CSCI 620: Introduction to Big Data

12/9/2023

FIFA Unleashed: Eight Seasons of FIFA Player Trends (2015 - 23)

Group 7
Niharika Ahirekar
Rutvi Bheda
Mohammed Raeesul Irfan

TABLE OF CONTENTS

01 Phase I → About our dataset, how we formed and loaded it.

02 Phase II → Interesting queries over our dataset, Indexing our dataset for faster query execution, Functional Dependencies, Normalization.

03 Phase III → Document-oriented model Vs Relational model, Operations performed on the dataset:
Data Cleaning, Data Integration, Itemset Mining

FIFA Unleashed: Eight Seasons of Player Trends

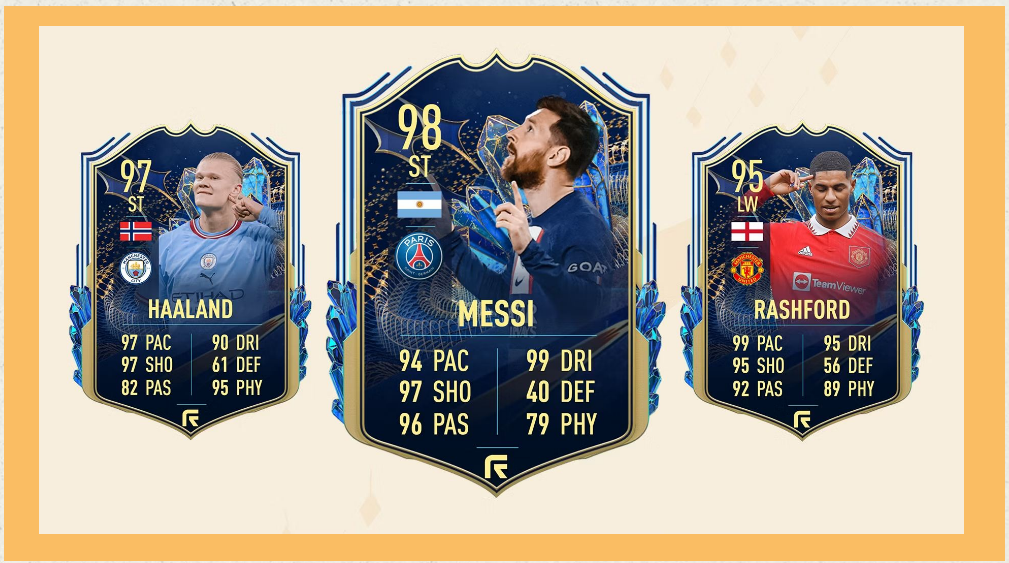
❏ Based on the videogame FIFA
❏ Allows comparisons for the same players across the last 8 version of the videogame (FIFA'15 - FIFA'23)
❏ Basically the data on the player cards

Player Stats:

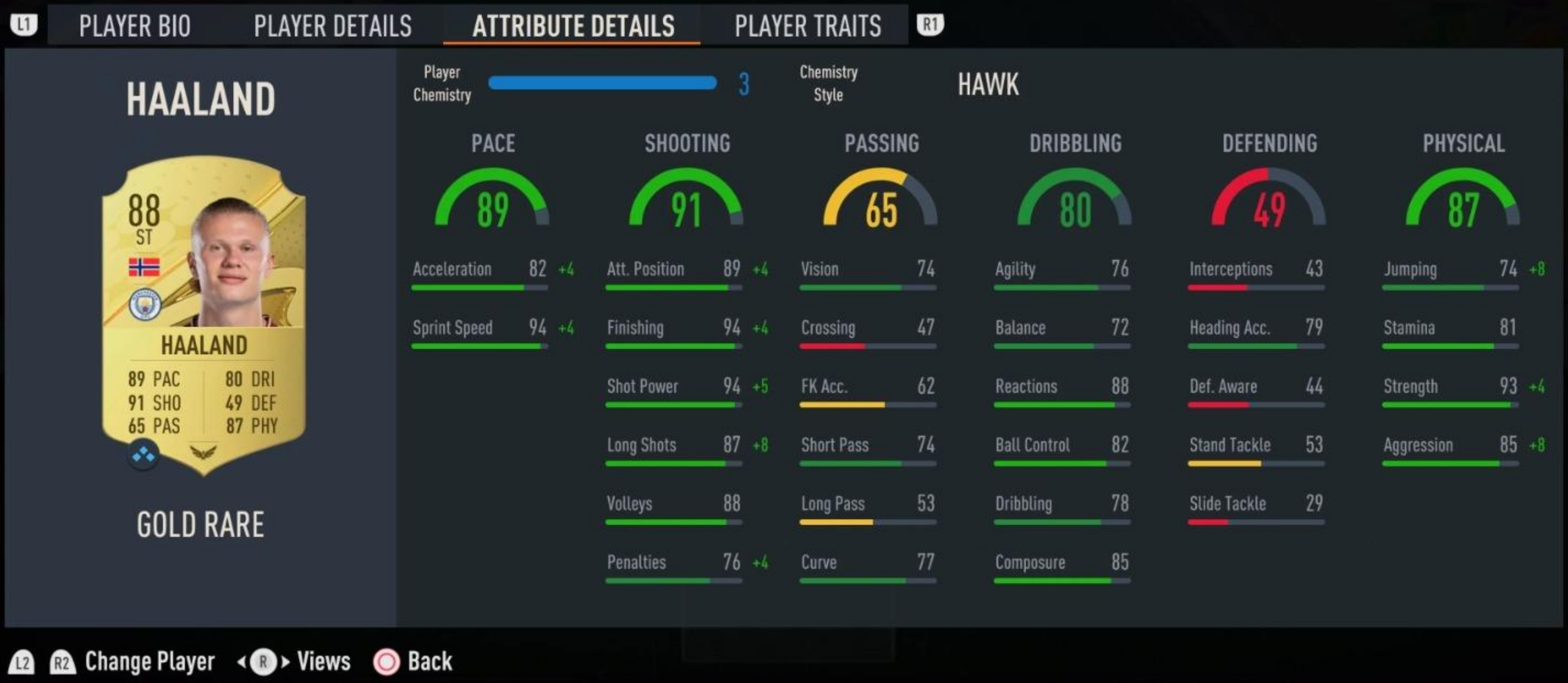
❑ Player stats are the real life data of the players analysed by EA

