

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»
КАФЕДРА СИСТЕМНОГО ПРОГРАМУВАННЯ І СПЕЦІАЛІЗОВАНИХ
КОМП'ЮТЕРНИХ СИСТЕМ

ЛАБОРАТОРНА РОБОТА №2

З дисципліни «Організація баз даних»

На тему: «Розробка Web-орієнтованої інформаційно-пошукової
системи»

Виконала

Студентка групи КВ-32

Ахмедова Д.Н.

ЗАВДАННЯ

1. Розробити модель «сутність-зв'язок» предметної галузі, обраної студентом, відповідно до пункту «Вимоги до ER-моделі».
2. Перетворити розроблену модель у структуру бази даних MySQL.
3. Виконати нормалізацію бази даних до 3НФ, затвердити базу даних у викладача.
4. Реалізувати функціональні вимоги, наведені нижче

Вимоги до ER-моделі:

1. Модель предметної галузі має відповідати схемі «зірка», що складається з декількох сутностей «виміри» та однієї сутності «факти»
2. Кількість атрибутів у кожній сутності: від двох до семи.
3. Сутності мають включати атрибути для коректної реалізації особливостей пошуку, наведених у варіанті.

Функціональні вимоги:

1. Для усіх сутностей «виміри» реалізувати внесення даних із зовнішнього файлу. Формат файлу (JSON, XML, CSV) обрати згідно з варіантом. Внесення даних передбачити з повною заміною існуючих даних у сутностях (таблицях) з урахуванням залежностей у сутності «факти».
2. Для сутності «факти» реалізувати операції вставлення, вилучення та редагування окремих записів, використовуючи Web-орієнтований інтерфейс користувача.
3. Забезпечити реалізацію пошуку за двома-трьома атрибутами кожної з сутностей «виміри»: для числових атрибутів – у рамках діапазону, для рядкових – як перелічення, для логічного типу – значення True/False, для дат – у рамках діапазону дат.
4. Забезпечити реалізацію повнотекстового пошуку по будь-якому рядковому або текстовому атрибуту бази даних засобами MySQL в режимі BooleanMode.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

№	Формат файлу	Пошук за атрибутами	Повнотекстовий пошук
21	XML	діапазон чисел, перелічення	слово не входить, ціла фраза

ФРАГМЕНТИ КОДУ

Db.py

```
from django.shortcuts import render
from django.http import HttpResponse, HttpResponseRedirect
import datetime

from db import DB_1
from .forms import *
import MySQLdb as mdb1

def filelist(request):

    db = DB_1()
    File = db.get_file_list()
    Cloud = db.get_Cloud_storage_list()
    User = db.get_user_list()
    form = Diapason()
    form1 = Diapason_1()
    search = Search()
    if request.method == 'POST':

        if request.POST['key'] == 'Diapason':
            form = Diapason(request.POST)
            D = db.diapason(request.POST['first'], request.POST['last'])
            return render(request, 'filelist.html', {'File': File, 'Cloud_storage': Cloud, 'user': User,
            'form': form, 'Dia': D, 'form1': form1, 'search': search},)

        if request.POST['key'] == 'Diapason_1':
            form1 = Diapason_1(request.POST)
            D_1 = db.diapason_1(request.POST['first'], request.POST['last'])
            return render(request, 'filelist.html', {'File': File, 'Cloud_storage': Cloud, 'user': User,
            'Dia_1': D_1, 'form1': form1, 'form': form, 'search': search},)

        if request.POST['key'] == 'Reset':
            db.load()
            return HttpResponseRedirect('/student/file/')

        if request.POST['key'] == 'Poisk':
            Poisk_user = db.poisk_1(request.POST['Poisk'])
            print request.POST
            return render(request, 'filelist.html', {'File': File, 'Cloud_storage': Cloud, 'user': User,
            'p': Poisk_user, 'form1': form1, 'form': form, 'search': search })

        if request.POST['key'] == 'Poisk_1':
            Poisk_cloud_name = db.poisk_2(request.POST['Poisk_1'])
            print request.POST
            return render(request, 'filelist.html', {'File': File, 'Cloud_storage': Cloud, 'user': User,
            'c': Poisk_cloud_name, 'form': form, 'form1': form1, 'search': search})

        if request.POST['key'] == 'search':
            search_1 = db.search_1(request.POST['search'], request.POST['search_1'])
            print request.POST
            return render(request, 'filelist.html', {'File': File, 'Cloud_storage': Cloud, 'user': User,
            'form': form, 'form1': form1, 'search': search, 'search_1': search_1})
```

```

        return render(request, 'filelist.html', {'File': File, 'Cloud_storage': Cloud, 'user': User, 'form':
form, 'form1': form1, 'search': search})

def addfile(request):
    if request.method == 'POST':
        form = FileForm(request.POST)
        if form.is_valid():
            db = DB_1()
            db.save_file(request.POST['name_f'], request.POST['size'], request.POST['type'],
request.POST['user_name_u'], request.POST['Cloud_storage_idstorage'])
            return HttpResponseRedirect('/student/file/')
        else:
            form = FileForm()
            return render(request, 'add.html', {'form': form})

def adduser(request):
    if request.method == 'POST':
        form = userForm(request.POST)
        if form.is_valid():
            db = DB_1()
            print request.POST
            db.save_user(request.POST['name_u'], request.POST['B_day'], request.POST['year'])
            return HttpResponseRedirect('/student/file/')
        else:
            form = userForm()
            return render(request, 'add.html', {'form': form})

def addCloud_storage(request):
    if request.method == 'POST':
        form = CloudForm(request.POST)
        if form.is_valid():
            db = DB_1()
            print request.POST
            db.save_cloud(request.POST['name'], request.POST['date_s_add'], request.POST['date_s_change'])
            return HttpResponseRedirect('/student/file/')
        else:
            form = CloudForm()
            return render(request, 'add.html', {'form': form})

def change_u(request):
    if request.method == 'POST':
        user = user_id_edit(request.POST)

        if user.is_valid():
            form = userForm_Edit()
            return render(request, 'change.html', {'form': form, 'name_u_edit': request.POST['Name_key']})
        form = userForm_Edit(request.POST)
        if form.is_valid():
            db = DB_1()
            db.change_user(request.POST['name_u'], request.POST['B_day'], request.POST['year'],
request.POST['key'])
            return HttpResponseRedirect('/student/file/')
        else:
            user = user_id_edit()
            return render(request, 'change.html', {'user': user})

def change_f(request):
    if request.method == 'POST':
        file = File_id_edit(request.POST)

        if file.is_valid():
            form = FileForm_Edit()
            return render(request, 'change_file.html', {'form': form, 'name_f_edit':
request.POST['Name_key_f']})
        form = FileForm_Edit(request.POST)
        if form.is_valid():
            db = DB_1()
            db.change_file(request.POST['name_f'], request.POST['size'], request.POST['type'],
request.POST['user_name_u'], request.POST['Cloud_storage_idstorage'], request.POST['key'])

            return HttpResponseRedirect('/student/file/')
        else:
            file = File_id_edit()
            return render(request, 'change_file.html', {'file': file})

def delet_f(request):
    if request.method == 'POST':

