SQL Intelligent

how to start…

SQL Intelligent is an application for swift and easy publication of SQL queries for an end user within a web environment.

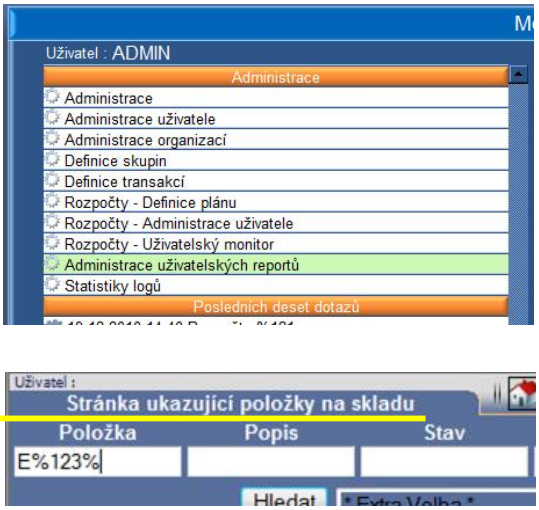
It is necessary to follow rules and procedures mentioned in this manual when entering SQL and creating individual pages (modules).

SQL Intelligent makes it possible to produce a form, which has dynamic features at data generation and display.

Administrator rights are required for work under SQL Intelligent.

1. Creating Basic Entry Fields of a Form

Fig. No. 1



You will produce a form (module) generated by a standard basic SQL query in this part of application. By such form you create reports from a database, to which the form is linked.

Click on ***"Administration of User Reports"*** in the "***Administration***" section, see fig. no.1.
Enter future name of a page in the form, field "**Page**" and "**Page Description**". The text in field "**Page Description**" is shown in the heading of a form (module) and the text in field "**Page**" is shown in the main menu. Further, it is necessary to set a database instance, if more of them are defined, see fig.No.2.

A text field for entering the basic SQL query is shown upon a click in the field below the line. SQL Intelligent provides some processes automatically and that is why **it is necessary to adhere to essential procedures**.

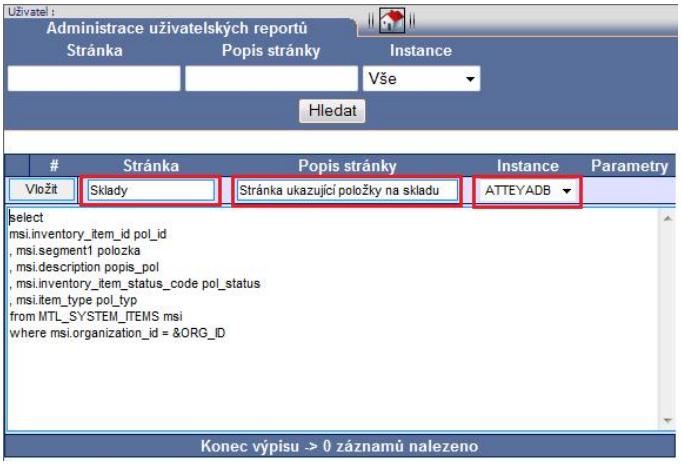


Fig . No. 2

a. **The first column should be a unique line index.** The program will try to prepare summing according to it later on, it will make graphic jumps.

b. **Each column should have its own unique alias**, sorting will be done according to it. If two similar ones are defined, an error will occur in the resulting SQL query.

c. It is possible to use variables *&ORG_ID* and *&SOB_ID* in a report, which applies **only** to an installation with **Oracle E-Business Suite**, that mean organizations or sets of ledgers from the basic combo-box. These combo-boxes will be displayed only if the variable is used.

d. **All tables should be used with the APPS prefix,** e.g. APPS.MTL_SYSTEM_ITEMS. They can function even without it, but only in case that ApplStream® uses directly the APPS database user for connecting to the database (applies to *Oracle E-Business Suite*).

Within the given example (Fig.No.2) we define page "Inventory" and we start with the basic SQL only for an item. All, what we enter now, will be a minimum of what will be visible on the page.
We have used the &ORG_ID parameter and so a combo-box for organizations has been also added to the resulting form.
A report is created upon pressing the "Enter" key.

It is in the “Correction” state now and all entered parameters can be modified *fig. No. 3*.

