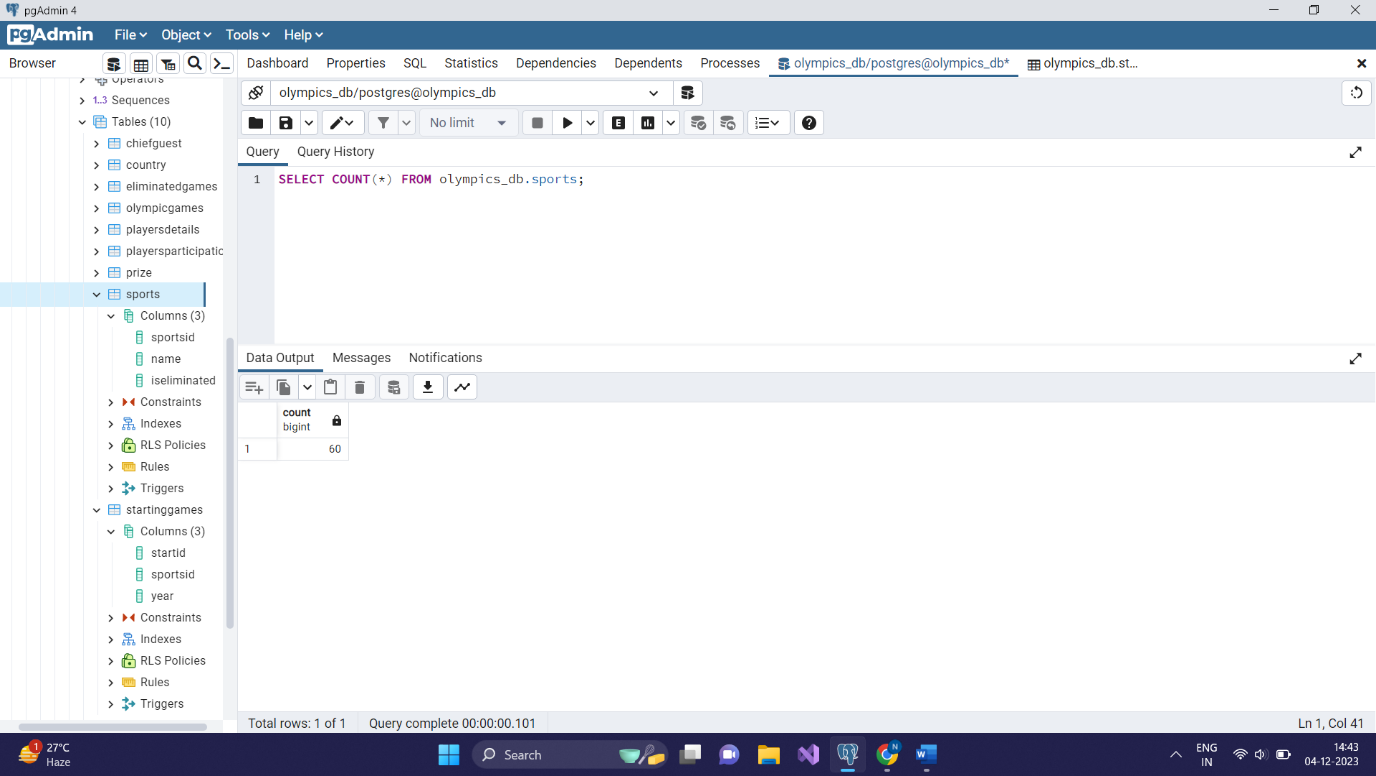
**1. Select the total number of sports:**

