

Assignment Two - Scraping Twitter

Project Name – Grocery Invoice Billing Management System

Member Name – Kalind Joshi

NUID – 002752274

I. FIVE USE CASES, THEIR SQL QUERIES AND RELATIONAL ALGEBRA:

1. Use Case:

Description: Which product has the highest sales in a timeframe and is it the most discounted one

Actor: Admin

Precondition: Product should have been sold more than once in the timeframe

Steps:

Actor action: Admin checks for products which are the most selling in a given timeframe

System Responses: System will check the best-selling item and compare it to the list of most discounted items to see if the top discounted product matches the best selling item

Post Condition: Both the products are retrieved and check against each other

Alternate Path: Most discounted product is not equal to most selling item

Error: No error possible

SQL for US1:

```
1 • SELECT
2     COUNT(DISTINCT invoiceid), prodid
3 FROM
4     invoice
5 WHERE
6     prodid = '1'
7     AND invoicedt BETWEEN '2020-09-01' AND '2020-09-31'
```

This will return the number of times prod ID '1' has appeared in different invoices for the month of September 2020

```

1 • SELECT
2     prodid, proddis
3 FROM
4     prod
5 ORDER BY proddis DESC;

```

When we compare the top result of these queries, and if they match then the answer of the above use case will be true or else false.

Relational Algebra for US1:

Code 1: π *COUNT* (\delta *invoiceid*), *prodid*

γ *COUNT* (\delta *invoiceid*)

σ *prodid* = "1" AND ("2020-09-01" <= *invoicedt* AND *invoicedt* <= "2020-09-31") *invoice*

Code 2: τ *proddis* ↓

π *prodid*, *proddis* *prod*

2. Use Case:

Description: Which employee has generated the most invoices in a month?

Actor: Admin

Precondition: There should be more than one employee working in the specified timeframe

Steps:

Actor action: Admin creates a view which joins two tables that is *invoice* and *emp*. Then, admin can check the emp with most invoice id by using count on the view created.

System Responses: System will create a view and check for emp with most invoice id against them.

Post Condition: View is created and emp of the month is checked subsequently.

Alternate Path: check for emp id without creating view and joining tables directly.

Error: View is not created or there are more than one emp with same number of invoice generated

SQL for US2:

Creating a view to check employee of the month:

```
1 • CREATE VIEW empofmonth AS
2     SELECT
3         emp.empid, emp.empname, invoice.invoiceid, invoice.invoicedt
4     FROM
5         emp,
6         invoice
7     WHERE
8         emp.empid = invoice.empid;
```

Then use the following SQL command on view to check employee with highest number of invoices generated for month of September:

```
1     SELECT
2         COUNT(empid), empname, invoice id
3     FROM
4         empofmonth
5     WHERE
6         invoicedt BETWEEN '2020-09-31' AND '2020-09-01'
```

Relational algebra for US2:

Code 1: $\pi_{emp . empid, emp . empname, invoice . invoiceid, invoice . invoicedt}$
 $\sigma_{emp . empid = invoice . empid} (emp \times invoice)$

Code 2: $\pi_{COUNT (empid), empname, invoiceid}$
 $\gamma_{COUNT (empid)}$
 $\sigma_{"2020-09-31" \leq invoicedt \text{ AND } invoicedt \leq "2020-09-01"} empofmonth$

3. Use Case:

Description: Is there profit or loss of a particular product in a given time?

