GAMEFUSION - VIDEO GAME CONSOLE RENTAL PROVIDER

GAMEFUSION

PROVIDING AN UNFORGETTABLE GAMING EXPERIENCE

KUSSAINOV ANSAR | SE-2201

PROJECT OVERVIEW

OUR AIM: **To provide a convenient and enjoyable gaming experience by offering gaming console rentals with flexible options.**

PROJECT GOALS:
1) Build a reliable and profitable gaming console rental service.
2) Attract a diverse customer base, including gaming enthusiasts, families, and pro gamers.
3) Develop streamlined processes for renting, returning, and maintaining gaming consoles.
4) Create a welcoming environment in the store for gamers to gather, connect, and play together.
5) Implement robust data security measures, particularly when handling customer information and ID cards.

PROJECT OVERVIEW

ABOUT US:

GameFusion is a leading video game console rental provider, committed to delivering an unparalleled gaming experience.

SERVICES OFFERED:

- We offer rental services for popular gaming consoles such as PlayStations and Xboxes.
- Customers can choose to rent consoles for in-store play or enjoy the convenience of home delivery.

RENTAL PROCESS:

- In-store rentals: Customers visit our store, rent a console, and enjoy gaming on-site.
- Home delivery: Free delivery service available with a deposit required (ID card as collateral).
- Rental options: Hourly rentals (minimum 2 hours), with each hour priced at 1,000 tenge.

RELEVANCE OF OUR WORK: NEXT-LEVEL GAMING

GameFusion is your go-to choice for a gaming experience that goes beyond the ordinary. Offering a unique console rental service, we provide gamers with unmatched flexibility and variety, meeting the demand for diverse gaming experiences. Choose GameFusion for convenience, adaptability, and a connected gaming community. With a focus on tech-savvy users, our seamless digital and in-store solutions redefine gaming expectations. Elevate your gaming choices with GameFusion – where gaming is not just an activity but an immersive, secure, and socially connected experience.

OVERVIEW TO OUR BUSINESS MODEL:

GameFusion operates on a versatile business model, encompassing both in-store and home-based gaming services. The hourly pricing structure ensures affordability and caters to diverse customer preferences. The use of ID cards as collateral adds a layer of security for home deliveries. The commitment to building an in-store gaming community fosters a sense of belonging for gamers.

COMPARATIVE ANALYSIS:

1. Versatility and Convenience:
   - GameFusion: Stands out by offering both in-store and home delivery services, providing customers with flexibility based on their preferences.
   - Competitors: Some competitors may focus solely on one aspect, either in-store gaming or home-based rentals.
2. Pricing Structure:
   - GameFusion: Adopts an hourly pricing model, ensuring affordability and adaptability to varying customer needs.
   - Competitors: Pricing models may vary, with some competitors adopting subscription-based plans, pay-per-rental, or fixed-duration rental fees.

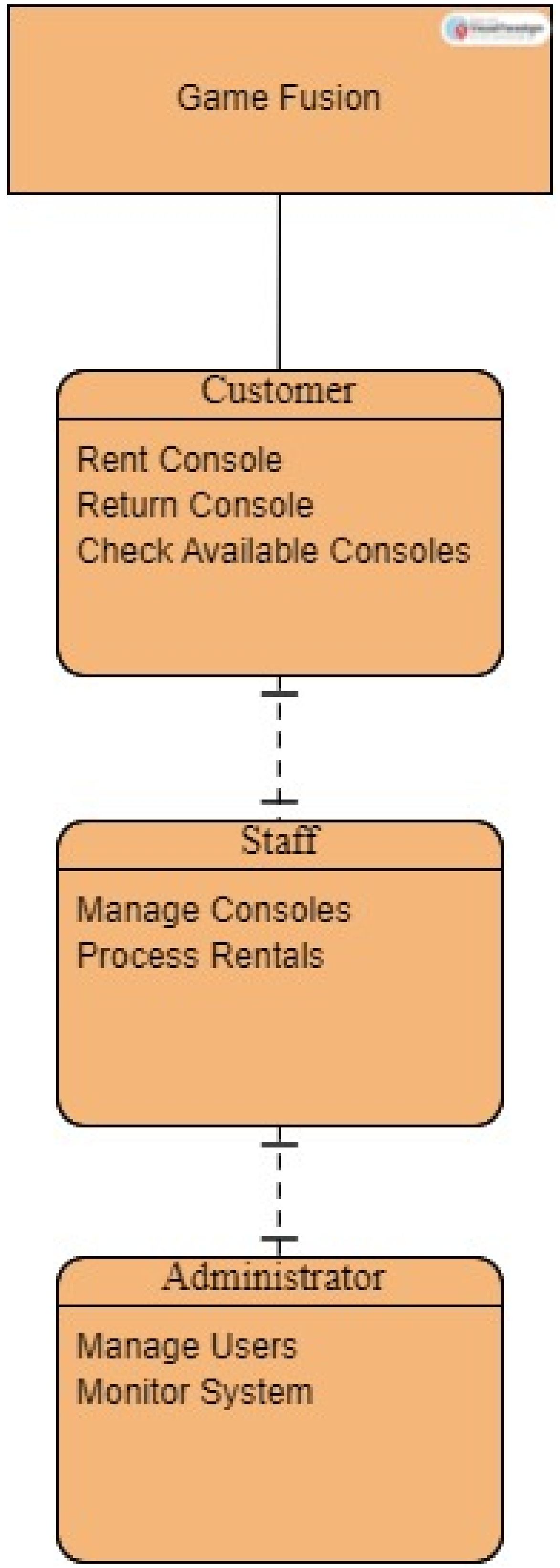
COMPARATIVE ANALYSIS:

1. Security Measures:
   - GameFusion: Utilizes ID cards as collateral for home deliveries, enhancing security for both the customer and the business.
   - Competitors: Security measures may differ, with some platforms relying solely on user accounts and payment information.
2. Community Building:
   - GameFusion: Prioritizes the creation of an in-store gaming community, fostering social connections among gamers.
   - Competitors: Social aspects may vary, with some platforms focusing more on the transactional aspect of rentals.

In conclusion, GameFusion positions itself as a dynamic and customer-centric gaming console rental service by combining versatility, affordability, and community engagement. The unique features and business model set it apart in the market, providing a well-rounded gaming experience for customers.

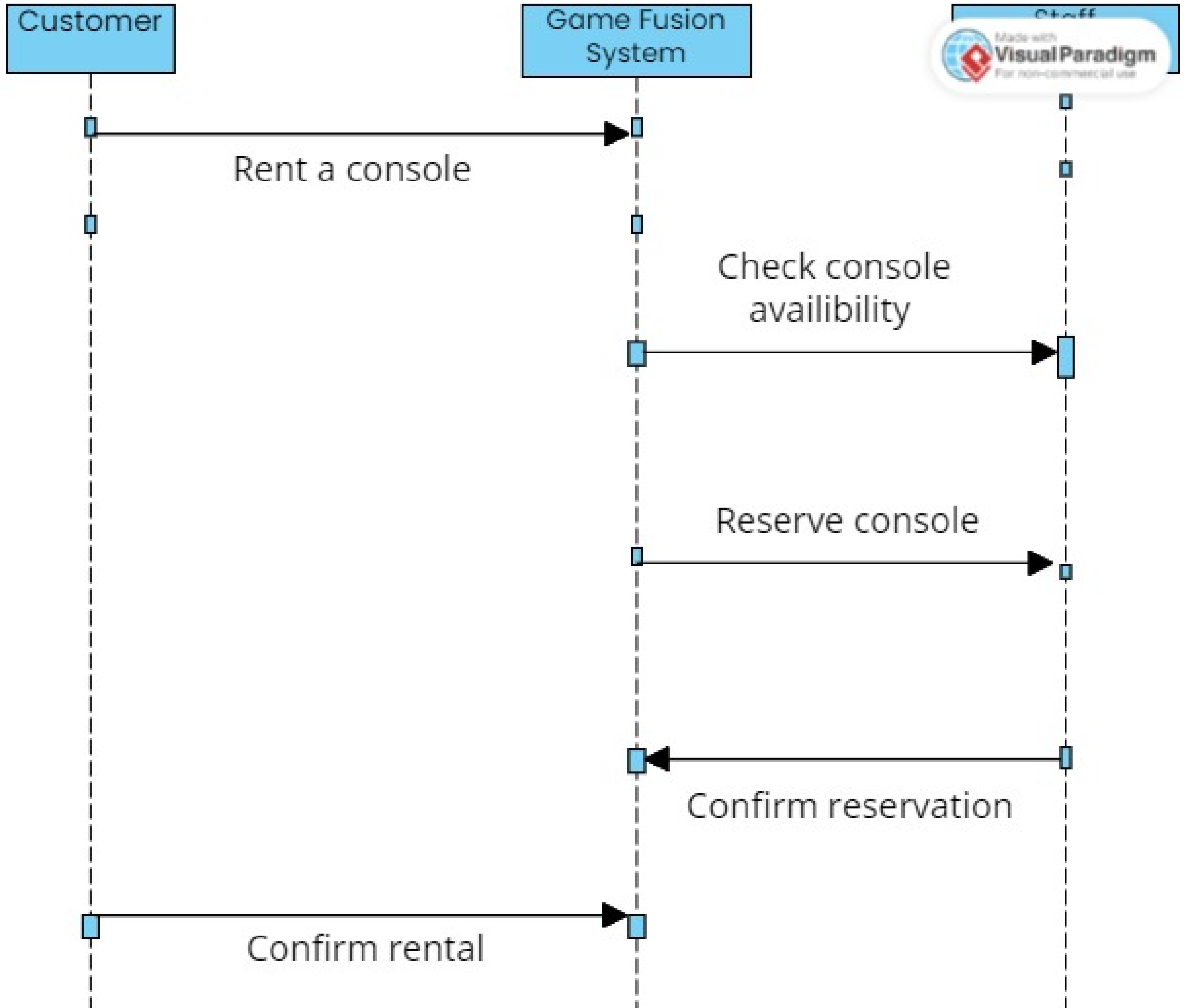
USE CASE DIAGRAM:

A USE CASE DIAGRAM REPRESENTS THE INTERACTIONS BETWEEN THE VARIOUS PARTICIPANTS AND THE SYSTEM ITSELF. IN OUR CASE, PARTICIPANTS CAN BE CUSTOMERS, STORE EMPLOYEES AND ADMINISTRATORS. HERE'S A SIMPLIFIED REPRESENTATION:

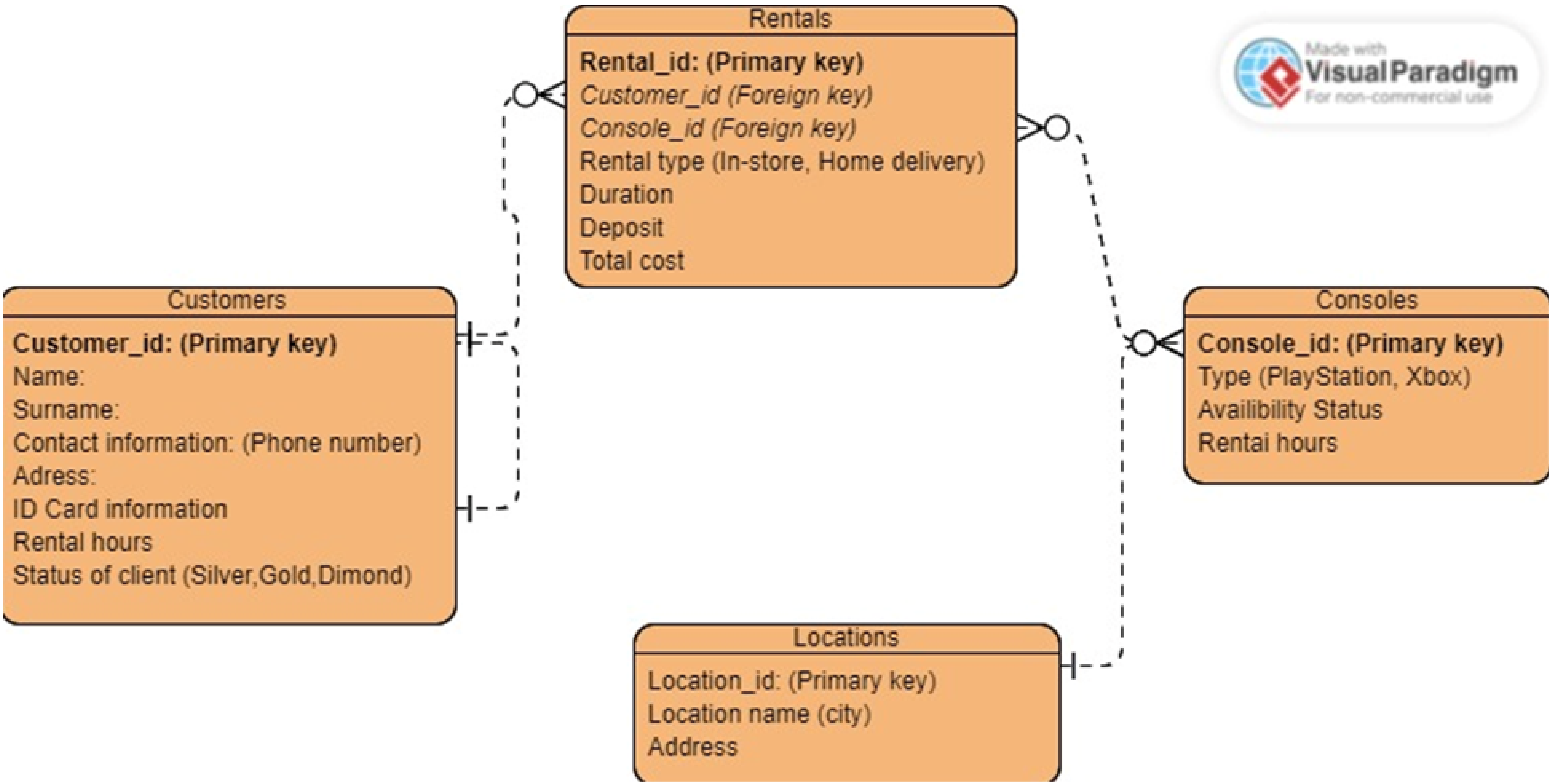


SEQUENCE DIAGRAM:

A SEQUENCE DIAGRAM DEPICTS THE INTERACTIONS BETWEEN DIFFERENT COMPONENTS OR OBJECTS OVER TIME. LET'S CONSIDER THE PROCESS OF A CUSTOMER RENTING A CONSOLE:



ENTITY RELATIONSHIP DIAGRAM:



Rentals

Rental_id: (Primary key)
Customer_id (Foreign key)
Console_id (Foreign key)
Rental type (In-store, Home delivery)
Duration
Deposit
Total cost

Customers

Customer_id: (Primary key)
Name:
Surname:
Contact information: (Phone number)
Adress:
ID Card information
Rental hours
Status of client (Silver,Gold,Dimond)

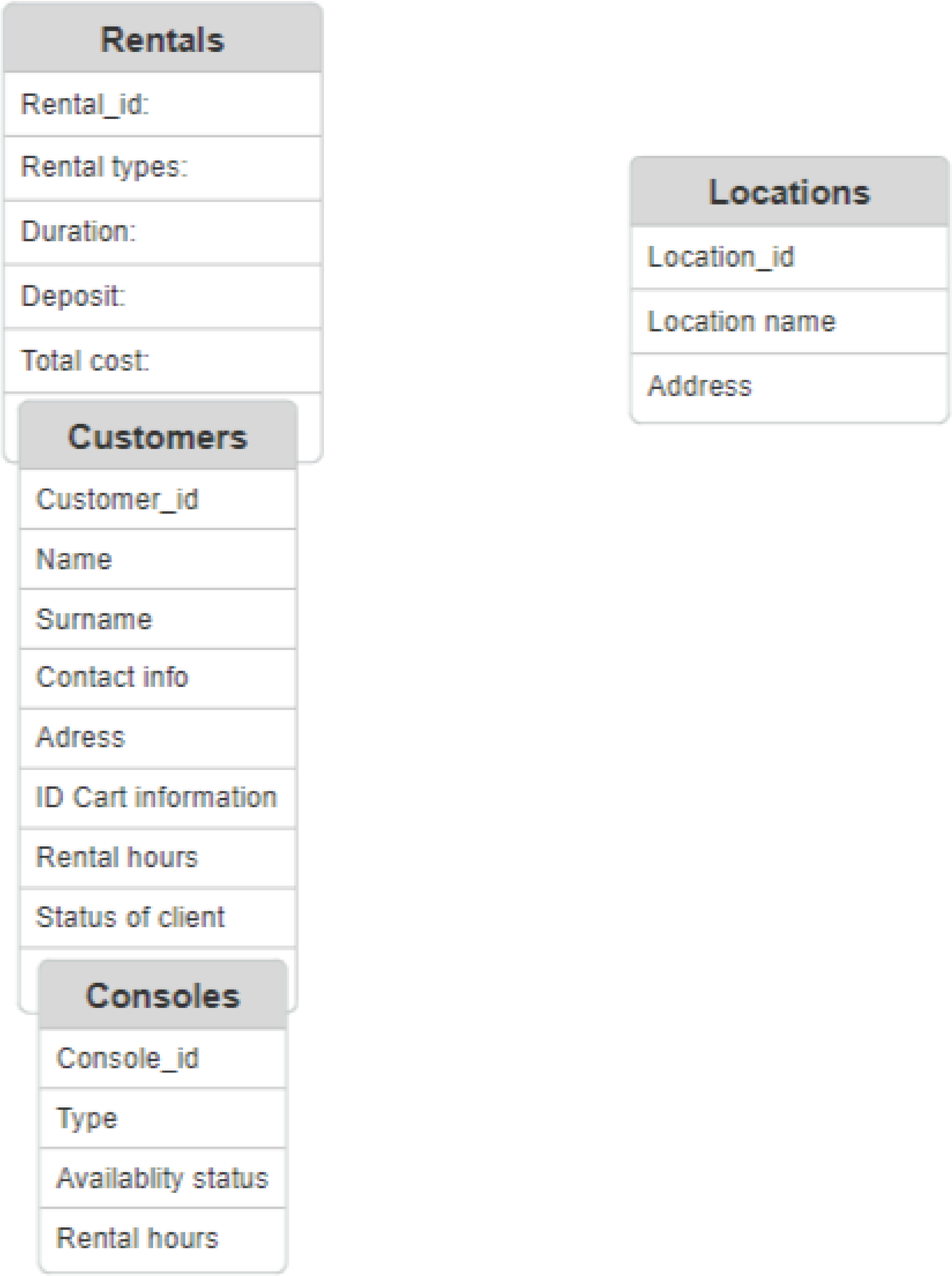
Consoles

Console_id: (Primary key)
Type (PlayStation, Xbox)
Availibility Status
Rentai hours

Locations

Location_id: (Primary key)
Location name (city)
Address

COLLECTION RELATIONSHIP DIAGRAM:



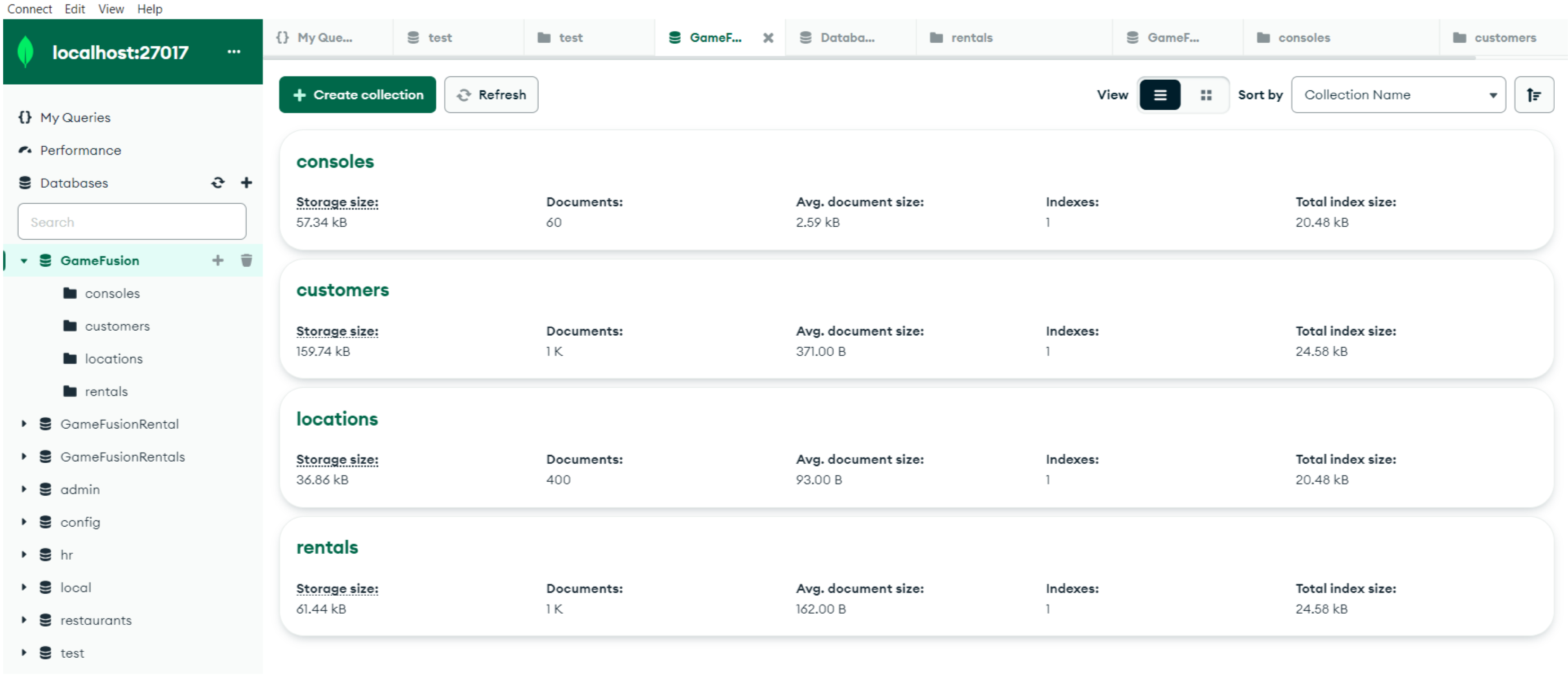
Rentals
Rental_id:
Rental types:
Duration:
Deposit:
Total cost:

Locations
Location_id
Location name
Address

Customers
Customer_id
Name
Surname
Contact info
Adress
ID Cart information
Rental hours
Status of client

Consoles
Console_id
Type
Availablity status
Rental hours

GAME FUSION DATABASE COLLECTIONS



consoles

Storage size:	Documents:	Avg. document size:	Indexes:	Total index size:
57.34 kB	60	2.59 kB	1	20.48 kB

customers

Storage size:	Documents:	Avg. document size:	Indexes:	Total index size:
159.74 kB	1 K	371.00 B	1	24.58 kB

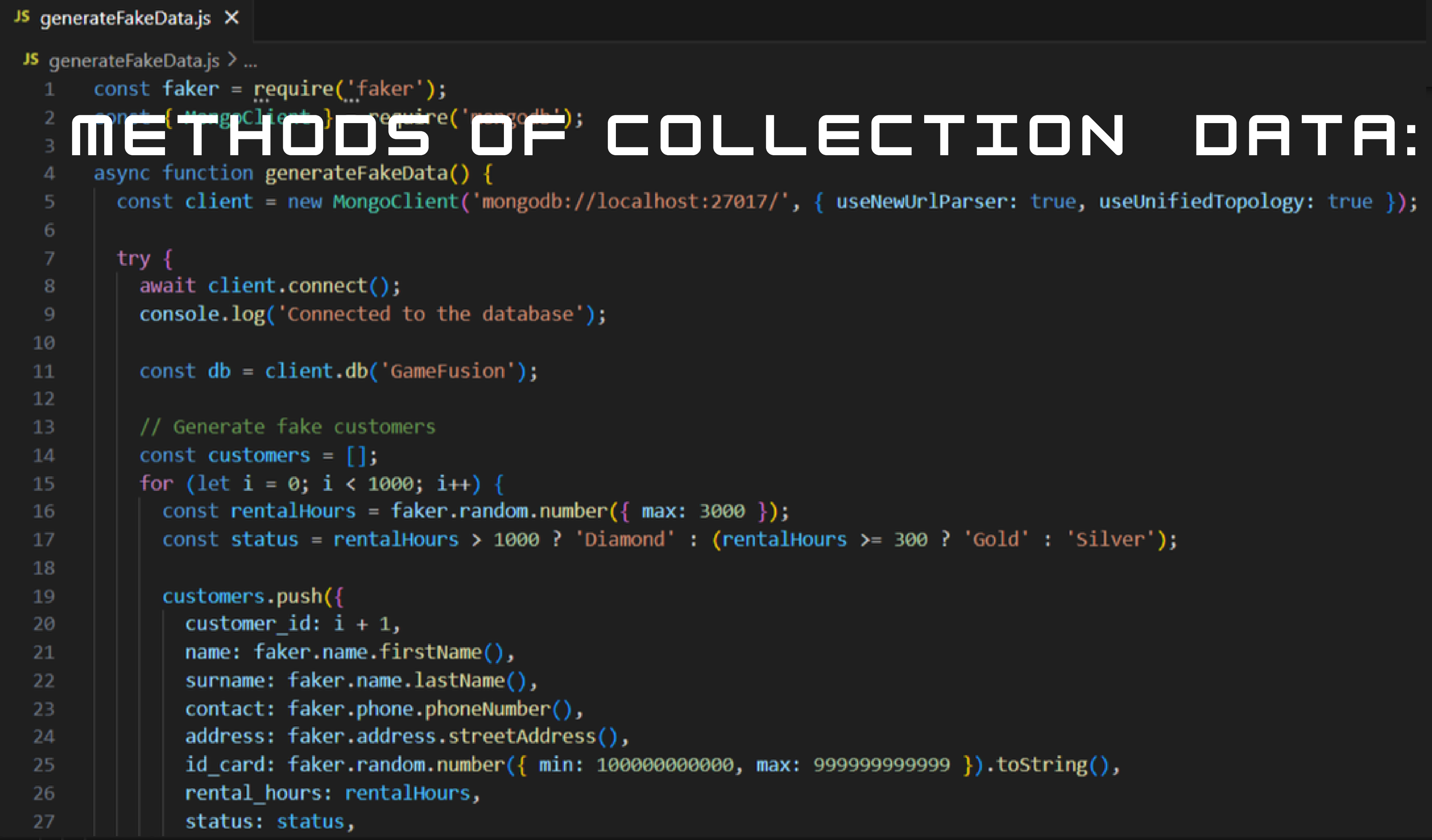
locations

Storage size:	Documents:	Avg. document size:	Indexes:	Total index size:
36.86 kB	400	93.00 B	1	20.48 kB

