COP4521 Homework 6

Name: FSU login:

1 (10 points). A company wants to maintain the commissions for each employee

when the employee sells a product. An employee has two attributes,

employee ID (Eid) and name (Name). A product has two attributes,

product ID (Pid) and maker (Maker). Draw the ER diagram to model the data

for this task.

Answer:

commission

Name

Maker

Pid

Eid

Employee

Sells

Product

2. (10 points). Many real world relations are multiway. For example, in

Question 1, the sells relation can have three parties: an employee sells

a product to a customer (attributes: customer ID (Cid) and name (Name)). How

does the ER model handle such cases? Update the ER diagram in Question 1

for this three-way relation.

Eid

Employee

SellBy

Commission

Name

Cid

Customer

SellTo

Sells

Name

Pid

SellItem

Product

Maker

3. (20 points) Write Python SQLite3 code to create three tables: Employee

(Eid, Name) where Eid is the key, product (Pid, Name) where Pid is the key,

and sells (Eid, Pid). Initialize each table with one entry. You can assume

any appropriate data types for the attributes.

import sqlite3

conn = sqlite3.connect('./sql\_demo.db')

cur = conn.cursor()

# create table in database

cur.execute('''CREATE TABLE EMPLOYEE(

Eid INTEGER PRIMARY KEY NOT NULL,

Name TEXT NOT NULL);

''')

cur.execute('''CREATE TABLE PRODUCT(

Pid INTEGER PRIMARY KEY NOT NULL,

Name TEXT NOT NULL);

''')

cur.execute('''CREATE TABLE SELLS(

Eid INTEGER NOT NULL,

Pid INTEGER NOT NULL);

''')

# commit and save changes to database

conn.commit()

# execute one command

cur.execute('''Insert Into EMPLOYEE (Eid, Name)

Values (1, 'David')’'')

cur.execute('''Insert Into PRODUCT (Pid, Name)

Values (1, 'Hammer')'’')

cur.execute('''Insert Into SELLS (Eid,Pid)

Values (1, 1)’'')

# commit and save changes to database

conn.commit()

conn.close()

4. (20 points) Write Python SQLite3 code to query the database in Question

3 to find the names of all products sold by David. The code must also

display the result of the query.

import sqlite3

conn = sqlite3.connect('./sql\_demo.db')

cur = conn.cursor()

for rows in cur.execute(‘’’

SELECT PRODUCT.Name

FROM EMPLOYEE, PRODUCE, SELLS

WHERE EMPLOYEE.name = ‘David’ and SELLS.Eid = EMPLOYEE.Eid and Product.Pid = SELLS.Pid;’’’):

 print(rows)

conn.close()

5. (20 points) Write the Python Flask code for the static web page

http://127.0.0.1:50001/ and dynamic web-pages

http://127.0.0.1:50001/user/{username} where username can be any string. The

http://127.0.0.1:50001/ page displays "Welcome" while

http://127.0.0.1:50001/user/{username} displays f"Hello {username}!".

Answer:

from flask import Flask
app = Flask(\_\_name\_\_)

@app.route('/')
def welcome():
 return 'Welcome’

@app.route(‘/user/<name>’)

def hello(name):

 return f‘Hello {name}!’

if \_\_name\_\_ == '\_\_main\_\_':
 app.run(host=‘127.0.0.1’, port = 50001)

6. (20 points) Let tableVar be a variable in python. For example,

 tableVar = [[1, 2, 3], ["one", "two", "three"]]

Write the python Flask code (the combine of Python code and html template)

to display the content of tableVar in an html table format in page

<http://127.0.0.1:50001/content>.

Python:

tableVar = …..

return render\_template(‘table.html’, table = tableVar);

table.html:

<!doctype **html**>
<**html**>
 <**body**>
 <**table border = 1**>
 {% for row in table) %}
 <**tr**>
 {% for item in row %}
 <**td**> {{ item }} </**td**>

 {% endfor %}
 </**tr**>
 {% endfor %}
 </**table**>
 </**body**>
</**html**>