INF1343: Data Modeling and Database Design

Assignment #2: RA, SQL and Accessing a Database

Submitted by:

Assel Kushkeyeva, Zhaotian Li, and Shabrina J. Mardevi

Part I. The .sql schema and data files containing the SQL statements to create and populate the database.

Submitted as two separate files

Part II: The ten queries translated into Relational Algebra (where possible) and SQL

1. What are the total sales made of Americano?

```
SELECT SUM(beverage_total_price)

FROM product INNER JOIN beverage_sold_in

ON product.sku = beverage_sold_in.sku_beverage_sold_in

WHERE product.name = 'americano';
```

```sql
SELECT SUM(beverage_total_price)
FROM product INNER JOIN beverage_sold_in
ON product.sku = beverage_sold_in.sku_beverage_sold_in
WHERE product.name = 'americano';
```

100% 34:4

Result Grid Filter Rows: Search Export:

SUM(beverage_total_pri...
21.400000000000002

2. Which Tim Horton's customer accounts are using debits?

$\pi_{account\_number}(\sigma_{payment\_info = \text{'Debit'}}(customer))$

SELECT account_number FROM customer

WHERE payment_info = 'Debit';

```sql
SELECT account_number FROM customer
WHERE payment_info = 'Debit';
```

100% 30:2

Result Grid Filter Rows: Search Edit: Export/Import:

account_number
1001
1005
1006
NULL

3. What orders consist of one beverage and at least one food item?

$\pi_{order\_no}(\sigma_{food\_quantity>=1}(food\_sold\_in) \cap \sigma_{beverage\_quantity=1}(beverage\_sold\_in))$

SELECT order_no FROM food_sold_in

WHERE food_quantity >= 1 IN (SELECT order_no FROM beverage_sold_in

WHERE beverage_quantity = 1);

```
1  SELECT order_no FROM food_sold_in
2  WHERE food_quantity >= 1 IN (SELECT order_no FROM beverage_sold_in
3  WHERE beverage_quantity = 1);
```

100% 30:3

Result Grid Filter Rows: Search Export:

order_no
1
3
5
7
9

4. What's the name of the ingredients restocked last time (on March 09, 2020) that have quantity less than 5?

$\pi_{\text{ingredient\_name}}$ (σ $_{\text{ingredient\_quantity} < 5 \wedge \text{restock\_date} = \text{'2020-03-09'}}$ ingredients $\bowtie$ restock)

SELECT ingredient_name

FROM ingredient INNER JOIN restock ON ingredient.barcode_number = restock.restocked_barcode_number

WHERE restock_date = "2020-03-09" AND ingredient_quantity < 5;

```
1 •  SELECT ingredient_name
2    FROM ingredient INNER JOIN restock
3    ON ingredient.barcode_number = restock.restocked_barcode_number
4    WHERE restock_date = "2020-03-09" AND ingredient_quantity < 5;
```

100% 63:4

Result Grid Filter Rows: Search Export:

ingredient_name
sugar glaze
honey
baking soda

5. What's the employee's job type who handled restocking duties on 2020-03-09?

$\pi_{job\_type}(\sigma_{restock\_date='2020-03-09'}(restock \bowtie employee))$

SELECT DISTINCT job_type

FROM employee INNER JOIN restock ON employee.employee_id = restock.employee_id

WHERE restock_date = "2020-03-09";

```
1 •  SELECT DISTINCT job_type
2    FROM employee INNER JOIN restock
3    ON employee.employee_id = restock.employee_id
4    WHERE restock_date = "2020-03-09";
```

100% 35:4

Result Grid Filter Rows: Search Export:

job_type
cook

6. Which employees handled the cash registry in morning shifts on Mondays, Tuesdays and Wednesdays?

SELECT employee_id

FROM employee

WHERE job_type = 'cashier'

AND LOCATE('MON_AM', availability)

AND LOCATE('TUE_AM', availability)

AND LOCATE('WED_AM', availability);

```
1 •  SELECT employee_id FROM employee
2    WHERE job_type = 'cashier'
3    AND LOCATE('MON_AM', availability)
4    AND LOCATE('TUE_AM', availability)
5    AND LOCATE('WED_AM', availability);
```

100% 37:5

Result Grid Filter Rows: Search Edit: Export/Import:

employee_id
1002

7. Which employees are making more than the hourly minimum wage?

$\pi_{employee\_id}(\sigma_{hourly\_wage > 14}(employee))$

SELECT employee_id

FROM employee

WHERE hourly_wage >= 14;

```sql
SELECT employee_id
FROM employee
WHERE hourly_wage >= 14;
```

100% 20:1

Result Grid Filter Rows: Search Edit: Export/Import:

employee_id
1001
1004
1005
1006
1007
1008

8. What are the job types of employees that are available on the weekends?

SELECT DISTINCT job_type

FROM tim_hortons.employee

WHERE LOCATE('SAT_AM', availability)

OR LOCATE('SAT_PM', availability)

OR LOCATE('SUN_AM', availability)

OR LOCATE('SUN_PM', availability);

```sql
SELECT DISTINCT job_type FROM tim_hortons.employee
WHERE LOCATE('SAT_AM', availability)
OR LOCATE('SAT_PM', availability)
OR LOCATE('SUN_AM', availability)
OR LOCATE('SUN_PM', availability);
```

