COSC344
Database Theory and Applications

Lecture 16 Views & NULL

Yawen Chen
yawen.chen@otago.ac.nz

Lecture Schedule

Lecture 16	Views and Null
Lecture 17	Database Files and Storage
Lecture 18	Database Indexing (1)
Lecture 19	Database Indexing (2)
Lecture 20	Database Security and Auditing
Lecture 21	Transactions
Lecture 22	Concurrency Control
Lecture 23	Database Recovery
Lecture 24	Query Optimization
Lecture 25	Other Data Models
Lecture 26	Revision

Overview

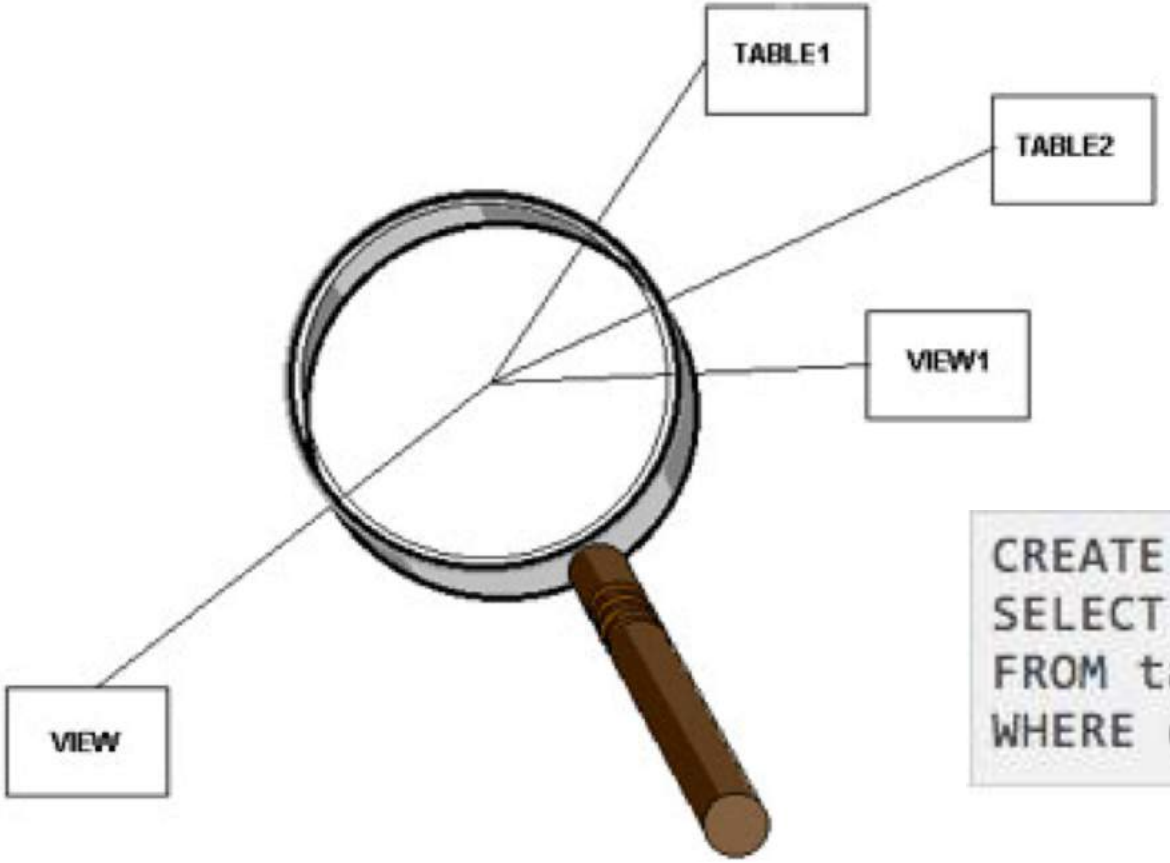
- Last Lecture
  - Triggers
- This Lecture
  - Views
  - NULL
  - Source: Chapter 5.3 Pages: 129-133
    Chapter 3.1 Pages: 61- 62
- Next Lecture
  - Transactions

Concept of a View

- Photo “views”: what are you interested?

