



SQL Query Manipulation (DML)

- Example & Explanation -

Created by: Felice Benita



Overview

- [Background](#)
- [Scope of Problem](#)
- [Example of ERD](#)
- [Example of Stored Procedure](#)
- [Example of Query Reporting \(1\)](#)
- [Example of Query Reporting \(2\)](#)
- [Example of Query Reporting \(3\)](#)
- [Conclusion](#)
- [Closing](#)

Background

- The basis is to demonstrate full proficiency in SQL, both DDL and DML.
- Provides examples of creating complex SQL queries for reporting and database programming /stored procedure and have been optimized.

Scope of Problem

- SQL query manipulation is built using SQL Server (in version 2005).
- What is shown is just an example of some that have been successfully implemented.

ERD SMS

The ERD diagram for the SMS system shows the following tables and their attributes:

- fgtr_hdr**: Attributes are fgtr_id and fgtr_hdr.
- fgtr_dtl**: Attributes are fgtr_id, tr_seq, and fgtr_dtl.
- fgtr_unit**: Attributes are fgtr_id, tr_seq, unit_id, and fgtr_unit.
- fgbal_period**: Attributes are period_id, fgbal_id1, fgbal_id2, and fgbal_period.
- fgbal_avail**: Attributes are fgbal_id1 and fgbal_avail.
- fgbal_sum**: Attributes are fgbal_id1, fgbal_id2, and fgbal_sum.
- fgbal_unit**: Attributes are unit_id, fgbal_id1, fgbal_id2, and fgbal_unit.
- deliv_hdr**: Attributes are deliv_id and deliv_hdr.
- deliv_packing**: Attributes are deliv_id, deliv_seq, cont_id, and deliv_packing.
- deliv_picking**: Attributes are deliv_id, deliv_seq, cont_id, deliv_pick_seq, and deliv_picking.
- deliv_unit**: Attributes are deliv_id, cont_id, order_id, item_num, origin_mach, loc_id, unit_id, and deliv_unit.

The relationships between the tables are as follows:

- fgtr_hdr** is connected to **fgtr_dtl** with a one-to-many relationship.
- fgtr_dtl** is connected to **fgtr_unit** with a one-to-many relationship.
- fgtr_unit** is connected to **fgbal_period** with a one-to-many relationship.
- fgtr_unit** is connected to **fgbal_sum** with a one-to-many relationship.
- fgbal_period** is connected to **fgbal_avail** with a one-to-many relationship.
- fgbal_period** is connected to **fgbal_sum** with a one-to-many relationship.
- fgbal_avail** is connected to **fgbal_unit** with a one-to-many relationship.
- fgbal_sum** is connected to **fgbal_unit** with a one-to-many relationship.
- deliv_hdr** is connected to **deliv_packing** with a one-to-many relationship.
- deliv_packing** is connected to **deliv_picking** with a one-to-many relationship.
- deliv_packing** is connected to **deliv_unit** with a one-to-many relationship.

