Sami Shamoon
College of Engineering (R.A)

המכללה האקדמית להנדסה
סמי שמעון (ע"ר)
מדור בחינות

המחלקה להנדסת תוכנה

05/02/08
09:00-12:00

בסיסי נתונים
מועד א'
ד"ר יוסף שפונגין

חומר עזר – אסור

השאלון מכיל 7 דפים (כולל דף זה). **בהצלחה !**

==========

.1

(9) (א) נתונים ה-Interfaces הבאים.

```
1) interface Int_A (key (x)){
2) attribute string x;
3) attribute List<Struct S {string y, string z}>lt;
4) relationship Set<Int_B> R_A_B
        inverse Int_B :: R_B_A;
5) relationship Int_C R_A_C
        inverse Int_C :: R_C_A;
   };

1) interface Int_B (key (u)) {
2) attribute string u;
3) attribute Set<Struct S {string v, string w}>st;
4) relationship Int_A R_B_A
        inverse Int_A :: R_A_B;
5) relationship Int_C R_B_C
        inverse Int_C :: R_C_B;
   };

1) interface Int_C (key (k)) {
2) attribute string k;
3) attribute Struct S {string v, string w};
4) relationship Int_A R_C_A
        inverse Int_A :: R_A_C;
5) relationship Int_B R_C_B
        inverse Int_A :: R_B_C;
   };
```

(1) כתוב את ה-RM (Relational Model) המתאים
ל-Interfaces הנ"ל. ציין את ה-keys של הטבלאות.
(2) בנה את ה- E/R דיאגראמה שלפי המשמעות (objects, relationships) יהיה
מתאימה ל- interfaces הנ"ל.
(3) כתוב את ה-RM המתאים ל-E/R שבנית ב-(2).

(8) (ב) ב- R(A, B, C, D, E) relation מתקיימים ה- FD's הבאים:

$$AC \to B, AB \to C, BC \to A, CD \to E, CE \to D, ED \to C$$

(1) מצא את כל ה- keys עבור R.
(2) על סמך ה-FD`s הנתונים כתוב את הבסיס המינימאלי המתאים
(minimal base)
(3) האם R נמצא ב- BCNF? אם לא, פרק אותו לפי BCNF.
(4) האם R נמצא ב- 3NF? אם לא, פרק אותו לפי 3NF.

(8) (ג) (1) נניח כי נתונה טבלא R(A, B, C) שמכילה 3 שורות. נתון עוד כי $A \to\to B$.
האם נכון כי מתקיים $A \to B$? אם כן, הוכח זאת. אם לא, תן דוגמא נגדית.
(2) (בלי קשר ל-(1)) בטבלה R(A, B, C) נתון כי $A \to B$.
האם נכון כי מתקיים: $A \to\to C$?
אם כן, הוכח זאת לפי הגררה של MD. אם לא, תן דוגמא נגדית.
(3) נתון כי $A \to\to B \cup C$ וגם $A \to\to B$. האם נכון כי $A \to\to C$?
אם כן, הוכח זאת לפי הגררה של MD. אם לא, תן דוגמא נגדית.

2.

נתונה הסכמה הבאה (Relational Database Schema):

Books(bID, author, cat)
Library(lID, country, city)
Lib-Books(lID, bID, lnum)
Store(sID, country, city)
Store_Lib_Books(sID, lID, bID, year, snum)
Store_Lib(sID, lID, year, num_total)

נסביר את משמעות ה-attributes:

bID- ID של book;
author- מחבר;
cat (category) - סוג של ספר (מתמטיקה, מחשבים וכו');
lID- ID של ספרייה;
lnum- מספר של עותקים;
sID- ID של חנות ספרים;
snum- מס' ספרים bID שמכרה חנות sID לספרייה lID בשנה year;
num_total- מס' ספרים סה"כ שמכרה חנות sID לספרייה lID בשנה year.
הערה: lnum לא בהכרח שווה לסכום של כל הקניות של הספרייה.

