Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 1998 Proceedings

Americas Conference on Information Systems (AMCIS)

December 1998

Normalizing Relations: One-to-One Relationships

George Philip
University of Wisconsin Oshkosh

Normalizing Relations: One-to-One Relationships

George C. Philip
College of Business
University of Wisconsin Oshkosh

Introduction

This paper discusses problems and confusion that can arise in normalizing certain relations, using the commonly accepted practices of identifying primary keys and defining normal forms. Specifically, normalization of relations that combines information on two different entities with a one-to-one relationship is discussed. It is shown that for such relations, using the standard normalization procedures may lead to the conclusion that a relation in Boyce-Codd normal form (BCNF) still suffers from insertion and deletion anomalies. Methods to deal with such cases are presented.

The Paradox

Consider a relation, ASSIGNMENT, which combines information on employees and offices, with information on the current assignment of employees to offices:

ASSIGNMENT (ID, NAME, TITLE, OFFICE_NO, AREA, FLOORING)

In the above relation, ID, NAME, TITLE, and OFFICE_NO represent the unique identification number, the name (non-unique), the title, and the office number of the employee, respectively. AREA and FLOORING represent the floor area, and the type of flooring of the employee's office, respectively. Each office has a unique office number.

It is assumed that a maximum of one employee is assigned to an office; thus an office may not have an employee assigned to it all the time. An employee has at most one office. Since some employees work from home, an employee may not have an office.

We will now present the paradox that although the above relation fits the definition of Boyce-Codd normal form, it suffers from insertion and deletion anomalies.

Test for Boyce-Codd Normal Form (BCNF)

To apply the definition of BCNF normal form to ASSIGNMNET, we first identify the candidate keys. For an attribute to be a candidate key of a relation, it must satisfy the constraint that at any given time, no two tuples (rows) of the relation have the same value for the attribute. It should be noted that candidate keys that are not selected as the primary key, may or may not have nulls allowed (Date, 1990, P. 281). For example, in the relation, STUDENT (ID, SSN, NAME, PHONE), the social security number (SSN) is a candidate key even if some students may not have a social security number. In the relation, ASSIGNMENT, the attributes, ID and OFFICE_NO are candidate keys, since no two assignments have the same value for OFFICE_NO or for ID.

Since a relation that is in BCNF is also in first, second, and third normal forms, we consider only BCNF here. A relation is in BCNF if every determinant is a candidate key. A determinant is any attribute on which some other attribute is functionally dependent. An attribute Y is functionally dependent on attribute X if and only if each X-value has associated with it exactly one Y-value at any one time.

In the ASSIGNMENT relation, ID determines NAME, TITLE and OFFICE_NO, and in addition, AREA and FLOORING. Similarly, OFFICE_NO determines AREA, FLOORING, ID, NAME and TITLE. Thus, ID and OFFICE_NO are determinants. There are no other determinants. Since every determinant is a candidate key, ASSIGNMENT is in BCNF. Since there are no multivalued dependencies in ASSIGNMENT, it is also in fourth normal form.

Insertion and Deletion Anomalies

Is the design of ASSIGNMENT good? Does it suffer from any insertion or deletion anomalies? To answer these questions, it is necessary to choose a primary key. Any candidate key can be selected as the primary key if it satisfies the entity integrity rule that no component of the primary key is allowed to accept nulls. First, we consider the candidate key, ID. It is clear that if an ASSIGNMENT tuple has a value for ID, then the value will be unique, since an employee is assigned to exactly one office. But, will there be tuples with null values for ID?

To answer the above question, we will present two examples that illustrate the assumptions and procedures that typically are used in the literature in selecting a primary key and in testing for anomalies. First, consider a relation presented by Date (1990, P. 534), that combines information on shipments (of parts by suppliers at a given time) and suppliers.

FIRST (S#, STATUS, CITY, P#, QTY)

The above relation, FIRST, consists of the attributes, S#, STATUS, CITY, P#, and QTY, that represent the number (unique), status, and location of the supplier, and the part number and quantity of the part supplied by the supplier, respectively.

