Generate Test Data

General Methodology

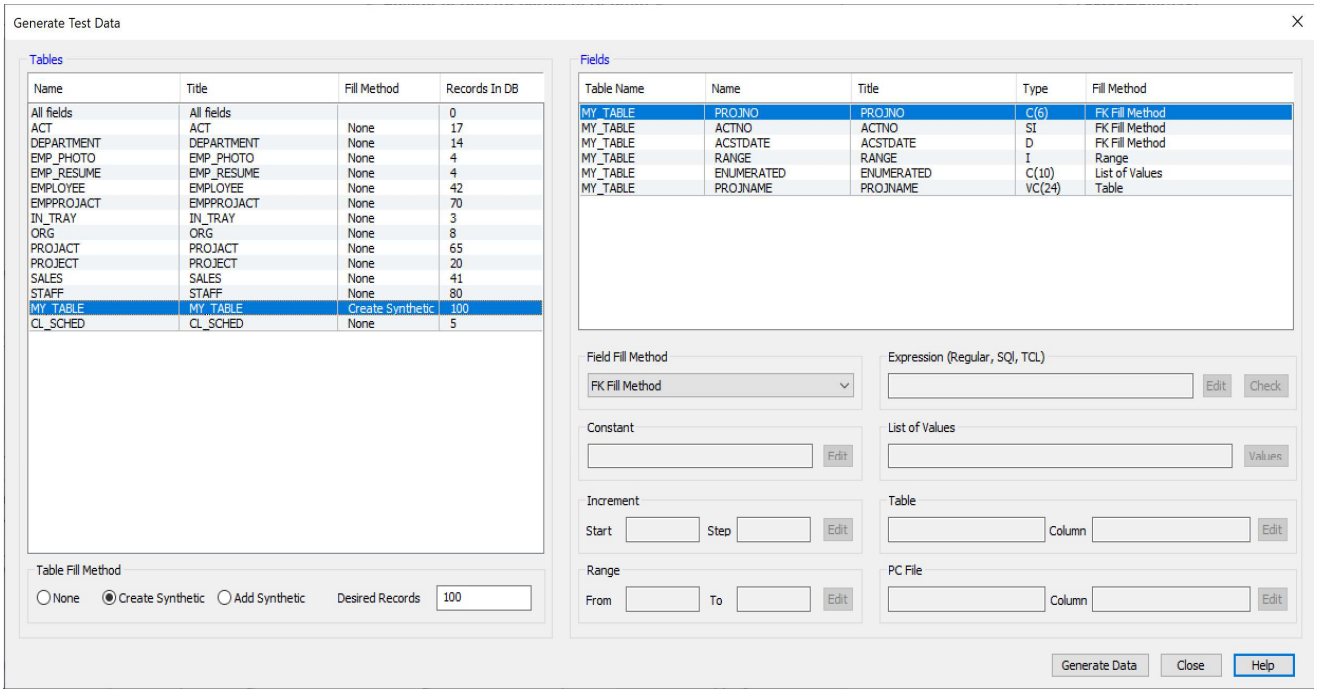
Xcase enables the filling or modification of data into an existing database, which we will refer to as the Target Database. The Model used to generate the Test Data should accurately describe the Target Database. To ensure this, synchronize the database with the model or reverse engineer the database into the model.
Each table in the Target database can be located in a different schema/library and can have a different filling method:

- **None**
  Existing records should be preserved, and fields will not be modified.
- **Sample**
  Existing records in the Target Table will be deleted, and new ones will be added from the Source Table.
- **Use Current**
  Existing records should be preserved, but the content of fields can be modified.
- **Create Synthetic**
  Existing records will be deleted from the Target Table, and automatically generated records will be added to the Target Table.
- **Add Synthetic**
  Automatically generated records will be added to the Target Table.

Note:
If at least one of the Target Tables was set to have the “Sample” Fill Method, records could be added or removed from other tables independently of their Fill Method, to preserve the Referential Integrity of the database.

