USE CASE STUDY REPORT

Group No.: Group 01

Student Names: Rishabh Madani and Omkar Narkar

Executive Summary

The Entertainment Agency Management System (EAMS) addresses critical challenges within the expansive yet fragmented landscape of the Entertainment Industry, valued at 2.82 billion USD. Currently lacking centralized control, this industry struggles to maximize its potential. EAMS emerges as a comprehensive solution, aiming to organize and streamline various facets of entertainment, fostering growth and efficiency. By leveraging extensive data analysis capabilities, EAMS empowers stakeholders to make informed decisions regarding talent representation, project investments, and overall agency performance. This platform not only enhances operational effectiveness but also enables the identification of profitable opportunities amidst the vast production of movies, albums, and concerts. With a focus on data-driven decision-making, talent management, and risk assessment, EAMS aims to become the cornerstone of the entertainment industry, facilitating its evolution into a more structured and lucrative sector. Through strategic integration and analytics, EAMS promises to catalyze growth, stability, and success within the dynamic realm of Indian entertainment.

The implemented schema consists of following key entities:

1. Talent (ArtistID, First Name, Last Name, DOB, Age, Address, Email, Phone)
2. Singer (ArtistID, Specialty)
3. Actor (ArtistID, Specialty)
4. Company (CompanyID, Name, Email)
5. Agency (AgencyID, Name, Address, Email, Phone)
6. Stats (StatsID, Revenue, NetProfit, Investment)
7. Concert (ConcertID, Venue, Capacity, Date, Time)
8. Album (AlbumID, Name, Year, Duration, Genre, *ArtistID*, *StatID*)
9. Movie (MovieID, Title, Duration, Genre, Rating, Year, *StatID*)
10. Award (AwardID, Title, Organization)
11. Production (Identifier)

Achievements of this study include the successful creation of a relational database that effectively models the Entertainment Agency Management System (EAMS)'s requirements. The structured design facilitates data integrity and relationships between different entities. Recommendations for future enhancements may include optimizing query performance, ensuring data security, and implementing regular updates to adapt to evolving business needs.

Conceptual Data Modelling

Conceptual Data Modeling plays a pivotal role in designing a robust and effective database system. This phase of data modeling involves the creation of an Entity-Relationship Diagram (ERD) using Enhanced Entity-Relationship (EER) modeling techniques and Unified Modeling Language (UML). This report outlines the key components of conceptual data modeling, including domains, cardinalities, constraints, and abstractions.

