**Описание таблиц и работы проверок**

Все таблицы лежат в **БД DWMON** в схеме MONITORING (monitoring // ware22mon):

DWMON=(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=exa3-scan)(PORT=1521))(CONNECT\_DATA=(SERVER=dedicated)(SERVICE\_NAME=DWMONs)))

*(если tns устарела обратиться к Дмитрию Кузнецову)*

Таблица MONITORING.DQ\_DETAIL – таблица результатов всех проверок

Таблица MONITORING.DQ\_PARAMS\_CONTROL – информация о запусках потоков загрузок. Записи в этой таблице вставляются сессиями в Информатике следящего и запускающего проверку потока.

Таблица MONITORING.DQ\_LIST\_CONTROL – содержит информацию о проверке. Название, место в общем списке и sql-запрос.

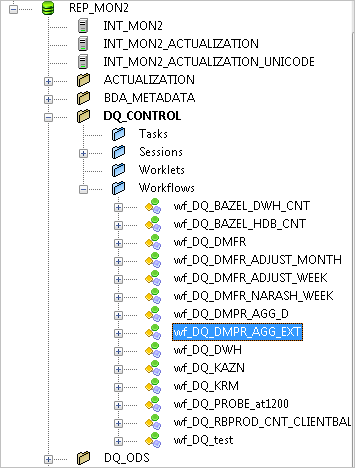
Таблица MONITORING.DQ\_HEADINGS\_CONTROL – содержит описание заголовков для вывода результатов проверок в общем списке и рассылке.

Таблица MONITORING. DQ\_CONTROL\_MAIL – содержит список почтовых адресов пользователей, получающих отчеты о работе проверок.

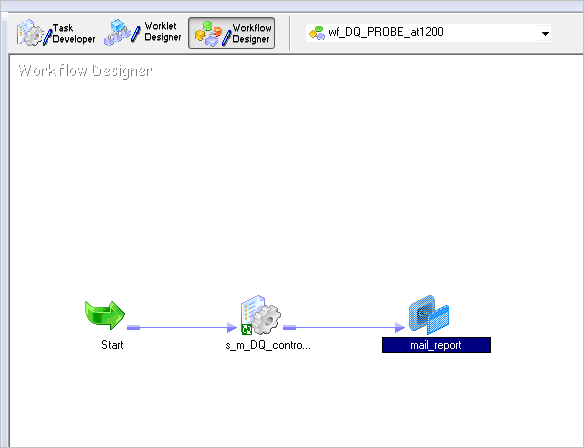
Процедура procedure MONITORING.DQ\_RUN\_CONTROL – запускает проверки по номеру. Вызывается в Информатике сессией проверки.

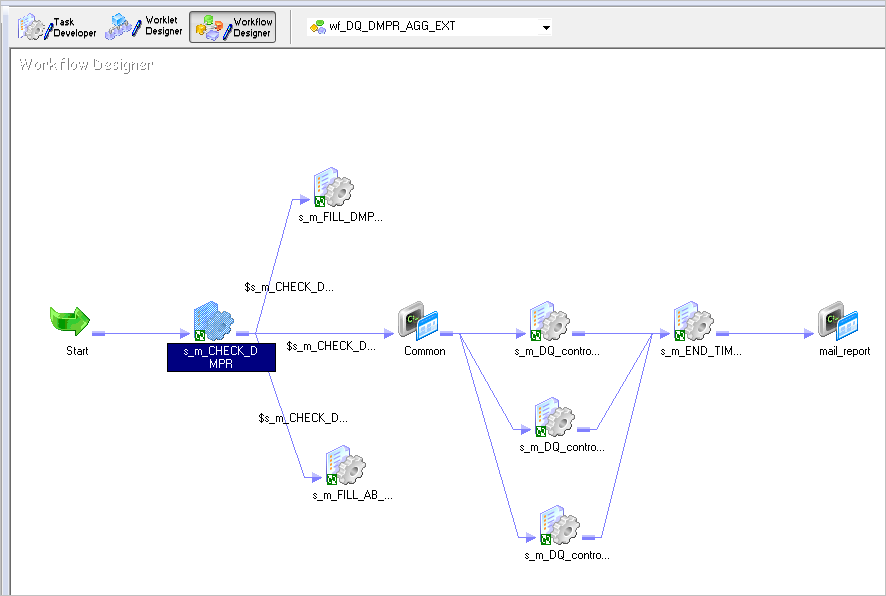
Как работает проверка:

Все потоки с проверками находятся REP\_MON2 / DQ\_CONTROL:



Есть проверки, которые запускаются просто по шедулеру:



И есть, которые запускаются после завершения работы определенного потока:

Описание следящих потоков (автоматический запуск проверок после завершения работы определенного потока)

Потоки стартуют каждые 15 минут.

