Вариант В3

```python
# Шаг 1: Определение классов данных

# Класс "Водитель"
class Driver:
    def __init__(self, driver_id, surname, salary, fleet_id):
        self.driver_id = driver_id
        self.surname = surname
        self.salary = salary
        self.fleet_id = fleet_id

# Класс "Автопарк"
class Fleet:
    def __init__(self, fleet_id, name):
        self.fleet_id = fleet_id
        self.name = name

# Класс "Водители автопарка" для реализации отношения многие-ко-многим
class FleetDrivers:
    def __init__(self, driver_id, fleet_id):
        self.driver_id = driver_id
        self.fleet_id = fleet_id


# Шаг 2: Создание списков объектов классов с тестовыми данными

# Создание объектов класса "Водитель"
drivers = [
    Driver(1, "Ivanov", 50000, 1),
    Driver(2, "Alexeev", 60000, 1),
    Driver(3, "Alekseeva", 55000, 2),
    Driver(4, "Abramov", 52000, 2),
    Driver(5, "Petrov", 58000, 2)
]
```

```python
# Создание объектов класса "Автопарк"
fleets = [
    Fleet(1, "Fleet 1"),
    Fleet(2, "Fleet 2")
]

# Создание объектов класса "Водители автопарка" для связи многие-ко-многим
fleet_drivers = [
    FleetDrivers(1, 1),
    FleetDrivers(2, 1),
    FleetDrivers(3, 2),
    FleetDrivers(4, 2),
    FleetDrivers(5, 2)
]


# Шаг 3: Разработка запросов

# Запрос №1: Вывод списка водителей с фамилией, начинающейся на "A", и названием автопарка
drivers_with_a = [(driver.surname, fleet.name) for driver in drivers for fleet in fleets if driver.surname.startswith("A")]
print("Водители на 'A':", drivers_with_a)

# Запрос №2: Вывод списка автопарков с минимальной зарплатой водителей, отсортированный по минимальной зарплате
min_salary_per_fleet = {fleet.name: min(driver.salary for driver in drivers if driver.fleet_id == fleet.fleet_id) for fleet in fleets}
sorted_fleets_by_min_salary = [(fleet, min_salary) for fleet, min_salary in sorted(min_salary_per_fleet.items(), key=lambda x: x[1])]
print("Автопарки по минимальной зарплате:", sorted_fleets_by_min_salary)

# Запрос №3: Вывод списка всех связанных водителей и автопарков, сортировка по водителям (автопарки произвольно)
fleet_drivers_sorted_by_driver = sorted(fleet_drivers, key=lambda x: drivers[x.driver_id-1].surname)
drivers_fleets = [(drivers[fd.driver_id-1].surname, fleets[fd.fleet_id-1].name) for fd in fleet_drivers_sorted_by_driver]
print("Водители и автопарки", drivers_fleets)
```

Результат работы программы:

Водители на 'A': [('Alexeev', 'Fleet 1'), ('Alekseeva', 'Fleet 2'), ('Abramov', 'Fleet 2')]

Автопарки по минимальной зарплате: [('Fleet 1', 50000), ('Fleet 2', 52000)]

Водители и автопарки [('Abramov', 'Fleet 2'), ('Alekseeva', 'Fleet 2'), ('Alexeev', 'Fleet 1'), ('Ivanov', 'Fleet 1'), ('Petrov', 'Fleet 2')]