Concept of a View

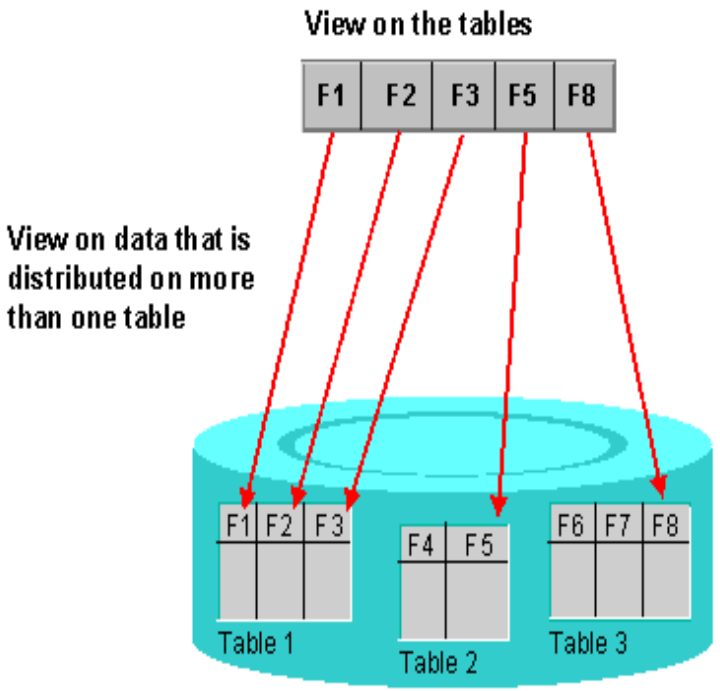
- Database “views”: what are you interested?




```
CREATE VIEW view_name AS
SELECT column_name(s)
FROM table_name
WHERE condition
```

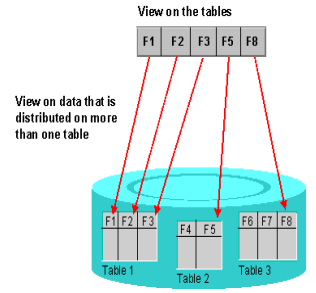
Concept of a View

- A **virtual table** derived from other base tables or previously defined views
- Does **not** necessarily exist in **physical forms**, in contrast to base tables
- Looks like, and in many ways acts like, a base table
  - No limitations on querying on a view
  - Possible limitations on updating



Why Need View

- Provides a powerful and flexible **security** and **access control** mechanism
  - Hide parts of the database from certain users (video)
- Provides a way to access data in a way customized to user needs.
- Provides a way to rename attributes or change the order of attributes.
- Simplify complex operations on the base tables



Specification of View in SQL

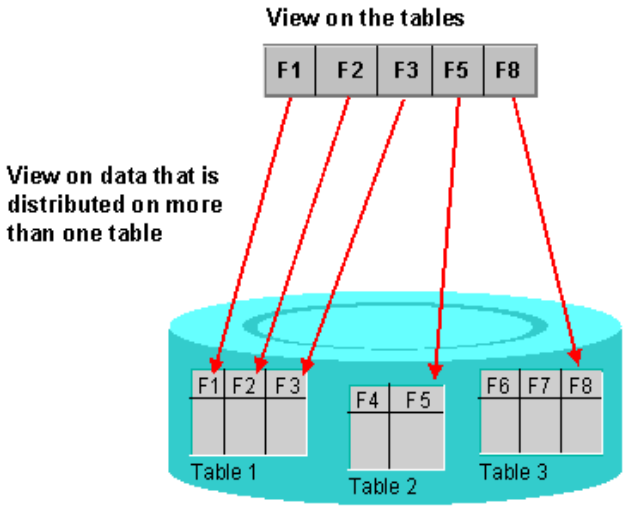
- Create view

```
CREATE VIEW <view name>
  AS <select statement>;
```

Can have an arbitrarily complex SELECT statement.

- Drop view

```
DROP VIEW <view name>;
```



View Example 1

- Customized to user needs.
- Rename attributes

```
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;
```

- Creates new attribute names
- 1:1 correspondence between view columns and select attributes

View Example 1 (cont.)

