BOOTCAMP BIG DATA & MACHINE LEARNING

TALLER DATA 101

Rico Abilowo Hardjono Hartini

Data 101

Contenido

PARTE 1

Estudio del schema STAGE

```
mysql> use STAGE; show tables;
+-----------------------------------+
| Tables_in_STAGE                   |
+-----------------------------------+
| STG_CLIENTES_CRM                  |
| STG_CONTACTOS_IVR                 |
| STG_FACTURAS_FCT                  |
| STG_ORDERS_CRM                    |
| STG_PRODUCTOS_CRM                 |
+-----------------------------------+
5 rows in set (0,01 sec)
```

El estudio y análisis de la base de datos origen con toda la información de los operacionales y que se han importado al schema STAGE como parte de una primera fase en el proceso de creación del Data Warehouse (consultar Scripts PARTE1 - STAGE), son las tareas previas para definir el modelo de datos. Del mismo modo, disponer del conocimiento sobre las líneas de negocio repercute destacadamente a la hora de interpretar los datos de forma más precisa y disponer así de un modelo de datos más real.

Sin embargo, en esta práctica realizada se ha tomado la decisión de seguir el modelo propuesto. A continuación, se describen los criterios generales que se han tomado para la definición del modelo de datos:

Seleccionar los campos que van a ser Primary Key (PK) verificando que todos sus valores son distintos y no existe ningún valor nulo o vacío.

Comprobar que los campos están rellenos:

- Si un campo está completamente vació se puede considerar excluirlo del modelo de datos.
- Si un campo tiene parcialmente valores a nulo o vacío se les asignará posteriormente un valor por defecto según el tipo del campo.
- Si el número de valores distintos del campo es considerablemente menor que el número total de elementos o registros de la tabla entonces se propone la creación de una tabla de dimensión para dicho campo.

A. CLIENTES

```
mysql> DESCRIBE STG_CLIENTES_CRM;
| Field          | Type         | Null | Key  | Default | Extra |
| CUSTOMER_ID    | varchar(512) | NO   |      | NULL    |       |
| FIRST_NAME     | varchar(512) | YES  |      | NULL    |       |
| LAST_NAME      | varchar(512) | YES  |      | NULL    |       |
| IDENTIFIED_DOC | varchar(512) | YES  |      | NULL    |       |
| GENDER         | varchar(512) | YES  |      | NULL    |       |
| CITY           | varchar(512) | YES  |      | NULL    |       |
| ADDRESS        | varchar(512) | YES  |      | NULL    |       |
| POSTAL_CODE    | varchar(512) | YES  |      | NULL    |       |
| STATE          | varchar(512) | YES  |      | NULL    |       |
| COUNTRY        | varchar(512) | YES  |      | NULL    |       |
| PHONE          | varchar(512) | YES  | MUL  | NULL    |       |
| EMAIL          | varchar(512) | YES  |      | NULL    |       |
| BIRTHDAY       | varchar(512) | YES  |      | NULL    |       |
| PROFESION      | varchar(512) | YES  |      | NULL    |       |
| COMPANY        | varchar(512) | YES  |      | NULL    |       |
15 rows in set (0,00 sec)
```

TOTAL_REGISTROS	
17558	
TOTAL_CUSTOMER_ID	TOTAL_DISTINTOS_CUSTOMER_ID
17558	17558
TOTAL_FIRST_NAME	TOTAL_DISTINTOS_FIRST_NAME
17558	7314
TOTAL_LAST_NAME	TOTAL_DISTINTOS_LAST_NAME
17497	14577
TOTAL_IDENTIFIED_DOC	TOTAL_DISTINTOS_IDENTIFIED_DOC
17497	17498
TOTAL_GENDER	TOTAL_DISTINTOS_GENDER
17497	3
TOTAL_CITY	TOTAL_DISTINTOS_CITY
17497	82
TOTAL_ADDRESS	TOTAL_DISTINTOS_ADDRESS
17497	17439
TOTAL_POSTAL_CODE	TOTAL_DISTINTOS_POSTAL_CODE
17497	274
TOTAL_STATE	TOTAL_DISTINTOS_STATE
17497	4
TOTAL_COUNTRY	TOTAL_DISTINTOS_COUNTRY
17497	2
TOTAL_PHONE	TOTAL_DISTINTOS_PHONE
17497	17498
TOTAL_EMAIL	TOTAL_DISTINTOS_EMAIL
17497	17498
TOTAL_BIRTHDAY	TOTAL_DISTINTOS_BIRTHDAY
17497	10753
TOTAL_PROFESION	TOTAL_DISTINTOS_PROFESION
17497	196
TOTAL_COMPANY	TOTAL_DISTINTOS_COMPANY
17451	384

Análisis

A partir de la tabla anterior se indican las siguientes conclusiones:

El campo **CUSTOMER_ID** es el PK de la tabla **STG_CLIENTES_CRM**. No incluye valores nulos y no es autoincremental (se insertan los propios identificadores de cliente que son todos distintos puesto que **TOTAL_DISTINTOS_CUSTOMER_ID** coincide con **TOTAL_REGISTROS**)

Se crean tablas dimensión para los campos: **GENDER**, **COUNTRY**, **PROFESION** y **COMPANY** puesto que el número de elementos distintos en cada campo es destacadamente inferior que **TOTAL_REGISTROS**.

Para el caso del campo **POSTAL_CODE** no se ha definido una tabla dimensión a pesar de cumplir los requisitos para ello porque no se ha considerado parte de la jerarquía de información asociada a una dirección.

No se ha creado tabla dimensión conjunta para los campos **CITY** y **STATE**. Esta decisión se explicará con mayor detalle en el apartado PARTE2 - Cuestión de diseño.

Con respecto a las tablas dimensión, cabe destacar que la existencia de tablas maestras hubiera facilitado la creación y mantenimiento de las mismas.

Para los campos **FIRST_NAME**, **LAST_NAME**, **BIRTHDAY** y **ADDRESS** se asume que los clientes pueden compartir entre sí un mismo nombre, apellido o fecha de nacimiento e incluso un mismo domicilio y no tendría sentido crear tablas dimensión para estos campos.

