



**UNIVERSIDAD DE SANTIAGO DE CHILE**  
**Facultad de Ciencia**  
**Departamento de Matemática y Ciencia de la Computación**

**MODELAMIENTO DE BASE DE DATOS CIENTÍFICOS**  
**LABORATORIO 3**

**Integrantes:** Gabriel Badilla Alfaro  
Rodrigo Kobayashi Araya  
Ignacio Sanhueza Zúñiga  
Javier Gómez Gallegos

**Profesor:** Fredi Palominos Villavicencio

**Fecha de entrega:** 17/01/2021

**Santiago de Chile – Chile**

**AÑO 2021**



## CONTENIDO

1.	Introducción.....	2
2.	Objetivos. ....	2
2.1.	General.....	2
2.2.	Específicos.....	2
3.	Enunciado del problema. ....	3
4.	Modelamiento conceptual. ....	5
4.1.	Análisis del problema. ....	5
4.2.	Modelo Conceptual.....	5
5.	Diseño Lógico. ....	6
5.1.	Modelo estrella de la variable multidimensional. ....	6
6.	Implementación de la Variable Multidimensional.....	7
6.1.	Creación de tablas lock-up.....	7
6.2.	Creación de la tabla central.....	7
6.3.	Creación de claves primarias y foráneas.....	7
6.4.	Script de SQL para la creación de las tablas. ....	8
6.5.	Script de SQL para la creación de claves primarias y foráneas. ....	9
6.6	Diagrama de la variable multidimensional implementada.....	9
7.	Carga inicial de datos (ETL). ....	10
8.	Carga Inicial.....	13
8.1.	Extracción.....	13
8.2.	Transformación.....	14
8.3.	Carga.....	15
9.	Conclusiones. ....	16

## Ilustraciones

Ilustración 1:	DER de la base de datos.....	4
Ilustración 2:	Modelo lógico de la base de datos restaurant.....	4
Ilustración 3:	Modelo lógico de la base de datos comercio.....	5
Ilustración 4	modelo conceptual de la variable multidimensional.....	5
Ilustración 5	modelo conceptual de la variable multidimensional.....	6
Ilustración 6:	Script para crear las tablas de la variable multidimensional.....	8
Ilustración 7:	Script para crear las claves primarias y foráneas de la variable multidimensional.....	9
Ilustración 8:	Diagrama las tablas de la variable multidimensional implementada.....	9
Ilustración 9:	Grafo del proceso de carga inicial de datos.....	10
Ilustración 10:	Script de extracción de datos.....	13
Ilustración 11:	Scripts de transformación de datos.....	14
Ilustración 12:	Scripts de carga de datos.....	15

## Tablas

Tabla 1:	Especificación de datos de entrada y salida del Grafo del proceso inicial de carga de datos.....	11
----------	--	----



## 1. Introducción.

En el laboratorio 3 se pide crear una variable multidimensional para el problema planteado a continuación. Las variables multidimensionales bajo el enfoque R-OLAP modelan hechos de la base de datos que se extraen para luego ser analizadas para la futura toma de decisiones de la institución de donde se toman los datos. Para la correcta creación de una variable multidimensional se requiere tomar en cuenta las necesidades por las cuales se crea la variable, el contexto el cual la rodea y las especificaciones de la o las bases de datos de donde se extrae la información.

A continuación se presentará el planteamiento del problema, la resolución y el resultado final del laboratorio.

## 2. Objetivos.

### 2.1. General.

- Generar una variable multidimensional que esté de acuerdo a los requisitos planteados por el problema.

### 2.2. Específicos.

- Crear un modelo conceptual de la variable de acuerdo al problema planteado.
- Realizar el diseño lógico de la variable bajo un enfoque R-OLAP.
- Realizar la carga inicial de datos de acuerdo a la metodología estudiada en clases.
- Realizar la carga inicial de datos con scripts de SQL diseñados correctamente



### 3. Enunciado del problema.

Un empresario es dueño de dos empresas diferentes. La primera es un restaurante de larga trayectoria y la segunda una ferretería que expende herramientas y materiales de construcción.

Cada una de las empresas tiene sus propios sistemas de información para cada una de las tareas administrativas que realiza: Por ejemplo, control de inventario, gestión de contratos de personal y sueldos, contabilidad, actividades operacionales y registro de ventas.

En particular, en este último ámbito tiene un problema de control, puesto que no le es simple comparar las ventas y los ingresos asociados debido a que están en sistemas diferentes. Para solución este problema necesita que se le implemente una variable multidimensional que le permita llevar un seguimiento de los siguientes indicadores:

1. Construya el modelo conceptual de la variable.
2. Realice el diseño lógico de la variable en un enfoque R-OLAP.
3. Implemente la variable multidimensional.
4. Planifique la carga inicial de datos (ETL) utilizando la metodología estudiada en clases, especificando cada nodo.
5. Realice el proceso de carga inicial, con las correspondientes instrucciones SQL e implemente la variable multidimensional a partir de las bases de datos del restaurante (BDR\_restaurant) y de la ferretería (BDR\_comercio).

Las especificaciones de la base de datos restaurant son la siguientes:

Descripción del problema:

Un restaurante organiza su trabajo en turnos cada día de la semana. Cada turno tiene un horario de inicio y término. Como cada turno se realiza una vez a la semana, cada ejecución del turno se denomina ejercicio.

En cada ejercicio (de un turno) pueden participar diferentes camareros y uno de ellos actúa como coordinador o jefe del turno.

En el ejercicio de cada turno un camarero atiende varias mesas y cada mesa es atendida por un solo camarero.

Producto de la atención a los clientes, cada atención de vincula a un cliente y a una mesa, y obviamente el camarero responsable es el que tiene a cargo la mesa.

Finalmente, cada cuenta se relaciona con uno o más tipos de productos y la cuanta registra la información del cliente, así como el tipo de medio de pago con el que se paga el monto de la cuenta.



El DER de la base de datos es el siguiente:

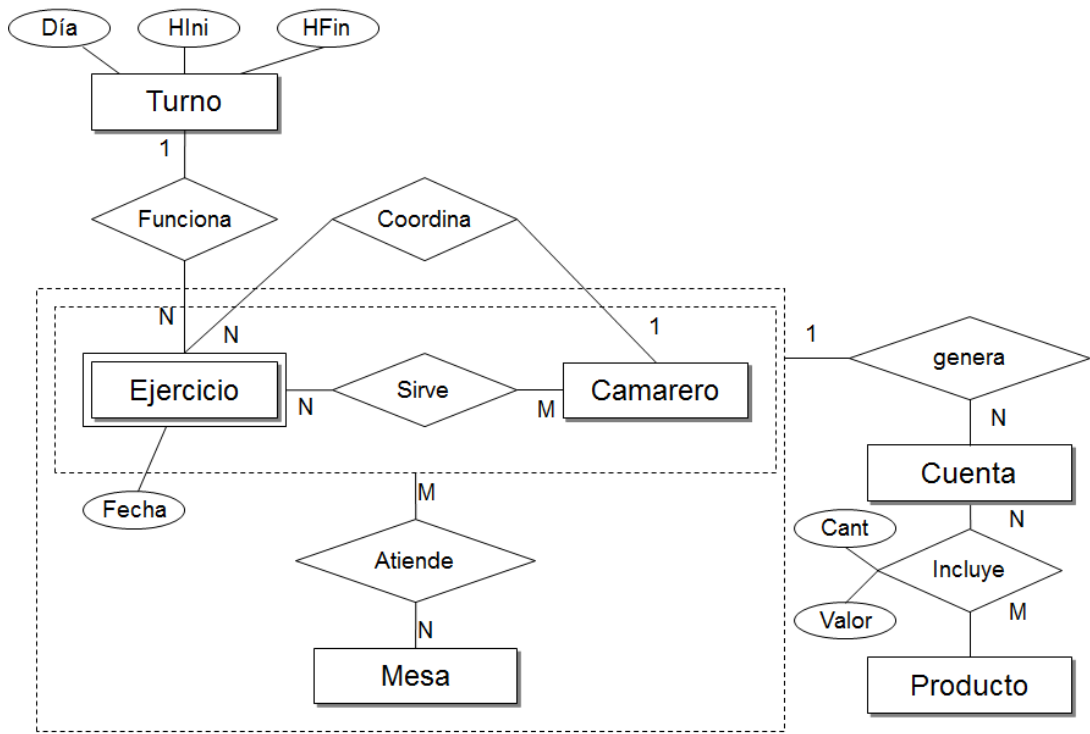


Ilustración 1: DER de la base de datos

El modelo lógico de la base de datos es el siguiente:

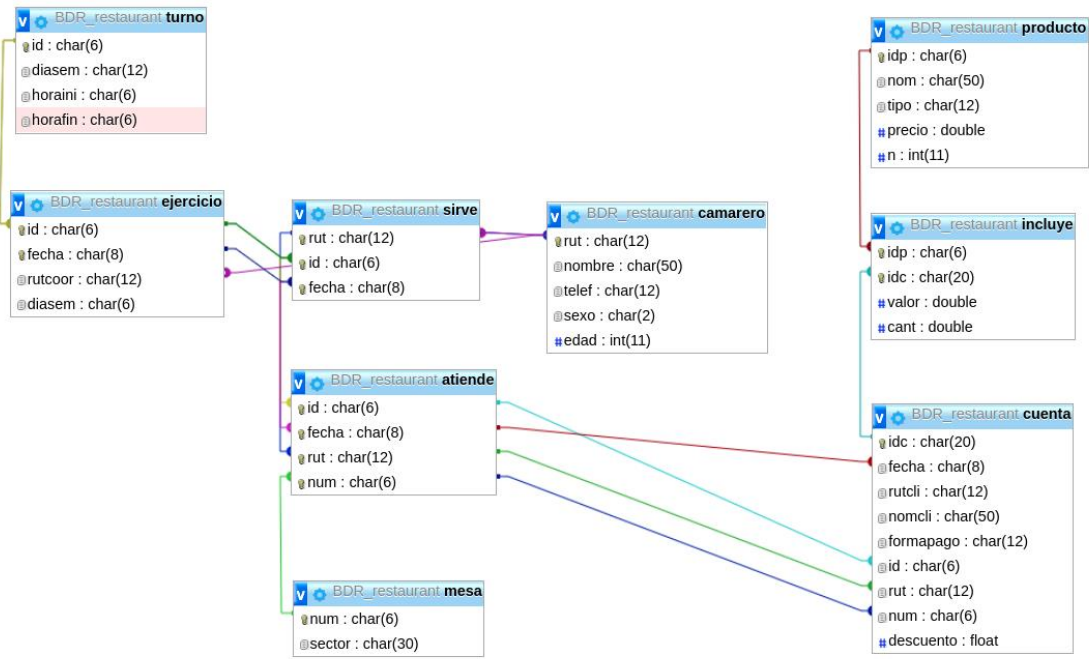


Ilustración 2: Modelo lógico de la base de datos restaurant



El modelo lógico de la base de datos comercio es el siguiente:

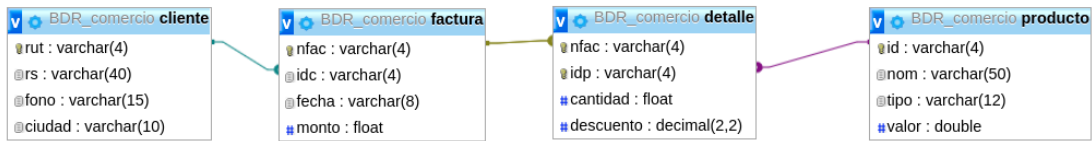


Ilustración 3: Modelo lógico de la base de datos comercio

## 4. Modelamiento conceptual.

El modelamiento conceptual toma la información redactada en el planteamiento del problema, además de la estructura de las tablas de las bases de datos y extrae la información importante para crear indicadores y dimensiones. A continuación se presentará el análisis hecho y el resultado expresado en el modelo conceptual de la variable multidimensional.

### 4.1. Análisis del problema.

Tomando el planteamiento del problema, los indicadores necesarios son los siguientes:

- Ingresos generados por las ventas de cada producto.
- Descuentos asociados a las ventas de cada producto.
- Cantidad de productos vendidos.
- Cantidad de clientes que adquieren dichos productos.

Además, el seguimiento debe realizarse a nivel de día de la semana (Lunes a Domingo), por semana, mes y año.

De esta manera se definen las siguientes dimensiones:

- Tiempo: Se requiere el seguimiento por día de la semana, semana, mes y año.
- Producto: Identificación y precio del producto.
- Cliente: Identificación del cliente.

### 4.2. Modelo Conceptual.

Luego del análisis dimensional del problema planteado, se llega al siguiente modelo conceptual:

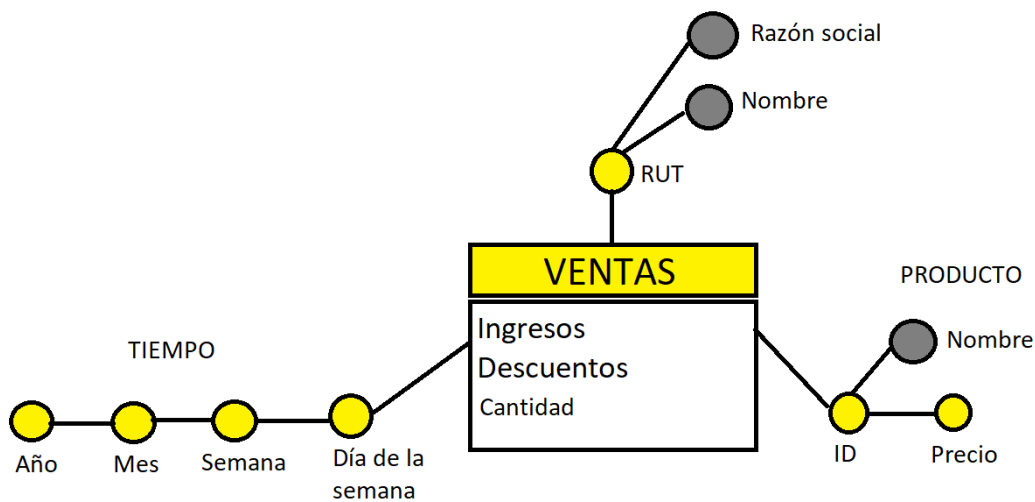


Ilustración 4 modelo conceptual de la variable multidimensional



5. Diseño Lógico.

El diseño lógico toma la información del modelamiento conceptual para generar la variable multidimensional bajo un enfoque, en este caso R-OLAP y un modelo, para este caso el modelo estrella. A continuación se presentará el diseño lógico de la variable multidimensional.

