CS312: Database Management Systems - Final Project

By Nikola Jocic

Instructor: Christos Christodoulou

Date: 30/03/2022

American College of Thessaloniki

Table of Contents

[**Introduction**](#_47cvbh87st2s) **2**

[**Table Description**](#_sxird56lo6s8) **3**

[ERD](#_jn6eu2abac54) 3

[Data](#_yqij9qckjsme) 4

[**Enhancements**](#_xgwzt6u28z6) **6**

[**Queries**](#_i7rgqf1ov6zr) **8**

[TRIGGER](#_6140z2ia6gi4) 8

[THE VIEW](#_ixav56j402gz) 8

[THE FUNCTIONS](#_g6p00erv4utg) 9

[getCurrentPrice()](#_rrjj1p9lizsy) 9

[orders\_cost()](#_5kknsxl655s9) 9

[CURSOR](#_7krtoro7jyrk) 10

[HIERARCHICAL QUERY](#_orwgskbfx6zt) 11

[**Further Development**](#_83x4hgxdnbm) **12**

# Introduction

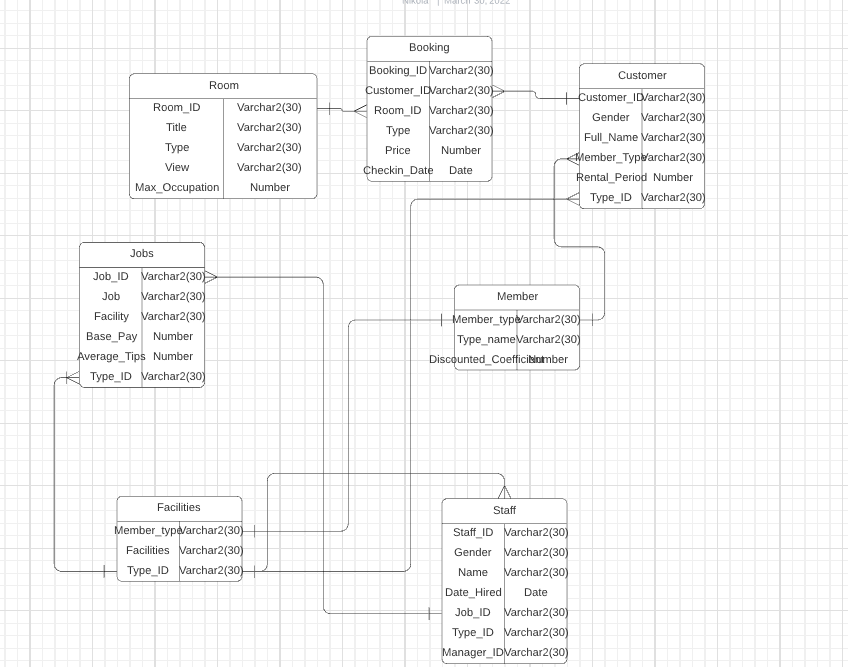
This project aims to incorporate the material taught throughout the course and use it to manipulate a real-life-like database system. The database system in question is a Vacation Resort which provides rooms and other services to its customers based on their arrangements.

The database is made to be useful to both workers for the resort and customers (through workers usage). It is accommodated so that the employees of all likelihood can benefit from the data including but not limited to (staff members, managers, receptionists, bookkeepers, etc..).

The table is created to represent real world scenarios.

# Table Description

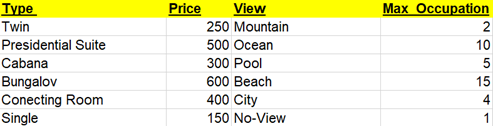
## ERD



## Data

The data presented in the table is randomized and hand picked out so that it represents several different and realistic scenarios when it comes to bookings of the hotel rooms.

Customers have a choice between different types of rooms which come at a different price per night as well as different views. Additionally, each room has a different inhabitant capacity. Moreover, the table displays the number of the days that the customer spends in the selected, respective bedroom. The pricing for each room per night, as well as additional information mentioned below is as found below:



The customer books a room with its room ID, and its price which could be deducted depending on the membership that the ‘customer’ has with the hotel. The customer, although can book rooms multiple times at different times only, retains only one customer ID. However, each time the customer books a room (regardless of whether the customer ID is the same), the booking ID is different. The customer is only able to book one room at the time and the room is only bookable on one customer’s ID.

Each membership comes with different ‘benefits’.

