

## Relational databases

### Keys

A key is the tool to unlock access to database tables. By knowing the key, we know how to locate specific records, and traverse the relationships between tables.

A candidate key is any field, or combination of fields, that uniquely identifies a record. The field/s of the candidate key must contain unique values (if the values were duplicated, they would be no longer identify unique records), and cannot contain a null value.

A primary key is the candidate key that has been chosen to identify unique records in a particular table.

Examine the following table:

#### Poet

| Code | First Name | Surname | Age |
|------|------------|---------|-----|
| 1    | Mongane    | Afrika  | 62  |
| 2    | Stephen    | Serote  | 58  |
| 3    | Tatumkhulu | Watson  | 29  |

At first it seems there are two candidate keys for this table. Both *code* and the combination of *first\_name* and *surname* would suffice. It is always better to choose the candidate key with the least number of fields for the actual primary key, so we would choose code in this case. Also, if we thought about it some more, we'd soon realize that there is a possibility of the second combination not being unique. The combination of *first\_name* and *surname* could be duplicated. So to consider this for a choice of primary key, we'd have to be sure that none of our poets could ever have the same name. This is the reason we assign code fields. Codes are assigned by our system, so we can ensure there are never any duplicate codes. After the primary key has been assigned, any remaining candidate keys are labelled alternate keys.

## Foreign Keys

A relation between two tables is created by creating a common field to the two tables. The common field must be a primary key to the one table (the table that would be the one component of the one-to-many relationship). Consider a relation between a poet table and a poetry anthology table. The relation is of little use if instead of using the primary key from the poet table, code, to create the relationship with the anthology table, we use another field that is not unique, such as the poet's surname. We would never know for certain which poet we're referring to in the poetry anthology. The *poet\_code* field is called the foreign key in the anthology table, which means it's the primary key (code) in the poet table.

### Anthology

- Anthology\_code
- Poet\_code

### Poet

- Code
- Firstname
- Surname
- Age

Foreign keys allow us to ensure what is called referential integrity. This means that if a foreign key contains a value, the value must refer to an existing record in the related table. For example, take a look at these two tables:

### Poet

**Code Firstname Surname**

|   |            |        |
|---|------------|--------|
| 1 | Mongane    | Afrika |
| 2 | Stephen    | Serote |
| 3 | Tatumkhulu | Watson |

### Poem

|  | <b>Title</b>        | <b>Poet</b> |
|--|---------------------|-------------|
|  | Wakening Night      | 1           |
|  | Thrones of Darkness | 2           |
|  | Once                | 3           |

Referential integrity exists here, as all the poets listed in the poem table exist in the poet table. Stephen Serote now pulls out of the anthology, and we delete him from the poet table. In a situation where referential integrity is not enforced, he would not appear in the poet table, but the code he used to have would still appear in the poem table. So, when we look up the poet who wrote *Thrones of Darkness* (which is poet code 2) we are sent to a non-existent record.

### Poet

**Code Firstname Surname**

|   |            |        |
|---|------------|--------|
| 1 | Mongane    | Afrika |
| 3 | Tatumkhulu | Watson |

### Poem

|  | <b>Title</b>        | <b>Poet</b> |
|--|---------------------|-------------|
|  | Wakening Night      | 1           |
|  | Thrones of Darkness | 2           |
|  | Once                | 3           |

The delete results in poor data integrity.

Foreign keys also allow what are called cascading deletes and updates. This means that we can delete Stephen Serote from the poet table, and all poems written by him, with one SQL statement. The delete "cascades" through the relevant tables, removing all relevant records. Foreign keys can contain null values if the relationship is optional, which indicates that no relationship exists. If the relationship is mandatory, the foreign key cannot contain nulls.