❑ The 6 main stats mentioned in these cards are:

Pace,
Shooting,
Passing,
Dribbling,
Defending,
and Physical



However, these stats are an average of the complete player stats



HAALAND

88
ST

HAALAND

89	PAC	80	DRI
91	SHO	49	DEF
65	PAS	87	PHY

GOLD RARE

Player Chemistry 3

Chemistry Style HAWK

PACE 89		
Acceleration	82	+4
Sprint Speed	94	+4

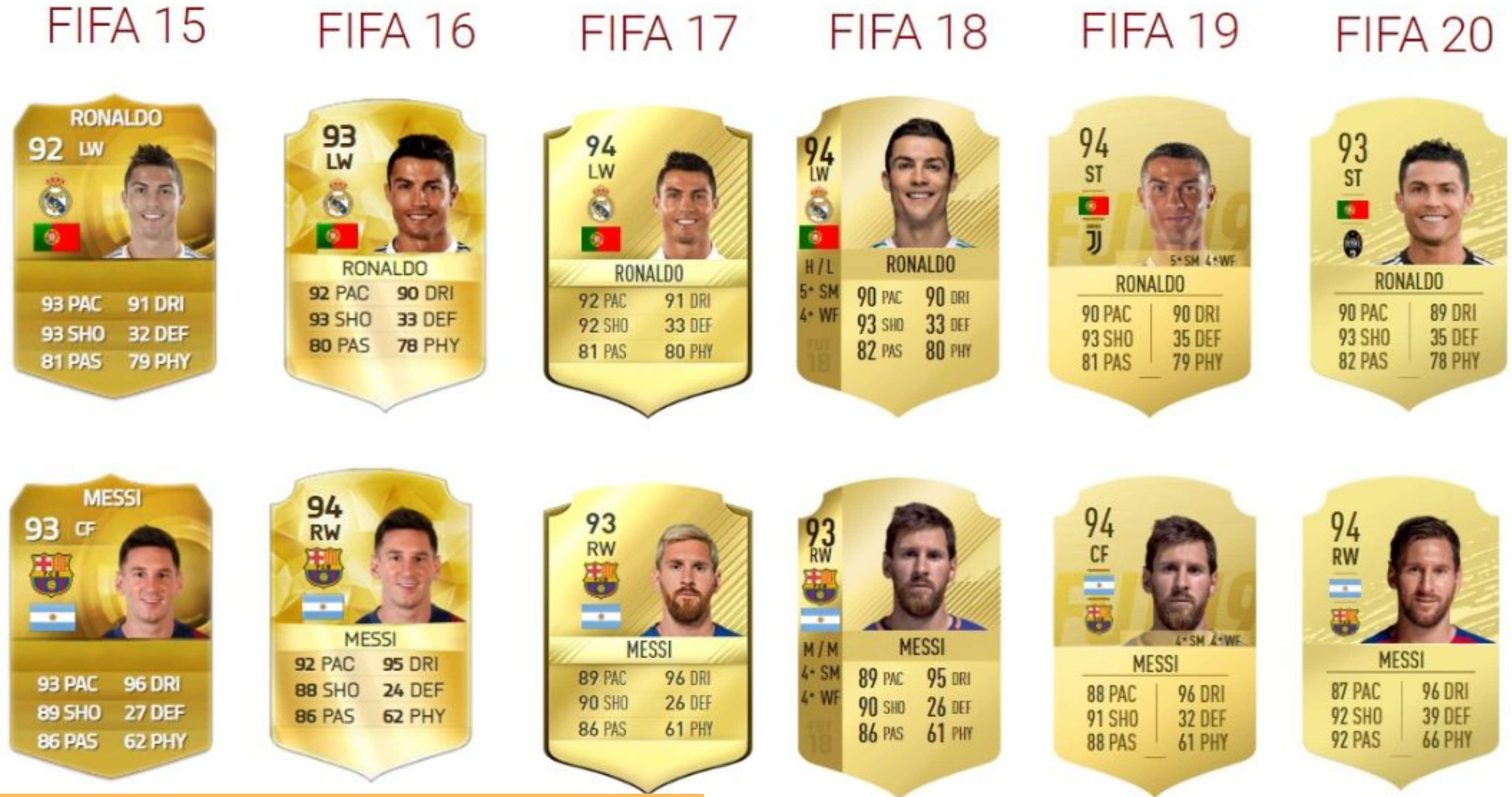
SHOOTING 91		
Att. Position	89	+4
Finishing	94	+4
Shot Power	94	+5
Long Shots	87	+8
Volleys	88	
Penalties	76	+4

PASSING 65	
Vision	74
Crossing	47
FK Acc.	62
Short Pass	74
Long Pass	53
Curve	77

DRIBBLING 80	
Agility	76
Balance	72
Reactions	88
Ball Control	82
Dribbling	78
Composure	85

DEFENDING 49	
Interceptions	43
Heading Acc.	79
Def. Aware	44
Stand Tackle	53
Slide Tackle	29

PHYSICAL 87		
Jumping	74	+8
Stamina	81	
Strength	93	+4
Aggression	85	+8



Player stats change over the years:

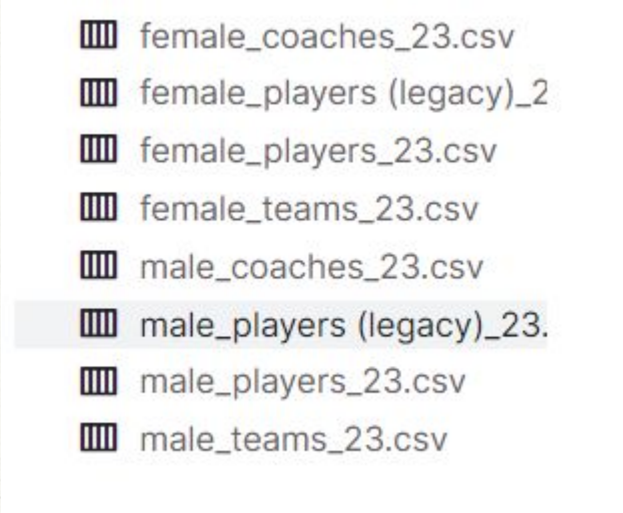
Kaggle Data Set

The data set on Kaggle had many tables.
Due to the accumulation of stats of players, coaches and the teams over the years, the data was HUGE!

We decided to narrow our focus onto Male players and coaches.

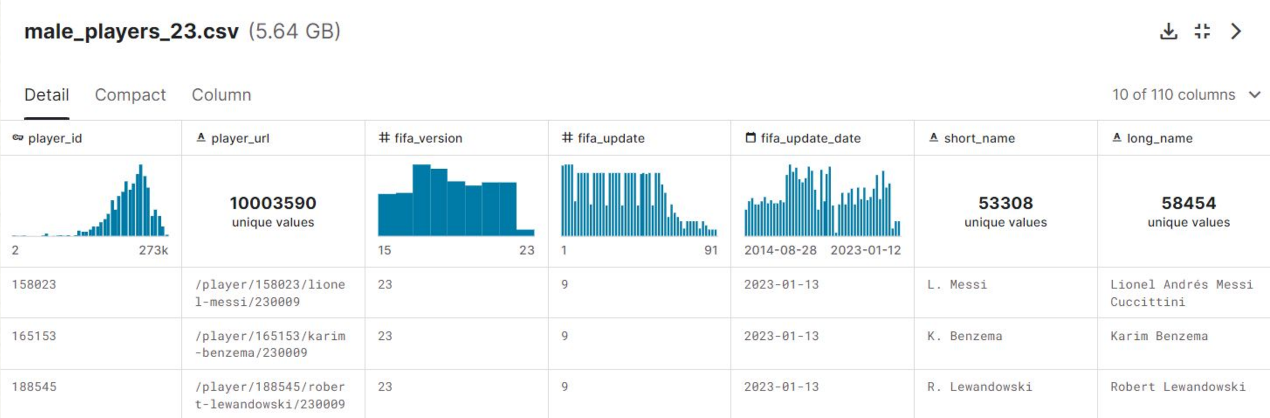
We chose 2 tables to extract our data from:

❑ male_players_23.csv
❑ male_coaches_23.csv



female_coaches_23.csv
female_players (legacy)_2
female_players_23.csv
female_teams_23.csv
male_coaches_23.csv
male_players (legacy)_23.
male_players_23.csv
male_teams_23.csv

Working on the CSV files



male_players_23.csv (5.64 GB)

player_id	player_url	# fifa_version	# fifa_update	fifa_update_date	short_name	long_name
2 273k	10003590 unique values	15 23	1 91	2014-08-28 2023-01-12	53308 unique values	58454 unique values
158023	/player/158023/lionel-messi/230009	23	9	2023-01-13	L. Messi	Lionel Andrés Messi Cuccittini
165153	/player/165153/karim-benzema/230009	23	9	2023-01-13	K. Benzema	Karim Benzema
188545	/player/188545/robert-lewandowski/230009	23	9	2023-01-13	R. Lewandowski	Robert Lewandowski

- ❏ 110 columns on unfiltered raw data
- ❏ Contained player personal details as well as the performance statistics all in one csv file

DIVIDING THE CONTENT INTO TABLES:

Players

24 columns
Contains only personal details such as name, jersey number, club, nationality_id, dob, etc

★ playerid : gives the unique id of the players over the years

Metrics

72 columns
Contains attributes such as positions (like center back, right wing, etc), goalkeeping, pace, etc

★ metrics_id: contains the unique id of the stats over the years

Player Metrics

2 columns
Contains playerid and metrics_id of the corresponding players

MORE TABLES:

Coach

4 columns
Contains personal details of the coach

★ Coach_id: unique id given to each coach

Nationality

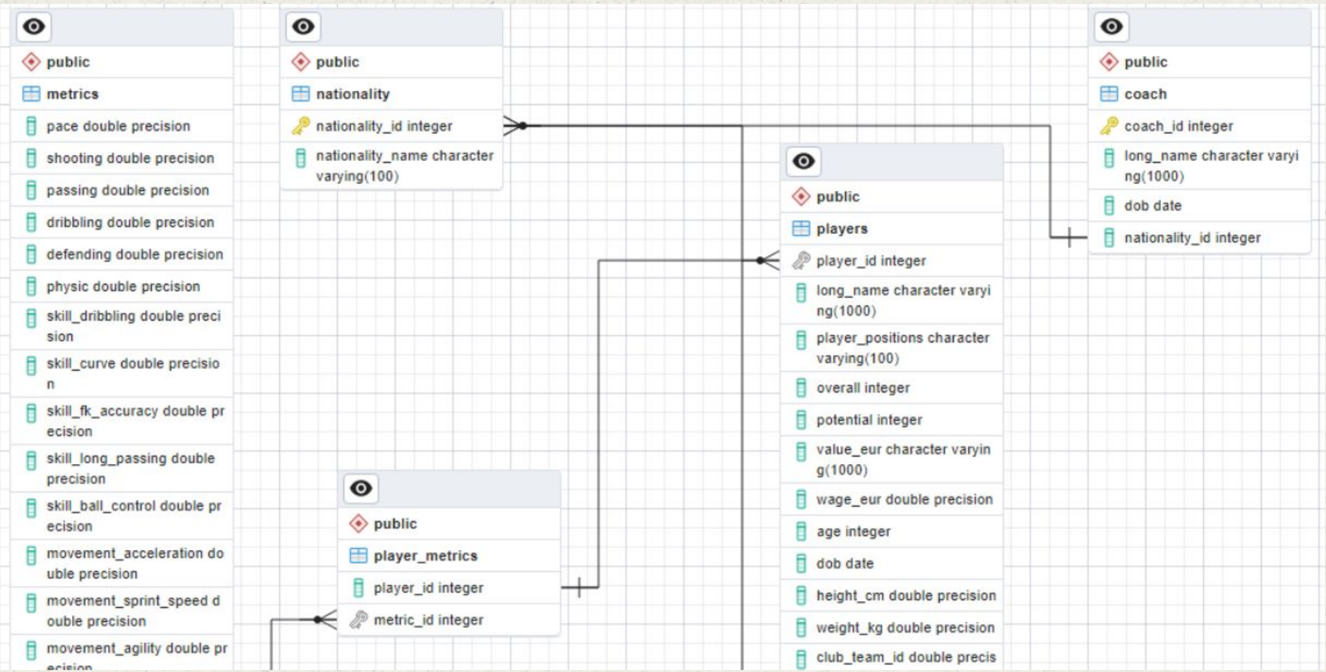
2 columns
Contains nationality_id and nationality

★ Nationality_id: unique id given to each country

ER Diagram of the database

- ❏ 1 to 1 relation between Coach and Nationality.
- ❏ 1 to 1 relation between Players and Player_metrics.
- ❏ 1 to 1 relation between Metrics and Player_metrics.

