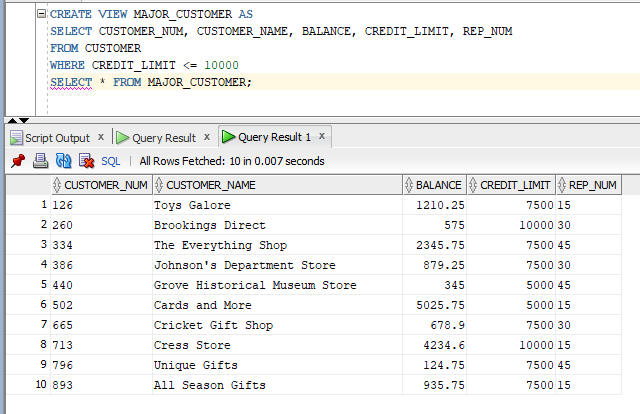
EOC # 7

.

1. Create a view name MAJOR\_CUSTOMER.  It consists of the customer number, name, balance, credit limit, and rep number for every customer whose credit limit is $10,000 or less.

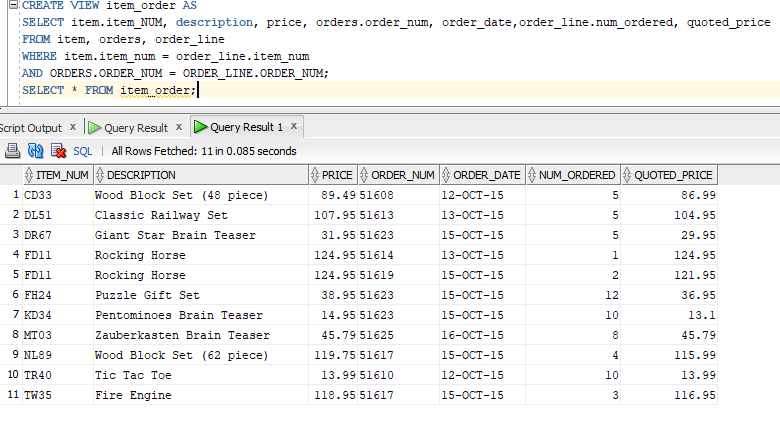
Write and execute the CREATE\_VIEW command to create the MAJOR\_CUSTOMER view.

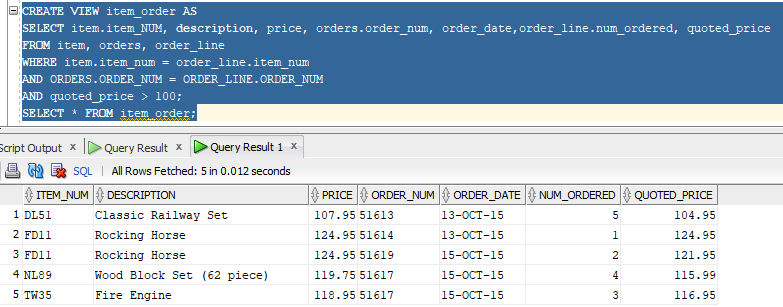
* + Write and execute the command to retrieve the customer number and name of each customer in the MAJOR\_CUSTOMER view with a balance that exceeds the credit limit.
  + Write and execute the query that the DBMS actually executes.

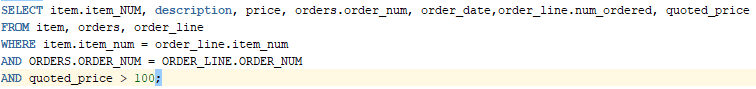
 

* + Does updating the database through this view create any problems?  If so, what are they?  If not, why not? There are no error messages displayed, I don’t think there’s any problems.

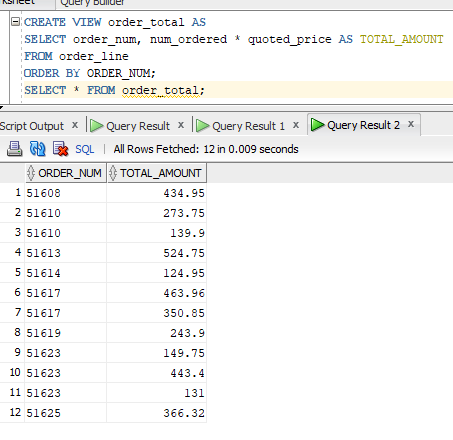
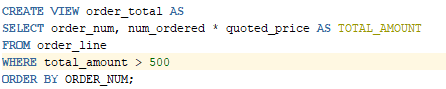
1. Create a view named ITEM\_ORDER.  It consists of the item number, description, price, order number, order date, number ordered, and quoted price for all order lines currently on file.
   * Write and execute the CREATE\_VIEW command to create the ITEM\_ORDER view.



* + Write and execute the command to retrieve the item number, description, order number, and quoted price for all orders in the ITEM\_ORDER view for parts with quoted prices that exceed $100. 
  + Write and execute the query that the DBMS actually executes.



* + Does updating the database through this view create any problems? If so, what are they? If not, why not? No errors displayed. I think the code is correct.

