

Trabajo Nociones de Arquitectura de la Información

Versión: 2021.05.23 12:00

Observación: Cada vez que agregue nuevos elementos al documento, o que modifique algún componente del informe, revise la coherencia y consistencia con los otros elementos que hacen parte del mismo.

RESPONSABLES

Nombre Completo – Documento de Identificación
1. Juan David Rivera Builes - 1128397071
2. Daniel Cardona Velasquez - 1053851588
3. Carolina Loaiza Salazar - 1036656435
4. Santiago Vasquez Rodriguez - 1035428867
REPO EN GITHUB: https://github.com/judriverabu/Trabajo_AGD_2021-S1

Realiza este trabajo considerando los datos que generan los sistemas transaccionales e información no estructurada de tu dominio (si trabajas por ejemplo para TCC tu dominio es la mensajería; también puedes explorar en la página <https://www.kaggle.com/datasets> o <https://arxiv.org/>). Considera tener acceso a esta información, de al menos 10 MB (puede ser uno o varios archivos de texto), y **tener al menos cuatro clases conceptuales. Este documento también debe almacenarse en el REPO. Plazo Máximo de Entrega 23 de Mayo, NO SE recibirá por correo electrónico, envía por <https://forms.gle/h7ty3yZykaUq5m7y6>**

1 COMPRENSIÓN DEL NEGOCIO

1.1 DESCRIPCIÓN DEL CONTEXTO DEL NEGOCIO.

Los datos con los que se cuentan para trabajar hacen referencia a una parte de la operación de Rappi. Son datos anonimizados, (a los que se tiene acceso por parte de uno de los integrantes del equipo que trabaja para Rappi y con autorización de esta misma) de: los usuarios, los restaurantes, los rappitenderos y las órdenes del último mes. Estos datos son generados a partir de la interacción usuario-tienda-rappitendero todos orquestados por la infraestructura de la app de Rappi. La promesa de valor de Rappi se basa en ofrecer servicios de intermediación entre usuarios, restaurantes y rappitenderos esta intermediación se puede lograr gracias a la aplicación móvil que registra cada una de las transacciones. Como bien se dijo anteriormente solo se cuenta con información de una parte de la

operación de Rappi, toda vez que estos prestan otros servicios que no se toman en cuenta, es por eso que se delimitó el análisis en 4 entidades: **entidad usuario**, **entidad rappidero**, **entidad restaurante** y **entidad orden**. Donde se pudo extraer información relevante de cada una de estas entidades.

IDENTIFICACIÓN DEL PROBLEMA:

La problemática que se quiere resolver en este trabajo consiste en conocer las diferencias entre el tiempo de inicio y el tiempo que finaliza la orden. Esta problemática se puede observar a diferentes niveles de agregación de los datos: La ciudad, el tipo de tienda, el género del usuario, el tipo de usuario, el tipo de celular del usuario (device) y demás variables que se identifican en los datos; con esto se pretende conocer si hay factores demográficos que determinan o influyen sobre los tiempos de entrega.

1.2 DETERMINACIÓN DE OBJETIVOS:

El objetivo del trabajo es conocer el grado de complejidad operacional que tiene Rappi. Esta medición del grado de complejidad estará dada por los tiempos que tarda en entregar una orden. Esto bajo el supuesto que las órdenes no sufren cambios durante su ejecución (bien se sabe que esto en la realidad pasa generalmente) y que por supuesto complejizan mucho más la operación.

Otro de los objetivos del trabajo es poder determinar si hay factores propios de la operación que ocasionan que los tiempos de respuesta sean mayores (o menores) según la ciudad, el género del usuario, el tipo de celular del usuario, el tipo de repartidor (rt), el valor de la orden, el método de pago o el tipo de restaurantes.

1.3 EVALUACIÓN DE LA SITUACIÓN ACTUAL:

La situación actual del problema está determinada por las diferencias en los tiempos de entrega de una orden. Si bien el hecho de poder homogeneizar los tiempos de entrega se tornan complejos (factores exógenos que no se pueden controlar) si se podría decir que al menos algunas tendencias se podrían controlar. Como por ejemplo, que en determinadas horas del día, o en algunas ciudades se esperaría que la operación fuese más o menos parecida. De esta manera, se espera poder realizar algunas propuestas que indiquen cuales son esos factores que pueden llevar a una homogeneización del proceso y hacer que los tiempos de entrega no sean tan variables.

2 COMPRENSIÓN DE LOS DATOS

2.1 RECOLECCIÓN DE DATOS

Como se ha dicho anteriormente los datos son reales provenientes de algunas órdenes hechas por usuarios bajo la aplicación de Rappi. Estos datos inicialmente provienen de microservicios (**ms**) que luego mediante tareas programadas se migran al datawarehouse de Rappi. Una de los posibles riesgos

Departamento de Ciencias de la Computación y la Decisión

que existen corren por cuenta de fallos en la arquitectura de Rappi, bien sea porque los ms se queden colgados o porque las bases de datos no escalen lo suficiente y se rompan los flujos. Una forma para contener estos problemas consiste en realizar seguimientos a los ms: Que no presenten 404, 501 u otros errores de integración; en cuanto a los errores de migración de bases de datos es igual que en lo ms, toda vez que se tienen paneles donde se le hacen seguimientos a las tablas y en los casos de presentar errores levanta alarmas a los desarrolladores.

