Saroj Lamichhane

BOSTON UNIVERSITY

CS469 TERM PROJECT

Netflix 1.0

ODS Requirement and Design Document

Table of Contents

[Overview 3](#_Toc468830612)

[Business Requirement: 3](#_Toc468830613)

[Design: 3](#_Toc468830614)

[1) Table design 3](#_Toc468830615)

[**Functional design:** 4](#_Toc468830616)

[2) Business Rules: 6](#_Toc468830617)

[3) Conceptual ERD 8](#_Toc468830618)

[4) Logical ERD 8](#_Toc468830619)

[5) Use Case 9](#_Toc468830620)

[6) Index Creation 17](#_Toc468830621)

**Overview:**

Netflix 1.0 is a term project to determine the business requirement and design for an operational data storage(ODS) for the business.

Netflix 1.0 is a DVD rental business. The ODS will focus on the value chain operation of the DVD rental business. Any human resource, transactional or other data/design is out of scope.

**Business Requirement:**

|  |  |  |
| --- | --- | --- |
| S.N. | Requirement Description | Required(R)/Optional(O)/ Out of scope(NiS) |
| BRQ1 | Database should store:   1. member information 2. membership information 3. movie information 4. rental information 5. rental history 6. billing information | R |
| BRQ2 | There will be specific set of membership types offered to the Netflix 1.0 members, which is used to limit DVD issued | R |
| BRQ3 | Lost DVD, either reported or failing to return before membership cancellation will incur 1 time fixed charge for loss | R |
| BRQ4 | Despite member enrollment, DVD issue cycle follows calendar month schedule. | R |
| BRQ5 | Prorated fees calculation based on the date of member enrollment/reactivation | NiS |
| BRQ6 | ODS should handle archiving of rental history | R |
| BRQ7 | Rental return should trigger movie issue if valid for the members | R |
| BRQ8 | Automated monthly billing/bill totaling is out of scope; this has to be implemented in conjunction with Transactional database to support the DVD rental business. | NiS |
|  |  |  |

**Design:**

## **Table design**

The normalization choice of various data has been determined with consideration to the scope of the ODS, and primary focus of this project- the direct SQL user accessibility to query the needed data with ease. For this reason, the tables will not be fully normalized to BCNF or 3.5 Normal form. Tables will be normalized to 2nd degree and 3rd degree Normal form.

A typical Production database typically implements a high level of normalization to atomize data to avoid redundancy and achieve high database performance. Also at times atomized and normalized data would provide for better data validation. Example, member names and addresses could be designed into separate tables, and address validation (with street/zip/state lookups) could be easily maintained at data entry to ensure correct and clean data gets in the system.

For our intent and purpose data has been normalized to mostly 2nd degree normal form, and 3rd degree normal form.

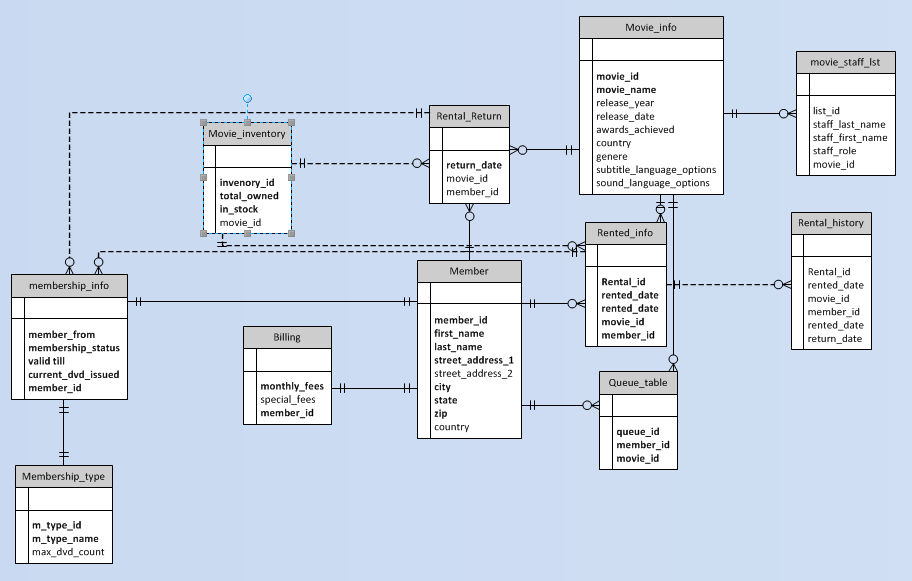
### **Functional design:**

|  |  |  |  |
| --- | --- | --- | --- |
| MEMBER Table |  |  |  |
| **Field Name** | **Data Type** | **Constraint(if any)** | **example** |
| MEMBER\_ID | INTEGER | PRIMARY KEY | 1 |
| MEMBER\_FIRST\_NAME | VARCHAR(30) | NOT NULL | Saroj |
| MEMBER\_LAST\_NAME | VARCHAR(30) | NOT NULL | Lamichhane |
| STREET\_ADDRESS\_1 | VARCHAR(50) | NOT NULL | 238 apple st |
| STREET\_ADDRESS\_2 | VARCHAR(50) |  |  |
| CITY | VARCHAR(50) | NOT NULL | Hartford |
| STATE | VARCHAR(50) | NOT NULL | CT |
| ZIP | DECIMAL(5,0) | NOT NULL | 3123 |
| COUNTRY | VARCHAR(30) |  | USA |
|  |  |  |  |
|  |  |  |  |
| MEMBERSHIP\_TYPE table |  |  |  |
| **Field Name** | **Data Type** | **Constraint(if any)** | **example** |
| MEMBERSHIP\_TYPE\_ID | INTEGER | PRIMARY KEY | 1 |
| MEMBERSHIP\_TYPE\_DESCRIPTION | VARCHAR(30) |  | 2 at a time |
| MAX\_DVD\_ALLOWED\_PER\_MONTH | INTEGER |  | 3 |
| DVD\_AT\_A\_TIME | INTEGER |  | 2 |
|  |  |  |  |
| MEMBERSHIP\_INFO table |  |  |  |
| **Field Name** | **Data Type** | **Constraint(if any)** | **example** |
| MEMBER\_ID | INTEGER |  | 1 |
| MEMBER\_FROM | DATE |  | 12/8/2016 |
| MEMBER\_TILL | DATE |  | 12/9/2016 |
| MEMBERSHIP\_STATUS | VARCHAR(15) | CURRENT/EXPIRED | CURRENT |
| MEMBERSHIP\_TYPE\_ID | INTEGER |  | 1 |
| CURRENT\_DVD\_ISSUED | INTEGER |  | 2 |
| MONTHLY\_DVD\_ISSUED\_COUNT | NUMBER |  | 4 |
| UPDATED\_DATESTAMP | DATE |  | 12/8/2016 |
| CURRENT\_MONTHEND | DATE |  | 31/8/2016 |
|  |  |  |  |
| MOVIE\_INFO table |  |  |  |
| **Field Name** | **Data Type** | **Constraint(if any)** | **example** |
| MOVIE\_ID | INTEGER | PRIMARY KEY | 1 |
| MOVIE\_NAME | VARCHAR(50) | NOT NULL | Black Magic |
| RELEASE\_DATE | DATE |  | 12/9/2016 |
| AWARDS\_ACHIEVED | VARCHAR(100) |  | Oscar |
| COUNTRY\_OF\_ORIGIN | VARCHAR(30) |  | USA |
| GENERE | VARCHAR(100) |  | Comedy |
| SUBTITLE\_LANGUAGE\_OPTIONS | VARCHAR(100) |  | Spanish |
| SOUND\_LANGUAGES\_OPTIONS | VARCHAR(100) |  | Spanish, French |
|  |  |  |  |
|  |  |  |  |
| MOVIE\_STAFF\_LIST table |  |  |  |
| **Field Name** | **Data Type** | **Constraint(if any)** | **example** |
| MOVIE\_ID | INTEGER |  | 1 |
| M\_STAFF\_FIRST\_NAME | VARCHAR(30) | NOT NULL | Steven |
| M\_STAFF\_LAST\_NAME | VARCHAR(30) |  | Splat |
| M\_STAFF\_ROLE | VARCHAR(100) |  | Actor |
|  |  |  |  |
|  |  |  |  |
| RENTAL\_INFO table |  |  |  |
| **Field Name** | **Data Type** | **Constraint(if any)** | **example** |
| RENTAL\_ID | INTEGER | NOT NULL | 1 |
| MEMBER\_ID | INTEGER | NOT NULL | 1 |
| MOVIE\_ID | INTEGER | NOT NULL | 1 |
| RENTED\_DATE | DATE | NOT NULL | 12/9/2016 |
| RETURN\_DATE | DATE |  | 12/9/2016 |
| RETURN\_LOST\_INDICATOR | VARCHAR(10) |  | N |
|  |  |  |  |
|  |  |  |  |
| RENTAL\_HISTORY table |  |  |  |
| **Field Name** | **Data Type** | **Constraint(if any)** | **example** |
| RENTAL\_ID | INTEGER |  | 1 |
| MEMBER\_ID | INTEGER |  | 1 |
| MOVIE\_ID | INTEGER |  | 1 |
| RENTED\_DATE | DATE |  | 12/9/2016 |
| RETURN\_DATE | DATE |  | 12/9/2016 |
| RETURN\_LOST\_INDICATOR | VARCHAR(6) |  | Y |
|  |  |  |  |
|  |  |  |  |
| QUEUE\_TABLE table |  |  |  |
| **Field Name** | **Data Type** | **Constraint(if any)** | **example** |
| QUEUE\_ID | INTEGER |  | 1 |
| MEMBER\_ID | INTEGER | NOT NULL | 1 |
| MOVIE\_ID | INTEGER | NOT NULL | 1 |
| SHIPPED\_STATUS | VARCHAR(1) | NOT NULL,DEFAULT('N') | N |
|  |  |  |  |
|  |  |  |  |
| MOVIE\_INVENTORY table |  |  |  |
| **Field Name** | **Data Type** | **Constraint(if any)** | **example** |
| MOVIE\_ID | INTEGER | NOT NULL | 1 |
| MOVIE\_NAME | VARCHAR(50) |  | Titanic |
| INITIAL\_TOTAL\_INVENTORY | INTEGER | NOT NULL | 5 |
| INVENTORY\_IN\_HAND | INTEGER | NOT NULL | 5 |
| INVENTORY\_OUT | INTEGER | NOT NULL | 0 |
| LOST\_DVD\_COUNT | INTEGER |  | 0 |
|  |  |  |  |
| BILLING table |  |  |  |
| **Field Name** | **Data Type** | **Constraint(if any)** | **example** |
| MEMBER\_ID | INTEGER |  | 1 |
| MONTHLY\_FEES | DECIMAL(5,2) |  | 11.55 |
| INCIDENTAL\_FEES | DECIMAL(5,2) |  | 25.52 |

