ASSIGNMENT 2 REPORT

Álvaro Castillo García
Pablo Ernesto Soëtard

Universidad Autónoma de Madrid

Escuela Politécnica Superior

Ingeniería Informática bilingüe, group 1251

Data Structures

For every program there is a main file: dvdreq.c, dvdfilm.c and dvdrent.c. In addition, all functions are declared on a file called functions.h and they are programmed on the file functions.c.

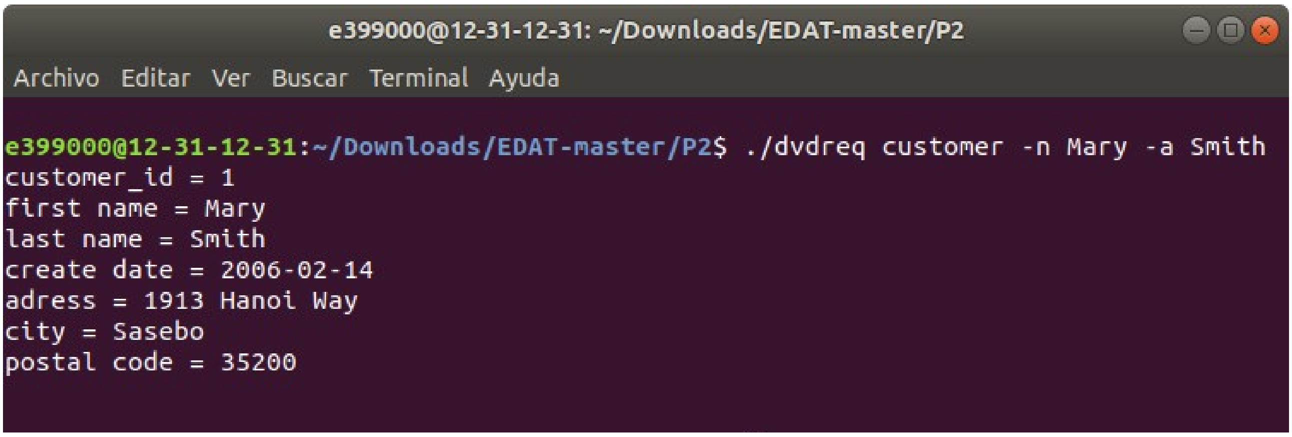
DVDREQ

dvdreq customer -n <First Name> -a <Last Name>

```
SELECT customer.customer_id,
       customer.first_name,
       customer.last_name,
       customer.create_date,
       address.address,
       city.city,
       address.postal_code
FROM   customer
       INNER JOIN address
              ON customer.address_id = address.address_id
       INNER JOIN city
              ON address.city_id = city.city_id
WHERE  customer.first_name = '%s'
        OR customer.last_name = '%s'
```

This query prints: the customer id, the customer’s first name, the customer’s last name, the registration date and the complete address of the customer indicated by the first and/or last name that is entered by the user. It just join the customer table with the city one and the address one to take all the parameters requested with the constraint of the name and last name (%s).

A sample run of the program is:


e399000@12-31-12-31: ~/Downloads/EDAT-master/P2
Archivo Editar Ver Buscar Terminal Ayuda

```
e399000@12-31-12-31:~/Downloads/EDAT-master/P2$ ./dvdreq customer -n Mary -a Smith
customer_id = 1
first name = Mary
last name = Smith
create date = 2006-02-14
adress = 1913 Hanoi Way
city = Sasebo
postal code = 35200
```

