

Porting in dbt

[Analisi MIAMI IT001.pdf](#)

Migrazione dell'attuale processo che utilizza varie stored procedures e funzioni SQL verso dbt.

Prerequisiti

Tabella `dwh.paghefatture` popolata con il contenuto del file `fatture.txt` già pulito.

Azienda	Anno	Mese	Matricola	GGLav	GGRetr	HHLav	HHRetr
000010	2015	1	0572969	19.00	22.00	146.000	168.500

▼ Schema `dwh.paghefatture`

column_name	data_type	is_nullable	column_default
Azienda	varchar	YES	NULL
Anno	varchar	YES	NULL
Mese	varchar	YES	NULL
Matricola	varchar	YES	NULL
GGLav	numeric	YES	NULL
GGRetr	numeric	YES	NULL
HHLav	numeric	YES	NULL
HHRetr	numeric	YES	NULL
Stipendio	numeric	YES	NULL
RetribTFR	numeric	YES	NULL
ImponImps	numeric	YES	NULL
Netto	numeric	YES	NULL
CostoRetr	numeric	YES	NULL
CtrRetr	numeric	YES	NULL
InailRetr	numeric	YES	NULL
Ratei	numeric	YES	NULL
CtrRatei	numeric	YES	NULL
InailRatei	numeric	YES	NULL
Tfr	numeric	YES	NULL
EbiFormRetr	numeric	YES	NULL
EbiFormRatei	numeric	YES	NULL
RimbSpese	numeric	YES	NULL
Trasferte	numeric	YES	NULL
CtrTrasferte	numeric	YES	NULL
UnaTantum	numeric	YES	NULL
CtrUnaTantum	numeric	YES	NULL
PremioDecontrib	numeric	YES	NULL
CtrDecontrib	numeric	YES	NULL
CtrAuto	numeric	YES	NULL

Arretrati	numeric	YES	NULL
CtrArretrati	numeric	YES	NULL
ArretratiAP	numeric	YES	NULL
CtrArrAP	numeric	YES	NULL
UnaTantumAP	numeric	YES	NULL
CtrTantumAP	numeric	YES	NULL
Causale90	numeric	YES	NULL
Causale91	numeric	YES	NULL
CtrCaus9091	numeric	YES	NULL
Causale92	numeric	YES	NULL
CtrCaus9296	numeric	YES	NULL
Causale93	numeric	YES	NULL
Causale94	numeric	YES	NULL
Causale96	numeric	YES	NULL
Causale99	numeric	YES	NULL
Rival_Ratei	numeric	YES	NULL
Contrib_Riv_Ratei	numeric	YES	NULL
Ind_dispon	numeric	YES	NULL

Per ottenere info sugli schemi delle tabelle:

```
SELECT
  COLUMN_NAME,
  DATA_TYPE,
  IS_NULLABLE,
  COLUMN_DEFAULT
FROM
  INFORMATION_SCHEMA.COLUMNS
WHERE
  TABLE_NAME = 'FactFatture';
```

Pipeline

[dwh.PagheFatture da BigQuery] → [modelli dbt (miami_ibcnif, miami_ibdiif, ecc.)] → [Talend legge i dati finali]

Step 1

Creare il modello `[[IAM_TABLE_FILL]]`. Questo modello dovrà essere l'equivalente della stored procedure `[[IAM]].[[IAM_TABLE_FILL]]` la quale fa:

1. Controlli preliminari:

- Verifica se il processo è già stato eseguito per quel periodo controllando la tabella `[[IAM_LOG]]`
- Se già eseguito, blocca l'esecuzione con messaggio di errore
- Controlla se esistono dati nel data warehouse per il periodo richiesto nella tabella `FactFatture`

▼ Schema `FactFatture`

COLUMN_NAME	DATA_TYPE	IS_NULLABLE	COLUMN_DEFAULT
id_fact	int	NO	NULL
id_DimDataDocumento	int	NO	NULL

id_DimDataCompetenza	int	NO	NULL
id_DimClienti	int	NO	NULL
id_DimContiContabili	int	NO	NULL
id_DimContrattiPrestazione	int	NO	NULL
id_DimDocumenti	int	NO	NULL
id_DimLineeBusiness	int	NO	NULL
id_DimStatiElaborazione	int	NO	NULL
id_DimCausali	int	NO	NULL
id_DimLavoratori	int	NO	NULL
id_DimAlbero	int	NO	NULL
id_DimContatti	int	NO	NULL
id_DimBusinessLineCDG	int	NO	NULL
id_dimProdottoCDG	int	NO	NULL
Quantita	numeric	YES	NULL
PrezzoUnitario	numeric	YES	NULL
Fatturato	numeric	YES	NULL
CostoTecnicoUnitario	numeric	YES	NULL
CostoPresuntoUnitario	numeric	YES	NULL

2. Backup dei dati:

a. Esegue un backup dei dati precedenti tramite `IAM.IAM_TABLE_BACKUP`

▼ Logica `IAM.IAM_TABLE_BACKUP`

```
USE [BINEW_HELPER]
GO

/***** Object: StoredProcedure [IAM].[IAM_TABLE_BACKUP]  Script Date: 07/07/2025 11:09:43 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [IAM].[IAM_TABLE_BACKUP]
    @Periodo INT
AS
BEGIN
    SET NOCOUNT ON;

    -- Controllo se già eseguita tramite log nella tabella IAM_LOG
    DECLARE @Count INT = (SELECT COUNT(*) FROM IAM.IAM_LOG WHERE Period = @Periodo)
    IF (@Count > 0) BEGIN
        PRINT 'Impossibile eseguire - voce nella tabella IAM_LOG già presente.'
        RAISERROR (N'Impossibile eseguire'
            ,1, -- Severity,
            1, -- State,
            7, -- First argument used for width.
            3, -- Second argument used for precision.
```

```

        @Periodo)
    RETURN
END
--ELSE BEGIN
-- INSERT INTO IAM_LOG VALUES (@Periodo, GETDATE())
--END

-----

truncate table IAM.IAM_BAK_IBCNIF
truncate table IAM.IAM_BAK_IBDIIF
truncate table IAM.IAM_BAK_IBFUIF
truncate table IAM.IAM_BAK_IBLKIF
truncate table IAM.IAM_BAK_IBOZIF

-----

INSERT INTO IAM.IAM_BAK_IBCNIF SELECT * FROM IAM.IAM_IBCNIF

-----

INSERT INTO IAM.IAM_BAK_IBDIIF SELECT * FROM IAM.IAM_IBDIIF

-----

INSERT INTO IAM.IAM_BAK_IBFUIF SELECT * FROM IAM.IAM_IBFUIF

-----

INSERT INTO IAM.IAM_BAK_IBLKIF SELECT * FROM IAM.IAM_IBLKIF

-----

INSERT INTO IAM.IAM_BAK_IBOZIF SELECT * FROM IAM.IAM_IBOZIF

-----

END

GO

```

3. Caricamento tabelle:

Esegue in sequenza diverse stored procedure per aggiornare le tabelle:

- a. `stp_Update_IBDIIF`
- b. `stp_Update_IBCNIF`
- c. `stp_Update_IBFUIF`
- d. `stp_Update_IBOZIF` (con parametro periodo)
- e. `stp_Update_IBLKIF` (con parametro periodo + giorno)

4. Logging:

- a. Registra l'esecuzione nella tabella `IAM_LOG` con periodo, timestamp e ID file

▼ Logica `IAM_TABLE_FILL`

```
USE [BINEW_HELPER]
GO

/***** Object: StoredProcedure [IAM].[IAM_TABLE_FILL]  Script Date: 07/07/2025 10:51:38 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

--- esecuzione del processo di caricamento di tutte le tabelle IAM
--EXEC [IAM].[IAM_TABLE_FILL] 201809
CREATE PROCEDURE [IAM].[IAM_TABLE_FILL] @periodo int
AS
BEGIN
    SET NOCOUNT ON;

    declare @PeriodoGiorno int = @Periodo * 100 + 1
    Declare @idfile int =(select MAX(IdFile)+1 from IAM.IAM_LOG)

    -- Controllo se già eseguita tramite log nella tabella IAM_LOG
    DECLARE @Count INT = (SELECT COUNT(*) FROM IAM.IAM_LOG WHERE Period = @Periodo)
    IF (@Count > 0) BEGIN
        PRINT 'Impossibile eseguire - voce nella tabella IAM_LOG già presente.'

        RETURN
    END

    -- Controllo se esistono i dati del periodo richiesto
    SET @Count = (select COUNT(*) from BINEW_DWH.dwh.FactFatture where id_DimDataDocumento = @PeriodoGic)
    IF (@Count = 0) BEGIN
        PRINT 'Impossibile eseguire - Dati non presenti.'

        RETURN
    END ELSE BEGIN
        --INSERT INTO IAM.IAM_LOG VALUES (@Periodo, GETDATE())
    END

    -- Backup dati precednente elaborazione
    exec IAM.IAM_TABLE_BACKUP @periodo

    -- Caricamento tabelle (non verrà eseguito il porting della seguente logica)
    -- exec IAM.stp_Update_IBDIIF
    -- exec IAM.stp_Update_IBCNIF
    -- exec IAM.stp_Update_IBFUIF
    -- exec IAM.stp_Update_IBOZIF @Periodo
    -- exec IAM.stp_Update_IBLKIF @PeriodoGiorno
```

```

INSERT INTO IAM.IAM_LOG VALUES (@Periodo, GETDATE(),@idfile)

END
;

GO

```

▼ Schema `IAM_LOG`

column_name	data_type	is_nullable	column_default
Period	int	NO	NULL
InsertDate	datetime	NO	NULL
IdFile	int	NO	NULL

Sarebbe meglio avere un backup che va oltre il mese.

- Si potrebbe optare per modelli incrementali `*_bak` partizionati per `backup_period` dove in automatico ogni run mensile salva uno snapshot dei dati e si può impostare una retention storica secondo delle policy (es: 12 mesi, 24 mesi). Inoltre le query per un periodo sono più performanti rispetto ad un modello non partizionato.
 - La velocità di lettura non ci interessa quindi i partizionamenti si possono anche non fare e usare tabelle normali
 - Non ci interessa neanche risparmiare spazio
- Oppure si potrebbe `dbt snapshot`. Un **snapshot in dbt** permette di **tracciare lo storico delle modifiche ai dati** su una tabella sorgente (ad esempio `IAM.IBDIIF`, `IAM.IBCNIF`) creando una **tabella di audit/versioning** che conserva:
 - Lo **stato del record** al momento della cattura
 - Le **date di validità del record** (`valid_from`, `valid_to`)
 - Permette di sapere **cosa è cambiato e quando**

Con dbt snapshot ottieni:

- Versioning record-level **automatico**
- Non sovrascrivi i vecchi dati, ma questi vengono **storicizzati in modo sicuro**
- Puoi ricostruire lo stato dei dati in una certa data (utile per audit o ricostruzioni).

Controindicazioni

- Le entità variano completamente ogni mese quindi il diff non è molto efficiente perchè terrebbe sempre traccia di tutte i valori

Step 2

Creazione di un modello per ogni SP lanciata da `IAM_TABLE_FILL` che ottiene esattamente gli stessi dati delle SP.

Mart Layer (`models/marts/miami/`)

- `miami_ibcnif.sql` : logica per `IAM_IBCNIF` (equivalente alla SP `IAM.stp_Update_IBCNIF`).

▼ Schema `IAM_IBCNIF`

column_name	data_type	is_nullable	column_default
Record Type	char	NO	NULL
Interface File Name	char	NO	NULL
Mutation Code	char	NO	NULL
Country Code	char	NO	NULL
Division Code	char	NO	NULL

Concern Code	char	NO	NULL
Concern Name	char	NO	NULL
Filler	char	YES	NULL
Concern Level	char	NO	NULL
Upper Level	char	YES	NULL
Report Record	nvarchar	YES	NULL

▼ Codice `stp_Update_IBCNIF`

```
USE [BINEW_HELPER]
GO

/***** Object: StoredProcedure [IAM].[stp_Update_IBCNIF]  Script Date: 07/07/2025 15:17:31 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

-- *****
-- CREATE REPORT IBCNIF RECORD START ---
-- *****

CREATE procedure [IAM].[stp_Update_IBCNIF]

as

--begin tran;

with cteA as
(
    SELECT * FROM [IAM].[udf_Dati_IBCNIF] ('DIV_RIH')
    union
    SELECT * FROM [IAM].[udf_Dati_IBCNIF] ('DIV_RIT')
    union
    SELECT * FROM [IAM].[udf_Dati_IBCNIF] ('DIV_MAG')
    union
    SELECT * FROM [IAM].[udf_Dati_IBCNIF] ('DIV_MIN')
)

, cteB AS (

SELECT
    A.*,
    A.[Division Code] + '_' + S.DESC_SPECIALTY + '_TEMP' AS [Division Code Specialty]
FROM cteA AS A
CROSS APPLY (SELECT DISTINCT DESC_SPECIALTY AS DESC_SPECIALTY FROM [IAM].[udf_SPECIALTY] ()) WH

)
```

```

merge
  into IAM.IAM_IBCNIF as s
using
  cteB as u
on
  s.[Record Type] = u.[Record Type]
  and s.[Interface File Name] = u.[Interface File Name]
  and s.[Country Code] = u.[Country Code]
  and s.[Division Code] = u.[Division Code Specialty]
  and s.[Concern Code] = u.[Concern Code]

when MATCHED then
  update set
    s.[Mutation Code] = 'U',
    s.[Concern Name] = u.[Concern Name],
    s.[Concern Level] = u.[Concern Level],
    s.[Upper Level] = u.[Upper Level]

when NOT MATCHED then
  insert ([Record Type],[Interface File Name],[Mutation Code],[Country Code],[Division Code],[Concern Code],
    values (u.[Record Type], u.[Interface File Name], 'I', u.[Country Code], u.[Division Code Specialty], u.[Concern C

---when NOT MATCHED BY SOURCE then Delete
;

update iam.IAM_IBCNIF
set   [Report Record] = [Record Type]
      + [Interface File Name]
      + [Mutation Code]
      + [Country Code]
      + [Division Code]
      + [Concern Code]
      + [Concern Name]
      + Filler
      + convert(char(1), [Concern Level])
      + isnull([Upper Level], convert(char(15), ''))

--select * from BI_DWH.dbo.IAM_IBCNIF --where [Mutation Code] <> 'U'
--select * from IAM.IAM_IBCNIF

-- rollback;

GO

```


▼ Spiegazione logica `stp_Update_IBCNIF`

La stored procedure utilizza due CTE per preparare i dati da aggiornare o inserire nella tabella `IAM.IAM_IBCNIF`. La prima, `cteA`, unisce i dati provenienti dalla funzione `IAM.udf_Dati_IBCNIF` per le divisioni specificate (`DIV_RIH`, `DIV_RIT`, `DIV_MAG`, `DIV_MIN`). La seconda, `cteB`, estende i dati di `cteA` aggiungendo informazioni su Division name Specialty utilizzando la funzione `IAM.udf_SPECIALTY`. Successivamente, la stored procedure esegue un'operazione di MERGE per aggiornare o inserire i dati nella tabella `IAM.IAM_IBCNIF`.

