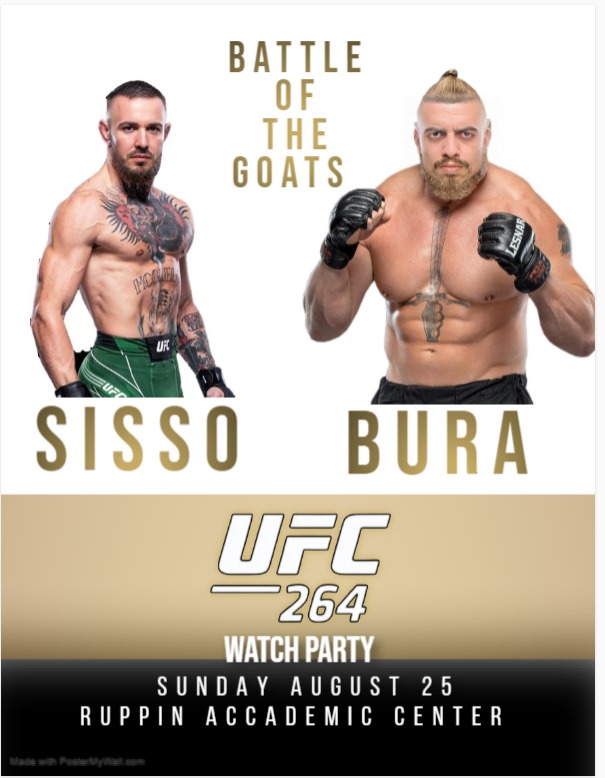
**פרויקט גמר MongoDB**

מגישים:

אלמוג סיסו 204307516

אלמוג בורה 206295115



**תוכן עניינים**

[פירוט הקבצים להגשה ב ZIP 3](#_Toc175309194)

[הקדמה 3](#_Toc175309195)

[Data Modeling 4](#_Toc175309196)

[שאילתות 5](#_Toc175309197)

[1. הקדמה 5](#_Toc175309198)

[2. האלופים הנוכחיים 5](#_Toc175309199)

[3. לוחמים לפי קטגורית משקל 5](#_Toc175309200)

[4. הענקים של הזירה 5](#_Toc175309201)

[5. סטטיסטיקת הניצחונות 6](#_Toc175309202)

[6. קידום הכוכבים העתידיים 6](#_Toc175309203)

[7. סיכום האירוע 7](#_Toc175309204)

[8. גיבוי נתונים קריטי 7](#_Toc175309205)

[9. שימור תהילת האלופים 8](#_Toc175309206)

[10. ניקוי מסד הנתונים 9](#_Toc175309207)

[11. הוספת לוחם חדש 10](#_Toc175309208)

[12. הוספת אירוע וקרב 11](#_Toc175309209)

[13. עדכון – אלוף חדש 11](#_Toc175309210)

[14. עדכון מערכת השיפוט 12](#_Toc175309211)

[15. שינוי תוצאות ב-UFC 236: הפתעה בקרב האליפות 13](#_Toc175309212)

[16. המשקל כן קובע : ניתוח מחלקות המשקל ב UFC 14](#_Toc175309213)

[17. המיקס המנצח: ניתוח שילובי אומנויות לחימה ב UFC 15](#_Toc175309214)

[18. מעצמת ה-MMA: זיהוי המדינה המובילה בייצוא לוחמי UFC 16](#_Toc175309215)

[19. מלך הכסף: חשיפת הלוחם המרוויח ביותר ב-UFC 17](#_Toc175309216)

[20. סמבו באמריקה: מהפכת האימונים ב-UFC 18](#_Toc175309217)

[21. עלייה לגדולה: ג'ון ג'ונס כובש את תואר האלוף במשקל כבד 19](#_Toc175309218)

[עבודה עם Claude AI 20](#_Toc175309219)

# פירוט הקבצים להגשה ב ZIP

קבצי קוד וטקסט :

* insert\_Fighters.js
* insert\_Events.js
* All\_Queries.txt

קובץ word עם אפיון והסבר על השאילתות

# הקדמה

ה-Ultimate Fighting Championship (UFC) היא הפלטפורמה המובילה בעולם למגרשי הקרב בסגנון Mixed Martial Arts (MMA). הוקמה ב-1993 וזכתה לפופולריות עולמית בשל קרבותיה המרגשים שבהם מתמודדים לוחמים מכל רחבי הגלובוס. ה-UFC מציעה קרבות בקטגוריות משקל שונות ומקדמת אירועים חיים והפכה למותג ספורט אולטימטיבי וגלובלי.

