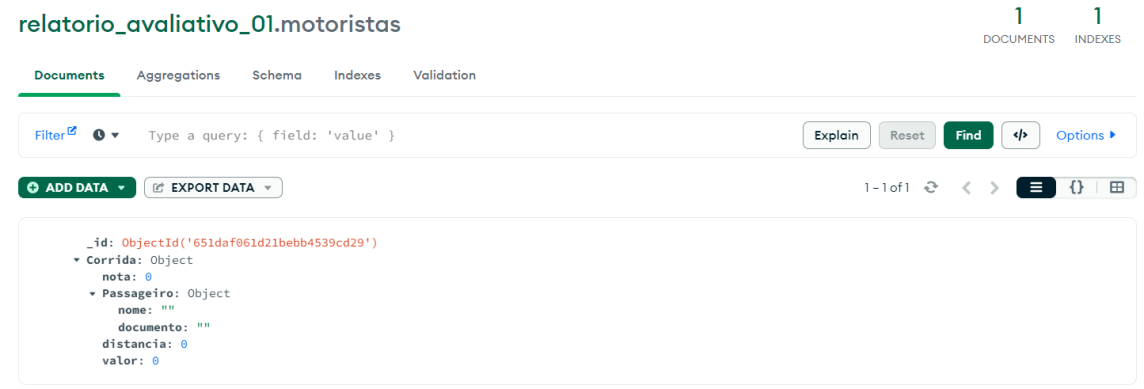
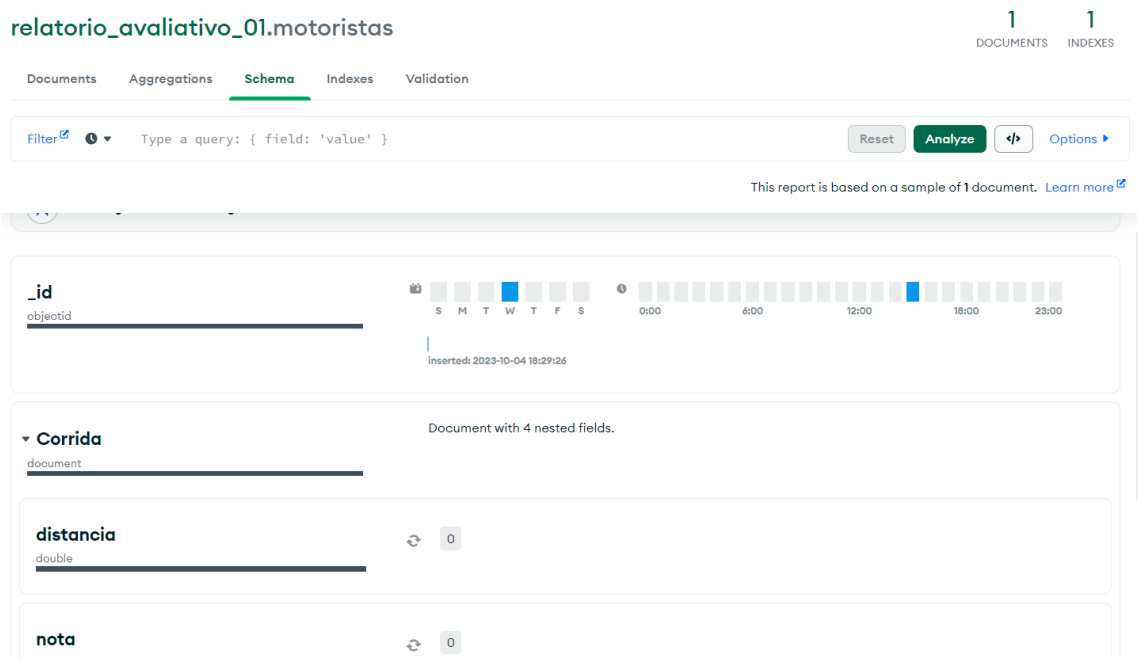
```python
class Database:
    def __init__(self, database, collection):
        self.connect(database, collection)

    def connect(self, database, collection):
        try:
            connectionString = "mongodb+srv://root:root@cluster0.ddrq2fn.mongodb.net/"
            self.clusterConnection = pymongo.MongoClient(
                connectionString,
                tlsAllowInvalidCertificates=True
            )
            self.db = self.clusterConnection[database]
            self.collection = self.db[collection]
            print("Conectado ao banco de dados com sucesso!")