Los campos **IDENTIFIED_DOC**, **PHONE**, **EMAIL** se importarán directamente sin ninguna modificación.

Hay que señalar que el número total de elementos que en la mayoría de los campos es de 17497 es inferior al **TOTAL_REGISTROS** que es 17558. Esto se interpreta en que la diferencia entre dichos valores corresponde al número de elementos que son nulos o vacíos. Como se ha indicado anteriormente, estos valores nulos o vacíos serán modificados en la importación por un valor por defecto.

Finalmente, a partir de los datos de los campos **ADDRESS**, **POSTAL_CODE**, **CITY**, **STATE**, y **COUNTRY** se define una entidad propia para la información de las direcciones.

[*] Para los siguientes modelos tablas ya muestran en una columna el número de elementos nulos o vacíos.

B. PRODUCTOS

```
mysql> describe STG_PRODUCTOS_CRM;
| Field                | Type         | Null | Key | Default | Extra |
| PRODUCT_ID           | varchar(512) | NO   |     | NULL    |       |
| CUSTOMER_ID          | varchar(512) | YES  | MUL | NULL    |       |
| PRODUCT_NAME         | varchar(512) | YES  |     | NULL    |       |
| ACCESS_POINT         | varchar(512) | YES  |     | NULL    |       |
| CHANNEL              | varchar(512) | YES  |     | NULL    |       |
| AGENT_CODE           | varchar(512) | YES  |     | NULL    |       |
| START_DATE           | varchar(512) | YES  |     | NULL    |       |
| INSTALL_DATE         | varchar(512) | YES  |     | NULL    |       |
| END_DATE             | varchar(512) | YES  |     | NULL    |       |
| PRODUCT_CITY         | varchar(512) | YES  |     | NULL    |       |
| PRODUCT_ADDRESS      | varchar(512) | YES  | MUL | NULL    |       |
| PRODUCT_POSTAL_CODE  | varchar(512) | YES  | MUL | NULL    |       |
| PRODUCT_STATE        | varchar(512) | YES  |     | NULL    |       |
| PRODUCT_COUNTRY      | varchar(512) | YES  |     | NULL    |       |
14 rows in set (0,00 sec)
```

TOTAL_REGISTROS		
78495		
TOTAL_PRODUCT_ID	TOTAL_NULOSVACIOS_PRODUCT_ID	TOTAL_DISTINTOS_PRODUCT_ID
78495	0	78495
TOTAL_CUSTOMER_ID	TOTAL_NULOSVACIOS_CUSTOMER_ID	TOTAL_DISTINTOS_CUSTOMER_ID
78495	0	8001
TOTAL_PRODUCT_NAME	TOTAL_NULOSVACIOS_PRODUCT_NAME	TOTAL_DISTINTOS_PRODUCT_NAME
78495	0	6
TOTAL_ACCESS_POINT	TOTAL_NULOSVACIOS_ACCESS_POINT	TOTAL_DISTINTOS_ACCESS_POINT
78274	221	78275
TOTAL_CHANNEL	TOTAL_NULOSVACIOS_CHANNEL	TOTAL_DISTINTOS_CHANNEL
78274	221	5
TOTAL_AGENT_CODE	TOTAL_NULOSVACIOS_AGENT_CODE	TOTAL_DISTINTOS_AGENT_CODE
42630	35865	701
TOTAL_START_DATE	TOTAL_NULOSVACIOS_START_DATE	TOTAL_DISTINTOS_START_DATE
78495	0	8035
TOTAL_INSTALL_DATE	TOTAL_NULOSVACIOS_INSTALL_DATE	TOTAL_DISTINTOS_INSTALL_DATE
75363	3132	75360
TOTAL_END_DATE	TOTAL_NULOSVACIOS_END_DATE	TOTAL_DISTINTOS_END_DATE
46684	31811	46683
TOTAL_PRODUCT_CITY	TOTAL_NULOSVACIOS_PRODUCT_CITY	TOTAL_DISTINTOS_PRODUCT_CITY
78274	221	83
TOTAL_PRODUCT_ADDRESS	TOTAL_NULOSVACIOS_PRODUCT_ADDRESS	TOTAL_DISTINTOS_PRODUCT_ADDRESS
78274	221	77037
TOTAL_PRODUCT_POSTAL_CODE	TOTAL_NULOSVACIOS_PRODUCT_POSTAL_CODE	TOTAL_DISTINTOS_PRODUCT_POSTAL_CODE
78274	221	274
TOTAL_PRODUCT_STATE	TOTAL_NULOSVACIOS_PRODUCT_STATE	TOTAL_DISTINTOS_PRODUCT_STATE
78090	405	5
TOTAL_PRODUCT_COUNTRY	TOTAL_NULOSVACIOS_PRODUCT_COUNTRY	TOTAL_DISTINTOS_PRODUCT_COUNTRY
78274	221	3

Análisis

A partir de la tabla anterior se indican las siguientes conclusiones:

El campo **PRODUCT_ID** es el PK de la tabla **STG_PRODUCTOS_CRM**. No incluye valores nulos y no es autoincremental.

El campo **CUSTOMER_ID** es el Foreign Key (FK) de la tabla **STG_CLIENTES_CRM**

Se crean tablas dimensión para los campos **PRODUCT_NAME** y **CHANNEL**.

Los campos **ACCESS_POINT**, **AGENT_CODE**, **START_DATE**, **INSTALL_DATE** y **END_DATE** se importarán directamente sin ninguna modificación. Si el campo **AGENT_CODE** no fuera numérico se puede plantear crear una tabla dimensión para él.

Los campos **PRODUCT_ADDRESS**, **PRODUCT_POSTAL_CODE**, **PRODUCT_CITY**, **PRODUCT_STATE**, y **PRODUCT_COUNTRY** definen la información de la dirección de la instalación de un servicio o producto, pero para nuestro modelo se considera la misma que la dirección de cliente y por tanto se agregará a la misma entidad de direcciones.

