

Índice de contenidos

Índice de contenidos	1
Estudio de la tabla FACTURAS y CONTACTOS	2
FACTURAS	2
CONTACTOS	3
Diagrama ODS	4
Diagrama con número de registros que contiene cada tabla	4
Modelo de DIRECCIONES. ¿CIUDADES y ESTADOS?	6
Separar el campo DE_DIRECCION en NOMBRE_VIA y NUM_VIA	6
Data Management	10
Mejoras sobre el ecosistema DWH	10
Reglas o Mandamientos de un DWH	11
Anexo. Scripts	12
Estudio facturas	12
Estudio Contactos	13
Modelado Facturas	14
Modelado Llamadas	16
Inserción Datos Facturas	18
Inserción Datos Llamadas	20
Separación del campo DE_DIRECCION a NUM_VIA e ID_NOMBRE_VIA	22

1. Estudio de la tabla FACTURAS y CONTACTOS

1.1. FACTURAS

TOTAL_REGISTROS:	420000
TOTAL_BILL_REF_NO:	420000
TOTAL_DISTINTOS_BILL_REF_NO:	420000
TOTAL_CUSTOMER_ID:	420000
TOTAL_DISTINTOS_CUSTOMER_ID:	20000
TOTAL_START_DATE:	420000
TOTAL_DISTINTOS_START_DATE:	40
TOTAL_END_DATE:	420000
TOTAL_DISTINTOS_END_DATE:	20
TOTAL_STATEMENT_DATE:	420000
TOTAL_DISTINTOS_STATEMENT_DATE:	40
TOTAL_PAYMENT_DATE:	420000
TOTAL_DISTINTOS_PAYMENT_DATE:	400
TOTAL_BILL_CYCLE:	420000
TOTAL_DISTINTOS_BILL_CYCLE:	2
TOTAL_AMOUNT:	420000
TOTAL_DISTINTOS_AMOUNT:	5604
TOTAL_BILL_METHOD:	420000
TOTAL_DISTINTOS_BILL_METHOD:	3

- Se ha optado por sacar dos dimensiones: ODS_DM_CICLO_FACTURACION y ODS_DM_METODO_PAGO, por sus estadísticas y las veces que se repetían en la tabla.
- Se ha creado una FK con la tabla de hechos de clientes desde el campo CUSTOMER_ID.
- Para las fechas, aunque se ha barajado la posibilidad de sacar dimensiones, creo que no aporta mayores beneficios el hecho de hacerlo. Sacar una dimensión de fechas, que potencialmente puede crecer muchísimo, y además puede ocupar más espacio si le adicionamos los campos de control hace que no merezca la pena, además de la penalización de rendimiento de tener que cruzar 4 campos por registro. Eso sí, se han convertido de STRING a DATETIME.
- Para el campo AMOUNT, aunque parece que es un valor que también se repite con cierta frecuencia, también se ha optado por dejarlo en la propia tabla, puesto que no parece que tenga sentido que este valor esté en otra tabla. Se ha convertido a DECIMAL(11,2).
- Quedarían pendientes la construcción de algún mecanismo de control o trigger para permitir la inserción en las nuevas dimensiones creadas para salvaguardar los IDs para los valores DESCONOCIDO y NO APLICA.

1.2. CONTACTOS

TOTAL_REGISTROS:	202717
TOTAL_ID:	202717
TOTAL_DISTINTOS_ID:	150000
TOTAL_PHONE_NUMBER:	185018
TOTAL_DISTINTOS_PHONE_NUMBER:	18226
TOTAL_START_DATETIME:	202717
TOTAL_DISTINTOS_START_DATETIME:	201098
TOTAL_END_DATETIME:	186535
TOTAL_DISTINTOS_END_DATETIME:	183678
TOTAL_SERVICE:	202502
TOTAL_DISTINTOS_SERVICE:	7
TOTAL_FLG_TRANSFER:	202717
TOTAL_DISTINTOS_FLG_TRANSFER:	2
TOTAL_AGENT:	194739
TOTAL_DISTINTOS_AGENT:	594

- Se ha optado por sacar dos dimensiones: ODS_DM_AGENTES y ODS_DM_SERVICIOS, por la cantidad de veces que se repetían sus valores.
- Se barajó la posibilidad de sacar una dimensión de teléfonos, y enlazar esta dimensión con la tabla de hechos de CLIENTES además de con la de CONTACTOS. Pero dado su carácter numérico y a pesar de que los valores de teléfono no cruzaban con la tabla de CLIENTES, se han creado dos índices, uno para cada tabla, en previsión de futuras cargas de datos pudiesen cruzar. (Sería conveniente indagar con negocio el porqué no cruzan estos datos). Como reflexión, es posible que en un futuro haya que cambiar la tipología de este campo, por si hubiese que guardar prefijos internacionales o extensiones, y es posible que de ahí salga alguna dimensión.
- Otro campo digno de mención es el FLG_TRANSFER. Este campo es claramente de tipo BOOLEAN, y por ello no merece la pena sacar dimensión. Queda aportando valor dentro de la misma tabla.
- En esta tabla me he encontrado con la problemática de que lo que parecía ser su clave, no era única. Por tanto, se ha tenido que generar un nuevo campo único y hacerlo clave. Se ha dejado una copia de esta "ID" en un campo llamado STG_ID_CONTACTO, por si en un futuro fuese relevante o hiciese falta cruzarlo con algún otro dato que a día de hoy no se tenga pero que pudiese ser importante y poder generar algún tipo de histórico.

