K.M. Tahlil Mahfuz Faruk

Islamic University of Technology

Name: K.M. Tahlil Mahfuz Faruk

Student ID: 200042158

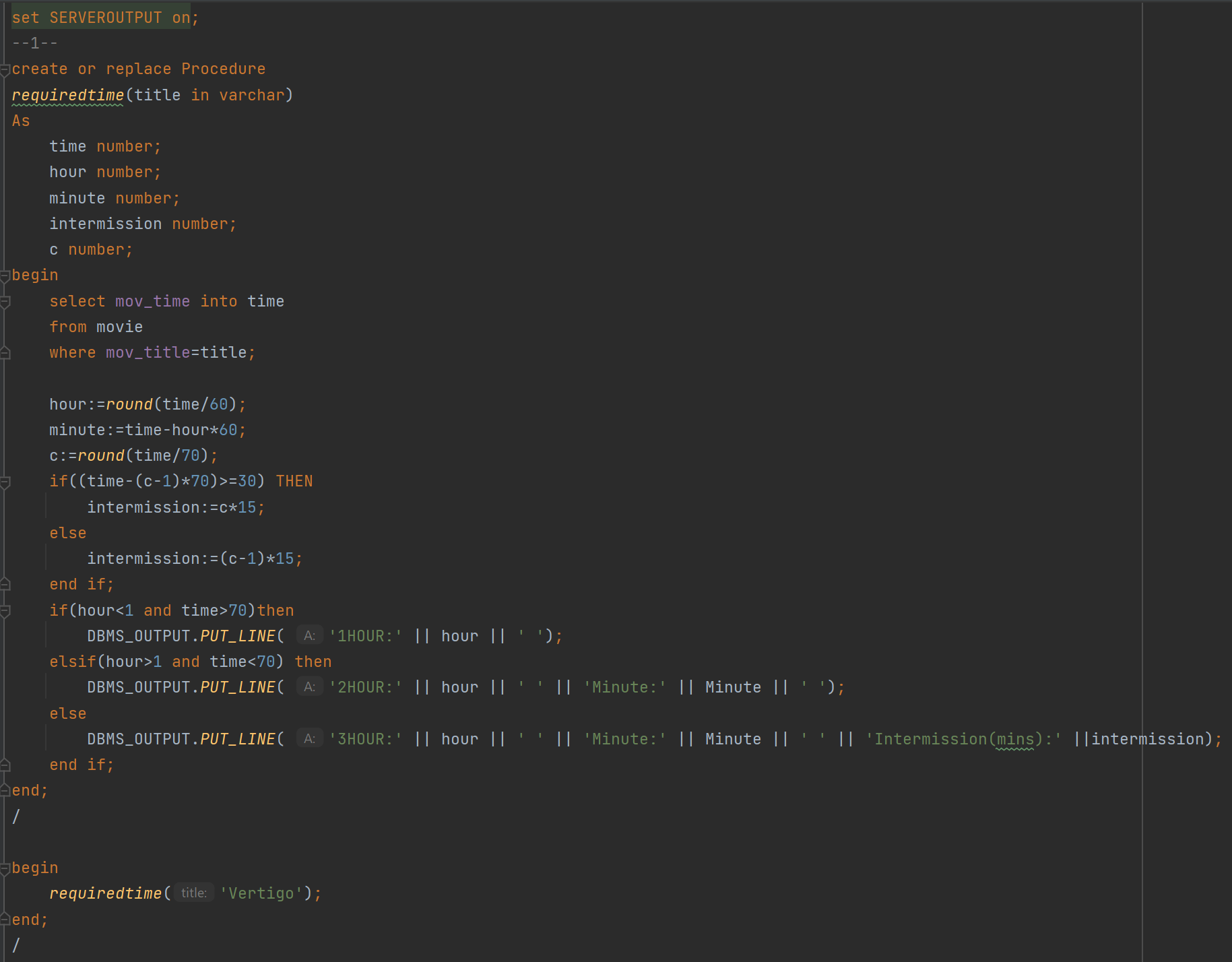
Department: CSE(SWE)

