduction to Oracle and SQL	Introduction to SQL in Oracle Create a table in SQL SQL Conditions Working with SQL
SQL and SQL*Plus	More on maintaining an SQL table SQL functions More SQL functions Group SQL functions Introduction to multiple tables (relational) - joins Relational database examples Oracle Join statement Subqueries More on subqueries Using input variables More on variables with Oracle's SQL*Plus Table maintenance revisited (again) Continuation of table maintenance revisited (again) Introduction to views Introduction to reports Indexes in Oracle - An introduction More on Primary and Foreign keys Notes on key questions More on views Information on database objects Script for input data
Older versions	Introduction to user security Copying a Table from DEMO Exporting an Oracle table (older version)
PL/SQL	Introduction to PL/SQL More examples of inner/outer procedures Introduction to IF statements More on PL/SQL IF statements Introduction to loops Data manipulation in PL/SQL - introduction Record and table types Introduction to Cursors Example 1 using cursors Continuing example 1 using cursors Another set of cursor examples Introduction to exceptions Introduction to procedures

www.pgrocer.net/Cis50/oracle1.html

Type here to search

10:09 AM
9/5/2019

Introduction to Oracle - SQL

In introducing SQL, I will be using three tables that form a contribution system. The first table is donor, the second is donation and the third is drive:

Donor (<u>idno</u>, name, stadr, city, state, zip, datefst, yrgoal, contact)
Donation (<u>idno, driveno, datecont</u>, amtcont)
Drive (<u>driveno</u>, drname, drchair, goal, ytdtotal, lastyrtot)

SELECT:

The SELECT statement can be used to obtain information from a table. The select can retrieve selected rows, specified columns or you can join information from two or more tables using a link to specified columns. Selection is the name given to selecting rows based on specified criteria. Projection is the name given to specifying certain columns for your query. Join is the name given to linking multiple tables through common columns.

The SELECT statement must include a clause which tells what columns to show and a from clause that designates the table to be used. A where clause can be used to specify a condition that results in only selected rows being displayed. SQL is not case sensitive and the SQL statement can be written on one or many lines. Frequently lines and indenting of clauses are used to enhance readability.

SELECT {columns}
FROM {table};

Example: I want to select all columns from the table called donor. To select all columns you can put an * after the select statement. I can code on one line or two as shown.

SELECT *
FROM donor;

SQL CODE:

```
SQL> select * from donor;

IDNO  NAME            STADR           CITY       ST ZIP   DATEFST      YRGOAL CONTACT
----- --------------- --------------- ---------- -- ----- --------- --------- ------------
11111 Stephen Daniels 123 Elm St      Seekonk    MA 02345 03-JUL-98       500 John Smith
12121 Jennifer Ames   24 Benefit St   Providence RI 02045 24-MAY-97       400 Susan Jones
22222 Carl Hersey     24 Benefit St   Providence RI 02045 03-JAN-98       400 Susan Jones

SQL>
```

After keying the semi-colon the user can press enter to execute. You can also execute a statement by keying a slash at the SQL prompt or issuing the RUN command at the SQL prompt.

To specify certain fields for selection, you list the fields by name separated by a comma. The order that you use in listing determines the order in the results. For example: idno, name means idno lists first while name, idno means name lists first.

SELECT idno, name, city
FROM donor;

SQL CODE:

```
SQL> select idno, name, city from donor;
```

SELECT *
FROM donor;

SQL CODE:

```
SQL> select * from donor;

IDNO  NAME            STADR           CITY       ST ZIP   DATEFST      YRGOAL CONTACT
----- --------------- --------------- ---------- -- ----- --------- --------- ------------
11111 Stephen Daniels 123 Elm St      Seekonk    MA 02345 03-JUL-98       500 John Smith
12121 Jennifer Ames   24 Benefit St   Providence RI 02045 24-MAY-97       400 Susan Jones
22222 Carl Hersey     24 Benefit St   Providence RI 02045 03-JAN-98       400 Susan Jones

SQL>
```

After keying the semi-colon the user can press enter to execute. You can also execute a statement by keying a slash at the SQL prompt or issuing the RUN command at the SQL prompt.