1.3. Modelo, tablas, FK, y población de datos

Adjuntado en los scripts. (Ver anexo o ficheros adjuntos)

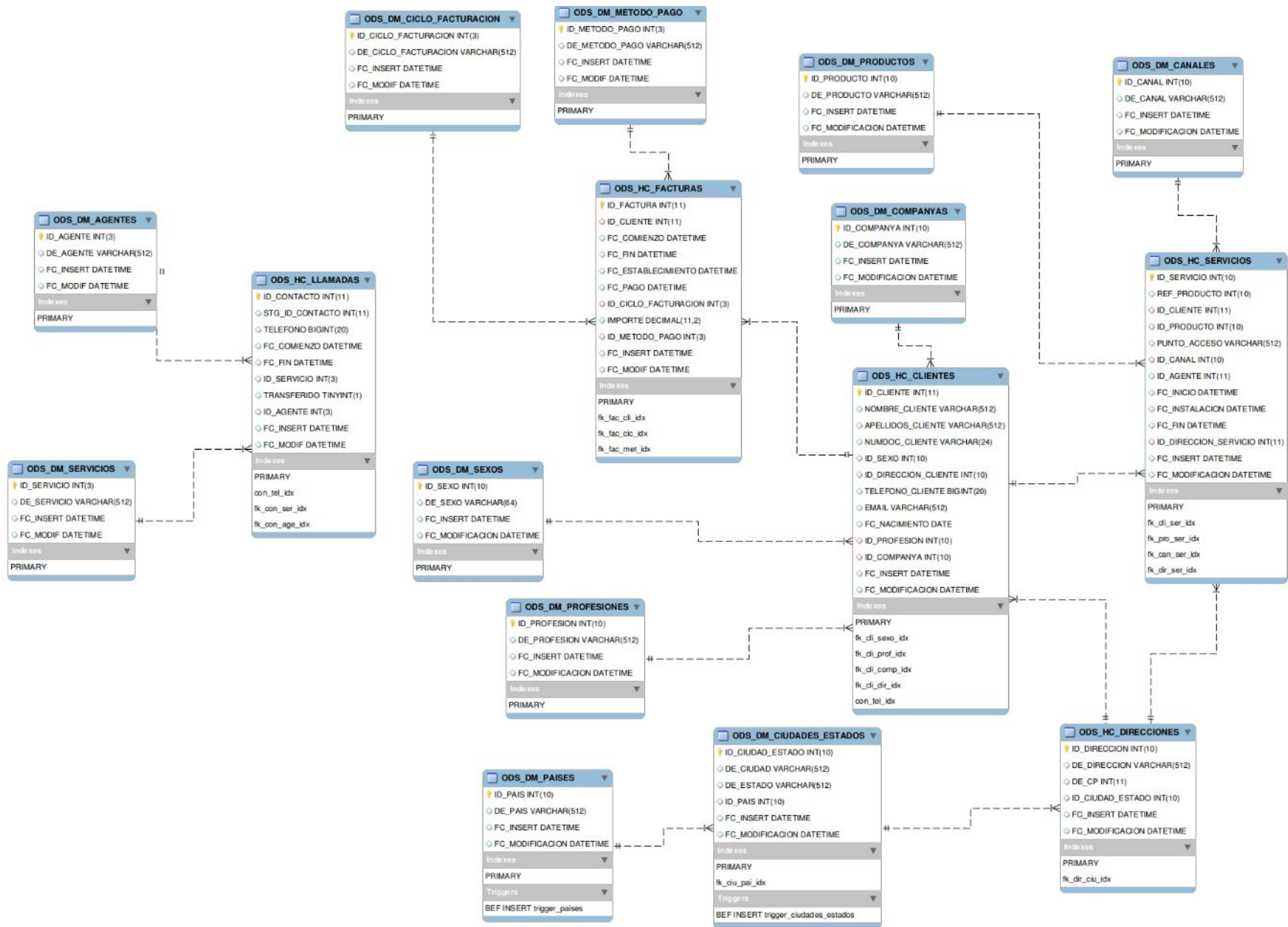
2. Diagrama ODS

2.1. Diagrama con número de registros que contiene cada tabla

COUNTS:

- SELECT COUNT(*) FROM ODS_DM_AGENTES; // 595
- SELECT COUNT(*) FROM ODS_DM_CANALES; // 6
- SELECT COUNT(*) FROM ODS_DM_CICLO_FACTURACION; // 4
- SELECT COUNT(*) FROM ODS_DM_CIUDADES_ESTADOS; // 159
- SELECT COUNT(*) FROM ODS_DM_COMPANYAS; // 385
- SELECT COUNT(*) FROM ODS_DM_METODO_PAGO; // 5
- SELECT COUNT(*) FROM ODS_DM_NOMBRE_VIA; // 10.502
- SELECT COUNT(*) FROM ODS_DM_PAISES; // 3
- SELECT COUNT(*) FROM ODS_DM_PRODUCTOS; // 8
- SELECT COUNT(*) FROM ODS_DM_PROFESIONES; // 197
- SELECT COUNT(*) FROM ODS_DM_SERVICIOS; // 8
- SELECT COUNT(*) FROM ODS_DM_SEXOS; // 4
- SELECT COUNT(*) FROM ODS_HC_CLIENTES; // 17.558
- SELECT COUNT(*) FROM ODS_HC_DIRECCIONES; // 113.260
- SELECT COUNT(*) FROM ODS_HC_FACTURAS; // 420.000
- SELECT COUNT(*) FROM ODS_HC_LLAMADAS; // 202.717
- SELECT COUNT(*) FROM ODS_HC_SERVICIOS; // 78.467

Práctica: mod data 101 - Taller DWH



2.2. Modelo de DIRECCIONES. ¿CIUDADES y ESTADOS?

El motivo de dejar en la misma tabla las CIUDADES y los ESTADOS es por la relación N:M que existe entre ambas entidades, ya que una ciudad puede pertenecer a uno o varios estados, y un estado puede contener una o varias ciudades, por tanto, la unicidad del dato, lo da el par ciudad - estado. También podría haber sido una solución válida, haber creado una tabla intermedia con todas las combinaciones posibles de ciudades - estado y cruzarlo con el par concreto que correspondiese en cada dirección.