1. Raccolta dati da più divisioni (CTE A)

```
sql
with cteA as (
  SELECT * FROM [IAM].[udf_Dati_IBCNIF] ('DIV_RIH')-- Divisione RIH
  union
  SELECT * FROM [IAM].[udf_Dati_IBCNIF] ('DIV_RIT')-- Divisione RIT
  union
  SELECT * FROM [IAM].[udf_Dati_IBCNIF] ('DIV_MAG')-- Divisione MAG
  union
  SELECT * FROM [IAM].[udf_Dati_IBCNIF] ('DIV_MIN')-- Divisione MIN
)
```

- Chiama una funzione `udf_Dati_IBCNIF` per 4 divisioni diverse
- Unisce tutti i risultati in un singolo dataset

2. Combinazione con specialità (CTE B)

```
sql
cteB AS (
  SELECT
    A.*,
    A.[Division Code] + '_' + S.DESC_SPECIALTY + '_TEMP' AS [Division Code Specialty]
  FROM cteA AS A
  CROSS APPLY (
    SELECT DISTINCT DESC_SPECIALTY
    FROM [IAM].[udf_SPECIALTY] ()
    WHERE [DivisionCode] IN (SELECT DISTINCT [Division Code] FROM cteA)
  ) AS S
)
```

- Fa un **prodotto cartesiano** tra divisioni e specialità
- Crea un nuovo codice divisione che include la specialità: `DIV_RIH_CARDIOLOGY_TEMP`
- Questo significa che **ogni divisione viene duplicata per ogni specialità applicabile**

3. Merge dei dati

```
sql
merge into IAM.IAM_IBCNIF as s
using cteB as u
on s.[Record Type] = u.[Record Type]
and s.[Interface File Name] = u.[Interface File Name]
and s.[Country Code] = u.[Country Code]
```

```
and s.[Division Code] = u.[Division Code Specialty]-- Usa il codice con specialità
and s.[Concern Code] = u.[Concern Code]
```

Chiave di matching: Record Type + Interface File Name + Country Code + Division Code + Concern Code

WHEN MATCHED (record esiste):

- Aggiorna `Mutation Code = 'U'` (Update)
- Aggiorna Concern Name, Concern Level, Upper Level

WHEN NOT MATCHED (record nuovo):

- Inserisce nuovo record con `Mutation Code = 'I'` (Insert)

4. Creazione del campo Report Record

```
sql
update iam.IAM_IBCNIF
set [Report Record] = [Record Type]
    + [Interface File Name]
    + [Mutation Code]
    + [Country Code]
    + [Division Code]
    + [Concern Code]
    + [Concern Name]
    + Filler
    + convert(char(1), [Concern Level])
    + isnull([Upper Level], convert(char(15), ''))
```

- Concatena tutti i campi in un unico campo `Report Record`
- Sembra essere un formato fisso per l'esportazione

Caratteristiche importanti

Espansione per specialità

- **Input:** 4 divisioni con N record ciascuna
- **Output:** 4 divisioni × M specialità = 4M combinazioni
- Esempio: Se DIV_RIH ha 100 record e ci sono 5 specialità → 500 record finali

Gestione mutazioni

- `'I'` = Insert (nuovo record)
- `'U'` = Update (record modificato)
- Traccia le operazioni per sistemi downstream

▼ Codice funzione `IAM.udf_Dati_IBCNIF`

```
USE [BINEW_HELPER]
GO

/***** Object: UserDefinedFunction [IAM].[udf_Dati_IBCNIF]  Script Date: 10/07/2025 10:57:00 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
```

```

GO

CREATE function [IAM].[udf_Dati_IBCNIF] (@DivisionCode varchar(7) )

returns table
as
return

select
    [Record Type] = 'D'
    , [Interface File Name] = 'IBCNIF'
    , [Mutation Code] = 'I'
    , [Country Code] = 'IT'
    ---, [Division Code] = 'DIV_RIT' -- Division Code] = 'DIV_RIH'
    , [Division Code] = @DivisionCode
    , [Concern Code]= [CodInternazionale]
    , [Concern Name]= ConcernName
    , Filler = ' '
    , [Concern Level] = isnull(convert(char(1), GC.ConcernLevel), convert(char(1),'0'))
    , [Upper Level] = isnull(convert(char(15), GC.UpperLevel), convert(char(15),''))
    , [Report Record] = ''

from
    [BINEW_STG].[stg].[tblGruppiClienti] GC
WHERE
    (CodInternazionale is not null) and (ConcernName is not null)

GROUP BY
    CodInternazionale, ConcernName, ConcernLevel, UpperLevel

;

GO

```

▼ Codice funzione IAM.udf_SPECIALTY

```

USE [BINEW_HELPER]
GO

/***** Object: UserDefinedFunction [IAM].[udf_SPECIALTY]  Script Date: 10/07/2025 10:57:34 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

-- SELECT * FROM [IAM].[udf_SPECIALTY] () WHERE [COD_MANSIONE] = ''
-- SELECT DISTINCT DESC_SPECIALTY AS DESC_SPECIALTY FROM [IAM].[udf_SPECIALTY] () WHERE [DivisionC

CREATE function [IAM].[udf_SPECIALTY] ()

```

```
returns table
as
return
```

```
WITH CTE AS (
```

```
    SELECT 'DIV_RIT' AS DivisionCode
    UNION ALL
    SELECT 'DIV_RIH' AS DivisionCode
    UNION ALL
    SELECT 'DIV_INT' AS DivisionCode
```

```
)
```

```
SELECT [cod_mansione_randstad_italy] AS [COD_MANSIONE]
, [desc_mansione_randstad_italy] AS [DESC_MANSIONE]
, CASE WHEN D.DivisionCode = 'DIV_INT' THEN 'OP' ELSE
CASE
    WHEN [specialty] = 'digital'      THEN 'DI'
    WHEN [specialty] = 'operational'  THEN 'OP'
    WHEN [specialty] = 'professional' THEN 'PR'
    ELSE ''
END
END AS [DESC_SPECIALTY]
, CASE WHEN D.DivisionCode = 'DIV_INT' THEN 'operational' ELSE
CASE
    WHEN [specialty] = 'digital'      THEN 'digital'
    WHEN [specialty] = 'operational'  THEN 'operational'
    WHEN [specialty] = 'professional' THEN 'professional'
    ELSE ''
END
END AS [DESC_ESTESA_SPECIALTY]
, D.DivisionCode
FROM [MDS].[dbo].[Mansioni_Trascodifica_RND] as cod
left join [MDS].[dbo].[Mansioni_RASCO_ISCO_SPEC] as spec on spec.[code_6_dgt_rasco] = cod.code_6_dgt_rasco
CROSS JOIN (SELECT * FROM CTE ) AS D
```

```
UNION ALL
```

```
SELECT
'' AS [COD_MANSIONE],
'' AS [DESC_MANSIONE],
CASE WHEN D.DivisionCode = 'DIV_INT' THEN 'OP' ELSE 'OP' END AS [DESC_SPECIALTY],
CASE WHEN D.DivisionCode = 'DIV_INT' THEN 'operational' ELSE 'operational' END AS [DESC_ESTESA_SPEC],
D.DivisionCode
FROM CTE AS D
```

```
;
```

```
GO
```

- `miami_ibdiif.sql` : logica per `IAM_IBDIIF` (equivalente alla SP `IAM.stp_Update_IBDIIF`).

▼ Codice `stp_Update_IBDIIF`

```
USE [BINEW_HELPER]
GO

/***** Object: StoredProcedure [IAM].[stp_Update_IBDIIF]  Script Date: 07/07/2025 15:17:58 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

----- ***** -----
----- UPDATE TABLE IBDIIF RECORD START ---
----- ***** -----

CREATE procedure [IAM].[stp_Update_IBDIIF]
as

with cteA
as
(
    select  'D' as [Record Type]
           , 'IBDIIF' as [Interface File Name]
           , 'I' as [Mutation Code]
           , 'IT' as [Country Code]
           , 'DIV_RIH' as [Division Code]
           , 'Randstad in House' as [Division Name]
    union
    select  'D' as [Record Type]
           , 'IBDIIF' as [Interface File Name]
           , 'I' as [Mutation Code]
           , 'IT' as [Country Code]
           , 'DIV_RIT' as [Division Code]
           , 'Randstad Italia' as [Division Name]
)

,cteB AS (

SELECT
    A.*,
    A.[Division Code] + '_' + S.DESC_SPECIALTY + '_TEMP' AS [Division Code Specialty],
    A.[Division Name] + ' ' + S.DESC_ESTESA_SPECIALTY AS [Division Name Specialty]
FROM cteA AS A
```

```

CROSS APPLY (SELECT DISTINCT DESC_SPECIALTY AS DESC_SPECIALTY, DESC_ESTESA_SPECIALTY AS DES
)

-- SELECT * FROM cteB

merge
  into IAM.IAM_IBDIIF as s
using
  cteB as u
on
  s.[Record Type] = u.[Record Type]
  and s.[Interface File Name] = u.[Interface File Name]
  and s.[Country Code] = u.[Country Code]
  and s.[Division Code] = u.[Division Code Specialty]

when MATCHED then
  update set
    s.[Mutation Code] = 'U',
    s.[Division Name] = u.[Division Name Specialty]

when NOT MATCHED then
  insert ([Record Type],[Interface File Name],[Mutation Code],[Country Code],[Division Code],[Division Name])
  values (u.[Record Type], u.[Interface File Name], 'I', u.[Country Code], u.[Division Code Specialty], u.[Division N

---when NOT MATCHED BY SOURCE then Delete
;

update iam.IAM_IBDIIF
set [Report Record] =
  [Record Type]
  + [Interface File Name]
  + [Mutation Code]
  + [Country Code]
  + [Division Code]
  + [Division Name]
;

GO

```

▼ Spiegazione logica `stp_Update_IBDIIF`

Questa stored procedure utilizza due CTE. La prima, cteA, crea un set di dati iniziale con informazioni di base sulle divisioni. La seconda, cteB, estende questi dati agganciando dettagli su Division name Specialty. Successivamente, la stored procedure esegue un'operazione di MERGE per aggiornare o inserire i dati nella tabella `IAM.IAM_IBDIIF`.

1. **CTE A (cteA)** - Dati base delle divisioni

```

sql
with cteA as (
  select 'D' as [Record Type], 'IBDIIF' as [Interface File Name],

```

```

        'I' as [Mutation Code], 'IT' as [Country Code],
        'DIV_RIH' as [Division Code], 'Randstad in House' as [Division Name]
union
select 'D', 'IBDIIF', 'I', 'IT', 'DIV_RIT', 'Randstad Italia'
)

```

Crea un dataset con due divisioni italiane di Randstad con i loro codici e nomi base.

2. CTE B (cteB) - Espansione per specialità

```

sql
,cteB AS (
    SELECT A.*,
        A.[Division Code] + '_' + S.DISC_SPECIALTY + '_TEMP' AS [Division Code Specialty],
        A.[Division Name] + ' ' + S.DISC_ESTESA_SPECIALTY AS [Division Name Specialty]
    FROM cteA AS A
    CROSS APPLY (SELECT DISTINCT DISC_SPECIALTY, DISC_ESTESA_SPECIALTY
        FROM [IAM].[udf_SPECIALTY]()
        WHERE [DivisionCode] IN (SELECT DISTINCT [Division Code] FROM cteA)) AS S
)

```

Esponde ogni divisione combinandola con tutte le specialità disponibili, creando codici come "DIV_RIH_SPECIALTY_TEMP".

3. Operazione MERGE

```

sql
merge into IAM.IAM_IBDIIF as s
using cteB as u
on s.[Record Type] = u.[Record Type]
and s.[Interface File Name] = u.[Interface File Name]
and s.[Country Code] = u.[Country Code]
and s.[Division Code] = u.[Division Code Specialty]

```

- **WHEN MATCHED:** Se il record esiste già, aggiorna il **Mutation Code** a 'U' (Update) e il nome della divisione
- **WHEN NOT MATCHED:** Se il record non esiste, lo inserisce con **Mutation Code** = 'I' (Insert)

4. Aggiornamento finale

```

sql
update iam.IAM_IBDIIF
set [Report Record] = [Record Type] + [Interface File Name] + [Mutation Code] +
    [Country Code] + [Division Code] + [Division Name]

```

Concatena tutti i campi in un unico campo **Report Record** per facilitare il reporting.

▼ Schema **IAM_IBDIIF**

column_name	data_type	is_nullable	column_default
Record Type	char	NO	NULL
Interface File Name	char	NO	NULL
Mutation Code	char	NO	NULL
Country Code	char	NO	NULL
Division Code	char	NO	NULL

Division Name	char	NO	NULL
Report Record	nvarchar	YES	NULL

- `miami_ibfuif.sql` : logica per `IAM_IBFUIF` (equivalente alla SP `IAM.stp_Update_IBFUIF`).

▼ Schema `IAM_IBFUIF`

column_name	data_type	is_nullable	column_default
Record Type	char	NO	NULL
Interface File Name	char	NO	NULL
Mutation Code	char	NO	NULL
Country Code	char	NO	NULL
Division Code	char	NO	NULL
Function Code	char	NO	NULL
Function Description	char	NO	NULL
Function Group	char	YES	NULL
Function Group Description	char	YES	NULL
Filler	char	YES	NULL
ILO Code	char	YES	NULL
Report Record	nvarchar	YES	NULL

▼ Codice `stp_Update_IBFUIF`

```
USE [BINEW_HELPER]
GO

/***** Object: StoredProcedure [IAM].[stp_Update_IBFUIF]  Script Date: 07/07/2025 15:18:17 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE procedure [IAM].[stp_Update_IBFUIF]

as

--begin tran;

with cteA
as

(
    select * from [IAM].[udf_Dati_IBFUIF] ('DIV_RIT')
    union
    select * from [IAM].[udf_Dati_IBFUIF] ('DIV_RIH')
)

merge
    into IAM.IAM_IBFUIF as s
    using
```



```

cteA as u
on
  s.[Record Type] = u.[Record Type]
  and s.[Interface File Name] = u.[Interface File Name]
  and s.[Country Code] = u.[Country Code]
  and s.[Division Code] = u.[Division Code]
  and s.[Function Code]= u.[Function Code]

when MATCHED then
  update set
    s.[Mutation Code] ='U',
    s.[ILO Code]=u.[ILO Code],
    s.[Function Group] = u.[Function Group],
    s.[Function Group Description] = u.[Function Group Description]

when NOT MATCHED then insert
  ( [Record Type]
    ,[Interface File Name]
    ,[Mutation Code]
    ,[Country Code]
    ,[Division Code]
    ,[Function Code]
    ,[Function Description]
    ,[Filler]
    ,[ILO Code]
    ,[Function Group]
    ,[Function Group Description]
  )

VALUES
  (u.[Record Type]
    ,u.[Interface File Name]
    ,''
    ,u.[Country Code]
    ,u.[Division Code]
    ,u.[Function Code]
    ,u.[Function Description]
    ,''
    ,u.[ILO Code]
    ,u.[Function Group]
    ,u.[Function Group Description]
  )

---when NOT MATCHED BY SOURCE then Delete
;

UPDATE IAM_IBFUIF
SET [Report Record] =
  [Record Type]
  + [Interface File Name]
  + [Mutation Code]

```

```

+ [Country Code]
+ [Division Code]
+ CAST( ISNULL([Function Code], '') AS CHAR(15))
+ CAST( ISNULL([Function Description] , '') AS CHAR(50))
+ CAST( ISNULL([Function Group] , '') AS CHAR(15))
+ CAST( ISNULL([Function Group Description] , '') AS CHAR(50))
+ [Filler]
+ CAST( ISNULL([ILO Code] , '') AS CHAR(4))--modificato G Timpa
FROM IAM_IBFUIF

;

--select top 10 * from IAM.IAM_IBFUIF where [Function Code] in ('90733','ACQ002') and [Division Code] in ('DIV_R
--select top 10 * from BI_DWH.dbo.IAM_IBFUIF where [Function Code] in ('90733','ACQ002') and [Division Code] in

--rollback;

;

GO

```

▼ Spiegazione logica `stp_Update_IBFUIF`

Aggiorna (merge) la tabella `IAM.IAM_IBFUIF` (oggetto finale). La sorgente dei dati è la union di quanto esposto dalla funzione `IAM.udf_Dati_IBFUIF`, che prende in ingresso il Periodo, e restituisce una tabella che segue la struttura della tabella di riferimento, processata per le seguenti divisioni:

1. CTE A (cteA) - Raccolta dati dalle funzioni

```

sql
with cteA as (
  select * from [IAM].[udf_Dati_IBFUIF] ('DIV_RIT')
  union
  select * from [IAM].[udf_Dati_IBFUIF] ('DIV_RIH')
)

```

Combina i dati delle funzioni per entrambe le divisioni Randstad:

- **DIV_RIT**: Randstad Italia
- **DIV_RIH**: Randstad in House

La funzione `udf_Dati_IBFUIF` restituisce probabilmente informazioni strutturate sulle funzioni aziendali per ciascuna divisione.

2. Operazione MERGE

```

sql
merge into IAM.IAM_IBFUIF as s
using cteA as u
on s.[Record Type] = u.[Record Type]
and s.[Interface File Name] = u.[Interface File Name]
and s.[Country Code] = u.[Country Code]

```

```
and s.[Division Code] = u.[Division Code]
and s.[Function Code] = u.[Function Code]
```

Condizioni di match: I record vengono confrontati su 5 campi chiave che identificano univocamente una funzione.

WHEN MATCHED (record esistente)

```
sql
update set
  s.[Mutation Code] = 'U',
  s.[ILO Code] = u.[ILO Code],
  s.[Function Group] = u.[Function Group],
  s.[Function Group Description] = u.[Function Group Description]
```

Aggiorna i dati della funzione esistente con codice mutazione 'U' (Update).

