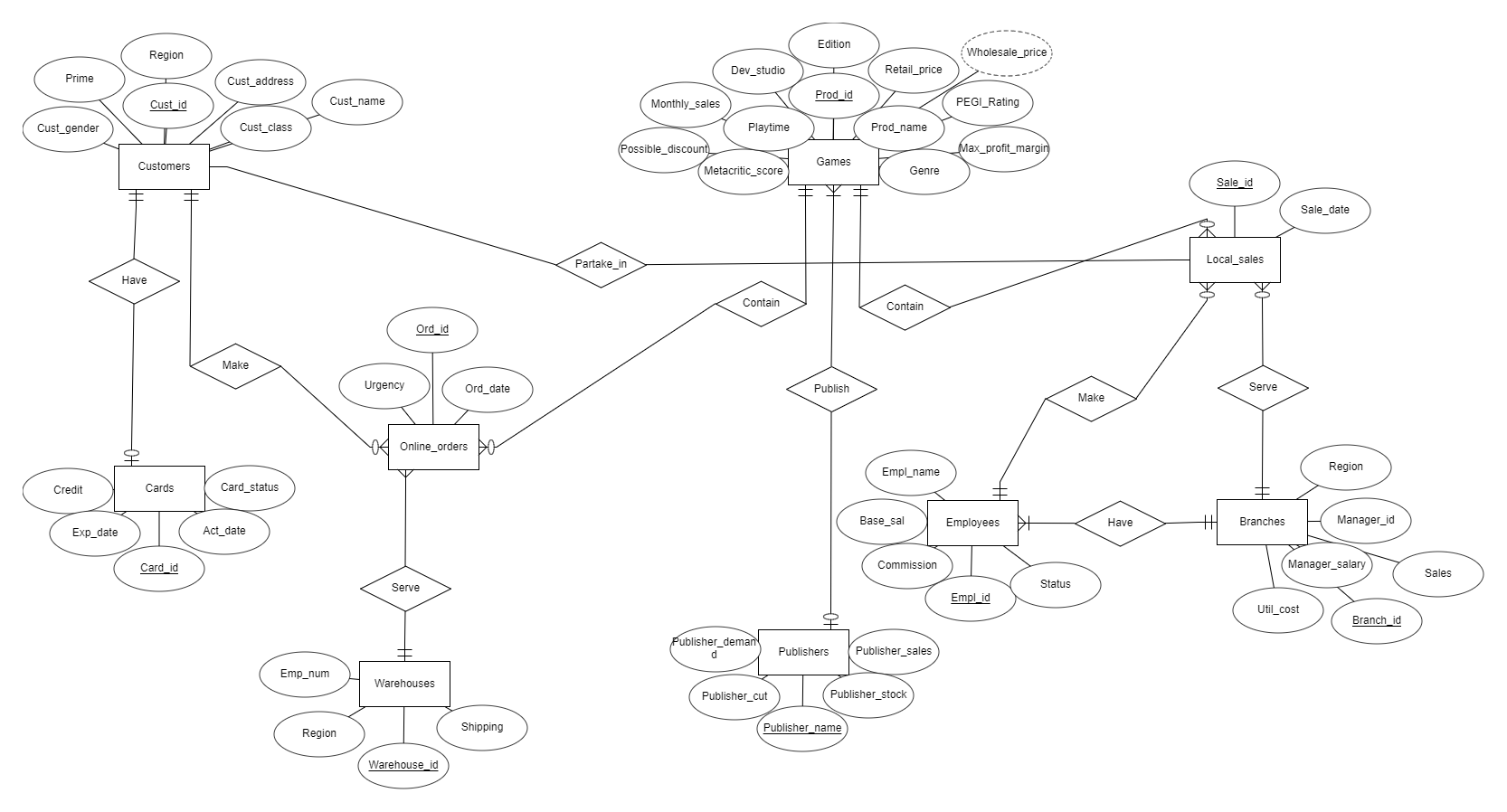
# CS205 Business Data Management

# Fall 2020 – Final Project Report

## Presentation

This application is meant to partially monitor the data of a video game store franchise. It holds both logistic data useful for making financial decisions on a higher authority level, and information that can be used by retail workers for increased efficiency when helping customers. It is important to note that several data fields are not used for the queries of this application. These data can be useful in simple displays or queries of a subpart complexity for the nature of this project. Finally, this project was created in order to display a variety of usable techniques, and thus some of the queries are unrealistic or even slightly comedic in nature.