דיינה וויט, הנשיא של ארגון ה UFC פנה לצוות שלנו עם בקשה מיוחדת. בשל צמיחת הארגון והצורך בניהול יעיל של נתונים, וויט ביקש להקים מסד נתונים מורכב ומדויק המכיל את כל המידע הנדרש על הלוחמים, האירועים, הקרבות והאלופים.

**הנחות עבודה כלליות :**

* סט הנתונים הוא אמיתי, ולכן כל הקרבות האירועים והלוחמים מבוססים על המציאות, אך מוגבלים עד לתאריך 11/11/23 , ולכן חלק מהשאילתות יהיו עד השנה הנ"ל.
* המטלה ב SQL והמטלה ב MongoDB בוצעו על אותו סט נתונים, בהתחלה עשינו את המטלה על SQL ולכן מומלץ לבדוק אותה קודם על מנת להכיר את הנושא טוב יותר.

# Data Modeling

1. **Fighters:** 
   1. \_id (מזהה ייחודי)
   2. firstName
   3. lastName
   4. nickname
   5. dateOfBirth
   6. joinedUFC
   7. nationality
   8. heightCm
   9. reachCm
   10. weightLbs
   11. weightClass
   12. **Record: (Embedded)**
       1. wins
       2. losses
       3. draws
       4. noContests
       5. knockouts
       6. submissions
   13. martialArts (מערך של סגנונות לחימה)
   14. championships (מערך של אליפויות):
       1. weightClass
       2. startDate
       3. endDate
2. **Events:**
3. \_id (מזהה ייחודי)
4. eventName
5. eventDate
6. eventLocation
7. country
8. **fights (Embedded):**
9. fightId(מזהה ייחודי)
10. fighter1 (Fightersהפניה ל)
11. fighter2 (Fightersהפניה ל)
12. weightClass
13. winner (Fightersהפניה ל)
14. winMethod
15. roundNumber

# שאילתות

## הקדמה

לצורך הקמת בסיס הנתונים, צוות הפיתוח של ה-UFC כתב את הקוד בפורמט JSON . לאחר שהקוד נכתב ונבדק, הצוות פשוט הריץ את הקוד הנ"ל, מה שאפשר להם ליצור ולאכלס את בסיס הנתונים במהירות וביעילות.

Use UFC\_DataBase

load("C:/Users/almog/Almog\_codes/mongoDB/insert\_Fighters.js")

load("C:/Users/almog/Almog\_codes/mongoDB/insert\_Events.js")

## האלופים הנוכחיים

דיינה, מעוניין לסקור את האלופים הנוכחיים בכל קטגוריית משקל. הוא מתכנן לחפש בטבלת האליפויות את הלוחמים שתאריך סיום האליפות שלהם ריק, המציין את מעמדם כאלופים נוכחיים.

db.Fighters.find(

{ "championships.endDate": null },

{ firstName: 1, lastName: 1, championships: 1, \_id: 0 }

).forEach(fighter => {

fighter.championships.forEach(championship => {

if (championship.endDate === null) {

const startDate = new Date(championship.startDate);

const today = new Date();

const daysHeld = Math.floor((today - startDate) / (1000 \* 60 \* 60 \* 24)); // Calculate days

print(`${fighter.firstName} ${fighter.lastName} holds the "${championship.weightClass}" championship from ${championship.startDate} until now (${daysHeld} days).`);

}

});

});

## לוחמים לפי קטגורית משקל

דיינה וייט מעוניין לקבל תמונה מקיפה של מספר הלוחמים בכל קטגוריית משקל. הוא מתכנן לערוך ניתוח שיציג את הנתונים הללו בסדר עולה, כדי להבין טוב יותר את המבנה והעומק של כל קטגוריה**.**

db.Fighters.aggregate([

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

{ $sort: { \_id: 1 } }

]).forEach(result =>

print(`In the "${result.\_id}" weight class, there are ${result.count} fighters`)

);

## הענקים של הזירה

דיינה וייט מתעניין בלוחמים הגדולים ביותר בארגון. הוא מחליט לאתר את שלושת הלוחמים הכבדים ביותר, סקרן לגבי השפעת משקלם על קרבותיהם.

