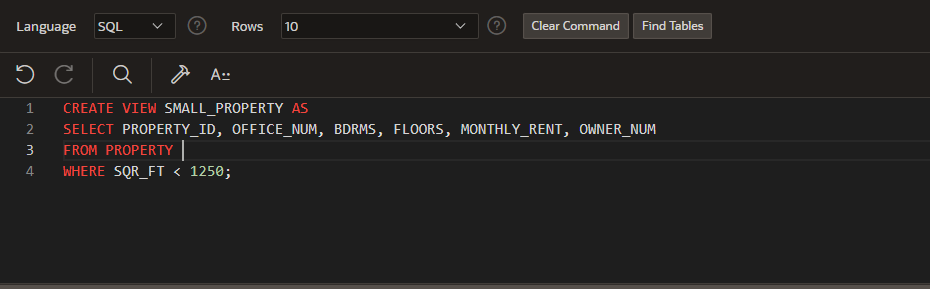
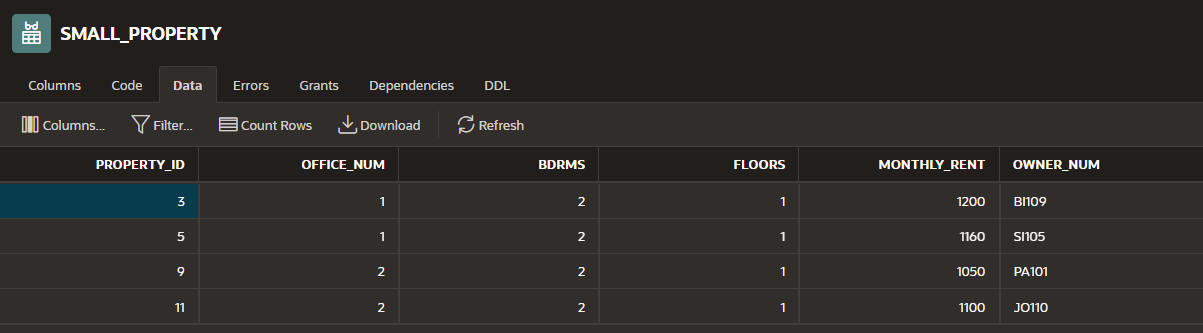
Collaboration: I worked alone.

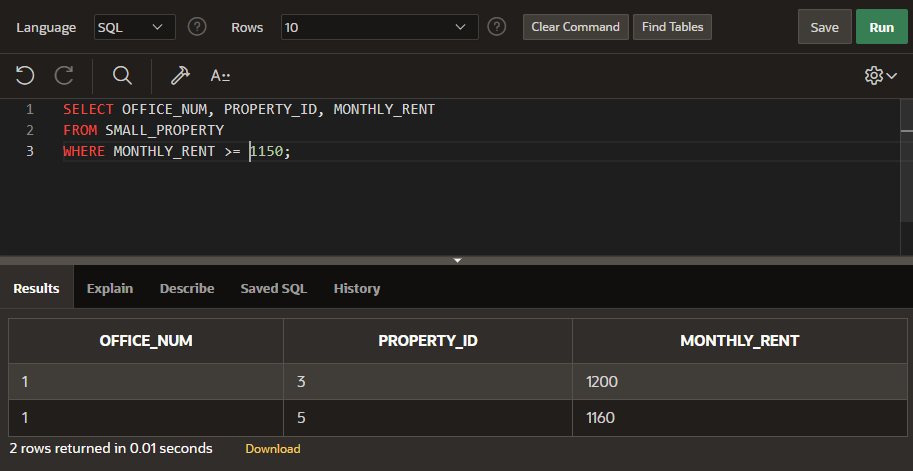
1. Create a view named SMALL\_PROPERTY. It consists of the property ID, office number, bedrooms, floor, monthly rent, and owner number for every property whose square footage is less than 1,250 square feet.

a. Write and execute the CREATE VIEW command to create the SMALL\_PROPERTY view.

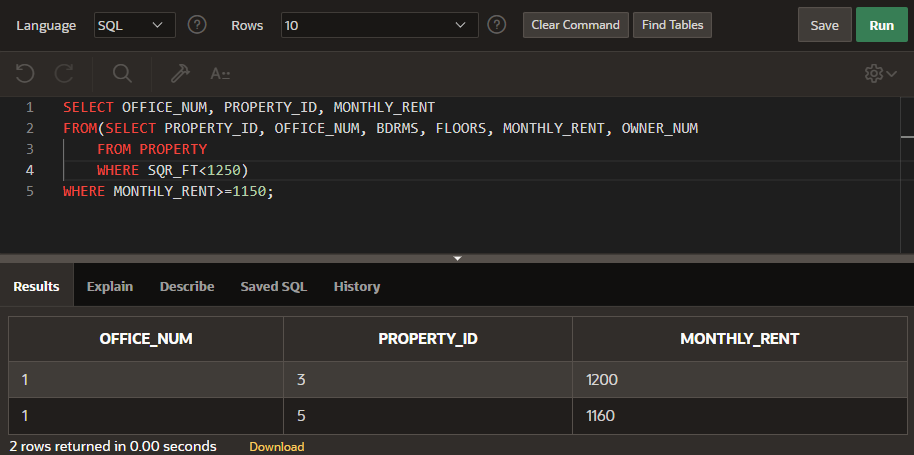




b. Write and execute the command to retrieve the office number, property ID, and monthly rent for every property in the SMALL\_PROPERTY view with a monthly rent of $1150 or more.



c. Write and execute the query that the DBMS actually executes.

