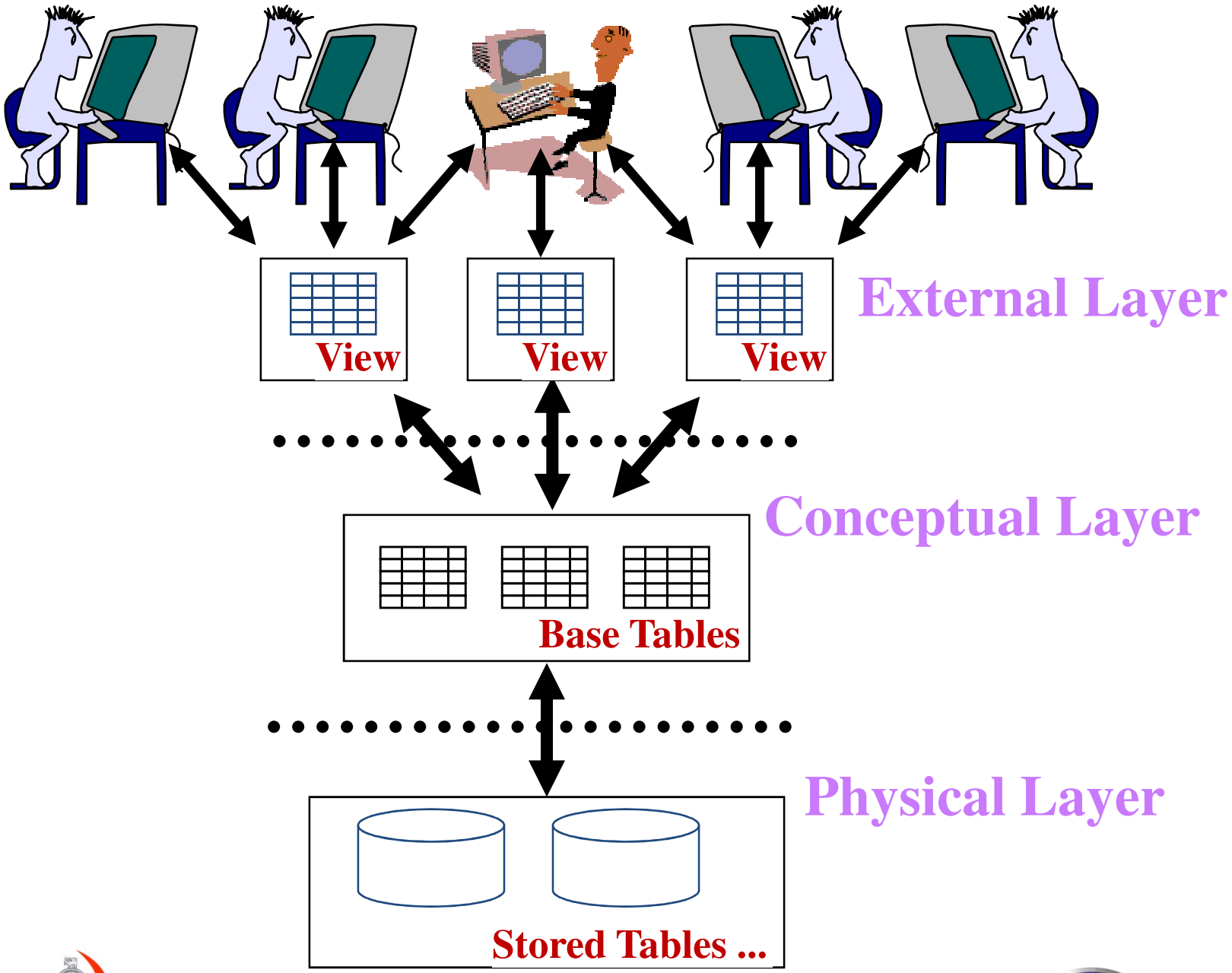
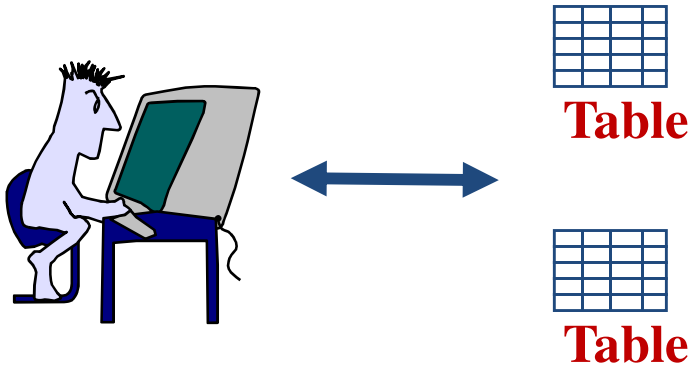
Database Management Systems
Views

Dr. Jeevani Goonetillake
UCSC





Architecture

Conceptual Layer

- The ***conceptual model*** is a logical representation of the entire contents of the database.
- The conceptual model is made up of ***base tables***.
- Base tables are “real” in that they contain physical records.