WHEN NOT MATCHED (nuovo record)

```
sql
insert ([Record Type], [Interface File Name], [Mutation Code], ...)
values (u.[Record Type], u.[Interface File Name], 'I', ...)
```

Inserisce nuove funzioni con codice mutazione 'I' (Insert) e un campo **Filler** con spazio vuoto.

3. Aggiornamento Report Record

```
sql
UPDATE IAM_IBFUIF
SET [Report Record] =
  [Record Type] + [Interface File Name] + [Mutation Code] +
  [Country Code] + [Division Code] +
  CAST(ISNULL([Function Code],'') AS CHAR(15)) +
  CAST(ISNULL([Function Description],'') AS CHAR(50)) +
  CAST(ISNULL([Function Group],'') AS CHAR(15)) +
  CAST(ISNULL([Function Group Description],'') AS CHAR(50)) +
  [Filler] + CAST(ISNULL([ILO Code],'') AS CHAR(4))
```

Concatena tutti i campi in un formato fisso per il reporting, con:

- Gestione dei valori NULL tramite **ISNULL**
- Casting a lunghezze fisse (**CHAR(n)**)
- Nota del modificatore "G Timpa" per il campo ILO Code

▼ Codice funzione **IAM.udf_Dati_IBFUIF**

```
USE [BINEW_HELPER]
GO

/***** Object: UserDefinedFunction [IAM].[udf_Dati_IBFUIF]  Script Date: 10/07/2025 10:59:18 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO
```

```

-- *****
-- CREATE REPORT IBFUIF RECORD START ---
-- *****

CREATE function [IAM].[udf_Dati_IBFUIF] (@DivisionCode varchar(7))

returns table
as
return

SELECT DISTINCT
    [Record Type] = 'D'
    ,[Interface File Name] = 'IBFUIF'
    ,[Mutation Code] = 'I'
    ,[Country Code] = 'IT'
    --,[Division Code] = 'DIV_RIT'
    ,[Division Code] = CONVERT(char(15), CONVERT(nvarchar(15), @DivisionCode) + '_' + S.DESC_SPECIALTY +
    ,[Function Code] = MAN.C_MANIST_COD
    ,[Function Description] = isnull(LEFT(MAN.C_MANIST_DES,50), convert(char(50),''))
    ,[Function Group] = isnull(MAN.C_RAGGRMANS_COD , convert(char(15),''))
    ,[Function Group Description] = convert(char(50),'')
    ,[Filler] = CAST(' ' AS CHAR(1))
    ,[ILO Code] = CAST(Replace(C_ISTAT_COD,','),'')AS CHAR(4)) --CAST(' ' AS CHAR(4)) modifica G.Timpanelli 20
    ,[Report Record] = CAST(' ' AS VARCHAR(512))
FROM BINEW_STG.stg.tblMansioni MAN
CROSS APPLY (SELECT * FROM [IAM].[udf_SPECIALTY] ()) AS S
WHERE MAN.C_MANIST_COD = S.COD_MANSIONE AND S.DivisionCode = @DivisionCode
;

GO

```

- `miami_iblkif.sql` : logica per `IAM_IBLKIF` (equivalente alla SP `IAM.stp_Update_IBLKIF`).

▼ Schema `IAM_IBLKIF`

column_name	data_type	is_nullable	column_default
Record Type	varchar	NO	NULL
Interface File Name	char	NO	NULL
Mutation Code	char	NO	NULL
Country Code	char	NO	NULL
Division Code	char	NO	NULL
Client Number 1	char	YES	NULL
Client Number 2	char	YES	NULL
Client Number 3	char	YES	NULL
Client Name	char	YES	NULL
Street Name	char	NO	NULL
House Number	char	YES	NULL
Postal Code	char	YES	NULL
Municipality	char	NO	NULL

Concern Code	char	NO	NULL
Filler	char	NO	NULL
Client Size	char	YES	NULL
Filler2	char	NO	NULL
Indication International Account	char	NO	NULL
Search Name	char	YES	NULL
NACE Code	char	YES	NULL
Filler3	char	NO	NULL
Client Status	char	NO	NULL
Filler4	char	NO	NULL
Date First Invoice	varchar	NO	NULL
Report Record	nvarchar	YES	NULL

▼ Codice `stp_Update_IBLKIF`

```

USE [BINEW_HELPER]
GO

/***** Object: StoredProcedure [IAM].[stp_Update_IBLKIF]  Script Date: 07/07/2025 15:18:42 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE procedure [IAM].[stp_Update_IBLKIF] @Periodo int

as

--begin tran;

TRUNCATE TABLE [BINEW_HELPER].[IAM].[CostiCP_TMP]
INSERT INTO [BINEW_HELPER].[IAM].[CostiCP_TMP]
SELECT distinct
    vwcosti.[id_DimClienti]
    , vwcosti.[id_DimDataDocumento]
    , vwcosti.[id_DimAlbero]
    , [id_DimContrattiPrestazione] = NULL
    , vwcosti.id_DimProdottoCDG
    , vwcosti.id_DimSocieta

FROM [BINEW_DWH].[olap].[vwFactCostoTotaleperCP] vwcosti
WHERE vwcosti.id_DimDataDocumento = @Periodo

TRUNCATE TABLE [BINEW_HELPER].[IAM].[RicaviCP_TMP]

```

```

INSERT INTO [BINEW_HELPER].[IAM].[RicaviCP_TMP]
SELECT
    vwfatt.[id_DimClienti]
    , vwfatt.[id_DimDataDocumento]
    , vwfatt.[id_DimAlbero]
    , vwfatt.[id_DimContrattiPrestazione]
    , vwfatt.[id_DimProdottoCDG]
    , vwfatt.[id_DimSocieta]
FROM BINEW_DWH.olap.vwFactFattureCDG vwfatt
WHERE vwfatt.id_DimDataDocumento = @Periodo;

```

```

with cteA
as

```

```

(
    select * from [IAM].[udf_Dati_IBLKIF] (@Periodo,'DIV_RIT')
    union
    select * from [IAM].[udf_Dati_IBLKIF] (@Periodo, 'DIV_RIH')
    union
    select * from [IAM].[udf_Dati_IBLKIF] (@Periodo, 'DIV_MAG')
    union
    select * from [IAM].[udf_Dati_IBLKIF] (@Periodo, 'DIV_MIN')
),

```

--Modifica temporanea per gestire anomalia cod. Cliente C280261 N.L. 2018.09.18

```
cte AS
```

```

(
    SELECT * FROM cteA WHERE ([Client Number 1] = 'C280261' AND [Concern Code] = '028') OR [Client Number 1] = 'C280261'
)

```

```
merge
```

```
    into IAM.IAM_IBLKIF as s
```

```
using
```

```
    -- Modifica temporanea per gestire anomalia cod. Cliente C280261 N.L. 2018.09.18
```

```
    cte AS u
```

```
    --cteA as u
```

```
on
```

```

    s.[Record Type] = u.[Record Type]
    and s.[Interface File Name] = u.[Interface File Name]
    and s.[Country Code] = u.[Country Code]
    and s.[Division Code] = u.[Division Code]
    and s.[Client Number 1]= u.[Client Number 1]
    and s.[Client Number 2]= u.[Client Number 2]
    and s.[Client Number 3]= u.[Client Number 3]

```

```
when MATCHED then
```

```
    update set
```

```
        s.[Mutation Code] ='U',
```

```
        s.[Client Name] = u.[Client Name],
```

```

s.[Street Name] = u.[Street Name],
s.[House Number] = u.[House Number],
s.[Postal Code] = u.[Postal Code],
s.[Municipality] = u.[Municipality],
s.[Concern Code] = u.[Concern Code],
s.[Client Size] = u.[Client Size],
s.[Indication International Account] = u.[Indication International Account],
s.[Search Name] = u.[Search Name],
s.[NACE Code] = u.[NACE Code],
s.[Client Status] = u.[Client Status],
s.[Date First Invoice] = u.[Date First Invoice]

```

when NOT MATCHED then insert

```

([Record Type]
,[Interface File Name]
,[Mutation Code]
,[Country Code]
,[Division Code]
,[Client Number 1]
,[Client Number 2]
,[Client Number 3]
,[Client Name]
,[Street Name]
,[House Number]
,[Postal Code]
,[Municipality]
,[Concern Code]
,[Filler]
,[Client Size]
,[Filler2]
,[Indication International Account]
,[Search Name]
,[NACE Code]
,[Filler3]
,[Client Status]
,[Filler4]
,[Date First Invoice]
,[Report Record])

```

VALUES

```

(u.[Record Type]
,u.[Interface File Name]
,'I'
,u.[Country Code]
,u.[Division Code]
,u.[Client Number 1]
,u.[Client Number 2]
,u.[Client Number 3]
,u.[Client Name]
,u.[Street Name]
,u.[House Number]
,u.[Postal Code]
,u.[Municipality]

```

```

,u.[Concern Code]
,u.[Filler]
,u.[Client Size]
,u.[Filler2]
,u.[Indication International Account]
,u.[Search Name]
,u.[NACE Code]
,u.[Filler3]
,u.[Client Status]
,u.[Filler4]
,u.[Date First Invoice]
,u.[Report Record])

---when NOT MATCHED BY SOURCE then Delete
;

update iam.IAM_IBLKIF
set    [Report Record] = [Record Type]
                        + [Interface File Name]
                        + [Mutation Code]
                        + [Country Code]
                        + [Division Code]
                        + [Client Number 1]
                        + [Client Number 2]
                        + [Client Number 3]
                        + [Client Name]
                        + convert(char(50),[Street Name])
                        + [House Number]
                        + [Postal Code]
                        + [Municipality]
                        + isnull([Concern Code] , convert(char(15),''))
                        + Filler
                        + isnull(convert(char(1), [Client Size]), convert(char(1),''))
                        + Filler2
                        + isnull(convert(char(1), [Indication International Account]), convert(char(1),''))
                        + isnull([Search Name], convert(char(50),''))
                        + isnull([NACE Code], convert(char(6),''))
                        + Filler3
                        + convert(char(1), [Client Status])
                        + Filler4
                        + isnull([Date First Invoice], convert(char(8),''))

;

GO

```

▼ Spiegazione logica `stp_Update_IBLKIF`

Questa stored procedure tronca e ricarica i dati nella tabella `BINew_Helper.IAM.CostiCP_TMP` (oggetto finale), leggendo da `BINew_DWH.olap.vwFactCostoTotaleperCP` (sorgente) filtrando per periodo. Inoltre, tronca e ricarica i dati nella tabella `BINew_Helper.IAM.RicaviCP_TMP` (oggetto finale), leggendo da `BINew_DWH.olap.vwFactFattureCDG` (sorgente) filtrando

per periodo. Infine, aggiorna (merge) la tabella `IAM.IAM_IBLKIF` (oggetto finale). La sorgente dei dati è la union dei dati esposti dalla funzione `IAM.udf_Dati_IBLKIF`, che prende in ingresso il Division Code e Periodo, e restituisce una tabella che segue la struttura della tabella di riferimento, processata per le seguenti divisioni:

Parametri

- `@Periodo int`: Identifica il periodo temporale per cui processare i dati

Struttura generale

1. Preparazione tabelle temporanee

```
sql
TRUNCATE TABLE [BINEW_HELPER].[IAM].[CostiCP_TMP]
INSERT INTO [BINEW_HELPER].[IAM].[CostiCP_TMP]
SELECT distinct vwcosti.[id_DimClienti], vwcosti.[id_DimDataDocumento], ...
FROM [BINEW_DWH].[olap].[vwFactCostoTotaleperCP] vwcosti
WHERE vwcosti.id_DimDataDocumento = @Periodo
```

```
sql
TRUNCATE TABLE [BINEW_HELPER].[IAM].[RicaviCP_TMP]
INSERT INTO [BINEW_HELPER].[IAM].[RicaviCP_TMP]
SELECT vwfatt.[id_DimClienti], vwfatt.[id_DimDataDocumento], ...
FROM BINEW_DWH.olap.vwFactFattureCDG vwfatt
WHERE vwfatt.id_DimDataDocumento = @Periodo
```

Popola due tabelle temporanee con dati di **costi** e **ricavi** per il periodo specificato, probabilmente utilizzate dalle funzioni successive.

2. CTE A - Raccolta dati dalle divisioni

```
sql
with cteA as (
  select * from [IAM].[udf_Dati_IBLKIF] (@Periodo,'DIV_RIT')
  union
  select * from [IAM].[udf_Dati_IBLKIF] (@Periodo, 'DIV_RIH')
  union
  select * from [IAM].[udf_Dati_IBLKIF] (@Periodo, 'DIV_MAG')
  union
  select * from [IAM].[udf_Dati_IBLKIF] (@Periodo, 'DIV_MIN')
)
```

Combina i dati dei clienti da quattro divisioni:

- **DIV_RIT**: Randstad Italia
- **DIV_RIH**: Randstad in House
- **DIV_MAG**: Probabilmente Magazzino/Logistica
- **DIV_MIN**: Probabilmente Minerali/Settore specifico

3. CTE per gestione anomalia

```
sql
cte AS (
  SELECT * FROM cteA
```

```
WHERE ([Client Number 1] = 'C280261' AND [Concern Code] = '028')
OR [Client Number 1] <> 'C280261'
)
```

Gestione speciale: Filtra i dati per gestire un'anomalia specifica del cliente C280261, mantenendo solo i record con Concern Code '028' per questo cliente, mentre per tutti gli altri clienti non applica filtri.

4. Operazione MERGE

```
sql
merge into IAM.IAM_IBLKIF as s
using cte AS u
on s.[Record Type] = u.[Record Type]
and s.[Interface File Name] = u.[Interface File Name]
and s.[Country Code] = u.[Country Code]
and s.[Division Code] = u.[Division Code]
and s.[Client Number 1] = u.[Client Number 1]
and s.[Client Number 2] = u.[Client Number 2]
and s.[Client Number 3] = u.[Client Number 3]
```

Condizioni di match: 7 campi chiave che identificano univocamente un cliente.

WHEN MATCHED (cliente esistente)

Aggiorna tutti i dati del cliente:

- `Mutation Code = 'U'`
- Nome cliente, indirizzo, codici, status, data prima fattura

WHEN NOT MATCHED (nuovo cliente)

Inserisce nuovo cliente con:

- `Mutation Code = 'I'`
- Tutti i campi inclusi i vari `Filler` per mantenere la struttura fissa

5. Aggiornamento Report Record

```
sql
update iam.IAM_IBLKIF
set [Report Record] = [Record Type] + [Interface File Name] + [Mutation Code] +
[Country Code] + [Division Code] + [Client Number 1] +
[Client Number 2] + [Client Number 3] + [Client Name] +
convert(char(50),[Street Name]) + [House Number] +
[Postal Code] + [Municipality] +
isnull([Concern Code], convert(char(15),'')) +
Filler + ... +
isnull([Date First Invoice], convert(char(8),''))
```

Concatena tutti i campi in un record a lunghezza fissa per il reporting, con:

- Conversioni esplicite a `CHAR(n)`
- Gestione dei NULL con `ISNULL`
- Campi Filler per allineamento