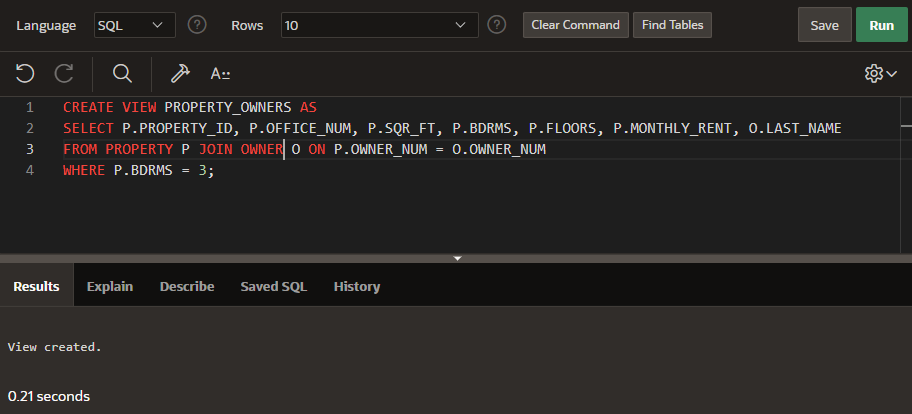


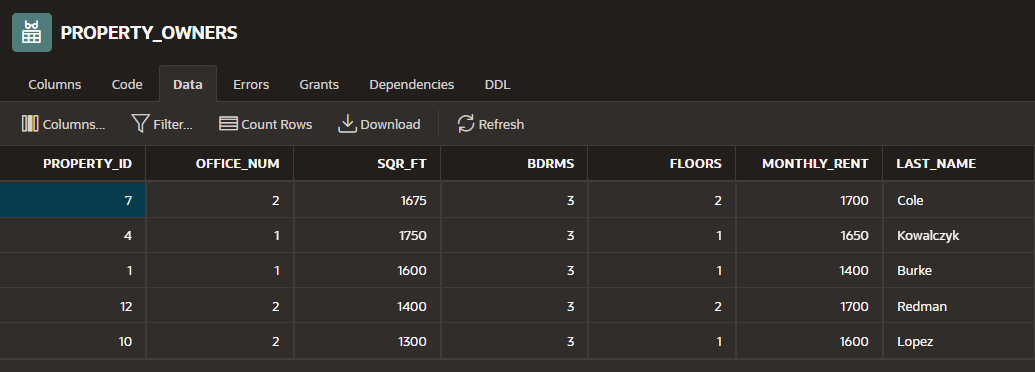
d. Does updating the database through this view create any problems? If so, what are they? If not, why not?

- No, updating the database through this view does not create any problems because the view doesn’t have a calculated field, and each row is updatable.

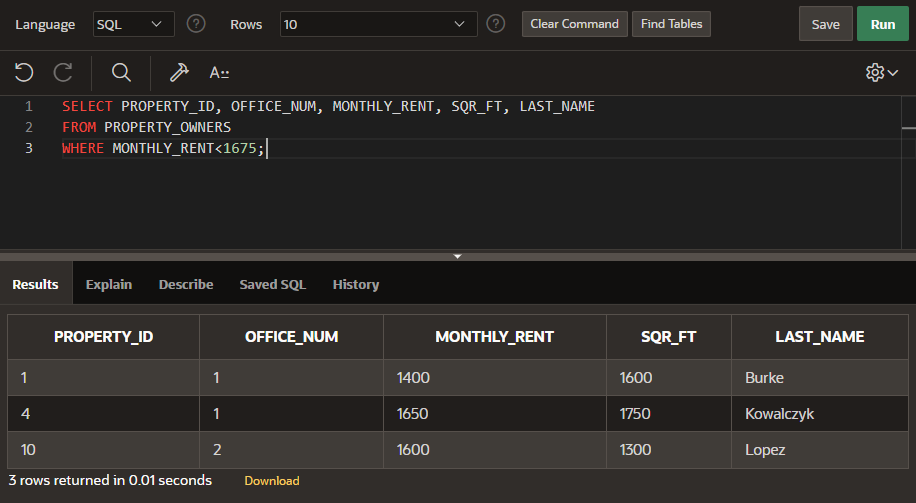
2. Create a view named PROPERTY\_OWNERS. It consists of the property ID, office number, square footage, bedrooms, floors, monthly rent, and owner’s last name for every property in which the number of bedrooms is three.

1. Write and execute the CREATE VIEW command to create the PROPERTY\_OWNERS view.

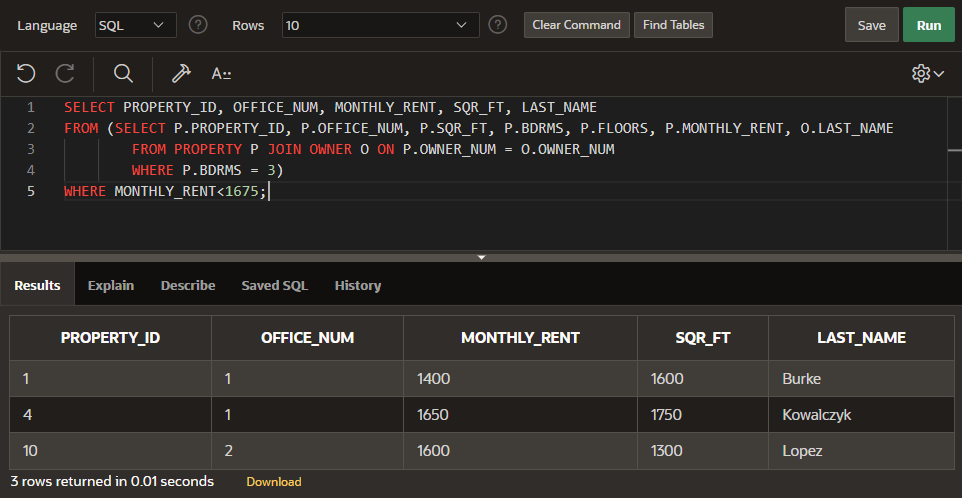




1. Write and execute the command to retrieve the property ID, office number, monthly rent, square footage, and owner’s last name for every property in the PROPERTY\_OWNERS view with a monthly rent of less than $1675.