Сессия **s\_m\_CHECK\_........** также как и поток запускается каждые 15 минут и выполняет запрос:

SELECT CTL\_WORKFLOW\_NAME, D.START\_TIME, DQ\_RUN, M.END\_TIME, SYSDATE DQ\_RUN\_TIME

FROM DQ\_PARAMS\_CONTROL D

JOIN

(SELECT SERVER\_NAME, WORKFLOW\_NAME, START\_TIME, END\_TIME,

DECODE (RUN\_STATUS\_CODE,

1, 'Suceeded',

2, 'Disabled',

3, 'Failed',

4, 'Stopped',

5, 'Aborted',

6, 'Running',

7, 'Suspending',

8, 'Suspended',

9, 'Stopping',

10, 'Aborting',

11, 'Waiting',

12, 'Scheduled',

13, 'Unscheduled',

14, 'Unknown',

15, 'Terminated',

'Unknown') AS STATUS

FROM REP\_EX\_PROD.OPB\_WFLOW\_RUN@EXINFPRD

WHERE SERVER\_NAME='INT\_EX\_PROD'

AND WORKFLOW\_NAME IN ('WF\_REG\_DMPR\_LOADING\_AGG\_EXT')

AND START\_TIME>=SYSDATE-7

ORDER BY START\_TIME DESC ) M

ON D.CTL\_WORKFLOW\_NAME = M.WORKFLOW\_NAME

AND D.START\_TIME = M.START\_TIME

AND M.SERVER\_NAME = 'INT\_EX\_PROD'

AND d.dq\_run = 'N'

AND M.STATUS = 'Suceeded'

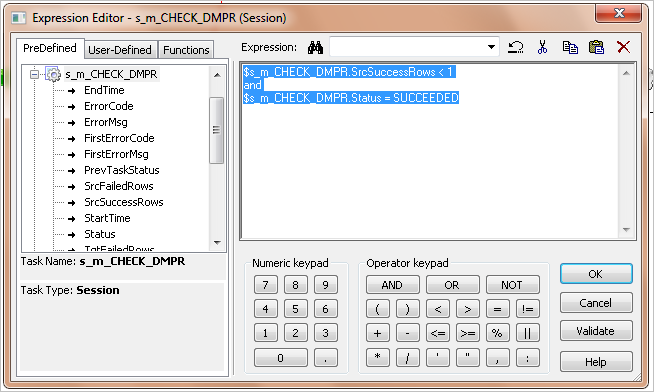
Запрос выдает строку, если завершил работу поток WF\_REG\_DMPR\_LOADING\_AGG\_EXT(например) и есть строка в таблице DQ\_PARAMS\_CONTROL, с записью о том, что этот поток запустился и флаг dq\_run = ‘N’



В этот же момент флаг dq\_run меняется мапингом на = ‘Y’.

Если соответствующей строки в таблице нет или поток не работает/не завершился, то сессия **s\_m\_CHECK\_........** ничего не выдаст и сработают сессии **s\_m\_FILL\_.....** и **s\_m\_FILL\_AB.....**

т.к. условием для них является:



Сессия **s\_m\_FILL\_.....** выполняет insert / update в таблицу DQ\_PARAMS\_CONTROL на основании запроса:

SELECT WORKFLOW\_NAME, START\_TIME, END\_TIME

FROM REP\_EX\_PROD.OPB\_WFLOW\_RUN@EXINFPRD

WHERE SERVER\_NAME='INT\_EX\_PROD'

AND WORKFLOW\_NAME IN ('WF\_REG\_DMPR\_LOADING\_AGG\_EXT')

AND START\_TIME between SYSDATE-7 and SYSDATE

т.е. берет информаци. о запусках потока за неделю и вставляет в таблицу. Если строки уже были они остаются как есть, а новые добавляются:



Сессия **s\_m\_FILL\_AB.....** содержит запрос

SELECT CTL\_WORKFLOW\_NAME, D.START\_TIME, DQ\_RUN, M.END\_TIME

FROM DQ\_PARAMS\_CONTROL D

JOIN

(SELECT SERVER\_NAME, WORKFLOW\_NAME, START\_TIME, END\_TIME,

DECODE (RUN\_STATUS\_CODE,

1, 'Suceeded',

2, 'Disabled',

3, 'Failed',

4, 'Stopped',

5, 'Aborted',

6, 'Running',

7, 'Suspending',

8, 'Suspended',

9, 'Stopping',

10, 'Aborting',

11, 'Waiting',

12, 'Scheduled',

13, 'Unscheduled',

14, 'Unknown',

15, 'Terminated',

'Unknown') AS STATUS

FROM REP\_EX\_PROD.OPB\_WFLOW\_RUN@EXINFPRD

WHERE SERVER\_NAME='INT\_EX\_PROD'

AND WORKFLOW\_NAME IN ('WF\_REG\_DMPR\_LOADING\_AGG\_EXT')

AND START\_TIME>=SYSDATE-7

ORDER BY START\_TIME DESC ) M

ON D.CTL\_WORKFLOW\_NAME=M.WORKFLOW\_NAME

AND D.START\_TIME=M.START\_TIME

AND M.SERVER\_NAME='INT\_EX\_PROD'

AND d.dq\_run='N'

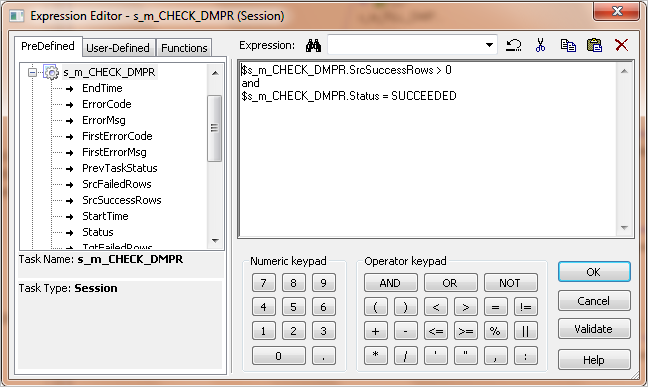
AND WORKFLOW\_NAME='WF\_REG\_DMPR\_LOADING\_AGG\_EXT'