Note tecniche

- Utilizza tabelle temporanee per performance

▼ Codice vista `olap.vwFactCostoTotaleperCP`

```
USE [BINEW_DWH]
GO

/***** Object: View [olap].[vwFactCostoTotaleperCP]  Script Date: 10/07/2025 10:47:13 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE VIEW [olap].[vwFactCostoTotaleperCP]
AS

SELECT
    id_DimSocieta = 1
    ,cp.QuantitaCosto
    ,cp.CostoUnitario
    ,cp.CostoTotale
    ,cp.id_DimDataDocumento
    ,cp.id_DimDataCompetenza
    --***** GG 20180410 modifiche per cambiare id dim albero (easyflex)
    --,cp.id_DimAlbero,
    --,coalesce(easFlex.id_DimAlbero, cp.id_DimAlbero) as id_DimAlbero
    -- 20181218: SR Fix rimappatura solo se unit Easyflex
    ,case when cp.id_DimAlbero in (2448,2805, 4949) -- 19/03/2025 Alessio Bagno: aggiunta unit easy flex CIUA
    then ISNULL(easFlex.id_DimAlbero, cp.id_DimAlbero) else cp.id_DimAlbero end as id_DimAlbero

    ,cp.id_DimClienti
    ,cp.id_DimContrattiPrestazione
    ,cp.id_DimLavoratori
    ,id_DimCausale = -1
    ,cp.id_DimContatti
    ,cp.id_DimContrattiQuadro
    ,id_DimProdottoCDG = 1
    ,id_dim_Transfer = case
        when ct.id_DimClientiTransfer is null then 0
        else 1
    end
    ,id_DimContrattiFornitura = isnull(cf.id_DimContrattiFornitura,-1)
    -- SR 20181121: Aggiunti id nuove dimensioni DimSorgente e DimTipoBusiness
    ,Id_DimSorgente = 1
    -- 20181218: SR Fix rimappatura solo se unit Easyflex
    ,Id_DimTipoBusiness = case when cp.id_DimAlbero in (2448,2805,4949) then 2 else 1 end -- 'RIUA0035','IHU'
FROM
    dwh.FactCostoTotaleperCP cp
    left join dwh.vw_BridgeCP_CF cf on cf.Id_DimContrattiPrestazione = cp.id_DimContrattiPrestazione
    left join BINEW_STG.etl.vwClientiTransferIdDim ct on cp.id_DimAlbero = ct.id_DimAlbero
        and cp.id_DimClienti = ct.id_DimClienti
        and cp.id_DimDataDocumento >= ct.id_DimDataInizio
        and cp.id_DimDataDocumento <= ct.id_DimDataFine
```

```

--***** GG 20181004 modifiche per cambiare id dim albero (easyflex) tramite unit si può correggere id_Dim
--left join ContrattiPrestazEasyflex as easFlex
-- on cp.id_DimContrattiPrestazione = easFlex.id_DimContrattiPrestazione
--***** GG 20181004 fine

--***** SR 20180607: Rimossa cte in favore di Tabella Bridge per fix caricamento cubo servizi digitali - Iniz
left join dwh.Bridge_EasyFlex as easFlex
    on cp.id_DimContrattiPrestazione = easFlex.id_DimContrattiPrestazione
--***** NL 20201116: Aggiunta condizione su periodo di validità del mapping
AND id_DimDataDocumento
    BETWEEN (YEAR(easFlex.scd2_valid_from) * 100 + MONTH(easFlex.scd2_valid_from))*100+1
    AND ISNULL((YEAR(easFlex.scd2_valid_to)*100 + MONTH(easFlex.scd2_valid_to))*100+1,20990101)
--***** SR 20180607: Rimossa cte in favore di Tabella Bridge per fix caricamento cubo servizi digitali - Fin

UNION ALL

-- tabella correzioni post caricamento: [BINEW_DWH].[dbo].[tbsysCorrezioniCDG]
SELECT
    id_DimSocieta = isnull(cp2.id_DimSocieta,1)
    ,cp2.[Qta Costo]
    ,cp2.[Costo Unitario]
    ,cp2.[Costo Totale]
    ,cp2.id_DimDataDocumento
    ,cp2.id_DimDataCompetenza
    ,case when cp2.id_DimAlbero in (2448,2805,4949 -- 19/03/2025 Alessio Bagno: aggiunta unit easy flex CIUA
    ) then ISNULL(easFlex.id_DimAlbero, cp2.id_DimAlbero) else cp2.id_DimAlbero end as id_DimAlbero --,cp2.id
    ,cp2.id_DimClienti
    ,cp2.id_DimContrattiPrestazione
    ,cp2.id_DimLavoratori
    ,cp2.id_DimCausali
    ,cp2.id_DimContatti
    ,cp2.id_DimContrattiQuadro
    ,id_DimProdottoCDG = 1
    ,id_dim_Transfer = case
        when ct.id_DimClientiTransfer is null then 0
        else 1
    end
    ,id_DimContrattiFornitura = isnull(cf.id_DimContrattiFornitura,-1)
-- SR 20181121: Aggiunti id nuove dimensioni DimSorgente e DimTipoBusiness
    ,Id_DimSorgente = 3
    ,Id_DimTipoBusiness = case when cp2.id_DimAlbero in (2448,2805,4949) -- 19/03/2025 Alessio Bagno: aggi
    then 2 else 1 end
FROM
    [BINEW_DWH].[dbo].[tbsysCorrezioniCDG] cp2
left join dwh.vw_BridgeCP_CF cf on cf.Id_DimContrattiPrestazione = cp2.id_DimContrattiPrestazione
left outer join BINEW_STG.etl.vwClientiTransferIdDim ct on cp2.id_DimAlbero = ct.id_DimAlbero
    and cp2.id_DimClienti = ct.id_DimClienti
    and cp2.id_DimDataDocumento >= ct.id_DimDataInizio
    and cp2.id_DimDataDocumento <= ct.id_DimDataFine
left join dwh.Bridge_EasyFlex as easFlex
    on cp2.id_DimContrattiPrestazione = easFlex.id_DimContrattiPrestazione
--***** NL 20201116: Aggiunta condizione su periodo di validità del mapping
AND id_DimDataDocumento

```

```

        BETWEEN (YEAR(easFlex.scd2_valid_from) * 100 + MONTH(easFlex.scd2_valid_from))*100+1
        AND ISNULL((YEAR(easFlex.scd2_valid_to)*100 + MONTH(easFlex.scd2_valid_to))*100+1,20990101)
    WHERE
    cp2.Source='C'

UNION ALL

-- dati StartPeople
SELECT
    id_DimSocieta = 2
    ,[Qta Costo] = cp3.QuantitaCosto
    ,[Costo Unitario] = cp3.CostoUnitario
    ,[Costo Totale] = cp3.CostoTotale
    ,ISNULL(cp3.id_DimDataDocumento, -1)
    ,ISNULL(cp3.id_DimDataCompetenza, -1)
    ,ISNULL(cp3.id_DimAlbero, -1)
    ,ISNULL(cp3.id_DimClienti, -1)
    ,ISNULL(cp3.id_DimContrattiPrestazione, -1)
    ,ISNULL(cp3.id_DimLavoratori, 1)
    ,id_DimCausali = -1
    ,ISNULL(cp3.id_DimContatti, -1)
    ,id_DimContrattiQuadro = -1
    ,id_DimProdottoCDG = 1
    ,id_dim_Transfer = case
        when ct.id_DimClientiTransfer is null then 0
        else 1
    end
    ,id_DimContrattiFornitura = isnull(cf.id_DimContrattiFornitura,-1)
    -- SR 20181121: Aggiunti id nuove dimensioni DimSorgente e DimTipoBusiness
    ,Id_DimSorgente = 7
    ,Id_DimTipoBusiness = 1
FROM
    sp.FactCostoTotaleperCP cp3
    left join dwh.vw_BridgeCP_CF cf on cf.Id_DimContrattiPrestazione = cp3.id_DimContrattiPrestazione
    left join BINEW_STG.etl.vwClientiTransferIdDim ct on cp3.id_DimAlbero = ct.id_DimAlbero
        and cp3.id_DimClienti = ct.id_DimClienti
        and cp3.id_DimDataDocumento >= ct.id_DimDataInizio
        and cp3.id_DimDataDocumento <= ct.id_DimDataFine

UNION ALL

-- dati disponibilità extra file fattura
Select
    id_DimSocieta = 1
    ,[qtaCosto]
    ,[CostoUnitario] = null
    ,[CostoTotale]
    ,[id_DimDataDocumento]
    ,[id_DimDataCompetenza]
    ,[id_DimAlbero]
    ,[id_dimClienti]
    ,cp.[Id_dimcontrattiPrestazione]
    ,[id_DimLavoratori]
    ,ID_DimCausale = -1

```

```

,Id_DimContatti = -1
,id_DimContrattiQuadro
,Id_dimProdottoCdG = 1
,Id_transfer = 0
,id_DimContrattiFornitura = isnull(cf.id_DimContrattiFornitura,-1)
-- SR 20181121: Aggiunti id nuove dimensioni DimSorgente e DimTipoBusiness
,Id_DimSorgente = 9
,Id_DimTipoBusiness = 1
from
    dwh.FactDisponibilitaCosti cp
    left join dwh.vw_BridgeCP_CF cf on cf.Id_DimContrattiPrestazione = cp.id_DimContrattiPrestazione

UNION ALL

-- dati OL
SELECT
    id_DimSocieta = 3
    ,cp.QuantitaCosto
    ,cp.CostoUnitario
    ,cp.CostoTotale
    /*
RE Rivoiro 20170216 Ripristinata data documento = data competenza come richiesto da Controller (Fabrizia La
*/
    --,[id_DimDataDocumento]
    ,[id_DimDataDocumento] = cp.[id_DimDataCompetenza]
    ,cp.id_DimDataCompetenza
    ,cp.id_DimAlbero
    ,cp.id_DimClienti
    ,cp.id_DimContrattiPrestazione
    ,cp.id_DimLavoratori
    ,id_DimCausale = -1
    ,cp.id_DimContatti
    ,cp.id_DimContrattiQuadro
    ,id_DimProdottoCDG = 1
    ,id_dim_Transfer = case
        when ct.id_DimClientiTransfer is null then 0
        else 1
    end
    ,id_DimContrattiFornitura = isnull(cf.id_DimContrattiFornitura,-1)
-- SR 20181121: Aggiunti id nuove dimensioni DimSorgente e DimTipoBusiness
,Id_DimSorgente = 6
,Id_DimTipoBusiness = 1
FROM
    dwh_ol.FactCostoTotaleperCP cp
    left join dwh.vw_BridgeCP_CF cf on cf.Id_DimContrattiPrestazione = cp.id_DimContrattiPrestazione
    left join BINEW_STG.etl.vwClientiTransferIdDim ct on cp.id_DimAlbero = ct.id_DimAlbero
        and cp.id_DimClienti = ct.id_DimClienti
        and cp.id_DimDataDocumento >= ct.id_DimDataInizio
        and cp.id_DimDataDocumento <= ct.id_DimDataFine
    left join olap.vwDimAlbero a on a.id_DimAlbero = cp.id_DimAlbero
where
    cp.id_DimAlbero not in (-999,-998) -- esclusione TEMP0, TEMP1

UNION ALL

```

```

-- dati disponibilità extra file fattura
Select
    id_DimSocieta = 3
    ,[qtaCosto]
    ,[CostoUnitario] = null
    ,[CostoTotale]
    /*
    RE Rivoiro 20170216 Ripristinata data documento = data competenza come richiesto da Controller (Fabrizia L
    */
    --,[id_DimDataDocumento]
    ,[id_DimDataDocumento] = [id_DimDataCompetenza]
    ,[id_DimDataCompetenza]
    ,[id_DimAlbero]
    ,[id_dimClienti]
    ,cp.[Id_dimcontrattiPrestazione]
    ,[id_DimLavoratori]
    ,ID_DimCausale = -1
    ,ID_DimContatti = -1
    ,id_DimContrattiQuadro
    ,Id_dimProdottoCdg = 1
    ,Id_transfer = 0
    ,id_DimContrattiFornitura = isnull(cf.id_DimContrattiFornitura,-1)
    -- SR 20181121: Aggiunti id nuove dimensioni DimSorgente e DimTipoBusiness
    ,Id_DimSorgente = 10
    ,Id_DimTipoBusiness = 1
from
    dwh_ol.FactDisponibilitaCosti cp
    left join dwh.vw_BridgeCP_CF cf on cf.Id_DimContrattiPrestazione = cp.id_DimContrattiPrestazione
where
    id_DimAlbero not in (-999,-998) -- esclusione TEMP0, TEMP1

UNION ALL

-- dati IN
SELECT
    id_DimSocieta = 4
    ,QuantitaCosto
    ,CostoUnitario
    ,CostoTotale
    /*
    RE Rivoiro 20170216 Ripristinata data documento = data competenza come richiesto da Controller (Fabrizia L
    RE Rivoiro 20170330 Modifica per gestione datadocumento a partire da Gennaio 2017 come richiesto da Cont
    */
    --[id_DimDataDocumento],
    --[id_DimDataDocumento] = [id_DimDataCompetenza]
    ,[id_DimDataDocumento] = case when [id_DimDataDocumento] < 20170101 then [id_DimDataCompetenza] els
    ,id_DimDataCompetenza
    ,cp.id_DimAlbero
    ,cp.id_DimClienti
    ,cp.id_DimContrattiPrestazione
    ,id_DimLavoratori
    ,id_DimCausale=-1
    ,id_DimContatti

```

```

,id_DimContrattiQuadro
,id_DimProdottoCDG = 1
,id_dim_Transfer = case
    when ct.id_DimClientiTransfer is null then 0
    else 1
end
,id_DimContrattiFornitura = isnull(cf.id_DimContrattiFornitura,-1)
-- SR 20181121: Aggiunti id nuove dimensioni DimSorgente e DimTipoBusiness
,id_DimSorgente = 5
,id_DimTipoBusiness = 1
FROM
    dwh_in.FactCostoTotaleperCP cp
left join dwh.vw_BridgeCP_CF cf on cf.Id_DimContrattiPrestazione = cp.id_DimContrattiPrestazione
left join BINEW_STG.etl.vwClientiTransferIdDim ct on cp.id_DimAlbero = ct.id_DimAlbero
    and cp.id_DimClienti = ct.id_DimClienti
    and cp.id_DimDataDocumento >= ct.id_DimDataInizio
    and cp.id_DimDataDocumento <= ct.id_DimDataFine
left join olap.vwDimAlbero a on a.id_DimAlbero = cp.id_DimAlbero
WHERE
    cp.id_DimAlbero not in (-999,-998) -- esclusione TEMP0, TEMP1

UNION ALL
--GG 20180523 add union costi welfare

--forzature dei costi welfare da excel
Select
    w.[id_DimSocieta]
    ,w.[QuantitaCosto]
    ,w.[CostoUnitario]
    ,w.[CostoTotale]
    ,w.[id_DimDataDocumento]
    ,w.[id_DimDataCompetenza]
    ,case when W.id_DimAlbero in (2448,2805,4949) -- 19/03/2025 Alessio Bagno: aggiunta unit easy flex CIUA0C
    then ISNULL(easFlex.id_DimAlbero, W.id_DimAlbero) else w.id_DimAlbero end as id_DimAlbero
    ,w.[id_DimClienti]
    ,w.[id_DimContrattiPrestazione]
    ,w.[id_DimLavoratori]
    ,w.[id_DimCausale]
    ,w.[id_DimContatti]
    ,w.[id_DimContrattiQuadro]
    ,w.[id_DimProdottoCDG]
    ,w.[id_dim_Transfer]
    ,w.[id_DimContrattiFornitura]
    -- SR 20181121: Aggiunti id nuove dimensioni DimSorgente e DimTipoBusiness
    ,Id_DimSorgente = 4 -- welfare
    ,Id_DimTipoBusiness = 1
from dwh.CorrezioneCostiWelfare W
    left join dwh.Bridge_EasyFlex as easFlex
        on W.id_DimContrattiPrestazione = easFlex.id_DimContrattiPrestazione
--***** NL 20201116: Aggiunta condizione su periodo di validità del mapping
AND id_DimDataDocumento
    BETWEEN (YEAR(easFlex.scd2_valid_from) * 100 + MONTH(easFlex.scd2_valid_from))*100+1
    AND ISNULL((YEAR(easFlex.scd2_valid_to)*100 + MONTH(easFlex.scd2_valid_to))*100+1,20990101)

