2024 Ohio State Offensive Players

Karissa Merillat | Huntington University | CS 415 Database Management – Fall 2025 - Final Project

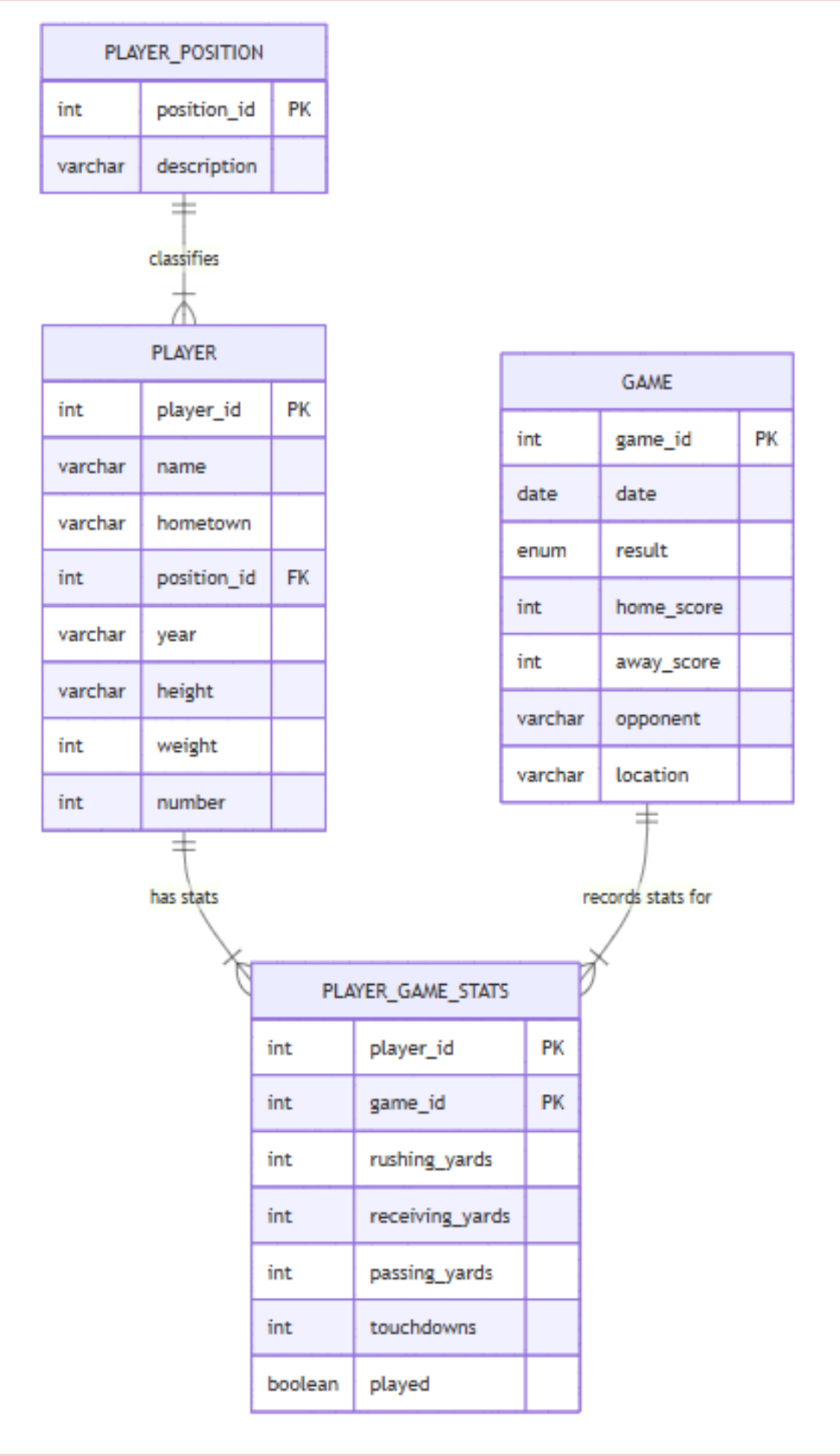
Overview

- My database is designed for a college football program that needs to efficiently store and analyze individual player performance throughout a season. This organization relies on accurate statistical tracking to evaluate athletes, prepare weekly scouting reports, and support coaching decisions. By centralizing information about game results, detailed offensive statistics, players, and their positions, the database provides the coaching staff, analysts, and support personnel with a unified and reliable system for understanding team performance.
- From a user's perspective, the database functions as an organized system that allows coaches, analysts, and student staff members to quickly retrieve and update information about players and their performance. After each game, a user can enter rushing, receiving, passing, and touchdown statistics, as well as whether the player participated in the game, into the player_game_stats table. Users can also review player information such as position, class year, and jersey number to support tasks like roster sheet creation, scouting breakdowns, and media preparation.