1. Write and execute the query that the DBMS actually executes.

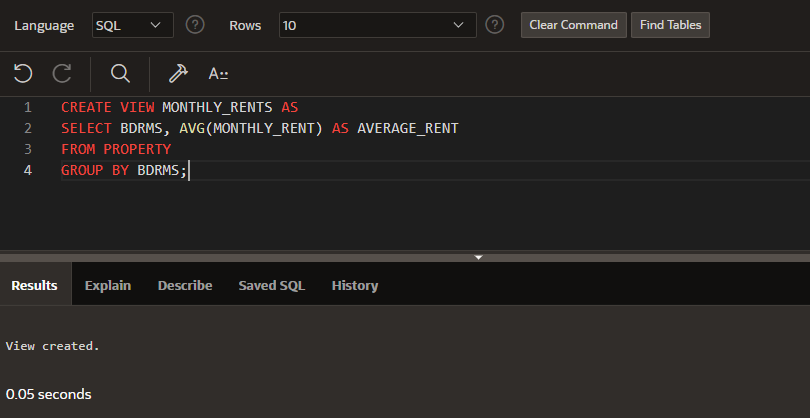


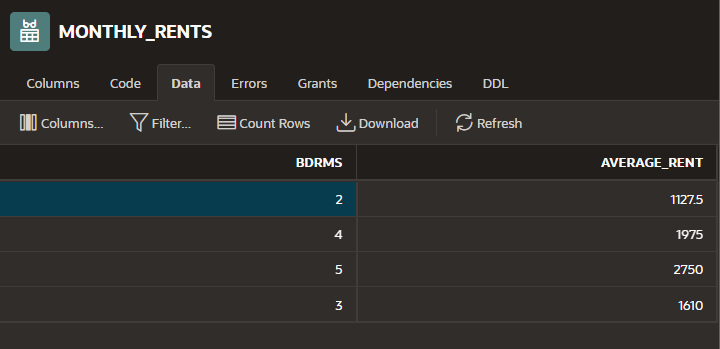
1. Does updating the database through this view create any problems? If so, what are they? If not, why not?

- Yes, updating the database through this view does create problems because the view is created using two tables. It would not be able to update owner number in both tables.

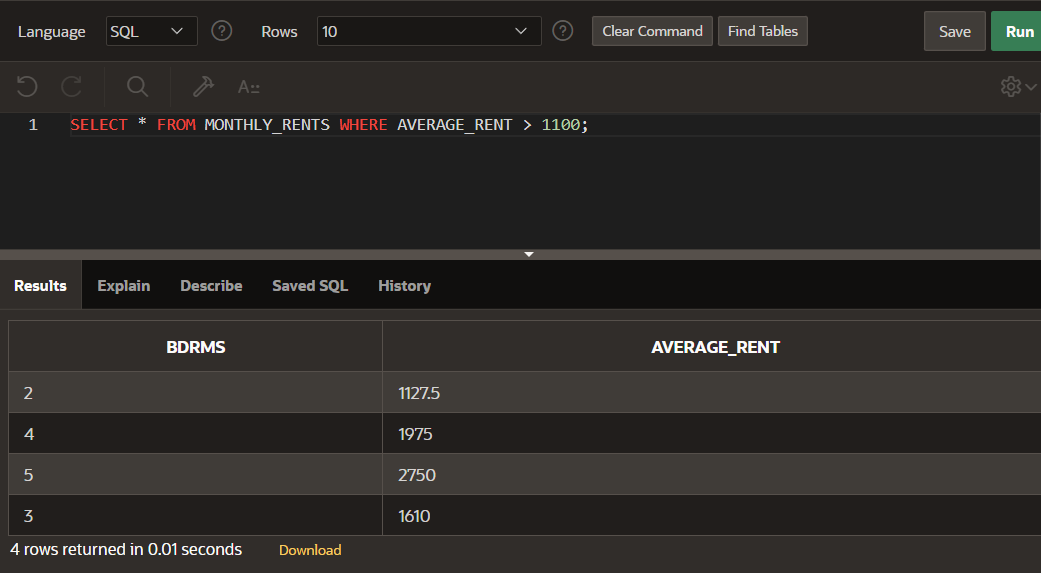
3. Create a view named MONTHLY\_RENTS. It consists of two columns: The first is the number of bedrooms, and the second is the average monthly rent for all properties in the PROPERTY table that have that number of bedrooms. Use AVERAGE\_RENT as the column name for the average monthly rent. Group and order the rows by number of bedrooms.