```


GO

▼ Codice vista `olap.vwFactFattureCDG`

```
USE [BINEW_DWH]
GO

/***** Object: View [olap].[vwFactFattureCDG]  Script Date: 10/07/2025 10:49:02 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE VIEW [olap].[vwFactFattureCDG] as

with CTE_Easyflex as (
SELECT
    id_DimSocieta = 1
    --***** GG 20180410 modifiche per cambiare id dim albero (easyflex)
    --,fatt.id_DimAlbero
    --,coalesce(easFlex.id_DimAlbero, fatt.id_DimAlbero) as id_DimAlbero
    -- 20181218: SR Fix rimappatura solo se unit Easyflex
    ,case when fatt.id_DimAlbero in (2448,2805,4949) -- 19/03/2025 Alessio Bagno: aggiunta unit easy flex CIUA
    then ISNULL(easFlex.id_DimAlbero, fatt.id_DimAlbero) else fatt.id_DimAlbero end as id_DimAlbero
    ,id_DimLavoratori
    ,id_DimDataDocumento = Case when id_DimDataDocumento < 20060101 THEN -1 ELSE id_DimDataDocumento
    ,id_DimDataCompetenza=Case when id_DimDataCompetenza < 20060101 THEN -1 ELSE id_DimDataCompetenza
    ,fatt.id_DimClienti
    ,fatt.id_DimContiContabili
    ,fatt.id_DimContrattiPrestazione
    ,id_DimDocumenti
    ,id_DimLineeBusiness
    ,id_DimStatiElaborazione
    ,fatt.id_DimCausali
    ,fatt.id_DimContatti
    ,id_DimContrattiQuadro
    ,id_DimBusinessLineCDG
    ,id_DimProdottoCDG
    --,id_dim_Transfer = case when ct.id_DimClientiTransfer is null then 0 else 1 end
    ,Quantita
    ,PrezzoUnitario
    ,Fatturato
    ,CostoPresuntoUnitario
    ,CostoTecnicoUnitario
    ,CostoPresuntoTotale=CostoPresuntoUnitario*Quantita
    ,CostoTecnicoTotale=CostoTecnicoUnitario*Quantita
    ,OreAssenzaFatt = Case when bk_GruppoStatistico = 9 Then Quantita Else 0 End
    ,OreLavorabiliFatt = Case when bk_GruppoStatistico in (1,3,9) Then Quantita Else 0 End
    ,id_DimContrattiFornitura = isnull(cf.id_DimContrattiFornitura,-1)
    -- SR 20181121: Aggiunti id nuove dimensioni DimSorgente e DimTipoBusiness
    ,Id_DimSorgente = 1
```

```

-- ,Id_DimTipoBusiness = case when easFlex.id_DimAlbero is null then 1 else 2 end
,Id_DimTipoBusiness = case when fatt.id_DimAlbero in (2448,2805,4949) then 2 else 1 end -- 'RIUA0035','IHL

from dwh.FactFatture as fatt

left join dwh.vw_BridgeCP_CF cf on cf.Id_DimContrattiPrestazione = fatt.id_DimContrattiPrestazione
join olap.vwDimContiContabili as conti on conti.id_DimContiContabili = fatt.id_DimContiContabili
join olap.vwDimCausali as causali on causali.id_DimCausali = fatt.id_DimCausali

-- /

--***** GG 20180410 modifiche per cambiare id dim albero (easyflex) tramite unit si può correggere id_DimAl
--left join ContrattiPrestazEasyflex as easFlex
-- on fatt.id_DimContrattiPrestazione = easFlex.id_DimContrattiPrestazione
--***** GG 20180410 fine
--***** SR 20180607: Rimossa cte in favore di Tabella Bridge per fix caricamento cubo servizi digitali - Inizio

left join dwh.Bridge_EasyFlex as easFlex
on fatt.id_DimContrattiPrestazione = easFlex.id_DimContrattiPrestazione
--***** NL 20201016: Aggiunta condizione su periodo di validità del mapping
AND id_DimDataDocumento
BETWEEN (YEAR(easFlex.scd2_valid_from) * 100 + MONTH(easFlex.scd2_valid_from))*100+1
AND ISNULL((YEAR(easFlex.scd2_valid_to)*100 + MONTH(easFlex.scd2_valid_to))*100+1,20990101)
where
--id_DimDataDocumento<20190801 and --gt 20190903 eliminiamo i dati di agosto
fatt.id_DimAlbero <> 1387 -- esclusione Politiche Attive Unit PAHO0001 20150216
AND ISNULL(causali.bk_GruppoStatistico,-1) <> 24
and causali.bk_Causale_WN<>'A$COSTO_TIRO'--modifica gt 20200313 per elinare costo tirocinanti
and exists (
select top 1 1 from [MDS].[mdm].[Conti_Contabili] cc
where
(conti.bk_ContoContabile = cc.Name and cc.Anno is null)
OR
(conti.bk_ContoContabile = cc.Name and LEFT(fatt.id_DimDataDocumento, 4) >= cast(cc.Anno as int))
)
)
-- transfer su easyflex
SELECT
id_DimSocieta
,ef.id_DimAlbero
,id_DimLavoratori
,id_DimDataDocumento
,id_DimDataCompetenza
,ef.id_DimClienti
,id_DimContiContabili
,id_DimContrattiPrestazione
,id_DimDocumenti
,id_DimLineeBusiness
,id_DimStatiElaborazione
,id_DimCausali
,id_DimContatti
,id_DimContrattiQuadro
,id_DimBusinessLineCDG
,id_DimProdottoCDG

```

```

,id_dim_Transfer = case when ct.id_DimClientiTransfer is null then 0 else 1 end
,Quantita
,PrezzoUnitario
,Fatturato
,CostoPresuntoUnitario
,CostoTecnicoUnitario
,CostoPresuntoTotale
,CostoTecnicoTotale
,OreAssenzaFatt
,OreLavorabiliFatt
,id_DimContrattiFornitura
,Id_DimSorgente
,Id_DimTipoBusiness
from CTE_Easyflex as ef
left join BINEW_STG.etl.vwClientiTransferIdDim as ct on ct.id_DimAlbero = ef.id_DimAlbero
and ct.id_DimClienti = ef.id_DimClienti
and ef.id_DimDataDocumento >= ct.id_DimDataInizio
and ef.id_DimDataDocumento <= ct.id_DimDataFine

UNION ALL --tabella Martellate

Select
id_DimSocieta = isnull(id_DimSocieta,1)
,mart.id_DimAlbero
,id_DimLavoratori
,id_DimDataDocumento
,id_DimDataCompetenza
,mart.id_DimClienti
,mart.id_DimContiContabili
,mart.id_DimContrattiPrestazione
,id_DimDocumenti
,id_DimLineeBusiness
,id_DimStatiElaborazione
,mart.id_DimCausali
,id_DimContatti
,IsNull(ID_dimcontrattiquadro,-1) as ID_dimcontrattiquadro
,id_DimBusinessLineCDG =
case when id_DimBusinessLineCDG is null then
(case
when TipoRiga = 'Somministrazione' then 8
when TipoRiga = 'Ricerca e Selezione' then 7
else -1
end)
else id_DimBusinessLineCDG
end
,id_DimProdottoCDG =
case when id_DimProdottoCDG is null then
(case
when causali.bk_GruppoStatistico = '26' then 2
When Causale='A$R&S_PROF' then 3
when TipoRiga = 'Somministrazione' then 1
when TipoRiga = 'Ricerca e Selezione' then 2
else -1
end)
end)

```

```

        else id_DimProdottoCDG
        end
    ,case
        when ct.id_DimClientiTransfer is null then 0
        else 1
    end as id_dim_Transfer
    ,[Qta FT]
    ,[Importo Unitario FT]
    ,[Importo Totale FT]
    ,CostoPresuntoUnitario = 0
    ,CostoTecnicoUnitario = 0
    ,CostoPresuntoTotale=0 --CostoPresuntoUnitario*Quantita
    ,CostoTecnicoTotale=0 --CostoTecnicoUnitario*Quantita
    ,OreAssenzaFatt=Case when bk_GruppoStatistico =9 Then [Qta FT] Else 0 End
    ,OreLavorabiliFatt=Case when bk_GruppoStatistico in (1,3,9) Then [Qta FT] Else 0 End
    ,id_DimContrattiFornitura = isnull(cf.id_DimContrattiFornitura,-1)
    -- SR 20181121: Aggiunti id nuove dimensioni DimSorgente e DimTipoBusiness
    ,Id_DimSorgente = 3
    --,Id_DimTipoBusiness = 1 AS: 20220721 - ho inserito la condizione sottostante al posto di questa per
    -- prendere gli adjustment effettuati sulle unit easyflex
    ,Id_DimTipoBusiness = case when Mart.id_DimAlbero in (2448,2805,4949) -- 19/03/2025 Alessio Bagno: agg
    then 2 else 1 end
from
    dbo.tbsysCorrezioniCDG as Mart
    left join dwh.vw_BridgeCP_CF cf on cf.Id_DimContrattiPrestazione = Mart.id_DimContrattiPrestazione
    join olap.vwDimContiContabili as conti on conti.id_DimContiContabili = Mart.id_DimContiContabili
    join olap.vwDimCausali as causali on causali.id_DimCausali = Mart.id_DimCausali
    left join BINEW_STG.etl.vwClientiTransferIdDim ct on mart.id_DimAlbero = ct.id_DimAlbero
        and mart.id_DimClienti = ct.id_DimClienti
        and mart.id_DimDataDocumento >= ct.id_DimDataInizio
        and mart.id_DimDataDocumento <= ct.id_DimDataFine
WHERE
    Source='F'

UNION ALL --- dati StartPeople

SELECT
    id_DimSocieta = 2
    ,isnull( ff.id_DimAlbero, -1)
    ,isnull ( [id_DimLavoratori], -1)
    ,isnull ( [id_DimDataDocumento], -1)
    ,isnull ( [id_DimDataCompetenza], -1)
    ,isnull ( ff.id_DimClienti, -1)
    ,isnull ( [id_DimContiContabili], -1)
    ,isnull ( ff.[id_DimContrattiPrestazione], -1)
    ,isnull ( [id_DimDocumenti], -1)
    ,isnull ( [id_DimLineeBusiness], -1)
    ,isnull ( [id_DimStatiElaborazione], -1)
    ,isnull ( [id_DimCausali], -1)
    ,isnull ( [id_DimContatti], -1)
    ,-1
    ,case
        when bk_tipo_documento in ('FATTURE SELEZIONE','NOTE CREDITO SELEZIONE') then 7
        else 8
    end

```

```

end as id_DimBusinessLineCDG
,case
  when bk_tipo_documento in ('FATTURE SELEZIONE','NOTE CREDITO SELEZIONE') then 2
  else 1
end as id_DimProdottoCDG
,case
  when ct.id_DimClientiTransfer is null then 0
  else 1
end as id_dim_Transfer
,[Quantita]
,[PrezzoUnitario]
,[Fatturato]
,[CostoPresuntoUnitario]
,[CostoTecnicoUnitario]
,[CostoPresuntoTotale] = null
,[CostoTecnicoTotale] = null
,[OreAssenzaFatt] = null
,[OreLavorabiliFatt] = null
,id_DimContrattiFornitura = isnull(cf.id_DimContrattiFornitura,-1)
-- SR 20181121: Aggiunti id nuove dimensioni DimSorgente e DimTipoBusiness
,id_DimSorgente = 7
,id_DimTipoBusiness = 1
FROM
  sp.FactFatture ff
  left join dwh.vw_BridgeCP_CF cf on cf.Id_DimContrattiPrestazione = ff.id_DimContrattiPrestazione
  left join BINEW_STG.etl.vwClientiTransferIdDim ct on ff.id_DimAlbero = ct.id_DimAlbero
    and ff.id_DimClienti = ct.id_DimClienti
    and ff.id_DimDataDocumento >= ct.id_DimDataInizio
    and ff.id_DimDataDocumento <= ct.id_DimDataFine
WHERE
  bk_codice_causale not in ('ATTFORMA', 'Bolli','SC')

UNION ALL

-- dati OL SPA
SELECT
  id_DimSocieta = 3
  ,fatt.id_DimAlbero
  ,id_DimLavoratori
  /*
RE Timpa 20160802 deciso con M Santa di usare la data di competenza
Remmata la riga sotto per eventuale RollBack e cambiato da id_DimDataDocumento a id_DimDataCompetenza
RE Rivoiro 20160202 Ripristinata data competenza come richiesto da Controller
RE Rivoiro 20170216 Ripristinata data documento = data competenza come richiesto da Controller (Fabrizia L&
*/
  --,id_DimDataDocumento=Case when id_DimDataDocumento<20060101
  -- THEN -1 ELSE id_DimDataDocumento  END
  ,id_DimDataDocumento = id_DimDataCompetenza
  ,id_DimDataCompetenza = id_DimDataCompetenza
  ,fatt.id_DimClienti
  ,fatt.id_DimContiContabili
  ,fatt.id_DimContrattiPrestazione
  ,id_DimDocumenti
  ,id_DimLineeBusiness

```

```

,id_DimStatiElaborazione
,fatt.id_DimCausali,
,fatt.id_DimContatti,
id_DimContrattiQuadro,
id_DimBusinessLineCDG
,id_DimProdottoCDG
,case
    when ct.id_DimClientiTransfer is null then 0
    else 1
end as id_dim_Transfer
,Quantita
,PrezzoUnitario
,Fatturato,
CostoPresuntoUnitario
,CostoTecnicoUnitario
,CostoPresuntoTotale = CostoPresuntoUnitario * Quantita
,CostoTecnicoTotale = CostoTecnicoUnitario * Quantita
,OreAssenzaFatt = Case when bk_GruppoStatistico = 9 Then Quantita Else 0 End
,OreLavorabiliFatt = Case when bk_GruppoStatistico in (1,3,9) Then Quantita Else 0 End
,id_DimContrattiFornitura = isnull(cf.id_DimContrattiFornitura,-1)
-- SR 20181121: Aggiunti id nuove dimensioni DimSorgente e DimTipoBusiness
,Id_DimSorgente = 6
,Id_DimTipoBusiness = 1
from
    dwh_ol.FactFatture as fatt
left join dwh.vw_BridgeCP_CF cf on cf.Id_DimContrattiPrestazione = fatt.id_DimContrattiPrestazione
join olap.vwDimProdottoCDG as p on p.IdProdottoCDG = fatt.id_dimProdottoCDG
join olap.vwDimContiContabili as conti on conti.id_DimContiContabili=fatt.id_DimContiContabili
join olap.vwDimCausali as causali on causali.id_DimCausali=fatt.id_DimCausali
left join BINEW_STG.etl.vwClientiTransferIdDim ct on fatt.id_DimAlbero = ct.id_DimAlbero
    and fatt.id_DimClienti = ct.id_DimClienti
    and fatt.id_DimDataDocumento >= ct.id_DimDataInizio
    and fatt.id_DimDataDocumento <= ct.id_DimDataFine
left join olap.vwDimAlbero a on a.id_DimAlbero = fatt.id_DimAlbero
where
    p.ProdottoCDG in ('Temps','Permanent')
    and fatt.id_DimAlbero not in (-999,-998) -- esclusione TEMP0, TEMP1

UNION ALL

-- dati IN

select
    id_DimSocieta = 4
    ,fatt.id_DimAlbero
    ,id_DimLavoratori
    /*
        RE Timpa 20160802 deciso con M Santa di usare la data di competenza
        Remmata la riga sotto per eventuale RollBack e cambiato da id_DimDataDocumento a id_DimDataCompeten
        RE Rivoiro 20160202 Ripristinata data competenza come richiesto da Controller
        RE Rivoiro 20170216 Ripristinata data documento = data competenza come richiesto da Controller (Fabrizia
        RE Rivoiro 20170330 Modifica per gestione datadocumento a partire da Gennaio 2017 come richiesto da Cc
    */
    --,id_DimDataDocumento=Case when id_DimDataDocumento<20060101

```

```

-- THEN -1 ELSE id_DimDataDocumento END
,[id_DimDataDocumento] = case when [id_DimDataDocumento] < 20170101 then [id_DimDataCompetenza] els
-- ,id_DimDataDocumento = id_DimDataCompetenza
, id_DimDataCompetenza = id_DimDataCompetenza
, fatt.id_DimClienti,
fatt.id_DimContiContabili,
fatt.id_DimContrattiPrestazione
, id_DimDocumenti, id_DimLineeBusiness, id_DimStatiElaborazione
, fatt.id_DimCausali,
fatt.id_DimContatti,
id_DimContrattiQuadro,
id_DimBusinessLineCDG,id_DimProdottoCDG = 1
,case
    when ct.id_DimClientiTransfer is null then 0
    else 1
end as id_dim_Transfer
,Quantita, PrezzoUnitario, Fatturato,CostoPresuntoUnitario
,CostoTecnicoUnitario
,CostoPresuntoTotale = CostoPresuntoUnitario * Quantita
,CostoTecnicoTotale = CostoTecnicoUnitario * Quantita
,OreAssenzaFatt = Case when bk_GruppoStatistico = 9 Then Quantita Else 0 End
,OreLavorabiliFatt = Case when bk_GruppoStatistico in (1,3,9) Then Quantita Else 0 End
,id_DimContrattiFornitura = isnull(cf.id_DimContrattiFornitura,-1)
-- SR 20181121: Aggiunti id nuove dimensioni DimSorgente e DimTipoBusiness
,Id_DimSorgente = 5
,Id_DimTipoBusiness = 1
from
    dwh_in.FactFatture as fatt
left join dwh.vw_BridgeCP_CF cf on cf.Id_DimContrattiPrestazione = fatt.id_DimContrattiPrestazione
Inner join olap.vwDimContiContabili as conti on conti.id_DimContiContabili=fatt.id_DimContiContabili
inner join olap.vwDimCausali as causali on causali.id_DimCausali=fatt.id_DimCausali
left outer join BINEW_STG.etl.vwClientiTransferIdDim ct on fatt.id_DimAlbero = ct.id_DimAlbero
--where id_DimDataDocumento<20190801 --AS 20190903 eliminiamo i dati di agosto

GO