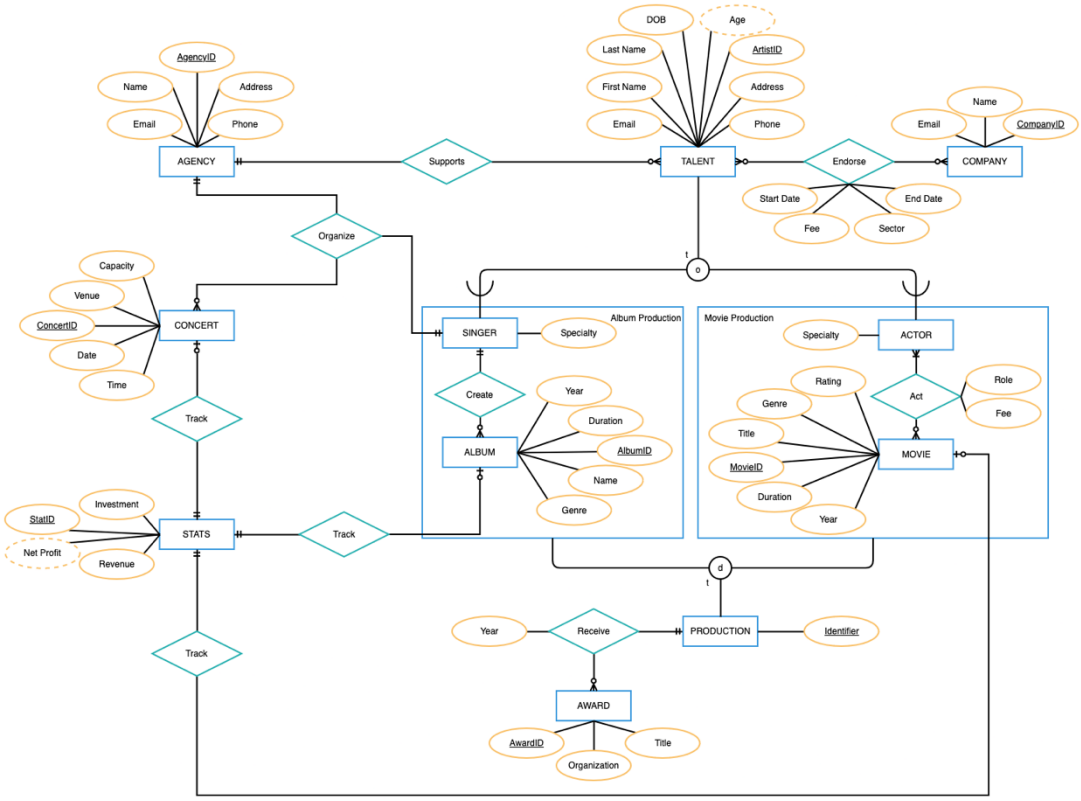


Figure 1 : EER Diagram

Assumptions and Constraints:

1. An entity can contain only 1 email id.
2. An entity can contain only 1 phone number.
3. Every album has only 1 singer.

4. Only one singer may perform for a concert.
5. A concert can be organized by only one agency.
6. Award will be given to artists and albums but not movies.
7. A talent is associated with only one agency throughout lifetime.