2.3. Separar el campo DE_DIRECCION en NOMBRE_VIA y NUM_VIA

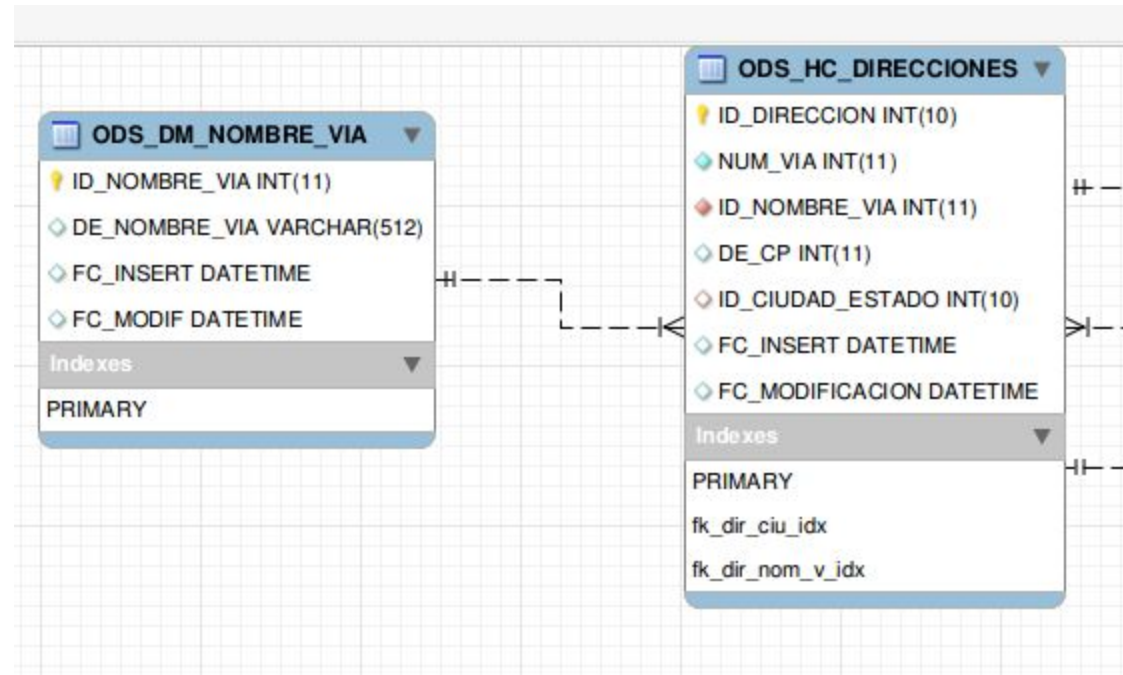
```

26 -- ESTUDIO NOMBRE VIAS
27 • SELECT COUNT(SUBSTR(OHD.DE_DIRECCION, LOCATE(' ', OH.DE_DIRECCION) + 1)),
28       COUNT(DISTINCT(SUBSTR(OHD.DE_DIRECCION, LOCATE(' ', OH.DE_DIRECCION) + 1)))
29       FROM ODS.ODS HC DIRECCIONES OH
30       WHERE SUBSTR(OHD.DE_DIRECCION, LOCATE(' ', OH.DE_DIRECCION) + 1) <> ' ';
31

```

#	COUNT(SUBSTR(OHD.DE_DIRECCION, LOCATE(' ', OH.DE_DIRECCION) + 1))	COUNT(DISTINCT(SUBSTR(OHD.DE_DIRECCION, LOCATE(' ', OH.DE_DIRECCION) + 1)))
1	113262	10502

Resultado del modelo con la nueva tabla:



Durante la realización de este ejercicio me he dado cuenta de que en la tabla ODS_HC_DIRECCIONES están duplicados los registros que representan DESCONOCIDO y NO APLICA:

4 • `SELECT * FROM ODS_HC_DIRECCIONES WHERE DE_CP=99999;`
5

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:

#	ID_DIRECCION	NUM_VIA	ID_NOMBRE_VIA	DE_CP	ID_CIUDAD_ESTADO	FC_INSERT	FC_MODIFICACION
1	999998	999998	999999998	99999	999	2018-12-22 12:13:44	2018-12-23 13:39:16
2	999999	999999	999999999	99999	999	2018-12-22 12:13:44	2018-12-23 13:39:16
3	1095761	999999	999999999	99999	999	2018-12-22 12:43:54	2018-12-23 13:39:16
4	1095762	999998	999999998	99999	999	2018-12-22 12:43:54	2018-12-23 13:39:16
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Tras comprobar que ambos valores no están siendo referenciados (con el fin de reajustarlos), se ha procedido a su eliminación:

Práctica: mod data 101 - Taller DWH

```

1 • USE ODS;
2
3 • SELECT * FROM ODS_HC_CLIENTES OHC WHERE OHC.ID_DIRECCION_CLIENTE=1095762 OR OHC.ID_DIRECCION_CLIENTE=1095761;
4 • SELECT * FROM ODS_HC_SERVICIOS OHC WHERE OHC.ID_DIRECCION_SERVICIO=1095762 OR OHC.ID_DIRECCION_SERVICIO=1095761;
5
6 • SELECT * FROM ODS_HC_DIRECCIONES WHERE DE_CP=99999;
7

```

Result Grid							
Filter Rows: <input type="text"/>							
Edit: Export/Import: Wrap Cell Content:							
#	ID_DIRECCION	NUM_VIA	ID_NOMBRE_VIA	DE_CP	ID_CIUADAD_ESTADO	FC_INSERT	FC_MODIFICACION
1	999998	999998	999999999	99999	999	2018-12-22 12:13:44	2018-12-23 13:39:16
2	999999	999999	999999999	99999	999	2018-12-22 12:13:44	2018-12-23 13:39:16
3	1095761	999999	999999999	99999	999	2018-12-22 12:43:54	2018-12-23 13:39:16
4	1095762	999998	999999998	99999	999	2018-12-22 12:43:54	2018-12-23 13:39:16
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

ODS_HC_CLIENTES 37 ✕	ODS_HC_SERVICIOS 38 ✕	ODS_HC_DIRECCIONES 39 ✕
----------------------	-----------------------	-------------------------

Action Output			
		Time	Action
✓	1	16:57:06	USE ODS
✓	2	16:57:06	SELECT * FROM ODS_HC_CLIENTES OHC WHERE OHC.ID_DIRECCION_CLIENTE=1095762 OR OHC.ID_DIRECCION_CLIENTE=1095761 LIMIT 0, 1000
✓	3	16:57:06	SELECT * FROM ODS_HC_SERVICIOS OHC WHERE OHC.ID_DIRECCION_SERVICIO=1095762 OR OHC.ID_DIRECCION_SERVICIO=1095761 LIMIT 0, 1000
✓	4	16:57:06	SELECT * FROM ODS_HC_DIRECCIONES WHERE DE_CP=99999 LIMIT 0, 1000

```

3 • DELETE FROM ODS_HC_DIRECCIONES WHERE ID_DIRECCION=1095761;
4 • DELETE FROM ODS_HC_DIRECCIONES WHERE ID_DIRECCION=1095762;
5
6 • SELECT * FROM ODS_HC_DIRECCIONES WHERE DE_CP=99999;

