SE 344
Section 10 worksheet

1.

XML/XPath/XQuery

(25 points)

Consider an XML document that contains data about books. Consider the following two XPath queries on this XML data:

```
P1 = /bib/book[@year<2005][editor/last/text()='Samet'][price < 99]/title/text()
P2 = /bib/book[author/last/text()='Hull'][author/last/text()='Vianu']/title/text()
```

In this problem you are asked to design a relational schema for this data, then translate the two XPath queries into SQL over that schema.

(a) (10 points) Assume that the XML data conforms to the following DTD:

```
<!DOCTYPE bib [
<!ELEMENT bib  (book* )>
<!ELEMENT book  (title,  (author+ | editor+ ), publisher?, price )>
<!ATTLIST book  year CDATA  #REQUIRED >
<!ELEMENT author  (last, first )>
<!ELEMENT editor  (last, first, affiliation )>
<!ELEMENT title  (#PCDATA )>
<!ELEMENT last  (#PCDATA )>
<!ELEMENT first  (#PCDATA )>
<!ELEMENT affiliation  (#PCDATA )>
<!ELEMENT publisher  (#PCDATA )>
<!ELEMENT price  (#PCDATA )>
]>
```

1. Design a relational schema for the XML data. Your schema should have relations corresponding to entity sets such as `Book`, `Author`, etc., as well as relationships between these entity sets. Write only the relation names and their columns; for example, `Author(aid, last, first)`; do not write the field types nor the key/foreign key constraints.
   Answer (write a relational schema):

2. Translate the two XPath queries P1 and P2 into SQL queries over your relational schema.

```
P1 = /bib/book[@year<2005][editor/last/text()='Samet'][price < 99]/title/text()
P2 = /bib/book[author/last/text()='Hull'][author/last/text()='Vianu']/title/text()
```

Answer (write two SQL queries):

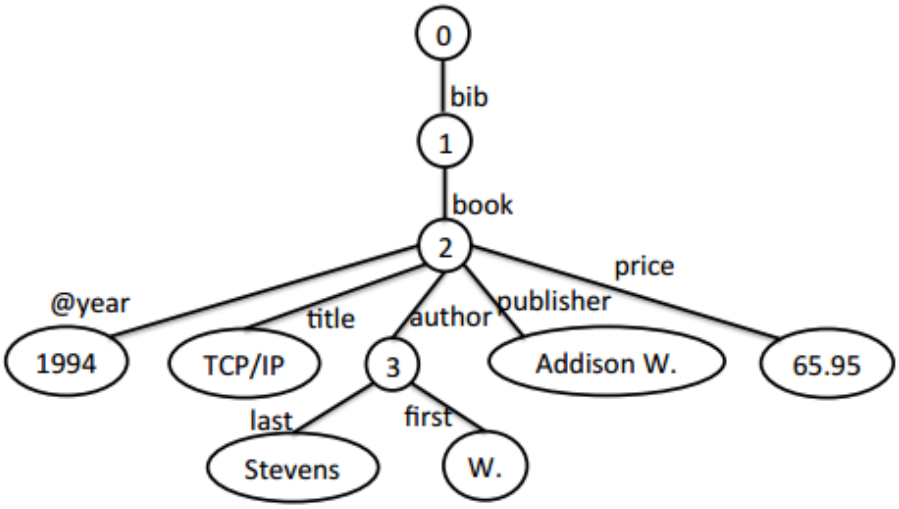
(b) (10 points) Next, assume that no DTD is given, and that we do not know anything about the structure of the XML document. In this case we store it in generic table:

```
Element(id, tag, child)
```

`id` is a node identifier, `tag` is a tag (or attributes) in that node, `child` is the identifier, or the content of the child. For example, if the document were:

```
<bib> <book year="1994">
        <title>TCP/IP Illustrated</title>
        <author><last>Stevens</last><first>W.</first></author>
        <publisher>Addison-Wesley</publisher>
        <price>65.95</price>
      </book>
</bib>
```

Then its tree representation, and tabular representation are:



Element:

id	tag	child
0	bib	1
1	book	2
2	@year	1994
2	title	TCP/IP
2	author	3
2	publisher	Addison Wesley
2	price	65.95
3	last	Stevens
3	first	W.

The root node has always identifier 0, but all others can be arbitrary.

Translate the two XPath queries P1 and P2 into SQL over the table `Element`.

For example, if the XPath query were:

```
P0 = /bib/*/title/text()
```

then your SQL query would be:

```
select z.child
from Element x, Element y, Element z
where x.id = 0 and x.tag = 'bib'
  and x.child = y.id
  and y.child = z.id and z.tag = 'title'
```

Answer write two SQL queries for P1, P2:

```
P1 = /bib/book[@year<2005][editor/last/text()='Samet'][price < 99]/title/text()
P2 = /bib/book[author/last/text()='Hull'][author/last/text()='Vianu']/title/text()
```

Conceptual Design, Constraints, Views

1.

(35 points)

(a) (10 points) Consider a relation $R(A, B, C, D, E)$ that satisfies the following functional dependencies:

$$ABC \rightarrow D$$
$$E \rightarrow B$$
$$AD \rightarrow C$$

Decompose the schema in BCNF. Show all your steps.