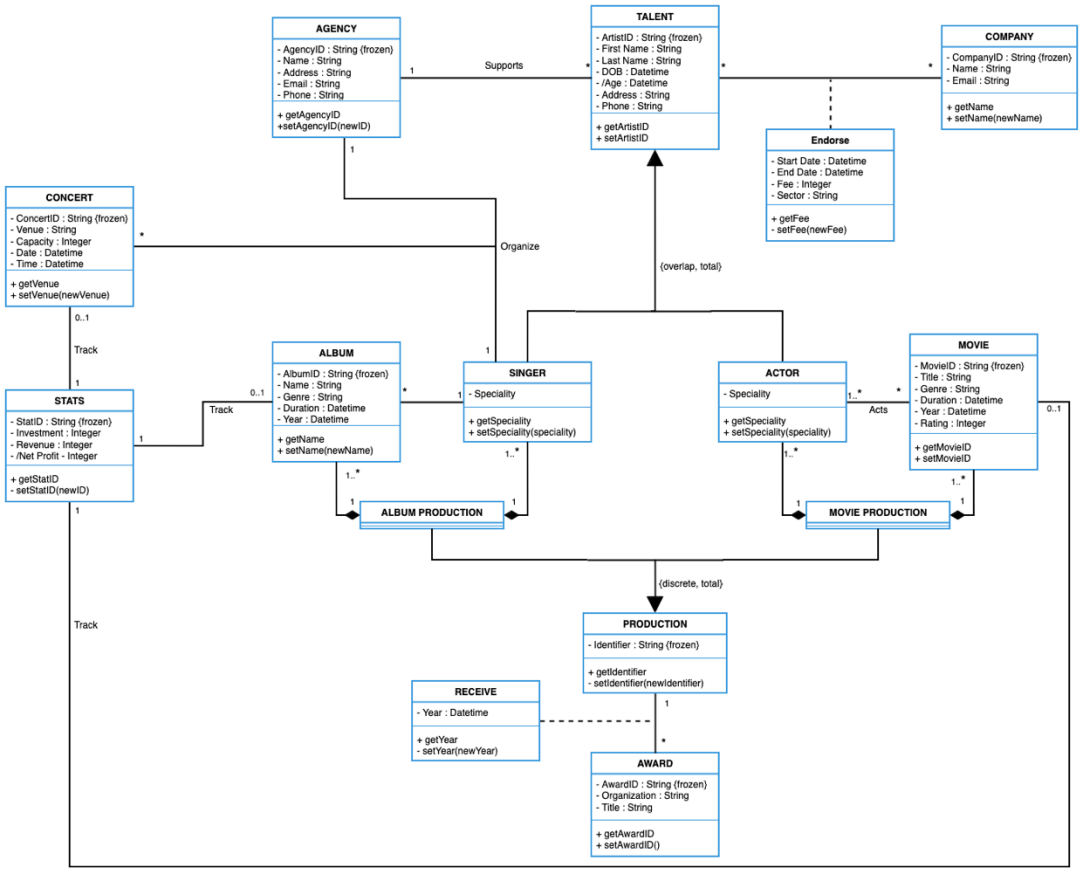


Figure 2 : UML Diagram

Mapping Conceptual Model to Relational Model

Mapping from an EER and UML model to a relational model is a meticulous process that involves translating conceptual representations into a structured database schema. The resulting relational model forms the basis for creating an efficient and well-organized database system capable of handling the complexities of the Fixed Deposits Booking Management System.

Relational Model:

1. Talent (**ArtistID**, First Name, Last Name, DOB, Age, Address, Email, Phone, *AgencyID*)
   - AgencyID refers to AgencyID in Agency.
2. Singer (***ArtistID***, Specialty)

   - ArtistID refers to ArtistID in Talent.
3. Actor (***ArtistID***, Specialty)
   - ArtistID refers to ArtistID in Talent.
4. Company (**CompanyID**, Name, Email)
5. Endorse (***ArtistID***, ***CompanyID***, Fee, StartDate, EndDate, Sector)
   - ArtistID refers to ArtistID in Talent.
   - CompanyID refers to CompanyID in Company.
6. Agency (**AgencyID**, Name, Address, Email, Phone)
7. Stats (**StatsID**, Revenue, NetProfit, Investment)
8. Concert (**ConcertID**, Venue, Capacity, Date, Time, *StatsID*)
   - StatsID refers to StatsID in Stats.
9. OrganizeConcert (***ConcertID***, ***AgencyID***, ***ArtistID***)
   - ConcertID refers to ConcertID in Concert
   - AgencyID refers to AgencyID in Agency.
   - ArtistID refers to ArtistID in Talent.
10. Production (**Identifier**)
11. Album (**AlbumID**, Name, Year, Duration, Genre, *ArtistID*, *StatID*)
   - ArtistID refers to ArtistID in Singer.
   - StatID refers to StatID in Stats.
12. AlbumProduction (***Identifier***, ***AlbumID***, ***ArtistID***)
   - Identifier refers to Identifier in Production.
   - AlbumID refers to AlbumID in Album.
   - ArtistID refers to ArtistID in Singer/Actor.
13. Movie (**MovieID**, Title, Duration, Genre, Rating, Year, *StatID*)
   - StatID refers to StatID in Stats.
14. Act (***ArtistID***, ***MovieID***, Role, Fee)
   - ArtistID refers to ArtistID in Actor.
   - MovieID refers to MovieID in Movie.
15. MovieProduction (***Identifier***, ***MovieID***, ***ArtistID***)
   - Identifier refers to Identifier in Production.
   - MovieID refers to MovieID in Movie.
   - ArtistID refers to ArtistID in Actor.
16. Award (**AwardID**, Title, Organization)
17. Recipient (***Identifier***, ***AwardID***)
   - Identifier refers to Identifier in Production.
   - AwardID refers to AwardID in Award.

Implementation of Relation Model via MySQL

The implementation of the relational model in MySQL involves translating the designed schema into a functioning database, creating tables, defining relationships, and populating the database with sample data. This process is integral to organize The Entertainment Agency Management System (EAMS) to exponentially grow the entertainment industry.

1. List the concerts organized by agencies where the net profit exceeded a certain threshold.

   It allows agencies to understand which events were particularly profitable and potentially replicate their success in future planning. Investors would be interested in this query as it helps them identify successful concerts organized by agencies, indicating potentially lucrative investment opportunities.