```

▼ Codice funzione IAM.udf_Dati_IBLKIF

```

USE [BINEW_HELPER]
GO

/***** Object: UserDefinedFunction [IAM].[udf_Dati_IBLKIF]  Script Date: 10/07/2025 11:01:04 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE function [IAM].[udf_Dati_IBLKIF] (@periodo int, @Divisioncode varchar(7))

returns table
as
return

```

```

/*
--INIZIO COMMENTO: MODIFICHE Huawei patch - FU 11/01/2018
SELECT Distinct

    [Record Type] = 'D'
    , [Interface File Name] = 'IBLKIF'
    , [Mutation Code] = 'I'
    , [Country Code] = 'IT'
    , [Division Code] = CONVERT(char(15), @Divisioncode)
    , [Client Number 1] =-- IAM.[udf_Get_IDclientiWN](con.[bk_CustomerNo]) --
    con.[bk_CustomerNo]

    , [Client Number 2] = CASE WHEN NOT FORN.Unit IS NULL THEN FORN.Unit
    ELSE isnull(CON.bk_Unit , convert(char(15),'')) END
    , [Client Number 3] = convert(char(15),'')

    , [Client Name] = LEFT(isnull(CLI.RagioneSociale , convert(char(50),'')), 50)
    , [Street Name] = isnull(CLI.Indirizzo , convert(char(50),''))
    , [House Number] = convert(char(15),'') -- LEFT(ISNULL(CLI.[Address 2], convert(char(15),'')),15)
    , [Postal Code] = LEFT(isnull(CLI.cap , convert(char(10),'')),10)
    , [Municipality] = Isnull(CLI.Citta , convert(char(50),''))
    , [Concern Code] = isnull(CLIAM.CodInternazionale , convert(char(15),''))
    , Filler = ' '
    , [Client Size] = max(convert(char(1),isnull(CON.ContattoNumeroDipendenti , '')))
    , Filler2 = ' '
    , [Indication International Account] = CASE
    WHEN LEN(CLIAM.CodiceContabileCliente ) > 0
    THEN '1'
    ELSE '0'
    END
    , [Search Name] = LEFT(isnull(CLI.RagioneSociale, convert(char(50),'')), 50) -- VERRA' TRONCATO A 50 CARA'
    , [NACE Code] = convert(char(6),Replace(isnull(AtecoCodice,0),',',''))--convert(char(6),'') Cambiato da G.Timpa
    , Filler3 = ' '
    , [Client Status] = MAX(CASE CON.ClientStatus -- aggiunta max 21/02/2017 andrea molteni duplica cir food
    WHEN 'PROSPECT' THEN 1
    WHEN 'MAI' THEN 1
    WHEN 'POTENZIALE' THEN 1
    WHEN 'PERSO' THEN 2
    WHEN 'LATENTE' THEN 3
    WHEN 'ATTIVO' THEN 3
    WHEN 'NUOVO' THEN 3
    WHEN 'SVILUPPATO' THEN 3 -- CODIFICA EFFETTUATA CON GIORGIO SAVINO IL 12.05.2010
    ELSE 0
    END) -- PROBLEMA: I VALORI POSSONO CAMBIARE PERCHE' FANNO PARTE DI
    -- UNA TABELLA DI BASE: PER QUESTO E' STATA EMESSA UNA SEGNALAZIONE
    , Filler4 = ' '
    , [Date First Invoice] = isnull(convert(varchar(10),MIN(id_DimDataDocumento)), '00000000')
    , null as [Report Record]

FROM
    BINEW_DWH.dwh.DimContatti CON

left outer join BINEW_DWH.dwh.DimClienti CLI

```



```

on CON.bk_CustomerNo = CLI.bk_CodiceContabile

left outer join BINEW_DWH.dwh.FactFatture CDG
  ON CDG.[id_DimClienti] = CLI.id_DimClienti
  AND CDG.id_DimContatti = CON.id_DimContatti
  AND CDG.id_DimDataDocumento = @periodo

left outer join BINEW_DWH.dwh.DimContrattiPrestazione CP
  on CDG.id_DimContrattiPrestazione = CP.id_DimContrattiPrestazione

left outer join BINEW_DWH.dwh.DimContrattiFornitura FORN
  ON CP.cf = FORN.bk_ContrattoFornitura
  --AND FORN.YYYYMM = @periodo

left outer join BINEW_STG.stg.tblGruppiClienti CLIAM
on CON.bk_CustomerNo = CLIAM.CodiceContabileCliente

WHERE
  NOT CON.ClientStatus IS NULL
  AND NOT CON.bk_Unit IS NULL
  and con.bk_CustomerNo <> '-1'
  and not IAM.[udf_Get_IDclientiWN](con.[bk_CustomerNo]) is null -- 2015.09.11 FP per passare i contatti senz

GROUP BY
Replace(isnull(AtecoCodice,0),',',''),-- Cambiato da G.Timpanelli 20111121
CON.bk_CustomerNo,
CASE WHEN NOT FORN.Unit IS NULL THEN FORN.Unit
      ELSE isnull(CON.bk_Unit , convert(char(15),'')) END
,isnull(CLI.RagioneSociale , convert(char(50),''))
,isnull(CLI.Indirizzo , convert(char(50),''))
--,LEFT(ISNULL(CLI.[Address 2], convert(char(15),'')),15)
, LEFT(isnull(CLI.CAP , convert(char(10),'')),10)
, Isnull(CLI.Citta, convert(char(50),''))
, Isnull(CLIAM.CodInternazionale , convert(char(15),''))
--, convert(char(1),isnull(CON.ContattoNumeroDipendenti , ''))
,CASE
  WHEN LEN(CLIAM.CodiceContabileCliente) > 0
  THEN '1'
  ELSE '0'
END
,isnull(CLI.RagioneSociale, convert(char(50),''))
;
--FINE COMMENTO: MODIFICHE Huawei patch - FU 11/01/2018
*/

WITH docFirstInvoice AS (
  SELECT
    vwfatt.[id_DimClienti]
    , [id_DimDataDocumento] = MIN(vwfatt.[id_DimDataDocumento])
  FROM BINEW_DWH.olap.vwFactFattureCDG vwfatt
  INNER JOIN BINEW_DWH.dwh.DimProdottoCDG pr ON vwfatt.id_DimProdottoCDG = pr.IdProdottoCDG AND pr.P
  INNER JOIN BINEW_DWH.dwh.DimAlbero ALB on vwfatt.id_DimAlbero = alb.id_DimAlbero AND alb.Divisione IN (
  INNER JOIN BINEW_DWH.dwh.DimSocieta s ON vwfatt.id_DimSocieta = s.id_DimSocieta AND s.bk_Societa = 'R';

```

```

GROUP BY
    vwfatt.[id_DimClienti]
)
, CDG AS (
    SELECT
        vwfatt.[id_DimClienti]
        , vwfatt.[id_DimDataDocumento]
        , vwfatt.[id_DimAlbero]
        , vwfatt.[id_DimContrattiPrestazione]
    FROM [BINEW_HELPER].[IAM].[RicaviCP_TMP] vwfatt
    INNER JOIN BINEW_DWH.dwh.DimProdottoCDG pr ON vwfatt.id_DimProdottoCDG = pr.IdProdottoCDG AND pr.P
    INNER JOIN BINEW_DWH.dwh.DimAlbero ALB on vwfatt.id_DimAlbero = alb.id_DimAlbero AND alb.Divisione IN (
    INNER JOIN BINEW_DWH.dwh.DimSocieta s ON vwfatt.id_DimSocieta = s.id_DimSocieta AND s.bk_Societa = 'R
    WHERE vwfatt.id_DimDataDocumento = @periodo
    GROUP BY
        vwfatt.[id_DimClienti]
        , vwfatt.[id_DimDataDocumento]
        , vwfatt.[id_DimAlbero]
        , vwfatt.[id_DimContrattiPrestazione]
    UNION
    SELECT
        vwcosti.[id_DimClienti]
        , vwcosti.[id_DimDataDocumento]
        , vwcosti.[id_DimAlbero]
        , [id_DimContrattiPrestazione] = NULL
    FROM [BINEW_HELPER].[IAM].[CostiCP_TMP] vwcosti
    INNER JOIN BINEW_DWH.dwh.DimProdottoCDG pr ON vwcosti.id_DimProdottoCDG = pr.IdProdottoCDG AND pr
    INNER JOIN BINEW_DWH.dwh.DimSocieta s ON vwcosti.id_DimSocieta = s.id_DimSocieta AND s.bk_Societa = 'I
    INNER JOIN BINEW_DWH.dwh.DimAlbero ALB on vwcosti.id_DimAlbero = ALB.id_DimAlbero AND alb.Divisione I
    WHERE vwcosti.id_DimDataDocumento = @periodo
    GROUP BY
        vwcosti.[id_DimClienti]
        , vwcosti.[id_DimDataDocumento]
        , vwcosti.[id_DimAlbero]
)
SELECT DISTINCT
    [Record Type] = 'D'
    , [Interface File Name] = 'IBLKIF'
    , [Mutation Code] = 'I'
    , [Country Code] = 'IT'
    , [Division Code] = CONVERT(char(15), CONVERT(NVARCHAR(15), @Divisioncode + '_' + S.DESC_SPECIALTY +
    , [Client Number 1] = con.[bk_CustomerNo]
    , [Client Number 2] = CASE
        WHEN alb.bk_Unit IS NOT NULL THEN alb.bk_Unit
        WHEN FORN.Unit IS NOT NULL THEN FORN.Unit
        ELSE isnull(CON.bk_Unit , convert(char(15),'')) END
    , [Client Number 3] = convert(char(15),'')
    , [Client Name] = LEFT(isnull(CLI.RagioneSociale , convert(char(50),'')), 50)
    , [Street Name] = isnull(CLI.Indirizzo , convert(char(50),''))
    , [House Number] = convert(char(15),'')
    , [Postal Code] = LEFT(isnull(CLI.cap , convert(char(10),'')),10)
    , [Municipality] = Isnull(CLI.Citta , convert(char(50),''))
    , [Concern Code] = isnull(CLIAM.CodInternazionale , convert(char(15),''))
    , Filler = ' '

```

```

, [Client Size] = max(convert(char(1),isnull(CON.ContattoNumeroDipendenti ,'')))
, Filler2 = ''
, [Indication International Account] = CASE
    WHEN LEN(CLIAM.CodiceContabileCliente ) > 0 THEN '1'
    ELSE '0' END
, [Search Name] = LEFT(isnull(CLI.RagioneSociale, convert(char(50),'')), 50) -- VERRA' TRONCATO A 50 CARAT
, [NACE Code] = convert(char(6),Replace(isnull(AtecoCodice,0),','))
, Filler3 = ''
, [Client Status] =
    MAX(CASE CON.ClientStatus
        WHEN 'PROSPECT' THEN 1
        WHEN 'MAI' THEN 1
        WHEN 'POTENZIALE' THEN 1
        WHEN 'PERSO' THEN 2
        WHEN 'LATENTE' THEN 3
        WHEN 'ATTIVO' THEN 3
        WHEN 'NUOVO' THEN 3
        WHEN 'SVILUPPATO' THEN 3 -- CODIFICA EFFETTUATA CON GIORGIO SAVINO IL 12.05.2010
        ELSE 0
    END) -- PROBLEMA: I VALORI POSSONO CAMBIARE PERCHE' FANNO PARTE DI
        -- UNA TABELLA DI BASE: PER QUESTO E' STATA EMESSA UNA SEGNALAZIONE
, Filler4 = ''
, [Date First Invoice] = isnull(convert(varchar(10),MIN(fi.id_DimDataDocumento)), '00000000')
, NULL as [Report Record]
FROM BINEW_DWH.dwh.DimContatti CON
LEFT OUTER JOIN BINEW_DWH.dwh.DimClienti CLI on CON.bk_CustomerNo = CLI.bk_CodiceContabile
LEFT OUTER JOIN CDG ON CDG.[id_DimClienti] = CLI.id_DimClienti AND CDG.id_DimDataDocumento = @periodo
LEFT OUTER JOIN docFirstInvoice fi ON fi.[id_DimClienti] = CLI.id_DimClienti
LEFT OUTER JOIN BINEW_DWH.dwh.DimAlbero alb ON CDG.id_DimAlbero = alb.id_DimAlbero
LEFT OUTER JOIN BINEW_DWH.dwh.DimContrattiPrestazione CP on CDG.id_DimContrattiPrestazione = CP.id_Dim
LEFT OUTER JOIN BINEW_DWH.dwh.DimContrattiFornitura FORN ON CP.cf = FORN.bk_ContrattoFornitura
LEFT OUTER JOIN BINEW_STG.stg.tblGruppiClienti CLIAM on CON.bk_CustomerNo = CLIAM.CodiceContabileClier
CROSS APPLY (SELECT DISTINCT DESC_SPECIALTY AS DESC_SPECIALTY FROM [IAM].[udf_SPECIALTY] ()) WHEF
--WHERE
-- NOT CON.ClientStatus IS NULL
-- AND NOT CON.bk_Unit IS NULL
-- AND con.bk_CustomerNo <> '-1'
-- AND NOT IAM.[udf_Get_IDclientiWN](con.[bk_CustomerNo]) IS NULL
GROUP BY
    Replace(isnull(AtecoCodice,0),',')) -- Cambiato da G.Timpanelli 20111121
, CON.bk_CustomerNo
, CASE
    WHEN alb.bk_Unit IS NOT NULL THEN alb.bk_Unit
    WHEN FORN.Unit IS NOT NULL THEN FORN.Unit
    ELSE isnull(CON.bk_Unit , convert(char(15),'')) END
, isnull(CLI.RagioneSociale , convert(char(50),''))
, isnull(CLI.Indirizzo , convert(char(50),''))
, LEFT(isnull(CLI.CAP , convert(char(10),'')),10)
, isnull(CLI.Citta, convert(char(50),''))
, isnull(CLIAM.CodInternazionale , convert(char(15),''))
, CASE
    WHEN LEN(CLIAM.CodiceContabileCliente) > 0 THEN '1'
    ELSE '0' END
, isnull(CLI.RagioneSociale, convert(char(50),''))

```

```
, CONVERT(char(15), CONVERT(NVARCHAR(15), @Divisioncode + '_' + S.DESC_SPECIALTY + '_TEMP'))
;

GO
```

- `miami_ibozif.sql` : logica per `IAM_IBOZIF` (equivalente alla SP `IAM.stp_Update_IBOZIF`).

