**PostgreSQL(OLAP)**

**Views:**

Create public.Avgyear

AS select event\_id, fiscal\_year, total\_amount from public.ankur

group by fiscal\_year,event\_id,total\_amount;

**Create view** public.city as select donor\_city,donor\_state,donor\_county, (gift\_amount+additional\_gift\_amount) as sum,(gift\_amount::float+additional\_gift\_amount::float)/2 as Average

from public.donation;

**Create view** gender as select donor\_gender,(gift\_amount+additional\_gift\_amount) as sum,(gift\_amount::float+additional\_gift\_amount::float)/2 as Average, fiscal\_year

from public.donation

**Q1.** **What is the total amount of givings every team(is prior participants)did per year?**

select sum(gift\_amount), is\_prior\_participant, fiscal\_year

from Donation

Group by is\_prior\_participant, Fiscal\_year, gift\_amount;

**Q2. What are the cities and respective states where is done the majority of givings?**

select max(sum),donor\_state,donor\_city from public.city

group by donor\_state,donor\_city

**Q3**.**What is the average amount of donations for a particular event in a given year?**

select \*,avg(total\_amount) from public.avgyear group by event\_id,fiscal\_year,total\_amount

limit 500

**Q4.What is the max amount of givings and max average of givings based on donor’s gender ?**

select max(sum), max(average),donor\_gender from public.gender

group by donor\_gender

limit 1000;

**Q5.What is the max amount of givings and max average of givings based on donor’s gender per each year?**

select max(sum), max(average),donor\_gender,fiscal\_year from public.gender

group by fiscal\_year,donor\_gender

limit 1000

**Q6.What is the total amount of givings every team did based on their name and number of participants?**

Select a.team\_name,a.number\_of\_participants,b.total\_amount

from public.bike\_teams a left outer join public.ankur b on a.event\_id=b.event\_id

limit 1000

**Queries referring to specific OLAP extentions of PostgreSQL for windows and window functions**

**-Computing rankings and partitioning**

Select p.participant\_connection\_to\_ms, a.total\_amount, dense\_rank() over( Partition by a.fiscal\_year)

from public.participants p left join public.ankur a on p.event\_id=a.event\_id

where a.fiscal\_year=2013

**-Computing cumulative totals ( window framing)**

Select donor\_state,fiscal\_year, sum(net\_transaction\_amount)

OVER(ORDER BY donor\_state RANGE BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW)

from public.donation

group by donor\_state, net\_transaction\_amount,fiscal\_year

**-Computing mobile aggregates [window framing]**

select event\_id,fiscal\_year,

sum(net\_transaction\_amount),avg(net\_transaction\_amount::float)

OVER(Partition by event\_id order by fiscal\_year rows 1 preceding )

from public.ankur

group by event\_id,net\_transaction\_amount,fiscal\_year

order by event\_id

limit 10000