ER Modeling II

Design a database for an investment company that records the data about brokers, offices, investors, stocks and how much of the stock a particular investor holds, and dividends. Moreover we know that:

1. every broker has only one office,
2. every investor is assigned to one broker,
3. stocks pay the same dividend to everyone holding the stock.

Draw an ER diagram for this database, identify constraints that hold in the diagram, and translate the result into relational schema.

Reasoning about Functional Dependencies.

- Show that whenever $X \rightarrow YZ$ holds in $R$ then $X \rightarrow Y$ also holds.
- Show that every relation must have at least one superkey and one candidate key.
- Consider a relation $R(A_1, \ldots, A_k, B_1, \ldots, B_k)$ with $2k$ attributes and a set of functional dependencies $F = \{ A_i \rightarrow B_i, B_i \rightarrow A_i \mid 0 < i \leq k \}$. How many candidate keys does $R$ have?

FDs and Normal Forms

Suppose instead we have the following database for the investment company, organized as a single relation $R$ with the following attributes: $B$ (broker), $O$ (office of a broker), $I$ (investor), $S$ (stock), $Q$ (quantity of stock owned by an investor), and $D$ (dividend paid by a stock). In addition we know the following functional dependencies hold: $S \rightarrow D$, $I \rightarrow B$, $IS \rightarrow Q$, and $B \rightarrow O$. 
1. Find all candidate keys for $R$.

2. Find a lossless-join decomposition of $R$ into BCNF.

3. Find a lossless-join dependency-preserving decomposition of $R$ into 3NF.

Compare results of (2) and (3) with your answer to problem 1 (and explain the differences, if any).

**Relational Algebra**

Convert the following SQL queries to *Relational Algebra*:

**Q1:**

```sql
SELECT DISTINCT a.name, p.title
FROM author a, wrote w, publication p
WHERE a.aid=w.author AND w.publication=p.pubid
```

**Q2:**

```sql
SELECT p.pubid, p.title
FROM publication p
WHERE p.title NOT IN (SELECT pubid
                     FROM article)
```

**Q3:**

```sql
SELECT p.pubid, p.title
FROM publication p, (
    (SELECT pubid FROM book)
UNION ALL
    (SELECT pubid FROM journal)) bj
WHERE p.pubid=bj.pubid
```

For each of the relational algebra expressions indicate what *indices* should be created on the relations involved in the queries (and on which attributes) to improve the performance of the queries.