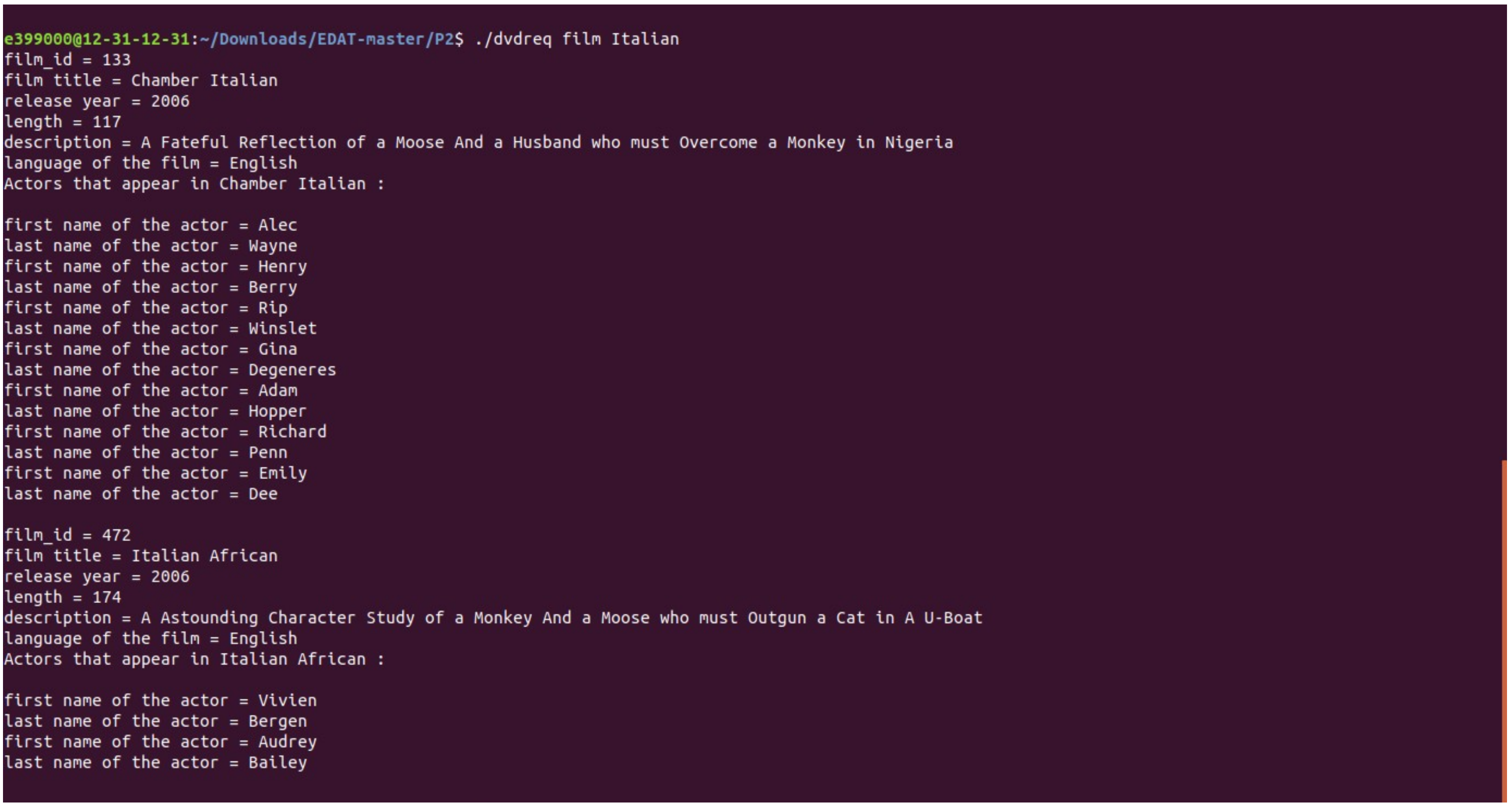
dvdreq film <title>

```
SELECT film.film_id,
       film.title,
       film.release_year,
       film.length,
       film.description,
       LANGUAGE.name
FROM   film,
       LANGUAGE
WHERE  film.title LIKE '%%s%'
       AND film.language_id = LANGUAGE.language_id;
```

```
SELECT actor.first_name,
       actor.last_name
FROM   actor,
       film_actor,
       film
WHERE  film.film_id = film_actor.film_id
       AND film_actor.actor_id = actor.actor_id
       AND film.film_id = %inputvalue;
```

The first query prints: the film id, the film title, the release year, the length, the language and the description for each film that matches fully or partially the title (%s). For this we have used the like operator.
The second query prints a list of actors with their first and last name for each film (%inputvalue). It just joins film and film_actor to get the name and last name.

