

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 == proddb.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()

```

```

$ python3 RK1.py
Задание 1
[('get_log', 'log_db'), ('create_user', 'main_db'), ('delete_user', 'main_db'), ('update_info_user', 'main_db'),
 ('describe_db(mean, median, 25%, 50%, 75%, min, max)', 'ml_db'), ('get_user_by_id', 'ml_db')]
Задание 2
[('main_db', 3), ('ml_db', 2), ('log_db', 1)]
Задание 3
[('create_user', 'main_db'), ('delete_user', 'main_db'), ('get_user_by_id', 'ml_db'), ('create_user', 'main_oth
er_db'), ('delete_user', 'log_other_db'), ('get_user_by_id', 'log_other_db'), ('delete_user', 'log_other_db')]

```