## Entity Relationship Diagram



## Field Description

**Customers:**

* **Cust\_id:** A unique id for every registered customer. This application will only monitor purchases made from registered customers.
* **Cust\_gender:** Used to deduce the title of a customer when addressing them.
* **Cust\_address:** Used for the competition query. Can be used for statistical analyses when it comes to regional productivity and allocation of resources.
* **Region:** An extension of the address, kept in a separate field so it can be easily accessed for large scale statistics.
* **Cust\_class:** The class of the customer, customers could change class based on activity and commitment to the brand. Will be later used to provide discounts for local purchases.
* **Cust\_card\_code:** Some of the customers have opted to sign in for a card. The card collects credits based on a number of things like purchases or store events.
* **Prime:** Prime is a paid service that a customer may choose to sign on, that will remove the urgency fees on online orders. Other bonuses like extra credits can be awarded to prime members.

**Cards:**

* **Card\_id:** A unique card id, associated with Cust\_card\_code.
* **Credit:** The number of credits in any given card.
* **Card\_status:** Some clients might not opt to renew their cards and they will be rendered inactive. The credits will count for the rest of the year, but the customer will not be eligible for the grand prize unless he reactivates the card before a certain deadline.
* **Activation/Expiration dates:** Useful to let the customer know about the status of their card.

**Games:**

* **Prod\_id**: Important for tracking and storage. Could also be used to gather statistical data about the performance of a specific product.
* **Prod\_name**: The title of the product. Can be used for informative purposes.
* **Genre**: The Genre of the game. Can be used for informative purposes as well as for statistical evidence as to what genre is more popular. Could guide future pricing and distributor deals. (Useless in our case since the pseudo-random generator distributes all data equally, thus eliminating any possible statistical value.)
* **Publisher/Dev\_studio**: Can be important to strike deals with distributors depending on how well the games of a specific publisher/studio are selling. Could also be used for marketing purposes and to inform the clients.
* **PEGI-rating/Metacritic-score/Playtime**: Trivial information about the game. Can be used for marketing, to inform the client or deduce pricing and discounts for a game.
* **Edition**: Creates different pricing margins for the same games and is also used for dictating discounts.
* **Retail\_price/Wholesale\_price/Max\_Profit\_margin**: Extremely important data for book-keeping.
* **Possible\_discount/Monthly\_sales**; Based on how well a game is selling the maximum possible discount is decided. If a game does extremely well the possible discount drops and if it does poorly the discounts rise to boost sales.

**Publishers:**

* **Publisher\_name:** The name of the publisher, associated with publisher in the games table.
* **Publisher\_cut:** A cut that the publisher takes from every purchase when the store buys its stock. Based on publisher\_demand. Not used in this application since all the stocking is calculated in the util\_cost of the branch.
* **Publisher\_sales:** Statistics for the publisher world-wide used to calculate publisher demand.
* **Publisher\_stock:** The number of items the publisher is willing to sell for the year. Could be used to calculate cases of shortage and change the prices accordingly.
* **Publisher\_demand:** The general demand for a publisher.

**Employees:**

* **Empl\_id:** The unique id for every employee in the franchise.
* **Empl\_name:** The name of the employee.
* **Branch\_id:** The branch in which the employee belongs.
* **Status:** Weather an employee is active or on leave. It is a paid leave so it still counts as a branch expense.
* **Base\_sal:** The standard monthly salary for every employee.
* **Commission:** The percentage of a sale allocated towards the employee.

**Online\_orders:**

* **Ord\_id:** The unique id of an order.
* **Ord\_date:** The date the order was placed.
* **Cust\_id:** The id of the customer that placed the order.
* **Prod\_id:** The id of the product ordered.
* **Urgency:** The selected urgency selected by the user. Is used to calculate extra fees.
* **Warehouse\_id:** The unique id of the warehouse that will service the order.

**Local\_sales:**

* **Sale\_id:** The unique id of any local sale.
* **Cust\_id:** The id of the customer that made the purchase.
* **Sale\_date:** The date the sale took place.
* **Prod\_id:** The id of the product bought.
* **Branch\_id:** The branch the sale took place in.
* **Empl\_id:** The employee that closed the sale.