1. Write and execute the CREATE VIEW command to create the MONTHLY\_RENTS view.

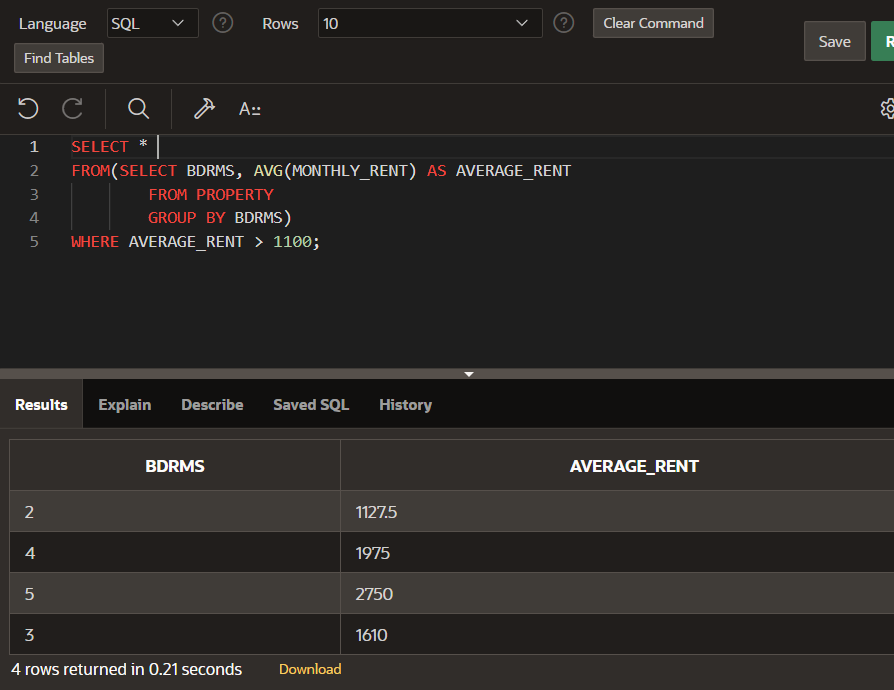




1. Write and execute the command to retrieve the number of bedrooms and average monthly rent for which the average monthly rent is greater than $1,100.



1. Write and execute the query that the DBMS actually executes.

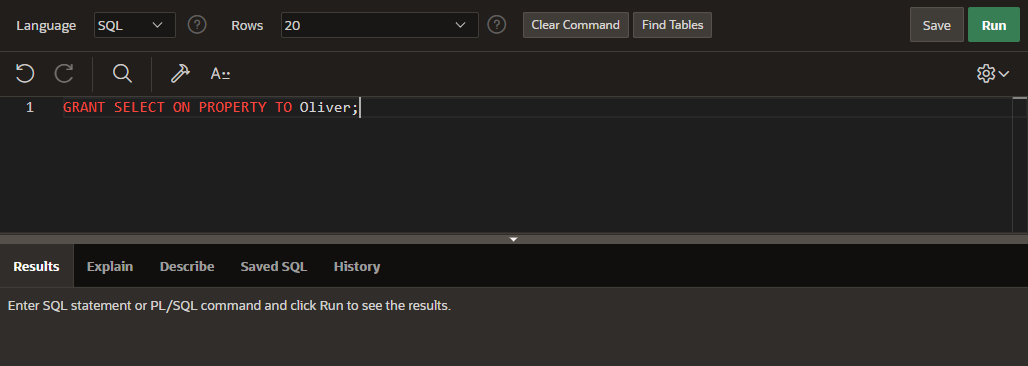


1. Does updating the database through this view create any problems? If so, what are they? If not, why not?

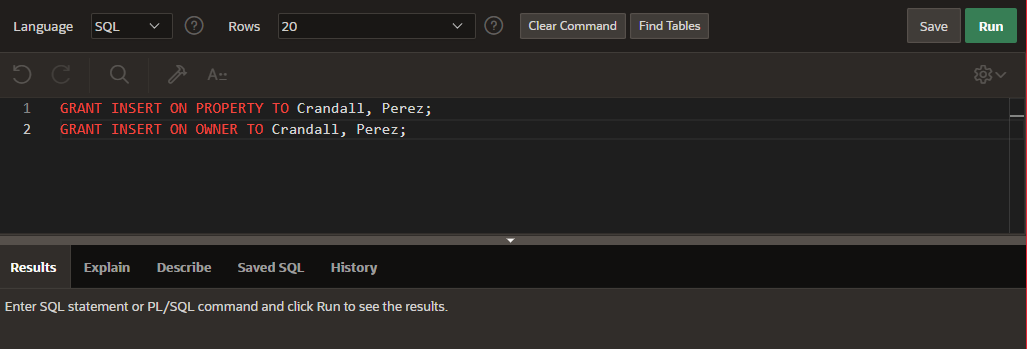
- Yes, updating the database through this view does create problems because the view is using a calculated value (AVG) which conflicts with updates.

4. Write, but do not execute, the commands to grant the following privileges:

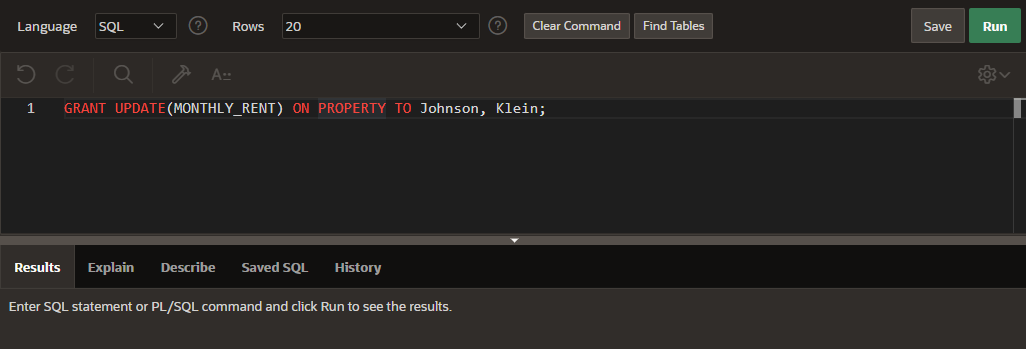
1. User Oliver must be able to retrieve data from the PROPERTY table.



