Восстановление работы конвейеров после пересоздания базы данных (из БД основного сервера).

Так как таблицы конвейера CONVEYORS и CONVEYOR\_LOAD не репликационные, то после пересоздания БД из базы основного сервера, таблицы конвейеров будут таблицами из БД основного сервера. Чтобы восстановить работоспособность конвейера, необходимо в программе IBExpert, на старой базе сделать экспорт таблиц CONVEYORS и CONVEYOR\_LOAD в скрипт, выбрав в настройке Export as – INSERT Statements. Сохранить скрипт (см. рис. 1).

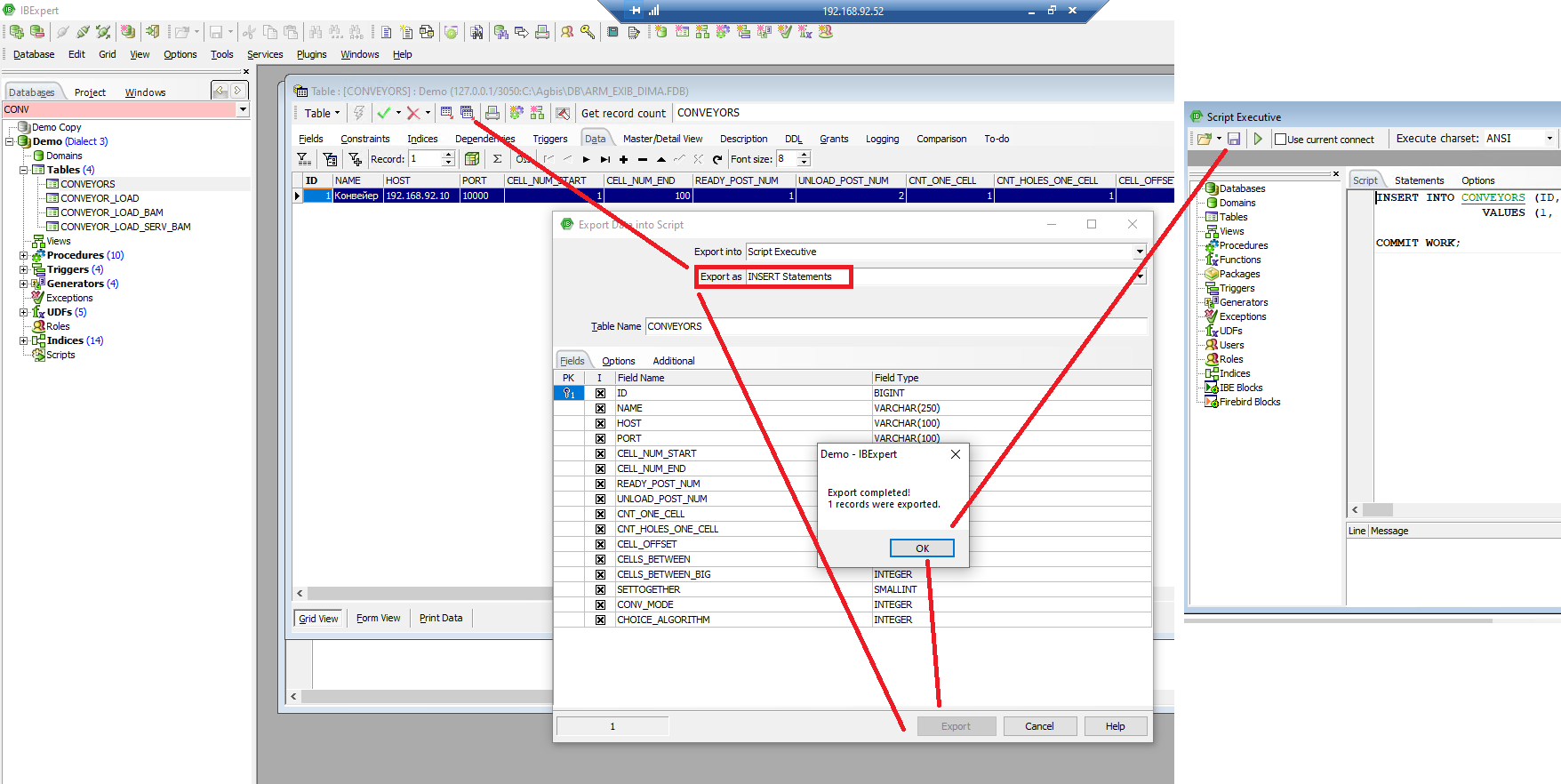


Рис. 1 В новой БД, обнулить таблицу CONVEYOR\_LOAD, если там есть записи. В IBExpert выбрать Tools -> SQL Editor (F12), выполнить команду delete from CONVEYER\_LOAD, выполнить Commit Transaction. Запустить из меню Tools -> Script Executive (Ctrl+F12), открыть скрипт для таблицы CONVEYORS (1) и выбрав новую БД (2) и Use current connection (3), запустить скрипт на выполнение. Удалить из Таблицы CONVEYORS, запись для конвейера из основной БД. Аналогично запустить скрипт для изменения таблицы CONVEYOR\_LOAD (см. рис. 2)