C. FACTURAS

```
mysql> describe STG_FACTURAS_FCT;
| Field          | Type         | Null | Key | Default | Extra |
| BILL_REF_NO    | varchar(512) | NO   |     | NULL    |       |
| CUSTOMER_ID    | varchar(512) | YES  |     | NULL    |       |
| START_DATE     | varchar(512) | YES  |     | NULL    |       |
| END_DATE       | varchar(512) | YES  |     | NULL    |       |
| STATEMENT_DATE | varchar(512) | YES  |     | NULL    |       |
| PAYMENT_DATE   | varchar(512) | YES  |     | NULL    |       |
| BILL_CYCLE     | varchar(512) | YES  |     | NULL    |       |
| AMOUNT         | varchar(512) | YES  |     | NULL    |       |
| BILL_METHOD    | varchar(512) | YES  |     | NULL    |       |
9 rows in set (0,00 sec)
```

TOTAL_REGISTROS		
420000		
TOTAL_BILL_REF_NO	TOTAL_NULOSVACIOS_BILL_REF_NO	TOTAL_DISTINTOS_BILL_REF_NO
420000	**0**	**420000**
TOTAL_CUSTOMER_ID	TOTAL_NULOSVACIOS_CUSTOMER_ID	TOTAL_DISTINTOS_CUSTOMER_ID
420000	**0**	**20000**
TOTAL_START_DATE	TOTAL_NULOSVACIOS_START_DATE	TOTAL_DISTINTOS_START_DATE
420000	**0**	**40**
TOTAL_END_DATE	TOTAL_NULOSVACIOS_END_DATE	TOTAL_DISTINTOS_END_DATE
420000	**0**	**20**
TOTAL_STATEMENT_DATE	TOTAL_NULOSVACIOS_STATEMENT_DATE	TOTAL_DISTINTOS_STATEMENT_DATE
420000	**0**	**40**
TOTAL_PAYMENT_DATE	TOTAL_NULOSVACIOS_PAYMENT_DATE	TOTAL_DISTINTOS_PAYMENT_DATE
420000	**0**	**400**
TOTAL_BILL_CYCLE	TOTAL_NULOSVACIOS_BILL_CYCLE	TOTAL_DISTINTOS_BILL_CYCLE
420000	**0**	**2**
TOTAL_AMOUNT	TOTAL_NULOSVACIOS_AMOUNT	TOTAL_DISTINTOS_AMOUNT
420000	**0**	**5604**
TOTAL_BILL_METHOD	TOTAL_NULOSVACIOS_BILL_METHOD	TOTAL_DISTINTOS_BILL_METHOD
420000	**0**	**3**

Análisis

A partir de la tabla anterior se indican las siguientes conclusiones:

El campo **BILL_REF_NO** es el PK de la tabla **STG_FACTURAS_FCT**. No incluye valores nulos y no es autoincremental.

El campo **CUSTOMER_ID** es FK de la tabla **STG_CLIENTES_CRM**.

Se crean tablas dimensión para los campos **BILL_CYCLE** y **BILL_METHOD**.

El resto de los campos se importarán directamente sin ninguna modificación. Cabe destacar que los campos de tipo fecha (**START_DATE**, **END_DATE**, **STATEMENT_DATE** y **PAYMENT_DATE**) tienen un número de valores distintos muy bajo (20) lo que sugiere que los periodos de facturación se establecen en fechas fijas. Así tenemos que la fecha de inicio de facturación se fija el día 1 o 15 de cada mes, la fecha de finalización de facturación se fija el día 1 de cada mes y que la fecha de declaración de la factura se realiza dos días después de la fecha de inicio, es decir, los 3 o 17 de cada mes.

Esto se verifica mediante las siguientes consultas:

```
SELECT start_date, COUNT(*) total FROM STAGE.STG_FACTURAS_FCT GROUP BY start_date;
SELECT end_date, COUNT(*) total FROM STAGE.STG_FACTURAS_FCT GROUP BY end_date;
SELECT statement_date, COUNT(*) total FROM STAGE.STG_FACTURAS_FCT GROUP BY statement_date;
```

D. LLAMADAS

```
mysql> describe STG_CONTACTOS_IVR;
| Field          | Type         | Null | Key | Default | Extra |
| ID             | varchar(512) | NO   |     | NULL    |       |
| PHONE_NUMBER   | varchar(512) | YES  | MUL | NULL    |       |
| START_DATETIME | varchar(512) | YES  |     | NULL    |       |
| END_DATETIME   | varchar(512) | YES  |     | NULL    |       |
| SERVICE        | varchar(512) | YES  |     | NULL    |       |
| FLG_TRANSFER   | varchar(512) | YES  |     | NULL    |       |
| AGENT          | varchar(512) | YES  |     | NULL    |       |
7 rows in set (0,00 sec)
```

# TOTAL_REGISTROS		
202717		
TOTAL_ID	TOTAL_NULOSVACIOS_ID	TOTAL_DISTINTOS_ID
202717	**0**	**150000**
TOTAL_PHONE_NUMBER	TOTAL_NULOSVACIOS_PHONE_NUMBER	TOTAL_DISTINTOS_PHONE_NUMBER
185018	**17699**	**18226**
TOTAL_START_DATETIME	TOTAL_NULOSVACIOS_START_DATETIME	TOTAL_DISTINTOS_START_DATETIME
202717	**0**	**201098**
TOTAL_END_DATETIME	TOTAL_NULOSVACIOS_END_DATETIME	TOTAL_DISTINTOS_END_DATETIME
186535	**16182**	**183678**
TOTAL_SERVICE	TOTAL_NULOSVACIOS_SERVICE	TOTAL_DISTINTOS_SERVICE
202502	**215**	**7**
TOTAL_FLG_TRANSFER	TOTAL_NULOSVACIOS_FLG_TRANSFER	TOTAL_DISTINTOS_FLG_TRANSFER
202717	**0**	**2**
TOTAL_AGENT	TOTAL_NULOSVACIOS_AGENT	TOTAL_DISTINTOS_AGENT
194739	**7978**	**594**

Análisis

A partir de la tabla anterior se indican las siguientes conclusiones:

El campo **ID** no puede ser utilizado como PK de la tabla **STG_CONTACTOS_IVR**. Puesto que a pesar de no tener valores nulos sí que tiene valores repetidos. Por este motivo, se precisa definir un campo identificador para la tabla **ODS_HC_LLAMADAS** que sea autoincremental.

Se crean tablas dimensión para los campos **SERVICE** y **AGENT**.

Los campos **PHONE_NUMBER**, **START_DATE**, **END _DATE** y **FLG_TRANSFER** se importarán directamente sin ninguna modificación. El campo **FLG_ TRANSFER** es similar al campo **GENDER** de Clientes, tiene dos valores posibles (True y False) pero en este caso no se crea una tabla dimensión por ser de tipo booleano, aunque se podría aplicar una transformación a 1 y 0.

E. ORDENES

```
mysql> describe STG_ORDERS_CRM;
| Field      | Type         | Null | Key | Default | Extra |
| ID         | varchar(512) | NO   |     | NULL    |       |
| ORDER      | varchar(512) | YES  |     | NULL    |       |
| PHASE      | varchar(512) | YES  |     | NULL    |       |
| AGENT      | varchar(512) | YES  |     | NULL    |       |
| START_DT   | varchar(512) | YES  |     | NULL    |       |
| END_DT     | varchar(512) | YES  |     | NULL    |       |
6 rows in set (0,00 sec)
```

TOTAL_REGISTROS		
360067		
TOTAL_ID	TOTAL_NULOSVACIOS_ID	TOTAL_DISTINTOS_ID
360067	**0**	**324081**
TOTAL_ORDER	TOTAL_NULOSVACIOS_ORDER	TOTAL_DISTINTOS_ORDER
360067	**0**	**78000**
TOTAL_PHASE	TOTAL_NULOSVACIOS_PHASE	TOTAL_DISTINTOS_PHASE
360067	**0**	**7**
TOTAL_AGENT	TOTAL_NULOSVACIOS_AGENT	TOTAL_DISTINTOS_AGENT
360032	**35**	**101**
TOTAL_START_DT	TOTAL_NULOSVACIOS_START_DT	TOTAL_DISTINTOS_START_DT
360067	**0**	**342069**
TOTAL_END_DT	TOTAL_NULOSVACIOS_END_DT	TOTAL_DISTINTOS_END_DT
282067	**78000**	**270383**

Análisis

A partir de la tabla anterior se indican las siguientes conclusiones:

El campo **ID** no puede ser utilizado como PK de la tabla **STG_ORDERS_CRM**. Puesto que a pesar de no tener valores nulos sí que tiene valores repetidos. Por este motivo, se precisa definir un campo identificador para la tabla **ODS_HC_PROVISION** que sea autoincremental.

Se crean tabla dimensión para el campo **PHASE** y **ORDERS.**

Los campos **START_DT** y **END _DATE** se importarán directamente sin ninguna modificación y el campo **AGENT** se agregará a la tabla dimensión de AGENT definida anteriormente.

Valores por defecto

Para cargar los datos desde la capa STAGE a la capa ODS de la arquitectura Data Warehouse (Data Warehouse - Arquitectura) planteada, se realiza un proceso ETL basada en queries y CTAS cuyo resultado es la creación de una base de datos normalizada en el que se definen las tablas de Dimensiones y de Hechos.

Durante este proceso ETL no se ha dispuesto de tablas de equivalencia o maestras que permitiesen evitar inconsistencia y duplicidad de los datos. Por otro lado, para obtener la normalización de la base de datos se ha cargado un conjunto de registros por defecto para asegurar que no hubiese atributos o campos con valores nulos. Estos registros son los siguientes:

TABLAS DIMENSIONES

DIMENSION GENDER/SEXOS
(ID, DESCRIPCION)
99, 'DESCONOCIDO'
98, 'NO APLICA'

DIMENSION PROFESION/PROFESIONES
(ID, DESCRIPCION)
999, 'DESCONOCIDO'
998, 'NO APLICA'

DIMENSION COMPANY/COMPANYAS
(ID, DESCRIPCION)
999, 'DESCONOCIDO'
998, 'NO APLICA'

DIMENSION COUNTRY/PAISES
(ID, DESCRIPCION)
99, 'DESCONOCIDO'
98, 'NO APLICA'

DIMENSION CITY_STATE/CIUDADES_ESTADOS
(ID_CIUDAD_ESTADO, DESCRIPCION_CIUDAD, DESCRIPCION_ESTADO, ID_PAIS)
999, 'DESCONOCIDO', 'DESCONOCIDO', 99
998, 'NO APLICA', 'NO APLICA', 98

DIMENSION CHANNEL/CANALES
(ID, DESCRIPCION)
999, 'DESCONOCIDO'
998, 'NO APLICA'

DIMENSION PRODUCT/PRODUCTOS
(ID, DESCRIPCION)
999, 'DESCONOCIDO'
998, 'NO APLICA'

DIMENSION BILL_METHOD/METODOS_PAGO
(ID, DESCRIPCION)
999, 'DESCONOCIDO'
998, 'NO APLICA'

DIMENSION BILL_CYCLE/CICLOS_FACTURACION
(ID, DESCRIPCION)
999, 'DESCONOCIDO'
998, 'NO APLICA'

DIMENSION AGENT/AGENTES_CC
(ID, DESCRIPCION)
99999, 'DESCONOCIDO'
99998, 'NO APLICA'

DIMENSION SERVICE/DEPARTAMENTOS_CC
(ID, DESCRIPCION)
999, 'DESCONOCIDO'
998, 'NO APLICA'

TABLAS HECHOS

HECHO DIRECCIONES
(ID_DIRECCION, DESCRIPCION_DIRECCION, DESCRIPCION_CODIGO_POSTAL, ID_CIUDAD_ESTADO**)**
999999, 'DESCONOCIDO', 99999, 999
999998, 'NO APLICA', 99998, 998

HECHO CLIENTES
(ID_CLIENTE, DESCRIPCION_NOMBRE, DESCRIPCION_APELLIDO, DESCRIPCION_NUMDOC, ID_SEXO, ID_DIRECCION, DESCRIPCION_TELEFONO, DESCRIPCION_EMAIL, DESCRIPCION_FECHA_NACIMIENTO, ID_PROESION, ID_COMPANYA**)**
999999999, 'DESCONOCIDO', 'DESCONOCIDO', '99-999-9999', 99, 999999, 9999999999, 'DESCONOCIDO', STR_TO_DATE('31/12/9999', '%d/%m/%Y %T'), 999, 999

Otros resultados:

HECHO DIRECCIONES

Existen varias direcciones en común entre las tablas STG_CLIENTES_CRM y STG_PRODUCTOS_CRM por lo que es importante evitar que se generen registros duplicados en la tabla ODS_HC_DIRECCIONES al agregar las direcciones de Productos sobre las direcciones de clientes.

