香港中文大學（深圳）
The Chinese University of Hong Kong, Shenzhen

SCHOOL OF DATA SCIENCE
數據科學學院

CSC3170
Tutorial 2

School of Data Science

The Chinese University of Hong Kong, Shenzhen

SCHOOL OF DATA SCIENCE

Outline

- Views
  - Nested Views
- Authorization
  - Roles

Views

- Motivation: Sometimes, it is not desirable for all users to see the entire relation.
  - Example: a certain staff working at cuhksz can view the basic information of instructors, but not their salary.

Actual Relation

id	name	department	title	salary	address

Visible Relation

id	name	department	title

- View: provides a mechanism to hide certain data from certain users

Views

- Definition:

  create view v **as** <query expression>

  where <query expression> is any legal SQL expression.

Actual Relation

id	name	department	title	salary	address

Visible Relation

id	name	department	title

- A view of instructors:

```
create view faculty as
    select id, name, department, title
    from instructor
```

- Find all instructors in SDS:

```
select name
    from faculty
    where department = ‘SDS’
```

Views

- Practice: create a view of department salary totals (department, total_salary)

instructor

id	name	department	title	salary	address

Views

- Practice: create a view of department salary totals (department, total_salary)

instructor

id	name	department	title	salary	address

```
create view dept_salary_totals as
        select department, sum(salary) as total_salary
        from instructor
        group by department
```

Nested Views

- We can define a view *v2* based on another view *v1*.

customers

customer_id	customer_name	city

orders

order_id	customer_id	product_id	quantity	date

products

product_id	product_name	price

- Step 1: we need a view that shows the order details for each customer.

```
create view customerOrderDetailsView as
    select c.customer_id, c.customer_name, o.order_id, o.product_id, o.quantity
    from customers c
    join orders o on c.customer_id = o.customer_id;
```

Nested Views

- We can define a view *v2* based on another view *v1*.

customerOrderDetailsView

customer_id	customer_name	order_id	product_id	quantity

products

product_id	product_name	price

- Step 2: we need a view that shows the total amount of each customer's order.

```
create view customerOrderSummaryView as
    select cod.customer_id, cod.customer_name, cod.order_id, p.product_name,
              cod.quantity, (cod.quantity * p.price) as total_amount
    from customerOrderDetailsView cod
    join products p on cod.product_id = p.product_id;
```

SCHOOL OF DATA SCIENCE

Nested Views

- Advantages of nested views:
  - Simplification of Complex Queries

What will happen if we directly create the second view (customerOrderSummaryView) ?

```
create view customerOrderSummaryView as
    select cod.customer_id, cod.customer_name, cod.order_id, p.product_name,
            cod.quantity, (cod.quantity * p.price) as total_amount
    from (select cod.customer_id, cod.customer_name, cod.order_id, p.product_name,
                cod.quantity, (cod.quantity * p.price) as total_amount
                from customerOrderDetailsView cod
                join products p on cod.product_id = p.product_id) cod
    join products p on cod.product_id = p.product_id;
```

Nested Views

- Advantages of nested views:
  - Simplification of Complex Queries
  - Data Security (Permission Management)
  - Dynamic (Always returns real-time result)
  - Readability

Authorization

- We may assign a user several forms of authorization (privilege):
    - Read (select), Insert (insert), Update (update), Delete (delete), All Privileges (all)

- Example: a certain staff working at cuhksz can view the basic information of instructors, but not their salary.

faculty

id	name	department	title

- Confer Authorization: **grant** <privilege list> **on** <relation / view> **to** <user list>
  grant select **on** faculty **to** staffA: staffA can run "select" statement on faculty

- Revoke Authorization: **revoke** <privilege list> **on** <relation / view> **to** <user list>

Roles

- Problem: we may need to assign a group of user a certain privilege.
  - Example: 50 staffs in HR department can access *faculty*

    grant select **on** faculty **to** staff1, staff2, …, staff50

    what if we want to revoke the privilege?

- A role is a way to distinguish among various users as far as what these users can access/update in the database.

- Create a role: **create role** <name>
- Assign a user to a role: **grant** <role> **to** <users>

  grant select **on** faculty **to** hr_department

Reminder

1. Due date of As1 is 9.28 23:59.
2. The website cannot be accessed outside the school. Please connect to the campus network to open the website. https://oj.cuhk.edu.cn/d/csc3170_2025_fall/
3. After you login the OJ, it will automatically go to the main site, so you need manually copy the whole link again.
4. When you submit your solution (Q4 – Q23), you just need to write the SQL query (that is, **do not add the table creation and record insertion code** anymore)

SCHOOL OF DATA SCIENCE

Q&A

Thanks to the previous CSC3170 teaching team from which part of the content was sourced.