

LESSON 4.2

98-364 Database Administration Fundamentals

# Understand Primary, Foreign, and Composite Keys

## Lesson Overview

In this lesson, you will learn:

- How to choose appropriate keys
- What a primary key is
- What a foreign key is
- Considerations for maintaining data integrity

## Choosing Appropriate Keys

- Keys must be unique.
  - Not relatively unique, but truly unique for each record in the database.
  - If you are named Bob, this may be unique in your home, but it's not unique in the city of New York.
- Keys only need to have value for the database.
  - Student\_ID has the ability to be unique among the 500,000 current students in the New York school system.

Creating a key just for one student makes it unique and valuable only to the database. If the key gets compromised, it has no value outside the database.

On the other hand, if your Social Security number becomes public knowledge, it can cause problems for you because it has value outside the database.

## Choosing Appropriate Keys (continued)

- To be most useful, keys need to be of consistent type and style.
  - A key needs to be made up of only one data type.
  - This is why most keys are made of the same type of data and have the same length.
  - When keys are consistent, this helps keep the database in order and helps with retrieving records.

# Primary Keys

## General Rules:

- — The primary key uniquely identifies each record in a database table.
- — Primary keys must contain unique values.
- — A primary key column cannot contain NULL values.
- — Each table should have a primary key, and each table can have only one primary key.

## Working Definition:

- A primary key uniquely defines a relationship within a database. For an attribute to be a good primary key, it must not repeat.

Sometimes we try to use a person's name as the primary key. In the classroom, a teacher will call on a student by name, but it may not uniquely define that student. The class could have two Mary Smiths, for example.

Databases solve this problem by creating unique primary keys, such as the "Student\_ID" of a specific data type for data integrity. There will never be two students with the same ID.

## Primary Keys (continued)

### Example of a primary key within a table in normalized form

Table Teacher\_classes contains Teacher\_ID and Class\_ID:

<b>Teacher_ID</b>	Class_ID
<b>78v22</b>	124567
<b>78v22</b>	Lit200

- In the above table, we have a unique primary key for the table Teacher\_classes. Each teacher has a unique ID (primary key in bold) tied to many class IDs.
- In First Normal Form (1NF), we create tables for each group of related data and identify each row with a unique column.

In this case, the primary key is Teacher\_ID.

*Note:* Additional review materials that cover the topic of 1NF are in Lesson 4.1.

## Primary Keys (continued)

### Syntax Example

#### SQL CREATE PRIMARY KEY

```
CREATE TABLE Students_Nameof_School
( Student_ID int NOT NULL PRIMARY KEY,
  LastName varchar(255) NOT NULL,
  FirstName varchar(255),
  Address varchar(255),
  City varchar(255) )
```

- In this example (which is written to no particular version of SQL), a primary key is created and tied to `Student_ID`.
- It has the data type of integer and defined as `NOT NULL`.
- When one enters data into the field, it must have a value.
- `NOT NULL` is an added constraint on what can go into the field.
- Integer is the first constraint or restriction to what can be placed into the primary key field `Student_ID`.

# Foreign Keys

## General Rules

- A foreign key in one table points to a primary key in another table.
- A foreign key column should *not* contain NULL values.
  - A NULL value in a foreign key means it has no relations to other tables.

## Working Definition

- A foreign key is a reference to the primary key in another table.
  - — Using this method, we relate or reference all the tables together.
  - — Foreign keys do *not* need to have unique values.
  - — We use foreign keys as lookup items for another table.



## Foreign Keys (continued)

### Example of a foreign key

Table Teacher\_classes with Teacher\_ID and Class\_ID:

Teacher_ID	Class_ID 1	Class_ID 2	Class_ID 3
78v22	124567	Lit200	224567

- There is a unique primary key for the table Teacher\_classes.
- Each teacher will have a unique ID tied to many class IDs. The Class\_ID is the foreign key.
- The table above shows that one teacher can have different classes and multiple sections of the same class.
- To find the details of the classes, one would use the foreign key to find the description of the class in the class\_description table.
- The table above is a virtual table or a view of a query on Teacher\_ID “78v22”.

*Note:* See Lessons 2.3 and 3.1 for more on this subject.

## Foreign Key (continued)

Syntax Example to create a foreign key at the same time as a primary key

```
CREATE TABLE Students_Name of School
( Student_ID int NOT NULL PRIMARY KEY,
  LastName varchar(255) NOT NULL,
  FirstName varchar(255),
  Address varchar(255),
  ZipCode int(5) NOT NULL FOREIGN KEY)
```

- In this example (which is not written to any specific version of SQL), a foreign key is created and tied to ZipCode.
- It has the data type of integer and *cannot* be empty (NULL).
- When one enters data into the field, it must have a value. The foreign key is the lookup key to find city and state information for this student.

## Data Integrity

- Data integrity is enforced by a series of constraints or rules.
  - Three types of integrity constraints are an inherent part of the relational database model.
1. **Entity integrity**—Focus is on the primary key.
    - The rule is that every table must have a primary key, and that the column or columns chosen to be the primary key should be unique and not NULL.

## Data Integrity (continued)

2. **Referential integrity**—Focus is on the foreign key.
  - The rule is any foreign key value can only be in one of *two* states.
    - First state: a foreign key value refers to a primary key value of a table in the database.
    - Second state: a foreign key value can be NULL.
      - Either there is no relationship between the objects in the database, or this relationship is undefined.

## Data Integrity (continued)

**3. Domain integrity**—Specifies that all allowable values in a relational database must be declared.

- A domain is a set of values of the same type (data type).
  - Domains, therefore, are groups of values from which actual values are drawn.
- A domain describes the set of possible values for a given attribute.

Because a domain constrains the attribute's values, it can be considered a constraint.

### Example:

The character value “John” is not in the integer domain.

The integer value “42” satisfies the domain constraint.

## Lesson Review

1. Why are unique keys important?
2. What is the relationship between keys and data types?
3. What are the differences between a foreign key and a primary key?
4. Why can a foreign key have a NULL value but a primary key cannot?
5. List and describe the three rules of data integrity.