HECHO CLIENTES

Se ha creado este registro que corresponde a un usuario que no tienen ninguno de sus campos informados, es decir, tienen los valores por defecto. De este modo, cuando se agregue a la tabla ODS_HC_CLIENTES un nuevo cliente que cumpla estas condiciones no se generará un nuevo identificador, sino que se referenciará a este cliente desconocido con identificador 999999999.

Esta situación se produce con la tabla STG_PRODUCTOS_CRM donde hay productos que referencian a clientes que no existen o que fueron eliminados por algún motivo de la tabla de STG_CLIENTES_CRM sin indicar que fueron dados de baja

HECHO LLAMADAS

En el proceso de transformación entre la fase STAGE y la fase ODS se ha verificado que no existe relación entre los números de teléfono registrados en las llamadas (tablas STG_CONTACTOS_IVR) y los números de teléfono de los clientes (tabla STC_CLIENTES_CRM). Por este motivo, todas las llamadas se han asignado al cliente por defecto/desconocido.

Tampoco es posible establecer una relación entre los números de teléfono registrados en las llamadas y el producto contratado por el cliente.

Creación de los modelos de datos

A. PRODUCTOS

Consultar Scripts PARTE1 - ODS.

B. FACTURAS

Consultar Scripts PARTE1 - ODS.

C. LLAMADAS

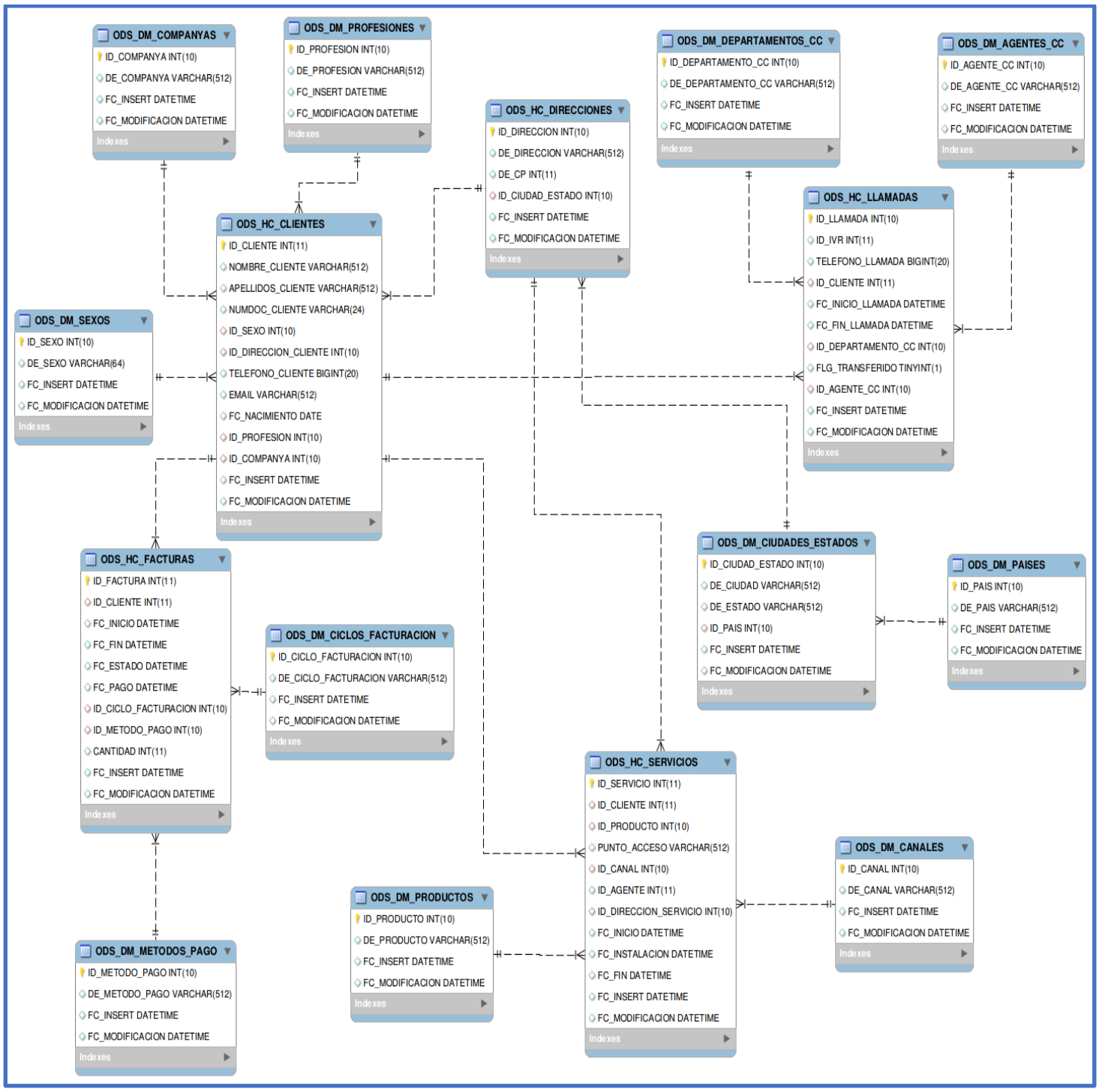
Consultar Scripts PARTE1 - ODS.

D. ORDENES

Consultar Scripts PARTE1 - ODS.

