Augmenting Databases: Views

Manipulating Databases

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Kevin C.C. Chang, Professor
Computer Science @ Illinois

Learning Objectives

By the end of this video, you will be able to:

- Define what views are and why we need them.
- Declare and use views in relational databases.
- Explain why views are not always updatable.

Views: What and Why?

- A view is a “virtual” table, a relation that is defined in terms of the contents of other tables (and views).
- We created tables via our data modeling process, to capture the essential needs of the applications.
- However, different users may have different “perspectives”, and they may want a particular subset organized in particular ways.
  - AcademicWorld:
    - Roles: Students vs. instructors vs. administrators.
    - Levels of concerns: CS department vs. EE department vs. College of Engineering.
- Views help to provide such different perspectives.

Creating and Querying a View

- Declare by:

  CREATE VIEW <name> **AS** <query>;

- To contrast, a relation whose value is really stored in the database is called a **base table**.

- You can “query” a view as if it were a base table.
  - But a view may not be “updatable”.

View Examples

- *Q1: Create a view GreenStDrinkers – those drinkers who live on Green St.*

```
mysql> CREATE VIEW GreenStDrinkers AS
    -> SELECT * FROM Drinkers WHERE addr = "Green St";
Query OK, 0 rows affected (0.01 sec)

mysql> SELECT * FROM GreenStDrinkers;
+-------+----------+---------+-----------+
| name  | addr     | hobby   | frequent  |
+-------+----------+---------+-----------+
| Alex  | Green St | Reading | Sober Bar |
| Cindy | Green St | Hiking  | Green Bar |
+-------+----------+---------+-----------+
2 rows in set (0.00 sec)
```

CREAE VIEW query example in MySQL

View Examples

- *Q2: Create a view HappyDrinkers -- drinker and beer pairs for those drinkers who have bars on the same street they live and the bar sells those beers that they drink.*

```
mysql> CREATE VIEW HappyDrinkers AS
    ->   SELECT dr.name AS drinker, dk.beer
    ->   FROM Drinkers dr, Bars b, Sells s, Drinks dk
    ->   WHERE dr.addr = b.addr AND b.name = s.bar AND
    ->         s.beer = dk.beer AND dk.drinker = dr.name;
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT * FROM HappyDrinkers;
+---------+-----------+
| drinker | beer      |
+---------+-----------+
| Alex    | Bud       |
| Alex    | Sam Adams |
+---------+-----------+
2 rows in set (0.00 sec)
```

CREAE VIEW query example in MySQL

View-Querying Examples

- *Q1: Find GreetStDrinkers who frequent Green Bar.*

```
mysql> SELECT name FROM GreenStDrinkers g WHERE g.frequent = "Green Bar";
+-------+
| name  |
+-------+
| Cindy |
+-------+
1 row in set (0.00 sec)
```

View-querying example in MySQL

- *Q2: Find HappyDrinkers whose beers are made by Boston Beer.*

```
mysql> SELECT h.drinker FROM HappyDrinkers h, Beers b WHERE h.beer = b.name AND b.brewer = "Boston Beer";
+---------+
| drinker |
+---------+
| Alex    |
+---------+
1 row in set (0.00 sec)
```

View-querying example in MySQL

Querying a View: What Happens When a View Is Used?

- **View Expansion:** DBMS will replace a view with its definition, thus turning the query into one with only base tables.
  - It will first rewrite the query and view definition both into some internal expressions, e.g., in relational algebra.
  - The view definition expression is then "sliced into" the query expression.

- SELECT name FROM GreenStDrinkers g WHERE g.frequent = "Green Bar";
  - **Query:** $Q = \pi_{name}(\sigma_{\text{frequent="Green Bar"}}(\text{GreenStDrinkers}))$
  - **View:** $\text{GreenStDrinkers} = \sigma_{\text{addr="Green St"}}(\text{Drinkers})$
  - **Query with View Expanded:**

$$Q' = \pi_{name}(\sigma_{\text{frequent="Green Bar"}}(\sigma_{\text{addr="Green St"}}(\text{Drinkers})))$$

Updating a View

- Can we insert a tuple into a “virtual” table?
- Not all views are “updatable”.
- GreenStDrinkers?