5.1. Modelo estrella de la variable multidimensional.

Luego de crear el modelo conceptual de la variable multidimensional, se crea el modelo estrella, especificando las tablas por crear y su contenido:

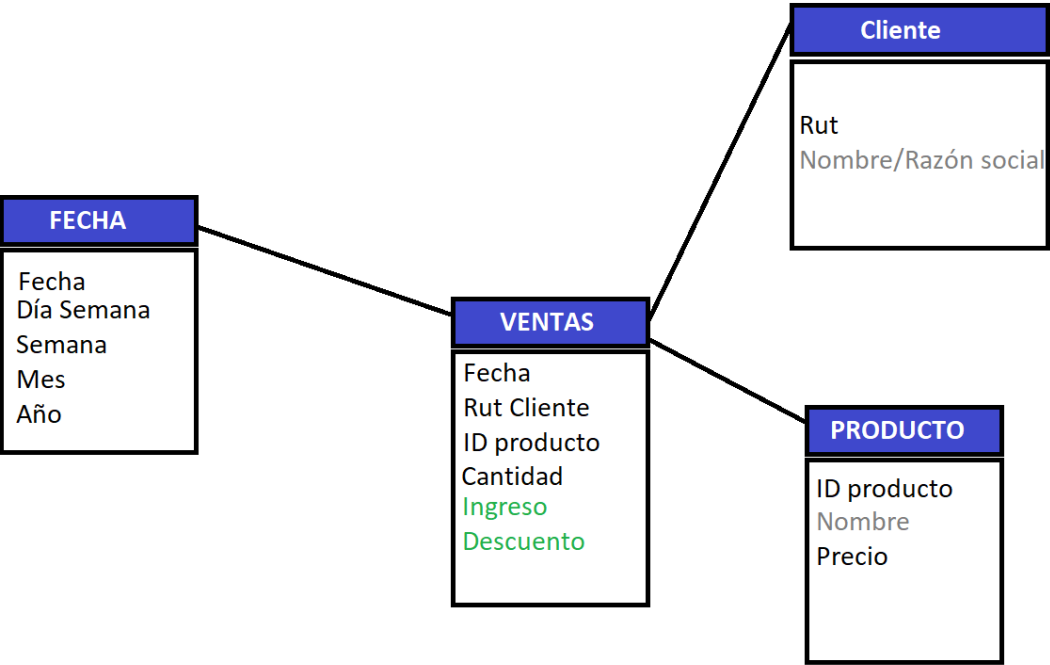


Ilustración 5 modelo conceptual de la variable multidimensional



## 6. Implementación de la Variable Multidimensional.

Luego de realizar el diseño lógico de la variable multidimensional, se realiza la implementación de esta, creando las tablas lock-up para cada dimensión, la tabla central o Fact Table para los indicadores y definiendo las claves primarias y foráneas de las tablas creadas.

### 6.1. Creación de tablas lock-up.

Para la dimensión tiempo se define la tabla Fecha, la cual se compone por los siguientes atributos:

- IDF: Representa la fecha de compra.
- Día\_Semana: Representa el día de la semana en que se hace la compra (1=Lunes, ..., 7=Domingo)
- Semana: Representa la semana en que se hace la compra.
- Mes: Representa el mes en que se hace la compra.
- Año: Representa el año en que se hace la compra.

Para la dimensión Cliente se crea la tabla Cliente con sus atributos:

- Rut: representa el rut del cliente o empresa
- Nombre\_Razon\_Social: Representa el nombre del cliente como persona natural o razón social

Cabe mencionar la fusión de los atributos de las tablas clientes, de BDR\_comercio y cuenta de BDR\_restaurant, en la cual se alberga el rut y nombre del cliente. Ya que no es necesario diferenciar de empresas y personas, ya que estas son clientes para la funcionalidad de la variable multidimensional. Además para el rut, se escoge el varchar con mayor tamaño de los rut de las tablas de BDR\_comercio y BDR\_restaurant.

Para la dimensión Producto se crea la tabla Producto, con sus atributos presentados a continuación:

- IDP: Representa la ID del producto.
- Nombre: El nombre del producto.
- Precio: Representa el precio del producto.

Es necesario mencionar la conversión de ciertos atributos, ya que en alguno de estos, como la cantidad, se presentan de distintas unidades, double y float en este caso. Luego de observar la cantidad en la tabla detalle de BDR\_comercio, no se encuentran valores que tengan decimales dentro de este atributo, por lo que se decide usar double como unidad en la cantidad de la tabla descrita anteriormente.

### 6.2. Creación de la tabla central.

Para la tabla central, se define la tabla Ventas, la cual contiene los siguientes atributos:

- IDFecha: contiene la fecha de transacción.
- Rut\_Cliente: contiene el rut del cliente.
- IDP: representa el ID del producto comprado.
- Ingreso: el ingreso de la compra del producto.
- Descuento: el descuento aplicado a la compra del producto.
- Cantidad: Representa la cantidad del producto comprada.

### 6.3. Creación de claves primarias y foráneas.

Para las tablas descritas anteriormente, se definen las claves primarias:

- IDF para la tabla Fecha
- Rut para la tabla Cliente
- IDP para la tabla producto
- La combinación de las claves anteriores para la tabla Ventas (IDFecha, Rut\_Cliente y IDP)





En cuanto a las claves foráneas, las claves primarias de la tabla Ventas son todas claves foráneas de las tablas lock\_up, donde IDFecha es referencia de IDF de la tabla fecha, Rut\_Cliente es referencia de Rut de la tabla Cliente e IDP es referencia de IDP de la tabla producto.

#### 6.4. Script de SQL para la creación de las tablas.

Para crear las tablas, se presenta el siguiente script de SQL:

```
1  /* Crear Tabla Fecha */
2  CREATE TABLE Fecha(
3      IDF varchar(8) NOT NULL,
4      Dia_Semana varchar(1) NOT NULL,
5      Semana Varchar(2) NOT NULL,
6      Mes varchar(2) NOT NULL,
7      Annio varchar(4) NOT NULL
8  );
9
10 /* Crear Tabla Cliente */
11 CREATE TABLE Cliente (
12     Rut varchar(12) NOT NULL,
13     Nombre_Razon_Social varchar(50) NOT NULL
14 );
15
16 /* Crear Tabla Producto */
17 CREATE TABLE Producto (
18     IDP varchar(6) NOT NULL,
19     Nombre varchar(50) NOT NULL,
20     Precio double NOT NULL
21 );
22
23 /* Crear Tabla Ventas*/
24 CREATE TABLE Ventas (
25     IDFecha varchar (8) NOT NULL,
26     Rut_Cliente varchar(12) NOT NULL,
27     IDP varchar(6) NOT NULL,
28     Ingreso double NOT NULL,
29     Descuento float NOT NULL,
30     Cantidad double NOT NULL
31 );
32
```

*Ilustración 6: Script para crear las tablas de la variable multidimensional*



### 6.5. Script de SQL para la creación de claves primarias y foráneas.

Para la creación de las claves primarias y foráneas detalladas anteriormente, se usa el siguiente script:

```
33  /*Creación de Claves primarias*/
34
35  ALTER TABLE Cliente ADD PRIMARY KEY (Rut);
36  ALTER TABLE Fecha ADD PRIMARY KEY (IDF);
37  ALTER TABLE Producto ADD PRIMARY KEY (IDP);
38  ALTER TABLE Ventas ADD PRIMARY KEY (Rut_Cliente, IDFecha, IDP);
39
40  /* Creación de claves foráneas*/
41
42  ALTER TABLE Ventas ADD FOREIGN KEY (Rut_Cliente) REFERENCES Cliente(Rut);
43  ALTER TABLE Ventas ADD FOREIGN KEY (IDFecha) REFERENCES Fecha(IDF);
44  ALTER TABLE Ventas ADD FOREIGN KEY (IDP) REFERENCES Producto(IDP);
```

Ilustración 7: Script para crear las claves primarias y foráneas de la variable multidimensional

### 6.6 Diagrama de la variable multidimensional implementada.

A continuación se presenta el diagrama de la variable multidimensional ya implementada en la siguiente figura:

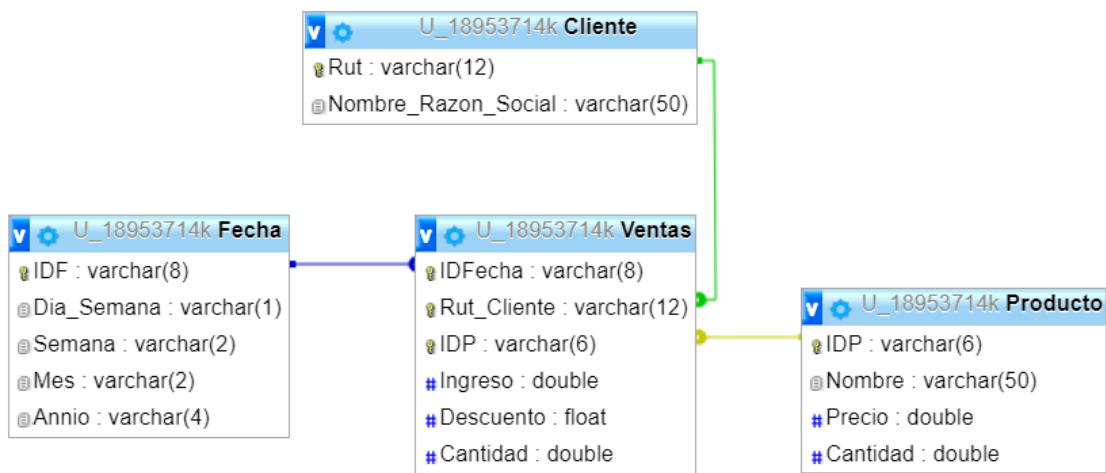


Ilustración 8: Diagrama las tablas de la variable multidimensional implementada



7. Carga inicial de datos (ETL).

La carga inicial de datos consta de tres pasos, la extracción, donde se obtienen los datos provenientes de diferentes sistemas, para este caso, las bases de datos BDR\_comercio y BDR\_restaurant. La transformación, que consiste en la aplicación sistemática de reglas o leyes de derivación para transformar los datos, los cuales deben cumplir con los estándares del modelo descrito anteriormente. Y la carga, en la cual se ingresan los datos refinados a la base de datos que contiene nuestra variable dimensional.

El grafo que describe el proceso de carga inicial de datos se presenta a continuación:

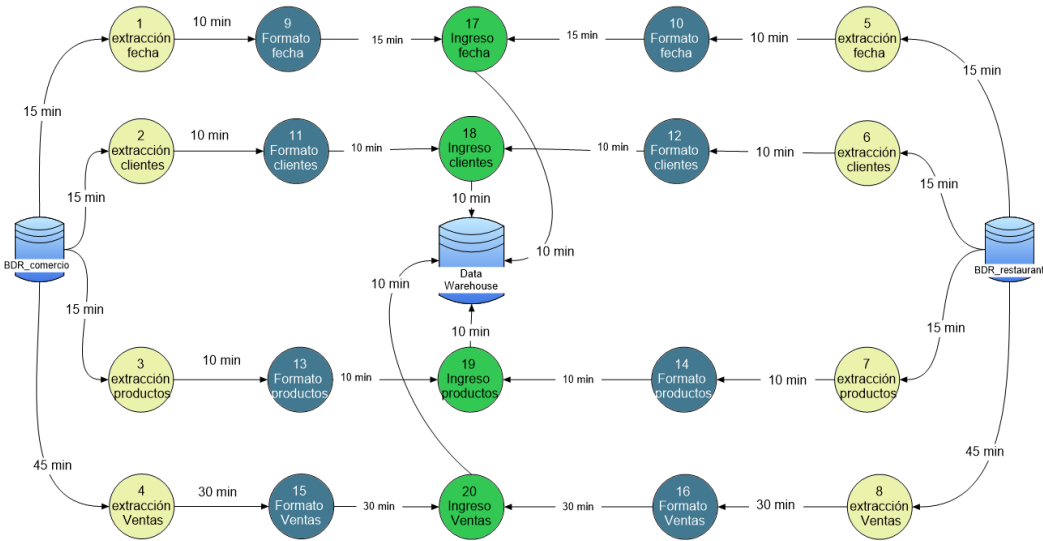


Ilustración 9: Grafo del proceso de carga inicial de datos

En la figura anterior se puede apreciar la especificación de nodo antecesor y sucesor, descripción de la tarea y el tiempo estimado para el desarrollo de la tarea, se omite el costo de la tarea, ya que el proceso ETL se hace bajo un contexto de estudio, en el cual no se tiene presente la ganancia o gasto de recursos. Finalmente, se presenta la especificación de datos de entrada y salida en la siguiente tabla:

Tabla 1: Especificación de datos de entrada y salida del Grafo del proceso inicial de carga de datos

Nombre Proceso	Datos de entrada	Datos de salida
1 extracción fecha	BDR_comercio.factura.fecha	Comercio_fechas
2 extracción clientes	BDR_comercio.cliente.rut BDR_comercio.cliente.rs	Comercio_clientes
3 extracción producto	BDR_comercio.producto.id BDR_comercio.producto.nom BDR_comercio.producto.valor	Comercio_productos
4 extracción Ventas	BDR_comercio.cliente.rut BDR_comercio.factura.fecha BDR_comercio.producto.id SUM(BDR_comercio.detalle.cantidad * BDR_comercio.producto.valor) AVG(BDR_comercio.detalle.descuento) SUM(BDR_comercio.detalle.cantidad)	Comercio_Ventas
5 extracción fecha	BDR_restaurant.cuenta.fecha	Restaurant_fechas
6 extracción clientes	BDR_restaurant.cuenta.rutcli BDR_restaurant.cuenta.nomcli	Restaurant_clientes
7 extracción producto	BDR_restaurant.producto.id BDR_restaurant.producto.nom BDR_restaurant.producto.valor	Restaurant_productos
8 extracción Ventas	BDR_restaurant.cuenta.rutcli BDR_restaurant.cuenta.fecha BDR_restaurant.producto.idp SUM(BDR_restaurant.incluye.cantidad * BDR_restaurant.producto.precio) AVG(BDR_restaurant.cuenta.descuento) SUM(BDR_restaurant.incluye.cant)	Restaurant_Ventas
9 Formato fecha	Comercio_fechas	Comercio_fechas (formateadas)
10 Formato fecha	Restaurant_fechas	Restaurant_fechas (formateadas)
11 Formato clientes	Comercio_clientes	Comercio_clientes (formateados)
12 Formato clientes	Restaurant_clientes	Restaurant_clientes (formateados)
13 Formato productos	Comercio_productos	Comercio_productos (formateados)



14 Formato productos	Restaurant_productos	Restaurant_productos (formateados)
15 Formato Ventas	Comercio_Ventas	Comercio_Ventas (formateados)
16 Formato Ventas	Restaurant_Ventas	Restaurant_Ventas (formateados)
17 Ingreso clientes	Comercio_fechas Restaurant_fechas	Fecha
18 Ingreso clientes	Comercio_clientes Restaurant_clientes	Cliente
19 Ingreso productos	Comercio_productos Restaurant_productos	Producto
20 Ingreso Ventas	Comercio_Ventas Restaurant_Ventas	Ventas



## 8. Carga Inicial

A continuación se presentarán los scripts de SQL para la extracción, transformación y carga de los datos a la variable multidimensional.

### 8.1. Extracción.

Para extraer los datos de las bases de datos de comercio y restaurant, se utilizan los siguientes scripts:

```
1  /*-----EXTRACCION-----*/
2  /*-----EXTRACCION-----*/
3  /*-----EXTRACCION-----*/
4
5  /* -----tablas de fechas transicionales -----*/
6  /*Se extraen las fechas desde las bases de datos*/
7  CREATE TABLE Comercio_fechas AS
8  select DISTINCT factura.fecha as fecha from BDR_comercio.factura;
9
10 CREATE TABLE Restaurant_fechas AS
11 select DISTINCT cuenta.fecha as fecha from BDR_restaurant.cuenta
12
13
14 /* -----tablas de clientes transicionales -----*/
15
16 /*Se extraen los clientes desde las bases de datos*/
17 CREATE TABLE comercio_clientes AS
18 SELECT rut, rs as razon_social from BDR_comercio.cliente
19
20 CREATE TABLE Restaurant_clientes AS
21 SELECT DISTINCT rutcli, nomcli from BDR_restaurant.cuenta
22
23 /* -----tablas de productos transicionales -----*/
24
25 /*Se extraen los productos desde las bases de datos*/
26 CREATE TABLE Comercio_productos AS
27 SELECT DISTINCT producto.id as IDP, producto.nom as Nombre, producto.valor as Precio
28 FROM BDR_comercio.producto;
29
30 CREATE TABLE Restaurant_productos AS
31 SELECT DISTINCT producto.idp as IDP, producto.nom as Nombre, producto.precio as Precio
32 FROM BDR_restaurant.producto;
33
34
35
36
37
38 /*-----tablas de Ventas-----*/
39 CREATE TABLE Comercio_Ventas AS
40 SELECT factura.fecha as IDFecha,
41 cliente.rut as Rut_Cliente,
42 producto.id as IDP,
43 SUM(detalle.cantidad * producto.valor) as Ingreso,
44 AVG(detalle.descuento) as Descuento,
45 SUM(detalle.cantidad) as Cantidad
46 FROM BDR_comercio.cliente, BDR_comercio.factura, BDR_comercio.detalle, BDR_comercio.producto
47 WHERE
48 cliente.rut = factura.idc AND
49 factura.nfac = detalle.nfac AND
50 detalle.idp = producto.id
51 GROUP BY IDFecha, Rut_Cliente, IDP
52
53 CREATE TABLE Restaurant_Ventas AS
54 SELECT cuenta.fecha as IDFecha,
55 cuenta.rutcli as Rut_Cliente,
56 producto.idp as IDP,
57 SUM(include.cant * producto.precio) as Ingreso,
58 AVG(cuenta.descuento) as Descuento,
59 SUM(include.cant) as Cantidad
60 FROM BDR_restaurant.cuenta, BDR_restaurant.include, BDR_restaurant.producto
61 WHERE
62 include.idc = cuenta.idc AND
63 producto.idp = include.idp
64 GROUP BY IDFecha, Rut_Cliente, IDP
```

Ilustración 10: Script de extracción de datos

Cabe mencionar que para la tabla de ventas, se decide aplicar la suma de los ingresos, el promedio de los descuentos y la suma de las cantidades de los productos vendidos a los mismos clientes, los cuales compran el mismo producto en el mismo día. Ya que al usar DISTINCT se perdería información de los ingresos de estas ventas. Esto es especialmente importante en las ventas hechas en el restaurant, ya que hay muchos registros que no indican ni rut ni nombre.



## 8.2. Transformación.

Para la transformación de los datos extraídos, se realizan cambios de unidades descritos en los siguientes scripts:

```
65  /*-----*/
66  /*-----TRANSFORMACION-----*/
67  /*-----*/
68
69
70  /* Se modifica el tipo de dato de las fechas de restaurant a varchar: */
71  ALTER TABLE Restaurant_fechas
72  ALTER COLUMN fecha varchar(8)
73
74
75  /* Se modifica el tipo de dato de rut para calzar a varchar(12) y nombre para
76  calzar a varchar(50) */
77
78  ALTER TABLE comercio_clientes
79  MODIFY COLUMN rut varchar(12);
80
81  ALTER TABLE comercio_clientes
82  MODIFY COLUMN razon_social varchar(50);
83
84  ALTER TABLE Restaurant_clientes
85  MODIFY COLUMN rutcli varchar(12);
86
87  ALTER TABLE Restaurant_clientes
88  MODIFY COLUMN nomcli varchar(50);
89
90  /* Se modifica el tipo de dato de IDP para calzar a varchar(6), Nombre para
91  calzar a varchar(50) y Cantidad para calzar a double */
92
93  ALTER TABLE Restaurant_productos
94  MODIFY COLUMN IDP varchar(6);
95
96  ALTER TABLE Restaurant_productos
97  MODIFY COLUMN Nombre varchar(50);
98
99  ALTER TABLE Comercio_productos
100  MODIFY COLUMN IDP varchar(6);
101
102
103  /*se modifican las columnas de rut del cliente, ID del producto y cantidad
104  vendida de la tabla Comercio_Ventas para calzar con la tabla Ventas*/
105  ALTER TABLE Comercio_Ventas
106  MODIFY COLUMN Rut_Cliente varchar(12);
107
108  ALTER TABLE Comercio_Ventas
109  MODIFY COLUMN IDP varchar(6);
110
111  ALTER TABLE Comercio_Ventas
112  MODIFY COLUMN Descuento float;
113
114  ALTER TABLE Comercio_Ventas
115  MODIFY COLUMN Cantidad double;
116  /*se modifican las columnas de ID de fecha, Rut del cliente y ID del producto
117  de la tabla Restaurant_Ventas para calzar con la tabla Ventas*/
118  ALTER TABLE Restaurant_Ventas
119  MODIFY COLUMN IDFecha varchar(8);
120
121  ALTER TABLE Restaurant_Ventas
122  MODIFY COLUMN Rut_Cliente varchar(12);
123
124  ALTER TABLE Restaurant_Ventas
125  MODIFY COLUMN IDP varchar(6);
```