Relation between the tables:



public
metrics
pace double precision
shooting double precision
passing double precision
dribbling double precision
defending double precision
physic double precision
skill_dribbling double precision
skill_curve double precision
skill_fk_accuracy double precision
skill_long_passing double precision
skill_ball_control double precision
movement_acceleration double precision
movement_sprint_speed double precision
movement_agility double precision

public
nationality
nationality_id integer
nationality_name character varying(100)

public
player_metrics
player_id integer
metric_id integer

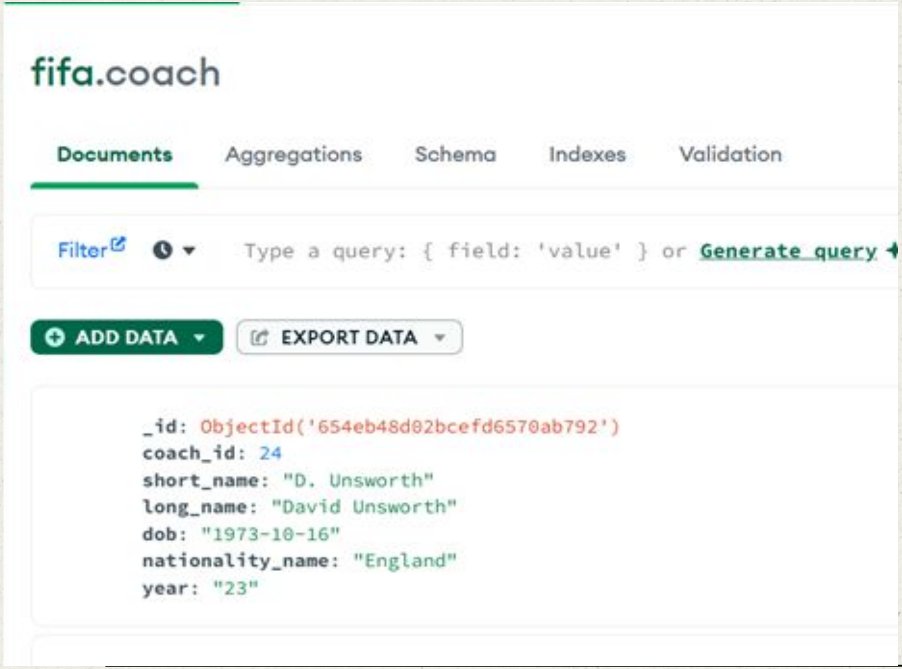
public
players
player_id integer
long_name character varying(1000)
player_positions character varying(100)
overall integer
potential integer
value_eur character varying(1000)
wage_eur double precision
age integer
dob date
height_cm double precision
weight_kg double precision
club_team_id double precis

public
coach
coach_id integer
long_name character varying(1000)
dob date
nationality_id integer

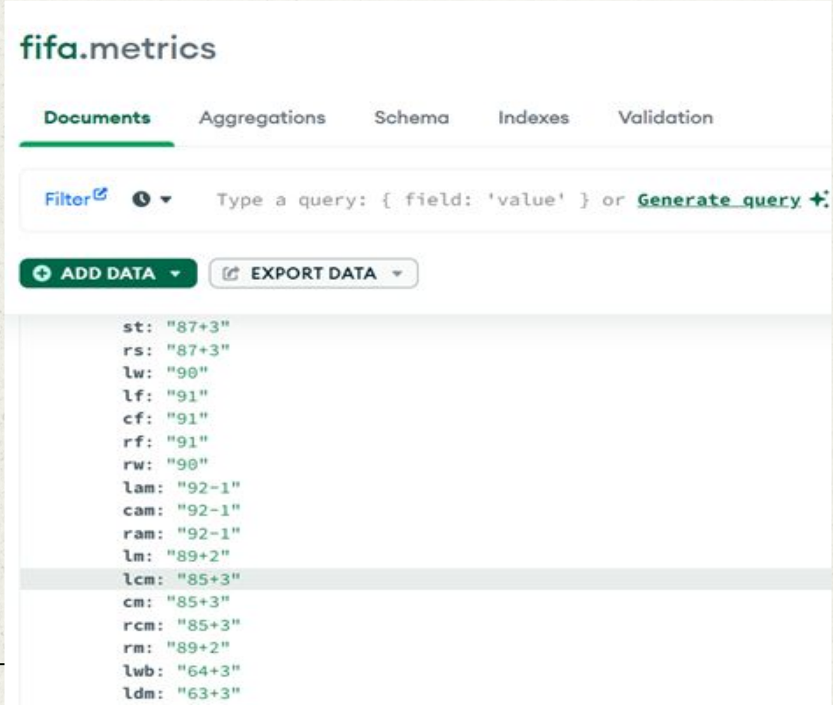
Document Oriented Modeling

- ❏ Document-oriented modeling is a methodology employed in document-oriented databases, a category of NoSQL databases.
- ❏ This approach is centered around the concept of documents, which are independent data entities akin to JSON or XML objects.
- ❏ Unlike traditional relational databases that require a predefined schema and store data in rows across multiple interconnected tables, document-oriented databases are either schema-less or have very flexible schemas. This means that each document can have a unique structure, allowing for a wide variety of data types and structures within the same database.
- ❏ Some examples are - MongoDB, RavenDB , Amazon DynamoDB, Couchbase, etc.