**Branches:**

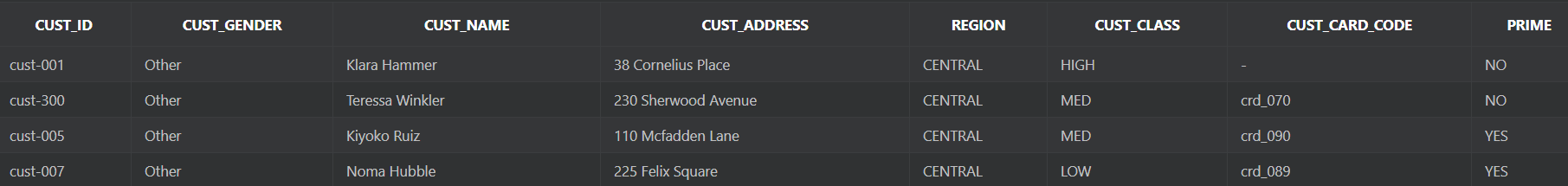
* **Branch\_id:** The unique id of a branch.
* **Region:** The regional coverage of a branch.
* **Manager\_id:** The id of the manager of the branch.
* **Manager\_salary:** The salary of the Manager of the branch.
* **Sales:** The number of sales of non-game products in a store. These are accounted for as an average revenue value.
* **Util\_cost:** The monthly utility costs of the branch.

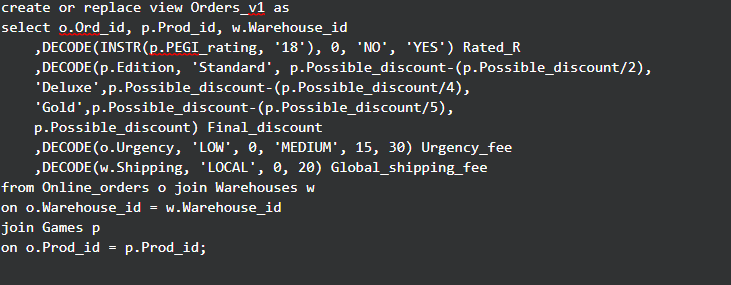
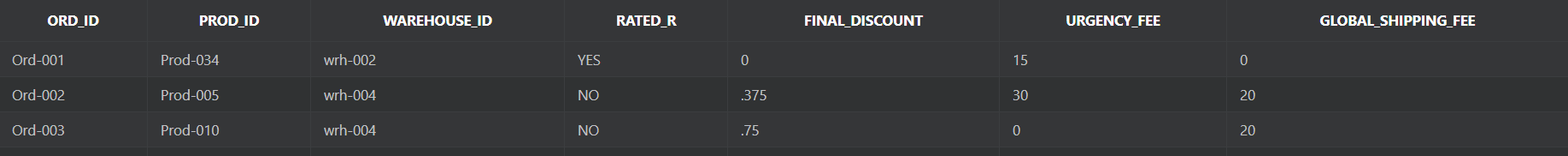
**Warehouses:**