▼ Schema `IAM_IBOZIF`

column_name	data_type	is_nullable	column_default
Record Type	char	NO	NULL
Interface File Name	char	NO	NULL
Mutation Code	char	NO	NULL
Country Code	char	NO	NULL
Division Code	char	NO	NULL
Timeframe Code	char	NO	NULL
Filler	char	YES	NULL
Accounting Year	char	NO	NULL
Filler2	char	YES	NULL
Accounting Period	char	NO	NULL
Client Number 1	char	NO	NULL
Client Number 2	char	NO	NULL
Client Number 3	char	NO	NULL
Function Code	char	NO	NULL
Employee Number 1	char	NO	NULL
Employee Number 2	char	NO	NULL
Employee Number 3	char	NO	NULL
Filler3	char	YES	NULL
Hours Invoiced	char	YES	NULL
Filler4	char	YES	NULL
Hours Payrolled	char	YES	NULL
Filler5	char	YES	NULL
Number of Timesheets	char	YES	NULL
Total Sales Amount	char	NO	NULL
Sales Amount Hours	char	YES	NULL
Direct Costs	char	YES	NULL
Filler6	char	YES	NULL
Start Date Job Order	char	NO	NULL
Client Rate	char	NO	NULL
Employee Rate	char	NO	NULL
Tariff Coefficient	char	NO	NULL
Report Record	nvarchar	YES	NULL

▼ Codice `stp_Update_IBOZIF`

```
USE [BINEW_HELPER]
GO
```

```

/***** Object: StoredProcedure [IAM].[stp_Update_IBOZIF]  Script Date: 07/07/2025 15:19:05 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE procedure [IAM].[stp_Update_IBOZIF] @Periodo int

as
declare @PeriodoGiorno int = @Periodo *100 + 1;

delete
from IAM.IAM_IBOZIF
where [Accounting Year]= SUBSTRING (convert(char(6),@Periodo),1,4)
and [Accounting Period] = SUBSTRING (convert(char(6),@Periodo),5,2)
;

--20180518 inglobata la table function [IAM].[udf_Dati_IBOZIF]
WITH cteFT as (
    SELECT
        [Accounting Year] = substring (convert(char(8),ft.id_DimDataDocumento),1,4)
        ,[Accounting Period] = substring (convert(char(8),ft.id_DimDataDocumento),5,2)
        ,CON.CodiceLavoratore as CodLavoratore
        ,CLI.bk_CodiceContabilie as [Nr. Cliente]
        ,CONVERT(CHAR(15),ALB.bk_Unit) as [Client Number 2]
        ,CON.bk_CP as [Nr. Contratto Prestazione]
        ,ft.Fatturato as [Importo Totale FP]
        ,[Division Code] = alb.Divisione + '_' + ISNULL(SPE_ADJ.Cod_Specialty_Adj, Spe.DESC_SPECIALTY) + '_TEMP'
        ,ISNULL(caus.bk_GruppoStatistico,-1) AS [New Statistic Group]
        ,ISNULL(CON.MansioneCodice , '') AS Position
        ,REPLACE(ISNULL(CONVERT(VARCHAR(10),CON.DataInizio ,120), '19000101'), '-', '') AS [Start Date Job Order]
        ,forn.TariffaOraOrdinaria as [Client Rate]
    FROM BINEW_DWH.olap.vwFactFattureCDG ft
    INNER JOIN BINEW_DWH.dwh.DimProdottoCDG pr ON ft.id_DimProdottoCDG = pr.IdProdottoCDG
    INNER JOIN BINEW_DWH.dwh.DimSocieta s ON ft.id_DimSocieta = s.id_DimSocieta
    left outer join BINEW_DWH.dwh.DimCausali CAUS on ft.id_DimCausali = CAUS.id_DimCausali
    left outer join BINEW_DWH.dwh.DimAlbero ALB on ft.id_DimAlbero = ALB.id_DimAlbero
    left outer join BINEW_DWH.dwh.DimContrattiPrestazione CON on ft.id_DimContrattiPrestazione = CON.id_DimCc
    left outer join BINEW_DWH.dwh.DimClienti CLI on ft.id_DimClienti = CLI.id_DimClienti
    left outer join BINEW_DWH.dwh.DimContrattiFornitura FORN on CON.CF = FORN.bk_ContrattoFornitura
    CROSS APPLY [IAM].[udf_SPECIALTY] () AS Spe
    LEFT JOIN [IAM].[SPECIALTY_ADJUSTMENTS] AS SPE_ADJ ON SPE.DESC_SPECIALTY = SPE_ADJ.Cod_Special
    WHERE ft.id_DimDataDocumento = @PeriodoGiorno
    AND pr.ProdottoCDG = 'Temps'
    AND alb.Divisione IN ('DIV_RIT', 'DIV_RIH')
    --AND s.bk_Societa IN ('Randstad', 'N/D')
    AND s.bk_Societa = 'Randstad'
    -- G.T. 2019.02.15

```

```

--Condizione inserita per gestire le righe con CodiceLavoratore a NULL.
--i dati arrivano dalla vista olap che include le "martellate" che pososno generare questo tipo di problema.
AND CON.CodiceLavoratore is not null
AND ALB.Divisione = Spe.DivisionCode
AND ISNULL(CON.MansioneCodice , '') = Spe.COD_MANSIONE
)
,cteCosti AS (
SELECT
    [Accounting Year] = substring (convert(char(8),COSTI.id_DimDataDocumento),1,4)
    ,[Accounting Period] = substring (convert(char(8),COSTI.id_DimDataDocumento),5,2)
    ,CON.CodiceLavoratore as CodLavoratore
    ,CLI.bk_CodiceContabilie as [Nr. Cliente]
    ,CONVERT(CHAR(15),ALB.bk_Unit) as [Client Number 2]
    ,CON.bk_CP as [Nr. Contratto Prestazione]
    ,COSTI.[CostoTotale] as [Costo Totale]
    ,[Division Code] = alb.Divisione + '_' + ISNULL(SPE_ADJ.Cod_Specialty_Adj, Spe.DESC_SPECIALTY) + '_TEMP'
    ,ISNULL(CON.MansioneCodice , '') AS Position
    ,con.MatricolaGenerale
    ,REPLACE(ISNULL(CONVERT(VARCHAR(10),CON.DataInizio ,120), '19000101'), '-', '') AS [Start Date Job Order]
    --,ISNULL(CON.DataInizio, '19000101') AS [Start Date Job Order]
    ,FORN.TariffaOraOrdinaria as [Client Rate] --aggiunto da timpa
    --,REPLACE(ISNULL(CONVERT(VARCHAR(10),CON.DataInizio ,120), '19000101'), '-', '') AS [Start Date Job Ord]
    --,forn.TariffaOraOrdinaria as [Client Rate]
FROM [BINEW_DWH].[olap].[vwFactCostoTotaleperCP] COSTI
INNER JOIN BINEW_DWH.dwh.DimProdottoCDG pr ON COSTI.id_DimProdottoCDG = pr.IdProdottoCDG
INNER JOIN BINEW_DWH.dwh.DimSocieta s ON COSTI.id_DimSocieta = s.id_DimSocieta
left outer join BINEW_DWH.dwh.DimAlbero ALB on COSTI.id_DimAlbero = ALB.id_DimAlbero
left outer join BINEW_DWH.dwh.DimContrattiPrestazione CON on COSTI.id_DimContrattiPrestazione = CON.id_C
left outer join BINEW_DWH.dwh.DimClienti CLI on COSTI.id_DimClienti = CLI.id_DimClienti
left outer join BINEW_DWH.dwh.DimContrattiFornitura FORN --rem tolto da timpa
    on CON.CF = FORN.bk_ContrattoFornitura
CROSS APPLY [IAM].[udf_SPECIALTY] () AS Spe
LEFT JOIN [IAM].[SPECIALTY_ADJUSTMENTS] AS SPE_ADJ ON SPE.DESC_SPECIALTY = SPE_ADJ.Cod_Special
WHERE COSTI.id_DimDataDocumento = @PeriodoGiorno
AND pr.ProdottoCDG = 'Temps'
AND alb.Divisione IN ('DIV_RIT', 'DIV_RIH')
--AND s.bk_Societa IN ('Randstad', 'N/D')
AND s.bk_Societa = 'Randstad'
AND ALB.Divisione = Spe.DivisionCode
AND ISNULL(CON.MansioneCodice , '') = Spe.COD_MANSIONE
)
,cteFT2 AS (
SELECT
    [Accounting Year]
    ,[Accounting Period]
    ,CodLavoratore
    ,[Nr. Cliente]
    ,[Client Number 2]
    ,[Nr. Contratto Prestazione]
    ,SUM([Importo Totale FP]) AS [Importo Totale FP]
    ,[Division Code]
    ,SUM(CASE WHEN [New Statistic Group] IN (1, 2, 3, 8, 11) THEN [Importo Totale FP] ELSE 0 END) AS [Somma]
    ,Position
    ,MAX([Start Date Job Order]) AS [Start Date Job Order]

```

```

    ,MAX([Client Rate]) AS [Client Rate]
FROM cteFT
GROUP BY
    [Accounting Year]
    ,[Accounting Period]
    ,CodLavoratore
    ,[Nr. Cliente]
    ,[Client Number 2]
    ,[Nr. Contratto Prestazione]
    ,[Division Code]
    ,Position
)
, cteCosti2 AS (
    SELECT
        [Accounting Year]
        ,[Accounting Period]
        ,CodLavoratore
        ,[Nr. Cliente]
        ,[Client Number 2]
        ,[Nr. Contratto Prestazione]
        ,[Costo Totale] = SUM([Costo Totale])
        ,[Division Code]
        ,Position
        ,MatricolaGenerale
        ,[Start Date Job Order]
        ,[Client Rate]
    FROM cteCosti
    GROUP BY
        [Accounting Year]
        ,[Accounting Period]
        ,CodLavoratore
        ,[Nr. Cliente]
        ,[Client Number 2]
        ,[Nr. Contratto Prestazione]
        ,[Division Code]
        ,Position
        ,MatricolaGenerale
        ,[Start Date Job Order]
        ,[Client Rate]
    )
SELECT
    COALESCE( ft.[Accounting Year],costi.[Accounting Year]) as [Accounting Year],
    COALESCE( ft.[Accounting Period],costi.[Accounting Period]) as [Accounting Period],
    COALESCE( ft.CodLavoratore,costi.CodLavoratore) as CodLavoratore,
    COALESCE( ft.[Nr. Cliente],costi.[Nr. Cliente]) as [Nr. Cliente],
    COALESCE( ft.[Client Number 2],costi.[Client Number 2]) as [Client Number 2],
    COALESCE( ft.[Nr. Contratto Prestazione],costi.[Nr. Contratto Prestazione]) as [Nr. Contratto Prestazione],
    COALESCE( ft.[Division Code],costi.[Division Code]) as [Division Code],
    COALESCE(ft.Position,costi.Position) as Position,
    ISNULL(COALESCE( ft.[Client Rate],costi.[Client Rate]),0) as [Client Rate],--Modificato Timpa
    IsNULL( ft.[Somma FP Per NSG],0) as [Somma FP Per NSG],
    COALESCE( ft.[Start Date Job Order],costi.[Start Date Job Order]) as [Start Date Job Order] ,--Modificato Timpa
    ISNULL(ft.[Importo Totale FP],0) as [Importo Totale FP],
    COSTI.MatricolaGenerale,

```

```

        IsNULL( COSTI.[Costo Totale],0) as [Costo Totale]
INTO #tblAll
FROM cteFT2 FT
FULL OUTER JOIN cteCosti2 COSTI
    on ft.[Accounting Year] = COSTI.[Accounting Year]
    and ft.[Accounting Period] = COSTI.[Accounting Period]
    and ft.CodLavoratore = COSTI.CodLavoratore
    and ft.[Nr. Cliente] = COSTI.[Nr. Cliente]
    and ft.[Client Number 2] = COSTI.[Client Number 2]
    and ft.[Nr. Contratto Prestazione]= COSTI.[Nr. Contratto Prestazione]
    and ft.[Division Code] = COSTI.[Division Code]
    and ft.Position = COSTI.Position

SELECT
    tblall.[Accounting Year]
    ,tblall.[Accounting Period]
    ,tblall.[CodLavoratore]
    ,tblall.[Nr. Cliente]
    ,tblall.[Client Number 2]
    ,tblall.[Nr. Contratto Prestazione]
    ,tblall.[Division Code]
    ,tblall.[Position]
    ,tblall.[Client Rate]
    ,tblall.[Somma FP Per NSG]
    ,tblall.[Start Date Job Order]
    ,tblall.[Importo Totale FP]
    ,tblall.[MatricolaGenerale]
    ,tblall.[Costo Totale]
into #tblAllPulita
FROM #tblAll tblall
INNER JOIN BINEW_DWH.dwh.DimContatti CO on tblall.[Nr. Cliente] = CO.bk_CustomerNo
GROUP BY
    [Accounting Year]
    ,[Accounting Period]
    ,[CodLavoratore]
    ,[Nr. Cliente]
    ,[Client Number 2]
    ,[Nr. Contratto Prestazione]
    ,[Division Code]
    ,[Position]
    ,[Client Rate]
    ,[Somma FP Per NSG]
    ,[Start Date Job Order]
    ,[Importo Totale FP]
    ,[MatricolaGenerale]
    ,[Costo Totale]

SELECT
    [Record Type] = 'D'
    ,[Interface File Name] = 'IBOZIF'
    ,[Mutation Code] = 'I'
    ,[Country Code] = 'IT'
    ,[Division Code]
    ,[Timeframe Code] = 'M'

```



```

,Filler = ' '
,[Accounting Year]
,Filler2 = ' '
,[Accounting Period]
,[Client Number 1] = BINEW_HELPER.IAM.[udf_Get_IDclientiWN]([Nr. Cliente]) -- Non è necessario creare la versione
,[Client Number 2]
,[Client Number 3] = CONVERT(CHAR(15), '')
,[Function Code] = Position
,[Employee Number 1] = CodLavoratore
,[Employee Number 2] = CONVERT(CHAR(15), '')
,[Employee Number 3] = CONVERT(CHAR(15), '')
,Filler3 = ' '
,[Hours Invoiced] = BINEW_HELPER.utl.FormatNumber(coalesce(SUM(FPF.[HHLav]),0),7,2,0)
,Filler4 = ' '
,[Hours Payrolled] = BINEW_HELPER.utl.FormatNumber(coalesce(SUM(FPF.[HHRetr]),0),7,2,0)
,Filler5 = ' '
,[Number of Timesheets] = '0000001'
,[Total Sales Amount] = CONVERT(CHAR(17),BINEW_HELPER.utl.FormatNumber(
    SUM([Importo Totale FP])
    ,13,2,1))
---- verificato con Savino: la formula va bene, i gruppi statistici sono solo questi
,[Sales Amount Hours] = CONVERT(CHAR(17),BINEW_HELPER.utl.FormatNumber(
-- SUM(CASE WHEN [Somma FP Per NSG] > 0 THEN [Somma FP Per NSG] ELSE 0 END)
    SUM([Somma FP Per NSG])
    ,13,2,1))
---- verificata con Giorgio Savino: la formula del campo è giusta
,[Direct Costs] = CONVERT(CHAR(17),BINEW_HELPER.utl.FormatNumber(
    SUM([Costo Totale])
    ,13,2,1))
,Filler6 = ' '
,[Start Date Job Order] = CONVERT(VARCHAR,MIN([Start Date Job Order]))
-- -- Emisfera ci ha messo a disposizione questo campo nel contratto di fornitura
,[Client Rate] = CONVERT(CHAR(9),BINEW_HELPER.utl.FormatNumber([Client Rate],5,2,1)) ----- *****
,[Employee Rate] = CONVERT(CHAR(9),
    BINEW_HELPER.utl.FormatNumber(ISNULL(
        CASE WHEN SUM(FPF.[HHRetr]) <> 0 THEN SUM(FPF.Stipendio) / SUM(FPF.[HHRetr]) ELSE 0 END
        , 0)
    , 5, 2, 1)) --modificato da G.timpanelli 20111121 prima il divisore era SUM([Somma FP Per NSG]) a posto di SUM(f
-- CONVERT(CHAR(9),0)
---- FORMULA CORRETTA RISULTANTE DALL'ULTIMA VERSIONE DELLE SPECIFICHE
,[Tariff Coefficient] = CONVERT(CHAR(9),
    BINEW_HELPER.utl.FormatNumber(ISNULL(
        CASE WHEN SUM(FPF.[HHRetr]) <> 0 THEN SUM([Client Rate]) / (SUM(FPF.Stipendio) / SUM(FPF.[HHRetr]))
        , 0)
    , 5, 2, 1)) --modificato da G.timpanelli 20111121 prima il divisore era SUM([Somma FP Per NSG]) a posto di SUM(f
into #tblFinale
FROM #tblAIIPulita a
LEFT OUTER JOIN BINEW_DWH.dwh.PagheFatture FPF
    ON RIGHT('0000000000' + a.MatricolaGenerale,10) = RIGHT('0000000000' + FPF.Matricola,10)
-- AND a.[Accounting Year] + a.[Accounting Period] + '01' = @periodo
    AND CAST(a.[Accounting Year] AS INT)*10000 + CAST(a.[Accounting Period] AS INT)*100 + 1 = @PeriodoGiorn
    AND FPF.Anno = CAST(a.[Accounting Year] AS INT) AND FPF.Mese = CAST(a.[Accounting Period] AS INT)
--a.[Accounting Period]
--INNER JOIN factPagheFatture FPF

```

```

--ON FPF.[Nr. Contratto Prestazione] = DUE.[Nr. Contratto Prestazione]
--AND FPF.YYYYMM = @periodo
GROUP BY
  [Accounting Year]
,[Accounting Period]
,CodLavoratore
,[Nr. Cliente]
,[Client Number 2]
,[Division Code]
,Position
,[Client Rate]
HAVING [Nr. Cliente] is not null

insert into IAM.IAM_IBOZIF
([Record Type],
[Interface File Name],
[Mutation Code],
[Country Code],
[Division Code],
[Timeframe Code],
Filler,
[Accounting Year],
Filler2,
[Accounting Period],
[Client Number 1],
[Client Number 2],
[Client Number 3],
[Function Code],
[Employee Number 1],
[Employee Number 2],
[Employee Number 3],
Filler3,
[Hours Invoiced],
Filler4,
[Hours Payrolled],
Filler5,
[Number of Timesheets],
[Total Sales Amount],
[Sales Amount Hours],
[Direct Costs],
Filler6,
[Start Date Job Order],
[Client Rate],
[Employee Rate],
[Tariff Coefficient])

select
[Record Type],
[Interface File Name],
[Mutation Code],
[Country Code],
[Division Code],
[Timeframe Code],
Filler,

```

```

[Accounting Year],
Filler2,
[Accounting Period],
[Client Number 1],
[Client Number 2],
[Client Number 3],
[Function Code],
[Employee Number 1],
[Employee Number 2],
[Employee Number 3],
Filler3,
[Hours Invoiced],
Filler4,
[Hours Payrolled],
Filler5,
[Number of Timesheets],
[Total Sales Amount],
[Sales Amount Hours],
[Direct Costs],
Filler6,
[Start Date Job Order],
[Client Rate],
[Employee Rate],
[Tariff Coefficient]
-- from [IAM].[udf_Dati_IBOZIF] (@PeriodoGiorno)
FROM #tblFinale
;

```

```

update IAM.IAM_IBOZIF
set   [Report Record] = [Record Type]
                        + [Interface File Name]
                        + [Mutation Code]
                        + [Country Code]
                        + [Division Code]
                        + [Timeframe Code]
                        + Filler
                        + [Accounting Year]
                        + Filler2
                        + [Accounting Period]
                        + [Client Number 1]
                        + [Client Number 2]
                        + [Client Number 3]
                        + [Function Code]
                        + [Employee Number 1]
                        + [Employee Number 2]
                        + [Employee Number 3]
                        + Filler3
                        + [Hours Invoiced]
                        + Filler4
                        + [Hours Payrolled]
                        + Filler5
                        + [Number of Timesheets]
                        + [Total Sales Amount]

```

```

+ [Sales Amount Hours]
+ [Direct Costs]
+ Filler6
+ [Start Date Job Order]
+ [Client Rate]
+ [Employee Rate]
+ [Tariff Coefficient]

```

```

where [Accounting Year]= SUBSTRING (convert(char(6),@Periodo),1,4)
and [Accounting Period] = SUBSTRING (convert(char(6),@Periodo),5,2)
;
GO

```

▼ Spiegazione logica `stp_Update_IBOZIF`

Cancella il periodo in esame dalla tabella `IAM.IAM_IBOZIF` (oggetto finale).