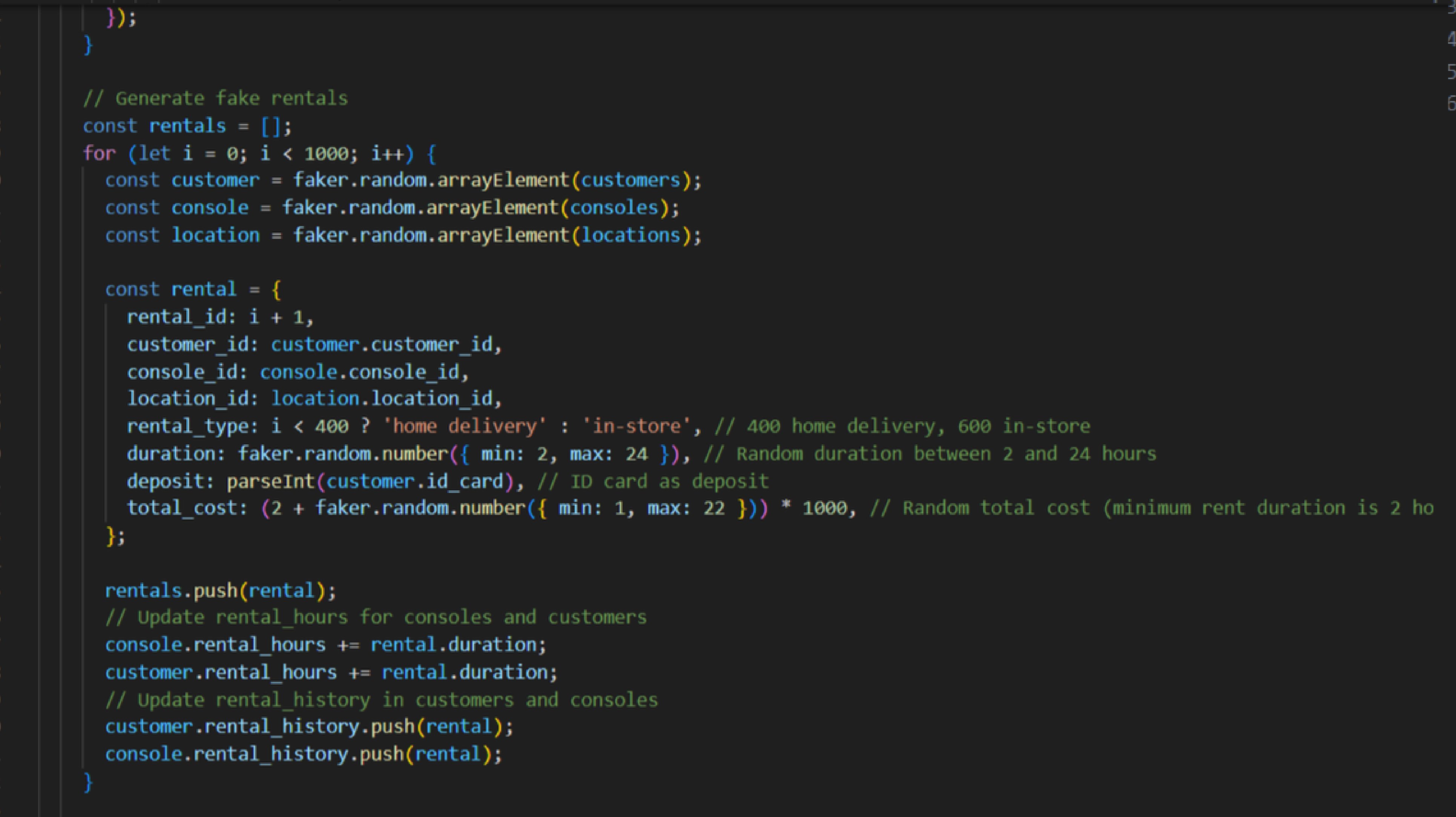
rentals

Storage size:	Documents:	Avg. document size:	Indexes:	Total index size:
61.44 kB	1 K	162.00 B	1	24.58 kB

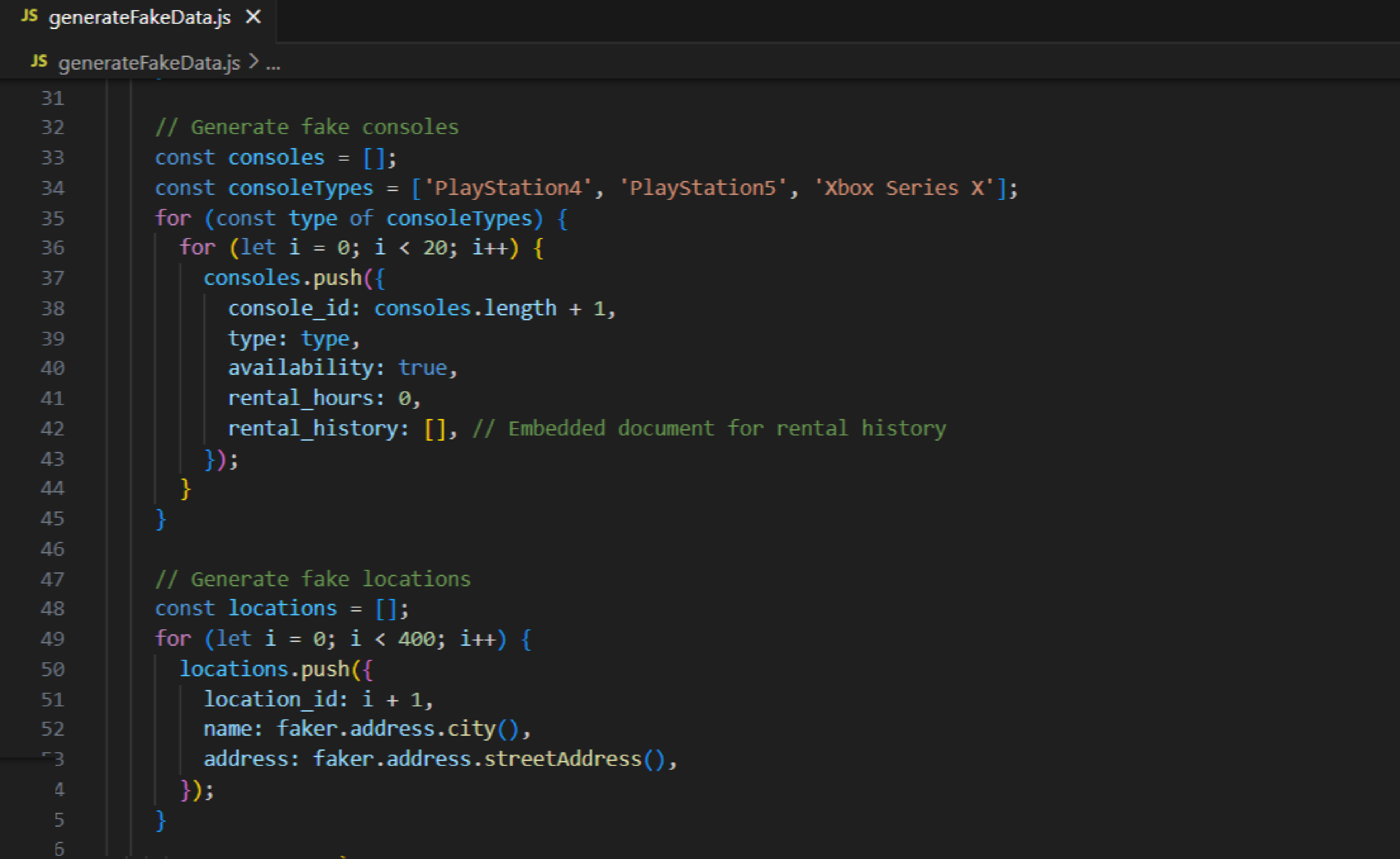
METHODS OF COLLECTION DATA:



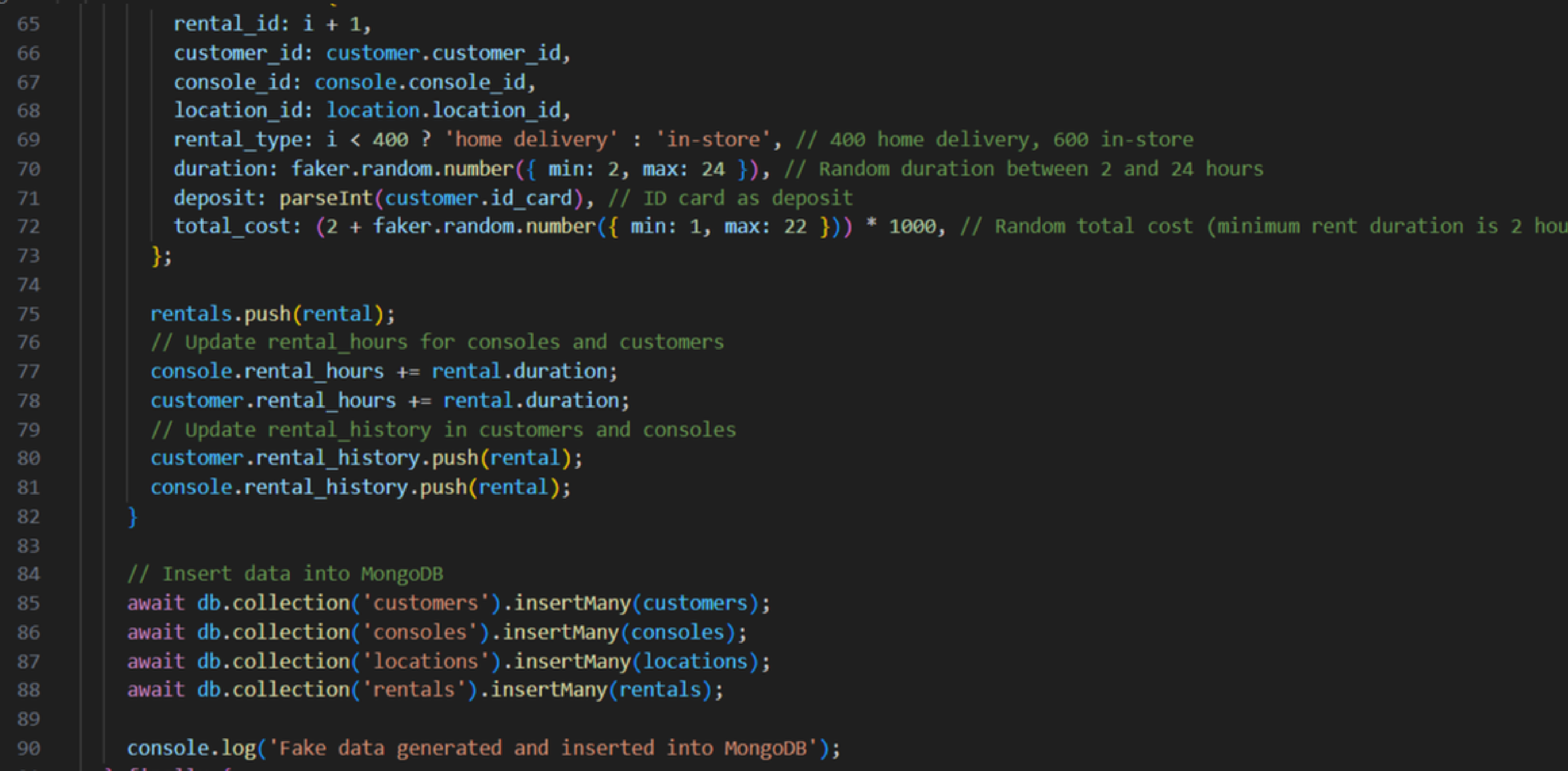
```javascript
const faker = require('faker');
const { MongoClient } = require('mongodb');

async function generateFakeData() {
  const client = new MongoClient('mongodb://localhost:27017/', { useNewUrlParser: true, useUnifiedTopology: true });

  try {
    await client.connect();
    console.log('Connected to the database');

    const db = client.db('GameFusion');

    // Generate fake customers
    const customers = [];
    for (let i = 0; i < 1000; i++) {
      const rentalHours = faker.random.number({ max: 3000 });
      const status = rentalHours > 1000 ? 'Diamond' : (rentalHours >= 300 ? 'Gold' : 'Silver');

      customers.push({
        customer_id: i + 1,
        name: faker.name.firstName(),
        surname: faker.name.lastName(),
        contact: faker.phone.phoneNumber(),
        address: faker.address.streetAddress(),
        id_card: faker.random.number({ min: 100000000000, max: 999999999999 }).toString(),
        rental_hours: rentalHours,
        status: status,
```



```javascript
      });
    }

    // Generate fake rentals
    const rentals = [];
    for (let i = 0; i < 1000; i++) {
      const customer = faker.random.arrayElement(customers);
      const console = faker.random.arrayElement(consoles);
      const location = faker.random.arrayElement(locations);

      const rental = {
        rental_id: i + 1,
        customer_id: customer.customer_id,
        console_id: console.console_id,
        location_id: location.location_id,
        rental_type: i < 400 ? 'home delivery' : 'in-store', // 400 home delivery, 600 in-store
        duration: faker.random.number({ min: 2, max: 24 }), // Random duration between 2 and 24 hours
        deposit: parseInt(customer.id_card), // ID card as deposit
        total_cost: (2 + faker.random.number({ min: 1, max: 22 })) * 1000, // Random total cost (minimum rent duration is 2 ho
      };

      rentals.push(rental);
      // Update rental_hours for consoles and customers
      console.rental_hours += rental.duration;
      customer.rental_hours += rental.duration;
      // Update rental_history in customers and consoles
      customer.rental_history.push(rental);
      console.rental_history.push(rental);
    }
```



```javascript

    // Generate fake consoles
    const consoles = [];
    const consoleTypes = ['PlayStation4', 'PlayStation5', 'Xbox Series X'];
    for (const type of consoleTypes) {
      for (let i = 0; i < 20; i++) {
        consoles.push({
          console_id: consoles.length + 1,
          type: type,
          availability: true,
          rental_hours: 0,
          rental_history: [], // Embedded document for rental history
        });
      }
    }

    // Generate fake locations
    const locations = [];
    for (let i = 0; i < 400; i++) {
      locations.push({
        location_id: i + 1,
        name: faker.address.city(),
        address: faker.address.streetAddress(),
      });
    }

```

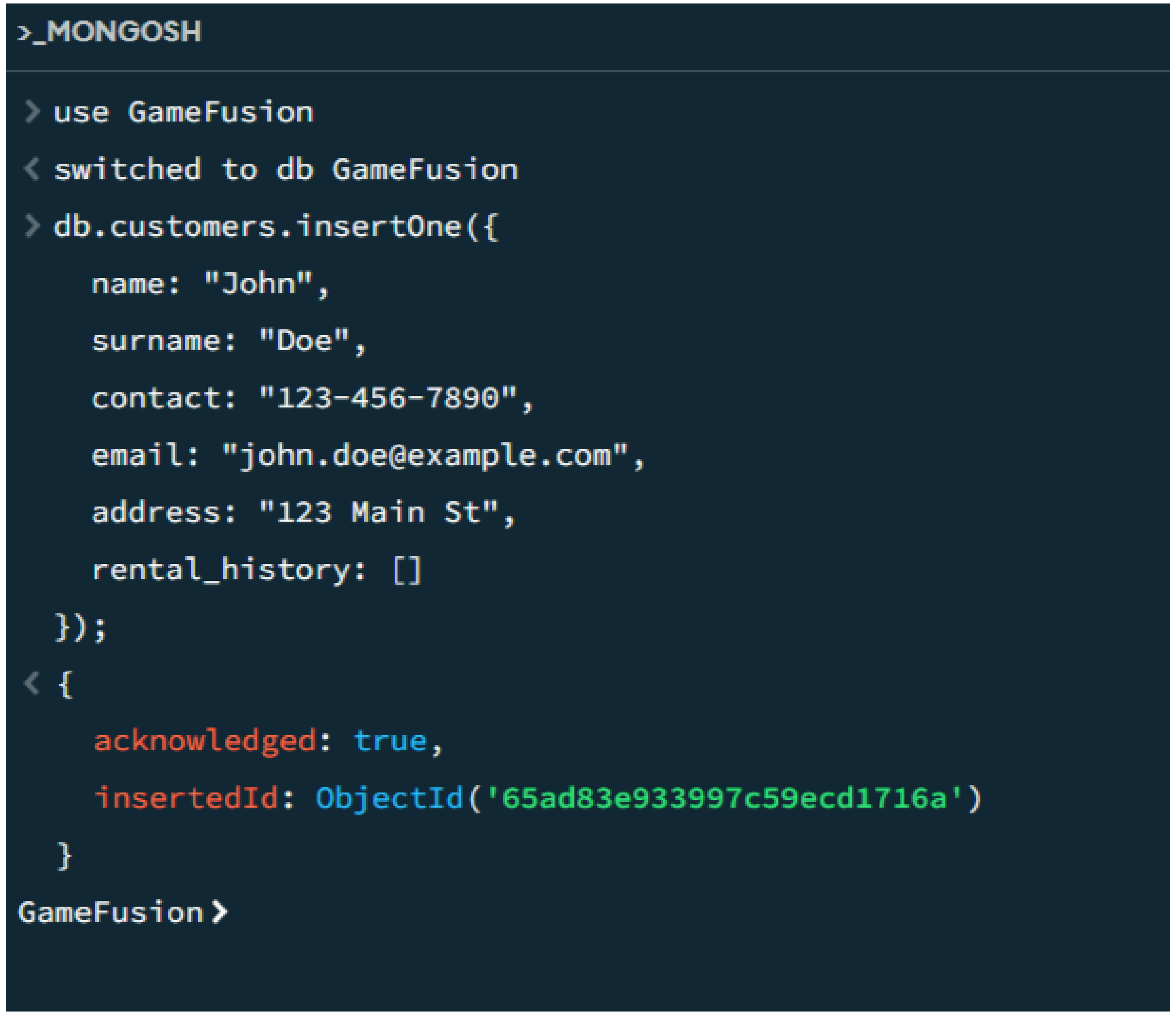


```javascript
        rental_id: i + 1,
        customer_id: customer.customer_id,
        console_id: console.console_id,
        location_id: location.location_id,
        rental_type: i < 400 ? 'home delivery' : 'in-store', // 400 home delivery, 600 in-store
        duration: faker.random.number({ min: 2, max: 24 }), // Random duration between 2 and 24 hours
        deposit: parseInt(customer.id_card), // ID card as deposit
        total_cost: (2 + faker.random.number({ min: 1, max: 22 })) * 1000, // Random total cost (minimum rent duration is 2 hou
      };

      rentals.push(rental);
      // Update rental_hours for consoles and customers
      console.rental_hours += rental.duration;
      customer.rental_hours += rental.duration;
      // Update rental_history in customers and consoles
      customer.rental_history.push(rental);
      console.rental_history.push(rental);
    }
```

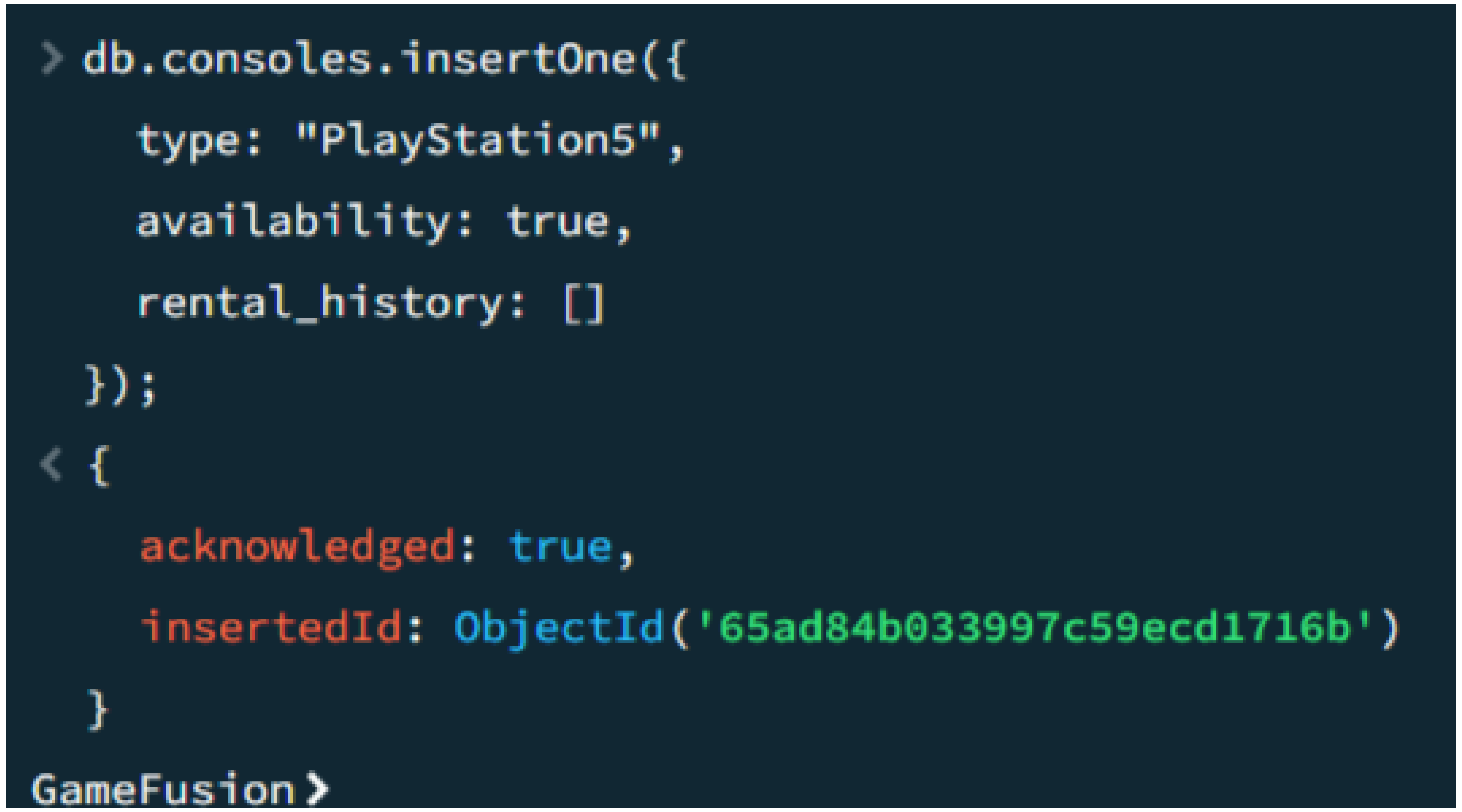
```javascript

    // Insert data into MongoDB
    await db.collection('customers').insertMany(customers);
    await db.collection('consoles').insertMany(consoles);
    await db.collection('locations').insertMany(locations);
    await db.collection('rentals').insertMany(rentals);

    console.log('Fake data generated and inserted into MongoDB');
```

CRUD OPERATIONS EXAMPLE :

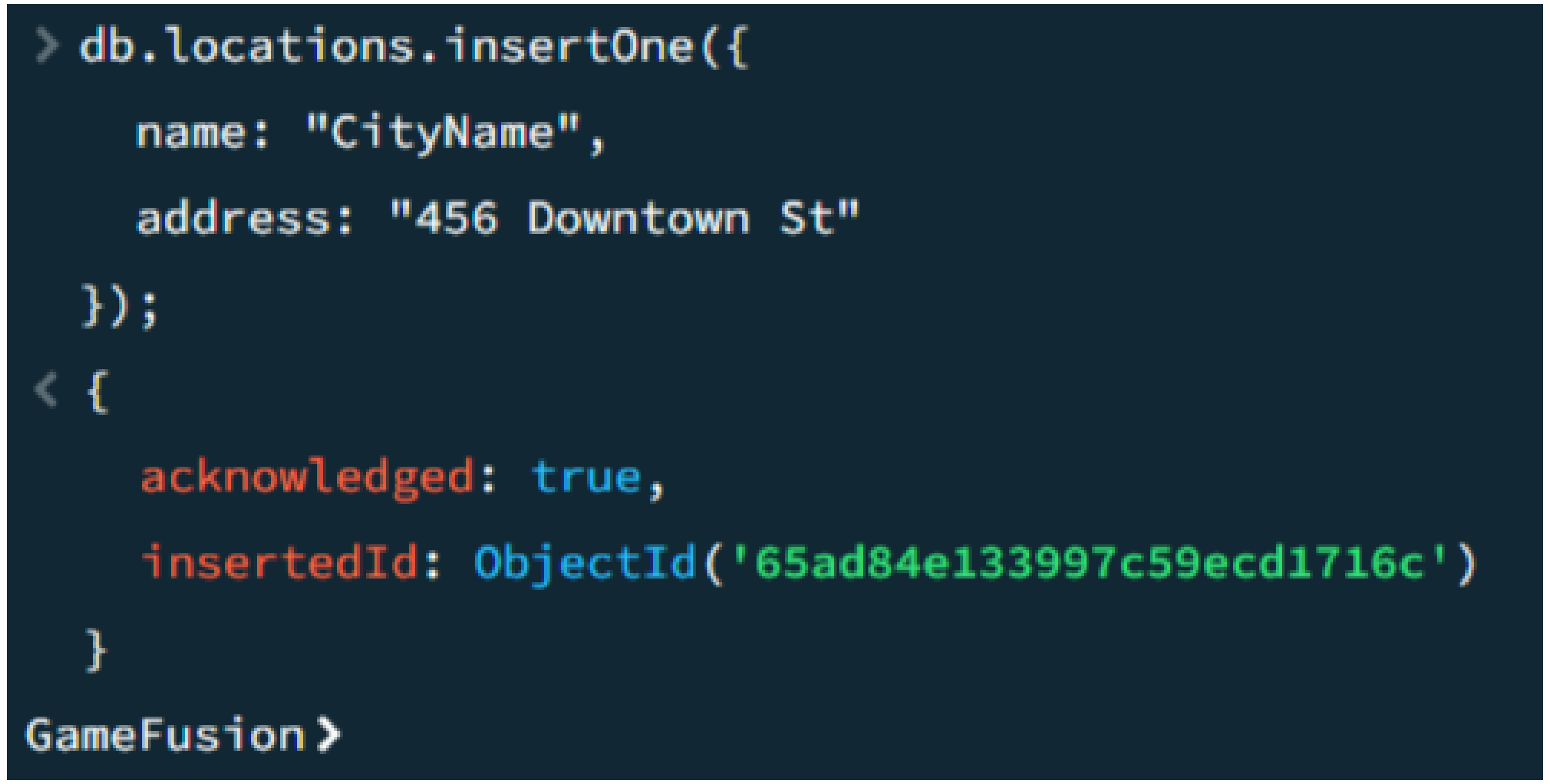
CREATE:



```
>_MONGOSH
> use GameFusion
< switched to db GameFusion
> db.customers.insertOne({
    name: "John",
    surname: "Doe",
    contact: "123-456-7890",
    email: "john.doe@example.com",
    address: "123 Main St",
    rental_history: []
  });
< {
    acknowledged: true,
    insertedId: ObjectId('65ad83e933997c59ecd1716a')
  }
GameFusion>
```



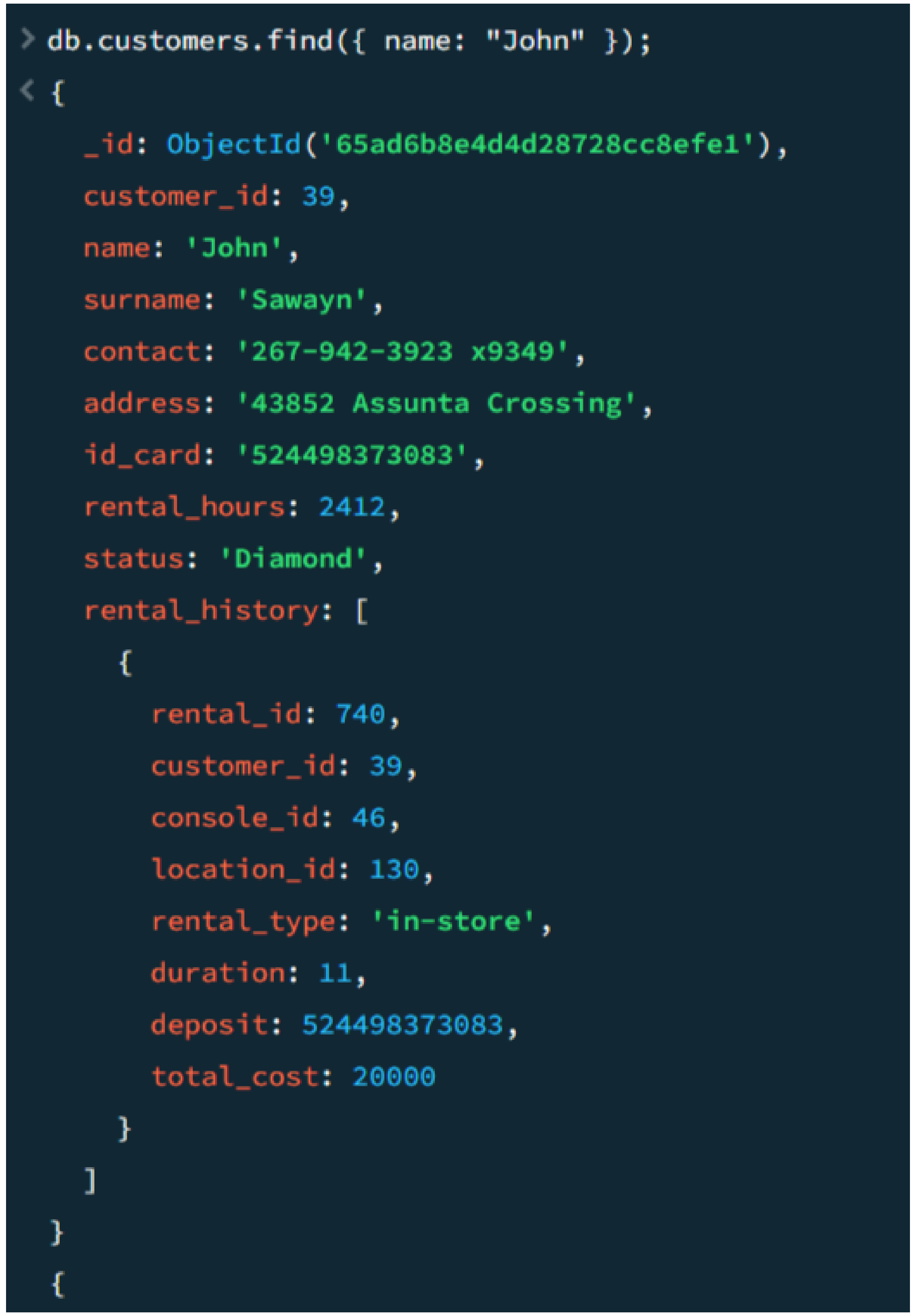
```
> db.consoles.insertOne({
    type: "PlayStation5",
    availability: true,
    rental_history: []
  });
< {
    acknowledged: true,
    insertedId: ObjectId('65ad84b033997c59ecd1716b')
  }
GameFusion>
```



