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

Sami Shamoon
College of Engineering (R.A)

המחלקה להנדסת תוכנה

24/02/08
09:00-12:00

בסיסי נתונים
מועד ב'
ד"ר יוסף שפונגין

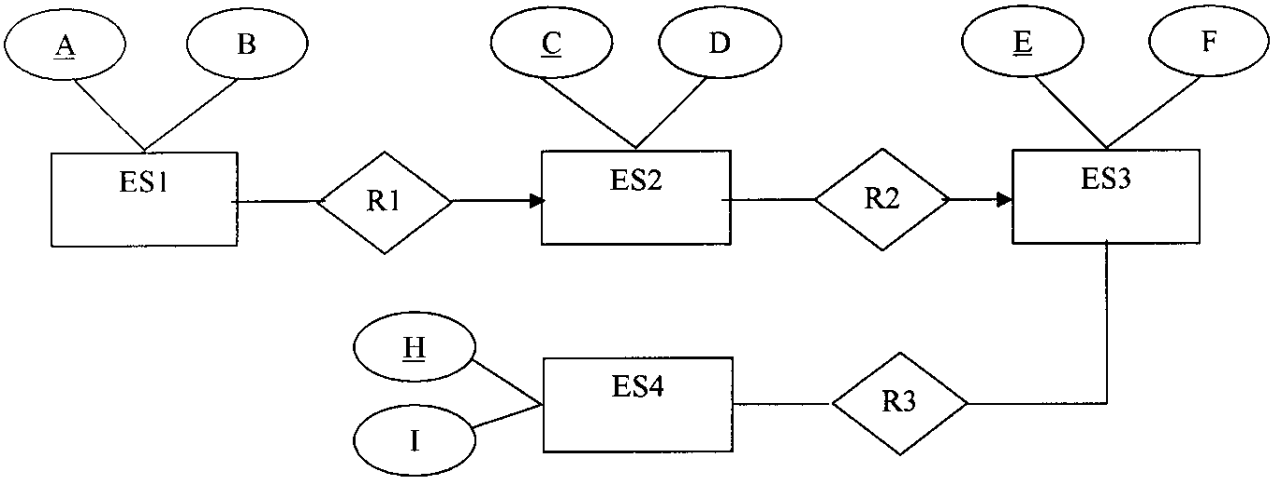
חומר עזר – אסור

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

==========

.1

(9) (א) נתונה E/R schema הבאה:



(1) כתוב את ה-RM (Relational Model) המתאים. ציין את ה-keys של הטבלאות.
(2) בנה את ה- Interfaces שלפי המשמעות יהיו מתאימים ל- E/R schema הנ"ל.
(3) כתוב את ה-RM המתאים ל-Inerfaces שבנית ב-(2). ציין את ה-keys של הטבלאות.

(8) (ב) (6) (ב) נתון relation:

A	B	C	D	E
1	2	3	1	1
2	1	4	5	1
1	3	3	1	1
2	2	4	5	1
3	2	2	1	1

(1) האם ב- relation קיימים MD's שהם לא FD's? הסבר
(2) האם ה-relation נמצא ב-4NF? אם לא, פרק אותו לפי 4NF. הסבר.
(3) האם ה-relation נמצא ב-BCNF? אם לא, פרק אותו לפי BCNF.
הסבר.
(4) האם ה-relation נמצא ב-3NF? אם לא, פרק אותו לפי 3NF. הסבר.

(8) (ג) ב-relation R(A, B, C, D) נתונות 2 עמודות- A ו- B:

A	B	C	D
1	2	*	*
2	3	*	*
1	2	*	*
2	3	*	*

(1) האם ניתן להשלים את הטבלה כך שיתקיים $C \rightarrow\rightarrow AB$ (MD), אך לא יתקיים $A \rightarrow D$ (FD). אם כן, השלם. הסבר.

(2) (בלי קשר ל-(1)) נתון כי $A \rightarrow\rightarrow B$ ו- $C \rightarrow\rightarrow B$. האם נכון כי מתקיים $AC \rightarrow\rightarrow B$? אם כן, הוכח זאת לפי ההגדרה של MD. אם לא, תן דוגמא נגדית.

