

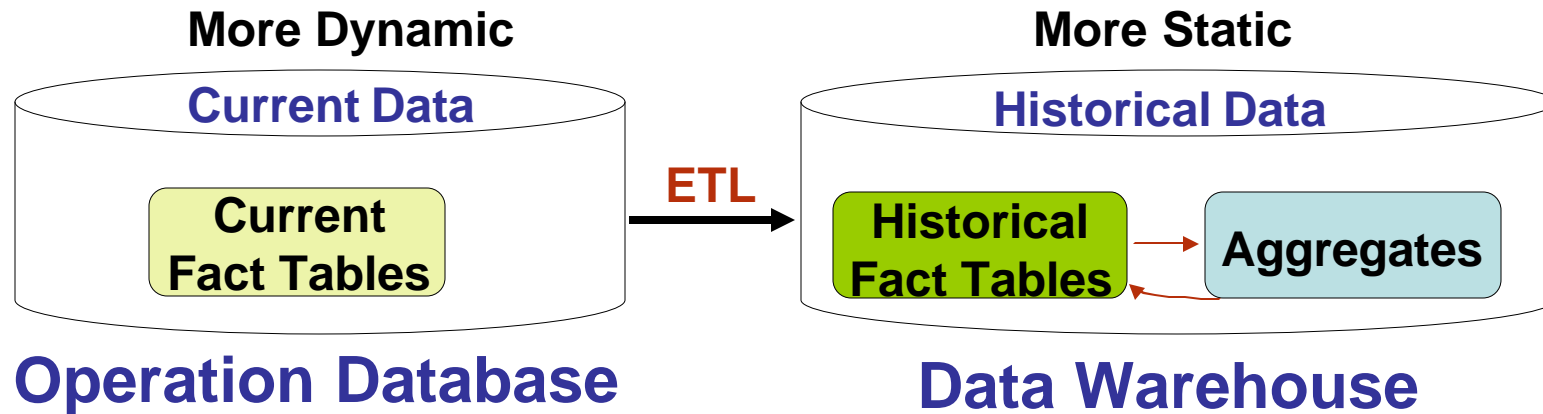
Real-Time Data Warehouse Management

Qiming Chen

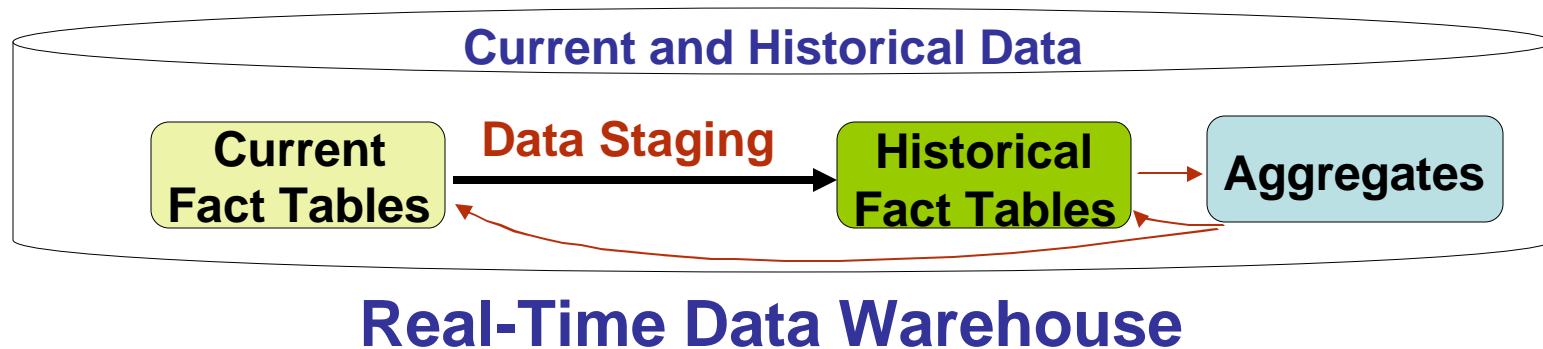
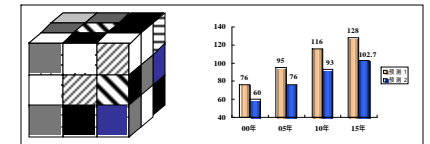
HP Labs