```

Result Grid: Filter Rows: Edit: Export/Import: Wrap Cell Content:							
#	ID_DIRECCION	NUM_VIA	ID_NOMBRE_VIA	DE_CP	ID_CIUDAD_ESTADO	FC_INSERT	FC_MODIFICACION
1	999998	999998	999999998	99999	999	2018-12-22 12:13:44	2018-12-23 13:39:16
2	999999	999999	999999999	99999	999	2018-12-22 12:13:44	2018-12-23 13:39:16
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

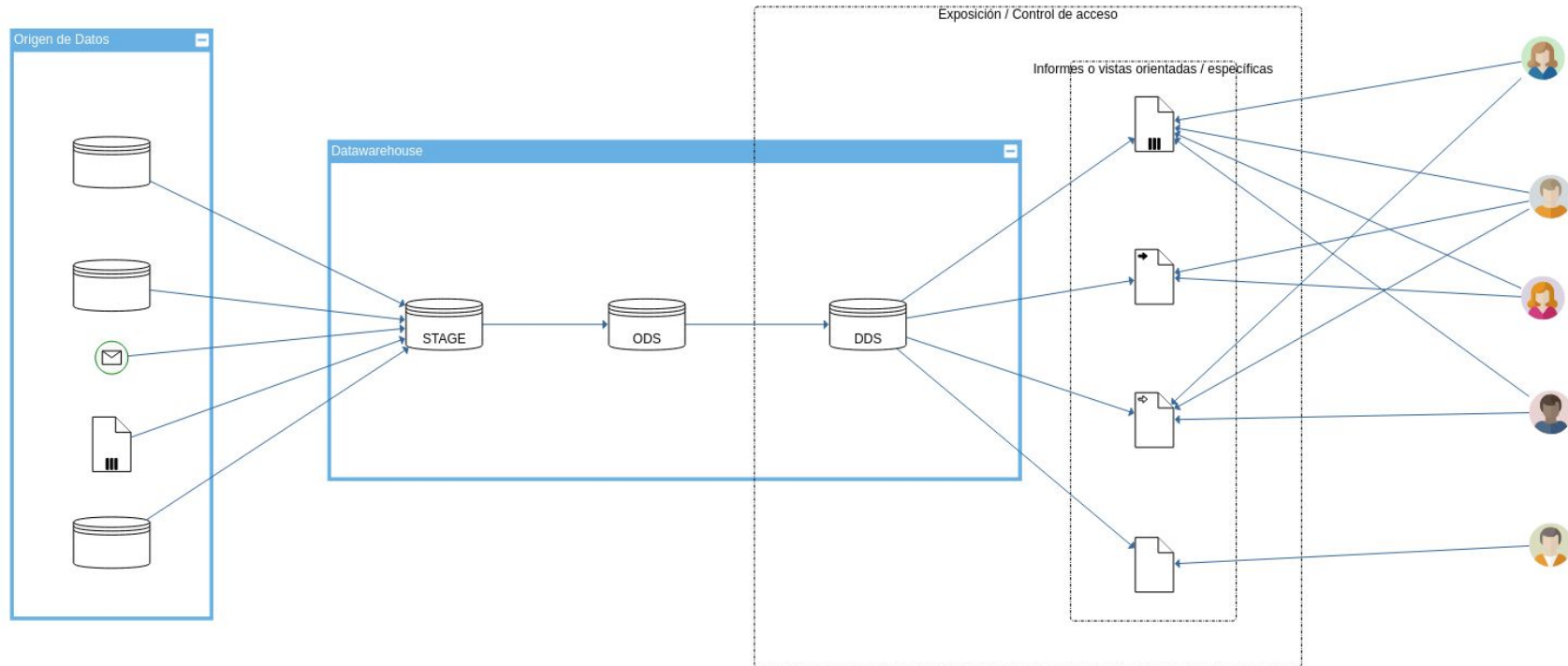
3. Data Management

99999998 - NO APLICA

4. Mejoras sobre el ecosistema DWH

La arquitectura planteada para el DWH es correcta, pero sólo es el punto de partida. Con el ecosistema actual, tenemos una réplica de los datos operacionales que han podido venir de unidades de negocio distintas, sistemas muy heterogéneos donde el dato podía o no estar estructurado. Lo que conseguimos con el diseño actual es tener un dato estructurado y dotado de consistencia. Es a partir de este esquema, el cual establece la base, donde se tiene que empezar a trabajar para transformar y exponer estos datos orientados al rol y/o unidad de negocio que requiera sacar algún tipo de analítica / conclusión.

Por tanto, parece lógico pensar en extender este modelo en distintas “proyecciones” donde cada rol y/o unidad de negocio necesita mirar a través de un prisma distinto. Es en esta capa de exposición / proyección donde permitiría / controlaría el acceso a los datos mediante las oportunas herramientas de visualizado de los datos o generación de informes.



5. Reglas o Mandamientos de un DWH

- Construir un ODS sólido, estructurado y consistente. Será la pieza clave donde partirán todas las analíticas que se lleven a cabo.
 - Evitar valores nulos.
 - Normalización de los datos (estructura y contenido).
 - Estudio de la cantidad de campos desconocidos y comunicar / colaborar con los responsables del operacional que genera estos datos para plantear posibles soluciones.
 - Concienciar de la importancia de guardar la zona horaria en los campos fecha, sobre todo en proyectos multinacionales.
- Puesto que el Datawarehouse no es estático, automatizaría los procesos de carga de datos del operacional (preferiblemente de forma incremental).
 - Valorar la posibilidad de enviar los datos del operacional mediante operaciones asíncronas (colas de mensajería) o tomar los datos de algún nodo secundario de sólo lectura que contenga copia exacta del nodo principal en todo momento.
- Revisión e iteración continua sobre el modelado de datos. Los distintos modelos del operacional pueden crecer y se deberá valorar la inclusión de estos nuevos campos.
- Considerar en función del tipo de dato, la anonimización de los mismos desde su inserción en el STAGE.
- No perder el foco pensando en tablas y columnas. Pensar en los procesos de negocio que queremos conocer / detallar / descubrir.