ConcertID	Venue	Date	AgencyID	NetProfit
4	Staples Center	2022-08-16	3	290612022.60
25	Barclays Center	2019-08-23	10	278355330.08
13	Staples Center	2021-04-16	10	271220278.04
21	Barclays Center	2021-11-27	7	265888108.39
2	Barclays Center	2020-11-24	10	262210919.19
1	Staples Center	2019-04-28	7	255182486.88

   ```
   SELECT c.ConcertID, c.Venue, c.Date, oc.AgencyID, s.NetProfit
   FROM Concert c
   JOIN OrganizeConcert oc ON c.ConcertID = oc.ConcertID
   JOIN Stats s ON c.StatsID = s.StatsID
   WHERE s.NetProfit > 250000000
   ORDER BY s.NetProfit DESC;
   ```

2. Determine the total revenue generated from concerts organized by each agency.

   Investors can use this query to evaluate the performance of different agencies in terms of revenue generation, helping them make informed decisions about potential investments in agency stocks or partnerships. It also helps in This helps in evaluating the effectiveness and profitability of each agency's concert management operations.

AgencyID	Name	TotalRevenue
10	Jast-Strosin	1200416097.32
7	Wisoky LLC	930228843.73
8	Gaylord-Block	707563192.05
2	Von-Walker	506917824.77
9	Russel PLC	473631150.34
4	Jones-Rohan	472254116.75
5	Buckridge and Sons	424326966.11
3	Hegmann-Klein	394128909.03
6	Muller-Roberts	382024970.64
1	Reichel, Flatley and Marvin	378415614.17

   ```
   SELECT a.AgencyID, a.Name, SUM(s.Revenue) AS TotalRevenue
   FROM Concert c
   JOIN OrganizeConcert oc ON c.ConcertID = oc.ConcertID
   JOIN Agency a ON oc.AgencyID = a.AgencyID
   JOIN Stats s ON c.StatsID = s.StatsID
   GROUP BY a.AgencyID
   ORDER BY TotalRevenue DESC;
   ```

3. List top 5 agency that has the highest average revenue per concert in descending order.

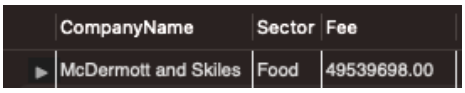
   The query helps in showcasing efficiency and effectiveness in organizing profitable events. This information can aid agencies in benchmarking and improving performance. The query also allows investors to identify agencies with a consistently high revenue per concert, indicating efficiency and potential profitability for investors who are considering partnerships or investments in those agencies.

AgencyID	Name	AvgRevenuePerConcert
4	Jones-Rohan	236127058.375000
7	Wisoky LLC	232557210.932500
5	Buckridge and Sons	212163483.055000
10	Jast-Strosin	200069349.553333
3	Hegmann-Klein	197064454.515000

   ```
   SELECT a.AgencyID, a.Name, AVG(s.Revenue) AS AvgRevenuePerConcert
   FROM Concert c
   JOIN OrganizeConcert oc ON c.ConcertID = oc.ConcertID
   ```

```
JOIN Agency a ON oc.AgencyID = a.AgencyID
JOIN Stats s ON c.StatsID = s.StatsID
GROUP BY a.AgencyID, a.Name
ORDER BY AvgRevenuePerConcert DESC
LIMIT 5;
```

4. List the companies that have endorsed artists with the highest fees.

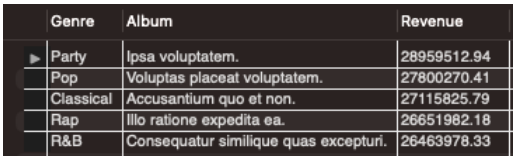


CompanyName	Sector	Fee
McDermott and Skiles	Food	49539698.00

The query helps in highlighting companies investing significant amounts in artist endorsements, indicating strong brand associations or marketing strategies. This insight can guide partnership decisions and negotiation strategies. Talents can identify high-paying endorsement opportunities, allowing them to negotiate better contracts and maximize their earnings.

```
SELECT
   c.Name AS CompanyName, e.Sector, e.Fee
FROM Endorse e
JOIN Company c ON e.CompanyID = c.CompanyID
WHERE
   e.Fee = (SELECT
         MAX(Fee)
      FROM
         Endorse);
```