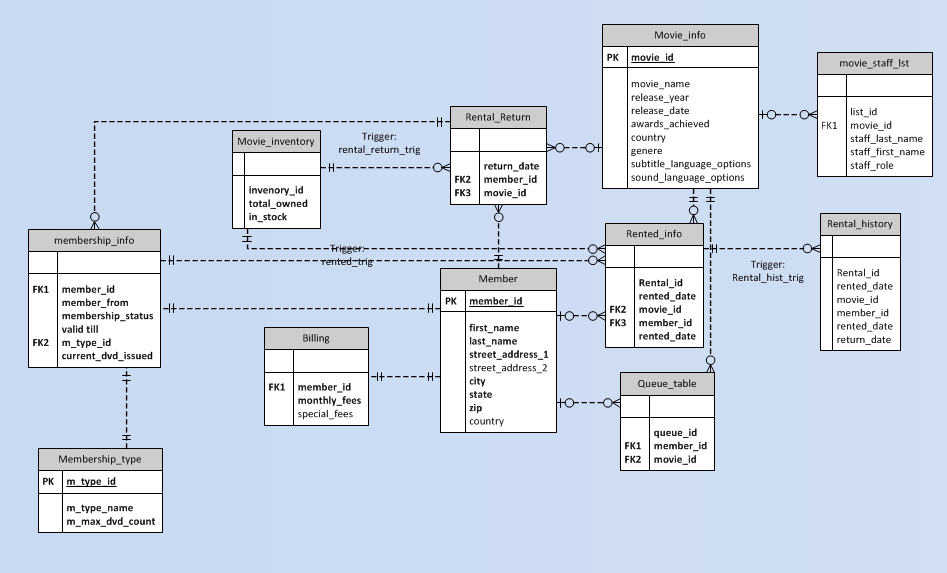
## **Business Rules:**

|  |  |
| --- | --- |
| **S.N.** | **Rule Description** |
| BR1 | Each Member will have a Membership information  Each Membership information is for a Member |
| BR2 | Each Member has a monthly Billing information  Each Billing information relates to a Member |
| BR3 | Each Membership information is related to a Membership type  A Membership type relates to a member’s Membership information |
| BR4 | Each Member has one or many current rental information  A rental information is for a member |
| BR5 | A movie has many staffs in the movie (like actor, director)  Staffs in a movie are related to a movie. |
| BR6 | A member can have no or many rental information  A Rental information will be of a member. |
| BR7 | Rental Info relates to the movie inventory  Movie inventory is related to none or many rented info |
| BR8 | Rental\_info is related to a movie.  A movie (has multiple DVDs in inventory, so) has none or many rented\_info |
| BR9 | A member has none or many items in queue  A queue information is of one member. |
| BR10 | A queue table entry has one movie info  A movie info might be none or many queue entries. |
| BR11 | A Rental\_return is related to an inventory entry  An inventory entry will be related none or many rental\_return |
| BR12 | A Rental\_return is related to exactly a member  A member is related with one or many rental\_return |
| BR13 | A Rental return is related to exactly one movie  A movie is related with none or many movie dvd returns. |
|  |  |

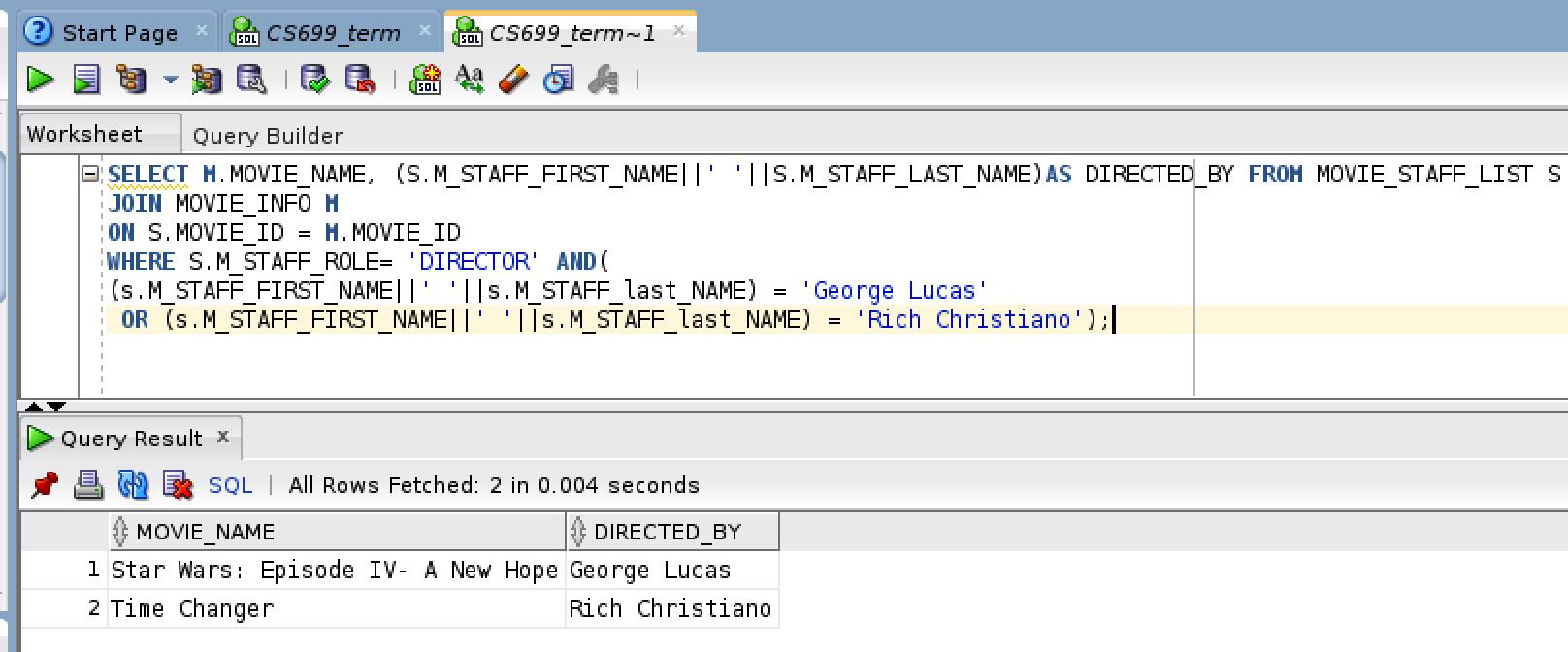
## **Conceptual ERD**

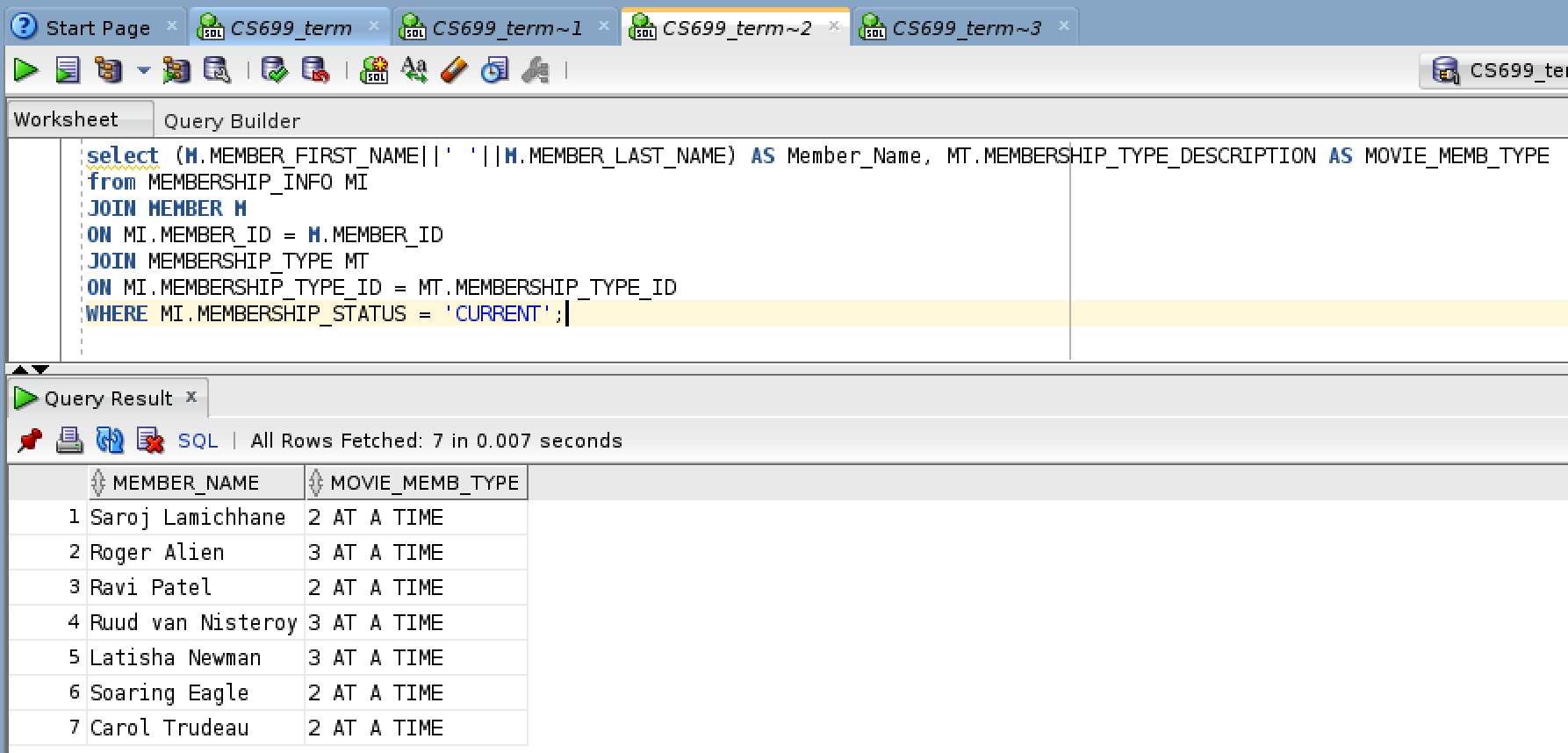


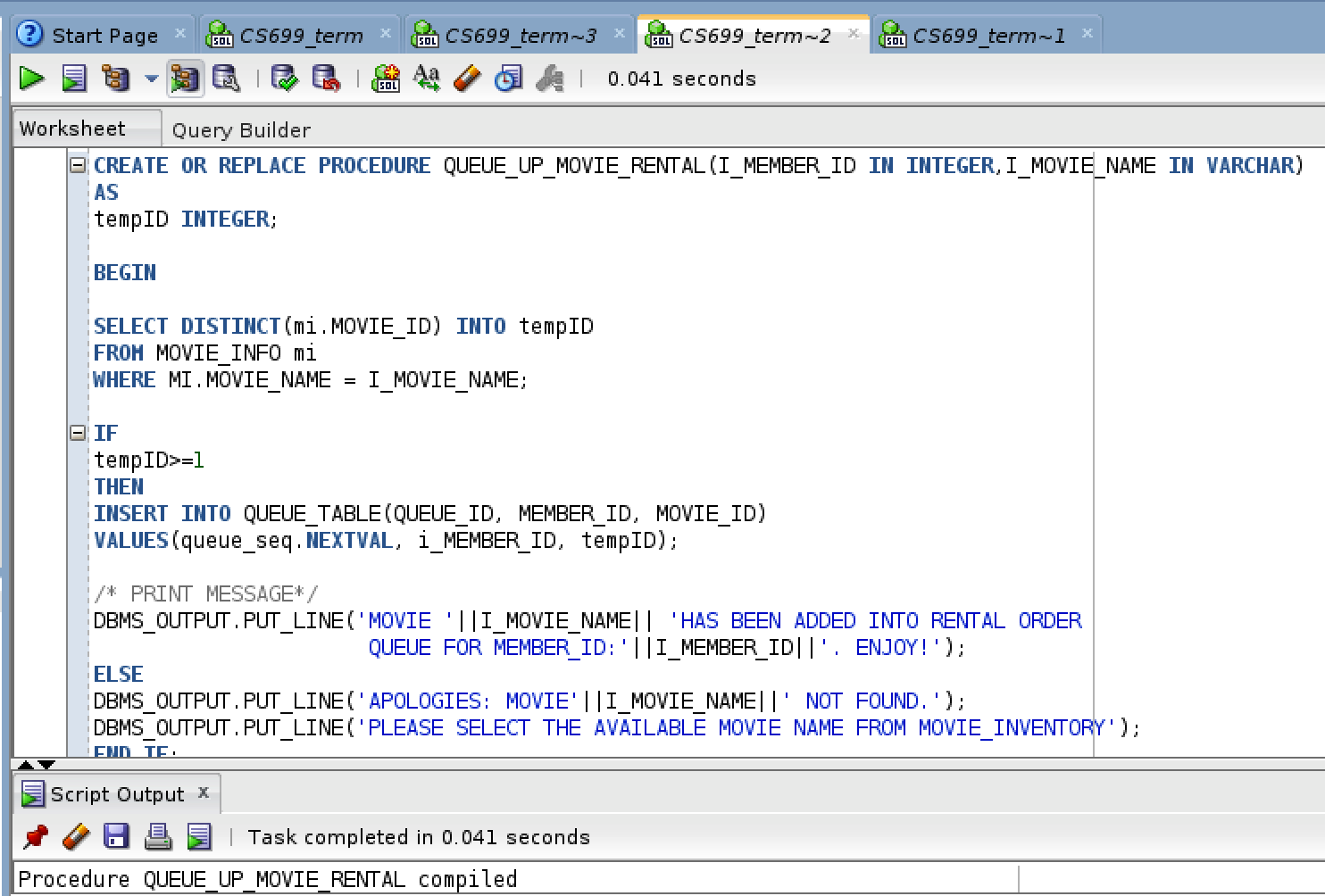
## **Logical ERD**

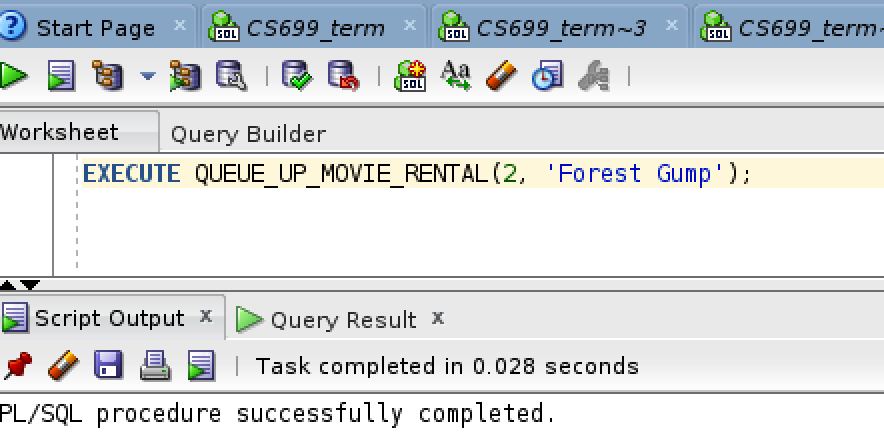


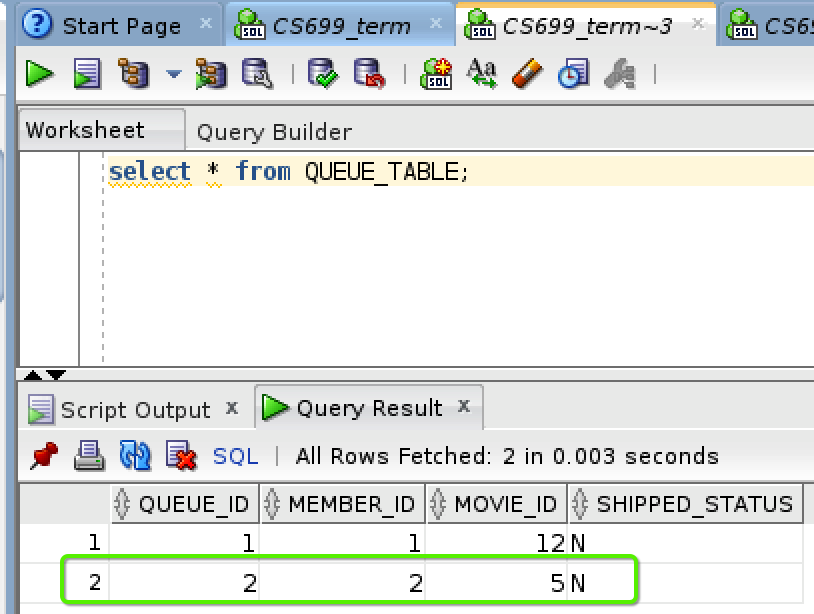
## **Use Case**

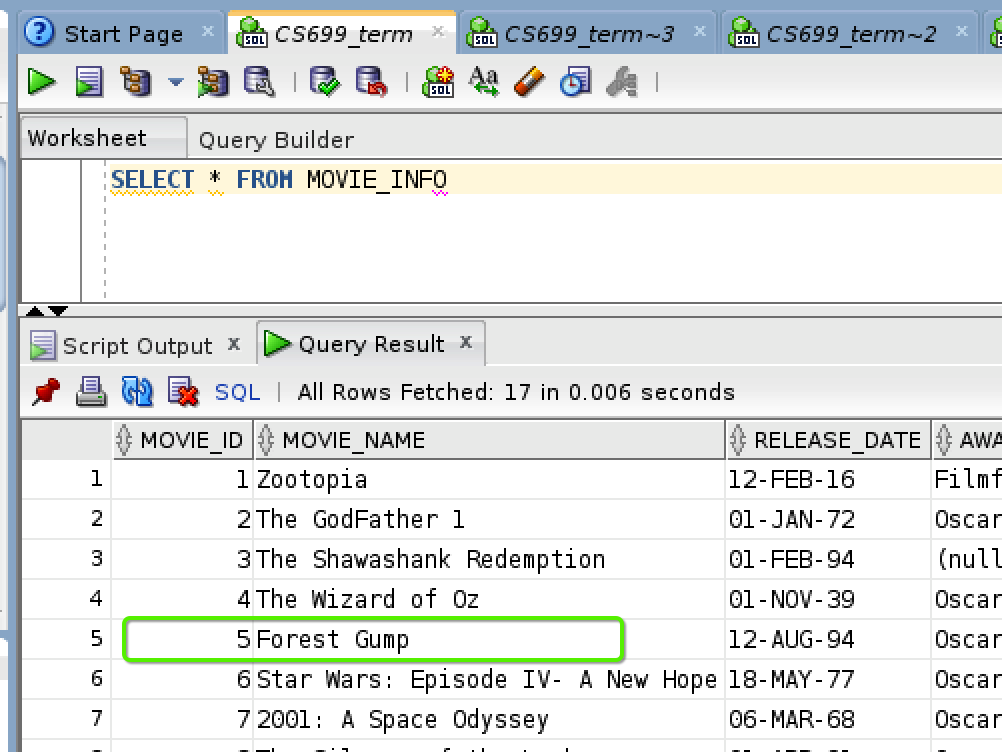


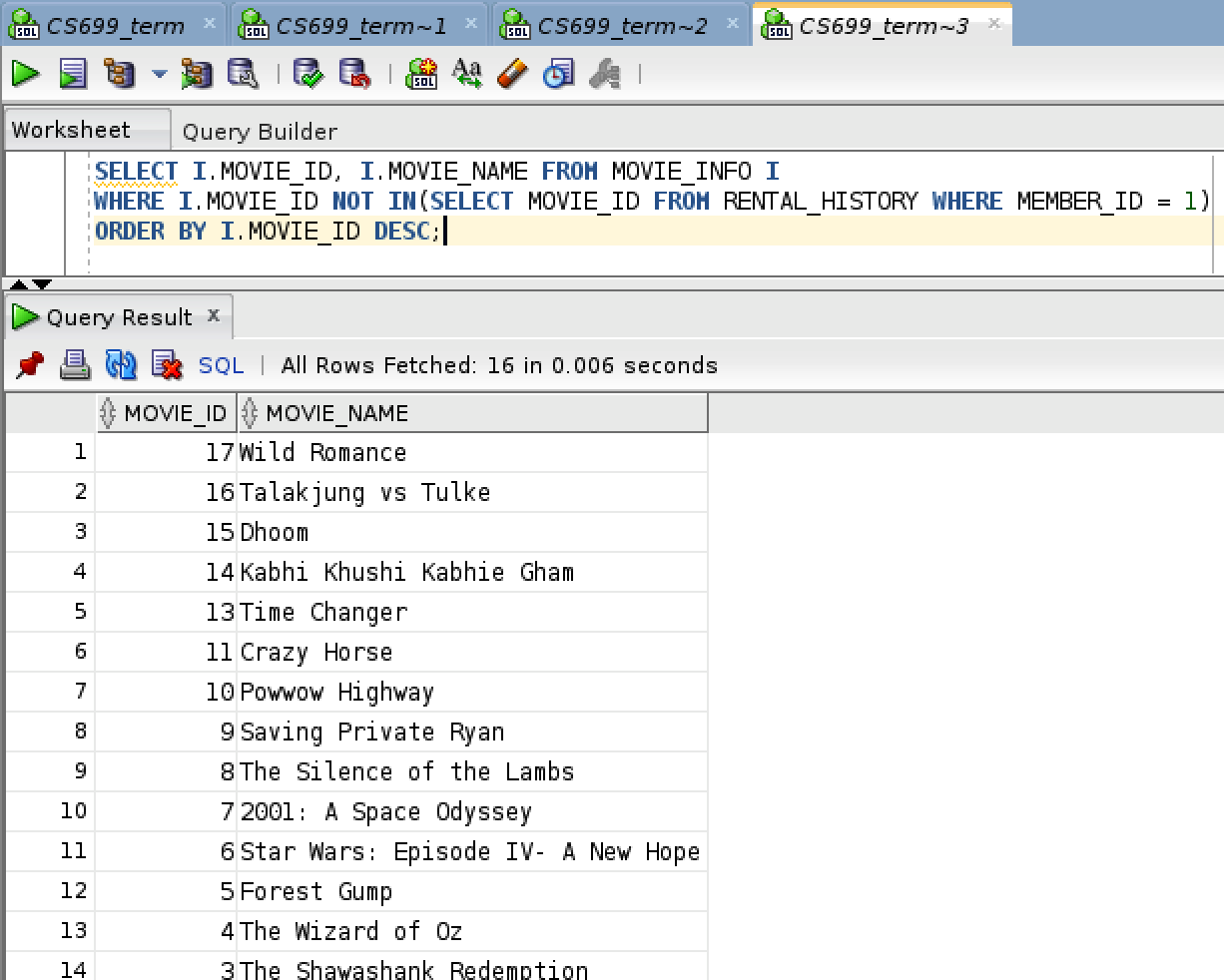