6. Anexo. Scripts

6.1. Estudio facturas

```
USE STAGE;  
  
SELECT COUNT(*) TOTAL_REGISTROS  
, SUM(CASE WHEN LENGTH(TRIM(BILL_REF_NO))<>0 THEN 1 ELSE 0 END) TOTAL_BILL_REF_NO  
, COUNT(DISTINCT CASE WHEN LENGTH(TRIM(BILL_REF_NO))<>0 THEN BILL_REF_NO ELSE 0 END)
```

```
TOTAL_DISTINTOS_BILL_REF_NO
, SUM(CASE WHEN LENGTH(TRIM(CUSTOMER_ID))<>0 THEN 1 ELSE 0 END) TOTAL_CUSTOMER_ID
, COUNT(DISTINCT CASE WHEN LENGTH(TRIM(CUSTOMER_ID))<>0 THEN CUSTOMER_ID ELSE 0 END)
TOTAL_DISTINTOS_CUSTOMER_ID
, SUM(CASE WHEN LENGTH(TRIM(START_DATE))<>0 THEN 1 ELSE 0 END) TOTAL_START_DATE
, COUNT(DISTINCT CASE WHEN LENGTH(TRIM(START_DATE))<>0 THEN START_DATE ELSE 0 END)
TOTAL_DISTINTOS_START_DATE
, SUM(CASE WHEN LENGTH(TRIM(END_DATE))<>0 THEN 1 ELSE 0 END) TOTAL_END_DATE
, COUNT(DISTINCT CASE WHEN LENGTH(TRIM(END_DATE))<>0 THEN END_DATE ELSE 0 END) TOTAL_DISTINTOS_END_DATE
, SUM(CASE WHEN LENGTH(TRIM(STATEMENT_DATE))<>0 THEN 1 ELSE 0 END) TOTAL_STATEMENT_DATE
, COUNT(DISTINCT CASE WHEN LENGTH(TRIM(STATEMENT_DATE))<>0 THEN STATEMENT_DATE ELSE 0 END)
TOTAL_DISTINTOS_STATEMENT_DATE
, SUM(CASE WHEN LENGTH(TRIM(PAYMENT_DATE))<>0 THEN 1 ELSE 0 END) TOTAL_PAYMENT_DATE
, COUNT(DISTINCT CASE WHEN LENGTH(TRIM(PAYMENT_DATE))<>0 THEN PAYMENT_DATE ELSE 0 END)
TOTAL_DISTINTOS_PAYMENT_DATE
, SUM(CASE WHEN LENGTH(TRIM(BILL_CYCLE))<>0 THEN 1 ELSE 0 END) TOTAL_BILL_CYCLE
, COUNT(DISTINCT CASE WHEN LENGTH(TRIM(BILL_CYCLE))<>0 THEN BILL_CYCLE ELSE 0 END)
TOTAL_DISTINTOS_BILL_CYCLE
, SUM(CASE WHEN LENGTH(TRIM(AMOUNT))<>0 THEN 1 ELSE 0 END) TOTAL_AMOUNT
, COUNT(DISTINCT CASE WHEN LENGTH(TRIM(AMOUNT))<>0 THEN AMOUNT ELSE 0 END) TOTAL_DISTINTOS_AMOUNT
, SUM(CASE WHEN LENGTH(TRIM(BILL_METHOD))<>0 THEN 1 ELSE 0 END) TOTAL_BILL_METHOD
, COUNT(DISTINCT CASE WHEN LENGTH(TRIM(BILL_METHOD))<>0 THEN BILL_METHOD ELSE 0 END)
TOTAL_DISTINTOS_BILL_METHOD
FROM STAGE.STG_FACTURAS_FCT;
```

6.2. Estudio Contactos

```
USE STAGE;

SELECT COUNT(*) TOTAL_REGISTROS
, SUM(CASE WHEN LENGTH(TRIM(ID))<>0 THEN 1 ELSE 0 END) TOTAL_ID
, COUNT(DISTINCT CASE WHEN LENGTH(TRIM(ID))<>0 THEN ID ELSE 0 END) TOTAL_DISTINTOS_ID
, SUM(CASE WHEN LENGTH(TRIM(PHONE_NUMBER))<>0 THEN 1 ELSE 0 END) TOTAL_PHONE_NUMBER
, COUNT(DISTINCT CASE WHEN LENGTH(TRIM(PHONE_NUMBER))<>0 THEN PHONE_NUMBER ELSE 0 END)
TOTAL_DISTINTOS_PHONE_NUMBER
, SUM(CASE WHEN LENGTH(TRIM(START_DATETIME))<>0 THEN 1 ELSE 0 END) TOTAL_START_DATETIME
, COUNT(DISTINCT CASE WHEN LENGTH(TRIM(START_DATETIME))<>0 THEN START_DATETIME ELSE 0 END)
TOTAL_DISTINTOS_START_DATETIME
, SUM(CASE WHEN LENGTH(TRIM(END_DATETIME))<>0 THEN 1 ELSE 0 END) TOTAL_END_DATETIME
, COUNT(DISTINCT CASE WHEN LENGTH(TRIM(END_DATETIME))<>0 THEN END_DATETIME ELSE 0 END)
TOTAL_DISTINTOS_END_DATETIME
, SUM(CASE WHEN LENGTH(TRIM(SERVICE))<>0 THEN 1 ELSE 0 END) TOTAL_SERVICE
, COUNT(DISTINCT CASE WHEN LENGTH(TRIM(SERVICE))<>0 THEN SERVICE ELSE 0 END) TOTAL_DISTINTOS_SERVICE
, SUM(CASE WHEN LENGTH(TRIM(FLG_TRANSFER))<>0 THEN 1 ELSE 0 END) TOTAL_FLG_TRANSFER
, COUNT(DISTINCT CASE WHEN LENGTH(TRIM(FLG_TRANSFER))<>0 THEN FLG_TRANSFER ELSE 0 END)
TOTAL_DISTINTOS_FLG_TRANSFER
, SUM(CASE WHEN LENGTH(TRIM(AGENT))<>0 THEN 1 ELSE 0 END) TOTAL_AGENT
, COUNT(DISTINCT CASE WHEN LENGTH(TRIM(AGENT))<>0 THEN AGENT ELSE 0 END) TOTAL_DISTINTOS_AGENT
FROM STAGE.STG_CONTACTOS_IVR;
```

