Project #2 PL/SQL (Procedures and Triggers)

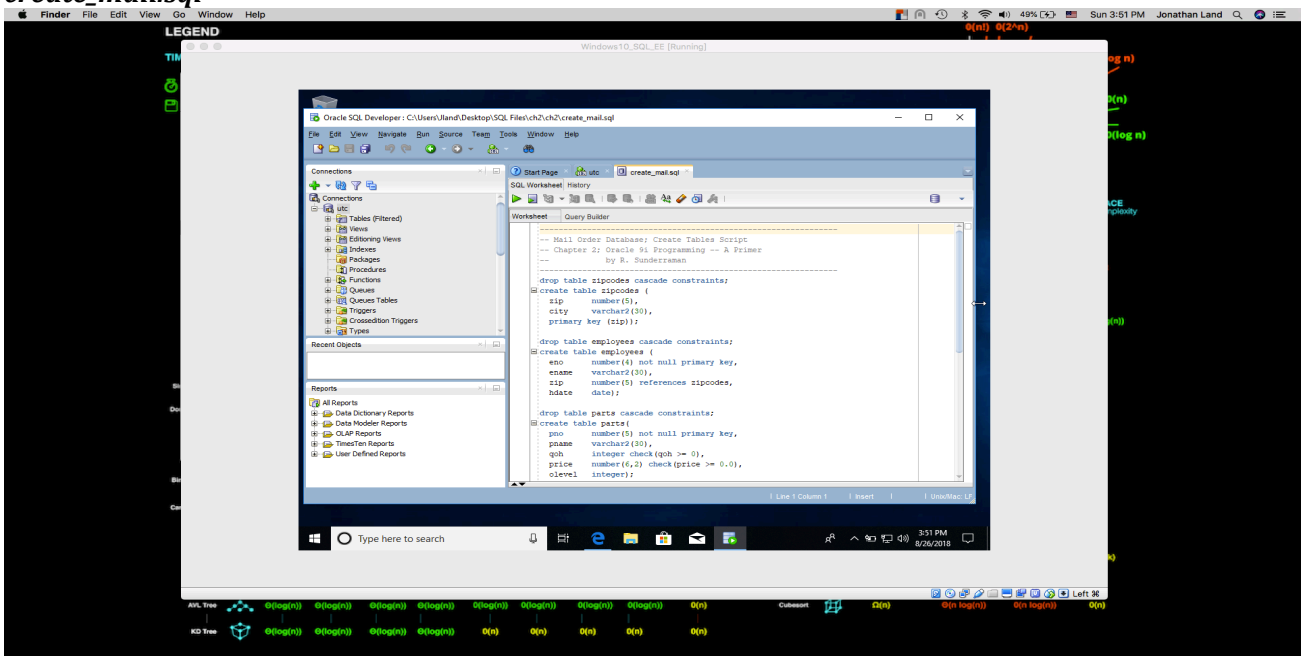
CPSC 5270 Advanced Database and Database Security

Jonathan Land

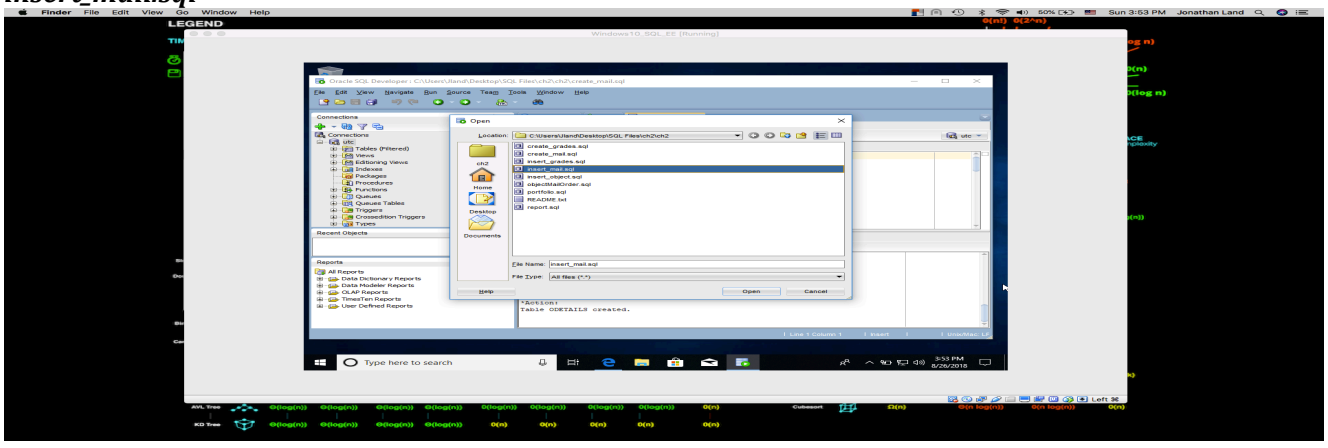
Preparation (20 points):

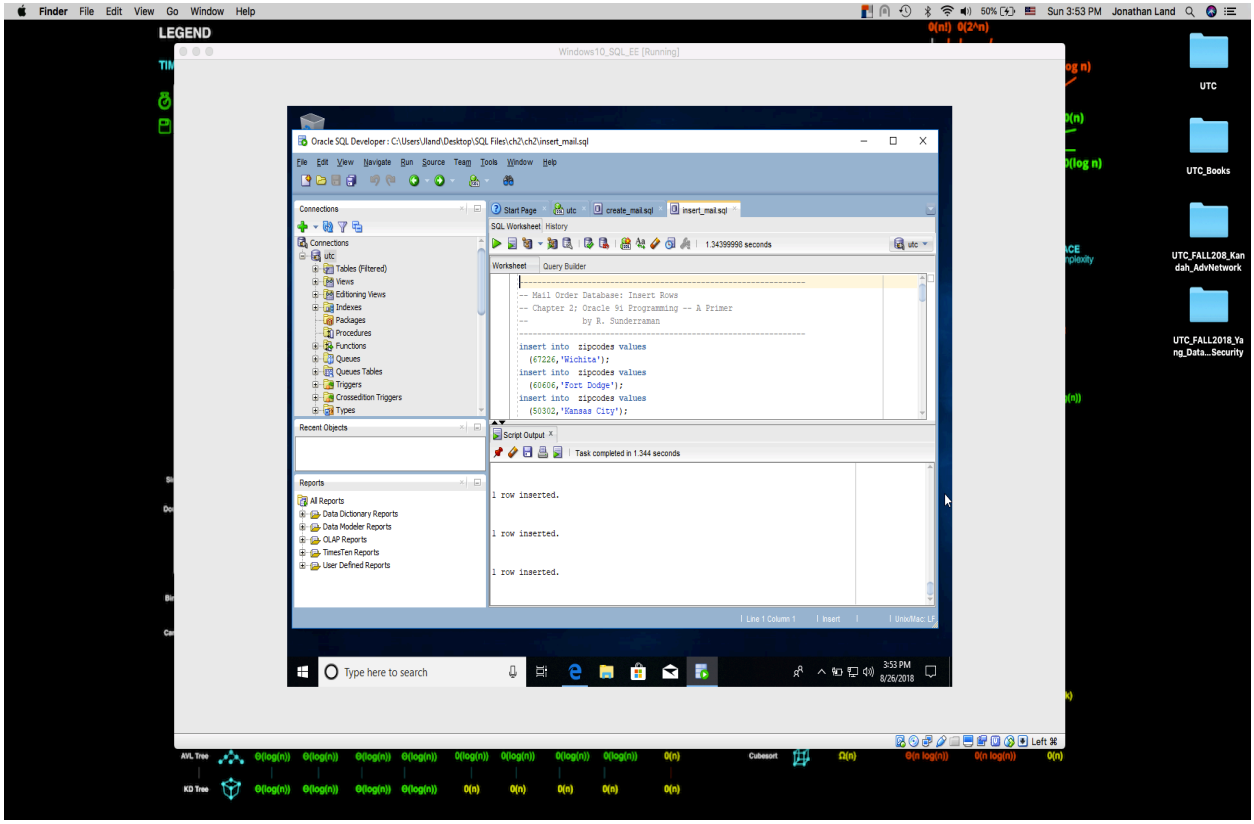
1. Execute scripts *such as create_mail.sql, insert_mail.sql,* create_grades*.sql, and insert_grades.sql* scripts in the zip file of plsql.ch2.zip.