Example of data in MongoDB



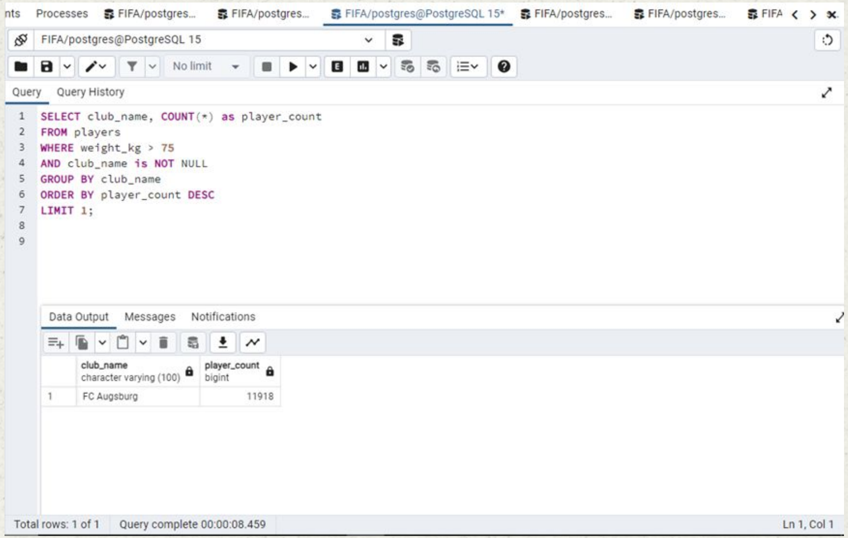
```
_id: ObjectId('654eb48d02bcefd6570ab792')
coach_id: 24
short_name: "D. Unsworth"
long_name: "David Unsworth"
dob: "1973-10-16"
nationality_name: "England"
year: "23"
```



```
st: "87+3"
rs: "87+3"
lw: "90"
lf: "91"
cf: "91"
rf: "91"
rw: "90"
lam: "92-1"
cam: "92-1"
ram: "92-1"
lm: "89+2"
lcm: "85+3"
cm: "85+3"
rcm: "85+3"
rm: "89+2"
lwb: "64+3"
ldm: "63+3"
```

Five Interesting Queries

1. Identify the particular sports club exhibiting a high count of players surpassing a weight threshold of 75 kilograms.



FIFA/postgres@PostgreSQL 15

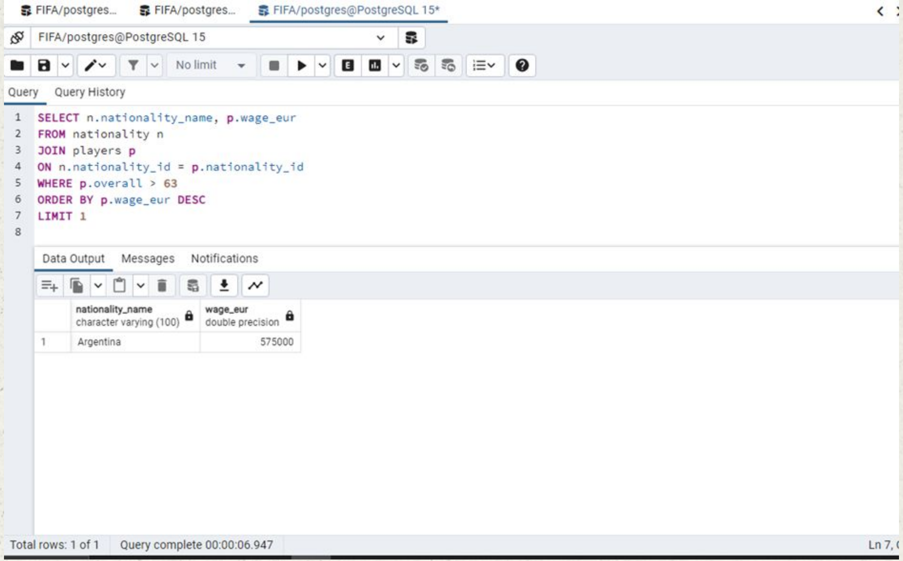
No limit

```sql
SELECT club_name, COUNT(*) as player_count
FROM players
WHERE weight_kg > 75
AND club_name is NOT NULL
GROUP BY club_name
ORDER BY player_count DESC
LIMIT 1;
```

	club_name character varying (100)	player_count bigint
1	FC Augsburg	11918

Total rows: 1 of 1   Query complete 00:00:08.459   Ln 1, Col 1

2. Determine the specific national origin attributed to the player with the most substantial compensation package whose overall performance registers above the threshold of 63.

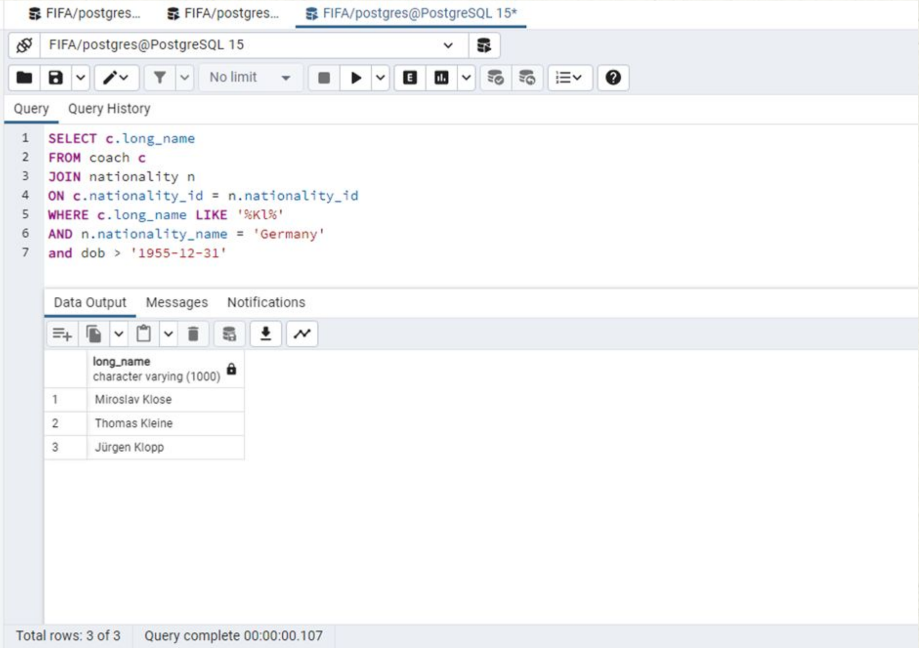


```sql
SELECT n.nationality_name, p.wage_eur
FROM nationality n
JOIN players p
ON n.nationality_id = p.nationality_id
WHERE p.overall > 63
ORDER BY p.wage_eur DESC
LIMIT 1
```

	nationality_name character varying (100)	wage_eur double precision
1	Argentina	575000

Total rows: 1 of 1 Query complete 00:00:06.947 Ln 7, (

3. Enlist all coaching personnel hailing from Germany, specifically those whose surname initiates with the consonantal sequence 'Kl' and whose date of birth falls after the year 1955.