Design

- The design of my database is fully normalized and has a relational structure that intends to store and analyze football player performance data across a full season. This database model separates the data into 4 distinct tables. We can see the basic information for each player on the team, each game in the season, and the positions on the team. Using those three tables, they can be combined to get the 4th table that describes each player's stats for every game. The tables depend solely on their primary keys, following the Third Normal Form. For example, player details such as jersey number, height, and hometown are stored only once in the player table to prevent unnecessary duplication when generating statistics for multiple games
- A key design choice I made was creating a junction table, player_game_stats, to represent the many-to-many relationship between players and games. To avoid this issue, the player_game_stats table has a composite primary key of player_id and game_id. This composite key is used to uniquely identify each performance entry. The player_game_stats table also limited the stats to the measurable performance metrics: rushing, receiving, passing, and touchdowns, which allows game-by-game analytics without repeating player or game attributes.
- The database also normalizes player positions by storing them in a separate player_position table. I could have kept the full position name for each player, but I decided to create a whole table for the position. The position_id serves as a foreign key in the player table that references a single record in the player_position table. Making this choice avoids inconsistency in position naming, makes updates easier, and maintains referential integrity. Overall, the design enforces clean separation of entities while supporting flexible reporting and efficient statistical queries.

ER Model



Data

- **Player**
- The player table stores personal and static information for each football player, such as name, class year, hometown, physical attributes, and jersey number. It links to the player_position table through a foreign key to classify each player. Each entry represents one unique athlete in the system.
- **Player_Position**
- The player_position table defines the set of football positions, each assigned to an ID number. Storing these in a separate table prevents duplication and ensures consistency. It is referenced by the player table to assign each player exactly one position category.
- **Game**
- The game table contains information on each scheduled game, including the date, opponent, location, and final scores. It serves as the parent entity for all statistics recorded during that game. Each entry represents one unique matchup.
- **Player_Game_Stats**
- The player_game_stats table records all performance statistics for a single player in a single game. Because players participate in multiple games, and games include multiple players, this table resolves the many-to-many relationship between player and game tables. It includes rushing, receiving, passing yards, touchdowns, and whether the player participated in that game. A player having zero stats because they did not participate means something different than a player having zero stats while playing.

Queries

- This query shows the total yards, including passing, rushing, and receiving, each player has when they played in the game. This can be useful when figuring out the leaders of a certain game. Also, excluding players who did not play makes the data easier to read.

```
SELECT
    p.name,
    pgs.game_id,
    (pgs.rushing_yards + pgs.receiving_yards + pgs.passing_yards) AS total_yards
FROM player_game_stats pgs
JOIN player p ON pgs.player_id = p.player_id
WHERE pgs.played <> 0;
```

- This query shows which players had more the 3 touchdowns in the entire season. This is helpful when tracking leaders on the team and what positions have to be filled when players leave the program.