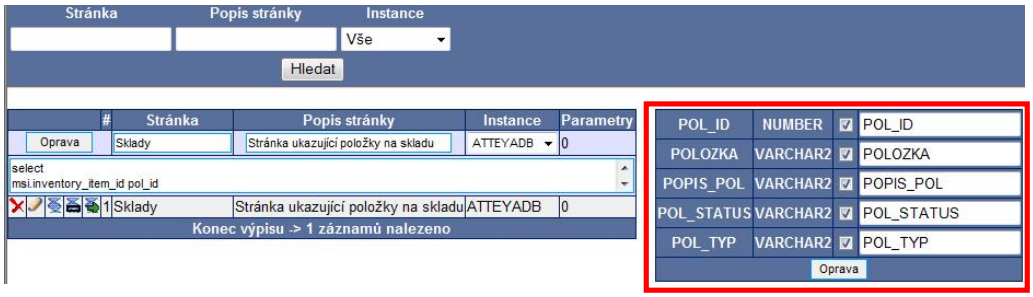
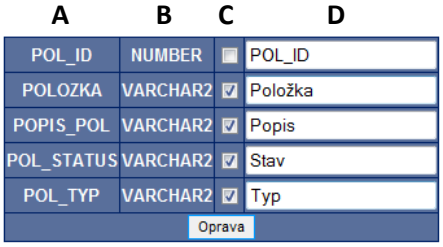


Fig. No. 3

- Page
- Page Description
- Instance
- Basic SQL

This can be done after a change by the “Correction” key on the left side in the form.
Figure no. 3 shows all parameters, which will create columns in the report.
We enter a description with which we want to denominate the column in the report into the white field.
For saving, it is necessary to push the “Correction” key on the right below the listing of parameters.

Column „A“ is the SQL identification of a parameter
Column „B“ is the type of a parameter
Column „C“ the visibility of a column in a report
Column „D“ the denomination of column in a report.



If you entered a unique line index, then it is appropriate to make it invisible by scratching it out from the list (e.g. POL_ID). This way you can make invisible any number of future columns in a listing.
Only the first of them will be used for summing the report.

Upon clicking the “Correction” key it is necessary to check whether all parameters were downloaded correctly.

Click the page denomination (“Inventory” in our case) in the created form, see fig. 4.

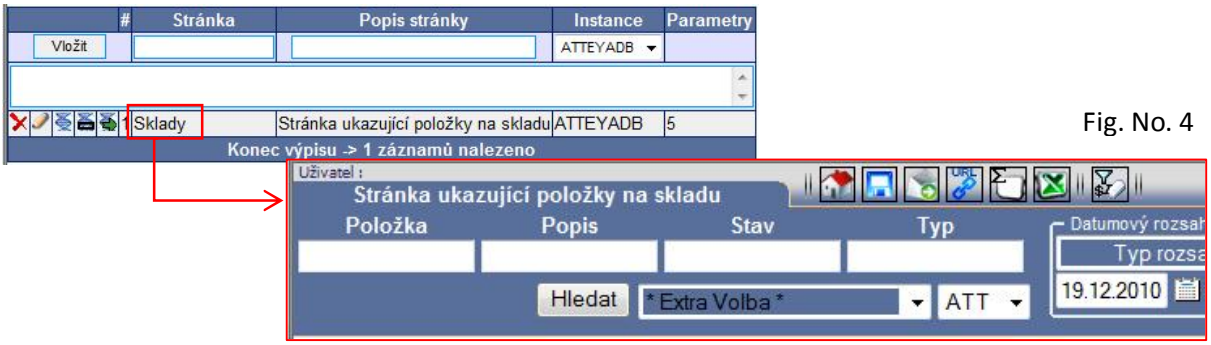
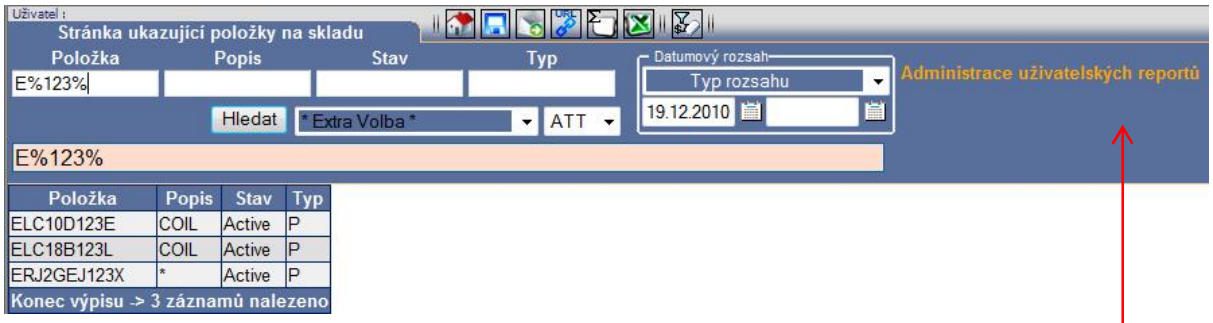


Fig. No. 4

The newly created form provides standard possibilities and functions for data searching now, the same way as the ApplStream® application.



The application administrator has a possibility to click the orange link, which brings him directly to the page administration without the need to click through the Menu all the time.

Section **"Reports"** is created in the Menu, where a reference to the new page is shown, **"Inventory"** in our case. There is an icon in front of the page name, which symbolizes user developed pages. Access rights to individual pages are assigned under the **"User Administration".**

2. Adding Dynamic Features to the Report

This part of the application gives you the possibility to improve the functionality of the basic SQL by **dynamic elements.** The form acquires a new freely definable dynamic field, by which you can modify display of the report. This feature is unique for the **DNA Technology** and is used by all products of ATTEYA Group.