5. Find the top-selling album in terms of revenue for each genre.



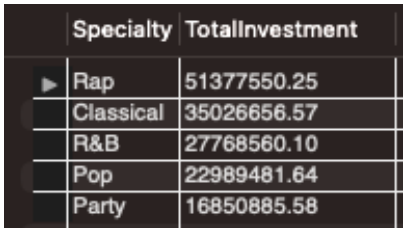
Genre	Album	Revenue
Party	Ipsa voluptatem.	28959512.94
Pop	Voluptas placeat voluptatem.	27800270.41
Classical	Accusantium quo et non.	27115825.79
Rap	Illo ratione expedita ea.	26651982.18
R&B	Consequatur similique quas excepturi.	26463978.33

Agencies can use this query to understand the market demand for different genres of music and tailor their artist representation and marketing strategies accordingly.

```
WITH RankedAlbum AS (
    SELECT
            a.Genre, a.Name AS 'Album', s.Revenue,
            RANK() OVER (PARTITION BY a.Genre ORDER BY s.Revenue DESC)
as AlbumRank
    FROM
            Album a,
            Stats s
    WHERE
            a.StatID = s.StatsID
    ORDER BY AlbumRank, s.Revenue DESC
)
SELECT Genre, Album, Revenue
FROM RankedAlbum
WHERE AlbumRank = 1;
```

6. Find the total investment made in concerts for each genre of music.

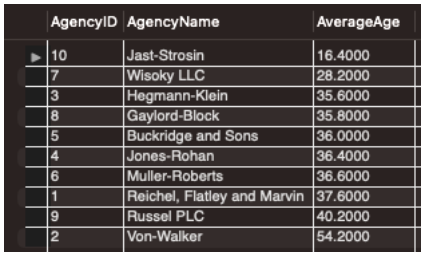
The query helps in understand investment trends across different music genres, guiding resource allocation and strategic planning for concert events. It helps in identifying lucrative genres and potential areas for growth.



Specialty	TotalInvestment
Rap	51377550.25
Classical	35026656.57
R&B	27768560.10
Pop	22989481.64
Party	16850885.58

```
SELECT sg.Specialty, SUM(s.Investment) AS TotalInvestment
FROM Concert c
JOIN OrganizeConcert oc ON c.ConcertID = oc.ConcertID
JOIN Singer sg ON oc.ArtistID = sg.ArtistID
JOIN Stats s ON c.StatsID = s.StatsID
GROUP BY sg.Specialty
ORDER BY TotalInvestment DESC;
```

7. Find the average age of artists represented by each agency.
   This query allows agencies to understand the demographics of their talent pool, which can inform their recruitment strategies and help them better cater to the needs of their artists. It assists artists in making informed decisions by identifying which agency is more inclined to provide support based on demographic alignment.



AgencyID	AgencyName	AverageAge
10	Jast-Strosin	16.4000
7	Wisoky LLC	28.2000
3	Hegmann-Klein	35.6000
8	Gaylord-Block	35.8000
5	Buckridge and Sons	36.0000
4	Jones-Rohan	36.4000
6	Muller-Roberts	36.6000
1	Reichel, Flatley and Marvin	37.6000
9	Russel PLC	40.2000
2	Von-Walker	54.2000

