**ЛАБОРАТОРНА РОБОТА №1**

З дисципліни «Організація баз даних»

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

|  |  |
| --- | --- |
| **Виконав**  Студент групи КВ-21  Бондарчук М.Ю.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | **Прийняв**  Доцент кафедри СПСКС  Замятін Д.С.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

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 Boolean Mode.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

|  |  |  |  |
| --- | --- | --- | --- |
| № | Формат файлу | Пошук за атрибутами | Повнотекстовий пошук |
| 2 | JSON | діапазон дат, логічний тип | слово не входить, ціла фраза |

ФРАГМЕНТИ КОДУ

Фрагмент зміни таблиці фактів

import MySQLdb

def remove\_transaction(transaction\_id):

db = MySQLdb.connect(host="localhost", user="root", passwd="5131", db="lab1")

cursor = db.cursor()

command = "select TransactionID from transactions where TransactionID='" + str(transaction\_id) + "';"

cursor.execute(command)

if cursor.fetchone(): # If we find

command = "delete from transactions where TransactionID='" + str(transaction\_id) + "';"

cursor.execute(command)

db.commit()

def add\_transaction(transaction\_id):

db = MySQLdb.connect(host="localhost", user="root", passwd="5131", db="lab1")

cursor = db.cursor()

command = "select TransactionID from transactions where TransactionID='" + str(transaction\_id) + "';"

cursor.execute(command)

if not cursor.fetchone(): # If no transaction with that ID

command = "INSERT INTO transactions (TransactionID) VALUES ('" + str(transaction\_id) + "');"

cursor.execute(command)

db.commit()

def change\_transaction(transaction\_id, field, new):

db = MySQLdb.connect(host="localhost", user="root", passwd="5131", db="lab1")

cursor = db.cursor()

cursor.execute("select TransactionID from transactions where TransactionID='" + str(transaction\_id) + "';")

if cursor.fetchone(): # If we find

if field == "TransactionID":

cursor.execute("select TransactionID from transactions where TransactionID='" + str(new) + "';")

if not cursor.fetchone():

cursor.execute("UPDATE transactions SET TransactionID='" + str(new) + "' WHERE TransactionID='" + str(transaction\_id) + "';")

elif field == "BillID":

cursor.execute("select BillID from Bill where BillID='" + str(new) + "';")

if cursor.fetchone():

cursor.execute("UPDATE transactions SET BillID='" + str(new) + "' WHERE TransactionID='" + str(transaction\_id) + "';")

elif field == "TimeID":

cursor.execute("select TimeID from Time where TimeID='" + str(new) + "';")

if cursor.fetchone():

cursor.execute("UPDATE transactions SET TimeID='" + str(new) + "' WHERE TransactionID='" + str(transaction\_id) + "';")

elif field == "UserID":

cursor.execute("select UserID from User where UserID='" + str(new) + "';")

if cursor.fetchone():

cursor.execute("UPDATE transactions SET UserID='" + str(new) + "' WHERE TransactionID='" + str(transaction\_id) + "';")

elif field == "Successful":

cursor.execute("UPDATE transactions SET Successful='" + str(new) + "' WHERE TransactionID='" + str(transaction\_id) + "';")

db.commit()

Фрагмент отримання таблиць

import MySQLdb

def get\_users():

db = MySQLdb.connect(host="localhost", user="root", passwd="5131", db="lab1")

cursor = db.cursor()

cursor.execute("show columns from User")

x0 = ""

x1 = ""

x2 = ""

i = 0

for x in cursor.fetchall():

if i == 0:

x0 = x[0]

elif i == 1:

x1 = x[0]

else:

x2 = x[0]

i += 1

ret = [(x0, x1, x2)]

cursor.execute("select \* from User")

return ret + [(x[0], x[1], x[2]) for x in cursor.fetchall()]

def get\_bills():

db = MySQLdb.connect(host="localhost", user="root", passwd="5131", db="lab1")

cursor = db.cursor()

cursor.execute("show columns from Bill")

x0 = ""

x1 = ""

x2 = ""

i = 0

for x in cursor.fetchall():

if i == 0:

x0 = x[0]

elif i == 1:

x1 = x[0]

else:

x2 = x[0]

i += 1

ret = [(x0, x1, x2)]

cursor.execute("select \* from Bill")

return ret + [(x[0], x[1], x[2]) for x in cursor.fetchall()]

def get\_times():

db = MySQLdb.connect(host="localhost", user="root", passwd="5131", db="lab1")

cursor = db.cursor()

cursor.execute("show columns from Time")

x0 = ""

x1 = ""

x2 = ""

i = 0

for x in cursor.fetchall():

if i == 0:

x0 = x[0]

elif i == 1:

x1 = x[0]

else:

x2 = x[0]

i += 1

ret = [(x0, x1, x2)]

cursor.execute("select \* from Time")

return ret + [(x[0], x[1], x[2]) for x in cursor.fetchall()]

def get\_transactions():

db = MySQLdb.connect(host="localhost", user="root", passwd="5131", db="lab1")

cursor = db.cursor()

cursor.execute("show columns from Transactions")

x0 = ""

x1 = ""

x2 = ""

x3 = ""

i = 0

for x in cursor.fetchall():

if i == 0:

x0 = x[0]

elif i == 1:

x1 = x[0]

elif i == 2:

x2 = x[0]

else:

x3 = x[0]

i += 1

ret = [(x0, x1, x2, x3)]

cursor.execute("select \* from Transactions")

return ret + [(x[0], x[1], x[2], x[3]) for x in cursor.fetchall()]

def get\_users\_ids():

db = MySQLdb.connect(host="localhost", user="root", passwd="5131", db="lab1")

cursor = db.cursor()

cursor.execute("select UserID from User")

return [(x[0]) for x in cursor.fetchall()]