N - No membership, no discount and the access to Bar

B - Bronze membership, 5% discount and exclusive access to the Restaurant

S - Silver membership, 10% discount and exclusive access to the Pool Area

G - Gold membership, 20% discount and exclusive access to the Spa

P - Platinum membership, 30% discount and exclusive access to the Casino

For the simplicity of the data structure, the customers which have some benefits are presented in such a way where, hierarchically, the highest membership customer does not retain the right to the lower membership benefits. This fact is considered to be implied and is only presented in such a way to excel for simplicity. The memberships only display exclusive and both ‘extra’ access, meaning the members still retain rights to the lower membership benefits.

Depending on what kind of the benefits customers have a right to, there are different staff job positions at different facilities.

None - Bar (Facility), Bartender (Job Position)

Bronze - Restaurant (Facility), Host, Waiter, Busser (Job Position(s))

Silver - Pool Area (Facility), Lifeguard, Swim Instructor (Job Position(s))

Gold - Spa (Facility), Masseuse, Pedicure Tech (Job Position(s))

Platinum - Casino (Facility), Dealer, Security (Job Position(s))

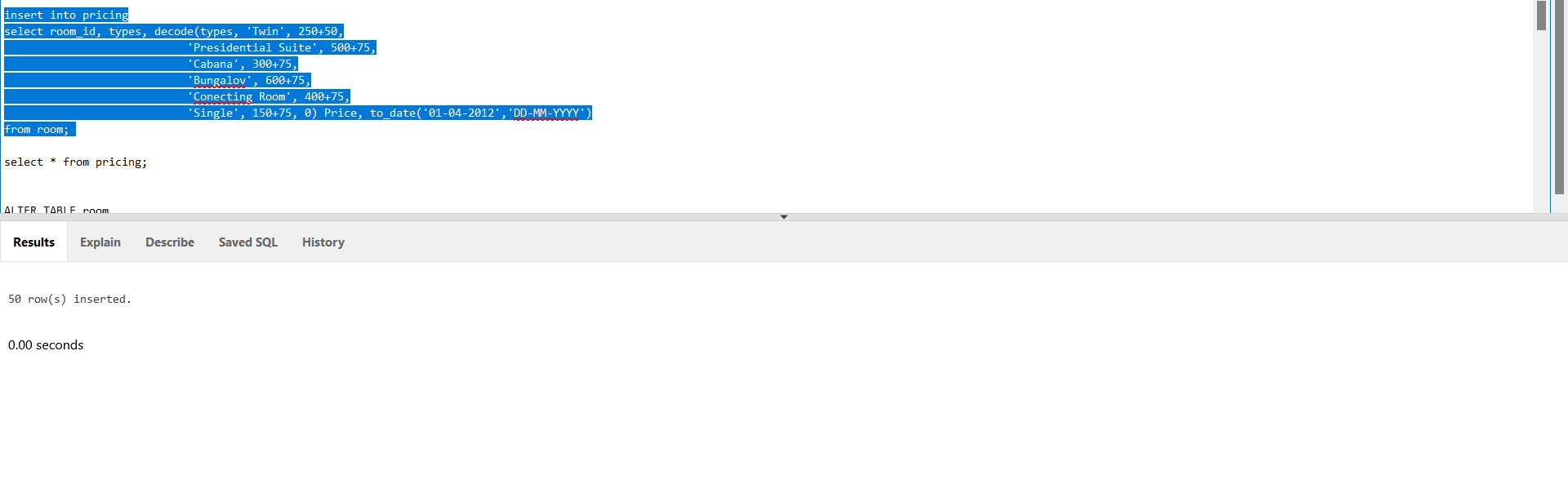
The job table takes care of displaying the vital information of the job position such as the facility where the job is located as well as the salary and possible tips. Tips have been coordinated so that only realistic job positions receive them. Furthermore, tips are calculated on a monthly basis.

Finally, the Staff represents the information structure of every employee of the hotel which displays their job position and facility at which they work at, along with their names and the date that they were hired. There is an additional manager column inserted containing randomized staff ID’s for latter use of hierarchical queries. One field is excluded from the staff ID since this person is the top manager.

The hotel is officially in operation for 10 years (between 2010 and 2020) and counting.