AND M.STATUS='Aborted'

и меняет флаг dq\_run на = ‘А’ если поток был зааборчен:



Если сессия **s\_m\_CHECK\_...** выдала строку, то срабатывает условие:



и выполняется пустой cmd Common , который нужен только как общая точка всех проверок и

дальше выполняется запуск проверок в сессиях вида:



в которых выполняется процедура **call MONITORING.DQ\_RUN\_CONTROL(1956);** выполняющая запрос и инсерт из таблицы DQ\_LIST\_CONTROL для этой проверки.



Когда все проверки отработают выполняется:

Сессия **s\_m\_END\_TIME\_.....** которая на основании запроса:

SELECT CTL\_WORKFLOW\_NAME, START\_TIME, DQ\_RUN, END\_TIME, DQ\_RUN\_TIME,

sysdate DQ\_RUN\_END\_TIME

FROM DQ\_PARAMS\_CONTROL

WHERE CTL\_WORKFLOW\_NAME = 'WF\_REG\_DMPR\_LOADING\_AGG\_EXT'

AND DQ\_RUN\_TIME IN (SELECT MAX(DQ\_RUN\_TIME)

FROM DQ\_PARAMS\_CONTROL WHERE CTL\_WORKFLOW\_NAME = 'WF\_REG\_DMPR\_LOADING\_AGG\_EXT')

заполняет в таблице DQ\_PARAMS\_CONTROL время окончания работы проверок



Работает поток следующим образом:

При запуске потока загрузки (например ядра WF\_CTL\_CORE\_LOADING) в таблице [**REP\_EX\_PROD.OPB\_WFLOW\_RUN@EXINFPRD**](mailto:REP_EX_PROD.OPB_WFLOW_RUN@EXINFPRD) появляется строчка с именем потока, датой запуска и состоянием ‘Running’.

Сессия **s\_m\_CHECK\_...** ничего не вернет и запустится сессия **s\_m\_FILL\_...** которая выберет из таблицы **OPB\_WFLOW\_RUN** информацию о потоке WF\_CTL\_CORE\_LOADING за последнюю неделю и вставит строку о последнем запуске в таблицу DQ\_PARAMS\_CONTROL.

Новая строка содержит время старта потока WF\_CTL\_CORE\_LOADING и значение ‘N’ в столбце DQ\_RUN (установлено дефолтное значение ‘N’ при вставке записи таблицу.

Поток будет запускаться каждые 15 минут и сессия s\_m\_FILL будет срабатывать каждый раз пока процесс загрузки не завершиться и значение в таблице **OPB\_WFLOW\_RUN** не измениться на Succeeded.

Как только это произойдет сработает мапинг m\_CHECK сессии s\_m\_CHECK, который изменит флаг ‘N’ на ‘Y’, запишет время окончания работы потока END\_TIME и время начала работы проверок DQ\_RUN\_TIME в строке таблицы DQ\_PARAMS\_CONTROL, сессия s\_m\_CHECK выдаст одну строку и процесс пойдет по пути выполнения CMD и сессий проверок.

Сведения о таблице MONITORING.DQ\_DETAIL:

Создание:

CREATE TABLE MONITORING.DQ\_DETAIL

(

XK NUMBER,

DEFAULTST\_FLAG CHAR(1 BYTE) NOT NULL,

DESCRIPTION VARCHAR2(255 BYTE),

DQ\_CONTROL\_UK NUMBER NOT NULL,

ERROR\_FLAG CHAR(1 BYTE),

REPORT\_TIME TIMESTAMP(6) NOT NULL,

VALUE\_DATE DATE NOT NULL,

RESULT\_DATE1 DATE,

RESULT\_DATE2 DATE,

RESULT\_DATE3 DATE,

RESULT\_DATE4 DATE,

RESULT\_NUM1 NUMBER,

RESULT\_NUM2 NUMBER,

RESULT\_NUM3 NUMBER,

RESULT\_NUM4 NUMBER,

RESULT\_CHAR1 VARCHAR2(255 BYTE),

RESULT\_CHAR2 VARCHAR2(255 BYTE),

RESULT\_CHAR3 VARCHAR2(255 BYTE),

RESULT\_CHAR4 VARCHAR2(255 BYTE),

RESULT\_CHAR5 VARCHAR2(255 BYTE),

RESULT\_CHAR6 VARCHAR2(255 BYTE),

RULE\_DESCRIPTION VARCHAR2(255 BYTE)

)

Для ключа XK создал сиквенс и тригер на авто добавление значения при вставке строки

-------------------------------------------------------------------------------------------

CREATE SEQUENCE MONITORING.SEQ\_DQ\_DETAIL

START WITH 71

MAXVALUE 9999999

MINVALUE 1

NOCYCLE

NOCACHE

ORDER;

-------------------------------------------------------------------------------------------

CREATE OR REPLACE TRIGGER MONITORING.TR\_DQ\_DETAIL

before insert ON MONITORING.DQ\_DETAIL for each row

declare

*-- local variables here*

v\_id number;

begin

select (SEQ\_DQ\_DETAIL.NEXTVAL)

into v\_id

from dual;

:new.XK := v\_id;

end TR\_DQ\_DETAIL;

/

-------------------------------------------------------------------------------------------

Для того чтобы в таблице не хранились записи глубже трех месяцев сделал джоб,

который запускается каждый день и удаляет все строки **<** SYSDATE – 90

BEGIN

SYS.***DBMS\_SCHEDULER.CREATE\_JOB***

(

job\_name => 'MONITORING.DELETE\_DQ\_DETAIL'

,start\_date => TO\_TIMESTAMP\_TZ('2017/10/05 14:00:00.000000 +04:00','yyyy/mm/dd hh24:mi:ss.ff tzr')

,repeat\_interval => 'FREQ=DAILY; BYHOUR=16; BYMINUTE=00; BYSECOND=0'

,end\_date => NULL

,job\_class => 'DEFAULT\_JOB\_CLASS'

,job\_type => 'PLSQL\_BLOCK'

,job\_action => 'BEGIN

DELETE FROM MONITORING.DQ\_DETAIL

WHERE REPORT\_TIME < SYSDATE - 90;

COMMIT;

END;'

,comments => NULL

);

SYS.***DBMS\_SCHEDULER.SET\_ATTRIBUTE***

( name => 'MONITORING.DELETE\_DQ\_DETAIL'

,attribute => 'RESTARTABLE'

,value => FALSE);

SYS.***DBMS\_SCHEDULER.SET\_ATTRIBUTE***

( name => 'MONITORING.DELETE\_DQ\_DETAIL'

,attribute => 'LOGGING\_LEVEL'

,value => SYS.***DBMS\_SCHEDULER.LOGGING\_RUNS***);

SYS.***DBMS\_SCHEDULER.SET\_ATTRIBUTE\_NULL***

( name => 'MONITORING.DELETE\_DQ\_DETAIL'

,attribute => 'MAX\_FAILURES');

SYS.***DBMS\_SCHEDULER.SET\_ATTRIBUTE\_NULL***

( name => 'MONITORING.DELETE\_DQ\_DETAIL'

,attribute => 'MAX\_RUNS');

BEGIN

SYS.***DBMS\_SCHEDULER.SET\_ATTRIBUTE***

( name => 'MONITORING.DELETE\_DQ\_DETAIL'

,attribute => 'STOP\_ON\_WINDOW\_CLOSE'

,value => FALSE);

EXCEPTION

*-- could fail if program is of type EXECUTABLE...*

WHEN OTHERS THEN

NULL;

END;

SYS.***DBMS\_SCHEDULER.SET\_ATTRIBUTE***

( name => 'MONITORING.DELETE\_DQ\_DETAIL'

,attribute => 'JOB\_PRIORITY'

,value => 1);

SYS.***DBMS\_SCHEDULER.SET\_ATTRIBUTE\_NULL***

( name => 'MONITORING.DELETE\_DQ\_DETAIL'

,attribute => 'SCHEDULE\_LIMIT');

SYS.***DBMS\_SCHEDULER.SET\_ATTRIBUTE***

( name => 'MONITORING.DELETE\_DQ\_DETAIL'

,attribute => 'AUTO\_DROP'

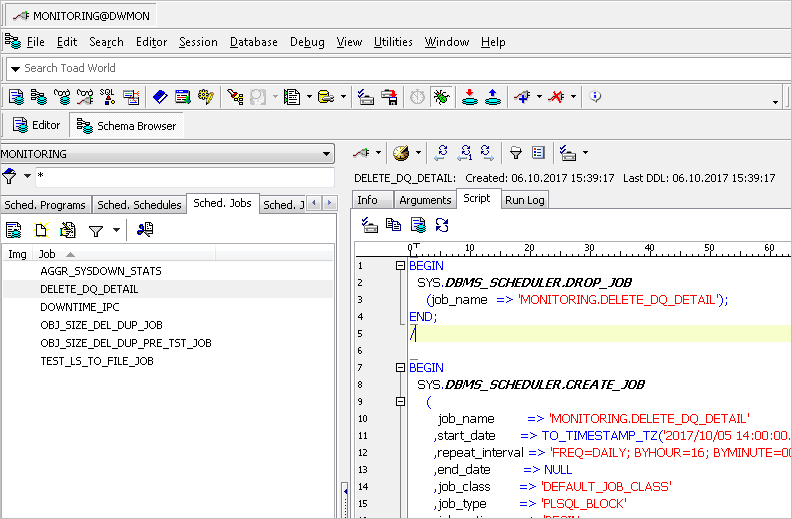
,value => FALSE);

SYS.***DBMS\_SCHEDULER.ENABLE***

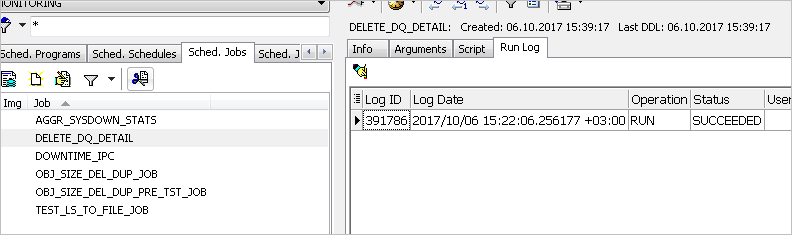
(name => 'MONITORING.DELETE\_DQ\_DETAIL');