create_mail.sql

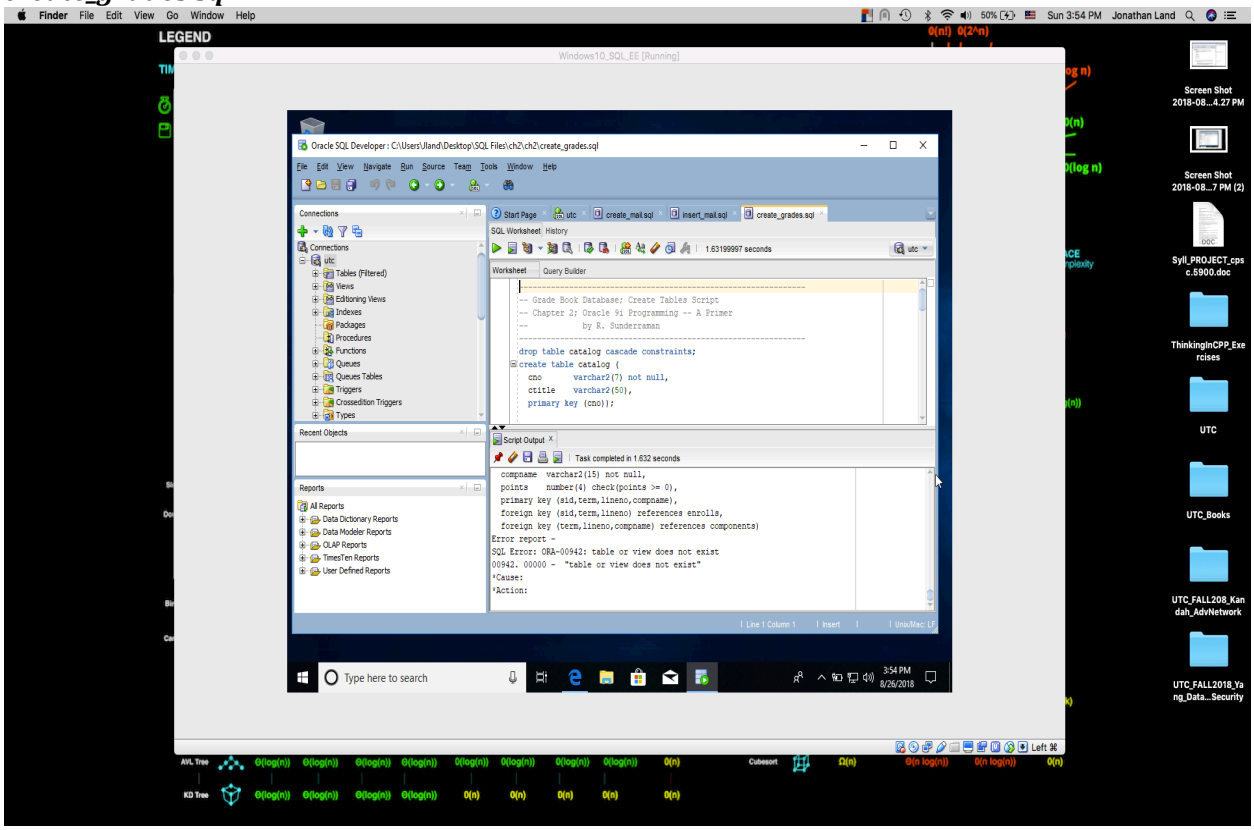


insert_mail.sql

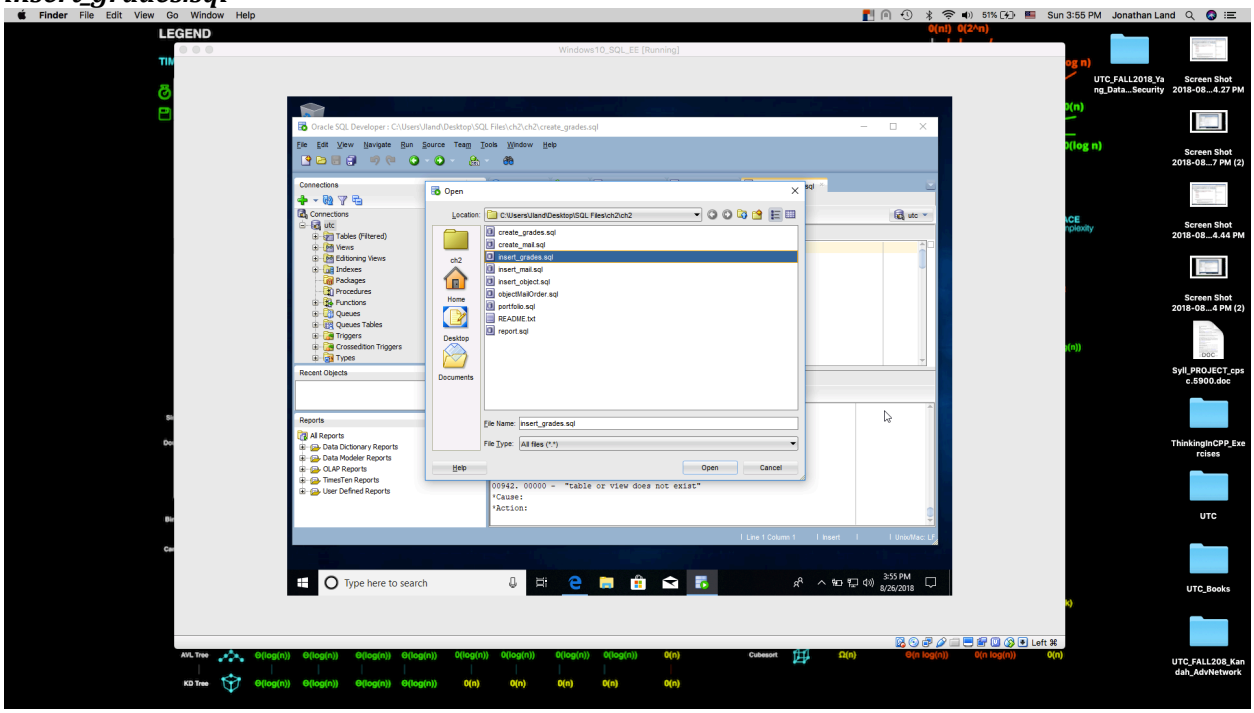


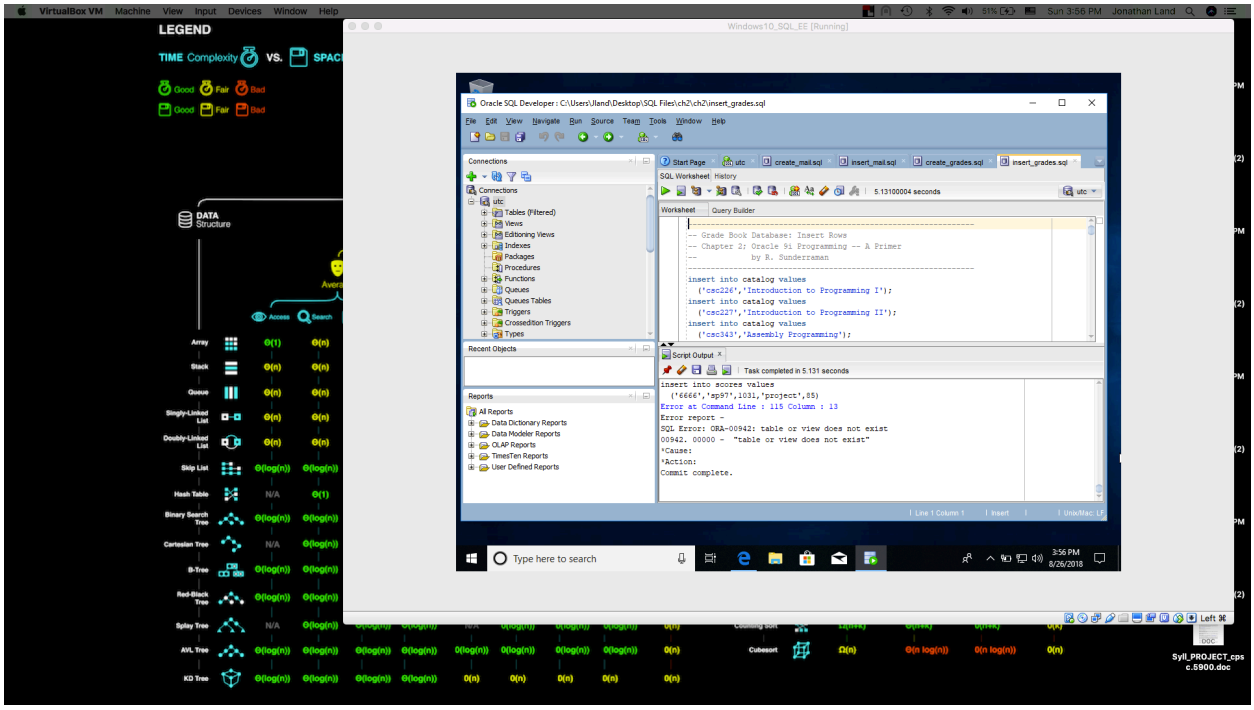


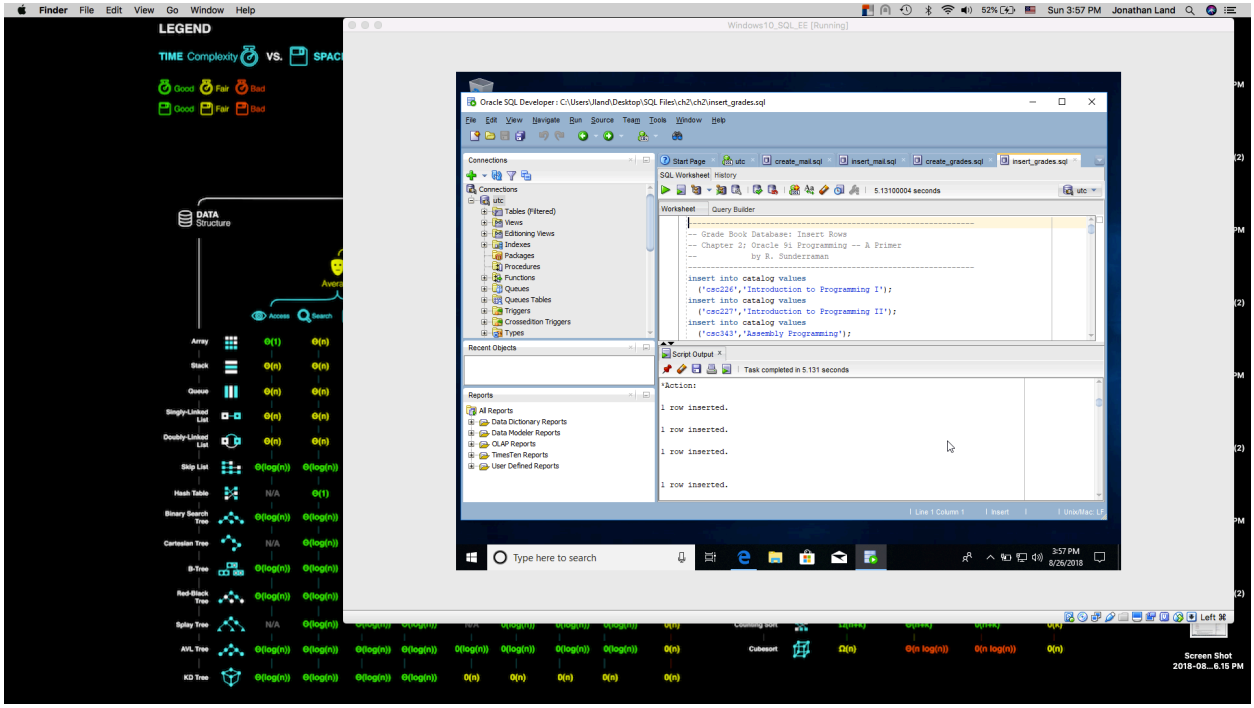
create_grades.sql



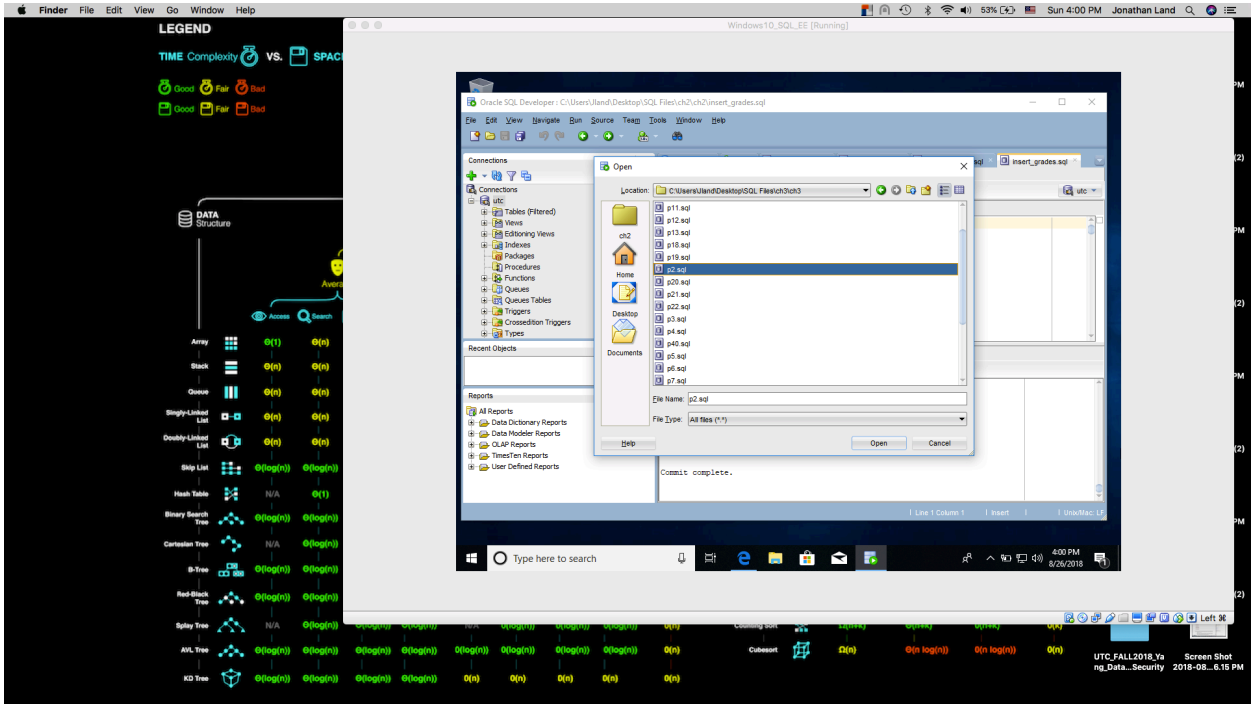
insert_grades.sql

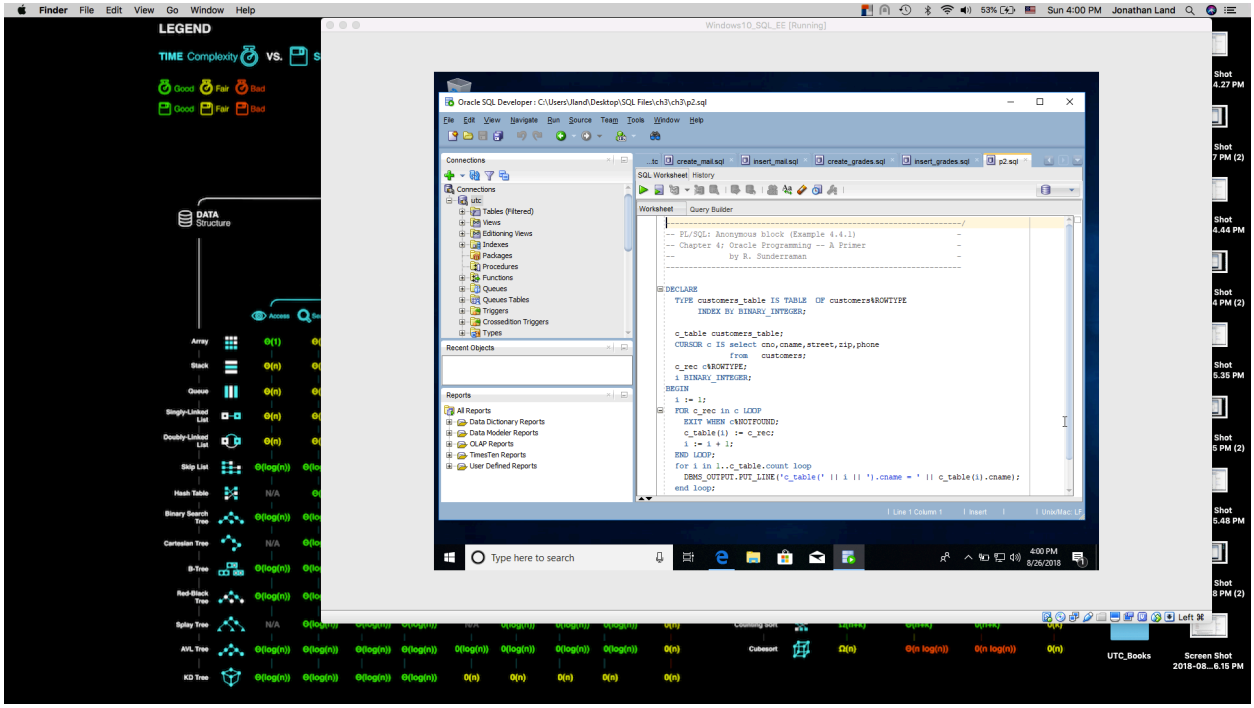


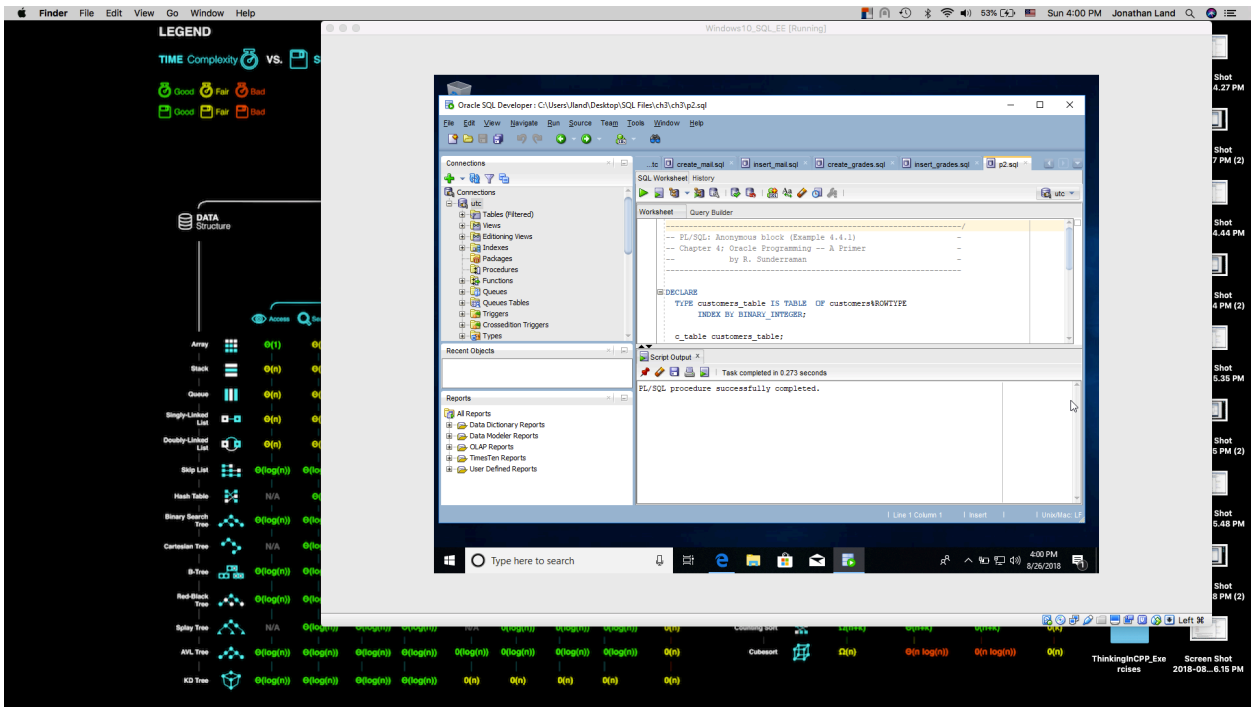




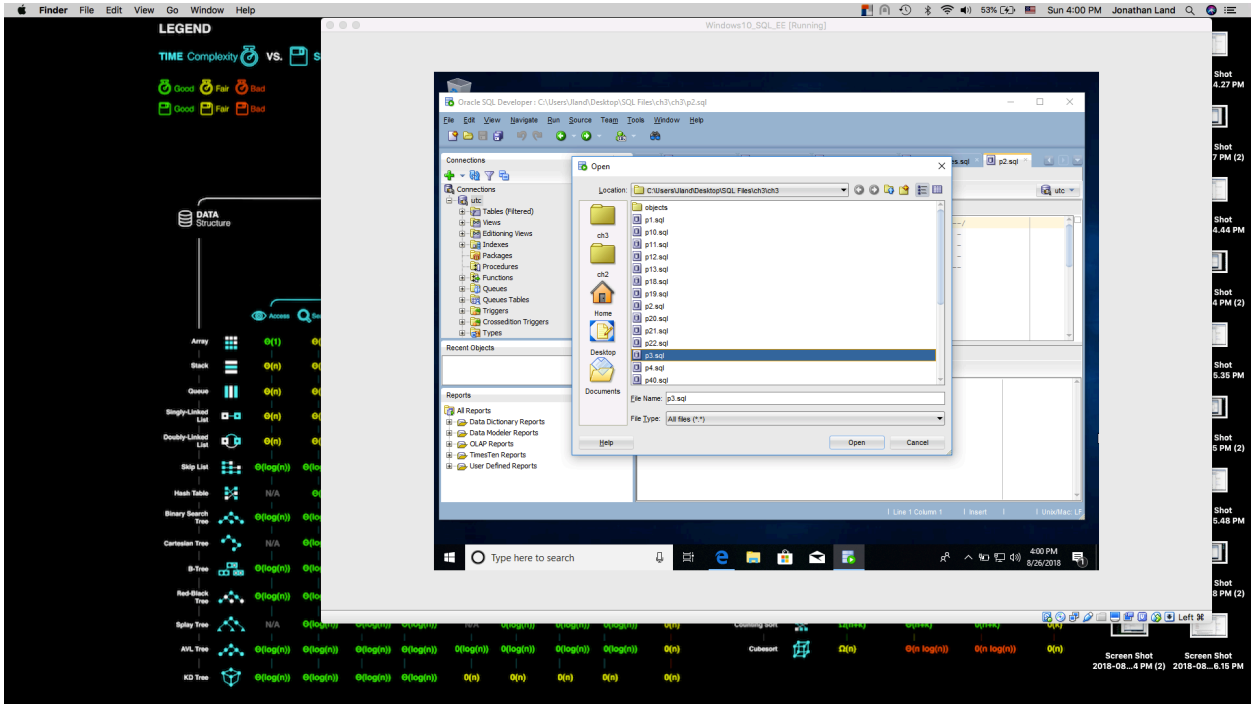
2. Execute scripts in plsql.ch3.zip including *p2.sql*, *p3.sql*, *p4.sql*, *p5.sql*, *p6.sql* by typing: **start p2**
    a. p2.sql gets all the rows from the customers table and prints the names of the customers on the screen.