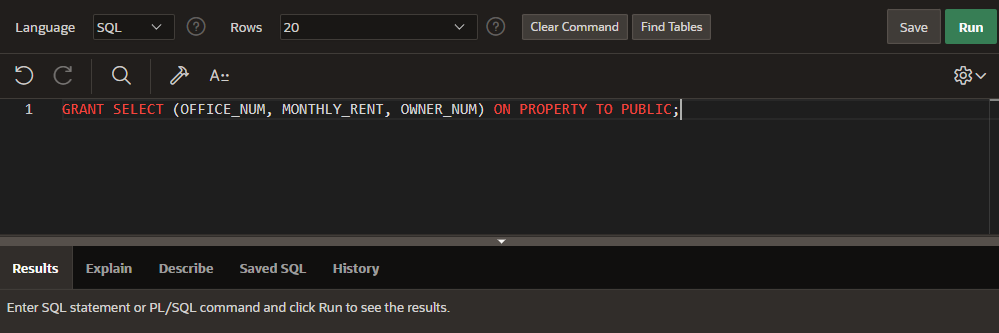
1. Users Crandall and Perez must be able to add new owners and properties to the database.



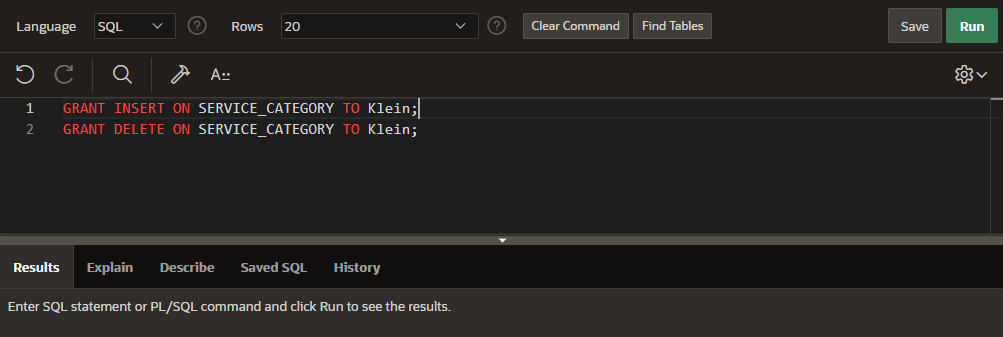
1. Users Johnson and Klein must be able to change the monthly rent of any unit.



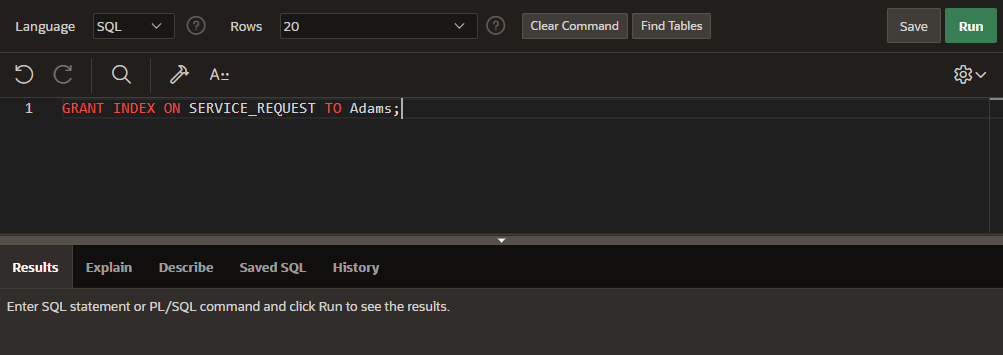
1. All users must be able to retrieve the office number, monthly rent, and owner number for every property.



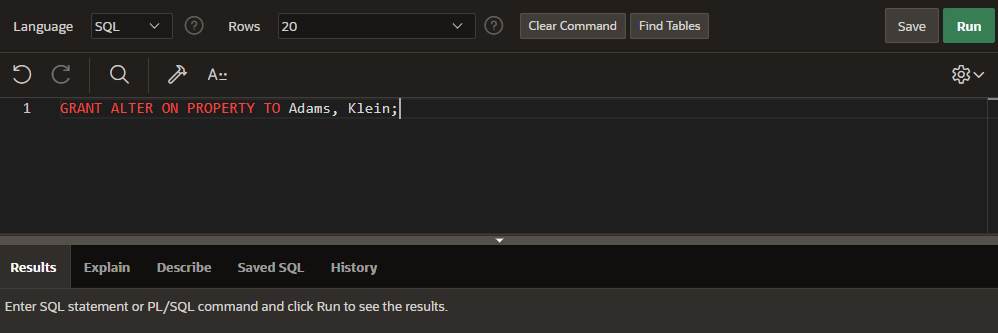
1. User Klein must be able to add and delete service categories.



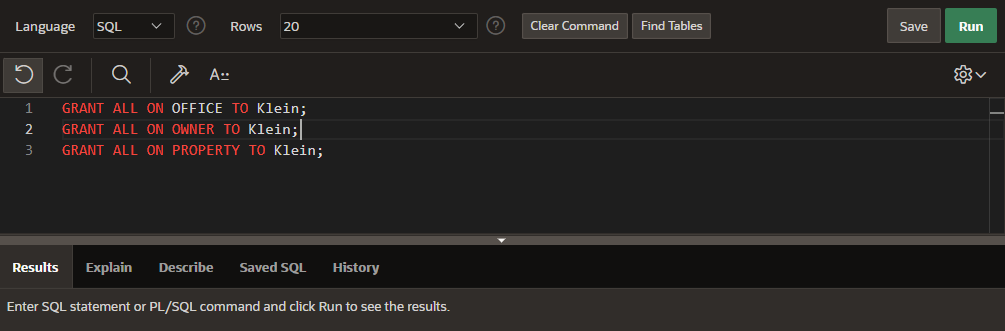
1. User Adams must be able to create an index on the SERVICE\_REQUEST table.



