CSC12001

Data Security in Information Systems

C03 - Access Control

Dr. Phạm Thị Bạch Huệ
MSc. Lương Vĩ Minh

Information System Department – Faculty of Information Technology
University of Science, VNU-HCM

Outline

1. Objectives.
2. View is for maintaining privacy.
3. View is for hiding column.
4. View is for masking the data values.

Objectives

- Introduce various ways of restricting access to the data tables.
- You will see how database views can be used as effective security mechanisms for providing security for **the individual table rows and columns.**
  - Row-level security or fine-grained access control
  - Tools developed by Oracle for securing the data at a fine level of granularity.

- A user with the select privilege on a table will be able to read all records within that table → the database privileges apply to accessing the object (table or view) but do not specify the security within the object.
- Limiting access to data within tables is desirable and is often a security requirement.

- Views are database objects and access to them occurs at the object level.
- Privileges on the view are separate and distinct from the privileges on the underlying objects the view accesses.
- Allowing users access to a view and not to the underlying objects is an effective security technique for insulating your sensitive data.

Views are used maintaining privacy

- EMP, DEPT
- User SCOTT wishes to allow certain users to access the number of employees for each department. He does not have to allow access to the EMP and DEPT tables.
- Create a view that performs the calculation.

```
Scott: CREATE OR REPLACE VIEW emp_dist
AS
SELECT INITCAP (d.dname) "Department",
COUNT (e.ename) "Total Employees"
FROM dept d, emp e
WHERE e.deptno = d.deptno
GROUP BY dname;
```

- Granting access to the view then allows users to retrieve this summary data while simultaneously maintaining separate security for the underlying objects

```
Scott:  GRANT SELECT ON emp_dist TO BLAKE;
```

```
Blake: SELECT * FROM scott.emp_dist;
Department Total Employees
Accounting         3
Research           5
Sales              6
Blake: SELECT * FROM scott.emp;

SELECT * FROM scott.emp * ERROR at line 1: ORA-00942: table or
view does not exist
```

By using view, sensitive information of individual is hidden.

Views can hide columns

- EMP(ename, job, sal)
- Views are ideal tool for providing column-level security (CLS).
- Removing access to the SAL column in EMP table because it contains sensitive data.

```
ALTER TABLE emp RENAME TO emp$;
-- create view that removes sensitive columns
CREATE VIEW emp AS SELECT ename, job
FROM emp$;
-- grant access to view
GRANT SELECT ON emp TO user1;
-- revoke access from table
REVOKE SELECT ON emp$ FROM user1;
```

Views can mask data values

- Users can access only their salary.
  - Users should be prohibited from accessing other users' salary data.
  - Because the user can access their salary, you cannot simply omit the SAL column from the view definition.
- The view will use the Oracle built-in DECODE function to implement the column masking. If the user accessing the record is the same as the person in the record, then they are allowed to see the salary; otherwise, a null value is returned.

```
CREATE OR REPLACE VIEW people_cls
AS
SELECT username,
job,
deptno,
DECODE (username, USER, salary, NULL) salary
FROM people;
```

You don’t grant privileges on the base table. When granting privileges on the view, you can allow the user to read all records, except the other users’ salaries.

Scott:

```
SELECT * FROM people_cls;
USERNAME    JOB          DEPTNO      SALARY
SMITH       CLERK        20
JONES       MANAGER      20
SCOTT       ANALYST      20          3000
ADAMS       CLERK        20
FORD        ANALYST      20
```

Computing salaries for everyone will only return user's salary

```
Scott: SELECT SUM (salary)
FROM people_cls;
SUM(SALARY)
3000
```

- A user cannot issue direct updates on the view. Example, give everyone a 10% raise.

```
UPDATE people_cls
SET salary = salary * 1.1;
```

- SET salary = salary * 1.1 * ERROR at line 2: ORA-01733: virtual column not allowed here

- Instead-of triggers for performing DML operations on complex views.
- Scott:

```
CREATE OR REPLACE TRIGGER people_sal_update
INSTEAD OF UPDATE
ON people_cls
FOR EACH ROW
DECLARE
BEGIN
IF :OLD.salary IS NOT NULL
THEN
UPDATE people SET salary = :NEW.salary WHERE username = :NEW.username;
END IF;
END;
```

- scott> UPDATE people_cls

```
SET salary = salary * 1.1;
5 rows updated.
```

- Oracle built-in functions such as DECODE generally perform well and always outperform user-created PL/SQL functions.

- Column-level privileges allow the user to update the column value for all records in the table. This may or may not be desirable. In the previous example, it may be unlikely that a user should have the ability to update other users' phone numbers.
- `Scott: ALTER TABLE people ADD (phone_number VARCHAR2(20));`
  `Table altered.`
- `Scott: GRANT UPDATE (phone_number) ON people TO user1;`

`Grant succeeded.`

- `User1: UPDATE scott.people SET phone_number = '555-1212';`

- The column-level privileges allow the user to update the column value for all records in the table.

- A view is used for an application that **allows a user to update their personal record.**
- A user (or hacker) can't update or modify someone else's record.
- We create a view that ensures the only record displayed to the user will be theirs. The view's security will eliminate all other records. This is done by adding a predicate or where clause to the query on the base table.

```
Scott: CREATE OR REPLACE VIEW people_edit
AS
SELECT * FROM emp
WHERE ename = SYS_CONTEXT ('userenv', 'session_user')
WITH CHECK OPTION;

View created.
```

- Scott: SELECT empno, ename, sal
  FROM people_edit;

```
     EMPNO ENAME                SAL
---------- ---------- ----------
      7788 SCOTT              3000
```

Updating records outside of the view definition has no effect.

```
Scott: UPDATE people_edit
     SET ename = 'Bozo'
     WHERE ename = 'KING';
```

- Deletes are also constrained to the view definition. Inserts are allowed as long as the ENAME value matches the user's name:

```
Scott: DELETE FROM people_edit;
1 row deleted.
Scott: INSERT INTO people_edit
                (empno, ename, sal) VALUES (7788, 'SCOTT', '3000');
1 row created.
Scott: INSERT INTO people_edit
               (empno, ename, sal) VALUES (7788, 'SCOTT2', '3000');

INSERT INTO people_edit
            *
ERROR at line 1:
ORA-01402: view WITH CHECK OPTION where-clause violation
```

Q&A

Dr. Phạm Thị Bạch Huệ - ptbhue@fit.hcmus.edu.vn

MSc. Lương Vĩ Minh - lvminh@fit.hcmus.edu.vn