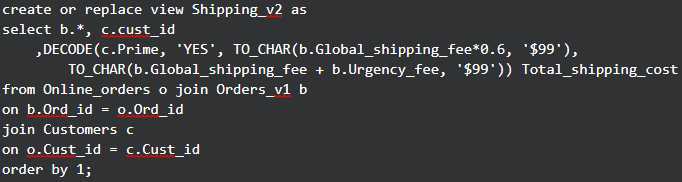
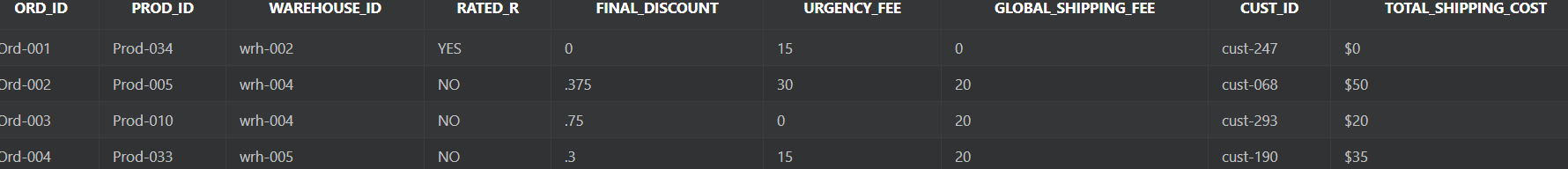
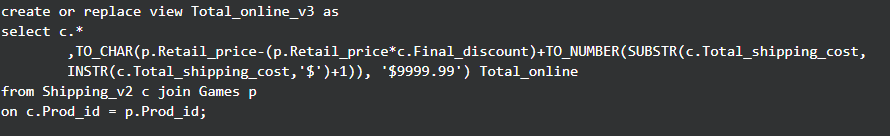
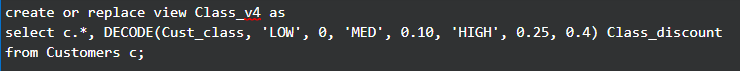
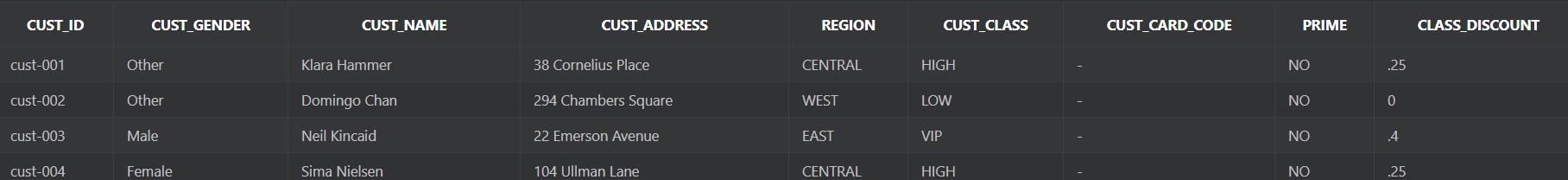
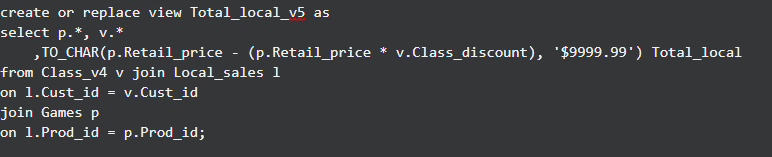
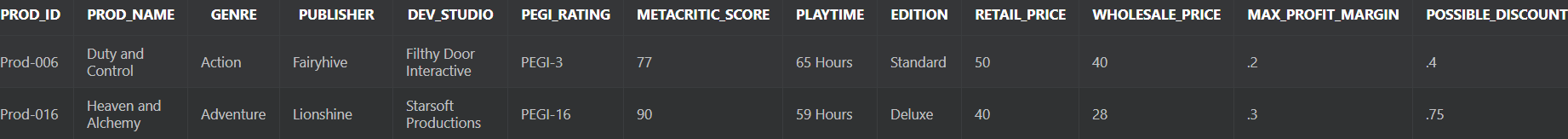
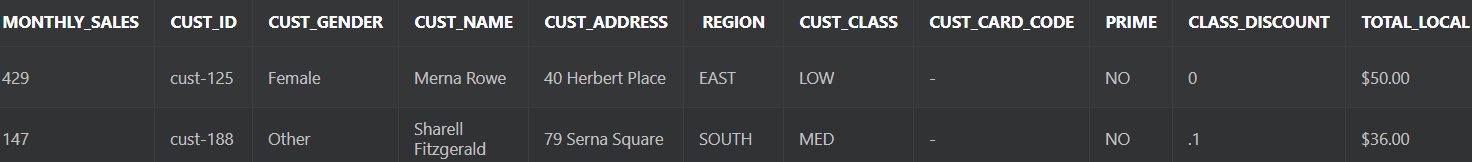
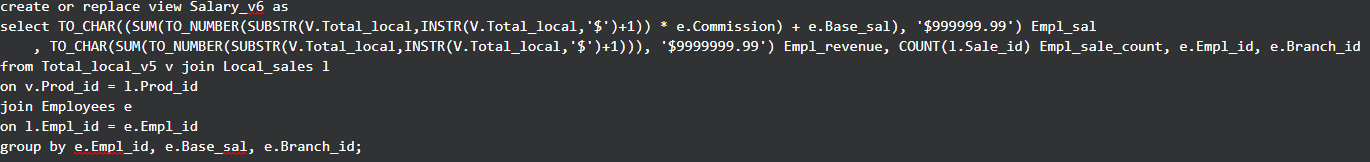
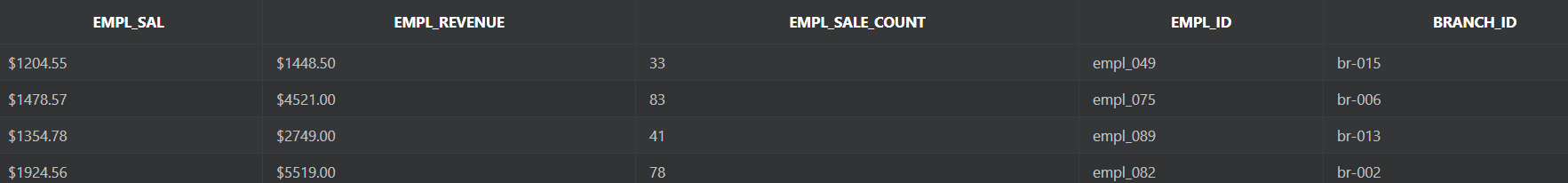
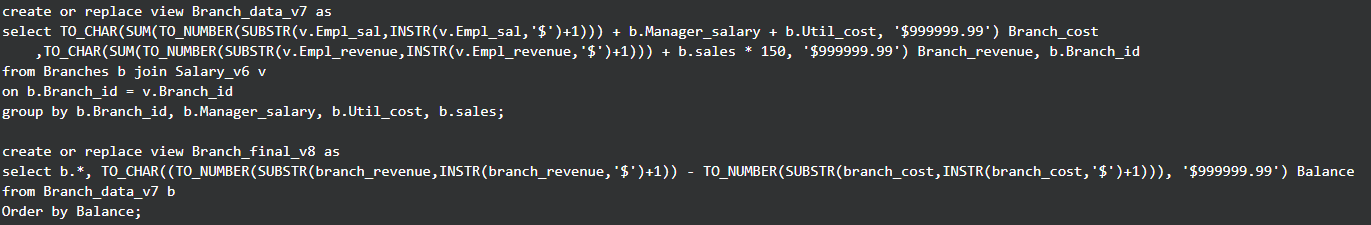
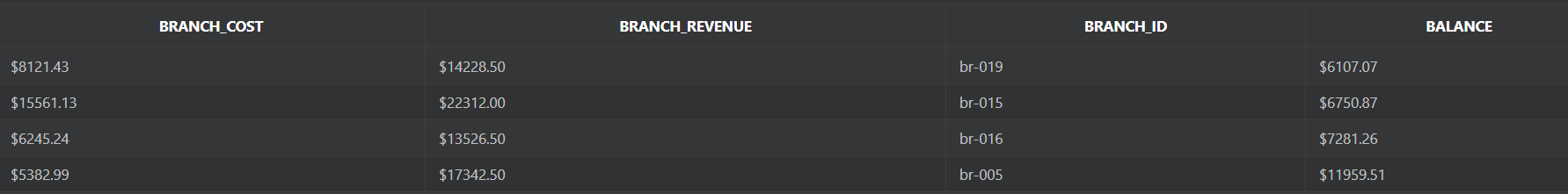