6.3. Modelado Facturas

```
USE ODS;
```

```
DROP TABLE IF EXISTS ODS_HC_FACTURAS;
CREATE TABLE ODS_HC_FACTURAS (
    ID_FACTURA INT(11) NOT NULL PRIMARY KEY,
    ID_CLIENTE INT(11),
    FC_COMIENZO DATETIME,
    FC_FIN DATETIME,
    FC_ESTABLECIMIENTO DATETIME,
    FC_PAGO DATETIME,
    ID_CICLO_FACTURACION INT(3) UNSIGNED,
    IMPORTE DECIMAL(11,2),
    ID_METODO_PAGO INT(3) UNSIGNED,
    FC_INSERT DATETIME,
    FC_MODIF DATETIME
);

DROP TABLE IF EXISTS ODS_DM_CICLO_FACTURACION;
CREATE TABLE ODS_DM_CICLO_FACTURACION (
    ID_CICLO_FACTURACION INT(3) UNSIGNED AUTO_INCREMENT NOT NULL PRIMARY KEY,
    DE_CICLO_FACTURACION VARCHAR(512),
    FC_INSERT DATETIME,
    FC_MODIF DATETIME
);

DROP TABLE IF EXISTS ODS_DM_METODO_PAGO;
CREATE TABLE ODS_DM_METODO_PAGO (
    ID_METODO_PAGO INT(3) UNSIGNED AUTO_INCREMENT NOT NULL PRIMARY KEY,
    DE_METODO_PAGO VARCHAR(512),
    FC_INSERT DATETIME,
    FC_MODIF DATETIME
);

ALTER TABLE ODS_HC_FACTURAS ADD INDEX fk_fac_cli_idx (ID_CLIENTE ASC);
```



```
ALTER TABLE ODS_HC_FACTURAS ADD CONSTRAINT fk_fac_cli FOREIGN KEY (ID_CLIENTE)
REFERENCES ODS_HC_CLIENTES(ID_CLIENTE);

ALTER TABLE ODS_HC_FACTURAS ADD INDEX fk_fac_cic_idx (ID_CICLO_FACTURACION ASC);
ALTER TABLE ODS_HC_FACTURAS ADD CONSTRAINT fk_fac_cic FOREIGN KEY (ID_CICLO_FACTURACION)
REFERENCES ODS_DM_CICLO_FACTURACION(ID_CICLO_FACTURACION);

ALTER TABLE ODS_HC_FACTURAS ADD INDEX fk_fac_met_idx (ID_METODO_PAGO ASC);
ALTER TABLE ODS_HC_FACTURAS ADD CONSTRAINT fk_fac_met FOREIGN KEY (ID_METODO_PAGO)
REFERENCES ODS_DM_METODO_PAGO(ID_METODO_PAGO);
```

6.4. Modelado Llamadas

```
USE ODS;

DROP TABLE IF EXISTS ODS_HC_LLAMADAS;
CREATE TABLE ODS_HC_LLAMADAS (
    ID_CONTACTO INT(11) UNSIGNED AUTO_INCREMENT NOT NULL PRIMARY KEY,
    STG_ID_CONTACTO INT(11),
    TELEFONO BIGINT(20),
    FC_COMIENZO DATETIME,
    FC_FIN DATETIME,
    ID_SERVICIO INT(3) UNSIGNED,
    TRANSFERIDO BOOLEAN,
    ID_AGENTE INT(3) UNSIGNED,
    FC_INSERT DATETIME,
    FC_MODIF DATETIME
);
```

```
DROP TABLE IF EXISTS ODS_DM_SERVICIOS;
CREATE TABLE ODS_DM_SERVICIOS (
    ID_SERVICIO INT(3) UNSIGNED AUTO_INCREMENT NOT NULL PRIMARY KEY,
    DE_SERVICIO VARCHAR(512),
    FC_INSERT DATETIME,
    FC_MODIF DATETIME
);

DROP TABLE IF EXISTS ODS_DM_AGENTES;
CREATE TABLE ODS_DM_AGENTES (
    ID_AGENTE INT(3) UNSIGNED AUTO_INCREMENT NOT NULL PRIMARY KEY,
    DE_AGENTE VARCHAR(512),
    FC_INSERT DATETIME,
    FC_MODIF DATETIME
);

ALTER TABLE ODS_HC_LLAMADAS ADD INDEX con_tel_idx (TELEFONO ASC);
ALTER TABLE ODS_HC_CLIENTES ADD INDEX con_tel_idx (TELEFONO_CLIENTE ASC);

ALTER TABLE ODS_HC_LLAMADAS ADD INDEX fk_con_ser_idx (ID_SERVICIO ASC);
ALTER TABLE ODS_HC_LLAMADAS ADD CONSTRAINT fk_con_ser FOREIGN KEY (ID_SERVICIO)
    REFERENCES ODS_DM_SERVICIOS(ID_SERVICIO);

ALTER TABLE ODS_HC_LLAMADAS ADD INDEX fk_con_age_idx (ID_AGENTE ASC);
ALTER TABLE ODS_HC_LLAMADAS ADD CONSTRAINT fk_con_age FOREIGN KEY (ID_AGENTE)
    REFERENCES ODS_DM_AGENTES(ID_AGENTE);
```