For each field of each table, you specify a fill method. This allows for obfuscating sensitive data and obtaining a comprehensive yet controlled range of values, enabling thorough testing of your database application.

The User Interface



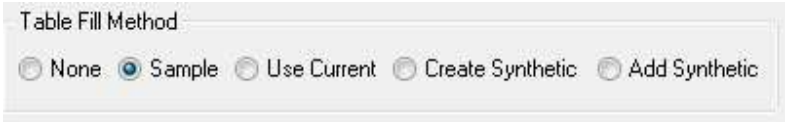
The Table and Field Lists

The “Generate Test Data” dialog displays a list of model tables and their corresponding fields. One or multiple columns can sort both lists in ascending or descending order. Clicking a column title will sort the list according to that column. Clicking again will reverse the sorting order. To order the lists by multiple columns, press the Shift key while clicking the column titles.

For each table, you can select its Fill Method. Note that it is possible to select multiple tables in the list and to assign a Fill method to the selected tables. To assign a fill method for the table, click one of the radio buttons in the “Table Fill Method” group. When a table has been processed, the number of records in the table is displayed in the “Records” column of the table list.

For each field, you can select its Fill Method. This allows you to specify the value of the field in the Test Database table. To assign a fill method for the chosen field, select a Field Fill Method from the Field Fill Method Dropdown.

Table Fill Methods



- **None**
  Existing records will be preserved. New ones will be added only if necessary to maintain the database's integrity. For example, suppose the table is a parent of another table that contains records in the Target database. In that case, the necessary documents will be added to ensure referential integrity is preserved.
- **Create Synthetic**
  Existing records will be deleted from the table, and automatically generated records will be added to the Target Table.

  Add Synthetic

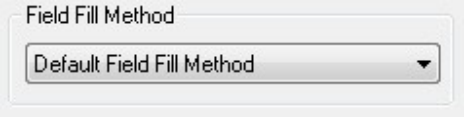
  Automatically generated records will be added to the Target Table.

  Desired Records

  You can specify the number of records to be added in the Target Table in the Desired Records field.

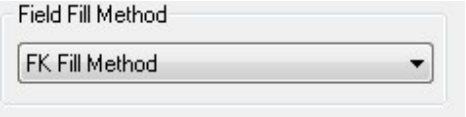
Field Fill Methods

- **Default Field Fill Method**



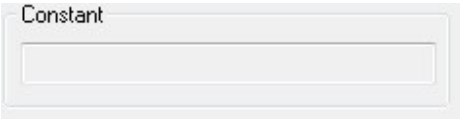
When the Table Fill method is set to “Sample,” the field will retain the same value as its source.
When the Table Fill method is “Use Current,” the field will not be modified. When the Table Fill method is set to “Create Synthetic” or “Add Synthetic,” the Field will be filled with a random value that matches its type.

- **FK Fill Method**



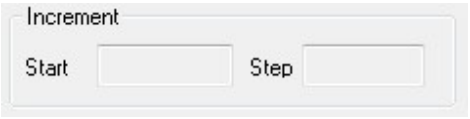
The Field will be filled with the same value as its Parent Field. This is the default when the Field is a Foreign Key.

- **Constant**



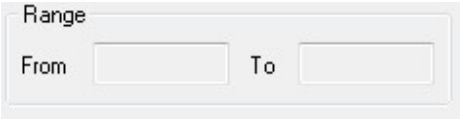
The Field will be filled with the value entered in the Constant Box.

- **Increment**



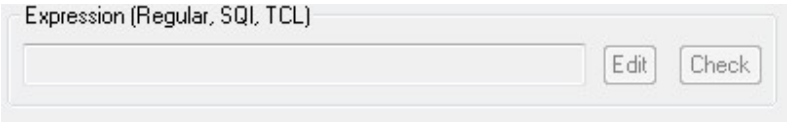
The Field will be filled with a value starting with the “Start” entry and incremented by “Step” as defined in the Increment Box.

- **Range**



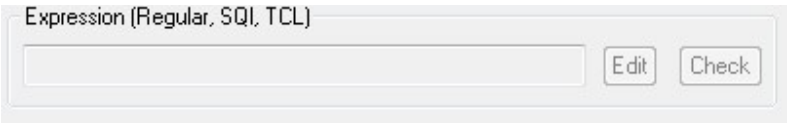
The Field will be filled with a random value from a range as defined in the Range Box.

- **Expression (Regular)**



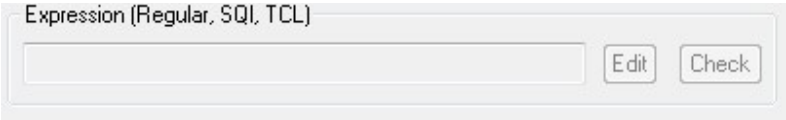
The Field will be filled with a random value matching the Regular Expression defined in the Expression Box. Clicking the Check Button allows you to verify if a syntax error exists. If no error is found, a value will be generated and displayed, enabling you to check if it conforms to your specification.
Example:
5([0-9]){3,3} ([0-9]){4,4} ([0-9]){4,4} ([0-9]){4,4}
This will produce a random Credit Card number starting with the digit 5. Clicking the check button produces a result like: 5890 4328 9854 4378

- **Expression (SQL)**



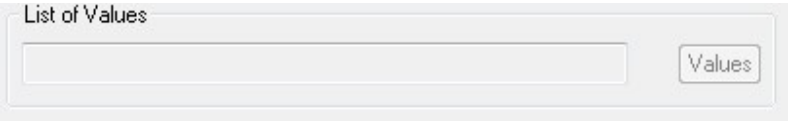
The Field will be filled with a random value picked from the first column returned by the SQL statement defined in the Expression Box. Clicking the Edit button opens a simple editor allowing the entry of the SQL Statement. Clicking the Check Button enables you to verify if a syntax error exists. If no error is found, a value will be generated and displayed, allowing you to check if it conforms to your specification. Example:
SELECT CUSTOMER_NAME FROM MYSCHEMA.CUSTOMER_TABLE WHERE CUSTOMER_TYPE='A'

- **Expression (TCL)**

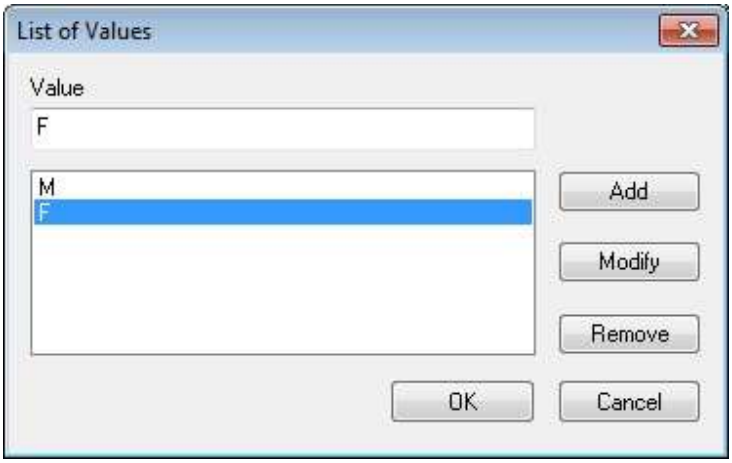


The Field will be filled with the value returned by the TCL expression defined in the Expression Box. Clicking the Edit button opens a simple editor allowing TCL statements. Clicking the checkbox allows you to check for a syntax error; otherwise, a value will be generated and displayed, enabling you to verify whether it conforms to your specification.
The TCL statements can utilize the generate_data_current_value function. The parameter of this function is the name of a field of the Table. It returns the value of that Field for the current record. Example:
>> [expr [generate_data_current_value SALARY] * 2]

- **List of Values**



The Field will be filled with a random value picked from the list of values defined in the List of Values Box. Clicking the Values Button produces a dialog allowing editing of the list. Note that you can enter the value <Null> in the list. This will put a Null in the field when <Null> is randomly selected.

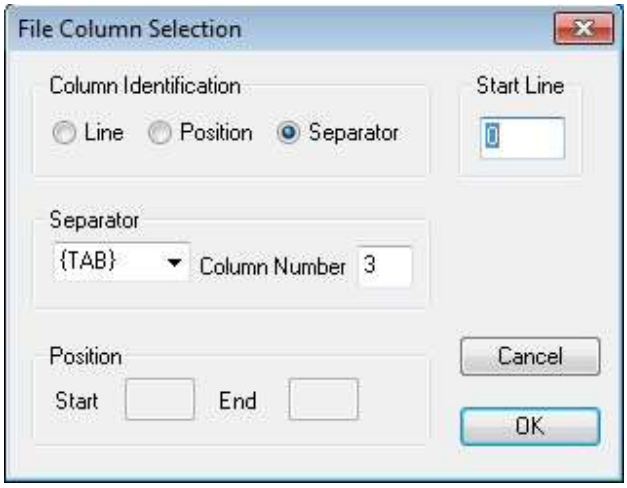


- **Table**

The Field will be filled with a random value picked from a Field of a Model Table defined in the Table Box. Clicking the first "…" button in the box produces a list of the Tables of the Model, and clicking the second "…" button produces a list of the Fields of the selected Table.

- **PC File**

The Field will be filled with a random value picked from one of the columns of the PC File specified in the PC File Box. Clicking the first "…" button in the box allows you to select the PC File, and clicking the second "…" button produces a dialog allowing you to identify the column in the PC File.
In this dialog, you can select how the Column is identified:
Line – the complete line (ended by a carriage return) is considered to be the value
Position – you can specify the starting and ending position of the value
Separator – you can specify which character separates the columns in the file and which column number represents the value
You can also specify from what line the values start, in case the file contains headings for the columns.
Note that you can enter the value <Null> in the file. This will put a Null in the field when <Null> is randomly selected.



- **Null**

The Field will be filled with a Null value. This method is available only if the field is defined as nullable

- **None**

The Field will be filled with its Default value as defined in the Database. If no default value is determined, it will be filled with a Null. Note that this method is available only if the field is nullable or has a Default.

Generate Data Button

Clicking this button initiates the generation of test data. The process starts with a validation of the correctness of the various input parameters in the dialog. If no incorrect parameters are found, Triggers, Journaling, Identity, and Referential Constraints are disabled before generating the data. They will be re-enabled once the data has been generated. During the Generation process, a message window appears displaying a log of the actions performed and their results.

Errors when Generating Test Data

If an error occurs while generating test data, a dialog will appear allowing you to do one of the following:

- Press the "**Stop**" Button to stop the process

- Press the "**Skip Current Table**" to skip the processing on the table on which the error occurred
- Press the "**Continue**" Button to move to the next step in the process

You can also set what should happen when the following error occurs by clicking on one of the three radio buttons in the "On Next Error" Box:

- **Report Error**: The Generate Data Error dialog will appear again
- **Ignore All Errors**: The Generate Data Error dialog will not appear
- **Ignore Same Error**: The Generate Data Error dialog will appear only if the error is different than the last error

Errors when re-enabling Referential Constraints

Once the Generate Data process is completed, the Triggers, Journaling, Identity, and Referential Constraints will be re-enabled. Note that if the Source Database has integrity errors, they may also appear in the Target Database. In that case, depending on the target DBMS, the constraints will be kept disabled, or it will not be possible to create the constraint. If you wish to enable or create those constraints, you should apply the following procedure:

- ### Detect and fix the offending data
  Click the "Check Data Integrity" icon in the main **Xcase** toolbar. Check "Foreign Keys". In the Entities Box, select the Tables that need to be checked. A list of referential integrity issues will appear. You can edit and fix the data till all integrity violations are eliminated.

- ### Synchronize the database with the model
  Once the data has been fixed, click the "Synchronize Database with Model" icon in the main **Xcase** toolbar. This will enable you to generate and submit a script to create or allow referential constraints in the database.