

```

/*
Noah Zhou
CNIT 272 Fall 2023
Lab Time: Friday 7:30 AM - 9:20 AM
*/

--
*****
*****
--Question 1
/*
Use the FOOD_SUPPLIER and FOOD tables from the LunchesDB for the following
questions.
A) List the supplier ID, supplier name, description, price upcharge, and
price for all food supplied.
☐ Use an INNER JOIN
☐ Order by the supplier name
☐ 35 rows selected.
B) List the supplier ID, supplier name, description, price upcharge, and
price for all food supplied
from suppliers with their food items including cheese. Include records
of the suppliers
providing stocks with 'Cheese' to the organization (the price upcharge
will be null in the result set).
☐ Use a LEFT JOIN
☐ 5 rows selected.
C) List the supplier ID, supplier name, description, price upcharge, and
price for only suppliers
who have provided items with 'Cheese' that have charged increased
rates to the organization.
☐ Use a LEFT JOIN and include a WHERE clause to find non-null values of
upcharges from the FOOD table
☐ 1 row selected.
*/

--Question 1-A
SELECT f.supplier_id, supplier_name, description, price_upcharge, price
FROM food_supplier fs INNER JOIN food f
ON fs.supplier_id = f.supplier_id
ORDER BY supplier_name;

--Question 1-B
SELECT f.supplier_id, supplier_name, description, price_upcharge, price
FROM food_supplier fs LEFT JOIN food f
ON fs.supplier_id = f.supplier_id
WHERE description LIKE '%Cheese%';
ORDER BY supplier_name;

--Question 1-C
SELECT f.supplier_id, supplier_name, description, price_upcharge, price
FROM food_supplier fs LEFT JOIN food f
ON fs.supplier_id = f.supplier_id
WHERE description LIKE '%Cheese%' AND price_upcharge IS NOT NULL;

```

/*

Results:

[Question 1-A]

SUP	SUPPLIER_NAME	DESCRIPTION	PRICE_UPCHARGE
-----	---------------	-------------	----------------

PRICE

---	-----	-----	---

Ard Arnoldo Deli		PB Cookie	
1.25			
Ard Arnoldo Deli		Chicken Avocado Wrap	
5.25			
Ard Arnoldo Deli		Veggie Pizza	
6.25			
Blu Blue Sky Deli		Turkey Club	.5
7.5			
Blu Blue Sky Deli		Chips	
2.8			
Crm Crystal Market		Ham Melt	.3
7.2			
Crm Crystal Market		Grilled Cheese	.4
5.25			
Crm Crystal Market		Philly Melt	.7
8.6			
Crm Crystal Market		Wheat Bagel	.25
4			
Dpz Downtowner Pizza		Dinner Roll	.25
1			
Dpz Downtowner Pizza		Cheese Pizza	
8.5			

SUP	SUPPLIER_NAME	DESCRIPTION	PRICE_UPCHARGE
-----	---------------	-------------	----------------

PRICE

---	-----	-----	---

Dpz Downtowner Pizza		Cheese Sauce	
.75			
Foi Fontinas Italian		Apple Pie	
4.5			
Foi Fontinas Italian		Barley Soup	.2
4.75			
Foi Fontinas Italian		French Fries	
1.5			
Foi Fontinas Italian		Broccoli Salad	.05
4			
Gls Great Lakes Station		Breadstick	
1.25			
Gls Great Lakes Station		Mango Smoothie	
4.95			
Gls Great Lakes Station		Sugar Cookie	.5
3.5			
Hsd Harper Street Deli		Yellow Cake	.5
4.5			

Hsd Harper Street Deli	Chicken Soup	
4.25		
Hsd Harper Street Deli	Chef Salad	.25
5.75		