Actor: Admin

Precondition: Profit or loss should be calculated by SP and MP

Steps:

Actor action: Admin's creates a view which will have the PL column and get it calculated by SP and MP

Then, after the view is created then the admin can group by product id to get PL of each product

System Responses: System will create a view and check for Profit or loss of each product by using product ID

Post Condition: View is created, and P/L of each product is checked subsequently

Alternate Path: None

Error: View is not created, or new column PL is not created to calculate profit loss

SQL for US3:

Create a view which joins the prod and invoice table to calculate profit/loss of the product for the month of September:

```
1  CREATE VIEW PL AS
2      SELECT
3          prod.prodname,
4          prod.prodmp,
5          prod.prodsp,
6          prod.prodid,
7          invoice.invoicedt,
8          prod.prodsp - prod.prodmp as PL
9      FROM
10         prod,
11         invoice
12     WHERE
13         prod.prodid = invoice.invoiceid
14         AND invoice.invoicedt BETWEEN '2020-09-31' AND '2020-09-01'
```

After view is created and profit column is also created as difference of market price and selling price, we can use select statement to get sum of prodid individually

```
1 • select sum(PL) from PL where prodid = "1" group by prodid, prodname;
2 • select sum(PL) from PL where prodid = "2" group by prodid, prodname;
3 • select sum(PL) from PL where prodid = "3" group by prodid, prodname;
```

Relational Algebra for US3:

Code 1:

$$\pi_{prod.prodname, prod.prodmp, prod.prodsp, prod.prodid, invoice.invoicedt, prod.prodsp - prod.prodmp} \rightarrow pl$$
$$\sigma_{prod.prodid = invoice.invoiceid \text{ AND } ("2020-09-31" \leq invoice.invoicedt \text{ AND } invoice.invoicedt \leq "2020-09-01")} (prod \times invoice)$$

Code 2:

$$\pi_{SUM(pl)}$$
$$\gamma_{prodid, prodname, SUM(pl)}$$
$$\sigma_{prodid = "1"} pl$$
$$\pi_{SUM(pl)}$$
$$\gamma_{prodid, prodname, SUM(pl)}$$
$$\sigma_{prodid = "2"} pl$$
$$\pi_{SUM(pl)}$$
$$\gamma_{prodid, prodname, SUM(pl)}$$
$$\sigma_{prodid = "3"} pl$$

4. Use Case:

Description: Which customers has the highest orders in a period?

Actor: Admin

Precondition: There should be more than one customer to order during given timeframe

Steps:

Actor action: Admin's creates a view which will be the join of customer and invoice table with invoice id as foreign key to join tables
Then, after the view is created then the admin can group by invoice total to get the total of each customer order.

System Responses: System will create a view and check for total of each customer by cust ID

Post Condition: View is created, and total amount of each customer is checked subsequently

Alternate Path: None

Error: View is not created

SQL for US4:

Creating a view which will join tables cust and invoice to calculate the total of all customers order for month of September

```
1 • CREATE VIEW custhigh AS
2     SELECT
3         cust.custid,
4         cust.custname,
5         invoice.invoicetotal,
6         invoice.invoicedt
7     FROM
8         cust,
9         invoice
10    WHERE
11        cust.custid = invoice.invoiceid
12        AND invoice.invoicedt BETWEEN '2020-09-31' AND '2020-09-01'
```

After view is created which will have desired columns, we can use select statement to get sum of all the orders of an individual customer to see which customer has highest total of order amount for the month September

```

1  SELECT
2      SUM(total)
3  FROM
4      custhigh
5  WHERE
6      custid = '1'
7  GROUP BY total;
8
9  • SELECT
10     SUM(total)
11  FROM
12     custhigh
13  WHERE
14     custid = '2'
15  GROUP BY total
16

```

Relational Algebra for US4:

Code 1: $\pi_{cust . custid, cust . custname, invoice . invoicetotal, invoice . invoicedt}$

$\sigma_{cust . custid = invoice . invoiceid \text{ AND } ("2020-09-31"$

$\leq invoice . invoicedt \text{ AND } invoice . invoicedt \leq "2020-09-01") (cust \times invoice)$

Code 2: $\pi_{SUM (total)}$

$\gamma_{total, SUM (total)}$

$\sigma_{custid = "1"} custhigh$

$\pi_{SUM (total)}$

$\gamma_{total, SUM (total)}$

$\sigma_{custid = "2"} custhigh$

5. Use Case:

Description: Is there profit or loss of a particular product in a given time?