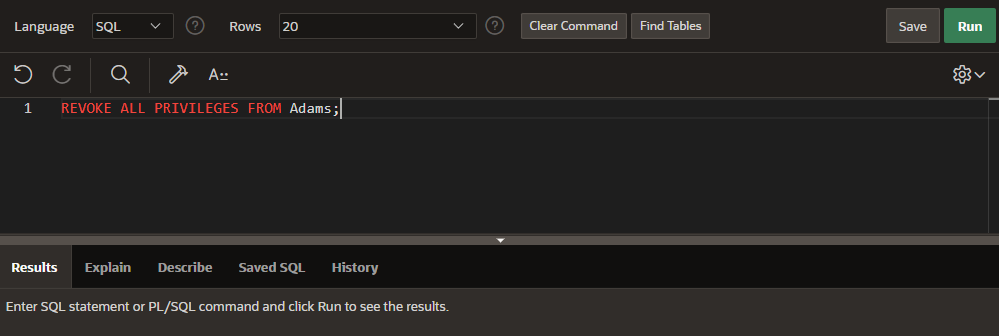
1. Users Adams and Klein must be able to change the structure of the PROPERTY table.



1. User Klein must have all privileges on the OFFICE, OWNER, and PROPERTY tables.

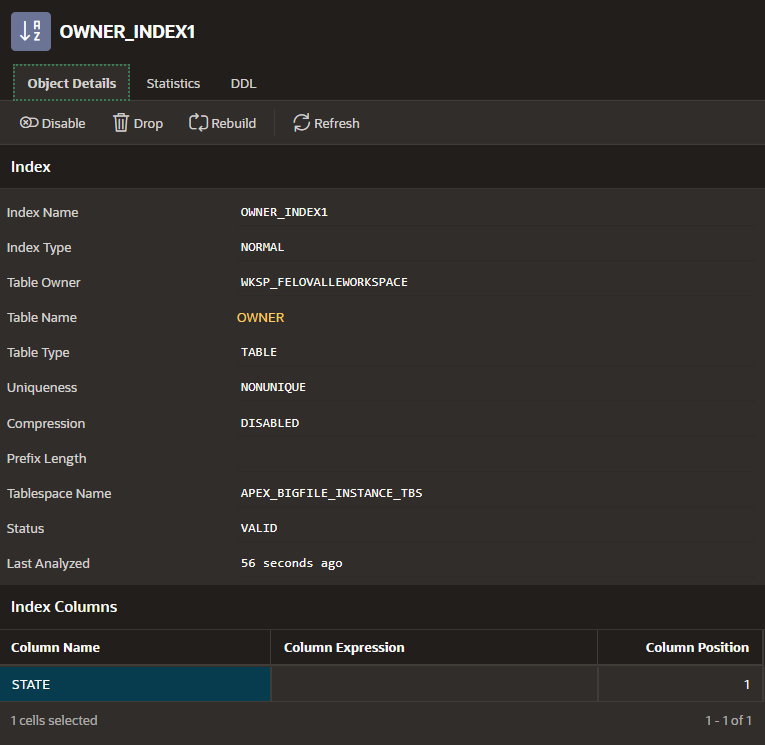
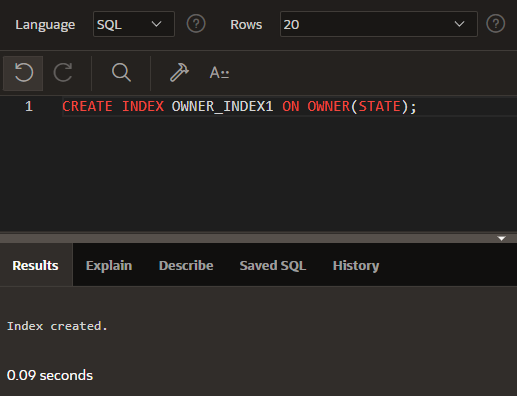


5. Write, but do not execute, the command to revoke all privileges from user Adams.

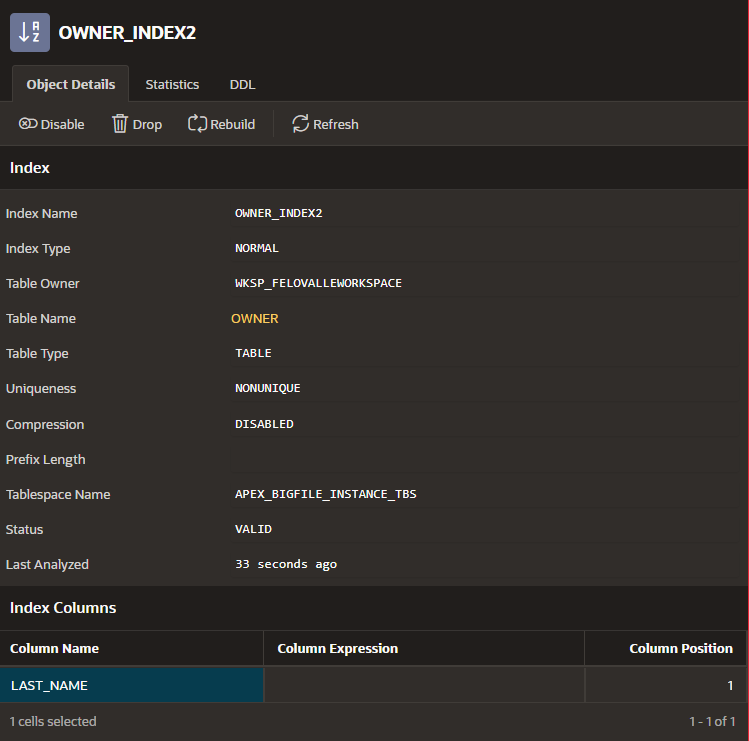
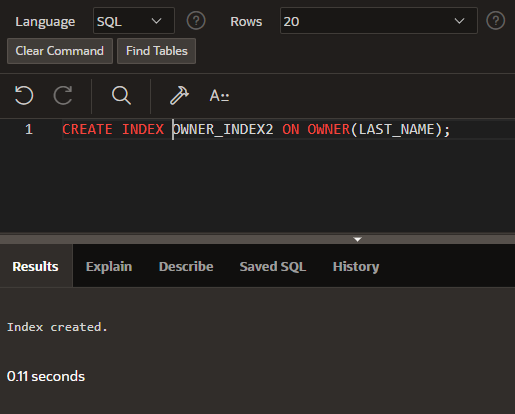


