

Usage Example

These are a few usage examples of how to perform common operations using the Rent-A-Film database.

- Rent a DVD
- Return a DVD
- Find Overdue DVDs

- Rent a DVD

To rent a DVD, first confirm that the given inventory item is in stock, and then insert a row into the `rental` table. After the `rental` table is created, insert a row into the `payment` table. Depending on business rules, you may also need to check whether the customer has an outstanding balance before processing the rental.

```
mysql> SELECT inventory_in_stock(10);
+-----+
| inventory_in_stock(10) |
+-----+
| 1 |
+-----+
1 row in set (0.01 sec)

mysql> INSERT INTO rental(rental_date, inventory_id, customer_id, staff_id)
VALUES(NOW(), 10, 3, 1);
Query OK, 1 row affected (0.00 sec)

mysql> SET @rentID = LAST_INSERT_ID(),
@balance = get_customer_balance(3, NOW());
Query OK, 0 rows affected (0.14 sec)

mysql> SELECT @rentID, @balance;
+-----+-----+
| @rentID | @balance |
+-----+-----+
| 16050 | 4.99 |
+-----+-----+
1 row in set (0.00 sec)

mysql> INSERT INTO payment (customer_id, staff_id, rental_id, amount,
payment_date)
VALUES(3, 1, @rentID, @balance, NOW());
Query OK, 1 row affected (0.00 sec)
```

- **Return a DVD**

To return a DVD, update the `rental` table and set the return date. To do this, first identify the `rental_id` to update based on the `inventory_id` of the item being returned. Depending on the situation, it may be necessary to check the customer balance and perhaps process a payment for overdue fees by inserting a row into the `payment` table.

```
mysql> SELECT rental_id
        FROM rental
        WHERE inventory_id = 10
        AND customer_id = 3
        AND return_date IS NULL
        INTO @rentID;

Query OK, 1 row affected (0.01 sec)
```

```
mysql> SELECT @rentID;
+-----+
| @rentID |
+-----+
|    16050 |
+-----+
1 row in set (0.00 sec)
```

```
mysql> UPDATE rental
      SET return_date = NOW()
      WHERE rental_id = @rentID;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> SELECT get_customer_balance(3, NOW());  
+-----+  
| get_customer_balance(3, NOW()) |  
+-----+  
|                                0.00 |  
+-----+  
  
1 row in set (0.13 sec)
```

- Find Overdue DVDs

Many DVD stores produce a daily list of overdue rentals so that customers can be contacted and asked to return their overdue DVDs.

```
mysql> SELECT CONCAT(customer.last_name, ', ', customer.first_name) AS
customer,
        address.phone, film.title
        FROM rental INNER JOIN customer ON rental.customer_id =
customer.customer_id
        INNER JOIN address ON customer.address_id = address.address_id
        INNER JOIN inventory ON rental.inventory_id =
inventory.inventory_id
        INNER JOIN film ON inventory.film_id = film.film_id
        WHERE rental.return_date IS NULL
        AND rental_date + INTERVAL film.rental_duration DAY <
CURRENT_DATE()
        ORDER BY title
        LIMIT 5;
```

```
+-----+-----+-----+
| customer      | phone      | title      |
+-----+-----+-----+
| OLVERA, DWAYNE | 62127829280 | ACADEMY DINOSAUR |
| HUEY, BRANDON  | 99883471275 | ACE GOLDFINGER    |
| OWENS, CARMEN  | 272234298332 | AFFAIR PREJUDICE  |
| HANNON, SETH   | 864392582257 | AFRICAN EGG       |
| COLE, TRACY    | 371490777743 | ALI FOREVER       |
+-----+-----+-----+
5 rows in set (0.10 sec)
```

To create such a list, search the `rental` table for films with a return date that is `NULL` and where the rental date is further in the past than the rental duration specified in the `film` table. If so, the film is overdue and we should produce the name of the film along with the customer name and phone number.