To specify certain fields for selection, you list the fields by name separated by a comma. The order that you use in listing determines the order in the results. For example: idno, name means idno lists first while name, idno means name lists first.

SELECT idno, name, city
FROM donor;

SQL CODE:

Developing a column alias as the heading for a particular output.

```
SQL> select idno, name, city from donor;

IDNO  NAME            CITY
----- --------------- ----------
11111 Stephen Daniels Seekonk
12121 Jennifer Ames   Providence
22222 Carl Hersey     Providence

SQL>
```

When showing columns, numeric data is justified right and date and character data are justified to the left. Justification applies to both the data and the column headings. The default for column headings is the name of the field displayed in upper case. The developer may specify a column alias which will serve as the column heading. For example, this would show the headings as ID, NAME, STREET instead of IDNO, NAME, STADR. The AS is optional.

SELECT idno AS ID, name, stadr AS STREET
FROM donor;

SELECT idno ID, name, stradr STREET
FROM donor;

SQL CODE:

```
SQL> select idno AS ID, name, stadr AS STREET
  2  FROM donor;

ID    NAME            STREET
----- --------------- ---------------
```

Type here to search

10:13 AM
9/5/2019

File Edit View History Bookmarks Tools Help

CIS50 - Introduction to Oracle

pgrocer@cisweb:~

```
SQL> select * whatever;
select * whatever
         *
ERROR at line 1:
ORA-00923: FROM keyword not found where expected


SQL> edit
Wrote file afiedt.buf

  1* select * from whatever
SQL> /

IDNO NAME                   AMT
---- --------------- ----------
1111 Susan Ash              500

SQL> select idno ID, name, amt from whatever;

ID   NAME                   AMT
---- --------------- ----------
1111 Susan Ash              500

SQL>
```

FROM don

SQL CODE

```
SQL> selec

IDNO  NAME
----- ---
11111 Step
12121 Jenn
22222 Carl

SQL>
```

When show... ...tion applies to both the data and the column headings. The default for column headings is the name of the field displayed in upper case. The developer m... ...eadings as ID, NAME, STREET instead of IDNO, NAME, STADR. The AS is optional.

SELECT id
FROM don

SELECT id
FROM don

SQL CODE:

```
SQL> select idno AS ID, name, stadr AS STREET
  2  FROM donor;

ID    NAME            STREET
----- --------------- ---------------
11111 Stephen Daniels 123 Elm St
12121 Jennifer Ames   24 Benefit St
22222 Carl Hersey     24 Benefit St
```

You can also specify a multiword heading or a mixed case heading by enclosing it in double quotes:

SELECT idno "ID #", name "Name", stadr "Street Adr"
FROM donor;

SQL CODE:

```
SQL> SELECT idno "ID #", name "Name", stadr "Street Adr"
  2  FROM donor;

ID #  Name            Street Adr
----- --------------- ---------------
11111 Stephen Daniels 123 Elm St
12121 Jennifer Ames   24 Benefit St
22222 Carl Hersey     24 Benefit St

SQL>
```

Type here to search

10:16 AM
9/5/2019

Title: Sep 5-10:16 AM (12 of 20)

```
SQL> SELECT idno "ID #", name "Name", stadr "Street Adr"
  2  FROM donor;

ID #  Name            Street Adr
----- --------------- ---------------
11111 Stephen Daniels 123 Elm St
12121 Jennifer Ames   24 Benefit St
22222 Carl Hersey     24 Benefit St

SQL>
```

The SELECT statement can also include Arithmetic expressions:

Operator	Process
+	Add
-	Subtraction
*	Multiplication
/	Division

One of the fields on the donor field is yrgoal. If I want to display a column with monthly goal I would divide by 12. The resulting header would be YRGOAL/12.

SELECT idno, name, yrgoal/12
FROM donor;

SQL CODE:

```
SQL> SELECT idno, name, yrgoal/12
  2  FROM donor;

IDNO  NAME            YRGOAL/12
----- --------------- ---------
11111 Stephen Daniels 41.666667
12121 Jennifer Ames   33.333333
22222 Carl Hersey     33.333333

SQL>
```

If I want the header to say Month Goal then I would include the following:

SELECT idno, name, yrgoal/12 "Month Goal"
FROM donor;

Calculations follow the standard order of operation:

- Evaluation works from left to right
- Multiplication and division first
- Addition and subtraction second
- Parentheses are evaluated/resolved first (use to change order of operation)

For example: If you want to add 100 to the yrgoal and then divide by 12 you use the parenthesis to do the addition before the division.