2.2 DESCRIPCIÓN DE DATOS (DICCIONARIO):

Order Entity:

Nombre del atributo / variable	Formato o Tipo de Dato	Descripción
ORDER_ID	INTEGER	Identificador único de cada orden
APPLICATION_USER_ID	INTEGER	Identificador único en la app para cada usuario
STOREKEEPER_ID	INTEGER	Identificador único del repartidor
CREATED_AT	TEXT	Fecha de creación de la orden (cuando el usuario ordena)
CLOSE_AT	TEXT	Fecha de finalización de la orden (cuando es entregada)
PAYMENT_METHOD	TEXT	Método de pago de la orden
TOTAL_VALUE	INTEGER	Valor total de la orden
TIP	INTEGER	Propina dada por el usuario

Rt Entity:

Nombre del atributo / variable	Formato o Tipo de Dato	Descripción
STOREKEEPER_ID	INTEGER	Identificador único del repartidor
GENDER_RT	TEXT	Género del repartidor
BIRTHDAY_RT	TEXT	Fecha de cumpleaños del repartidor
TRANSPORT_MEDIA_TYPE	TEXT	Medio de transporte del repartidor
DELIVERY_KIT_SIZE	TEXT	Tamaño de la maleta del repartidor

Store Entity:

Nombre del atributo / variable	Formato o Tipo de Dato	Descripción
STORE_ID	INTEGER	Identificador de la tienda
ORDER_ID	INTEGER	Identificador de la orden
CITY	TEXT	Ciudad
PRICE	INTEGER	Precio de la orden
PRODUCT_NAME	TEXT	Nombre del producto
VERTICAL_SUB_GROUP	TEXT	Tipo de establecimiento

Departamento de Ciencias de la Computación y la Decisión

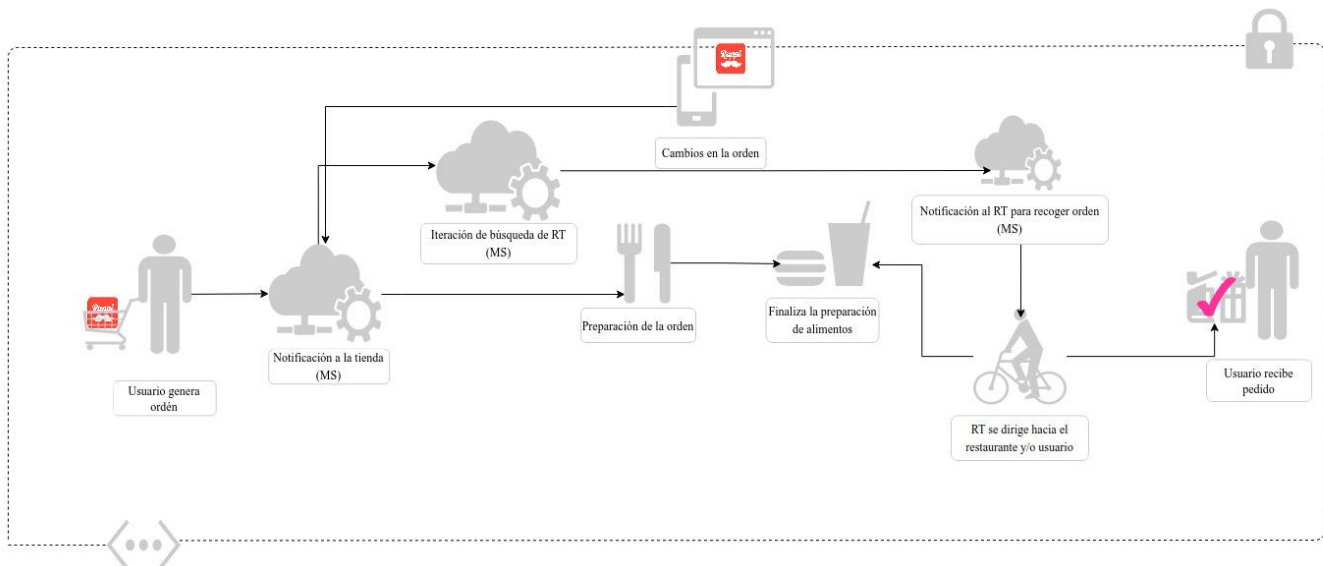
CREATED_AT	TEXT	Fecha de creación de la orden
------------	------	-------------------------------

User Entity:

Nombre del atributo / variable	Formato o Tipo de Dato	Descripción
APPLICATION_USER_ID	INTEGER	Identificador único en la app para cada usuario
GENDER	TEXT	Género del usuario
REGISTER_DEVICE	TEXT	Dispositivo del usuario
SEGMENT_RFM	TEXT	Segmento del usuario
IS_PRIME	TEXT	Indica si es un usuario con rappi prime

