

Proyecto SQL

Customer Shopping (Latest Trends) Dataset

Ulises Rodrigo Gonzalez Osnaya

Dataset

Criterios de selección

- alto volumen de datos
- contexto simple
- contar con más de 10 características (columnas)

Dataset

Problemática

Dataset: [Customer Shopping \(Latest Trends\) Dataset](#)

Objetivo: Apoyar la **toma de decisiones** del sector retail (Moda)

Dataset

Problemática

el dataset propuesto recomienda los siguientes enfoques:

- *Analyze consumer purchasing patterns over time.*
- ***Identify popular product categories and high-performing segments.***
- *Develop customer segmentation and personalization strategies.*
- *Build predictive models for sales forecasting or customer retention.*

Dataset

Análisis

```
1 Customer ID, Age, Gender, Item Purchased, Category, Purchase Amount (USD), Location, Size, Color, Season, Review Rating, Subscription
  Status, Payment Method, Shipping Type, Discount Applied, Promo Code Used, Previous Purchases, Preferred Payment Method, Frequency
  of Purchases
2 1, 55, Male, Blouse, Clothing, 53, Kentucky, L, Gray, Winter, 3.1, Yes, Credit Card, Express, Yes, Yes, 14, Venmo, Fortnightly
3 2, 19, Male, Sweater, Clothing, 64, Maine, L, Maroon, Winter, 3.1, Yes, Bank Transfer, Express, Yes, Yes, 2, Cash, Fortnightly
4 3, 50, Male, Jeans, Clothing, 73, Massachusetts, S, Maroon, Spring, 3.1, Yes, Cash, Free Shipping, Yes, Yes, 23, Credit Card, Weekly
5 4, 21, Male, Sandals, Footwear, 90, Rhode Island, M, Maroon, Spring, 3.5, Yes, PayPal, Next Day Air, Yes, Yes, 49, PayPal, Weekly
6 5, 45, Male, Blouse, Clothing, 49, Oregon, M, Turquoise, Spring, 2.7, Yes, Cash, Free Shipping, Yes, Yes, 31, PayPal, Annually
7 6, 46, Male, Sneakers, Footwear, 20, Wyoming, M, White, Summer, 2.9, Yes, Venmo, Standard, Yes, Yes, 14, Venmo, Weekly
8 7, 63, Male, Shirt, Clothing, 85, Montana, M, Gray, Fall, 3.2, Yes, Debit Card, Free Shipping, Yes, Yes, 49, Cash, Quarterly
9 8, 27, Male, Shorts, Clothing, 34, Louisiana, L, Charcoal, Winter, 3.2, Yes, Debit Card, Free Shipping, Yes, Yes, 19, Credit Card, Weekly
10 9, 26, Male, Coat, Outerwear, 97, West Virginia, L, Silver, Summer, 2.6, Yes, Venmo, Express, Yes, Yes, 8, Venmo, Annually
11 10, 57, Male, Handbag Accessories, 31, Missouri, M, Pink, Spring, 4.8, Yes, PayPal, 2-Day Shipping, Yes, Yes, 4, Cash, Quarterly
```

in* 0 0 0 CSVLint Query Align Rainbow OFF Col 7: Location Ln 9, Col 38 Spaces: 4 UTF-8

Dataset

Rows:	3,900	Columns:	19
-------	-------	----------	----

Customer ID	Age	Gender
Item Purchased	Category	Purchase Amount (USD)
Location	Size	Color
Season	Review Rating	Subscription Status
Payment Method	Shipping Type	Discount Applied
Promo Code Used	Previous Purchases	
Preferred Payment Method	Frequency of Purchases	

Dataset

preprocesamiento

- se reemplazan valores "yes", "no" por constantes TRUE, FALSE
- se eliminan espacios en blanco despues del separador

Base de Datos

Diseño *modelado de entidades independientes*

Notamos que se pueden crear entidades para almacenar valores discretos que pueden funcionar como Enums (id, valor/etiqueta).