(13) (א) מצא ב-RA את כל ה-stores כך, שלכל אחד מהם לא הייתה שנה שה-store לא מכר
ל-library אחד כלשהו, ספרים מכל הסוגים (category) האפשריים.
(12) (ב) מצא ב- DATALOG את כל ה-cities כך, שבכל אחד מהם יש לפחות 2 ספריות
שכל אחת מהן קונה ספרים רק מהחנויות של אותו country שבו היא נמצאת.

(10) (ג) כתוב ב-RA את ה-Constraint שמקיים את התנאי הבא: בכל city קיים store שמוכר ספרים גם לחו"ל (ז"א למדינות שהוא לא נמצא בהן).

.3

נניח כי בסיס נתונים מוגדר על-ידי 6 טבלאות משאלה 2.

(10) (א) מצא בשאילתה אחת, את כל הספריות כך, שכל אחת מהן קונה בכל שנה מספר מקסימאלי של ספרים (באותה שנה) בין כל הספריות.

(10) (ב) נניח כי הספריות "העשירות" החליטו לתרום ספרים למוסדות שונים. הספרייה היא עשירה אם יש בה ספרים מכל סוג (category) שמופיע בכל ספרייה מאותו country. הוחלט מכל ספרייה עשירה לקחת 10% ממספר עותקים של כל ספר שנמצא בספרייה בכמות מקסימאלית (ז"א להקטין lnum ב-10%). כתוב את השאילתא המתאימה.

.4

(10) נתון DTD הבא:

```
<!DOCTYPE Company [
<!ELEMENT Company (Department+|Person+)>
<!ATTLIST Company Name CDATA #REQUIRED>
  <!ELEMENT Department (SubDepartment+|Person+)>
  <!ATTLIST Department Name CDATA #REQUIRED>
    <!ELEMENT SubDepartment (#PCDATA)>
    <!ELEMENT Person EMPTY>
    <!ATTLIST Person
      Name CDATA #REQUIRED
      Address CDATA #REQUIRED
      Tel_Num CDATA #IMPLIED>
]>
```

(א) תן דוגמא ל-XML-File המתאים.

(ב) מהו הפלט של ה-XPathes הבאים (על סמך ה-XML-File שבנוי בסעיף א')?

(1) //preceeding::Person

(2) /Company/SubDepartment/ancestor::*

.5

(5) (א) נניח כי hash function מייצרת 8 ביטים. תכניס ל- extensible hash table את הערכים הבאים (משמאל לימין):

00110000, 00010000, 11110000, 11110001, 11000000, 11100000, 10110000, 11101000

כל block יכול להכיל 3 רשומות. הסבר את כל השלבים.

(5) (ב)

(1) בנה B-tree עבור נתונים הבאים:20 ,20 ,20 ,10 ,5 ,5 ,5 ,5 ,5 ,5 ,5 ,5 ,5 , 5 (14 נתונים). כל קודקוד יכול להכיל מקסימום 3 keys.

(2) הסבר מפורט איך מתבצע חיפוש של כל רשומות עם key=3.

(3) הסבר מפורט איך מתבצע חיפוש של כל רשומות עם key=6

בהצלחה!

1. ODL

(a) Example

```
Interface Star (key name) {
attribute string name;
attribute Struct Addr {string street, string city} address;
relationship Set<Movie> starredIn
        inverse Movie::stars; };
```

(b) Types in ODL Atomic: integer, float, character, string, boolean, enumeration.
Complex (T denotes the type) : Set<T>, Bag<T>, List<T>,
Array<T,i>, Struct N {T_1, F_1, ...,T_n, F_n}.