END;

/

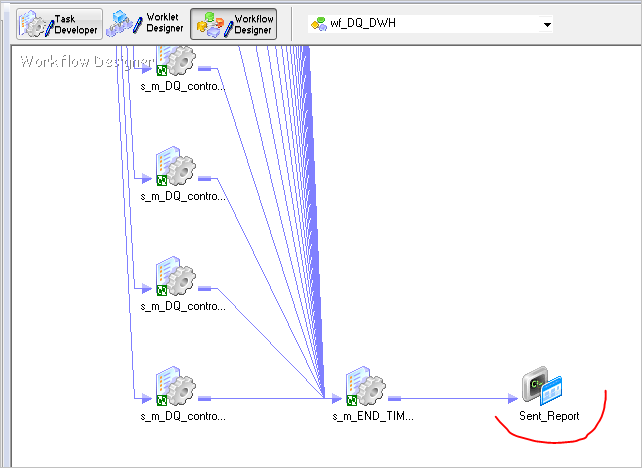


Посмотреть когда он работал можно во вкладке RunLog:

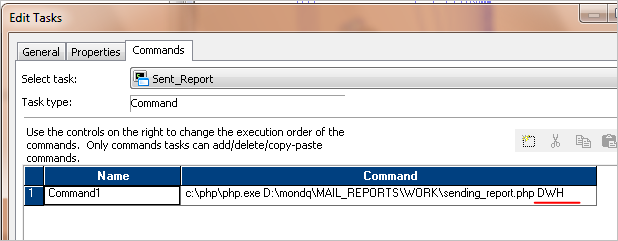


Рассылка отчетов об ошибках

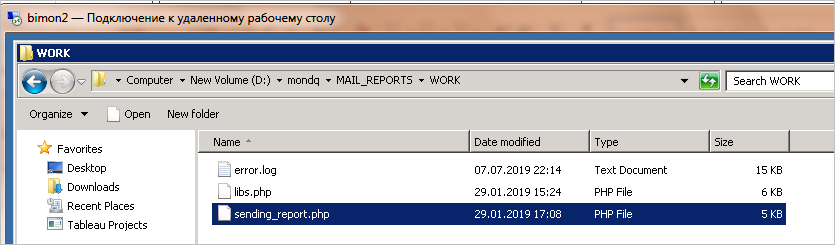
Рассылка запускается из последнего cmd **Sent\_Report** в потоке проверок:



Запускается файл sending\_report.php на сервере **bimon2**, с параметром той системы, по которой выполняются проверки (например DWH):

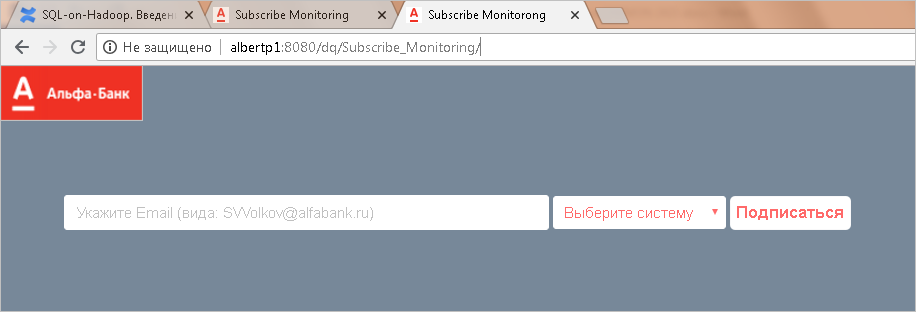


Програмные файлы лежат на сервере **bimon2** в каталоге **D:\mondq\MAIL\_REPORTS\WORK**:



Адреса получателей выбираются из таблицы MONITORING.DQ\_CONTROL\_MAIL, которая заполняется через страницу подписки на рассылку:

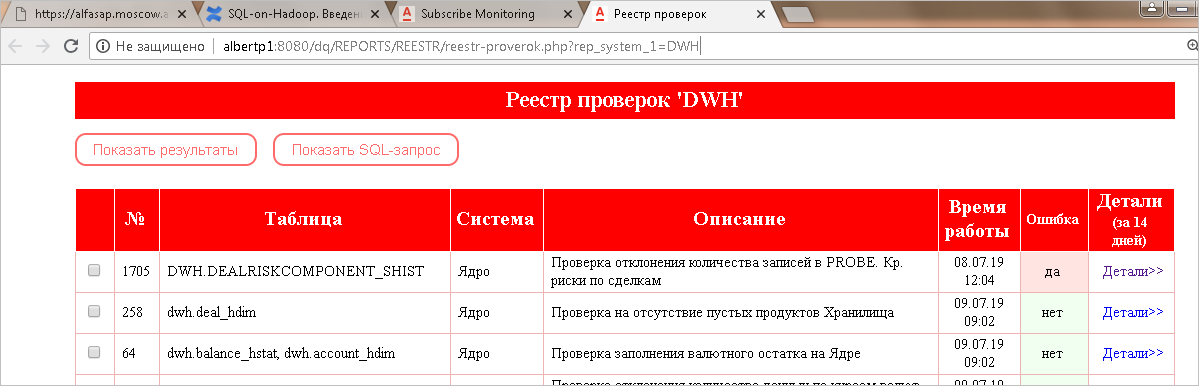
[**http://albertp1:8080/dq/Subscribe\_Monitoring/**](http://albertp1:8080/dq/Subscribe_Monitoring/)



