Homework Assignment 2: BIU Shoes

Tal Fiskus
sorte411@gmail.com

Database Systems – Semester A 2025/2026
Due 30/12/2025

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 35 .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).

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.), or `q{n_m}.py` (for example, `q2_1.py`, `q2_2.py`, etc.) for each query in the question. That 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 the format:

Listing 1: Python file format example

```python
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:

Listing 2: Python file format example

```python
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()
    ))
    cursor.close()
    mydb.close()
```

Important Notes

- Do not use commands that have not yet been learned 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.) to make them readable (see the example format below).

Listing 3: SQL query example

```sql
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 shoe store, BIU Shoes. 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

Congratulations on landing your first job as a data analyst at **BIU Shoes**, a rising star company in the footwear world! From stylish everyday sneakers to rare, limited-edition collections, BIU Shoes has something for every shoe lover out there.
But here's the catch: as the company grows, it's becoming harder to keep track of everything—inventory levels, sizes, special collections, customer preferences, and even those big orders flying in from all over the globe. That's where you come in! We hired you especially to construct a powerful database to keep the store running smoothly.
You'll not only create tables and fill them with data but also write smart queries to answer important business questions. The success of BIU Shoes is now in your capable hands. Time to lace up your data skills and step into the spotlight!

Part 1: Construct the database

Question 1:

Create a database for BIU Shoes with the name: *biu_shoes* Write a query that does so.

Question 2:

Using the requirements provided by our architecture team.
Write proper queries (10) to create the following tables in the database.
Ensure you select the appropriate data types and constraints.

1. Stores shoe details:
   shoe : ($\underline{shoe\_id}$: INT, $shoe\_name$: $VARCHAR(31)$ $NOT\ NULL$, $price$: $SMALLINT\ NOT\ NULL$)

2. Sizes:
   size : ($\underline{size\_id}$: INT, $european\_number$: $TINYINT\ NOT\ NULL$, us_number : $TINYINT$)

3. Connection table of shoes and sizes:
   shoe_size : ($\underline{shoe\_id}$: INT, $\underline{size\_id}$: INT)

   Foreign Keys:
   shoe_size($shoe\_id$) $\rightarrow$ **shoe**($shoe\_id$)
   shoe_size($size\_id$) $\rightarrow$ **size**($size\_id$)

4. Special edition of upcoming sneakers releases:
   upcoming : ($\underline{special\_id}$: INT, $shoe\_id$: $INT\ NOT\ NULL$, $collection\_name$: $VARCHAR(31)$, $release\_date$: $DATETIME$)
   Foreign Keys:
   upcoming($shoe\_id$) $\rightarrow$ **shoe**($shoe\_id$)

5. Countries information:
   country : ($\underline{country\_id}$: INT, $country\_name$: $VARCHAR(63)$ $NOT\ NULL$)

6. Cities information:
   city : ($\underline{city\_id}$: INT, $city\_name$: $VARCHAR(63)$ $NOT\ NULL$, $country\_id$:

INT NOT NULL)
Foreign Keys:
city(*country_id*) $\rightarrow$ **country**(*country_id*)

7. Customers details:
   customer : (<u>*customer_id*</u> : *VARCHAR*(15) *CHECK LENGTH EXACTLY* 9, *first_name* : *VARCHAR*(31) *NOT NULL*, *last_name* : *VARCHAR*(31) *NOT NULL*, *email* : *VARCHAR*(255) *UNIQUE NOT NULL*, *city_id* : *INT NOT NULL*)

   Foreign Keys:
   customer(*city_id*) $\rightarrow$ **city**(*city_id*)

   Tip:
   Use the save word LENGTH in the CHECK constraint when creating the table
   W3Schools Length

8. All of the company orders:
   company_order : (<u>*order_id*</u> : *INT*
   order_date : *DATETIME NOT NULL*)

9. Connection table of shoes and orders:
   order_shoe : (<u>*order_id*</u> : *INT*, <u>*shoe_id*</u> : *INT*)

   Foreign Keys:
   order_shoe(*order_id*) $\rightarrow$ **company_order**(*order_id*)
   order_shoe(*shoe_id*) $\rightarrow$ **shoe**(*shoe_id*)

10. Connection table of shoes and orders: **order_customer** : (<u>*order_id*</u> : *INT*, <u>*customer_id*</u> : *VARCHAR*(15))
    Foreign Keys:
    order_customer(*order_id*) $\rightarrow$ **company_order**(*order_id*)
    order_customer(*customer_id*) $\rightarrow$ **customer**(*customer_id*)

Question 3:

A team member has handed you all the data of the company to add it to the database.