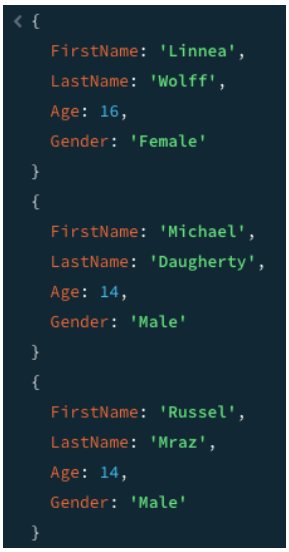
```
SELECT a.AgencyID, a.Name AS AgencyName, AVG(t.Age) AS AverageAge
FROM Talent t
JOIN Agency a ON t.AgencyID = a.AgencyID
GROUP BY a.AgencyID, AgencyName
ORDER BY AverageAge;
```

Implementation of Model via NoSQL

The implementation of the relational model in NoSQL involves creating collections and populating with documents. This process is integral to organize The Entertainment Agency Management System (EAMS) to exponentially grow the entertainment industry.

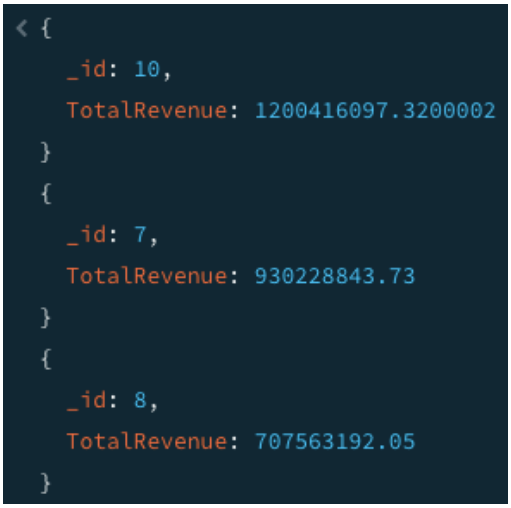
1. Retrieve all records from the Talent collection where the age is less than equal to 16.
   Examining the age distribution of talents aged 16 or younger can help identify trends and patterns in the emergence of talent across different age brackets. This insight can inform talent development strategies, educational programs, or scouting efforts targeting specific age groups.

```
db.Talent.find(     { Age: { $lte: 16 } },
                    { FirstName: 1, LastName: 1, _id: 0 , Age: 1, Gender: 1}
            )
```

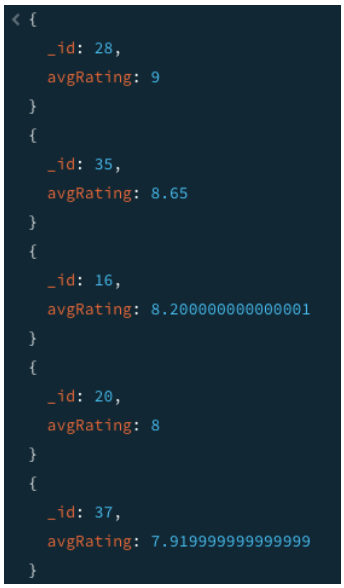


2. Analyze the revenue data by agency.
Analyzing revenue data by agency provides insights into the effectiveness of agency partnerships in generating concert revenue. Organizations can use this information to optimize their partnerships by prioritizing collaborations with high-performing agencies or adjusting contractual terms to incentivize revenue growth.

```
db.OrganizeConcert.aggregate([
   {$lookup: {
                  from: "Concert",
        localField: "ConcertID",
        foreignField: "ConcertID",
        as: "concert"
     }
   },
   {$unwind: "$concert"},
   {$lookup: {
        from: "Stats",
        localField: "concert.StatsID",
        foreignField: "StatsID",
        as: "stats"
     }
   },
   {$unwind: "$stats"},
   {$group: {
        _id: "$AgencyID",
        TotalRevenue: { $sum: "$stats.Revenue" }
     }
   },
   {$sort: { TotalRevenue: -1 }},
   {$limit:3 }
])
```



3. Analyze actors' average ratings in movies
   The query analyzes actors' average ratings in movies, aiding in assessing their performance and popularity. Insights guide casting decisions, marketing efforts, and strategic planning in the film industry. By understanding audience preferences and competitive dynamics, businesses optimize resources and tailor content for success.




```
db.Act.aggregate([
  { $lookup: {
     from: "Movie",
     localField: "MovieID",
     foreignField: "MovieID",
     as: "movie"
    }
  },
  { $unwind: "$movie" },
  { $group: {
     _id: "$ArtistID",
     avgRating: { $avg: "$movie.Rating" }
    }
  },
  { $sort: { avgRating: -1 } },
  { $limit: 10 }
])
```