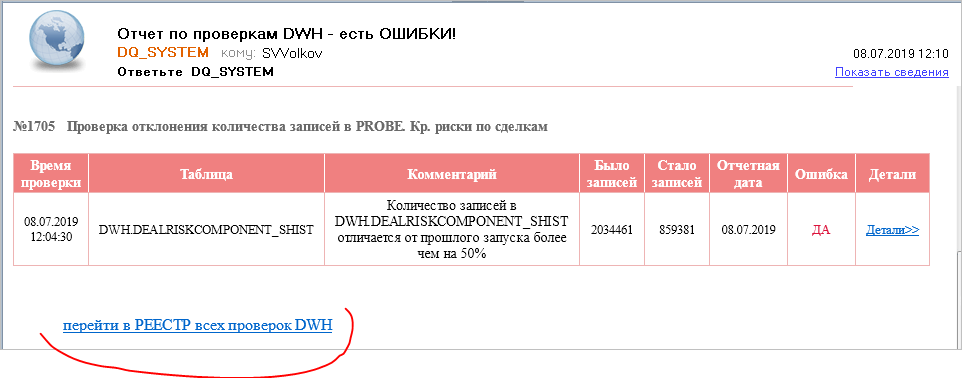
Реестр проверок

Ссылка: http://albertp1:8080/dq/REPORTS/REESTR/reestr-proverok.php?rep\_system\_1=**DWH**

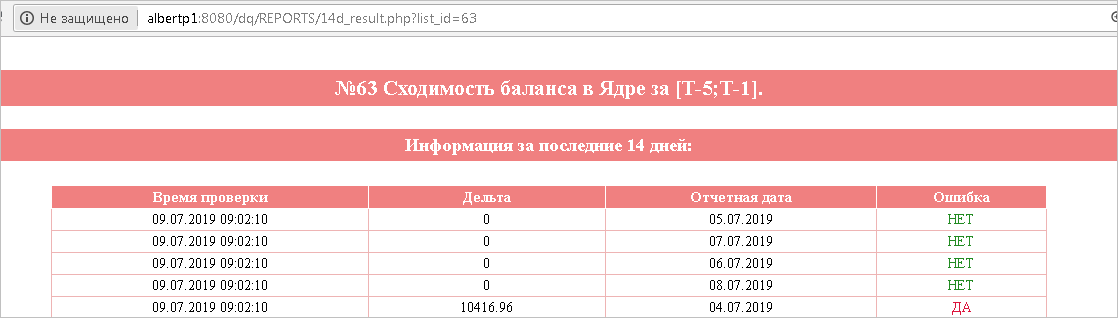
В конце ссылки указывается параметр требуемой системы (например, DWH):



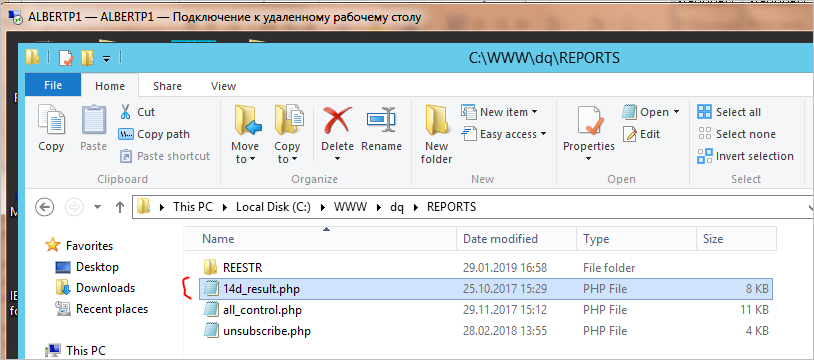
Ссылка на реестр каждой системы приходит в рассылке ошибок:



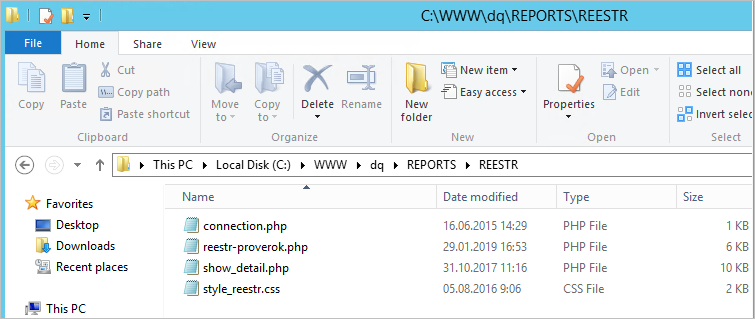
Результаты за 14 дней можно посмотреть пройдя по ссылке **Детали>>**



Программный файл на сервере ALBERTP1 - C:\WWW\dq\REPORTS\**14d\_result.php**



Работает реестр проверок на сервере ALBERTP1 в каталоге **C:\WWW\dq\REPORTS\REESTR**:



Как добавить новую проверку.

Для новой проверки надо:

- Добавить названия столбцов для рассылки в таблицу **DQ\_HEADINGS\_CONTROL**.