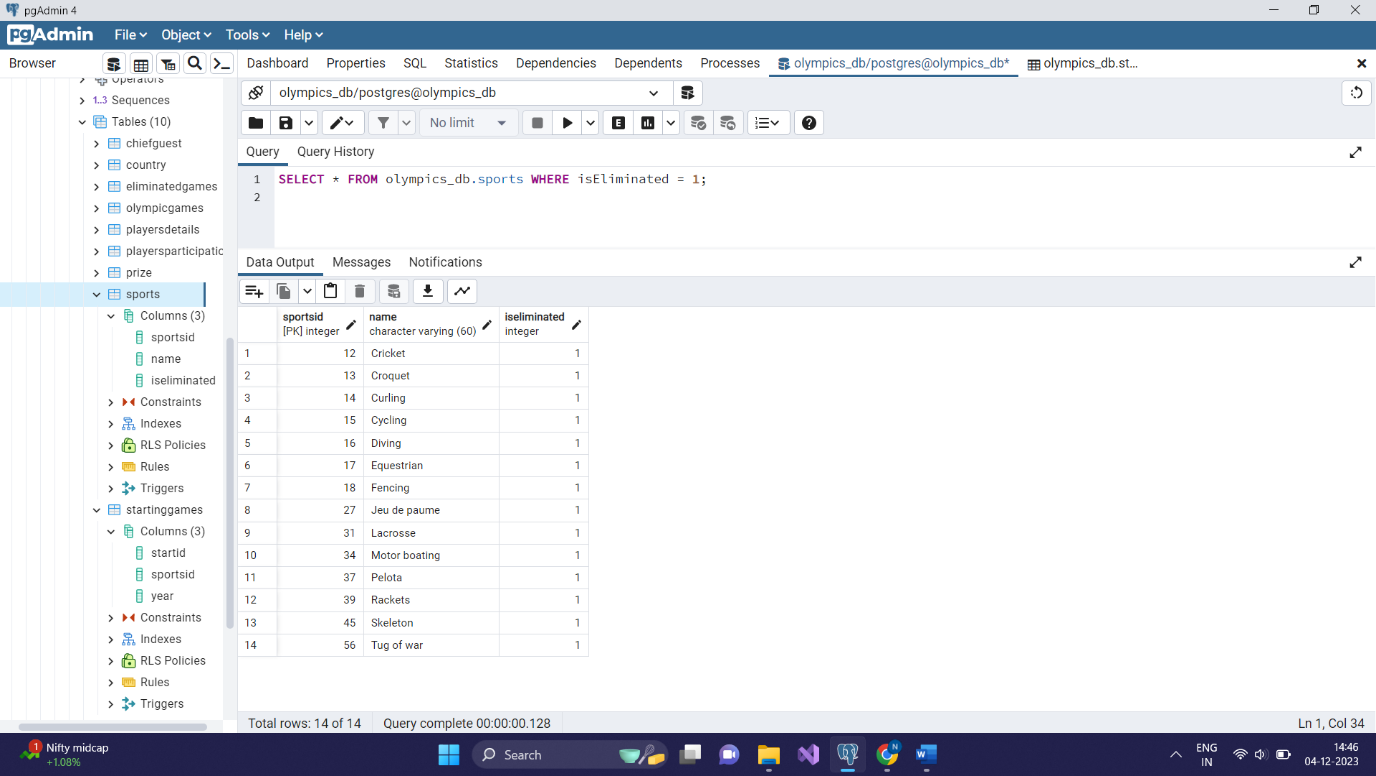
* SELECT COUNT(\*) FROM olympics\_db.sports;
* π\_COUNT(\*) (σ\_sports(olympics\_db))



* Number of tuples-1

**2. Select the sports that are eliminated:**

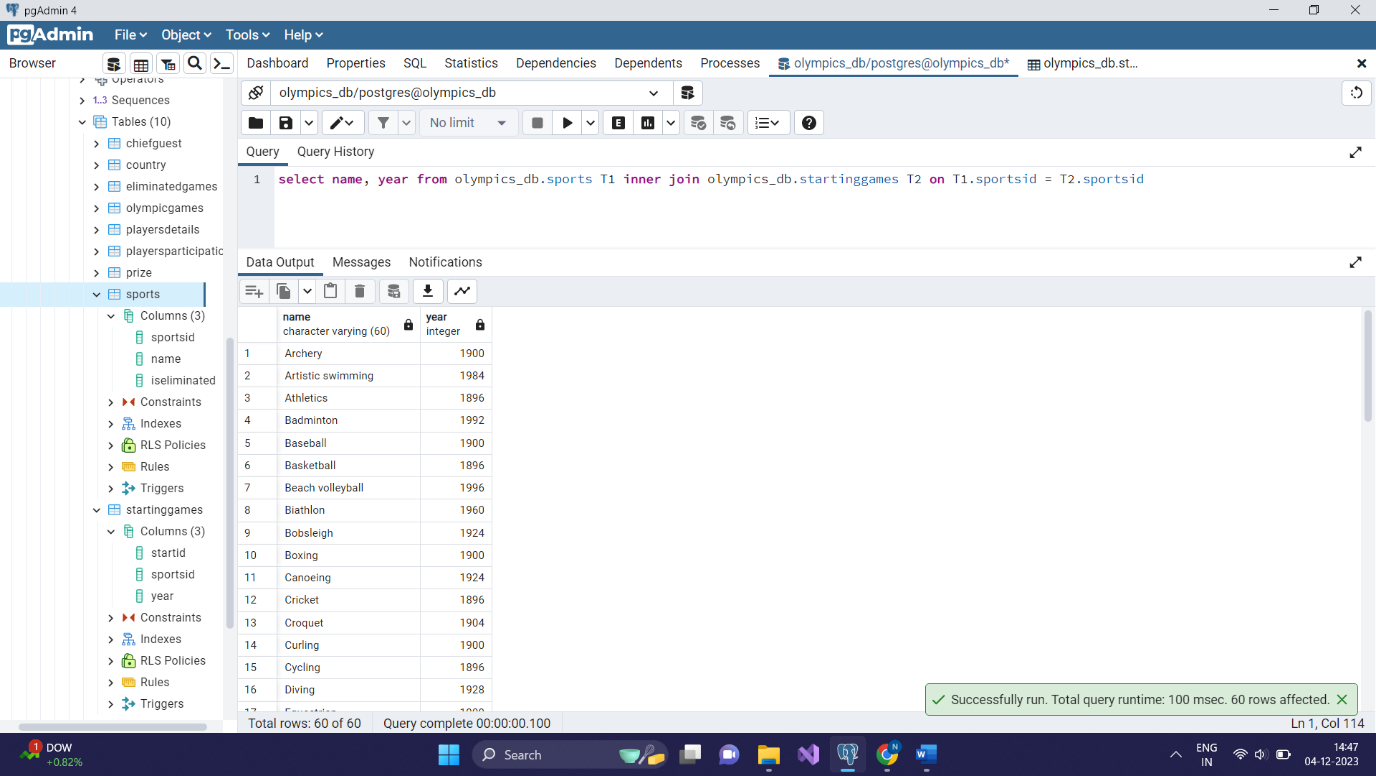
* SELECT \* FROM olympics\_db.sports WHERE isEliminated = 1;
* Πsport\_id,name,iseliminated(σ\_isEliminated=1(sports))