A sample run of the program is:



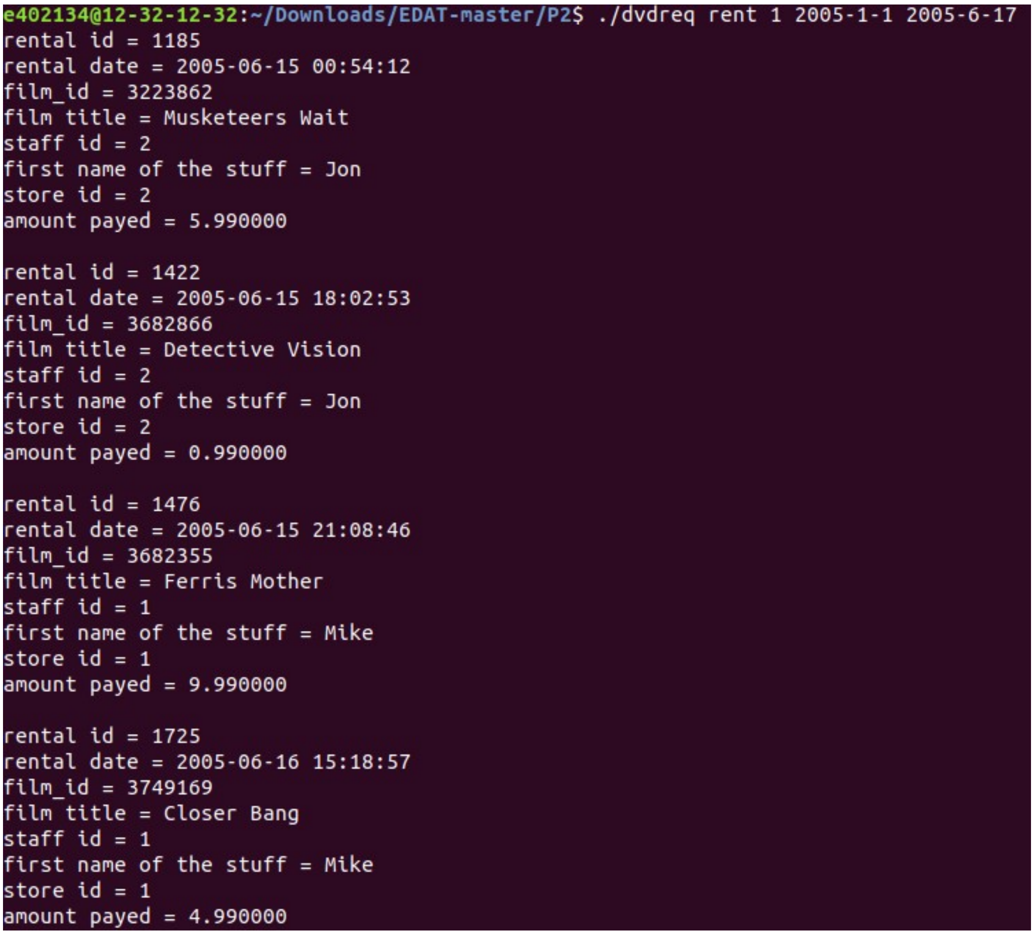
```
e399000@12-31-12-31:~/Downloads/EDAT-master/P2$ ./dvdreq film Italian
film_id = 133
film title = Chamber Italian
release year = 2006
length = 117
description = A Fateful Reflection of a Moose And a Husband who must Overcome a Monkey in Nigeria
language of the film = English
Actors that appear in Chamber Italian :

first name of the actor = Alec
last name of the actor = Wayne
first name of the actor = Henry
last name of the actor = Berry
first name of the actor = Rip
last name of the actor = Winslet
first name of the actor = Gina
last name of the actor = Degeneres
first name of the actor = Adam
last name of the actor = Hopper
first name of the actor = Richard
last name of the actor = Penn
first name of the actor = Emily
last name of the actor = Dee

film_id = 472
film title = Italian African
release year = 2006
length = 174
description = A Astounding Character Study of a Monkey And a Moose who must Outgun a Cat in A U-Boat
language of the film = English
Actors that appear in Italian African :

first name of the actor = Vivien
last name of the actor = Bergen
first name of the actor = Audrey
last name of the actor = Bailey
```

dvdreq rent <customer_id> <init date> <end date>

```
SELECT rental.rental_id,
      rental.rental_date,
      inventory.film_id,
      film.title,
      rental.staff_id,
      staff.first_name,
      staff.store_id,
      payment.amount
FROM   rental
      INNER JOIN inventory
             ON inventory.inventory_id = rental.inventory_id
      INNER JOIN film
             ON film.film_id = inventory.film_id
      INNER JOIN staff
             ON staff.staff_id = rental.staff_id
      INNER JOIN payment
             ON payment.rental_id = rental.rental_id
WHERE  rental.rental_date > 'InputDate'
      AND rental.rental_date < 'InputDate'
      AND payment.customer_id = InputCustomerId
ORDER  BY rental.rental_date
```