SUP SUPPLIER_NAME	DESCRIPTION	PRICE_UPCHARGE
PRICE		

Hsd Harper Street Deli	Protein Box	.5
9		
Jmd Jebston Montrose Deli	Potato Soup	
6.25		
Jmd Jebston Montrose Deli	Mushroom Pizza	.15
8		
Jmd Jebston Montrose Deli	Cole Slaw	.5
1.5		
Jmd Jebston Montrose Deli	Iced Tea	.15
2.85		
Jd6 Justin's Deli at 652	Brownie	.25
3.45		
Jd6 Justin's Deli at 652	Soda	.25
2.25		
Jd6 Justin's Deli at 652	Coffee	.15
1.55		
Lak Lakeshore Bakery	Caprese	.5
5.5		
Lak Lakeshore Bakery	Cheese Stick	
2.35		
Lak Lakeshore Bakery	Spinach Salad	
6.25		

SUP SUPPLIER_NAME	DESCRIPTION	PRICE_UPCHARGE
PRICE		

Lss Lucias Sub Shop	Veggie Soup	
4.85		
Lss Lucias Sub Shop	Grilled Cheese	
5.95		

35 rows selected.

[Question 1-B]

SUP SUPPLIER_NAME	DESCRIPTION	PRICE_UPCHARGE
PRICE		

Crm Crystal Market	Grilled Cheese	.4
5.25		
Dpz Downtowner Pizza	Cheese Pizza	
8.5		
Dpz Downtowner Pizza	Cheese Sauce	
.75		

Lak Lakeshore Bakery	Cheese Stick
2.35	
Lss Lucias Sub Shop	Grilled Cheese
5.95	

[Question 1-C]

SUP	SUPPLIER_NAME	DESCRIPTION	PRICE_UPCHARGE
PRICE			

```

-----
-----
Crm Crystal Market      Grilled Cheese      .4
5.25
*/

```

--

--Question 2

/*

List the supplier ID, food description, and total quantity (Hint: Use SUM) for each supplied food item.

If the food item has never been purchased, still list the supplier ID and description.

Group by supplier ID and description. Sort by Description.

A) Use LEFT JOIN

☐ The relationship includes a composite primary key connection in the FROM clause.

☐ 35 rows selected

B) Change the LEFT JOIN to an INNER JOIN and rerun it. Explain the difference and

what is missing in the result set. What causes the difference?

☐ 31 rows selected

*/

--Question 2-A

```

SELECT li.supplier_id, description, SUM(quantity) AS TotalQuantity
FROM food f LEFT JOIN lunch_item li
ON f.supplier_id = li.supplier_id AND f.product_code = li.product_code
GROUP BY li.supplier_id, description;
ORDER BY description;

```

--Question 2-B

```

SELECT li.supplier_id, description, SUM(quantity) AS TotalQuantity
FROM food f INNER JOIN lunch_item li
ON f.supplier_id = li.supplier_id AND f.product_code = li.product_code
GROUP BY li.supplier_id, description
ORDER BY description;
/*

```

Results:

[Question 2-A]

SUP	DESCRIPTION	TOTALQUANTITY
---	-----	-----

Foi Apple Pie	3
Foi Barley Soup	4

Gls Breadstick	1
Foi Broccoli Salad	1
Jd6 Brownie	10
Lak Caprese	2
Dpz Cheese Pizza	4
Dpz Cheese Sauce	2
Lak Cheese Stick	5
Hsd Chef Salad	14
Chicken Avocado Wrap	

SUP DESCRIPTION	TOTALQUANTITY
---	-----
Hsd Chicken Soup	13
Blu Chips	2
Jd6 Coffee	6
Cole Slaw	
Dpz Dinner Roll	8
Foi French Fries	11
Crm Grilled Cheese	7
Lss Grilled Cheese	4
Crm Ham Melt	10
Jmd Iced Tea	10
Mango Smoothie	

SUP DESCRIPTION	TOTALQUANTITY
---	-----
Jmd Mushroom Pizza	3
PB Cookie	
Crm Philly Melt	12
Jmd Potato Soup	1
Hsd Protein Box	2
Jd6 Soda	25
Lak Spinach Salad	2
Gls Sugar Cookie	9
Blu Turkey Club	1
Ard Veggie Pizza	4
Lss Veggie Soup	2

SUP DESCRIPTION	TOTALQUANTITY
---	-----
Crm Wheat Bagel	1
Hsd Yellow Cake	6

35 rows selected.