* Number of tuples-14

**3. Find the starting games for a specific sport:**

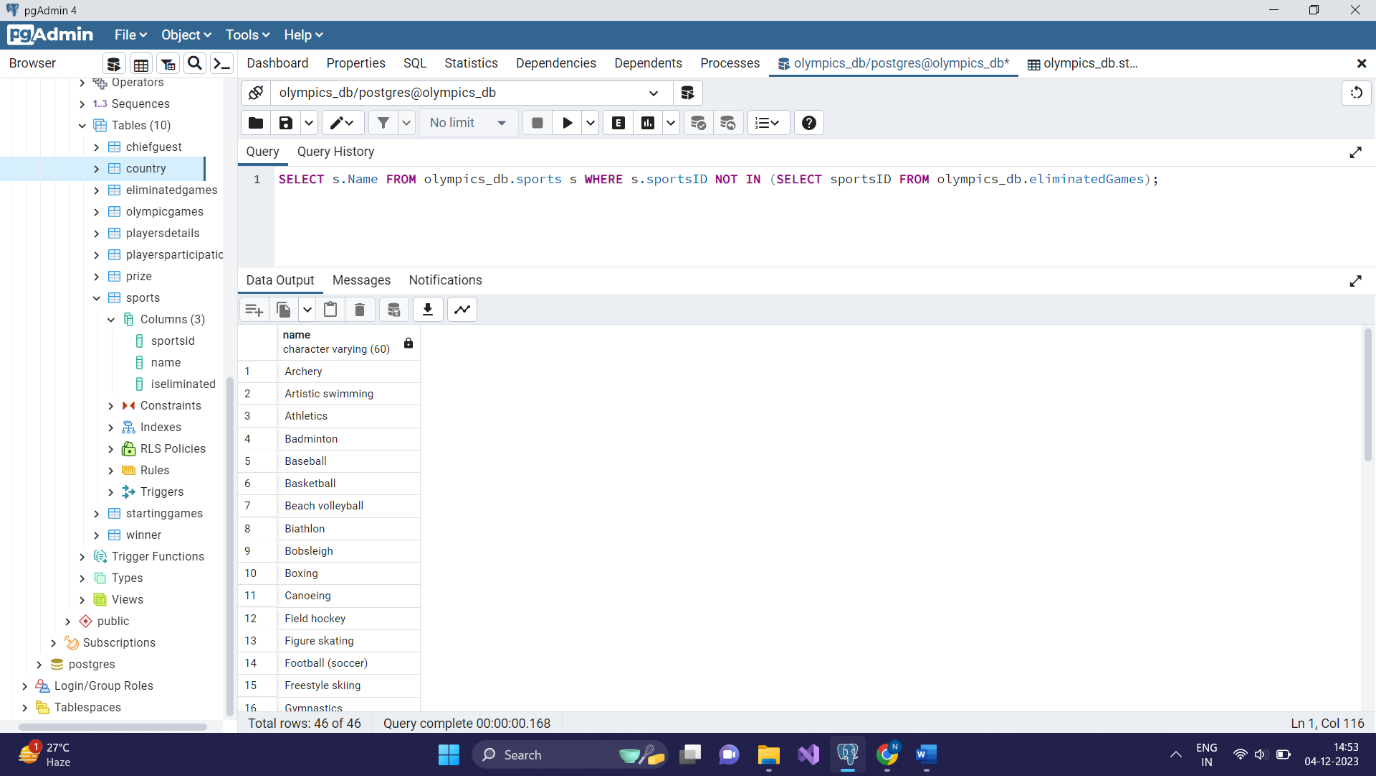
* select name, year from olympics\_db.sports T1 inner join olympics\_db.startinggames T2 on T1.sportsid = T2.sportsid
* π\_name,year(σ\_sportsid(T1 ⋈ T2))