db.Fighters.find(

{},

{ firstName: 1, lastName: 1, weightLbs: 1 }

).sort({ weightLbs: -1 }).limit(3).toArray().forEach(fighter =>

print(`Name: ${fighter.firstName} ${fighter.lastName}, Weight: ${fighter.weightLbs} lbs`));

## סטטיסטיקת הניצחונות

דיינה וייט, תמיד מתעניין בביצועי הלוחמים, מחליט לבחון את אחוזי הניצחון של כל לוחם. הוא מבקש מצוות הנתונים לחשב את הסטטיסטיקה הזו עבור כל לוחם, בהתבסס אך ורק על הרקורד שלהם.

db.Fighters.aggregate([

{ $addFields: {

totalFights: { $sum: ["$record.wins", "$record.losses", "$record.draws", "$record.noContests"] },

winPercentage: { $cond: [ { $gt: [{ $sum: ["$record.wins", "$record.losses", "$record.draws", "$record.noContests"] }, 0] },

{ $multiply: [{ $divide: ["$record.wins", { $sum: ["$record.wins", "$record.losses", "$record.draws", "$record.noContests"] }] }, 100] }, 0 ] }

}

},

{ $sort: { winPercentage: -1 } },

{ $limit: 10 },

{ $project: { firstName: 1, lastName: 1, nickname: 1, "record.wins": 1, totalFights: 1, winPercentage: { $round: ["$winPercentage", 2] } } }

]).forEach(fighter => {

print(`Fighter: ${fighter.firstName} ${fighter.lastName} (${fighter.nickname})`);

print(`Wins: ${fighter.record.wins}`);

print(`Total Fights: ${fighter.totalFights}`);

print(`Win Percentage: ${fighter.winPercentage}%`);

print('------------------------');

});

## קידום הכוכבים העתידיים

דיינה וייט, בחיפושיו אחר כישרונות מבטיחים, מתמקד בלוחמים שנמצאים מחוץ לעשירייה הראשונה. הוא מבקש לראות את הלוחמים המדורגים במקומות 11 עד 20, בתקווה לזהות פוטנציאל לקידום ולפיתוח כוכבים חדשים בזירה.

db.Fighters.aggregate([

{ $addFields: {

totalFights: { $sum: ["$record.wins", "$record.losses", "$record.draws", "$record.noContests"] },

winPercentage: { $cond: [ { $gt: [{ $sum: ["$record.wins", "$record.losses", "$record.draws", "$record.noContests"] }, 0] },

{ $multiply: [{ $divide: ["$record.wins", { $sum: ["$record.wins", "$record.losses", "$record.draws", "$record.noContests"] }] }, 100] }, 0 ] }

}

},

{ $sort: { winPercentage: -1 } },

{ $skip: 10 },

{ $limit: 10 },

{ $project: { firstName: 1, lastName: 1, nickname: 1, "record.wins": 1, totalFights: 1, winPercentage: { $round: ["$winPercentage", 2] } } }

]).forEach(fighter => {

print(`Fighter: ${fighter.firstName} ${fighter.lastName} (${fighter.nickname})`);

print(`Wins: ${fighter.record.wins}`);

print(`Total Fights: ${fighter.totalFights}`);

print(`Win Percentage: ${fighter.winPercentage}%`);

print('------------------------');

});

## סיכום האירוע

לאחר כל אירוע UFC, דיינה וייט מבקש לקבל סיכום מקיף. הוא רוצה לראות את תוצאות כל הקרבות שהתקיימו באירוע, כדי לנתח את הביצועים ולהעריך את ההשלכות על דירוגי הלוחמים ועל התחרות העתידית.

// Function to get a summary of a UFC event

function getEventSummary(eventId) {

// Find the event by ID

var event = db.Events.findOne({ \_id: eventId });

if (!event) return { error: "Event not found" };

var summary = {

eventName: event.eventName,

eventDate: event.eventDate,

fights: []

};

// Process each fight in the event

event.fights.forEach(function(fight) {

var fighter1 = db.Fighters.findOne({ \_id: parseInt(fight.fighter1) });

var fighter2 = db.Fighters.findOne({ \_id: parseInt(fight.fighter2) });

if (!fighter1 || !fighter2) {

summary.fights.push({ error: "Fighter details not found for this fight" });

return;

}

var fightSummary = {

weightClass: fight.weightClass,

description: ""

