Query:

db.games.insertMany([

{

"name": "Spy Hunter",

"genre": "Racing",

"rating": 76,

"achievements": ["Speed Demon"]

},

{

"name": "Mario Kart 64",

"genre": "Racing",

"rating": 96,

"achievements": ["Game Master", "Speed Demon"]

},

{

"name": "Tetris",

"genre": "Puzzle",

"rating": 83,

"achievements": ["Puzzle Solver"]

},

{

"name": "Mega Man 5",

"genre": "Platformer",

"rating": 81,

"achievements": ["Robot Master"]

},

{

"name": "Star Fox",

"genre": "Action",

"rating": 71

},

{

"name": "The Legend of Zelda: Ocarina of Time",

"genre": "Action"

},

{

"name": "Banjo-Kazooie",

"genre": "Platformer",

"rating": 92,

"achievements": ["Game Master", "Speed Demon"]

}

]);

// Confirm insertion

print("Games inserted successfully.");

Output:

{

"acknowledged" : true,

"insertedIds" : [

ObjectId("6626c23aa620cfbeca17ef2e"),

ObjectId("6626c23aa620cfbeca17ef2f"),

ObjectId("6626c23aa620cfbeca17ef30"),

ObjectId("6626c23aa620cfbeca17ef31"),

ObjectId("6626c23aa620cfbeca17ef32"),

ObjectId("6626c23aa620cfbeca17ef33"),

ObjectId("6626c23aa620cfbeca17ef34")

]

}

Games inserted successfully.

**(3)** Retrieve all the games in the collection.

Query:

var allGames = db.games.find();

allGames.forEach(function(game) {

printjson(game);

});

Output:

{

"\_id" : ObjectId("6626c8b6c3af7369f67446c2"),

"name" : "Spy Hunter",

"genre" : "Racing",

"rating" : 76,

"achievements" : [

"Speed Demon"

]

}

{

"\_id" : ObjectId("6626c8b6c3af7369f67446c3"),

"name" : "Mario Kart 64",

"genre" : "Racing",

"rating" : 96,

"achievements" : [

"Game Master",

"Speed Demon"

]

}

{

"\_id" : ObjectId("6626c8b6c3af7369f67446c4"),

"name" : "Tetris",

"genre" : "Puzzle",

"rating" : 83,

"achievements" : [

"Puzzle Solver"

]

}

{

"\_id" : ObjectId("6626c8b6c3af7369f67446c5"),

"name" : "Mega Man 5",

"genre" : "Platformer",

"rating" : 81,

"achievements" : [

"Robot Master"

]

}

{

"\_id" : ObjectId("6626c8b6c3af7369f67446c6"),

"name" : "Star Fox",

"genre" : "Action",

"rating" : 71

}

{

"\_id" : ObjectId("6626c8b6c3af7369f67446c7"),

"name" : "The Legend of Zelda: Ocarina of Time",

"genre" : "Action"

}

{

"\_id" : ObjectId("6626c8b6c3af7369f67446c8"),

"name" : "Banjo-Kazooie",

"genre" : "Platformer",

"rating" : 92,

"achievements" : [

"Game Master",

"Speed Demon"

]

}

**(4)** Display only the name and genre of all games, excluding their \_id.

Query:

var projection = { \_id: 0, name: 1, genre: 1 };

var games = db.games.find({}, projection);

games.forEach(function(game) {

print("Name: " + game.name + ", Genre: " + game.genre);

});

Output:

Name: Spy Hunter, Genre: Racing

Name: Mario Kart 64, Genre: Racing

Name: Tetris, Genre: Puzzle

Name: Mega Man 5, Genre: Platformer

Name: Star Fox, Genre: Action

Name: The Legend of Zelda: Ocarina of Time, Genre: Action

Name: Banjo-Kazooie, Genre: Platformer

A screenshot of a computer

Description automatically generated

**(5)** Locate a specific game by its name, such as “Mario Kart 64”, without employing limit(). Instead, use the findOne function.

Query:

var game = db.games.findOne({ name: "Mario Kart 64" });

printjson(game);

Result:

A screenshot of a computer code

Description automatically generated

**(6)** Find the top 3 highest-rated games.

Query:

var topRatedGames = db.games.find().sort({ rating: -1 }).limit(3);

var rank = 1;

topRatedGames.forEach(function(game) {

print("Rank " + rank + ": " + game.name + " - Rating: " + game.rating);

rank++;

});

Output:

Rank 1: Mario Kart 64 - Rating: 96

Rank 2: Banjo-Kazooie - Rating: 92

Rank 3: Tetris - Rating: 83

A screenshot of a number of text

Description automatically generated

**(7)** List all unique genres available in the collection.

Query:

var uniqueGenres = db.games.distinct("genre");

printjson(uniqueGenres);

Output:

A close up of a sign

Description automatically generated

**(8)** Find games with ratings above 90.

Query:

var highlyRatedGames = db.games.find({ rating: { $gt: 90 } });

highlyRatedGames.forEach(function(game) {

printjson(game);

});