* Number of tuples-60

**4. Select the sports that have never been eliminated:**

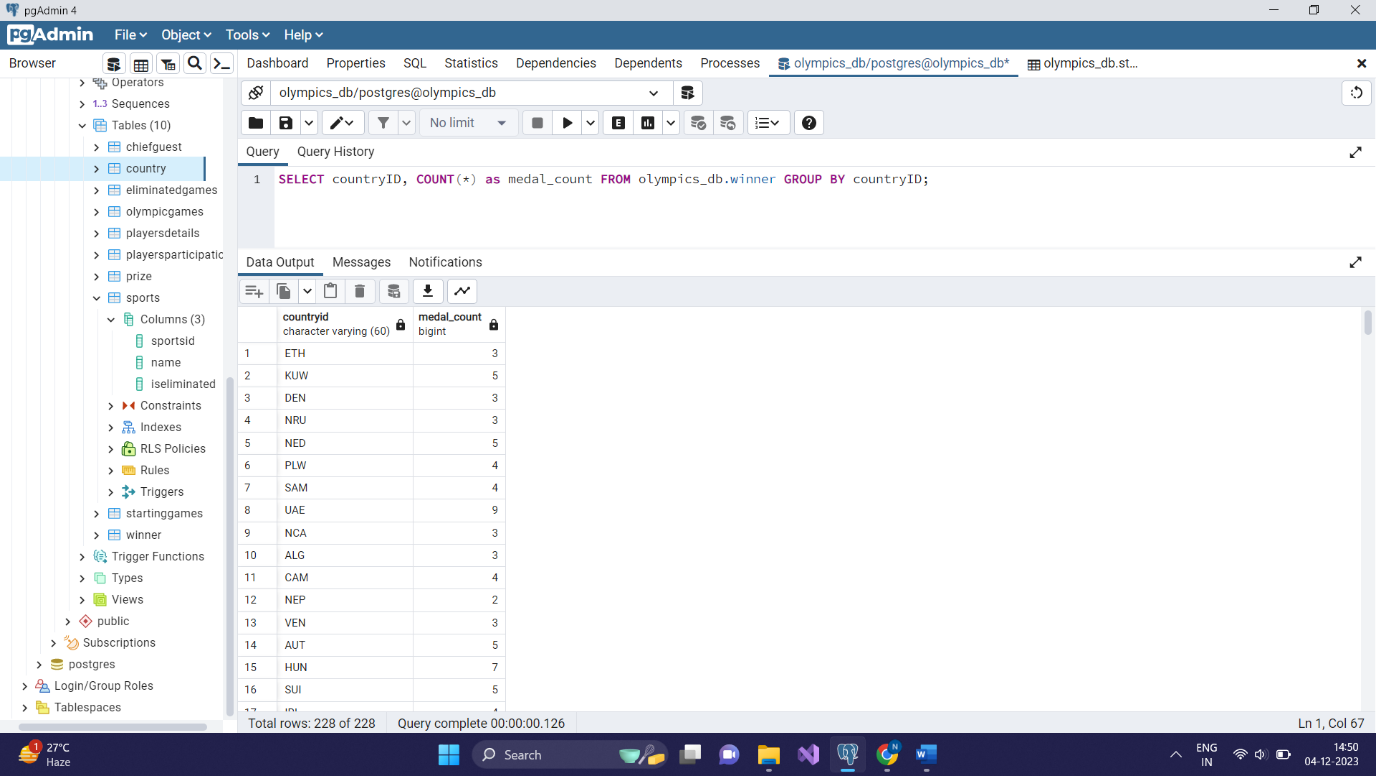
* SELECT s.Name FROM olympics\_db.sports s WHERE s.sportsID NOT IN (SELECT sportsID FROM olympics\_db.eliminatedGames);
* π\_Name(σ\_sportsID∉(eliminatedGames.sportsID)(sports))



* Number of tuples-46

**5. Get the total number of medals won by each country**:

* SELECT countryID, COUNT(\*) as medal\_count FROM olympics\_db.winner GROUP BY countryID;
* π\_countryID,COUNT(\*)(σ\_winner(olympics\_db))



