

МІНІСТЕРСТВО ОСВІТИ І НАУКИ І УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

Факультет прикладної математики
Кафедра програмного забезпечення комп’ютерних систем

Лабораторна робота №1

з дисципліни “ Організація баз даних ”

“Розробка Web-орієнтованої інформаційно-пошукової системи”

Виконав:
студент Лецик Андрій
Перевірив:
к. т. н. Петрашенко А. В.

Київ 2014

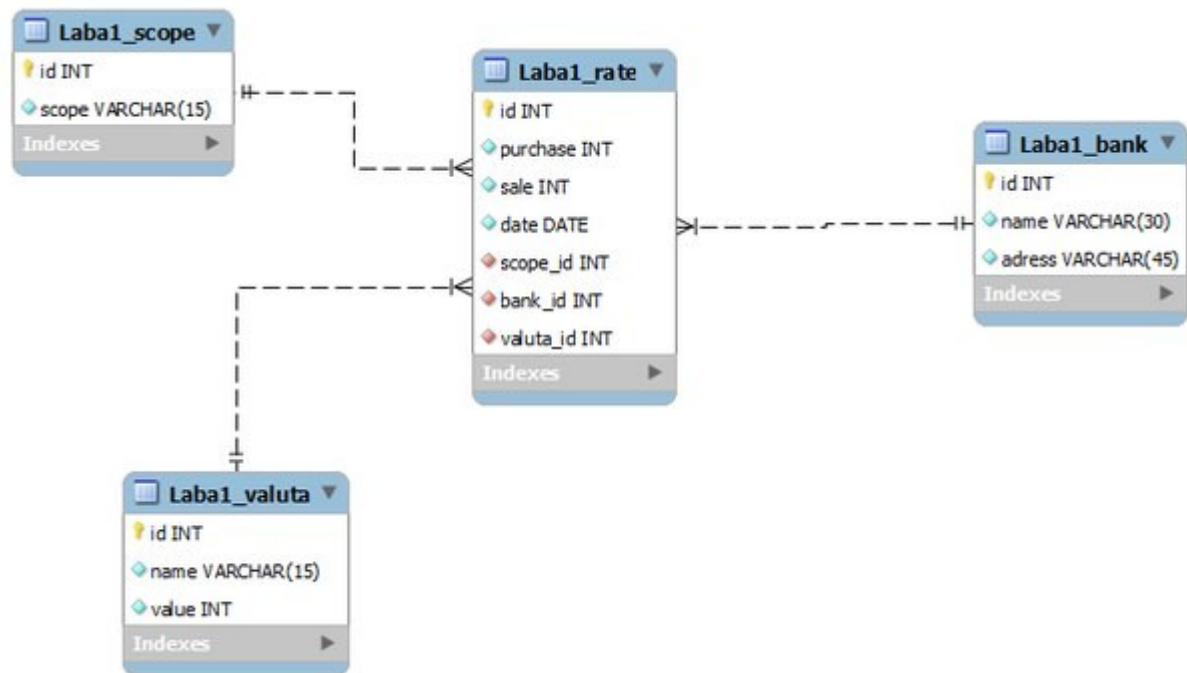
Варіант 12

Формат файлу — Jason

Пошук за атрибутами - діапазон чисел, перелічення

Повнотекстовий пошук - обов'язкове входження слова, ціла фраза

Графічне подання розробленої моделі «сутність-зв'язок»



Лістинг

```
class MyORM():
    def __init__(self, params):
        self.params = params
    def get_table_field(self):
        return ', '.join(self.table_fields)
    def get_params(self):
        fields_data = []
        for field in self.table_fields:
            field_data = '\'%s\' ' % self.params[field]
            fields_data.append(field_data)
        return ', '.join(fields_data)
    def insert(self):
        print ("INSERT INTO %s(%s) VALUES(%s);" %
              (self.table_name, self.get_table_field(), self.get_params()))
        execute_query("INSERT INTO %s(%s) VALUES(%s);" %
                      (self.table_name, self.get_table_field(), self.get_params()), 'false')
        self.get_params()
    def delete(self, _where):
        execute_query("DELETE FROM %s WHERE (%s);" %
                      (self.table_name, _where), 'false')
    def delete_all(self):
        execute_query("DELETE FROM %s;" %
                      (self.table_name), 'false')
    def select_id(self, id):
        result = execute_query("SELECT * FROM %s where id = %s;" % (self.table_name, id) ,
        'true')
        print ("select * from %s where id = 1;" % self.table_name)
        return result
    def update(self, data, id):
        result = execute_query("UPDATE %s set %s where id = %s;" % (self.table_name, data,
        id), 'false')
        return result
    def parse_XML_file(self):
        print self.table_name
        if (os.path.isfile('Laba1/%s.xml' % (self.table_name))):
            tree = ET.parse('Laba1/%s.xml' % (self.table_name))
            print 'yes'

            root = tree.getroot()

            self.delete_all()
            for r in root:
                t = ""
                for field in self.table_fields:
                    t += "\"" + r.find(field).text + "\", "
                t = t[0:-2]
```

```

        self.params = t
        print t
        execute_query("INSERT INTO %s(%s) VALUES(%s);" %
            (self.table_name, self.get_table_field(), t), 'false')
def search_field_value(self, field, value):
    print '1111111111111111'
    result = execute_query("SELECT * FROM %s where %s = \"%s\";" %
        (self.table_name, field, value), 'true')
    return result
def search_field_beetween(self, field, value1, value2):
    #print ("SELECT * FROM " + self.table_name + " where " + field + " BEETWEEN " +
str(value1) + " AND " + str(value2))
    result = execute_query("SELECT * FROM %s where %s >= %s AND %s <= %s;" %
        (self.table_name, field, value1, field, value2), 'true')
    return result
def fulltext_search(self, text, field, table):
    print ("SELECT * FROM %s WHERE MATCH (%s) AGAINST (\'+%s\' in BOOLEAN
MODE);"
        % (table, field, text))
    result = execute_query("SELECT * FROM %s WHERE MATCH (%s) AGAINST (\'+
%s\' in BOOLEAN MODE);"
        % (table, field, text), 'true')
    return result
def result_to_norm_look(self, _str):
    _str = str(_str)
    _str = _str[2:-3]
    return _str

```

Скріншоти

