5 Databases (TGG)

Suppose that an Entity-Relationship model has been constructed that contains two entities $S(\underline{A},\ B)$ and $T(\underline{C},\ Amount)$, where A, B, C and $Amount$ are attributes and the underline indicates a key. Suppose that we also have a many-to-many relationship R between S and T.

We might expect that this model would be implemented in a relational schema such as $S(\underline{A},\ B)$, $T(\underline{C},\ Amount)$, and $R(\underline{A},\ \underline{C})$. However, the database implementor has noticed that a very common and expensive query is this: given an A-value a, find the sum of all $Amount$ values for records in T related to this a value in S. Therefore, the implementor has decided to "optimise" the database and replace table S with S' having schema

$$S'(\underline{A},\ B,\ Sum),$$

where the records in table S' will contain the precomputed values for this query. In this way the common and expensive query can be answered by a single key-based read. (Note: Sum should be 0 if no matching records exist.)

(a) Explain how the operation *insert* $(a,\ b)$ *into* S can be correctly implemented in the $\{S',\ R,\ T\}$ database. [4 marks]

(b) Explain how the operation *insert* (c, v) *into* T can be correctly implemented in the $\{S',\ R,\ T\}$ database. [4 marks]

(c) Explain how the operation *insert* $(a,\ c)$ *into* R can be correctly implemented in the $\{S',\ R,\ T\}$ database. [4 marks]

(d) For an OLTP database, discuss the performance implications of this so-called optimisation. [4 marks]

(e) This example illustrates a fundamental trade-off in the design and implementation of database applications. Discuss. [4 marks]