DEPT_NAME	NO_OF_EMPS	TOTAL_SAL
Administration	3	93000
Headquarters	1	55000
Research	4	133000

View Example 2

- Suppose we often want to access employees' names, the projects they work on, and the hours worked.

```
CREATE VIEW works_on1
  AS SELECT fname,lname,pname,hours
       FROM employee,project,
            works_on
      WHERE ssn = essn AND
            pno=pnumber;
```

3-table join

WORKS_ON1

FNAME	LNAME	PNAME	HOURS

Mapping

WORKS_ON1
FNAME
LNAME
PNAME
HOURS
Note: The "linking" attributes do not appear in the view
PROJECT
FNAME
LNAME
SSN
PNUMBER
PNAME
ESSN
PNO
HOURS
3-table join
EMPLOYEE
WORKS_ON

View Example (continued)

FNAME	LNAME	PNAME	HOURS
John	Smith	ProductX	32.5
John	Smith	ProductY	7.5
Franklin	Wong	ProductY	10.0
Franklin	Wong	ProductZ	10.0
Franklin	Wong	Computerisation	10.0
Franklin	Wong	Reorganisation	10.0
Alicia	Zelaya	Newbenefits	30.0
Alicia	Zelaya	Computerisation	10.0
Jennifer	Wallace	Newbenefits	20.0
Jennifer	Wallace	Reorganisation	15.0
Ramish	Narayan	ProductZ	40.0
Joyce	English	ProductX	20.0
Joyce	English	ProductY	20.0
Ahmad	Jabbar	Computerisation	35.0
Ahmad	Jabbar	Newbenefits	5.0
James	Borg	Reorganisation	NULL

View Example (continued)

- Query on a view

```
SELECT fname,lname
FROM WORKS_ON1
WHERE Pname='ProductX'
```

Implementation of Views

- Views are maintained by the DBMS
- Always up to date with the underlying base tables
- Efficient implementation is complex
  - Query modification (transform view to temporary query)
    - modify or transform the view query into a query on the underlying base tables
  - View materialisation (transform view to temporary table)
    - Physically create a temporary view table when the view is first queried
    - Incremental update is used to keep the view table updated with the underlying base tables.

Query Modification


```
CREATE VIEW works_on1
AS SELECT fname,lname,pname,hours
   FROM employee,project,works_on
   WHERE ssn = essn AND
         pno=pnumber;
```

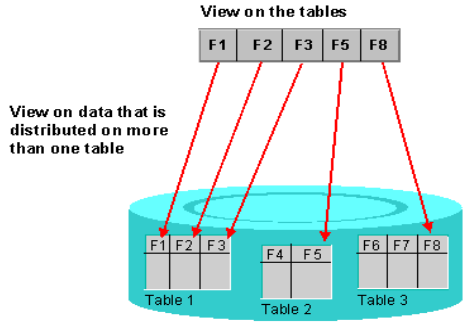
```
SELECT fname,lname
FROM WORKS_ON1
WHERE Pname='ProductX'
```

```
SELECT fname,lname
FROM employee,
     project,WORKS_ON
WHERE ssn = essn and
      pno = pnumber
      Pname='ProductX'
```

- It is inefficient for
  - views defined via complex queries that are time-consuming to execute
  - Multiple queries are going to be applied to the same view within a short period.

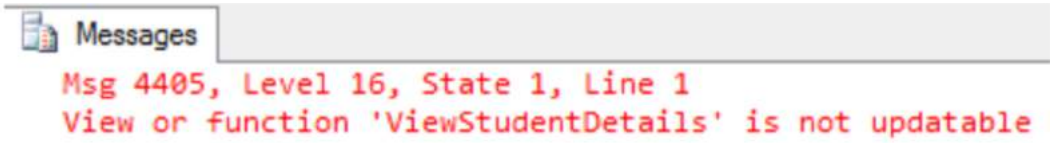
View Materialization

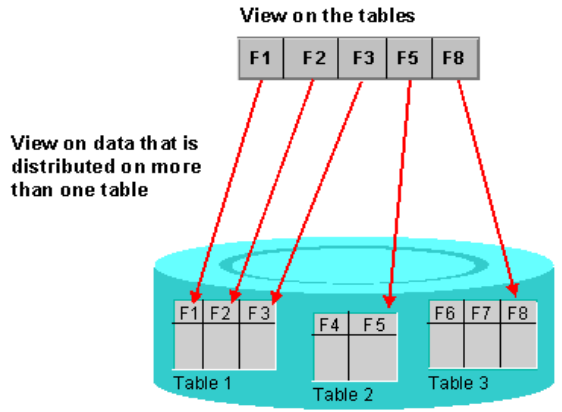
- Create a physical temporary view table when the view is first queried
- Using incremental update, DBMS determines what tuples should be inserted, deleted or modified in the materialized view table.
- The view is kept as a materialized table as long as it is being queried.
- If the view is not queried for a certain period of time, the system may remove the physical table and recompute it from scratch when future queries reference the view.



