CS 327E Lecture 7

Shirley Cohen

February 15, 2016
Agenda

- Reading Quiz
- Views Discussion
- Concept Questions
- Midterm #1 Discussion

Reminder: Midterm #1 is next class
Homework for Today

- Chapter 14 from the Learning SQL book
- Exercises at the end of Chapter 14
Question 1

What is a database view?

A. A mechanism for reading raw data files from disk
B. A mechanism for querying database tables
C. A mechanism for doing bulk imports and exports
D. A web-based interface for running SQL queries
E. None of the above
Question 2

What is NOT a motivation for views?

A. Aggregation: to appear as though data is aggregated
B. Complexity: making multiple tables appear to be a simple table
C. Security: to avoid having to reveal individual data rows
D. Space saving: to reduce the storage of database tables
Question 3

Can you update data through a view?

A. No, views are only designed to simplify a SELECT statement
B. No, views are statically-generated tables and do not update
C. Yes, with several restrictions on clauses and functions
D. Yes, for all views
Question 4

mysql> desc customer;
+-----------------------------+----------+---+--------+------------+------------------+
<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Null</th>
<th>Key</th>
<th>Default</th>
<th>Extra</th>
</tr>
</thead>
<tbody>
<tr>
<td>cust_id</td>
<td>int(10) unsigned</td>
<td>NO</td>
<td>PRI</td>
<td>NULL</td>
<td>auto_increment</td>
</tr>
<tr>
<td>fed_id</td>
<td>varchar(12)</td>
<td>NO</td>
<td></td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>cust_type_cd</td>
<td>enum('I','B')</td>
<td>NO</td>
<td></td>
<td>NULL</td>
<td></td>
</tr>
</tbody>
</table>
+----------------+------------------+------|-------|-----------|------------------+

Which of these views hides the `fed_id` field from the `customer` table?

A. CREATE VIEW `customer_vw` (cust_id, cust_type_cd) AS
   SELECT cust_id, cust_type_cd
   FROM customer;

B. CREATE VIEW `customer_vw` AS
   SELECT cust_id, cust_type_cd
   FROM customer;

C. CREATE VIEW `customer_vw` (cust_id, cust_type_cd) AS
   SELECT c.cust_id, c.cust_type_cd
   FROM customer c;

D. CREATE VIEW `customer_vw` (cust_num, cust_type) AS
   SELECT cust_id, cust_type_cd
   FROM customer;

E. All of the above
Views

- Views are like procedures in SQL
- They are defined by a SQL query
- They return a table of results from the SQL query

Example view:

Employees(ssn, first_name, last_name, role, title, salary)

```sql
CREATE VIEW SeniorStaff AS
    SELECT ssn, first_name, last_name, role, title, salary
    FROM Employees
    WHERE title LIKE 'Senior%'
    ORDER BY salary
```

SeniorStaff(ssn, first_name, last_name, role, title, salary) = virtual table

We can now use the *SeniorStaff* view as if it were a table
Types of Views

• **Virtual views:**
  – computed only on-demand
  – always up-to-date

• **Materialized views:**
  – pre-computed offline
  – requires extra storage
  – may be out-of-date with the base tables
CREATE VIEW CustomerSales AS
    SELECT o.customer_id, i.price
    FROM Orders o, Items i
    WHERE o.item_id = i.id

CustomerSales(customer_id, price) = virtual table

Using the view:

SELECT c.customer_id, c.price, o.store
FROM CustomerSales c, Orders o
WHERE c.customer_id = o.customer_id
AND c.price > 100

Question: How will this query be computed?
Query Modification

Using the view:

```
SELECT  c.customer_id, c.price, o.store
FROM    CustomerSales c, Orders o
WHERE   c.customer_id = o.customer_id
AND     c.price > 100
```