09/2006

Real-Time Data Warehouse



Call Detail Records
Network Traffic Logs



Why Real-Time DW

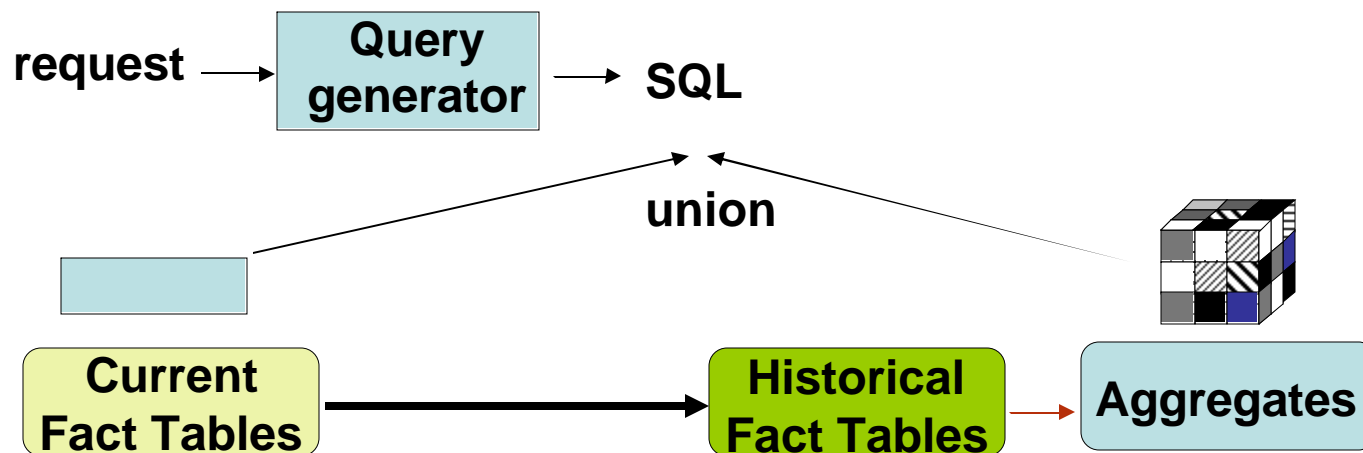
- Support “up-to-now” data measurement and drill-down analysis
- Handle event, alert
- Provide a single view to enterprise including something just happened
- Reduce dimension data duplication by keeping all fact data in one DB
- Examples: Network traffic data, CDR

Challenges

- **Query together historical data (summarized) and current data (detailed)**
- **Mix-workload for Read/Write**
 - Real-time data loading and reporting
- **Data staging without data movement**
 - Avoid `select ... into ...;delete ...from...;`
- **Need DWMS above DBMS to deal with common functionalities**

Query Generation Facility

- Query “summary of this week **up to now**”
- Result by the union of pre-aggregated and live-aggregated reports **running in parallel**
- SQL statement is automatically generated



Mix-Workload for R/W

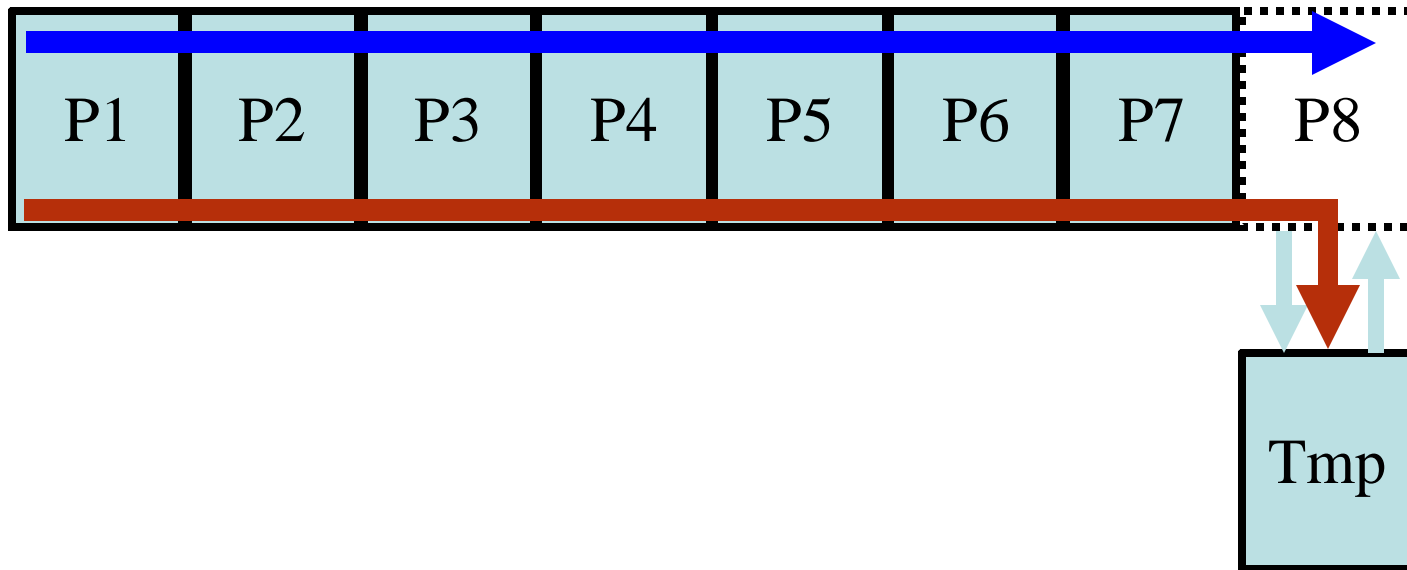
- Due to conflict between data loading and reporting (W/R fact table), a report (e.g. aggregate) covering t1-t2 may not be made in time
- Queries expect to read that portion of report
- A mix-workload solution
 - Write has higher priority than read
 - Then a periodically operated report query may not finish in-time, resulting the unavailability of the portion of report
 - In worst case, report generation should be halted and automatically catch-up in next report generation cycle
 - Query to a report (e.g. aggregate) must check the availability of each requested portion
 - if a portion is unavailable, do live-aggregation for it; query result is the union of pre-aggregation and live-aggregation