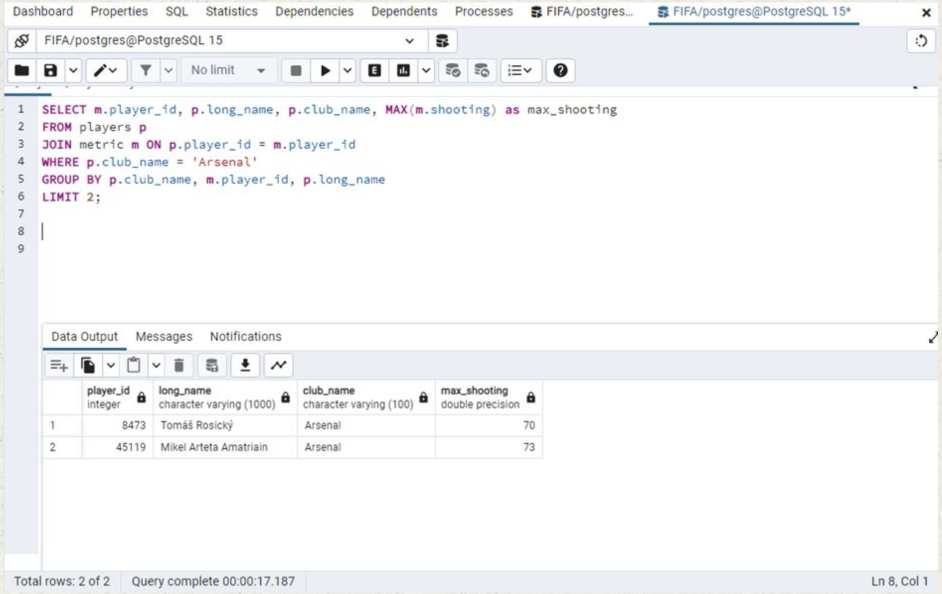


```sql
SELECT c.long_name
FROM coach c
JOIN nationality n
ON c.nationality_id = n.nationality_id
WHERE c.long_name LIKE '%Kl%'
AND n.nationality_name = 'Germany'
and dob > '1955-12-31'
```

	long_name character varying (1000)
1	Miroslav Klose
2	Thomas Kleine
3	Jürgen Klopp

Total rows: 3 of 3   Query complete 00:00:00.107

4. Identify top 2 athletes within the Arsenal roster who exhibits the most elevated measure of their shooting skill.

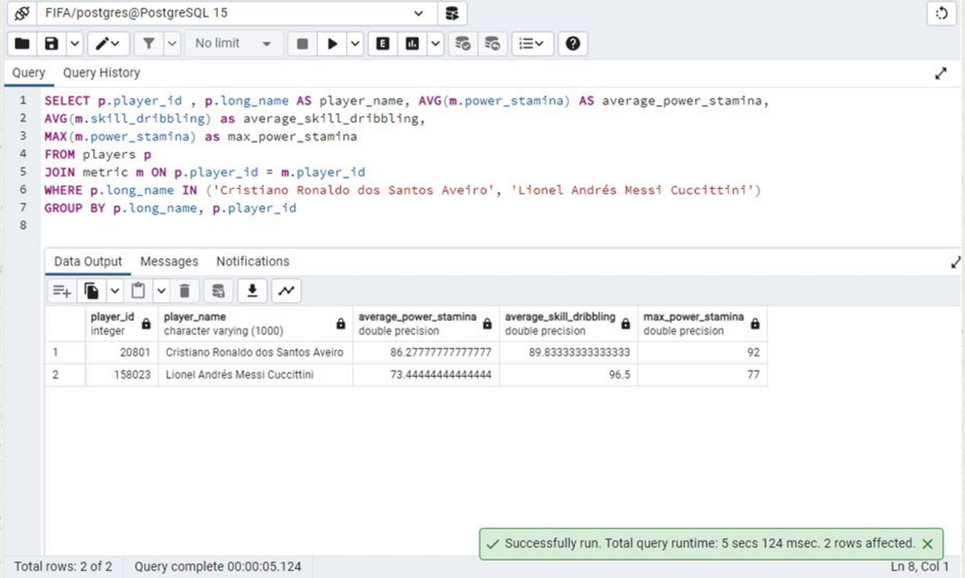


```sql
SELECT m.player_id, p.long_name, p.club_name, MAX(m.shooting) as max_shooting
FROM players p
JOIN metric m ON p.player_id = m.player_id
WHERE p.club_name = 'Arsenal'
GROUP BY p.club_name, m.player_id, p.long_name
LIMIT 2;
```

	player_id integer	long_name character varying (1000)	club_name character varying (100)	max_shooting double precision
1	8473	Tomáš Rosický	Arsenal	70
2	45119	Mikel Arteta Amatriain	Arsenal	73

Total rows: 2 of 2 Query complete 00:00:17.187 Ln 8, Col 1

5. Comparing the statistics for Lionel Messi and Cristiano Ronaldo’s performance in the last 8 years.



```
SELECT p.player_id , p.long_name AS player_name, AVG(m.power_stamina) AS average_power_stamina,
AVG(m.skill_dribbling) as average_skill_dribbling,
MAX(m.power_stamina) as max_power_stamina
FROM players p
JOIN metric m ON p.player_id = m.player_id
WHERE p.long_name IN ('Cristiano Ronaldo dos Santos Aveiro', 'Lionel Andrés Messi Cuccittini')
GROUP BY p.long_name, p.player_id
```

	player_id integer	player_name character varying (1000)	average_power_stamina double precision	average_skill_dribbling double precision	max_power_stamina double precision
1	20801	Cristiano Ronaldo dos Santos Aveiro	86.27777777777777	89.83333333333333	92
2	158023	Lionel Andrés Messi Cuccittini	73.44444444444444	96.5	77

✓ Successfully run. Total query runtime: 5 secs 124 msec. 2 rows affected.

