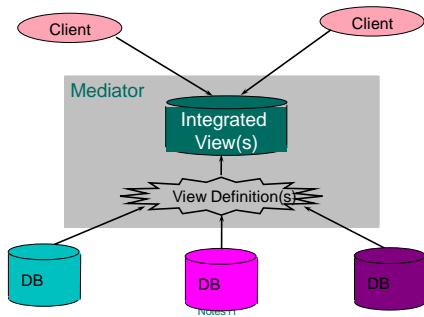
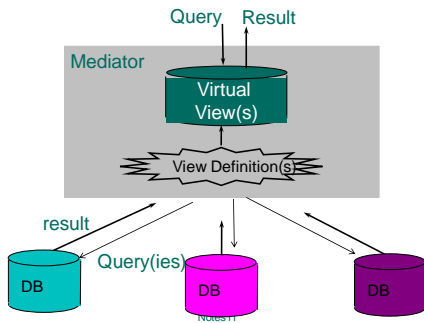


Virtual View -> Mediator needs Distributed Query Processor
 Materialized view (warehouse) -> Mediator also needs storage
 & Incremental View Maintenance



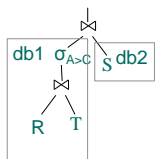
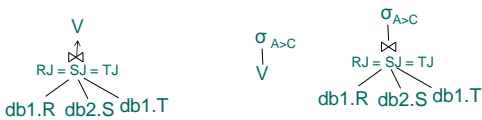
4

Distributed Query Processing in Mediators



5

Mediators as providers of view-based virtual views over distributed data



Translate subplans to SQL
 + How will the join happen?
 - What if the source offers multiple data services instead of JDBC access? (besides CSE232 material)

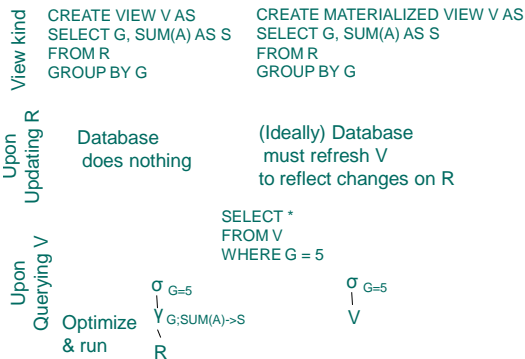
6

Join types

- Mediator-based Join
 - ◆ Ship results of queries at mediator
- Parameterized Join
 - ◆ Right subquery is enhanced with selection on join attribute
 - ◆ For each join value of left hand side, execute another right subquery
- Data Ship Join
 - ◆ Insert the result of left hand side (lhs) in the db of right hand side (rhs).
 - ◆ Execute join at db of right hand side
- Semijoin Reduction Join
 - ◆ Send rhs parameters to lhs
 - ◆ (Data ship alike variation) Lhs sends to rhs the semijoin of its subquery with the parameters set.
 - ◆ Execute join at db of rhs
 - ◆ Also, variation that looks like mediator-based join

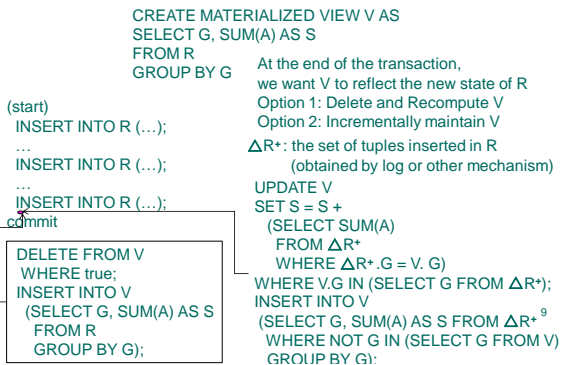
7

Virtual Views Vs Materialized Views

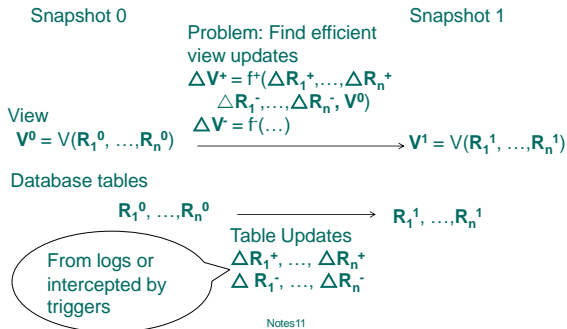


8

Recompute Vs Incremental (Materialized View) Maintenance – Informal Example



IVM: Self-maintaining version (not always possible)



Basic IVM Algorithm: Compose operator IVM rules

Example (wlog deferred, i.e., R means R^1 and S means S^1)

- Rule for $V = R \bowtie S$
 - ◆ $\Delta V^+ = ((\Delta R^+ \bowtie S) \cup (R \bowtie \Delta S^+)) - (\Delta R^+ \bowtie \Delta S^+)$
 - ◆ $\Delta V^- = ???$
- Rule for $V = \sigma_c R$
 - ◆ $\Delta V^+ = \sigma_c \Delta R^+$
 - ◆ $\Delta V^- = ???$
- Composition of rules leads to solutions for

$$V = T \bowtie \sigma_{A>5} W$$

$\Delta V^+ = \dots$

$\Delta V^- = \dots$

- May rewrite initial expression

14

IVM with Caching

- May associate intermediate views (caches) with subexpressions
- Bottom-up: From updating caches to reaching the materialized view
- Caches will typically need indices
- Caches may or may not pay off as they incur cost for maintaining them (and their indices)

15

Generalizations

- Multiple views
 - ◆ self maintenance may involve a view utilizing the other views in its computation
- Genuine updates
 - ◆ Not simulated via insertions/deletions
- Insertions, deletions, updates on tables and views expressed as DML statements

16

Comparisons

Materialized View

- High query performance
- Queries not visible outside warehouse
- Local processing at sources unaffected
- Can operate when sources unavailable
- Extra information at warehouse
 - ◆ Modify, summarize (store aggregates)
 - ◆ Add historical information

Virtual View

- No need for yet another database
 - More up-to-date data
 - ◆ Depending on specifics of IVM
 - Query needs can be unknown
 - Only query interface needed at sources
- => Lower Total Cost of Ownership

CSE232A

Notes11

17
