

Module - II

SQL - Structured Query Language

Lecture - 8

Views | Index Definition

SQL - Views

➤ Views

- View Definition
- View Definition and Use
- Views Defined Using Other Views
- View Expansion
- Materialized Views
- Update of a View
- View Updates in SQL

SQL - Index Definition

- Index Definition in SQL
 - Index Creation
 - Index Creation Example

SQL - Views

➤ Views

- View Definition
- View Definition and Use
- Views Defined Using Other Views
- View Expansion
- Materialized Views
- Update of a View
- View Updates in SQL

Views

- In some cases, it is not desirable for all users to see the entire logical model (that is, all the actual relations stored in the database.)
- **Ex:** *Consider a person who needs to know an instructors name and department, but not the salary.* This person should see a relation described, in SQL, by

```
select ID, name, dept_name  
from instructor
```

Views

- A **view** provides a mechanism to hide certain data from the view of certain users.
- Any relation that is not of the conceptual model but is made visible to a user as a “**virtual relation**” is called a **view**.

View Definition

- A **view** is defined using the **create view** statement which has the form

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

where <query expression> is any legal SQL expression.
The **view name** is represented by *v*.

- Once a **view** is defined, the **view name** can be used to refer to the **virtual relation** that the **view generates**.

View Definition

- **View definition** is not the same as creating a **new relation** by evaluating the **query expression**.
 - Rather, a **view definition** causes the **saving of an expression**; the **expression** is substituted into queries using the **view**.

View Definition and Use

```
select ID, name, dept_name  
from instructor
```

- **Ex-1:** *A view of instructors without their salary*

```
create view faculty as  
select ID, name, dept_name  
from instructor
```

- **Query using View:** *Find all instructors in the Biology department*

```
select name  
from faculty  
where dept_name = 'Biology'
```

View Definition and Use

- **Ex-2:** *Create a view of department salary totals*

```
create view departments_total_salary(dept_name,  
total_salary) as  
    select dept_name, sum (salary)  
    from instructor  
    group by dept_name;
```

Views Defined Using Other Views

- One view may be used in the expression defining another view
- A view relation v_1 is said to **depend directly** on a view relation v_2 if v_2 is used in the expression defining v_1
- A view relation v_1 is said to **depend on view relation** v_2 if either v_1 depends directly to v_2 or there is a path of dependencies from v_1 to v_2
- A view relation v is said to be **recursive** if it depends on itself.

Views Defined Using Other Views

Ex:

```
create view physics_fall_2017 as  
  select course.course_id, sec_id, building, room_number  
  from course, section  
  where course.course_id = section.course_id  
        and course.dept_name = 'Physics'  
        and section.semester = 'Fall'  
        and section.year = '2017';
```



```
create view physics_fall_2017_watson as  
  select course_id, room_number  
  from physics_fall_2017  
  where building= 'Watson';
```

View Expansion

- Expanded the view:

```
create view physics_fall_2017_watson as  
  select course_id, room_number  
  from physics_fall_2017  
  where building= 'Watson';
```

- To:

```
create view physics_fall_2017_watson as  
  select course_id, room_number  
  from (select course.course_id, building, room_number  
         from course, section  
         where course.course_id = section.course_id  
              and course.dept_name = 'Physics'  
              and section.semester = 'Fall'  
              and section.year = '2017')  
  where building= 'Watson';
```

View Expansion (cont.)

- A way to define the **meaning of views** defined in terms of **other views**.
- Let **view** v_1 be defined by an expression e_1 that may itself contain uses of view relations.
- **View expansion** of an expression repeats the following **replacement** step:
 - repeat**
 - Find any view relation v_i in e_1
 - Replace the view relation v_i by the expression defining v_i
 - until** no more view relations are present in e_1
- As long as the **view definitions** are not recursive, this loop will terminate

Materialized Views

- Certain database systems allow **view relations** to be **physically stored**.
 - **Physical copy** created when the **view** is defined.
 - Such **views** are called **Materialized view**:
- If **relations** used in the **query** are updated, the **materialized view** result becomes out of date
 - Need to **maintain** the **view**, by **updating the view** whenever the underlying relations are updated.

Update of a View

- Add a new tuple to *faculty* view which we defined earlier

```
insert into faculty  
values ('30765', 'Green', 'Music');
```

- This insertion must be represented by the insertion into the *instructor* relation

- Must have a **value for salary**.

- **Two approaches**

- Reject the insert
- Insert the tuple

```
('30765', 'Green', 'Music', null)
```

into the *instructor* relation

Some Updates Cannot be Translated Uniquely

- `create view instructor_info as`
 `select ID, name, building`
 `from instructor, department`
 `where instructor.dept_name = department.dept_name;`
- `insert into instructor_info`
 `values ('69987', 'White', 'Taylor');`
- **Issues**
 - Which department, if multiple departments in Taylor?
 - What if no department is in Taylor?

And Some Not at All

- **create view** *history_instructors* **as**
 select *
 from *instructor*
 where *dept_name*= 'History';
- What happens if we insert
 ('25566', 'Brown', 'Biology', 100000)
 into *history_instructors*?

View Updates in SQL

- Most SQL implementations allow updates only on **simple views**
 - The **from** clause has only one database relation.
 - The **select** clause contains only attribute names of the relation, and does not have any expressions, aggregates, or **distinct** specification.
 - Any attribute not listed in the **select** clause can be set to null
 - The query does not have a **group** by or **having** clause.

SQL - Index Definition

- Index Definition in SQL
 - Index Creation
 - Index Creation Example

Index Creation

- Many **queries** reference only a small proportion of the records in a **table**.
- It is **inefficient** for the system to read every record to find a record with particular value.
- An **index** on an attribute of a relation is a data structure that allows the database system to find those tuples in the relation that have a specified value for that attribute efficiently, without scanning through all the tuples of the relation.
- We **create** an **index** with the **create index** command

```
create index <name> on <relation-name> (attribute);
```

Index Creation Example

- **create table** *student*
(*ID* **varchar** (5),
name **varchar** (20) **not null**,
dept_name **varchar** (20),
tot_cred **numeric** (3,0) **default** 0,
primary key (*ID*))
- **create index** *studentID_index* **on** *student* (*ID*)

The query:

```
select *  
from student  
where ID = '12345'
```

can be executed by using the **index** to find the required record,
without looking at all records of *student*