};

if (fight.winner) {

var winner = fight.winner === fight.fighter1 ? fighter1 : fighter2;

var loser = fight.winner === fight.fighter1 ? fighter2 : fighter1;

fightSummary.description = `${winner.firstName} ${winner.lastName} defeated ${loser.firstName} ${loser.lastName}` +

(fight.winMethod ? ` by ${fight.winMethod}` : '') +

(fight.roundNumber ? ` in round ${fight.roundNumber}` : '');

} else {

fightSummary.description = `${fighter1.firstName} ${fighter1.lastName} fought ${fighter2.firstName} ${fighter2.lastName} to a draw`;

}

summary.fights.push(fightSummary);

});

return summary;

}

// Example usage

printjson(getEventSummary(229));

## גיבוי נתונים קריטי

דיינה וייט מורה על שכפול מלא של בסיס הנתונים. מטרתו ליצור גיבוי מקיף, המבטיח שמירה בטוחה של כל המידע החיוני בארגון.

// Create backups of all data

db.Fighters.find().forEach(fighter => db.Fighters\_Backup.insertOne(fighter));

db.Events.find().forEach(event => db.Events\_Backup.insertOne(event));

## שימור תהילת האלופים

דיינה וייט מבקש להעתיק את טבלת האלופים בלבד.

db.Fighters.aggregate([

{ $match: { "championships.endDate": null } },

{ $unwind: "$championships" },

{ $match: { "championships.endDate": null } },

{ $group: {

\_id: "$\_id",

"First name": { $first: "$firstName" },

"Last name": { $first: "$lastName" },

championship: { $push: {

weightClass: "$championships.weightClass",

startDate: "$championships.startDate",

endDate: "$championships.endDate"

}}

}},

{ $out: "CurrentChampions" }

]);

## ניקוי מסד הנתונים

דיינה וייט מחליט לעשות ניקיון במסד הנתונים של ה-UFC. הוא מורה למחוק את כל הרשומות של לוחמים ללא תיעוד מספק, במטרה לשמור על דיוק ורלוונטיות המידע בארגון.

// Step 1: Create a temporary collection for inactive fighters

db.createCollection("TempInactiveFighters");

// Step 2: Find and insert inactive fighters into the temporary collection

db.Fighters.aggregate([

{

$lookup: {

from: "Events",

let: { fighterId: "$\_id" },

pipeline: [

{ $unwind: "$fights" },

{

$match: {

$expr: {

$or: [

{ $eq: ["$fights.fighter1", "$$fighterId"] },

{ $eq: ["$fights.fighter2", "$$fighterId"] }

]

}

}

}

],

as: "matchingFights"

}

},

{ $match: { matchingFights: { $size: 0 } } },

{

$project: {

\_id: 1,

fullName: { $concat: ["$firstName", " ", "$lastName"] }

}

},

{ $out: "TempInactiveFighters" }

]);

// Step 3: Get the inactive fighters from the temporary collection

let inactiveFighters = db.TempInactiveFighters.find().toArray();

// Step 4: Remove inactive fighters using remove

let deleteResult = db.Fighters.remove({

\_id: { $in: inactiveFighters.map(f => f.\_id) }

});

// Step 5: Prepare summary

let summary = {

deletedFighters: deleteResult.nRemoved,

deletedFightersNames: inactiveFighters.map(f => f.fullName),

remainingFighters: db.Fighters.count()

};

// Step 6: Drop the temporary collection

db.TempInactiveFighters.drop();

printjson(summary);

## הוספת לוחם חדש

דיינה וייט, תמיד בחיפוש אחר כישרונות חדשים, מורה על הוספת לוחמים חדשים למסד הנתונים.

db.Fighters.insertOne({

\_id: 2000,

firstName: "Almog",

lastName: "Sisso",

nickname: "The Calculator",

dateOfBirth: "1994-03-15",

joinedUFC: 2024,

nationality: "Israel",

heightCm: 170,

reachCm: 175,

weightLbs: 160,

weightClass: "Welterweight",

record: {

wins: 15,

losses: 2,

draws: 0,

noContests: 0,

knockouts: 2,

submissions: 8

},

martialArts: ["Boxing", "Brazilian Jiu-Jitsu", "Wrestling"],

championships: []

});

db.Fighters.insertOne({

\_id: 2001,

firstName: "Almog",

lastName: "Bura",

nickname: "The Tribal Chief",

dateOfBirth: "1995-01-09",

joinedUFC: 2024,

nationality: "Israel",

heightCm: 173,

reachCm: 183,

weightLbs: 140,

weightClass: "Featherweight",

record: {

wins: 10,

losses: 1,

draws: 2,

noContests: 2,

knockouts: 2,

submissions: 4

},

martialArts: ["Boxing", "Brazilian Jiu-Jitsu"],

championships: []