2.3 MODELO DEL DOMINIO

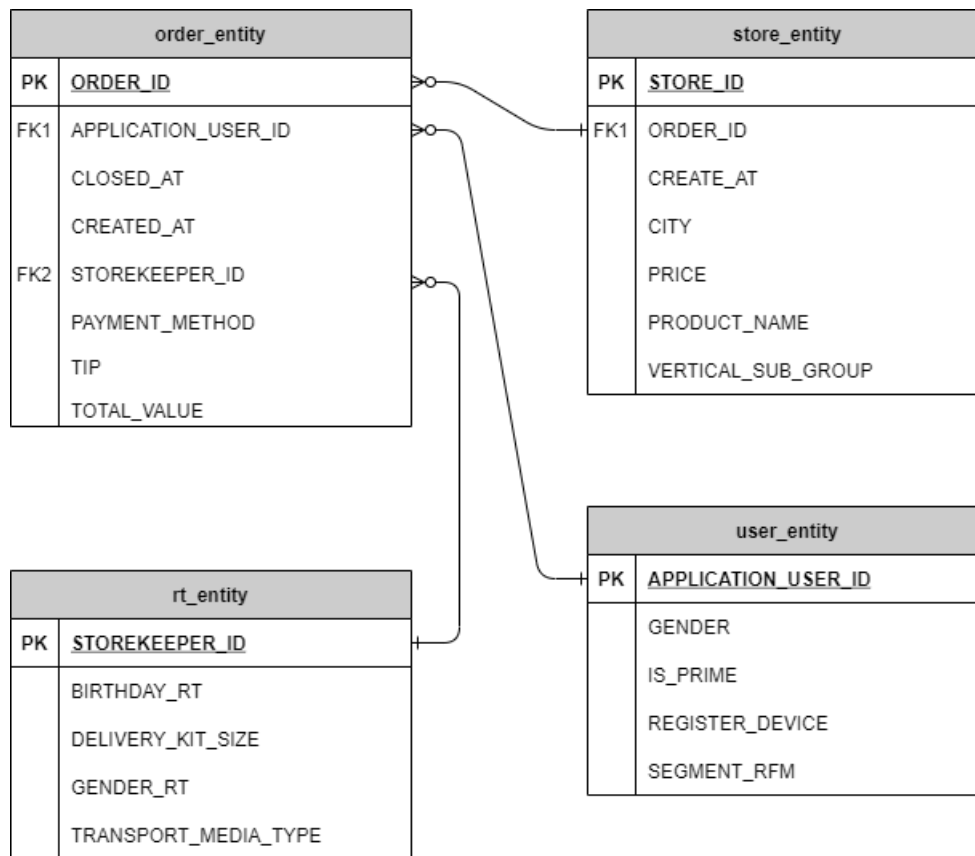
La forma en que se realiza extracción de los datos está soportada sobre la aplicación móvil de Rappi basado en una infraestructura de apis que permiten levantar la información correspondiente. A continuación, se explicará a grandes rasgos cómo es el flujo de ejecución de una orden: Una vez el usuario crea la orden el flujo que se activa es la de hacer visible al orden ante el partner (que puede decidir tomarla o no) una vez toma la orden al usuario le aparece que el partner tomo la orden y acto seguido la orden cambia de estado en la aplicación - pasando de tomado por el usuario a en ejecución o preparación de alimentos- en los próximos 5 minutos el algoritmo de asignación de repartidor (RT) comienza a iterar para encontrar los posibles prospectos de RT, paralelamente el restaurante está preparando la comida, una vez el partner termina de preparar los alimentos, se listan los posible prospectos de RT el algoritmo vuelve a iterar y solo escogerá a un solo RT a este último se le asignará la orden - en esta parte se le muestra al RT tanto la orden del restaurante como del usuario- y éste deberá dirigirse al restaurante, una vez llega a la dirección el RT cambia el estado de la orden a en manos del domiciliario y emprende la marcha hacia la dirección del usuario; finalmente llega al destino y la orden es cerrada por el RT.



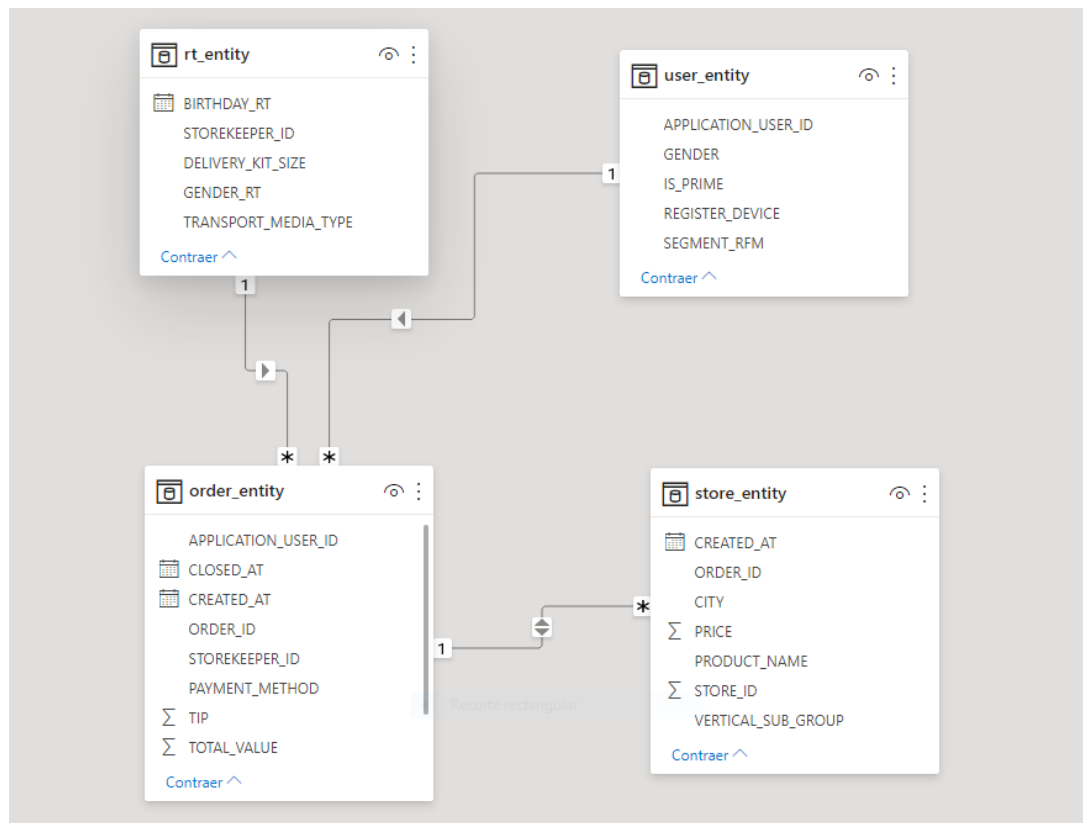
3 MODELO ENTIDAD-RELACIÓN

3.1 TOMA DE PANTALLA DEL MODELO E-R

Observación: lo que se pide, puede usar <https://draw.io> o Microsoft Visio® y modele usando la notación de Barker.