[Question 2-B]

SUP DESCRIPTION	TOTALQUANTITY
---	-----
Foi Apple Pie	3
Foi Barley Soup	4
Gls Breadstick	1
Foi Broccoli Salad	1
Jd6 Brownie	10
Lak Caprese	2

Dpz Cheese Pizza	4
Dpz Cheese Sauce	2
Lak Cheese Stick	5
Hsd Chef Salad	14
Hsd Chicken Soup	13

SUP DESCRIPTION	TOTALQUANTITY
---	-----
Blu Chips	2
Jd6 Coffee	6
Dpz Dinner Roll	8
Foi French Fries	11
Crm Grilled Cheese	7
Lss Grilled Cheese	4
Crm Ham Melt	10
Jmd Iced Tea	10
Jmd Mushroom Pizza	3
Crm Philly Melt	12
Jmd Potato Soup	1

SUP DESCRIPTION	TOTALQUANTITY
---	-----
Hsd Protein Box	2
Jd6 Soda	25
Lak Spinach Salad	2
Gls Sugar Cookie	9
Blu Turkey Club	1
Ard Veggie Pizza	4
Lss Veggie Soup	2
Crm Wheat Bagel	1
Hsd Yellow Cake	6

31 rows selected.

Explanation: LEFT JOIN returned all the records from the FOOD table and also returned any matching records from the LUNCH_ITEM table. INNER JOIN returned all the records that have matching values in both the FOOD and LUNCH_ITEM table.

*/

--

--Question 3

/*

Use the DEPARTMENT and WORKER tables for the following questions.

A) List the worker id, city, department code, and department name for all workers in the database.

- ☐ Set the linespace to 200 for readability.
- ☐ Use an INNER JOIN
- ☐ Order by the department code
- ☐ 26 rows selected.

B) List the worker id, city, department code, and department name for all workers in the database.

Include all workers, even if they have not been assigned to a department yet.

- ☐ Use a RIGHT JOIN
- ☐ Order by the worker id
- ☐ 34 rows selected.

C) List the worker id, city, department code, and department name for all workers in the database.

Also include all departments, even if they have no employees.

- ☐ Use a RIGHT JOIN
- ☐ Order by the department code
- ☐ 31 rows selected

D) Create a nested query to display all departments without assigned workers. Include

the department code and the department name in the result set.

- ☐ In the nested query, add a WHERE clause to find the department codes that are not NULL
- ☐ Remember that you need to use NOT IN in the WHERE clause when comparing to the nested query
- ☐ 5 rows selected.

*/

--Question 3-A

set linespace 200

```
SELECT worker_id, city, w.dept_code, department_name
FROM department d INNER JOIN worker w
ON d.dept_code = w.dept_code
ORDER BY dept_code;
```

--Question 3-B

```
SELECT worker_id, city, w.dept_code, department_name
FROM department d RIGHT JOIN worker w
ON d.dept_code = w.dept_code
ORDER BY worker_id;
```

--Question 3-C

```
SELECT w.worker_id, city, w.dept_code, department_name
FROM worker w RIGHT JOIN department d
ON w.dept_code = d.dept_code
ORDER BY dept_code;
```

--Question 3-D

```
SELECT dept_code, department_name
FROM department
WHERE dept_code NOT IN (SELECT dept_code FROM worker WHERE dept_code IS
NOT NULL);
```

/*

Results:

[Question 3-A]

WOR CITY	DEP DEPARTMENT_NAME
580	Acc Accounting
582 Hinsdale	Acc Accounting
563 Oak Brook	Acc Accounting

570	Aurora	Aud	Auditing
573		Aud	Auditing
560	Chicago	Aud	Auditing
562	Glencoe	Com	Compliance
555	Aurora	Exe	Executive
575	Glencoe	Fin	Finance
569	Evanston	Fin	Finance
558	Oak Brook	Hmn	Human Resources