});

## הוספת אירוע וקרב

הוספה של אירוע וקרב חדשים למסד הנתונים

// Insert the new event

db.Events.insertOne({

\_id: 505,

eventName: "UFC 505: Sisso vs. Makhachev",

eventDate: "2026-01-01",

eventLocation: "T-Mobile Arena",

country: "USA",

fights: [{

fightId: "505",

fighter1: 2000, // Almog Sisso's ID

fighter2: 1023, // Islam Makhachev's ID

weightClass: "Lightweight",

winner: 2000, // Almog Sisso's ID as the winner

winMethod: "Submission",

roundNumber: 3

}]});

## עדכון – אלוף חדש

דיינה וייט מעדכן את מסד הנתונים לאחר קרב אליפות מרגש. הוא מורה לעדכן את פרטי האלוף החדש ולהוסיפו לטבלת האלופים, מסמן תחילתו של עידן חדש בקטגוריה.

// Update Almog Sisso (winner)

db.Fighters.updateOne(

{ \_id: 2000 },

{

$push: {

championships: { weightClass: "Lightweight", startDate: "2026-01-01", endDate: null }

},

$inc: {

"record.wins": 1,

"record.submissions": 1

}

}

);

// Update Islam Makhachev (loser)

db.Fighters.updateOne(

{ \_id: 1023 },

{

$set: { "championships.$[championship].endDate": "2026-01-01" },

$inc: { "record.losses": 1 }

},

{

arrayFilters: [{ "championship.weightClass": "Lightweight", "championship.endDate": null }]

}

);

// Retrieve updated fighter data and print the result

let newChamp = db.Fighters.findOne({ \_id: 2000 });

let formerChamp = db.Fighters.findOne({ \_id: 1023 });

print(newChamp && formerChamp

? `New champion crowned! ${newChamp.firstName} ${newChamp.lastName} (${newChamp.record.wins}-${newChamp.record.losses}) defeated former champion ${formerChamp.firstName} ${formerChamp.lastName} (${formerChamp.record.wins}-${formerChamp.record.losses}) to become the new UFC Lightweight Champion.`

: "Error retrieving updated fighter data."

);

## עדכון מערכת השיפוט

דיינה וייט מחליט לשפר את השקיפות בשיפוט הקרבות. הוא מורה להוסיף שדה חדש לרשומות הקרבות, המציין את זהות השופט, מאפשר מעקב מדויק יותר אחר החלטות השיפוט בזירה. בנוסף הוא מבקש לצורך הדוגמא למלא מיידית ל 5 קרבות את השופטים שלהם.

db.Events.updateMany(

{},

{

$set: {

"fights.$[].referee": "TBA"

}})

db.Events.updateOne(

{ \_id: 295, "fights.fightId": "295A" },

{ $set: { "fights.$.referee": "Herb Dean" } }

)

db.Events.updateOne(

{ \_id: 294, "fights.fightId": "294A" },

{ $set: { "fights.$.referee": "Marc Goddard" } }

)

db.Events.updateOne(

{ \_id: 293, "fights.fightId": "293A" },

{ $set: { "fights.$.referee": "Dan Miragliotta" } }

)

db.Events.updateOne(

{ \_id: 292, "fights.fightId": "292A" },

{ $set: { "fights.$.referee": "Keith Peterson" } }

)

db.Events.updateOne(

{ \_id: 291, "fights.fightId": "291A" },

{ $set: { "fights.$.referee": "Jason Herzog" } }

)

## שינוי תוצאות ב-UFC 236: הפתעה בקרב האליפות

דיינה וייט מורה לעדכן את תוצאות הקרב המרכזי ב-UFC 236 בין Max "Blessed" Holloway " לבין  
Dustin "The Diamond" Poirier. לאחר בחינה מחדש, נקבע כי מקס הלוואי ניצח בהכרעת שופטים, שינוי דרמטי מהתוצאה המקורית. עדכון זה צפוי להשפיע משמעותית על דירוגי קטגוריית המשקל הקל ועל מסלול האליפות

// Step 1: Update the fight result

db.Events.updateOne(

{ \_id: 236, "fights.fightId": "236A" },

{

$set: {

"fights.$.winner": "1003" // Holloway's ID

}

}

)

