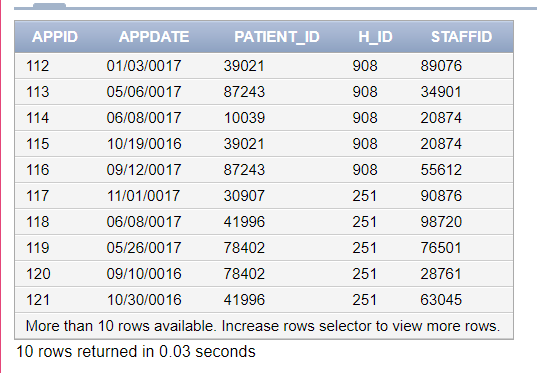
**Minsun Kim**

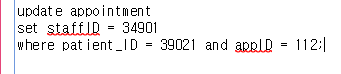
**Due: 18/Dec/2017**

1. In one SQL window, change the doctor of a previously performed procedure. Don’t commit. In another SQL window, change the doctor again for the same procedure. Don’t commit. Explain your results. Resolve the problem. Disable the auto commit flag at the top of the window before performing this operation.

Original output of appointment table:

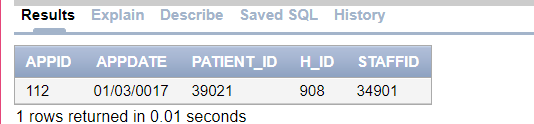


Step 1: Update at the first SQL window.

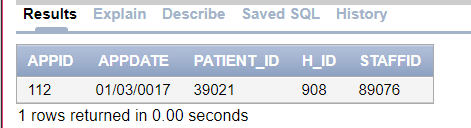




* Result of this first SQL window:



Step2: Update same appointment at the second SQL window



Before update, the data that I’ve changed at the first SQL window was not saved.

Update



Then..



Result: It is not working. It has an error.

Solution:

It will make an error when more than one person access to the same data. For make sure the new data(one that updated) is saved, we should commit it. Otherwise, it will make an error. Like the first result that I showed before update the data in second SQL window, it shows the previous data which is the one that saved before first SQL window updated.

1. In one SQL window, delete all procedures for the patient John Smith. Don’t commit. In another SQL window, change all procedures for patient John Smith. Don’t commit. Explain your result. Resolve the problem. Create a backup of your table before implementing. To create a backup table, enter create table <table> as select \* from <original table>; Commit; Then you can rename a table using the rename table commit. Disable the auto commit flag at the top of the window before performing this operation.

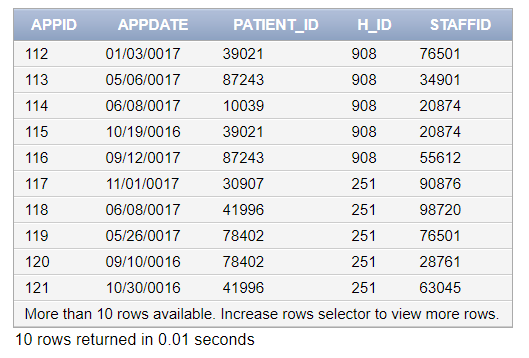
Step 1: Create a backup table;



 🡺

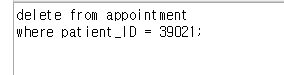
Step 2: Check the backup table is created well.



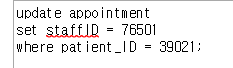


Step 3: Delete all procedures from John Smith.





Step 4: Then change all procedures for John Smith.



Result: It is not working.

Solution:

For delete or update data, the user should commit the data. Otherwise, it will be impossible to update the data. Second SQL will not be able to update the data until the first SQL window commit or rollback the data.

1. In one SQL window, change the zip code for patient John Smith. Don’t commit. In another SQL window, change the address of the patient John Smith. Don’t commit. Quit both Oracle sessions. Login to Oracle and display all information for the patient John Smith. Explain your results. Disable the auto commit flag at the top of the window before performing this operation.

Step 1: check the data first.



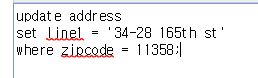
Step 2: update zipcode for John Smith.



Step 3: check the original address for John Smith from second SQL window.

Step 4: update the address for John Smith.



Step 5: close all and reopen and check the data.



Result: Not changed.

