

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Звіт

з лабораторної роботи № 3

з дисципліни «Ефективність та якість архітектурних рішень
інформаційних систем»

Виконав:

студент групи ІКМ-М225В

Суліма Нікіта Володимирович

Перевірив:

аспірант каф. КМПС Хорошун Андрій Сергійович

Харків 2025

Реалізовано будівельник SQL-запитів, який дозволяє формувати SQL-інструкції для різних типів баз даних (PostgreSQL та MySQL) з використанням одного інтерфейсу.

Структура класів

Містить базові методи для побудови SQL-запиту:

```
class IQueryBuilder {
public:
    virtual IQueryBuilder* select(string fields) = 0;
    virtual IQueryBuilder* where(string condition) = 0;
    virtual IQueryBuilder* limit(int n) = 0;
    virtual string getSQL() = 0;
};
```

Цей інтерфейс забезпечує уніфікований спосіб побудови запитів.

Конкретні будівельники

PostgreSQLQueryBuilder

Реалізує побудову запитів для PostgreSQL:

```
class PostgreSQLQueryBuilder : public IQueryBuilder {
private:
    string query;
public:
    IQueryBuilder* select(string fields) override { ... }
    IQueryBuilder* where(string condition) override { ... }
    IQueryBuilder* limit(int n) override { ... }
    string getSQL() override { return query + ";;" }
};
```

MySQLQueryBuilder

Реалізує побудову запитів для MySQL:

```
class MySQLQueryBuilder : public IQueryBuilder {
private:
    string query;
public:
    IQueryBuilder* select(string fields) override { ... }
    IQueryBuilder* where(string condition) override { ... }
    IQueryBuilder* limit(int n) override { ... }
    string getSQL() override { return query + ";;" }
};
```

Клієнтський код

```

int main() {
    // PostgreSQL
    IQueryBuilder* pgBuilder = new PostgreSQLQueryBuilder();
    string pgSQL = pgBuilder->select("")
        ->where("age > 18")
        ->limit(10)
        ->getSQL();
    cout << "[PostgreSQL] " << pgSQL << endl;

    // MySQL
    IQueryBuilder* myBuilder = new MySQLQueryBuilder();
    string mySQL = myBuilder->select("id, name")
        ->where("status = 'active'")
        ->limit(5)
        ->getSQL();
    cout << "[MySQL] " << mySQL << endl;

    delete pgBuilder;
    delete myBuilder;
    return 0;
}

```

Клієнт створює будівельник (PostgreSQLQueryBuilder або MySQLQueryBuilder). Викликаються методи select(), where(), limit() — кожен додає частину запиту. В кінці викликається метод getSQL(), який повертає сформований запит. Код побудови не залежить від конкретного типу бази даних, тому його можна легко адаптувати для будь-якої СУБД.

Висновок

В ході лабораторної роботи було ознайомлено з патерном “Будівельник”, розділено логіку побудови запитів для PostgreSQL та MySQL, зберігаючи єдиний інтерфейс використання. Патерн “Будівельник” забезпечив гнучкість і розширюваність коду, дозволивши легко додавати нові типи будівельників без зміни клієнтського коду.