Course: CSE 4410

**Database Management System II LAB 3**

**SQL Commands:**

In task 1,



**Explanation:**

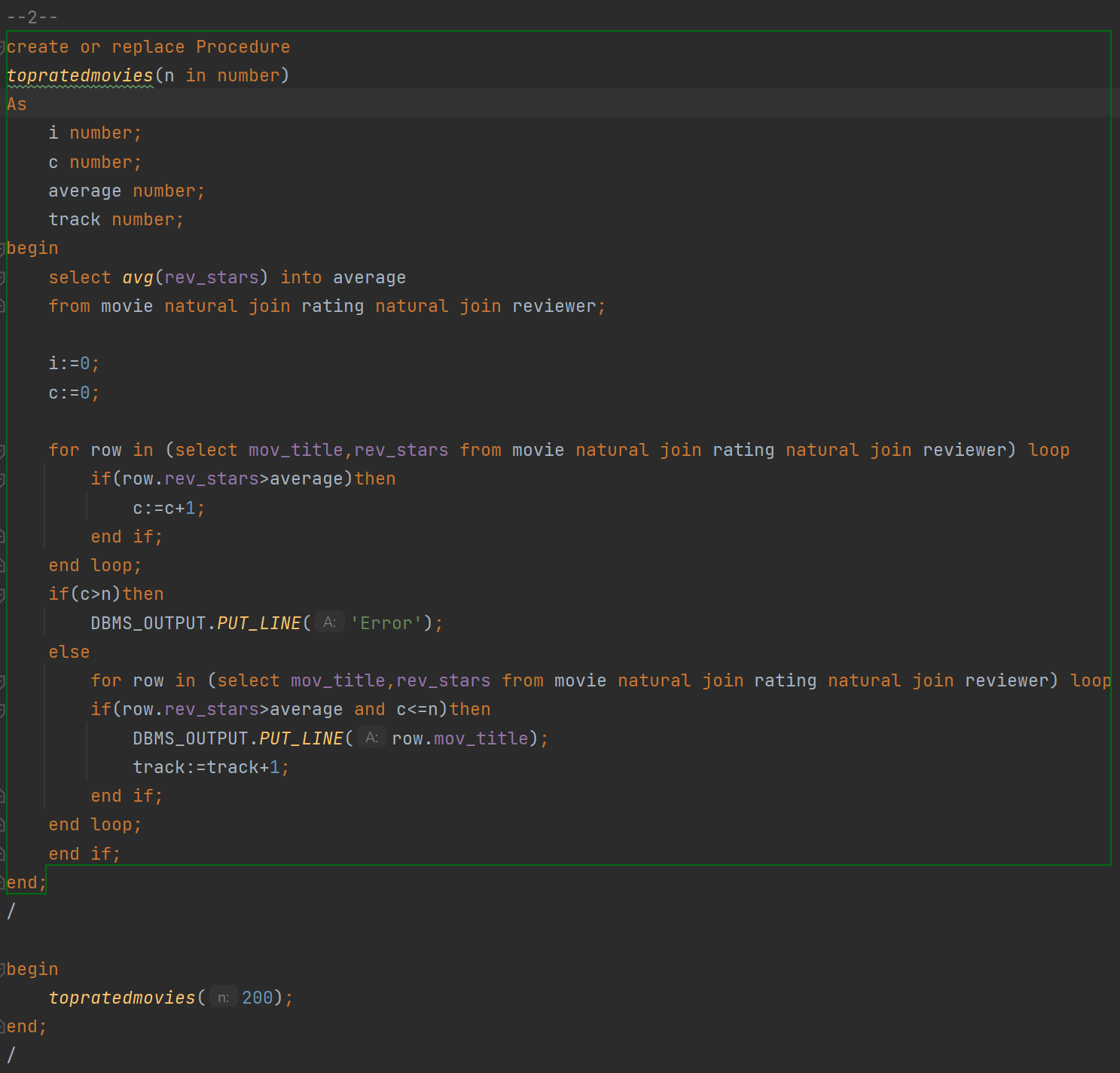
* A mov title will be passed in the function parameter.
* Find the title’s time. Divide it by 60 and round to find the hour. Then find the minute by subtracting the hour\*60 from the time.
* Now divide the time by 70 and round to find the interval time. Interval would be valid if remaining time is greater than 30 mins. Check that condition by

(c-1)\*70>=30.

**Difficulties:**

* No significant difficulty appeared.