Departamento de Ciencias de la Computación y la Decisión



3.2 SENTENCIA O CONSULTA DE CREACIÓN DEL TABLA(S)

Observación: Escriba el código en el Sistema de Gestión de Bases de Datos Relacionales de su elección (se recomienda SQLite por simplicidad, mediante <https://sqlitebrowser.org/>) para crear las tablas que corresponda con su conjunto de datos específico. Almacene en el repositorio **(REPO EN GITHUB)** el script con el nombre de T1.3.2.Creacion_Tablas.sql

Se recomienda repasar SQL en <https://www.w3schools.com/sql/default.asp>

order_entity:

```
1 CREATE TABLE "order_entity" (  
2     "ORDER_ID" INTEGER NOT NULL UNIQUE,  
3     "APPLICATION_USER_ID" INTEGER,  
4     "CLOSE_AT" TEXT,  
5     "CREATED_AT" TEXT,  
6     "STOREKEEPER_ID" INTEGER,  
7     "PAYMENT_METHOD" TEXT,  
8     "TIP" NUMERIC,  
9     "TOTAL_VALUE" NUMERIC,  
10    PRIMARY KEY("ORDER_ID")  
11 );
```

rt_entity:

```
1 CREATE TABLE "rt_entity" (  
2     "STOREKEEPER_ID"    INTEGER NOT NULL UNIQUE,  
3     "GENDER_RT" TEXT,  
4     "BIRTHDAY_RT" TEXT,  
5     "TRANSPORT_MEDIA_TYPE" TEXT,  
6     "DELIVERY_KIT_SIZE" TEXT,  
7     PRIMARY KEY("STOREKEEPER_ID")  
8 );
```

store_entity:

```
1 CREATE TABLE "store_entity" (  
2     "STORE_ID"    INTEGER NOT NULL UNIQUE,  
3     "ORDER_ID"    INTEGER,  
4     "CITY" TEXT,  
5     "PRICE" INTEGER,  
6     "PRODUCT_NAME" TEXT,  
7     "VERTICAL_SUB_GROUP" TEXT,  
8     "CREATED_AT" TEXT,  
9     PRIMARY KEY("STORE_ID")  
10 );
```

user_entity:

```
1 CREATE TABLE "user_entity" (  
2     "APPLICATION_USER_ID"    INTEGER NOT NULL UNIQUE,  
3     "GENDER" TEXT,  
4     "REGISTER_DEVICE" TEXT,  
5     "SEGMENT_RFM" TEXT,  
6     "IS_PRIME" TEXT,  
7     PRIMARY KEY("APPLICATION_USER_ID")  
8 );
```


3.3 SENTENCIAS PARA INSERTAR DATOS

Observación: Escriba el código para insertar los datos en cada una de las tablas creadas. Almacene en el repositorio (REPO EN GITHUB) el script con el nombre de T1.3.3.Insertar_Datos.sql

```
--Carga de datos con SQL Server
BULK INSERT order_entity
FROM "D:\Escritorio\raw-data\order_entity.csv"
WITH(
FIELDTERMINATOR = ","
ROWTERMINATOR = "\n"
);

BULK INSERT rt_entity
FROM "D:\Escritorio\raw-data\rt_entity.csv"
WITH(
FIELDTERMINATOR = ","
ROWTERMINATOR = "\n"
);

BULK INSERT store_entity
FROM "D:\Escritorio\raw-data\store_entity.csv"
WITH(
FIELDTERMINATOR = ","
ROWTERMINATOR = "\n"
);

BULK INSERT user_entity
FROM "D:\Escritorio\raw-data\user_entity.csv"
WITH(
FIELDTERMINATOR = ","
ROWTERMINATOR = "\n"
);
```

3.4 SENTENCIA DE CONSULTA

Observación: realice la exploración básica de los datos, conteos totales y por categorías, máximos, promedio y mínimos. Es decir, aplique estadística descriptiva con el fin de conocer las propiedades de los datos y entenderlos lo mejor posible. Use solamente sentencias SQL. Anexe las tomas de pantalla donde evidencie la sentencia SQL y su correspondiente ejecución. Además, Almacene en el repositorio (REPO EN GITHUB) el script con el nombre de T1.3.4.Consultar_Datos.sql

Departamento de Ciencias de la Computación y la Decisión

Conteo de device por application_user_id

```
SELECT
REGISTER_DEVICE
, COUNT(APPLICATION_USER_ID) AS DEVICE_COUNT
FROM user_entity
WHERE 1 = 1
AND REGISTER_DEVICE <> 'OTRO'
GROUP BY REGISTER_DEVICE
ORDER BY DEVICE_COUNT DESC;
```

	REGISTER_DEVICE	COUNT_DEVICE
1	IPHONE 7	52458
2	IPHONE 6	45847
3	IPHONE 7 PLUS	39265
4	IPHONE 8 PLUS	38138
5	ANE-LX3	37794
6	IPHONE X	35527
7	IPHONE11,8	32652
8	IPHONE 6S	30153
9	IPHONE 8	27502
10	SM-A715F	23241

Conteo de gender por application user_id

```
SELECT
GENDER
, COUNT(APPLICATION_USER_ID) AS GENDER_COUNT
FROM user_entity
WHERE 1 = 1
AND GENDER <> ''
GROUP BY GENDER
ORDER BY GENDER_COUNT DESC;
```

	GENDER	GENDER_COUNT
1	F	619329
2	M	572034

Conteo de segmento rfm por application user_id

```
SELECT
SEGMENT_RFM
, COUNT(APPLICATION_USER_ID) AS SEGMENT_COUNT
FROM user_entity
WHERE 1 = 1
AND SEGMENT_RFM <> 'Undefined Segment RFM'
GROUP BY SEGMENT_RFM
ORDER BY SEGMENT_COUNT DESC;
```

	SEGMENT_RFM	SEGMENT_COUNT
1	Low Value	578384
2	Regular	279701
3	Potential Power	256050
4	Power	69397
5	Potential Diamond	29245
6	Diamond	19467
7	Zero	1548