```

```

file = File_del(request.POST)

if file.is_valid():
    db = DB_1()
    db.delete(request.POST['idFile'])

    return HttpResponseRedirect('/student/file/')
else:
    file = File_del()
return render(request, 'delete.html', {'file': file})

```

views.py

```

from django.shortcuts import render
from django.http import HttpResponseRedirect
import datetime

from db import DB_1
from .forms import *
import MySQLdb as mdb1

def filelist(request):

    db = DB_1()
    File = db.get_file_list()
    Cloud = db.get_Cloud_storage_list()
    User = db.get_user_list()
    form = Diapason()
    form1 = Diapason_1()
    search = Search()
    if request.method == 'POST':

        if request.POST['key'] == 'Diapason':
            form = Diapason(request.POST)
            D = db.diapason(request.POST['first'],request.POST['last'])
            return render(request, 'filelist.html', {'File': File , 'Cloud_storage': Cloud, 'user':User,
'form': form, 'Dia': D,'form1': form1, 'search': search},)

        if request.POST['key'] == 'Diapason_1':
            form1 = Diapason_1(request.POST)
            D_1 = db.diapason_1(request.POST['first'],request.POST['last'])
            return render(request, 'filelist.html', {'File': File , 'Cloud_storage': Cloud, 'user':User,
'Dia_1': D_1,'form1': form1, 'form': form, 'search': search},)

        if request.POST['key'] == 'Reset':
            db.load()
            return HttpResponseRedirect('/student/file/')

        if request.POST['key'] == 'Poisk':
            Poisk_user = db.poisk_1(request.POST['Poisk'])
            print request.POST
            return render(request, 'filelist.html', {'File': File , 'Cloud_storage': Cloud, 'user':User,
'p':Poisk_user,'form1': form1,'form': form, 'search': search })

        if request.POST['key'] == 'Poisk_1':
            Poisk_cloud_name = db.poisk_2(request.POST['Poisk_1'])
            print request.POST
            return render(request, 'filelist.html', {'File': File , 'Cloud_storage': Cloud, 'user':User,
'c':Poisk_cloud_name,'form': form,'form1': form1, 'search': search})

        if request.POST['key'] == 'search':
            search_1 = db.search_1(request.POST['search'],request.POST['search_1'])
            print request.POST
            return render(request, 'filelist.html', {'File': File , 'Cloud_storage': Cloud, 'user':User,
'form': form, 'form1': form1, 'search': search, 'search_1': search_1})

    return render(request, 'filelist.html', {'File': File , 'Cloud_storage': Cloud, 'user':User, 'form':
form, 'form1': form1, 'search': search})

def addfile(request):
    if request.method == 'POST':
        form = FileForm(request.POST)
        if form.is_valid():
            db = DB_1()
            db.save_file(request.POST['name_f'], request.POST['size'], request.POST['type'],
request.POST['user_name_u'], request.POST['Cloud_storage_idstorage'])