Ilustración 11: Scripts de transformación de datos



### 8.3. Carga.

Para la carga de datos, se ingresan los contenidos de las tablas con el prefijo Restaurant\_ y Comercio\_ en sus respectivas tablas lockup y tabla central. Los scripts para el ingreso de datos se muestran a continuación:

```
130  /*-----*/
131  /*-----CARGA-----*/
132  /*-----*/
133
134
135  /*METODOS PARA OBTENER COLUMNAS ADICIONALES DE FECHA */
136
137  /* DIA DE LA SEMANA */
138  SELECT DAYOFWEEK(fecha) from Comercio_fechas
139
140  /*SEMANA*/
141  SELECT WEEKOFYEAR(fecha) from Comercio_fechas
142
143  /*MES*/
144  SELECT SUBSTR(fecha, 5, 2) from Comercio_fechas
145
146  /*AÑO*/
147  SELECT SUBSTR(fecha, 1, 4) from Comercio_fechas
148
149  /*Carga de Fechas*/
150
151  INSERT INTO Fecha (IDF, Dia_semana, Semana, Mes, Annio)
152  SELECT fecha,
153  DAYOFWEEK(fecha),
154  WEEKOFYEAR(fecha),
155  SUBSTR(fecha, 5, 2),
156  SUBSTR(fecha, 1, 4)
157  FROM Comercio_fechas
158
159  INSERT INTO Fecha (IDF, Dia_semana, Semana, Mes, Annio)
160  SELECT fecha,
161  DAYOFWEEK(fecha),
162  WEEKOFYEAR(fecha),
163  SUBSTR(fecha, 5, 2),
164  SUBSTR(fecha, 1, 4)
165  FROM Restaurant_fechas
166
167  /*se testea que no existan datos repetidos*/
168  SELECT IDF, count(IDF)
169  FROM Fecha
170  GROUP BY IDF
171  ORDER BY count(IDF) DESC
172
173  /*carga de fechas*/
174  INSERT INTO Cliente (Rut, Nombre_Razon_Social)
175  SELECT rut, razon_social from comercio_clientes;
176
177  INSERT INTO Cliente (Rut, Nombre_Razon_Social)
178  SELECT rutcli, nomcli from Restaurant_clientes;
179
180  /*carga de productos*/
181  INSERT INTO Producto (IDP, Nombre, Precio)
182  SELECT IDP, Nombre, Precio FROM Comercio_productos;
183
184  INSERT INTO Producto (IDP, Nombre, Precio)
185  SELECT IDP, Nombre, Precio FROM Restaurant_productos;
186
187  /*Carga de Ventas*/
188  INSERT INTO Ventas (IDFecha, Rut_Cliente, IDP, Ingreso, Descuento, Cantidad)
189  SELECT IDFecha, Rut_Cliente, IDP, Ingreso, Descuento, Cantidad
190  from Comercio_Ventas;
191
192  INSERT INTO Ventas (IDFecha, Rut_Cliente, IDP, Ingreso, Descuento, Cantidad)
193  SELECT IDFecha, Rut_Cliente, IDP, Ingreso, Descuento, Cantidad
194  from Restaurant_Ventas;
```

Ilustración 12: Scripts de carga de datos





## 9. Conclusiones.

Luego de llevar a cabo la tarea de construir un modelo conceptual de la variable multidimensional, realizar el diseño lógico de dicha variable bajo un enfoque R-OLAP, implementarla, planificar y llevar a cabo la carga inicial de datos, se concluye que el diseño e implementación de variables multidimensionales es uno de los pilares fundamentales de distintas empresas, las cuales necesitan del análisis de sus datos para tomar decisiones en base a estos. Sin el estudio de este diseño de variables, la información dentro de bases de datos relacionales puede verse totalmente oscura y confusa para el entendimiento y análisis de los datos. Además, el diseño de una variable multidimensional depende totalmente de las características que se piden, ya que, si se pidiera otro enfoque con estas mismas bases de datos, la variable cambiaría a una sumamente diferente. Por lo tanto, el diseño de variables multidimensionales no tiene una guía clara de como llevarse a cabo, por lo que su práctica es totalmente necesaria. Sólo la experiencia puede llevar al correcto diseño de estas variables.

Además, se concluye que al diseñar estas variables bajo bases de datos que pueden estar incompletas o defectuosas en sus datos, se deben tomar decisiones difíciles sobre los datos que se deben agregar a las dimensiones e indicadores, ya que, por una parte se necesita la totalidad de datos indicadores para representar la realidad al cien por ciento, pero por otra, al agregar estos datos se corre el riesgo de perjudicar esta representación, generando distorsiones en el posterior análisis de la variable.