# MIS 381N
# HW3– 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. A query was run on the customer payment table pulling the columns cardholder_first_name, cardholder_last_name, expiration_date, and card_number sorting by earliest expiration dates.
2. A query was run on the customer table pulling the first_name and last_name columns and combining them with a space in between. This new column was renamed as customer_full_name and sorted in alphabetical order. A sub query was run to only select the customers' first names that began with an A, B, or a C.
3. A query was run on the reservation table pulling the columns customer_id, confirmation_nbr, date_created, check_in_date, and number of guests where the reservation is upcoming and the check in date is between today/end of year sorted by earliest check in date.
4. The same query was run as in the previous question except the where condition utilized the between function.
5. A query was run on the reservation table pulling the first 10 rows for customer_id, location_id, check_out_date, and check_in_date with a completed reservation. The difference of check out/in dates was taken as length_of_stay sorted by longest length of stay and largest customer_id.
6. A query was run on the customer table pulling the columns first_name, last_name, email, stay_credits_earned, and stay_credits_used. The difference of stay_credits_earned and stay_credits_used was taken as credits_available and was sorted by least credits available.
7. A query was run on the customer_payment table pulling cardholder_first_name, cardholder_mid_name, and cardholder_last_name where the customer has a middle name ordered in ascending order in columns 2 and 3. A potential problem with this query is if the order of the columns changes because the sorting is hard coded on columns 2 & 3 instead of column names.
8. A dual table was displayed with columns today_unformatted, today_formatted, credits_earned, stays_earned, redeemable_stays, and next_stay_to_earn. A first example row was created.
9. A query was run on the reservation table pulling the columns customer_id, location_id, check_out_date, and check_in_date with a completed reservation. The difference of check out/in dates was taken as length_of_stay sorted by longest length of stay and largest customer_id. The table consists of only the first 20 rows of the query.
10. A join was performed on the customer and reservation tables on the customer_id column in both tables where the reservation was completed. The new table has columns first_name, last_name, confirmation_nbr, date_created, check_in_date, check_out_date sorted in descending order by customer_id and check out date.
11. A join was performed between tables customer, reservation, reservation_details, and room where reservation status was upcoming, and more than 40 credits were earned. The join was on the primary key/foreign key connections between the tables.
12. A left join was performed on the customer and reservation tables on the customer_id column in both tables where the reservation customer_id was blank (null). Columns first_name, last_name, confirmation_nbr, date_created, check_in_date, and check_out_date were selected.
13. A new table was created from the union of three queries of the customer table. Each of the queries was selected based on a condition of the number of credits earned. A new column based on the level of credit earned was created. From customer table, the columns first_name, last_name, email, stay_credit_earned were pulled. The table was sorted alphabetically by last name and level of membership.