Departamento de Ciencias de la Computación y la Decisión

Conteo de prime por application user_id

```
SELECT
IS_PRIME
, COUNT(APPLICATION_USER_ID) AS PRIME_COUNT
FROM user_entity
WHERE 1 = 1
GROUP BY IS_PRIME
ORDER BY PRIME_COUNT DESC;
```

	IS_PRIME	PRIME_COUNT
1	false	969907
2	true	268629

Conteo de genero, segmento y prime de usuarios

```
SELECT
IS_PRIME
, SEGMENT_RFM
, GENDER
, COUNT(APPLICATION_USER_ID) AS USER_COUNT
FROM user_entity
WHERE 1 = 1
AND SEGMENT_RFM <> 'Undefined Segment RFM'
AND GENDER <> ''
GROUP BY IS_PRIME, SEGMENT_RFM, GENDER;
```

	IS_PRIME	SEGMENT_RFM	GENDER	USER_COUNT
1	false	Diamond	F	898
2	false	Diamond	M	796
3	false	Low Value	F	283223
4	false	Low Value	M	236918
5	false	Potential Diamond	F	1884
6	false	Potential Diamond	M	2163
7	false	Potential Power	F	79777
8	false	Potential Power	M	70898
9	false	Power	F	8798
10	false	Power	M	7727
11	false	Regular	F	116012
12	false	Regular	M	113382

Conteo de gener de rt por storekeeper_id

```
SELECT
GENDER_RT
, COUNT(STOREKEEPER_ID) AS GENDER_RT_COUNT
FROM rt_entity
WHERE 1 = 1
GROUP BY GENDER_RT;
```

	GENDER_RT	GENDER_RT_COUNT
1	F	1325
2	H	23978
3	M	7260

Departamento de Ciencias de la Computación y la Decisión

Conteo de transporte de rt por storekeeper_id

```
SELECT
TRANSPORT_MEDIA_TYPE
,COUNT(STOREKEEPER_ID) AS TRANSPORT_COUNT
FROM rt_entity
WHERE 1 = 1
GROUP BY TRANSPORT_MEDIA_TYPE;
```

	TRANSPORT_MEDIA_TYPE	TRANSPORT_COUNT
1	bicycle	15966
2	car	2490
3	motorbike	14103
4	neither	4

Conteo de kit size de rt por storekeeper_id

```
SELECT
DELIVERY_KIT_SIZE
,COUNT(STOREKEEPER_ID) AS DELIVERY_KIT_COUNT
FROM rt_entity
WHERE 1 = 1
AND DELIVERY_KIT_SIZE <> ''
GROUP BY DELIVERY_KIT_SIZE;
```

	DELIVERY_KIT_SIZE	DELIVERY_KIT_COUNT
1	big	14598
2	medium	2
3	other	5879
4	small	12083

Edad mínima de rt

```
WITH AGE AS(
SELECT
STOREKEEPER_ID
,BIRTHDAY_RT
,DATETIME(JULIANDAY(DATETIME('NOW')) - DATETIME(JULIANDAY(BIRTHDAY_RT))) AS AGE_RT
FROM rt_entity
WHERE 1 = 1
)
SELECT
MIN(AGE_RT) AS AGE_RT
FROM AGE
WHERE 1 = 1
AND BIRTHDAY_RT > '1918-11-16'
AND BIRTHDAY_RT < '2019-02-12';
```

	AGE_RT
1	18

Edad máxima de rt

```
WITH AGE AS(
SELECT
STOREKEEPER_ID
,BIRTHDAY_RT
,DATETIME(JULIANDAY(DATETIME('NOW')) - DATETIME(JULIANDAY(BIRTHDAY_RT))) AS AGE_RT
FROM rt_entity
WHERE 1 = 1
)
SELECT
MAX(AGE_RT) AS AGE_RT
FROM AGE
WHERE 1 = 1
AND BIRTHDAY_RT > '1918-11-16'
AND BIRTHDAY_RT < '2019-02-12';
```

	AGE_RT
1	85

Departamento de Ciencias de la Computación y la Decisión

Edad promedio de rt

```
WITH AGE AS(  
  SELECT  
    STOREKEEPER_ID  
    , BIRTHDAY_RT  
    , DATETIME(JULIANDAY(DATETIME('NOW'))) - DATETIME(JULIANDAY(BIRTHDAY_RT)) AS AGE_RT  
  FROM rt_entity  
  WHERE 1 = 1  
)  
SELECT  
  round(AVG(AGE_RT),0) AS AGE_RT  
FROM AGE  
WHERE 1 = 1  
AND BIRTHDAY_RT > '1918-11-16'  
AND BIRTHDAY_RT < '2019-02-12';
```

AGE_RT	
1	32.0

Conteo edades del rt

```
WITH AGE AS(  
  SELECT  
    STOREKEEPER_ID  
    , BIRTHDAY_RT  
    , DATETIME(JULIANDAY(DATETIME('NOW'))) - DATETIME(JULIANDAY(BIRTHDAY_RT)) AS AGE_RT  
  FROM rt_entity  
  WHERE 1 = 1  
)  
SELECT  
  AGE_RT  
  , COUNT(AGE_RT) AS AGE_RT_COUNT  
FROM AGE  
WHERE 1 = 1  
AND BIRTHDAY_RT > '1918-11-16'  
AND BIRTHDAY_RT < '2019-02-12'  
GROUP BY AGE_RT  
ORDER BY AGE_RT_COUNT DESC;
```

