 📚 **Lesson** | Joining Data

— Inner Join

PROMPT: Data can be found in some unexpected spaces. Take the museum, for example. While it's not the first institution that comes to mind when you think of "data-driven decisions", like any business, museums can benefit from an analysis of who is visiting, what customers are interacting with, and other metrics that can help improve their offerings and financial stability. Let’s practice using inner joins to stitch data together. It's time to apply a join to our NYC MoMA dataset, and explore the combined artist and artwork information.

1. Please write a query that combines data from the **moma\_artworks** and **moma\_artists** tables.

|  |
| --- |
| select \*  from moma\_artworks ma  join moma\_artists mart  on ma.artistid = mart.artistid |

— Aggregating Data with INNER JOINS (Part 1)

PROMPT: After we join two tables together, the result is a single table that combines the information from the original two. You can use everything else you learned so far on that joined table, such as grouping and aggregation. Let's go through an example of this, using data about the MoMA.

1. How can we determine the most common nationalities among artists in MoMA? What are the three most common?

|  |
| --- |
| select  nationality,  count(nationality) as number\_of\_artists  from moma\_artworks ma  join moma\_artists mart  on ma.artistid = mart.artistid  group by 1  order by 2 DESC  LIMIT 3 |

|  |
| --- |
| American, French, German |

1. How many pieces of artwork in MoMA were created by an artist born in the 21st century?

|  |
| --- |
| select count(artworkid)  from moma\_artworks ma  join moma\_artists mart  on ma.artistid = mart.artistid  where birthyear>=2001 |

|  |
| --- |
| 116 |

1. Is it more common for pieces of artwork in MoMA to have been created by an artist who is still alive or who has perished?

|  |
| --- |
| select  case when deathyear is not null then 'living' else 'dead' end as status,  count(artworkid)  from moma\_artworks ma  join moma\_artists mart  on ma.artistid = mart.artistid  group by 1 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | dead | 57614 | | living | 84667 | |

— Aggregating Data with INNER JOINS (Part 2)

PROMPT: After we join two tables together, the result is a single table that combines the information from the original two. You can use everything else you learned so far on that joined table, such as grouping and aggregation. Let's go through an example of this, using data about the NBA.

1. Which NBA division has the highest average points scored per game based on the available team statistics?

|  |
| --- |
| select  division,  avg(pts\_scored\_per\_game) as avg\_pts  from nba\_team\_stats nts  join nba\_team\_conferences\_divisions ntcd  on ntcd.team = nts.team  group by 1  order by 2 desc |

|  |
| --- |
| Pacific Division 104.346 pts |

1. Which NBA conference has the highest average points scored per game based on the available team statistics?

|  |
| --- |
| select  conference,  avg(pts\_scored\_per\_game) as avg\_pts  from nba\_team\_stats nts  join nba\_team\_conferences\_divisions ntcd  on ntcd.team = nts.team  group by 1  order by 2 desc |

|  |
| --- |
| Western Conference 103.647 pts |

— WITH Keyword

PROMPT: We have several tables of data on the NBA (nba\_games, nba\_players, nba\_team\_conference\_divisions, nba\_team\_stats, and nba\_teams). Imagine being able to define a temporary dataset and its output, then recalling that temporary data for later use. Kind of like a 'save as' option for queries. Useful, right? Well, meet the WITH keyword that does just that! It's a way to simplify complex queries and retrieve them for later use without having to constantly modify the query. Let's see it in action on some more NBA data.

1. Can you create a temporary table that lists all teams along with their division, conference, and performance metrics such as points and rebounds at home for the 2019-20 NBA season?

|  |
| --- |
| with temp\_table as (  select  team,  division,  conference,  pts\_home,  reb\_home  from nba\_games ng  join nba\_team\_conferences\_divisions ntcd  on ng.team\_home = ntcd.team  where season = '2019-20'  )  select \* from temp\_table |

1. Using the temporary table created, how can you calculate the average points and rebounds scored at home by teams in each division?

|  |
| --- |
| with temp\_table as (  select  team,  division,  conference,  pts\_home,  reb\_home  from nba\_games ng  join nba\_team\_conferences\_divisions ntcd  on ng.team\_home = ntcd.team  where season = '2019-20'  )  select  division,  avg(pts\_home) as avg\_pts,  avg(reb\_home) as avg\_rebound  from temp\_table  group by 1 |

|  |  |  |
| --- | --- | --- |
| Atlantic Division | 111.197183098592 | 46.4460093896714 |
| Central Division | 111.09595959596 | 45.3989898989899 |
| Northwest Division | 112.548076923077 | 45.0576923076923 |
| Pacific Division | 112.649769585253 | 45.1013824884793 |
| Southeast Division | 110.303482587065 | 44.1791044776119 |
| Southwest Division | 115.808823529412 | 46.4117647058824 |