Example experiment:

**Flights dataset**

A : *year, month, week, day, carrier, origin, destination*

M : *arrivaldelay, departuredelay, weatherdelay, distance*

F : *avg, sum, max, min*

Total possible views: *112 views*

**Naïve Method (one by one query view execution):**

Q: select year, avg(arrivaldelay) from flights where carrier='AA' group by year

R: select year, avg(arrivaldelay) from flights group by year

Total number of query executions: *2\*112 query executions*

**Query sharing Method:**

Q: select **year**, avg(arrivaldelay), avg(departuredelay), avg(weatherdelay), avg(distance),

sum(arrivaldelay), sum(departuredelay), sum(weatherdelay), sum(distance)

max(arrivaldelay), max(departuredelay), max(weatherdelay), max(distance),

min(arrivaldelay), min(departuredelay), min(weatherdelay), min(distance)

from flights where carrier='AA' **group by year**

R: select **year**, avg(arrivaldelay), avg(departuredelay), avg(weatherdelay), avg(distance),

sum(arrivaldelay), sum(departuredelay), sum(weatherdelay), sum(distance)

max(arrivaldelay), max(departuredelay), max(weatherdelay), max(distance),

min(arrivaldelay), min(departuredelay), min(weatherdelay), min(distance)

from flights **group by year**

Total number of query executions: *2\*7 query executions*

SeeDB works as follows:

During each phase, SeeDB keeps an estimate of the mean utility for every aggregate view *Vi* and a confidence interval around that mean.

At the end of a phase, SeeDB uses the following rule to prune low-utility views: If the upper bound of the utility of view *Vi* is less than the lower bound of the utility of k or more views, then *Vi* is discarded