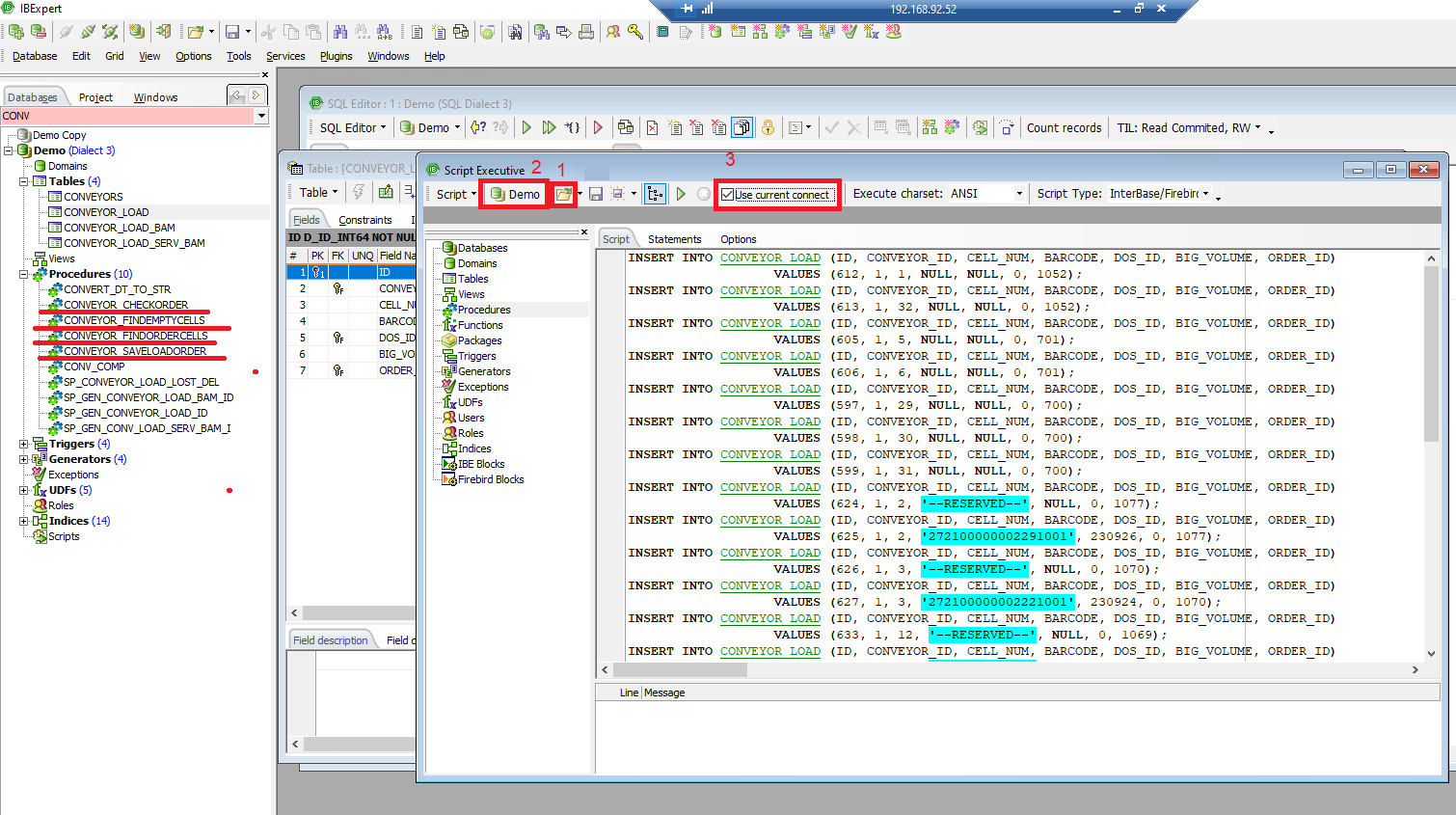


Рис. 2

Перезагрузить Агента.