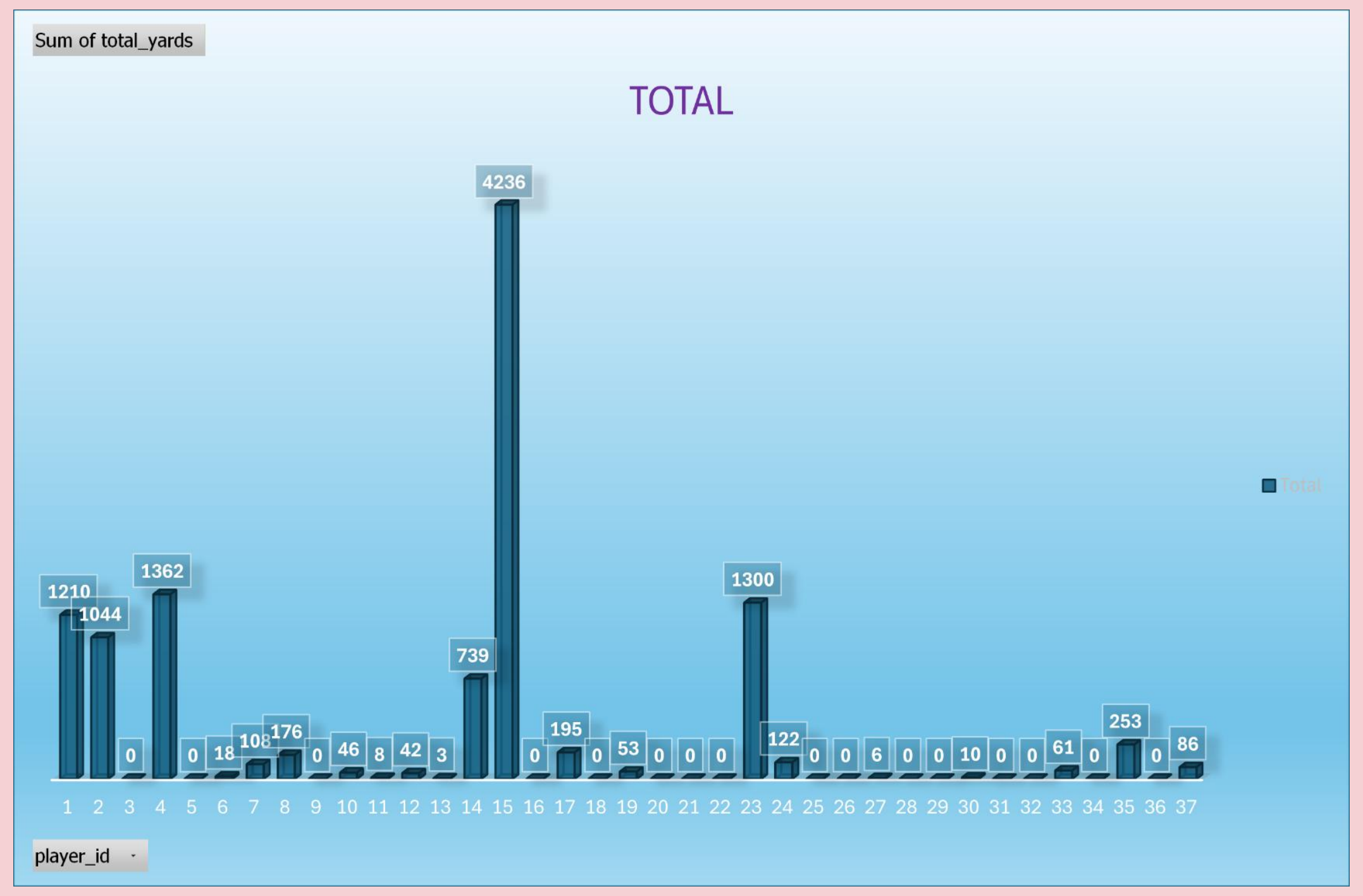
```
SELECT
    p.name,
    SUM(s.touchdowns) AS total_tds
FROM player_game_stats AS s
JOIN player AS p ON s.player_id = p.player_id
GROUP BY p.player_id
HAVING total_tds > 3;
```

- This query creates a view called offensive_totals that sums the total rushing, receiving, and passing yards and touchdowns for each player. This type of view makes it easy to see the impact each player had on the team.

```
CREATE VIEW offensive_totals AS
SELECT
    p.player_id,
    p.name,
    SUM(s.rushing_yards) AS total_rush,
    SUM(s.receiving_yards) AS total_receive,
    SUM(s.passing_yards) AS total_pass,
    SUM(s.touchdowns) AS total_tds
FROM player_game_stats s
JOIN player p ON s.player_id = p.player_id
GROUP BY p.player_id;

SELECT * FROM offensive_totals
ORDER BY total_tds DESC;
```

Reports



Number of Touchdowns	Game															
Player	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Grand Total
Air Noland	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bennett Christian	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Brandon Inniss	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1
Brennen Schramm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bryson Rodgers	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Carnell Tate	1	0	1	0	0	0	1	0	0	2	0	0	0	0	0	0	5
Chad Ray	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Chase Brecht	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Damarion Witten	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
David Adolph	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Devin Brown	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Dorian Williams	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Emeka Egbuka	0	0	1	1	3	1	0	1	1	0	1	0	0	1	0	0	10
Gee Scott Jr.	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	2
Jace Middleton	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
James People	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
Jayden Ballard	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Jelani Thurman	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1
Jeremiah Smith	2	1	1	2	1	1	1	0	1	0	0	1	2	2	0	1	16
Joop Mitchell	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Julian Sayin	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Kojo Antwi	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Lincoln Kienholz	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Mason Maggs	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Maxence LeBlanc	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Mylan Graham	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Nolan Baudo	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Patrick Gurd	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Quinshon Judkins	1	2	2	0	0	1	1	0	0	2	0	0	2	0	2	3	16
Rashid SeSay	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Sam Dixon	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Shawn Lodge	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TC Caffey	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TreVeyon Henderson	0	2	2	0	0	0	0	0	1	0	1	0	2	2	1	0	11
Will Howard	4	2	3	3	5	3	3	2	4	2	2	1	2	3	1	2	42
Will Kacmarek	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Zak Herbstreit	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Grand Total	**9**	**10**	**10**	**7**	**9**	**6**	**6**	**4**	**8**	**6**	**5**	**2**	**8**	**8**	**4**	**6**	**108**

Future Work

- This database can be used in the future for any football team. All the information needed would be players, games, and the offensive stats. The database will also be helpful when signing players in NFL drafts or when looking at the transfer portal.

Works Cited

- https://ohiostatebuckeyes.com/sports/football