SELECT idno, name, (yrgoal + 100)/12 "New Month Goal"
FROM donor;

FROM donor;

SQL CODE:

```
SQL> SELECT idno, name, yrgoal/12
  2  FROM donor;

IDNO  NAME            YRGOAL/12
----- --------------- ---------
11111 Stephen Daniels 41.666667
12121 Jennifer Ames   33.333333
22222 Carl Hersey     33.333333

SQL>
```

If I want the header to say Month Goal then I would include the following:

SELECT idno, name, yrgoal/12 "Month Goal"
FROM donor;

Calculations follow the standard order of operation:

- Evaluation works from left to right
- Multiplication and division first
- Addition and subtraction second
- Parentheses are evaluated/resolved first (use to change order of operation)

For example: If you want to add 100 to the yrgoal and then divide by 12 you use the parenthesis to do the addition before the division.

SELECT idno, name, (yrgoal + 100)/12 "New Month Goal"
FROM donor;

SQL CODE:

```
SQL> 1  SELECT idno, name, (yrgoal + 100)/12 "New Month Goal"
  2* FROM donor
SQL> /

IDNO  NAME            New Month Goal
----- --------------- --------------
11111 Stephen Daniels             50
12121 Jennifer Ames        41.666667
22222 Carl Hersey          41.666667
```

Note that in the example above there is a / at the SQL prompt. This is because I edited the select statement in notepad and then executed it when I returned. The / causes the statement to be executed.

NULL values are values that are unassigned. A null value should not be confused with a zero which is a number or a space which is a character. Null means there is not a data value for a particular field. When a table is created the user can specify that a field cannot be null. Note also that a field defined as a primary key can not be null. When an arithmetic expression contains a null value, the result of the calculation evaluates to null. Looking at the example of dividing yrgoal by 12, any donor with a null value in yrgoal would have a null value as their monthly goal.

SELECT idno, yrgoal, yrgoal/12
FROM donor;

yrgoal would have a null value as their monthly goal.

SELECT idno, yrgoal, yrgoal/12
FROM donor;

SQL CODE:

```
SQL> SELECT idno, yrgoal, yrgoal/12
  2  FROM donor;

IDNO     YRGOAL YRGOAL/12
----- --------- ---------
11111       500 41.666667
12121       400 33.333333
22222
```

CONCATENATION strings columns or character strings together as a character expression. The concatenation operator is two vertical bars ||.

If the city was PROVIDENCE and the state was RI:
SELECT city || state would result in PROVIDENCERI while
SELECT city || ', ' || state would result in PROVIDENCE, RI

SQL CODE:

```
SQL> SELECT city || state
  2  FROM donor;

CITY||STATE
------------
SeekonkMA
ProvidenceRI
ProvidenceRI

SQL>

SQL> SELECT city || ', ' || state
  2  FROM donor;

CITY||','||STA
--------------
Seekonk, MA
Providence, RI
Providence, RI
```

Another example:

SELECT name || ' has a yearly goal of ' || yrgoal
FROM donor;

In this case, notice that there is a space after the quote so that there will be a space between name and has and there is a space before the quote which means there will be a space between of and the yearly goal.

For Stephen Daniels with a goal of 500 you would see:
Stephen Daniels has a yearly goal of 500

File Edit View History Bookmarks Tools Help

CIS50 - Introduction to Oracle

www.pgrocer.net/Cis50/oracle1.html

```
Jennifer Ames has a yearly goal of 400
Carl Hersey has a yearly goal of 400
```

If you want to eliminate duplicate rows, you can use the DISTINCT keyword. For example, if you want to see the contacts but you only want to see each contact once, you would use the following:

SELECT DISTINCT contact
FROM donor;

SQL CODE:

```
SQL> SELECT DISTINCT contact
  2  FROM donor;

CONTACT
------------
John Smith
Susan Jones

SQL>
```

SORT:

The ORDER BY clause is used to SORT rows in a specified order. Ascending order is the default (it can be indicated by ASC after the column name you are sorting on). If you want to sort in descending order, the DESC clause is placed after the column name you are sorting on. In the example below I have chosen to sort by datefst. Notice that it is in ascending order since that is the default.