Если конвейер работает с использованием сенсорного поста, по старой схеме (сенсорный пост общается с БД напрямую, в логах Агента, отсутствует лог ConvAgent) , нужно убедится, что в новой БД есть процедуры CONVEYOR\_CHECKORDER , CONVEYOR\_FINDEMPTYCELLS, CONVEYOR\_FINDEORDERCELLS и CONVEYOR\_SAVELOADORDER (см. рис. 2). Если их нет, необходимо скопировать текс процедуры:

SET TERM ^ ;

create or alter procedure CONVEYOR\_CHECKORDER (

ORDER\_BARCODE varchar(25))

returns (

RESULT integer,

ID bigint,

NUM varchar(50))

as

declare variable STATUS integer;

begin

select count(c.id), o.id, d.doc\_num, o.status\_id

from conveyor\_load c

right join docs\_order o on c.order\_id = o.id

inner join docs d on d.doc\_id= o.doc\_id

where o.barcode = :ORDER\_BARCODE

group by o.id, d.doc\_num, o.status\_id

into :RESULT, :ID, :NUM, STATUS;

if (RESULT is null)

then RESULT = -1;

else if (STATUS>4) then RESULT = -2;

suspend;

end^

SET TERM ; ^

COMMENT ON PARAMETER CONVEYOR\_CHECKORDER.RESULT IS

'-1 - заказ не существует, -2 - заказ закрыт, 0 - заказ не на конвейере, >0 - заказ на конвейере';

COMMENT ON PARAMETER CONVEYOR\_CHECKORDER.ID IS

'id заказа (null если не существует)';

COMMENT ON PARAMETER CONVEYOR\_CHECKORDER.NUM IS

'номер заказа';

COMMENT ON PARAMETER CONVEYOR\_CHECKORDER.ORDER\_BARCODE IS

'штрихкод заказа';

/\* Following GRANT statetements are generated automatically \*/

GRANT SELECT ON CONVEYOR\_LOAD TO PROCEDURE CONVEYOR\_CHECKORDER;

GRANT SELECT ON DOCS\_ORDER TO PROCEDURE CONVEYOR\_CHECKORDER;

GRANT SELECT ON DOCS TO PROCEDURE CONVEYOR\_CHECKORDER;

/\* Existing privileges on this procedure \*/

GRANT EXECUTE ON PROCEDURE CONVEYOR\_CHECKORDER TO SYSDBA;

SET TERM ^ ;

create or alter procedure CONVEYOR\_FINDEMPTYCELLS (

CONVEYOR\_ID integer not null,

NEED\_CELLS integer not null)

returns (

TARGET\_CELL integer,

BEGIN\_CELL integer,

END\_CELL integer)

as

declare variable MIN\_CELL integer; /\* номер начальной ячейки конвейера \*/

declare variable MAX\_CELL integer; /\* номер последней ячейки конвейера \*/

declare variable PREV\_CELL integer;

declare variable CUR\_CELL integer;

begin

TARGET\_CELL = -1;

select c.cell\_num\_start, c.cell\_num\_end

from conveyors c

where c.id = :CONVEYOR\_ID

into :MIN\_CELL, :MAX\_CELL;

PREV\_CELL = MIN\_CELL-1;

for select c.cell\_num

from conveyor\_load c

where (c.conveyor\_id = :CONVEYOR\_ID)and(c.dos\_id is not null or c.order\_id is not null)

order by c.cell\_num

into :CUR\_CELL

do begin

if (CUR\_CELL-PREV\_CELL > NEED\_CELLS) then

begin

TARGET\_CELL = PREV\_CELL+trunc(NEED\_CELLS/2)+1;

BEGIN\_CELL = PREV\_CELL+1;

END\_CELL = PREV\_CELL+NEED\_CELLS;

end

else

PREV\_CELL = CUR\_CELL;

end

if ((TARGET\_CELL = -1)and(MAX\_CELL+1-PREV\_CELL > NEED\_CELLS)) then

begin

TARGET\_CELL = PREV\_CELL+trunc(NEED\_CELLS/2)+1;

BEGIN\_CELL = PREV\_CELL+1;

END\_CELL = PREV\_CELL+NEED\_CELLS;

end

suspend;

end^

SET TERM ; ^

COMMENT ON PARAMETER CONVEYOR\_FINDEMPTYCELLS.TARGET\_CELL IS

'номер ячейки перехода (-1 - нет свободных ячеек)';

COMMENT ON PARAMETER CONVEYOR\_FINDEMPTYCELLS.BEGIN\_CELL IS

'начальная ячейка диапазона';

COMMENT ON PARAMETER CONVEYOR\_FINDEMPTYCELLS.END\_CELL IS

'конечная ячейка диапазона';

COMMENT ON PARAMETER CONVEYOR\_FINDEMPTYCELLS.CONVEYOR\_ID IS

'id конвейера';

COMMENT ON PARAMETER CONVEYOR\_FINDEMPTYCELLS.NEED\_CELLS IS

'запрашиваемый диапазон пустых ячеек';