Architecture Conceptual Layer

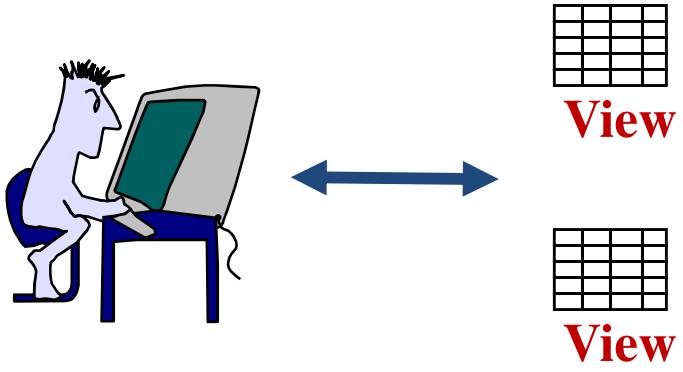
Department

Dept_Code	Dep_Name	Manager
SAL	Sales	179
FIN	Finance	857

Employee

Emp_No	Emp_Name	Designation	DOB	Dept
179	Silva	Manager	12-05-74	SAL
857	Perera	Accountant	01-04-67	FIN
342	Dias	Programmer	25-09-74	SAL

Base Tables



Architecture

External Layer

- The ***external model*** represents how data is presented to users.
- It is made up of ***view tables***. View tables are "virtual"-- they do not exist in physical storage, but appear to a user as if they did

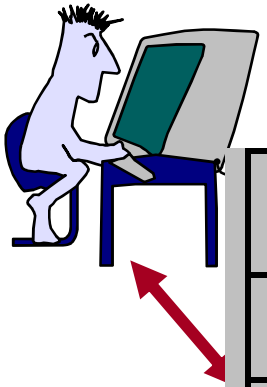
External Layer

Department_View

Dept_Code	Dep_Name	Manager
SAL	Sales	Silva
FIN	Finance	Perera

Employee_View

Emp_No	Emp_Name	Designation	Age	Dept
179	Silva	Manager	27	SAL
857	Perera	Accountant	34	FIN
342	Dias	Programmer	26	SAL



External Layer

Emp_Personnel

Emp_No	Emp_Name	Designation	Age	Dept
179	Silva	Manager	27	Sales
857	Perera	Accountant	34	Finance
342	Dias	Programmer	26	Sales

View Tables

Department

Dept_Code	Dep_Name	Manager
SAL	Sales	179
FIN	Finance	857

Employee

Emp_No	Emp_Name	Designation	DOB	Dept
179	Silva	Manager	12-05-74	SAL
857	Perera	Accountant	01-04-67	FIN
342	Dias	Programmer	25-09-74	SAL

Base Tables

What Are User views?

A view is a "Virtual Table". In SQL terminology, view is a single table that is derived from other tables.

- *User views*
  - Derived or virtual tables that are visible to users
  - Do not occupy any storage space
- Base Tables
  - Store actual rows of data
  - Occupy a particular amount of storage space

Characteristics of User views

- Behave as if it contains actual rows of data, but in fact contains none.
- Rows are derived from base table or tables from which the view is defined.
- Being virtual tables the possible update operations that can be applied to views are limited. However, it does not provide any limitations on querying a view.

SQL and User Views

Creating a View

CREATE VIEW *view-name*

(*list of attribute names,)*

AS *query*

SQL User Views

- Column names specified must have the same number of columns derived from the query
- Data definitions for each column are derived from the source table
- Columns will assume corresponding column names in the source table. Names must be specified for calculated or identical columns.

Specification of Views

Works_On1

Fname	Lname	Pname	Hours

CREATE VIEW Works_On1 **AS**
SELECT Fname, Lname, Pname, Hours
FROM Employee, Project, Works_On
WHERE Essn = Empid
AND Pno = Pnumber ;

Specification of Views

Dept_Info

Dept_name	No_Of_Emps	Total_Sal

```
CREATE VIEW Dept_Info (Dept_Name,
                No_Of_Emps, Total_Sal) AS
SELECT   Dname, COUNT(*), SUM(Salary)
FROM     Department, Employee
WHERE    Dnumber = Dno
GROUP BY  Dname ;
```

Specification of Views

- Retrieve the last name and first name of all employees who work on ‘ProjectX’.

```
SELECT Fname, Lname
  From Works_On1
  WHERE Pname = ‘ProjectX’ ;
```

Why User views?

Benefits

Security

Protect data from unauthorized access. Each user is given permission to access the database via only a small set of views that contain specific data the user is authorized to see.