# Enhancements

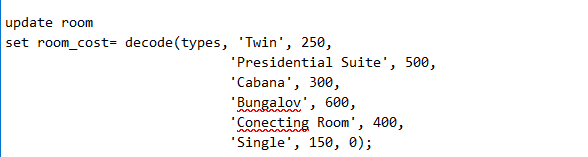
In order to better put the variety of queries to use, additional changes to the table were made in Oracle SQL. For starters, a new table has been added containing the list of all room types, their ID’s, their price, and the date at which those prices have changed. The following two figures better depict the changes made:



The ‘INSERT INTO’ statement was used twice, once in order to incorporate the price of the rooms in 2015, and the second time in 2012. In 2012 the rooms were more expensive. In 2015 those prices have dropped and that is why the values of the prices are added by 50/75 for the year 2012.

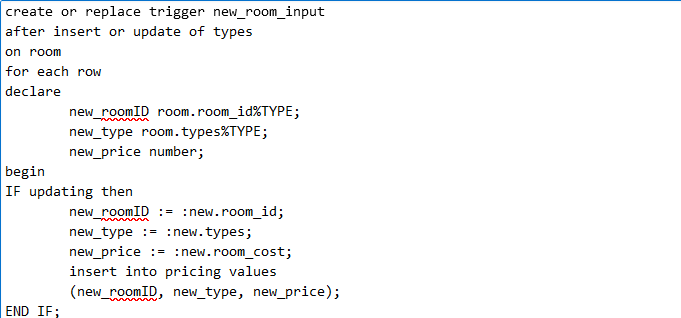
Moving onto the pre-existing tables, the ‘Room’ table has been altered. The final column room\_cost has been added for latter use in functions. As the title of the column suggests, the contents of it are the prices of the bedrooms. Following are pictures of those altercations:





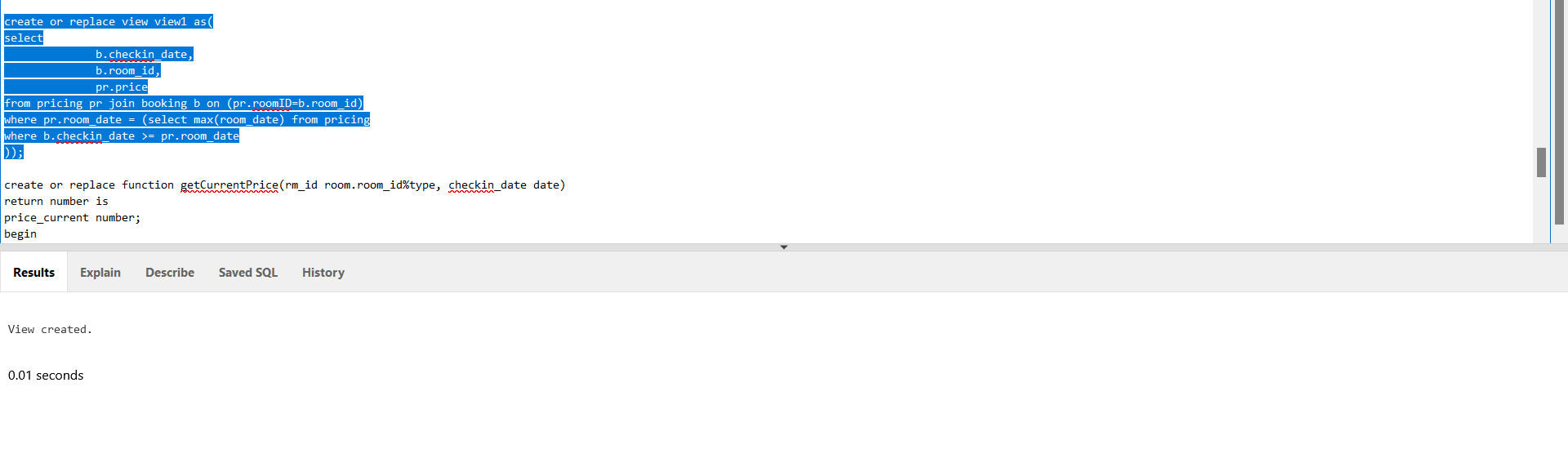
# Queries

### TRIGGER



This trigger is created for the purposes of checking when a new room has been installed inside a hotel. More specifically, a new room and a new room type. Upon the instance of updating or inserting into the room table, the trigger is activated and it serves to insert everything inputted into the room table into the newly created table ‘pricing’.