- 
- Back

Example of stored procedure to update or insert detail transactions dan update the quantity in balancing table

```
USE [tk_ol1]
GO
/***** Object: StoredProcedure [dbo].[APFPGTrDtlInsertUpdate] Script
Date: 12/30/2009 10:58:46 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
--
-- Author: Felice Benita
-- Create date: ##/##/####
-- Description: Update or Insert fgtr_dtl and fgbal_sum
--
ALTER PROCEDURE [dbo].[APFPGTrDtlInsertUpdate]
-- Add the parameters for the stored procedure here
@fgtr_id int,
@tr_seq smallint,
@fgbal_id1 int,
@sap_reason char(5),
@cost_ctr1 char(10),
@cost_ctr2 char(10),
@reff_num1 char(20),
@reff_num2 char(20),
@reff_num3 char(20),
@reff_num4 char(20),
@loc_id char(5),
@bin_id char(10),
@qty1 money,
@qty2 money,
@wgt1 money,
@wgt2 money,
@user_id char,
@fgbal_id2 int = 0 OUTPUT,
@dt_modified datetime = 0 OUTPUT,
@aafcRcds int = 0 OUTPUT
AS
BEGIN
-- SET NOCOUNT ON added to prevent extra result sets from
-- interfering with SELECT statements.
SET NOCOUNT ON;
DECLARE @aafcRcdsBS int, @aafcRcdsDtl int
DECLARE @BalSumRowCnt int, @RowCnt int, @ErrID int, @RetStat smallint
/* Standard Error & Row Count buffer */
DECLARE @OldQty1 money, @OldWgt1 money

SET @RetStat = 0
SET @aafcRcdsBS = 0
SET @aafcRcdsDtl = 0
SET @dt_modified = GETDATE()
BEGIN TRAN
SELECT @fgbal_id2 = fgbal_id2 FROM fgbal_sum WHERE
fgbal_id1=@fgbal_id1 AND loc_id=@loc_id AND bin_id=@bin_id
```

```
SELECT @BalSumRowCnt=@@ROWCOUNT, @ErrID=@@ERROR
IF @ErrID <> 0 BEGIN
SET @RetStat=@ErrID
END
ELSE BEGIN
IF @BalSumRowCnt = 0 BEGIN
INSERT INTO fgbal_sum
(fgbal_id1,loc_id,bin_id,qty1,wgt1,dt_modified,[user_id])
VALUES ( @fgbal_id1, @loc_id,
@bin_id,@qty1,@wgt1,@dt_modified,@user_id)
SELECT @aafcRcdsBS=@@ROWCOUNT, @ErrID=@@ERROR
IF @ErrID <> 0 BEGIN
SET @RetStat=@ErrID
END
ELSE BEGIN
SELECT @fgbal_id2 = @@IDENTITY
END
END
END
IF @RetStat = 0 BEGIN
SELECT @OldQty1 = qty1, @OldWgt1 = wgt1 FROM fgtr_dtl WHERE
tr_seq = @tr_seq AND fgtr_id = @fgtr_id
SELECT @RowCnt=@@ROWCOUNT, @ErrID=@@ERROR
IF @ErrID <> 0 BEGIN
SET @RetStat=@ErrID
END
ELSE BEGIN
IF @RowCnt = 1 BEGIN
UPDATE fgtr_dtl SET
fgbal_id1=@fgbal_id1,fgbal_id2=@fgbal_id2,

sap_reason=@sap_reason,cost_ctr1=@cost_ctr1,cost_ctr2=@cost_ctr2,

reff_num1=@reff_num1,reff_num2=@reff_num2,reff_num3=@reff_num3,

reff_num4=@reff_num4,qty1=@qty1,qty2=@qty2,wgt1=@wgt1,wgt2=@wgt2,

dt_modified=@dt_modified,

[user_id]=@user_id

WHERE tr_seq = @tr_seq AND fgtr_id
= @fgtr_id
SELECT @aafcRcdsDtl=@@ROWCOUNT, @ErrID=@@ERROR
IF @ErrID <> 0 BEGIN
SET @RetStat=@ErrID
END
ELSE BEGIN
SET @OldQty1 = 0
SET @OldWgt1 = 0
INSERT INTO fgtr_dtl
(fgtr_id,tr_seq,fgbal_id1,fgbal_id2,sap_reason,cost_ctr1,

cost_ctr2,reff_num1,reff_num2,reff_num3,reff_num4,qty1,qty2,
```

```
wgt1,wgt2,sap_doc_seq,dt_modified,[user_id])
VALUES (
@fgtr_id,@tr_seq,@fgbal_id1,@fgbal_id2,@sap_reason,@cost_ctr1,

@cost_ctr2,@reff_num1,@reff_num2,@reff_num3,@reff_num4,@qty1,@qty2,

@wgt1,@wgt2,0,@dt_modified,@user_id)
SELECT @aafcRcdsDtl=@@ROWCOUNT, @ErrID=@@ERROR
IF @ErrID <> 0 BEGIN
SET @RetStat=@ErrID
END
END
IF @RetStat = 0 AND @BalSumRowCnt = 0 BEGIN
UPDATE fgbal_sum SET qty1=@OldQty1+@qty1,

dt_modified=@dt_modified,

WHERE fgbal_id1 = @fgbal_id1
SELECT @aafcRcdsBS=@@ROWCOUNT, @ErrID=@@ERROR
IF @ErrID <> 0 BEGIN
SET @RetStat=@ErrID
END
END
END
IF (@aafcRcdsBS + @aafcRcdsDtl = 2) BEGIN
SET @aafcRcds = 1
END
IF @RetStat=0 and @aafcRcds = 1 BEGIN
COMMIT TRAN
END
ELSE BEGIN
ROLLBACK TRAN
END
SET NOCOUNT OFF
RETURN @aafcRcds
END
```



