Computer Science Department
Michigan State University
CSE480 Database Systems, Spring 2003
Lab Week #7 Relational Views and Materialized Views

You will define, use and update views in this lab.

(60 minutes)

1. Create a view named EastLansingStudent by running the following in sqlPlus:

   create or replace view EastLansingStudent(Id, Name)
   AS Select Sid, Sname
   From Student
   Where CurAddress='East Lansing';

Now you may run queries on EastLansingStudent view.
Run the following queries on the above view:
(a) Select * from EastLansingStudent;
(b) Select Id
    From EastLansingStudent
    Where Name='Susan';
(c) Check the answer of the second query to see if it is included in the first query.
(d) View can be considered a virtual table with no data in it. Where does the data come from when a query is run on a view?
(e) When Oracle runs an sql statement on a view, it first transforms this sql statement into an equivalent sql statement on the base tables, and then runs this transformed sql statement to retrieve the answer. Following is a transformed sql
for query (b) above. Run this transformed sql statement. Compare the answer of this run with that of the above.

Select Sid
From Student
Where Sname='Susan' and CurAddress='East Lansing';

(f) Now run the following:
   insert into EastLansingStudent(Id, Name)
   values('A000', 'Scott');

(g) Select * from EastLansingStudent;
   Do you see the tuple you have just inserted?

(h) Select * from Student;
   Do you see the tuple you have just inserted?

(i) You have to be careful about the semantic of the view when you insert tuples into a view. You rather not allow insert into this view.

(j) Change the above view definition of EastLansingStudent to make it read only type as follows:
   create or replace type EastLansingStudent
   AS Select Sid, Sname
   From Student
   Where CurAddress='East Lansing'
   With Read Only;
   run the following queries:
     i. select * from EastLansingStudent;
     ii. Insert into EastLansingStudent(Sid, Sname)
          values('A000', 'Scott');
     iii. Are you able to insert the tuple?

2. Now create another view:

Create or Replace View GoodStudents (Sid, Degree, ProgramName, GPA)
AS Select s.Sid, ProgType, ProgramName, AVG(Grade)
From DegreeProgram dp, Student s, taken t
Where dp.ProgId=s.Major and s.Sid=t.Sid
Group By s.Sid, ProgType, ProgramName
Having AVG(Grade)>3.0;

(a) What is the semantic of the above view?
(b) Run select * from GoodStudents;
(c) Run the following query which works on the above view, GoodStudents:

Select *
From GoodStudents
Where Degree='BS';

(d) Check to see if the result of this select is included in the result of the previous select.
(e) Design the transformed sql statement for the above sql statement, which runs directly on the base tables. Run this sql statement.
(f) Check your answer by comparing the result of this transformed sql statement with that of the above sql statement.
(g) Run the following to insert a tuple into this view:

Insert into GoodStudents(Sid) values('A0000');

Why are you not able to insert the tuple?

3. Create the following view:

create or Replace view StudentCourse(Sid, Sname, Cno, Grade)
AS Select s.Sid, Sname, cno, Grade
From Student s, taken t, CourseOffering co
Where s.Sid=t.Sid and t.SeqId=co.SeqId;

(a) Run the following queries on this view:

i. Select * from StudentCourse

ii. Select Cno, COUNT(*), MAX(Grade), MIN(Grade), AVG(Grade)
    From StudentCourse
    Where cno='CSE480'
    Group By Cno
(b) Give the transformed sql statement for the second sql above that runs on base tables.

(c) Run this transformed sql and check the answer by comparing it with that of the views.

4. Create the following view:

```sql
Create or replace view CSEfaculty(FacSSNo, FacName, DeptName)
AS Select FacSSNo, FacName, DeptName
From FAulty f, Department d
Where f.WorksFor=d.DeptId;
```

(a) Run the following queries:
   i. Select * From CSEfaculty;
   ii. Select *
       From CSEfaculty
       where DeptName Like '%Computer%';

(b) Check if the answer of the second sql statement is properly included in the result of the first.