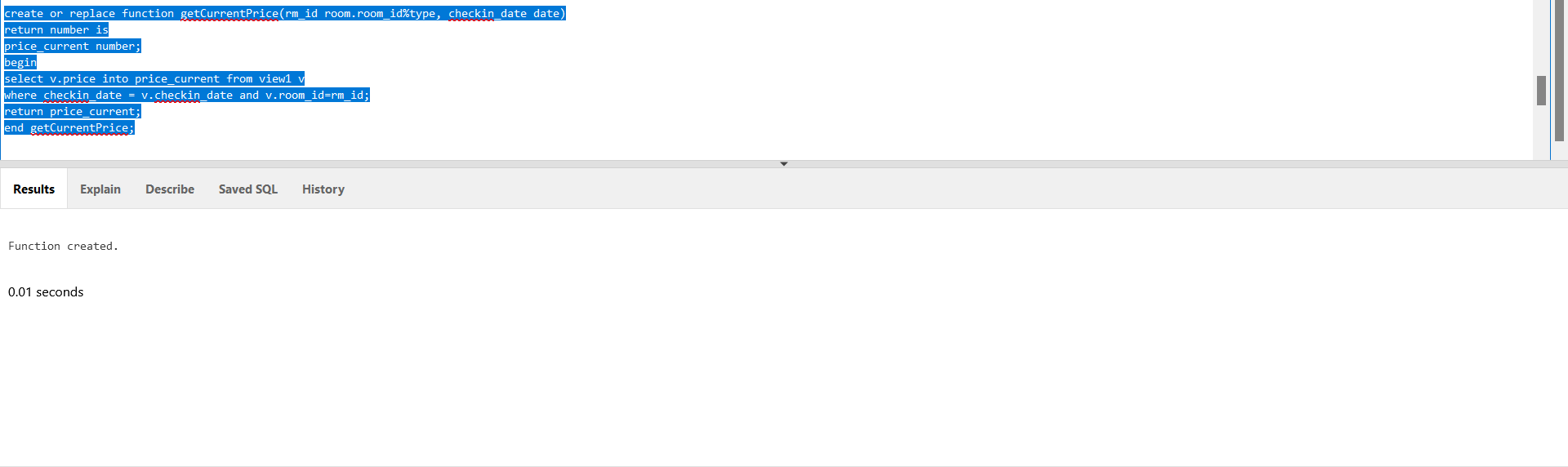
### THE VIEW



The view was created for the purpose of calculating the current price of the room at the time that the booking was made. Since the prices of rooms have been updated throughout time, the booking at a certain time for a particular room might contain a different price than originally instructed in the room table, and therefore the view is there to control just that.