SQL CODE:

```
SQL> SELECT *
  2  FROM donor
  3  ORDER BY datefst;

IDNO  NAME            STADR           CITY       ST ZIP   DATEFST      YRGOAL CONTACT
----- --------------- --------------- ---------- -- ----- --------- --------- ------------
23456 Susan Ash       21 Main St      Fall River MA 02720 04-MAR-92       100 Amy Costa
33333 Nancy Taylor    26 Oak St       Fall River MA 02720 04-MAR-92        50 John Adams
12121 Jennifer Ames   24 Benefit St   Providence RI 02045 24-MAY-97       400 Susan Jones
22222 Carl Hersey     24 Benefit St   Providence RI 02045 03-JAN-98           Susan Jones
34567 Robert Brooks   36 Pine St      Fall River MA 02720 04-APR-98        50 Amy Costa
11111 Stephen Daniels 123 Elm St      Seekonk    MA 02345 03-JUL-98       500 John Smith

6 rows selected.
```

Specifying DESC after datefst caused the sort to be done in descending order.

SQL CODE:

```
SQL> SELECT *
  2  FROM donor
  3  ORDER BY datefst DESC;

IDNO  NAME            STADR           CITY       ST ZIP   DATEFST      YRGOAL CONTACT
----- --------------- --------------- ---------- -- ----- --------- --------- ------------
11111 Stephen Daniels 123 Elm St      Seekonk    MA 02345 03-JUL-98       500 John Smith
```

Type here to search

10:27 AM
9/5/2019

SORT:

The ORDER BY clause is used to SORT rows in a specified order. Ascending order is the default (it can be indicated by ASC after the column name you are sorting on). If you want to sort in descending order, the DESC clause is placed after the column name you are sorting on. In the example below I have chosen to sort by datefst. Notice that it is in ascending order since that is the default.

SQL CODE:

```
SQL> SELECT *
  2  FROM donor
  3  ORDER BY datefst;

IDNO  NAME            STADR           CITY       ST ZIP   DATEFST      YRGOAL CONTACT
----- --------------- --------------- ---------- -- ----- --------- --------- ------------
23456 Susan Ash       21 Main St      Fall River MA 02720 04-MAR-92       100 Amy Costa
33333 Nancy Taylor    26 Oak St       Fall River MA 02720 04-MAR-92        50 John Adams
12121 Jennifer Ames   24 Benefit St   Providence RI 02045 24-MAY-97       400 Susan Jones
22222 Carl Hersey     24 Benefit St   Providence RI 02045 03-JAN-98           Susan Jones
34567 Robert Brooks   36 Pine St      Fall River MA 02720 04-APR-98        50 Amy Costa
11111 Stephen Daniels 123 Elm St      Seekonk    MA 02345 03-JUL-98       500 John Smith

6 rows selected.
```

Specifying DESC after datefst caused the sort to be done in descending order.

SQL CODE:

```
SQL> SELECT *
  2  FROM donor
  3  ORDER BY datefst DESC;

IDNO  NAME            STADR           CITY       ST ZIP   DATEFST      YRGOAL CONTACT
----- --------------- --------------- ---------- -- ----- --------- --------- ------------
11111 Stephen Daniels 123 Elm St      Seekonk    MA 02345 03-JUL-98       500 John Smith
34567 Robert Brooks   36 Pine St      Fall River MA 02720 04-APR-98        50 Amy Costa
22222 Carl Hersey     24 Benefit St   Providence RI 02045 03-JAN-98           Susan Jones
12121 Jennifer Ames   24 Benefit St   Providence RI 02045 24-MAY-97       400 Susan Jones
23456 Susan Ash       21 Main St      Fall River MA 02720 04-MAR-92       100 Amy Costa
33333 Nancy Taylor    26 Oak St       Fall River MA 02720 04-MAR-92        50 John Adams

6 rows selected.
```

In the example below the sort is being done on two fields. The field that is listed first is the primary sort field and other fields are secondary. If there are multiple secondary the order is from left to right. The primary sort is by state and since no order was given it is sorted in the default which is ascending order. The secondary sort is on datefst and this sort is done in descending order. Essentially the rows are shown in order by state and datefst within state.

