MIS 381N

HW4– DDL Script Assignment

Team Members

Name	**UT EID**
Bhavana Reddy	bc35833
Safiuddin Mohammed	sm76833
Brandt Green	bwg537
Ramya Madhuri Desineedi	rd32895
Suchit Das	sd38448
Lucas Fernandez	lf23234

Executive Summary

1. We are running a query on the customer table pulling the count of customers, the minimum stay credits earned, and the maximum stay credits earned. A column alias was performed to change the names of the columns to count_of_customers, min_credits and max_credits respectively.
2. A query was run to perform a join between the customer and reservation table. The join was performed on customer_id and the number of reservations and earliest check in date for each customer was pulled from the query.
3. A query was run on the customer table pulling out city, state and average of the stay credits earned. The results were grouped by state and city and then ordered by state in ascending, and average credits earned in descending.
4. A query was run to perform a join between customer, reservation, reservation_details and room. The customer_id, last_name, room_number and count of reservation_id was pulled for location_id = 1. The results were grouped by customer_id, last_name and room_number and ordered by customer_id in ascending and count of reservation_id in descending.
5. The query from the previous question was modified to only show reservations with status = C. The results were further filtered to only show the count of reservation_id greater than 2.
6. A join was performed customer, reservation and location tables. The location_name, check_in_date and sum of number_of_guests. The check_in_dates in the past were filtered out by using check_in_date > SYSDATE and the ROLLUP operator was used to include a row that gives the subtotal of location_name, check_in_date.
7. A join was performed between the location_features_linking, location and features table. The feature_name and count of the location_id was pulled from this. The result was grouped by feature_name and all features with count of location_id < 3 were filtered out.
8. A subquery was used to pull all the distinct customer_id from reservations table. The results of this sub query were used in the main query to return all the customer_id that are NOT in the reservation table and only in the customer table.
9. A subquery was used to pull the average of stay_credits_earned from the customer table. This returned a value of 13.86667. The result of this subquery was used in the main query to pull the first_name, last_name, email, phone and stay_credits_earned from customer where the stay_credits_earned was greater than the average value of 13.86667.
10. A subquery was run on the customer table pulling city, state, the sum of stay_credits_earned, and the sum of stay_credits_used. The results were grouped and ordered by state and city. The results were used in the main query to pull the city, state and the total credits remaining for each city in a state.
11. A join was performed between reservation, reservation_details and room. The room_id and count of reservation were pulled from this and grouped by room_id. The room_id with reservation count > 5 were filtered out. The results of this subquery were used in the main query to return the confirmation_nbr, date_created check_in_date, status, and room_id and all the reservations with status = C were filtered out.
12. A join was performed between customer and reservation table to return all the customers with only 1 completed reservation. The result of this sub query was used in the main query using an inline view to pull only the customers using card_type = MSTR.