Write proper queries (10) to insert them into the different tables:

1. The shoes the company has:
`(1, 'Air CS 0/1', 150), (2, 'Yeezy Gauss 360', 220), (3, '1B All Star', 60), (4, 'Jordan 1 Engineering', 170), (5, 'BIU Superstar', 90), (6, 'Lunar Glow', 200), (7, 'HyperFlux', 180), (8, 'Nebula Runner', 140), (9, 'Quantum Charge', 250), (10, 'Echo Boost', 160), (11, 'Storm Runner', 130), (12, 'Apex Boost', 190), (13, 'Velocity Wave', 170), (14, 'Stride Master', 100), (15, 'Trail Blazer', 200), (16, 'Infinity Glide', 180), (17, 'Solar Charge', 150), (18, 'Pace Leader', 120), (19, 'Canyon Racer', 250), (20, 'Sky High', 210)`

2. The sizes available:
`(1, 38, 6), (2, 39, 7), (3, 40, 8), (4, 41, 9), (5, 42, 10), (6, 43, 11), (7, 44, 12), (8, 45, 13), (9, 46, 14), (10, 47, 15), (11, 36, 4), (12, 37, 5), (13, 48, 16), (14, 49, 17), (15, 50, 18)`

3. Map the shoes to the sizes:
`(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (2, 3), (2, 4), (2, 5), (2, 6), (2, 7), (3, 4), (3, 5), (3, 6), (3, 7), (3, 8), (4, 6), (4, 7), (4, 8), (4, 9), (4, 10), (5, 5), (5, 6), (5, 7), (5, 8), (5, 9), (6, 4), (6, 5), (6, 6), (6, 7), (6, 8), (7, 5), (7, 6), (7, 7), (7, 8), (7, 9), (8, 6), (8, 7), (8, 8), (8, 9), (8, 10), (9, 7), (9, 8), (9, 9), (9, 10), (9, 11), (10, 8), (10, 9), (10, 10), (10, 11), (10, 12)`

4. Upcoming sneaker releases:
`(1, 1, 'Air Max Day 2025', '2025-03-26 10:00:00'), (2, 1, 'Spring Breeze Collection', '2025-05-15 12:00:00'), (3, 2, 'Yeezy Glow Pack', '2025-06-15 09:00:00'), (4, 2, 'Summer Neon Edition', '2025-08-10 11:00:00'), (5, 4, 'Jordan Anniversary', '2025-08-22 12:00:00'), (6, 4, 'Jordan Lunar Collection', '2025-10-05 14:00:00'), (7, 6, 'Lunar Bright Collection', '2025-09-30 08:00:00'), (8, 7, 'HyperFlux Supreme', '2025-10-10 15:00:00'), (9, 9, 'Quantum Special Edition', '2025-11-05 18:00:00'), (10, 9, 'Winter Starlight Pack', '2025-12-20 20:00:00'), (11, 10, 'Echo Runner Series',`

’2025-01-18 09:00:00’), (12, 10, ’Trail Explorer Edition’, ’2025-07-25 17:00:00’)

5. The countries where the company operates:
   (1, ’Israel’), (2, ’Canada’), (3, ’United States’), (4, ’Germany’), (5, ’France’), (6, ’Japan’), (7, ’Australia’), (8, ’Italy’), (9, ’Spain’), (10, ’Brazil’)

6. Cities in each country:
   (1, ’Jerusalem’, 1), (2, ’Tel Aviv’, 1), (3, ’Haifa’, 1), (4, ’Toronto’, 2), (5, ’Vancouver’, 2), (6, ’Montreal’, 2), (7, ’Los Angeles’, 3), (8, ’New York’, 3), (9, ’Chicago’, 3), (10, ’Houston’, 3), (11, ’Berlin’, 4), (12, ’Munich’, 4), (13, ’Frankfurt’, 4), (14, ’Hamburg’, 4), (15, ’Paris’, 5), (16, ’Lyon’, 5), (17, ’Marseille’, 5), (18, ’Tokyo’, 6), (19, ’Osaka’, 6), (20, ’Kyoto’, 6), (21, ’Sydney’, 7), (22, ’Melbourne’, 7), (23, ’Brisbane’, 7), (24, ’Rome’, 8), (25, ’Milan’, 8), (26, ’Naples’, 8), (27, ’Madrid’, 9), (28, ’Barcelona’, 9), (29, ’Seville’, 9), (30, ’Rio de Janeiro’, 10), (31, ’São Paulo’, 10), (32, ’Brasilia’, 10)