User views

Query Simplicity

Turning multiple table queries to single table queries against views, by drawing data from several tables. It provides flexible and powerful data access capabilities.

User views

Query Simplicity contd.

It also improves productivity of end-user and programmers by:

- Simplifying database access by presenting the structure of data that is most natural to the user.
- Simplifying the use of routine and repetitive statements

User views

Natural Interface

"Personalized" view of database structure, that make sense for the user. Restructure or tailor the way in which tables are seen, so that different users see it from different perspectives, thus allowing more natural views of the same enterprise (e.g. item names)

User views

Insulation from change

Data independence - maintain independence among different user views and between each user view and the physical constructs.

A view can present a consistent image of the database structure, even if the underlying source tables are restructured.

User view Design Considerations

- User view design is driven by specific application requirements
- User may be defined for individual user, or a group of users, of the transaction or application

Design Considerations

- User view may be defined to control and restrict access to specific columns and/or rows in one or more tables
- User views can be defined to help simplify queries, application development and maintenance
- User views may be derived from base tables or other user views

Remove a User View

Drop a View

DROP VIEW *view-name*

E.g.

DROP VIEW Emp_Payroll

Removes only the definition of the view table.
Data that it used to retrieve is not affected.

View Implementation

- Two main approaches have been suggested for efficiently implementing a view for querying.
  - query modification
  - view materialization

Query Modification

- Involves modifying the view query into a query on the underlying base tables. For example:

```
SELECT Fname, Lname
 From Works_On1
WHERE Pname = 'ProjectX' ;
```

```
SELECT Fname, Lname
   From Employee, Project, Works_On
WHERE Empid = Essn AND Pno = Pnumber
    AND Pname = 'ProjectX' ;
```

Query Modification

- The disadvantage of this approach is that it is inefficient for views defined via complex queries that are time consuming to execute.

View Materialization

- Involves physically creating a temporary view table when the view is first queried and
- Keeping the table on the assumption that other queries on the view will follow.
- An efficient strategy for automatically updating the view table when the base tables are updated must be developed – incremental update.
- The view is kept as long as it is being queried.

View Update

- Updating of views is complicated and can be ambiguous.

- An update on a view defined on a single table without any aggregate functions can be mapped to an update on the underlying base table under certain conditions.

- For a view involving joins, an update operation may be mapped to update operations on the underlying base relations in multiple ways.

View Update

- Consider the view Works_On1 and issue command to update the Pname attribute of ‘Sunil Perera’ from ‘ProductX’ to ‘ProductY’.

```
UPDATE Works_On1
 SET Pname = ‘ProductY’
 WHERE Lname = ‘Perera’ AND Fname = ‘Sunil’
   AND Pname = ‘ProductX’;
```

This query can be mapped into several updates on the base relations to give the desired update effect on the view.

View Update

a) UPDATE Works_On
SET Pno = (SELECT Pnumber
FROM Project
WHERE Pname = 'ProductY'
WHERE Essn IN (SELECT Empid
FROM Employee
WHERE Lname = 'Perera' AND Fname = 'Sunil')
AND Pno = (SELECT Pnumber
FROM Project
WHERE Pname = 'ProductX') ;

b) UPDATE Project
SET Pname = 'ProductY'
WHERE Pname = 'ProductX' ;

View Update

- A view update is feasible when only one possible update on the base relations can accomplish the desired update effect on the view.

- Whenever an update on the view can be mapped to more than one update on the underlying base relations, there must be a certain procedure for choosing the desired update.

With Check Option

Migrating rows

When a row is altered such that it no longer satisfies WHERE condition then it will disappear from the view. New rows will appear within the as a result of update or insert statement. Rows that enter or leave a view are called migrating rows.

- WITH CHECK OPTION clause of the CREATE VIEW statement prohibits a row migrating out of the view.

- Ensures that if a row fails to satisfy WHERE clause of defining query, it is not added to underlying base table

Eg:

```
CREATE VIEW  Emp_View AS
SELECT       *
FROM         Employee
WHERE        Dept = 'SAL'
WITH CHECK OPTION
```

If we now attempt to update the Department of one of the rows from SAL to FIN for example ;

```
UPDATE Emp_View
SET Dept = FIN
WHERE Emp_no = 179
```

This would cause the row to migrate from this view. But WITH CHECK OPTION clause in the definition of the view prevents this from happening

Limitations of User views

Restrictions on views processing

SELECT, INSERT, UPDATE and DELETE statements may refer to views, but there are a number of limitations.

Update may be possible for ‘simple’ views but not ‘complex’ views.

Limitations

Performance

DBMS must translate queries against the view to queries against the source tables.

These disadvantages means that we cannot indiscriminately define and use views instead of source tables.