4.2.5 Relational Keys

As stated earlier, there are no duplicate tuples within a relation. Therefore, we need to be able to identify one or more attributes (called **relational keys**) that uniquely identifies each tuple in a relation. In this section, we explain the terminology used for relational keys.

Superkey

An attribute, or set of attributes, that uniquely identifies a tuple within a relation.

A superkey uniquely identifies each tuple within a relation. However, a superkey may contain additional attributes that are not necessary for unique identification, and we are interested in identifying superkeys that contain only the minimum number of attributes necessary for unique identification.

Candidate key

A superkey such that no proper subset is a superkey within the relation.

A candidate key K for a relation R has two properties:

- *Uniqueness.* In each tuple of R, the values of K uniquely identify that tuple.
- *Irreducibility.* No proper subset of K has the uniqueness property.

There may be several candidate keys for a relation. When a key consists of more than one attribute, we call it a **composite key**. Consider the `Branch` relation shown in Figure 4.1. Given a value of city, we can determine several branch offices (for example, London has two branch offices). This attribute cannot be a candidate key. On the other hand, because *DreamHome* allocates each branch office a unique branch number, given a branch number value, `branchNo`, we can determine at most one tuple, so that `branchNo` is a candidate key. Similarly, postcode is also a candidate key for this relation.

Now consider a relation Viewing, which contains information relating to properties viewed by clients. The relation comprises a client number (clientNo), a property number (propertyNo), a date of viewing (viewDate) and, optionally, a comment (comment). Given a client number, clientNo, there may be several corresponding viewings for different properties. Similarly, given a property number, propertyNo, there may be several clients who viewed this property. Therefore, clientNo by itself or

propertyNo by itself cannot be selected as a candidate key. However, the combination of clientNo and propertyNo identifies at most one tuple, so for the Viewing relation, clientNo and propertyNo together form the (composite) candidate key. If we need to take into account the possibility that a client may view a property more than once, then we could add viewDate to the composite key. However, we assume that this is not necessary.

Note that an instance of a relation cannot be used to prove that an attribute or combination of attributes is a candidate key. The fact that there are no duplicates for the values that appear at a particular moment in time does not guarantee that duplicates are not possible. However, the presence of duplicates in an instance can be used to show that some attribute combination is not a candidate key. Identifying a candidate key requires that we know the "real-world" meaning of the attribute(s) involved so that we can decide whether duplicates are possible. Only by using this semantic information can we be certain that an attribute combination is a candidate key. For example, from the data presented in Figure 4.1, we may think that a suitable candidate key for the Staff relation would be IName, the employee's surname. However, although there is only a single value of "White" in this instance of the Staff relation, a new member of staff with the surname "White" may join the company, invalidating the choice of IName as a candidate key.

Primary key

The candidate key that is selected to identify tuples uniquely within the relation.

Because a relation has no duplicate tuples, it is always possible to identify each row uniquely. This means that a relation always has a primary key. In the worst case, the entire set of attributes could serve as the primary key, but usually some smaller subset is sufficient to distinguish the tuples. The candidate keys that are not selected to be the primary key are called **alternate keys**. For the Branch relation, if we choose branchNo as the primary key, postcode would then be an alternate key. For the Viewing relation, there is only one candidate key, comprising clientNo and propertyNo, so these attributes would automatically form the primary key.

Foreign key

An attribute, or set of attributes, within one relation that matches the candidate key of some (possibly the same) relation.

When an attribute appears in more than one relation, its appearance usually represents a relationship between tuples of the two relations. For example, the inclusion of branchNo in both the Branch and Staff relations is quite deliberate and links each branch to the details of staff working at that branch. In the Branch relation, branchNo is the primary key. However, in the Staff relation, the branchNo attribute exists to match staff to the branch office they work in. In the Staff relation, branchNo is a

foreign key. We say that the attribute branchNo in the Staff relation **targets** the primary key attribute branchNo in the **home relation,** Branch. These common attributes play an important role in performing data manipulation, as we see in the next chapter.

4.2.6 Representing Relational Database Schemas

A relational database consists of any number of normalized relations. The relational schema for part of the *DreamHome* case study is:

```
Branch                  (branchNo, street, city, postcode)

Staff                   (staffNo, fName, IName, position, sex, DOB,
                        salary, branchNo)

PropertyForRent         (propertyNo, street, city, postcode, type,
                        rooms, rent, ownerNo, staffNo, branchNo)

Client                  (clientNo, fName, IName, telNo, prefType,
                        maxRent, eMail)

PrivateOwner            (ownerNo, fName, IName, address, telNo, eMail,
                        password)

Viewing                 (clientNo, propertyNo, viewDate, comment)

Registration            (clientNo, branchNo, staffNo, dateJoined)
```

The common convention for representing a relation schema is to give the name of the relation followed by the attribute names in parentheses. Normally, the primary key is underlined.

The *conceptual model,* or *conceptual schema,* is the set of all such schemas for the database. Figure 4.3 shows an instance of this relational schema.