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

--*********************************************************************************
--Question 1
/*
List supplier id, name, and city of any supplied food with its price is more than
$7. Use
a nested query to find food items in the FOOD table that the price are more than
$7.
^ Use a subquery or also called a nested query
^ FOOD_SUPPLIER (and FOOD table inside of the subquery)
^ 5 rows selected
*/
SELECT supplier_id, supplier_name, supplier_city
FROM food_supplier
WHERE supplier_id IN
    (SELECT supplier_id FROM food WHERE price>7.00);

/*
Results:
SUP SUPPLIER_NAME                  SUPPLIER_CITY
--- ------------------------------ ------------------------------
Blu Blue Sky Deli                  Oak Brook
Crm Crystal Market                 Hinsdale
Dpz Downtowner Pizza               Chicago
Hsd Harper Street Deli             Chicago
Jmd Jebston Montrose Deli          Chicago

5 rows selected.
*/


--*********************************************************************************
--Question 2
/*
List the worker ID, first name, last name, and hire date of the workers with no
lunch
orders. Sort the list by Hire Date.
^ Type ‘set linespace 200’ in the first line to help with column wrapping, then
write your
query start in the second line.
^ Use a subquery with NOT IN
^ WORKER table (and LUNCH table inside of the subquery)
^ 8 rows selected
*/
set linespace 200

SELECT worker_id, first_name, last_name, hire_date
FROM worker
WHERE worker_id NOT IN
    (SELECT worker_id FROM lunch)
ORDER BY hire_date;


/*
```

```
Results:
WOR FIRST_NAME LAST_NAME            HIRE_DATE
--- ---------- -------------------- ---------
574 Cassie     Irwin                13-SEP-03
561 Maria      Bensen               15-SEP-07
580 Roy        Gonzalez             19-DEC-11
583 Tonya      Montre               07-MAR-12
569 Blair      Reynolds             07-FEB-13
558 Melody     Campbell             24-MAY-18
566 Tami       Tevona               21-AUG-19
588 Gail       Walsh                22-MAR-21

8 rows selected.

*/


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

--Question 3
/*
Find the food supplier id, the supplier name, product code, price and supplier city
for all
Suppliers from the city of Chicago and food priced more than $4. Sort by price.
^ Type ‘set linespace 200’ in the first line, then write your query start in the
second line.
^ Use an inner join to obtain this result.
^ FOOD SUPPLIER and FOOD tables
^ Try using table aliases such as ‘fs’ for ‘food_supplier’ to avoid typing long
table names
over and over.
^ Note that the attribute Supplier_ID is used in both tables
^ 11 rows selected.
*/
set linespace 200

SELECT fs.supplier_id AS Supplier_ID, supplier_name AS Supplier_Name, description
AS Description, price AS Price
FROM food_supplier fs INNER JOIN food f ON fs.supplier_id = f.supplier_id
WHERE Price>4.00 AND supplier_city = 'Chicago'
ORDER BY Price;
/*
Results:
SUP SUPPLIER_NAME                  DESCRIPTION               PRICE
--- ------------------------------ -------------------- ----------
Hsd Harper Street Deli             Chicken Soup               4.25
Foi Fontinas Italian               Apple Pie                   4.5
Hsd Harper Street Deli             Yellow Cake                 4.5
Foi Fontinas Italian               Barley Soup                4.75
Lak Lakeshore Bakery               Caprese                     5.5
Hsd Harper Street Deli             Chef Salad                 5.75
Jmd Jebston Montrose Deli          Potato Soup                6.25
Lak Lakeshore Bakery               Spinach Salad              6.25
Jmd Jebston Montrose Deli          Mushroom Pizza                8
Dpz Downtowner Pizza               Cheese Pizza                8.5
Hsd Harper Street Deli             Protein Box                   9

11 rows selected.
*/
```

```
--*********************************************************************************
--Question 4
/*
(The results of question 4 are not required to paste in your submission)
A) Find the last name, first name, department code, lunch date and lunch id, for
all
workers who placed lunch orders in May 2021(within May 1st,2021 to May 31st, 2021).
^ Use an inner join to obtain this result.
^ Use WORKER and LUNCH tables
^ Sort by the lunch date and the lunch id
^ 13 rows selected.

B) Find the last name, first name, department code, lunch date and lunch id, and
item
number for all workers who placed lunch orders in May 2021.
^ Builds on part 4A… notice that you add item number in the SELECT clause
^ Note that this involves 3 tables and 2 relationships.
^ Use WORKER, LUNCH, and LUNCH_ITEM tables
^ Sort by the lunch date, lunch id, and the item number
^ 57 rows selected.

