Views and Other Features

- Relational Views
  - View Definition
  - View Deletion
  - Data Retrieval from Views
  - View Update
- System Catalog

Relational Views

- Derived relations from other relations (base relations).
- Has no stored tuples.
- Provide multiple user views
- Support logical data independence.
- Provide data security
- Simplification of users' perspective

View Creation

CREATE VIEW view-name [ ( attr [ , attr ] ... ) ]
AS subquery;

DROP VIEW view-name;

Create a view containing the student ID, Name, Age and GPA for those who are qualified to take 3000-level courses, i.e., GPA >= 2.9.

CREATE VIEW Qualified_Student
AS
select ID, Name, Age, GPA
from Student
where GPA >= 2.9;
More on Views

- Can assign a name to each column in view.
- If list of column names is specified, it must have same number of items as number of columns produced by subquery.
- If omitted, each column takes name of corresponding column in subquery.
- A view may be derived from multiple base relations
- Need SELECT privilege on all tables referenced in subselect

A view for all CIS 2010 students:

```sql
create view CIS2010_Student (ID, Name, Instructor_Name) as
  select STUDENT.ID, STUDENT.Name, INSTRUCTOR.Name
  from STUDENT, CLASSROOMS, INSTRUCTOR
  where STUDENT.ID = CLASSROOMS.SID and
  CLASSROOMS.Emp# = INSTRUCTOR.Emp# and
  CLASSROOMS.course# = 'CIS2010';
```

View Retrieval

- Queries on views are the same as that on base relations.
- Queries on views are expanded into queries on their base relations.

```sql
select Name, Instructor_Name
from CIS2010_Student
where Name = Instructor_Name;
```

What is this query asking in plain English?

View Update

- Update on a view actually changes its base relations
- Some views are not updatable
  - A view may be updatable if it contains all primary keys of its base relations

```sql
update Qualified_Student
set GPA = 0.1
where SID = 's3';
```

```sql
insert into Qualified_Student
values ('s9', 'Lisa', 23, 4.0)
```

```sql
insert into Qualified_Student
values ('s10', 'Peter', 32, 1.7)
```
**View Update...**

```
insert into CIS2010_Student
values
('s10', 'Bill', 'Lisa');
```
**View Materialization**

- View resolution mechanism may be slow, particularly if view is accessed frequently.
- View materialization stores view as temporary table when view is first queried.
- Thereafter, queries based on materialized view can be faster than recomputing view each time.
- Difficulty is maintaining the currency of view while base tables(s) are being updated.

**System Catalog**

Stores the meta-data information about relations and users

**ORACLE Example:**

The `SYS_TAB` relation tells about relation or views:

```
SYS_TAB ( TName, TabType, ClusterID )
```

- `TName` is relation/view name
- `TabType` distinguishes a relation from a view
- `ClusterID` indicates which cluster stores the table.

The `SYSTEM.SYSCATALOG` relation records other information about a relation:

```
SYSTEM.SYSCATALOG ( TName, Creator, TableType, ClusterID, LogBlk, ReqBlk, IXComp, Remarks )
```

The `SYS.COL` relation stores information about attributes:

```
SYS.COL ( CName, TName, Creator, ColNo, ColType, ... )
```

The `SYS.ALL_USERS` relation keeps user information:

```
SYS.ALL_USERS ( UserID, UserName, TimeStamp, ConnectAuth, ... )
```

and more...