This query prints: the rental id, the rental date, the film id, the film title, the staff id, the first name of the staff, the store id and the amount payed, everything sorted by rental date. It just do several joins to get everything that we need in a single table, given the constrains that we have (customer_id, init_date, end_date), and then it selects everything that needs to be returned and orders it by rental_date.

A sample run of the program is:



```
e402134@12-32-12-32:~/Downloads/EDAT-master/P2$ ./dvdreq rent 1 2005-1-1 2005-6-17
rental id = 1185
rental date = 2005-06-15 00:54:12
film_id = 3223862
film title = Musketeers Wait
staff id = 2
first name of the stuff = Jon
store id = 2
amount payed = 5.990000

rental id = 1422
rental date = 2005-06-15 18:02:53
film_id = 3682866
film title = Detective Vision
staff id = 2
first name of the stuff = Jon
store id = 2
amount payed = 0.990000

rental id = 1476
rental date = 2005-06-15 21:08:46
film_id = 3682355
film title = Ferris Mother
staff id = 1
first name of the stuff = Mike
store id = 1
amount payed = 9.990000

rental id = 1725
rental date = 2005-06-16 15:18:57
film_id = 3749169
film title = Closer Bang
staff id = 1
first name of the stuff = Mike
store id = 1
amount payed = 4.990000
```

dvdreq recommend <customer Id>

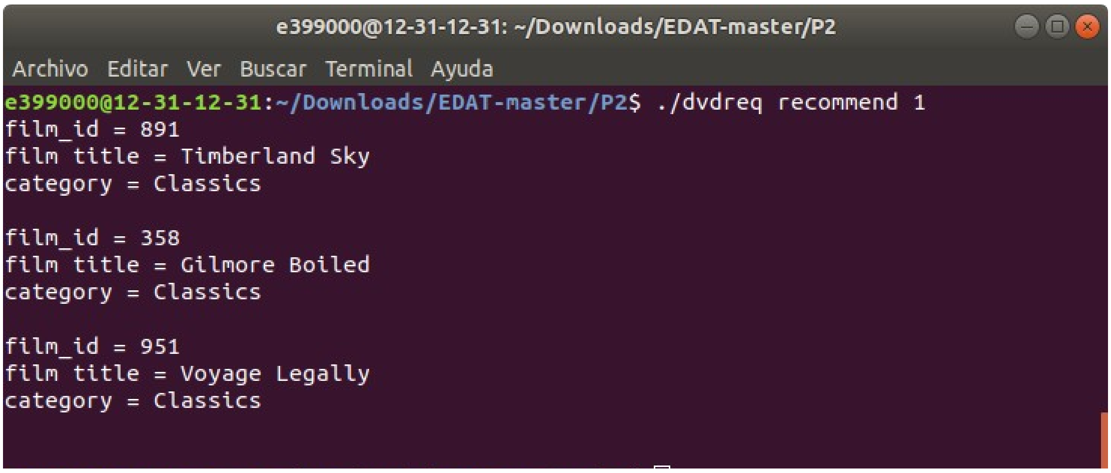
```
SELECT category.category_id
FROM   customer,
       rental,
       inventory,
       film,
       film_category,
       category
WHERE  customer.customer_id = "InputId"
       AND rental.customer_id = customer.customer_id
       AND rental.inventory_id = inventory.inventory_id
       AND inventory.film_id = film.film_id
       AND film.film_id = film_category.film_id
       AND film_category.category_id = category.category_id
GROUP  BY category.category_id,
          category.name
HAVING Count(*) IN
       (SELECT Count(*)
        FROM   customer,
               rental,
               inventory,
               film,
               film_category,
               category
        WHERE  customer.customer_id = "InputId"
               AND rental.customer_id = customer.customer_id
               AND rental.inventory_id = inventory.inventory_id
               AND inventory.film_id = film.film_id
               AND film.film_id = film_category.film_id
               AND film_category.category_id = category.category_id
        GROUP  BY category.category_id,
                  category.name
        ORDER  BY Count(*) DESC
        LIMIT  1)
```

This first query takes the most rented category of a customer, given by the customer id introduced ("InputId"). It is done in such a way that if the customer has two categories with the same number of films, then both will appear. We first count the number of times that rentals grouped by the category and we take the maximum. Then we select the categories that have that maximum.

```
SELECT film.film_id,
       film.title,
       category.name
FROM   customer,
       rental,
       inventory,
       film,
       film_category,
       category
WHERE  rental.customer_id = customer.customer_id
       AND rental.inventory_id = inventory.inventory_id
       AND inventory.film_id = film.film_id
       AND film.film_id = film_category.film_id
       AND film_category.category_id = category.category_id
       AND category.category_id = "%s"
       AND film.film_id NOT IN (SELECT film.film_id
                                FROM   customer,
                                       rental,
                                       film,
                                       inventory
                                WHERE  customer.customer_id = "InputId"
                                       AND
                               rental.customer_id = customer.customer_id
                                       AND rental.inventory_id =
                                           inventory.inventory_id
                                       AND inventory.film_id = film.film_id)
GROUP  BY category.category_id,
          film.film_id,
          film.title
ORDER  BY Count(*) DESC
LIMIT  3
```

The second query takes every film in the database that corresponds to the favorite category of the customer ("InputId") given by ("%s"). Then it takes the films that the customer has not watched and show all films except for this last ones. This query will be executed for all the favorite categories of the user.

A sample run of the program is:


e399000@12-31-12-31: ~/Downloads/EDAT-master/P2
Archivo Editar Ver Buscar Terminal Ayuda

```
e399000@12-31-12-31:~/Downloads/EDAT-master/P2$ ./dvdreq recommend 1
film_id = 891
film title = Timberland Sky
category = Classics

film_id = 358
film title = Gilmore Boiled
category = Classics

film_id = 951
film title = Voyage Legally
category = Classics
```

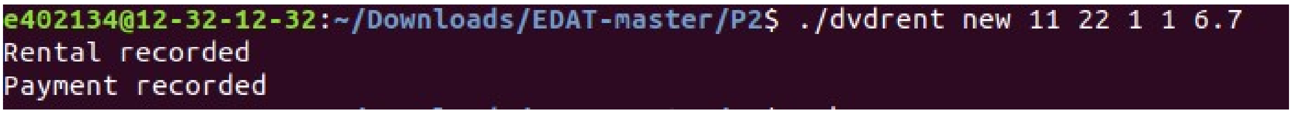
DVDRENT

dvdrent new <customer Id> <film id> <staff id> <store id> <amount>

```
SELECT customer_id
FROM   customer
WHERE  customer_id = InputCustomerId

SELECT staff_id
FROM   staff
WHERE  staff_id = InputStaffId
      AND store_id = InputStoreId

SELECT inventory_id
FROM   inventory
WHERE  film_id = InputFilmId
      AND store_id = InputStoreId
      AND inventory_id NOT IN(SELECT rental.inventory_id
                          FROM   rental
                                inner join inventory
                                       ON rental.inventory_id =
                                          inventory.inventory_id
                          WHERE  film_id = InputFilmId
                                AND rental.return_date IS NULL)

INSERT INTO rental
VALUES      ((SELECT Max(rental_id) + 1
             FROM   rental),
           Now(),
           %d,
           %ld,
           NULL,
           %ld,
           Now())

INSERT INTO payment
VALUES      ((SELECT Max(payment_id) + 1
             FROM   payment),
           %ld,
           %ld,
           (SELECT Max(rental_id)
             FROM   rental),
           %ld,
           Now())
```

First we check that the arguments are correct, thus checking that the customer id exists, and that the staff_id and store_id exists and the staff works in that store. Then we check if there is any item in the inventory that is available to be rented with the film_id provided and store_id. After checking everything we insert the new rental in the rental table, getting the rental_id by selecting the maximum one and adding one to it, then we do the same to add the payment associated to the rental just recorded, to do that we select the max rental_id from the rental table, that should be the one that we have just added.