```

```

        return HttpResponseRedirect('/student/file/')
    else:
        form = FileForm()
        return render(request, 'add.html', {'form': form})

def adduser(request):
    if request.method == 'POST':
        form = userForm(request.POST)
        if form.is_valid():
            db = DB_1()
            print request.POST
            db.save_user(request.POST['name_u'], request.POST['B_day'], request.POST['year'])
            return HttpResponseRedirect('/student/file/')
        else:
            form = userForm()
            return render(request, 'add.html', {'form': form})

def addCloud_storage(request):
    if request.method == 'POST':
        form = CloudForm(request.POST)
        if form.is_valid():
            db = DB_1()
            print request.POST
            db.save_cloud(request.POST['name'], request.POST['date_s_add'], request.POST['date_s_change'])
            return HttpResponseRedirect('/student/file/')
        else:
            form = CloudForm()
            return render(request, 'add.html', {'form': form})

def change_u(request):
    if request.method == 'POST':
        user = user_id_edit(request.POST)

        if user.is_valid():
            form = userForm_Edit()
            return render(request, 'change.html', {'form': form, 'name_u_edit': request.POST['Name_key']})
        form = userForm_Edit(request.POST)
        if form.is_valid():
            db = DB_1()
            db.change_user(request.POST['name_u'], request.POST['B_day'], request.POST['year'],
request.POST['key'])
            return HttpResponseRedirect('/student/file/')
        else:
            user = user_id_edit()
            return render(request, 'change.html', {'user': user})

def change_f(request):
    if request.method == 'POST':
        file = File_id_edit(request.POST)

        if file.is_valid():
            form = FileForm_Edit()
            return render(request, 'change_file.html', {'form': form, 'name_f_edit':
request.POST['Name_key_f']})
        form = FileForm_Edit(request.POST)
        if form.is_valid():
            db = DB_1()
            db.change_file(request.POST['name_f'], request.POST['size'], request.POST['type'],
request.POST['user_name_u'], request.POST['Cloud_storage_idstorage'], request.POST['key'])

            return HttpResponseRedirect('/student/file/')
        else:
            file = File_id_edit()
            return render(request, 'change_file.html', {'file': file})

def delet_f(request):
    if request.method == 'POST':
        file = File_del(request.POST)

        if file.is_valid():
            db = DB_1()
            db.delete(request.POST['idFile'])

            return HttpResponseRedirect('/student/file/')
        else:
            file = File_del()
            return render(request, 'delete.html', {'file': file})

```

СКРИНШОТЫ

All information

ID	Name file	Size file	Type file	Login	Year	B-day	Storage id	Name storage	Data add	Data change
15	James Arthur - Recovery	545	MP3	Miny	19	Feb. 2, 2000	45	Mini	Oct. 10, 2014, 9:21 p.m.	Oct. 10, 2014, 9:21 p.m.
16	James.	4654	IMG	U_D	33	Feb. 2, 1996	47	Darina	Oct. 10, 1996, 9:21 p.m.	Oct. 10, 1996, 9:21 p.m.

Search in general table

Search:
Search 1:

search

ID	Name file	Size file	Type file	Login	Year	B-day	Storage id	Name storage	Data add	Data change
16	James.	4654	IMG	U_D	33	Feb. 2, 1996	47	Darina	Oct. 10, 1996, 9:21 p.m.	Oct. 10, 1996, 9:21 p.m.

Add file

Change file

Delete file

User

Login	B-day	Year
Miny	Feb. 2, 2000	19
U_D	Feb. 2, 1996	33
U_S	Feb. 2, 2015	20

Add user

Change user

Search in user

Miny

Poisk

First:
Last:

Diapason

Cloud storage

ID	Name	Data add	Data change
45	Mini	Oct. 10, 2014, 9:21 p.m.	Oct. 10, 2014, 9:21 p.m.
46	Sasha	Oct. 10, 2000, 9:21 p.m.	Oct. 10, 2000, 9:21 p.m.
47	Darina	Oct. 10, 1996, 9:21 p.m.	Oct. 10, 1996, 9:21 p.m.

Add cloud

Search in Cloud storage

Mini

Poisk_1

First:
Last:

Diapason_1

Reset