2. FD Rules about FD's.

(a) The Splitting/Combining Rule;
(b) An FD $A_1A_2...A_n \rightarrow B_1B_2...B_m$ is Trivial if the B's are a subset of the A's;
(c) The above FD is Nontrivial if at least one of the B's is not among the A's;
(d) The above FD is Completely nontrivial if none of the B's is also one of the A's;
(e) The above FD is equivalent to $A_1A_2...A_n \rightarrow C_1C_2...C_k$, where the C's are all those B's that are not also A's;
(f) The Transitive Rule
(g) If $\{B_1B_2,...,B_m\} \subseteq \{A_1A_2,...,A_n\}$, then $A_1A_2...A_n \rightarrow B_1B_2...B_m$;
(h) If $A_1A_2...A_n \rightarrow B_1B_2...B_m$, then $A_1A_2...A_nC_1...C_k \rightarrow B_1B_2...B_mC_1...C_k$ for any set of C's

3. MD Rules about MD's.

(a) The Trivial Rule: If $A_1A_2...A_n \rightarrow\rightarrow B_1B_2...B_m$ then $A_1A_2...A_n \rightarrow C_1C_2...C_k$, where the C's are the B's plus one or more of the A's;
(b) Combining Rule;
(c) If $A_1A_2...A_n \rightarrow\rightarrow B_1B_2...B_m$ is a MD for relation R, then also $A_1A_2...A_n \rightarrow C_1C_2...C_k$, where the C's are all attributes of R not among the A's and B's;
(d) If X, Y and Z are sets of attributes, $X \rightarrow\rightarrow Y$ and $X \rightarrow\rightarrow Z$, then $X \rightarrow\rightarrow Y \cap Z$;
(e) If X, Y and Z are sets of attributes, $X \rightarrow\rightarrow Y$ and $X \rightarrow\rightarrow Z$, then $X \rightarrow\rightarrow (Y - Z)$;
(f) If $Y \subseteq X$, then $X \rightarrow\rightarrow Y$;
(g) If $X \cup Y = R$, then $X \rightarrow\rightarrow Y$;
(h) If $X \rightarrow\rightarrow Y$, then $X \rightarrow\rightarrow Y - X$;
(i) The Transitive Rule.

4. RA

(a) Set Operations: $R \cup S, R \cap S, R - S$.
(b) Projection: $\pi_{A_1,A_2,...,A_n}(R)$
(c) Selection: $\sigma_C(R)$, where C - condition
(d) Cartesian Product: $R \times S$
(e) Natural Join: $R \bowtie S$
(f) Theta-Join: $R \bowtie_C S$, where C – condition
(g) Renaming: $\rho_{S(A_1,...,A_n)}(R)$

5. Datalog

Example: LongMovie(t,y) ← Movie(t,y,l,c,s,p) AND l ≥ 100

6. SQL

(a) SELECT Example:

```
SELECT Star1.name, Star2.name
FROM MovieStar AS Star1, MovieStar AS Star2
WHERE Star1.address = Star2.address
      AND Star1.name < Star2.name;
```

(b) UNION, EXCEPT, INTERSECT

Example:

```
(SELECT name, address FROM MovieStar)
    INTERSECT
(SELECT name, address FROM MovieExec);
```

(c) Subqueries

Example:

```
SELECT name FROM MovieExec
WHERE cert =
      (SELECT producer FROM Movie
       WHERE title = 'Star Wars' );
```

(d) Conditions Involving Relations

(1) EXISTS R is a condition that is true iff R is not empty;
(2) s IN R is true iff s is equal to one of the values in R;
(3) s > ALL R is true iff s is greater than every value in R;
(4) s > ANY R is true iff s is greater than at least one value in R;

(5) Example:

```
SELECT name FROM MovieExec
WHERE cert IN
    (SELECT producer FROM Movie
     WHERE (title, year) IN
         (SELECT movieTitle, movieYear FROM StarsIn
          WHERE starName = 'Harrison Ford')
     );
```

(e) Correlated Subqueries

Example:

```
SELECT title FROM Movie AS Old
WHERE year < ANY
    (SELECT year FROM Movie
     WHERE title = Old.title);
```

(f) Aggregation functions: SUM, AVG, MIN, MAX, COUNT.