We have got our own defined first page "**Inventory**", which has a basic field in "**User Report Administration"**. Now, we will expand the page by a dynamic SQL feature.

Basic description of features: 

First icon, Delete report 

Second icon, Correction 

a. Third icon, Genome definition  We will add new features to SQL with the help of this function. These features are activated by a user on the form page in a standard way.

b. Fourth icon, the Dynamic Field definition.  A dynamic field makes it possible to add other fields/columns, which are not present in the basic SQL. This way the output of the report can be extended or its features modified. In such case it is necessary to use predefined SQL genomes (point a.), and even conditions (point c.) in some cases.

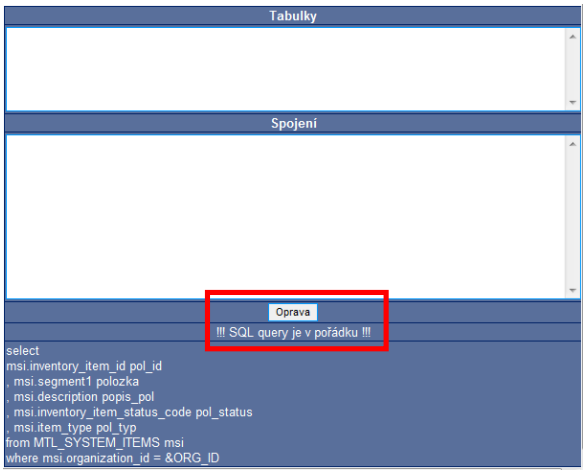
c. Fifth icon, the Link conditions definition.  Conditions of a link make it possible to define cases when a specific genome is to be activated in the case of activation of the search field.

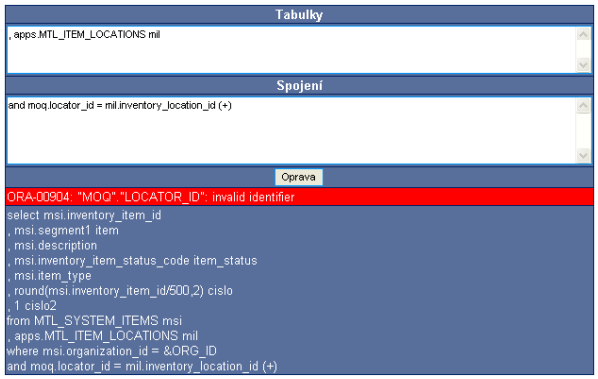
a. The Definition of a Genome

For the definition of a new dynamic genome it is necessary to enter a "Denomination" and a "Description".
The genome is transferred to the correction mode upon clicking the "Enter" key. It is possible to enter tables and links here, see fig. No. 5.

Fig. No. 5

Fig. No. 6

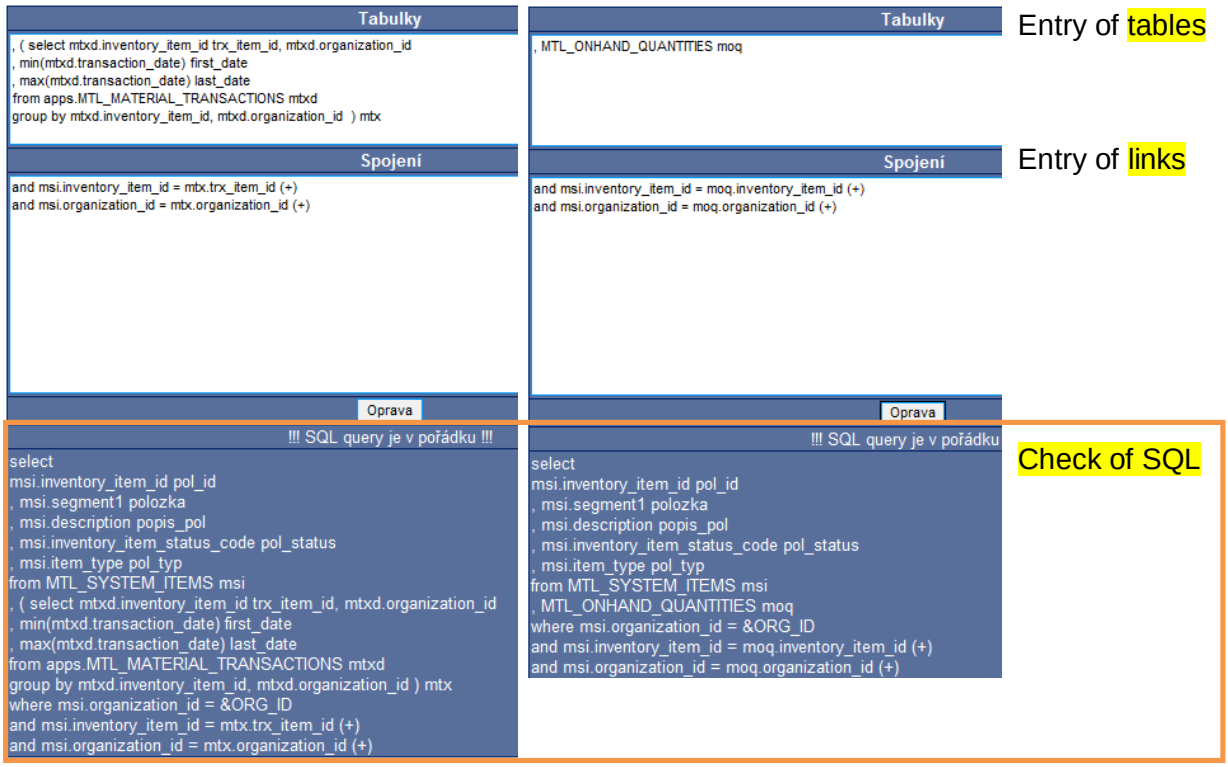




Incorrect SQL

The key "Correct" enters new features.
SQL Intelligent assigns newly defined features to the original SQL query and executes a check whether it is in order. In our case the check was performed without a problem, see fig. No. 6. It is necessary **to adhere to bracket conventions** when creating embedded selects. It can happen that the query is not correct in the case of higher number of mutually linked genomes. In such case it would not be a critical problem.

In our example we add a table of Inventories and values of quantities inventories. Further we define the first and the last transaction per an item.

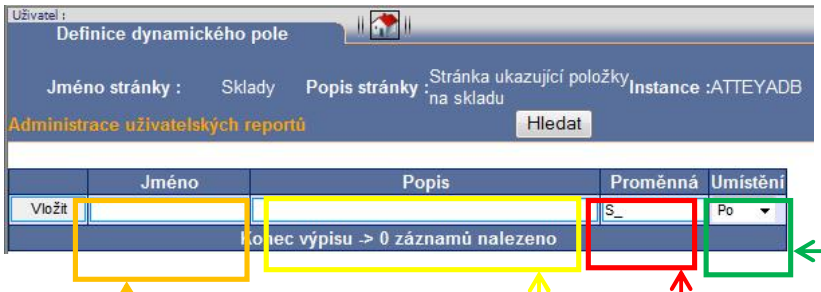


By this we have prepared new features, which can be used in the **"Dynamic Field Definition"** function.

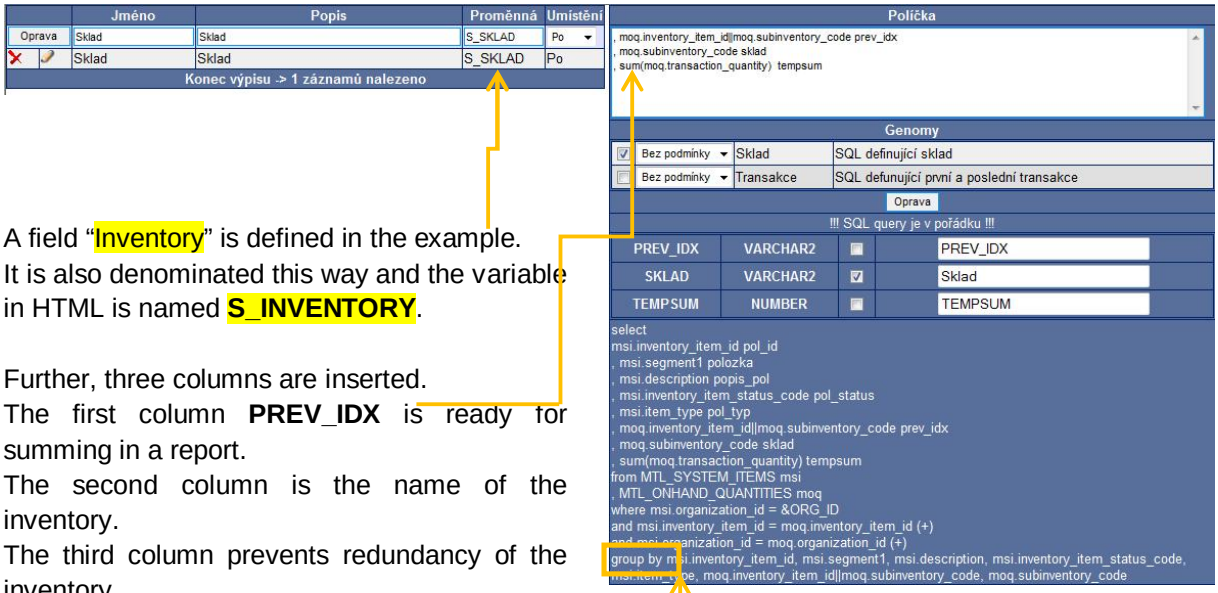
b. The Definition of a Dynamic Field

STEP 1

We will get back to Page entry by clicking the orange text "Administration of User Reports".



"Dynamic Field Definition" function, here we will enter such name of the entry field into the "Denomination" field, which we want to be used in the form.
Description is not shown anywhere else and is for an administrator only.
It is necessary to define a variable, which will be used in the page. **It is recommended to use uppercase and S and an underline if possible.**
It is possible to specify whether columns will be displayed in front of columns of the basic SQL or after them in the "Position" column.

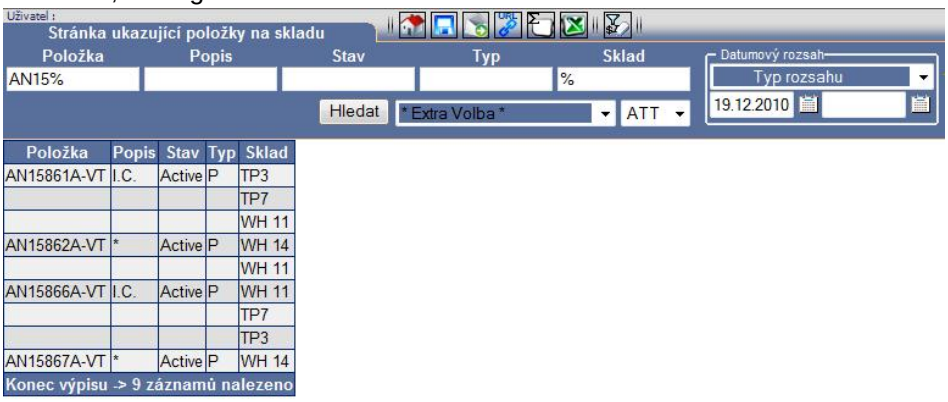


A field "Inventory" is defined in the example.
It is also denominated this way and the variable in HTML is named **S_INVENTORY**.

Further, three columns are inserted.
The first column **PREV_IDX** is ready for summing in a report.
The second column is the name of the inventory.
The third column prevents redundancy of the inventory.

Please notice, at our exemplary SQL, that the **GROUP BY** clause was automatically added at the bottom. GROUP BY is added because **SQL Intelligent recognized the SUM() function**, which requires this.

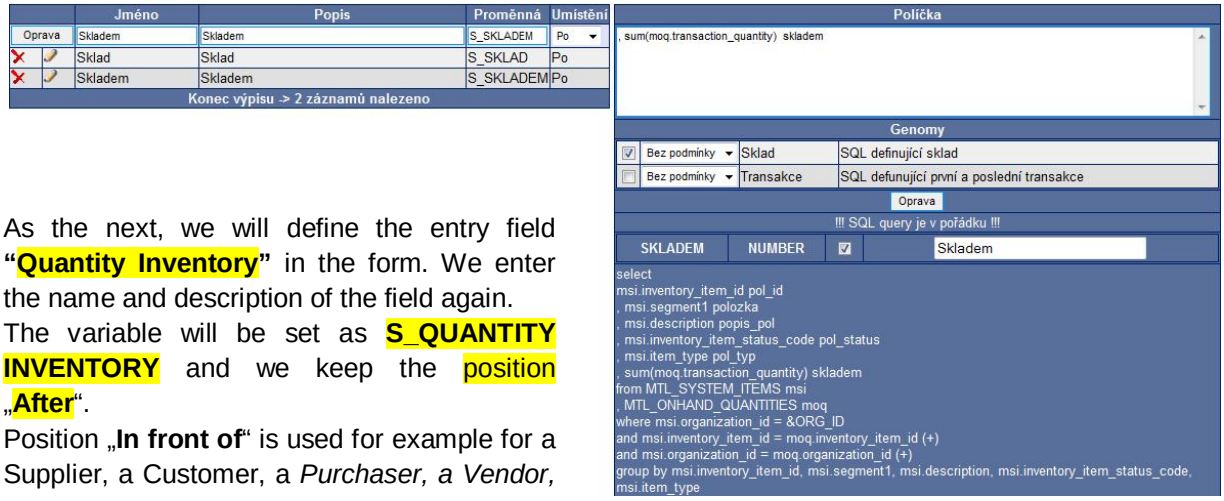
For majority of added fields it is necessary to tick off genomes, which will be incorporated into the original SQL. In our example a genome is added, which we have named "Inventory" and it has no condition. If the SQL is in order, a list of found parameters … new columns accordingly will be shown. We will leave ticked those, which we want to see in the listing and we will overwrite their denomination. This denomination will be the one, which will be show in the listing on a page. One field can thus activate display of any number of columns. We can test the new field "Inventory" immediately after the definition, see fig. No. 13.