SQL CODE:

```
SQL> SELECT *
  2  FROM donor
  3  ORDER by state, datefst DESC;

IDNO  NAME            STADR           CITY       ST ZIP   DATEFST      YRGOAL CONTACT
----- --------------- --------------- ---------- -- ----- --------- --------- ------------
11111 Stephen Daniels 123 Elm St      Seekonk    MA 02345 03-JUL-98       500 John Smith
```

File Edit View History Bookmarks Tools Help

CIS50 - Introduction to Oracle

www.pgrocer.net/Cis50/oracle1.html

```
33333 Nancy Taylor    26 Oak St       Fall River MA 02720 04-MAR-92        50 John Adams

6 rows selected.
```

In the example below the sort is being done on two fields. The field that is listed first is the primary sort field and other fields are secondary. If there are multiple secondary the order is from left to right. The primary sort is by state and since no order was given it is sorted in the default which is ascending order. The secondary sort is on datefst and this sort is done in descending order. Essentially the rows are shown in order by state and datefst within state.

SQL CODE:

```
SQL> SELECT *
  2  FROM donor
  3  ORDER by state, datefst DESC;

IDNO  NAME            STADR           CITY       ST ZIP   DATEFST      YRGOAL CONTACT
----- --------------- --------------- ---------- -- ----- --------- --------- ------------
11111 Stephen Daniels 123 Elm St      Seekonk    MA 02345 03-JUL-98       500 John Smith
34567 Robert Brooks   36 Pine St      Fall River MA 02720 04-APR-98        50 Amy Costa
23456 Susan Ash       21 Main St      Fall River MA 02720 04-MAR-92       100 Amy Costa
33333 Nancy Taylor    26 Oak St       Fall River MA 02720 04-MAR-92        50 John Adams
22222 Carl Hersey     24 Benefit St   Providence RI 02045 03-JAN-98           Susan Jones
12121 Jennifer Ames   24 Benefit St   Providence RI 02045 24-MAY-97       400 Susan Jones

6 rows selected.

SQL>
```

This sort is done on state and then on yrgoal divided by 12 within state. Since no order is specified for either sort, everything is in ascending order.

SQL CODE:

```
SQL> SELECT name, city, state, yrgoal/12 "Month Goal"
  2  FROM donor
  3  ORDER BY state, yrgoal/12;

NAME            CITY       ST Month Goal
--------------- ---------- -- ----------
Nancy Taylor    Fall River MA  4.1666667
Robert Brooks   Fall River MA  4.1666667
Susan Ash       Fall River MA  8.3333333
Stephen Daniels Seekonk    MA  41.666667
Jennifer Ames   Providence RI  33.333333
Carl Hersey     Providence RI

6 rows selected.
```

This sort is very similiar to the one above. However instead of using the formula in the ORDER BY, I used the alias that was assigned and I sorted the monthly goal in descending order.

SQL CODE:

```
SQL> SELECT name, city, state, yrgoal/12 month
  2  FROM donor
  3  ORDER BY state, month DESC;

NAME            CITY       ST     MONTH
--------------- ---------- -- ---------
Stephen Daniels Seekonk    MA 41.666667
```

Type here to search

10:29 AM
9/5/2019

File Edit View History Bookmarks Tools Help

CIS50 - Introduction to Oracle

www.pgrocer.net/Cis50/oracle1.html

SQL CODE:

```
SQL> SELECT *
  2  FROM donor
  3  ORDER by state, datefst DESC;

IDNO  NAME            STADR           CITY       ST ZIP   DATEFST      YRGOAL CONTACT
----- --------------- --------------- ---------- -- ----- --------- --------- ------------
11111 Stephen Daniels 123 Elm St      Seekonk    MA 02345 03-JUL-98       500 John Smith
34567 Robert Brooks   36 Pine St      Fall River MA 02720 04-APR-98        50 Amy Costa
23456 Susan Ash       21 Main St      Fall River MA 02720 04-MAR-92       100 Amy Costa
33333 Nancy Taylor    26 Oak St       Fall River MA 02720 04-MAR-92        50 John Adams
22222 Carl Hersey     24 Benefit St   Providence RI 02045 03-JAN-98           Susan Jones
12121 Jennifer Ames   24 Benefit St   Providence RI 02045 24-MAY-97       400 Susan Jones

6 rows selected.

SQL>
```

This sort is done on state and then on yrgoal divided by 12 within state. Since no order is specified for either sort, everything is in ascending order.

