Task 2:

Using the database from Task 1, write queries in natural language, relational algebra and SQL that contains:

1. Joining more than two tables
2. Aggregate function
3. Nested query

1. Joining more than two tables

Natural Language

Which customers from Greece have purchased both Intel i7 14700K and Nvidia RTX 4060 and which employee from which branch handled the sales?

Relational Algebra

$$
\begin{aligned}
&\pi(\text{customer\_id},\text{employee\_id},\text{branch\_id})\ (\\
&\ \gamma(\text{customer\_id},\text{employee\_id},\text{branch\_id};\ \text{COUNT(DISTINCT tech\_name)}=2)\ (\\
&\ \ \sigma(\text{country='Greece'} \wedge (\text{tech\_name='Intel i7 14700K'} \vee \text{tech\_name='Nvidia RTX 4060'}))\ (\\
&\ \ \ \text{customer} \bowtie \text{bill} \bowtie \text{bill\_item} \bowtie \text{items} \bowtie \text{employees} \bowtie \text{branch}\\
&\ \ )\\
&\ )\\
&)
\end{aligned}
$$

SQL

```
SELECT c.customer_id, e.employee_id, br.branch_id
FROM customer as c
JOIN bill as b ON c.customer_id = b.customer_id
JOIN bill_item as bi ON b.bill_id = bi.bill_id
JOIN items as i ON bi.item_id = i.item_id
JOIN employees as e ON e.employee_id = b.ref_employee_id
JOIN branch as br ON br.branch_id = e.branch_id
WHERE c.country = 'Greece'
  AND (i.tech_name = 'Intel i7 14700K'
  OR i.tech_name = 'Nvidia RTX 4060')
GROUP BY c.customer_id, e.employee_id, br.branch_id
HAVING COUNT (DISTINCT i.tech_name) = 2;
```

2. Aggregate function

Natural Language

What is the maximum sales amount that each employee has achieved?

Relational Algebra

```
π(employee_id, MAX(amount)) (
  γ(ref_employee_id; MAX(amount)) (
    employees ⋈ bill
  )
)
```

SQL

```
SELECT e.employee_id, max(b.amount)
FROM employees as e
JOIN bill as b ON b.ref_employee_id = e.employee_id
GROUP BY b.ref_employee_id
```

3. Nested query

Natural Language

Which employees work in branches that have more sales than the average sales per branch?

Relational Algebra

```
π(employee_id, branch_id) (
  γ(employee_id) (
    σ(branch_id ∈
      π(branch_id) (
        σ(SUM(amount) >
          AVG(branch_sales) (
            ρ(branch_sales/SUM(amount)) (
              γ(branch_id; SUM(amount)) (bill)
```

```
          )
        )
      ) (
        γ(branch_id; SUM(amount)) (bill)
      )
    )
  ) (
    employees ⋈ bill ⋈ branch
  )
 )
)
```

SQL

```
SELECT e.employee_id, e.branch_id
FROM employees as e
JOIN bill as b ON b.branch_id = e.branch_id
JOIN branch as br ON br.branch_id = b.branch_id
WHERE e.branch_id IN (
                    SELECT branch_id
                    FROM bill
                    GROUP BY branch_id
                    HAVING SUM(amount) > (
                    SELECT AVG(branch_sales)
                    FROM (SELECT SUM(amount) as branch_sales
                        FROM bill
                            GROUP BY branch_id)
                    )
)
GROUP BY e.employee_id;
```

σ (sigma - selection/WHERE)
π (pi - projection/SELECT)
⋈ (join)
∧ (AND)
∨ (OR)
γ (gamma - GROUP BY)