Gender	Size	Color	Category
Season	PaymentMethod	ShippingType	Frenquency
State			

Base de Datos

Diseño *modelado de entidades independientes*

Cloth
Customer
Purchase

Base de Datos

Diseño script para tablas

```
create table Frenquency (  
    id int not null AUTO_INCREMENT,  
    name varchar(30),  
    primary key (id)  
);
```

script

Base de Datos

Diseño scripts para inserts

cargamos el csv en Atlas y con ayuda de filtros obtuvimos los valores de los Enums, para poblar los inserts. [script](#)

```
insert into Gender (label) values
("MALE"), ("FEMALE");

insert into Size (size) values
("L"),
("M"),
("S"),
("XL");
```

Base de Datos

Diseño scripts para inserts

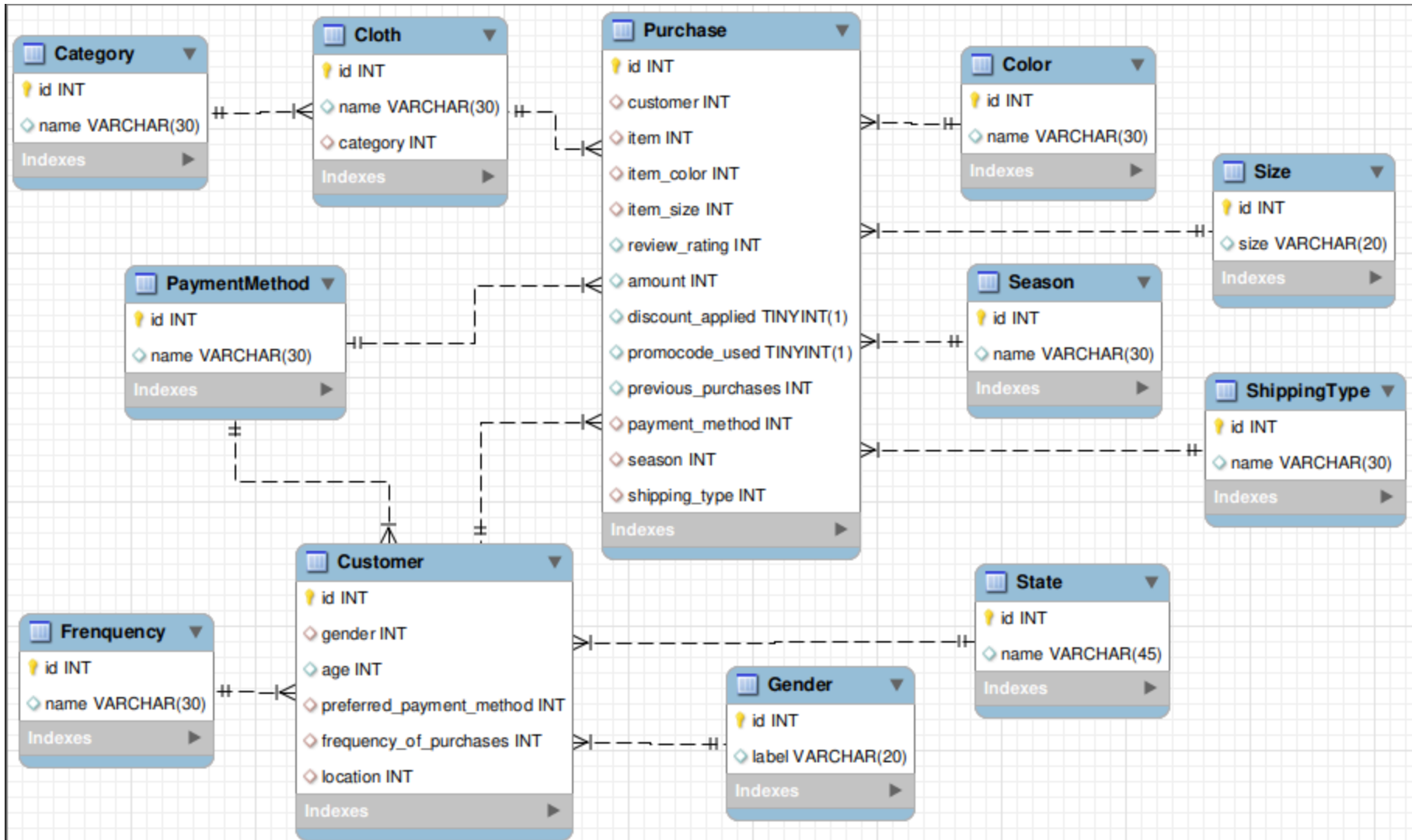
Para insertar en las tablas dependientes creamos un script en Bash para Parsear el CSV, que genera un [archivo de salida](#) con los inserts correspondientes y select anidados para obtener los Id de los Enum. [script](#)

Base de Datos

implementación local

ejecutamos:

- construcción del esquema (createDatabase.sql)
- insertar Enums (enum_insert.sql)
- insertar tablas dependientes (insertCustomers.sql, insertPurchases.sql)



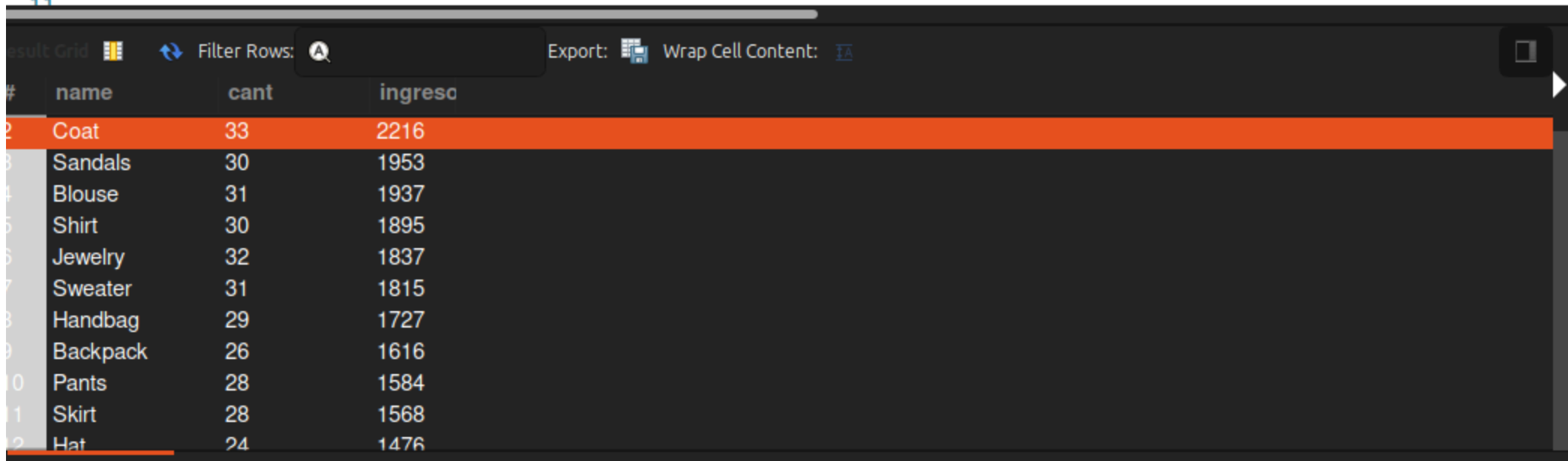
Análisis ¿A qué estado se envían más productos en cada temporada y de cuánto es el ingreso?

```
31  from Purchase
32  inner join Customer on Customer.id = Purchase.customer
33  inner join State on State.id = Customer.location
34  inner join Season on Purchase.season = Season.id
35  where Purchase.shipping_type in (select id from ShippingType where name != "Store Pickup")
36  group by Season.name, State.name
37  order by cantidad DESC, Season.name;
```

#	season	estado	cantidad
1	Winter	Vermont	28
2	Fall	California	27
3	Spring	Nevada	27
4	Summer	North Dakota	26

Análisis ¿Cuál es el producto más vendido en tiendas?

```
5 • select Cloth.name, count(Cloth.id) as cant, sum(Purchase.amount) as ingreso
6   from Purchase join Cloth on Purchase.item = Cloth.id
7   inner join Customer on Customer.id = Purchase.customer
8   inner join ShippingType on Purchase.shipping_type = ShippingType.id
9   group by Purchase.item, ShippingType.name
10  having ShippingType.name = "Store Pickup" order by ingreso DESC, cant DESC;
11
```



The screenshot shows a database query result grid with a dark theme. At the top, there are controls for 'Filter Rows' (with a magnifying glass icon) and 'Export' (with a document icon). Below these, the table has four columns: '#', 'name', 'cant', and 'ingreso'. The data is sorted by 'ingreso' in descending order. The first row, 'Coat', is highlighted in orange. The table lists 12 items in total.