Modified query (at runtime):

```
SELECT  c.customer_id, c.price, o.store
FROM    (SELECT x.customer_id, y.price,
           FROM Orders x, Items y
           WHERE x.item_id = y.id) c, Orders o
WHERE   c.customer_id = o.customer_id
AND     c.price > 100
```
Query Modification

Rewritten query (at runtime):

```sql
SELECT  c.customer_id, c.price, o.store 
FROM    (SELECT x.customer_id, y.price, 
           FROM Orders x, Items y 
           WHERE x.item_id = y.id) c, Orders o 
WHERE   c.customer_id = o.customer_id 
AND     c.price > 100
```

Flattened query (at runtime):

```sql
SELECT  o.customer_id, i.price, o.store 
FROM    Orders o, Items i 
WHERE   o.item_id = i.id 
AND     i.price > 100
```
**Concept Question 1**

Orders(order_id, item_id, customer_id, quantity, store)  
Items(id, item_name, price)

CREATE VIEW CustomerSales AS  
SELECT o.customer_id, o.store, i.price  
FROM Orders o, Items i  
WHERE o.item_id = i.id

CustomerSales(customer_id, store, price) = virtual table

Using the View:

SELECT customer_id  
FROM CustomerSales  
WHERE store = 'Texas Union'

Question: Which base table(s) will be used to answer this query?

A. Only Orders  
B. Orders and Item  
C. Only Item  
D. Customer_Sales
Applications of Views

• Logical Data Independence
  (recall: Physical Data Independence)

• Optimizations
  - vertical partitioning
  - horizontal partitioning

• Security
  - controlled access to fields and records
Vertical Partitioning

CREATE VIEW StudentsView AS
SELECT s.eid, s.first_name, s.middle_initial, s.last_name, p.photo, p.date_taken
FROM Students s, Student_Photo p
WHERE s.eid = p.eid

Using the View:

SELECT eid, first_name, middle_initial
FROM StudentsView
WHERE last_name = 'Evans'

Concept Question 2: Which base table(s) will be used to answer this query?

A. Only Students_Photo  B. Students and Students Photo  C. Only Students
Horizontal Partitioning

Students(eid, first_name, middle_initial, last_name)
Students_Photo_2015(eid, photo, date_taken)
Students_Photo_2016(eid, photo, date_taken)

CREATE VIEW Students_Photos AS
SELECT eid, photo, date_taken
FROM Student_Photo_2015
UNION ALL
SELECT eid, photo, date_taken
FROM Student_Photo_2016

Using the View:

SELECT s.eid, s.first_name, s.middle_initial, s.last_name,
       p.photo, p.date_taken
FROM Students s, Students_Photos p
WHERE s.eid = p.eid
AND p.date_taken < '2015-09-01'

Concept Question 3: Which base table(s) will be used to answer this query?

A. Only Students  B. Students and Students_Photo_2015  C. All base tables
Security Views

Employees(ssn, first_name, last_name, role, title, salary)

CREATE VIEW All_Employee_View AS
  SELECT first_name, last_name, role, title
  FROM Employees
  ORDER BY last_name, first_name

CREATE VIEW Manager_Employee_View AS
  SELECT ssn, first_name, last_name, role, title, salary
  FROM Employees
  WHERE role <> 'Executive'
  ORDER BY last_name, first_name

Concept Question 4: what data do these two views hide?

A. Salary information for all employees
B. Salary information for executives
C. All employee records
D. Only executive employee records
E. A and D
Midterm #1 Topics

- CREATE TABLE
- SELECT CLAUSE
- FROM CLAUSE
- WHERE CLAUSE
- ORDER BY CLAUSE
- Null values
- INNER JOINS
- OUTER JOINS
- GROUP BY and HAVING CLAUSE
- Aggregate functions (count, sum, avg, min, max)
- CREATE VIEWS
Midterm #1 Format

- Closed book exam
- Lasts 90 minutes
- 11 short-answer questions
- Budget 5-7 minutes per question