Parametri e Setup

```

sql
CREATE procedure [IAM].[stp_Update_IBOZIF] @Periodo int
declare @PeriodoGiorno int = @Periodo *100 + 1;

```

- `@Periodo` : Periodo in formato YYYYMM (es. 202501)
- `@PeriodoGiorno` : Converta il periodo nel primo giorno del mese (es. 20250101)

Struttura generale

1. Pulizia dati esistenti

```

sql
delete from IAM.IAM_IBOZIF
where [Accounting Year]= SUBSTRING (convert(char(6),@Periodo),1,4)
and [Accounting Period] = SUBSTRING (convert(char(6),@Periodo),5,2)

```

Rimuove tutti i dati del periodo specificato prima di reinserirli.

2. CTE `cteFT` - Dati Fatturato

```

sql
WITH cteFT as (
    SELECT [Accounting Year], [Accounting Period], CON.CodiceLavoratore,
           CLI.bk_CodiceContabile as [Nr. Cliente], ft.Fatturato as [Importo Totale FP],
           [Division Code] = alb.Divisione + '_' + ISNULL(SPE_ADJ.Cod_Specialty_Adj, Spe.DESC_SPECIALTY) +
'_TEMP'
    FROM BINEW_DWH.olap.vwFactFattureCDG ft
    WHERE ft.id_DimDataDocumento = @PeriodoGiorno
    AND pr.ProdottoCDG = 'Temps'
    AND alb.Divisione IN ('DIV_RIT', 'DIV_RIH')
    AND s.bk_Societa = 'Randstad'
    AND CON.CodiceLavoratore is not null
)

```

Estrae i dati di fatturato per i lavoratori temporanei dalle divisioni Randstad Italia e Randstad in House.

3. CTE **cteCosti** - Dati Costi

```
sql
cteCosti AS (
  SELECT [Accounting Year], [Accounting Period], CON.CodiceLavoratore,
         COSTI.[CostoTotale] as [Costo Totale]
  FROM [BINEW_DWH].[olap].[vwFactCostoTotaleperCP] COSTI
  WHERE COSTI.id_DimDataDocumento = @PeriodoGiorno
  AND pr.ProdottoCDG = 'Temps'
)
```

Estrae i costi sostenuti per gli stessi lavoratori temporanei.

4. CTE di **Aggregazione**

```
sql
cteFT2 AS (
  SELECT [Accounting Year], [Accounting Period], CodLavoratore,
         SUM([Importo Totale FP]) AS [Importo Totale FP],
         SUM(CASE WHEN [New Statistic Group] IN (1, 2, 3, 8, 11) THEN [Importo Totale FP] ELSE 0 END) AS
  [Somma FP Per NSG]
  FROM cteFT
  GROUP BY ...
)
```

Aggrega i dati di fatturato per lavoratore, calcolando anche una somma specifica per determinati gruppi statistici.

5. **Unione Fatturato e Costi**

```
sql
SELECT COALESCE(ft.[Accounting Year],costi.[Accounting Year]) as [Accounting Year],
       ISNULL(ft.[Importo Totale FP],0) as [Importo Totale FP],
       ISNULL(COSTI.[Costo Totale],0) as [Costo Totale]
INTO#tblAll
FROM cteFT2 FT
FULL OUTER JOIN cteCosti2 COSTI
```

Unisce i dati di fatturato e costi usando **FULL OUTER JOIN** per includere tutti i record.

6. **Pulizia finale**

```
sql
SELECT * into#tblAllPulita
FROM#tblAll tblall
INNER JOIN BINEW_DWH.dwh.DimContatti CO on tblall.[Nr. Cliente] = CO.bk_CustomerNo
```

Filtra solo i clienti validi presenti nella dimensione contatti.

7. **Calcolo campi finali**

```
sql
SELECT
  [Record Type] = 'D'
  ,[Interface File Name] = 'IBOZIF'
```

```

,[Mutation Code] = 'I'
,[Hours Invoiced] = BINEW_HELPER.utl.FormatNumber(coalesce(SUM(FPF.[HHLav]),0),7,2,0)
,[Hours Payrolled] = BINEW_HELPER.utl.FormatNumber(coalesce(SUM(FPF.[HHRetr]),0),7,2,0)
,[Employee Rate] = CASE WHEN SUM(FPF.[HHRetr]) <> 0 THEN SUM(FPF.Stipendio) / SUM(FPF.[HHRetr])
ELSE 0 END
,[Tariff Coefficient] = CASE WHEN SUM(FPF.[HHRetr]) <> 0 THEN SUM([Client Rate]) / (SUM(FPF.Stipendio) / SUM(FPF.[HHRetr])) ELSE 0 END
into#tblFinale
FROM#tblAllPulita a
LEFT OUTER JOIN BINEW_DWH.dwh.PagheFatture FPF

```

Calcoli principali:

- **Employee Rate:** Stipendio orario del lavoratore (Stipendio / Ore retribuite)
- **Tariff Coefficient:** Coefficiente tariffario (Tariffa cliente / Tariffa dipendente)
- **Hours Invoiced/Payrolled:** Ore fatturate e retribuite formattate

8. Inserimento finale

```

sql
insert into IAM.IAM_IBOZIF
([Record Type], [Interface File Name], [Mutation Code], ...)
select [Record Type], [Interface File Name], [Mutation Code], ...
FROM#tblFinale

```

9. Aggiornamento Report Record

```

sql
update IAM.IAM_IBOZIF
set [Report Record] = [Record Type] + [Interface File Name] + [Mutation Code] + ...
where [Accounting Year]= SUBSTRING (convert(char(6),@Periodo),1,4)
and [Accounting Period] = SUBSTRING (convert(char(6),@Periodo),5,2)

```

Caratteristiche tecniche

- **Gestione specialità:** Utilizza `SPECIALTY_ADJUSTMENTS` per adattamenti regionali
- **Formattazione:** Usa `utl.FormatNumber` per formati numerici specifici
- **Integrità dati:** Filtra lavoratori con `CodiceLavoratore NOT NULL`

▼ Schema `IAM.SPECIALTY_ADJUSTMENTS`

COLUMN_NAME	DATA_TYPE	IS_NULLABLE	COLUMN_DEFAULT
Cod_Region	nvarchar	NO	NULL
Region	nvarchar	YES	NULL
Cod_Specialty	nvarchar	NO	NULL
Specialty	nvarchar	NO	NULL
Cod_Specialty_Adj	nvarchar	NO	NULL
Specialty_Adj	nvarchar	NO	NULL

▼ Codice funzione `IAM.udf_Get_IDclientiWN`

```

USE [BINEW_HELPER]
GO

```

```

/***** Object: UserDefinedFunction [IAMINT].[udf_Get_IDclientiWN]  Script Date: 10/07/2025 11:03:56 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE FUNCTION [IAMINT].[udf_Get_IDclientiWN]
(
    @BK_CodiceContabile VARCHAR(15)
)
RETURNS char(15)
AS
BEGIN

    DECLARE @IdCliente char(15)

    SET @IdCliente = (
        select distinct [bk_CodiceContabile]--IDclientiWN
        from
            [BINEW_DWH].[dwh].[DimClienti]
        where
            [bk_CodiceContabile]= @BK_CodiceContabile
    )

    RETURN @IdCliente
END

GO

```

▼ Codice funzione `utl.FormatNumberFormatNumber`

```

USE [BINEW_HELPER]
GO

/***** Object: UserDefinedFunction [utl].[FormatNumber]  Script Date: 10/07/2025 11:05:26 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE FUNCTION [utl].[FormatNumber]
(
    @Valore DECIMAL(38,10),
    @Cifre INT,
    @Decimali INT,
    @Segno INT
)
RETURNS VARCHAR(100)
AS

```

```

BEGIN

DECLARE @Minus INT
DECLARE @Result VARCHAR(100)

IF @Valore < 0 BEGIN
    SET @Minus = 1
    SET @Valore = @Valore *- 1
END ELSE BEGIN
    SET @Minus = 0
END

SET @Result = CAST(
    CAST(@Valore AS DECIMAL(38, 2))
    AS VARCHAR(100)
)

SET @Result = REPLACE(@Result, ',', '')

SET @Result = RIGHT('0000000000000000000000000000000000000000000000000000000' + @Result, @Cifre +

IF @Segno > 0 BEGIN
    IF @Minus = 1 BEGIN
        SET @Result = '-' + @Result
    END ELSE BEGIN
        SET @Result = '' + @Result
    END
END

RETURN @Result

END

GO

```

Step 3

I dati raccolti ed elaborati potranno essere visibili da Talend attraverso una view o una table. Talend otterrà i dati e costruirà il file finale da consegnare.

Nel progetto originale i dati da scrivere nel file vengono raccolti dalla SP `IAM.GetDataIAMTable` che richiama la funzione `IAM.udf_IAMTable` la quale prende un parametro `@Periodo` (formato `YYYYMM`) e restituisce una tabella con una colonna `[Report Record]`. Più nello specifico:

▼ Codice `IAM.udf_IAMTable`

```

USE [BINEW_HELPER]
GO

/***** Object: UserDefinedFunction [IAM].[udf_IAMTable]  Script Date: 08/07/2025 10:17:42 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

```



```

CREATE Function [IAM].[udf_IAMTable] (@Periodo char(6))
returns table
as
return

with cteA
as
(

    select [Report Record] from IAM.IAM_IBDIIF

    union all
    select [Report Record] from IAM.IAM_IBCNIF
    where [Concern Code] in (SELECT distinct [Concern Code] FROM [IAM].[IAM_IBLKIF])

    union all
    --select REPLACE([Report Record], '$', '*') from IAM.IAM_IBLKIF
    select [Report Record] from IAM.IAM_IBLKIF
    where [Client Size] <> 'N'

    --NL 2024.11.18 Esclusi record con anomalie nella colonna [Report Record] - Presenti righe omologhe senza anomalie
    AND NOT([Report Record] like '%'+ char(13)+'%' OR [Report Record] like '%'+ char(10)+'%')
    AND [Client Number 1] != 'C220202 '

    union all
    select [Report Record] from IAM.IAM_IBFUIF

    union all
    -- select REPLACE([Report Record], '$', '*') from IAM.IAM_IBOZIF where [Accounting Year] =substring(@Periodo,1,4)
    select [Report Record] from IAM.IAM_IBOZIF where [Accounting Year] =substring(@Periodo,1,4) and [Accounting Period] =substring(@Periodo,5,4)
),

cteA1
as
(
    select * from cteA
    where not [Report Record] is null -- 2015.09.11 FP per evitare di avere record null (possono essere dovuti all'assenza di dati)
)

select
convert (nvarchar(512),'H' +
(select CONVERT (char(8), getdate(),112)) +
(select substring(CONVERT (char(8), getdate(),114),1,2) + substring(CONVERT (char(8), getdate(),114),4,2)) +
replace(str((select MAX (IdFile)from IAM.IAM_LOG),8,0),' ','0') +
'WHN 2.2IT') as [Report Record]

union all

-- NL 20241119 replace tab in string
select REPLACE([Report Record], char(9), '') from cteA1

--union all
--select [Report Record] from IAM.IAM_IBDIIF

```

```

--union all
--select [Report Record] from IAM.IAM_IBCNIF

--union all
--select [Report Record] from IAM.IAM_IBLKIF

--union all
--select [Report Record] from IAM.IAM_IBFUIF

--union all
--select [Report Record] from IAM.IAM_IBOZIF where [Accounting Year] =substring(@Periodo,1,4) and [Accounting Pe

union all
select 'F' + right('00000000' + convert(varchar(8),COUNT(*)),8) as [Report Record] from cteA1

;

GO

```

▼ Spiegazione logica `IAM.udf_IAMTable`

1. La funzione prende in input il parametro `@Periodo` che rappresenta il periodo nel formato `YYYYMM`. Restituisce una tabella con una singola colonna `[Report Record]`.
2. Definisce due CTE.:
 - a. La cteA raccoglie i dati da diverse tabelle (`IAM.IAM_IBDIIF`, `IAM.IAM_IBCNIF`, `IAM.IAM_IBLKIF`, `IAM.IAM_IBFUIF`, `IAM.IAM_IBOZIF`). Nella lettura di queste tabelle, applica i seguenti filtri:
 - i. `IAM.IAM_IBCNIF`: Include solo i record con *Concern Code* presente nella tabella `IAM.IAM_IBLKIF`.
 - ii. `IAM.IAM_IBLKIF`: Esclude i record con Client Size uguale a 'N' e quelli con anomalie nei campi `[Report Record]` (caratteri di controllo come `char(13)` o `char(10)`).
 - iii. `IAM.IAM_IBOZIF`: Filtra i record in base all'anno contabile e al periodo contabile derivati dal parametro `@Periodo`.
 - b. La seconda CTE, cteA1, seleziona solo i record non nulli dalla CTE cteA.
3. La select finale della funzione `[IAM].[udf_IAMTable]` esegue diverse operazioni in modo sequenziale e dettagliato:
 - a. Creazione di un Record di Intestazione: La funzione inizia creando un record di intestazione che include vari elementi concatenati insieme. Questo record inizia con la lettera 'H', seguita dalla data corrente nel formato `YYYYMMDD`. Successivamente, viene aggiunta l'ora corrente, estratta e formattata per includere solo le ore e i minuti. Inoltre, viene incluso l'ID massimo dalla tabella `IAM.IAM_LOG`, formattato come una stringa di otto caratteri con zeri iniziali. Infine, viene aggiunta una stringa fissa 'WHN 2.2IT'.
 - b. Selezione dei Record dalla CTE cteA1: La funzione procede selezionando i record dalla CTE cteA1. Durante questa selezione, ogni record viene pulito sostituendo i caratteri di tabulazione con stringhe vuote. Questo passaggio garantisce che i dati siano privi di caratteri di controllo che potrebbero causare problemi durante l'elaborazione successiva.
 - c. Aggiunta di un Record Finale: Infine, la funzione aggiunge un record che contiene il conteggio totale dei record presenti nella CTE cteA1. Questo conteggio è formattato come una stringa di otto caratteri con zeri iniziali, preceduta dalla lettera 'F'. Questo passaggio fornisce un riepilogo del numero totale di record elaborati, utile per verifiche e controlli.