Total rows: 2 of 2 Query complete 00:00:05.124 Ln 8, Col 1

Indexing

- Indexing in databases is a technique that improves the speed of data retrieval operations.

- The primary advantage of indexing is significantly faster data retrieval compared to scanning the entire table, especially in large databases.

- This speed-up is crucial for performance, particularly in search and join operations.

Queries	Time without Indexing (seconds)	Time after Indexing (seconds)
Query 1	8.459	6.1
Query 2	6.947	0.099
Query 3	0.107	0.075
Query 4	17.187	0.714
Query 5	5.124	3.8

Functional Dependencies

- A functional dependency is a relationship between two sets of attributes in a relational database.
- It represents the fact that the values of one set of attributes uniquely determine the values of another set. In other words, if we have a functional dependency A -> B, it means that for every unique combination of values in A, there is a unique corresponding combination of values in B.
- Types: Trivial, Non-trivial, Transitive.
- In the players table, several functional dependencies can be identified based on the relationships between different attributes. The primary identity dependency is established by the player ID, which uniquely determines all other attributes associated with a player, including personal details, club affiliations, and skill-related information.
- In the nationality table, the primary key is nationality_id, and it uniquely identifies each nationality. Therefore, the functional dependency can be expressed as: nationality_id → nationality_name.

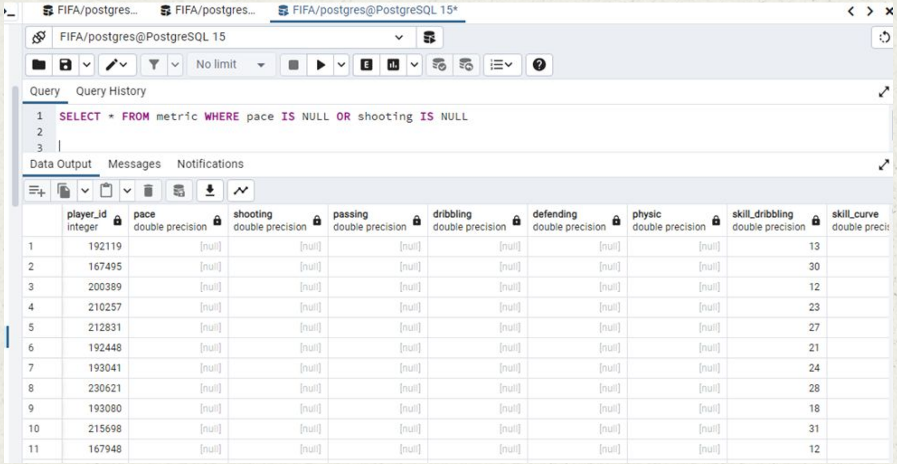
Normalization

- Normalization involves breaking down a large table into smaller, more organized groups of related information to reduce redundancy and improve data integrity.
- It ensures that data is neatly arranged and that identical information isn't repeated in multiple places.
- In the players table, the foreign key nationality_id referencing another table, aids to normalization. This design choice contributes to reducing redundancy in the database, as detailed information about nationalities is stored in separate tables, and player records only contain references to these entities.
- The use of foreign keys also ensures data integrity, as it enforces referential consistency between the players table and the referenced table nationality.
- The coach table has a unique identifier, coach_id and individual columns for coach details. It follows the basics of the First Normal Form (1NF) by keeping each piece of information separate.

Data Cleaning

- Data cleaning in SQL involves manipulating and modifying the data in a database to handle issues such as missing or inconsistent values. It aims to enhance the reliability and accuracy of datasets.
- We had some null values in some of the column, hence we would be eliminating those.
- Some ways of cleaning data:
  - Handling missing values
  - Removing duplicates
  - Data Validation
  - Data Type Conversion
  - Data Integration
  - Error Correction

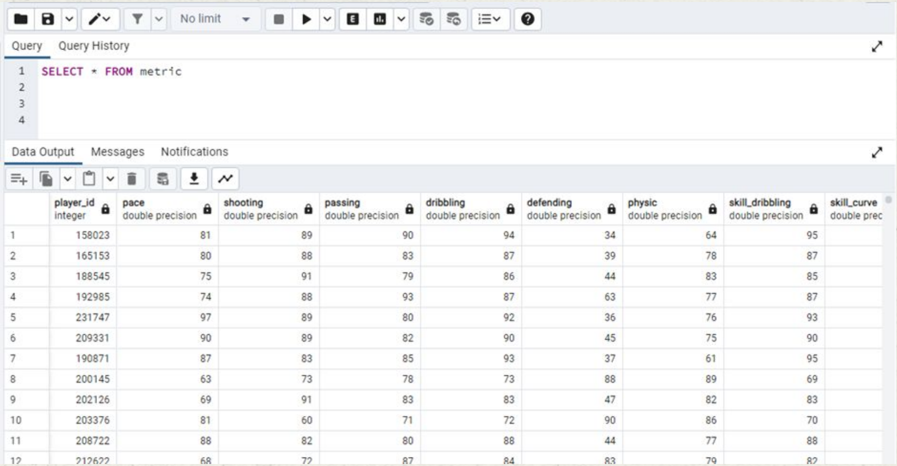
Data Cleaning



```
SELECT * FROM metric WHERE pace IS NULL OR shooting IS NULL
```

	player_id integer	pace double precision	shooting double precision	passing double precision	dribbling double precision	defending double precision	physic double precision	skill_dribbling double precision	skill_curve double precis
1	192119	[null]	[null]	[null]	[null]	[null]	[null]	13	
2	167495	[null]	[null]	[null]	[null]	[null]	[null]	30	
3	200389	[null]	[null]	[null]	[null]	[null]	[null]	12	
4	210257	[null]	[null]	[null]	[null]	[null]	[null]	23	
5	212831	[null]	[null]	[null]	[null]	[null]	[null]	27	
6	192448	[null]	[null]	[null]	[null]	[null]	[null]	21	
7	193041	[null]	[null]	[null]	[null]	[null]	[null]	24	
8	230621	[null]	[null]	[null]	[null]	[null]	[null]	28	
9	193080	[null]	[null]	[null]	[null]	[null]	[null]	18	
10	215698	[null]	[null]	[null]	[null]	[null]	[null]	31	
11	167948	[null]	[null]	[null]	[null]	[null]	[null]	12	