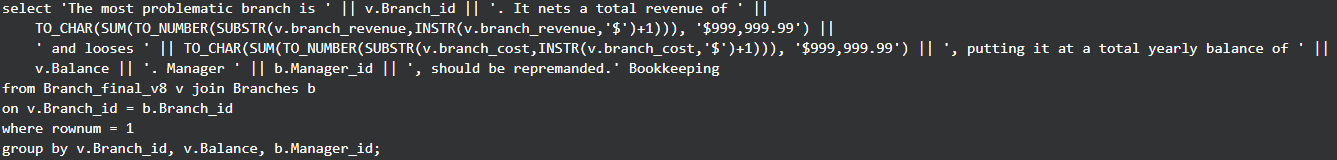
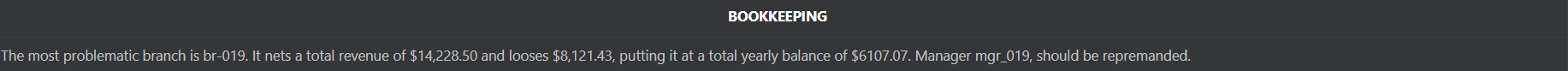
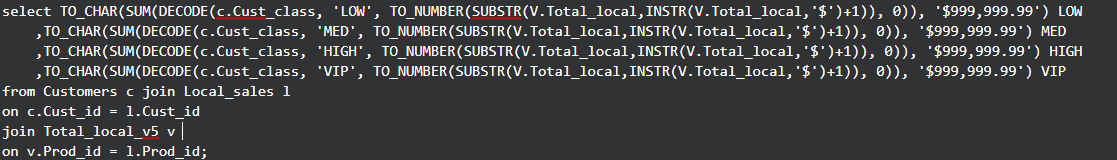
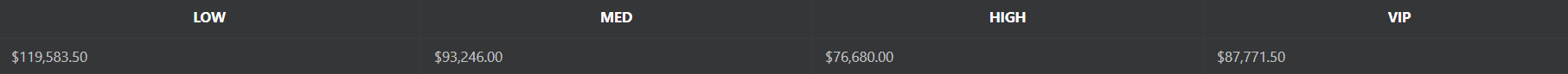
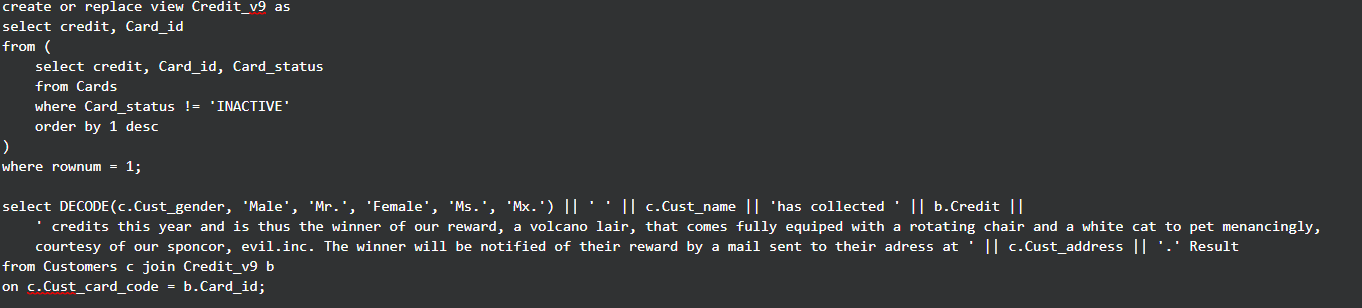
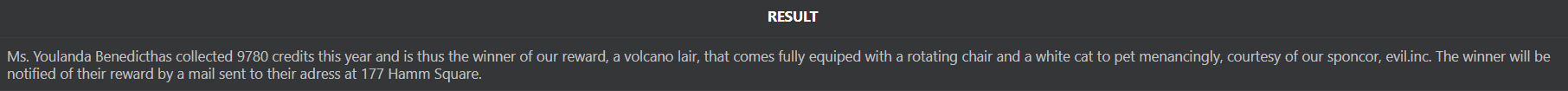
* **Warehouse\_id:** The unique id of the warehouse.
* **Region:** The region that the warehouse is located in.
* **Emp\_num:** The number of employees occupied in the warehouse.
* **Shipping:** Shows if the warehouse ships locally or globally.