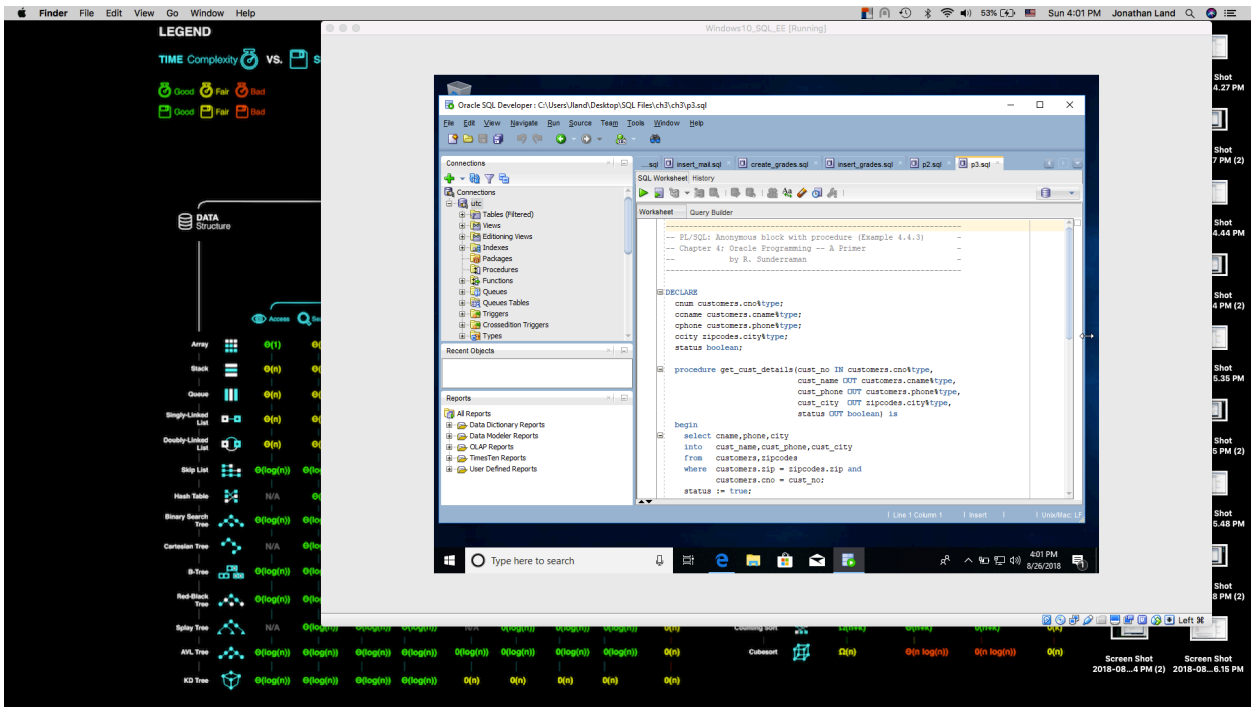


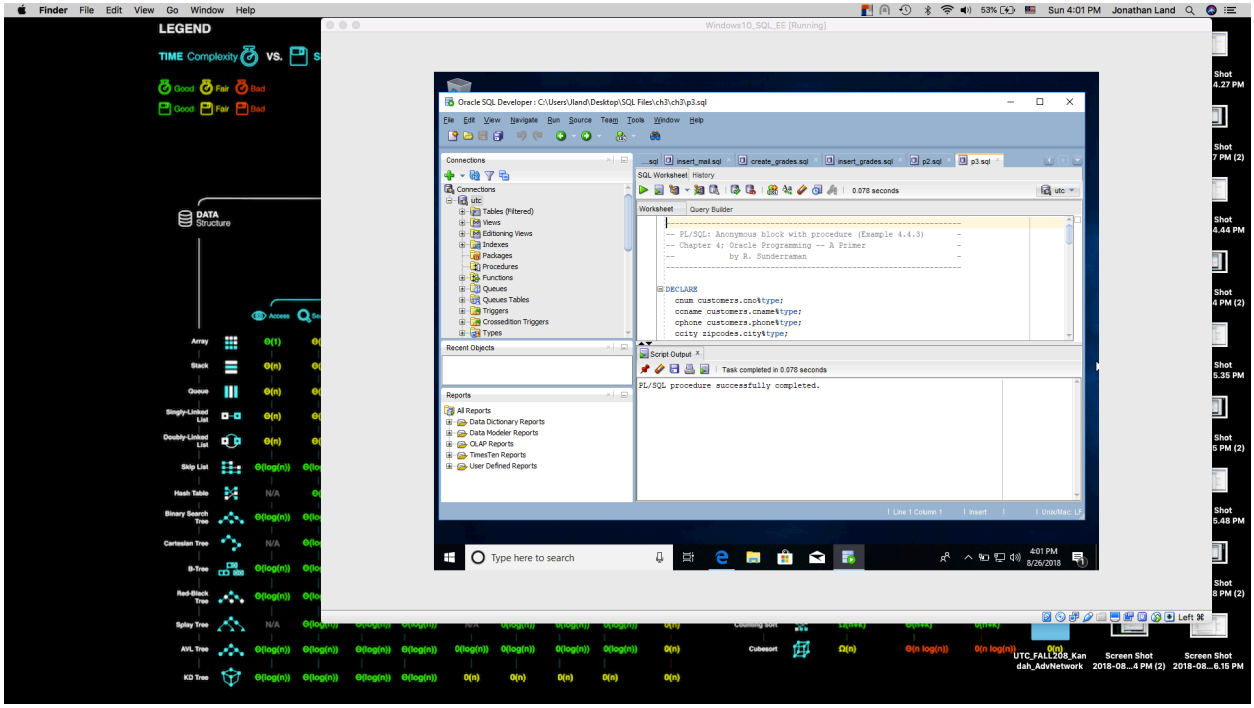




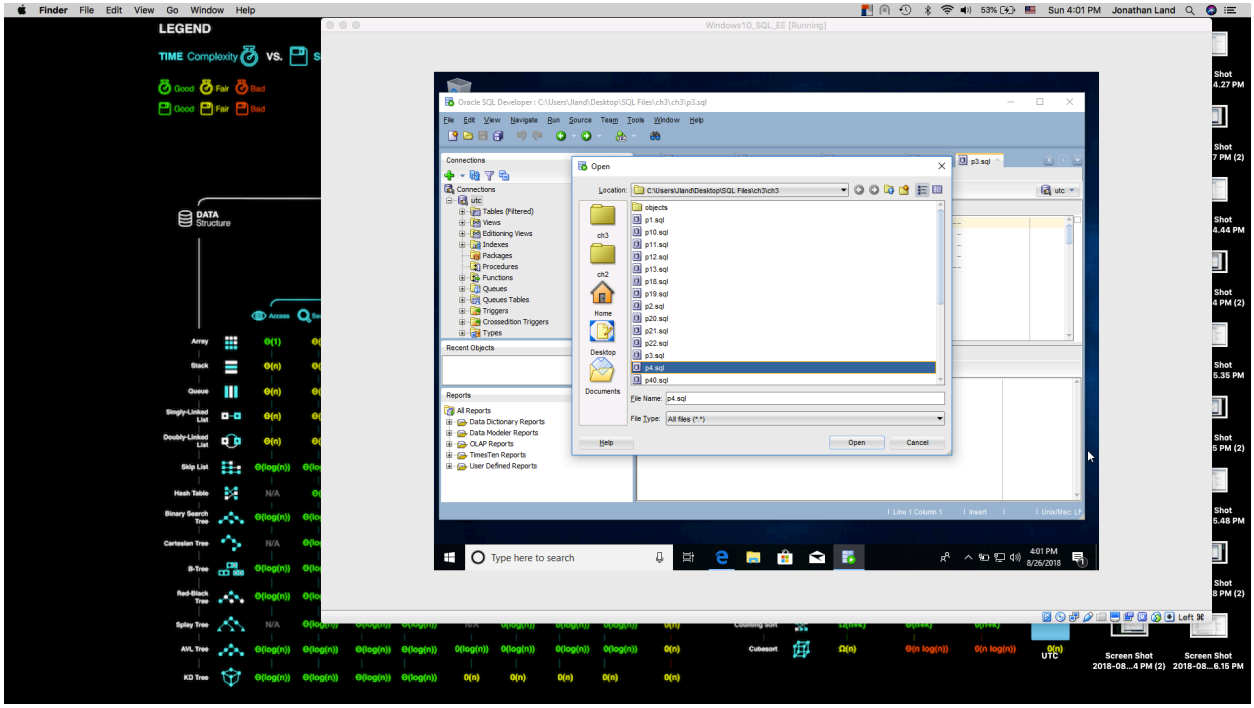
b. p3.sql contains a procedure that accepts as input a customer number and returns the name, phone, and city values for that customers. If such a customer is not found, status is returned as false; otherwise, status is returned as true.

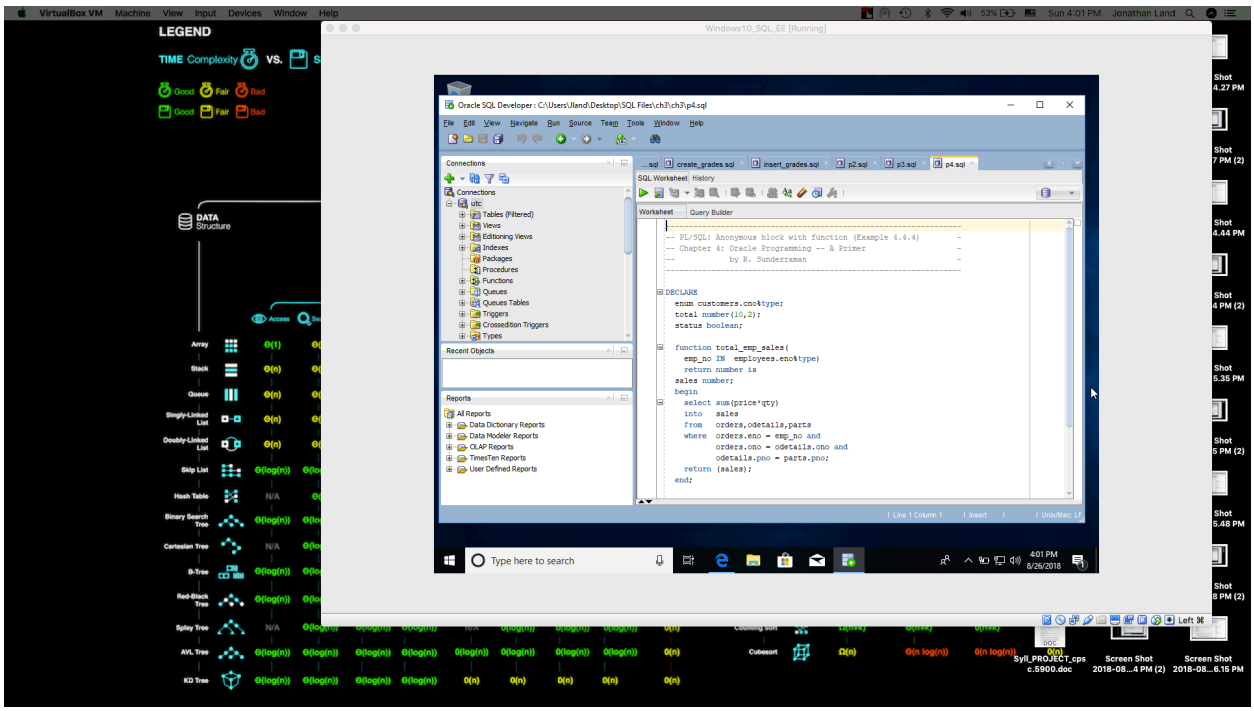

Finder File Edit View Go Window Help
Sun 4:00 PM Jonathan Land
LEGEND
TIME Complexity VS.
Good Fair Bad
DATA Structure
Oracle SQL Developer : C:\Users\Jland\Desktop\SQL Files\ch3\ch3\p2.sql
Open
Type here to search
4:00 PM 8/26/2018

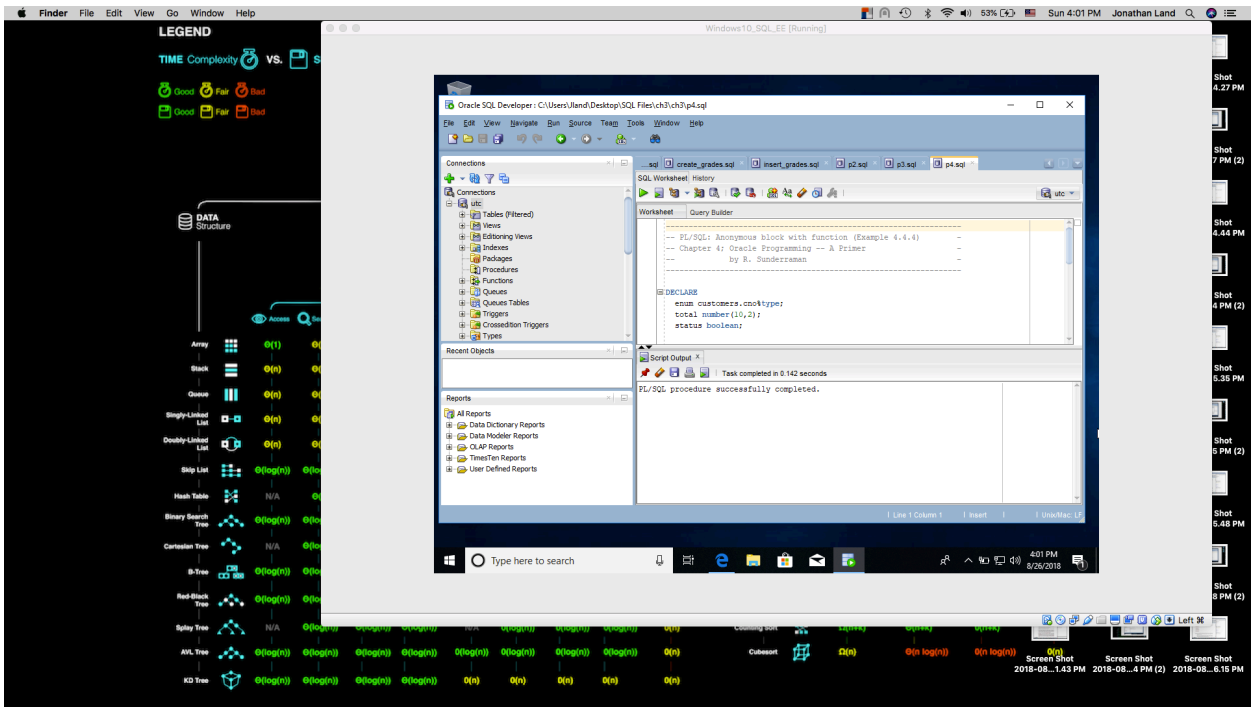

Finder File Edit View Go Window Help
Sun 4:01 PM Jonathan Land
LEGEND
TIME Complexity VS.
Good Fair Bad
DATA Structure
Oracle SQL Developer : C:\Users\Jland\Desktop\SQL Files\ch3\ch3\p3.sql
Type here to search
4:01 PM 8/26/2018