Položka	Popis	Stav	Typ	Sklad
AN15861A-VT	I.C.	Active	P	TP3
				TP7
				WH 11
AN15862A-VT	*	Active	P	WH 14
				WH 11
AN15866A-VT	I.C.	Active	P	WH 11
				TP7
				TP3
AN15867A-VT	*	Active	P	WH 14

Konec výpisu -> 9 záznamů nalezeno

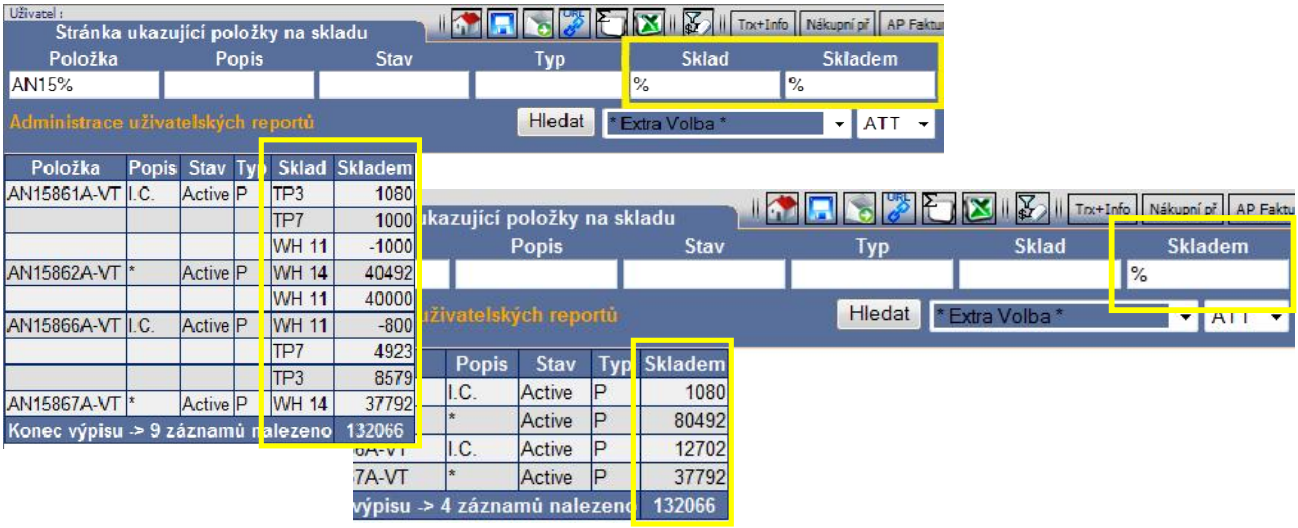
STEP 2



As the next, we will define the entry field **"Quantity Inventory"** in the form. We enter the name and description of the field again.
The variable will be set as **S_QUANTITY INVENTORY** and we keep the position „**After**".
Position „**In front of**" is used for example for a Supplier, a Customer, a *Purchaser, a Vendor,*

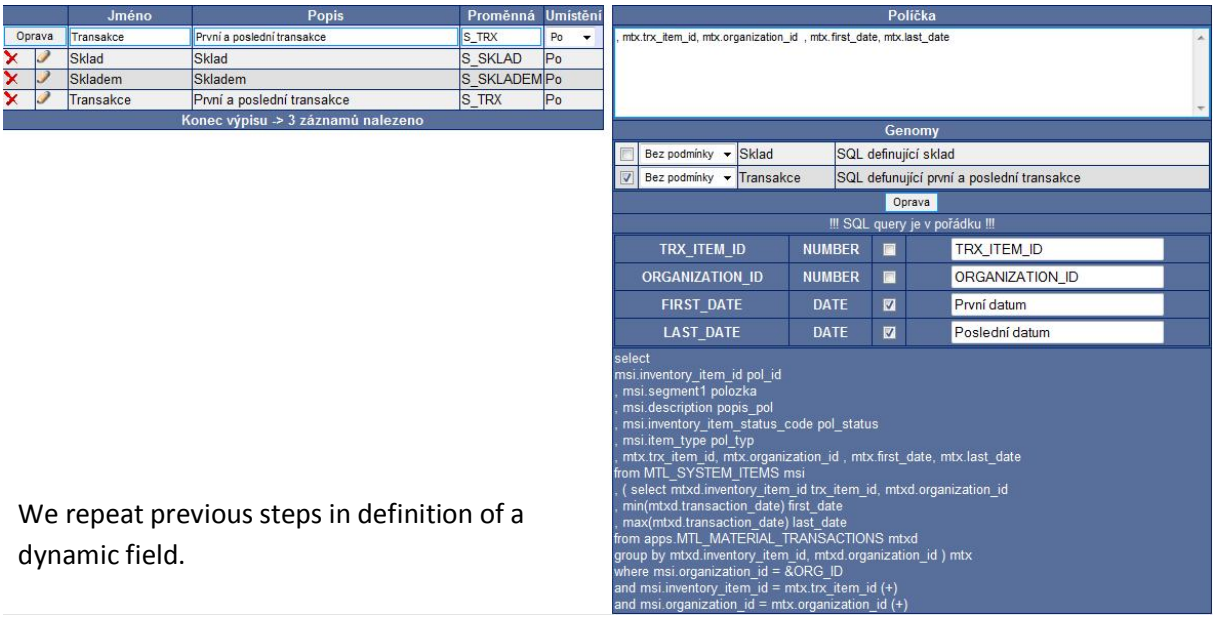
We add only one column to the text field for columns, namely the summary of transaction quantity. Again, we activate the previously defined genome **"Inventory"** and we select "Without condition". We rename the column to **"Quantity Inventory"** after the approval of the SQL.

Additional combinations with the **"Quantity Inventory"** entry field are now available within the form on the page



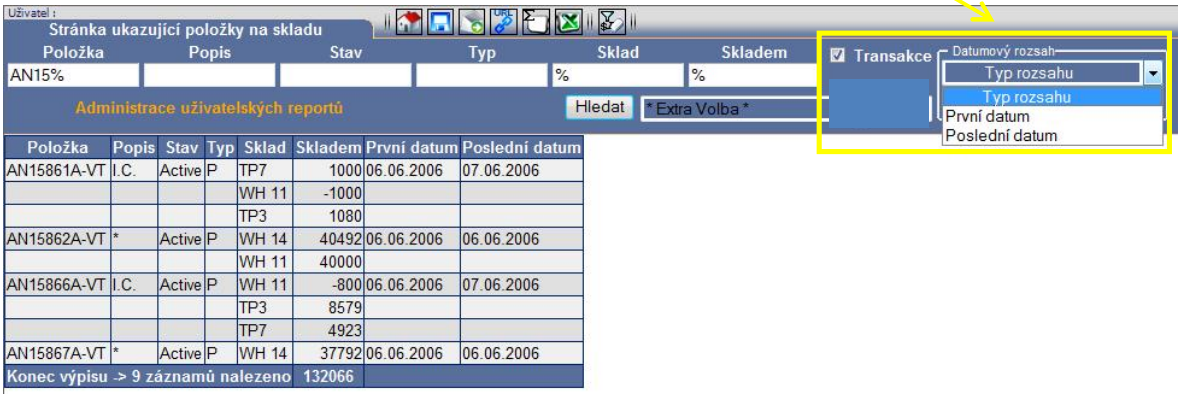
STEP 3

As a next function we will create a display of the first and the last transaction.



We repeat previous steps in definition of a dynamic field.

As none of displayed columns is either a number or a string, no entry field of the form is created but a check box. Both dates are automatically shown also in a combo-box "Type of Range" and it is possible to search according to them.

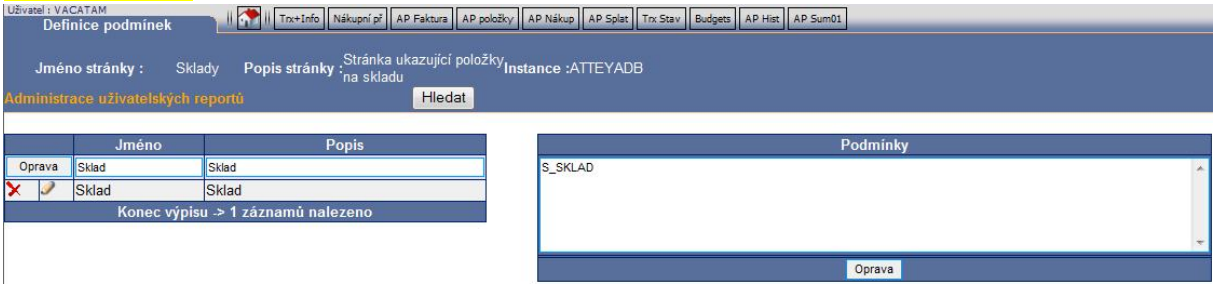


The current version of SQL Intelligent sorts columns in the order as they are defined. There are 7 entry fields on one line in the basic form. Additional fields are always created on a new line. Check boxes will be added by twos, one above the other, into a line, but if they are more than free positions, an intermediate line will be created.

C. The Definition of Conditions

Sometimes it is necessary to create more SQL genomes of a like type and link them according to conditions.
Then it is necessary to use the **"Definition of Conditions"** in the **"Administration of User Reports".**

Example:
As the picture shows, it possible to define for example a condition "Inventory", which tells us, whether the field "Inventory" is activated. For this it is enough to fill out the "Denomination" and "Description" the same way as in the previous steps and then add the condition consisting only of the string S_INVENTORY. This is the simplest condition.



However, it is possible to write down arbitrary mathematical expressions into such conditions, for example this way:

((S_INVENTORY) || S_QUANTITY INVENTORY=3 || S_QUANTITY INVENTORY>=5) || (S_QUANTITY INVENTORY =1) && S_TRX = ON

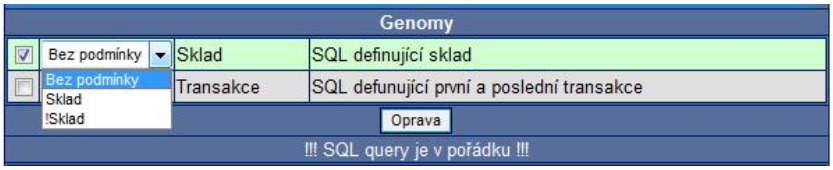
The character || means logical OR, character && means logical AND.
It is important to adhere to correct use of brackets.