6.5. Inserción Datos Facturas

```
USE ODS;

-- ODS_DM_CICLO_FACTURACION
INSERT INTO ODS_DM_CICLO_FACTURACION (DE_CICLO_FACTURACION, FC_INSERT, FC_MODIF)
    SELECT DISTINCT(UPPER(TRIM(BILL_CYCLE))) AS BILL_CYCLE, NOW(), NOW() FROM STAGE.STG_FACTURAS_FCT
    WHERE BILL_CYCLE<>' ';

INSERT INTO ODS_DM_CICLO_FACTURACION VALUES (999, 'DESCONOCIDO', NOW(), NOW());
INSERT INTO ODS_DM_CICLO_FACTURACION VALUES (998, 'NO APLICA', NOW(), NOW());

COMMIT;
ANALYZE TABLE ODS_DM_CICLO_FACTURACION;

-- ODS_DM_METODO_PAGO
INSERT INTO ODS_DM_METODO_PAGO (DE_METODO_PAGO, FC_INSERT, FC_MODIF)
    SELECT DISTINCT(UPPER(TRIM(BILL_METHOD))) AS BILL_METHOD, NOW(), NOW() FROM STAGE.STG_FACTURAS_FCT
    WHERE BILL_METHOD<>' ';

INSERT INTO ODS_DM_METODO_PAGO VALUES (999, 'DESCONOCIDO', NOW(), NOW());
INSERT INTO ODS_DM_METODO_PAGO VALUES (998, 'NO APLICA', NOW(), NOW());

COMMIT;
ANALYZE TABLE ODS_DM_METODO_PAGO;

-- ODS_HC_FACTURAS
SET FOREIGN_KEY_CHECKS=0;
```

```

INSERT INTO ODS_HC_FACTURAS
    SELECT TRIM(BILL_REF_NO) AS ID_FACTURA,
    TRIM(CUSTOMER_ID) AS ID_CLIENTE,
    CASE WHEN TRIM(SFF.START_DATE)<>' ' THEN STR_TO_DATE(SFF.START_DATE, '%Y-%m-%d %H:%i:%s') ELSE
STR_TO_DATE('31/12/9999', '%d/%m/%Y') END AS FC_COMIENZO,
    CASE WHEN TRIM(SFF.END_DATE)<>' ' THEN STR_TO_DATE(SFF.END_DATE, '%Y-%m-%d %H:%i:%s') ELSE
STR_TO_DATE('31/12/9999', '%d/%m/%Y') END AS FC_FIN,
    CASE WHEN TRIM(SFF.STATEMENT_DATE)<>' ' THEN STR_TO_DATE(SFF.STATEMENT_DATE, '%Y-%m-%d %H:%i:%s') ELSE
STR_TO_DATE('31/12/9999', '%d/%m/%Y') END AS FC_FC_ESTABLECIMIENTO,
    CASE WHEN TRIM(SFF.PAYMENT_DATE)<>' ' THEN STR_TO_DATE(SFF.PAYMENT_DATE, '%Y-%m-%d %H:%i:%s') ELSE
STR_TO_DATE('31/12/9999', '%d/%m/%Y') END AS FC_PAGO,
    ODCF.ID_CICLO_FACTURACION AS ID_CICLO_FACTURACION,
    AMOUNT AS IMPORTE,
    ODMP.ID_METODO_PAGO AS ID_METODO_PAGO,
    NOW(),
    NOW()
FROM STAGE.STG_FACTURAS_FCT SFF
    INNER JOIN ODS.ODS_DM_CICLO_FACTURACION ODCF ON CASE WHEN TRIM(SFF.BILL_CYCLE)<>' ' THEN
UPPER(TRIM(SFF.BILL_CYCLE)) ELSE 'DESCONOCIDO' END=ODCF.DE_CICLO_FACTURACION
    INNER JOIN ODS.ODS_DM_METODO_PAGO ODMP ON CASE WHEN TRIM(SFF.BILL_METHOD)<>' ' THEN
UPPER(TRIM(SFF.BILL_METHOD)) ELSE 'DESCONOCIDO' END=ODMP.DE_METODO_PAGO;

SET FOREIGN_KEY_CHECKS=1;

COMMIT;
ANALYZE TABLE ODS_HC_FACTURAS;

```

6.6. Inserción Datos Llamadas

```
USE ODS;

-- ODS_DM_AGENTES
INSERT INTO ODS_DM_AGENTES (DE_AGENTE, FC_INSERT, FC_MODIF)
    SELECT DISTINCT(UPPER(TRIM(SCI.AGENT))),
    NOW(),
    NOW()
    FROM STAGE.STG_CONTACTOS_IVR SCI
    WHERE SCI.AGENT<>' ';

INSERT INTO ODS_DM_AGENTES VALUES (999, 'DESCONOCIDO', NOW(), NOW());
INSERT INTO ODS_DM_AGENTES VALUES (998, 'NO APLICA', NOW(), NOW());

COMMIT;
ANALYZE TABLE ODS_DM_AGENTES;

-- ODS_DM_SERVICIOS
INSERT INTO ODS_DM_SERVICIOS (DE_SERVICIO, FC_INSERT, FC_MODIF)
    SELECT DISTINCT(UPPER(TRIM(SCI.SERVICE))),
    NOW(),
    NOW()
    FROM STAGE.STG_CONTACTOS_IVR SCI
    WHERE SCI.SERVICE<>' ';

INSERT INTO ODS_DM_SERVICIOS VALUES (999, 'DESCONOCIDO', NOW(), NOW());
INSERT INTO ODS_DM_SERVICIOS VALUES (998, 'NO APLICA', NOW(), NOW());
```

```
COMMIT;
ANALYZE TABLE ODS_DM_SERVICIOS;

-- ODS_HC_LLAMADAS
INSERT INTO ODS_HC_LLAMADAS (STG_ID_CONTACTO, TELEFONO, FC_COMIENZO, FC_FIN, ID_SERVICIO, TRANSFERIDO,
ID_AGENTE, FC_INSERT, FC_MODIF)
  SELECT ID,
    CASE WHEN TRIM(SCI.PHONE_NUMBER)<>' ' THEN TRIM(SCI.PHONE_NUMBER) ELSE 9999999999 END,
    CASE WHEN TRIM(SCI.START_DATETIME)<>' ' THEN STR_TO_DATE(TRIM(SCI.START_DATETIME), '%Y-%m-%d %H:%i:%s.%f')
ELSE STR_TO_DATE('31/12/9999', '%d/%m/%Y') END,
    CASE WHEN TRIM(SCI.END_DATETIME)<>' ' THEN STR_TO_DATE(TRIM(SCI.END_DATETIME), '%Y-%m-%d %H:%i:%s.%f') ELSE
STR_TO_DATE('31/12/9999', '%d/%m/%Y') END,
    ODS.ID_SERVICIO,
    CASE WHEN UPPER(TRIM(FLG_TRANSFER))='TRUE' THEN TRUE ELSE FALSE END,
    ODA.ID_AGENTE,
    NOW(),
    NOW()
  FROM STAGE.STG_CONTACTOS_IVR SCI
  INNER JOIN ODS_DM_AGENTES ODA ON CASE WHEN TRIM(SCI.AGENT)<>' ' THEN UPPER(TRIM(SCI.AGENT)) ELSE
'DESCONOCIDO' END = ODA.DE_AGENTE
  INNER JOIN ODS_DM_SERVICIOS ODS ON CASE WHEN TRIM(SCI.SERVICE)<>' ' THEN UPPER(TRIM(SCI.SERVICE)) ELSE
'DESCONOCIDO' END = ODS.DE_SERVICIO;

COMMIT;
ANALYZE TABLE ODS_HC_LLAMADAS;
```