* Number of tuples-228

**6.Select the names of players who won a gold medal:**

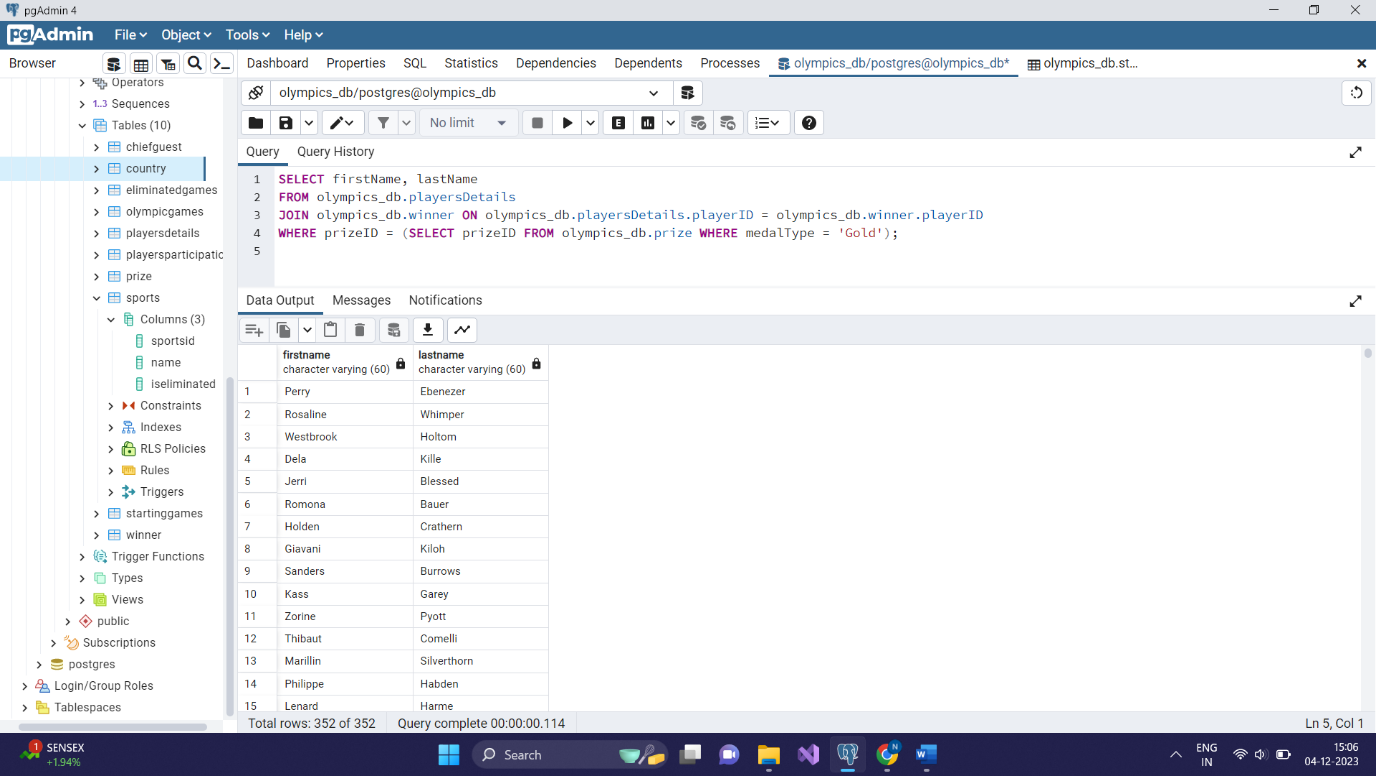
* SELECT firstName, lastName

FROM olympics\_db.playersDetails

JOIN olympics\_db.winner ON olympics\_db.playersDetails.playerID = olympics\_db.winner.playerID

WHERE prizeID = (SELECT prizeID FROM olympics\_db.prize WHERE medalType = 'Gold');

* π firstName, lastName ( σ prizeID = ( π prizeID ( σ medalType = 'Gold' (prize) ) ) ( playersDetails ⋈ winner ON playersDetails.playerID = winner.playerID ) )



* Number of tuples-352

**7. Calculate the total prize amount won by players between 1 to 100:**

* SELECT pd.firstName, pd.lastName, SUM(pr.amount) AS total\_amount

FROM olympics\_db.playersDetails pd

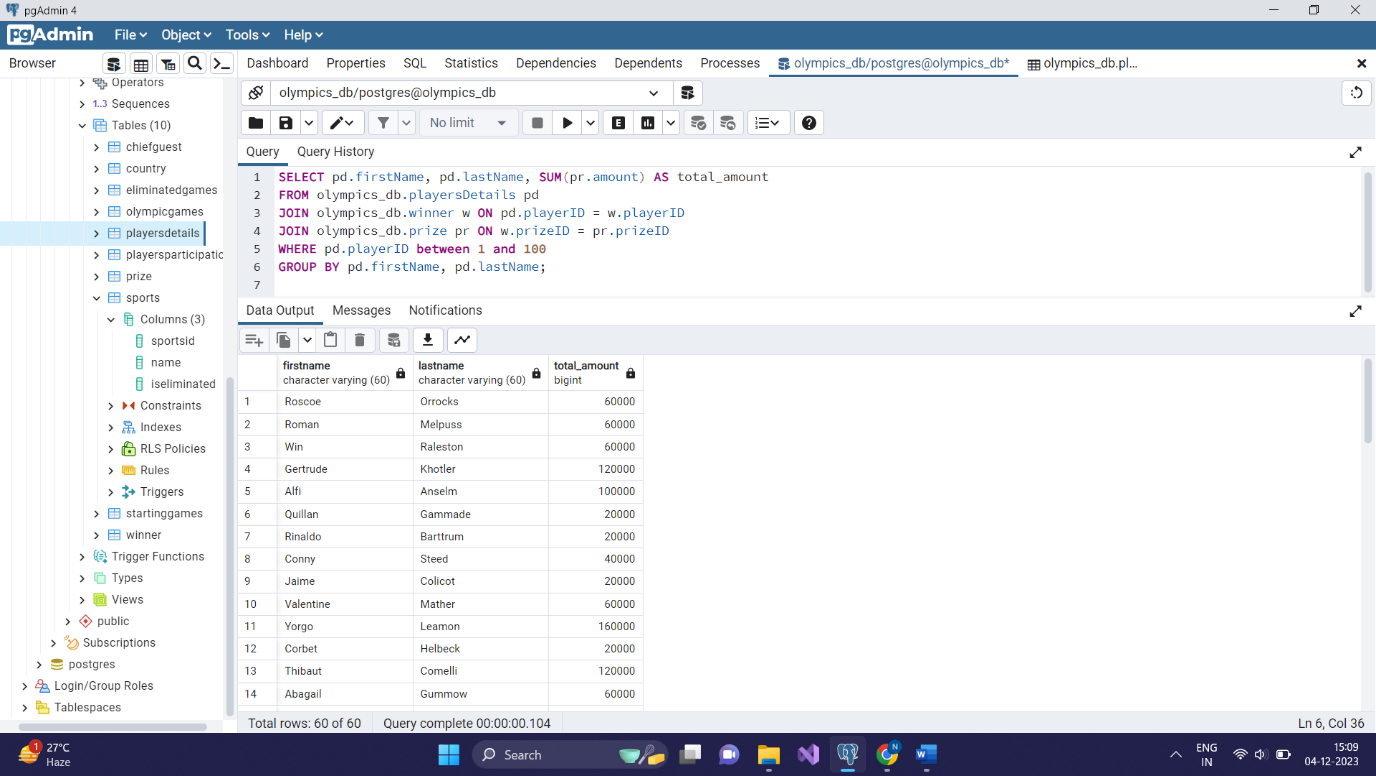
JOIN olympics\_db.winner w ON pd.playerID = w.playerID

