Database security

Vũ Tuyết Trinh

Learning objectives

- ***Upon completion of this lesson, students will be able to:***
  1. Create views and work correctly on predefined views
  2. Have experience with a DBMS: manage user account and database access permissions

Outline

1. View
2. Privileges and User Management in SQL

1. View

1.1. View definition
1.2. Accessing views
1.3. Updatable views
1.4. Materialized views

1.1. View definition

- A view is a relation defined in terms of stored tables (called base tables) and other views
- Two kinds:
  - Virtual = not stored in the database; just a query for constructing the relation
  - Materialized = actually constructed and stored
- Declaring views:

  CREATE [MATERIALIZED] VIEW <name> AS <query>;
  - Default is virtual

5

5

1.1. View definition: Removal

- Dropping views: DROP VIEW <name>;

  DROP VIEW female_student;
- Affection:
  - Deleting the definition of views: the female_student view no longer exists
  - No tuples of the base relation (student relation) is affected

6

6

1.2. Accessing views

- Declare:

```
CREATE VIEW monitor AS
SELECT student_id, first_name, last_name, dob, clazz_id
FROM student, clazz
WHERE student_id = monitor_id ;
```

- Query a view as if it were a base table

```
SELECT student_id, first_name, last_name, dob
FROM monitor
WHERE clazz_id = '20172201' ;
```

- A limited ability to modify views

7

7

1.3. Updatable views

- The SQL rules are complex
- They permit modifications on views that are defined by selecting (using SELECT, not SELECT DISTINCT) some attributes from one relation *R* (which may itself be an updatable view):
  - The **WHERE** clause must not involve *R* in a subquery
  - The **FROM** clause can only consist of one occurrence of *R* and no other relation
  - The **list in the SELECT clause** must include enough attributes that for every tuple inserted into the relation R (other attributes filled with NULL values or the proper default)
  - There is **no GROUP BY** clause

8

8

1.3. Updatable views: Example

- Base table:
  - student(**student_id**, first_name,last_name,dob,gender,address,note,*clazz_id*)
- Updatable view

```
CREATE VIEW female_student AS
        SELECT student_id, first_name, last_name FROM student
        WHERE gender = 'F';
```

- Insert into views:

```
INSERT INTO female_student VALUES('20160301', 'Hoai An', 'Tran');
```

means

```
INSERT INTO student(student_id, first_name, last_name)
VALUES ('20160301', 'Hoai An', 'Tran');
```

9

9

1.3. Updatable views: Example

- Delete from views:

```
DELETE FROM female_student WHERE first_name LIKE '%An' ;
```

means

```
DELETE FROM student
WHERE first_name LIKE '%An'  AND gender = 'F';
```

- Update views:

```
UPDATE female_student SET first_name = 'Hoài Ân'
WHERE first_name = 'Hoai An' ;
```

means

```
UPDATE student SET first_name = 'Hoài Ân'
WHERE first_name = 'Hoai An'  AND gender = 'F';
```

10

10

1.3. Updatable views: Views and INSTEAD OF trigger

- Generally, it is impossible to modify a virtual view, because it doesn't exist.
- But an INSTEAD OF trigger (next lesson) lets us interpret view modifications in a way that makes sense

```
CREATE TRIGGER delete_viewtrigger
  INSTEAD OF DELETE ON monitor
  FOR EACH ROW
  BEGIN
        UPDATE clazz SET monitor_id = NULL
        WHERE clazz_id = OLD.clazz_id;
  END;
```

11

11

1.4. Materialized Views

- Results of a query can be stored
  - This enables much more efficient access
- Problems
  - Each time a base table changes, the materialized view may change
- Solutions
  - Periodic reconstruction (REFRESH) of the materialized view
  - Triggers (next lesson)

12

12

2. Privileges and User Management in SQL

2.1. Privileges
2.2. Creating users
2.3. Granting privileges
2.4. Revoking privileges

13

13

2.1. Privileges

- SELECT, INSERT, DELETE, UPDATE: privileges on table/view
- REFERENCES: privilege on a relation; the right to refer to that relation in an integrity constraint
- USAGE: the right to use that element in one's own declarations
- TRIGGER: privilege on a relation; the right to define triggers on that relation
- EXECUTE: the right to execute a piece of code, such as a procedure or function
- UNDER: the right to create subtypes of a given type

14

14

2.2. Creating users

- Syntax: variations in different database platforms
  - Creating an user in Oracle, MySQL:

    CREATE USER *username* IDENTIFIED BY *password*;
  - Creating an user in PostgreSQL:

    CREATE USER *username*

    [[WITH] options] PASSWORD *password*;
  - Deleting:

    DROP USER *username* [CASCADE];
- Example:

  CREATE USER *toto* IDENTIFIED BY *pwdtoto*

15

2.3. Granting privileges

- Syntax:

  GRANT <privilege list> ON <database element> TO <user list>

  [WITH GRANT OPTION] ;
  - <privilege list> : INSERT, SELECT, …, ALL PRIVILEGES
  - <database element>: a table, a view
  - WITH GRANT OPTION:
    - the user may grant the privilege to other user
- Example:

  GRANT SELECT, INSERT ON student TO tom WITH GRANT OPTION;

16

2.4. Revoking privileges

- Syntax:

  REVOKE <privilege list> ON <database element> FROM <user list>

  [CASCADE| RESTRICT] ;

  - CASCADE : revoke any privileges that were granted only because of the revoked privileges
  - RESTRICT: the revoke statement cannot be executed if the revoked privileges have been passed on to others

  REVOKE GRANT OPTION FOR …..; : remove the grant option

- Example:

  REVOKE INSERT ON student FROM tom CASCADE;

17

Summary

- View
  - View definition
  - View accessing
  - Updatable view
  - Materialized view
- Privileges and User Managements
  - Privileges
  - Creating user
  - Granting / Revoking privileges

18