We created 3 SQL VIEW code blocks for display, both on app and Company.

In app part, in order to have a more intuitive display for each game category, we calculated average score and rank.

By using the INNER JOIN, we combined the app average score with the category entity and its contained APP name.

The upper one is average score selection code, the middle one is category and app combined code, the last one is how we join them together and group by categoryID to calculate each score with SUM divided by COUNT and DENSE\_RANK function.

In Company part, we write SQL statement to select every app developed by each company with their ID, name version and even their install quantity. And we joined them all together.

In the last part, we made some visualization for the view, we plot average score and rank separately in bar chart, in this way, when we have more category for apps, we can easily find the most popular one.

The other colorful bar chart is for company, with both company name and APP name as x, install quantity as y, we plot the trends and value for each APP.

Here is our Table Level Constraints, I just put one example here, We want to make sure is there has OrderID in OrderHeader table when we want to add some data into OrderDetail table. If this OrderID is not in the orderHeader table it will show conflicted with the check constraint error

Here is our Data encryption, first one is number encryption, we use this to encrypt credit card number and decryption

second one is String encryption, we use this to encrypt and decryption password of user account which contains both character and Number

Here is our procedure of calculate column, here are two example;

The first One is using loop to Calculate TotalAmount of OrderDetail table which Equals The quantity of OrderDetail table take the price of App table where they have same AppID

The second one is using loop to Calculate user’s InAppPurchase of UserAccount table which equals the sum of each user’s TotalAmount of OrderHeader table where has same UserAccountID