// Step 2: Get the fighters' IDs and fight details

let fightInfo = db.Events.aggregate([

{ $match: { \_id: 236, "fights.fightId": "236A" } },

{ $project: {

fighter1: { $arrayElemAt: ["$fights.fighter1", 0] },

fighter2: { $arrayElemAt: ["$fights.fighter2", 0] },

winner: { $arrayElemAt: ["$fights.winner", 0] },

winMethod: { $arrayElemAt: ["$fights.winMethod", 0] }

}

}

]).next();

let winnerId = fightInfo.winner;

let loserId = fightInfo.fighter1 === winnerId ? fightInfo.fighter2 : fightInfo.fighter1;

// Step 3: Update the winner's record (Holloway)

db.Fighters.updateOne(

{ \_id: parseInt(winnerId) },

{

$inc: {

"record.wins": 1,

"record.decisions": 1

}

}

)

// Step 4: Update the loser's record (Poirier)

db.Fighters.updateOne(

{ \_id: parseInt(loserId) },

{

$inc: {

"record.losses": 1

}

}

)

console.log(`Updated fight result and fighters' records. Winner: ${winnerId} (Holloway), Loser: ${loserId} (Poirier)`);

## המשקל כן קובע : ניתוח מחלקות המשקל ב UFC

דיינה וייט מנתח את היעילות של דפוסי הניצחון בקרב הלוחמים על פי קטגוריות משקל שונות. הניתוח חושף אילו לוחמים מובילים בקטגוריות השונות.

db.Events.aggregate([

{ $unwind: "$fights" },

// Group by weight class and winner to count wins

{ $group: { \_id: { class: "$fights.weightClass", fighter: "$fights.winner" }, wins: { $sum: 1 } } },

// Sort by weight class and then by number of wins descending

{ $sort: { "\_id.class": 1, "wins": -1 } },

// Group by weight class and collect the top fighters

{ $group: { \_id: "$\_id.class", fighters: { $push: { id: "$\_id.fighter", wins: "$wins" } } } },

// Limit to the top 3 fighters per weight class

{ $project: { \_id: 0, class: "$\_id", topFighters: { $slice: ["$fighters", 3] } } },

{ $sort: { class: 1 } },

// Lookup the fighters' names from the Fighters collection

{ $lookup: {

from: "Fighters",

localField: "topFighters.id",

foreignField: "\_id",

as: "fighterDetails"

}},

// Map the fighters array to include the name alongside ID and wins

{ $addFields: {

topFighters: {

$map: {

input: "$topFighters",

as: "fighter",

in: {

id: "$$fighter.id",

wins: "$$fighter.wins",

name: {

$let: {

vars: {

details: { $arrayElemAt: [{ $filter: { input: "$fighterDetails", as: "f", cond: { $eq: ["$$f.\_id", "$$fighter.id"] } } }, 0] }

},

in: { $concat: ["$$details.firstName", " ", "$$details.lastName"] }

}

}

}

}

}

}},

{ $project: { fighterDetails: 0 } }

]).forEach(weightClass => {

print(`\nWeight Class: ${weightClass.class}`);

weightClass.topFighters.forEach((fighter, index) => {

print(` ${index + 1}. Fighter ID: ${fighter.id}, Name: ${fighter.name}, Wins: ${fighter.wins}`);

});

});

## המיקס המנצח: ניתוח שילובי אומנויות לחימה ב UFC

דיינה וייט מנתח את יעילותם של שילובי אומנויות לחימה שונים בקרב לוחמי ה-UFC. הניתוח המעמיק חושף אילו שילובים מובילים לאחוזי ניצחון, נוקאאוט וכניעות גבוהים יותר, מספק תובנות חדשות על אסטרטגיות אימון ותכנון קרבות.

db.Fighters.aggregate([

{$unwind: "$martialArts"},

{$project: {

name: {$concat: ["$firstName", " ", "$lastName"]},

martialArt: "$martialArts",

totalFights: {$add: ["$record.wins", "$record.losses", "$record.draws", "$record.noContests"]},

wins: "$record.wins", losses: "$record.losses",

knockouts: "$record.knockouts", submissions: "$record.submissions",

winPercentage: {$multiply: [{$divide: ["$record.wins", {$add: ["$record.wins", "$record.losses", "$record.draws", "$record.noContests"]}]}, 100]}