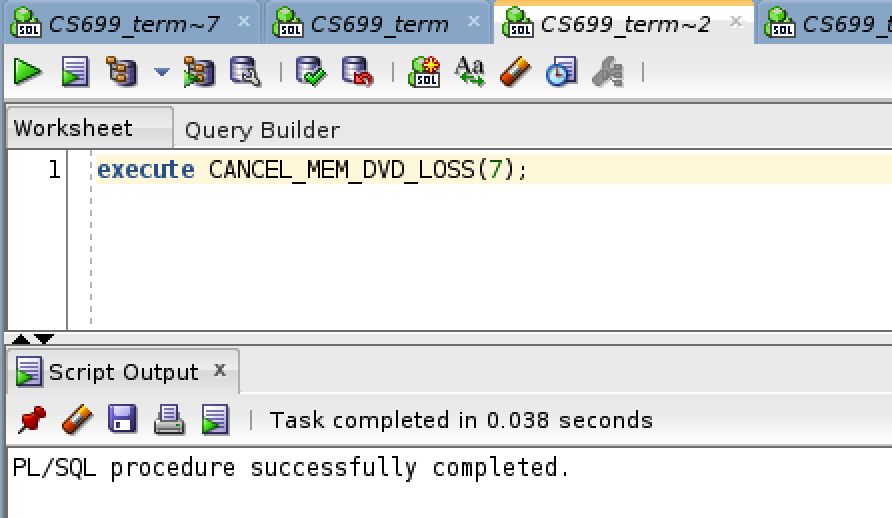


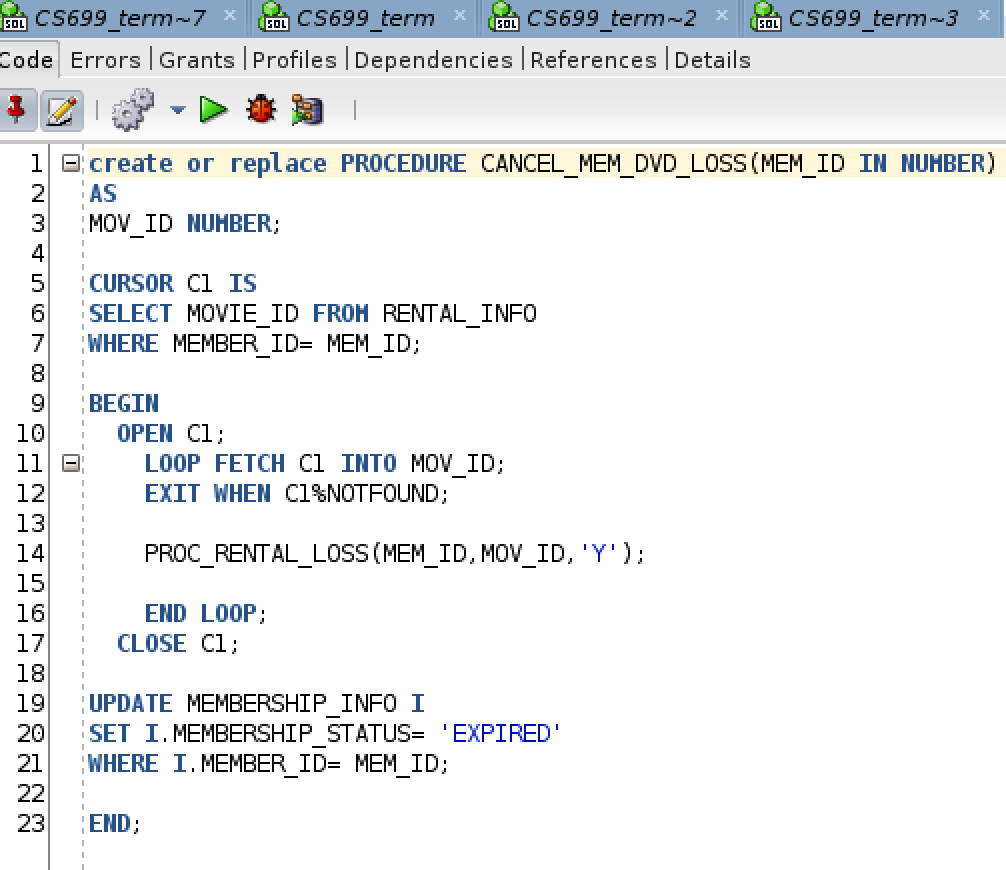


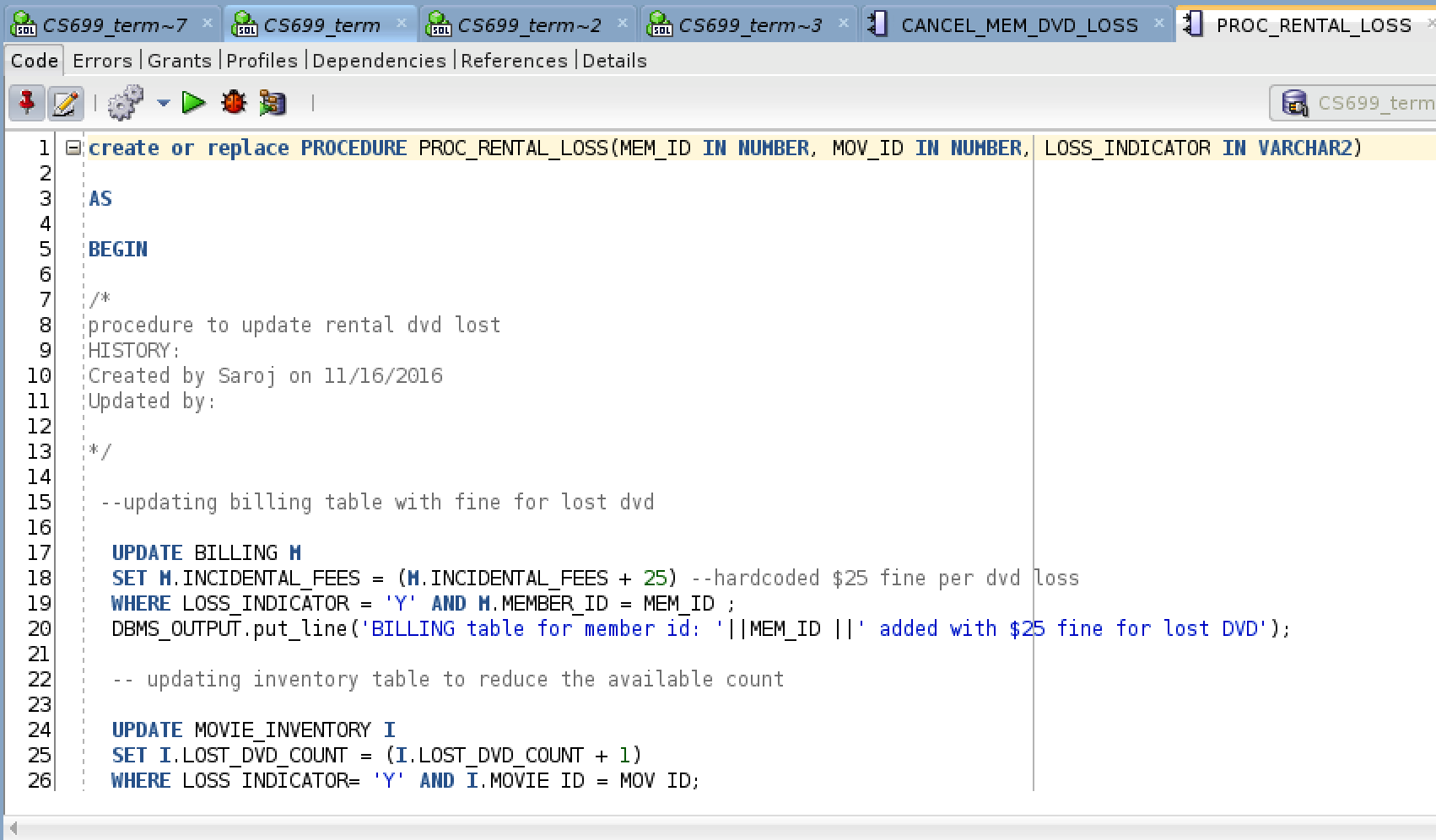


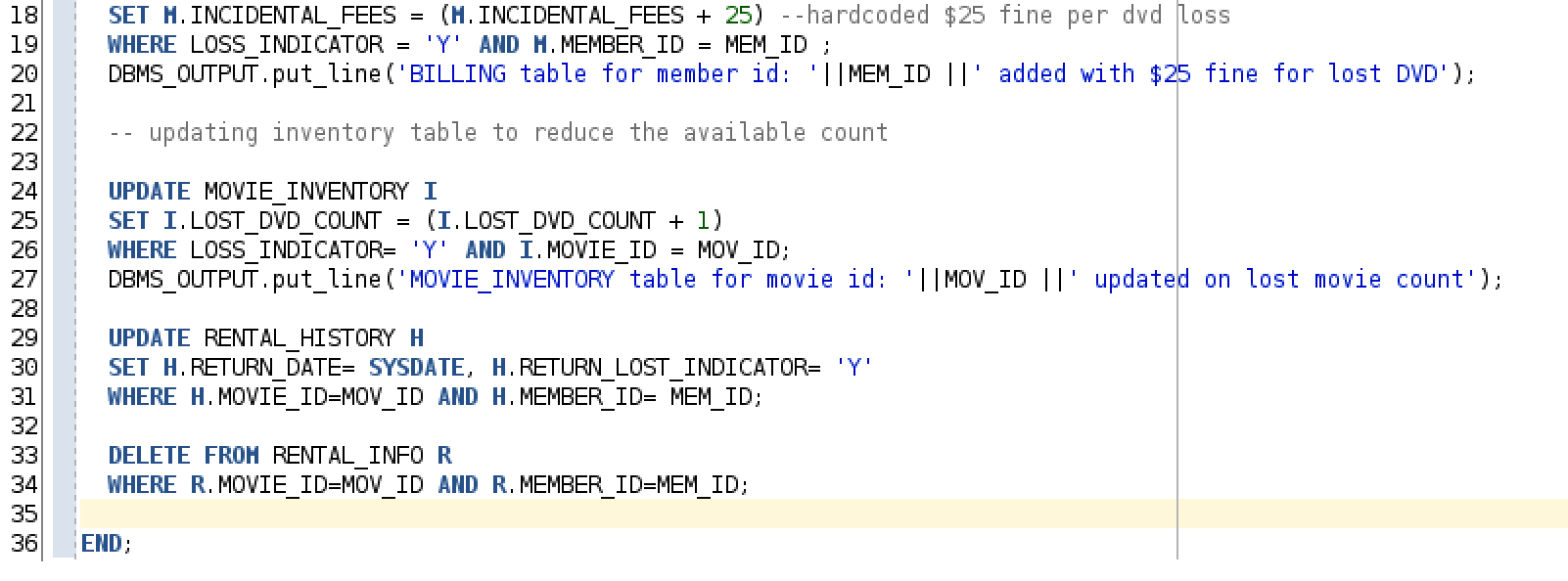


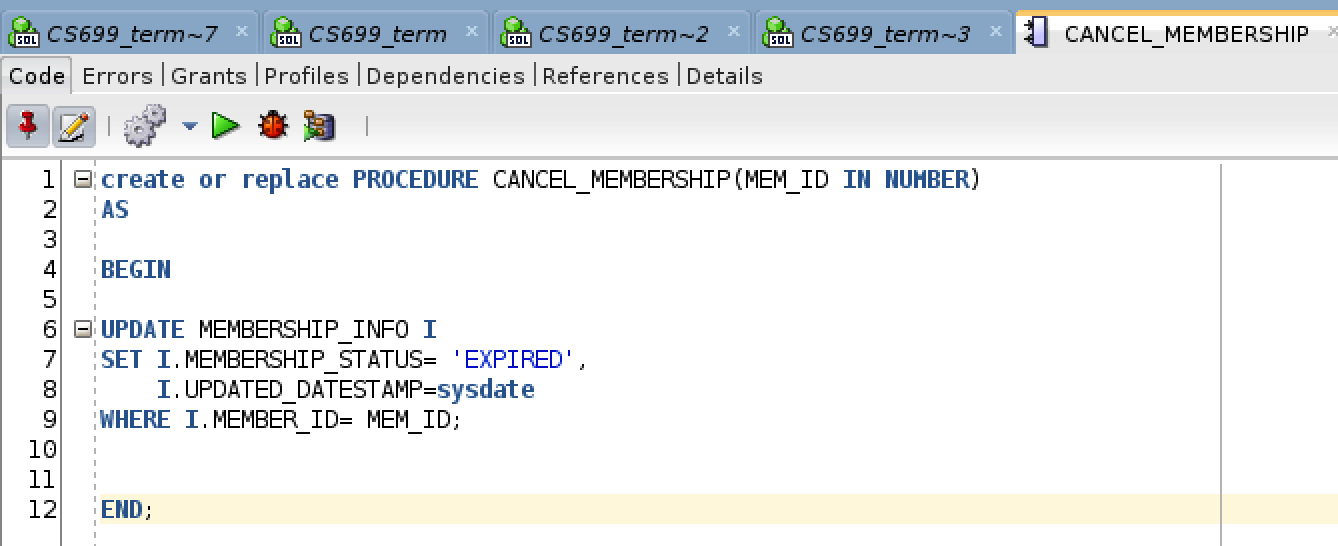


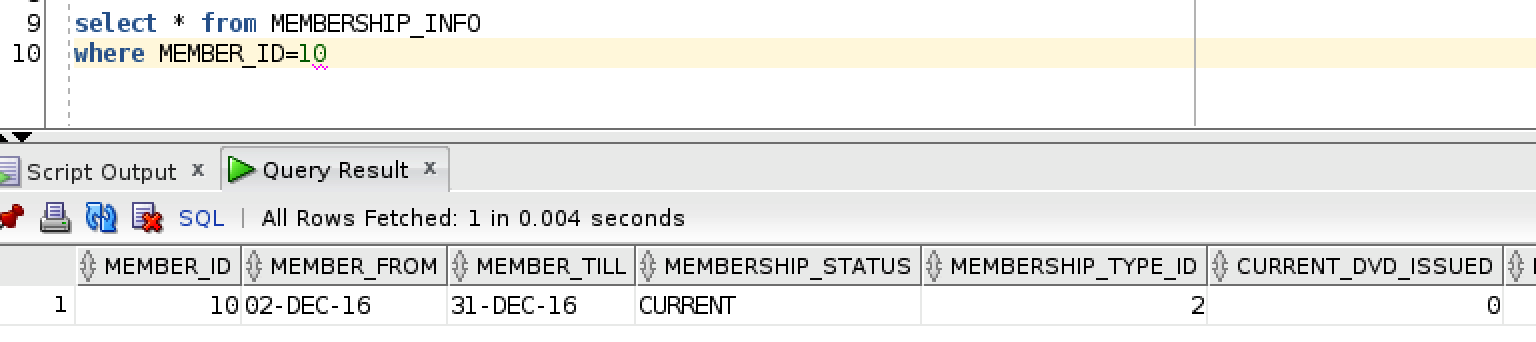


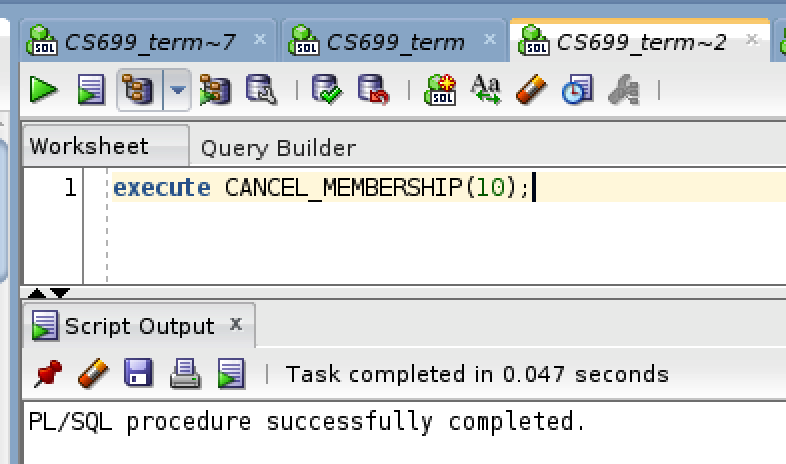


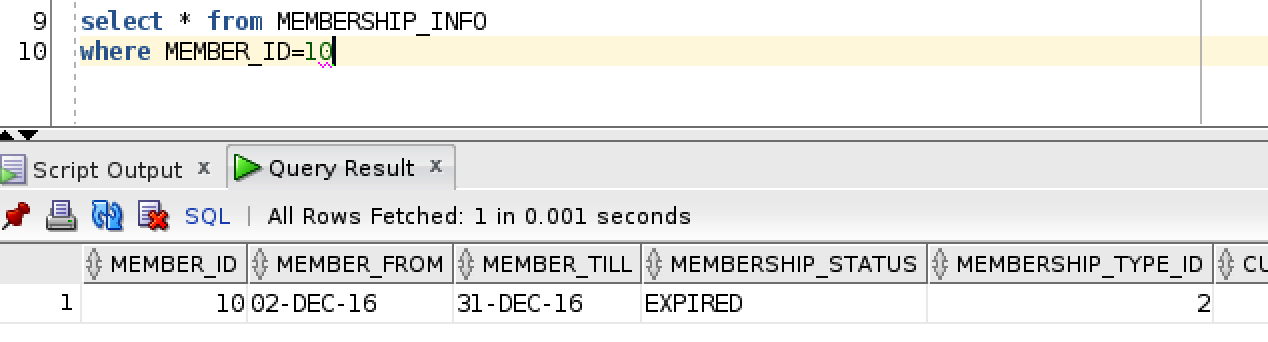


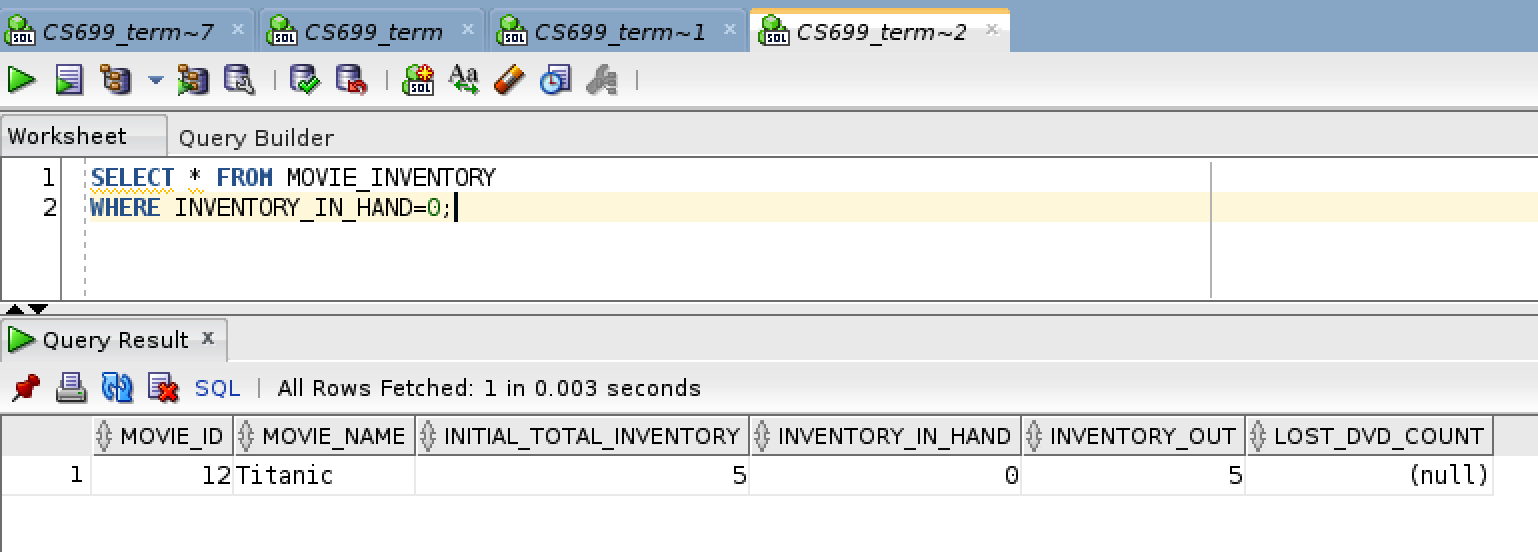


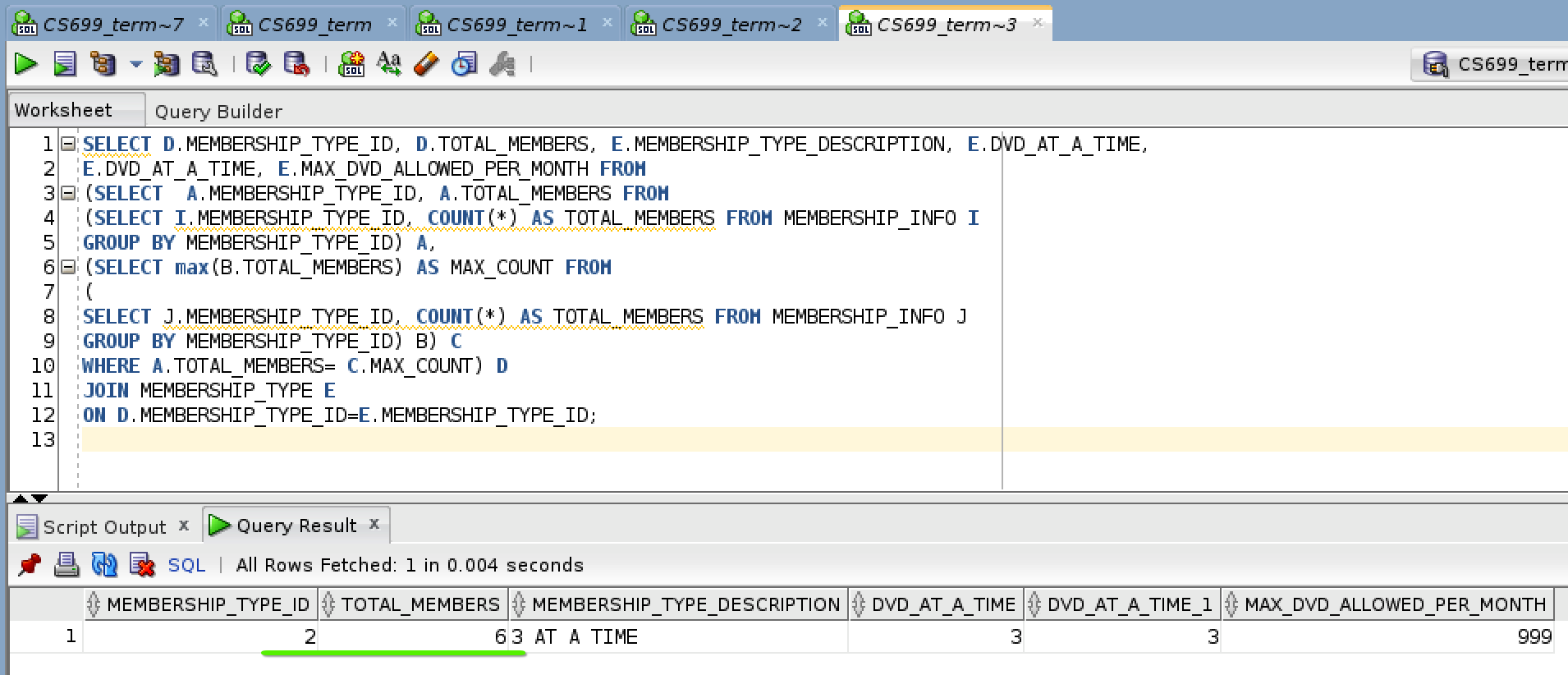


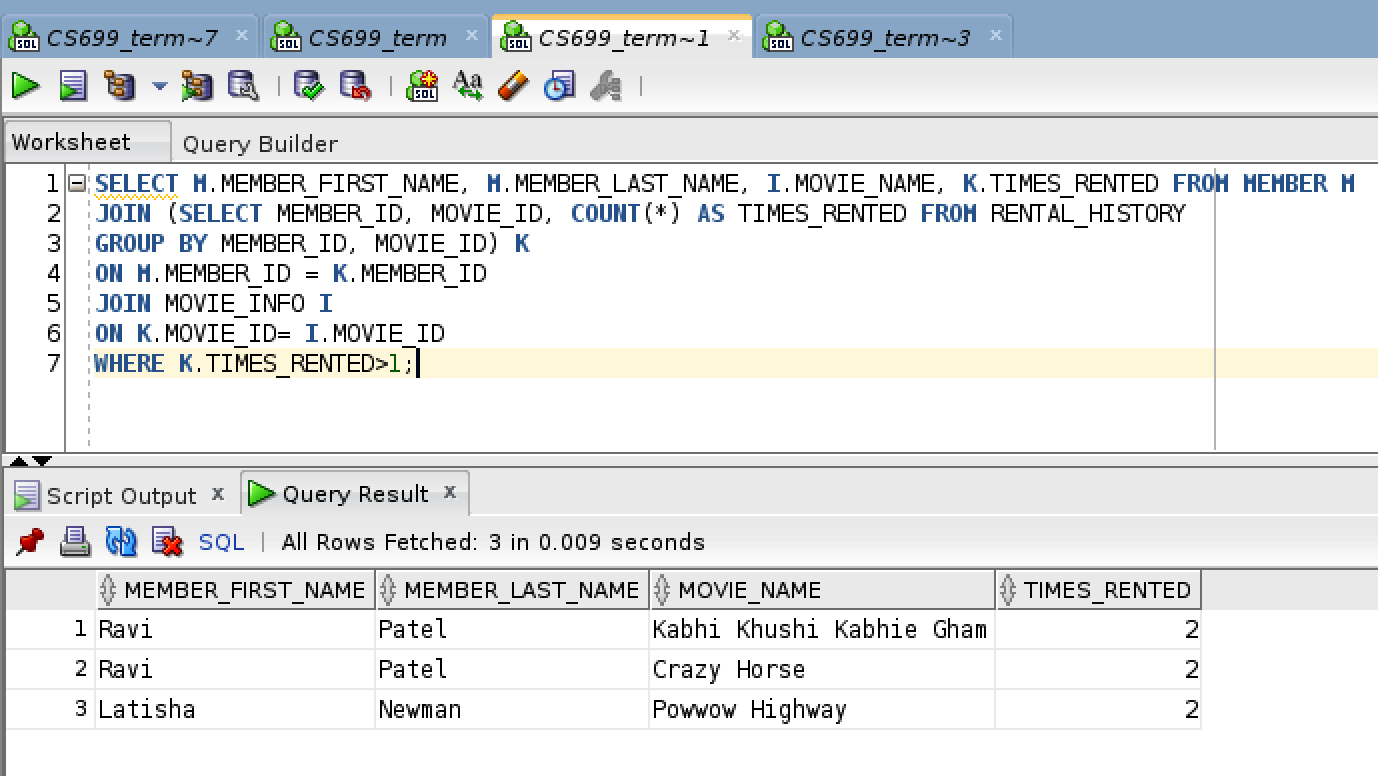




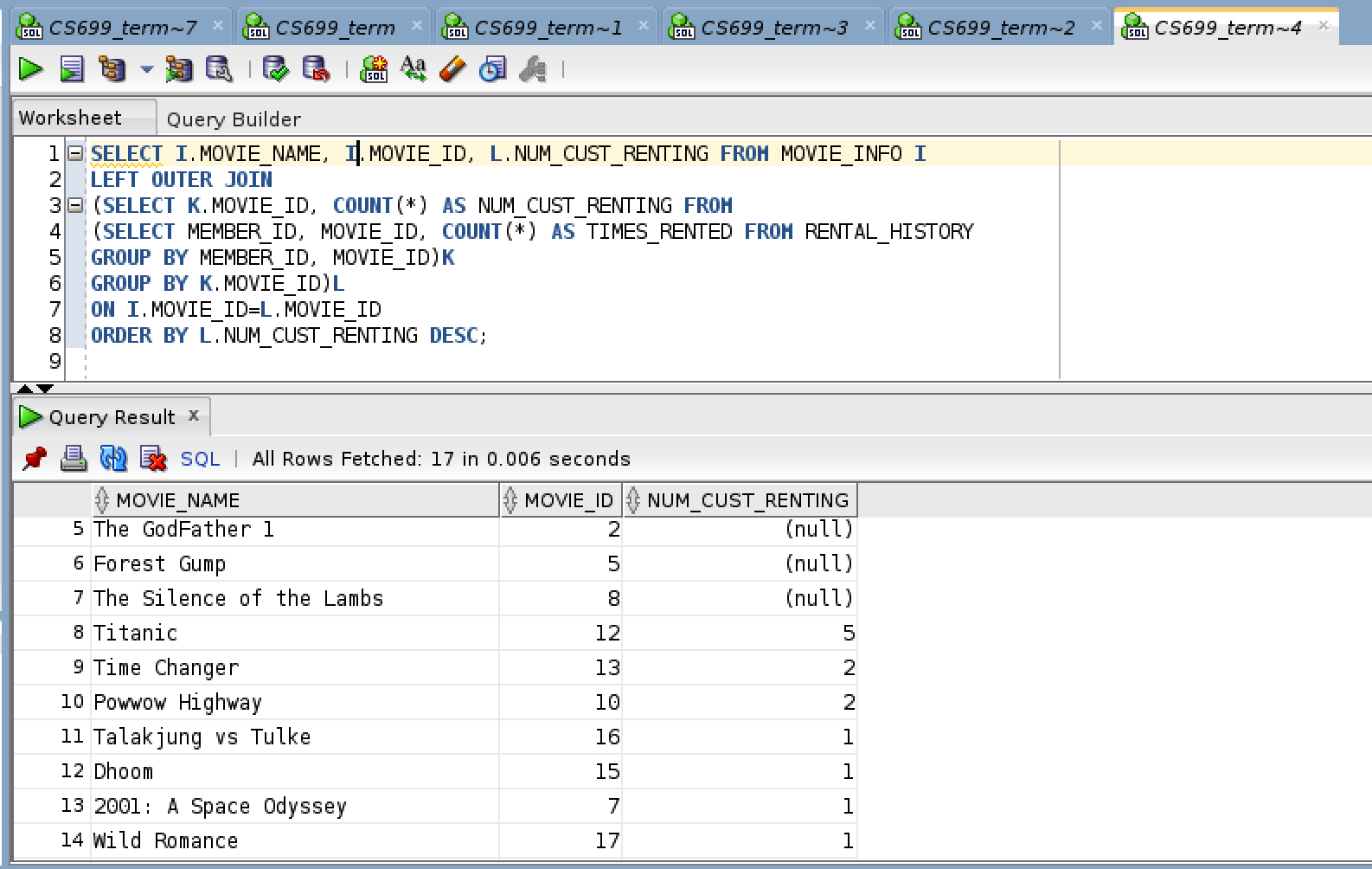




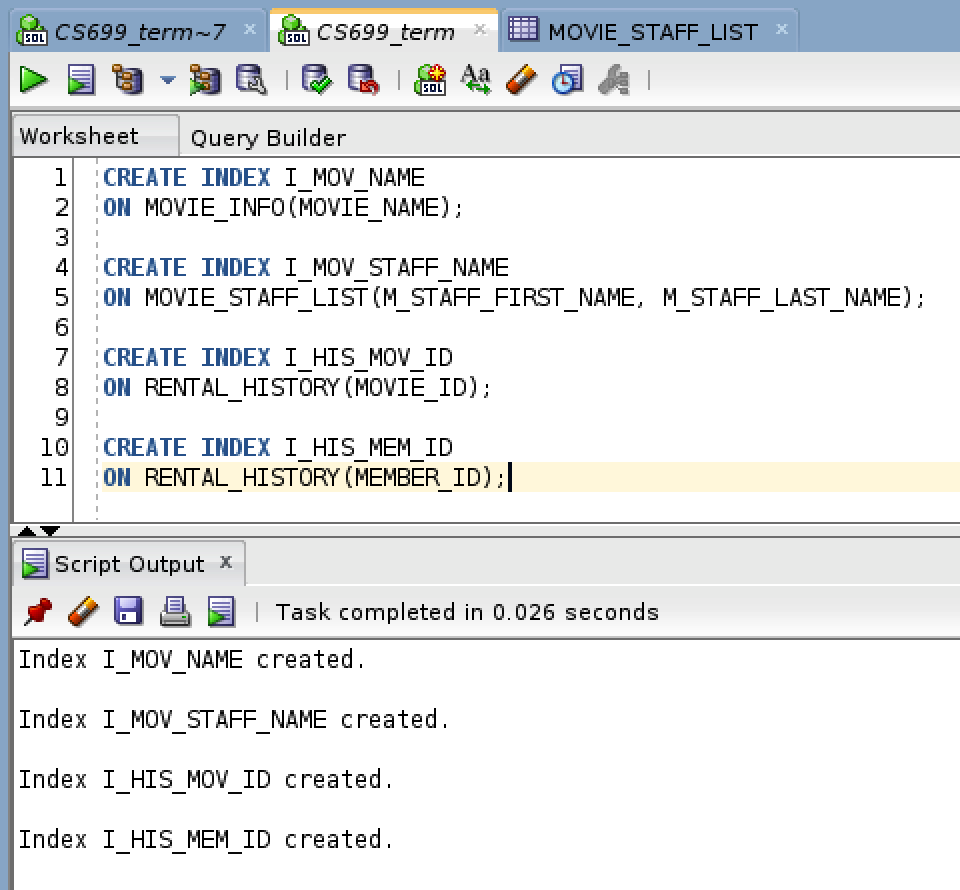




1. The following includes DVDs never rented as <null> coming on the top of the descending list. If these need to be removed the Join can be simply changed from LEFT OUTER to a simple join/inner join (JOIN).



## **Index Creation**



Four Indexes have been initially created anticipating the growth in the database for those particular tables, and their frequency of usage/querying.

**Index I\_MOV\_NAME:** This has been created in the Movie info table. The table holds all the movie names, awards, release information. This will be highly queried table particularly with Movie Name field. This is a varchar field. For the reason the index has been placed on it to optimize query performance.

**I\_Mov\_Staff\_Name:** Similar to movie name case, users will query the movies by staff names. To enhance the performance as the DB size(tablespace) grows for the movie\_staff\_list, an index has been created using both first and last name fields for the movie staff.

**I\_HIS\_MOV\_ID/ I\_HIS\_MEM\_ID:** Over time the movie history table will grow tremendously. Most query in the table is anticipated by movie\_id and/or member\_id fields and hence two indexes have been created on the table to optimize query time.