Output:

{

"\_id" : ObjectId("6626cac15f4a022ac5a070fc"),

"name" : "Mario Kart 64",

"genre" : "Racing",

"rating" : 96,

"achievements" : [

"Game Master",

"Speed Demon"

]

}

{

"\_id" : ObjectId("6626cac15f4a022ac5a07101"),

"name" : "Banjo-Kazooie",

"genre" : "Platformer",

"rating" : 92,

"achievements" : [

"Game Master",

"Speed Demon"

]

}

A computer screen shot of a computer program

Description automatically generated

**(9)** Find all games except those in the ‘Racing’ genre.

Query:

var nonRacingGames = db.games.find({ genre: { $ne: "Racing" } });

nonRacingGames.forEach(function(game) {

printjson(game);

});

Result:

{

"\_id" : ObjectId("6626cb1ac8a288ca88fbcfac"),

"name" : "Tetris",

"genre" : "Puzzle",

"rating" : 83,

"achievements" : [

"Puzzle Solver"

]

}

{

"\_id" : ObjectId("6626cb1ac8a288ca88fbcfad"),

"name" : "Mega Man 5",

"genre" : "Platformer",

"rating" : 81,

"achievements" : [

"Robot Master"

]

}

{

"\_id" : ObjectId("6626cb1ac8a288ca88fbcfae"),

"name" : "Star Fox",

"genre" : "Action",

"rating" : 71

}

{

"\_id" : ObjectId("6626cb1ac8a288ca88fbcfaf"),

"name" : "The Legend of Zelda: Ocarina of Time",

"genre" : "Action"

}

{

"\_id" : ObjectId("6626cb1ac8a288ca88fbcfb0"),

"name" : "Banjo-Kazooie",

"genre" : "Platformer",

"rating" : 92,

"achievements" : [

"Game Master",

"Speed Demon"

]

}

**(10)** Identify games that are missing the ‘rating’ attribute.

Query:

var gamesWithoutRating = db.games.find({ rating: { $exists: false } });

gamesWithoutRating.forEach(function(game) {

printjson(game);

});

Output:

{

"\_id" : ObjectId("6626cb86125e33eb5c0ae5b3"),

"name" : "The Legend of Zelda: Ocarina of Time",

"genre" : "Action"

}

**(11)** Count the number of games per genre

Query:

var genreCounts = db.games.aggregate([

{ $group: { \_id: "$genre", count: { $sum: 1 } } }

]);

genreCounts.forEach(function(genreCount) {

print("Genre: " + genreCount.\_id + ", Count: " + genreCount.count);

});

Result:

Genre: Racing, Count: 2

Genre: Platformer, Count: 2

Genre: Puzzle, Count: 1

Genre: Action, Count: 2

**(12)** Calculate the average rating for each genre.

Query:

var averageRatingByGenre = db.games.aggregate([

{

$group: {

\_id: "$genre",

averageRating: { $avg: "$rating" }

}

}

]);

averageRatingByGenre.forEach(function(genreRating) {

print("Genre: " + genreRating.\_id + ", Average Rating: " + genreRating.averageRating.toFixed(2));

});

Output:

Genre: Platformer, Average Rating: 86.50

Genre: Racing, Average Rating: 86.00

Genre: Puzzle, Average Rating: 83.00

Genre: Action, Average Rating: 71.00

**(13)** Locate all the games that possesses both the “Game Maser” and the “Speed Demon” achievements.

Query:

var gamesWithAchievements = db.games.find({ achievements: { $all: ["Game Master", "Speed Demon"] } });

gamesWithAchievements.forEach(function(game) {

printjson(game);

});

Output:

{

"\_id" : ObjectId("6626cc53ccb6f9abe447f4bb"),

"name" : "Mario Kart 64",

"genre" : "Racing",

"rating" : 96,

"achievements" : [

"Game Master",

"Speed Demon"

]

}

{

"\_id" : ObjectId("6626cc53ccb6f9abe447f4c0"),

"name" : "Banjo-Kazooie",

"genre" : "Platformer",

"rating" : 92,

"achievements" : [

"Game Master",

"Speed Demon"

]

}

**14)** Employee the update() function to modify a game named with “Star Fox” by adding two achievements with the following properties:

"name": "Game Master", "points": 100

"name": "Speed Demon", "points": 135

Query:

db.games.update(

{ name: "Star Fox" }, // Specify the game to update

{

$push: {

achievements: { $each: [

{ name: "Game Master", points: 100 },

{ name: "Speed Demon", points: 135 }

]}

}

}

);

Result:



**(15)** Add a common achievement “Fan Favorite” to all games with a rating of 90 or higher.

Query:

db.games.updateMany(

{ rating: { $gte: 90 } }, // Select games with a rating of 90 or higher

{ $push: { achievements: { name: "Fan Favorite" } } } // Add the "Fan Favorite" achievement

);

Output:



**(16)** Delete a game named with “Banjo-Kazooie”.

Query:

db.games.deleteOne({ name: "Banjo-Kazooie" });

Result:

