

# Database Systems- Assignment #2

## Submission Instructions

Your final submission should contain a zip file containing all submission requirements and a details.txt file that should contain the ID and name of each submitting student, such as:

first name last name 123456789

first name last name 987654321

- The zip file should be named with your ID, such as '123456789.zip'
- If multiple students submit, separate the ID numbers with an underscore, such as '123456789 987654321.zip'.
- In this assignment, your zip file needs to contain 8 .py files, make sure
- you don't add any other unnecessary files to the zip
- Submission is through the course website and can be either individually
- or in pairs:
  - If you submit in pairs, only one of the two should submit, and that student will also be the one to receive feedback.
  - A submission for three people is not permitted. If you cannot find a pair, you should submit by yourself (it is not necessary to ask for permission to submit by yourself).
- Submission due: 08.05.25

## Requirements

For each query, you are required to submit a single file called:

- **q{n}.py** if the question contains a single query (for example, q1.py, q2.py, etc.)
- **q{n m}.py** (for example, q2 1.py, q2 2.py, etc.) for each query in the question.

The file will include the following:

- The Python code and SQL query you used.
- Documentation - mandatory. Use # to add comments.

For Insert/Update queries each of the python files need to be in this format:

```
import mysql.connector

if __name__ == '__main__':
    mydb = mysql.connector.connect(
        host="localhost",
        user="root",
        password="root",
        database="## PUT THE CORRECT DATABASE HERE IF
NEEDED ##",
        port='3307',
    )
    cursor = mydb.cursor()
    cursor.execute("""
## PUT YOUR QUERY ##
""")
    # !!!Commit the transaction to save the changes to the
    database!!!
    mydb.commit()
    cursor.close()
    mydb.close()
```

For all other queries, each of the python files need to be in the format:

```
import mysql.connector

if __name__ == '__main__':
    mydb = mysql.connector.connect(

        host="localhost",
        user="root",
        password="root",
        database="## PUT THE CORRECT DATABASE HERE IF
NEEDED ##",
        port='3307',
    )
    cursor = mydb.cursor()
    cursor.execute("""
    ## PUT YOUR QUERY ##
    """)
    print(', '.join(str(row) for row in cursor.fetchall())
    ))
```

## Important Notes

- Do not use commands that have not yet been learned up to the publication of this assignment, or will not be learned at all.
- All of the required commands have already been learned by the time of posting this exercise.
- If a query uses such commands, all points will be deducted.
- Please make sure that the queries are well formatted (use the syntax conversion learned in class.)

```
SELECT film_id, title
FROM film
WHERE length > 10
ORDER BY title ASC
```

## Objective

The purpose of this assignment is to design, populate, and query a database for the infamous fast food chain branch, O'donald's. You will construct a database, create tables, and insert data using the SQL queries covered in class. Additionally, you will be required to extract meaningful insights (queries) by using the SQL commands learned during the course.

## Background Story

Noam recently graduated with a degree in Computer Science, but as you might know- it's pretty hard to find a position in tech right now, so Noam had to settle for a cooking job at O'donald's burgers.

With time, Noam's supervisors noticed her (yes, it's a she :)) potential, and decided to promote her from being a cook to develop the branch's database and analyze the data.

While thrilled by the opportunity, Noam feels slightly overwhelmed. She heard that **you** study database systems, and she asks you to help her.

## Part 1: Database Construction

### Question 1:

Write a query that creates a database for O'donald's burgers. The database is called *burgers*.

### Question 2:

Peter Griffin, the branch's IT manager, gave Noam some table requirements and guidelines for the database structure. For each table, write a query that creates it in the database.

Ensure you select the appropriate data types and constraints.

1. **menu\_item:** (item\_id : INT, item\_name: VARCHAR NOT NULL, price : SMALLINT NOT NULL)

2. **menu\_meal:** (meal\_id : INT, meal\_name: VARCHAR NOT NULL, price : SMALLINT NOT NULL, served\_at: VARCHAR NOT NULL)

3. **meal\_item:** (meal\_id: INT, item\_id: INT NOT NULL)

**Foreign Keys:**

**meal\_item**(meal\_id) → **menu\_meal**(meal\_id)

**meal\_item**(item\_id) → **menu\_item**(item\_id)

4. **city:** (city\_id: INT, city\_name: VARCHAR NOT NULL)

5. **street:** (street\_id: INT, street\_name: VARCHAR NOT NULL, in\_city: INT NOT NULL)

**Foreign Keys:**

**street**(in\_city) → **city**(city\_id)

6. **address:** (address\_id: INT, in\_street: INT NOT NULL, street\_number: SMALLINT NOT NULL, floor: SMALLINT NOT NULL)

**Foreign Keys:**

**address**(in\_street) → **street**(street\_id)

7. **client:** (client\_id: INT, client\_name: VARCHAR NOT NULL, client\_address: INT NOT NULL )

**Foreign Keys:**

**client**(client\_address) → **address**(address\_id)

8. **full\_order**: (order\_id: INT, order\_time: DATETIME NOT NULL, by\_client: INT NOT NULL)  
INSERT INTO "order" (order\_id, order\_time, by\_client) VALUES

**Foreign Keys:**

**full\_order**(by\_client) → **client**(client\_id)

9. **order\_item**: (order\_id: INT, item\_id: INT, is\_meal: BOOLEAN)

