# Discussion: BlinkDB

# BlinkDB: My main takeaways

- First contribution: Organize sampling around query column sets
  - (a) A small set may “cover” all queries
    - Drawn from workloads
  - (b) MILP formulation which picks these column sets
- Second contribution: Determine on the fly, the qcs that gives the best “bang for the buck”
  - Neat idea of selectivity – number of rows selected divided by the number of rows read

# Where could the BlinkDB approach fail?

# Where could the BlinkDB approach fail?

- QCSes are not stable
- # of rare subgroups are high, dimensionality bad
  - For example, if three groups have dimensionality 10000 each, the stratified sample of the cross-product could be GIGANTIC
  - Other approaches would also fail

# What are the drawbacks of the BlinkDB system?

Let’s talk about

- Optimization Techniques
- Query Class
- Repeated Queries

# What are the drawbacks of the BlinkDB system: Optimization Techniques?

# What are the drawbacks of the BlinkDB system: Optimization Techniques?

- Not clear
  - how to tune various parameters: K, M
  - whether the MILP is close to optimal
  - if techniques will apply to other workloads or case studies (two datasets is limited)

# What are the drawbacks of the BlinkDB system: Query Classes?

# What are the drawbacks of the BlinkDB system: Query Classes?

- Does not handle joins/nesting
  - Simple queries

# Other Drawbacks

- A QCS either receives K or none at all. Is this wise?
  - Should depend on variance, distribution
    - If variance is small, a smaller sample may suffice
- Paper claims partial covering is OK if need be. Is that true?
  - Partial covering may lead to a biased sample
  - May completely miss some groups (incomplete answers)

# Aqua vs. BlinkDB

- Very similar ideas for offline precomputed samples
- Aqua
  - Is query agnostic, will take full collection of group-by columns to construct stratified sample
    - May be too much
  - Broader class of queries
    - General enough to apply to joins (foreign key)

# When would you use BlinkDB vs. Materialized Views?