100% 36:5

Result Grid Filter Rows: Search Export:

job_type
cashier
cook
barista
cleaner

9. Which employees handled orders with both food and beverage?

$\pi_{employee\_id}(\sigma_{f\_subtotal <> 0 \wedge b\_subtotal <> 0}(order))$

SELECT DISTINCT employee_id

FROM tim_hortons.order

WHERE f_subtotal <> 0 AND b_subtotal <> 0;

```
1 •  SELECT DISTINCT employee_id
2    FROM tim_hortons.order
3    WHERE f_subtotal <> 0 AND b_subtotal <> 0;
```

100% 44:3

Result Grid Filter Rows: Search Export:

employee_id
1002
1003
1001
1004

10. What are the SKU numbers of products on the order that has largest total sales?

SELECT B.sku_beverage_sold_in, F.sku_food_sold_in

FROM beverage_sold_in B LEFT OUTER JOIN food_sold_in F ON B.order_no = F.order_no

WHERE B.order_no IN

(SELECT order_no FROM tim_hortons.order WHERE total IN

(SELECT MAX(total) FROM tim_hortons.order));

```
1 •  SELECT B.sku_beverage_sold_in, F.sku_food_sold_in
2    FROM beverage_sold_in B LEFT OUTER JOIN food_sold_in F
3    ON B.order_no = F.order_no
4    WHERE B.order_no IN
5    (SELECT order_no FROM tim_hortons.order WHERE total IN
6    (SELECT MAX(total) FROM tim_hortons.order));
```

100% 46:6

Result Grid Filter Rows: Search Export:

sku_beverage_sold_in	sku_food_sold_in
1004	NULL

Part III. SQL queries: 2 deletions, 2 insertions and 2 updates, and their results

INSERT

1) INSERT query 1:

INSERT INTO product (sku, name, price, ingredients, nutrition_facts) VALUES ("1012", " nutella croissant ", "2.9", " nutella and croissant ", " super yummy ");

sku	name	price	ingredients	nutrition_facts
1001	espresso	2.5	just coffee	low carb!
1002	espresso	3	just coffee but larger	low carb!
1003	americano	2.7	coffee and water	healthy choice
1004	americano	3.2	coffee and water larger	healthy choice
1005	latte	3	coffee and milk	contains milk
1006	latte	3.5	coffee and milk larger	contains more milk
1007	regular donut	0.99	just donut	high carb sugar fat!
1008	glazed donut	1.2	just donut but with glaze	unhealthy but tasty
1009	chicken sandwich	3.5	bread, veggie, chicken	healthy combination
1010	chicken sandwich	3.5	bread, veggie, chicken, large size	larger and healther
1011	honey cruller donuts	2	donut with honey	high carb sugar fat!
1012	nutella croissant	2.9	nutella and croissant	super yummy
NULL	NULL	NULL	NULL	NULL

2) INSERT query 2:

SELECT * FROM product;

INSERT INTO food (food_sku, size) VALUES ("1012", "regular ");

food_sku	size
1007	regular
1008	regular
1009	regular
1010	large
1011	regular
1012	regular
NULL	NULL

UPDATE

1) UPDATE query 1:

UPDATE restock SET restock_date = "2020-03-18" WHERE employee_id = "1005";

SELECT * FROM restock WHERE employee_id = "1005";

restock_id	restock_date	restocked_barcode_num...	ingredient_quant...	subtotal	employee_id
6	2020-03-18 00:00:00	10002	5	50	1005
7	2020-03-18 00:00:00	10005	3	120	1005
8	2020-03-18 00:00:00	10006	5	150	1005
9	2020-03-18 00:00:00	10007	2	100	1005
10	2020-03-18 00:00:00	10009	2	40	1005
NULL	NULL	NULL	NULL	NULL	NULL

2) UPDATE query 2:

SELECT * FROM customer;

UPDATE customer SET payment_info = " Interac " WHERE customer_name = "Alice";

SELECT * FROM customer WHERE customer_name = "Alice";

account_numb...	customer_name	payment_info
1001	Alice	Interac
NULL	NULL	NULL

DELETE

1) DELETE query 1:

DELETE FROM ingredient WHERE vendor_phone_no = "1112223334";
SELECT * FROM ingredient;

barcode_number	ingredient_name	vendor	price	vendor_phone_no
10000	grind coffee	tim hortons	10	1112223333
10002	flour	flour factory	10	1012023313
10003	egg	poultry store	20	1002003000
10004	chicken breast	poultry store	30	1002003000
10005	sugar glaze	sweet shop	40	1012003313
10006	bread	flour factory	30	1012023313
10007	honey	sweet shop	50	1012003313
10008	lettuce	green farm	20	1002023223
10009	baking soda	flour factory	20	1012023313
10010	yeast	flour factory	20	1012023313
NULL	NULL	NULL	NULL	NULL

2) DELETE query 2:

```
DELETE FROM beverage WHERE hot_or_cold = "cold";
SELECT * FROM beverage;
```

beverage_sku	cup_size	hot_or_cold
1001	regular	hot
1002	large	hot
1005	regular	hot
1006	large	hot
NULL	NULL	NULL