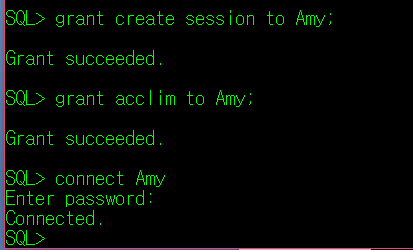
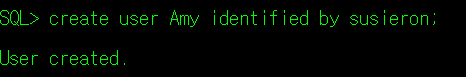
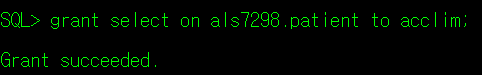
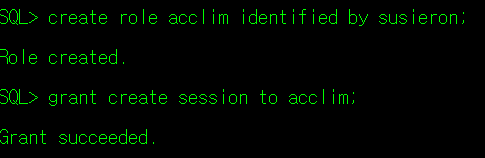
Solution:

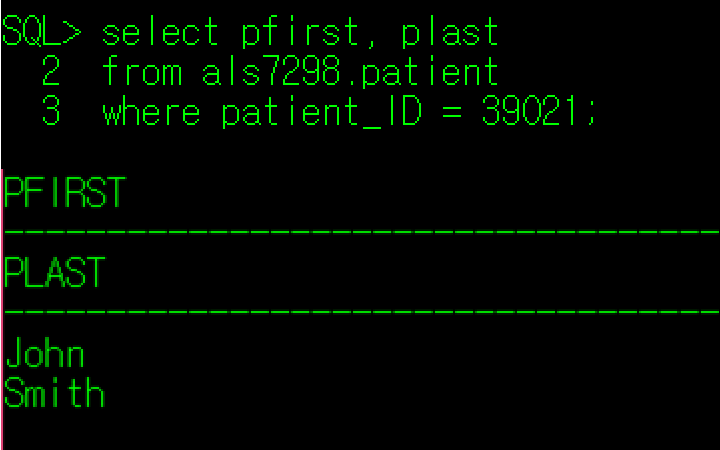
If the user wants to save one between first and second SQL window, then the user should commit the value before close it.

1. Utilize Oracle SQL security roles to limit access to display and not edit patient non-treatment information saved for patient John Smith. For instance name, address, date of birth. This security role will be able to only display and not add, delete or change patient information. Identify the SQL operations to implement and demonstrate the functionality of the security roles.

Steps:

* Create role
* Grant create session
* Grant select on patient table
* Create new user
* Grant create session to new user
* Grant role to new user
* Connect new user
* Check the access



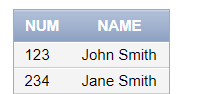


1. Utilize Oracle SQL security to allow users to create new patient procedures, but prevent the deletion or changes to patient procedures. Identify the SQL operations to implement and demonstrate the functionality of the security roles.

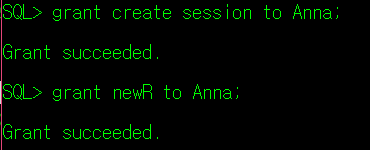
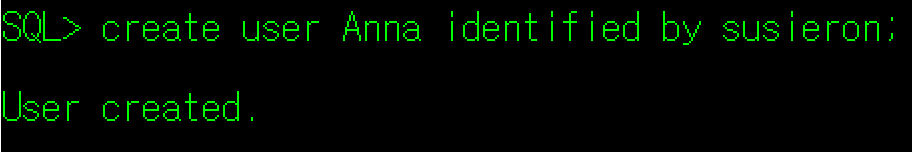
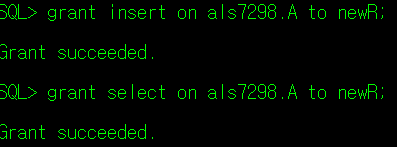
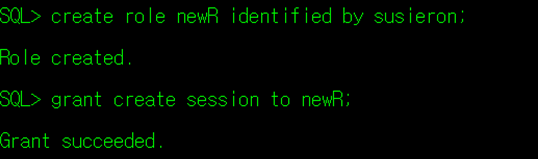
* Do same process with #4 but add
* Grant insert on als7298.A to newR

//For this question, I created table call A that has 2 types of data (num, name)

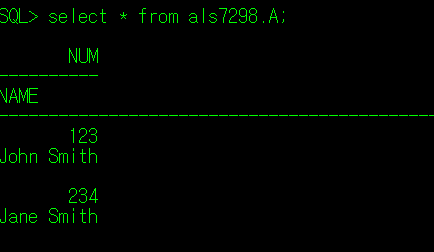




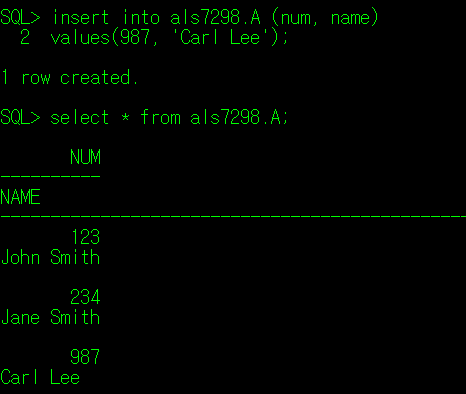
This is original data in A table.



//check the select and insert privileges are working on Anna



//on Anna insert value into A



1. Use the SQL DESCRIBE operation to display the structure for all tables.