PARTE 2

Diagrama completo del schema ODS



Cuestión de diseño

¿Por qué en el modelo de DIRECCIONES dejo en la misma tabla las CIUDADES y los ESTADOS y no los separo en dos tablas distintas para ser más estricta con la jerarquía: PAIS → ESTADOS → CIUDADES → DIRECCIONES

Para establecer en el modelo de Direcciones (ODS_HC_DIRECCIONES) una relación de jerarquía: PAIS --> ESTADO --> CIUDAD --> DIRECCION debe existir una relación 1 a 1 entre la entidad hijo con respecto a la entidad padre, por ejemplo, debe suceder que una Ciudad pertenezca a un solo Estado y que un Estado pertenezca a su vez a un solo País.

Sin embargo, tras analizar los campos City y State de la tabla STAGE.STG_CLIENTES_CRM se ha observado la existencia de una Ciudad que pertenece a dos Estados diferentes. De este modo si consultásemos o filtrásemos por dicha ciudad obtendríamos dos valores de Estados distintos.

No se puede establecer una relación de jerarquía entre Estado y Ciudad porque no hay una relación 1 a 1 entre la entidad hija con respecto a la entidad padre, en concreto existe una Ciudad que pertenece a dos Estados diferentes.

Esto lo podemos comprobar con la siguiente consulta que devuelve el número de estados al que pertenece una ciudad:

```
SELECT DISTINCT CITY, COUNT(*) TOTAL_STATES FROM (
        SELECT DISTINCT CITY, STATE FROM STAGE.STG_CLIENTES_CRM GROUP BY CITY, STATE HAVING
CITY<>'' OR STATE<>'') CITIES_STATES
GROUP BY CITY
ORDER BY TOTAL_STATES DESC;
```

```
# CITY, TOTAL_STATES
 Glendale, 2
 Henderson, 1
 Oakland, 1
 Sacramento, 1
 Simi Valley, 1
 Brea, 1
 Huntington Beach, 1
 Oceanside, 1
 Salinas, 1
 South Lake Tahoe, 1
```

Para el modelo de Direcciones se ha planteado una tabla auxiliar (ODS_DM_CIUDADES_ESTADOS) que relaciona la Ciudad y el Estado generando un nuevo campo identificador/clave.

Esto sería equivalente a crear las tablas de dimensiones de Ciudad y Estado y luego definir una tabla intermedia de Ciudades_Estados pero esto se puede considerar menos óptimo o eficiente pues habría que mantener 3 tablas.

Otras conclusiones:

Puede ser una opción de diseño alternativa, crear las dimensiones para País, Estado, Ciudad y Código Postal de forma independiente para luego a partir de ellas crear el modelo de Direcciones definiendo como identificador una PK compuesta por dichos campos o como una PK autoincremental.

¿Qué haríamos si en un futuro tuviésemos un mismo estado en diferentes países?. Aunque hay que tener un conocimiento del modelo de negocio para saber qué datos existen o se van a utilizar, en principio con el modelo de Dirección actual se necesitaría realizar modificaciones.

Cuestión de diseño Opcional

¿Serías capaz de separar el campo DE_DIRECCION de la tabla de direcciones en dos campos NOMBRE_VIA y NUM_VIA?

Para realizar esta operación hemos comprobado que el campo DE_DIRECCION tiene un formato común consistente en: [numero_via(espacio)nombre_via]

Se utiliza la función SUBSTRING_INDEX() para extraer las dos partes que componen la dirección.

```
SELECT
SUBSTRING_INDEX(DE_DIRECCION, ' ', -1) NOMBRE_VIA,
SUBSTRING_INDEX(DE_DIRECCION, ' ', 1) NUM_VIA
FROM ODS.ODS_HC_DIRECCIONES
WHERE ID_DIRECCION NOT IN (999999, 999998)
ORDER BY NOMBRE_VIA;
```

Consultar Scripts PARTE2.

PARTE 3

Data Management

La realidad es que si hubiésemos aplicado el "Data Management", muchas de las acciones que hemos tenido que realizar nos las hubiésemos evitado porque deberían estar controladas de otra forma. Explica qué habrías hecho diferente centrándote en las "patas":

- Data Quality

Data Quality es la disciplina dentro del Data Management encargada de definir, controlar y mejorar la calidad de los datos. Se persigue que los datos sean consistentes, precisos y completos y que tengan una correcta interrelación con todas las fuentes.

No tenemos una perspectiva general del negocio para poder evaluar de forma completa a priori si todos los datos disponibles en Operacional o en STAGE tienen una buena calidad. Sin embargo, tras estudiar los resultados de varios análisis realizados a los datos podemos comentar algunos aspectos a tener en cuenta para poder mejorar la calidad de los datos:

Diversos atributos o campos no están completamente rellenos, es decir, tienen valores vacíos o nulos. Hay que determinar si es debido a fallos en la toma del dato o es la propia naturaleza del campo.

Los datos de teléfono recogidos en la tabla de Llamadas no se han podido vincular a Clientes o Productos, por lo que la información que nos pueda dar no tendría suficiente valor o pudiera ser prescindible.

- Master Data

El Master Data Management (MDM) tiene como objetivo definir y crear una estructura de los datos maestros de la empresa que permita conseguir una visión consistente, confiable y compartida de los datos. Comprende el conjunto de procesos y herramientas para agregar, unificar y asegurar la consistencia y calidad de los datos, evitando que la información quede obsoleta, incompleta o duplicada.

Si esto se traslada al desarrollo realizado en la práctica, se observa que es necesario la existencia de una BBDD confiable donde se gestionase de forma unificada todos los datos de las Dimensiones. Esta BBDD sería utilizada por las distintas áreas operacionales de CRM, FACTURADOR e IVR y se resolverían de esta forma posibles inconsistencias o redundancias como así ha sucedido con la Dimensión de País dónde varios valores ("US" y "United States") supuestamente hacían referencia a un mismo país.

- Data Modeling & Design

Esta parte es la encargada del diseño y el modelado de las bases de datos especificando la estructura y la organización de los datos. Bajo mi punto de vista los datos origen de los operacionales presentan en ciertos casos un diseño del modelo incompleto. Un claro error es la de no incluir los datos del número de teléfono que están asociados al producto que contrata un cliente. Dado que es una empresa de telecomunicaciones es fundamental conocer para los servicios que se ofertan (telefonía fija, telefonía móvil, internet, etc.) qué números están vinculados a los mismos. Esta información proporcionaría a su vez la posibilidad de identificar si las llamadas realizadas al call-center pertenecen a un cliente o no.