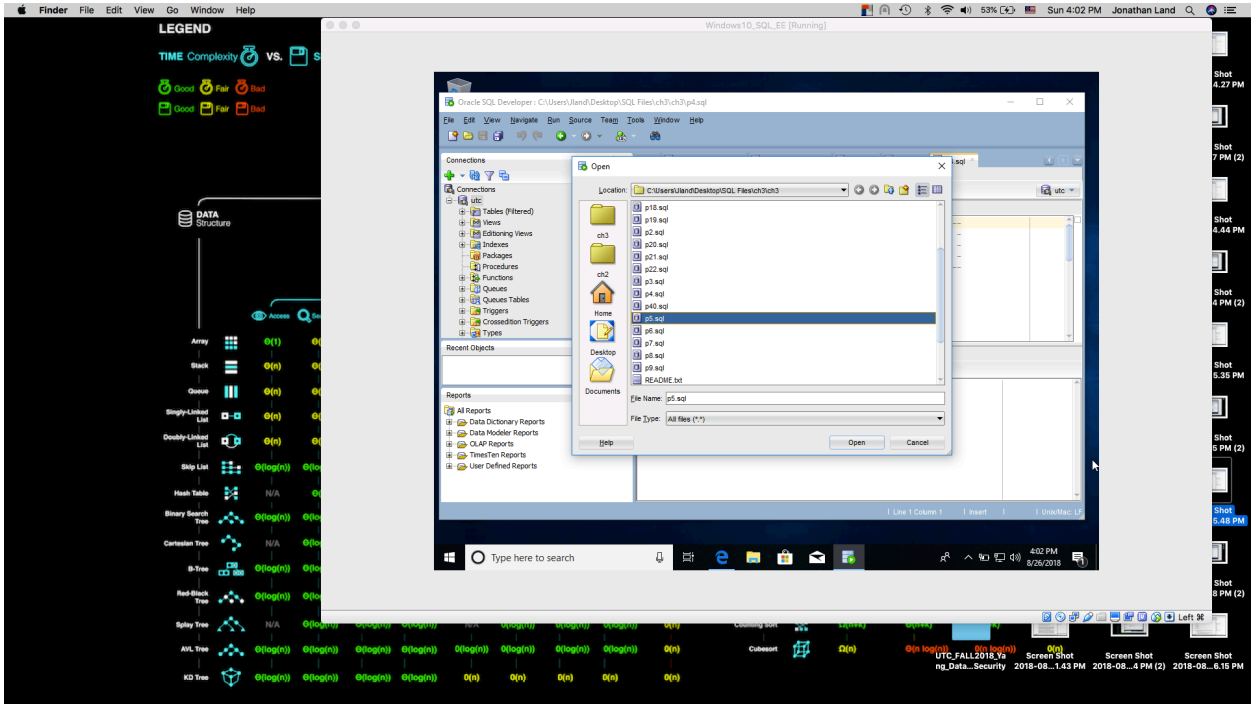
c. p4.sql contains a function declaration. Given an employee number, the function computes and returns the total sales for that employee.

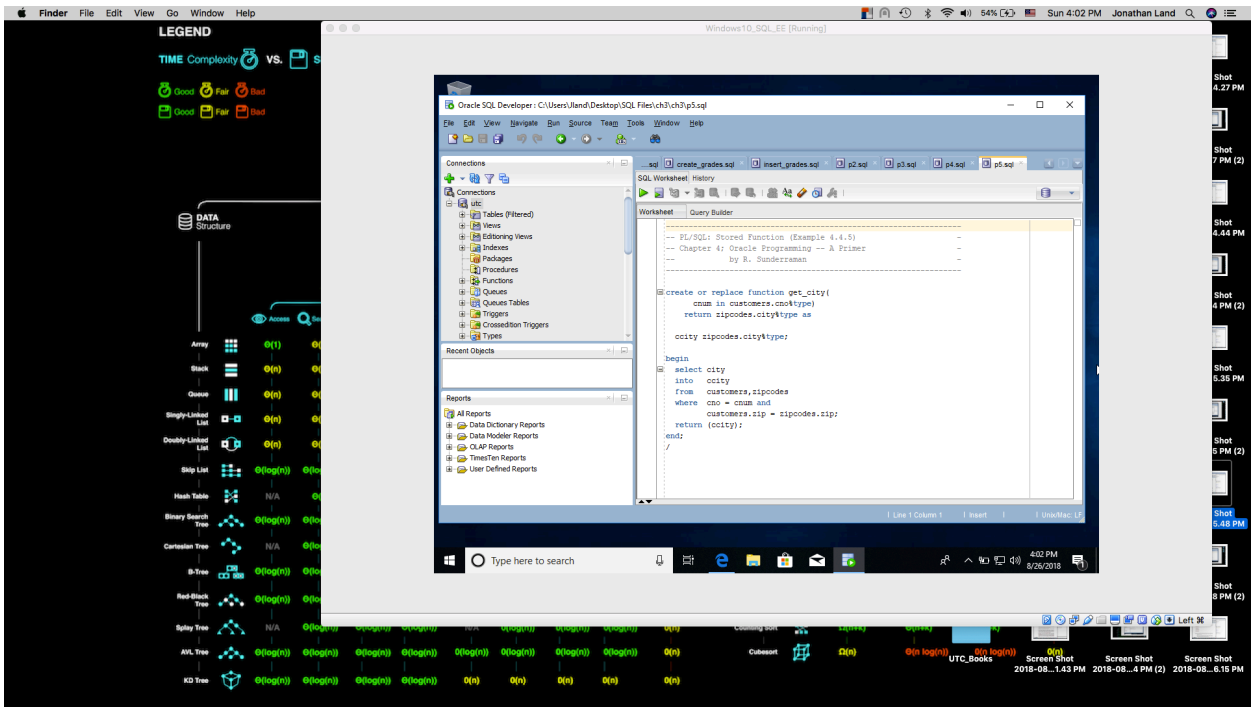


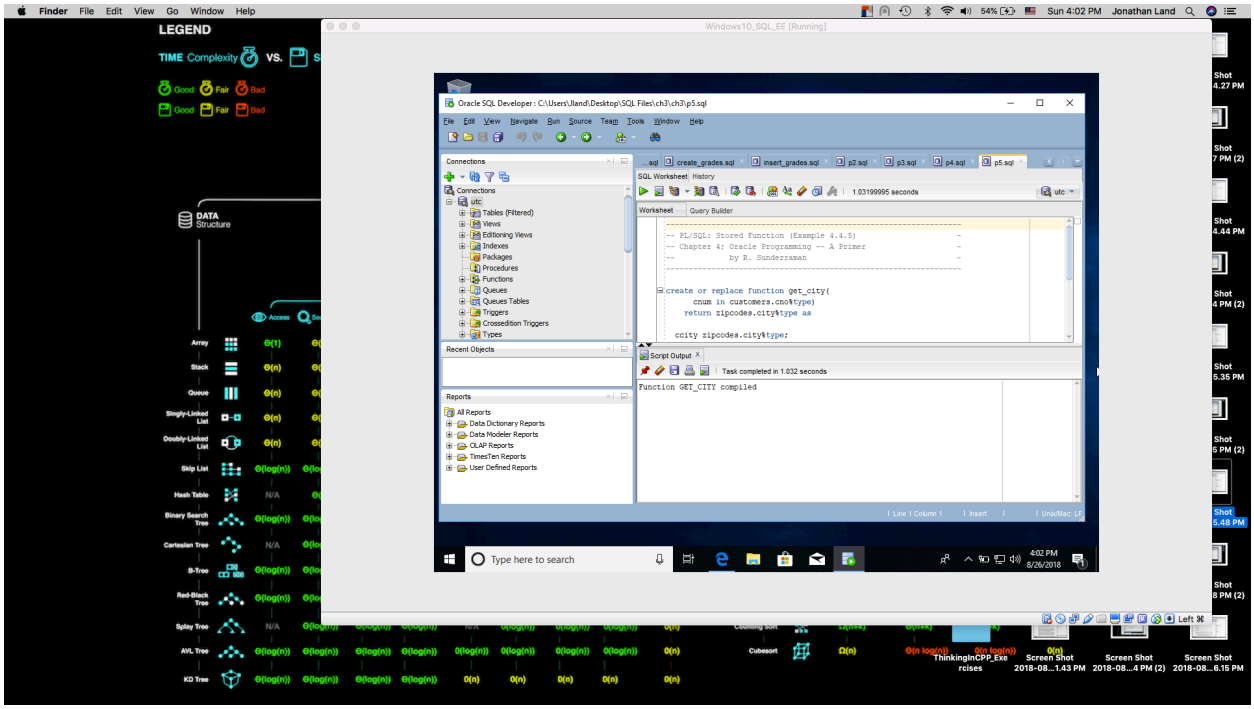




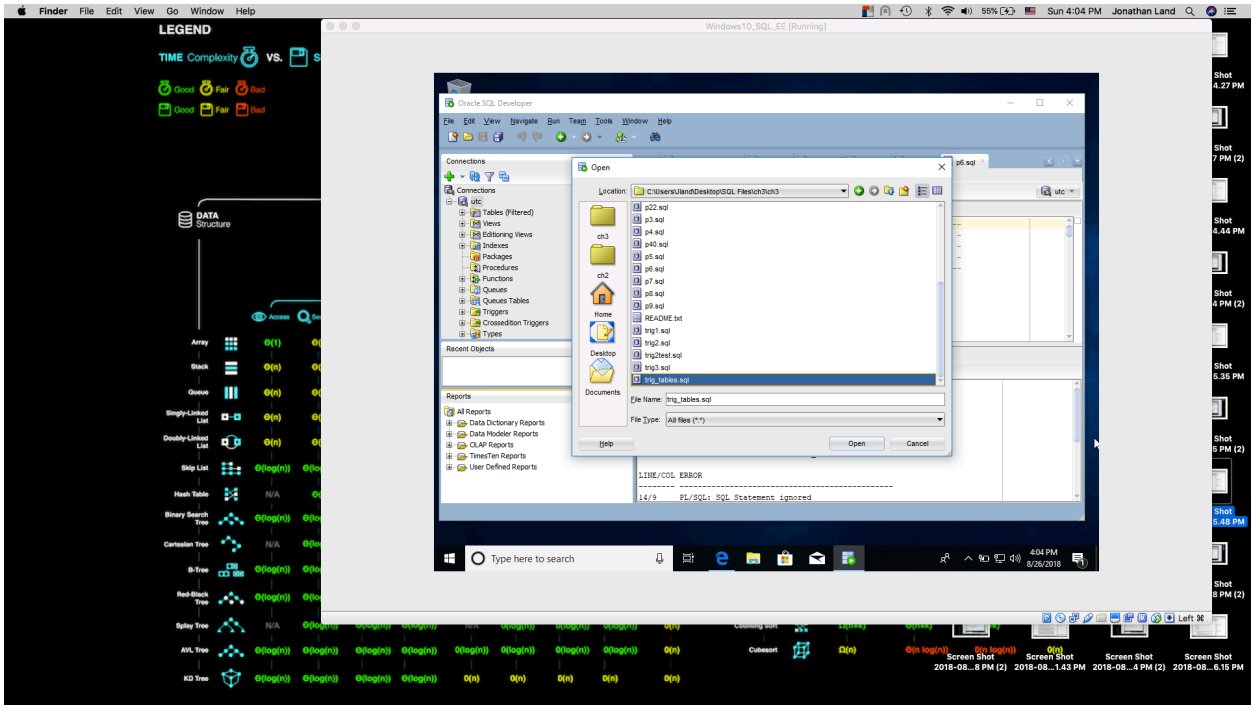
d. P5.sql contains stored function that takes a customer number as input and returns the city in which the customer lives. A stored procedure can be invoked from various environments, including an SQL statement, another stored function or procedure, an anonymous block or a procedure or function defined within it, an embedded SQL program, a database trigger, and Oracle tools. The stored function get_city can be called by typing: **select cno, cname, get_city(cno);**

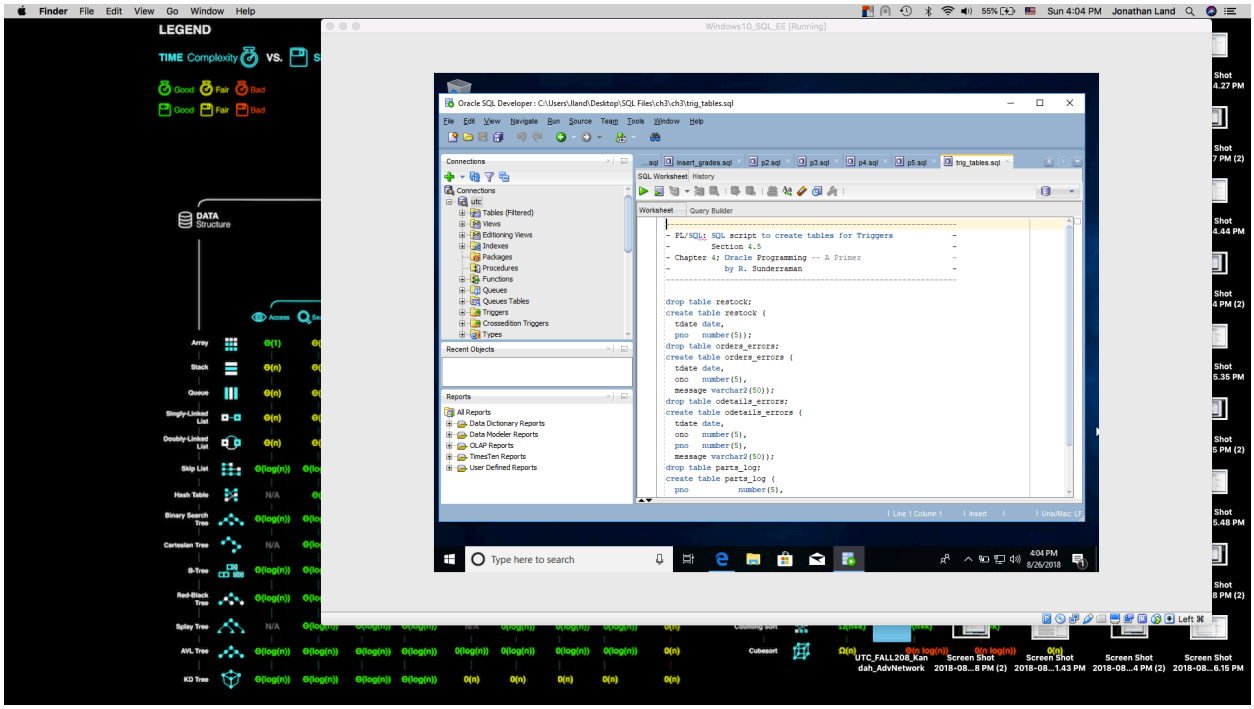

Finder File Edit View Go Window Help
Sun 4:02 PM Jonathan Land
LEGEND
TIME Complexity VS.
Good Fair Bad
DATA Structure
Oracle SQL Developer : C:\Users\Jland\Desktop\SQL Files\ch3\ch3\p4.sql
Open
p5.sql
Type here to search
4:02 PM 8/26/2018