Before



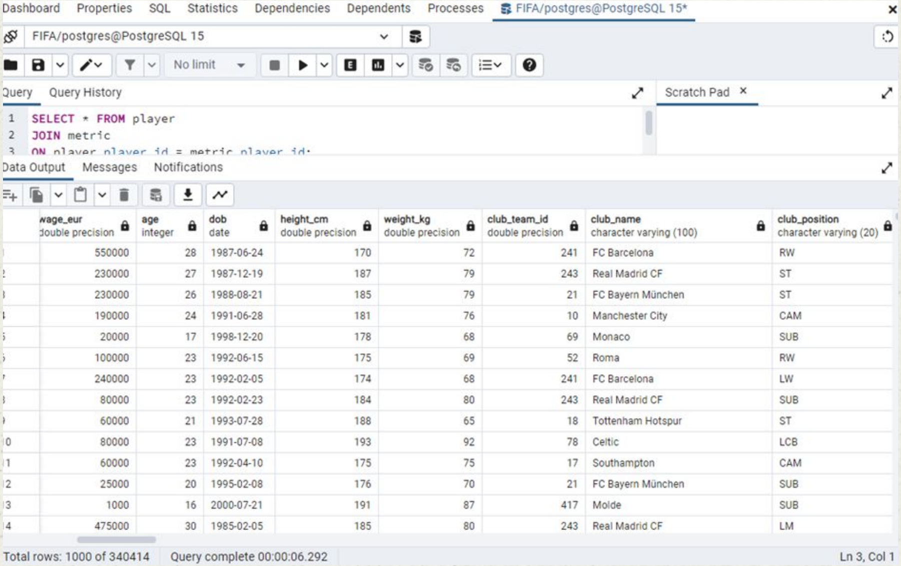
```
SELECT * FROM metric
```

	player_id integer	pace double precision	shooting double precision	passing double precision	dribbling double precision	defending double precision	physic double precision	skill_dribbling double precision	skill_curve double prec
1	158023	81	89	90	94	34	64	95	
2	165153	80	88	83	87	39	78	87	
3	188545	75	91	79	86	44	83	85	
4	192985	74	88	93	87	63	77	87	
5	231747	97	89	80	92	36	76	93	
6	209331	90	89	82	90	45	75	90	
7	190871	87	83	85	93	37	61	95	
8	200145	63	73	78	73	88	89	69	
9	202126	69	91	83	83	47	82	83	
10	203376	81	60	71	72	90	86	70	
11	208722	88	82	80	88	44	77	88	
12	212622	68	72	87	84	83	79	82	

After

Data Integrity

- ❑ Maintaining data integrity in SQL, especially when joining tables, involves enforcing constraints, relationships, and actions that ensure the accuracy and consistency of the data.
- ❑ Additionally, integrating tables can significantly enhance the accuracy of the data. By breaking down data into tables and defining clear relationships, reducing the risk of data redundancy and inconsistencies.
- ❑ Hence we had integrated some tables in our database for ease of access and getting accuracy.



```sql
SELECT * FROM player
JOIN metric
ON player.player_id = metric.player_id;
```

wage_eur double precision	age integer	dob date	height_cm double precision	weight_kg double precision	club_team_id double precision	club_name character varying (100)	club_position character varying (20)
550000	28	1987-06-24	170	72	241	FC Barcelona	RW
230000	27	1987-12-19	187	79	243	Real Madrid CF	ST
230000	26	1988-08-21	185	79	21	FC Bayern München	ST
190000	24	1991-06-28	181	76	10	Manchester City	CAM
20000	17	1998-12-20	178	68	69	Monaco	SUB
100000	23	1992-06-15	175	69	52	Roma	RW
240000	23	1992-02-05	174	68	241	FC Barcelona	LW
80000	23	1992-02-23	184	80	243	Real Madrid CF	SUB
60000	21	1993-07-28	188	65	18	Tottenham Hotspur	ST
80000	23	1991-07-08	193	92	78	Celtic	LCB
60000	23	1992-04-10	175	75	17	Southampton	CAM
25000	20	1995-02-08	176	70	21	FC Bayern München	SUB
1000	16	2000-07-21	191	87	417	Molde	SUB
475000	30	1985-02-05	185	80	243	Real Madrid CF	LM

Total rows: 1000 of 340414 Query complete 00:00:06.292 Ln 3, Col 1

Relational Model vs Document Oriented Model

Based on	Relational Model	Document Oriented Model
Schema	Consists of a fixed schema	It is schema less consisting of documents of different data structures.
Data Structure	Tables with rows and columns	The documents are of JSON,XML type
Query Language	SQL	Differs system by system
Normalization	Highly normalized data	Often denormalized, data might be duplicated.
Data Integrity	Good data integrity consisting of ACID properties.	Mostly focused on performance and scalability.

Itemset Mining

- In itemset mining, the concept of lattices plays a pivotal role, particularly in how itemsets are organized and systematically explored.
- A lattice in this context represents the complete set of all possible itemsets, structured in a way that mirrors their subset-superset relationships.
- The lattice starts with the smallest possible itemsets (individual items) and expands to include larger and more complex combinations of items.
- Algorithms like Apriori leverage this lattice structure to efficiently mine for frequent itemsets.
- In our project, we implemented itemset mining and got a total of 5 lattices.
- Lattice 1 - 5132 combinations of players who have been in the same club for a long time.
- Lattice 2 - 741 combination of players.
- Lattice 3 - 39 combination of players.
- Lattice 4 - 7 combination of players - meaning 7 combinations of 4 players.
- And lastly, lattice 5 had no combination of players.

Conclusion

- Through this project, we have discovered and understood various concepts of databases and querying.
- We started with getting the dataset, processing it , exploring SQL and querying it.
- Followed by generating interesting queries and understanding various clauses and features in SQL.
- Various concepts like indexing, functional dependencies, normalization were implemented.
- We introduced document oriented system and explored the loading of data to it.
- The last phase consisted of data cleaning, data integrity and itemset mining.
- Overall, this project helped us to dive deep into various database concepts and also taught us the ways to handle huge datasets.

THANK YOU