HappyDrinkers?

```
mysql> CREATE VIEW GreenStDrinkers AS
    -> SELECT * FROM Drinkers WHERE addr = "Green St";
Query OK, 0 rows affected (0.01 sec)

mysql> SELECT * FROM GreenStDrinkers;
+-------+----------+---------+-----------+
| name  | addr     | hobby   | frequent  |
+-------+----------+---------+-----------+
| Alex  | Green St | Reading | Sober Bar |
| Cindy | Green St | Hiking  | Green Bar |
+-------+----------+---------+-----------+
2 rows in set (0.00 sec)
```

Example view GreenStDrinkers in MySQL

```
mysql> CREATE VIEW HappyDrinkers AS
    ->   SELECT dr.name AS drinker, dk.beer
    ->   FROM Drinkers dr, Bars b, Sells s, Drinks dk
    ->   WHERE dr.addr = b.addr AND b.name = s.bar AND
    ->         s.beer = dk.beer AND dk.drinker = dr.name;
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT * FROM HappyDrinkers;
+---------+-----------+
| drinker | beer      |
+---------+-----------+
| Alex    | Bud       |
| Alex    | Sam Adams |
+---------+-----------+
2 rows in set (0.00 sec)
```

Example view HappyDrinkers in MySQL

Why Not All Views Are Updatable?

- Values of a view may not sufficiently determine values of its base tables.
- 1. Values may be missing.
  - CREATE VIEW BarHasBeer AS (SELECT bar, beer FROM Sells)
  - INSERT (“Green Bar”, “Goose Island IPA”)
    - What is price? Note that price must be NOT NULL in Sells.
- 2. Values cannot not be uniquely determined.
  - CREATE VIEW DeptGPA AS (SELECT major, AVERAGE(gpa) FROM Students GROUPBY major).
  - INSERT (“CS”, 3.0)
    - How to map 3.0 average GPA to every student in CS?
- Views from joins are more complicated, involving multiple base tables.
  - Cannot uniquely determine the values for each base table.

View-Updating Examples

- *Q1: Insert ("Gary", "Green St", "Singing", "Purple Bar") into GreenStDrinkers.*

```
mysql> INSERT INTO GreenStDrinkers VALUES ("Gary", "Green St", "Singing", "Purple Bar");
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM GreenStDrinkers;
+-------+----------+---------+------------+
| name  | addr     | hobby   | frequent   |
+-------+----------+---------+------------+
| Alex  | Green St | Reading | Sober Bar  |
| Cindy | Green St | Hiking  | Green Bar  |
| Gary  | Green St | Singing | Purple Bar |
+-------+----------+---------+------------+
3 rows in set (0.00 sec)

mysql> SELECT * FROM Drinkers;
+-------+----------+---------+------------+
| name  | addr     | hobby   | frequent   |
+-------+----------+---------+------------+
| Alex  | Green St | Reading | Sober Bar  |
| Betty | King St  | Singing | Green Bar  |
| Cindy | Green St | Hiking  | Green Bar  |
| Gary  | Green St | Singing | Purple Bar |
+-------+----------+---------+------------+
4 rows in set (0.00 sec)
```

View-updating example in MySQL

View-Updating Examples

- *Q2: Insert (Harry, "Bud Lite") into HappyDrinkers.*

```
mysql> INSERT INTO HappyDrinkers VALUES ("Harry", "Bud Lite");
ERROR 1394 (HY000): Can not insert into join view 'FridayNight.HappyDrinkers' without fields list
mysql>
```

View-updating example in MySQL

Not All Views Are “Updatable”

- GreenStDrinkers: Yes.
- HappyDrinkers: No.

- Complex rules to determine if a view is updatable, but intuitively:

  A view is updatable if it selects some **enough attribute** from **one relation**.
  - A view that joins multiple tables are not updatable.
  - A view with too few attributes is not updatable.

Food for Thought

Even when a view is updatable, its updates can be tricky. Try inserting (“Paul”, “Purple St”, “Riding”, “Green Bar”) to GreenStDrinkers.
Can you?