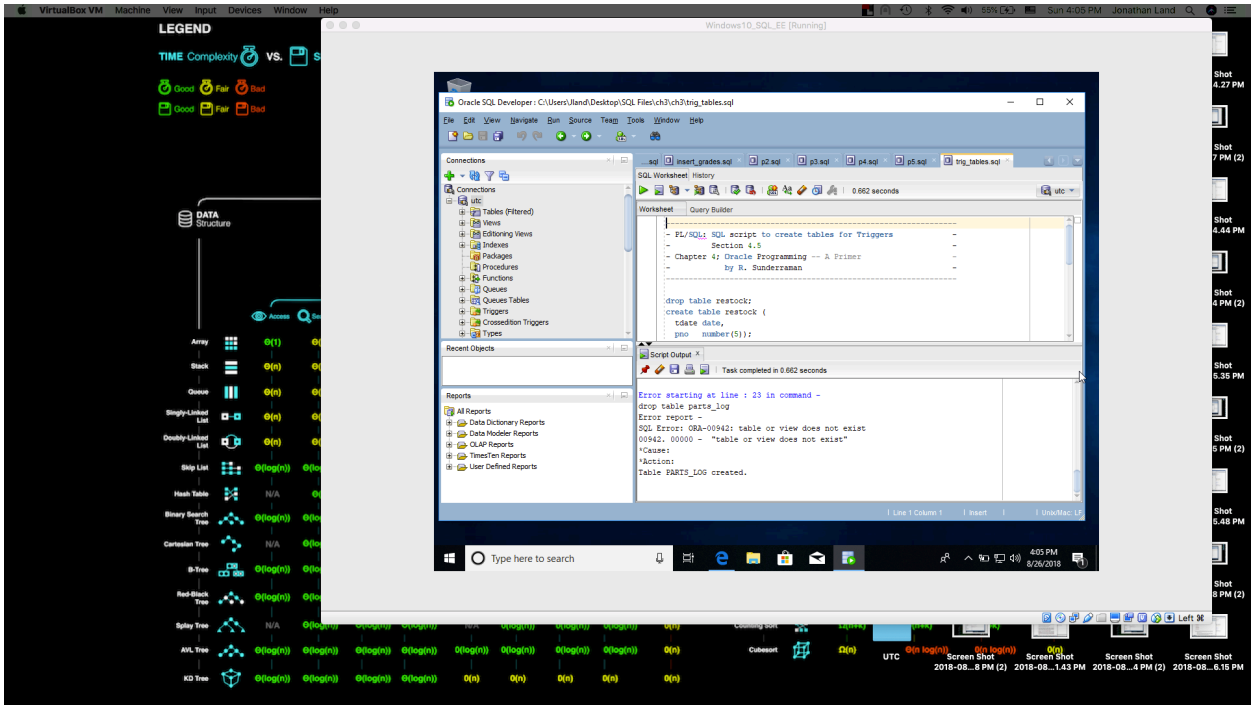

Finder File Edit View Go Window Help
Sun 4:02 PM Jonathan Land
LEGEND
TIME Complexity VS.
Good Fair Bad
DATA Structure
Oracle SQL Developer : C:\Users\Jland\Desktop\SQL Files\ch3\ch3\p5.sql
create or replace function get_city(
cnum in customers.cno%type)
return zipcodes.city%type as
ccity zipcodes.city%type;
begin
select city
into ccity
from customers,zipcodes
where cno = cnum and
customers.zip = zipcodes.zip;
return (ccity);
end;
/
Type here to search
4:02 PM 8/26/2018

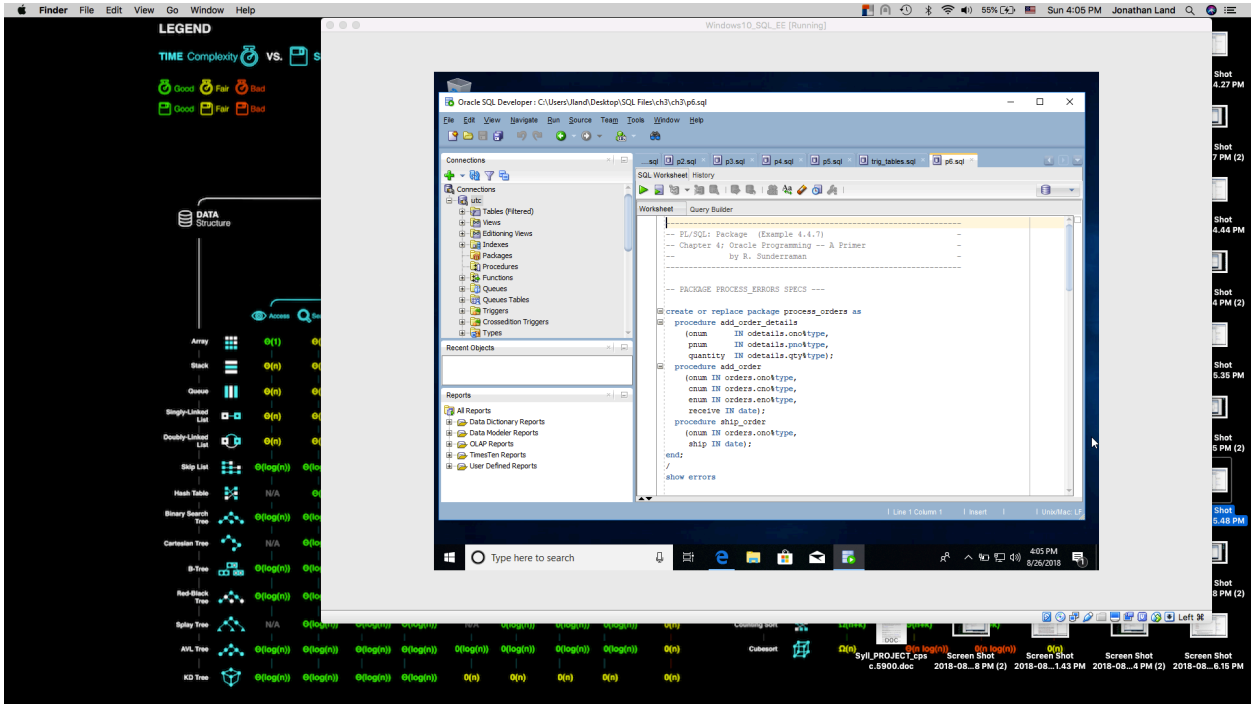


e. p6.sql defines a package called process_orders containg three procedures. add_order takes as input an order number, customer number, employee number, and received date and tries to insert a new row in the orders table. add_order_details receives as input an order number, part number, and quantity and attempts to add a row corresponding to the input in the odetails table. ship_order takes as input an order number and a shipped date and tries to update the shipped value for the order. Note: You need run trig_tables in ch03 first before you run p6.sql.



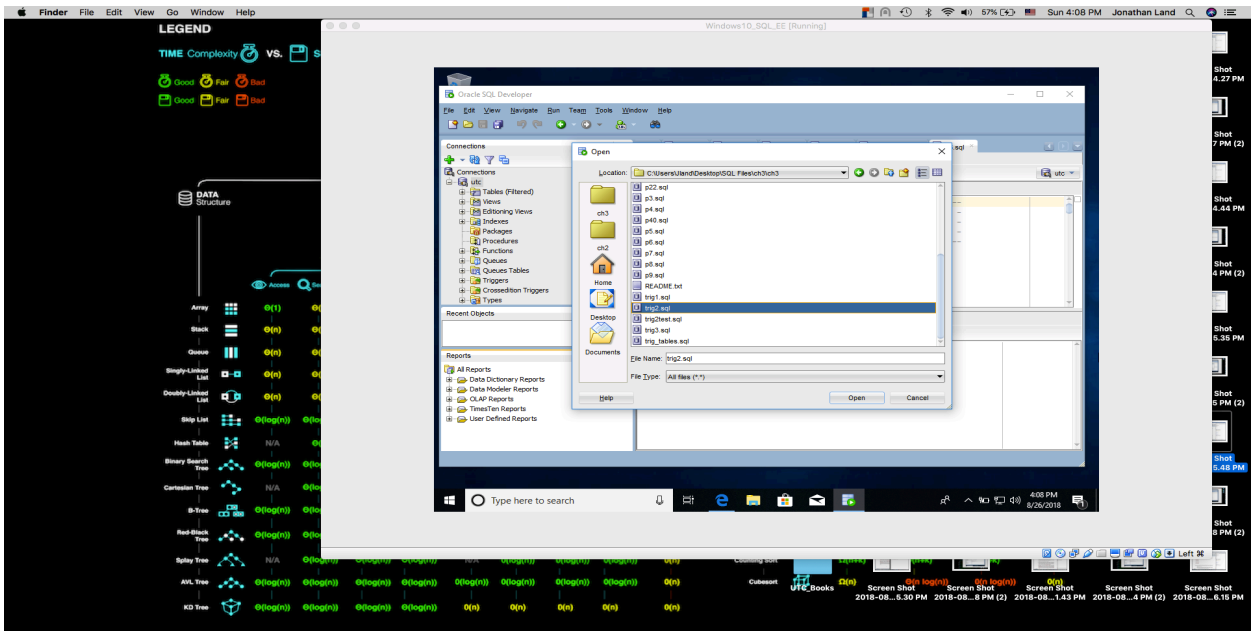

Finder File Edit View Go Window Help
Sun 4:04 PM Jonathan Land
LEGEND
TIME Complexity VS.
Oracle SQL Developer : C:\Users\Jland\Desktop\SQL Files\ch3\ch3\trig_tables.sql
Type here to search
4:04 PM
8/26/2018

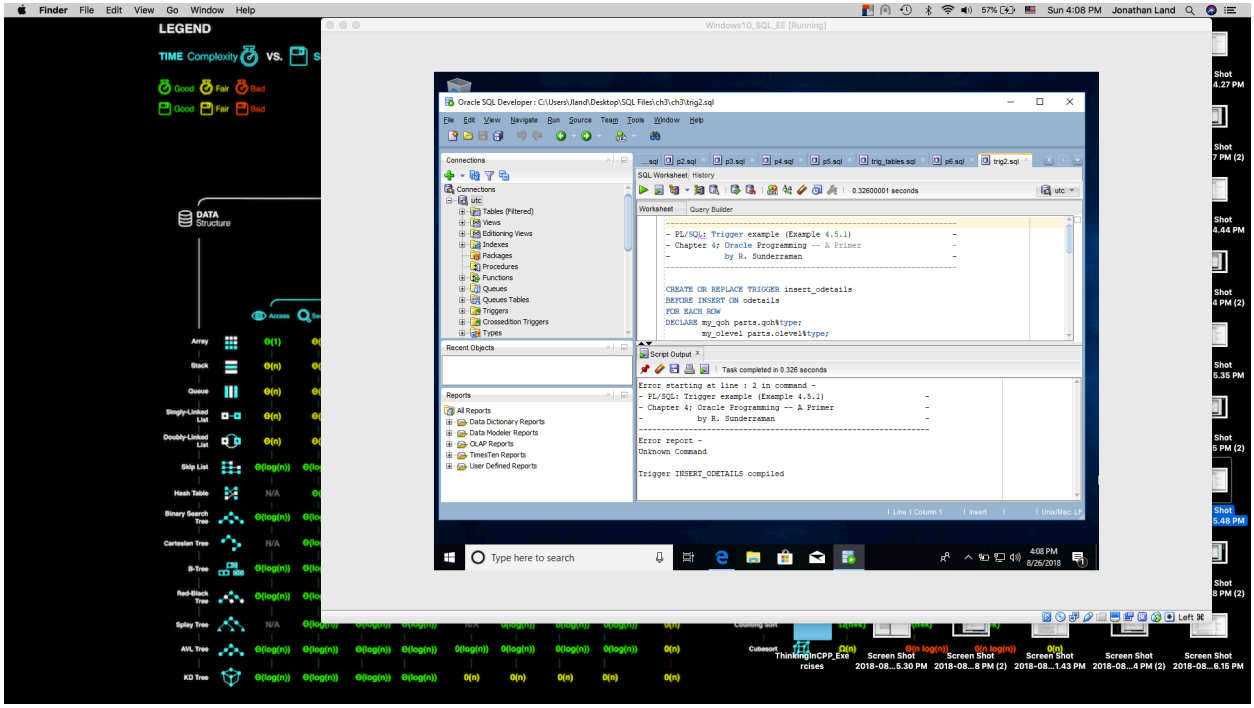

VirtualBox VM Machine View Input Devices Window Help
Sun 4:05 PM Jonathan Land
LEGEND
TIME Complexity VS.
Oracle SQL Developer : C:\Users\Jland\Desktop\SQL Files\ch3\ch3\trig_tables.sql
Error starting at line : 23 in command -
drop table parts_log
Error report -
SQL Error: ORA-00942: table or view does not exist
00942. 00000 - "table or view does not exist"
*Cause:
*Action:
Table PARTS_LOG created.
Type here to search
4:05 PM
8/26/2018



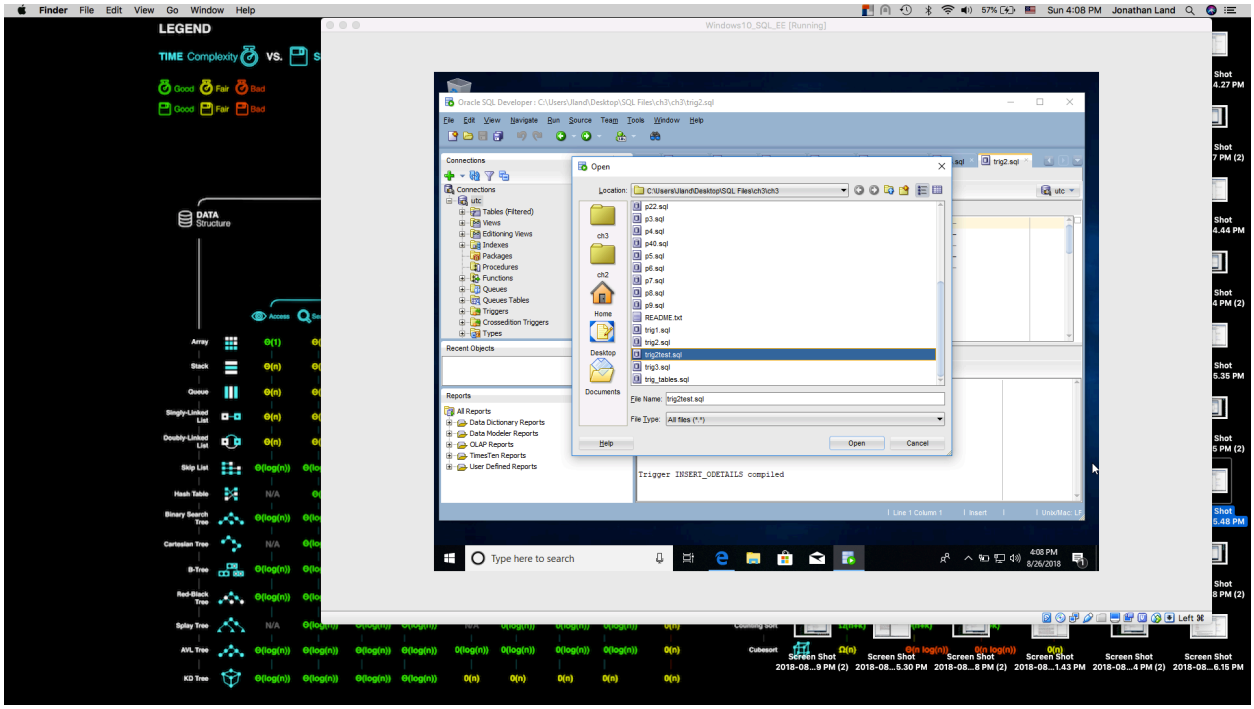
3. Execute *trig2.sql*, *trig2test.sql*, *trig3.sql* in plsql.ch3.zip

