Informe de Calidad de Datos

Alice Caronna

Adrián Edreira Gantes

Índice

1. Introducción

La base de datos corresponde a la de una empresa de ventas de diferentes productos, con tablas para los empleados, ventas, almacenes… La falta de calidad en este conjunto de datos puede desencadenar en problemas mayores que afecten a decisiones de negocio, como pueden ser menos ventas de un producto, mal control del inventario en los almacenes o clientes y/o empleados fantasma.

También podría afectar al Data Warehouse, que al no disponer de todos los datos, o no disponer de una versión correcta de los mismo, puede afectar a que su posterior explotación y análisis afecte a las decisiones que se tomen de cara al futuro del negocio, inclusive pudiendo dejar en situación de quiebra a la empresa.

Para evitar esta situación, hemos redactado este informe con el objetivo de encontrar el máximo de errores posibles y darle una solución a todos ellos. Cabe destacar que todos los campos no reportados en el actual documento son valorados como correctos y no serán analizados. La estructura de dicho informe será la siguiente.

En primer lugar se analizará la base de datos buscando errores. El objetivo de este análisis es comprobar la calidad de los datos almacenados en cada tupla de cada tabla de la base de datos; buscando duplicados, valores nulos y/o valores incorrectos, entre otras cosas. Dentro de este análisis se explicará cómo, y en qué medida, los errores más críticos para el negocio pueden afectar en él.

Seguidamente, se estudiarán las dimensiones de calidad de la ISO 25012 según los resultados obtenidos en el análisis anterior, buscando la calidad inherente a los datos como puede ser su completitud o su consistencia, y la calidad dependiente del sistema como puede ser la precisión o la comprensibilidad. Pueden existir campos de esta ISO más complicados de valorar como pueden ser la portabilidad o la recuperabilidad debido a la ejecución de la base de datos de forma local.

Con todo esto, podremos valorar una serie de mejoras y acciones recomendadas para solucionar los errores presentes y evitar que nuevos errores puedan producirse en el futuro. De esta forma se intentará mejorar la calidad de los datos de cara a la próxima auditoría y reducir los costos que una mala calidad de los datos puede acarrear.

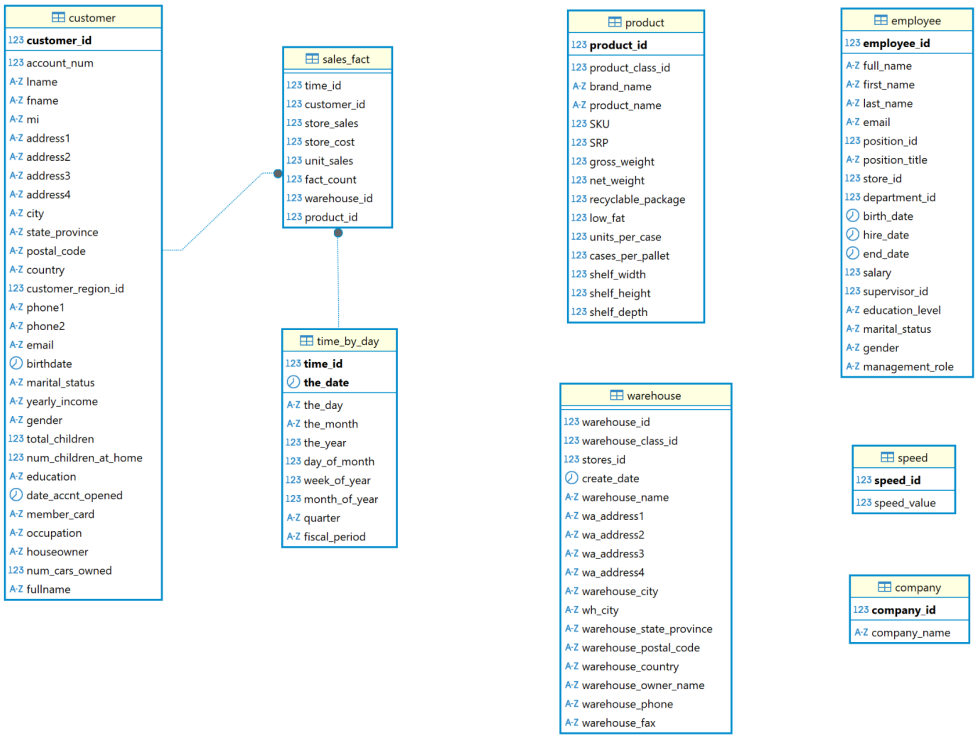
Finalmente se hará una valoración final a la base de datos teniendo una visión general de todo lo aportado en este reporte.

2. Análisis

Este análisis se realiza con el objetivo de evaluar la situación actual de la base de datos de la empresa, encontrar deficiencias y reportar su impacto en el negocio. Para ello analizaremos cada tabla de forma individual para una mayor limpieza. No obstante, algunas de estas tablas tienen columnas con claves foráneas a otras tablas, lo que será necesario comprobar para evitar problemas de consistencia.

Dos de las ocho tablas llaman especialmente la atención. Una de ellas, company, está completamente vacía y solo dispone de dos columnas, company_id y company_name. La otra, speed, tiene dos columnas llamadas speed_id y speed_value, ocupa un total de 28 MB en memoria, un 90% del total del tamaño de la base de datos, y sus valores para speed_id van de 3.000.001 a 4.000.000 y el valor de speed_value correspondiente va de 1 a 1.000.000, siendo para el id 3.000.001 el valor 1, para 3.000.002 el valor 2 y así sucesivamente.

Tras obtener el diagrama mediante la herramienta DBeaver podemos ver gran cantidad de fallos, como son falta de relaciones, de claves… Todas ellas serán revisadas con mayor atención en sus apartados correspondientes.



product

Con una simple consulta para observar los campos de esta tabla vemos dos cosas que nos llaman la atención, el nombre de los productos incluye la marca, valor que tiene su propia columna; y SKU, que corresponde al “Stock Keeping Unit”, es único para cada producto y podría usarse como clave primaria. Observemos el siguiente código creador de esta tabla:

Previsualización de SQL:

```sql
-- tbi.product definition

CREATE TABLE `product` (
  `product_class_id` int NOT NULL,
  `product_id` int NOT NULL,
  `brand_name` varchar(60) DEFAULT NULL,
  `product_name` varchar(60) NOT NULL,
  `SKU` bigint NOT NULL,
  `SRP` decimal(10,4) DEFAULT NULL,
  `gross_weight` double DEFAULT NULL,
  `net_weight` double DEFAULT NULL,
  `recyclable_package` tinyint(1) DEFAULT NULL,
  `low_fat` tinyint(1) DEFAULT NULL,
  `units_per_case` smallint DEFAULT NULL,
  `cases_per_pallet` smallint DEFAULT NULL,
  `shelf_width` double DEFAULT NULL,
  `shelf_height` double DEFAULT NULL,
  `shelf_depth` double DEFAULT NULL,
  UNIQUE KEY `i_product_id` (`product_id`),
  KEY `i_prod_brand_name` (`brand_name`),
  KEY `i_prod_class_id` (`product_class_id`),
  KEY `i_product_name` (`product_name`),
  KEY `i_product_SKU` (`SKU`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

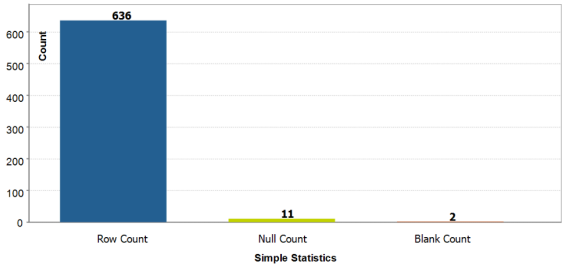
Vemos como la clave está formada por el valor “product_id” usando UNIQUE. Sin embargo, si no existe una norma específica de asignación exterior a la base de datos debería marcarse como PRIMARY KEY y asignarse sucesivamente a los nuevos productos introducidos en la tabla.

Un campo que permite nulos y no debería es la marca de un producto, ya que para la venta de los mismos podría ser necesario este valor en determinadas búsquedas o filtros de los compradores. Vemos que 11 productos tienen el nombre de marca a nulo y 2 están vacíos.

Por otra parte, hay 3 productos que tienen su nombre también vacío y uno de ellos con id = 27 se llama “kjl”. Esto también podría afectar al número de ventas de estos productos debido a que no podrán ser encontrados por su nombre.

Simple Statistics

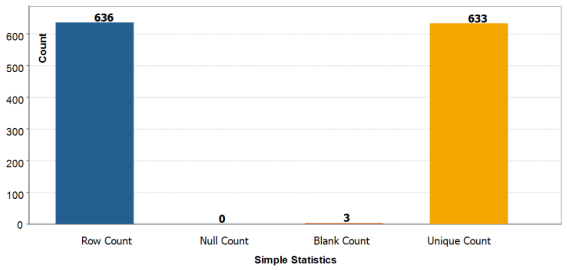
Label	Count	%
Row Count	636	100.00%
Null Count	11	1.73%
Blank Count	2	0.31%



Column: product.product_name

Simple Statistics

Label	Count	%
Row Count	636	100.00%
Null Count	0	0.00%
Blank Count	3	0.47%
Unique Count	633	99.53%



Si nos fijamos en el SRP (Sale Recommended Price), vemos que dos productos tienen este valor negativo, lo cual no es posible ya que se trata del precio recomendado de venta. Esto podría acarrear pérdidas si una página web que vende este producto lo vende por el SRP marcado en su tupla.

Valor	Count	%
9.9999	634	99.69%
-9.9999	2	0.31%

No encontramos ninguna fila que pueda considerarse duplicada ya que vemos que no hay duplicados en los valores SKU, nombre de producto o id del producto. Algunos productos tienen campos nulos, esto podría ser aceptado si sabemos que tendremos un producto nuevo pero no disponemos de toda su información o no podemos obtener ese valor concreto, aunque puede ser necesario rellenarlos en algún futuro.

El atributo “product_class_id” no podemos saber con exactitud que representa debido a la ausencia de un diccionario de datos, pero podemos intuir que es un código que identifica el tipo de producto que es. Así, por ejemplo, vemos que el 52 puede agrupar bebidas, el 30 zumos y el 19 sodas. Deberíamos comprobar que todos los nombres del producto contengan alguna de las palabras clave aceptadas en esa categoría; por ejemplo, tomando la clase con valor 1, frutos secos, buscaríamos alguno que no tuviese en su nombre de producto palabras clave como “Almonds”, “Nuts” o “Walnuts”, entre otros. De encontrar algún producto, deberemos cambiar la categoría del mismo. Esto puede afectar en las ventas de una determinada categoría si no están todos sus productos en la categoría correcta.

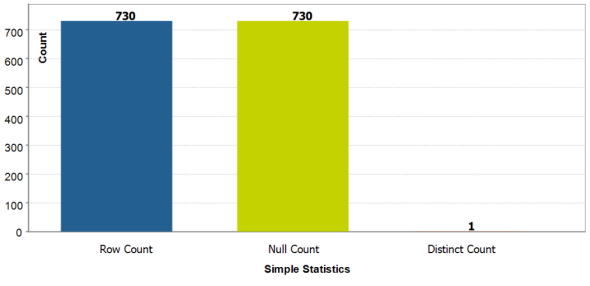
time_by_day

En esta tabla vemos que la clave primaria está formada por “time_id” y “the_date”, podríamos usar solo un campo para evitar crear nuevas tuplas con información duplicada, pues si para un mismo valor “the_date” le damos dos valores diferentes en “time_id” podríamos crear dos tuplas diferentes para guardar la misma información.

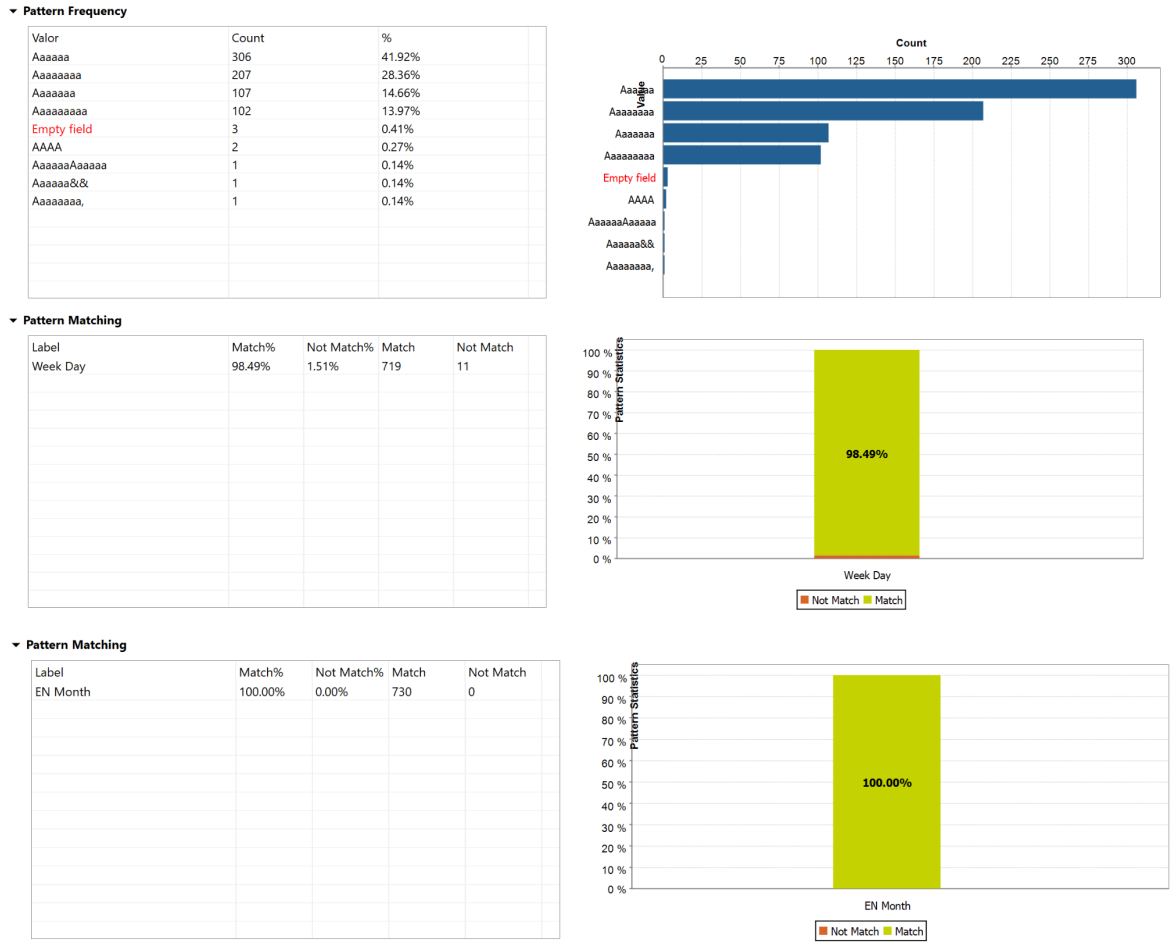
En “the_year” tenemos 3 filas con el año incorrecto, 2.997. La columna “fiscal_period” está completamente vacía, como se ve en la siguiente imagen, lo que podría dar problemas a la hora de obtener las ventas de un periodo fiscal concreto. Además, la columna “week_of_year” está completamente errónea ya que arrastra un error, pues algunas semanas de diciembre están marcadas como las primeras del año arrastrando el error a todas las demás semanas, lo que genera una gran inconsistencia e inexactitud.

Simple Statistics

Label	Count	%
Row Count	730	100.00%
Null Count	730	100.00%
Distinct Count	1	0.14%



En cuanto a los valores de las columnas, la columna “the_day” tiene 11 filas con valores incorrectos, de los cuales 5 son valores “nulos” o vacíos y el resto de escritura. En la siguiente imagen vemos que esos 11 valores, menos de un 2%, no matchean con el patrón Week Day. No obstante, todos los valores de “the_month” están correctos y cumplen el patrón “EN Month”, correspondiente al nombre del mes en inglés.



Pattern Frequency

Valor	Count	%
Aaaaaa	306	41.92%
Aaaaaaaa	207	28.36%
Aaaaaaa	107	14.66%
Aaaaaaaaa	102	13.97%
Empty field	3	0.41%
AAAA	2	0.27%
AaaaaaAaaaaa	1	0.14%
Aaaaaa&&	1	0.14%
Aaaaaaaa,	1	0.14%

Pattern Matching

Label	Match%	Not Match%	Match	Not Match
Week Day	98.49%	1.51%	719	11

Pattern Matching

Label	Match%	Not Match%	Match	Not Match
EN Month	100.00%	0.00%	730	0

sales_fact

Al observar las columnas vemos dos de ellas llamadas “store_cost” y “store_sales” que representan por cuanto se ha comprado y vendido la mercancía. Según el proceso de negocio, sería mejor mantener las adquisiciones y ventas en tablas separadas, de esta forma podremos saber dónde está cada producto que compramos, su coste o precio de venta y otros aspectos. Esto conlleva una completa ineficiencia a la hora de gestionar el inventario, pues no sabemos el total de unidades de un producto concreto.

Por otra parte, lo que más llama la atención es, sin duda alguna, la falta de relación entre la tabla “sales_fact” con las tablas “warehouse” y “product”. No podemos saber con certeza si los campos “warehouse_id” y “product_id” hacen referencia a los id en las tablas nombradas con anterioridad, pero suponemos que no tendría sentido almacenar esta información si no está relacionada entre si. De esta forma, concluimos que esta es la tabla con mayor inconsistencia, pues más de la mitad de las tuplas referencian a productos inexistentes, y la mayoría, a almacenes inexistentes. Podemos obtener este resultado con la sentencia SQL observada en la siguiente imagen:

```sql
SELECT 'product' AS tipo, p.product_id AS id, COUNT(*) AS total
FROM sales_fact sf
LEFT JOIN product p ON sf.product_id = p.product_id
GROUP BY p.product_id

UNION ALL

SELECT 'warehouse' AS tipo, w.warehouse_id AS id, COUNT(*) AS total
FROM sales_fact sf
LEFT JOIN warehouse w ON sf.warehouse_id = w.warehouse_id
GROUP BY w.warehouse_id
```

tipo	id	total
warehouse		2.481
product		1.452

Esto se debe a la falta de sentencias “CONSTRAINT” en la creación de la tabla, por lo que no se impide que se cumpla la integridad referencial, como vemos en la siguiente imagen:

Previsualización de SQL:

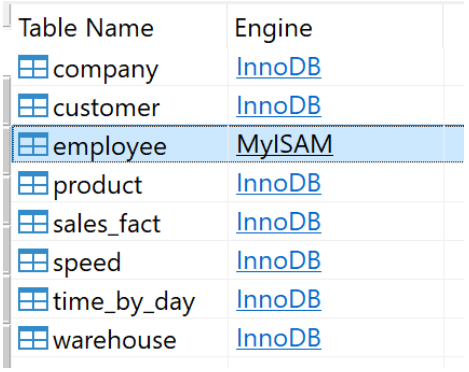
```sql
-- tbi.sales_fact definition

CREATE TABLE `sales_fact` (
  `time_id` int NOT NULL,
  `customer_id` int NOT NULL,
  `store_sales` decimal(10,4) NOT NULL,
  `store_cost` decimal(10,4) NOT NULL,
  `unit_sales` decimal(10,4) NOT NULL,
  `fact_count` int NOT NULL,
  `warehouse_id` int DEFAULT NULL,
  `product_id` int DEFAULT NULL,
  KEY `customer_id` (`customer_id`),
  KEY `time_id` (`time_id`),
  CONSTRAINT `sales_fact_ibfk_1` FOREIGN KEY (`customer_id`) REFERENCES
`customer` (`customer_id`),
  CONSTRAINT `sales_fact_ibfk_2` FOREIGN KEY (`time_id`) REFERENCES
`time_by_day` (`time_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Además vemos que “warehouse_id” y “product_id” pueden ser nulos, lo que ocasionará problemas en el presente y en el futuro. En el presente, una incorrecta gestión del inventario, y en el futuro, la imposibilidad de estudiar las ventas.

employee

Esta tabla, a diferencia del resto que usa el motor InnoDB, usa el motor MyISAM; lo que consideramos meramente como curiosidad de cara a este análisis. Esto podría deberse a alguna necesidad de mayor velocidad de lectura, muchas consultas o ahorro de espacio y no necesitar soporte para transacciones ni claves foráneas.



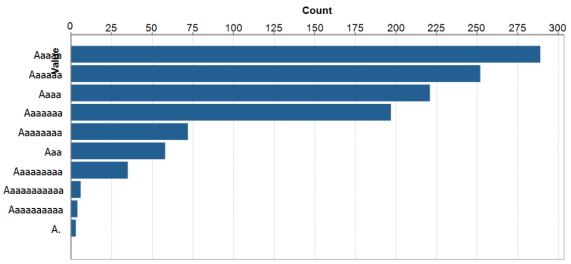
Al observar la tabla lo primero que llama especialmente la atención es la columna “last_name”, pues es igual a la columna “first_name” para todas sus tuplas. Sin embargo, vemos como la columna “full_name” podría mantener la información real. A cerca de la columna “last_name”, vemos que más de un 23% de las tuplas tienen el valor “********”. Además, en la columna “first_name” vemos tres personas con la inicial de su nombre, por lo que es imposible obtener su nombre completo ya que en “full_name” tenemos el mismo problema. Esto imposibilita identificar a un empleado por su nombre con facilidad.

Column: employee.first_name

Simple Statistics

Pattern Frequency

Valor	Count	%
Aaaaa	289	25.02%
Aaaaaa	252	21.82%
Aaaa	221	19.13%
Aaaaaaa	197	17.06%
Aaaaaaaa	72	6.23%
Aaa	58	5.02%
Aaaaaaaaa	35	3.03%
Aaaaaaaaaa	6	0.52%
Aaaaaaaaa	4	0.35%
A.	3	0.26%

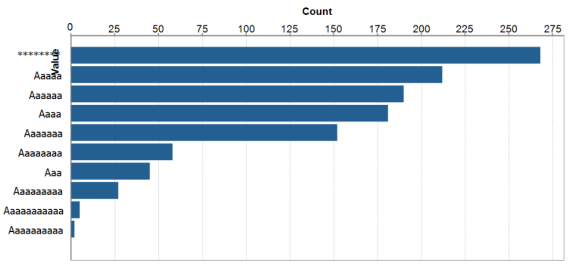


Column: employee.last_name

Simple Statistics

Pattern Frequency

Valor	Count	%
********	268	23.20%
Aaaaa	212	18.35%
Aaaaaa	190	16.45%
Aaaa	181	15.67%
Aaaaaaa	152	13.16%
Aaaaaaaa	58	5.02%
Aaa	45	3.90%
Aaaaaaaaa	27	2.34%
Aaaaaaaaaa	5	0.43%
Aaaaaaaaa	2	0.17%



La probabilidad de duplicados en esta tabla es baja ya que la columna “full_name” tiene un 99.8% de valores únicos. No obstante, el caso de la persona con Id 131 y 516 llama la atención, pues estas dos tuplas tienen valores muy similares en todas las columnas. Sobre estas dos tuplas destacamos el nombre completo y su género, pues “Mary Smith” en una tupla tiene género femenino y en otra masculino, lo que provoca incertidumbre acerca de la validez de la información.

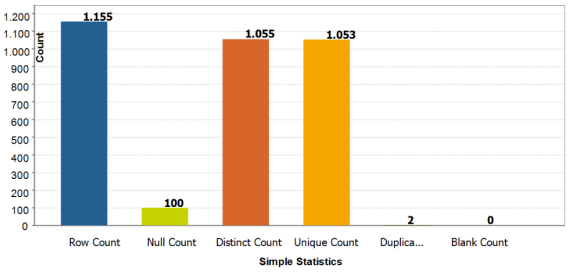
En relación a las columnas “position_id” y “position_title” podemos afirmar que introducen mucha redundancia al replicar información, lo que puede dar problemas de inconsistencia si no se introduce la información de forma correcta. Además, vemos que el id 10 no está en uso y el 19 no tiene un título asignado a esa posición, pues está a nulo y tenemos 32 empleados en dicha posición. A mayores, las columnas “store_id” y “departament_id” no tienen ningúna tabla a la que hagan referencia en la base de datos, por lo que no sabemos donde está trabajando un empleado al no tener más información de las tiendas o los departamentos.

Por otra parte, la columna “email” tiene un alto porcentaje de nulos, y los registrados tienen espacios en las direcciones de correo electrónico, como vemos en la gráfica de patrones. Esto puede desencadenar problemas a la hora de enviar correos a nuestros empleados, como pueden ser avisos o horarios laborales.

Column: employee.email

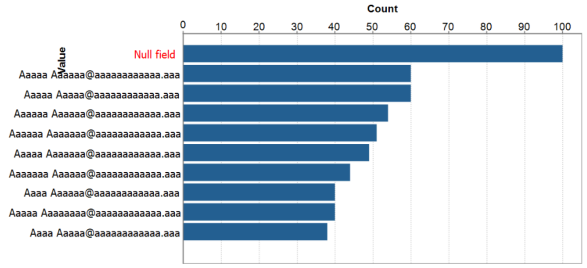
Simple Statistics

Label	Count	%
Row Count	1155	100.00%
Null Count	100	8.66%
Distinct Count	1055	91.34%
Unique Count	1053	91.17%
Duplicate Count	2	0.17%
Blank Count	0	0.00%



Pattern Frequency

Valor	Count	%
Null field	100	8.66%
Aaaaa Aaaaa@aaaaaaaaaaa.aaa	60	5.19%
Aaaaa Aaaaa@aaaaaaaaaaa.aaa	60	5.19%
Aaaaaa Aaaaaa@aaaaaaaaaaa.aaa	54	4.68%
Aaaaaa Aaaaaaa@aaaaaaaaaaa.aaa	51	4.42%
Aaaaa Aaaaaaa@aaaaaaaaaaa.aaa	49	4.24%
Aaaaaaa Aaaaaa@aaaaaaaaaaa.aaa	44	3.81%
Aaaa Aaaaaa@aaaaaaaaaaa.aaa	40	3.46%
Aaaaa Aaaaaaaa@aaaaaaaaaaa.aaa	40	3.46%
Aaaa Aaaaa@aaaaaaaaaaa.aaa	38	3.29%



Si nos fijamos en los valores de la columna “marital_status” vemos que existen 3: S (Single), M (Married) y 3(Suponemos que equivale a: No se sabe). Usar este 3 cuando no conocemos el valor es incorrecto y sería aceptado usar NULL en este caso para registrar esa falta de conocimiento, al igual que sucede con la columna “management_role” para los empleados que no tienen ningún rol de gestión.

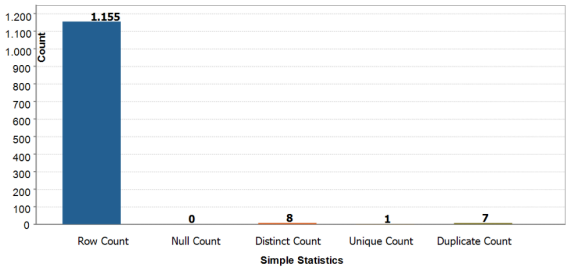
Sobre la columna “supervisor_id”, existe un empleado, probablemente el dueño, con un 0 en esta columna para indicar que no tiene supervisor. Esto es incorrecto ya que podría existir algún empleado con ese id, lo más correcto sería usar NULL; el resto de empleados tienen un supervisor que existe en la base de datos.

También llama la atención tener empleados nacidos en 1912 sin fecha de fin de contrato “end_date”, no sabemos si están vivos o si siguen trabajando con nosotros ya que dicha columna está completamente vacía. Esto generará problemas a la hora de obtener la plantilla real actual. Además, la fecha de contratación en la mayoría de las tuplas es la misma, tan solo 8 fechas distintas de las cuales 1 es única para una tupla, el resto comparten fecha de contratación. Esto también sucede en la fecha de nacimiento con 43 únicas y 9 duplicadas, lo que reduce en gran medida la credibilidad de la tabla.

Column: employee.hire_date

Simple Statistics

Label	Count	%
Row Count	1155	100.00%
Null Count	0	0.00%
Distinct Count	8	0.69%
Unique Count	1	0.09%
Duplicate Count	7	0.61%

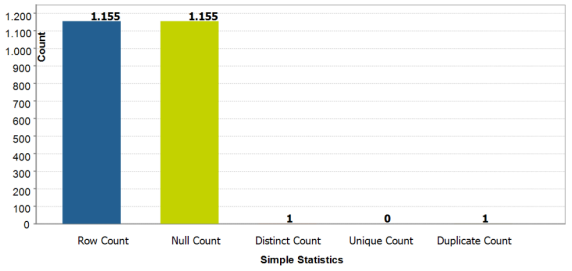


Pattern Frequency

Column: employee.end_date

Simple Statistics

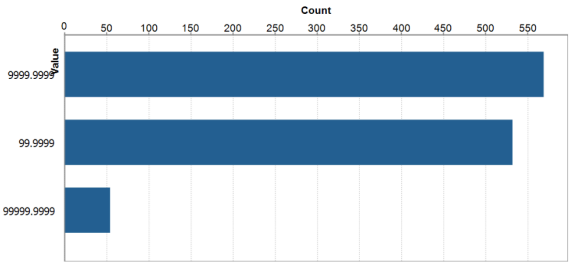
Label	Count	%
Row Count	1155	100.00%
Null Count	1155	100.00%
Distinct Count	1	0.09%
Unique Count	0	0.00%
Duplicate Count	1	0.09%



Por último, destaca el amplio rango de valores en la columna “salary”, con salarios de 2 a 5 dígitos. Probablemente unos representen el salario por horas, otros por meses y otros por años. Esta forma de registrar la información es incorrecta, pues impide, entre otras cosas, saber el total de salarios a pagar en un mes. En el siguiente gráfico podemos observar ese desequilibrio entre salarios.

Pattern Frequency

Valor	Count	%
9999.9999	569	49.26%
99.9999	532	46.06%
99999.9999	54	4.68%

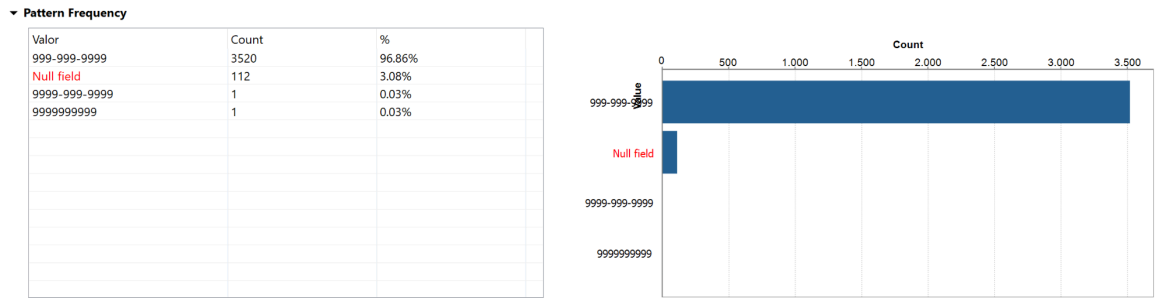


customer

Lo primero de lo que nos percatamos es de la presencia de un posible duplicado, pues los clientes con el id 0 y 1 tienen datos muy similares, e idénticos en la columna “account_num”y “fullname” entre otras, por lo que suponemos que es una tupla duplicada. Además, salvo este caso, “account_num” podría ser usada como clave primaria, con el respectivo cambio en la tabla de ventas.

En relación a los nombres, vemos que existe un pequeño porcentaje de nulos y vacíos en la columna para apellidos y un alto porcentaje de nulos en la columna “mi”, que suponemos que es la inicial del segundo nombre al no disponer de un diccionario de datos. En la columna de nombres de pila vemos como casi un 5% de los nombres son diminutivos, como Ed o Jo, lo que afecta a la exactitud sintáctica. Finalizando con los nombres, vemos que en la columna “fullname” existen 11 duplicados. Analizándolo más a fondo podemos ver que salvo el caso comentado anteriormente, no se trata de información duplicada, ya que tienen ubicaciones, email e información diferente entre ellos.

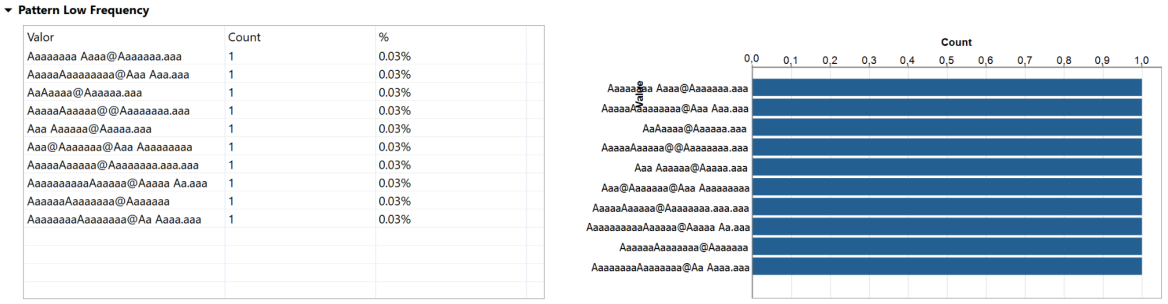
Relacionado con las formas de contacto, vemos que la columna “phone1” tiene errores de formato y un 3% de ellos con valor nulo, como vemos en la “imagen 1”. Valorando los clientes con teléfono 1 nulo, nos damos cuenta que la mayoría de ellos tienen un teléfono 2, lo que debería de ser al revés, siendo obligatorio el primero y el segundo opcional. Acerca de la columna “email” vemos que los formatos de algunos ellos son incorrectos, con espacios en las direcciones electrónicas, más arrobas de los permitidos o sin punto, como vemos en la “imagen 2”.



Pattern Frequency

Valor	Count	%
999-999-9999	3520	96.86%
Null field	112	3.08%
9999-999-9999	1	0.03%
9999999999	1	0.03%

imagen 1

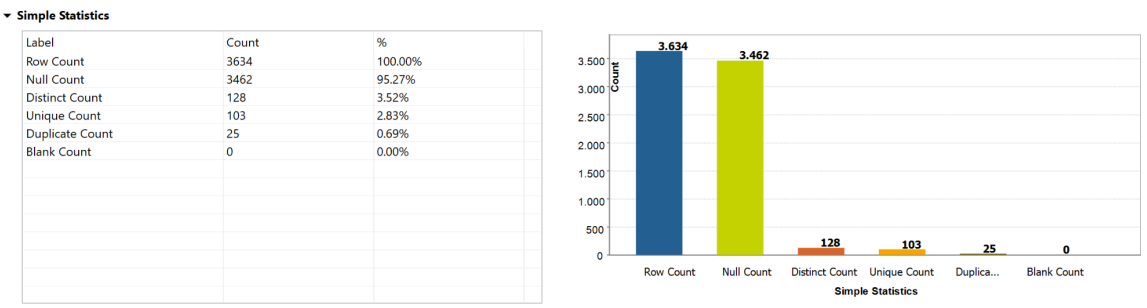


Pattern Low Frequency

Valor	Count	%
Aaaaaaaa Aaaa@Aaaaaaa.aaa	1	0.03%
AaaaaAaaaaaaaa@Aaa Aaa.aaa	1	0.03%
AaAaaaa@Aaaaaa.aaa	1	0.03%
AaaaaAaaaaa@@Aaaaaaaa.aaa	1	0.03%
Aaa Aaaaaa@Aaaaa.aaa	1	0.03%
Aaa@Aaaaaaa@Aaa Aaaaaaaaa	1	0.03%
AaaaaAaaaaa@Aaaaaaaa.aaa.aaa	1	0.03%
AaaaaaaaaaAaaaaa@Aaaaa Aa.aaa	1	0.03%
AaaaaaAaaaaaaa@Aaaaaaa	1	0.03%
AaaaaaaaAaaaaaaa@Aa Aaaa.aaa	1	0.03%

imagen 2

Acerca de las direcciones, vemos que la gran mayoría de tuplas tienen los valores de las columnas para direcciones 2, 3 y 4 con nulos (imagen 1). Por otra parte, las columnas “city”, “state_province”, “country” y “postal_code” tienen variedad de errores ortográficos, nulos y diferentes formatos (imagen 2). Esto puede conllevar a ventas que no llegan correctamente a su destino y por consiguiente, pérdidas en gastos por reembolsos. Además, la columna “customer_region_id” sugiere que podría existir una tabla con ids para cada región y de esta manera evitar redundancia en la tabla de clientes, como sucede con el id 99 que representa a la ciudad Oak Bay del estado BC, en Canadá. Registrarlo de la forma actual puede acarrear problemas de consistencia en el futuro si no se registran correctamente los datos.



Simple Statistics

Label	Count	%
Row Count	3634	100.00%
Null Count	3462	95.27%
Distinct Count	128	3.52%
Unique Count	103	2.83%
Duplicate Count	25	0.69%
Blank Count	0	0.00%

imagen 1

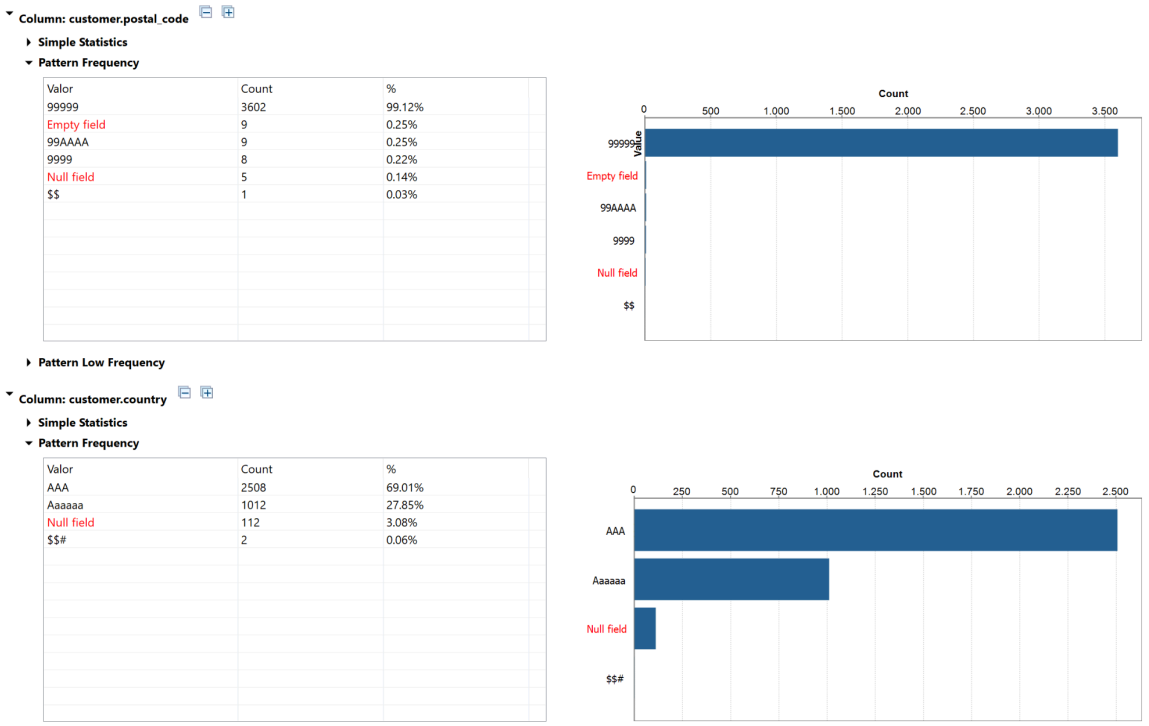
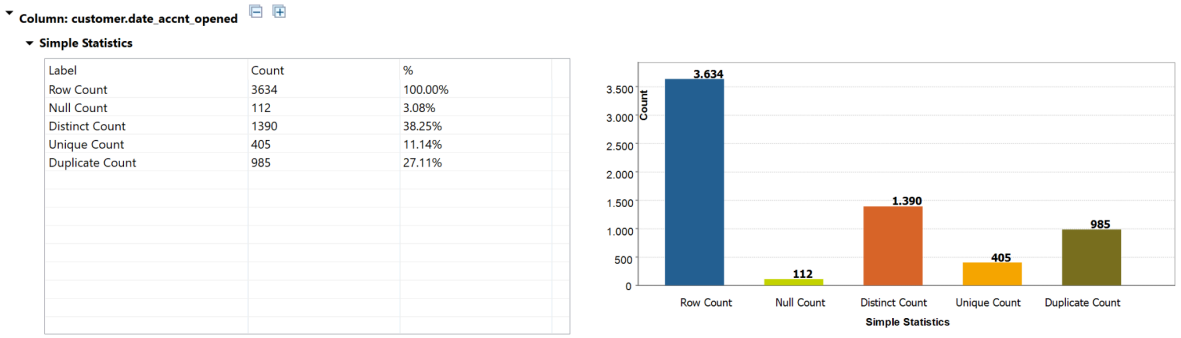


imagen 2

En cuanto a otras columnas, “bithdate” vemos que 100 personas no tienen fecha de cumpleaños registrada y 1 persona tiene una “H” en “marital_status”. En la columna “gender” sucede algo similar, tenemos los valores M y F y a mayores 9 filas con el valor nulo y otras 10 con el valor 2, lo que no sabemos si significa ambos, además de la tupla con la “H” anteriormente mencionada que vuelve a salir en esta columna. Esto conlleva a que no dispongamos de información que podría ser necesaria para promociones o descuentos.

Por último, al igual que sucede en la tabla de empleados, llama la atención que la mayoría de los clientes se han registrado en la base de datos el mismo día. En “date_accnt_opened” vemos que el número de valores únicos es mucho menor al total de tuplas de la tabla y un porcentaje demasiado elevado de nulos para este campo, pues debería ser el día en el que dicha tupla fue creada. Esto reduce la credibilidad de la tabla, y ahora, al ver que sucede repetidas veces, de la base de datos en general.



warehouse

Lo que más alerta de esta última tabla es la ausencia de un campo id para cada entidad. Aunque existe la columna “warehouse_id” esta no está declarada como clave primaria, lo que permite duplicar ids con todos los problemas que ello conlleva. No solo a nivel de base de datos, si no que no podemos saber cuántos almacenes tenemos ni si la información de ellos es correcta al existir la posibilidad de duplicación con información distinta. En la siguiente imagen vemos el código de creación de esta tabla.

