

# Submission 4: Table Data visualization & App functionalities

## Data visualization

Table 1(customers)

|   | customer_id | first_name | last_name | email                  | phone        |
|---|-------------|------------|-----------|------------------------|--------------|
| ▶ | 1           | John       | Doe       | john.doe@example.com   | 555-123-4567 |
|   | 2           | Jane       | Smith     | jane.smith@example.com | 555-987-6543 |
|   | 3           | Alice      | Johnson   | alice.j@example.com    | 555-222-3333 |
|   | 4           | Bob        | Brown     | bob.b@example.com      | 555-444-5555 |
| * | 5           | Emily      | Clark     | emily.c@example.com    | 555-777-8888 |
| * | NULL        | NULL       | NULL      | NULL                   | NULL         |

Table 2(Cars)

|   | car_id | car_type    | car_color | car_price |
|---|--------|-------------|-----------|-----------|
| ▶ | 1      | Compact     | Red       | 25.00     |
|   | 2      | Truck       | Blue      | 30.00     |
|   | 3      | Sedan       | Black     | 20.00     |
|   | 4      | SUV         | White     | 35.00     |
| * | 5      | Convertible | Yellow    | 40.00     |
| * | NULL   | NULL        | NULL      | NULL      |

Table3 (Rentals)

|   | rental_id | customer_id | car_id | rental_start_date | rental_end_date |
|---|-----------|-------------|--------|-------------------|-----------------|
| ▶ | 1         | 1           | 1      | 2023-04-01        | 2023-04-07      |
|   | 2         | 2           | 2      | 2023-04-02        | 2023-04-04      |
|   | 3         | 3           | 3      | 2023-04-05        | 2023-04-06      |
|   | 4         | 4           | 4      | 2023-04-06        | 2023-04-08      |
|   | 5         | 5           | 5      | 2023-04-07        | 2023-04-09      |
|   | 6         | 1           | 2      | 2023-05-01        | 2023-05-03      |
|   | 7         | 3           | 1      | 2023-05-04        | 2023-05-06      |
| * | NULL      | NULL        | NULL   | NULL              | NULL            |

Table 4(Invoice)

|   | invoice_id | rental_id | invoice_amount |
|---|------------|-----------|----------------|
| ▶ | 1          | 1         | 50.00          |
|   | 2          | 2         | 60.00          |
|   | 3          | 3         | 20.00          |
|   | 4          | 4         | 70.00          |
|   | 5          | 5         | 80.00          |
|   | 6          | 6         | 60.00          |
|   | 7          | 7         | 50.00          |
|   | NULL       | NULL      | NULL           |

Table5 (payments)

|   | payment_id | invoice_id | payment_date | amount | payment_method |
|---|------------|------------|--------------|--------|----------------|
| ▶ | 1          | 1          | 2023-04-04   | 150.00 | card           |
|   | 2          | 1          | 2023-04-05   | 20.00  | cash           |
|   | 3          | 2          | 2023-04-06   | 180.00 | card           |
|   | 4          | 2          | 2023-04-07   | 15.00  | cash           |
|   | 5          | 3          | 2023-04-13   | 90.00  | card           |
|   | 6          | 3          | 2023-04-14   | 10.00  | card           |
|   | 7          | 4          | 2023-04-19   | 200.00 | card           |
|   | 8          | 4          | 2023-04-20   | 25.00  | cash           |
|   | 9          | 5          | 2023-05-05   | 210.00 | card           |
|   | 10         | 5          | 2023-05-06   | 30.00  | cash           |
| * | NULL       | NULL       | NULL         | NULL   | NULL           |

## App functionalities

D:\> Karthik DBMS > Project\_1 > App.py > ...

```

1081
1082     for label in ["All Customers", "All Cars", "All Rentals", "All Invoices", "All Payments"]:
1083         ttk.Button(
1084             frame1,
1085             text=label,
1086             width=30,
1087             command=lambda q=label: run_report(q)
1088         ).pack(side="left", fill="x", expand=True)
1089
1090     for label in [
1091         "Total Earnings",
1092         "Total Rentals",
1093         "Most Rented Car",
1094         "Invoice Payment"
1095     ]:
1096         ttk.Button(
1097             frame2,
1098             text=label,
1099             width=30,
1100             command=lambda q=label: run_report(q)
1101         ).pack(side="left", fill="x", expand=True)
1102
1103     # =====
1104     # Main Loop
1105     # =====
1106
1107     app.mainloop()
1108

```

D:\> Karthik DBMS > Project\_1 > App.py > ...

```

1081
1082     for label in ["All Customers", "All Cars", "All Rentals", "All Invoices", "All Payments"]:
1083         ttk.Button(
1084             frame1,
1085             text=label,
1086             width=30,
1087             command=lambda q=label: run_report(q)
1088         ).pack(side="left", fill="x", expand=True)
1089
1090     for label in [
1091         "Total Earnings",
1092         "Total Rentals",
1093         "Most Rented Car",
1094         "Invoice Payment"
1095     ]:
1096         ttk.Button(
1097             frame2,
1098             text=label,
1099             width=30,
1100             command=lambda q=label: run_report(q)
1101         ).pack(side="left", fill="x", expand=True)
1102
1103     # =====
1104     # Main Loop
1105     # =====
1106
1107     app.mainloop()
1108

```

The screenshot shows a Python application window titled "Car Rental Management". The window has a tab bar with "Customers", "Inventory", "Rentals", and "Reports". The "Reports" tab is selected, displaying a table with columns: rental\_id, first\_name, last\_name, car\_type, and rental\_start\_date. The data in the table is as follows:

| rental_id | first_name | last_name | car_type    | rental_start_date |
|-----------|------------|-----------|-------------|-------------------|
| 1         | John       | Doe       | Compact     | 2023-04-01        |
| 6         | Jane       | Smith     | Truck       | 2023-04-01        |
| 2         | Alice      | Johnson   | Sedan       | 2023-04-05        |
| 7         | Bob        | Johnson   | Compact     | 2023-04-04        |
| 4         | Emily      | Clark     | SUV         | 2023-04-06        |
| 5         |            |           | Convertible | 2023-04-07        |

Below the table, there is a section titled "View & Modify Rentals" with a "Rental Mode" dropdown set to "Existing Rentals".

The screenshot shows the same Python application window, but the "Reports" tab is not selected. Instead, the "Customers" tab is selected, displaying a "List All Tables" section with four buttons: "All Customers", "All Cars", "All Rentals", and "All Payments". Below this is a "Predefined Queries" section with four buttons: "Total Earnings per Car", "Total Rentals per Customer", "Most Rented Cars", and "Invoice Payment Status".

The screenshot shows the results of the "Most Rented Cars" predefined query. The table has columns: car\_id, car\_type, and NumberOfRentals. The data is as follows:

| car_id | car_type    | NumberOfRentals |
|--------|-------------|-----------------|
| 1      | Compact     | 2               |
| 2      | Truck       | 2               |
| 3      | Sedan       | 1               |
| 4      | SUV         | 1               |
| 5      | Convertible | 1               |

The screenshot shows the results of the "Total Earnings per Car" predefined query. The table has columns: car\_id, car\_type, and TotalEarnings. The data is as follows:

| car_id | car_type    | TotalEarnings |
|--------|-------------|---------------|
| 1      | Compact     | 100.00        |
| 2      | Truck       | 120.00        |
| 3      | Sedan       | 20.00         |
| 4      | SUV         | 70.00         |
| 5      | Convertible | 80.00         |