SQL CODE:

```
SQL> SELECT name, city, state, yrgoal/12 "Month Goal"
  2  FROM donor
  3  ORDER BY state, yrgoal/12;

NAME            CITY       ST Month Goal
--------------- ---------- -- ----------
Nancy Taylor    Fall River MA  4.1666667
Robert Brooks   Fall River MA  4.1666667
Susan Ash       Fall River MA  8.3333333
Stephen Daniels Seekonk    MA  41.666667
Jennifer Ames   Providence RI  33.333333
Carl Hersey     Providence RI

6 rows selected.
```

This sort is very similiar to the one above. However instead of using the formula in the ORDER BY, I used the alias that was assigned and I sorted the monthly goal in descending order.

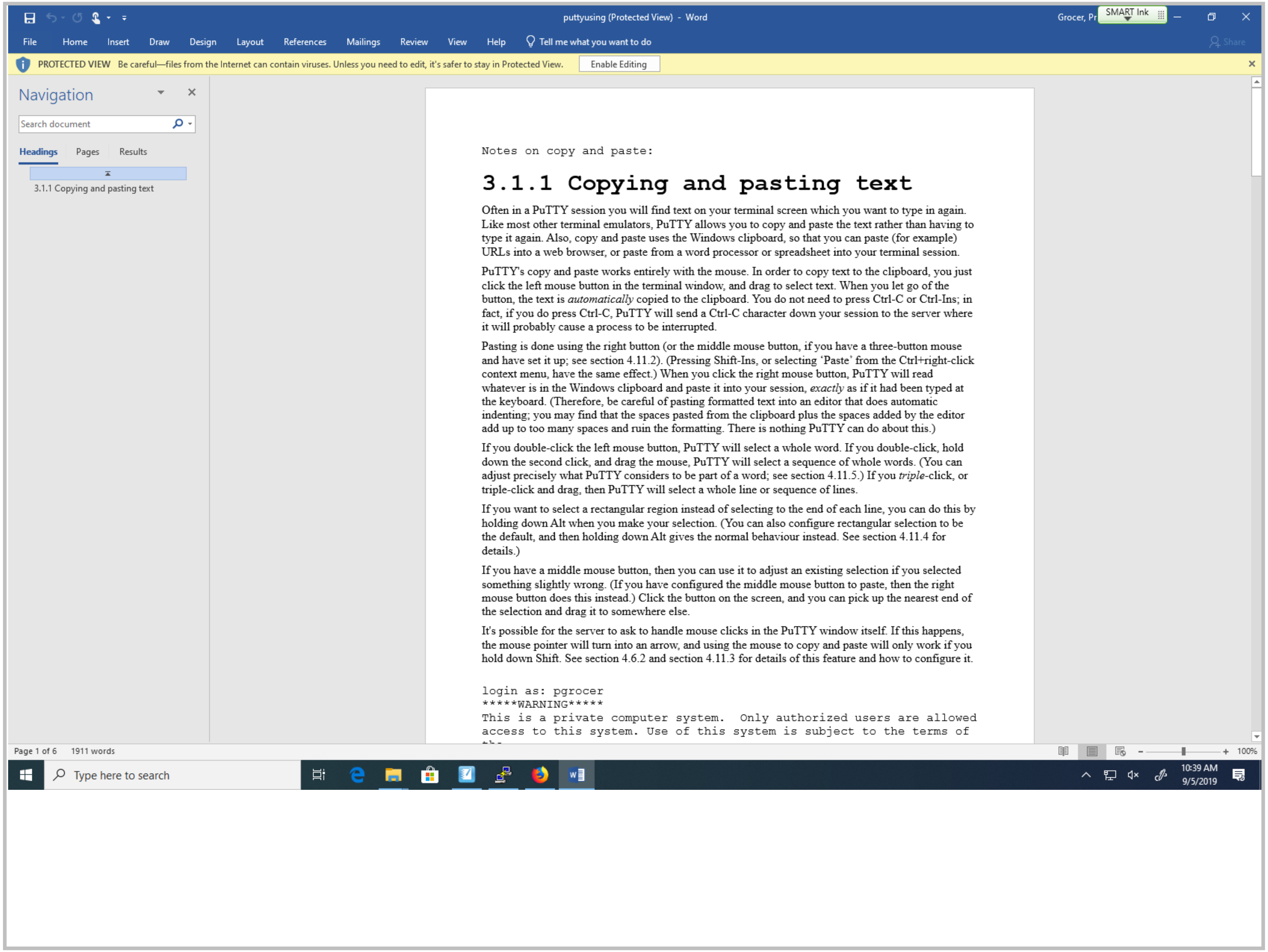
SQL CODE:

```
SQL> SELECT name, city, state, yrgoal/12 month
  2  FROM donor
  3  ORDER BY state, month DESC;

NAME            CITY       ST     MONTH
--------------- ---------- -- ---------
Stephen Daniels Seekonk    MA 41.666667
Susan Ash       Fall River MA 8.3333333
Nancy Taylor    Fall River MA 4.1666667
Robert Brooks   Fall River MA 4.1666667
Carl Hersey     Providence RI
Jennifer Ames   Providence RI 33.333333

6 rows selected.
```