4. Analyze how movies performed during the COVID-19 Pandemic.
   The query provides insights into the financial performance of the movie industry during the COVID-19 pandemic. By examining movies' net profits, stakeholders can assess the industry's resilience, identify successful releases, and inform strategic decision-making for future projects and investments.

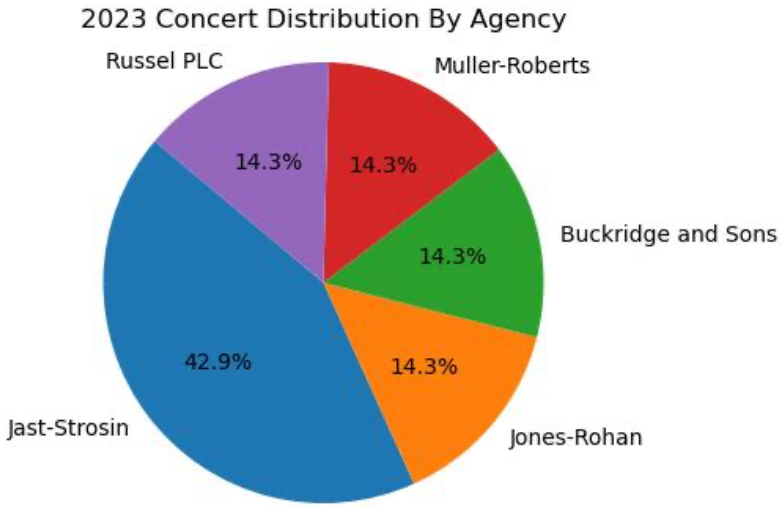

```
db.Movie.aggregate([
  {
   $lookup: {
     from: "Stats",
     localField: "StatID",
     foreignField: "StatsID",
     as: "stats"
    }
  },
  {
   $match: {
     Year: { $gte: 2021, $lte: 2022 }
    }
```

```
  },
  { $unwind: "$stats" },
  {
   $project: {
    Title: 1,
    NetProfit: "$stats.NetProfit"
   }
  }
 ])
```

Database Access via Python

1. In the framework of the Entertainment Agency Management System (EAMS), this code generates a pie chart representing the distribution of concerts among agencies in 2023. It visually showcases each agency's share, aiding in understanding their involvement in concert production.

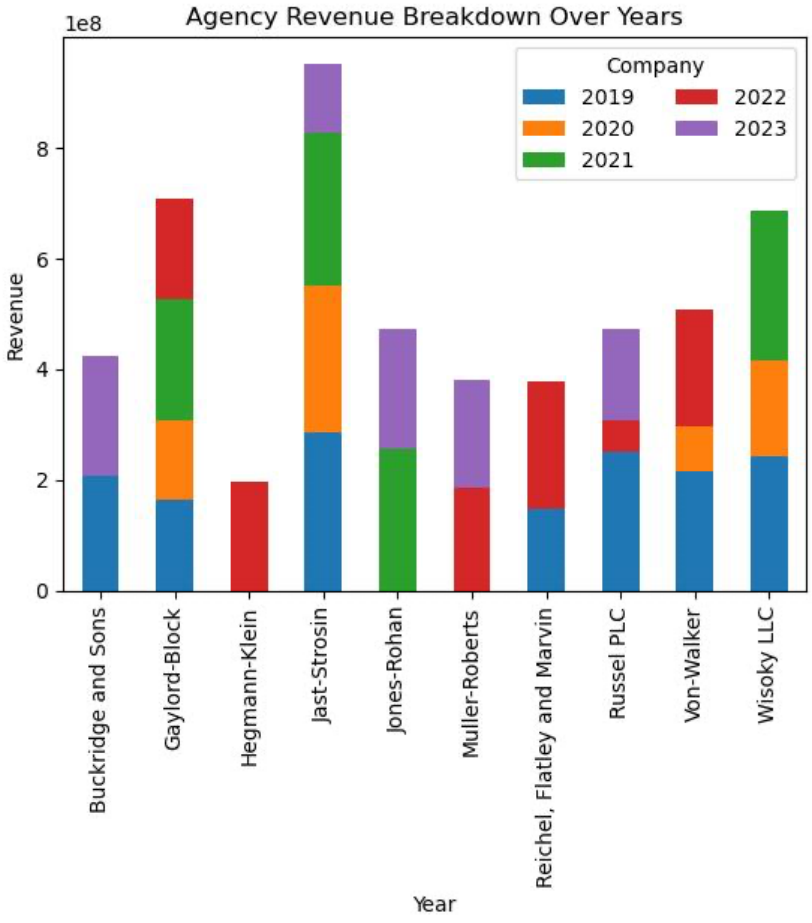
```
plt.figure(figsize=(6, 4))
plt.pie(df3['ConcertCount'], labels=df3['Name'], autopct='%1.1f%%', startangle=140)
plt.title('2023 Concert Distribution By Agency')
plt.axis('equal')
plt.show()
```