На примере проверки 1983, указываем название колонки рассылки в поле *COLUMN\_NAME*  и откуда будет браться значение *TABLE\_NAME* и *FIELD\_NAME*:

*insert into MONITORING.DQ\_HEADINGS\_CONTROL ( LIST\_ID, SORT\_NUM, TABLE\_NAME, FIELD\_NAME, COLUMN\_NAME, REP\_FLAG, DESCR)*

*values(1983, 1, 'DQ\_LIST\_CONTROL', 'REV\_NAME', 'Название проверки', 1, NULL);*

*insert into MONITORING.DQ\_HEADINGS\_CONTROL ( LIST\_ID, SORT\_NUM, TABLE\_NAME, FIELD\_NAME, COLUMN\_NAME, REP\_FLAG, DESCR)*

*values(1983, 2, 'TO\_CHAR (DQ\_DETAIL', 'REPORT\_TIME, ''DD.MM.YYYY HH24:MI:SS'')', 'Время проверки', 1, NULL);*

*insert into MONITORING.DQ\_HEADINGS\_CONTROL ( LIST\_ID, SORT\_NUM, TABLE\_NAME, FIELD\_NAME, COLUMN\_NAME, REP\_FLAG, DESCR)*

*values(1983, 3, 'DQ\_DETAIL', 'DESCRIPTION', 'Комментарий', 1, NULL);*

*insert into MONITORING.DQ\_HEADINGS\_CONTROL ( LIST\_ID, SORT\_NUM, TABLE\_NAME, FIELD\_NAME, COLUMN\_NAME, REP\_FLAG, DESCR)*

*values(1983, 4, 'DQ\_DETAIL', 'RESULT\_CHAR5', 'Таблица', 1, NULL);*

*insert into MONITORING.DQ\_HEADINGS\_CONTROL ( LIST\_ID, SORT\_NUM, TABLE\_NAME, FIELD\_NAME, COLUMN\_NAME, REP\_FLAG, DESCR)*

*values(1983, 5, 'DQ\_DETAIL', 'RESULT\_CHAR1', 'Конечный продукт', 1, NULL);*

*insert into MONITORING.DQ\_HEADINGS\_CONTROL ( LIST\_ID, SORT\_NUM, TABLE\_NAME, FIELD\_NAME, COLUMN\_NAME, REP\_FLAG, DESCR)*

*values(1983, 6, 'DQ\_DETAIL', 'RESULT\_NUM1', 'Количество', 1, NULL);*

*insert into MONITORING.DQ\_HEADINGS\_CONTROL ( LIST\_ID, SORT\_NUM, TABLE\_NAME, FIELD\_NAME, COLUMN\_NAME, REP\_FLAG, DESCR)*

*values(1983, 7, 'DQ\_DETAIL', 'ERROR\_FLAG', 'Ошибка', 1, NULL);*

*insert into MONITORING.DQ\_HEADINGS\_CONTROL ( LIST\_ID, SORT\_NUM, TABLE\_NAME, FIELD\_NAME, COLUMN\_NAME, REP\_FLAG, DESCR)*

*values(1983, 8, 'DQ\_DETAIL', 'RULE\_DESCRIPTION', 'Детали', 1, NULL);*

- Добавить информацию о проверке в таблицу **DQ\_ LIST \_CONTROL**.

На примере проверки 1983:

***insert into MONITORING.DQ\_LIST\_CONTROL (LIST\_ID, REV\_NAME, DEVELOPER, COMMENTS, REP\_SYSTEM, GROUP\_NAME, GROUP\_ORDER, LIST\_ORDER, SQL\_QUERY, FLAG\_NUM, FLAG\_STR)***

***values( 1983, 'Проверка наличия 3-х конечных продуктов(КП) и количества строк в каждом', 'Волков С.В.', NULL, 'DMPR', 'Проверки конечных продуктов в ВП', 3, 1, 'SQL', NULL, NULL) ;***

*REV\_NAME -* название проверки,

*GROUP\_ORDER -* место в ряду проверок всех систем

*LIST\_ORDER -* место в ряду проверок этой системы.

*GROUP\_NAME* – обязательно, название системы. По нему информация о проверке будет сортироваться для отчетов и реестра. Какое название какой системе соответствует можно посмотреть в таблице *DQ\_LIST\_CONTROL,* по тем проверкам, которые там уже есть.

*SQL\_QUERY* – запрос, пока не указываем, оставляем заглушку *'SQL'*

- Добавить селект в поле *SQL\_QUERY*  таблицы **DQ\_ LIST \_CONTROL**.

На примере проверки *1705*, пишем update и там, где в запросе есть кавычки, экранируем их еще двумя одинарными кавычками.

*update MONITORING.DQ\_LIST\_CONTROL*

*set SQL\_QUERY = ('*

*DECLARE*

*cnt int;*

*BEGIN*

*SELECT count(\*) into cnt FROM ( select as\_of\_day from DWH.DEALRISKCOMPONENT\_SHIST@DW\_ST\_PROD*

