# Column Constraints

## Description

This section is from the "[Practical PostgreSQL](http://chestofbooks.com/computers/databases/postgresql/practical-postgresql/index.html)" book, by John Worsley and Joshua Drake. Also available from Amazon: [Practical PostgreSQL](http://www.amazon.com/gp/product/1565928466?ie=UTF8&tag=theultimatlearna&link_code=as3&camp=211189&creative=373489&creativeASIN=1565928466).

# Column Constraints

If you perform a slash h (\h) on create table, you may have noticed that there are additional notes on how to create a table or column constraint. The portion that applies to a column constraint is as follows:

where column\_constraint can be:

[CONSTRAINT constraint\_name]

{

NOT NULL | NULL | UNIQUE | PRIMARY KEY

| DEFAULT

*value*

| CHECK (

*condition*

)

| REFERENCES

*table*

[ (

*column*

) ]

[ MATCH FULL | MATCH PARTIAL ]

[ ON DELETE

*action*

]

[ ON UPDATE

*action*

]

[ DEFERRABLE | NOT DEFERRABLE ]

[ INITIALLY DEFERRED | INITIALLY IMMEDIATE ]

}

There are five column constraints available:

1. Not Null
2. Unique
3. Primary Key
4. Check
5. References

The definitions for each clause are:

NULL | NOT NULL

NULL specifies that this column is allowed to have NULL values. This is the default so you don't need to implicitly specify this. This is only allowed as a column constraint, not a table constraint.

NOT NULL specifies that this column is not allowed to contain NULL values. Using the constraint CHECK ( *column* NOT NULL) is the equivalent to using NOT NULL.

UNIQUE

This column can contain only unique non-repeating values. UNIQUE does not necessarily mean NOT NULL. UNIQUE allows repeating NULL values to be in a column.

PRIMARY KEY

This column may contain only unique and non-null values. A table or column primary key is restricted to having only one primary key.

CHECK *condition*

This constraint defines tests that the column must satisfy for an insert or update operation to succeed on that row. The condition is an expression that must return a boolean result. For column constraint definitions, only one column can be referenced by the CHECK clause.

The following clauses apply to reference constraints:

REFERENCES *reftable* ( *refcolumn* )

The values in this column are checked against the values of another column that this references.

* reftable - this table contains the data that are compared with.
* refcolumn - this column is located in the reftable to compare data against. If refcolumn is left empty, then the PRIMARY KEY of the reftable is used.

MATCH FULL | MATCH PARTIAL

MATCH FULL rules out foreign key columns that contain NULL values, unless all foreign key columns are NULL. MATCH PARTIAL is not supported, but a default type is. The default allows NULL columns to satisfy the constraint.

ON DELETE *action*

When a DELETE is performed on a referenced row in the referenced table, one of these possible actions should likewise be executed:

* NO ACTION - Produces an error if the foreign key is violated. This is the default if an action is not specified.
* RESTRICT - Same as NO ACTION.
* CASCADE - Removes all rows which references the deleted row.
* SET NULL - Assigns a NULL to all referenced column values.
* SET DEFAULT - Sets all referenced columns to their default values.

ON UPDATE *action*

When an UPDATE is performed on a referenced row in the referenced table, an action occurs. If a row is updated, but the referenced column is not affected, then the action will not occur. The possible actions that can occur when an UPDATE is applied to a referenced column are the same as with ON DELETE. The only exception is the CASCADE action. CASCADE updates all of the rows which references the updated row.

DEFERRABLE | NOT DEFERRABLE

DEFERRABLE specifies the constraint to be postponed to the end of the transaction.

NOT DEFERRABLE means that the constraint is not postponed to the end of the transaction. This is the default when DEFERRABLE is not specified.

INITIALLY *checktime*

The constraint must be DEFERRABLE for you to specify a check time. The possible check times for a constraint to be deferred are:

* DEFERRED - postpone constraint checking until the end of the transaction is reached.
* IMMEDIATE - perform constraint checking after each statement. This is the default when a *checktime* is not specified.

To create a primary key column constraint on an employees table, use:

**Example 5-21. Creating a Primary Key Constraint**

CREATE TABLE employees

(

emp\_id INTEGER PRIMARY KEY,

name TEXT

);

This is the equivalent to the above operation:

CREATE TABLE employees

(

emp\_id INTEGER,

name TEXT,

PRIMARY KEY (emp\_id)

);

This creates a new table with a check column constraint that applies a rule which makes sure that employee identification numbers are greater than 100 and are non-NULL values. It also makes sure that an employee name exists for each employee id:

CREATE TABLE employees

(

emp\_id INTEGER NOT NULL CHECK (emp\_id > 100),

name TEXT NOT NULL CHECK (name <> '')

);

|  |  |
| --- | --- |
| Note | **Note** |
|  | When using the Check option to restrict the columns from containing empty values, there are different ways to express this depending on the data type of the column. You can specify a 0 for integer data type, or a pair of empty single quotes for text. If you use a pair of empty quotes for integer type, then it will automatically convert that into a 0.  Keep in mind that when you perform a check condition, the condition must return the same data type as the column value that you are checking. |

If you have an existing table that you want to apply a constraint on, then you can use the alter table command. The command below performs the exact same check condition as the above command for employee identification numbers:

**Example 5-22. Using a Constraint on an Existing Table**

ALTER TABLE employees

ADD CONSTRAINT emp\_id\_check

CHECK (emp\_id > 100);

The employee table currently contains these data values for the employee identification numbers:

SELECT emp\_id FROM employees;

emp\_id

--------

100

101

102

103

(4 rows)

After having executed the **ALTER TABLE** command, this message may have appeared:

ERROR: AlterTableAddConstraint:

rejected due to CHECK constraint emp\_id\_check

This problem arises when the constraint you are creating conflicts with the data that is currently stored in the table. We can correct this problem by changing the constraint to check for employee identification numbers greater than and equal to 100:

ALTER TABLE employees

ADD CONSTRAINT emp\_id\_check

CHECK (emp\_id >= 100);

|  |  |
| --- | --- |
| Note | **Creating a Constraint with ALTER TABLE** |
|  | To create a constraint using the ALTER TABLE command requires the following:   * the table which holds the column must exist * the column in which the constraint will apply on must exist * the constraint being created must apply to the data currently stored in the table.   If the above rules are not met, the constraint is not created. Using the CREATE TABLE command does not require these rules because the table and column is in the stage of creation and the constraint can be any valid constraint because there are no data values stored in the table yet. |

Continue to:

* prev: [Constraints](http://chestofbooks.com/computers/databases/postgresql/practical-postgresql/Constraints.html)
* [Table of Contents](http://chestofbooks.com/computers/databases/postgresql/practical-postgresql/index.html)
* next: [Table Constraints](http://chestofbooks.com/computers/databases/postgresql/practical-postgresql/Table-Constraints.html)

# Tags

[postgresql](http://chestofbooks.com/search.html?cx=000348145676127462126%3Anjhvjbiv4do&cof=FORID%3A10&sa=Search&q=postgresql), [psql](http://chestofbooks.com/search.html?cx=000348145676127462126%3Anjhvjbiv4do&cof=FORID%3A10&sa=Search&q=psql), [relational database](http://chestofbooks.com/search.html?cx=000348145676127462126%3Anjhvjbiv4do&cof=FORID%3A10&sa=Search&q=relational%20database), [sql](http://chestofbooks.com/search.html?cx=000348145676127462126%3Anjhvjbiv4do&cof=FORID%3A10&sa=Search&q=sql), [standard](http://chestofbooks.com/search.html?cx=000348145676127462126%3Anjhvjbiv4do&cof=FORID%3A10&sa=Search&q=standard), [query](http://chestofbooks.com/search.html?cx=000348145676127462126%3Anjhvjbiv4do&cof=FORID%3A10&sa=Search&q=query), [programming](http://chestofbooks.com/search.html?cx=000348145676127462126%3Anjhvjbiv4do&cof=FORID%3A10&sa=Search&q=programming), [administration](http://chestofbooks.com/search.html?cx=000348145676127462126%3Anjhvjbiv4do&cof=FORID%3A10&sa=Search&q=administration)

**Table Constraints**

## Description

This section is from the "[Practical PostgreSQL](http://chestofbooks.com/computers/databases/postgresql/practical-postgresql/index.html)" book, by John Worsley and Joshua Drake. Also available from Amazon: [Practical PostgreSQL](http://www.amazon.com/gp/product/1565928466?ie=UTF8&tag=theultimatlearna&link_code=as3&camp=211189&creative=373489&creativeASIN=1565928466).

# Table Constraints

Unlike column constraints, a table constraint can be defined on one or more columns of a table. There are five variations of a table constraint. They are:

1. PRIMARY KEY
2. UNIQUE
3. CHECK
4. FOREIGN KEY
5. REFERENCES

The syntax to create a table constraint is:

[ CONSTRAINT

*name*

] { PRIMARY KEY | UNIQUE } (

*column*

[, ...] )

[ CONSTRAINT

*name*

] CHECK (

*condition*

)

[ CONSTRAINT

*name*

] FOREIGN KEY (

*column*

[, ...] )

REFERENCES

*reftable*

(

*refcolumn*

[, ...] )

[ MATCH

*matchtype*

]

[ ON DELETE

*action*

]

[ ON UPDATE

*action*

]

[ [ NOT ] DEFERRABLE ]

[ INITIALLY

*checktime*

]

[CONSTRAINT *name* ] PRIMARY KEY *column* [,...]

The PRIMARY KEY table constraint is similar to the PRIMARY KEY column constraint. The only difference is that the table constraint allows multiple columns to be defined. This column may contain only unique and non-null values.

[CONSTRAINT *name* ] UNIQUE *column* [,...]

This specifies that there cannot be any repeating values in the column(s).

CHECK *condition*

This condition must be a boolean expression. It can use more than one column in this expression.

FOREIGN KEY

This means that a group of one or more distinct columns of a table is related to a group of distinct columns in the referenced table. Foreign keys are similar to column constraints, except for that it has the ability to include multiple columns.

REFERENCES

Refers or points to another table column(s). More than one table can be used as a reference. The possible actions that the reference is associated with are the same as with a column constraint.

This creates a table constraint and a reference to the isbn column in the book table.

**Example 5-23. Creating a Table Constraint**

CREATE TABLE publish

(

isbn INTEGER

CONSTRAINT exist CHECK

(isbn > 99999 and publ\_name <> '')

REFERENCES book (isbn) ON DELETE SET DEFAULT,

publ\_name TEXT,

sugg\_price MONEY

);

Continue to:

* prev: [Column Constraints](http://chestofbooks.com/computers/databases/postgresql/practical-postgresql/Column-Constraints.html)
* [Table of Contents](http://chestofbooks.com/computers/databases/postgresql/practical-postgresql/index.html)
* next: [Destroying a Constraint](http://chestofbooks.com/computers/databases/postgresql/practical-postgresql/Destroying-a-Constraint.html)

# Tags

[postgresql](http://chestofbooks.com/search.html?cx=000348145676127462126%3Anjhvjbiv4do&cof=FORID%3A10&sa=Search&q=postgresql), [psql](http://chestofbooks.com/search.html?cx=000348145676127462126%3Anjhvjbiv4do&cof=FORID%3A10&sa=Search&q=psql), [relational database](http://chestofbooks.com/search.html?cx=000348145676127462126%3Anjhvjbiv4do&cof=FORID%3A10&sa=Search&q=relational%20database), [sql](http://chestofbooks.com/search.html?cx=000348145676127462126%3Anjhvjbiv4do&cof=FORID%3A10&sa=Search&q=sql), [standard](http://chestofbooks.com/search.html?cx=000348145676127462126%3Anjhvjbiv4do&cof=FORID%3A10&sa=Search&q=standard), [query](http://chestofbooks.com/search.html?cx=000348145676127462126%3Anjhvjbiv4do&cof=FORID%3A10&sa=Search&q=query), [programming](http://chestofbooks.com/search.html?cx=000348145676127462126%3Anjhvjbiv4do&cof=FORID%3A10&sa=Search&q=programming), [administration](http://chestofbooks.com/search.html?cx=000348145676127462126%3Anjhvjbiv4do&cof=FORID%3A10&sa=Search&q=administration)