Introduction to Data Management: Assignment 5

Group Members:

Yash Warty	yvw57		Sahil Natu	sn25936
Shivarjun Sarkar	ss92758		Vivek Mehendiratta	vm24395
Mahika Bansal	mb62835		Ankita Kundra	ak44675

database, as required for the analysis.

Tasks & Methodology:

1. Set the server output on, then used a PL/SQL block to obtain a count of reservations for a given Customer ID and then the IF-ELSE statements to check if there were more than 15 reservations. Then we deleted a particular reservation ID, ran the code again for our new results and finally rolled back the deletion to obtain the original code.

2. Updated the previous PL/SQL block to take in user input for Customer ID and then pull and display the count of reservations for the ID while also displaying if there were more or less than 15 reservations made by the customer, similar to the procedure in (1).

3. Wrote a script using PL/SQL code blocks to insert a new customer into the customer table. Utilizing the customer_id_seq we assign a customer_id. Then, by only filling in the required values we use column list to complete the insert statement and display a message with the result.

4. Used PL/SQL and a bulk collect to capture a list of available features that begin with the letter P. Sorted the rows in the result set by feature name. Then displayed a string variable for each feature number and its associated feature.

5. Made use of a cursor to capture location, city and feature names. Then sorted the output and presented it by location, city and feature name. Also made the code dynamic so it prompts the user for a city and then updates the output list to the location name, city and features names for the city.

6. Created a named procedure called insert_customer that inserts a new customer by passing in customer ID, first name, last name, email, phone, address_line_1, city, state, and zip. Continued using the customer_id_seq to generate new customer ID's and rolled back the transaction when others occurred.

7. Created a function called hold_count that returns the total number of rooms that a customer has reserved with customer_id as input.