Tutorial 10 (SC2207-CZ2007)

SQL & XML

Classroom Exercise

1. Consider the following relation:

Articles (ID, title, journal, issue, year, startpage, endpage, TR-ID)

It contains information on articles published in scientific journals. Each article has a unique ID, a title, and information on where to find it (name of journal, what issue, and on which pages). Also, if results of an article previously appeared in a "technical report" (TR), the ID of this technical report can be specified. We have the following information on the attributes:

- For each journal, an issue with a given number is published in a single year.
- The endpage of an article is never smaller than the startpage.
- There is never (part of) more than one article on a single page.

Consider the following six queries on Articles:

a. SELECT title FROM Articles WHERE year=2005;

b. SELECT title FROM Articles WHERE endpage=100;

c. SELECT title FROM Articles WHERE year>1995 AND year<2000;

d. SELECT title FROM Articles WHERE journal='JACM' AND issue=55;

e. SELECT title FROM Articles WHERE issue=55 AND journal='JACM';

f. SELECT title FROM Articles WHERE endpage-startpage>50;

Indicate which of the above queries would likely be faster (based on the knowledge you have from the course), if all of the following indexes were created.

- CREATE INDEX Idx1 ON Articles (year, startpage);
- CREATE INDEX Idx2 ON Articles (startpage, endpage);
- CREATE INDEX Idx3 ON Articles (journal, issue, year);

2 (i) Consider relation STUDENT (id, name, age, address). Build an index for relation STUDENT and give an example query whose execution can be sped up by the index.

(ii) Consider relation CUSTOMER (NRIC, name, phone, address, email, sex, age). Build an index for relation CUSTOMER to best speed up the following query and explain your answer:

```
SELECT name, phone FROM CUSTOMER WHERE age > 30 AND sex = ‘male’
```

3. Consider the following XML DTD for an employee database.

```
<!DOCTYPE emp [
<!ELEMENT emp (ename, children*, skills*)>
<!ELEMENT children (name, birthday)>
<!ELEMENT birthday (day, month, year)>
<!ELEMENT skills (type, exams+)>
<!ELEMENT exams (year, city)>
<!ELEMENT ename( #PCDATA )>
<!ELEMENT name( #PCDATA )>
<!ELEMENT day( #PCDATA )>
<!ELEMENT month( #PCDATA )>
<!ELEMENT year( #PCDATA )>
<!ELEMENT type( #PCDATA )>
<!ELEMENT city( #PCDATA )>
] >
```

Create a valid XML document that follows the rules of the above DTD.

Your document must illustrate all the rules of the DTD.

4. Consider the following relational database schema about students and courses (primary keys are underlined):

STUDENT(SID, NAME, EMAIL)
COURSE(CID, NAME, INSTRUCTOR, ROOM)
ENROLLS(SID, CID, GRADE)

In ENROLLS, SID and CID are foreign keys into STUDENT and COURSE, respectively. The following two constraints also hold: (a) each student is enrolled in some course, and (b) each course has at least one student enrolled.

(I) Design a DTD for exporting this data as an XML view to the chair’s office. The chair wants to see the data grouped by courses and needs to have access to all the information in the database.

(ii) Give an example of a database instance for the relational schema and show the resulting document for the XML view according to the DTD that you have designed.