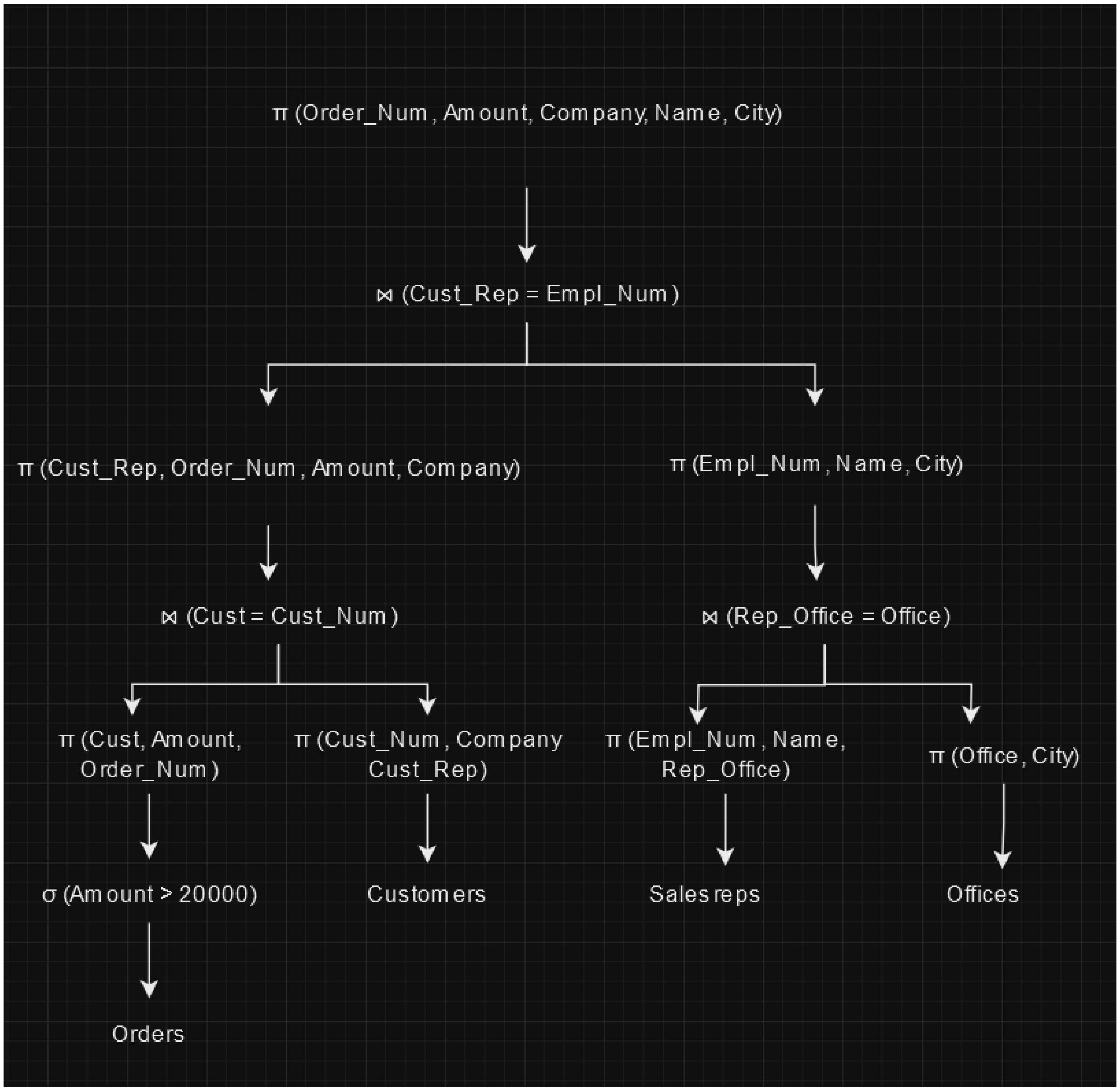
Lab 6

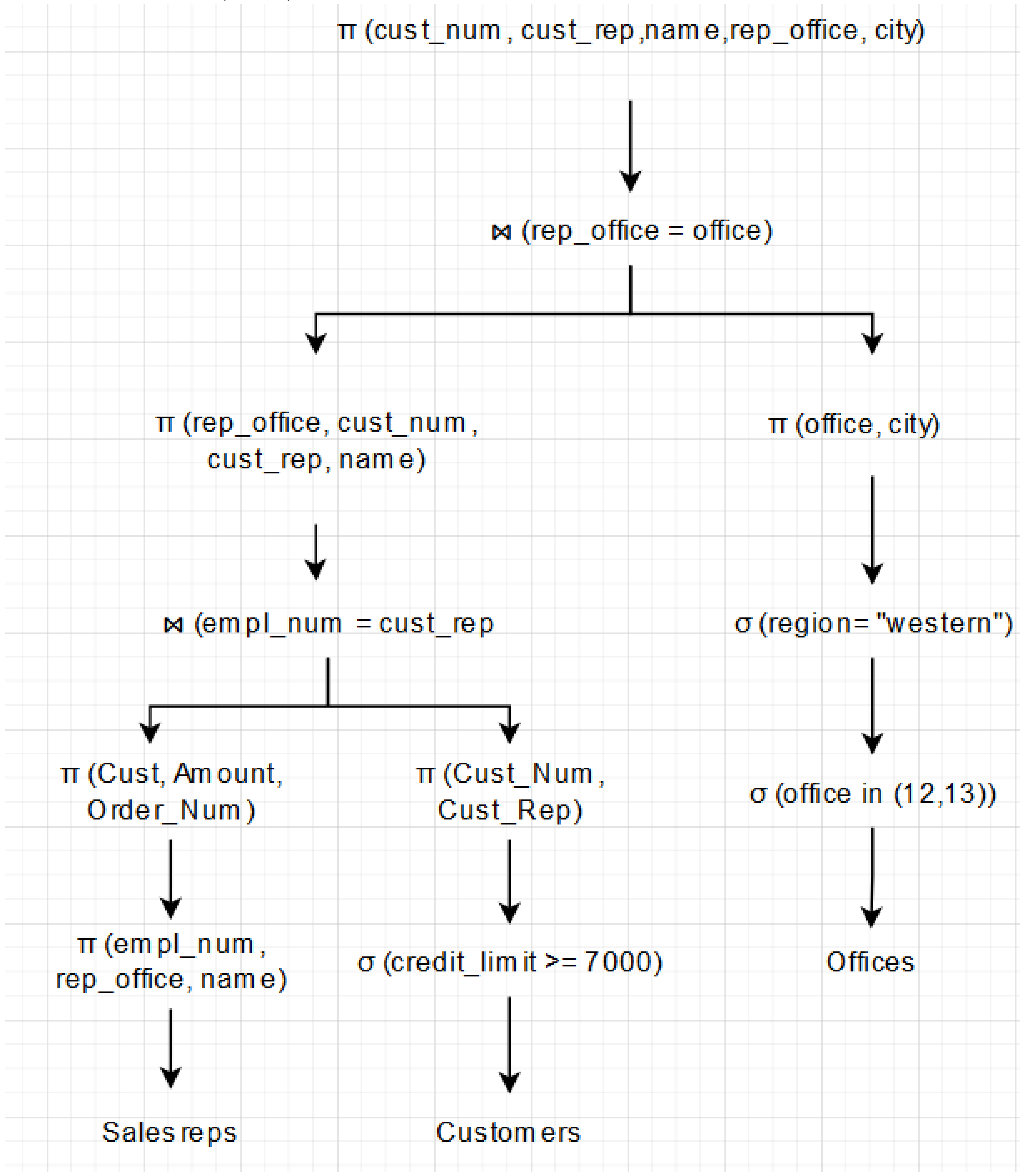
1) Draw the optimization tree for the following query.

