

1. Знайти загальну вартість, середню вартість, максимальну та мінімальну вартість автомобілів за маркою.

```
> db.cars.aggregate([
  {
    $group: {
      _id: "$brand",
      totalPrice: { $sum: "$price" },
      averagePrice: { $avg: "$price" },
      maxPrice: { $max: "$price" },
      minPrice: { $min: "$price" }
    }
  }
])
< {
  _id: 'Nissan',
  totalPrice: 23000,
  averagePrice: null,
  maxPrice: 23000,
  minPrice: 23000
}
```

2. Знайти кількість автомобілів за маркою, випущених після 2018 року.

```
> db.cars.aggregate([
  {
    $match: { year: { $gt: 2018 } }
  },
  {
    $count: "$_id"
  }
])
< {
  _id: 2,
  brand: 'Honda',
  model: 'Accord',
  year: 2019,
  price: 22000,
  ownerId: 2,
  mileage: 15000,
  insurance: 700
}
```

3. Знайти власників з масивами моделей їхніх автомобілів.

```
db.cars.aggregate([
{
  $lookup: {
    from: "owners",
    localField: "ownerId",
    foreignField: "_id",
    as: "owner"
  }
},
{
  $group: {
    _id: "$ownerId",
    name: { $first: "$owner.name" },
    cars: { $push: {model: "$model", brand: "$brand", year: "$year", price: "$price"} }
  }
},
{
  $project: {
    _id: 0,
    name: 1,
    cars: 1
  }
}
])
{
  name: [
    'John Doe'
  ],
  cars: [
    {
      model: 'Camry',
      brand: 'Toyota',
      year: 2018,
      price: 24000
    },
  ],
}
```

4. Сортувати автомобілі за віком власника.

```
> db.cars.aggregate([
  { $lookup: { from: "owners", localField: 'ownerId', foreignField: '_id', as: 'ownerInfo' }},
  { $unwind: '$ownerInfo' },
  { $group: {
    _id: '$ownerId',
    name: { $first: '$ownerInfo.name' },
    age: { $first: '$ownerInfo.age' },
    cars: { $push: {
      model: "$model",
      brand: "$brand",
      year: "$year",
      price: "$price"
    }}
  }},
  { $sort: { age: 1 } },
  {
    $project: { _id: 1, name: 1, age: 1, cars: 1 }
  }
])
{
  _id: 2,
  name: 'Jane Smith',
  age: 34,
  cars: [
    {
      model: 'Accord',
      brand: 'Honda',
      year: 2019,
      price: 22000
    }
  ],
}
```

5. Знайти загальну вартість автомобілів кожного власника.

```
> db.cars.aggregate([
  { $lookup: {from: 'owners', localField: 'ownerId', foreignField: '_id', as: 'ownerInfo' }},
  { $unwind: '$ownerInfo' },
  { $group: {
    _id: '$ownerId',
    name: { $first: '$ownerInfo.name' },
    totalPriceOfCars: { $sum: '$price' }
  }},
  { $project: { _id: 0 ,name: 1, totalPriceOfCars: 1 } }
])
< {
  name: 'John Doe',
  totalPriceOfCars: 107000
}
{
  name: 'Jane Smith',
  totalPriceOfCars: 79000
}
{
  name: 'Elon Musk',
  totalPriceOfCars: 113000
}
```

6. Обчислити середній вік автомобілів кожної марки, для чого краще використовувати додаткове поле age, додане до кожного документу.

```
> db.cars.aggregate([
  {$addField: { age: {$subtract: [ { $year: new Date() }, '$year' ] } }},
  { $group: {
    _id: '$brand',
    averageAge: { $avg: '$age' }
  }},
  { $project: { _id: 0, brand: '$_id', averageAge: 1 } }
])
< {
  averageAge: 5,
  brand: 'Chevrolet'
}
{
  averageAge: 6,
  brand: 'Toyota'
}
{
  averageAge: 6,
  brand: 'Audi'
}
{
  averageAge: 7,
  brand: 'Nissan'
}
```

7. Додати поле "expensive", яке приймає значення "True" для машин дорожче \$30000 та

```
> db.cars.aggregate([
  { $project: {
    _id: 0,
    model: 1,
    brand: 1,
    year: 1,
    price: 1,
    expensive: { $cond: {
      if: { $gt: [ '$price', 30000 ] },
      then: true,
      else: false
    } }
  } }
])
```