Nota: Además de las obvias que nos han salido al crear ODS que hay que describirlas, ¿se te ocurre alguna otra normalización?

En el schema ODS se ha realizado la normalización en la tercera forma normal (3FN) por la cual:

Cumple la primera forma normal (1FN):

- Todos los atributos son atómicos
- No hay registros duplicados.
- La tabla contiene una clave primaria única sin valores nulos.

Cumple la segunda forma normal (2FN):

- Cada columna de la tabla depende de forma completa de la clave principal.

Además cumple:

- Ninguna columna depende de otra columna que no sea clave principal.

Cuestión Opcional#1

¿Aconsejarías algún cambio en los sistemas origen extra teniendo en cuenta el resto de disciplinas del Data Governance?

Asignar privilegios específicos sobre los datos a los usuarios en función de roles predeterminados de modo que se puede controlar quien puede acceder a los datos y para qué fin. En la práctica se ha podido descubrir que hay productos que están relacionados con clientes cuyos identificadores no están en la tabla de Clientes. Esta situación puede haberse debido a que dichos clientes se han eliminado de manera manual. Con el mecanismo anteriormente indicado y bajo un consenso de todas las áreas implicadas se podría garantizar que solo el departamento de bajas pudiera realizar la baja del cliente y siguiendo unos procedimientos específicos (por ejemplo realizar bajas lógicas de clientes, en lugar de eliminar el registro del cliente completamente de la base de datos)

Cuestión Opcional#2

Utilizando alguna herramienta del mercado o inventándote un modelo en BBDD genera la trazabilidad de la información (Data Governance).

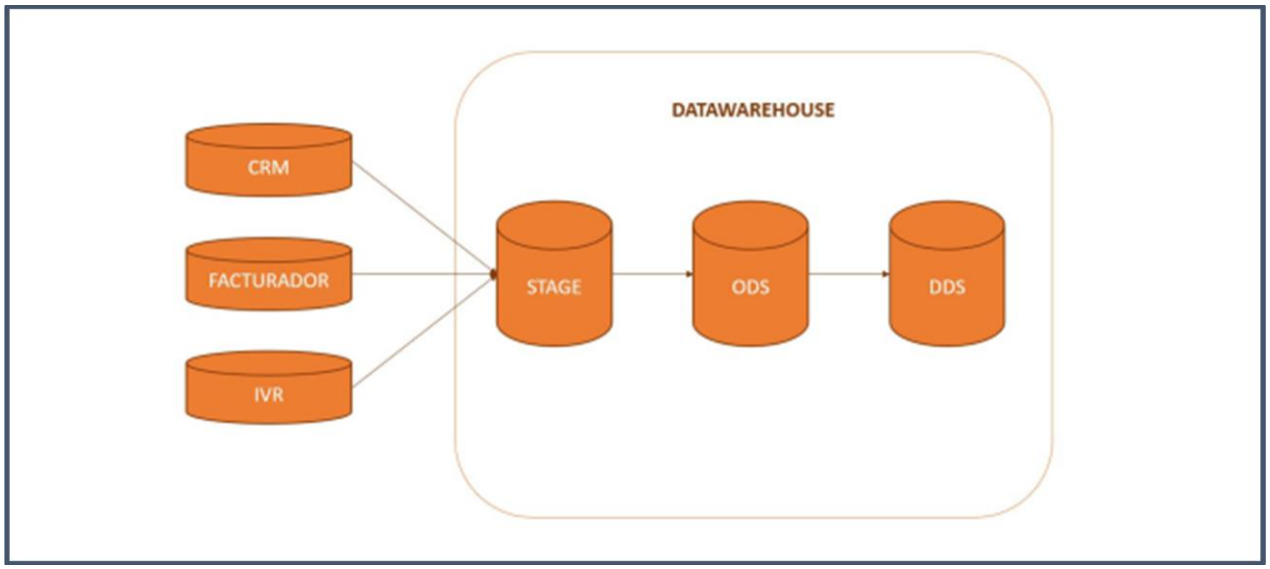
Por ejemplo para la capa STAGE se puede definir la siguiente tabla para controlar y gestionar la trazabilidad de los datos que son cargados desde los operacionales:

MONITORIZACION_STAGE
Tabla_Cargada
Fecha_Carga
Total_Registros
Total_Registros_Campo1
Total_RegistrosNulos_Campo1
Total_RegistrosDistintos_Campo1
Total_Registros_CampoN
Total_RegistrosNulos_CampoN
Total_RegistrosDistintos_CampoN

PARTE 4

Data Warehouse - Arquitectura

Después de todo lo visto nuestro ecosistema quedaría así:

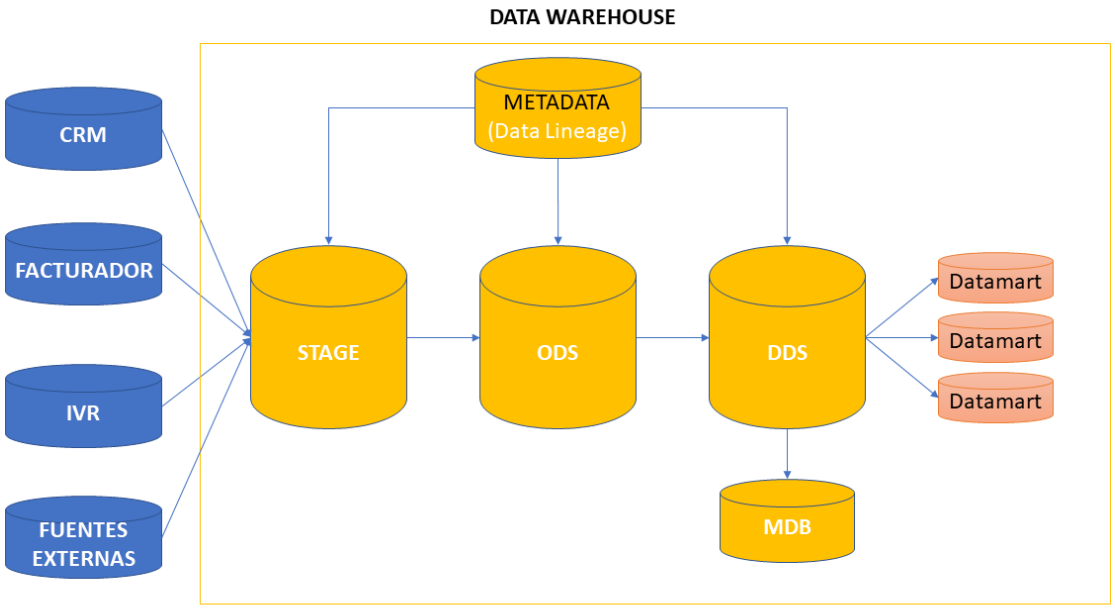


En el área de STAGE se almacena toda la información en bruto (en esta fase no se considera la realización de transformaciones en los datos, aunque sí es posible realizar aplicar filtros relacionados al negocio) tanto de las bases de datos de operación de la empresa como de otras fuentes externas. Es en definitiva una copia temporal de los sistemas orígenes.
Estos datos se actualizan generalmente en cargas mensual en procesos batch que se llevan a cabo fuera del horario laboral y cuando menos repercuta en el uso normal de la fuente de datos.

En la capa ODS (*operational data store*) se cargan los datos en un modelo relacional y normalizado. Se integran los datos de las distintas fuentes mediante transformaciones que incluyen la limpieza de datos y el control de integridad referencial así como la creación de las tablas de Dimensiones y Hechos. Los datos almacenados son la versión más reciente del Master Data, es decir, del momento actual y no se actualiza con los cambios de las fuentes origen hasta que finaliza la carga correspondiente.

Finalmente, en la capa DDS (*dimensional data store*) los datos que pueden estar desnormalizados, se orientan a la explotación analítica de la información por los usuarios o aplicaciones finales.

¿Lo dejarías así o plantearías otro diseño mejorado?



Los Datamart son un subconjunto de los datos del Data Warehouse (DDS) con la información resumida ajustada a los requerimientos de un determinado departamento.

MDB (Multidimensional Data Base) es un tipo de base de datos donde los datos se almacenan en celdas conformando estructuras de 'cubos'. y la posición de cada celda se define mediante una serie de dimensiones. Cada celda representa un evento de negocios, y el valor de las dimensiones muestra cuándo y dónde sucedió este evento. A partir del MDB se pueden realizar tareas de Data Mining, Reporting u otras aplicaciones Business Intelligence.

Metadata es un repositorio de los metadatos asociados al Data Warehouse. Estos metadatos pueden incluir información de diferentes tipos como la definición de la estructura del Data Warehouse, la propiedad de los datos o la trazabilidad de los datos desde que son cargados, transformados y finalmente utilizados por los usuarios finales (Data Lineage)

Según el flujo de datos la arquitectura del Data Warehouse puede presentar varios tipos:

- Single DDS
- NDS+DDS
- ODS+DDS (este sería la arquitectura planteada en la práctica)

PARTE 5

Data Warehouse - Mandamientos

Escribe tus propias reglas o mandamientos de un Data Warehouse

Tener un conocimiento completo del modelo de negocio y tener claro qué objetivos y resultados se persiguen, ya que es la base para poder definir los procesos ETL que permiten dar valor a la carga de los datos en el Data Warehouse.

Tener una estrategia corporativa común de todas las áreas funcionales de la empresa hacia la implantación y desarrollo de Data Management. Como hemos podido comprobar en esta práctica el haber tenido un buen diseño de los modelos así como una buena calidad de datos hubiesen facilitado la creación del Data Warehouse.

PARTE 6

Nivel SQL

¿Nivel de SQL antes y después?

	ANTES	DESPUES
Estoy más perdido que un muelle en las escaleras de Hogwarts		
Dejad que las queries se acerquen a mí	X	X
Todo lo que quiso saber sobre SQL y no se atrevió a preguntar		
TRUNCATE TABLE "sql_problems"		

ANEXOS

Scripts PARTE1 - STAGE

https://github.com/rhardjono/BootcampBDML/tree/master/Modulo01%20-%20Data101/02_STG

Scripts PARTE1 - ODS

https://github.com/rhardjono/BootcampBDML/tree/master/Modulo01%20-%20Data101/03_ODS

Scripts PARTE2

https://github.com/rhardjono/BootcampBDML/tree/master/Modulo01%20-%20Data101/scripts_extra

TOTAL_REGISTROS	
17558	
TOTAL_CUSTOMER_ID	TOTAL_DISTINTOS_CUSTOMER_ID
17558	17558
TOTAL_FIRST_NAME	TOTAL_DISTINTOS_FIRST_NAME
17558	7314
TOTAL_LAST_NAME	TOTAL_DISTINTOS_LAST_NAME
17497	14577
TOTAL_IDENTIFIED_DOC	TOTAL_DISTINTOS_IDENTIFIED_DOC
17497	17498
TOTAL_GENDER	TOTAL_DISTINTOS_GENDER
17497	3
TOTAL_CITY	TOTAL_DISTINTOS_CITY
17497	82
TOTAL_ADDRESS	TOTAL_DISTINTOS_ADDRESS
17497	17439
TOTAL_POSTAL_CODE	TOTAL_DISTINTOS_POSTAL_CODE
17497	274
TOTAL_STATE	TOTAL_DISTINTOS_STATE
17497	4
TOTAL_COUNTRY	TOTAL_DISTINTOS_COUNTRY
17497	2
TOTAL_PHONE	TOTAL_DISTINTOS_PHONE
17497	17498
TOTAL_EMAIL	TOTAL_DISTINTOS_EMAIL
17497	17498
TOTAL_BIRTHDAY	TOTAL_DISTINTOS_BIRTHDAY
17497	10753
TOTAL_PROFESION	TOTAL_DISTINTOS_PROFESION
17497	196
TOTAL_COMPANY	TOTAL_DISTINTOS_COMPANY
17451	384