Olympic SQL Project Documentation

- Downloaded file from Kaggle, data is hosted on my local computer: https://www.kaggle.com/datasets/heesoo37/120-years-of-olympic-history-athletes-and-results

SQL queries performed in PG ADMIN 4

PART: 1 LOAD DATA INTO DATASETS

- Created Database in PG Admin 4 called Olympic_history
- Created Table in olympic_history
    - Used the following query

```
DROP TABLE IF EXISTS OLYMPICS_HISTORY;
CREATE TABLE IF NOT EXISTS OLYMPICS_HISTORY
(
	id		INT,
	name	VARCHAR,
	sex		VARCHAR,
	age		VARCHAR,
	height	VARCHAR,
	weight	VARCHAR,
	team	VARCHAR,
	noc		VARCHAR,
	games	VARCHAR,
	year	INT,
	seasons	VARCHAR,
	city	VARCHAR,
	sport	VARCHAR,
	event	VARCHAR,
	medal	VARCHAR
);

DROP TABLE IF EXISTS OLYMPICS_HISTORY_NOC_REGIONS;
CREATE TABLE IF NOT EXISTS OLYMPICS_HISTORY_NOC_REGIONS
(
	noc		VARCHAR,
	region	VARCHAR,
	notes	VARCHAR

)
```

- Imported athelete_events data into olympics_history table
    - Changed Escape from ' to " after finding error in dataset on line A250019
- Imported noc_regions data into olympics_history_noc_regions table

```
1  select * from olympics_history
```

Data Output | Notifications | Messages

	id integer	name character varying	sex character varying	age character varying	height character varying	weight character varying	team character varying
1	1	A Dijiang	M	24	180	80	China
2	2	A Lamusi	M	23	170	60	China
3	3	Gunnar Nielsen Aaby	M	24	NA	NA	Denmark
4	4	Edgar Lindenau Aabye	M	34	NA	NA	Denmark/Sweden
5	5	Christine Jacoba Aaftink	F	21	185	82	Netherlands
6	5	Christine Jacoba Aaftink	F	21	185	82	Netherlands
7	5	Christine Jacoba Aaftink	F	25	185	82	Netherlands
8	5	Christine Jacoba Aaftink	F	25	185	82	Netherlands
9	5	Christine Jacoba Aaftink	F	27	185	82	Netherlands
10	5	Christine Jacoba Aaftink	F	27	185	82	Netherlands
11	6	Per Knut Aaland	M	31	188	75	United States

Total rows: 1000 of 271116 | Query complete 00:00:00.762 | Ln 1, Col 31

PART 2: IDENTIFY SPORTS PLAYED IN ALL SUMMER OLYMPIC GAMES

1. FIND TOTAL NO OF SUMMMER OLYMPIC GAMES

```
With t1 as (SELECT count(distinct games) as total_summer_games
            FROM OLYMPICS_HISTORY
           WHERE season = 'Summer'
           )
2.FIND FOR EACH SPORT, HOW MANY GAMES WERE PLAYED IN
    t2 as
           (select distinct sport, games
            from Olympics_history
            where season = 'Summer' order by games),
    t3 as
           (Select sport, count(games) as no_of_games
                 from t2
                group by sport)
```

3. Compare t1 with t2,t3 with an JOIN STATEMENT

```
Select *
from t3
JOIN t1 on t1.total_summer_games = t3.no_of_games;
```

```
t2 as
(select distinct sport, games
 from Olympics_history
 where season = 'Summer' order by games),
t3 as
(Select sport, count(games) as no_of_games
from t2
         group by sport)
Select *
from t3
JOIN t1 on t1.total_summer_games = t3.no_of_games;
```

Data Output | Notifications | Messages

	sport character varying	no_of_games bigint	total_summer_games bigint
1	Swimming	29	29
2	Cycling	29	29
3	Fencing	29	29
4	Gymnastics	29	29
5	Athletics	29	29

PART 3: WRITE SQL QUERY TO FETCH THE TOP 5 ATHLETES WHO WON THE MOST GOLD MEDALS.

```
with t1 as
(Select name , count(1) as total_medals
From olympics_history
Where medal = 'Gold'
group by name
order by count (1) desc),

t2 as
        (Select *, dense_rank() over(order by total_medals desc) as rnk
        from t1)
Select *
From t2
where rnk <= 5;
```

Had to use rank because I couldnt limit by 5 without getting the first 5 athletes from t1
Results:

	name character varying	total_medals bigint	rnk bigint
1	Michael Fred Phelps, II	23	1
2	Raymond Clarence "Ray" Ewry	10	2
3	Larysa Semenivna Latynina (Diriy-)	9	3
4	Frederick Carlton "Carl" Lewis	9	3
5	Paavo Johannes Nurmi	9	3
6	Mark Andrew Spitz	9	3
7	Sawao Kato	8	4
8	Usain St. Leo Bolt	8	4
9	Ole Einar Bjrndalen	8	4
10	Matthew Nicholas "Matt" Biondi	8	4
11	Birgit Fischer-Schmidt	8	4
12	Jennifer Elisabeth "Jenny" Thompson (-Cumpelik)	8	4
13	Aladr Gerevich (-Gerei)	7	5
14	Borys Anfiyanovych Shakhlin	7	5
15	Nikolay Yefimovich Andrianov	7	5
16	Viktor Ivanovych Chukarin	7	5
17	Donald Arthur "Don" Schollander	7	5
18	Vra slavsk (-Odloilov)	7	5