JOIN olympics\_db.prize pr ON w.prizeID = pr.prizeID

WHERE pd.playerID between 1 and 100

GROUP BY pd.firstName, pd.lastName;

* πfirstName, lastName, total\_amount​(γpd.firstName, pd.lastName, SUM(pr.amount) AS total\_amount​(σpd.playerID≥1∧pd.playerID≤100​(ρplayerID→w.playerID​(playersDetails⋈pd.playerID=w.playerID​winner)⋈prize)))



* Number of tuples-60

**8. Select the players who won a medal in every Olympic game they participated in:**

* SELECT pd.firstName, pd.lastName

FROM olympics\_db.playersDetails pd

WHERE pd.playerID NOT IN (

SELECT pd.playerID

FROM olympics\_db.playersDetails pd

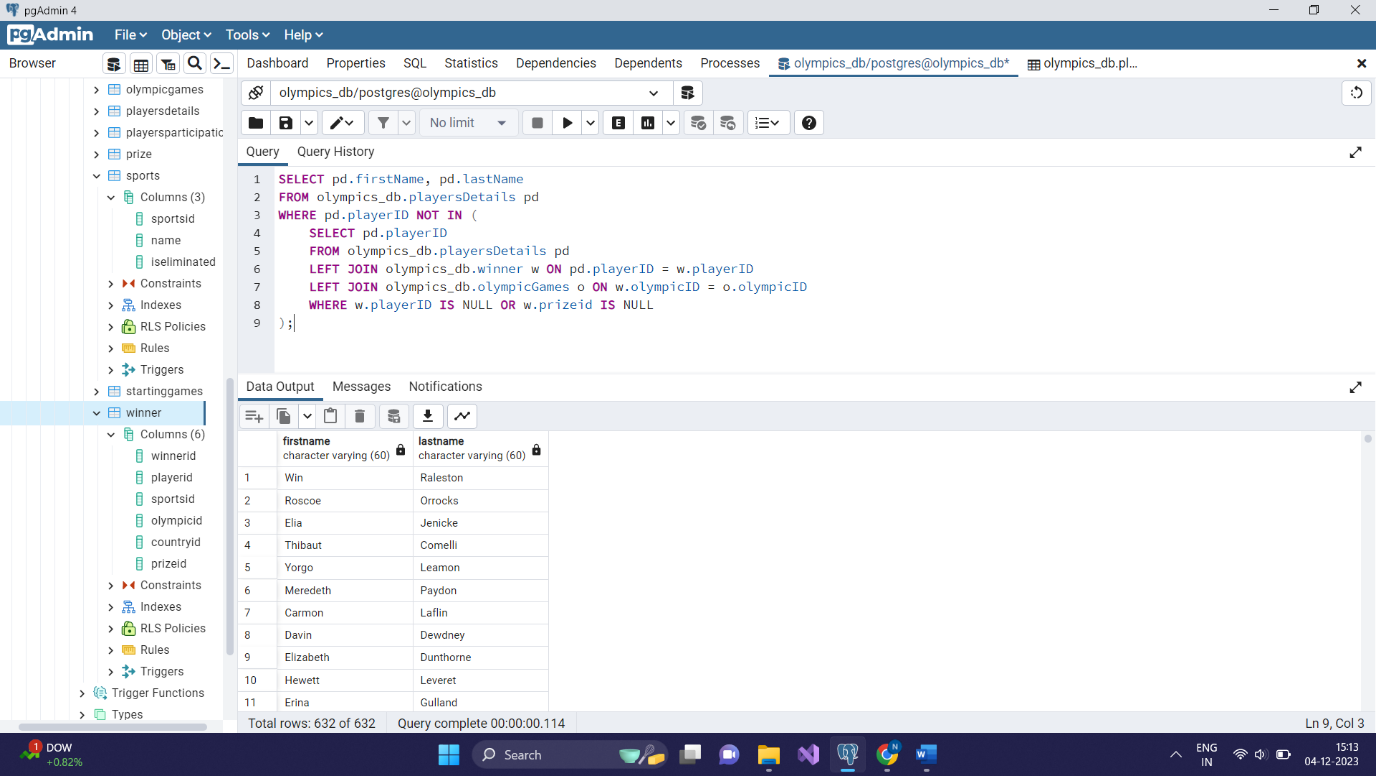
LEFT JOIN olympics\_db.winner w ON pd.playerID = w.playerID