Example:

```
SELECT COUNT(DISTINCT name)
FROM MovieExec;
```

(g) GROUP + HAVING

Example:

```
SELECT name, SUM(length)
FROM MovieExec, Movie
WHERE producer = cert
GROUP BY name
HAVING MIN(year) < 1930;
```

(h) INSERT

Example:

```
INSERT INTO Studio(name)
    SELECT DISTINCT studioName FROM Movie
    WHERE studioName NOT IN
        (SELECT name FROM Studio);
```

(i) DELETE

Example:

```
DELETE FROM StarsIn
WHERE movieTitle = 'The Maltese Falcon' AND
movieYear = 1942;
```

(j) UPDATE

Example:

```
UPDATE MovieExec
SET name = 'Pres. ' || name
WHERE cert IN (SELECT pres FROM Studio);
```

7. XML

Example:

```
<doc attr="ABC">
    <subdoc> TEXT</subdoc>
</doc>
```

8. DTD

(a) Example:

```
<!DOCTYPE NEWSPAPER [
<!ELEMENT NEWSPAPER (ARTICLE+)>
<!ELEMENT ARTICLE (HEADLINE, BODY)>
<!ELEMENT HEADLINE(#PCDATA)>
<!ELEMENT BODY (#PCDATA)>
<!ATTLIST ARTICLE AUTHOR CDATA #REQUIRED>
<!ATTLIST ARTICLE EDITOR CDATA #IMPLIED>
]>
```

(b) Number of occurrences:

```
<!ELEMENT element_name (child_name)>    - exactly one occurrence;
<!ELEMENT element_name (child_name+)>   - minimum one occurence;
<!ELEMENT element_name (child_name*)>   - zero or more occurrences;
<!ELEMENT element_name (child_name?)>   - zero or one occurrence.
```

(c) Declaring either/or content

Example: <!ELEMENT note(header, (message | body))>

(d) Attribute Types: PCDATA, CDATA, (en1|en2|...), ID, IDREF, IDREFS.

(e) Default attribute value can have:

value	The default value of the attribute
#REQUIRED	The attribute value must be included in the element
#IMPLIED	The attribute does not have to be included
#FIXED value	The attribute value is fixed

Example. DTD:

```
<!ELEMENT element_name EMPTY>
<!ATTLIST element_name attr CDATA "0">
```

Valid XML: <element_name attr="100"/>

9. Xpath

(a) Path Expressions

node_name	Selects all child nodes of the node
/	Selects from the root node
//	Selects nodes from the current node, no matter where they are
@	Selects attributes

(b) Predicates Examples:

```
/bookstore/book[1]
/bookstore/book[last()-1]
//title[@lang]
/bookstore/book[price>35.00]/title
```

(c) Unknown nodes

*	Matches any element node
@*	Matches any attribute
node()	Matches any node of any kind

Examples: /bookstore/*
//title[@*]

(d) Several paths Example: //book/title | //book/price

(e) XPath Axes

ancestor	Selects all ancestors (parent, grandparent, etc.) of the current node
attribute	Selects all attributes of the current node
child	Selects all children of the current node
parent	Selects the parent of the current node
descendant	Selects all descendants (children, grandchildren, etc.) of the current node
following	Selects everything in the document after the closing tag of the current node
preceding	Selects everything that is before the start tag of the current node
self	Selects the current node

Examples: //chapter[2]/preceding::*
/library/book[last()]/following::*

10. XQuery FLWOR Expression

Example:

```
for $x in doc("books.xml")/bookstore/book
where $x/price>30
order by $x/title
return $x/title
```

Example:

```
for $x in doc("books.xml")/bookstore/book
return if ($x/@category="Children")
then <child>{data($x/title)}</child>
else <adult>{data($x/title)}</adult>
```

11. B-Trees

The rules: (a) At the root, there are at least two used pointers;
(b) At a leaf, the last pointer points to the next block. Among the other pointers, at least $\left\lfloor \frac{n+1}{2} \right\rfloor$ pointers are point to data;
(c) At an interior node, at least $\left\lceil \frac{n+1}{2} \right\rceil$ pointers are actually used.

12. Linear Hash table

The rules: (a) The number of buckets n is always chosen so the average number of records per bucket is a fixed fraction;
(b) Overflow blocks are permitted;
(c) The number of bits used to number the entries of the bucket array is $\lceil \log_2 n \rceil$, where n is the current number of buckets.