Games Collection

```
{
  "game_ID": 1,
  "title": "Epic Quest",
  "developer": "QuestWorks",
  "releaseDate": "2023-01-01",
  "platform": "PC"
}
```

Players Collection

```
{
  "username": "hero123",
  "country": "USA",
  "joinDate": "2022-12-01",
  "gamesPlayed": [
    {
      "game_ID": 1,
      "title": "Epic Quest"
    },
    {
      "game_ID": 2,
      "title": "Another Adventure"
    }
  ]
}
```

Categories Collection

```
{
  "category_ID": 1,
  "game": {
    "game_ID": 1,
    "title": "Epic Quest"
  },
  "name": "Speedrun",
  "description": "Complete the game as fast as possible."
}
```

Runs Collection (Ranking Included)

```
{
  "run_ID": 1,
```

```json
  "game": {
    "game_ID": 1,
    "title": "Epic Quest"
  },
  "player": {
    "username": "hero123"
  },
  "category": {
    "category_ID": 1,
    "name": "Speedrun"
  },
  "completionTime": 3600,
  "completionDate": "2024-02-15",
  "videoLink": "http://example.com/video/1",
  "ranking": {
    "rank": 1
  }
}
```

Additional Notes for Database Engineers

Referencing vs. Embedding: Consider the trade-offs between referencing and embedding documents. Referencing (storing the ID of another document) avoids duplication but requires additional queries or join operations (aggregation pipeline in MongoDB). Embedding documents provides faster reads and simpler queries at the expense of potentially larger documents and more complex updates if the embedded information changes frequently.

Normalization: While document stores like MongoDB excel at handling denormalized data, consider the implications for data consistency, especially in cases where information like game titles or player usernames might change. Use MongoDB's powerful aggregation framework to maintain data integrity and performance.

Indexes: Ensure that indexes are created for frequently queried fields (e.g., game_ID, username, category_ID) to improve query performance.

Schema Evolution: Document databases allow for flexible schema evolution. While this flexibility is advantageous, maintaining some level of schema consistency across documents can help avoid confusion and errors in application development.