**Foreign Keys:**

**order\_item**(order\_id) → **full\_order**(order\_id)

**order\_item**(item\_id) → **menu\_item**(item\_id)

### Question 3:

For each of the following table, insert the given rows in a query

1. menu\_item table:  
(1, 'small fries', 2), (2, 'Os salad', 2), (3, 'large fries', 5), (4, 'Oburger', 6),  
(5, 'double Oburger', 10), (6, 'chicken Oburger', 3), (7, 'Onuggets', 3),  
(8, 'double Onuggets', 5), (9, 'Omlette', 3), (10, 'Opancake', 3), (11, 'Owaffle', 3),  
(12, 'light Obeverage', 2), (13, 'hot Obeverage', 2)
2. menu\_meal table:  
(1, 'small Oburger Meal', 8, 'all day'),  
(2, 'large Oburger Meal', 14, 'all day'),  
(3, 'chicken Oburger meal', 8, 'all day'),  
(4, 'small Onuggets meal', 7, 'all day'),  
(5, 'large Onuggets meal', 11, 'all day'),  
(6, 'Omfast', 4, 'morning'),  
(7, 'Opanfast', 4, 'morning'),  
(8, 'Owaffast', 4, 'morning')
3. meal\_item table:  
(1, 1), (1, 4), (1, 12),  
(2, 3), (2, 5), (2, 12),  
(3, 1), (3, 6), (3, 12),  
(4, 1), (4, 7), (4, 12),  
(5, 3), (5, 8), (5, 12),  
(6, 9), (6, 13),  
(7, 10), (7, 13),  
(8, 11), (8, 13)
4. city table:  
(1, 'MIA'), (2, 'NYC'), (3, 'SF')
5. street table:  
1, 'Ocean Drive st.', 1),

- (2, 'Brickell ave.', 1),
- (3, 'Park ave.', 2),
- (4, 'Wall st.', 2),
- (5, 'Lombard ave.', 3),
- (6, 'Market st.', 3)
- 6. address table:
  - (1, 2, 10, 2),
  - (2, 1, 1, 10),
  - (3, 3, 23, 101),
  - (4, 4, 52, 304),
  - (5, 5, 5, 2),
  - (6, 6, 34, 1)
- 7. client table:
  - (1, 'Lee Kang', 5),
  - (2, 'Wang Fu', 6),
  - (3, 'Sandro Moretti', 3),
  - (4, 'Giuseppe Angelo', 4),
  - (5, 'Favian Adames Alonso', 1),
  - (6, 'Edward Pagan Henriquez', 2)
- 8. full\_order table:
  - (1, '2025-04-20 08:15:00', 1),
  - (2, '2025-04-20 09:00:00', 1),
  - (3, '2025-04-20 10:00:00', 2),
  - (4, '2025-04-20 11:00:00', 2),
  - (5, '2025-04-20 13:00:00', 1),
  - (6, '2025-04-20 08:30:00', 3),
  - (7, '2025-04-20 14:00:00', 3),
  - (8, '2025-04-20 14:30:00', 4),
  - (9, '2025-04-20 12:00:00', 5),
  - (10, '2025-04-20 12:15:00', 6),
  - (11, '2025-04-20 08:15:00', 5)
- 9. order\_item table:
  - (1, 6, 1),
  - (2, 7, 1),
  - (3, 8, 1),
  - (4, 6, 1),
  - (5, 1, 1),
  - (6, 6, 0),
  - (6, 12, 0),
  - (6, 1, 0),
  - (6, 6, 1),
  - (7, 1, 0),
  - (7, 1, 1),
  - (8, 3, 0),
  - (8, 2, 1),

(9, 6, 1),  
(9, 4, 1),  
(9, 5, 1),  
(10, 2, 0),  
(10, 2, 1),  
(11, 1, 1)

## Part 2- Advanced Data Analysis

### Question 4: Business Analysis- Meal Type By City

Your first analysis task is to create a report on how the popularity of the different meal types (breakfast-morning, regular- all day) is distributed across the cities.

Write a query that will return a table of cities, and for each city- total of

1. breakfasts
2. regular meals

The cities should appear in an alphabetical order.

### Question 5: Business Analysis- Popular Meals

The business department asked you to analyze the different meal consumption, in order to optimize the revenue.

1. Provide a list of meals and how much of them were sold overall, when the most sold meal is on top and the least sold is on the bottom.
2. The business people are concerned with the least sold meal. They don't understand why people are not buying it. maybe the items in it are not popular?  
write a query to help them find out if people order the meal items separately instead of buying the meal, meaning provide the amount that include the meal, minus the amount of orders where each item of this meal is separately included.
3. As you probably already found out, for some reason, people prefer to order the items separately than order the meal. Maybe it's cheaper? Are there more such redundant meals?  
Write a query to find all meals where the meal doesn't cost less than the sum of price for each item in the meal. Only present the meal names.

## Part 3- Remodeling

### Question 6: Data Modeling

In order to resolve this issue of meals that aren't appealing, The accounting team has requested to enhance the meals table with the total items' cost for evaluation.

Write a query to alter the *menu\_meal* table such that it will include an integer *raw\_cost* column, which will store the accumulative cost of all items in the meal.

### Question 7: Updating the new column

Write SQL queries (in different files- q6.1, ...) to update the new *raw\_cost* column, for all meals that you found in question 3.3. The queries must be **the same** except for the specific names and ID's of the different meals.

GOOD LUCK :)