LEFT JOIN olympics\_db.olympicGames o ON w.olympicID = o.olympicID

WHERE w.playerID IS NULL OR w.prizeid IS NULL

);

* πfirstName, lastName​(playersDetails−πpd.playerID​(σw.playerID IS NULL OR w.prizeid IS NULL​(ρplayerID→w.playerID​(playersDetails⋈pd.playerID=w.playerID​winner⋈olympicGames))))



* Number of tuples-632

**9. Select the sports with the highest and lowest number of participants in a specific Olympic game:**

* (SELECT s.Name, COUNT(pp.playerID) AS participant\_count

FROM olympics\_db.sports s

JOIN olympics\_db.playersParticipation pp ON s.sportsID = pp.sportsID

WHERE pp.olympicID = 2

GROUP BY s.Name

ORDER BY participant\_count DESC

LIMIT 1)

UNION

(SELECT s.Name, COUNT(pp.playerID) AS participant\_count

FROM olympics\_db.sports s

JOIN olympics\_db.playersParticipation pp ON s.sportsID = pp.sportsID

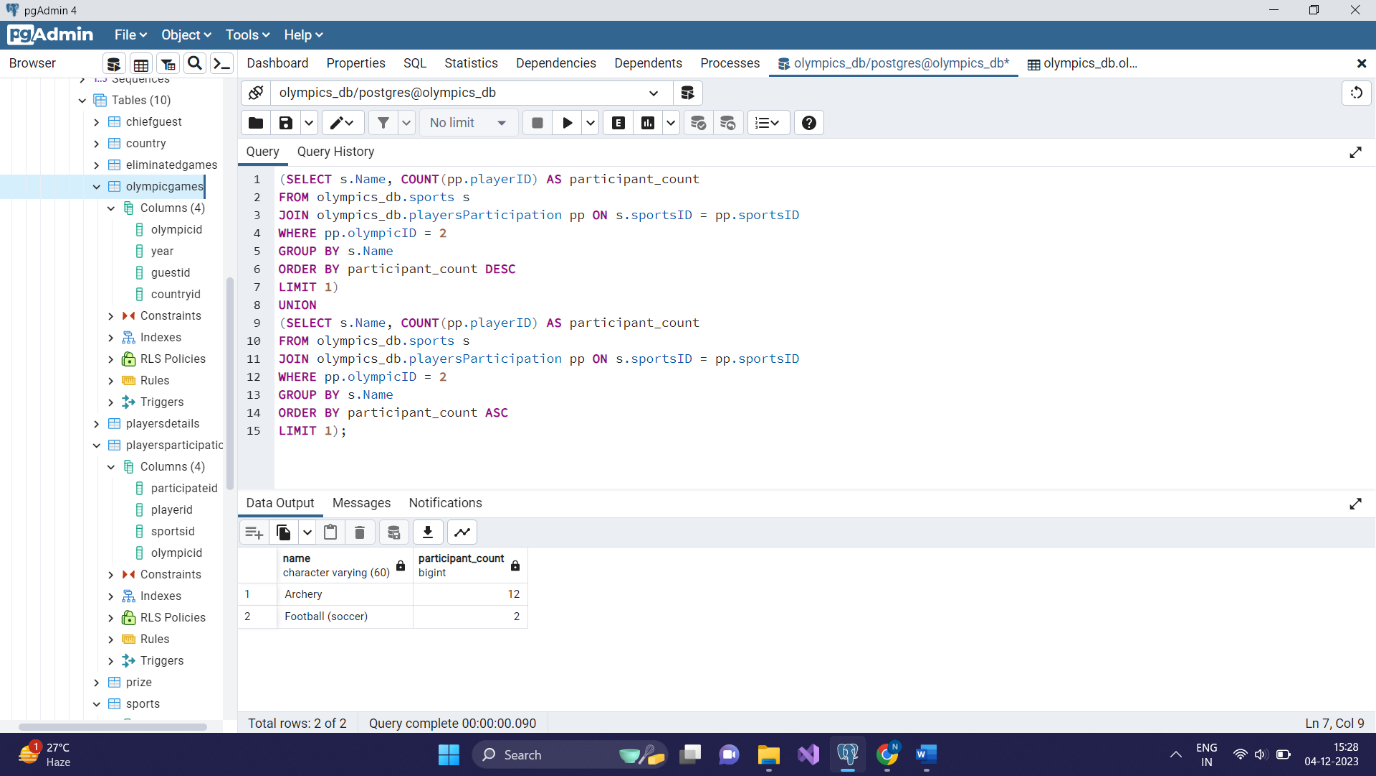
WHERE pp.olympicID = 2

GROUP BY s.Name

ORDER BY participant\_count ASC

LIMIT 1);