	AGE_RT	AGE_RT_COUNT
1	30	476
2	25	443
3	27	422
4	26	420
5	29	413
6	24	397
7	22	392
8	28	359
9	23	355
10	31	348
11	32	345
12	21	323
13	20	279
14	33	268
15	34	257
16	38	249

edad mínima por género edades del rt

```
WITH AGE AS(
SELECT
STOREKEEPER_ID
,BIRTHDAY_RT
,GENDER_RT
,TRANSPORT_MEDIA_TYPE
,DELIVERY_KIT_SIZE
,DATETIME(JULIANDAY(DATETIME('NOW'))) - DATETIME(JULIANDAY(BIRTHDAY_RT)) AS AGE_RT
FROM rt_entity
WHERE 1 = 1
)
SELECT
GENDER_RT
,TRANSPORT_MEDIA_TYPE
,DELIVERY_KIT_SIZE
,MIN(AGE_RT) AS AGE_RT_COUNT
FROM AGE
WHERE 1 = 1
AND BIRTHDAY_RT > '1918-11-16'
AND BIRTHDAY_RT < '2019-02-12'
GROUP BY GENDER_RT,TRANSPORT_MEDIA_TYPE,DELIVERY_KIT_SIZE
ORDER BY AGE_RT_COUNT DESC;
```

	GENDER_RT	TRANSPORT_MEDIA_TYPE	DELIVERY_KIT_SIZE	AGE_RT_COUNT
11	F	motorbike	other	19
12	M	car	big	19
13	M	car	other	19
14	F	bicycle	other	18
15	M	bicycle	big	18
16	M	bicycle	other	18
17	M	motorbike	big	18
18	M	motorbike	other	18

Edad máxima por género edades del rt

```
WITH AGE AS(
SELECT
STOREKEEPER_ID
,BIRTHDAY_RT
,GENDER_RT
,TRANSPORT_MEDIA_TYPE
,DELIVERY_KIT_SIZE
,DATETIME(JULIANDAY(DATETIME('NOW'))) - DATETIME(JULIANDAY(BIRTHDAY_RT)) AS AGE_RT
FROM rt_entity
WHERE 1 = 1
)
SELECT
GENDER_RT
,TRANSPORT_MEDIA_TYPE
,DELIVERY_KIT_SIZE
,MAX(AGE_RT) AS AGE_RT_COUNT
FROM AGE
WHERE 1 = 1
AND BIRTHDAY_RT > '1918-11-16'
AND BIRTHDAY_RT < '2019-02-12'
GROUP BY GENDER_RT,TRANSPORT_MEDIA_TYPE,DELIVERY_KIT_SIZE
ORDER BY AGE_RT_COUNT DESC;
```

	GENDER_RT	TRANSPORT_MEDIA_TYPE	DELIVERY_KIT_SIZE	AGE_RT_COUNT
1	F	bicycle	other	85
2	M	bicycle	other	73
3	M	car	big	71
4	M	motorbike	other	68
5	F	car	big	67
6	F	motorbike	other	62
7	M	car	other	62
8	M	motorbike	big	59

Edad promedio por género edades del rt

```
WITH AGE AS(
SELECT
STOREKEEPER_ID
,BIRTHDAY_RT
,GENDER_RT
,TRANSPORT_MEDIA_TYPE
,DELIVERY_KIT_SIZE
,DATETIME(JULIANDAY(DATETIME('NOW'))) - DATETIME(JULIANDAY(BIRTHDAY_RT)) AS AGE_RT
FROM rt_entity
WHERE 1 = 1
)
SELECT
GENDER_RT
,TRANSPORT_MEDIA_TYPE
,DELIVERY_KIT_SIZE
,ROUND(AVG(AGE_RT),1) AS AGE_RT_COUNT
FROM AGE
WHERE 1 = 1
AND BIRTHDAY_RT > '1918-11-16'
AND BIRTHDAY_RT < '2019-02-12'
GROUP BY GENDER_RT,TRANSPORT_MEDIA_TYPE,DELIVERY_KIT_SIZE
```

	GENDER_RT	TRANSPORT_MEDIA_TYPE	DELIVERY_KIT_SIZE	AGE_RT_COUNT
1	M	car	big	36.2
2	F	car	big	35.7
3	M	bicycle	small	33.0
4	F	car	other	32.5
5	M	car	other	32.1
6	M	motorbike	medium	32.0
7	M	neither	other	31.0
8	F	bicycle	other	30.9
9	M	motorbike	big	30.5
10	M	motorbike	other	30.3
11	M	bicycle	other	30.2

Departamento de Ciencias de la Computación y la Decisión

Conteo de métodos de pago por order_id

```
SELECT
  PAYMENT_METHOD
, COUNT(DISTINCT ORDER_ID) AS ORDER_COUNTS
FROM order_entity
WHERE 1 = 1
GROUP BY PAYMENT_METHOD
ORDER BY ORDER_COUNTS DESC;
```

	PAYMENT_METHOD	ORDER_COUNTS
1	cc	3685510
2	cash	1181065
3	rappi_pay	221032
4	rappi_pay_gateway	13061
5	rappicorp	131
6	nequi	107

Conteo de total órdenes

```
SELECT
  COUNT(DISTINCT ORDER_ID) AS ORDER_COUNTS
FROM order_entity
WHERE 1 = 1;
```

	ORDER_COUNTS
1	5100906

Conteo total de órdenes por usuario

```
SELECT
  COUNT(DISTINCT APPLICATION_USER_ID) AS ORDER_COUNTS
FROM order_entity
WHERE 1 = 1;
```

	ORDER_COUNTS
1	1168462

Conteo de métodos de pago por usuario

```
SELECT
  PAYMENT_METHOD
, COUNT(DISTINCT APPLICATION_USER_ID) AS ORDER_COUNTS
FROM order_entity
WHERE 1 = 1
GROUP BY PAYMENT_METHOD
ORDER BY ORDER_COUNTS DESC;
```

	PAYMENT_METHOD	ORDER_COUNTS
1	cash	647352
2	cc	620647
3	rappi_pay	73111
4	rappi_pay_gateway	1332
5	nequi	90
6	rappicorp	66

Conteo de métodos de pago por rt

```
SELECT
PAYMENT_METHOD
,COUNT(DISTINCT STOREKEEPER_ID) AS ORDER_COUNTS
FROM order_entity
WHERE 1 = 1
GROUP BY PAYMENT_METHOD
ORDER BY ORDER_COUNTS DESC;
```