/\* Following GRANT statetements are generated automatically \*/

GRANT SELECT ON CONVEYORS TO PROCEDURE CONVEYOR\_FINDEMPTYCELLS;

GRANT SELECT ON CONVEYOR\_LOAD TO PROCEDURE CONVEYOR\_FINDEMPTYCELLS;

/\* Existing privileges on this procedure \*/

GRANT EXECUTE ON PROCEDURE CONVEYOR\_FINDEMPTYCELLS TO SYSDBA;

SET TERM ^ ;

create or alter procedure CONVEYOR\_FINDORDERCELLS (

CONVEYOR\_ID integer not null,

DOR\_ID bigint not null)

returns (

TARGET\_CELL integer,

BEGIN\_CELL integer,

END\_CELL integer)

as

declare variable TMP\_CELL integer;

declare variable CUR\_CELL integer;

begin

BEGIN\_CELL = -1;

END\_CELL = -1;

for select c.cell\_num

from conveyor\_load c

where (c.conveyor\_id = :CONVEYOR\_ID)and(c.order\_id = :DOR\_ID)

order by c.cell\_num

into :CUR\_CELL

do begin

if (CUR\_CELL-1 = END\_CELL)

then begin

END\_CELL = CUR\_CELL;

end

else begin

if (BEGIN\_CELL > 0)

then begin

TARGET\_CELL = BEGIN\_CELL+trunc((END\_CELL-BEGIN\_CELL)/2);

suspend;

end

BEGIN\_CELL = CUR\_CELL;

END\_CELL = CUR\_CELL;

end

end

if (BEGIN\_CELL > 0)

then begin

TARGET\_CELL = BEGIN\_CELL+trunc((END\_CELL-BEGIN\_CELL)/2);

suspend;

end

end^

SET TERM ; ^

COMMENT ON PARAMETER CONVEYOR\_FINDORDERCELLS.TARGET\_CELL IS

'номер ячейки перехода (-1 - нет ячеек с данным заказом)';

COMMENT ON PARAMETER CONVEYOR\_FINDORDERCELLS.BEGIN\_CELL IS

'начальная ячейка диапазона';

COMMENT ON PARAMETER CONVEYOR\_FINDORDERCELLS.END\_CELL IS

'конечная ячейка диапазона';

COMMENT ON PARAMETER CONVEYOR\_FINDORDERCELLS.CONVEYOR\_ID IS

'id конвейера';

COMMENT ON PARAMETER CONVEYOR\_FINDORDERCELLS.DOR\_ID IS

'id заказа';

/\* Following GRANT statetements are generated automatically \*/

GRANT SELECT ON CONVEYOR\_LOAD TO PROCEDURE CONVEYOR\_FINDORDERCELLS;

/\* Existing privileges on this procedure \*/

GRANT EXECUTE ON PROCEDURE CONVEYOR\_FINDORDERCELLS TO SYSDBA;

SET TERM ^ ;

create or alter procedure CONVEYOR\_SAVELOADORDER (

CONV\_ID bigint,

BEGIN\_CELL integer,

END\_CELL integer,

DOR\_ID integer)

returns (

RESULT integer)

as

declare variable CELL\_NUM integer;

begin

CELL\_NUM = BEGIN\_CELL;

while (CELL\_NUM <= END\_CELL) do

begin

update or insert into conveyor\_load

(conveyor\_id, cell\_num, order\_id)

values (:CONV\_ID, :CELL\_NUM, :DOR\_ID)

matching (cell\_num);

CELL\_NUM = CELL\_NUM +1;

end

RESULT = 0;

suspend;

end^

SET TERM ; ^

COMMENT ON PARAMETER CONVEYOR\_SAVELOADORDER.CONV\_ID IS

'id конвейера';

COMMENT ON PARAMETER CONVEYOR\_SAVELOADORDER.BEGIN\_CELL IS

'начальная ячейка диапазона';

COMMENT ON PARAMETER CONVEYOR\_SAVELOADORDER.END\_CELL IS

'конечная ячейка диапазона';

COMMENT ON PARAMETER CONVEYOR\_SAVELOADORDER.DOR\_ID IS

'id заказа';

/\* Following GRANT statetements are generated automatically \*/

GRANT SELECT,INSERT,UPDATE ON CONVEYOR\_LOAD TO PROCEDURE CONVEYOR\_SAVELOADORDER;

/\* Existing privileges on this procedure \*/

GRANT EXECUTE ON PROCEDURE CONVEYOR\_SAVELOADORDER TO SYSDBA;

Запустить Script Executive, вставить в окно текст процедуры и выбрав текущую базу и Use current connect, запустить скрипт на выполнение.

Перезагрузить сенсорный пост и Агента.