In task 2,



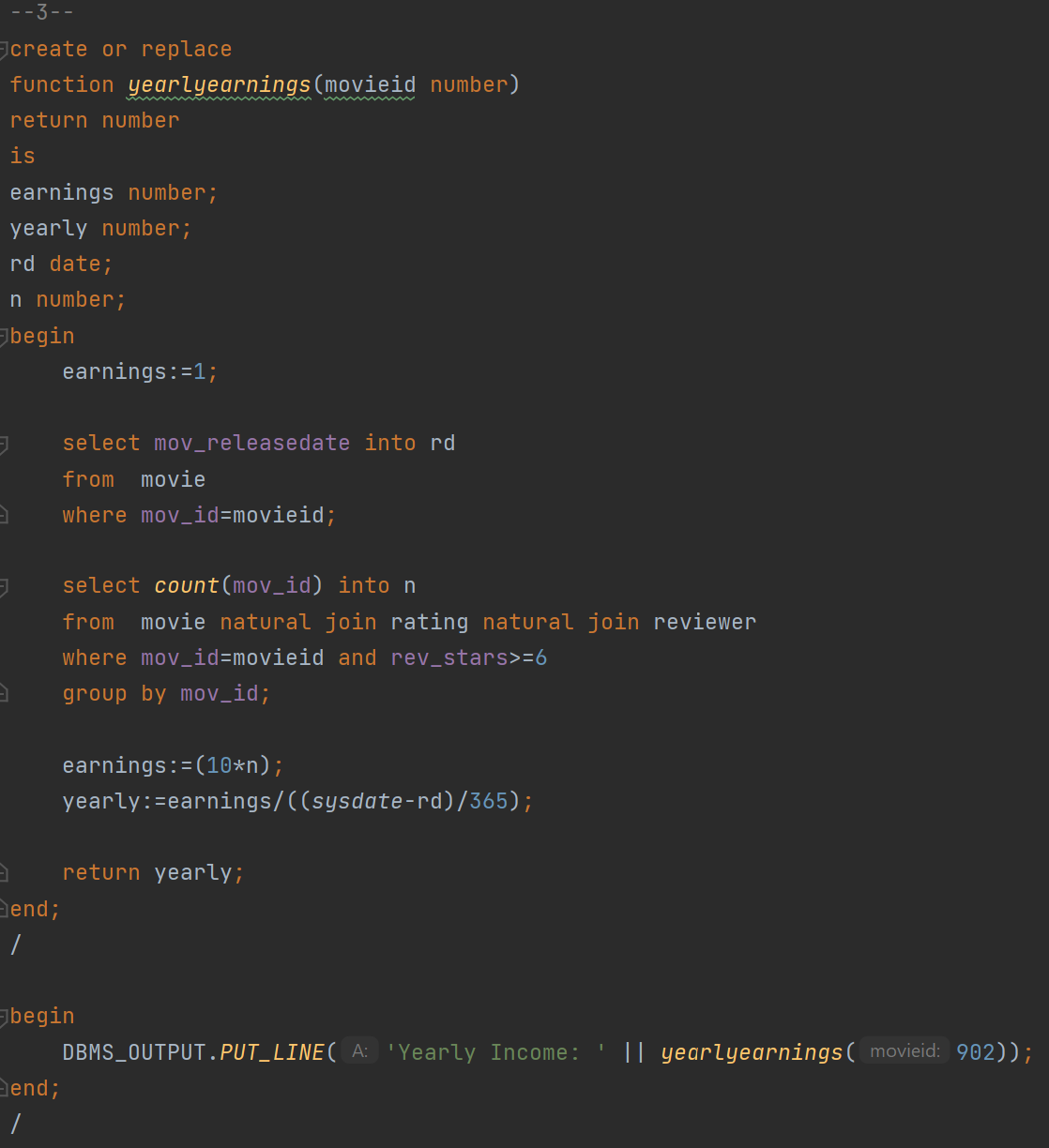
**Explanation:**

* First find the average rev\_stars and save it tin average variable.
* Now count the number of movies that has greater review than average.
* If it’s greater than n then print error.
* Else print those movies.

**Difficulties:**

* No such difficulties were faced during this task.

In task 3,

****

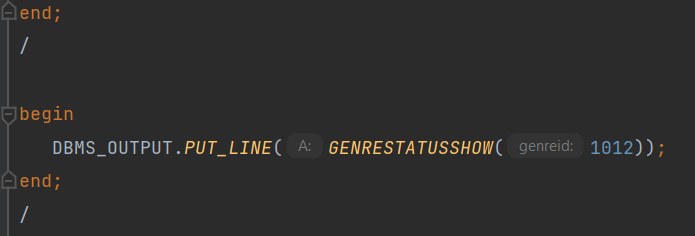
**Explanation:**

* First find the release data of the movie and also find the number of reviews that movie has got which is greater than 6.
* Now multiply by 10 with the number of reviews and then divide it by the year.
* That will provide the yearly income.

**Difficulties:**

* To find the year from the release data was the challenge.