(c) Run the following two sql INSERTS, one into StudentCourse and the other into the CSEfaculty views. Indicate, why it works for the second but not for the first.

```sql
insert into StudentCourse(Sid, Sname) values('A0001', 'Mary');

insert into CSEfaculty(FacSSNo, FacName) values('000000000', 'Jim');
```

5. Create a view for East Lansing good students by defining it on top of the two views EastLansingStudents and GoodStudents.

(15 minutes)

6. Now you design and create the following two views:

(a) Only those students in CSE 480 who belong to CSE department.
CSE480CSE

Students(Sid, Sname, SeqId, Semester, 
Year, Instructor, Grade)

(b) All courses with SeqId, Cno, Csemester, Year, Instructor, 
Text books and Prerequisites.

AllCourses(SeqId, Cno, Semester, Year, Instructorname, 
TextBook_name, Prerequisite_course_numbers)

For this view you have to create three additional tables indi- 
cated below.

You do not have to type for this part. You can cut and 
paste from the text file that I have provided for this part 
on this lab web page

Implement the entity and relationship types **Textbook, has, 
prerequisite** of the EER diagram of lab 5 (See the EER dia- 
gram of lab 5 available on the web). Insert tuples, given below, 
into these tables using sql INSERT.

drop table TextBook cascade constraints;
Create table Prerequisite(
  Cno       varchar2(10),
Pcno       varchar2(10),
  primary key(cno, PCno),
foreign key (Cno) references CourseDescription(Cno),
foreign key (PCno) references CourseDescription(Cno));

Create table TextBook(
  TextId   varchar2(10) primary key,
  TextTitle varchar2(50),
  TextAuth  varchar2(30));

Create table CourseText(
  SeqId     varchar2(5),
  TextId    varchar2(10),
  primary key(SeqId, TextId),
foreign key (SeqId) references CourseOffering(SeqId),
foreign key (TextId) references TextBook(TextId));
Now insert the following tuples into the tables.

<table>
<thead>
<tr>
<th>Prerequisite</th>
<th>CourseText</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chn</td>
<td>PreReqChn</td>
</tr>
<tr>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td>CSE480</td>
<td>CSE410</td>
</tr>
<tr>
<td>CSE480</td>
<td>CSE331</td>
</tr>
<tr>
<td>CSE880</td>
<td>CSE480</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TextBook</th>
</tr>
</thead>
<tbody>
<tr>
<td>textId</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>t0001</td>
</tr>
<tr>
<td>t0005</td>
</tr>
<tr>
<td>t0008</td>
</tr>
</tbody>
</table>

(30 minutes)

7. Materialized Views:

We will use materialized views to create a replicated database (or copy of tables) placed locally. The idea is to efficiently query a database from a remote place by making a local copy of the data and then performing queries on the local copy. Thus materialized views, kept locally, will implement such queries.

(a) Create and populate the table
TextBooks(TextId, TextName, TextAuth, Publishers, Type) in remote database cse480 as follows:
setenv ORACLE_SID cse4802
setenv TWO_TASK cse4802

sqlplus

Create table TextBooks(
    TextId   number(10) primary key,
    TextName  varchar2(30),
    TextAuth  varchar2(30),
    Publishers varchar2(30),
    Type      varchar2(5);

    /* populate ... */

    quit
(b) Create a materialized view RemoteTextBooks(TextId,TextName,TextAuth) in the local database cse480 using the base table, TextBooks, in cse4802 database as follows:
setenv ORACLE_SID cse480
setenv TWO_TASK cse480

sqlplus

CREATE DATABASE LINK cse4802 CONNECT TO YOUR-USERNAME IDENTIFIED BY YOUR-PID USING 'cse4802';

create materialized view RemoteTextBooks
    as select TextId, TextName, TextAuth
    from TextBooks@cse4802;

(c) Now query the view RemoteTextBooks in cse480 database.
    select * from RemoteTextBooks;

    Add a new tuple in the remote table TextBooks and run the above query on the materialized view again to see if the query result includes this tuple.

(d) Now you implement a regular view (not a materialized view) RemoteTextbooks as follows:
Create or replace view RemoteTextbooks
    as select TextId, TextName, TextAuth
    from TextBooks@cse4802;

Run the same query as above on this view. Do you see this query running faster than when you ran it on the materialized view?

You may not see any difference in response time because cse4802 database is mounted on the same machine as the cse480 database