A supplier may ship multiple parts, and multiple suppliers may ship a part. There can be at most one shipment at any given time for a given supplier and a given part. The combination of S# and P# was chosen as the primary key. In selecting the primary key, the implied assumption is that FIRST is to be used to store information only on suppliers who have a shipment. Entering information on suppliers who do not have a shipment will result in having null values for P#, thus making the combination S# and P# not satisfying the entity integrity rule for primary key. The reason for this assumption is that otherwise the relation doesn’t have a primary key. After selecting the primary key, the insertion anomaly and deletion anomaly are identified as two reasons for breaking down FIRST. The insertion anomaly is that it is not possible to insert information on a supplier until that supplier supplies a part. The deletion anomaly is that deleting the only tuple for a supplier will result in deleting the information, not only on the shipment, but also on the supplier. The fact that these insertion and deletion anomalies are of concern indicates that it is important for the users to store information also on suppliers who do not have a shipment. Thus, the assumptions made for choosing the primary key and for identifying the insertion and deletion anomalies appear to be contradictory.

An alternate approach is to recognize that the relation doesn’t have any attribute(s) that qualify as a primary key, under the realistic assumption that the relation, FIRST, needs to store information on all suppliers (including those who do not have a shipment) and shipments. Hence, the relation is to be modified.

A second example involves a relation that combines information on suppliers and supplies (Ullman, 1988, P. 377):

SUP_INFO (SNAME, SADDRESS, ITEM, PRICE)

A supplier may supply multiple items. Multiple suppliers may supply an item. The price of an item varies with the supplier. Thus, PRICE is functionally dependent on SNAME and ITEM, while SADDRESS is dependent on SNAME. Regarding the insertion anomaly, Ullman states, "We cannot record an address for a supplier if that supplier does not currently supply at least one item. We might put null values in the ITEM and PRICE components of a tuple for that supplier, … . ITEM and SNAME together form a key for the relation." Again, the implied assumption in choosing the primary key is that SUP_INFO is to store information only on suppliers who currently supply at least one item.

In examples like the one above involving a relation that combines information on two entities with a one-to-many or many-to-many relationship, the assumptions regarding possible null values of the primary key do not affect the final conclusion that the relation needs to be broken down. This is because such relations also suffer from duplication of data. However, such an approach could seriously affect the outcomes when a relation combines information on two entities with a one-to-one relationship, as in ASSIGNMENT.

When a relation combines information on two entities and their relationship, *the standard approach* used in selecting a primary key, as explained above, is to assume that, for any entity, the relation will store only those instances that are part of the relationship. These examples are not isolated cases. Almost all the authors use this approach (Ullman, 1988, P. 377; Hansen & Hansen, 1996, P. 149; Kroenke, 1998, P. 115). Adopting this approach, we assume that ASSIGNMENT is to store information only on employees and offices that are part of an assignment. In this case, the candidate key, ID, or OFFICE_NO can be selected as the primary key, since every tuple of ASSIGNMENT has a value (nonnull) for ID and OFFICE_NO. If ID is selected as the primary key, then we have the insertion anomaly that it is not possible to insert the AREA and FLOORING information for an OFFICE until that office has been assigned to an employee. Similarly, there is the deletion anomaly that if an employee leaves the company and the tuple corresponding to the employee is deleted, all the information on the office assigned to the employee also is lost. Thus the conclusion is that the relation, ASSIGNMENT, that is in BCNF (and 4NF) still suffers from insertion and deletion anomalies. The same conclusion will be reached if OFFICE_NO is chosen as the primary key. The question, then is, how can poor designs be identified?

The Solution

One option is to recognize that if we use the standard approach of choosing the primary key, BCNF does not guarantee the absence of insertion and deletion anomalies when a relation includes information on two entities that have a one-to-one relationship and the minimum cardinalities of the two entities are both zeroes. The rules for converting E/R diagrams to relations may be used in creating a better design.

When the minimum cardinality of one entity is not zero, then no insertion and deletion anomalies will be present. For example, if each employee has an office, then the minimum cardinality of the OFFICE entity is one. If OFFICE_NO is selected as the primary key, any employee record can be inserted, since each employee has an office. Since each employee should have an office, whenever an office is to be deleted, it should be replaced by another office, thus removing problems due to deletion anomaly.

A second and better option is to deviate from the standard procedure, and assume before selecting a primary key that the relation, ASSIGNMENT, is to store information on all employees, offices and their assignments. Under this assumption, since ID and OFFICE_NO may have null values, there is no attribute that qualifies as the primary key if the entity integrity rule is enforced. In this case, the lack of a qualified primary key indicates poor design.

References

References are available from the author (philip@uwosh.edu).