Updating Views

- Complicated
- Can be ambiguous
- An update on a view defined on a single table without aggregate functions can be mapped onto the underlying table.
- An update on a view involving joins could be interpreted in multiple ways.
- It is not often/not possible for the DBMS to determine which of the updates is intended.





View Update Example

```
UPDATE works_on1
  SET pname = 'ProductY'
    WHERE lname = 'Smith' AND
          fname = 'John' AND pname = 'ProductX';
```

FNAME	LNAME	PNAME	HOURS
John	Smith	ProductX	32.5
John	Smith	ProductY	7.5
Franklin	Wong	ProductY	10.0
Franklin	Wong	ProductZ	10.0
Franklin	Wong	Computerisation	10.0
Franklin	Wong	Reorganisation	10.0
Alicia	Zelaya	Newbenefits	30.0
Alicia	Zelaya	Computerisation	10.0
Jennifer	Wallace	Newbenefits	20.0
Jennifer	Wallace	Reorganisation	15.0
Ramish	Narayan	ProductZ	40.0
Joyce	English	ProductX	20.0
Joyce	English	ProductY	20.0
Ahmad	Jabbar	Computerisation	35.0
Ahmad	Jabbar	Newbenefits	5.0
James	Borg	Reorganisation	NULL

View Update Example (cont.)

- Map the update into several updates on the base relations
- Two possibilities:

```
First possibility:

UPDATE works_on
  SET pno =
      (SELECT pnumber FROM project
         WHERE pname='ProductY')
  WHERE essn IN
      (SELECT ssn FROM employee
         WHERE lname='Smith' AND
               fname='John')
    AND pno IN
      (SELECT pnumber FROM project
         WHERE pname='ProductX');
```

View Update Example (cont.)

Second possibility:

```
UPDATE project
   SET pname = 'ProductY'
     WHERE pname = 'ProductX';
```

Side effect of changing all the view tuples with pname=‘ProductX’

- Some view updates may not make much sense.

```
UPDATE Dept_info
     SET Total_sal = 100000
     WHERE dname = ’Research';
```

View Update Summary

- A view with a single defining table is updateable if the view attributes contain the primary key (or some candidate key) of the base relation.
- Views defined on multiple tables using joins are generally not updateable.
- Views defined using grouping and aggregate functions are not updateable.
- A view update is feasible when only one possible update on the base relations can accomplish the desired update effect on the view.
- In SQL the clause WITH CHECK OPTION must be added at the end of the view definition if a view is to be updated.

http://www.dba-oracle.com/t_with_check_option.htm

NULL

- Represents a value for an attribute is currently unknown or is not applicable for this tuple.
- It is not a value, but represents the absence of a value
- Has no data type
- Cannot be *compared* as a value
- Often used as the default

Name	**Home Phone**	**Address**
Dan	479001	ertyu
Emma	479002	qwer
Frank	479002	qwer
Gina	479002	qwer
Hamish	479005	tyufgd
Ian	479006	ghjgjh
Jo	479006	ghjghj
Mike	NULL	iuwer

NULL in Predicates

- A predicate is an expression that can be true, false, or unknown (Null)

NOT

	NOT
true	false
false	true
unknown	

NULL in Predicates (continued)

AND

AND	true	false	unknown
true	true	false	
false	false	false	
unknown			

OR

OR	true	false	unknown
true	true	true	
false	true	false	
unknown			

False < Unknown < True

AND: low OR: high

Thoughts on NULL

- NULL is not a value
- It cannot exist in a domain, which is defined as a set of values
- This wouldn't matter except that purists insist that the relational model is based on rigorous underlying theory
- Hugh Darwen and Chris Date were at the centre of this debate in the 1980s and 1990s, It's still going on