**Displaying Relational Data**

After implementing referential integrity, data across tables can be referenced to produce complex results. The next assignment you are given is to produce relevant data for the VaporGames data analytics team to review.

For this assignment, write queries that will answer the questions below. Save every query to a SQL script named DisplayRelationalData. Use -- to write a comment above each query, noting the question it refers to. Additionally, take a screenshot of the query and result set for each question, and insert it below the corresponding question.

**NOTE:** *no data found* may be an answer! This means the query ran without error, but there were no records found (no data present in the table matching the selected conditions).

The Data Analytics team over at VaporGames is interested in two general types of data and has composed a list of information they would like you to collect. The first six items pertain to user data, while the last six items pertain to product data.

1. Display every USERNAME and the lowest RATING they have left in a review.
2. Display every user’s EMAIL, QUESTION, and ANSWER.
3. Display the FIRSTNAME, EMAIL, and WALLETFUNDS of every user that does not have a WISHLIST.
4. Display every USERNAME and number of products they have ordered.
5. Display the age of any user who has ordered a product within the last 6 months.
6. Display the USERNAME and BIRTHDAY of the user who has the highest friend count.
7. Display the PRODUCTNAME, RELEASEDATE, PRICE, and DESCRIPTION for any product found in the WISHLIST table.
8. Display the PRODUCTNAME, highest RATING, and number of reviews for each product in the REVIEWS table. Order the results in descending order of the RATING.
9. Create a view that displays the PRODUCTNAME, GENRE, and RATING for every product with a 5 or a 1 RATING. Order the results in ascending order of the RATING.
10. Display the count of products ordered, grouped by GENRE. Order the results in alphabetical order of GENRE.
11. Create a view that displays each PUBLISHER, the average PRICE, and the sum of HOURSPLAYED for their products.
12. Display the sum of money spent on products and their corresponding PUBLISHER, from the ORDERS table. Order the results in descending order of the sum of money spent.

The Customer Support team at VaporGames is also interested in collecting new data. They have composed their own list of requested information and it is up to you to provide it. The first 6 items pertain to user support data and the last 6 items pertain to community rule infractions.

1. Display the TICKETID, USERNAME, EMAIL, and ISSUE only for tickets with a STATUS of ‘NEW’ or ‘IN PROGRESS’, sorted by the latest DATEUPDATED.
2. Display the USERNAME and count of TICKETID that users have submitted for user support.
3. Display the USERID and EMAIL of any user who has submitted a support ticket that used their FIRSTNAME, LASTNAME, or combination of the two in their EMAIL address.
4. Display the EMAIL address of any user who has a ‘NEW’ or ‘IN PROGRESS’ support ticket STATUS, where the EMAIL is not currently saved in the USERBASE table.
5. Display the TICKETID, FIRSTNAME, LASTNAME, and USERNAME of any user whose USERNAME is mentioned in the ISSUE of a support ticket.
6. Display the USERNAME and PASSWORD associated with the EMAIL address provided in the support tickets.
7. Create a view that displays the USERNAME, DATEASSIGNED, and PENALTY for any user whose PENALTY is not null and the infraction was assigned within the last month.
8. Display the USERNAME and EMAIL of any user who is at least 18 years old and has not received an infraction within the last 4 months.
9. Display the USERNAME, DATEASSIGNED, and full guideline name (RULENUM and TITLE with a blank space inbetween) for any user who has violated the community rules.
10. Display the USERID, USERNAME, EMAIL, and sum of all SEVERITYPOINTS each user has received.
11. Display the TITLE, DESCRIPTION, and PENALTY for all infractions assigned.
12. Display the USERNAME and count of infractions for users who have violated the community rules at least 15 times.