6. Create the following indexes:

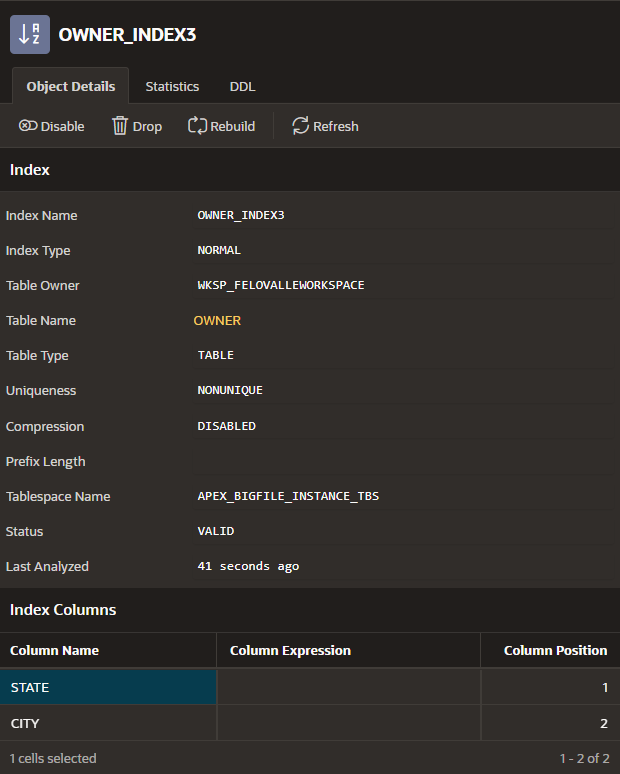
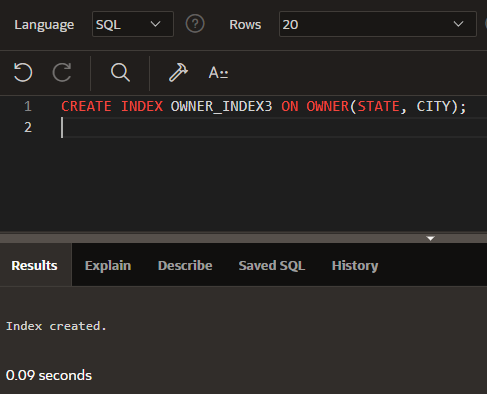
1. Create an index named OWNER\_INDEX1 on the STATE column in the OWNER table.

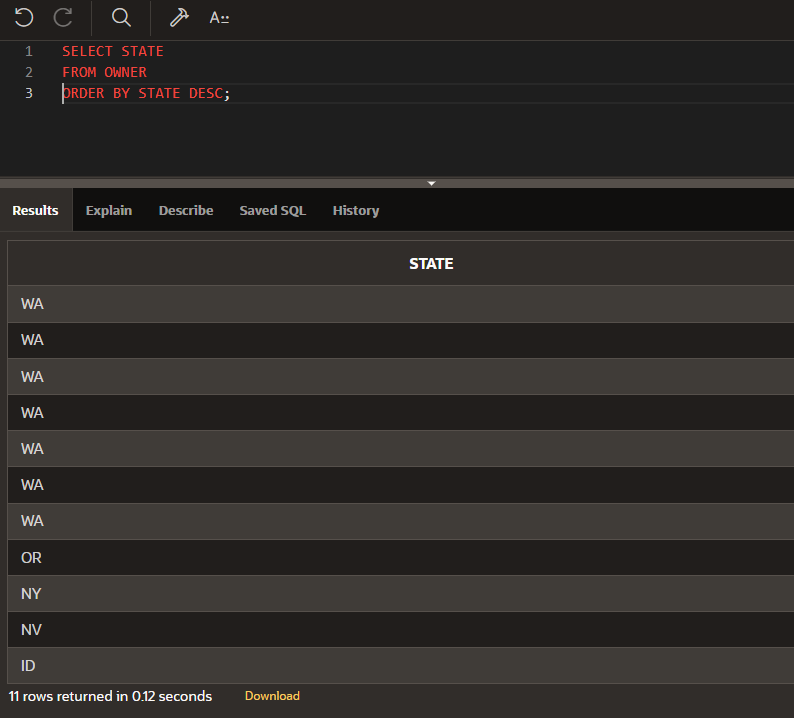


1. Create an index named OWNER\_INDEX2 on the LAST\_NAME column in the OWNER table.

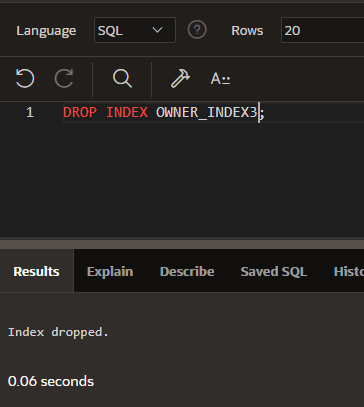


1. Create an index named OWNER\_INDEX3 on the STATE and CITY columns in the OWNER table. List the states in descending order.



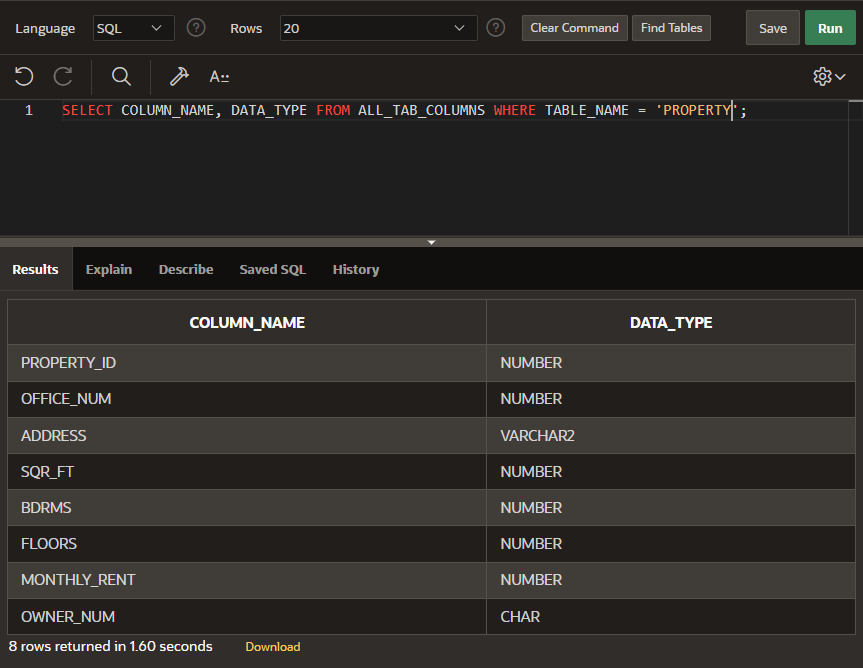


7. Delete the OWNER\_INDEX 3 index from the OWNER table.

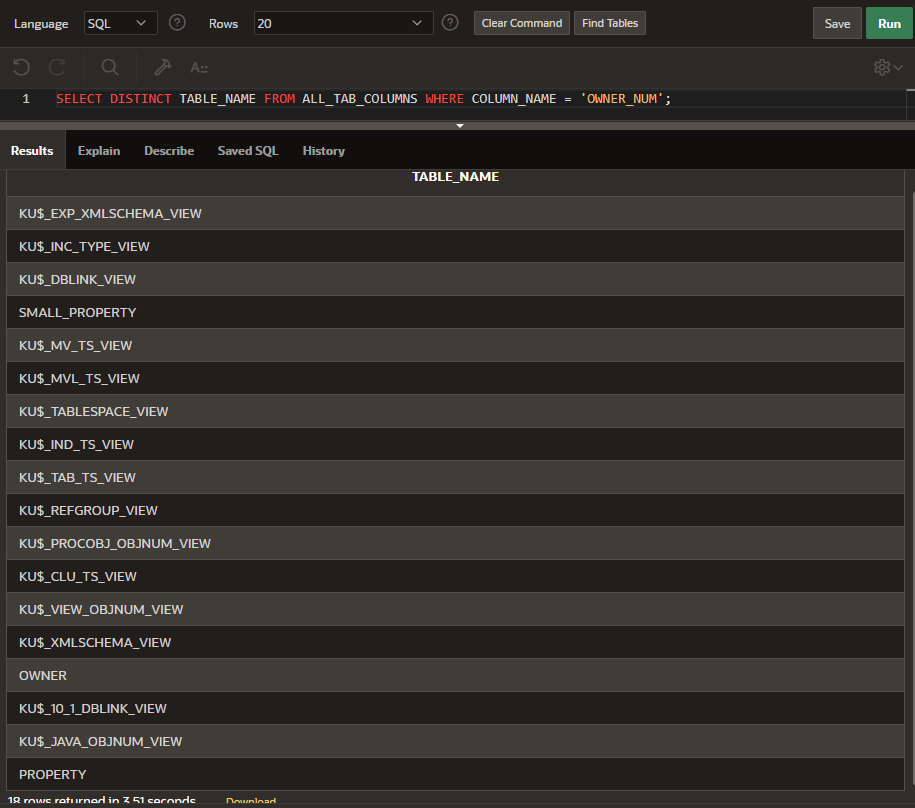


8. Write the commands to obtain the following information from the system catalog. Do not execute these commands unless your instructor specifically asks you to do so.

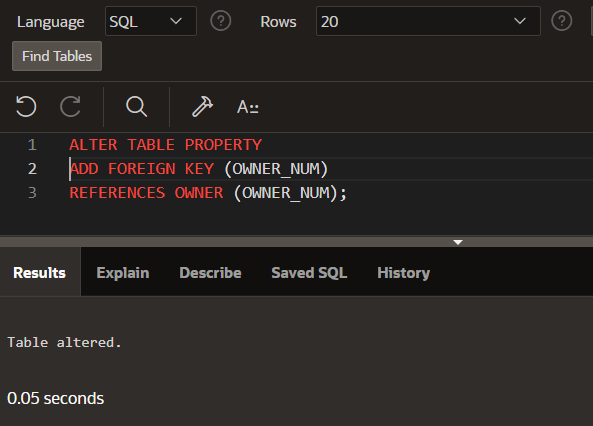
1. List every column in the PROPERTY table and its associated data type.



1. List every table that contains a column named OWNER\_NUM.



9. Add the OWNER\_NUM column as a foreign key in the PROPERTY table.



10. Ensure that the only legal values for the BDRMS column in the PROPERTY table are 1, 2, or 3.

- I had some issues with this problem, I cannot add the check needed in this question when values in the table are outside of the check. I included screenshots of my correct code that does what the problems asks, and some information about my error message.