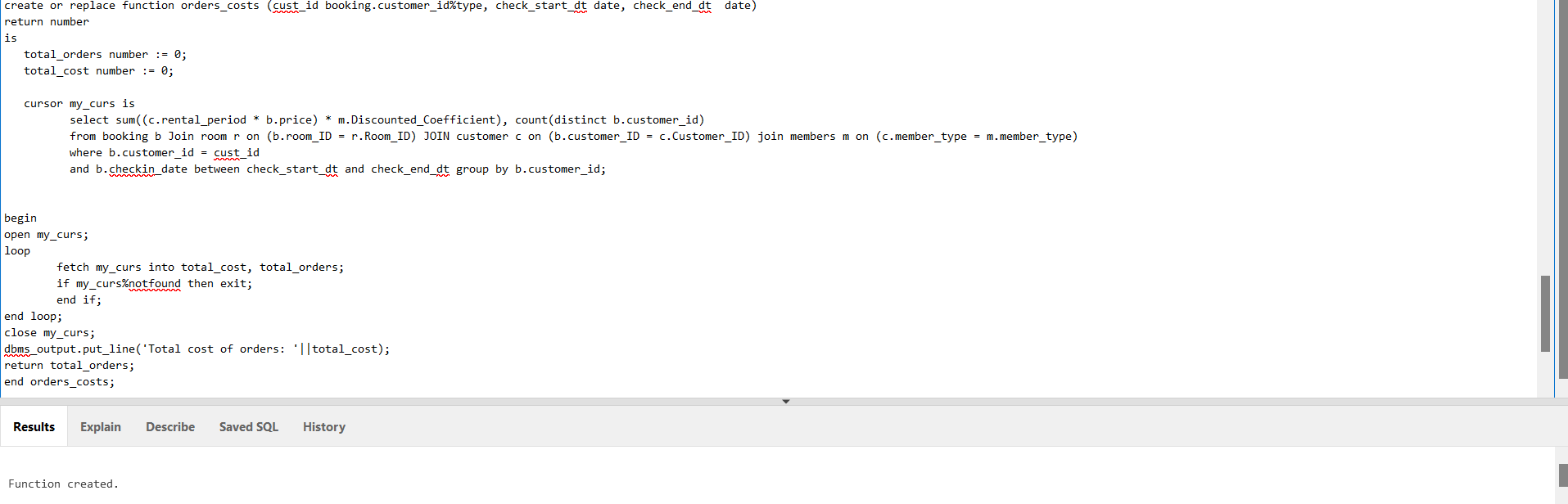
### THE FUNCTIONS

###### getCurrentPrice()



The aim of this function is to take data from the aforementioned view as well as two parameters (room ID and check in date) and return the current price of the room. This function is a continuation on the previous query (The view) so it works side by side with it to return the current price of the room at the time the booking was made.

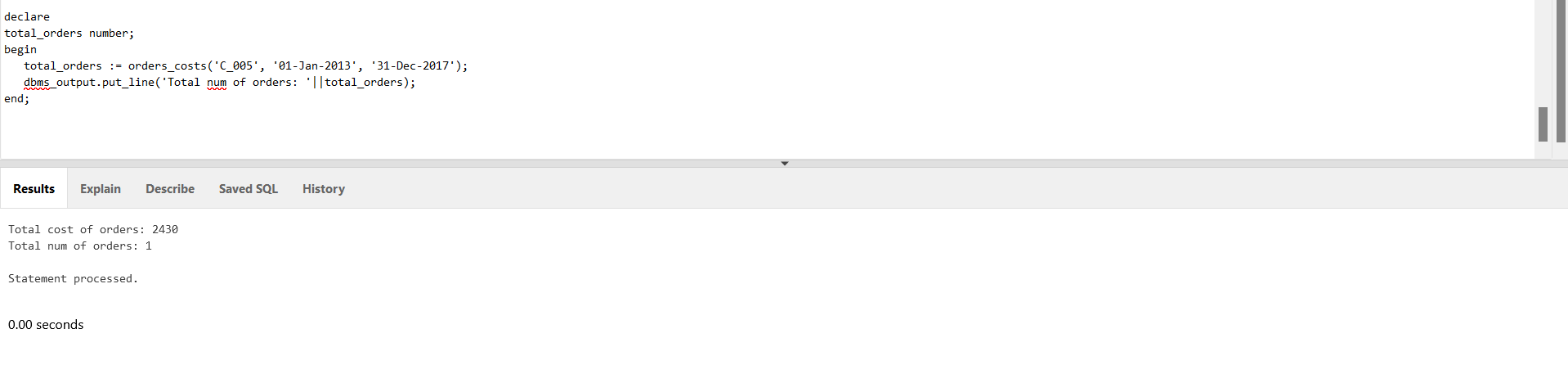
###### orders\_cost()



The purpose of this function is to take into consideration a singular customer at a specified point of time, and create a summary of their total booking expenses made in that interval. The function takes in parameters: booking\_id, start and end date. It returns the total number of orders made while printing the total amount of transactions they’ve made for their bookings. The function uses a cursor in order to calculate and sum all the expenses of the customer with respect to their royalty and benefits and it additionally keeps track of their ID.