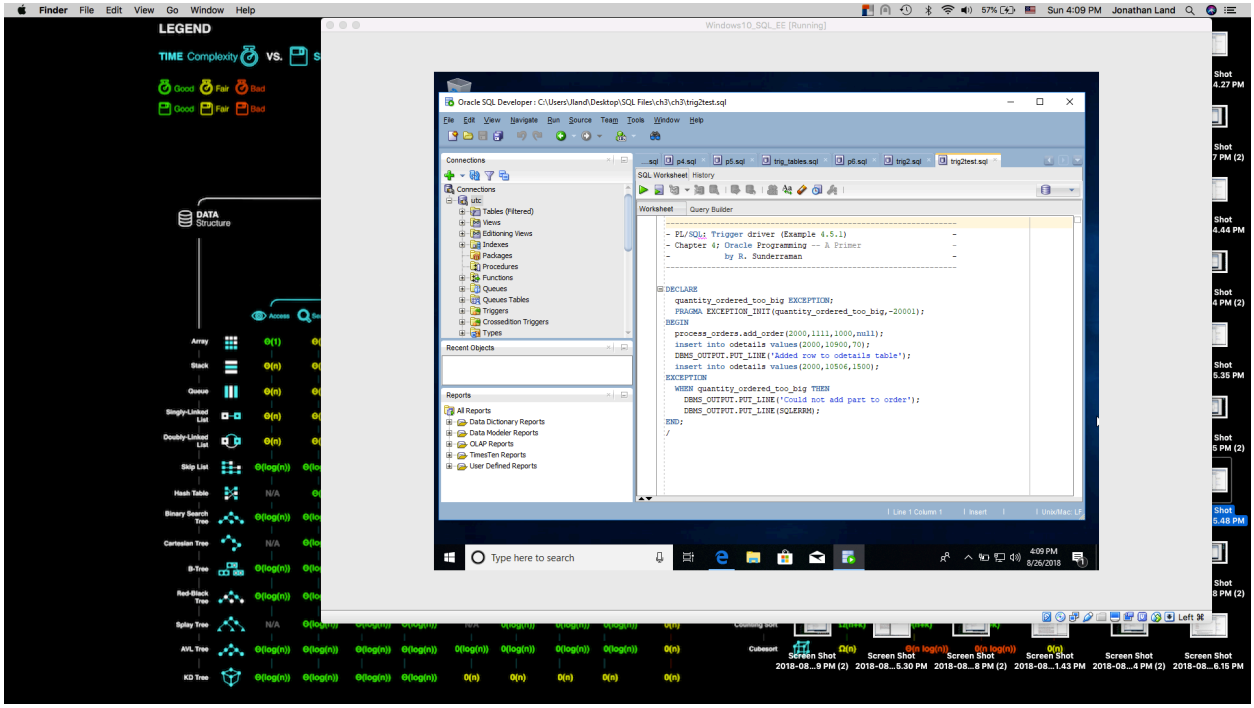
   a. In trig2.sql, a trigger is executed when a row is inserted into the odetails table. the trigger checks to see if the quantity ordered is more than the quantity on hand. If it is, an error message is generated, and the row is not inserted. Otherwise, the trigger updates the quantity on hand for the part and checks to see if it has fallen below the reorder level. If it has, it sends a row to the restock table indicating that the part needs to be reordered. You may need create a restock table here, with the first column is TDATE, and the second column is PNO.

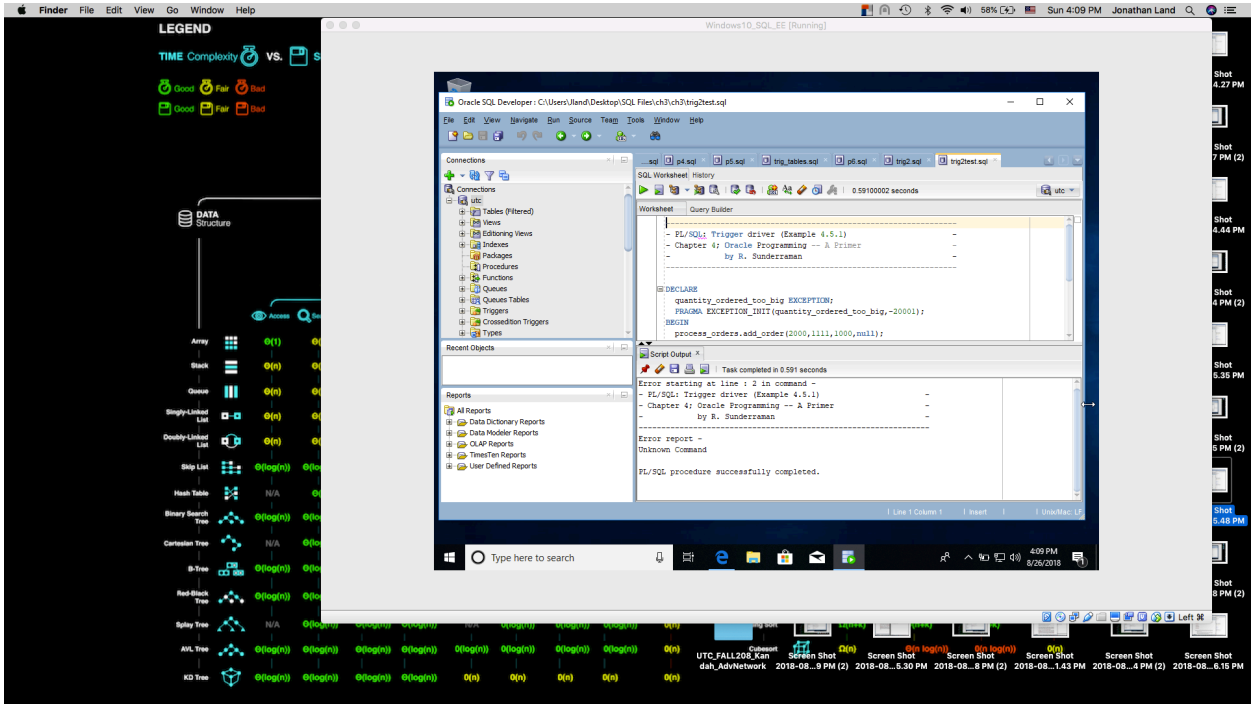




b. trig2test.sql is executed to create a new order.



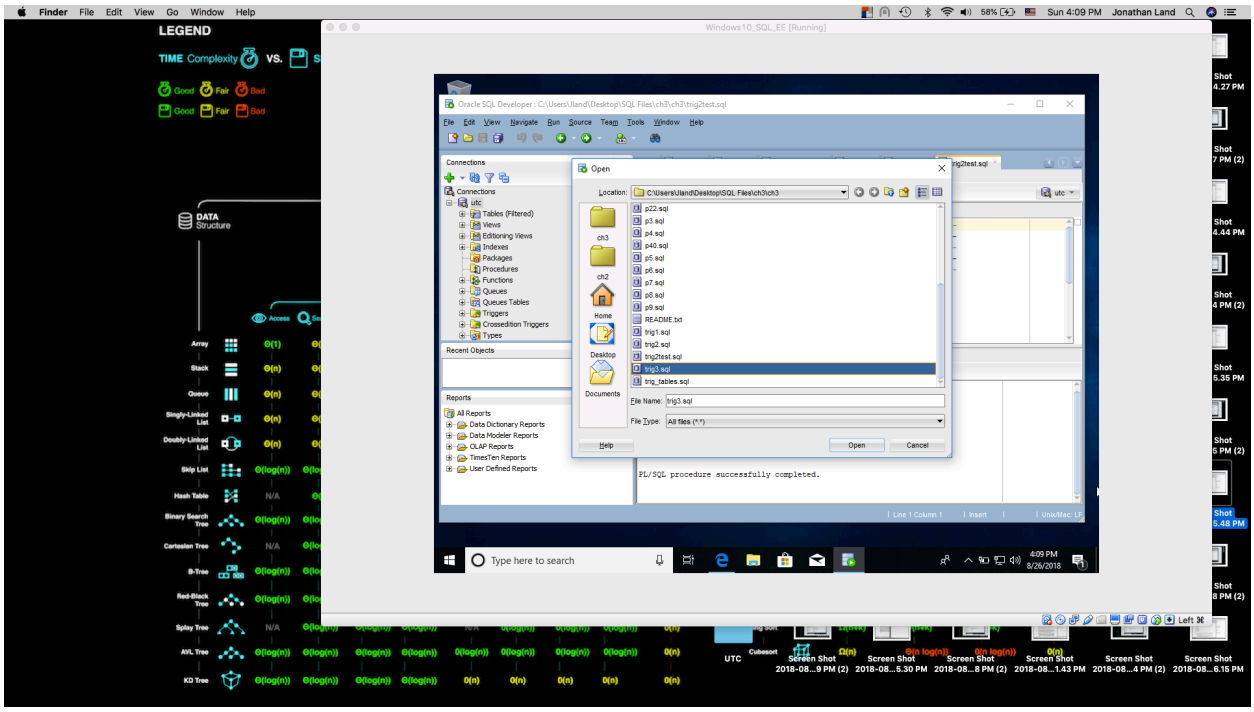




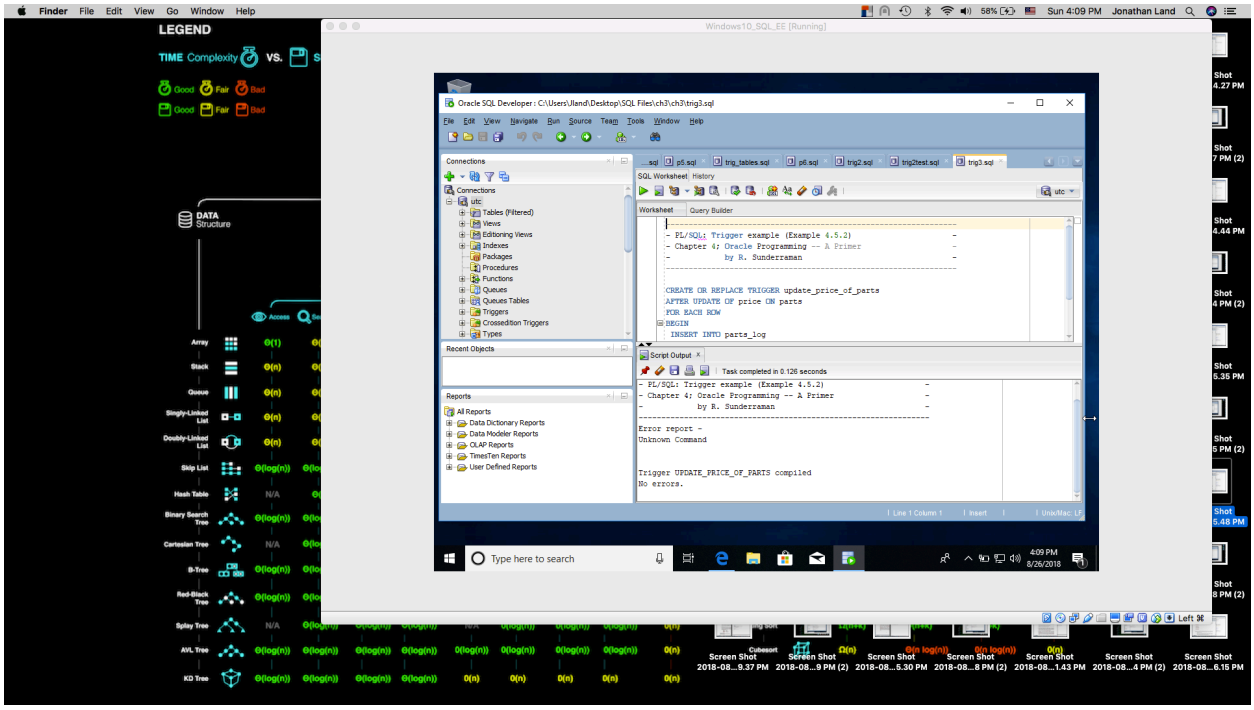
c. trig3 is defined on the parts table and is triggered when the price column is updated. Each time someone updates the price of a particular part, the trigger makes an entry in a log file of this update along with the userid of the person performing the update and the date the update. The log file is created using

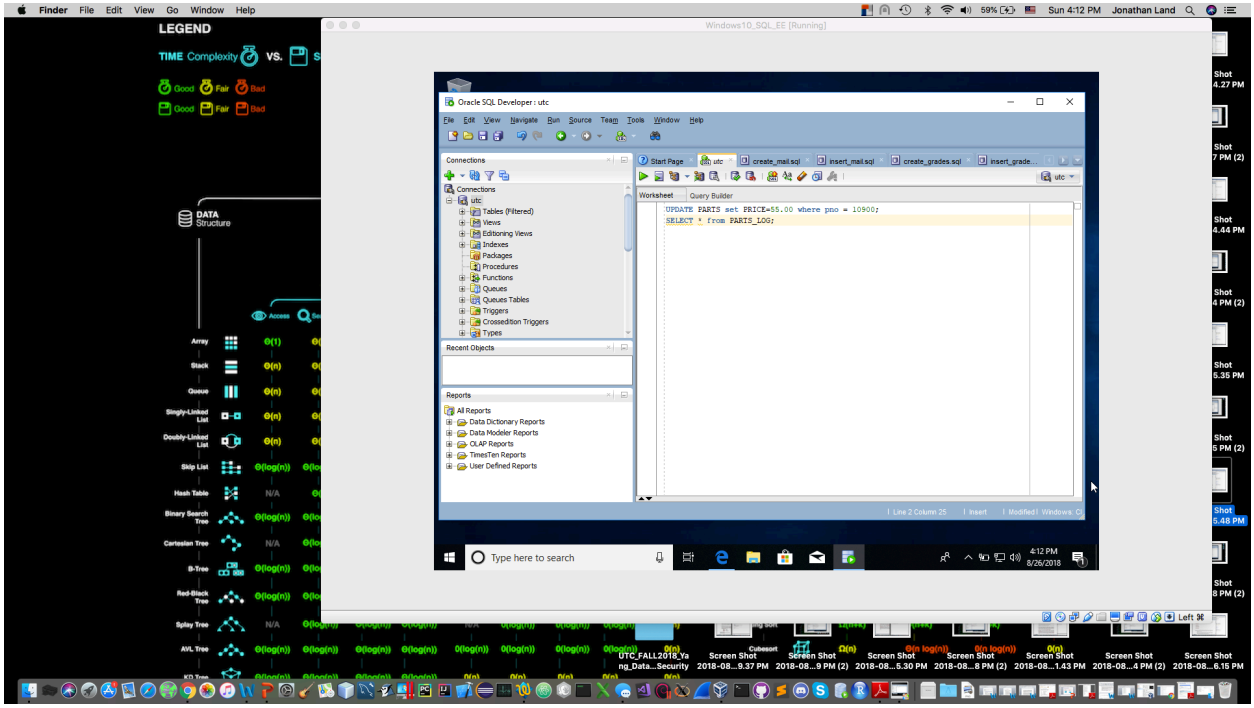
```
create table parts_log(
        pno             number(5),
        username        char(8),
```

```
update_date     date,
old_price       number(6,2),
new_price       number(6,2));
```

