Views

View Modifications — Introduction

Jennifer Widom

Why use views?

- Hide some data from some users

- Make some queries easier / more natural

- Modularity of database access

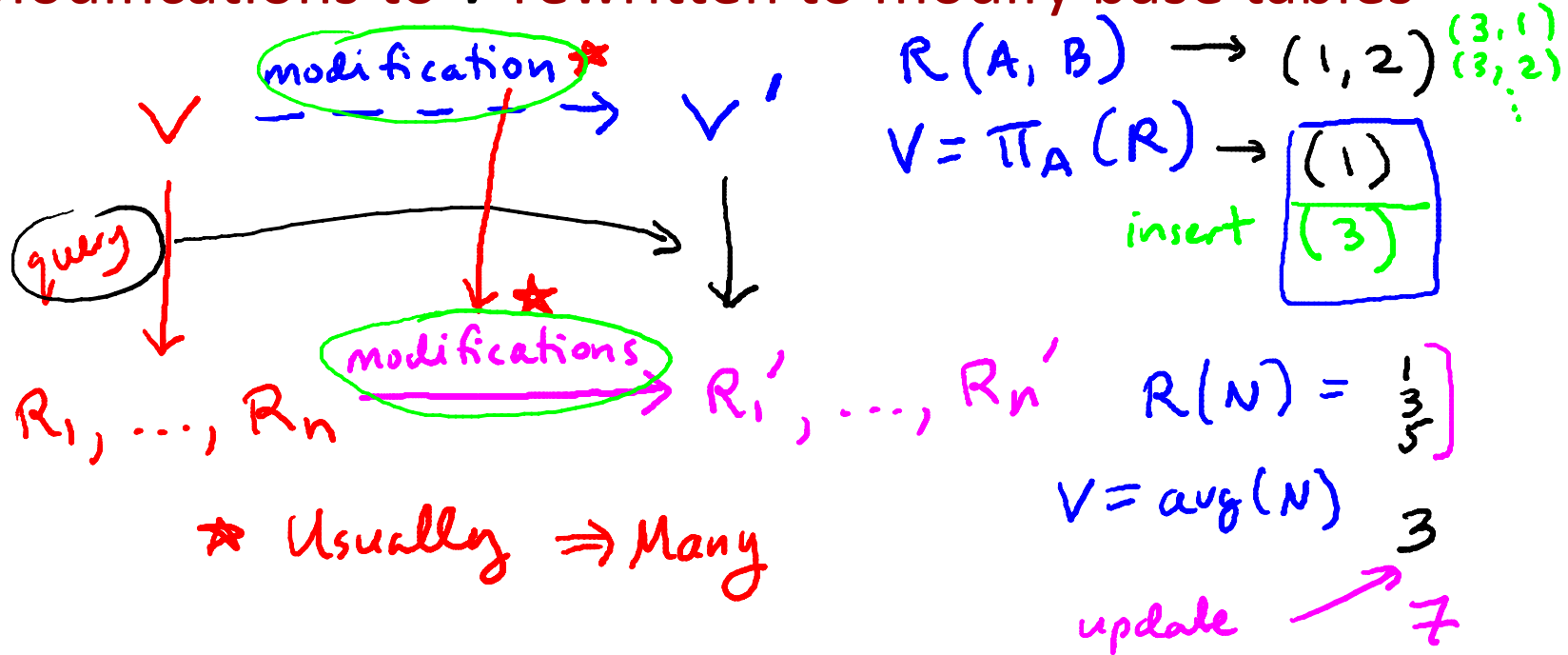> Real applications tend to use lots and lots (and lots and lots!) of views

Querying views

- Once *V* defined, can reference *V* like any table
- Queries involving *V* rewritten to use base tables

Modifying views

- Once *V* defined, can we modify *V* like any table **?**

- Doesn't make sense: *V* is not stored

- Has to make sense: views are some users' entire "view" of the database

❖ Solution: Modifications to *V* rewritten to modify base tables

❖ Modifications to *V* rewritten to modify base tables

V — modification → V'

query

$R_1, \ldots, R_n$ — modifications → $R_1', \ldots, R_n'$

★ Usually ⇒ Many

$R(A, B) \longrightarrow (1, 2)$ (3,1) (3,2)

$V = \pi_A(R) \longrightarrow$ (1) insert (3)

$R(N) = \begin{bmatrix} 1 \\ 3 \\ 5 \end{bmatrix}$

$V = avg(N)$ 3

update → 7

❖ <u>Modifications to *V*</u> rewritten to <u>modify base tables</u>

(1) Rewriting process specified explicitly by view creator
- (+) Can handle all modifications
- (−) No guarantee of correctness (or meaningful)

INSTEAD-OF triggers

RULES

(2) <u>Restrict views + modifications</u> so that translation to base table modifications is meaningful and unambiguous
- (+) No user intervention
- (−) Restrictions are significant

SQL standard

Jennifer Widom