

# RelationalDBDesign [Tweet](#)

## Database Analysis

## Database Design

Advertisement

### Database Analysis [«Prev](#) [Next»](#)

- Relational Data Analysis
- Relational Constraints
- DB Normal Forms
- Third Normal Form
- Table Joins User Views
- Physical Design

Advertisement

[Lesson 11](#)

[Primary keys](#)

[Objective](#)

[Define database primary keys.](#)

**Database Primary Key:** A primary key is a combination of one or more column values in a table that make a row of data unique within a database table. Identification of primary keys is an important part of *entity type identification*. Ideally, primary keys should be

1. stable and
2. single purpose.

*Stable* means that a *primary key* should never change after it has been assigned to an entity.

*Single purpose* means that a *primary key* attribute should have no purpose other than entity identification. Typically, good choices for primary keys are integer values automatically generated by a DBMS. Access has the *AutoNumber* data type for primary keys and *Oracle/PLSQL* uses the Sequences (Autonumber) for primary keys. As noted in the preceding lesson, tables consist of

1. *key columns* and
2. *descriptor columns*.

*Descriptor columns* contain information about the person, place, or thing described by a particular row in the table.

*Key columns* contain information that distinguishes elements from every other item represented in the table.

A key column (or set of key columns) that distinguishes a record from every other record in a table is referred to as a *primary key*.

#### Guidelines for Primary Keys

A primary key should never be a value that is also used outside the database, such as a

1. Social Security number (Social Insurance Number in Canada) or
2. license-plate number.

Social Security numbers are especially bad choices because they can be recycled, incorrectly assigned, or changed. Furthermore, the U.S. government strictly limits when Social Security numbers can be collected and used.

A better practice is to create a *unique identifier* to represent each row in a table and not try to store meaningful data in the primary-key column.

#### Relational Database Design

#### Advice on primary key fields

Most database designers mark primary-key fields in an ER diagram with an asterisk (\*).

Some *relational database management systems* underline the key fields or write the field's name in bold-face type.

#### Description of candidate key:

A column or (set of columns) that can be used as a *primary key* for a table.

A car's license-plate number is one example of a candidate key.

1. Within a given state, only one car can have a particular license number, so a table describing CAR might contain a column labeled PlateNo.
2. In a nationwide database of license-plate numbers, where the same license plate number could be assigned to cars in different states, the state and PlateNo key columns would form a candidate key.

The problem with using a license-plate number to identify a car is that

1. the car's owner can request new plates with a chosen message or
2. the state may change their numbering system to accommodate more cars.

In either case, the license-plate number associated with the car would change. Changing key values in a table can cause a great deal of confusion, especially if an update is done incorrectly and the wrong value is assigned to an item.

When you design a table, look for candidate keys with the following two properties:

1. The value in the key column can not change.
2. The key column cannot contain a null value.

If the candidate key meets these two additional requirements, it can be considered a *primary key*.

The next lesson defines *concatenated primary keys*.

*concatenated primary key*: A concatenated primary key is a database table key made up of more than one field.

---