"False"

```
{
  brand: 'Toyota',
  model: 'Camry',
  year: 2018,
  price: 24000,
  expensive: false
}
```

8. Додати поле "ageGroup", яке визначає вік автомобіля у групи: "Новий" (< 3 років), "Середній" (3-6 років) та "Старий" (> 6 років).

```
> db.cars.aggregate([
  {$addField: { age: { $subtract: [{ $year: new Date() }, '$year'] } }},
  { $project: {
    _id: 0,
    model: 1,
    brand: 1,
    price: 1,
    year: 1,
    ageGroup: {
      $switch: {
        branches: [
          { case: { $lt: ['$age', 3] }, then: "New" },
          { case: { $gt: ['$age', 6] }, then: "Old" },
        ],
        default: 'Average'
      }
    }
  }}
])
```

Додати поле "priceRange", яке визначає діапазон ціни автомобіля: "Дешевий" (< \$20000), "Середній" (\$20000-\$30000) та "Дорогий" (> \$30000)

```
> db.cars.aggregate([
  { $project: {
    _id: 0,
    model: 1,
    brand: 1,
    year: 1,
    price: 1,
    priceRange: {
      $switch: {
        branches: [
          { case: { $lt: ['$price', 20000] }, then: 'Cheap' },
          { case: { $gt: ['$price', 30000] }, then: 'Expensive' },
        ],
        default: 'Average'
      }
    }
  }
}]
< {
  brand: 'Toyota',
  model: 'Camry',
  year: 2018,
  price: 24000,
  priceRange: 'Average'
}
{
  brand: 'Honda',
  model: 'Accord',
  year: 2019,
  price: 22000,
  priceRange: 'Average'
}
```

10. Додати поле "ownerAgeCategory", яке визначає вікову категорію власника: "Молодий" (< 35 років), "Дорослий" (35-50 років) та "Похилий" (> 50 років).

```
> db.cars.aggregate([
  { $lookup: { from: 'owners', localField: 'ownerId', foreignField: '_id', as: 'ownerInfo' }},
  { $unwind: '$ownerInfo' },
  { $addFields: { ownerAgeCategory: {
    $switch: {
      branches: [
        { case: { $lt: ['$ownerInfo.age', 35] }, then: 'Young' },
        { case: { $gt: ['$ownerInfo.age', 60] }, then: 'Old' },
      ],
      default: 'Adult'
    }
  } } },
  {
    $project: {
      _id: 0,
      brand: 1,
      model: 1,
      year: 1,
      price: 1,
      ownerAge: '$ownerInfo.age',
      ownerAgeCategory: 1
    }
  }
])
< {
  brand: 'Toyota',
  model: 'Camry',
  year: 2018,
  price: 24000,
  ownerAgeCategory: 'Adult',
  ownerAge: 45
}
```


11. Додати поле "serviceNeeded", яке вказує на необхідність обслуговування автомобіля залежно від пробігу: "Потрібне" (> 50000 км), "Не потрібно" (<= 50000 км).

```
> db.cars.aggregate([
  {
    $project: {
      _id: 0,
      model: 1,
      brand: 1,
      year: 1,
      price: 1,
      mileage: 1,
      serviceNeeded: {
        $cond: {
          if: { $lte: ['$mileage', 50000] },
          then: 'No Nedeed',
          else: 'Needed'
        }
      }
    }
  }
])
< {
  brand: 'Toyota',
  model: 'Camry',
  year: 2018,
  price: 24000,
  mileage: 30000,
  serviceNeeded: 'No Nedeed'
}
```

12. Додати поле "insuranceCost", яке визначає вартість страхування автомобіля: "Низька" (< \$500), "Середня" (\$500-\$1500) та "Висока" (> \$1500).

```
> db.cars.aggregate([
  {
    $project: {
      _id: 0,
      model: 1,
      brand: 1,
      year: 1,
      price: 1,
      insurance: 1,
      insuranceCost: {
        $switch: {
          branches: [
            { case: { $lt: ['$insurance', 500] }, then: 'Low' },
            { case: { $gt: ['$insurance', 1500] }, then: 'Expensive' },
          ],
          default: 'Average'
        }
      }
    }
  ]
})
< {
  brand: 'Toyota',
  model: 'Camry',
  year: 2018,
  price: 24000,
  insurance: 600,
  insuranceCost: 'Average'
}
```