This expression means that it is valid in the following case:

- variable S_INVENTORY is activated
- or the value in the field S_QUANTITY INVENTORY is 3, or is higher than, or equals 5.
- or the value of S_QUANTITY INVENTORY equals 1 and the check box S_TRX is activated at the same time.

This is how it is necessary to use check box queries of conditions.

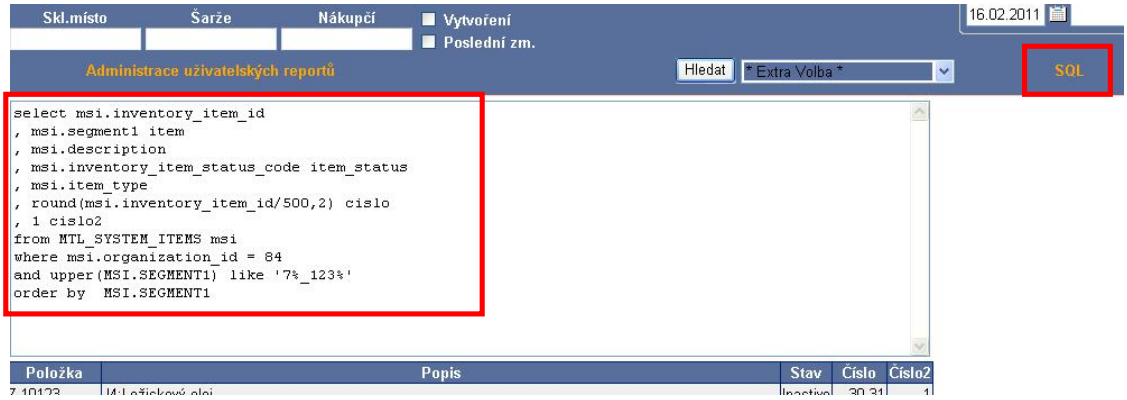
Conditions are consequently shown upon definitions of fields both in positive and in negative versions.



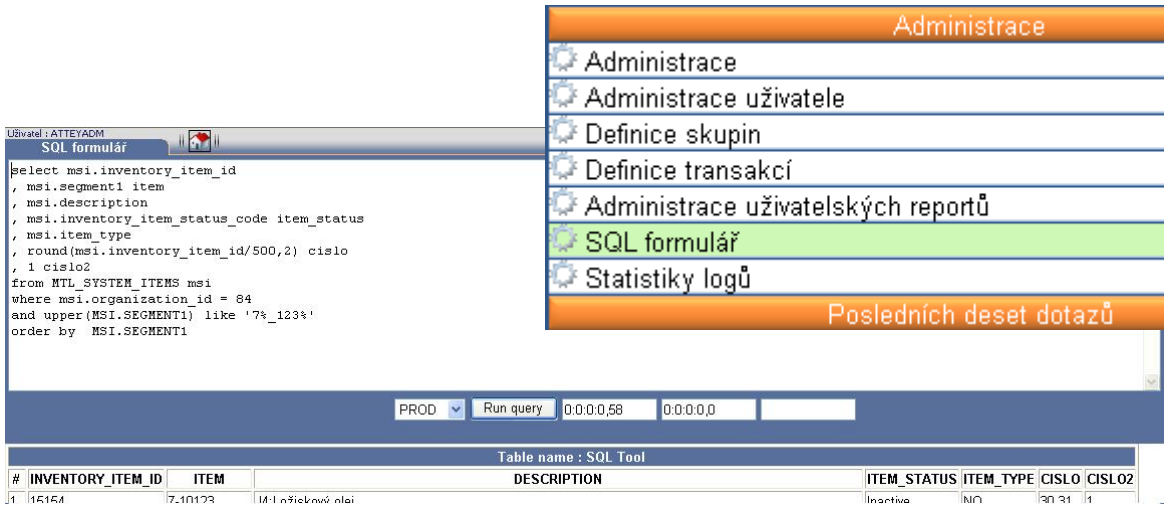
The use of a condition is usually given by a change of linking parameters.
If we, for example, created an internal select with a dependency on an item and an inventory and we subsequently added an "Inventory Position" into the report, then we would be obliged to create a second genome, which would be dependent on an item, an inventory and an inventory position.
Then we would have to activate genomes for prepared conditions in the definition of fields.

3. Tools for Administration and Management of Reports

Each SQL Intelligent administrator has a possibility to have the used SQL query written up from a report. A window containing the SQL is displayed upon clicking the orange SQL link.



The obtained SQL can be checked in the SQL form.



If the database query works in this form, it is also necessary to check whether final columns/parameters have **unique aliases**.

Attention, this applies to columns used in internal SQL too.

In case both conditions are fulfilled, the SQL must function in the SQL Intelligent application.