1. What operations do the following functions perform: film\_in\_stock, film\_not\_in\_stock, inventory\_in\_stock, get\_customer\_balance, inventory\_held\_by\_customer, rewards\_report, last\_day?

**film\_in\_stock** - Returns a list of inventory IDs for a given film that are currently available (in stock) at a specific store. It filters by film\_id, store\_id, and checks availability using inventory\_in\_stock(). Input is (film\_id, store\_id) and returns integer.

**film\_not\_in\_stock** - Opposite of film\_in\_stock(). Returns inventory IDs for a given film at a store that are currently not in stock. Input is (film\_id, store\_id) and returns integer.

**inventory\_in\_stock** - Checks if an inventory item is in stock. It returns TRUE if the item has never been rented, or if all past rentals have a return\_date. Otherwise it returns FALSE. Input it (inventory\_id) and returns boolean.

**get\_customer\_balance** - Calculates the current balance for a customer up to a specific date. It sums up rental fees, late fees, replacement costs (if overdue), and subtracts previous payments. Input is (p\_customer\_id, p\_effective\_date) and returns a numeric value.

**inventory\_held\_by\_customer** - Checks who currently holds a specific inventory item (based on inventory\_id). Returns customer\_id if it is rented and not yet returned and returns set of inventory IDs.

**rewards\_report** - Returns customers who qualify for a reward based on:

- how many purchases they made in the last full month (min\_monthly\_purchases)

- how much they spent (min\_dollar\_amount\_purchased)

Uses a temporary table and dynamic SQL.

**last\_day** - returns the last day of the month for the given timestamp. If the month is December, it returns Dec 31st. Otherwise, it returns the day before the 1st of the next month. Data type is timestamp.

2. Why does ‘rewards\_report’ function return 0 rows? Correct and recreate the function, so that it's able to return rows properly.

Because it uses dynamic SQL to build a query string using the tmpsql variable, but the query filters using a subquery that always returns no results because of incorrect date comparison or group aggregation logic. It also might be possible that CURENT\_DATE is beyond the date range of available payment data.

3. Is there any function that can potentially be removed from the dvd\_rental codebase? If so, which one and why?

\_group\_concat (an internal helper for group\_concat) could be removed if group\_concat is replaced with standard PostgreSQL aggregation like string\_agg, which is built-in and more efficient.  
This function exists mostly for MySQL compatibility or legacy reasons.

4. The ‘get\_customer\_balance’ function describes the business requirements for calculating the client balance. Unfortunately, not all of them are implemented in this function. Try to change function using the requirements from the comments.

The function mentions:

* Rental rate
* Payment made
* Late fees
* Lost films (not implemented)
* Outstanding balance

So we need to enhance it by:

* Calculating rental fees
* Subtracting payments
* Adding late fees and lost item cost (use replacement\_cost)

5. How do ‘group\_concat’ and ‘\_group\_concat’ functions work? (database creation script might help) Where are they used?

group\_concat - A user-defined aggregate function that concatenates multiple values into a single string with a separator (like comma).

\_group\_concat - The internal state transition function used by group\_concat. It appends each value during aggregation.

They are used in views or reports where multiple values (e.g., actors per film) need to be shown as a single string.

6. What does ‘last\_updated’ function do? Where is it used?

last\_updated() is a trigger function that automatically updates the last\_update column when a row is changed.

So basically, you don’t need to manually update that column — it does it for you behind the scenes.

It’s used with triggers (like BEFORE UPDATE) and you can find it under Trigger Functions in the database.

It’s just a simple way to keep track of when something was last modified.

7. What is tmpSQL variable for in ‘rewards\_report’ function? Can this function be recreated without EXECUTE statement and dynamic SQL? Why?

tmpSQL is a TEXT variable that stores a dynamically constructed SQL string.

It's used to build the query for inserting matching customers into the temporary table.

Yes, the function be rewritten using static SQL and JOINs, but dynamic SQL is used here likely for flexibility.