Databases

p 5q 8
JKMM

Papers 5,12

Solution

a) *entities* are items whose properties and interrelationships we may record in a database. An entity may be physical (a car, a person, a widget) or organizational (a company or a department) or abstract (a seat on an aircraft). The essential property of an entity is identity; it makes sense to say that two entity references are to the same entity, or to two different entities.

attributes are properties of entities, and are defined by functions taking values in some attribute value set — common instances are strings and integers, but an a.v.s. may contain structured data values.

Papers 5, 12

Solution

a) ctd) relationships are of a given degree n, and are defined for n-tuplets of entities — a specified relationship R either holds or does not hold for the n-tuplet $(e_1, e_2 \ldots e_n)$.

Often, indeed usually, entities will be typed, and attributes will be functions defined for elements of that type, taking values in a typed a.v.s. Each entity instance participating in a relationship will be typed, i.e there will be a specific type for e_i, $1 \leq i \leq n$.

---

b) the central requirement for recording information in a database is that the entities to be represented can be identified. One might

Papers 5.12

Solution

b) ctd) say that the key idea is that of key.

A key for identifying entities is a collection of attributes knowledge of whose values will, in ANY population of data, determine a unique entity. Often attributes have been established for precisely that purpose, for example the registration number of a car or the social security number of a UK citizen. In the relational model we shall choose a primary key for each entity type, that is a minimal set of attributes forming a key; often we shall use single attribute keys. SQL-92 has a schema definition sublanguage that allows us to ~~assert~~ a key for each relation.

The ODMG data definition language ODL

Papers 5,12

Solution

(b) ctd) allows keys to be identified, but the standard mechanism is to create a new object instance of a class corresponding to the entity type for each entity instance. The object identifier can then be used to refer to the unique entity.

Attribute values of an entity will be stored in a relation whose primary key is a suitable attribute set under SQL-92; in ODL they will be properties of the entity class.

Relationships in SQL-92 can be n-ary; instances of an n-ary relationship will specify keys for each entity of $(e_1, e_2, \ldots e_n)$. In ODL relationships are BINARY only, relating pairs of objects named by obj-ID.

Papers 5,12

Solution ctd) c) We must assume that some clients hold accounts with both enterprises already. It's likely that each has its own unique client ID; the clean solution is to extend both existing DBs with a new unique ID, noting that schema extension and key replacement may be non-trivial. There's then the difficult business of identifying common clients on the basis of name/address information; a mixture of program & human intervention.

That's the essential first step, and provides a federated database solution. Going further will mean looking at dependent attributes and trying to do semantic reconciliation. Then a global schema may be possible.

What about querying the federated database? It's only meaningful in so far as semantic mapping's been carried out.

NOT easy, but there's plenty to say!