A sample run of the program is:



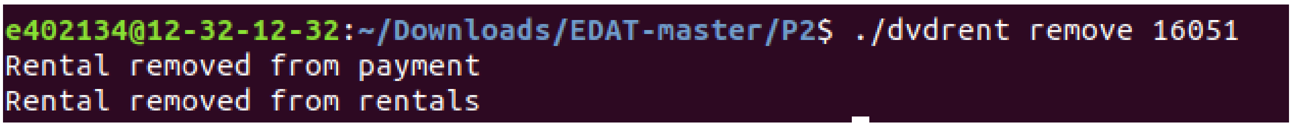
```
e402134@12-32-12-32:~/Downloads/EDAT-master/P2$ ./dvdrent new 11 22 1 1 6.7
Rental recorded
Payment recorded
```

dvdrent remove <rent Id>

```
SELECT rental_id
FROM   rental
WHERE  rental_id = %ld

DELETE FROM payment
WHERE  rental_id = %ld

DELETE FROM rental
WHERE  rental_id = %ld
```

To delete a rental from the database, first we need to check that the rental id (%ld) that we have introduced is in the database.
Then we delete the rental from the rental table by its rental_id and we do the same with the payment associated to that rental_id.

A sample run of the program is:



```
e402134@12-32-12-32:~/Downloads/EDAT-master/P2$ ./dvdrent remove 16051
Rental removed from payment
Rental removed from rentals
```

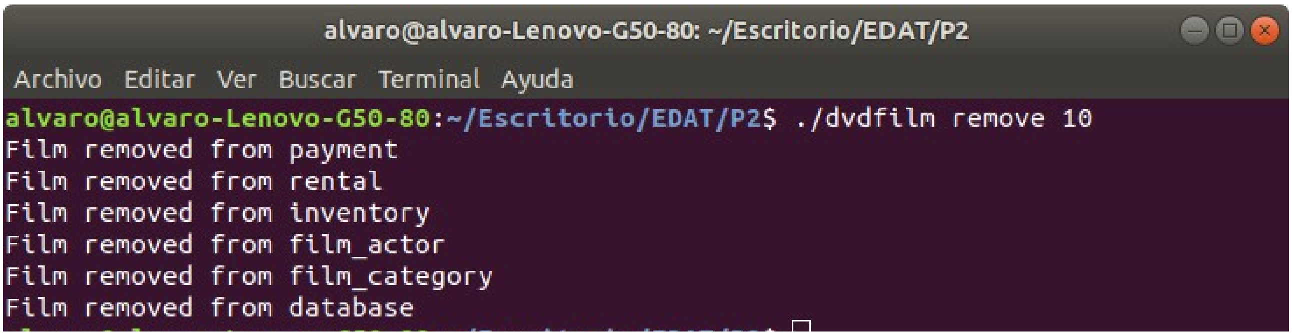
DVDFILM

dvdfilm remove <film id>

```
SELECT film_id
FROM    film
WHERE   film_id = %ld

DELETE FROM payment
USING  inventory,
       rental
WHERE  inventory.film_id =%ld
       AND rental.inventory_id = inventory.inventory_id
       AND rental.rental_id = payment.rental_id;

DELETE FROM rental
USING  inventory
WHERE  inventory.film_id = %ld
       AND rental.inventory_id = inventory.inventory_id;

DELETE FROM inventory
WHERE  film_id = %ld;

DELETE FROM film_actor
WHERE  film_id = %ld;

DELETE FROM film_category
WHERE  film_id = %ld;

DELETE FROM film
WHERE  film_id = %ld;
```

To delete a film from the database, first we need to check that the film id (%ld) that we have introduced is in the database. Then as the deletion is not on cascade in the source code, we will have to delete by hand everything.
The first query deletes the payment that matches to the rentals of the film.
The second query deletes all the rentals of that film.
The third one deletes the inventories that have the requested film.
The next ones deletes each relation made between the film and the actors and the categories.
The last query deletes finally the film from the film table.

A sample run of the program is:


alvaro@alvaro-Lenovo-G50-80: ~/Escritorio/EDAT/P2
Archivo Editar Ver Buscar Terminal Ayuda

```
alvaro@alvaro-Lenovo-G50-80:~/Escritorio/EDAT/P2$ ./dvdfilm remove 10
Film removed from payment
Film removed from rental
Film removed from inventory
Film removed from film_actor
Film removed from film_category
Film removed from database
```