PART 4: WRITE SQL QUERY TO DISPLAY THE TOTAL GOLD, SILVER, AND BRONZE MEDAL WON BY EACH COUNTRY

```
Select * From olympics_history where medal <> 'NA'

-- Need to convert data from row level data to column level data, will need to aggreagate data won by each country.
--Crosstab does the same thing as pivot table
--Create a country columns

create extension tablefunc;
```

- Now you can use crosstab function
    - Need query to be a string
    - This query can only have 3 columns
    - 1 main 2nd ordering 3rd aggregated value from each distinct value

– Specify column names and data type

```
select country
, coalesce(gold, 0) as gold
, coalesce(silver, 0) as silver
, coalesce(bronze, 0) as bronze
from crosstab('Select nr.region as country, medal, count(1) as total_medals
                From olympics_history oh
                join olympics_history_noc_regions nr on nr.noc = oh.noc
                where medal <> ''NA''
                group by  nr.region, medal
                order by  nr.region, medal',
                'values (''Bronze''), (''Gold''), (''Silver'')')
        as result(country VARCHAR, bronze bigint, gold bigint, silver
        bigint)
order by gold desc, silver desc, bronze desc;
```

-You have to add values(str) because it will not automatically compare

- You have to add , coalesce to get null entries to say zero
- Order by gold desc…. To get the top countries first

Results (Up to 10)

	country character varying	gold bigint	silver bigint	bronze bigint
1	USA	2638	1641	1358
2	Russia	1599	1170	1178
3	Germany	1301	1195	1260
4	UK	678	739	651
5	Italy	575	531	531
6	France	501	610	666
7	Sweden	479	522	535
8	Canada	463	438	451
9	Hungary	432	332	371
10	Norway	378	361	294

PART 5: WRITE SQL QUERY TO FETCH THE COUNTRY WHICH WON THE MOST GOLD, SILVER AND BROZNE MEDAL DURING EACH OLYMPIC GAME.

-1st part of query is to segregate the data of games and the associated country and the total number of medal they have won.

```
select substring(games_country, 1, position(' - ' in games_country) - 1) as games
```

```
, substring(games_country, position(' - ' in games_country) + 3) as country
, coalesce(gold, 0) as gold
, coalesce(silver, 0) as silver
, coalesce(bronze, 0) as bronze
from crosstab('Select concat(games, '' - '', nr.region) as games_country, medal, count(1) as
total_medals
                    From olympics_history oh
                    join olympics_history_noc_regions nr on nr.noc = oh.noc
                    where medal <> ''NA''
                    group by  games, nr.region, medal
                    order by  games, nr.region, medal',
                    'values (''Bronze''), (''Gold''), (''Silver'')')
                as result(games_country VARCHAR, bronze bigint, gold bigint, silver bigint)
order by games_country;
```

- 2nd part

  Move all query into with clause.
  For each olypmic game, create a separate partition for each game, in each partition see who has the highest medals
  Need to do partition of data and sorting of data at the same time
  Can use the max function or first value function of the window functions
  -For each game its going to create a partition and its going to order the data by gold in dec order
  -Corrresponding to each Olympic games I have a country that has the highest gold, '' silver, '' bronze with the total number of medals they've won

```
with temp as
    (select substring(games_country, 1, position(' - ' in games_country) -
    1) as games
    , substring(games_country, position(' - ' in games_country) + 3) as
    country
    , coalesce(gold, 0) as gold
    , coalesce(silver, 0) as silver
    , coalesce(bronze, 0) as bronze
    from crosstab('Select concat(games, '' - '', nr.region) as
    games_country, medal, count(1) as total_medals
                    From olympics_history oh
                    join olympics_history_noc_regions nr on
                    nr.noc = oh.noc
                    where medal <> ''NA''
                    group by  games, nr.region, medal
```

```
                                order by  games, nr.region, medal',
                                'values (''Bronze''), (''Gold''), (''Silver'')')
                            as result(games_country VARCHAR, bronze bigint,
                            gold bigint, silver bigint)
        order by games_country)
Select distinct games
, concat(first_value(country) over(partition by games order by gold desc)
                , ' - '
                , first_value(gold) over(partition by games order by gold desc))
                as max_gold
, concat(first_value(country) over(partition by games order by silver desc)
                , ' - '
                , first_value(silver) over(partition by games order by silver
                desc)) as max_silver
, concat(first_value(country) over(partition by games order by bronze
desc)
                , ' - '
                , first_value(bronze) over(partition by games order by bronze
                desc)) as max_bronze
from temp
order by games;
```

Results (up to 10)

	games text	max_gold text	max_silver text	max_bronze text
1	1896 Summer	Germany - 25	Greece - 18	Greece - 20
2	1900 Summer	UK - 59	France - 101	France - 82
3	1904 Summer	USA - 128	USA - 141	USA - 125
4	1906 Summer	Greece - 24	Greece - 48	Greece - 30
5	1908 Summer	UK - 147	UK - 131	UK - 90
6	1912 Summer	Sweden - 103	UK - 64	UK - 59
7	1920 Summer	USA - 111	France - 71	Belgium - 66
8	1924 Summer	USA - 97	France - 51	USA - 49
9	1924 Winter	UK - 16	USA - 10	UK - 11
10	1928 Summer	USA - 47	Netherlands - 29	Germany - 41