Relational Database Development

152-156

# Views, Indexes & Security

20 points

Using the MySQL Workbench, complete the following exercises. Save your queries as a .sql file. Indent **and number /\* 1 \*/** your queries to improve readability. Result sets for some queries are available on the course website.

**Premiere Products** (1 point each)

1. Create a view named vwMajorCustomers that includes the following fields: customer ID, customer name, balance, credit limit, remaining credit (limit – balance) and rep name (last and first names concatenated). Only include customers whose credit limit is $10,000 or less. Order the records by the remaining credit limit (largest to smallest).
2. Using vwMajorCustomers, display the customer name and rep name for customers whose balance exceeds their credit limit. Sort the results by rep name and then customer name.
3. Using vwMajorCustomers, display the rep name, total credit limit, total balance and total remaining credit for each of the reps. Assign appropriate aliases to each calculated field.
4. Create a view named vwOrderedParts that includes the following fields: part ID, description, price, order ID, order date, number ordered and quoted price. Sort the records by order date descending and then by part description.
5. Using vwOrderedParts, display the order lines that were tendered reduced prices on their parts. Include the part description, price and quoted price and the discount percent (1 – quoted / price). Sort the results discount percent, highest to lowest.
6. Create a view named vwOrderTotals that includes the order ID, order date and order total for all orders. The order total is sum of all quoted price times the number ordered for each of the order’s order lines. Sort the results by order date (ascending). Name the total field *totalAmount*.
7. Without modifying the view, include the customer’s name for each order. Tip: join to existing tables or views. Sort the results by customer name.
8. Create an index called idxByPart on the partId field in the orderlines table.
9. Create an index called idxByClassWarehouse using the class and warehouse ID fields in the parts table. Classes should appear in descending order.
10. Delete the idxByPart index.

(continued)

**Henry Books** (1 point each)

1. Create a view named vwPenguin that includes the following fields: book ID, title, type (name) and price for every book published by Penguin USA. Sort the list by type and then price.
2. Using vwPenguin, display the Penguin USA books with a price less than $10.
3. Create a view named vwHardCover that includes the following fields: bookd ID, title, publisher name and price. Include only hard cover books (not paperback). Sort the list by publisher name and then price descending.
4. Create a view named vwBranchInventory the includes the following fields: branch location and total number of books in inventory at that branch. Sort the results by location.
5. Provide the command to create an index of your choice on a table in the Henry Books database. **Why** do you feel this index would be valuable?

**Extra Credit** (+4 Points Max)

Provide the commands (one command each when possible) to provide the following database security. You will not be able to test these queries using the PortableApps version of XAMPP and MySQL Workbench.

1. User Ashton must be able to retrieve data from the Parts table.
2. Users Kelly and Morgan must be able to add new orders and order lines.
3. User James must be able to change the price for all parts.
4. User Danielson must be able to delete customers;
5. All users must be able to retrieve each customer’s ID, name, street, city, state, and zip code.
6. User Perez must be able to create an index on the Orders table.
7. User Washington must be able to change the structure of the Parts table.
8. User Grinstead must have all privileges on the Orders table.