Understanding Referential Data Integrity in R:BASE

Understanding Referential Data Integrity in R:BASE

by R:BASE Technologies, Inc.

This document provides detailed information regarding the built-in referential data integrity features of R:BASE.

Understanding Referential Data Integrity in R:BASE

Printed: May 2017 in Murrysville, PA

First Edition

Table of Contents

Part I Referential Data Integrity in R:BASE 2

Part II Constraints 4

1 Primary Key ... 4
2 Foreign Key ... 5
3 Unique Key ... 6
4 Unique Index ... 6
5 Not NULL ... 6
6 Comparison of Constraints ... 6

Part III Cascade 8

Part IV Defining Constraints 10

Part V Listing Constraints 17

Part VI Removing Constraints 21

Part VII Constraint Messages 23

Part VIII Useful Resources 25

Part IX Feedback 27

Part I

1 Referential Data Integrity in R:BASE

By definition, referential integrity is a property of data which, when satisfied, requires any field in a table that is declared a foreign key can contain only values from a parent table's primary key.

In legacy R:BASE versions, data integrity and referential integrity had to be enforced through rules and programming. They are now a part of the database structure called **constraints** and are automatically enforced when defined. The constraints that can be defined include:

- Primary Key
- Foreign Key
- Unique Key
- Unique Index
- Not NULL

Part II

2 Constraints

Constraints provide automatic, database-wide data integrity and referential integrity. The not null constraint restricts data entry. No matter what method is used to enter data, the constraint verifies that the specified column has a value and the value is unique.

The primary key, foreign key, and unique key constraints provide both data integrity **and** referential integrity. The primary key column is automatically not null and unique. Deletions to a table with a defined primary key are automatically restricted if a referenced foreign key is defined. A value cannot be entered into a table with a foreign key unless that value exists in the referenced primary key table.

Constraints are similar to rules in that they indicate valid values or conditions that must be met in a table or column before data can be added, changed, or deleted. For instance, deleting a record that contains a value referred to by a foreign key in another table would break referential integrity.

Some constraints automatically define indexes when they are created; where they use the index to quickly check the condition. For example, a primary key is a constraint and an index is created on the primary key column(s). A constraint may use an index, but constraints and indexes are two different things. Constraints and indexes are created and listed separately. Constraints are commonly used to refer to primary and foreign keys.

Constraint Benefits Over Rules
Using rules to enforce these same data constraints required several different rules. You would need a rule to prevent duplicates (replaced by the primary key or unique constraint), verify a value rules (foreign key), and require a value rules (not null). In addition, you would need to define delete rules to prevent rows being deleted from the primary key table if there were matching rows in the foreign key table. If you wanted to prevent users from changing a primary key value that exists in the foreign key table, you would need to define an additional verify a value rule. Other constraint benefits over rules include:

- Easier to define
- Cannot be turned off, and are always in place
- Faster performance

2.1 Primary Key

A primary key is a column, or set of columns, that uniquely identify a row, meaning that each value in a primary key column is unique. As a constraint, the primary key prevents duplicate (non-unique) and null values from being entered into the column.

A primary key could be something like an employee id column in an employee table or it could be a combination of the customer's id and the customer's phone number.

Notes:

- Only one Primary Key can be defined per table.
- Defining a Primary Key automatically enforces "not NULL" and "unique" constraints on the column(s).
- A Primary Key cannot be defined if any one of the columns included in the desired key already have a NULL or duplicate values.
- R:BASE automatically builds an index on the specified column(s) when a Primary Key is

defined.

- Every table should have a column or set of columns that identify a row, and (in a well-designed database) should have a Primary Key.
- A Primary Key definition is used instead of a Rule to prevent duplicates, and has the advantage of faster performance. It is recommended to delete the rules and indexes that are no longer needed before defining a Primary Key constraint.

2.2 Foreign Key

Like a primary key, a foreign key is a column or a group of columns. Foreign keys are also used to ensure that only valid data is entered into a column. A foreign key matches the values in a particular primary key or unique key constraint which is defined in a different table.

Primary and foreign keys must also match in terms of the specified columns. If you have a multi-column primary key, you can not have single column foreign keys reference it. If you have a multi column foreign key, it cannot reference a single column primary key. When a primary key is defined as more than one column, those columns are treated as a whole. Primary and foreign keys must match exactly. Thus, if your primary key is defined as text, then the corresponding foreign keys referencing it must also be text.

Primary and foreign keys automatically preserve referential integrity. You cannot delete a row from a table with a defined primary key if there are referenced foreign keys, thus you can never have detail records without a matching master record.

You can delete a row from a table with a foreign key. You cannot add a row to table with a foreign key defined unless the value entered matches a value in the referenced key table.

By default, users cannot update a primary key value if there are references, thus ensured that linking columns always match. The same applies for updating primary key values if there is not matching values in the referenced foreign key. However, implementing the r:base Cascade feature on primary keys will allow such changes. When a primary key is defined as a "cascading" primary key, the referenced foreign key values are automatically updated, deleted, or both. This allows the ability to update or delete referenced primary key values and retain the referential data integrity.

Notes:

- A Primary Key can exist without a Foreign Key, but a Foreign Key cannot exist without a Primary Key.
- A Foreign Key is always defined to reference a Primary Key or Unique Key.
- A Foreign Key automatically checks that the values in the Foreign Key exist in the referenced key.
- Many Foreign Keys can be defined in one table, and many Foreign Keys in different tables can reference the same Primary Key.
- Once Primary and Foreign Keys are defined accordingly, R:BASE automatically preserves the referential data integrity.
- A Foreign Key replaces a "Verify a value RULE".
- An index is automatically built when a Foreign Key constraint is defined.
- It is recommended to delete rules and indexes that are no longer needed before defining a Foreign Key constraint.

2.3 Unique Key

A unique key is a column or set of columns that uniquely identify a row; in other words, each value in a unique key column is unique. A unique key constraint prevents duplicate (non-unique) and null values from being entered into a table. The only difference between a unique key and a primary key is that you can define multiple unique keys per table.

Notes:

- A Unique Key constraint can replace a "Require a unique value" rule.
- A Unique Key constraint automatically builds an index.

2.4 Unique Index

A unique index is an index that uniquely identify a row. A unique index constraint prevents duplicate values from being entered into a table, and can prevent null values, if defined. The differences between a unique key and a unique index is that the unique key must be defined a Not NULL.

Notes:

- A Unique Index constraint can replace a "Require a unique value" rule.
- A Unique Index constraint can replace a "Require a value" rule.
- A Unique Index constraint automatically builds an index.

2.5 Not NULL

Placing a not null constraint on a column requires that the data in the column must contain a value, and cannot be null. This prevents users from adding a "blank" value. A not null constraint cannot be added if the column already contains null values.

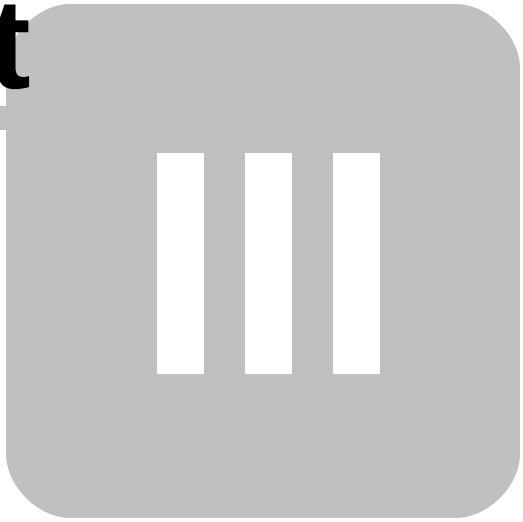
Notes:

- A not null constraint can replace a "Require a value" rule. R:BASE does not build an index for a not null constraint, but since it stores the not null as part of the column definition, it is able to check the constraint faster than it could check the rule.
- If null values already exist, the values must first be edited to actual data values, before the not null constraint can be added.

2.6 Comparison of Constraints

Constraint	Unique	Not NULL	Indexed	Can Be Referenced	Multi-Column Constraint	Can Replace a Rule
Primary Key	Yes	Yes	Yes	Yes	Yes	Yes
Foreign Key	No	Intended	Yes	No	Yes	Yes
Unique Key	Yes	Yes	Yes	Yes	Yes	Yes
Unique Index	Yes	Optional	Yes	No	Yes	Yes
Not NULL	No	Yes	No	No	No	Yes

Part



3 Cascade

Cascade is a referential integrity setting that can be applied to primary key tables, which maintains primary/foreign key relationships automatically. The Cascade options include:

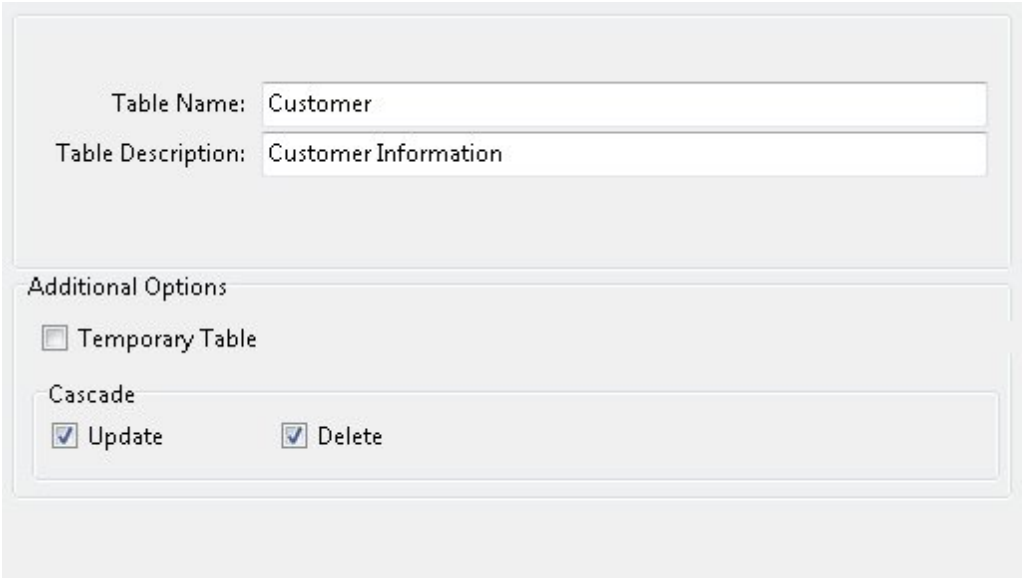
- Update
- Delete

Cascading updates enforce that when a primary key value is updated, the corresponding value in the foreign key table(s) will also be updated.
Cascading deletes enforce that when a primary key value is deleted, the corresponding value in the foreign key table(s) will also be deleted.

By not specifying either Update or Delete, both Cascade restrictions will be enforced upon the primary/foreign key tables and will prevent the values from being altered or deleted in the primary key table.

Separate Update and Delete data restrictions can allow a Cascade to be enforced for records that are updated, but not enforced when records are deleted, in order to avoid an accidental or undesired record delete. For example, if you either Update or Delete a primary key value from a table, the corresponding foreign key values are updated or deleted automatically. A Cascade can applied to Update, Delete or both to specific primary keys.

Cascade can be defined through the R:BASE command syntax using either the CREATE TABLE or ALTER TABLE commands. Cascade can also be set within the Data Designer when viewing the table properties.



Notes:

- Cascade can only be added to tables with defined Primary Keys.
- Cascade can be set to Update, Delete, or both.

Part IV

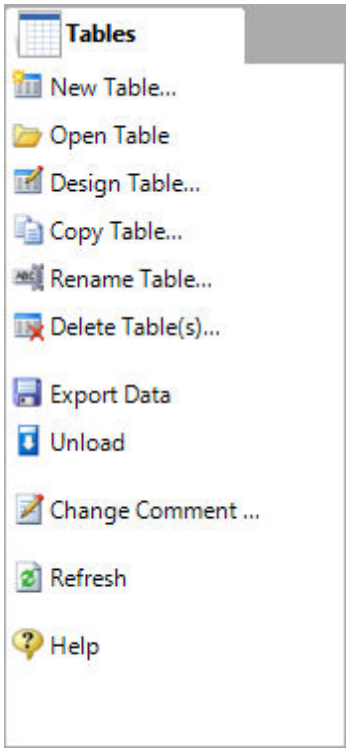
4 Defining Constraints

Constraints are defined using the Data Designer or through the CREATE TABLE or ALTER TABLE commands.

Before defining primary and foreign key constraints, decide which column or columns and which tables are best suited for the primary key. The column or columns selected should be the ones that uniquely identify the row from other rows in the table, and the table should be the one where that value is first entered into the database and is unique. Linking columns are generally good candidates for primary, foreign key constraints. For example, in the Concomp sample database, the empid column is used to link data in the Employee table with data in the Transmaster table and data in the Salesbonus table. Since a record is first entered into the Employee table, and when data is entered into the Transmaster or Salesbonus tables a matching record must already exist in the Employee table, the empid column in Employee becomes the primary key. The empid columns in Transmaster and Salesbonus become foreign keys referencing the Employee table primary key.

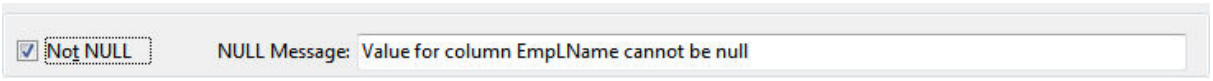
The direct benefit gained from designating empid as a primary key in the Employee table and foreign keys in the Transmaster and Salesbonus tables is the protection for your data from inadvertent changes. The empid value in Employee cannot be changed when there is a matching row in the Transmaster or Salesbonus table. Rows cannot be deleted from either table. When adding data to Transmaster or Salesbonus you automatically require a matching value in Employee without having to define a rule.

The Data Designer
Launching the Data Designer to defined constraints can be performed directly within the Database Explorer window. Select the "Tables" option from the Group Bar to view the available options. Then, highlight a table, and select "Design Table...".

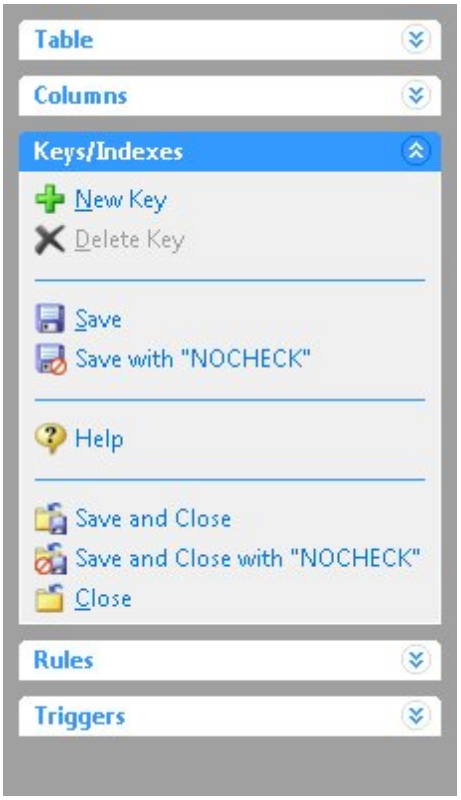


(Note: With an existing table selected, all of the above selections become enabled.)

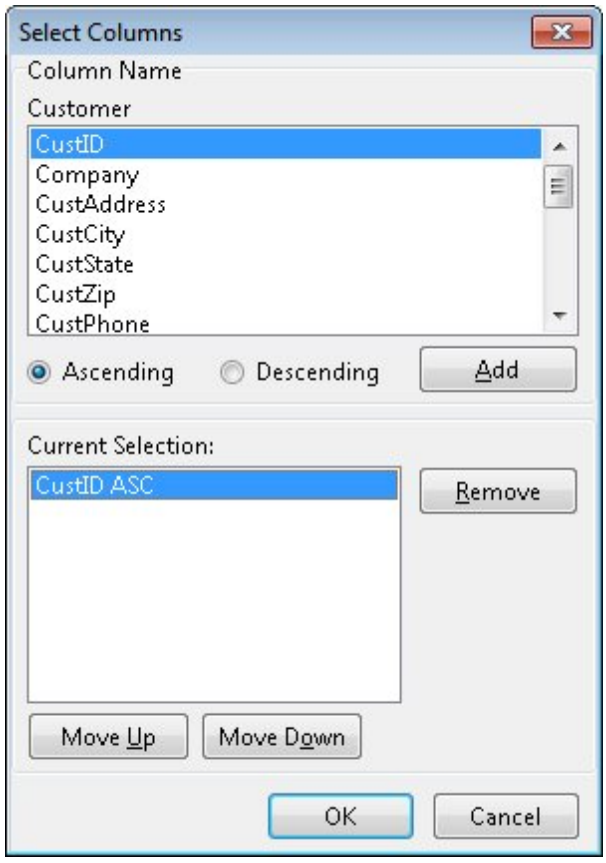
The "**Columns**" option will display the defined columns for the table, where the "Not NULL" constraint can be enabled.



The "**Key/Indexes**" option allows users to create primary key, foreign key, unique key, and unique index table constraints. The "New Key" option allows users to create a new key/index

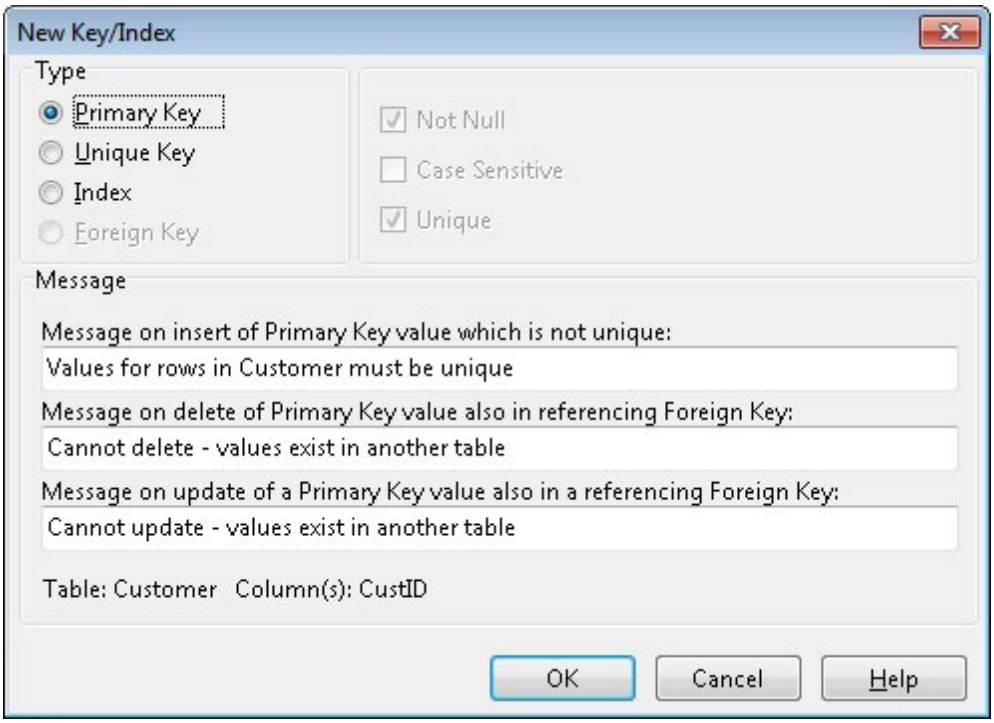


When creating a new key, the "Select Columns" dialog appears first allowing you to select the column(s) and sorting order (Ascending/Descending) before defining the key.



The "New Key/Index" dialog will display allowing for the creation of a primary key, foreign key, unique key, or index. If the Data Designer is launched for a table where a primary key is already defined, the option to specify a "Primary Key" type will be disabled. When adding a Primary Key, the options for "Not Null" and "Unique" are disabled by default and cannot be altered.

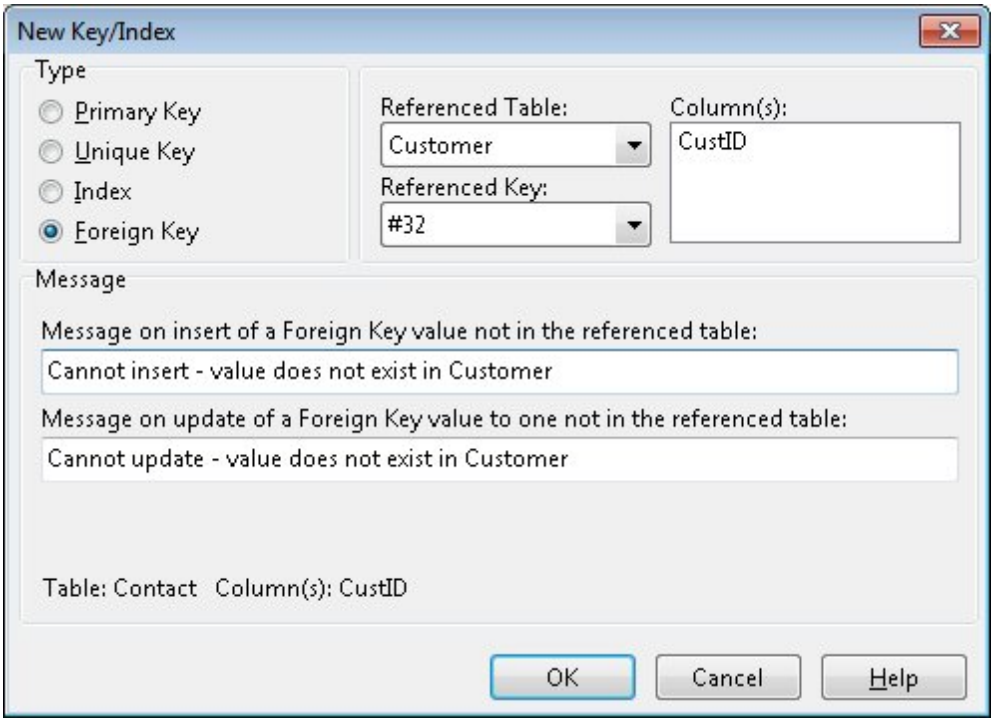
Custom constraint violation error messages can be specified for when an attempt is made to compromise the table's primary key referential integrity.



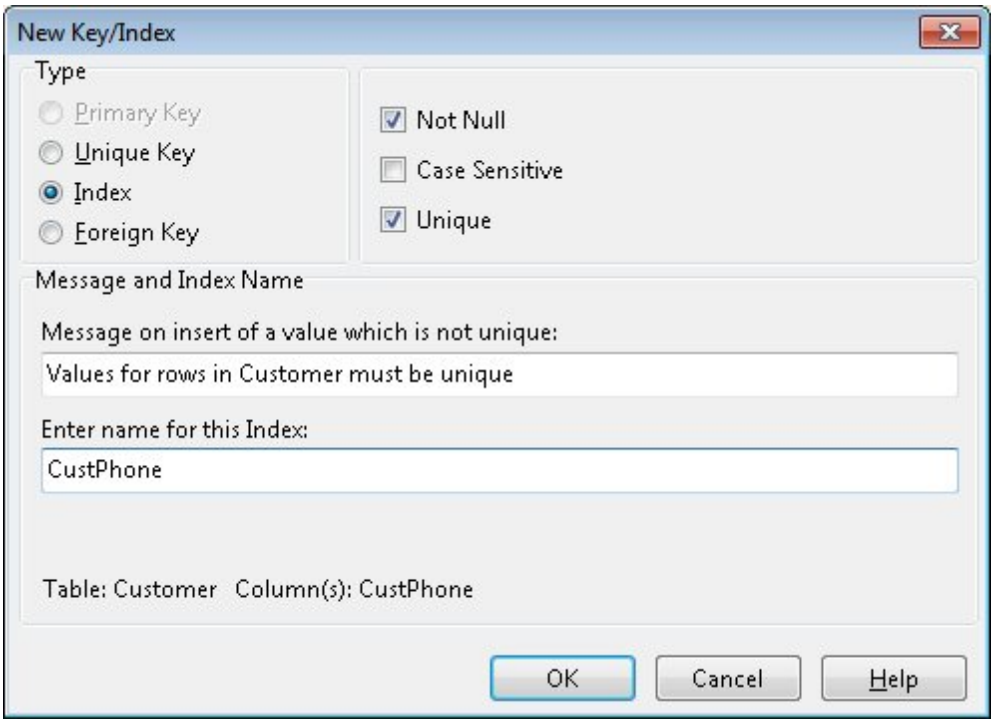
When adding a Unique Key, the options for "Not Null" and "Unique" are disabled by default and cannot be altered. Custom constraint violation error messages can be specified for when an attempt is made to compromise the table's unique key referential integrity.

When adding a Foreign Key, the options for to specify the referenced table and primary key is available. Along with a primary key, a unique key cal also be referenced by a foreign key.

Custom constraint violation error messages can be specified for when an attempt is made to compromise the table's foreign key referential integrity.



When adding a Index, the options for "Not Null" and "Unique" are optional. By enabling the "Unique"setting, the index will be defined as a unique index. A custom constraint violation error message can be specified for when an attempt is made to enter a non-unique value.



Commands
Defining a primary key or unique key constraint with the CREATE TABLE or ALTER TABLE commands requires the column be explicitly defined as NOT NULL as well as the unique or primary key designation. For example,

```
CREATE TABLE tblname (colname datatype NOT NULL PRIMARY KEY, ...)
```

Below are examples of the ALTER TABLE command to create a primary key for the Employee table, and to create a foreign key for the Employee table that references the Titles table.

```
ALTER TABLE `Employee` ADD PRIMARY KEY +
(`EmpID` ) +
('There must be a unique id number per employee.',+
 'You cannot delete an employee id that is being referenced in another
table.',+
 'Cannot change employee id value that is referenced in another table.')

ALTER TABLE `Employee` ADD FOREIGN KEY +
 ( `EmpTID` )+
 REFERENCES `Titles` +
('You must enter an employee id number that exists in the Titles
table.',+
 'You cannot update the title id to this value, use a value that exists
in +
 the Titles table.')
```

The following example displays the CREATE INDEX command to create a unique index.

```
CREATE UNIQUE INDEX TUII ON `TInvoiceHeader` (`TTransID` ASC )
```

Part V

5 Listing Constraints

The LIST command is used to display information about a database. Additional parameters to display specific constraint information include:

- LIST CONSTRAINTS
- LIST PKEYS
- LIST FKEYS
- LIST UKEYS
- LIST INDEXES
- LIST CASCADE

LIST CONSTRAINTS shows primary key, foreign key, unique key, and Not NULL constraints. The constraint ID, the type of constraint and if it is referenced, the table name, and table references if the key was a foreign key are displayed. Unique indexes are listed next with the index name, table and column names. Not null constraints are listed last with the table and column names.

```
R>LIST CONSTRAINTS

   Id     Type                         Table Name         References
   ------ ---------------------------- -----------------  --------------
   #31    PRIMARY KEY REFERENCED       Customer
   #33    PRIMARY KEY REFERENCED       Component
   #41    FOREIGN KEY                  SalesBonus         Employee
   #43    FOREIGN KEY                  CompUsed           Component
   #42    FOREIGN KEY                  CompUsed           Product
   #44    FOREIGN KEY                  ProdLocation       Product
   #45    FOREIGN KEY                  Levels             Product
   #34    PRIMARY KEY                  Levels
   #35    PRIMARY KEY REFERENCED       Product
   #47    FOREIGN KEY                  InvoiceHeader      Customer
   #46    FOREIGN KEY                  InvoiceHeader      Employee
   #36    PRIMARY KEY REFERENCED       InvoiceHeader
   #49    FOREIGN KEY                  InvoiceDetail      Product
   #48    FOREIGN KEY                  InvoiceDetail      InvoiceHeader
   #51    FOREIGN KEY                  ContactCallNotes   Contact
   #50    FOREIGN KEY                  ContactCallNotes   Employee
   #52    FOREIGN KEY                  Contact            Customer
   #37    PRIMARY KEY REFERENCED       Contact
   #38    PRIMARY KEY REFERENCED       Titles
   #53    FOREIGN KEY                  Employee           Titles
   #39    PRIMARY KEY REFERENCED       Employee
   #40    PRIMARY KEY                  StateAbr

   Unique Indexes:
   Index Name         Table Name           Column Name
   ------------------ -------------------- ---------------------------
   UI_Comp            Component            CompDesc


   NOT NULL Constraints:
   Table Name         Column Name
   ------------------ ------------------
```

```
Customer             CustID
Component            CompID
ProdLocation         OnHand
Levels               ModLevel
                     Model
Product              Model
InvoiceHeader        TransID
ContactCallNotes     CallTime
Contact              ContID
Titles               EmpTID
Employee             EmpID
StateAbr             State
RThemes_eXtreme      RTheme
```

LIST PKEYS shows only primary keys. The constraint ID, the type of constraint and if it is referenced, and the table name.

```
R>LIST PKEYS

Id     Type                          Table Name         References
------ ----------------------------- ------------------ ------------
#31    PRIMARY KEY REFERENCED        Customer
#33    PRIMARY KEY REFERENCED        Component
#34    PRIMARY KEY                   Levels
#35    PRIMARY KEY REFERENCED        Product
#36    PRIMARY KEY REFERENCED        InvoiceHeader
#37    PRIMARY KEY REFERENCED        Contact
#38    PRIMARY KEY REFERENCED        Titles
#39    PRIMARY KEY REFERENCED        Employee
#40    PRIMARY KEY                   StateAbr
```

LIST FKEYS shows only foreign keys. The constraint ID, the type of constraint, the table name, and table references are displayed. LIST UKEYS displays similar results.

```
R>LIST FKEYS

Id     Type                        Table Name         References
------ --------------------------- ------------------ ---------------
#41    FOREIGN KEY                 SalesBonus           Employee
#43    FOREIGN KEY                 CompUsed             Component
#42    FOREIGN KEY                 CompUsed             Product
#44    FOREIGN KEY                 ProdLocation         Product
#45    FOREIGN KEY                 Levels               Product
#47    FOREIGN KEY                 InvoiceHeader        Customer
#46    FOREIGN KEY                 InvoiceHeader        Employee
#49    FOREIGN KEY                 InvoiceDetail        Product
#48    FOREIGN KEY                 InvoiceDetail        InvoiceHeader
#51    FOREIGN KEY                 ContactCallNotes     Contact
#50    FOREIGN KEY                 ContactCallNotes     Employee
#52    FOREIGN KEY                 Contact              Customer
#53    FOREIGN KEY                 Employee             Titles
```

LIST INDEXES will display all database indexes, and notes if a unique index is defined with (U) to the left of the index name.

```
R>LIST INDEXES

Number of Indexes in Database RRBYW17 is 54.
   Index Name           Table Name          Column Name
   -------------------- ------------------- --------------------
   CustState            Customer            CustState
(U)UI_Comp              Component           CompDesc
```

LIST CASCADE displays tables with CASCADE, and whether UPDATE, DELETE, or BOTH is enabled.

```
R>LIST CASCADE
   Tables with CASCADE flag in the Database RRBYW17

   Name              Cascade Type
   ----------------- ---------------------------------
   Customer          CASCADE BOTH
   Employee          CASCADE BOTH
   Titles            CASCADE BOTH
```

The LIST command can also be used to display constraints for a single table. When specifying an individual table, the referenced column names are also displayed.

```
R>LIST CONSTRAINTS FOR Employee

         Table Name: Employee

   Id     Type          Column Name(s)  Ref Table Name  Ref Column Name(s)

   ------ ------------- --------------- --------------- ------------------
   #53    FOREIGN KEY   EmpTID          Titles          EmpTID
   #39    PRIMARY KEY   EmpID

   NOT NULL Constraints:
   Table Name         Column Name
   ------------------ ------------------
   Employee           EmpID


R>LIST FKEYS FOR InvoiceHeader

         Table Name: InvoiceHeader

   Id     Type           Column Name(s)  Ref Table Name  Ref Column Name(s)

   ------ -------------- --------------- --------------- ------------------
   #47    FOREIGN KEY    CustID          Customer        CustID
   #46    FOREIGN KEY    EmpID           Employee        EmpID
```

Part VI

6 Removing Constraints

A primary key constraint that is referenced by a foreign key cannot be removed until the foreign key constraint has first been deleted.

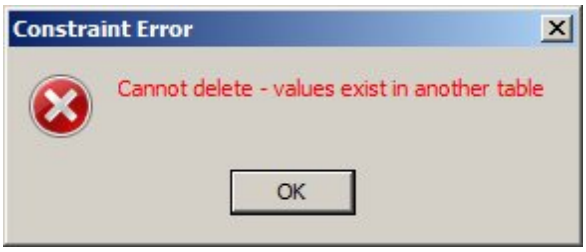
If a column was first defined as not null, and then as a primary key, the not null constraint on the column cannot be removed until the primary key constraint has been removed. Removing a primary key constraint does not remove the NOT NULL part of the constraint. That must be removed separately.

Part VII

7 Constraint Messages

When primary key, foreign key and not null constraints are defined, custom violation messages can be entered. The messages cannot be added or modified after the constraint is defined. The constraint must be deleted and re-defined to add or modify custom messages.

The following is an example of a constraint violation error message.

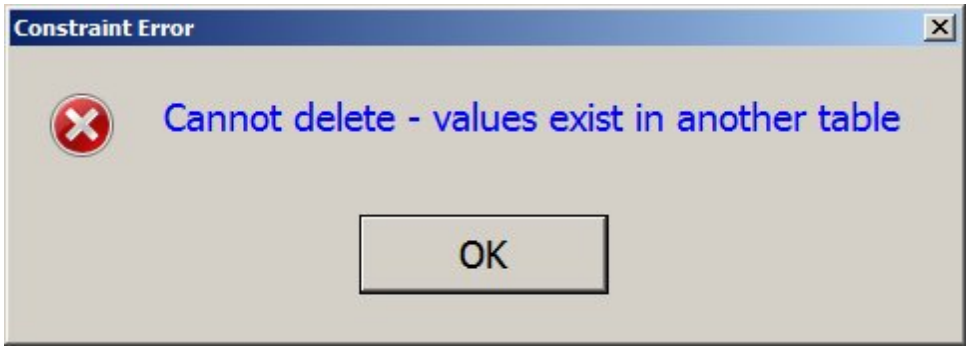


Using two RBTI System Variables, the font size and color can be adjusted for the constraint error message. The variables are:

- RBTI_CEM_FONTSIZE
- RBTI_CEM_FONTCOLOR

Using the following commands, the constraint error message will display a blue font size of 14

```
SET VAR RBTI_CEM_FONTSIZE INTEGER = 14
SET VAR RBTI_CEM_FONTCOLOR TEXT = BLUE
```



Part VIII

8 Useful Resources

. R:BASE Home Page:	http://www.rbase.com
. R:BASE X Home Page:	http://www.rbasex.com
. Up-to-Date R:BASE Updates:	http://www.rupdates.com
. Sample Applications:	http://www.rbasecommunity.com
. General R:BASE Syntax:	http://www.rsyntax.com
. Technical Documents - From The Edge:	http://www.razzak.com/fte
. More Sample Applications:	http://www.razzak.com/sampleapplications
. Education and Training:	http://www.rbaseuniversity.com
. Upcoming Events:	http://www.rbase.com/events
. R:BASE Beginners Tutorial:	http://www.rtutorial.com

Part

9 Feedback

Suggestions and Enhancement Requests:

From time to time, everyone comes up with an idea for something they'd like their software to do differently.

If you come across an idea that you think might make a nice enhancement, your input is always welcome.

Please submit your suggestion and/or enhancement request to the R:BASE Developers' Corner Crew (R:DCC) and describe what you think might make a nice enhancement. In R:BASE, the R:DCC Client is fully integrated to communicate with the R:BASE development team. From the main Menu Bar, choose "Help" > "R:DCC Client". If you do not have a login profile, select "New User" to create one.

If you have a sample you wish to provide, have the files prepared within a zip archive prior to initiating the request. You will be prompted to upload any attachments during the submission process.

Unless additional information is needed, you will not receive a direct response. You can periodically check the status of your submitted enhancement request.

If you are experiencing any difficulties with the R:DCC Client, please send an e-mail to rdcc@rbase.com.

Reporting Bugs:

If you experience something you think might be a bug, please report it to the R:BASE Developers' Corner Crew. In R:BASE, the R:DCC Client is fully integrated to communicate with the R:BASE development team. From the main Menu Bar, choose "Help" > "R:DCC Client". If you do not have a login profile, select "New User" to create one.

You will need to describe:

- What you did, what happened, and what you expected to happen
- The product version and build
- Any error messages displayed
- What computer operating system is in use
- Anything else you think might be relevant

If you have a sample you wish to provide, have the files prepared within a zip archive prior to initiating the bug report. You will be prompted to upload any attachments during the submission process.

Unless additional information is needed, you will not receive a direct response. You can periodically check the status of your submitted bug.

If you are experiencing any difficulties with the R:DCC Client, please send an e-mail to rdcc@rbase.com.

Index

- A -

ALTER TABLE 10

- B -

benefits 4

- C -

CASCADE 8, 17
comparison 6
constraint benefits 4
constraint violation 23
CONSTRAINTS 2, 17
CREATE INDEX 10
CREATE TABLE 10

- D -

Data Designer 10
data integrity 4
defining constraints 10
Delete 8

- E -

error message 23
error messages 10

- F -

FKEYS 17
Foreign Key 2, 5

- I -

INDEXES 17

- L -

LIST 17

- M -

message 23

- N -

New Key 10
Not NULL 2, 6, 10

- P -

PKEYS 17
Primary Key 2, 4

- R -

RBTI System Variables 23
RBTI_CEM_FONTCOLOR 23
RBTI_CEM_FONTSIZE 23
referential integrity 2, 4
removing constraints 21
resources 25

- S -

SET VAR 23

- U -

UKEYS 17
Unique Index 6
Unique Key 2, 6
Update 8

Notes