CSE 480, Spring 2003

Database Systems

Views in Relational Datamodel

S. Pramanik
Views (Virtual Tables)

• View:
  – A named SQL statement or
  – A named relational algebraic expression.

• Relational Algebra:
  GoodStudent(Id, Name, GPA):
  \[ \pi_{Sid,Sname,gpa}\sigma_{gpa>3.0}(Student) \]

• SQL:

  CREATE VIEW
  GoodStudents(Id, Name, GPA)
  AS
  SELECT Sid, Sname, gpa
  FROM Student
  WHERE gpa>3.0
How It Works

- Queries on Views:
  - Algebra:
    \[ \pi_{Id} \sigma_{Name='John'}(GoodStudents) \]
  - SQL:
    Select Id
    From GoodStudent
    Where Name='John'

- View Definition Replaces References:
  - \[ \pi_{Sid} \sigma_{Name='John' \& gpa>3.0}(Student) \]
  - Select Sid
    From Student
    Where Name='John' and gpa>3.0
Why It Works

• Closure Properties:
  – Algebraic Operators.
  – Group By?

• Use of Closure properties:
  – Algebraic Expression ⇒ Virtual Relation
  – Substitute and Merge Algebraic Expressions (Names to Base Tables)
    – \( \pi Id \sigma Name = 'John'(GoodStudents) \Rightarrow \pi Sid \sigma Name = 'John' \& gpa > 3.0(Student) \)
Advantages of Views

1. Security of Hidden Data
2. Simplify Complex Operations
3. Logical Data Independence.
Updates on Views

• Updates Allowed When
  – Defined on a Single relation
  – Contains PK or a Candidate Key

• Problems of View semantics and inserts.
  (lab problem: Insert into EastLansing (Sid, Sname))
  Base table gets data through view but view does not show it.
  Read Only views

• Problems of Unique Mapping
  Key preserving tables (See Oracle)

• Queries; Updates Nonprimary
Materialized Views

• Physical Copy of Data
• Data in Sync With Base Tables
• Achieved by Triggers
• Better Query Performance
• Oracle Features:
• Data Warehousing Example:
  – Huge Base Tables
  – Expensive Joins
  – Archival In Nature
• Mirroring Database