Alert: An Architecture for Transforming a Passive DBMS into an Active DBMS

Ulf Schreier, Hamid Pirasesh, Rakesh Agrawal, C. Mohan

IBM Almaden Research Center
Alert Idea

- Does not build an Active DMBS from scratch
- **Extends** a Passive DBMS into an Active DMBS

**Examples:**
- Active queries are written in SQL with minimal additions.
- Indexing and Query optimization are used for event detection
Figure 1: Architecture
Passive Tables / Cursors

- Cursor = bookmark for queries
- Query calls fetch which moves the cursor
- Loop until end of table
- No more results after table’s end
Active Cursors / Tables

- Active table = append only table
- Queries call fetch-**wait**
- Fetch-wait goes to sleep at end of table
- Awoken on update
- New results
Sample Active Query

Declare C cursor for
  SELECT name, email FROM classlist
Open C;
while(TRUE)
  { fetch_wait C into:name,,email
    ;; send welcome to class email
  }

Alert Rules

- Rule = named active query
- Conditions in FROM and WHERE clause.
- Ex: Create rule temp_watch as
  
  ```sql
  SELECT alarm('EVACUATE!')
  FROM temps
  WHERE location = 'Nuclear Reactor'
  AND temp > 500
  ```
Rule = SQL View

Create rule exam as
SELECT email("registrar")
FROM rooms
WHERE actual_capacity > max_capacity/2;
Create rule fire-code-violation as
SELECT phone("Fire Department")
FROM exam
WHERE actual_capacity > max_capacity;
Rule Activation

Rules can be activated with consideration for three categories:

- Transaction coupling = Same / Separate
- Time coupling = Synchronous / Asynchronous
- Assertion mode = Immediate / Deferred
Transaction Coupling

**Same Coupling**
- Triggered transaction is part of triggering transaction
- Ex: An integrity check should run as part of an insertion to correct it if necessary.

**Separate Coupling**
- Triggered transaction runs separately from its triggering transaction
- Ex: A stock order should run separately from a query over stock prices.
Time Coupling

- **Synchronous**
  - Triggered transaction runs and triggering transaction waits for it to complete before running

- **Asynchronous**
  - Triggered transaction runs in parallel
Assertion Mode

- **Immediate**
  - The rule is triggered as soon as its condition is satisfied.

- **Deferred**
  - The rule is triggered only within certain parts of the triggering transactions.
  - Ex: A certain professor wants to allow students to work together during tests.
Restrictions

- Same must be synchronous
- Separate must be asynchronous
- Deferred can only be used in the same coupling mode.

Why all these options? It seems like there are 8 combinations, but really only 3!
Monitor System -- Locks

Ex: Rule that monitors bank accounts for accounts below the minimum balance

- Most transactions will not result in a transaction applying to this rule.

Regular locking scheme requires rule to wait for locks for useless data.

Only waits for locks when a tuple meets rule conditions.
Monitor System – Adding tuples

- When a tuple is added we would like to quickly know to which rule it is applied.
- Modified B+ tree that alerts corresponding rules when a tuple is added to corresponding index.
- Tuples have increasing Ids and therefore each tuple will be added to the end of buckets.
What to take away

- Active DBMS can be created by simply extending normal DBMS features.
- DSMS must be able to perform better than this extended type of DBMS.