## Queries

**Note: The information mentioned in the code in the form of comments will not be repeated here for the sake of concision. To display results for after the creation of the view a select \* statement is used on said view.**

1. A simple display of all the data in Customers table, ordered by region.  
     
   
2. Creates a view that performs the following tasks:
   1. Returns Yes if the game is rated R based on the pegi system or No otherwise.
   2. Calculates the final discount by taking into consideration the max possible discount and the edition ordered by the client. The more expensive the edition, the higher the discount.
   3. Created a global shipping fee in case the product is not shipped locally.
   4. Adds an urgency fee based on the urgency of the order.
   5. It also includes Ord\_id, Prod\_id, and Warehouse\_id to make future joins possible.

1. View that calculates the total shipping cost.  
     
     
   
2. View that calculates the total cost of the order.  
     
     
   
3. View for calculating class discounts.  
     
     
   
4. View for calculating the total cost of local sales.  
     
     
     
   
5. View that calculates the employee salary and the number of sales per employee.  
     
     
   
6. Views for calculating the branch revenue, losses and overall balance.  
     
     
   
7. Bookkeeping info on the worst performing branch.  
     
     
   
8. A tabulation showing the clean revenue from local sales based on customer class.  
     
     
   
9. A contest based on the card credit of the customers.  
     
     
   

## Suggestions for further development.

The needs of a franchise set of businesses are vast. This application barely scratches the tip of the iceberg when it comes to covering all of them. Here are some suggestions for improving upon this application in the future.

* Many more sales need to be recorded. An average of 30.000 sales per year for 100 employees should be expected by a healthy business chain.
* Different tables are needed for recording weekly, monthly, and yearly data.
* More tables needed to expand upon the publisher relations and operations of warehouses.
* More queries that encompass both online and local environments could be created.
* More elegant handling of the complex relationships dictating the pricing of the items, the salaries, and bookkeeping data needed.
* The application should consider all customers, registered or not.
* Need to expand upon non-game products, opening an entire subtree of interconnected tables for things like providers etc.
* Separation between applications aimed at employees, managers, and global supervisors should be created, providing them with only the data needed to fit their needs.