#	name	cant	ingreso
2	Coat	33	2216
3	Sandals	30	1953
4	Blouse	31	1937
5	Shirt	30	1895
6	Jewelry	32	1837
7	Sweater	31	1815
8	Handbag	29	1727
9	Backpack	26	1616
10	Pants	28	1584
11	Skirt	28	1568
12	Hat	24	1476

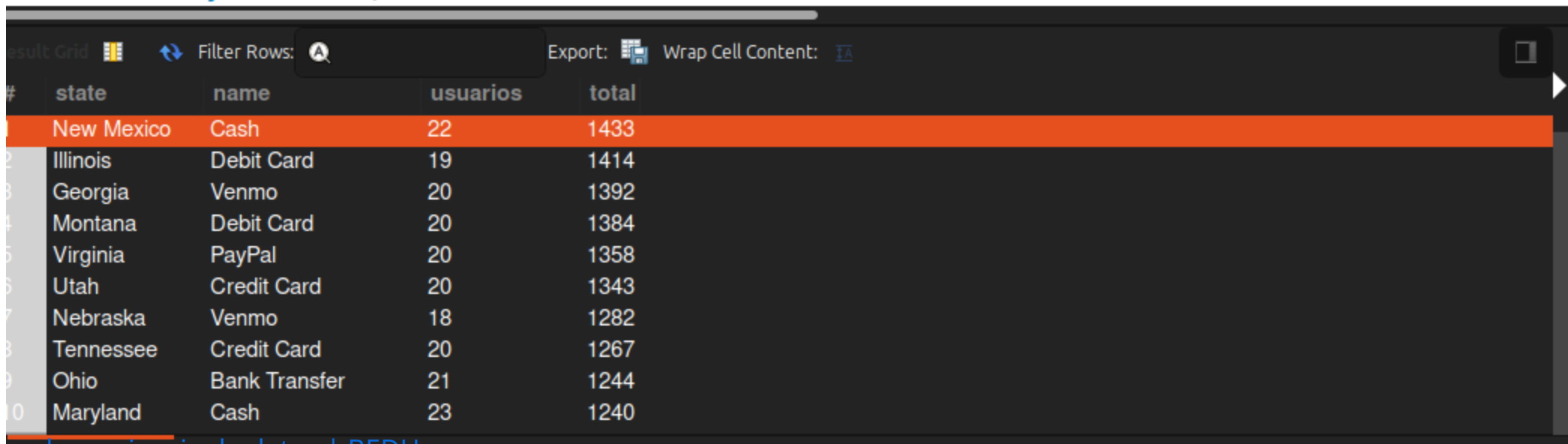
Análisis ¿En qué temporada se envían más productos y cuáles son?

```
14 • select Season.name as season, Cloth.name, count(Cloth.name) as cantidad
15 from Purchase
16 inner join Season on Purchase.season = Season.id
17 inner join Cloth on Purchase.item = Cloth.id
18 where Purchase.shipping_type in (select id from ShippingType where name != "Store Pickup")
19 group by Season.name, Cloth.name
20 order by cantidad DESC;
```

#	season	name	cantidad
1	Winter	Sunglasses	46
2	Winter	Pants	46
3	Fall	Jacket	44
4	Fall	Handbag	43
5	Fall	Hat	41
6	Fall	Skirt	40
7	Spring	Shorts	40
8	Spring	Skirt	40
9	Winter	Hoodie	40
10	Fall	Sweater	40

Análisis ¿en qué estado gastaron más, por tipo de método de pago?

```
18 # en que estado gastaron mas, por tipo de metodo de pago
19 • select State.name as state, PaymentMethod.name, count(Customer) as usuarios, sum(Purchase.amount) as total
20 from Purchase inner join Customer on Purchase.customer = Customer.id
21 inner join State on Customer.location = State.id
22 inner join PaymentMethod on Purchase.payment_method = PaymentMethod.id
23 group by Customer.location, PaymentMethod.id
24 order by total DESC;
```



#	state	name	usuarios	total
1	New Mexico	Cash	22	1433
2	Illinois	Debit Card	19	1414
3	Georgia	Venmo	20	1392
4	Montana	Debit Card	20	1384
5	Virginia	PayPal	20	1358
6	Utah	Credit Card	20	1343
7	Nebraska	Venmo	18	1282
8	Tennessee	Credit Card	20	1267
9	Ohio	Bank Transfer	21	1244
10	Maryland	Cash	23	1240

Conclusiones

- es importante cruzar mas datos para obtener información con mayor detalle para toma de decisiones especificas.
- dado el diseño normalizado de la BD los queries pueden ser complejos y dificil de mantener
- puese ser coveniente el uso de vistas para presentar los datos sin normalización.