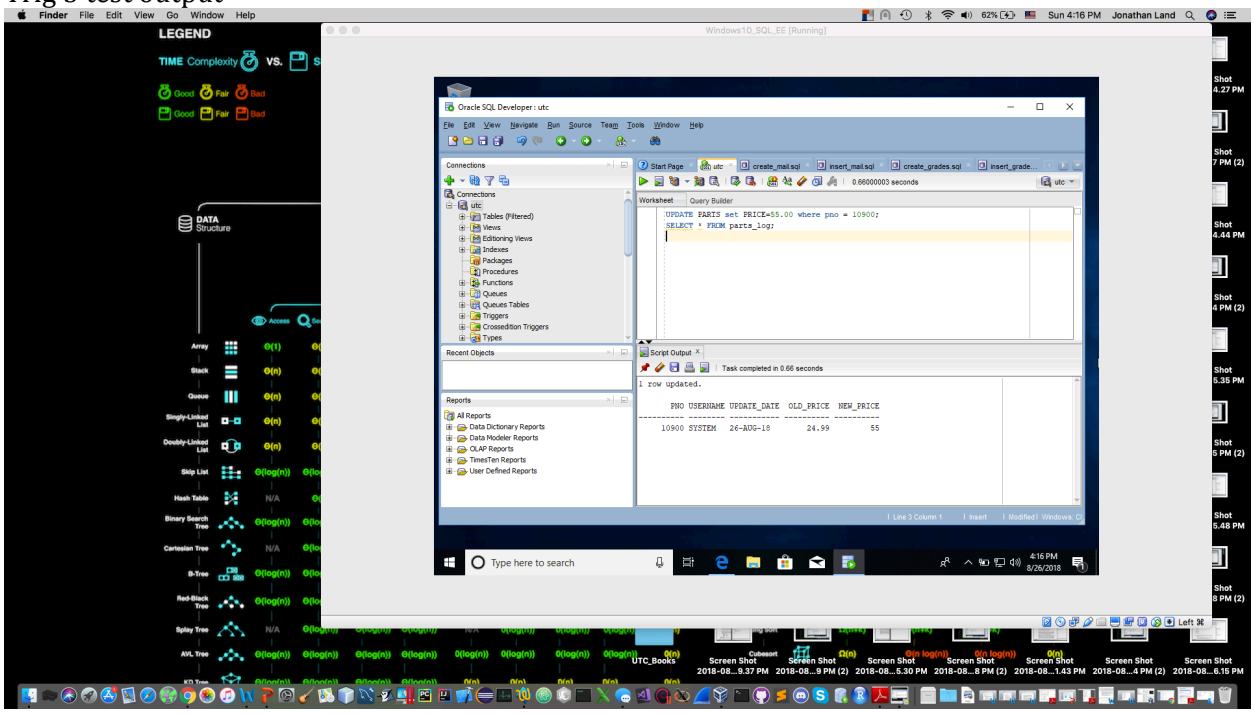


4. Test trig3.sql by typing:
   update parts set price=55.00 where pno=10900;
   select * from parts_log;





Trig 3 test output



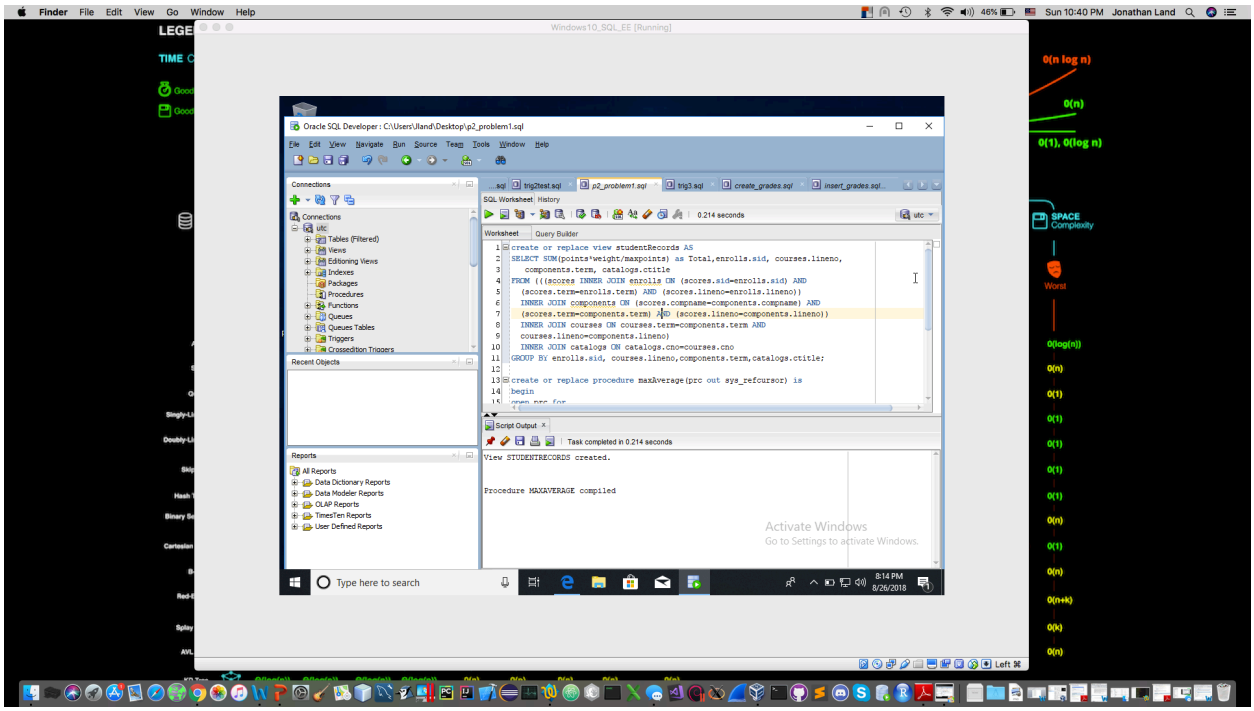
5. Create Grade book database using scripts of create_grade.sql and insert_grades.sql

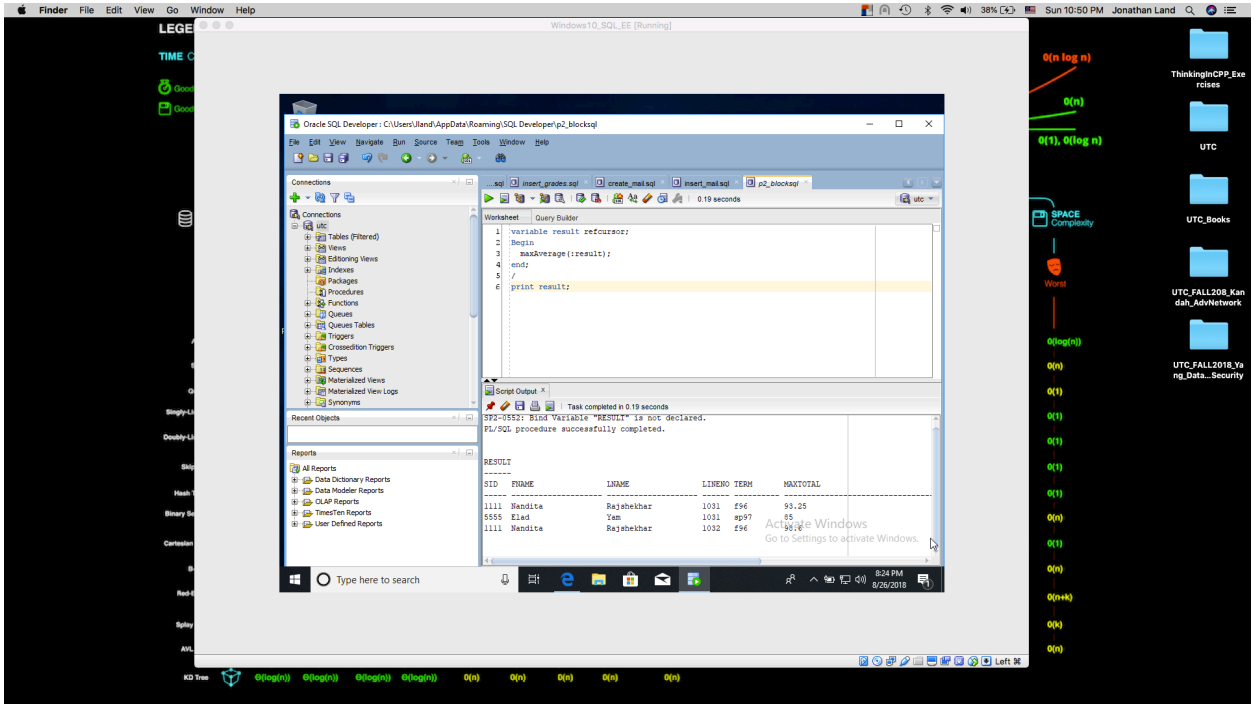
Problem one (40 points): Write a PL/SQL procedure that finds the student with the highest overall average in every course. In case there is more than one student tied for the highest average, the procedure should return all of the students. The results should be returned in a PL/SQL table of records, where each record has the field term, line number, course title, student ID, student name, and overall average. Also, write an anonymous

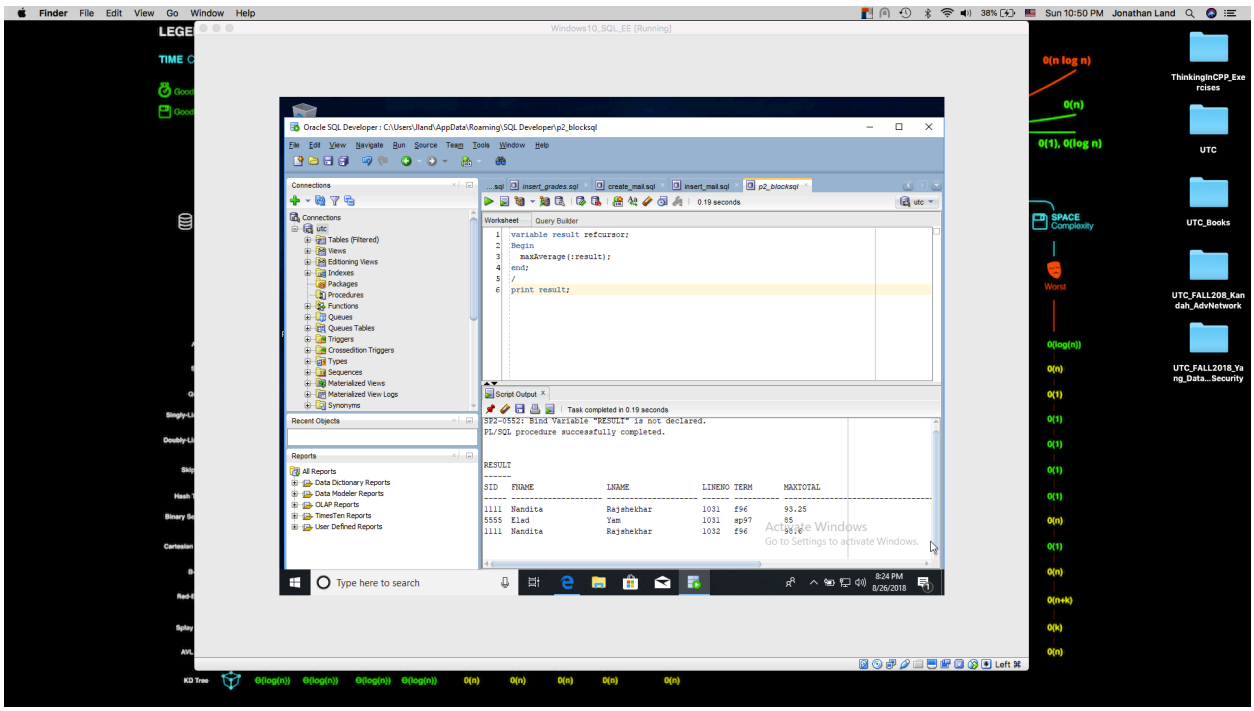
PL/SQL block that makes a call to the procedure and prints the results to the standard output. Refer to p4.sql for some ideas.

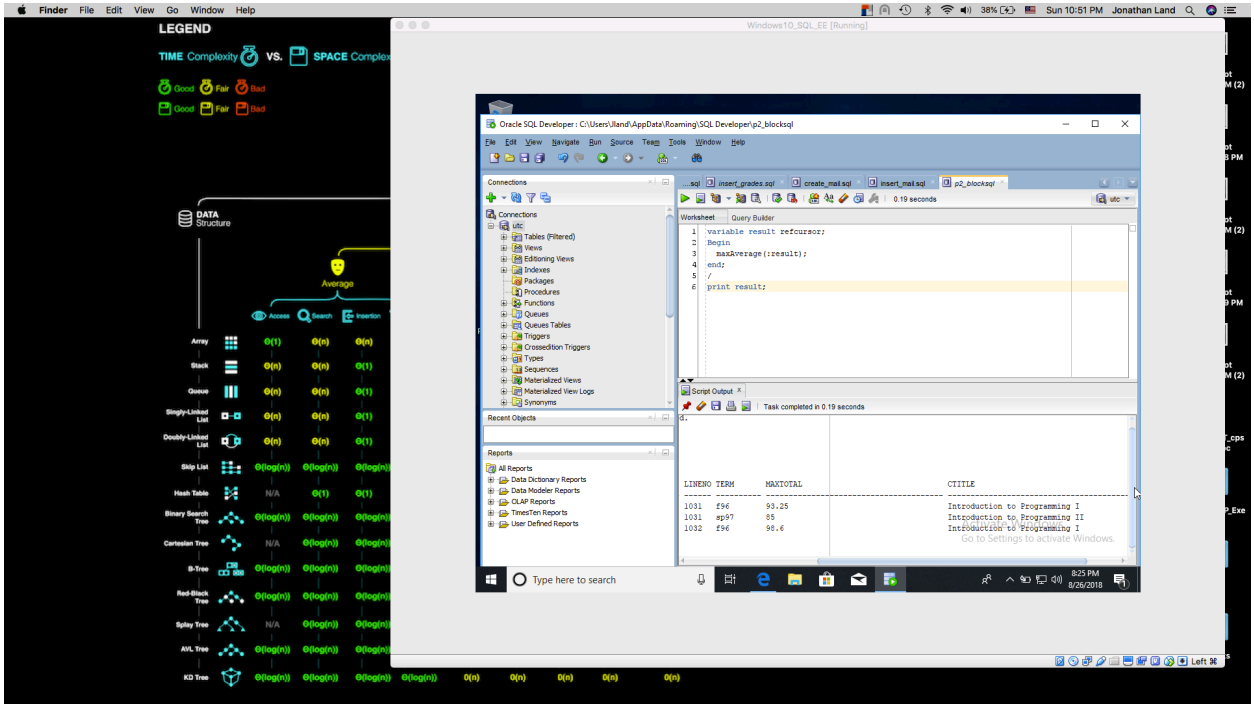
For this problem, two sql scripts were created (p2_problem1.sql and p2_blocksql.sql [code below]). In p2_problem1.sql the basic structure/outline for getting the average was as follows: (1) create a view (virtual table) called studentRecords so that the data would always be the current; (2) use the sum function to get the average of student data and then use an alias to reference the data; (3) use inner joins and group the student data from different tables, fields, etc.; (4) create a procedure called **maxAverage** and open a query using sys_refcursor output parameter to order and find the maxAverage of the total combined averages of the students (the cursor starts before the first row in the result set and then "begin" is where the executable table starts); (5) select student sid, fname, lname, studMax.linene, studMax.term, and MAXTotal, ctitle **from studentRecords** created earlier and group that data; (6) select everything from studentRecords and order the max averages via student id, student marks, and max total. The course number and title also show in the rows.

As can be seen below, both studentRecords and maxAverage were created below and the top averages are listed.