Example of query reporting to report container allocation by ship point (1)

```
"Container Allocation By Ship Point"
-- =====
-- Author:      Felice Benita
-- Create date:  ##/##/####
-- Description:  Get Container Allocation By Ship Point
-- =====

DECLARE @ship_point char(5)
SET @ship_point = '{0}'

SELECT opt.order_id, opt.deliv_id, COALESCE(opt.OrderCont20,0) 'OrderCont20',
COALESCE(opt.OrderCont40,0) 'OrderCont40', COALESCE(opt.OrderCont40H,0)
'OrderCont40H',
COALESCE(dp2.WgtPPL, 0) 'WgtPPL', opt.loading_date, c.cust_name
'Customer', cr.country_name 'Country', opt.gi_date, opt.ship_point,
da.dt_shipment_completion 'ClosingDate', da.vbi_status, oi.ts_req_ship 'ESD',
om.frgt_terms_id 'IncoTerm',
COALESCE(opt.CStuffing20, 0) 'CStuffing20',
COALESCE(opt.CStuffing40,0) 'CStuffing40', COALESCE(opt.CStuffing40H,0)
'CStuffing40H',
COALESCE(opt.CReady20, 0) 'CReady20', COALESCE(opt.CReady40,0)
'CReady40', COALESCE(opt.CReady40H,0) 'CReady40H',
COALESCE(opt.CNotReady20, 0) 'CNotReady20',
COALESCE(opt.CNotReady40,0) 'CNotReady40', COALESCE(opt.CNotReady40H,0)
'CNotReady40H'
FROM
(
SELECT DISTINCT opt.order_id, opt.deliv_id, opt.loading_date,
opt.gi_date, opt.ship_point, opt.ship_to,
COUNT(DISTINCT CASE WHEN opt.cont_type = '900010' THEN opt.cont_id
END) AS OrderCont20,
COUNT(DISTINCT CASE WHEN opt.cont_type = '900011' THEN opt.cont_id
END) AS OrderCont40,
COUNT(DISTINCT CASE WHEN opt.cont_type = '900012' THEN opt.cont_id
END) AS OrderCont40H,
COUNT(DISTINCT CASE WHEN opt.dn_date > '1900-01-01' AND opt.cont_type
= '900010' THEN opt.cont_id END) AS CStuffing20,
COUNT(DISTINCT CASE WHEN opt.dn_date > '1900-01-01' AND opt.cont_type
= '900011' THEN opt.cont_id END) AS CStuffing40,
COUNT(DISTINCT CASE WHEN opt.dn_date > '1900-01-01' AND opt.cont_type
= '900012' THEN opt.cont_id END) AS CStuffing40H,
COUNT(DISTINCT CASE WHEN opt.dt_cargo_optimized > '1900-01-01' AND
opt.dn_date = '1900-01-01' AND opt.cont_type = '900010' THEN opt.cont_id END)
AS CReady20,
COUNT(DISTINCT CASE WHEN opt.dt_cargo_optimized > '1900-01-01' AND
opt.dn_date = '1900-01-01' AND opt.cont_type = '900011' THEN opt.cont_id END)
AS CReady40,
COUNT(DISTINCT CASE WHEN opt.dt_cargo_optimized > '1900-01-01' AND
opt.dn_date = '1900-01-01' AND opt.cont_type = '900012' THEN opt.cont_id END)
AS CReady40H,

```

```

COUNT(DISTINCT CASE WHEN opt.dt_cargo_optimized = '1900-01-01' AND
opt.dn_date = '1900-01-01' AND opt.cont_type = '900010' THEN opt.cont_id END)
AS CNotReady20,
COUNT(DISTINCT CASE WHEN opt.dt_cargo_optimized = '1900-01-01' AND
opt.dn_date = '1900-01-01' AND opt.cont_type = '900011' THEN opt.cont_id END)
AS CNotReady40,
COUNT(DISTINCT CASE WHEN opt.dt_cargo_optimized = '1900-01-01' AND
opt.dn_date = '1900-01-01' AND opt.cont_type = '900012' THEN opt.cont_id END)
AS CNotReady40H
FROM
(
SELECT dp.order_id, h.deliv_id, h.loading_date, h.gi_date,
h.ship_point, h.ship_to, dp.cont_id, dp.cont_type, dp.dn_date,
da.dt_cargo_optimized FROM
(SELECT * FROM tk_ol1..deliv_hdr (nolock) WHERE ship_point =
@ship_point AND gi_date = '1900-01-01'
AND deliv_type IN (SELECT str1 FROM tk_ol1..appl_const (nolock) WHERE
group_id = 'DELI_TYP_CONT_OPTMZ')) as h
JOIN tk_ol1..deliv_packing dp (nolock) ON h.deliv_id = dp.deliv_id
LEFT JOIN tk_ol1..deliv_cont_additional da (nolock) ON da.deliv_id =
dp.deliv_id AND da.cont_id = dp.cont_id
) AS opt
GROUP BY opt.order_id, opt.deliv_id, opt.loading_date, opt.gi_date,
opt.ship_point, opt.ship_to
) AS opt
LEFT JOIN
(
SELECT deliv_id, SUM(wgtl1) 'WgtPPL' FROM tk_ol1..deliv_packing
(nolock) GROUP BY deliv_id
) AS dp2
ON dp2.deliv_id = opt.deliv_id
LEFT JOIN
tk_ol1..deliv_additional da (nolock) ON da.deliv_id = opt.deliv_id
LEFT JOIN
tk_ol1..customer c (nolock) ON c.customer_id = opt.ship_to
LEFT JOIN
tk_ol1..country cr (nolock) ON c.country_id = cr.country_id
LEFT JOIN
tk_ol1..order_mast om (nolock) ON om.order_id = opt.order_id
LEFT JOIN
tk_ol1..order_item oi (nolock) ON oi.order_id = opt.order_id
GROUP BY opt.order_id, opt.deliv_id, OrderCont20, OrderCont40,
OrderCont40H, WgtPPL,
opt.loading_date, c.cust_name, cr.country_name, opt.gi_date,
opt.ship_point, da.dt_shipment_completion, da.vbi_status,
CStuffing20, CStuffing40, CStuffing40H, CReady20, CReady40,
CReady40H,
CNotReady20, CNotReady40, CNotReady40H, oi.ts_req_ship,
om.frgt_terms_id

```

Example of query reporting to report container incoming to mill (2)

```

" Container Incoming To Mill By EMKL "
-- =====
-- Author:      Felice Benita
-- Create date:  ##/##/####
-- Description:  Get Container Incoming To Mill By EMKL
-- =====

DECLARE @millSiteId VARCHAR(3), @mDateFrom VARCHAR(20), @mDateTo VARCHAR(20),
@mflag varchar(1)
SET @mDateFrom = '{0}'
SET @mDateTo = '{1}' + ' 23:59:59'
SET @mflag = '{2}'
SET @millSiteId = 'PD' + '{3}'

SELECT ke.keterangan 'EMKL', c.FGWH, c.Cont20, c.Cont20CB, c.Cont40HC,
COALESCE((1*Cont20) + (1*Cont20CB) + (2*Cont40HC),0) 'Teus' FROM
(
    SELECT c.kdEkspedisi, c.gdtujuan 'FGWH', SUM(DISTINCT CASE
c.type_mobil WHEN 'CONTAINER 20 FEET' THEN COALESCE(JmlCont,0) ELSE 0 END) AS
Cont20,
SUM(DISTINCT CASE c.type_mobil WHEN 'CONTAINER 20 FEET (COMBO)' THEN
COALESCE(JmlCont,0) ELSE 0 END) AS Cont20CB,
SUM(DISTINCT CASE c.type_mobil WHEN 'CONTAINER 40 FEET' THEN
COALESCE(JmlCont,0) ELSE 0 END) AS Cont40HC
FROM
(
    SELECT c.kdEkspedisi, c.gdtujuan, c.type_mobil, COUNT(c.type_mobil)
'JmlCont' FROM sot..miml c (nolock)
WHERE c.factory = @millSiteId AND c.tgl1 BETWEEN @mDateFrom AND
@mDateTo AND c.flag = @mflag AND c.type_mobil IN ('CONTAINER 20
FEET','CONTAINER 20 FEET (COMBO)','CONTAINER 40 FEET') AND c.hapus = 0
GROUP BY c.kdEkspedisi, c.gdtujuan, c.type_mobil
) AS c
GROUP BY c.kdEkspedisi, c.gdtujuan
) AS c
JOIN sot..kodeekspedisi ke (nolock) ON c.kdEkspedisi = ke.kode AND
ke.factory = @millSiteId
WHERE ke.hapus = 0
GROUP BY ke.keterangan, c.FGWH, c.kdEkspedisi, c.Cont20, Cont20CB,
c.Cont40HC
ORDER BY ke.keterangan

```

```

" Container Incoming To Mill By ContainerType "
-- =====
-- Author:      Felice Benita
-- Create date:  ##/##/####
-- Description:  Get Container Incoming To Mill By ContainerType
-- =====

DECLARE @millSiteId VARCHAR(3), @mDateFrom VARCHAR(20), @mDateTo VARCHAR(20),
@mflag varchar(1)
SET @mDateFrom = '{0}'
SET @mDateTo = '{1}' + ' 23:59:59'
SET @mflag = '{2}'
SET @millSiteId = 'PD' + '{3}'

SELECT ke.keterangan 'EMKL', RTRIM(c.type_mobil) 'Container Type',
COALESCE(c.Cont20,0) 'Cont20', COALESCE(c.Cont20CB,0) 'Cont20CB',
COALESCE(c.Cont40HC,0) 'Cont40HC', COALESCE((1*Cont20) + (1*Cont20CB)
+ (2*Cont40HC),0) 'Teus'
FROM
(
    SELECT c.kdEkspedisi, c.type_mobil, SUM(DISTINCT CASE c.type_mobil
WHEN 'CONTAINER 20 FEET' THEN COALESCE(JmlCont,0) ELSE 0 END) AS Cont20,
SUM(DISTINCT CASE c.type_mobil WHEN 'CONTAINER 20 FEET (COMBO)' THEN
COALESCE(JmlCont,0) ELSE 0 END) AS Cont20CB,
SUM(DISTINCT CASE c.type_mobil WHEN 'CONTAINER 40 FEET' THEN
COALESCE(JmlCont,0) ELSE 0 END) AS Cont40HC
FROM
(
    SELECT c.kdEkspedisi, c.type_mobil, COUNT(c.type_mobil) 'JmlCont'
FROM sot..miml c (nolock)
WHERE c.factory = @millSiteId AND c.tgl1 BETWEEN @mDateFrom AND
@mDateTo AND c.flag = @mflag AND c.type_mobil IN ('CONTAINER 20
FEET','CONTAINER 20 FEET (COMBO)','CONTAINER 40 FEET')
AND c.hapus = 0
GROUP BY c.kdEkspedisi, c.type_mobil
) AS c
GROUP BY c.kdEkspedisi, c.type_mobil
) AS c
JOIN sot..kodeekspedisi ke (nolock) ON c.kdEkspedisi = ke.kode AND
ke.factory = @millSiteId
WHERE ke.hapus = 0
GROUP BY ke.keterangan, c.kdEkspedisi, c.type_mobil, c.Cont20,
Cont20CB, c.Cont40HC
ORDER BY ke.keterangan

```


Example of query reporting to report container incoming to mill (3)

```

" Container Incoming To Mill By Shift "
-- =====
-- Author:      Felice Benita
-- Create date:  ##/##/####
-- Description:  Get Container Incoming To Mill By Shift
-- =====

DECLARE @millSiteId VARCHAR(3), @mDateFrom VARCHAR(20), @mDateTo VARCHAR(20),
@mflag varchar(1)
SET @mDateFrom = '{0}'
SET @mDateTo = '{1}' + ' 23:59:59'
SET @mflag = '{2}'
SET @millSiteId = 'PD' + '{3}'

SELECT ke.keterangan 'EMKL', COALESCE(inco.C20Shift1,0) 'C20Shift1',
COALESCE(inco.C20CBShift1,0) 'C20CBShift1', COALESCE(inco.C40HCSHift1,0)
'C40HCSHift1', COALESCE((1*C20Shift1) + (1*C20CBShift1) + (2*C40HCSHift1),0)
'TeusShift1',
COALESCE(inco.C20Shift2, 0) 'C20Shift2', COALESCE(inco.C20CBShift2,0)
'C20CBShift2', COALESCE(inco.C40HCSHift2,0) 'C40HCSHift2',
COALESCE((1*C20Shift2) + (1*C20CBShift2) + (2*C40HCSHift2),0) 'TeusShift2',
COALESCE(inco.C20Shift3, 0) 'C20Shift3', COALESCE(inco.C20CBShift3,0)
'C20CBShift3', COALESCE(inco.C40HCSHift3,0) 'C40HCSHift3',
COALESCE((1*C20Shift3) + (1*C20CBShift3) + (2*C40HCSHift3),0) 'TeusShift3'
FROM
(
SELECT kdekspedisi,
COUNT(CASE WHEN CONVERT(VARCHAR,tgl1,8) >= '07:00:00' AND
CONVERT(VARCHAR,tgl1,8) < '15:00:00' AND type_mobil = 'CONTAINER 20 FEET'
THEN type_mobil END) AS C20Shift1,
COUNT(CASE WHEN CONVERT(VARCHAR,tgl1,8) >= '07:00:00' AND
CONVERT(VARCHAR,tgl1,8) < '15:00:00' AND type_mobil = 'CONTAINER 20 FEET
(COMBO)' THEN type_mobil END) AS C20CBShift1,
COUNT(CASE WHEN CONVERT(VARCHAR,tgl1,8) >= '07:00:00' AND
CONVERT(VARCHAR,tgl1,8) < '15:00:00' AND type_mobil = 'CONTAINER 40 FEET'
THEN type_mobil END) AS C40HCSHift1,
COUNT(CASE WHEN CONVERT(VARCHAR,tgl1,8) >= '15:00:00' AND
CONVERT(VARCHAR,tgl1,8) < '23:00:00' AND type_mobil = 'CONTAINER 20 FEET'
THEN type_mobil END) AS C20Shift2,
COUNT(CASE WHEN CONVERT(VARCHAR,tgl1,8) >= '15:00:00' AND
CONVERT(VARCHAR,tgl1,8) < '23:00:00' AND type_mobil = 'CONTAINER 20 FEET
(COMBO)' THEN type_mobil END) AS C20CBShift2,
COUNT(CASE WHEN CONVERT(VARCHAR,tgl1,8) >= '15:00:00' AND
CONVERT(VARCHAR,tgl1,8) < '23:00:00' AND type_mobil = 'CONTAINER 40 FEET'
THEN type_mobil END) AS C40HCSHift2,
COUNT(CASE WHEN ((CONVERT(VARCHAR,tgl1,8) >= '23:00:00' AND
CONVERT(VARCHAR,tgl1,8) <= '23:59:59')
OR (CONVERT(VARCHAR,tgl1,8) >= '00:00:00' AND CONVERT(VARCHAR,tgl1,8)
< '07:00:00')) AND type_mobil = 'CONTAINER 20 FEET' THEN type_mobil END) AS
C20Shift3,
COUNT(CASE WHEN ((CONVERT(VARCHAR,tgl1,8) >= '23:00:00' AND
CONVERT(VARCHAR,tgl1,8) <= '23:59:59')

```

Page 3 | 4

```

OR (CONVERT(VARCHAR,tgl1,8) >= '00:00:00' AND CONVERT(VARCHAR,tgl1,8)
< '07:00:00')) AND type_mobil = 'CONTAINER 20 FEET (COMBO)' THEN type_mobil
END) AS C20CBShift3,
COUNT(CASE WHEN ((CONVERT(VARCHAR,tgl1,8) >= '23:00:00' AND
CONVERT(VARCHAR,tgl1,8) <= '23:59:59')
OR (CONVERT(VARCHAR,tgl1,8) >= '00:00:00' AND CONVERT(VARCHAR,tgl1,8)
< '07:00:00')) AND type_mobil = 'CONTAINER 40 FEET' THEN type_mobil END) AS
C40HCSHift3
FROM
(
SELECT a.kdekspedisi, a.type_mobil, a.tgl1 FROM sot..miml a (nolock)
WHERE a.factory = @millSiteId AND a.tgl1 BETWEEN @mDateFrom AND
@mDateTo AND a.flag = @mflag
AND a.type_mobil IN ('CONTAINER 20 FEET','CONTAINER 20 FEET
(COMBO)','CONTAINER 40 FEET') AND a.hapus = 0
) AS miml
GROUP BY miml.kdekspedisi
) AS inco
JOIN sot..kodeekspedisi ke (nolock) ON inco.kdekspedisi = ke.kode AND
ke.factory = @millSiteId AND ke.hapus = 0
ORDER BY ke.keterangan

```

Page 4 | 4

Created by: Felice Benita



Conclusion

- SQL query manipulation has been successfully implemented.
- From the test results, it can be seen that the query that was built worked correctly and as desired.

Those are some examples of complex SQL query manipulation (DML) using SQL Server that I created and have tested well.

Thank You.

Created By: Felice Benita