* πName, participant\_count​(σpp.olympicID=2​(sports⋈s.sportsID=pp.sportsID​playersParticipation)⋈s1.participant\_count = s2.participant\_count​(ρs.Name, COUNT(pp.playerID) AS participant\_count​(γs.Name, COUNT(pp.playerID) AS participant\_count​(sports⋈s.sportsID=pp.sportsID​playersParticipation))))



* Number of tuples-2

**10. Select the countries where the average age of players is above the global average**:

* SELECT co.countryName

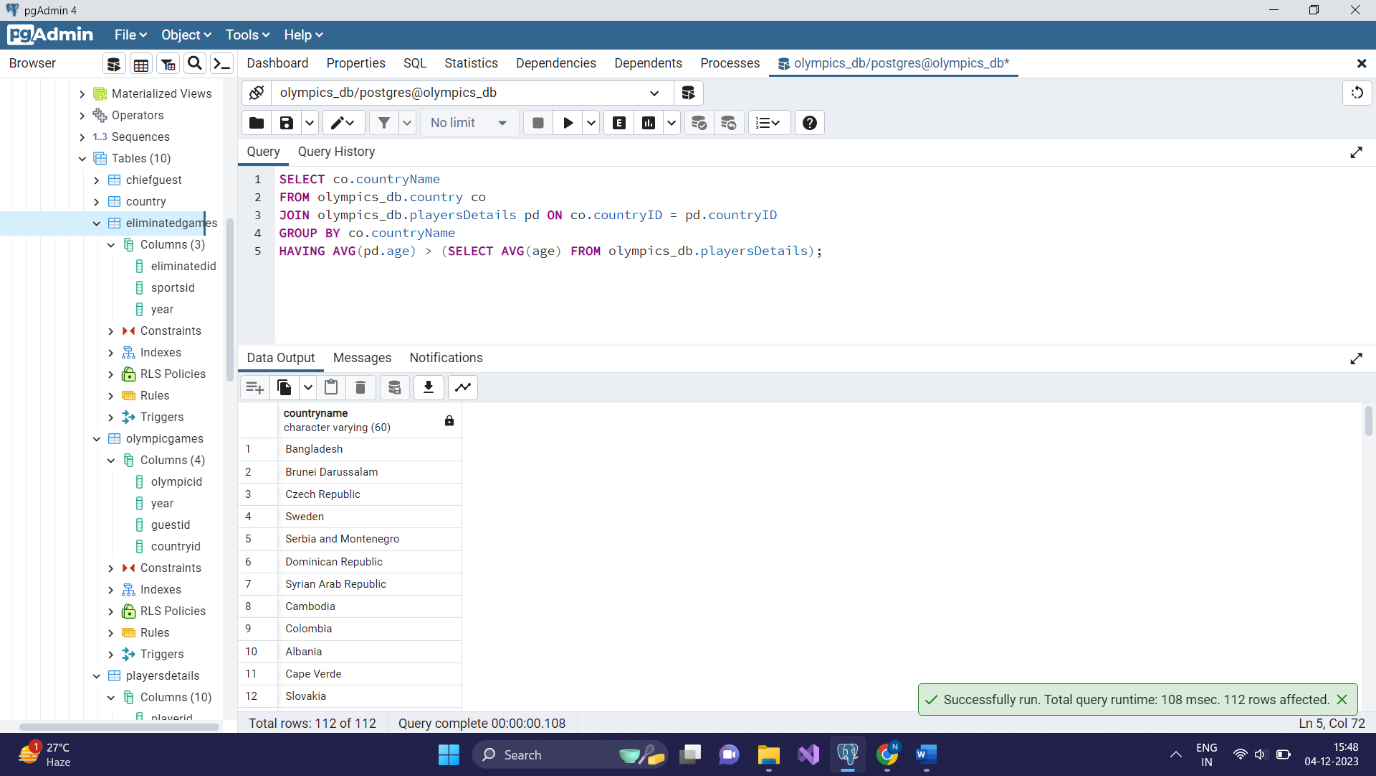
FROM olympics\_db.country co

JOIN olympics\_db.playersDetails pd ON co.countryID = pd.countryID

GROUP BY co.countryName

HAVING AVG(pd.age) > (SELECT AVG(age) FROM olympics\_db.playersDetails);

* πcountryName​(σAVG(pd.age) > AVG(age)​(γco.countryName​(country⋈co.countryID=pd.countryID​playersDetails)⋈γAVG(age)​(playersDetails)))



* Number of tuples-112