Previsualización de SQL:

```sql
-- tbi.warehouse definition

CREATE TABLE `warehouse` (
  `warehouse_id` int NOT NULL,
  `warehouse_class_id` int DEFAULT NULL,
  `stores_id` int DEFAULT NULL,
  `create_date` date DEFAULT '0000-00-00',
  `warehouse_name` varchar(60) DEFAULT NULL,
  `wa_address1` varchar(30) DEFAULT NULL,
  `wa_address2` varchar(30) DEFAULT NULL,
  `wa_address3` varchar(70) DEFAULT NULL,
  `wa_address4` varchar(30) DEFAULT NULL,
  `warehouse_city` varchar(30) DEFAULT NULL,
  `wh_city` varchar(30) DEFAULT NULL,
  `warehouse_state_province` varchar(30) DEFAULT NULL,
  `warehouse_postal_code` varchar(30) DEFAULT NULL,
  `warehouse_country` varchar(30) DEFAULT NULL,
  `warehouse_owner_name` varchar(30) DEFAULT NULL,
  `warehouse_phone` varchar(30) DEFAULT NULL,
  `warehouse_fax` varchar(30) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

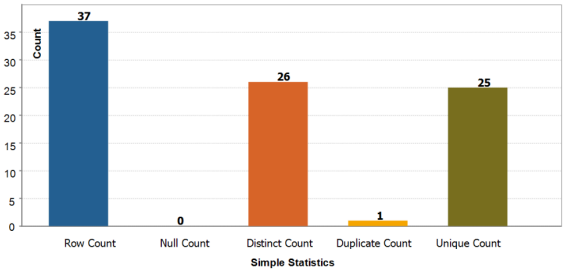
Relacionado con la imagen anterior, nos damos cuenta de que campos importantes para los almacenes como la ubicación, el código postal o la información de contacto permiten nulos, por lo que podría suceder que no pudiéramos localizar o contactar con el personal de dichos almacenes.

En relación a los duplicados, vemos que el almacén con id 24 está decuplicado, es decir, existen 10 tuplas con el mismo id e información muy similar en todas estas tuplas. De las 37 tuplas de la tabla, 26 valores de los id son distintos y uno de ellos, el 24, está duplicado como vemos en la siguiente imagen.

Column: warehouse.warehouse_id

Simple Statistics

Label	Count	%
Row Count	37	100.00%
Null Count	0	0.00%
Distinct Count	26	70.27%
Duplicate Count	1	2.70%
Unique Count	25	67.57%

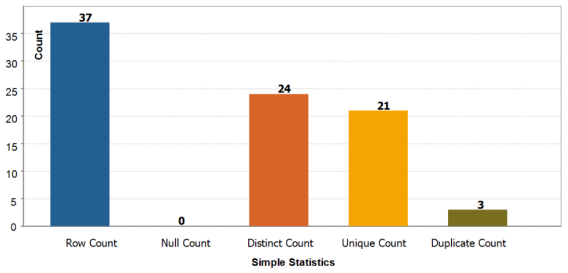


Existen 2 campos que son “warehouse_class_id” y “stores_id” que sugieren que existen 2 tablas para clases de almacenes y tiendas, pero no existen tales tablas en la base de datos y consecuentemente no son claves foráneas. No tenemos ningún tipo de información extra acerca de estas columnas, por lo que no podemos analizarlas en profundidad. No obstante, el nombre en plural para la columna “stores_id” sugiere que sería multivaluado, por ejemplo para las tiendas que son suministradas por un almacén, aun teniendo solo un valor.

Column: warehouse.stores_id

Simple Statistics

Label	Count	%
Row Count	37	100.00%
Null Count	0	0.00%
Distinct Count	24	64.86%
Unique Count	21	56.76%
Duplicate Count	3	8.11%

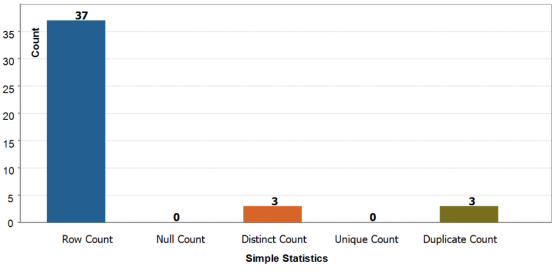


En la gráfica anterior podemos ver como tres de los valores están duplicados para “stores_id”. Analizando las filas de esas duplicaciones descubrimos que existen 2 duplicaciones a mayores de la nombrada anteriormente que son el almacén con id 25 y 19, y el almacén con id 27 y 17. Además, vemos que todos los valores que están en la columna “warehouse_id” están replicados en la columna “stores_id”, salvo las duplicaciones anteriores. Esto sugiere que es otra forma de nombrar al propio almacén, lo que dificulta la comprensibilidad de la base de datos.

También llama la atención que todos los almacenes han sido creados en días idénticos, solo existen 3 valores distintos por lo que podemos asegurar que esta información está falsificada. Además, al disponer de la tabla “time_by_day” podríamos crear una clave foránea a dicha tabla para disponer de más información por si fuese necesaria.

Simple Statistics

Label	Count	%
Row Count	37	100.00%
Null Count	0	0.00%
Distinct Count	3	8.11%
Unique Count	0	0.00%
Duplicate Count	3	8.11%

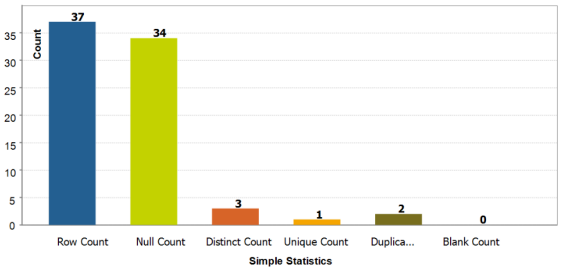


En cuanto a las direcciones de los almacenes, vemos que las columnas “address” 2, 3 y 4 están prácticamente vacías, como sucede en la tabla de clientes. Existen varias tuplas con información errónea en alguna de estas columnas, por lo que podemos considerarlas descartables. Por otra parte, vemos que existen 2 columnas idénticas, “warehouse_city” y “wh_city”, que son usadas para registrar el mismo valor en ambas, lo que no tiene ningún tipo de utilidad ni sentido aparente.

Column: warehouse.wa_address2

Simple Statistics

Label	Count	%
Row Count	37	100.00%
Null Count	34	91.89%
Distinct Count	3	8.11%
Unique Count	1	2.70%
Duplicate Count	2	5.41%
Blank Count	0	0.00%



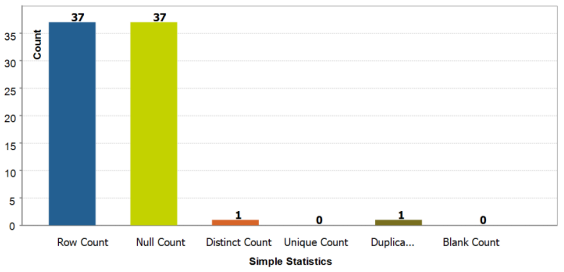
Pattern Frequency

Pattern Low Frequency

Column: warehouse.wa_address3

Simple Statistics

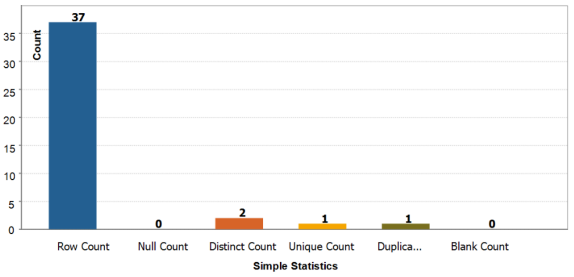
Label	Count	%
Row Count	37	100.00%
Null Count	37	100.00%
Distinct Count	1	2.70%
Unique Count	0	0.00%
Duplicate Count	1	2.70%
Blank Count	0	0.00%



En concordancia con las direcciones, vemos que el nombre del estado donde se encuentran y su país correspondiente tienen formatos diferentes como sucede en la tabla de clientes. Además, todos los códigos postales son el “55555” con la excepción que una tupla que tiene un error de formato en este campo, pues solo tiene 3 dígitos.

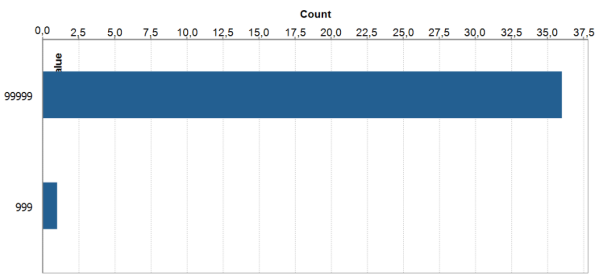
Simple Statistics

Label	Count	%
Row Count	37	100.00%
Null Count	0	0.00%
Distinct Count	2	5.41%
Unique Count	1	2.70%
Duplicate Count	1	2.70%
Blank Count	0	0.00%



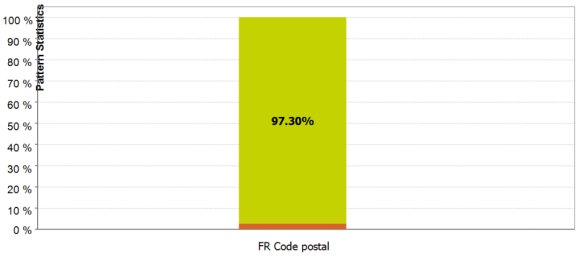
Pattern Frequency

Valor	Count	%
99999	36	97.30%
999	1	2.70%



Pattern Matching

Label	Match%	Not Match%	Match	Not Match
FR Code postal	97.30%	2.70%	36	1



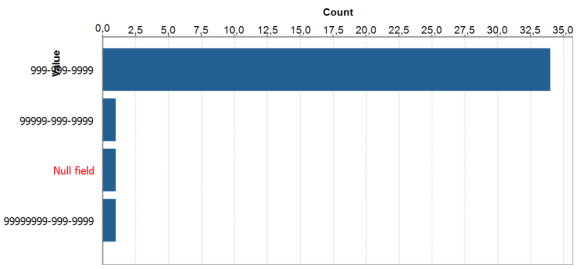
Por último, observamos que la columna “warehouse_owner_name” está completamente vacía, esto tiene sentido si somos dueños de todos los almacenes y puede darse el caso de alquilar alguno, pero deberíamos poner que dicho almacén es nuestro para reflejar esa información en la base de datos. Además, la información de contacto de nuestros almacenes en gran parte está errónea. Tanto en la columna de teléfono como en la columna de fax vemos tuplas con valores nulos o formatos incorrectos, lo que nos impide contactar con nuestros almacenes en caso de necesidad de reabastecimiento.

Column: warehouse.warehouse_phone

Simple Statistics

Pattern Frequency

Valor	Count	%
999-999-9999	34	91.89%
99999-999-9999	1	2.70%
Null field	1	2.70%
99999999-999-9999	1	2.70%



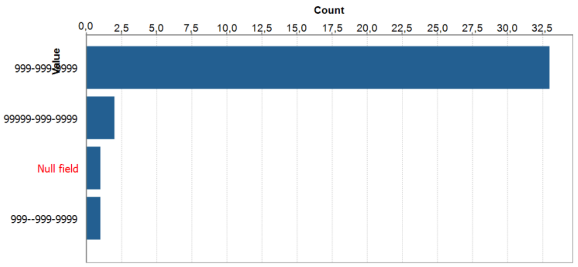
Pattern Low Frequency

Column: warehouse.warehouse_fax

Simple Statistics

Pattern Frequency

Valor	Count	%
999-999-9999	33	89.19%
99999-999-9999	2	5.41%
Null field	1	2.70%
999--999-9999	1	2.70%



3. ISO 25012

De los 15 campos que contempla la ISO 25012, algunos de ellos no podremos estudiarlos a la perfección debido a diversos factores. Aquellos campos son los siguientes:

- Accesibilidad: no disponemos de tecnologías de apoyo, como APIs REST, para valorar este aspecto.
- Confidencialidad: tenemos acceso a todos los datos al tener la base de datos de forma local y no disponemos de un backend que acceda a dicha base de datos, por lo que no podemos valorar la confidencialidad.
- Eficiencia: por los mismos motivos que la confidencialidad, no podemos valorar la eficiencia de la base de datos.