7. Customer details:
   (’123456789’, ’John’, ’Doe’, ’john.doe@example.com’, 1), (’987654321’, ’Jane’, ’Smith’, ’jane.smith@example.com’, 2), (’112233445’, ’Alice’, ’Brown’, ’alice.brown@example.com’, 3), (’223344556’, ’Bob’, ’White’, ’bob.white@example.com’, 4), (’334455667’, ’Eve’, ’Green’, ’eve.green@example.com’, 5), (’445566778’, ’Chris’, ’Black’, ’chris.black@example.com’, 6), (’556677889’, ’Anna’, ’Taylor’, ’anna.taylor@example.com’, 7), (’667788990’, ’Max’, ’Johnson’, ’max.johnson@example.com’, 8), (’778899001’, ’Lily’, ’Adams’, ’lily.adams@example.com’, 9), (’889900112’, ’Oscar’, ’Hill’, ’oscar.hill@example.com’, 10)

8. Company orders:
   (1, ’2025-01-15 14:30:00’), (2, ’2025-02-10 10:15:00’), (3, ’2025-03-05 16:45:00’), (4, ’2025-04-12 09:00:00’), (5, ’2025-05-20 11:00:00’), (6, ’2025-06-01 13:30:00’), (7, ’2025-07-10 15:45:00’), (8, ’2025-08-15 17:00:00’), (9, ’2025-09-05 19:30:00’), (10, ’2025-10-22 08:15:00’), (11, ’2025-11-01 12:30:00’), (12, ’2025-12-10

```
14:45:00'), (13, '2025-02-18 10:00:00'), (14, '2025-03-20 08:15:00'),
(15, '2025-04-22 16:30:00')
```

9. Mapping orders to shoes:
```
(1, 1), (1, 2), (2, 3), (3, 4), (4, 5), (5, 6), (6, 7), (7,
8), (8, 9), (9, 10)
```

10. Mapping orders to customers:
```
(1, '123456789'), (2, '987654321'), (3, '112233445'), (4, '223344556'),
(5, '334455667'), (6, '445566778'), (7, '556677889'), (8, '667788990'),
(9, '778899001'), (10, '889900112')
```

Question 4:

The company has decided to expand to the UK market.
Write a query that alter the *size* table to add the *uk_number* $TINYINT$ column.

Question 5:

The company has started offering pre-orders for special edition sneakers.
Write a query that add a *pre_order_available* column to the *upcoming* table to indicate whether pre-orders are enabled, making it a single-bit type and default value 0.

Question 6:

European Size	US Size	**UK Size**
38	6	5
39	7	6
41	9	7
42	10	8
43	11	9

Table 1: Shoe Size Chart

The company wants to populate the *uk_number* column in the *size* table based on the above size chart.
Write SQL queries (5) to update the *uk_number* column values.

Question 7 - Revenue Insights:

The marketing team at BIU Shoes is curious about which customers generate the most revenue for the company. They want a clear breakdown of the total revenue each customer has contributed, based on their purchases.
Write a query to calculate the **total revenue generated by each customer**, returning their $first\_name$, *last_name*, and total amount spent if at all (total amount spent must be a number).

Question 8 - Price Analysis by Size:

The product team wants to better understand the price distribution of shoes for different sizes. This analysis will help them determine if certain sizes have higher-priced shoes.
Write a query to find the **average price of shoes available in each size**, returning the European and US sizes among with the average, and order them from highest to lowest (by the average).

Question 9 - Shoe Availability:

To ensure their inventory is well-stocked, the operations team needs to know how many size options are available for each shoe. This will help them identify shoes with limited size options and adjust their stock accordingly.
Write a query to list **all shoe names with the count of sizes available for each** including shoes that don't have any size.

Question 10 - Inventory and Marketing Integration:

The marketing team is planning a major campaign that combines regular inventory with upcoming special releases. They need a consolidated list of all available shoe names and upcoming collections to design their promotional materials. Write a query to **combine the list of shoe names from the inventory with the list of special upcoming collection names under** *name***column**, and include an additional column, `source`, indicating whether the entry is from the 'Inventory' or an 'Upcoming Release' (a special edition of upcoming sneakers releases).
(Tip: we can hard code a value as a column using aliasing)

Question 11 - Sales Summary by Shoe:

The finance team wants a high-level summary of sales for each shoe to evaluate product performance. They need a summary that shows the total revenue generated by each shoe (that got sold), across all orders.
Write a query to **create a view (*total_sales_per_shoe*) summarizing total sales per shoe**, including the shoe ID, name, and total revenue.
Write another query to display the entire view data.

Question 12 - Identifying Unsold Shoes:

The BIU Shoes team has noticed that some shoes remain unsold for extended periods, tying up valuable inventory space. They need a list of shoes that have not yet been ordered by any customer.
Write a query to **find all shoes (names) that have not been ordered** without duplicates.