2. The code below produces a stacked bar plot presenting revenue distribution across years for multiple agencies. It visually communicates how each agency contributes to overall revenue trends over time, facilitating comparative analysis of their financial performance and aiding in identifying patterns or anomalies within the data.

```
plt.figure(figsize=(4, 2)) plot1.plot(kind='bar', stacked=True)
plt.title('Agency Revenue Breakdown Over Years') plt.xlabel('Year')
```

```
plt.ylabel('Revenue')
plt.legend(title='Company', loc='upper right', ncol=2) plt.xticks(rotation=90)
plt.show()
```



Summary and Recommendation

The development of The Entertainment Agency Management System (EAMS) centers on a robust relational database model. This model encompasses essential entities such as 'Agency,' 'Company,' 'Actor,' 'Stats,' 'Concert,' 'Production,' 'Award,' 'Act,' and 'Movie,' offering a comprehensive framework for effectively managing EAMS data. Leveraging MySQL for database implementation ensures smooth storage, retrieval, and analysis of information, while the system's integration with Python highlights its versatility.

Advantages:

1. Enhanced Decision-Making: The Entertainment Agency Management System (EAMS) empowers stakeholders with valuable insights derived from extensive data analysis, enabling informed decision-making regarding talent representation, project investments, and overall agency performance.
2. Improved Efficiency: By centralizing and organizing various elements of the entertainment industry, EAMS streamlines operations and workflows, reducing redundancy and enhancing overall efficiency in managing talent, projects, and finances.
3. Effective Risk Management: EAMS facilitates risk assessment through comprehensive data analysis, allowing agencies to identify and mitigate potential risks associated with talent representation, project investments, and market dynamics, thereby minimizing financial losses.
4. Competitive Advantage: With its robust features and data-driven approach, EAMS equips entertainment agencies with a competitive edge in navigating the dynamic and competitive landscape of the industry, fostering growth, stability, and success in an ever-evolving market.

Shortcomings:

Complexity in Querying: Users with limited SQL proficiency may find complex querying challenging, highlighting the need for user-friendly interfaces.

Recommendations:

1. Scalability Planning: Regularly assess and plan for scalability, ensuring the system can handle increased data volume and user interactions.
2. Security Measures: Implement robust security measures to protect sensitive financial data, including encryption, access controls, and regular security audits.
3. Continuous Optimization: Regularly optimize the database structure and queries to ensure optimal performance as the system evolves.

In conclusion, The Entertainment Agency Management System (EAMS) demonstrates the potential for efficient data management for an entertainment industry. While leveraging the advantages of a relational database and versatile language integration, addressing the identified shortcomings through strategic recommendations will ensure the system's continued success and adaptability in the ever-evolving financial landscape.