}},

{$sort: {winPercentage: -1, wins: -1}},

{$group: {

\_id: "$martialArt",

topFighters: {$push: {

name: "$name", totalFights: "$totalFights", wins: "$wins", losses: "$losses",

knockouts: "$knockouts", submissions: "$submissions",

winPercentage: {$round: ["$winPercentage", 2]}

}}

}},

{$project: {\_id: 0, martialArt: "$\_id", topFighters: {$slice: ["$topFighters", 3]}}},

{$sort: {martialArt: 1}},

{$out: "TopFightersByMartialArt"}

]);

print("Top 3 Fighters for Each Martial Art:");

print("--------------------------------");

db.TopFightersByMartialArt.find().forEach(function(art) {

print("Martial Art: " + art.martialArt);

art.topFighters.forEach(function(fighter, index) {

print((index + 1) + ". " + fighter.name);

print(" Total Fights: " + fighter.totalFights);

print(" Wins: " + fighter.wins + " | Losses: " + fighter.losses);

print(" Knockouts: " + fighter.knockouts + " | Submissions: " + fighter.submissions);

print(" Win Percentage: " + fighter.winPercentage + "%");

});

print("--------------------------------");

});

## מעצמת ה-MMA: זיהוי המדינה המובילה בייצוא לוחמי UFC

דיינה וייט מבקש לזהות את המדינה המובילה בייצוא לוחמי UFC. באמצעות ניתוח MapReduce מתקדם, הצוות מונה את מספר הלוחמים מכל מדינה ומזהה את המובילה.

db.Fighters.aggregate([

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

{ $sort: { count: -1 } },

{ $limit: 5 },

{ $project: {

\_id: 0,

country: "$\_id",

fighters: "$count",

output: { $concat: [

{ $toString: "$\_id" },

": ",

{ $toString: "$count" },

" fighters"

]}

}

}

]).forEach(function(doc) {

print(doc.output);

});

## מלך הכסף: חשיפת הלוחם המרוויח ביותר ב-UFC

דיינה וייט מעוניין לזהות את הלוחם בעל ההכנסות הגבוהות ביותר ב-UFC. באמצעות ניתוח MapReduce מורכב, הצוות מחשב את הרווחים של כל לוחם, כולל בונוסים על ניצחונות וקנסות על תיקו. התוצאות עשויות להשפיע על משא ומתן עתידי ועל אסטרטגיות תגמול בארגון.

// MapReduce operation to calculate fighter earnings and fight statistics

var mapFunction = function() {

this.fights.forEach(function(fight) {

var winBonus = 50000;

var result = { earnings: 0, wins: 0, losses: 0, draws: 0 };

if (fight.winner) {

result.earnings = winBonus;

result.wins = 1;

emit(fight.winner, result);

var loserId = fight.winner === fight.fighter1 ? fight.fighter2 : fight.fighter1;

emit(loserId, { earnings: 0, wins: 0, losses: 1, draws: 0 });

} else {

result.earnings = -5000; // Draw penalty

result.draws = 1;

emit(fight.fighter1, result);

emit(fight.fighter2, result);

}

});

};

var reduceFunction = function(key, values) {

return values.reduce(function(total, current) {

return {

earnings: total.earnings + current.earnings,

wins: total.wins + current.wins,

losses: total.losses + current.losses,

draws: total.draws + current.draws

};

}, { earnings: 0, wins: 0, losses: 0, draws: 0 });

};

db.Events.mapReduce(

mapFunction,

reduceFunction,

{ out: "fighterEarnings" }

);

// Find the top earner and add their name

var topEarner = db.fighterEarnings.find().sort({value: -1}).limit(1).toArray()[0];

var fighterDetails = db.Fighters.findOne({\_id: NumberInt(topEarner.\_id)});

var result = {

fighterId: topEarner.\_id,

name: fighterDetails.firstName + " " + fighterDetails.lastName,

earnings: topEarner.value.earnings,

wins: topEarner.value.wins,

losses: topEarner.value.losses,

draws: topEarner.value.draws,

totalFights: topEarner.value.wins + topEarner.value.losses + topEarner.value.draws

};

print("Top earning fighter:");

printjson(result);

## סמבו באמריקה: מהפכת האימונים ב-UFC

לאור ההצלחה המרשימה של לוחמים מחבל הקווקז, המשלבים את אמנות הלחימה סמבו בסגנונם, החלו לוחמים אמריקאים לאמץ את האמנות הזו.