C) Find the last name, first name, department code, lunch date and lunch id, and
item
number for all workers who placed lunch orders in May 2021.
Note that this involves 4 tables and 3 relationships (Build on Part 4B)
^ Use WORKER, LUNCH, LUNCH_ITEM, and FOOD tables
^ Same number of rows with 4B but need information from an additional
table(FOOD).
^ Note that FOOD to LUNCH_ITEM is a composite PK relationship.
^ 57 rows selected.
*/
--Question 4-A
SELECT last_name, first_name, dept_code, lunch_id, lunch_date
FROM worker w INNER JOIN lunch l ON w.worker_id = l.worker_id
WHERE lunch_date >= '01-MAY-2021' AND lunch_date <= '31-MAY-2021'
ORDER BY lunch_date, lunch_id;


--Question  4-B
SELECT last_name, first_name, dept_code, li.lunch_id, lunch_date, item_number
FROM worker w INNER JOIN lunch l ON w.worker_id = l.worker_id
INNER JOIN lunch_item li  ON l.lunch_id = li.lunch_id
WHERE lunch_date >= '01-MAY-2021' AND lunch_date <= '31-MAY-2021'
ORDER BY lunch_date, lunch_id, item_number;


--Question 4-C
SELECT last_name, first_name, dept_code, li.lunch_id, lunch_date, item_number
FROM worker w INNER JOIN lunch l ON w.worker_id = l.worker_id
INNER JOIN lunch_item li ON l.lunch_id = li.lunch_id
INNER JOIN food f ON li.supplier_id = f.supplier_id AND li.product_code =
f.product_code
WHERE lunch_date >= '01-MAY-2021' AND lunch_date <= '31-MAY-2021'
ORDER BY lunch_date, lunch_id, item_number;
/*

--*********************************************************************************
--Question 5
```

```
/*
List the lunch ID, lunch date, the count of menu items per purchased lunch, and the
total price of a purchased lunch per lunch id. Only include lunch dates from the
second
half of May 2021 (May 16, 2021 – May 31, 2021). Label the total price expression as
TOTAL_PRICE and the count of items as COUNT. Format the TOTAL_PRICE in
currency format. Sort by the lunch date, then the lunch id.
^ Type ‘col TOTAL_PRICE format $9999.99’ in the first line to format the
TOTAL_PRICE in currency format, then write your query start in the second line.
^ This involves 3 tables and 2 relationships.
^ LUNCH, LUNCH_ITEM, FOOD tables
^ Note again that FOOD to LUNCH_ITEM is a composite PK relationship.
^ Group by the non-aggregated columns from the SELECT clause.
^ 13 rows selected.
*/
col TOTAL_PRICE format $9999.99

SELECT l.lunch_id, lunch_date, COUNT(menu_item) AS Count, SUM(price) AS Total_Price
FROM lunch l INNER JOIN lunch_item li ON l.lunch_id = li.lunch_id
INNER JOIN food f ON li.supplier_id = f.supplier_id AND li.product_code =
f.product_code
WHERE lunch_date >= '16-MAY-2021' AND lunch_date <= '31-MAY-2021'
GROUP BY lunch_date, l.lunch_id
ORDER BY l.lunch_id, lunch_date;
/*
Results:
LUNCH_ID LUNCH_DAT      COUNT TOTAL_PRICE
---------- --------- ---------- -----------
       1 22-MAY-21          3    $12.25
       2 22-MAY-21          4    $16.25
       3 22-MAY-21          5    $23.20
       4 22-MAY-21          4    $19.40
       5 22-MAY-21          5    $22.25
       6 22-MAY-21          5    $23.10
       7 27-MAY-21          3    $17.80
       8 27-MAY-21          6    $24.40
       9 27-MAY-21          4    $19.40
      10 27-MAY-21          4    $14.45
      11 27-MAY-21          4    $15.20

LUNCH_ID LUNCH_DAT      COUNT TOTAL_PRICE
---------- --------- ---------- -----------
      12 27-MAY-21          5    $24.95
      13 27-MAY-21          5    $21.10

13 rows selected.
*/


--*********************************************************************************
--Question 6
/*
A) Count the number of lunches per lunch date purchased by workers who are not
assigned to any department and hired between December 01, 2012, and April 01 2020.
Format the lunch date in the fmMONTH-YYYY format.
^ Use an inner join to obtain this result
^ WORKER and LUNCH tables
^ Label the lunch date as LD and the count as LUNCH_COUNT
^ Group by the non-aggregated column in the SELECT clause
^ Sort by LD.
^ 2 rows selected.

B) Expand on 6A: Add a clause to remove any LD with a total lunch count greater
than 2.
^ 1 row selected.
*/

--Question 6-A
SELECT TO_CHAR(lunch_date,'fmMONTH,YYYY') AS LD, COUNT(lunch_id) AS Lunch_Count
FROM worker w INNER JOIN lunch l ON w.worker_id = l.worker_id
WHERE hire_date BETWEEN '01-DEC-2012' AND '01-APR-2020' AND dept_code IS NULL
GROUP BY TO_CHAR(lunch_date,'fmMONTH,YYYY');

--Question 6-B
SELECT TO_CHAR(lunch_date,'fmMONTH,YYYY') AS LD, COUNT(lunch_id) AS Lunch_Count
FROM worker w INNER JOIN lunch l ON w.worker_id = l.worker_id
WHERE hire_date BETWEEN '01-DEC-2012' AND '01-APR-2020' AND dept_code IS NULL
GROUP BY TO_CHAR(lunch_date,'fmMONTH,YYYY')
HAVING COUNT(lunch_id)<3;

/*
Results:
[Question 6-A]
LD             LUNCH_COUNT
-------------- -----------
JUNE,2021                3
MAY,2021                 1

[Question 6-B]

LD             LUNCH_COUNT
-------------- -----------
MAY,2021                 1
*/


--**********************************************************************************
```