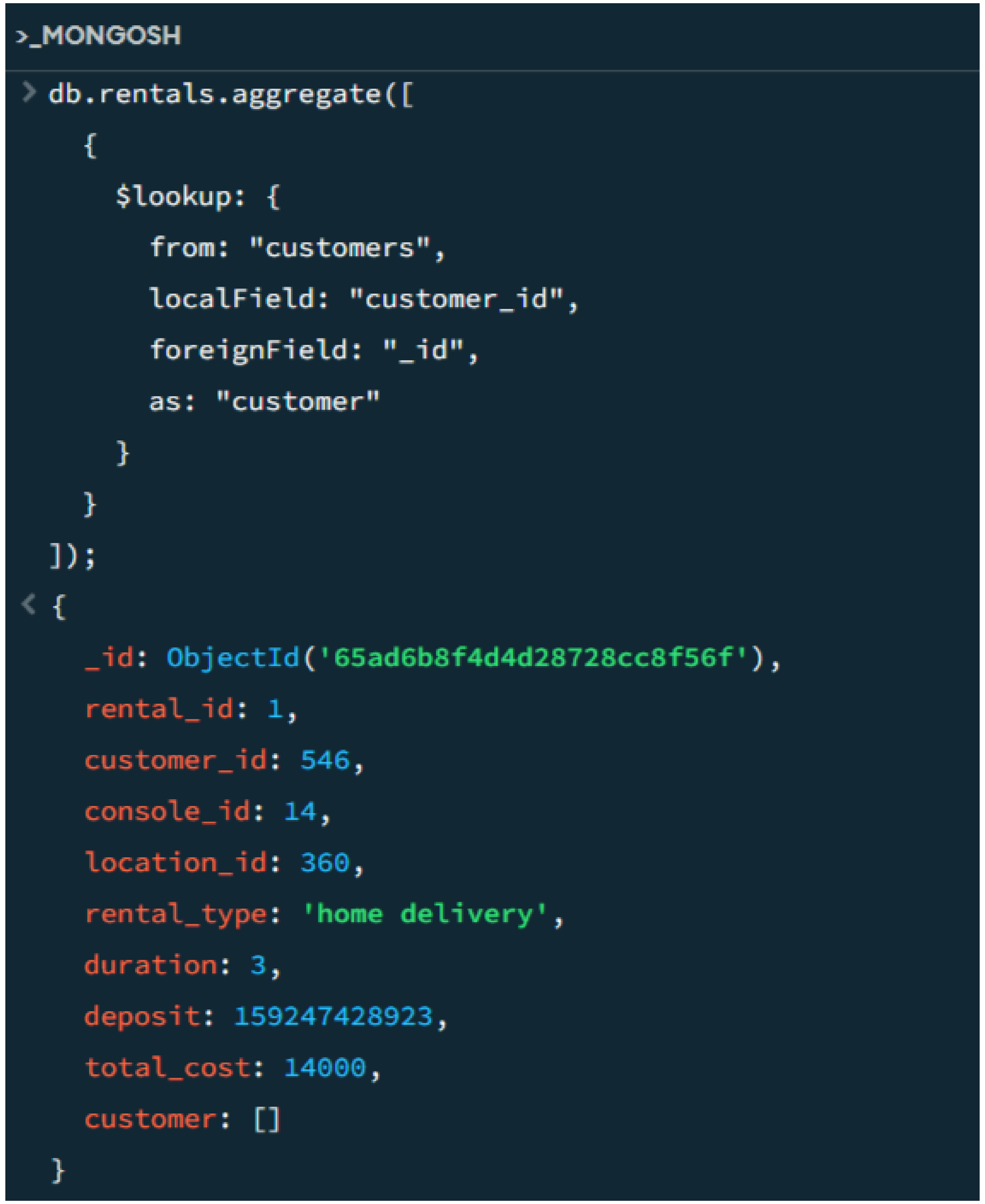
```
> db.locations.insertOne({
    name: "CityName",
    address: "456 Downtown St"
  });
< {
    acknowledged: true,
    insertedId: ObjectId('65ad84e133997c59ecd1716c')
  }
GameFusion>
```

CRUD OPERATIONS EXAMPLE :

READ:



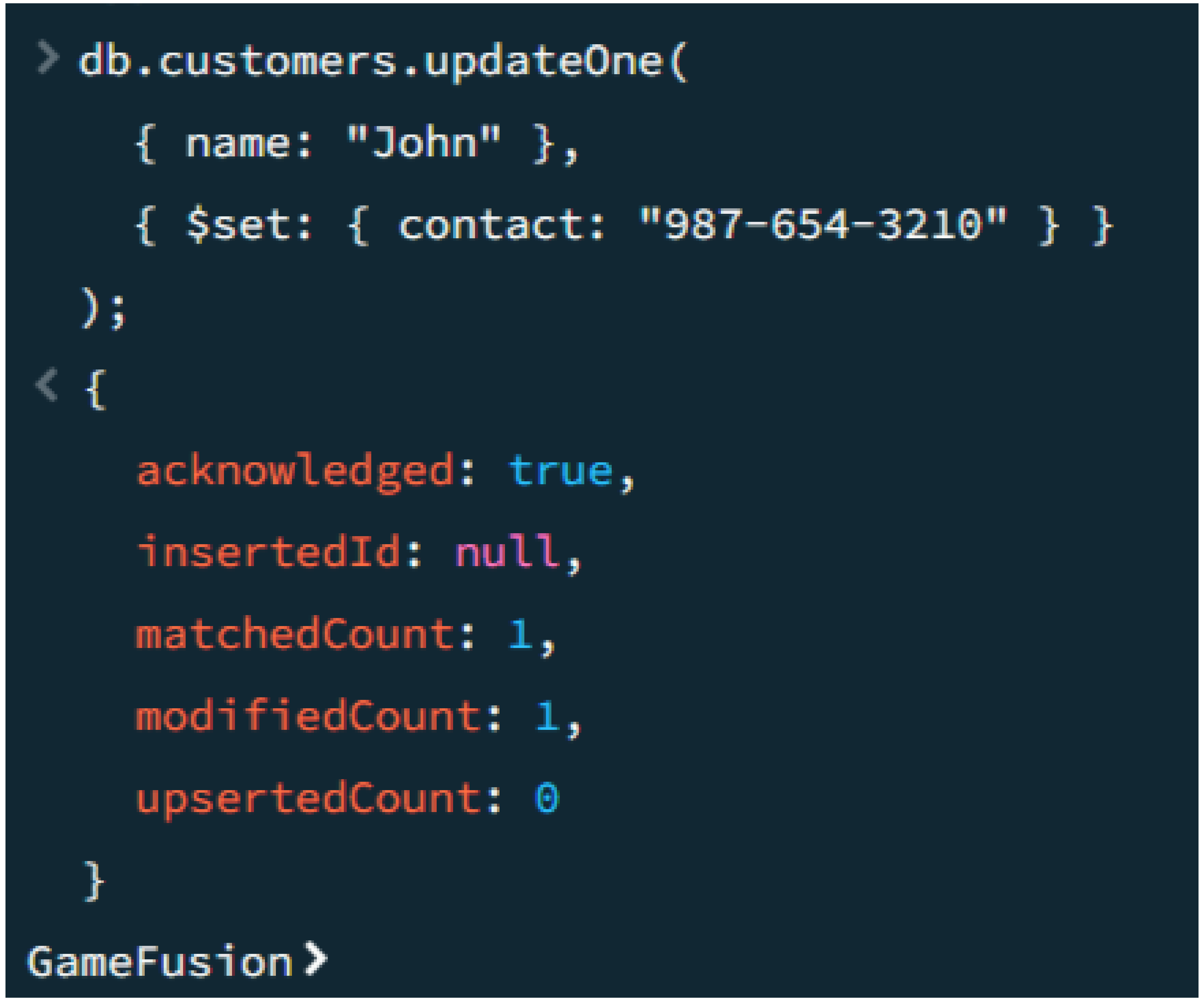
```
> db.customers.find({ name: "John" });
< {
    _id: ObjectId('65ad6b8e4d4d28728cc8efe1'),
    customer_id: 39,
    name: 'John',
    surname: 'Sawayn',
    contact: '267-942-3923 x9349',
    address: '43852 Assunta Crossing',
    id_card: '524498373083',
    rental_hours: 2412,
    status: 'Diamond',
    rental_history: [
      {
        rental_id: 740,
        customer_id: 39,
        console_id: 46,
        location_id: 130,
        rental_type: 'in-store',
        duration: 11,
        deposit: 524498373083,
        total_cost: 20000
      }
    ]
  }
  {
```



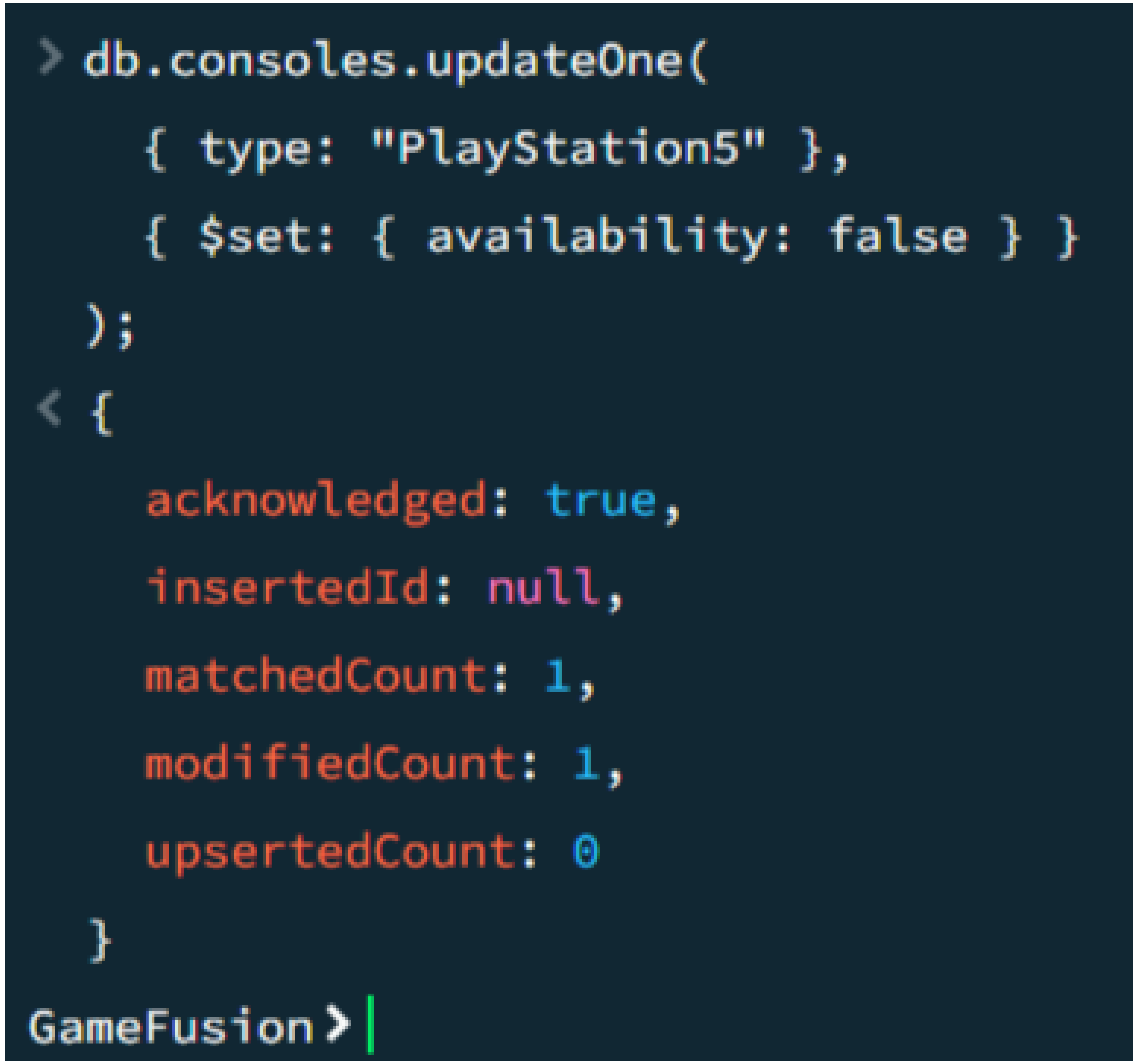
```
> db.rentals.aggregate([
    {
      $lookup: {
        from: "customers",
        localField: "customer_id",
        foreignField: "_id",
        as: "customer"
      }
    }
  ]);
< {
    _id: ObjectId('65ad6b8f4d4d28728cc8f56f'),
    rental_id: 1,
    customer_id: 546,
    console_id: 14,
    location_id: 360,
    rental_type: 'home delivery',
    duration: 3,
    deposit: 159247428923,
    total_cost: 14000,
    customer: []
  }
```

CRUD OPERATIONS EXAMPLE :

UPDATE:



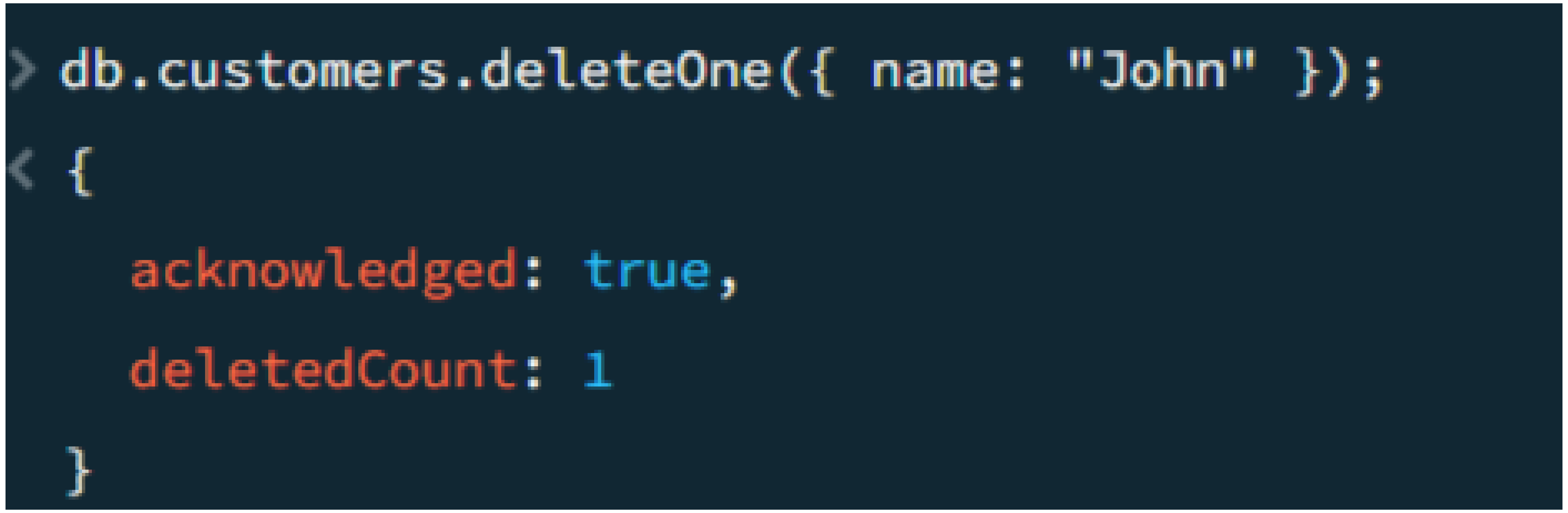
```
> db.customers.updateOne(
    { name: "John" },
    { $set: { contact: "987-654-3210" } }
  );
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
GameFusion>
```



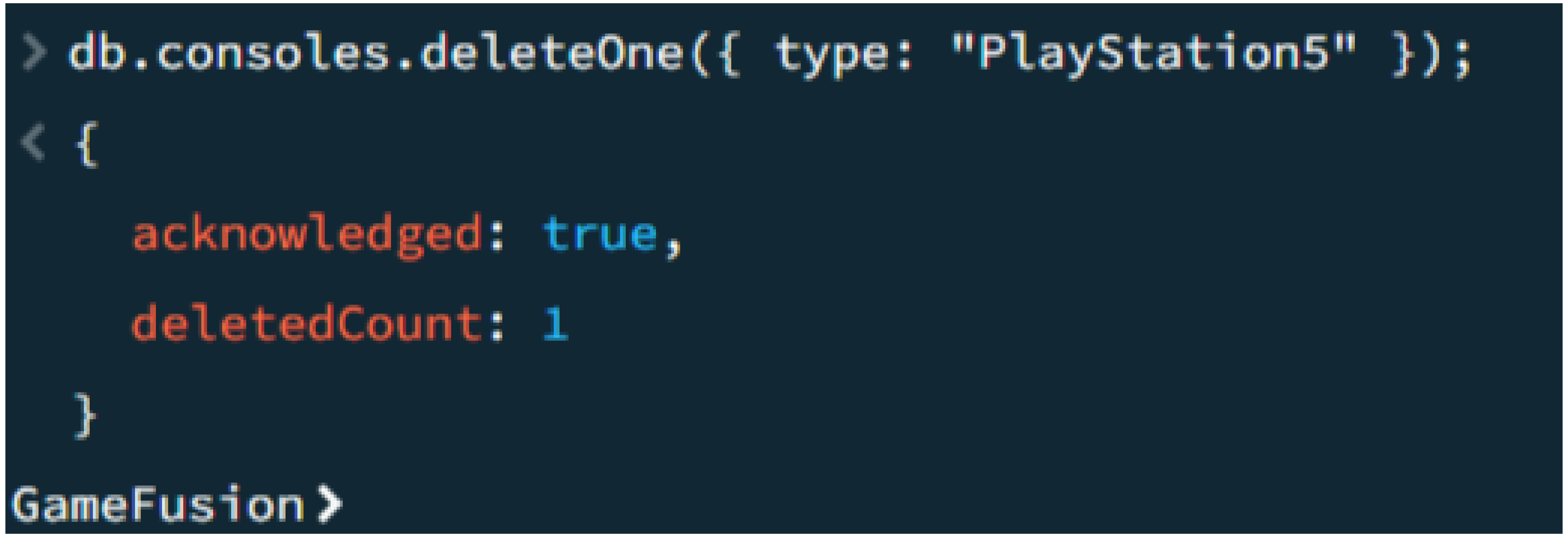
```
> db.consoles.updateOne(
    { type: "PlayStation5" },
    { $set: { availability: false } }
  );
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
GameFusion>
```

CRUD OPERATIONS EXAMPLE :

DELETE:



```
> db.customers.deleteOne({ name: "John" });
< {
    acknowledged: true,
    deletedCount: 1
  }
```



```
> db.consoles.deleteOne({ type: "PlayStation5" });
< {
    acknowledged: true,
    deletedCount: 1
  }
GameFusion>
```

THANK YOU

LINK TO THE GITHUB (DATABASE, WITH FAKER HERE):
HTTPS://GITHUB.COM/KUSSAINOVANSAR/GAMEFUSIONRENTAL