Type here to search

10:30 AM
9/5/2019

Title: Sep 5-10:30 AM (19 of 20)



Notes on copy and paste:

3.1.1 Copying and pasting text

Often in a PuTTY session you will find text on your terminal screen which you want to type in again. Like most other terminal emulators, PuTTY allows you to copy and paste the text rather than having to type it again. Also, copy and paste uses the Windows clipboard, so that you can paste (for example) URLs into a web browser, or paste from a word processor or spreadsheet into your terminal session.

PuTTY's copy and paste works entirely with the mouse. In order to copy text to the clipboard, you just click the left mouse button in the terminal window, and drag to select text. When you let go of the button, the text is *automatically* copied to the clipboard. You do not need to press Ctrl-C or Ctrl-Ins; in fact, if you do press Ctrl-C, PuTTY will send a Ctrl-C character down your session to the server where it will probably cause a process to be interrupted.

Pasting is done using the right button (or the middle mouse button, if you have a three-button mouse and have set it up; see section 4.11.2). (Pressing Shift-Ins, or selecting 'Paste' from the Ctrl+right-click context menu, have the same effect.) When you click the right mouse button, PuTTY will read whatever is in the Windows clipboard and paste it into your session, *exactly* as if it had been typed at the keyboard. (Therefore, be careful of pasting formatted text into an editor that does automatic indenting; you may find that the spaces pasted from the clipboard plus the spaces added by the editor add up to too many spaces and ruin the formatting. There is nothing PuTTY can do about this.)

If you double-click the left mouse button, PuTTY will select a whole word. If you double-click, hold down the second click, and drag the mouse, PuTTY will select a sequence of whole words. (You can adjust precisely what PuTTY considers to be part of a word; see section 4.11.5.) If you *triple*-click, or triple-click and drag, then PuTTY will select a whole line or sequence of lines.

If you want to select a rectangular region instead of selecting to the end of each line, you can do this by holding down Alt when you make your selection. (You can also configure rectangular selection to be the default, and then holding down Alt gives the normal behaviour instead. See section 4.11.4 for details.)

If you have a middle mouse button, then you can use it to adjust an existing selection if you selected something slightly wrong. (If you have configured the middle mouse button to paste, then the right mouse button does this instead.) Click the button on the screen, and you can pick up the nearest end of the selection and drag it to somewhere else.

It's possible for the server to ask to handle mouse clicks in the PuTTY window itself. If this happens, the mouse pointer will turn into an arrow, and using the mouse to copy and paste will only work if you hold down Shift. See section 4.6.2 and section 4.11.3 for details of this feature and how to configure it.

```
login as: pgrocer
*****WARNING*****
This is a private computer system.  Only authorized users are allowed
access to this system. Use of this system is subject to the terms of
```


Title: Sep 5-10:39 AM (20 of 20)