	PAYMENT_METHOD	ORDER_COUNTS
1	cc	1368499
2	cash	753312
3	rappi_pay	198401
4	rappi_pay_gateway	12591
5	rappicorp	130
6	nequi	98

Conteo total de órdenes por rt

```
SELECT
COUNT(DISTINCT STOREKEEPER_ID) AS ORDER_COUNTS
FROM order_entity
WHERE 1 = 1;
```

	ORDER_COUNTS
1	1505534

Propina mínima

```
SELECT
MIN(TIP) AS TIP_MIN
FROM order_entity;
```

	TIP_MIN
1	0

Suma propina

```
SELECT
SUM(TIP) AS TIP_MIN
FROM order_entity;
```

	TIP_MIN
1	5111857955

Departamento de Ciencias de la Computación y la Decisión

Propina máxima

```
SELECT  
MAX(TIP) AS TIP_MAX  
FROM order_entity;
```

TIP_MAX	
1	40000

Propina promedio

```
SELECT  
AVG(TIP) AS TIP_AVG  
FROM order_entity;
```

TIP_AVG	
1	1002.14706073784

Valor mínimo de la orden

```
SELECT  
MIN(TOTAL_VALUE) AS TOTAL_VALUE_MIN  
FROM order_entity;
```

TOTAL_VALUE_MIN	
1	0

Suma de valor de todas las órdenes

```
SELECT  
SUM(TOTAL_VALUE) AS TOTAL_VALUE  
FROM order_entity;
```

TOTAL_VALUE	
1	240522109246.689

Valor máximo de la orden

```
SELECT  
MAX(TOTAL_VALUE) AS TOTAL_VALUE_MAX  
FROM order_entity;
```

TOTAL_VALUE_MAX	
1	16840854

Valor promedio de la orden

```
SELECT  
AVG(TOTAL_VALUE) AS TOTAL_VALUE_AVG  
FROM order_entity;
```

TOTAL_VALUE_AVG	
1	47152.8213314829

Departamento de Ciencias de la Computación y la Decisión

Valor promedio de la orden por método de pago

```
SELECT  
PAYMENT_METHOD  
,ROUND(AVG(TOTAL_VALUE),0) AS TOTAL_VALUE_AVG  
FROM order_entity  
GROUP BY PAYMENT_METHOD;
```

	PAYMENT_METHOD	TOTAL_VALUE_AVG
1	cash	35940.0
2	cc	50907.0
3	nequi	133995.0
4	rappi_pay	43126.0
5	rappi_pay_gateway	68966.0
6	rappicorp	65109.0

Valor mínimo de la orden por método de pago

```
SELECT  
PAYMENT_METHOD  
,MIN(TOTAL_VALUE) AS TOTAL_VALUE_MIN  
FROM order_entity  
GROUP BY PAYMENT_METHOD;
```

	PAYMENT_METHOD	TOTAL_VALUE_MIN
1	cash	0
2	cc	0
3	nequi	8968
4	rappi_pay	0
5	rappi_pay_gateway	0
6	rappicorp	0

Suma total de las órdenes por método de pago

Departamento de Ciencias de la Computación y la Decisión

```
SELECT  
PAYMENT_METHOD  
,SUM(TOTAL_VALUE) AS TOTAL_VALUE_SUM  
FROM order_entity  
GROUP BY PAYMENT_METHOD;
```

	PAYMENT_METHOD	TOTAL_VALUE_SUM
1	cash	42446955586.8608
2	cc	187619371012.09
3	nequi	14337468
4	rappi_pay	9532150032.95991
5	rappi_pay_gateway	900765933.100001
6	rappicorp	8529213.69

Valor máximo de la orden por método de pago

```
SELECT  
PAYMENT_METHOD  
,MAX(TOTAL_VALUE) AS TOTAL_VALUE_MAX  
FROM order_entity  
GROUP BY PAYMENT_METHOD;
```

	PAYMENT_METHOD	TOTAL_VALUE_MAX
1	cash	4367000
2	cc	16840854
3	nequi	426200
4	rappi_pay	5285000
5	rappi_pay_gateway	6297800
6	rappicorp	358900

Duración máxima de una orden (en minutos)

```
SELECT  
MAX(CAST((JULIANDAY(CLOSED_AT) - JULIANDAY(CREATED_AT))*24*60 AS INTEGER)) AS DURATION_ORDER_MAX  
FROM order_entity  
WHERE 1 = 1  
AND CLOSED_AT IS NOT NULL;
```

	DURATION_ORDER_MAX
1	24738

Departamento de Ciencias de la Computación y la Decisión

Duración mínima de una orden (en minutos)

```
SELECT  
MIN(CAST((JULIANDAY(CLOSED_AT) - JULIANDAY(CREATED_AT))*24*60 AS INTEGER)) AS DURATION_ORDER_MIN  
FROM order_entity  
WHERE 1 = 1  
AND CLOSED_AT IS NOT NULL;
```

DURATION_ORDER_MIN	
1	0

Duración promedio de una orden (en minutos)

```
SELECT  
AVG(CAST((JULIANDAY(CLOSED_AT) - JULIANDAY(CREATED_AT))*24*60 AS INTEGER)) AS DURATION_ORDER_AVG  
FROM order_entity  
WHERE 1 = 1  
AND CLOSED_AT IS NOT NULL;
```

DURATION_ORDER_AVG	
1	61.8894106634581

Duración mínima de una orden por método de pago (en minutos)

```
SELECT  
PAYMENT_METHOD  
,MIN(CAST((JULIANDAY(CLOSED_AT) - JULIANDAY(CREATED_AT))*24*60 AS INTEGER)) AS DURATION_ORDER_MIN  
FROM order_entity  
WHERE 1 = 1  
AND CLOSED_AT IS NOT NULL  
GROUP BY PAYMENT_METHOD;
```