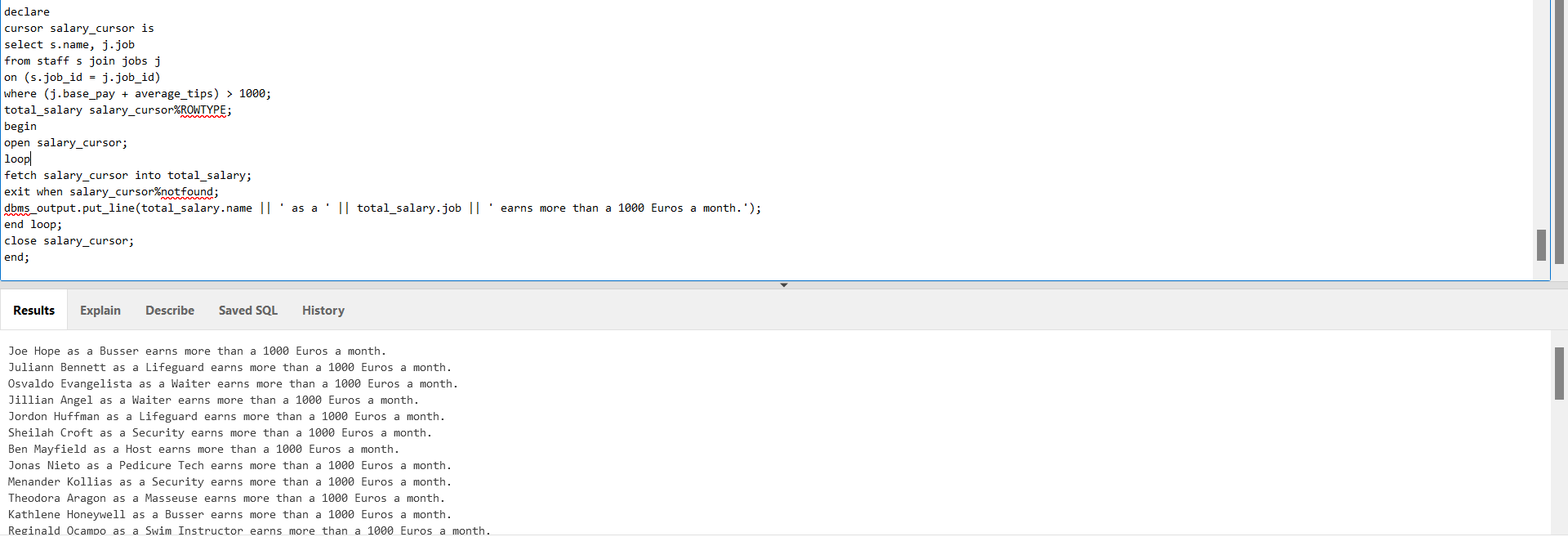
The loop fetches the data into the appropriate variables that are later returned/printed.

Orders\_cost Anonymous block



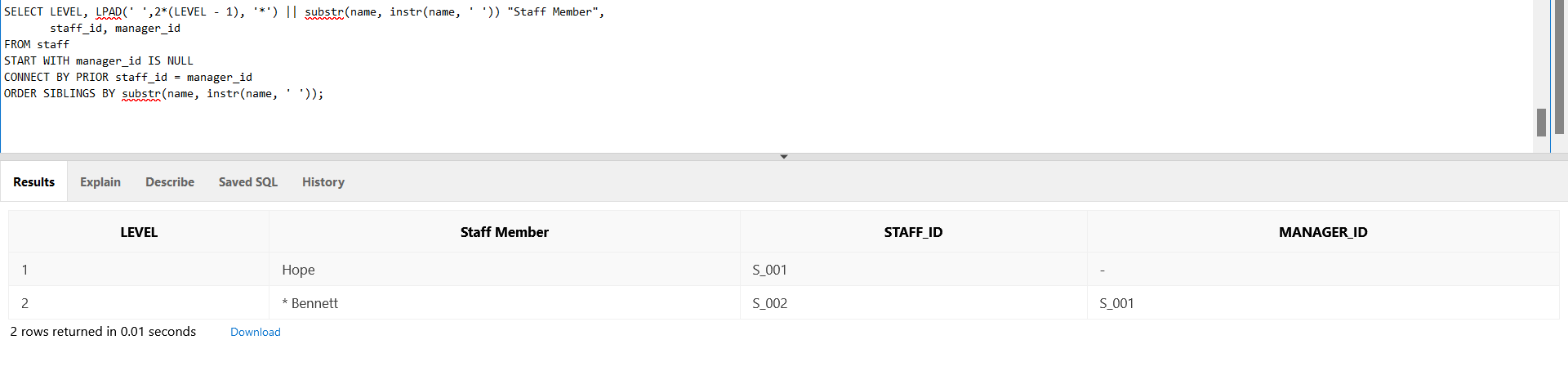
This is the anonymous block of the function that takes care of using and implementing the function.

### CURSOR



This cursor is used to print an employee name, their staff duty for every employee making over 1000 Euros (Combined tips and salary).

### HIERARCHICAL QUERY



This query returns the first two levels of managers and orders them by the last name. It adds an asterisk (\*) to the lower manager tier for every tier that they are lower than the first level (the top manager).

# Further Development

Depending on how the business is to be run in the future this project could be expanded in various different ways. For example, should the hotel resort create a chain across the country(s), they would have a larger database of employees or multiple tables containing different databases per resort location. This would create opportunities for better hierarchical queries to be run.

Furthermore, the tables could be worked on more in detail so that it would create more opportunistic queries overall. I.e. tables regarding jobs and facilities could be further expanded and added hierarchy within their own table so as to create a mini database of every facility within the resort.

Lastly, if the resort sees any large remodeling done, the table could see future possibilities of expansion, entering the pricing and new room types, expanding on facilities, etc.