**September 26**

Python comprehension query (Boolean) interaction before expression

Q.1 The Number of units for a course can’t exceed 6 (corresponds course table)

Answer

Course = UnivDB [“tables”] [“course”]

Answer = all ( [ c [“units”] < = for c in Course ])

Q.2 Everyone who is enrolled in a class must be a student (appear must be in student structure) use and any and then produce a secquence of function

Enrollment = UnivDB [“tables”] [“enrollment”]

student = UnivDB [“tables”] [“student”]

isastudent = lambda ssn: (any([s[“ssn”] == ssn for s in student]) {alternate: [true for s in student if s[“ssn] == ssn]}

Answer = all ( [isastudent(e[“ssn] for e in enrollement])

Q.5 Student majoring in CS must enroll in a class tittle “vacation in Hawaii” (nested query)

Enrollment = UnivDB [“tables”] [“enrollment”]

Class = UnivDB [“tables”] [“class”]

Student = UnivDB [“tables”] [“student”]

Course = UnivDB [“tables”] [“course”]

Enrolledinhawaii = lambda ssn: any(

[ True for e in enrollment,

For c in class

For co in course

If (e[“ssn”] == ssn and

e[“class”] == c[class] and

c[“decode”] == co[decode] and

c[“cno”] == co[“cno”] and

co[”title”] == ““vacation in Hawaii”

Answer = all([ enrolledinhawaii (s[“ssn”])

for s in student

If s[“masor”] == “cs”

])

Data Query

List course (decode, cno)

Along with the name of students who are currently taking them

Enrollment = UnivDB [“tables”] [“enrollment”]

Class = UnivDB [“tables”] [“class”]

Student = UnivDB [“tables”] [“student”]

Answer: [ ]

For co in course :

Decode = co[“decode”]

Cno = co[“cno”]

Enrolledstudents = [ {“name’ : s[“name”]}

for s in student

if any ([ true

for e in enrollment

for c in class

if e[“ssn”] == s[‘’ssn”] and

e[“class”] = c[“class”] and

c[“decode”] == decode and

C[“cno”] == cno

])

Answer: append({“decode”:decode, “cno”:cno, “student”: enrolledstudent})

Data Query 8: Find students (ssn,name) who are taking a class without satisfying all of its prereq

Enrollment = UnivDB [“tables”] [“enrollment”] (may have issue with enrollment because ssn is) redundant and name is not there)

Class = UnivDB [“tables”] [“class”]

Student = UnivDB [“tables”] [“student”]

Course = UnivDB [“tables”] [“course”]

[ {“ssn}: s[“ssn], “name” : s[“name”]}

for s in student

If (any([ True for e in enrollment

If e[“ssn”] == s[“ssn”]

For c in class

If (c[“class”] == e[“class’] and

Not ( studentsatcourseprerq(s[“ssn”],

C[“decode”],

C[“cno”])

]]

Def studentsatcourseprerq(ssn,decode,cno) :

Prereqs = [ for p in prereq

If (p[“decode”] == decode and p[“cno”] == cno)]

Issatisfied = all([ sat for p in prereqs)

For sat in [any([ true for t in transcript

If t[“ssn”] == ssn and

t[“decode”] == p[“pcode”] and

t[“cno”] == p[“cno” and

(t[“grade] == “A” or t[“grade] == “B”)

]]))

**October 3, 2019**

Create a list of answers and use method Sort to prove the key argument – HW2

More Python Comprehension practice

Data Query Q9: Finding classes in which at least two students are enrolled (no duplicate, sorted by class ID) hint: create a dictionary and get key

(or e can take if out and put True after any

Ans = [ C

For C in class\_

If ( any ([

For e1 in enrollment

For e2 in enrollment

If e1[“ssn] != e1 [“ssn”] and

e1[“class”] == C [“class”] and

e2[“class”] == C [“class”]

])

====================================================================================

What If it is for 200 students – same Q above

[ C

For C in class\_

For c in class

For studEnrolled in [[ e[“ssn”]

For e in enrollment

If e [“class”] == c [“class”]

]]

If len(studEnrolled) >= 200

===================== Q9.sort( key = lamda t : t [“class”]) =================================

Data Q 15. Find Department that offer classes taught by their faculty

Ans = [ d

For d in department

If ( any ([ c[“dcode”] == d[“dcode”] and c[“instr”] == f[“ssn”] and f[“dcode”] == d[“dcode’]

For f in faculty

For c in class\_

])

]

=====================================================================================

Q14. Find Department such that all their classes offered by their faculty

Ans = [ d

For d in department\_

If ( all ([ f[“dcode”] == d[“dcode”]

For f in faculty

if c [“dcode”] == d[“dcode”]

For f in faculty

If f[‘’ssn’’] == c[‘’instr’’]

])

]

=====================================================================================

{ ‘’department’’ : [

{ ‘’cs’’ : { ‘’class’’ : [

{ 17 : { studenst : [

{ssn: ….

]

}

}

Depts = [d [ ‘’dcode’’] for d in department

For d in depts :

Classtaughtbydept = [

Ideas ===

* Distributed DB (existing database management system, how you use them, transactions different, how to do a query)
* Convert the patriot web to a mobile app
* Note taking apps
* Creating new type of database
* DB for server models
* Deploy service/product in cloud