*where trunc(as\_of\_day)=trunc(sysdate)*

*and DWSCMIX in (3089,3090,3091,3092)*

*group by as\_of\_day ) t;*

*IF cnt < 1*

*THEN*

*INSERT INTO MONITORING.DQ\_DETAIL*

*(*

*DEFAULTST\_FLAG,*

*DESCRIPTION,*

*DQ\_CONTROL\_UK,*

*ERROR\_FLAG,*

*REPORT\_TIME,*

*VALUE\_DATE,*

*RESULT\_NUM1,*

*RESULT\_NUM2,*

*RESULT\_CHAR1,*

*RESULT\_CHAR5,*

*RESULT\_CHAR6,*

*RULE\_DESCRIPTION*

*)*

*SELECT*

*''N'',*

*''В таблице нет загруженных записей'',*

*1705,*

*''Y'',*

*SYSDATE AS REPORT\_TIME,*

*yesterday,*

*cnt\_yes,*

*0 cnt\_today,*

*tab,*

*''DWH.DEALRISKCOMPONENT\_SHIST'' as RESULT\_CHAR5,*

*''Ядро'' as RESULT\_CHAR6,*

*''<a target="\_blank" href="http://albertp1:8080/dq/REPORTS/14d\_result.php?list\_id=1705">Детали>></a>'' AS RULE\_DESCRIPTION*

*from*

*( select*

*RESULT\_NUM2 cnt\_yes ,*

*RESULT\_CHAR1 tab ,*

*REPORT\_TIME yesterday*

*from MONITORING.DQ\_DETAIL*

*where DQ\_CONTROL\_UK = 1705*

*and XK in ( select max(XK) from MONITORING.DQ\_DETAIL*

*where DQ\_CONTROL\_UK in (1705) ));*

*ELSE*

*INSERT INTO MONITORING.DQ\_DETAIL*

*(*

*DEFAULTST\_FLAG,*

*DESCRIPTION,*

*DQ\_CONTROL\_UK,*

*ERROR\_FLAG,*

*REPORT\_TIME,*

*VALUE\_DATE,*

*RESULT\_NUM1,*

*RESULT\_NUM2,*

*RESULT\_CHAR1,*

*RESULT\_CHAR5,*

*RESULT\_CHAR6,*

*RULE\_DESCRIPTION*

*)*

*SELECT*

*''N'',*

*case when err = 0 then ''ok'' else ''Количество записей в DWH.DEALRISKCOMPONENT\_SHIST отличается от прошлого запуска более чем на 50%'' end,*

*1705,*

*case when err = 0 then ''N'' else ''Y'' end,*

*SYSDATE AS REPORT\_TIME,*

*yesterday,*

*cnt\_yes,*

*cnt\_today,*

*tab,*

*''DWH.DEALRISKCOMPONENT\_SHIST'' as RESULT\_CHAR5,*

*''Ядро'' as RESULT\_CHAR6,*

*''<a target="\_blank" href="http://albertp1:8080/dq/REPORTS/14d\_result.php?list\_id=1705">Детали>></a>'' AS RULE\_DESCRIPTION*

*from*

*(select*

*tab,*

*LAG (dday) OVER (ORDER BY dday, tab) yesterday,*

*ind,*

*NVL (LAG (cnt) OVER (ORDER BY dday, tab), 0) cnt\_yes,*

*cnt cnt\_today,*

*CASE WHEN cnt BETWEEN NVL (LAG (cnt) OVER (ORDER BY dday, tab), 0)\*0.5*

*AND NVL (LAG (cnt) OVER (ORDER BY dday, tab), 0)\*1.5*

*THEN 0*

*ELSE 1*

*END err*

*from*

*(*

*select*

*count(\*) cnt,*

*''DWH.DEALRISKCOMPONENT\_SHIST'' tab ,*

*trunc(as\_of\_day) dday,*

*1 ind*

*from DWH.DEALRISKCOMPONENT\_SHIST@DW\_ST\_PROD*

*where trunc(as\_of\_day)=trunc(sysdate)*

*and DWSCMIX in (3089,3090,3091,3092)*

*group by as\_of\_day*

*union all*

*select*

*RESULT\_NUM2 cnt ,*

*RESULT\_CHAR1 tab ,*

*REPORT\_TIME dday,*

*0 ind*

*from MONITORING.DQ\_DETAIL*

*where DQ\_CONTROL\_UK = 1705*

*)*

*) t*

*where ind=1 ;*

*END IF;*

*COMMIT;*

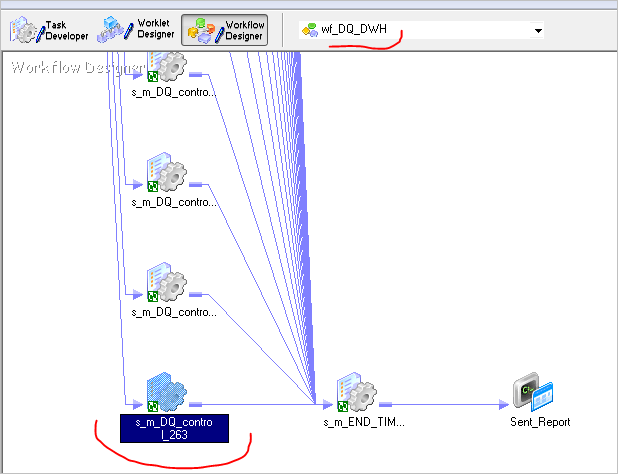
*END;*

*')*

*WHERE LIST\_ID in (1705);*

- Добавить сессию в потоке.

Чтобы проверка заработала, надо добавить сессию в потоке, который отвечает за соответствующую систему:



Для этого можно скопировать любую уже существующую сессию и изменить номер проверки в названии сессии и в вызове процедуры MONITORING.DQ\_RUN\_CONTROL:

