

# **Migrazione mago4 VED**

Steps migrazione

# STEP 1 - Clonazione Aziende

Database  
**VEDCONTAB**

**Azienda**  
VED\_Contabilita

**Clonare su**

Database  
**VEDMaster**

**Azienda**  
VedMaster

**duplicadbvedcontab.ps1**

Database  
**VEDBondife**

**Azienda**  
**VED\_Tubi**

**Clonare su**

Database  
**VEDBondifeClone**

**Azienda**  
**vedbondifeclone**

**duplicadbvedbondife.ps1**

Database  
**FurmaNet**

**Azienda**  
**FURMANITE**

**Clonare su**

Database  
**furmanetclone**

**Azienda**  
**furmanetclone**

**duplicadbfurmanet.ps1**

Database  
**gpxnet**

**Azienda**  
**GPX\_PRIOLO**

**Clonare su**

Database  
**gpxnetclone**

**Azienda**  
**gpxnetclone**

**duplicadbgpnet.ps1**

# **STEP 2 - Operazioni preparatorie all'esecuzione degli scripts step 3**

## **1 - Eliminare dati su db di destinazione**

Db: Vedmaster

**Ianciare script SvuotaVedmaster.ps1**

**eliminare manualmente tutte la causali di  
magazzino presenti su vedmaster**

La mappa è contenuta nel file excel creato da Signorelli "commesse duplicate.xlsx".

## **2- Creazione tabella MM4\_MappaJobsCodes (niene creata dall script)**

**Accertarsi che nella root dello script esista il file  
commesse duplicate.csv**

**Ianciare script  
CreaMappaCommesseDuplicate.ps1**

# **DEPOSITI**

## 1 Eliminare tutti i depositi su vedmaster

### Depositi fase 1 che si ripetono in Db diversi

#### StoragePhase1 che si Ripetono in Database Diversi

**01** (2 database)  
furmanetclone, gpxnetclone

**01MP** (2 database)  
furmanetclone, gpxnetclone

**COLLAUDI** (2 database)  
gpxnetclone, vedbondifeclone

**SANNAZZA** (2 database)  
gpxnetclone, vedbondifeclone



non modificare nulla su gpxnetclone

Creare deposito 01FRM su furmanetclone

Creare deposito 01MPFRM su furmanetclone

Creare deposito COLLBDF su vedbondifeclone

Creare deposito SANNABDF su vedbondifeclone

**CAUSALI**

# 1 Eliminare tutte le causali magazzino su vedmaster

**InvRsn che si Ripetono in Database Diversi**

**MOV-DEP** (3 database) furmanetclone, gpxnetclone, vedbondifeclone

**ACQ** (2 database) furmanetclone, gpxnetclone

**AMEUR** (2 database) furmanetclone, gpxnetclone

**CPEUR** (2 database) furmanetclone, gpxnetclone

**I-ACQC** (2 database) furmanetclone, vedbondifeclone (**standard**)

**MID** (2 database) furmanetclone, gpxnetclone

**MOV-LIB** (2 database) furmanetclone, gpxnetclone

**MUD** (2 database) furmanetclone, gpxnetclone

**VEN-O** (2 database) gpxnetclone, vedbondifeclone

Duplicare le causali **ed assegnare depositi** e disattivare le vecchie

**non modificare nulla su gpxnetclone**

Creare causale MOV-DEPF su furmanetclone

Creare causale ACQ-FRM su furmanetclone

Creare causale AMEUR-F su furmanetclone

Creare causale CPEUR-F su furmanetclone

Creare causale MID-FRM su furmanetclone

Creare causale MOV-LIBF su furmanetclone

Creare causale Mud-FRM su furmanetclone

Creare causale VEN-O-B su VEDBONDIFECLONE

Creare causale MOV-DEPB su bondifeclone

## STEP 3 - esecuzione script per commesse

Gli script .ps1 hanno delle varibili in cui sono contenuti i parametri del motore sql e dei database.

**Impostare correttamente le variabili prima di eseguire gli script**

1	<b>01rinumerajobid.ps1 **</b>	<b>Script per aggiornare IM_JobId nella tabella MA_Jobs per database specifici</b>
2	<b>02disableoldjobscodes.ps1 **</b>	<b>Script per disabilitare jobs basati su vecchi codici (usa la tabella MM4_MappaJobsCodes)</b>
3	<b>03replaceoldcodes.ps1 da eliminare (la duplicazione ha già fatto il lavoro) *</b>	<b>Script per aggiornare i codici Job in tutte le tabelle correlate non fatto da eliminare</b>
4	<b>04aggiornadocumenti.ps1 **</b>	<b>Aggiorna i documenti con i nuovi codici</b>
5	<b>05RinumeraldAcquisti.ps1 **</b>	<b>Rinumera id documenti di acquisto e costruisce mappa tra vecchio ide e nuovo id</b>

## STEP 3 - esecuzione script per commesse

6	<b>06RinumeraldOrdiniFornitori.ps1 **</b>	Rinumera id ordini forn. e costruisce mappa tra vecchio ide e nuovo id
7	<b>07RinumeraldOfferteFornitori.ps1 **</b>	Rinumera id offerte forn. e costruisce mappa tra vecchio ide e nuovo id
8	<b>08RinumeraldOfferteClienti **</b>	Rinumera id offerte clienti. e costruisce mappa tra vecchio id e nuovo id
9	<b>09RinumeraldOrdiniClienti **</b>	Rinumera id ordini clienti. e costruisce mappa tra vecchio id e nuovo id
10	<b>10RinumeraEntryId **</b>	Rinumera id movimenti magazzino e costruisce mappa tra vecchio id e nuovo id
11	<b>11RinumeralmQuotations.ps1 *</b>	Rinumera QuotationRequestId

## STEP 3 - esecuzione script per commesse

12	<b>12RinumeralmRapportini.ps1 *</b>	Rinumera gli id dei rapportini in tutte le tabelle contenenti WorkingReportId
13	<b>13RinumeraLibretti.ps1 *</b>	Rinumera gli id dei libretti in tutte le tabelle contenenti MeasuresBookId
14	<b>14RinumeraAnalisiPreventivo.ps1 *</b>	Rinumera gli id dei preventivi in tutte le tabelle contenenti JobQuotationId
15	<b>15RinumeraSal.p1 *</b>	Rinumera gli id dei SAL in tutte le tabelle contenuti WPRId
16	<b>27rinumerasaledoc.ps1</b>	Rinumera i ddt
18	<b>18CreaMappaRiferimenti.ps1 *</b>	Crea una mappa tra le specie archivio e i tipi documento di riferimento (usata per aggiornare cross reference)
19	<b>19AggiornaCRossReference *</b> <b>19BisRinumerasubid</b>	aggiorna i cross reference in base alla mappa dei riferimenti
20	<b>20AggCrossreferenceCommesse *</b>	aggiorna i cross reference delle commesse già rinumorate al primo script

## STEP 3 - esecuzione script per commesse

21	<b>21ExportCrossReference.ps1 *</b>	crea MA_CrossReferences\$suffix duffix = nome db su ogni origine
22	<b>22Aggiornacausaliedeposuorigini.ps1 *</b>	aggiorna sulle origini le causali e i depositi

# **TRASFERIMENTO DATI**

## **Prima di esportare dati da furmanetclone (previene errori di export)**

```
UPDATE IM_JobsNotes SET Note = REPLACE(CAST(Note AS VARCHAR(MAX)), CHAR(31), '')  
WHERE CAST(Note AS VARCHAR(MAX)) LIKE '%' + CHAR(31) + '%'
```

```
UPDATE ma_jobs SET Description = REPLACE(CAST(Description AS VARCHAR(MAX)), CHAR(31), '')  
WHERE CAST(Description AS VARCHAR(MAX)) LIKE '%' + CHAR(31) + '%'
```

```
UPDATE IM_WorkingReportsDetails SET Note = REPLACE(CAST(Note AS VARCHAR(MAX)), CHAR(31), '')  
WHERE CAST(Note AS VARCHAR(MAX)) LIKE '%' + CHAR(31) + '%'
```

# STEP 4 - Importazioni / esportazioni

## Tabelle

**Sottoinsieme Magonet Commesse nella cartella MigrazioneSottoinsiemeCommesse →**

- eseguire script **Migrate-JobsData** presente nella cartella

**Sottoinsieme Perfetto cartella MigrazioneSottoinsiemePerfetto01**

- eseguire su SSMS lo script sql **Fix-Perfetto-Lengths.sql**
- eseguire **Migrate-PerfettoData.ps1**

**Sottoinsieme Perfetto cartella MigrazioneSottoinsiemePerfetto02**

- eseguire su SSMS lo script sql **Fix-IM-Lengths.sql**
- eseguire **Migrate-ItemsData.ps1**

**Posizionarsi sulla cartella MigrazioneSottoinsiemeCrossReferences**

**Eseguire script Migrate-CrossReferencesData**

# Importazione Esportazione documenti di mago

- 1 - eliminare ordini clienti, offerte clienti, offerte da fornitore, ordini a fornitore su **vedmaster**
- 2 - eliminare bolle di carico, su **vedmaster**
- 3 - eliminare ddt, resi a fornitore su **vedmaster**

# **Esportare sottointieme acquisti \***

MA\_PurchaseDoc  
MA\_PurchaseDocDetail  
MA\_PurchaseDocNotes  
MA\_PurchaseDocPymtSched  
MA\_PurchaseDocReferences  
MA\_PurchaseDocShipping  
MA\_PurchaseDocSummary  
MA\_PurchaseDocTaxSummary

**Ianciare .\MigrazioneSottoinsiemeAcquisti\Migrate-PurchaseData.ps1**

# **Esportare sottoinsieme ordini fornitori**

**posizionarsi nella cartella MigrazioneSottoinsiemeOrdiniFornitore**

**eseguire lo script sql Increase-DescriptionColumnLength.sql**

**eseguire lo script ps1 Migrate-PurchaseOrdData.ps1**

# **Esportare sottoinsieme offerte fornitori fornitori**

**PRIMA DI ESPORTARE SUL DATABASE ESEGUIRE :**  
**posizionarsi sulla cartella MigrazioneSottoinsiemeOfferteFornitore**

**copiare ed incollare su ssms Fix-SuppQuotas-Lengths.sql e lanciarlo sul db  
vedmaster**

**lanciare lo script ps1 Migrate-SuppQuotasData.ps1**

# **Esportare sottoinsieme offerte clienti**

**Posizionarsi nella cartella: MigrazioneSottoinsiemeOfferteCliente**

**eseguire lo script sql Fix-CustQuotas-Lengths su SSMS nel db vedmaster**

**lanciare lo script ps1 Migrate-CustQuotasData.ps1**

**N.B nello script di migrazione le tabelle vanno importate da gpxnet non da gpxnetclone - perchè gli ordini clienti di gpx non devono essere rinumerati. lo script è già corretto**

## **Esportare sottoIntieme ordini clienti \***

**posizionarsi su cartella MigrazioneSottoinsiemeOrdiniCliente**

**eseguire da SSMS lo script sql Alter\_Description\_VEDMaster\_MA\_SaleOrdDetails su vedmaster**

**lanciare Migrate-SaleOrdData.ps1**

# **Articoli di magazzino**

## **eliminare da vedmaster tutti gli articoli**

```
DELETE FROM vedmaster.dbo.MA_ItemsWMSZones WHERE Item IN (SELECT Item FROM vedmaster.dbo.ma_items);
DELETE FROM vedmaster.dbo.MA_ItemsLIFO WHERE Item IN (SELECT Item FROM vedmaster.dbo.ma_items);
DELETE FROM vedmaster.dbo.MA_ItemsKit WHERE Item IN (SELECT Item FROM vedmaster.dbo.ma_items);
DELETE FROM vedmaster.dbo.MA_ItemsFIFO WHERE Item IN (SELECT Item FROM vedmaster.dbo.ma_items);
DELETE FROM vedmaster.dbo.MA_ItemsStorageQtyMonthly WHERE Item IN (SELECT Item FROM vedmaster.dbo.ma_items);
DELETE FROM vedmaster.dbo.MA_ItemsLIFODomCurr WHERE Item IN (SELECT Item FROM vedmaster.dbo.ma_items);
DELETE FROM vedmaster.dbo.MA_ItemsManufacturingData WHERE Item IN (SELECT Item FROM vedmaster.dbo.ma_items);
DELETE FROM vedmaster.dbo.MA_ItemsFIFODomCurr WHERE Item IN (SELECT Item FROM vedmaster.dbo.ma_items);
DELETE FROM vedmaster.dbo.MA_ItemsIntrastat WHERE Item IN (SELECT Item FROM vedmaster.dbo.ma_items);
DELETE FROM vedmaster.dbo.MA_ItemNotes WHERE Item IN (SELECT Item FROM vedmaster.dbo.ma_items);
DELETE FROM vedmaster.dbo.MA_ItemsPurchaseBarCode WHERE Item IN (SELECT Item FROM vedmaster.dbo.ma_items);
DELETE FROM vedmaster.dbo.MA_ItemsLanguageDescri WHERE Item IN (SELECT Item FROM vedmaster.dbo.ma_items);
DELETE FROM vedmaster.dbo.MA_ItemsSubstitute WHERE Item IN (SELECT Item FROM vedmaster.dbo.ma_items);
DELETE FROM vedmaster.dbo.MA_StandardCostHistorical WHERE Item IN (SELECT Item FROM vedmaster.dbo.ma_items);
DELETE FROM vedmaster.dbo.MA_ItemsComparableUoM WHERE Item IN (SELECT Item FROM vedmaster.dbo.ma_items);
DELETE FROM vedmaster.dbo.ma_items;
```

```
delete from vedmaster.dbo.MA_ItemCustomers
delete from vedmaster.dbo.ma_items
delete from vedmaster.dbo.MA_ItemSuppliers
```

## **Posizionarsi nella cartella `MigrazioneSottoinsiemeArticoli`**

- Eseguire script sql `Fix-MA_Items-Description-280.sql` su vedmaster
- Eseguire `Migrate-ItemsData.ps1`

# **Esportare sottoinsieme magazzino**

**eseguire script da cartella MigrazioneSottoinsiemeMagazzino  
Migrate-StockData.ps1**

# **Lavori post migrazione iniziale**

23	<b>22PostTrasfUpdateCrossReference *</b>	<b>aggiorna i cross references</b>
24	<b>23PostTrasfAggiornarfatturevedmaster1 *</b>	<b>i riferimenti alle fatture già presenti su vedmaster</b>
25	Inserire nella tabella IM_SpecificationsItems i seguenti campi : k (float, null) idSal (nchar(10), null) Riga (float, null)	<b>aggiunge dei campi su im_specificationsitems</b>
26	<b>24Opreazionilds.ps1</b>	<b>aggiorna i nuovi id</b>

# Dati gpx Ram

## Operazioni pre-esportazione

Controllare i campi chiave delle tabelle gpx\_ (per qualche motivo sono mancanti)

- Su Db vedmaster - Inserire chiave su gpx\_parametri campo chiave 'Codice'
- Su Db vedmaster - Inserire chiave multipla su gpx\_parametririghe campo chiave 'Codice' e 'Deposito'

## **dalla cartella MigrazioneSottoinsiemeGpx**

- **eseguire script sql Fix-gpx\_righeram-Descrizione-280**
- **eseguire script Migrate-GpxData.ps1**

# **Query da eseguire per aggiornare i riferimenti degli ordini clienti alle RAM di gpx**

```
UPDATE d
SET d.SaleDocId = c.SaleOrdId
FROM VEDMaster.dbo.gpx_saledocram d
INNER JOIN gpxnet.dbo.gpx_saledocram a ON d.IdRam = a.IdRam
INNER JOIN gpxnet.dbo.MA_SaleOrd b ON a.SaleDocId = b.SaleOrdId
INNER JOIN VEDMaster.dbo.MA_SaleOrd c ON c.InternalOrdNo = b.InternalOrdNo
```

**Dalla cartella MigrazioneSottoinsiemeEmployees**

**eseguire script Migrate-ItemsData.ps1**

## Migrazione per le viste report da gpxnetclone a vedmaster

27

25trasferisciviews.ps1

**trasferisce tutte le viste create per i report**

**da fare --- errore script**

# **Esportare sottoinsieme multi deposito**

**Dalla cartella MigrazioneSottoinsiemeMultstorages**

**eseguire Migrate-ItemsData.ps1**

**Dalla cartella MigrazioneSottoinsiemeLotti eseguire Migrate-LotsData.ps1**

# **Trasferimento ddt cartella MigrazioneSottoinsiemeDdt**

- Eseguire script sql Fix-MA\_SaleDocDetail-Description-512. (solo se la dimensione del campo Description di MA\_SaleDocDetail è inferiore a 512)
- eseguire script ps1 Migrate-ItemsData

# Problema con RAM

**Impostare in VEDMaster.dbo.MA\_IDNumbers il numeratore degli ordini in negativo  
update VEDMaster.dbo.MA\_IDNumbers set LastId = -6335 where CodeType = 3801098**

**6335 è un valore da aggiornare al momento verificando quale è l'ultimo id ordine cliente  
su gpxnet**

dal momento in cui viene fatta questa operazione tutti gli ordini cliente nuovi avranno un id negativo.

per gli ordini presenti su GPX che non hanno generato RAM sara necessario ricaricare gli ordini. **sarebbe preferibile che prima della migrazione gli ordini abbiano tutti le ram generate**

Creare su vedmaster nella tabella gpx\_saledocram la chiave multipla SaleDocId, IdRam

# da fare

**creare una tabella di raccordo tra id di vedcontab e id dei db cloni**

**Creare tabella vecchi id nuovi id**

```
CREATE TABLE [dbo].[SaleDocMapping]([  
    [DocumentType] [nvarchar](50) NULL,  
    [NewSaledocId] [int] NULL,  
    [DocNo] [nvarchar](50) NULL,  
    [DocumentDate] [datetime] NULL,  
    [OldSaleDocId] [int] NULL,  
    [SourceDB] [nvarchar](50) NULL,  
    [Customer] [nvarchar](50) NULL  
) ON [PRIMARY]
```

# popolare tabella SaleDocMapping

-- Inserisci in una tabella esistente

```
INSERT INTO VEDMaster.dbo.SaleDocMapping
(DocumentType, NewSaledocId, DocNo, DocumentDate, OldSaleDocId, Customer, SourceDB)
SELECT
    a.DocumentType, a.SaleDocId as NewSaledocId, a.DocNo, a.DocumentDate, g.SaleDocId as OldSaleDocId, a.CustSupp as Customer, 'gpxnetclone' as SourceDB
FROM VEDMaster.dbo.MA_SaleDoc a
INNER JOIN gpxnetclone.dbo.MA_SaleDoc g
    ON g.DocNo = a.DocNo
        AND g.DocumentDate = a.DocumentDate
    AND g.CustSupp = a.CustSupp
    where g.DocumentType = 3407874

UNION ALL
```

SELECT

```
    a.DocumentType, a.SaleDocId as NewSaledocId, a.DocNo,a.DocumentDate, v.SaleDocId as OldSaleDocId, a.CustSupp as Customer,'vedbondifeclone' as SourceDB
FROM VEDMaster.dbo.MA_SaleDoc a
INNER JOIN vedbondifeclone.dbo.MA_SaleDoc v
    ON v.DocNo = a.DocNo
        AND v.DocumentDate = a.DocumentDate
    AND v.CustSupp = a.CustSupp
    where v.DocumentType = 3407874
```

UNION ALL

SELECT

```
    a.DocumentType, a.SaleDocId as NewSaledocId, a.DocNo, a.DocumentDate, f.SaleDocId as OldSaleDocId,a.CustSupp as Customer,'furmanetclone' as SourceDB
FROM VEDMaster.dbo.MA_SaleDoc a
INNER JOIN furmanetclone.dbo.MA_SaleDoc f
    ON f.DocNo = a.DocNo
        AND f.DocumentDate = a.DocumentDate
    AND f.CustSupp = a.CustSupp
    where f.DocumentType = 3407874
```

# **Sistemazione riferimenti origine Fatture (gli id su vedmaster sono diversi dalle origini)**

## **-- Backup della tabella prima dell'UPDATE**

```
SELECT *
INTO VEDMaster.dbo.MA_CrossReferences_BACKUP
FROM VEDMaster.dbo.MA_CrossReferences
WHERE OriginDocType = 27066387
```

## **-- Aggiornamento dei riferimenti origine fatture**

```
UPDATE c
SET c.OriginDocID = s.NewSaledocId
FROM VEDMaster.dbo.MA_CrossReferences c
INNER JOIN SaleDocMapping s
    ON c.OriginDocType = 27066387
        AND c.OriginDocID = s.OldSaleDocId
WHERE s.NewSaledocId != s.OldSaleDocId
    AND s.DocNo != ""
    AND s.DocumentType = 3407874
```

# **Trasferimento sedi**

**ma\_custsuppbranches**

## **Bozza di query per aggiornamento documenti di vendita esiste il problema che nei cross references ci sono id documenti di vendita uguali tra i database origini (clone)**

```
UPDATE c
SET c.DerivedDocId = s.NewSaleDocId
from VEDMaster.dbo.SaleDocMapping s
inner join VEDMaster.dbo.MA_CrossReferences c
on c.DerivedDocID = s.OldSaleDocId and c.DerivedDocType = 27066387 and s.SourceDB = 'furmanetclone'
```

```
UPDATE c
SET c.DerivedDocId = s.NewSaleDocId
from VEDMaster.dbo.SaleDocMapping s
inner join VEDMaster.dbo.MA_CrossReferences c
on c.DerivedDocID = s.OldSaleDocId and c.DerivedDocType = 27066387 and s.SourceDB = 'vedbondifeclone'
```

```
UPDATE c
SET c.DerivedDocId = s.NewSaleDocId
from VEDMaster.dbo.SaleDocMapping s
inner join VEDMaster.dbo.MA_CrossReferences c
on c.DerivedDocID = s.OldSaleDocId and c.DerivedDocType = 27066387 and s.SourceDB = 'gpxnetclone'
```

**lanciare questa update per il problema degli ordini non rinumerati da gpx**

```
update VEDMaster.dbo.MA_CrossReferences set OriginDocID = OriginDocID -100000 where  
OriginDocType = 27066372
```

```
update VEDMaster.dbo.MA_CrossReferences set DerivedDocID = DerivedDocID -100000 where  
DerivedDocType = 27066372
```

# Eliminazione riferimenti inesistenti

```
DELETE cr
FROM dbo.MA_CrossReferences cr
INNER JOIN dbo.SaleDocMapping sm
    ON cr.DerivedDocID = sm.NewSaledocId
INNER JOIN dbo.MA_SaleDoc sd_fattura
    ON cr.DerivedDocID = sd_fattura.SaleDocId
INNER JOIN dbo.MA_SaleDoc sd_origine
    ON cr.OriginDocID = sd_origine.SaleDocId
WHERE sd_fattura.DocumentType IN (3407874, 3407876)
    AND sd_origine.CustSupp <> sd_fattura.CustSupp
    AND cr.OriginDocType = 27066383
```

**elimina il riferimento a bolle insistenti**