1. Create a view named ORDER\_TOTAL.  If consists of the order number and order total for each order currently on file.  (The order total is the sum of the number of units ordered multiplied by the quoted price on each order line for each order.)  Sort the row by order number.  Use TOTAL\_AMOUNT as the name for the order total.
   * Write and execute the CREATE\_VIEW command to create the ORDER\_TOTAL view. 
   * Write and execute the command to retrieve the order number and order total for only those orders totaling more than $500. 
   * Write and execute the query that the DBMS actually executes.



* + Does updating the database through this view create any problems? If so, what are they? If not, why not? An error message displayed when trying display total amount exceeding $500.

1. Write but do not execute, the commands to grant the following privileges:
   * User Ashton must be able to retrieve data from the ITEM table.

GRANT SELECT ON ITEM TO ASHTON;

* + Users Kelly and Morgan must be able to add new orders and order lines.
  + GRANT INSERT ON ORDERS, ORDER\_LINE TO KELLY, MORGAN;
  + User James must be able to change the price for all items.

GRANT UPDATE (PRICE) ON ITEM TO JAMES;

* + User Danielson must be able to delete customers.

GRANT DELETE ON CUSTOMER TO DANIELSON;

* + All users must be able to retrieve each customers number, name, street, city, state, and postal code. GRANT SELECT (CUSTOMER\_NUM, CUSTOMER\_NAME, STREET, CITY, STATE, POSTAL\_CODE) ON CUSTOMER TO PUBLIC;
  + User Perez must be able to create an index on the ORDERS table.

GRANT INDEX ON ORDERS TO PEREZ;

* + User Washington must be able to change the structure of the ITEM table.

GRANT ALTER ON ITEM TO WASHINGTON;

* + User Grinstead must have all privileges on the ORDERS table.
  + GRANT ALL ON ORDERS TO GRINSTEAD;

1. Write, but do not execute, the command to revoke all privileges from user Ashton in Exercise 4a.

REVOKE ALL ON ITEM TO ASHTON;

1. Perform the following tasks:
   * Create an index named ITEM\_INDEX1 on the ITEM\_NUM column in the ORDER\_LINE table.

* + Create an index named ITEM\_INDEX2 on the CATEGORY column in the ITEM table.

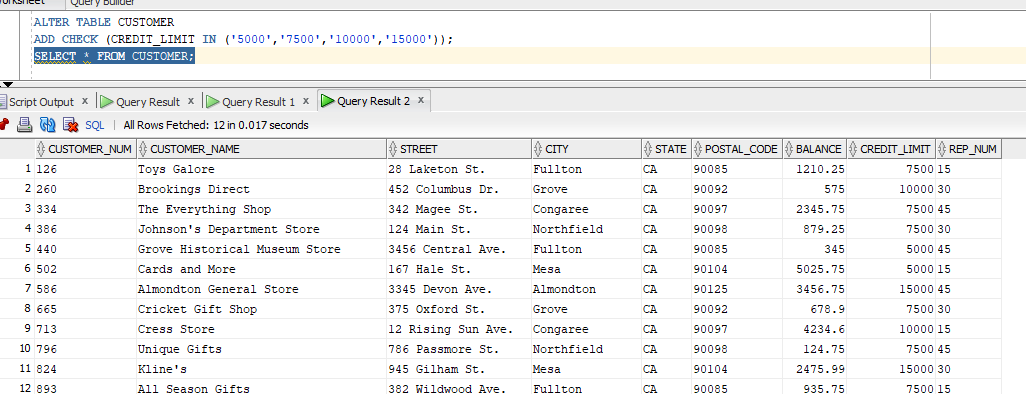
 

* + Create an index named ITEM\_INDEX3 on the CATEGORY and STOREHOUSE columns in the ITEM table.  
  + Create an index named ITEM\_INDEX4 on the CATEGORY and STOREHOUSE columns in the ITEM table.  List item categories in descending order.  

1. Delete the index name ITEM\_INDEX3.  
2. Write the commands to obtain the following information from the system catalog.  Do not execute these commands unless your instructor asks you to do so.
   * List every table that you own. SELECT TABLE\_NAME
   * FROM DBA\_TABLES;
   * WHERE TABLE\_NAME = I WAS UNSUCCESSFUL IN LOCATING OWNER NAME. NOR COULD I NARROW THE TABLES DOWN, THERE'S THOUSANDS!
   * List every column in the ITEM table and its associated data type.
   * SELECT COLUMN\_NAME, DATA\_TYPE
   * FROM DBA\_TAB\_COLUMNS
   * WHERE OWNER = 'N/A'
   * AND TABLE\_NAME= 'ITEM';
3. Add the ORDER\_NUM column as a foreign key in the ORDER\_LINE table.

ALTER TABLE ORDER\_LINE

ADD FOREIGN KEY (ORDER\_NUM) REFERENCES ORDERS;

1. Ensure that the only values entered into the CREDIT\_LIMIT column are 5000, 7500, 10000, AND 15000. 
2. Toys Galore currently has a credit limit of $7,500.  Because Toys Galore has an excellent credit rating, TAL Distributors is increasing the company's credit limit to $10,000.  If you run the SQL query in Exercise 1 after the credit limit has been increased, would Toys Galore still be included in the view?  Why or why not? Yes because the changes would reflect what was in the original query.