In task 4,

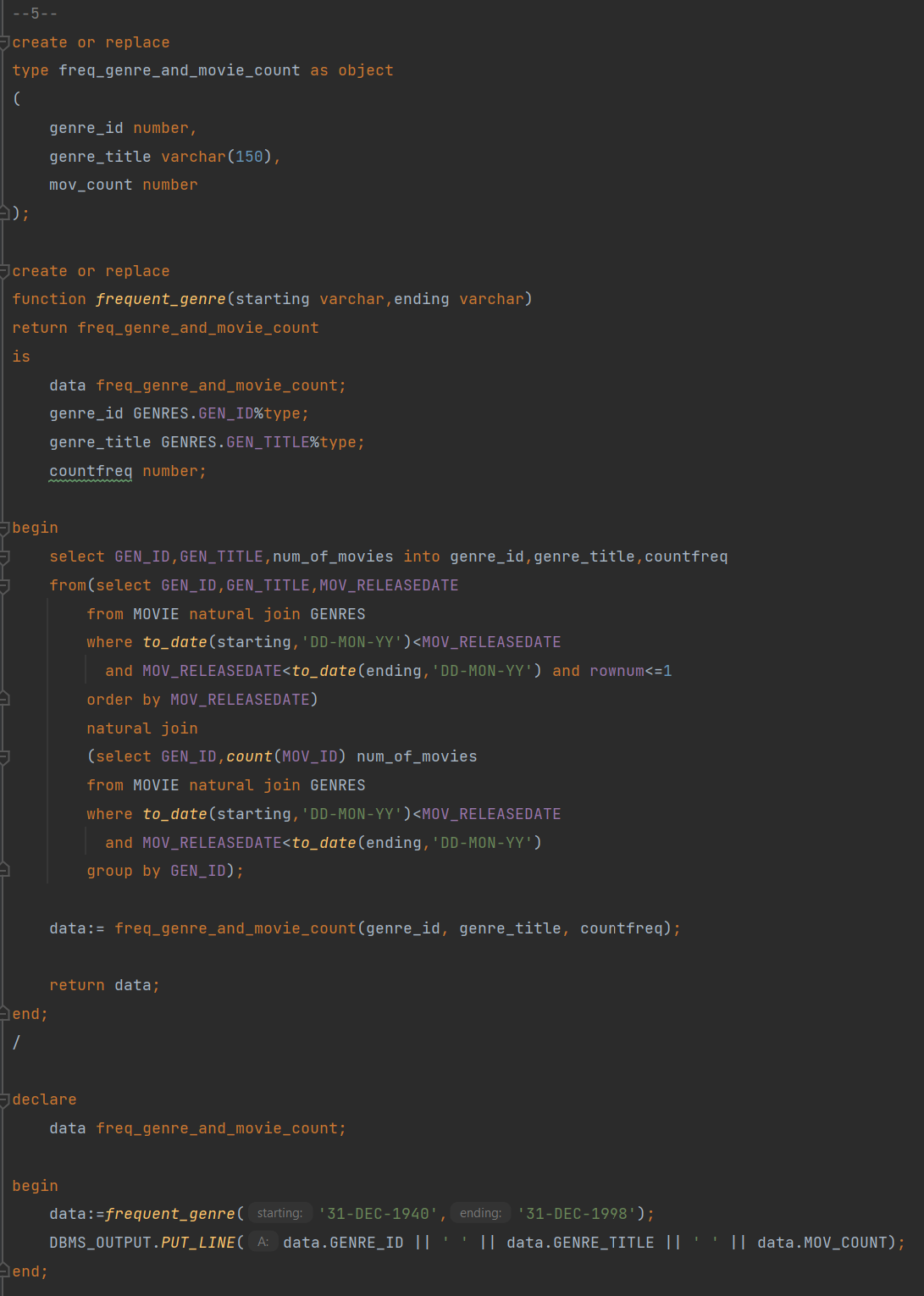
**Explanation:**

* First find the average review stars. To find it do a subquery.
* Also do a subquery to find the average review count.
* Now natural join both of these queries using the gen\_id that will provide the average stars and reviews and save the result in a cursor called dummytable.
* Now Find the average review and stars of all the genres using two queries and save them to variable avg\_review\_of\_all\_genres and avg\_stars\_of\_all\_genres respectively.
* Now compare according to condition given and return the genre status.

**Difficulties:**

* To do the subquery and get them to a table was the challenge.

In task 5,



**Explanation:**

* First create or replace a composite data type that will contain gen\_id, gen\_title and frequent movie counts.
* Now set that data type as the return value of the function.
* Do a subquery of finding the number of movies between the starting date and ending date and sort them according to release data. First row will contain the most frequent genre.
* Now to find the number of movies of that genre, just do a subquery to find the number of movies of each genre and in the range of given date and natural join with the previous subquery.
* Now save them is the composite datatype variable data.
* Return data.

**Difficulties:**

* The composite data type was showing an error that the “Reference to uninitialized composite”.
* To solve this we have to declare the data variable like this.