Actor: Admin

Precondition: Profit or loss should be calculated by SP and MP

Steps:

Actor action: Admin's creates a view which will have the PL column and get it calculated by SP and MP

Then, after the view is created then the admin can group by product id to get PL of each product

System Responses: System will create a view and check for Profit or loss of each product by using product ID

Post Condition: View is created, and P/L of each product is checked subsequently

Alternate Path: None

Error: View is not created, or new column PL is not created to calculate profit loss

SQL for US5:

Creating a view which will select columns like product name, id, and price with discount. Price after discount will be a calculated attribute derived from subtracting price from discount for month of September.


```

1 • CREATE VIEW discount AS
2     SELECT
3         prodid, prodname, prodwodis, proddis,
4         prodprice - proddis as prodwithdis
5     FROM
6         invoice
7     WHERE
8         invoice.invoicedt BETWEEN '2020-09-31' AND '2020-09-01'

```

After creating a view, admin can select sum of price with and without discount of a particular product by grouping it by product ID and name

```

1     SELECT
2         SUM(pricewithdis), SUM(pricewodis)
3     FROM
4         discount
5     WHERE
6         prodid = '1'
7     GROUP BY prodid, prodname;
8
9 • SELECT
10    SUM(pricewithdis), SUM(pricewodis)
11    FROM
12        discount
13    WHERE
14        prodid = '2'
15    GROUP BY prodid, prodname;

```

Relational Algebra for US5:

Code 1: $\pi_{prodid, prodname, prodwodis, prodprice - proddis} \rightarrow prodwithdis$

$\sigma_{"2020-09-31" \leq invoice.invoicedt \text{ AND } invoice.invoicedt \leq "2020-09-01"} invoice$

Code 2: π SUM (pricewithdis), SUM (pricewodis)
 γ $prodid$, $prodname$, SUM (pricewithdis), SUM (pricewodis)
 $\sigma_{prodid = "1"} discount$

π SUM (pricewithdis), SUM (pricewodis)
 γ $prodid$, $prodname$, SUM (pricewithdis), SUM (pricewodis)
 $\sigma_{prodid = "2"} discount$

II. SQL QUERIES FROM TWITTER TABLE (IMPORTED)

I made a new table called twitter and added a row twitter table which has attributes custid, price, item and date.

Here are the queries for the same and their output:

1. To find the date of creation

```
1 • use twitter;  
2 • select time from twittertable;  
3
```

[illegible]

2. Which user posted the tweet:

- 1 • `use twitter;`
- 2 • `select custid from twittertable;`
- 3

0%	33:2
result Grid	Filter Rows: Search
custid	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	

3. What tweets have this user posted in the past 24 hours?

```
1 • use twitter;  
2 • select * from twittertable;  
3
```

0%

1:3

result Grid

Filter Rows:

Q

Search

Ed

	custid	item	price	time
▶ 1		grocery item: milk	4	2022-11-13 00:00:00
2		grocery item: chocolate	3	2022-11-13 00:00:00
3		grocery item: eggs	9	2022-11-13 00:00:00
4		grocery item: bread	5	2022-11-13 00:00:00
5		grocery item: tomatoes	2	2022-11-13 00:00:00
6		grocery item: bananas	1	2022-11-13 00:00:00
7		grocery item: lays	8	2022-11-13 00:00:00
8		grocery item: salsa	10	2022-11-13 00:00:00
9		grocery item: nachos	5	2022-11-13 00:00:00
10		grocery item: apples	11	2022-11-13 00:00:00
11		grocery item: coffee	1	2022-11-13 00:00:00
12		grocery item: sugar	5	2022-11-13 00:00:00
13		grocery item: peach	8	2022-11-13 00:00:00
14		grocery item: pepsi	12	2022-11-13 00:00:00
15		grocery item: carrots	4	2022-11-13 00:00:00

III. ERD DIAGRAM OF THE PROJECT:

