

Application level

Notification

User[] recipients

String notificationType

DateTime notificationTime

Database

NotificationType

VarChar(255) **NotificationTemplateName** (Primary Key)

VarChar(255) **TableName** (it contains the table or view, if values are contained in more table, name)

TemplateFields

VarChar(255) **NotificationTemplateName**

VarChar(255) **FieldName**

Properties, for example

int **SingleField**

Notification workflow is based on the use of templates: a template is basically a String containing a certain number of keys to be replaced with the related table field value.

Example

Input data

TripId: 122

Start: Pisa

Stop: Roma

Template Name: NewTrip

Workflow:

-> execute select **FieldName** from TemplateFields where **NotificationTemplateName**='NewTrip'

query results:

String[] result= {"start","stop"}

String specializedQuery="select ";

while (results..){

specializedQuery+="results[i]+" ";

}

specializedQuery+=" from "+*TableName*;

Name of the table is dependent from the template and/or it could be simplified if another similar structure, containing the relation between templates and views, is created (such as an hash table)*

-> execute specializedQuery

while (results..){

-> execute `simpleReplaceInTemplate (NotificationTemplateName,fieldname[i],results[i])`
//results[i] contains the field values, for example start="Pisa"

`simpleReplaceInTemplate`: this is an example, because it must be managed the case in which a field name is related to multiple values (for example **intermediate stops**)