    SELECT Order_Num, Amount, Company, Name, City
    FROM Orders, Customers, Salesreps, Offices
    WHERE Cust = Cust_Num
    AND Cust_Rep = Empl_Num
    AND Rep_Office = Office
    AND Amount > 20000


π (Order_Num, Amount, Company, Name, City)
⋈ (Cust_Rep = Empl_Num)
π (Cust_Rep, Order_Num, Amount, Company)
π (Empl_Num, Name, City)
⋈ (Cust = Cust_Num)
⋈ (Rep_Office = Office)
π (Cust, Amount, Order_Num)
π (Cust_Num, Company Cust_Rep)
π (Empl_Num, Name, Rep_Office)
π (Office, City)
σ (Amount > 20000)
Customers
Salesreps
Offices
Orders

2) Draw the optimization tree for the following query.

```
SELECT cust_num, cust_rep, name, rep_office, city
FROM Customers, Salesreps, Offices
WHERE empl_num = cust_rep, and rep_office = office
AND credit_limit >= 7000
And region = “western”
And office in (12, 13)
```



3) Some of frequent attacks on the database include: Unauthorized Privilege Escalation, Privilege Abuse, Denial of Service, and Weak authentication. Explain Privilege Abuse and Weak Authentication in a few lines.

Privilege Abuse:

occurs when an authorized user intentionally misuses their access rights. For instance, a database administrator could change grades, alter financial records, or access restricted data for personal gain. This attack is internal and depends on the trusted user's actions.

Weak Authentication:

is vulnerabilities in the authentication mechanism, such as weak or easily guessable passwords. If the user authentication system is not secure, an attacker can steal or mimic the login credentials of a legitimate user, gaining unauthorized access to the database and its resources.

4) One form of Security Injection Attack is called "SQL manipulation". Suppose we have the following query

```
SELECT *
FROM users
WHERE username = 'user_input' AND password = 'password_input’;
```

How can an attacker insert a code in the above query to retrieve the database data without authorization?

An attacker can exploit the provided query through SQL injection by manipulating the input fields to bypass authentication and retrieve unauthorized data. The vulnerable query directly embeds user input into the SQL statement without proper sanitization. For example, the attacker can input ' OR '1'='1 as the username and ' OR '1'='1 as the password. When this input is substituted into the query, it becomes:

```
SELECT *
FROM users
WHERE username = '' OR '1'='1' AND password = '' OR '1'='1';.
```

The condition OR '1'='1' always evaluates to true, effectively bypassing the intended authentication logic. As a result, the query retrieves all rows from the users table, granting the attacker unauthorized access to sensitive data without needing valid credentials.

5) Rewrite the following query with a php code to prevent the attack

SELECT *
FROM users
WHERE username = 'user_input' AND password = 'password_input';

```
$stmt = $pdo->prepare("SELECT * FROM users
WHERE username = :username AND password = :password");

$stmt->execute(['username' => $user_input, 'password' => $password_input]);
```

6) What are the four types of "Code Injection"? Just name them (no need to explain them).

- SQL Injection
- Command Injection
- Script Injection (Cross-Site Scripting - XSS)
- Buffer Overflow

7) Provide examples of "Function Call Injection" where an attacker

a. can find the user id of a database user. Just provide the query.

```
' OR 1=1; SELECT USER() --
```

b. can find the version of the database. Just provide the query.

```
' OR 1=1; SELECT VERSION() --
```

c. can find the currently connected database.

```
' OR 1=1; SELECT DATABASE() --
```

d. can drop a table in the database.

```
' OR 1=1; DROP TABLE table_name --
```

8) How to prevent the Function call Injection? Give a PHP code with pdo□Prepare function that blocks attacker from function call injection.

```
$stmt = $pdo->prepare("SELECT * FROM employees WHERE first_name =
:first_name");
```

```
$stmt->execute(['first_name' => $user_input]);
```

9) What are the typical risks associated with SQL Injection. Just name them, no need to provide any explanation.

    - Database Fingerprinting
    - Denial of Service
    - Bypassing Authentication
    - Identifying Injectable Parameters
    - Executing Remote Commands
    - Performing Privilege Escalation

10) What are the three forms of protections against SQL Injection. Give example queries of each method.

    - Bind Variables (Parameterized Queries)

```
PreparedStatement stmtconn_PrepareStatement
(Select * FROM EMPLOYEE
WHERE EMPLOUYEE_ID=? AND PASSWORD=?”);
Stmt.setString1.employeee_id;
Stmt.setString2.password;
```

    - Filtering input (Input Validation)

```
cursor.execute("SELECT * FROM users WHERE username = ?",
(input_username_escaped,))
```

    - Function Security

```
cursor.execute("SELECT * FROM users WHERE username = ?",
(input_username_escaped,))
```