Фрагмент пошуку в таблицях «виміри»

import MySQLdb

def search\_bool(bool):

db = MySQLdb.connect(host="localhost", user="root", passwd="5131", db="lab1")

cursor = db.cursor()

cursor.execute("show columns from Time")

x0 = ""

x1 = ""

x2 = ""

i = 0

for x in cursor.fetchall():

if i == 0:

x0 = x[0]

elif i == 1:

x1 = x[0]

else:

x2 = x[0]

i += 1

ret = [(x0, x1, x2)]

cursor.execute("select \* from Time where At\_day\_time=" + bool)

return ret + [(x[0], x[1], x[2]) for x in cursor.fetchall()]

def search\_date(start, end):

db = MySQLdb.connect(host="localhost", user="root", passwd="5131", db="lab1")

cursor = db.cursor()

cursor.execute("show columns from Time")

x0 = ""

x1 = ""

x2 = ""

i = 0

for x in cursor.fetchall():

if i == 0:

x0 = x[0]

elif i == 1:

x1 = x[0]

else:

x2 = x[0]

i += 1

ret = [(x0, x1, x2)]

cursor.execute("select \* from Time where date(DateTime) between '" + start + "' and '" + end + "'")

return ret + [(x[0], x[1], x[2]) for x in cursor.fetchall()]

def search\_text\_not\_word(column, word):

db = MySQLdb.connect(host="localhost", user="root", passwd="5131", db="lab1")

cursor = db.cursor()

cursor.execute("show columns from User")

x0 = ""

x1 = ""

x2 = ""

i = 0

for x in cursor.fetchall():

if i == 0:

x0 = x[0]

elif i == 1:

x1 = x[0]

else:

x2 = x[0]

i += 1

ret = [(x0, x1, x2)]

cursor.execute("select \* from User where not match (" + column + ") against ('" + word + "' in boolean mode);")

return ret + [(x[0], x[1], x[2]) for x in cursor.fetchall()]

def search\_phrase(column, phrase):

db = MySQLdb.connect(host="localhost", user="root", passwd="5131", db="lab1")

cursor = db.cursor()

cursor.execute("show columns from User")

x0 = ""

x1 = ""

x2 = ""

i = 0

for x in cursor.fetchall():

if i == 0:

x0 = x[0]

elif i == 1:

x1 = x[0]

else:

x2 = x[0]

i += 1

ret = [(x0, x1, x2)]

cursor.execute("select \* from User where match (" + column + ") against ('\"" + phrase + "\"' in boolean mode)")

return ret + [(x[0], x[1], x[2]) for x in cursor.fetchall()]

def search\_bill(column, start, end):

db = MySQLdb.connect(host="localhost", user="root", passwd="5131", db="lab1")

cursor = db.cursor()

cursor.execute("show columns from Bill")

x0 = ""

x1 = ""

x2 = ""

i = 0

for x in cursor.fetchall():

if i == 0:

x0 = x[0]

elif i == 1:

x1 = x[0]

else:

x2 = x[0]

i += 1

ret = [(x0, x1, x2)]

cursor.execute("select \* from Bill where " + column + " between " + start + " and " + end)

return ret + [(x[0], x[1], x[2]) for x in cursor.fetchall()]

Фрагмент завантаження з файлу таблиць «виміри»

import json

import MySQLdb

def load\_from\_file(file\_name, table):

db = MySQLdb.connect(host="localhost", user="root", passwd="5131", db="lab1")

cursor = db.cursor()

file = open("files/" + file\_name)

data = json.load(file)

cursor.execute("delete from transactions")

if table == "Users":

cursor.execute("delete from user")

for i in data:

cursor.execute("insert into user (UserID, Name, comment) values ('"+str(i["UserID"])+"', '"+str(i["Name"])+"', '"+str(i["Comment"])+"')")

elif table == "Bills":

cursor.execute("delete from bill")

for i in data:

cursor.execute("insert into bill (BillID, Amount, Average\_amount\_for\_month) values ('"+str(i["BillID"])+"', '"+str(i["Amount"])+"', '"+str(i["Average\_amount\_for\_month"])+"')")

elif table == "Times":

cursor.execute("delete from time")

for i in data:

cursor.execute("insert into time (TimeID, DateTime, At\_day\_time) values ('" + str(i["TimeID"]) + "', '"+str(i["DateTime"])+"', '"+str(i["At\_day\_time"])+"')")

file.close()

db.commit()