        except Exception as e:
            print(e)

    def resetDatabase(self):
        try:
            self.db.drop_collection(self.collection)
            self.collection.insert_many(dataset)
            print("Banco de dados resetado com sucesso!")
        except Exception as e:
            print(e)
```

```python
class MotoristaDAO:
    def __init__(self, database):
        self.db = database

    def create_driver(self, nota: int, corridas: int):
        try:
            res = self.db.collection.insert_one({"nota": nota, "corridas": corridas})
            print(f"Driver created with id: {res.inserted_id}")
            return res.inserted_id
        except Exception as e:
            print(f"An error occurred while creating driver: {e}")
            return None

    def read_driver_by_id(self, id: str):
        try:
            res = self.db.collection.find_one({"_id": ObjectId(id)})
            print(f"Driver found: {res}")
            return res
        except Exception as e:
            print(f"An error occurred while reading driver: {e}")
            return None

    def update_driver(self, id: str, nota: int, corridas: int):
        try:
            res = self.db.collection.update_one({"_id": ObjectId(id)}, {"$set": {"nota": nota, "corridas": corridas}})
            print(f"driver updated: {res.modified_count} document(s) modified")
            return res.modified_count
        except Exception as e:
            print(f"An error occurred while updating driver: {e}")
            return None

    def delete_driver(self, id: str):
        try:
            res = self.db.collection.delete_one({"_id": ObjectId(id)})
            print(f"driver deleted: {res.deleted_count} document(s) deleted")
            return res.deleted_count
        except Exception as e:
            print(f"An error occurred while deleting driver: {e}")
            return None
```

```python
#Classes
class Passageiro:
    def __init__(self, nome: str, documento: str):
        self.nome = nome
        self.documento = documento

class Corrida:
    def __init__(self, nota: int, distancia: float, valor: float, passageiro: Passageiro):
        self.nota = nota
        self.distancia = distancia
        self.valor = valor
        self.passageiro = Passageiro

class Motorista:
    def __init__(self, corridas: [Corrida], nota: int):
        self.corridas = Corrida
        self.nota = nota
```



```python
class MotoristaCRUD(SimpleCLI):
    def __init__(self, driver_model):
        super().__init__()
        self.driver_model = driver_model
        self.add_command("create", self.create_driver)
        self.add_command("read", self.read_driver)
        self.add_command("update", self.update_driver)
        self.add_command("delete", self.delete_driver)

    def create_driver(self):
        nota = int(input("Entre com a nota:"))
        corrida = input("Entre com a corrida:")
        self.driver_model.create_driver(nota, corrida)

    def read_driver(self):
        id = input("Enter the id: ")
        driver = self.driver_model.read_driver_by_id(id)
        if driver:
            print(f"Titulo: {driver['titulo']}")
            print(f"Autor: {driver['autor']}")
            print(f"Ano: {driver['ano']}")
            print(f"Preco: {driver['preco']}")

    def update_driver(self):
        id = input("Entre com o id: ")
        titulo = input("Entre com o novo titulo: ")
        autor = input("Entre com o novo autor: ")
        ano = int(input("Entre com o novo ano: "))
        preco = float(input("Entre com novo o preco: "))
        self.driver_model.update_driver(id, titulo, autor, ano, preco)

    def delete_driver(self):
        id = input("Entre com o id: ")
        self.driver_model.delete_driver(id)

    def run(self):
        print("Bem vindo ao driverCRUD!")
        print("Comandos disponiveis: create, read, update, delete, quit")
        super().run()
```


relatorio_avaliativo_01.motoristas
1
DOCUMENTS
1
INDEXES
Documents
Aggregations
Schema
Indexes
Validation
Filter
Type a query: { field: 'value' }
Reset
Analyze
Options
This report is based on a sample of 1 document. Learn more
_id
objectid
S M T W T F S
0:00
6:00
12:00
18:00
23:00
inserted: 2023-10-04 18:29:26
Corrida
document
Document with 4 nested fields.
distancia
double
0
nota
0