Answer (Show the steps leading to the BCNF decomposition):

(b) (10 points) Consider the table below:

A	B	C
a_1	b_1	c_1
a_1	b_2	c_2
a_2	b_3	c_1
a_2	b_3	c_2

For each of the functional dependencies listed below, indicate whether it holds or not. If it holds, write OK. If it does not hold, indicate two tuples in the table above that violate the functional dependency. Refer to the tuples as 1,2,3,4; for example, you may say that $A \rightarrow C$ fails because of the tuples 3,4.

FD	Holds ?
$B \rightarrow A$	
$C \rightarrow A$	
$A \rightarrow B$	
$C \rightarrow B$	
$A \rightarrow C$	
$B \rightarrow C$	
$BC \rightarrow A$	
$AC \rightarrow B$	
$AB \rightarrow C$	

2.

(20 points)

(a) (10 points) Design an E/R diagram describing the following domain:

- A **Person** has attributes **pid** (key) and **name**.
- A **Skier** is a type of **Person** with attribute **aptitude**.
- A **Snowboarder** is a type of **Skier**.
- A **PairOfSkis** has attribute **sid** (key) and **model**.
- A **Snowboard** has attribute **sid** (key) and **model**.
- A Skier **owns** zero or more PairOfSkis. The ownership relation has a **purchase_price**. A PairOfSkis is owned by at most one Skier.
- A Snowboarder **owns** zero or more Snowboards. The ownership relation has a **purchase_price**. A Snowboard is owned by at most one Snowboarder.
- a Person can **rent** a PairOfSkis or a Snowboard. A person cannot rent more than one PairOfSkis or one Snowboard at the same time. A person cannot rent a PairOfSkis and a Snowboard at the same time either. A piece of equipment can be rented by at most one person at a time. The rental comes with a **start_date** and an **end_date**.

Answer (Draw an E/R Diagram):

(b) (10 points) Write the SQL `CREATE TABLE` statement for the **owns** relation between Skier and PairOfSkis. Make sure that your statement specifies the PRIMARY KEY and any FOREIGN KEYS. Additionally, we would like to enforce the constraint that `purchase_price` be greater than zero.

Answer (Write a CREATE TABLE statement):

3.

(15 points)

(a) (5 points) Consider the following relational schema and set of functional dependencies. List all superkey(s) for this relation. Which of these superkeys form a key (i.e., a minimal superkey) for this relation? Justify your answer in terms of functional dependencies and closures.

R(A,B,C,D,E) with functional dependencies AB $\rightarrow$ E and D $\rightarrow$ C.

Answer (Find all the superkeys and keys):

(b) (10 points) Decompose R into BCNF. Show your work for partial credit. Your answer should consist of a list of table names and attributes and an indication of the keys in each table (underlined attributes).

Answer (Decompose R into BCNF):

4. (30 points)

(a) (10 points) We have a large database, on which we need to run repeatedly SQL queries. Each SQL query has up to 5 joins, a group-by, and some selections. We use a parallel database system, and we consider the following alternative evaluation strategies: (a) inter-query parallelism, (b) inter-operator parallelism, (c) intra-operator parallelism. In each case we deploy only one strategy, i.e. we do not combine them. Consider a job J consisting of several SQL queries, and assume it has the following running time:

- Job J runs in $T = 100$ minutes on 10 nodes.

Estimate the running time of that job if we increase the number of nodes from 10 to 100, in each of the six cases below. In each case, assume that the database is capable of delivering linear speedup, when the execution strategy is parallelizable.

Write the running time T on 100 nodes			
Job J consists of:	Type of Parallelism		
	Inter-query	Inter-operator	Intra-operator
1 SQL query			
1000 SQL queries			

(b) (10 points) The query below computes the total number of customers with any given date of birth:

```
select birthdate, count(*)
from Customer x
group by x.birthdate
```

The attributes `birthdate` represents the date of birth of the customer: it contains the day and month only (not the year!). We evaluate the query using Map-Reduce. Assume:

- The relation `Customer` has 16MB

We consider choosing the block size to be one of the following three values: 128KB, 64KB, 32KB. (Recall that $1MB = 1024KB$; thus, if the block size is 128KB, then the `Customer` file has $16 \cdot 1024/128 = 128$ blocks.) Indicate the maximum number of instances that you can use and still achieve linear speedup, if the data is uniformly distributed. Assume that the number of map tasks is equal to the number of blocks, and that the number of reduce tasks is set to the number of instances.

i. The block size is 128KB.

i. _____

Maximum number of instances:

ii. The block size is 64KB.

ii. _____

Maximum number of instances:

iii. The block size is 32KB.

iii. _____

Maximum number of instances:

(c) (10 points) A Map/Reduce Job runs on 10 instances, and uses 100 Map Tasks and 50 Reduce Tasks. The input file has 5GB, and the block size is 50K. We assume that the map function produces an output whose size is approximatively equal to that of the input: in other words, the size of the intermediate result is also 5GB.

i. What is the total number of intermediate files to which the mappers write their outputs?

i. _____

Write the number of files:

ii. After a reducer copies, it needs to sort. How large is the file that needs to be sorted? Answer with the expected value.

ii. _____

Write the size of the file:

iii. At any time, one instance runs only one map task, or only one reduce task. Suppose that a map task takes 1 minute to finish, and a reduce task also takes 1 minute to finish. What is the total running time for the Map/Reduce job?

iii. _____

Write the time in minutes:

iv. Continue to assume that each map task and each reduce task takes 1 minute. However, one single map task exhibits a skew, and take 10 minutes to complete instead of 1 minute; all other 99 map tasks still take 1 minute. What is the total running time for the Map/Reduce job?

iv. _____

Write the time in minutes: