

Queries

Benny Chen, Disha Verma, Joshua Lee, Alina Baby

April 25, 2023

Query 1

The first query we made was to get a general overview of incidents from a certain customer from one destination to another (by zip code). This query shows the date, name, the description of what happened and the zip codes of the route. This would help with a consise report on the incident that happened from that route.

```
01 | SELECT r.Report_Date, c.NAME, i.Description,
02 | s.ZipCode AS Starting_location_ZipCode,
03 | d.ZipCode AS Destination_ZipCode
04 | FROM CUSTOMER c
05 | INNER JOIN HOUSEHOLD_GOODS h ON c.CustomerID = h.CustomerID
    AND c.Starting_location_NO_ID = h.
    Starting_location_NO_ID
06 | INNER JOIN Report r ON h.HouseholdGood_ID = r.
    HouseholdGood_ID
07 | INNER JOIN INCIDENT i ON r.Incident_ID = i.Incident_ID
08 | INNER JOIN `STARTING LOCATION` s ON h.
    Starting_location_NO_ID = s.Starting_location_NO_ID
09 | INNER JOIN DESTINATION d ON r.Destination_NO_ID = d.
    Destination_NO_ID
10 | WHERE c.NAME = 'Glenna Jarvis'
```

This outputs the following table:

	Date of Report	Customer Name	Description	Starting ZipCode	Destination ZipCode
1	2021-02-25	Stacey Johnson	Furniture parts were lost during the move between truck and plane.	86006	44084

Query 2

The second query we aggregate the data to find those who have made unsuccessful payments. In this table we can see the name, policy type, customer ID, phone number, payment amount, and the number of days since due date. This would help with customer service as we can see who has made unsuccessful payments and contact them to see if they would like to make a repayment or cancel their policy. This is also ordered by the number of days since the due date so we can see who has been delinquent the longest and contact them first.

```

01 | SELECT c.NAME AS 'Customer Name',
02 | po.Policy_Type AS 'Policy',
03 | c.CUSTOMERID AS 'Customer ID',
04 | c.PHONE_NUMBER AS 'Phone Number',
05 | p.Payment_Amount AS 'Payment Due',
06 | DATEDIFF(NOW(), p.Payment_Date) AS 'Days Delinquent'
07 | FROM CUSTOMER c
08 | INNER JOIN PAYMENT p ON c.CUSTOMERID = p.CustomerID
09 | INNER JOIN POLICY po ON c.CUSTOMERID = po.CustomerID
10 | WHERE p.Success_Status = 'Unsuccessful'
11 | ORDER BY DATEDIFF(NOW(), p.Payment_Date) DESC;

```

This outputs the following table:

	Customer Name	Policy	Customer ID	Phone Number	Payment Due	Days Delinquent
1	McKenzie Estrada	Basic Coverage	2,171,675,179	1-723-872-3355	17,395	1,068
2	Levi Joyner	Basic Coverage	8,494,448,728	1-454-697-1335	27,865	967
3	Mohammad Miles	Full-Value Coverage	1,765,735,042	1-738-178-2681	2,374	948
4	John Griffin	Basic Coverage	8,973,484,610	(781) 268-9331	33,848	607
5	Kennan Lewis	Basic Coverage	9,502,402,374	1-706-778-8296	4,101	540

Query 3






The third query we made was to filter the data by those who have been constantly using the service. This would show a company that this customer is a high risk customer. The filter narrows down those who have multiple reports and have only been using the service for a year. This would output all customers that fit this along with their name, policy, policy number, the number of reports, and the total number of goods. this would help catch fraud and help the company decide if they want to continue to do business with this customer.

```

01 | SELECT C.NAME, P.Policy_Number, COUNT(R.Report_ID) AS
      Num_Reports, SUM(R.Quantity_of_Goods) AS Total_Goods
02 | FROM CUSTOMER C
03 | JOIN POLICY P ON C.CUSTOMERID = P.CustomerID
04 | JOIN HOUSEHOLD_GOODS HG ON C.CUSTOMERID = HG.CustomerID
05 | JOIN Report R ON HG.HouseholdGood_ID = R.HouseholdGood_ID
06 | WHERE P.Policy_End_Date <= DATE_ADD(P.Policy_Start_Date,
      INTERVAL 1 YEAR)
07 | GROUP BY C.NAME, P.Policy_Number
08 | HAVING COUNT(R.Report_ID) > 1
09 | ORDER BY Num_Reports DESC;

```

This outputs the following table:

	 CUSTOMERID	 RBC Type/Category	 RBC Incident_Type	 RBC Severity	 123 Deductible
1	2,577,716,392	Artwork	Vandalism	Moderate	44,486
2	6,068,690,772	Furniture	Natural Disaster	High	30,369
3	2,171,675,179	Appliance	Damage	Low	35,511
4	8,973,484,610	Appliance	Vandalism	High	40,011
5	7,661,208,705	Furniture	Accident	High	12,930
6	9,502,402,374	Furniture	Accident	Moderate	19,473
7	8,494,448,728	Furniture	Natural Disaster	High	48,538
8	6,882,156,528	Furniture	Damage	High	14,630
9	7,551,557,454	Appliance	Damage	Moderate	20,288
10	1,929,429,564	Furniture	Accident	High	6,153
11	1,336,062,432	Artwork	Loss	Moderate	44,167
12	6,394,929,011	Furniture	Loss	Low	30,772

Query 4

In the fourth query we aggregate the data to find the departure and arrival times of the move along with how far it was. This would help with the company to see how long it takes to move a customer from one location to another depending on the distance and if that factor affects incidents that happen during the move. This query shows the CustomerID, name, the departure time, the arrival time, and the distance of the move.

```

01 | SELECT
02 | c.CUSTOMERID ,
03 | c.NAME ,
04 | s.Depart_Date ,
05 | d.Arrival_Date ,
06 | de.Travel_Distance_in_Miles
07 | FROM
08 | CUSTOMER c
09 | JOIN HOUSEHOLD_GOODS h ON c.CUSTOMERID = h.CustomerID
10 | JOIN `STARTING LOCATION` s ON h.Starting_Location_NO_ID = s
    .Starting_location_NO_ID
11 | JOIN Report r ON h.HouseholdGood_ID = r.HousholdGood_ID
12 | JOIN DESTINATION d ON r.Destination_NO_ID = d.
    Destination_NO_ID
13 | JOIN DESTINATION de ON r.Destination_NO_ID = de.
    Destination_NO_ID;

```

This outputs the following table:

	¹²⁴ CUSTOMERID	RBC NAME	¹²⁵ Depart_Date	¹²⁶ Arrival_Date	¹²³ Travel_Distance_in_Miles
1	2,577,716,392	Glenna Jarvis	2021-08-06	2021-08-06	851
2	2,646,972,860	Christen Warren	2021-04-04	2021-04-07	580
3	8,080,468,691	Wang Neal	2022-11-17	2022-11-19	37
4	2,076,726,177	Rose Ward	2022-01-15	2022-01-17	18
5	6,068,690,772	Brock Blanchard	2022-08-28	2022-08-30	688
6	2,171,675,179	McKenzie Estrada	2022-05-31	2022-06-02	137
7	8,973,484,610	John Griffin	2021-03-17	2021-03-19	242
8	7,661,208,705	Lucas Cotton	2022-08-26	2022-08-28	913
9	5,263,926,722	Melyssa Hayden	2021-12-14	2021-12-16	146
10	9,502,402,374	Kennan Lewis	2022-03-28	2022-03-30	748
11	8,494,448,728	Levi Joyner	2022-09-27	2022-09-29	54
12	6,882,156,528	Isaac Johns	2021-05-18	2021-05-20	34
13	8,959,170,226	Wesley Banks	2021-09-21	2021-09-24	734
14	7,100,141,024	Joan Herman	2021-03-14	2021-03-16	159
15	3,052,814,236	Alvin Giles	2022-04-08	2022-04-10	73
16	1,765,735,042	Mohammad Miles	2021-03-20	2021-03-23	264
17	7,551,557,454	Shoshana Shelton	2021-02-05	2021-02-07	41
18	1,929,429,564	Burke Allison	2022-05-25	2022-05-27	404
19	1,336,062,432	Jayne Pace	2022-12-20	2022-12-23	676
20	6,394,929,011	Stacey Johnson	2021-02-19	2021-02-21	210

Query 5

In this query we filter the data by the year, in this case 2021, and output the number of cases and percentages of the number of incidents that happened in the year. This would help with the company to see what types of incidents happen the most and shape the insurance policy to better cover those incidents. This query shows the type of the insured, the number of cases of it that year, and the percentage as a whole.

```

01 | SELECT H.`Type/Category` AS 'Type of Case',
02 | COUNT(H.`Type/Category`) AS 'Number of Cases',
03 | COUNT(H.`Type/Category`) * 100.0 / SUM(COUNT(H.`Type/
    Category`)) OVER() AS 'Percentage within Year'
04 | FROM HOUSEHOLD_GOODS H
05 | JOIN POLICY P ON H.CustomerID = P.CustomerID
06 | JOIN Report R ON H.HouseholdGood_ID = R.HouseholdGood_ID
07 | WHERE YEAR(R.Report_Date) = 2021
08 | GROUP BY H.`Type/Category`;
09 | ORDER BY

```

This outputs the following table:

	¹²⁷ Type of Case	¹²⁸ Number of Cases	¹²⁹ Percentage within Year
1	Appliance	3	30
2	Artwork	1	10
3	Furniture	6	60