6.7. Separación del campo DE_DIRECCION a NUM_VIA e ID_NOMBRE_VIA

```
USE ODS;

-- ESTUDIO NOMBRE VIAS
SELECT COUNT(SUBSTR(OHD.DE_DIRECCION, LOCATE(' ', OHD.DE_DIRECCION) + 1)),
       COUNT(DISTINCT(SUBSTR(OHD.DE_DIRECCION, LOCATE(' ', OHD.DE_DIRECCION) + 1)))
  FROM ODS.ODS_HC_DIRECCIONES OHD
 WHERE SUBSTR(OHD.DE_DIRECCION, LOCATE(' ', OHD.DE_DIRECCION) + 1) <> '';

-- CREACIÓN TABLA ODS_DM_NOMBRE_VIA
DROP TABLE IF EXISTS ODS_DM_NOMBRE_VIA;
CREATE TABLE ODS_DM_NOMBRE_VIA (
  ID_NOMBRE_VIA INT(11) UNSIGNED AUTO_INCREMENT NOT NULL PRIMARY KEY,
  DE_NOMBRE_VIA VARCHAR(512),
  FC_INSERT DATETIME,
  FC_MODIF DATETIME
);

-- INSERCIÓN ODS_DM_NOMBRE_VIA
INSERT INTO ODS_DM_NOMBRE_VIA (DE_NOMBRE_VIA, FC_INSERT, FC_MODIF)
SELECT DISTINCT(SUBSTR(OHD.DE_DIRECCION, LOCATE(' ', OHD.DE_DIRECCION) + 1)),
       NOW(),
       NOW()
  FROM ODS.ODS_HC_DIRECCIONES OHD
 WHERE OHD.DE_DIRECCION <> 'NO APLICA' AND OHD.DE_DIRECCION <> 'DESCONOCIDO' AND SUBSTR(OHD.DE_DIRECCION,
LOCATE(' ', OHD.DE_DIRECCION) + 1) <> '';

INSERT INTO ODS_DM_NOMBRE_VIA VALUES (999999999, 'DESCONOCIDO', NOW(), NOW());
```



```
INSERT INTO ODS_DM_NOMBRE_VIA VALUES (999999998, 'NO APLICA', NOW(), NOW());

COMMIT;
ANALYZE TABLE ODS_DM_NOMBRE_VIA;

-- CREACIÓN DE LAS COLUMNAS EN ODS_HC_DIRECCIONES
ALTER TABLE ODS.ODS_HC_DIRECCIONES
ADD COLUMN NUM_VIA INT(11) UNSIGNED NOT NULL AFTER DE_DIRECCION;
ALTER TABLE ODS.ODS_HC_DIRECCIONES
ADD COLUMN ID_NOMBRE_VIA INT(11) UNSIGNED NOT NULL AFTER NUM_VIA;

-- ES NECESARIO QUITAR LA OPCIÓN DE SAFE UPDATES/DELETES Y RECONECTAR CON EL DBMS

-- ACTUALIZACION DEL CAMPO ID_NOMBRE_VIA
UPDATE ODS_HC_DIRECCIONES OHD
INNER JOIN ODS_DM_NOMBRE_VIA ODNV ON
    SUBSTR(OHD.DE_DIRECCION, LOCATE(' ', OHD.DE_DIRECCION) + 1) = ODNV.DE_NOMBRE_VIA
SET OHD.ID_NOMBRE_VIA = ODNV.ID_NOMBRE_VIA, FC_MODIF = NOW()
WHERE OHD.DE_DIRECCION <> 'DESCONOCIDO' AND OHD.DE_DIRECCION <> 'NO APLICA';

UPDATE ODS_HC_DIRECCIONES OHD
SET OHD.ID_NOMBRE_VIA = 999999999, OHD.FC_MODIFICACION = NOW()
WHERE OHD.DE_DIRECCION='DESCONOCIDO';

UPDATE ODS_HC_DIRECCIONES OHD
SET OHD.ID_NOMBRE_VIA = 999999998, OHD.FC_MODIFICACION = NOW()
WHERE OHD.DE_DIRECCION='NO APLICA';

-- ACTUALIZACION DEL CAMPO NUM_VIA
```

```
UPDATE ODS_HC_DIRECCIONES OHD
  SET OHD.NUM_VIA = CAST(SUBSTRING_INDEX(OHD.DE_DIRECCION, ' ', 1) AS UNSIGNED), OHD.FC_MODIFICACION = NOW()
  WHERE OHD.DE_DIRECCION<>'DESCONOCIDO' AND OHD.DE_DIRECCION<>'NO APLICA';

UPDATE ODS_HC_DIRECCIONES OHD
  SET OHD.NUM_VIA = 999999999, OHD.FC_MODIFICACION = NOW()
  WHERE OHD.DE_DIRECCION='DESCONOCIDO';

UPDATE ODS_HC_DIRECCIONES OHD
  SET OHD.NUM_VIA = 999999998, OHD.FC_MODIFICACION = NOW()
  WHERE OHD.DE_DIRECCION='NO APLICA';

-- ESTABLECIMIENTO FK CON TABLA ODS_DM_NOMBRE_VIA
ALTER TABLE ODS_HC_DIRECCIONES ADD INDEX fk_dir_nom_v_idx (ID_NOMBRE_VIA ASC);
ALTER TABLE ODS_HC_DIRECCIONES ADD CONSTRAINT fk_dir_nom_v FOREIGN KEY (ID_NOMBRE_VIA)
  REFERENCES ODS_DM_NOMBRE_VIA(ID_NOMBRE_VIA);

-- BORRADO DE LA COLUMNA DE_DIRECCION
ALTER TABLE ODS_HC_DIRECCIONES
DROP COLUMN DE_DIRECCION;
```