# **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

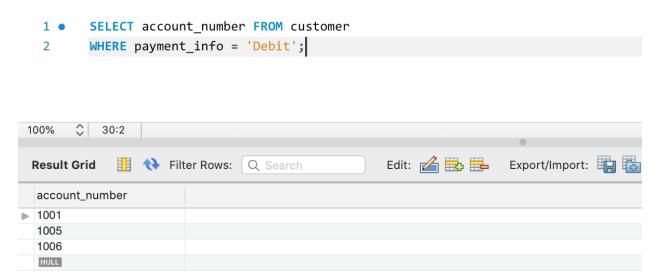
WHERE product.name = 'americano';

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

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

SELECT account\_number FROM customer

WHERE payment\_info = 'Debit';



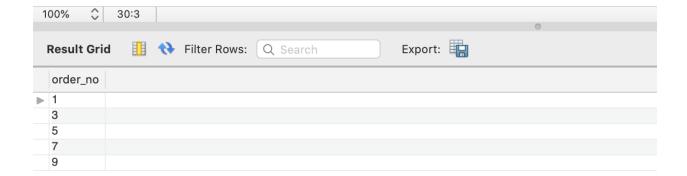
3. What orders consist of one beverage and at least one food item?

 $\pi_{\text{ order_no }}(\sigma_{\text{ food\_quantity}}=1(\text{food\_sold\_in})\cap\sigma_{\text{ beverage\_quantity}}=1(\text{beverage\_sold\_in}))$ 

SELECT order\_no FROM food\_sold\_in

WHERE food\_quantity >= 1 IN (SELECT order\_no FROM beverage\_sold\_in

WHERE beverage\_quantity = 1);



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

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

#### 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;

```
SELECT ingredient name
   1 •
   2
         FROM ingredient INNER JOIN restock
   3
         ON ingredient.barcode_number = restock.restocked_barcode_number
         WHERE restock_date = "2020-03-09" AND ingredient_quantity < 5;</pre>
   4
100%
       63:4
                                                    Export:
                 Filter Rows: Q Search
 Result Grid
 ingredient_name
sugar glaze
  honey
  baking soda
```

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

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

#### SELECT DISTINCT job\_type

FROM employee INNER JOIN restock ON employee.employee\_id = restock.employee\_id

WHERE restock\_date = "2020-03-09";

- **SELECT DISTINCT** job\_type 1 • 2 FROM employee INNER JOIN restock ON employee.employee\_id = restock.employee\_id 3 WHERE restock date = "2020-03-09"; 4 100% ♦ 35:4 Export: Filter Rows: **Result Grid** Q Search 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 WHERE job\_type = 'cashier' 2 3 AND LOCATE('MON\_AM', availability) 4 AND LOCATE('TUE\_AM', availability) AND LOCATE('WED\_AM', availability); 5 ♦ 37:5 100% Edit: 🚄 🖶 🖶 Filter Rows: Q Search Export/Import: **Result Grid** employee\_id ▶ 1002

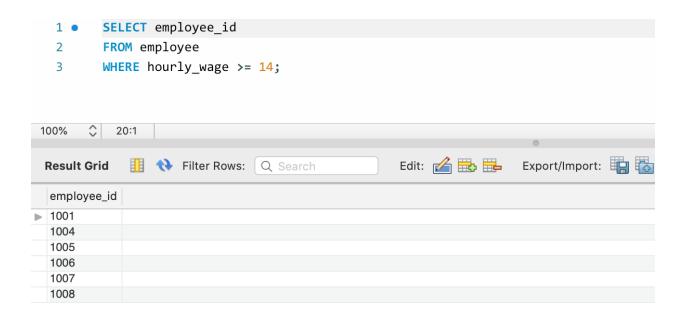
7. Which employees are making more than the hourly minimum wage?

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

SELECT employee\_id

FROM employee

WHERE hourly\_wage >= 14;



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); SELECT DISTINCT job\_type FROM tim\_hortons.employee 1 • 2 WHERE LOCATE('SAT\_AM', availability) OR LOCATE('SAT\_PM', availability) 3 OR LOCATE('SUN\_AM', availability) 4 OR LOCATE('SUN\_PM', availability); 5 \$ 36:5 100% Export: Filter Rows: Q Search **Result Grid** job\_type cashier cook barista

cleaner

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

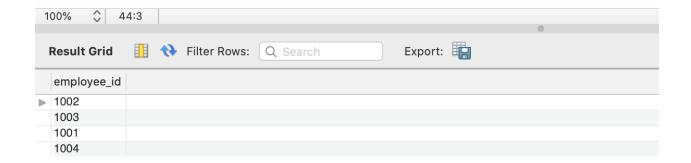
 $\pi_{employee\_id}(\sigma_{f\_subtotal} \Leftrightarrow 0 \land b\_subtotal \Leftrightarrow 0 (order))$ 

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;



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

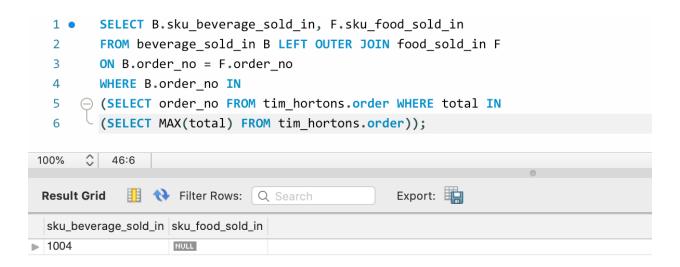
SELECT B.sku\_beverage\_sold\_in, F.sku\_food\_sold\_in

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));



# 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	HULL

#### 2) INSERT query 2:

# SELECT \* FROM product;

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

	food_sku	size
Þ	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