// Count fighters from USA with Sambo before the update

let countBefore = db.Fighters.count({

nationality: "USA",

martialArts: "Sambo"

});

print("Number of USA fighters with Sambo before update: " + countBefore);

// Add Sambo to all USA fighters

let updateResult = db.Fighters.updateMany(

{ nationality: "USA" },

{ $addToSet: { martialArts: "Sambo" } }

);

// Count fighters from USA with Sambo after the update

let countAfter = db.Fighters.count({

nationality: "USA",

martialArts: "Sambo"

});

print("Number of USA fighters with Sambo after update: " + countAfter);

// Display how many documents were actually modified

print("Number of fighters updated: " + updateResult.modifiedCount);

## עלייה לגדולה: ג'ון ג'ונס כובש את תואר האלוף במשקל כבד

דיינה וייט מתמודד עם תפנית בקריירה של ג'ון ג'ונס. למרות שיריבו היה כבד יותר, ג'ונס ניצח וזכה בתואר אלוף משקל כבד. וייט מורה להסיר את רישום אליפות ה-Light Heavyweight של ג'ונס ולהוסיף רישום חדש לאליפותו במשקל כבד, עם אותם תאריכים.

// Display Jon Jones' championships before the update

let fighterBefore = db.Fighters.findOne({ \_id: 1000 });

print("Jon Jones' championships before update:");

printjson(fighterBefore.championships);

// First, remove the Light Heavyweight championship

db.Fighters.updateOne(

{ \_id: 1000 },

{

$pull: {

championships: { weightClass: "Light Heavyweight" }

}

}

);

// Then, add the new Heavyweight championship

db.Fighters.updateOne(

{ \_id: 1000 },

{

$push: {

championships: {

weightClass: "Heavyweight",

startDate: "2018-12-29",

endDate: "2023-01-14"

}

}

}

);

// Display Jon Jones' championships after the update

let fighterAfter = db.Fighters.findOne({ \_id: 1000 });

print("\nJon Jones' championships after update:");

printjson(fighterAfter.championships);

# עבודה עם Claude AI

ראשית, חשוב להדגיש כי התהליך החל עם משימה ב SQL, ורק לאחר מכן התקדמנו למשימה ב-MongoDB . כתוצאה מכך, בסיס הנתונים שלנו כבר היה מוכן מראש. המשימה העיקרית שלנו הייתה להמיר את הנתונים לפורמט המתאים ל-MongoDB ולפתח את מודל הנתונים המתאים.

לצורך המרת הנתונים והתאמתם ל MongoDB נעזרתי בכלי הבינה המלאכותית .Claude AI סיפקנו לו את קבצי ה-DDL וה-INSERT מה-SQL המקורי, וביקשנו ממנו ליצור בסיס נתונים מותאם ל- .MongoDB את המבנה הכללי של בסיס הנתונים אנחנו ידענו מראש מה אנחנו רוצים ולכן התוצאה היא המבנה שראיתם בתחילת המסמך.

ככה נראה ה prompt :

Please convert the following SQL DDL and INSERT statements to MongoDB schema and operations:

Specific requirements:

- Create two main collections: "Fighters" and "Events"

- Embed the fights within the Events collection

- Pay attention to the keys between fighters and fights

- Any additional table should be inserted as an array where appropriate

לאחר שקיבלנו את המבנה שראינו למעלה , ביקשנו מהצאט לבצע את ההקמה עצמה של בסיס הנתונים כלומר לבצע הכנסה של נתונים.

Perform data insertion into the database based on the following data modeling:

[Insert your data modeling description here]

Requirements:

1. Insert data into the "Fighters" and "Events" collections

2. Ensure proper embedding of fights within the Events collection

3. Maintain correct foreign key relationships, especially between fighters and fights

4. Use appropriate data types for each field

5. Handle any additional data (from extra tables) as arrays where suitable

Please provide the MongoDB insertion operations, paying close attention to:

- Correct use of data types

- Proper referencing of foreign keys

- Accurate embedding of subdocuments

- Consistent formatting of dates, numbers, and other special data types

If any data transformation or preprocessing is needed, please explain the steps taken.

לאחר מכן כבר היה לנו בסיס נתונים מוכן, ומה שהיה צריך לעשות זה רק לבקש מהצאט לכתוב שאילתות שאנחנו חשבנו על הרעיונות שלה, אם היה חשוב להשתמש באופרטור ספציפי היינו מדגישים זאת בדומה לעבודה ב SQL.