(3) נתון כי $A \rightarrow\rightarrow B$ ו- $C \rightarrow\rightarrow D$. האם נכון כי $AC \rightarrow\rightarrow BD$? אם כן, הוכח זאת לפי ההגדרה של MD. אם לא, תן דוגמא נגדית.

2.

נתונה הסכמה הבאה (Relational Database Schema):

University(<u>uID</u>, <u>dep</u>)
Lib_of_Univ_Books(<u>uID</u>, <u>bID</u>, lnum)
Store_Lib_Books(<u>sID</u>, <u>uID</u>, <u>bID</u>, snum)
Student_Course(<u>stID</u>, uID, dep, <u>course</u>)
Books_Dep_Courses(<u>conum</u>, dep, <u>rbID</u>)
Student_Books(<u>studID</u>, <u>bID</u>)

נסביר את משמעות ה-attributes:
uID- ID של university;
dep (department)- מחלקה;
bID- ID של book;
lnum- מספר של עותקים;
sID- ID של חנות ספרים;
snum- מס' ספרים bID שמכרה חנות sID לספרייה של uID;
studID, stID- ID של סטודנט;
conum, course- מספר קורס;
rbid (recommended)- ספר שמומלץ לקורס.
<u>**הערה:**</u> לא בהכרח את כל הספרים המומלצים נמצאים בכל ספרייה.

(13) (א) מצא ב-RA את כל המחלקות (מכל האוניברסיטאות) כך, שבכל אחת מהן כל אחד מהסטודנטים לקח לפחות ספר אחד, עבור כל קורס שהוא לומד.
(12) (ב) מצא ב- DATALOG את כל החנויות כך, שכל אחת מהן מכרה את כל הספרים המומלצים לאוניברסיטה אחת לפחות (בהתאם למחלקות של אוניברסיטה זאת).
(10) (ג) כתוב ב-RA את ה-Constraint שמקיים את התנאי הבא: בכל ספרייה צריכים להיות את כל הספרים המומלצים (בהתאם למחלקות של האוניברסיטה), בכמות לא פחות

מ-10 עותקים כל אחד.

.3

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

(10) (א) מצא בשאילתה אחת את כל הסטודנטים כך, שכל אחד מהם עבור כל קורס שהוא לומד, השתמש במספר מקסימאלי של ספרים מומלצים (בין כל סטודנטים של אותה אוניברסיטה שלמדו אותו קורס).

(10) (ב) נניח כי עשו ראורגניזציה של הספריות לפי הכלל הבא: עבור כל קורס שאין עבורו בספרייה את כל הספרים המומלצים, החליפו מספר עותקים של כל ספר שנמצא בכמות מינימאלית (בין כל הספרים לאותו קורס הנמצאים באותה ספרייה), בממוצע של מספר ספרים המומלצים לאותו קורס באותה ספרייה. כתוב את השאילתא המתאימה.

.4

(10) נתון XML-file הבא:

```
<doc> This is Doc
    <sub1 par1="HELLO1">
      <sub2> What is this?
        <sub3> And this?
          <sub4 par2="HELLO2"/>
        </sub3>
      </sub2>
      <sub2> How are You? </sub2>
      <sub2> Where?</sub2>
    </sub1>
    <sub1>
      <sub2> What is this? </sub2>
    </sub1>
</doc>
```

(א) כתוב את ה-DTD המתאים ל- XML-file הנ"ל.

(ב) מהו הפלט של ה-XPathes הבאים?

(1) /doc//sub2/following::*

(2) //ancestor::sub2

.5

(5) (א) נניח כי ב-FILE מסוים יש 12 רשומות: FILE= 1, 2, 3, 3, 2, 1, 1, 1, 1, 1, 2, 1.

(1) כתוב Bitmap Index לכל ערך של ה-FILE;

(2) עבור כל Index, כתוב Compressed Index;

(5) (ב) נניח כי ה- hash function מייצרת 4 ביטים. בבניית Linear Hash Table נשתמש בקבוע 0.9 (ז"א, $(r/n)/b \approx 0.9$, כאשר r- מספר הרשומות ס"ה, n- מספר ה-buckets, b-מספר הרשומות ב- block אחד). כל בלוק יכול להכיל שתי רשומות.

תכניס ל- hash table את 8 הערכים הבאים (משמאל לימין):

1001, 0011, 0111, 1111, 1100, 1110, 1011, 1000

ציין את הקריטריון המתאים.

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.