- Trazabilidad: no disponemos de un histórico de cambios, sin embargo, será necesario hacerlo tras cambiar los fallos detectados en este análisis.
- Disponibilidad: al correr la base de datos de forma local no tenemos acceso al servidor ni podemos hacer pruebas de disponibilidad.
- Portabilidad: necesitaríamos diversidad de sistemas y arquitecturas para comprobar la portabilidad.
- Recuperabilidad: no disponemos de backups, por lo que no podemos valorar la recuperabilidad de la base de datos.

El resto de campos de la ISO 25012 pueden ser valorados con mayor facilidad gracias al análisis anteriormente realizado. Estos son los resultados que hemos obtenido:

Exactitud

Aunque la mayoría de valores representan fielmente la realidad, hemos comprobado que la exactitud es bastante deficiente. Observamos que existen fallos sintácticos en los apellidos de los empleados o en la semana del año, y semánticos, como en los países de los clientes o en los nombres de los días, lo que reduce la exactitud general de la base de datos. Algunos de ellos son Weak Accuracy Errors, ya que no afectan a la identidad de la tupla, como puede ser el nombre de un día, y otros son Strong Accuracy Errors, como el apellido de los empleados o ids de almacenes, ya que dos personas con el mismo nombre tienen el mismo apellido o dos almacenes tienen el mismo id. Respecto a la tabla de empleados como la de días, su porcentaje de exactitud, debido a los errores anteriores, tienen un porcentaje de exactitud del 0%. No obstante, la tabla de productos tiene más de un 95% de exactitud.

Completitud

Tras observar la tabla de clientes como la de almacenes vemos que gran parte de las tuplas tienen atributos vacíos, esto es un problema de “Attribute Completeness” ya que consultas realizadas para obtener la dirección de un cliente o almacén para un envío puede tener espacios en blanco. También destacar la falta de valores en la columna “fiscal_period” en la tabla “time_by_day”, ya que está completamente vacía. Esta última, al igual que la columna para guardar el dueño de un almacén o la fecha de fin de contrato de un empleado, tienen un porcentaje de completitud por atributo del 0% en las columnas correspondientes.

Otro aspecto a valorar es la “Entity Completeness”. En este caso, si nos basamos en la tabla “sales_fact” vemos que en las tablas “product” y “warehouse” faltan muchas entidades. Esto nos da un 30.5% de completitud en la tabla de productos y un 1.5% de completitud en la tabla de almacenes lo que sin duda alguna perjudica en gran medida a la base de datos de forma general.

Consistencia

Si observamos el código de creación de las tablas nos damos cuenta de la falta de reglas que existen en todas las tablas. No tan solo con la falta de claves primarias o foráneas, sino permitiendo valores, nulos o formatos que no deberían ser aceptados en ciertos atributos. Además, no sería de extrañar que alguna dirección no existiese en la región almacenada, lo que afectaría a una dependencia funcional; pero no lo podemos afirmar ya que no hemos podido analizar este aspecto a la perfección.

Credibilidad

Relacionado con la exactitud de los datos y al delegar sobre humanos la introducción de los mismos (en el caso actual, administrativos) no podemos asegurar que no se cometan errores, por lo que la credibilidad actual de los datos es bastante baja. Es necesario un software que maneje el acceso a la base de datos, así como normas que impidan la introducción de datos inválidos y un histórico de cambios para mejorar la trazabilidad. A medida que analizamos las tablas nuestra credibilidad sobre los mismos iba disminuyendo a causa de posibles duplicados, campos vacíos, valores erróneos… Un ejemplo pueden ser los apellidos de nuestros empleados, los precios de algunos productos o la tabla de ventas de forma general. Por todo esto, consideramos que la base de datos actual tiene una credibilidad baja.

Actualidad

Según la tabla “time_by_day”, los datos más recientes son del año 1998, por lo que podemos afirmar que la base de datos está completamente desactualizada. Por otra parte, el cliente más mayor es del año 1910, lo que no tiene mucho sentido seguir registrando si no se trata de un data warehouse. Es posible que este valor sea real ya que hay más clientes con esa fecha, pero muy probablemente mantengamos información de gente fallecida que puede no ser necesaria. Además tenemos empleados sin fecha de fin de contrato y ninguna venta posterior a 1998, por lo que podemos afirmar que la actualidad de la base de datos es nula.

Conformidad

Vemos una gran inconsistencia en relación a la forma de nombrar ciertos atributos como pueden ser los nombres de los países y las provincias, pues para Estados Unidos y Canadá se usan siempre siglas y para México el nombre completo. Además, ciertas fechas tienen formatos diferentes según la tabla y/o la columna, no siguen la norma ISO para el nombrado de fechas. Debido a la falta de estándares, denotamos como baja la conformidad de la base de datos actual.

Precisión

A lo largo de todas las tablas que hemos analizado hemos ido viendo que en gran parte de ellas se afecta a la precisión: apellidos de empleados, semanas del año, precios de productos, códigos postales de almacenes… Por lo que podemos concluir que de forma general la base de datos tiene una precisión escasa.

Comprensibilidad

La falta de un diccionario de datos y documentación afecta gravemente a este apartado, pues no podemos comprender a la perfección todos los campos de la base de datos. Aunque ciertas columnas sean auto descriptivas, SKU o SRP en la tabla de productos pueden llevar a confusión; así como fact_count, store_sales o store_cost en la tabla de ventas o las dos columnas para registrar la ciudad de un almacén. Por ello, la comprensibilidad actual de la base de datos es baja y se debe mejorar mediante documentación.

4. Acciones recomendadas

Tras valorar la calidad de la base de datos como muy baja, hemos realizado una serie de recomendaciones para mejorar la calidad de los datos de cara al futuro. Para realizar estos cambios será interesante mantener una historial de todos ellos para mantener la trazabilidad de los propios datos.

De forma general, es imperativo construir reglas “CONSTRAINT” en la mayoría de tablas, pues muchos problemas tanto de exactitud como de consistencia serían solucionados gracias al uso de claves foráneas correctamente definidas o valores numéricos que deben ser positivos no deben aceptar negativos, así como muchas otras columnas que no deberían aceptar nulos. Acerca de las tablas “company” y “speed”, consideramos que deberían ser borradas, pues tras investigarlas no encontramos ningún aporte especial a la base de datos, ninguna relación con las demás tablas y ocupan más del 95% del espacio total que ocupa en memoria la base de datos.

product

En esta tabla sería interesante valorar los siguientes puntos:

- Convertir el SKU en clave primaria, pues es único para cada producto. No sabemos si existe una necesidad externa de tener un id diferente, pero si no es el caso podremos aplicar este cambio.
- No permitir nulos en la marca, ya que todos los productos tienen una marca y es un valor del que se tiene conocimiento.
- Corregir valores erróneos, como son los nombres, SRPs… También se puede valorar quitar del nombre del producto la marca ya que tenemos una columna dedicada para ello.
- CONSTRAINT para SRP, con el cumplimiento de que el SRP debe ser mayor que 0, de esta forma se evitan errores de modificación, creación…
- Nueva tabla para clases de productos, pues solo tenemos un id y nada de información de cada categoría por lo que sería interesante crear una nueva tabla para registrar este tipo de información.

time_by_day

Esta tabla debería ser accedida únicamente a través de un backend diseñado para ella, de esta forma todos los errores obtenidos durante el análisis serían solucionados. Pues tanto el período fiscal, los días con el año mal registrado o la semana del año no introducirán datos erróneos en la base de datos, ya que de forma automática podríamos obtener esa información únicamente con la fecha y guardar la información necesaria en la base de datos. Por otra parte, sería interesante crear un único id para cada tupla, el cual podría ser la fecha completa.

sales_fact

Lo más importante de solucionar en esta tabla es la integridad referencial, con el uso de claves foráneas a los respectivos ids en las tablas correspondientes, como sucede con “customer” y “time_by_day”. También deberíamos impedir nulos en dichas claves foráneas, así como el uso de CONSTRAINT, impidiendo valores negativos, en campos como la cantidad, el precio, el coste…

Por último, cabe remarcar la importancia de crear dos tablas diferentes para las ventas y las adquisiciones, esto mejorará en gran medida nuestra base de datos en diferentes campos de la ISO 25012, ya que aumentará la limpieza y será más entendible la información residente en ellas.

employee

La serie de cambios a realizar en esta tabla son los siguientes:

- Corregir los apellidos de los empleados y valorar eliminar la columna “full_name” al tratarse de una columna compuesta por los valores de “first_name” y “last_name”.
- Comprobar la posible duplicación de Mary Smith.
- Todas las fechas podrían ser referenciadas en la tabla “time_by_day”, además se debe solucionar la duplicidad en las fechas de nacimiento y contratación y añadir las fechas de fin de contrato donde corresponda.
- Dar un formato específico a los correos electrónicos, ya que no pueden contener espacios en dichas direcciones.
- Crear tablas para las posiciones en la empresa, así como añadir a la base de datos, o crearlas si no existen, las tablas correspondientes a las tiendas y a los departamentos creando la correspondiente clave foránea. De esta forma se evitará la redundancia.
- Las columnas “marital_status” y “supervisor_id” deberían aceptar nulos, pues no es correcto asignar un valor no aceptado al no ser un objetivo para la creación de un data warehouse.
- Designar una métrica específica para los salarios, como puede ser por hora o por mes.
- Revisar todos los campos en los que es obligatorio algún valor como pueden ser las fechas de contratación o el salario y establecer la restricción “NOT NULL” en ellos.

customer

Existe gran cantidad de cambios a realizar en la tabla actual para mejorar su calidad. La lista de cambios es la siguiente:

- Es de vital importancia dar formatos a campos como emails y teléfonos, entre otros, pues debemos poder asegurarnos de que contactaremos con el cliente correctamente.
- Algunos campos como “houseowner” o “marital_status” deberían ser binarios para evitar problemas de consistencia.

- Se debe revisar la tupla duplicada con id 0 y 1, pues es muy probable que sea la misma persona y debemos saber a quien se le asigna una venta.
- Corregir los nombres de nuestros clientes y valorar eliminar la columna “fullname” al tratarse de un campo compuesto.
- “account_num” podría ser valorada como clave primaria para esta tabla.
- Las tres columnas para registrar direcciones (es decir, las que no son “address1”) deberían ser eliminadas ya que no aportan información crítica y podría agregarse en el valor de “address1”.
- Crear una tabla para registrar las regiones de nuestros clientes y usar la correspondiente clave foránea para ello, evitando inconsistencias y reduciendo la redundancia.
- La columna “phone1” no debe permitir nulos y el segundo teléfono debería ser opcional.
- Rellenar información necesaria de fechas como el cumpleaños o la creación de cuenta, esta última debería registrarse automáticamente en el momento en el que se crea la tupla.
- Sentencias CONSTRAINT para asegurar la validez de datos como que la cantidad de hijos sea mayor que 0 y que los hijos en casa no superen a la cantidad total de hijos; o que la fecha de creación de cuenta sea posterior a la fecha de nacimiento.

warehouse

Lo primordial para esta tabla es crear una clave primaria para cada almacén, la ausencia de ella puede acarrear grandes problemas de duplicación como hemos visto. Entre otras correcciones, podemos encontrar las siguientes:

- No permitir nulos en información importante como teléfono, direcciones, código postal…
- Eliminar duplicados de los almacenes para evitar errores de lectura o de búsqueda.
- Corregir la información de contacto de los almacenes existentes.
- Crear tablas para las clases de almacenes y tiendas con sus respectivas claves foráneas para mantener toda la información.
- Corregir las fechas de creación a la fecha real y relacionarla con la tabla “time_by_day” si se desea más información.
- Eliminar columnas sobrantes como las de direcciones 2, 3 y 4 o la del dueño del almacén.
- Corregir los códigos postales de cada almacén para poder realizar pedidos de reabastecimiento correctamente.

5. Valoración final

Tras analizar de forma general la base de datos llegamos a la conclusión de que tiene una calidad extremadamente baja. Consideramos que, debido a la gran cantidad de errores encontrados y de cambios necesarios, es más recomendable crear una base de datos nueva con un estudio de los requisitos necesarios que corregir la base de datos actual.

Se debe analizar correctamente el sector y las necesidades, tras ello crear un buen diseño y documentación necesaria para la introducción de datos. Se añadirán todas las restricciones y claves que se consideren oportunas y opcionalmente se podrá desarrollar un backend que permita acceder a la base de datos de forma sencilla para controlar el acceso y modificación de la información registrada.

Por último quedaría trasladar toda la información, una vez haya sido revisada y corregida, de la base de datos actual a la nueva para la puesta en producción.

6. Bibliografía

DBeaver: https://dbeaver.io/

Talend: https://www.talend.com/

ISO 25012: https://www.iso25000.com/index.php/normas-iso-25000/iso-25012

Chat GPT: https://chatgpt.com/