WOR	CITY	DEP	DEPARTMENT_NAME
-----	------	-----	-----------------

566	Oak Brook	Hmn	Human Resources
571	Chicago	Hmn	Human Resources
567	Oak Brook	Leg	Legal
557	Wilmette	Leg	Legal
587	Oak Brook	Leg	Legal
577	Wilmette	Sal	Sales
564	Chicago	Sal	Sales
559	Hinsdale	Srv	Service
585	Evanston	Srv	Service
581	Oak Brook	Srv	Service
556	Hinsdale	Tch	Technology

WOR	CITY	DEP	DEPARTMENT_NAME
-----	------	-----	-----------------

565		Tch	Technology
584	Chicago	Tch	Technology
576	Hinsdale	Tch	Technology
578	Evanston	Tch	Technology

26 rows selected.

[Question 3-B]

WOR	CITY	DEP	DEPARTMENT_NAME
-----	------	-----	-----------------

555	Aurora	Exe	Executive
556	Hinsdale	Tch	Technology
557	Wilmette	Leg	Legal
558	Oak Brook	Hmn	Human Resources
559	Hinsdale	Srv	Service
560	Chicago	Aud	Auditing
561	Wilmette		
562	Glencoe	Com	Compliance
563	Oak Brook	Acc	Accounting
564	Chicago	Sal	Sales
565		Tch	Technology

WOR	CITY	DEP	DEPARTMENT_NAME
-----	------	-----	-----------------

566	Oak Brook	Hmn	Human Resources
567	Oak Brook	Leg	Legal
568	Wilmette		
569	Evanston	Fin	Finance
570	Aurora	Aud	Auditing

571 Chicago	Hmn Human Resources
572 Hinsdale	
573	Aud Auditing
574	
575 Glencoe	Fin Finance
576 Hinsdale	Tch Technology

WOR CITY	DEP DEPARTMENT_NAME
---	-----

577 Wilmette	Sal Sales
578 Evanston	Tch Technology
579	
580	Acc Accounting
581 Oak Brook	Srv Service
582 Hinsdale	Acc Accounting
583 Aurora	
584 Chicago	Tch Technology
585 Evanston	Srv Service
586 Glencoe	
587 Oak Brook	Leg Legal

WOR CITY	DEP DEPARTMENT_NAME
---	-----

588 Chicago

34 rows selected.

[Question 3-C]

WOR CITY	DEP DEPARTMENT_NAME
---	-----

580	Acc Accounting
582 Hinsdale	Acc Accounting
563 Oak Brook	Acc Accounting
570 Aurora	Aud Auditing
573	Aud Auditing
560 Chicago	Aud Auditing
562 Glencoe	Com Compliance
555 Aurora	Exe Executive
575 Glencoe	Fin Finance
569 Evanston	Fin Finance
571 Chicago	Hmn Human Resources

WOR CITY	DEP DEPARTMENT_NAME
---	-----

566 Oak Brook	Hmn Human Resources
558 Oak Brook	Hmn Human Resources
567 Oak Brook	Leg Legal
587 Oak Brook	Leg Legal
557 Wilmette	Leg Legal
564 Chicago	Sal Sales
577 Wilmette	Sal Sales
581 Oak Brook	Srv Service
585 Evanston	Srv Service

```

559 Hinsdale      Srv Service
584 Chicago       Tch Technology

```

```

WOR CITY          DEP DEPARTMENT_NAME
-----
576 Hinsdale      Tch Technology
578 Evanston      Tch Technology
565               Tch Technology
556 Hinsdale      Tch Technology
                  Engineering
                  Facilities
                  Marketing
                  Purchasing
                  Shipping

```

31 rows selected.

```

[Question 3-D]
DEP DEPARTMENT_NAME
-----
Eng Engineering
Fac Facilities
Mkt Marketing
Pur Purchasing
Shp Shipping

```

*/

```

--
*****
*****

```

```

--Question 4
/*

```

Use a UNION operation to list the names of the supplier cities and the food suppliers,

as well as the concatenated first and last names of the workers and cities into one SQL statement

□ It is the UNION of two queries, one is to list supplier name and supplier city from FOOD_SUPPLIER table,

and the other is to list the worker name(concatenated) and city from WORKER table.

