Assignment-12 ITI-2180 Foundations of Data and Data Management

Answer each of the following in a similar manner to all Hands-On Assignments in the textbook. All problems below use the database provided with the textbook (JustLee Book Company).

Run JLDB\_Drop.sql followed by JLDB\_Build.sql from the student data files provided with the textbook. In addition, be sure to create the tax table from the tax.sql script in the Blackboard Resources.

NOTE: The first four questions are dependent and should be done in order. The remainder are independent and the order of completion is not important.

Submit: A file called *Assignment-12-Results.docx* with formatted SQL & output to Blackboard.

1. With a single SQL statement, create a new table called *customers\_alt* that contains the *customer#, lastname*, *firstname*, *state*, and *zip* codes loaded from the customers table. The column names should be identical in the new table except that the zip code column should be named *zipcode.* Stated another way, provide a single SQL statement that results in a new *customers\_al*t table with five columns that are pre-loaded with data the *customers* table.
2. Add a primary key constraint named *customers\_alt\_customer#\_pk* to the *customer#* column of the CUSTOMERS\_ALT table created above. Add a foreign key constraint named *tax\_state\_fk* to the CUSTOMER\_ALT table that references the *state* field in the TAX table.
3. Create an INDEX named *customers\_alt\_idx* on the *zipcode* column of the *customers\_alt* table. Verify that the index exists by querying the data dictionary. Remove the INDEX from the *customer\_alt* table after the data dictionary is queried.
4. Create a sequence named *orders\_order#\_seq* that will be used for order#'s when new orders are placed. If there is already a sequence named *order\_order#\_seq*, provide the command or commands to change the sequence. The company has decided that order numbers should always be even numbers. After your sequence is ready to be used, show proof that your sequence is working correctly.
5. Prepare a database select that shows the title of all books in the computer category that are from publisher #3 and that sell for at least $30.
6. Provide the first name and last name (concatenated together) for orders that have been shipped to Atlanta, GA. The resulting column should have the title “Customers in Atlanta, GA”. Account for the fact that you are not sure of the case of the city or state (i.e, your SQL should work if the city is ATLANTA, atlanta, Atlanta, or any other combination that spells Atlanta. The same for GA, Ga, or ga).
7. Management is proposing increasing the price of each book by 5%. Create a list that displays each book’s title, existing retail price, and proposed retail price. The prices should be displayed with two decimal places and have a dollar sign, and the proposed price must be rounded to a whole dollar minus one penny (e.g. If a book’s current price is $30.95, a 5% increase would be $32.188, but would be rounded to $32.00 minus one penny or $31.99). Sort the result by title. The first five lines of the output looks like this:

TITLE Existing Price Proposed Price

------------------------------ -------------- --------------

BIG BEAR AND LITTLE DOVE $8.95 $8.99

BODYBUILD IN 10 MINUTES A DAY $30.95 $31.99

BUILDING A CAR WITH TOOTHPICKS $59.95 $62.99

COOKING WITH MUSHROOMS $19.95 $20.99

DATABASE IMPLEMENTATION $55.95 $58.99

…

1. Find the number of books ordered by each customer. Display the customer # in sorted order along with the total number of books that the customer has ordered. The column heading for the books ordered should be "Ordered Books". The first five lines of the output looks like this:

CUSTOMER# Ordered Books

---------------------- ----------------------

1001 5

1003 2

1004 2

1005 3

1007 8

…

1. Create a READ ONLY view named *book\_authors* that includes the ISBN and the author’s last name and first name. After the view is created, select from the view all ISBN’s, author’s first names and last names that have the last name ‘WHITE’. I’ll provide the SELECT statement and the result below, you provide the SQL statement to create the view that generates this results:

SELECT isbn, lname, fname FROM book\_authors WHERE lname=’WHITE’;

ISBN LNAME FNAME

---------- ---------- ----------

1915762492 WHITE LISA

2147428890 WHITE LISA

9247381001 WHITE WILLIAM

1915762492 WHITE WILLIAM

1. Create a READ ONLY view named *order\_total* that has five columns:

*order#*  The order #

*subtotal* The subtotal of all merchandise on the order

*shipping* The shipping assigned to the order. Account for NULL shipping

*tax* The tax based on the subtotal rounded to the nearest cent

*total* The sum of subtotal, shipping, and tax

After the view is created, use the view to show all order#'s with total > $100.