Громов Владислав

ИУ5-33Б

Вариант запросов Б. Предметная область 34.

class DB():

def \_\_init\_\_(self, ID: int, name: str, descr: str):

self.ID = ID

self.descr = descr

self.name = name

class Proc():

def \_\_init\_\_(self, proc\_id: int, name: str, db\_id: int):

self.proc\_id = proc\_id

self.name = name

self.db\_id = db\_id

class ProcDB():

def \_\_init\_\_(self, proc\_id: int, db\_id: int):

self.proc\_id = proc\_id

self.db\_id = db\_id

procs = [

Proc(1, "create\_user", 1),

Proc(2, "delete\_user", 1),

Proc(3, "describe\_db(mean, median, 25%, 50%, 75%, min, max)", 3),

Proc(4, "get\_user\_by\_id", 3),

Proc(5, "get\_log", 2),

Proc(6, "update\_info\_user", 1)

]

DBs = [

DB(1, "main\_db", "основная бд"),

DB(2, "log\_db", "бд для логов"),

DB(3, "ml\_db", "бд для создания модели машинного обучения"),

DB(34, "main\_other\_db", "основная бд (другая)"),

DB(35, "log\_other\_db", "бд для логов (другая)"),

DB(36, "ml\_other\_db", "бд для мл (другая)")

]

procs\_db = [

ProcDB(1, 1),

ProcDB(2, 1),

ProcDB(3, 1),

ProcDB(4, 3),

ProcDB(5, 3),

ProcDB(1, 34),

ProcDB(2, 35),

ProcDB(3, 36),

ProcDB(4, 35),

ProcDB(2, 35)

]

def main():

#один-ко-многим

one\_to\_many = [(proc.name, db.name) for db in DBs for proc in procs if proc.db\_id == db.ID]

#многие-ко-многим

many\_to\_many\_temp = [(proc.name, procdb.db\_id, procdb.proc\_id)

for procdb in procs\_db

for proc in procs

if proc.proc\_id == procdb.proc\_id]

many\_to\_many = [(name, db.name)

for name, db\_id, proc\_id in many\_to\_many\_temp

for db in DBs

if db.ID == db\_id]

#Задачи

# задача 1: список связанных процедур и бд, отсортированный по бд.

print("Задание 1\n{}".format(sorted(one\_to\_many, key=lambda x: x[1])))

# задача 2: список баз данных с количеством процедур в бд и сортировка по количеству процеду.

ans = []

for db in DBs:

count = len(list(filter(lambda x: x.db\_id == db.ID, procs)))

if count > 0:

ans.append((db.name, count))

print("Задание 2\n{}".format(sorted(ans, key=lambda x: x[1], reverse=True)))

# задача 3: список всех процедур, в названии которых

# содержится слово "user" и названия их бд.

print("Задание 3\n{}".format(list(filter(lambda x: "user" in x[0], many\_to\_many))))

if \_\_name\_\_ == "\_\_main\_\_":

main()