□ The result set should only have two columns (a column for all of the names and a column for all of the cities).

□ Review concatenation (||) if needed. There is a space between worker first name and last name.

□ Limit the results to only food suppliers and workers NOT from the cities of Skokie, Oak Brook, and Chicago.

□ Sort by the name (sort by supplier name if you used the food supplier table in the first query,

otherwise, sort by the worker name)

□ 26 rows selected (8 from FOOD_SUPPLIER, 18 from WORKER)

*/

```

SELECT supplier_name, supplier_city

```

```

FROM food_supplier
WHERE supplier_city NOT IN ('Skokie', 'Oak Brook', 'Chicago')
UNION
SELECT first_name || ' ' || last_name, city
FROM worker
WHERE city NOT IN ('Skokie', 'Oak Brook', 'Chicago')
ORDER BY supplier_name;

```

/*

Results:

SUPPLIER_NAME	SUPPLIER_CITY
Angie Templeton	Hinsdale
Arnoldo Deli	Glencoe
Avery Trance	Aurora
Blair Reynolds	Evanston
Carey Martin	Wilmette
Carole Sumner	Hinsdale
Chase Johnson	Hinsdale
Crystal Market	Hinsdale
Dane Shreve	Aurora
Framer and Samson	Orland Park
Fresh Daily Vegetables	Aurora

SUPPLIER_NAME	SUPPLIER_CITY
Great Lakes Station	Wilmette
Harold Bakery	Hinsdale
Jodie Williams	Glencoe
Jose Sanchez	Glencoe
Katelynn Rayner	Evanston
Keyanna Jones	Hinsdale
Lucias Sub Shop	Aurora
Maria Bensen	Wilmette
Rita Gradle	Wilmette
Rosemont Bakery	Glencoe
Sam Frank	Evanston

SUPPLIER_NAME	SUPPLIER_CITY
Shawn Smith	Glencoe
Taylor Young	Wilmette
Tom Neal	Hinsdale
Tonya Montre	Aurora

26 rows selected.

*/

--

--Question 5

/*

Use the INTERSECT operation to find any food suppliers that are living in the same

city as any of the workers. 6 rows selected.

```
*/
SELECT supplier_city
FROM food_supplier
INTERSECT
SELECT city
FROM worker;
```

/*

Results:
SUPPLIER_CITY

Aurora
Chicago
Glencoe
Hinsdale
Oak Brook
Wilmette

6 rows selected.

*/

--

--Question 6

/*

Use the MINUS operation to list the cities of the workers where no food suppliers live

(i.e. city where there are workers living, but not suppliers).

A) Start by running a query that lists the Distinct worker cities.

□ 8 rows selected. (Hint: using distinct will only list each city once)

B) Then run a query that lists the Distinct food supplier cities.

□ 7 rows selected.

C) Using the MINUS operation, combine the two queries from 6A and 6B, starting

with the suppliers' cities first.

□ 1 row selected

*/

--Question 6-A

```
SELECT DISTINCT city
FROM worker;
```

--Question 6-B

```
SELECT DISTINCT supplier_city
FROM food_supplier;
```

--Question 6-C

```
SELECT DISTINCT supplier_city
FROM food_supplier
MINUS
SELECT DISTINCT city
FROM worker;
```

/*

Results:

[Question 6-A]

CITY

Aurora

Oak Brook

Chicago

Wilmette

Evanston

Hinsdale

Glencoe

8 rows selected.

[Question 6-B]

SUPPLIER_CITY

Oak Brook

Aurora

Orland Park

Chicago

Wilmette

Glencoe

Hinsdale

7 rows selected.

[Question 6-C]

SUPPLIER_CITY

Orland Park

*/

--