Play with VLDB or VSMD (Very Small Meta Data)

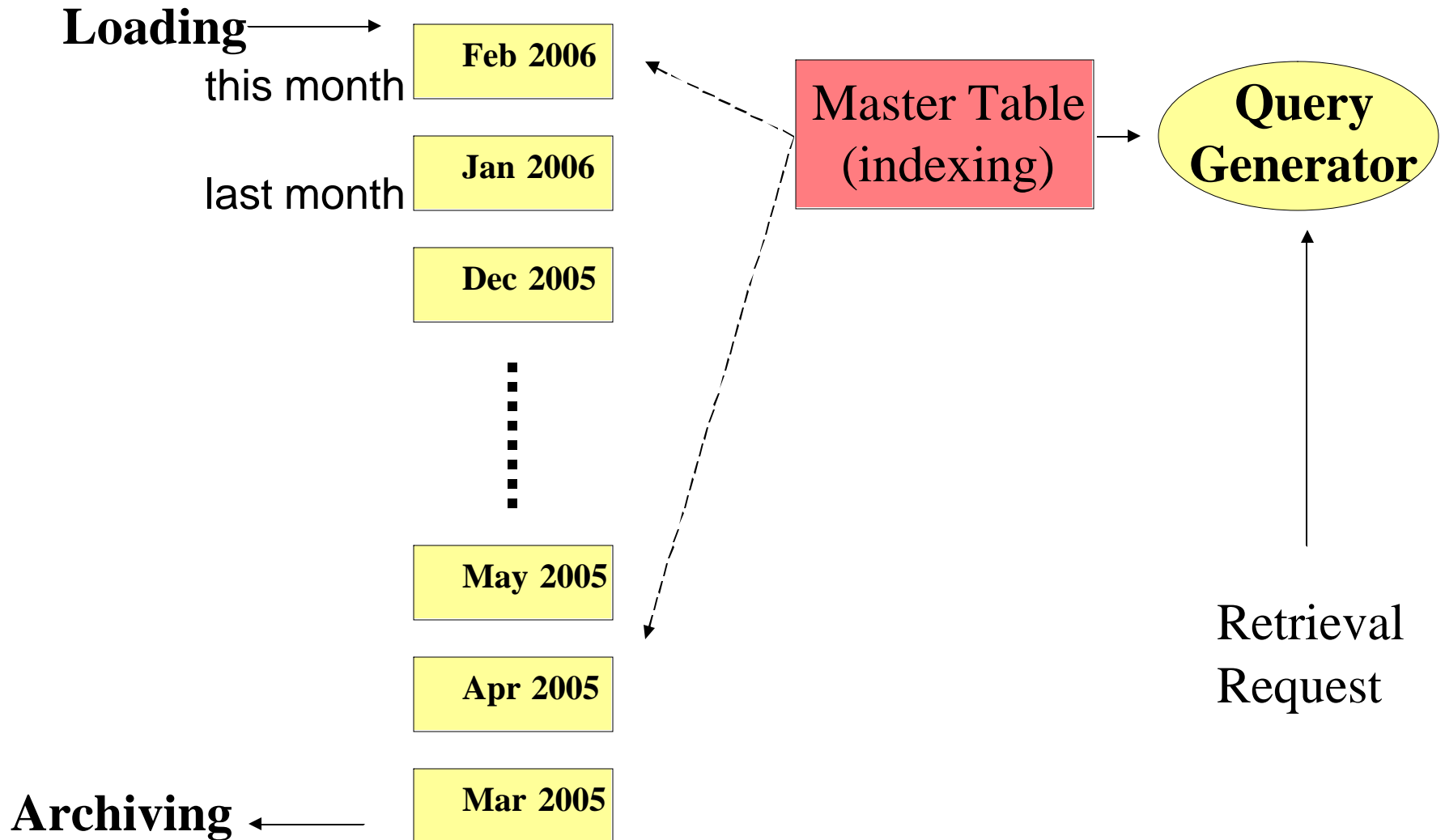
- Avoid one table to grow too big
- Avoid move data along staging
- Avoid update big table
- Meta data manipulation approaches
 - Change data object “tag” in Data Dictionary
 - Turn current fact table into a partition of historical fact table (Oracle EXCHANGE_PARTITION)
 - Treat historical fact table as union of multiple (once-current) fact tables
 - Round-Rabin management for data staging and archiving

Exchange Partition

- EXCHANGE PARTITION can be transparent to in-flight queries
 - Avoid updating big table
- Similar mechanisms of changing semantics but keep content



Staging by Round-Robin Table Switch



What is the Indication ?

- Need to build DWMS on top of DBMS to handle and automate common functionalities
- Domain-specific query processing not supported by DBMS
 - SQL has the expression power, but ...
- Scale to infinity require resemblance of database notions ...
 - Do you want to keep one table with unlimited size
 - ACID ?
 - Do you want hierarchical management (e.g. multi-DB portal)
 - Move data?