Oracle SQL Developer : C:\Users\Jland\AppData\Roaming\SQL Developer\p2_blocksql
variable result refcursor;
Begin
maxAverage(:result);
end;
/
print result;
Script Output
Task completed in 0.19 seconds
SP2-0552: Bind Variable "RESULT" is not declared.
PL/SQL procedure successfully completed.
RESULT
SID FNAME LNAME LINENO TERM MAXTOTAL
1111 Nandita Rajshekhar 1031 f96 93.25
5555 Elad Yam 1031 sp97 85
1111 Nandita Rajshekhar 1032 f96 93.25
Activate Windows
Go to Settings to activate Windows.
Type here to search
8:24 PM
8/26/2018


Oracle SQL Developer : C:\Users\Jland\AppData\Roaming\SQL Developer\p2_blocksql
variable result refcursor;
Begin
maxAverage(:result);
end;
/
print result;
Script Output
Task completed in 0.19 seconds
SP2-0552: Bind Variable "RESULT" is not declared.
PL/SQL procedure successfully completed.
RESULT
SID FNAME LNAME LINENO TERM MAXTOTAL
1111 Nandita Rajshekhar 1031 f96 93.25
5555 Elad Yam 1031 sp97 85
1111 Nandita Rajshekhar 1032 f96 93.25
Activate Windows
Go to Settings to activate Windows.
Type here to search
8:24 PM
8/26/2018



Problem 1 PL/SQL Code

p2_problem1.sql (maxAverage)

```
create or replace view studentRecords AS
SELECT SUM(points*weight/maxpoints) as Total,enrolls.sid, courses.lineno,
   components.term, catalogs.ctitle

FROM (((scores INNER JOIN enrolls ON (scores.sid=enrolls.sid) AND
  (scores.term=enrolls.term) AND (scores.lineno=enrolls.lineno))
  INNER JOIN components ON (scores.compname=components.compname) AND
  (scores.term=components.term) AND (scores.lineno=components.lineno))
  INNER JOIN courses ON courses.term=components.term AND
  courses.lineno=components.lineno)
  INNER JOIN catalogs ON catalogs.cno=courses.cno

GROUP BY enrolls.sid, courses.lineno,components.term,catalogs.ctitle;

create or replace procedure maxAverage(prc out sys_refcursor) is

begin
       open prc for

Select students.sid,fname,lname,studMax.lineno,studMax.term,MAXTotal,ctitle

FROM ((SELECT MAX(Total) as MAXTotal,studentRecords.lineno, studentRecords.term

       FROM studentRecords

       GROUP BY studentRecords.lineno,studentRecords.term)
```

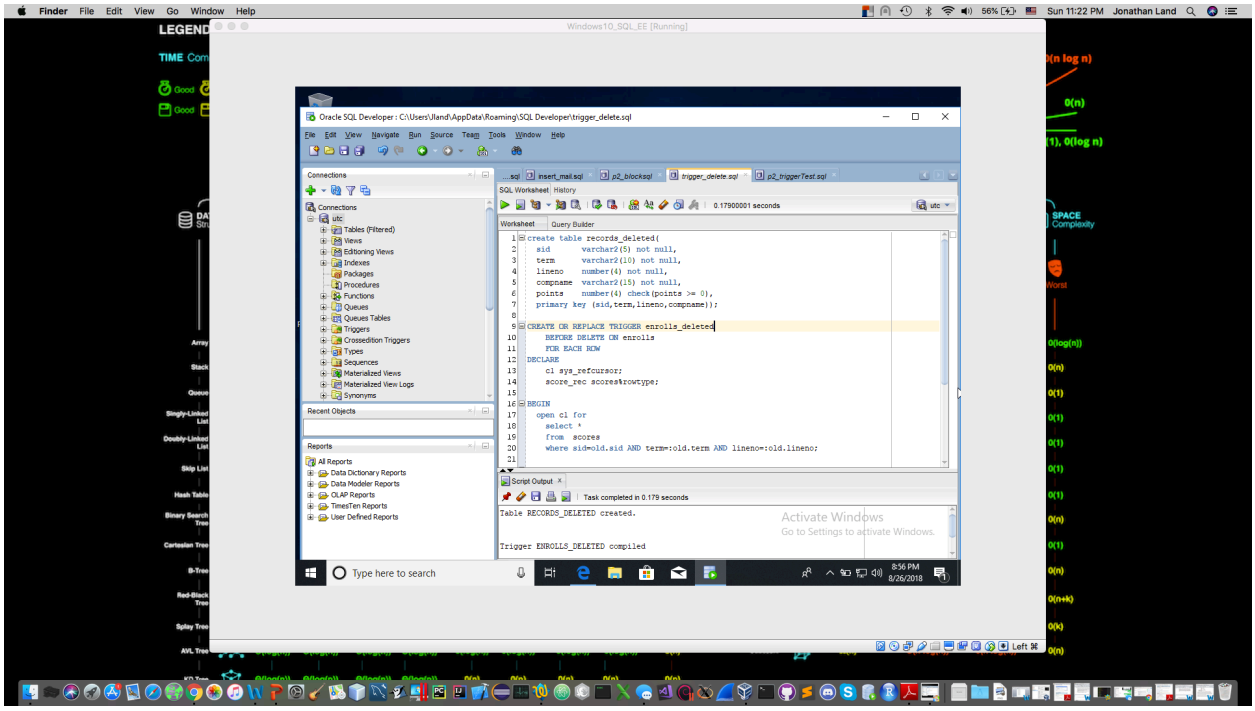
```
            studMax INNER JOIN

            (SELECT * FROM studentRecords) studMark ON (MAXTotal=Total) and
            (studMax.lineno=studMark.lineno)AND (studMax.term=studMark.term))

            INNER JOIN students ON students.sid=studMark.sid;
end;
```

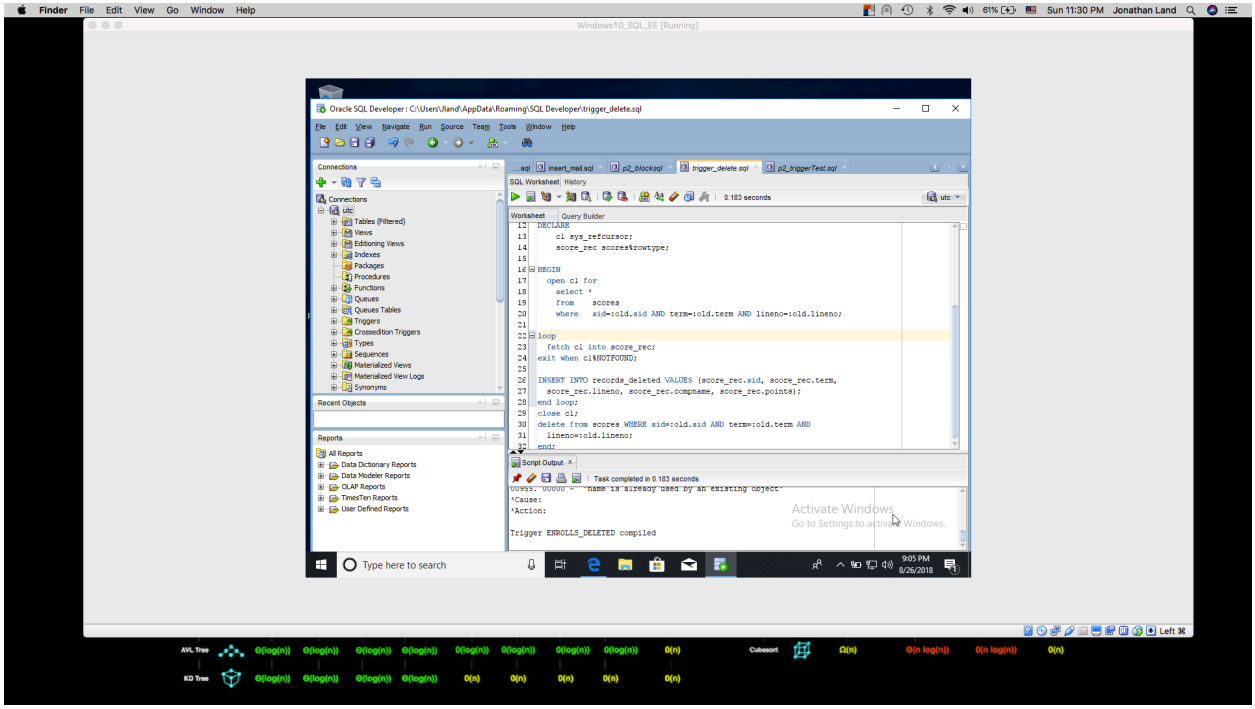
p2_blocksql.sql (tester)

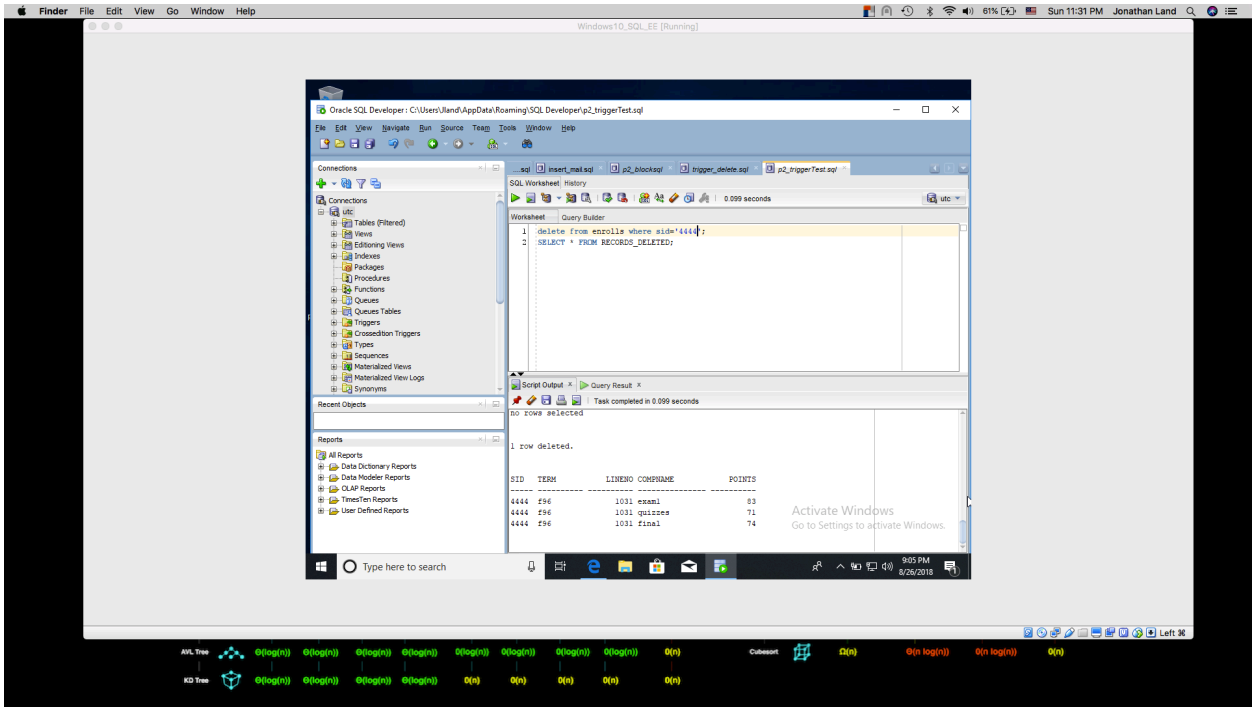
```
variable result refcursor;
Begin
  maxAverage(:result);
end;

/
print result;
```

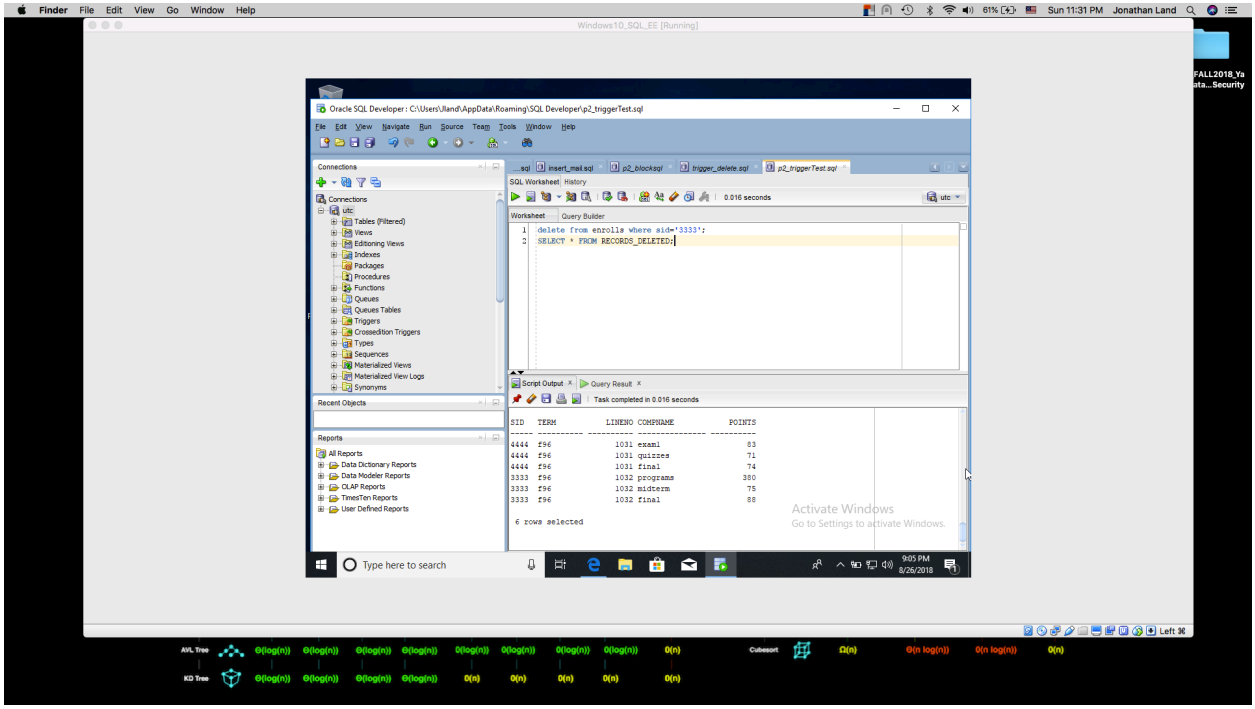
Problem two (40 points): Write a trigger that fires when a row is deleted from the enrolls table. The trigger should record the dropped student's scores in a temporary table, called deleted_scores, and cascade the deletes to ensure that the referential integrity constraints are maintained.

For each record we are doing an insert and a delete and will do this until the table is empty (using sys_refcursor). Instead of deleting a row, the trigger kicks off, copies the deleted record in records deleted and then it *actually* deletes it (screenshots and code below)




Oracle SQL Developer : C:\Users\lland\AppData\Roaming\SQL Developer\trigger_delete.sql
DECLARE
cl sys_refcursor;
score_rec scores%rowtype;
BEGIN
open cl for
select *
from scores
where sid=:old.sid AND term=:old.term AND lineno=:old.lineno;
loop
fetch cl into score_rec;
exit when cl%NOTFOUND;
INSERT INTO records_deleted VALUES (score_rec.sid, score_rec.term,
score_rec.lineno, score_rec.compname, score_rec.points);
end loop;
close cl;
delete from scores WHERE sid=:old.sid AND term=:old.term AND
lineno=:old.lineno;
end;
Task completed in 0.183 seconds
*Cause:
*Action:
Trigger ENROLLS_DELETED compiled
Activate Windows
Go to Settings to activate Windows.


Oracle SQL Developer : C:\Users\lland\AppData\Roaming\SQL Developer\p2_triggerTest.sql
delete from enrolls where sid='4444';
SELECT * FROM RECORDS_DELETED;
Task completed in 0.099 seconds
no rows selected
1 row deleted.
SID TERM LINENO COMPNAME POINTS
4444 f96 1031 exam1 83
4444 f96 1031 quizzes 71
4444 f96 1031 final 74
Activate Windows
Go to Settings to activate Windows.



Trigger

```
create table records_deleted(
  sid       varchar2(5) not null,
  term      varchar2(10) not null,
  lineno    number(4) not null,
  compname  varchar2(15) not null,
  points    number(4) check(points >= 0),
  primary key (sid,term,lineno,compname));

CREATE OR REPLACE TRIGGER enrolls_deleted
    BEFORE DELETE ON enrolls
    FOR EACH ROW

DECLARE
        c1 sys_refcursor;
        score_rec scores%rowtype;

BEGIN
      open c1 for
       select * from  scores
       where   sid=:old.sid AND term=:old.term AND lineno=:old.lineno;

loop
       fetch c1 into score_rec;
       exit when c1%NOTFOUND;

       INSERT INTO records_deleted VALUES (score_rec.sid, score_rec.term,
        score_rec.lineno, score_rec.compname, score_rec.points);
```

```
end loop;
close c1;

       delete from scores WHERE sid=:old.sid AND term=:old.term AND lineno=:old.lineno;

end;
```

Trigger Tester

```
delete from enrolls where sid='3333';

SELECT * FROM RECORDS_DELETED;
```