PAYMENT_METHOD		DURATION_ORDER_MIN
1	cash	0
2	cc	0
3	nequi	6
4	rappi_pay	4
5	rappi_pay_gateway	4
6	rappicorp	16

Duración máxima de una orden por método de pago (en minutos)

```
SELECT
PAYMENT_METHOD
,MAX(CAST((JULIANDAY(CLOSED_AT) - JULIANDAY(CREATED_AT))*24*60 AS INTEGER)) AS DURATION_ORDER_MAX
FROM order_entity
WHERE 1 = 1
AND CLOSED_AT IS NOT NULL
GROUP BY PAYMENT_METHOD;
```

	PAYMENT_METHOD	DURATION_ORDER_MAX
1	cash	24738
2	cc	19676
3	nequi	177
4	rappi_pay	10891
5	rappi_pay_gateway	7322
6	rappicorp	5295

Duración promedio de una orden por método de pago (en minutos)

```
SELECT
PAYMENT_METHOD
,AVG(CAST((JULIANDAY(CLOSED_AT) - JULIANDAY(CREATED_AT))*24*60 AS INTEGER)) AS DURATION_ORDER_AVG
FROM order_entity
WHERE 1 = 1
AND CLOSED_AT IS NOT NULL
GROUP BY PAYMENT_METHOD;
```

Departamento de Ciencias de la Computación y la Decisión

	PAYMENT_METHOD	DURATION_ORDER_AVG
1	cash	51.9978349857885
2	cc	64.9345080299578
3	nequi	32.9897959183673
4	rappi_pay	61.5499875192989
5	rappi_pay_gateway	90.3327548122436
6	rappicorp	210.853846153846

Conteo de productos ordenados

```
SELECT PRODUCT_NAME, COUNT(ORDER_ID) as Numero_Ordenes
From store_entity
GROUP BY 1;
```

259	3 cordilleras mestiza 330 ml	136
260	3 cordilleras mona 330 ml	20
261	3 cordilleras mulata 330 ml	141
262	3 cordilleras negra 330 ml	116
263	3 cordilleras rosada 330 ml	136
264	3 deditos de queso mas gaseosa 1.5	784
265	3 muslos de pollo	56
266	3 no son tantos	3
267	3 pizzas medianas	12
268	3 pizzas personal queso y jamon	296
269	3 presas con francesa	4
270	3 salchipapas sencillas	296
271	3 salchipapas trifasica	296
272	3 taco	1
273	3 x 2 premium roll	69

Departamento de Ciencias de la Computación y la Decisión

Conteo de productos diferentes por ciudad

```
SELECT CITY, COUNT(DISTINCT(PRODUCT_NAME)) as Productos
From store_entity
GROUP BY 1;
```

	CITY	Productos
1	Armenia	571
2	Barranquilla	3815
3	Bello	104
4	Bogotá	42314
5	Bucaramanga	924
6	Cali	1359
7	Cartagena	749
8	Chía	145
9	Cúcuta	519
10	Ibagué	142
11	Manizales	1004
12	Medellín	10979
13	Monteria	89
14	Mosquera/ Funza	595
15	Neiva	16

Valor mínimo de cada producto

```
SELECT PRODUCT_NAME, min(PRICE) as Valor_minimo
From store_entity
GROUP BY 1;
```

	PRODUCT_NAME	Valor_minimo
7	&jsw pkm ltsgo eevee	1
8	&mi primer pandereta	42900
9	&mi primera trompeta	42900
10	&pandereta panda	42900
11	&piano jirafa	42900
12	*pv arroz blanco surtimax 11350 g	32500
13	----- salchimanguera pqte kilo	7100
14	1 krusty arepa + gaseosa 400ml	14500
15	1 litro de helado	11000
16	1 natilla porción + 2 buñuelos pequeños	3800
17	1 pechuga entera	16000
18	1 perro suizo y gaseosa	15000
19	1 pierna	5900
20	1 salchipapa sencilla y 1 perro sencillo	11000
21	1 x pollo asado artesanal	35500

Departamento de Ciencias de la Computación y la Decisión

Valor máximo de cada producto

```
SELECT PRODUCT_NAME, max(PRICE) as Valor_maximo
From store_entity
GROUP BY 1;
```

PRODUCT_NAME	Valor_maximo
1 natilla porción + 2 buñuelos pequeños	3800
1 pechuga entera	16000
1 perro suizo y gaseosa	15000
1 pierna	5900
1 salchipapa sencilla y 1 perro sencillo	11000
1 x pollo asado artesanal	35500
1/2 carne	13000
1/2 carne gratinada	13000
1/2 costilla bbq	16500
1/2 costillita bbq	14000
1/2 lomo fino	10000
1/2 macarrones 4 quesos	15900
1/2 macarrones con ragú de cerdo	15900
1/2 molde de arándanos	7000
	7000

Valor promedio de cada producto

```
SELECT PRODUCT_NAME, avg(PRICE) as Valor_promedio
From store_entity
GROUP BY 1;
```

PRODUCT_NAME	Valor_promedio
1202b bikini/ bottom/ zahra lotto	90000.0
1202t bikini/ top/ kary lotto talla l m s	130000.0
1203 one piece/ larah lotto talla l m s	230000.0
1204bf bikini/ bottom full/ trisha lotto talla l m s	90000.0
1204t bikini/ top/ talia lotto talla l m s	170000.0
1205 blouse/ blusa waylla talla m s	170000.0
1206 skirt/ falda/ dianthus lotto talla m s	270000.0
1207 cover kimono/ yetty lotto talla m s	360000.0
1208 one piece/ alice provenza talla l m s	280000.0
1209b bikini/ bottom/ zahra provenza talla l m s	90000.0
1209t bikini/ top/ seti provenza talla l m s	110000.0
1210bf bikini/ bottom full/ trisha provenza talla ...	90000.0
1210t bikini/ top/ clau provenza talla m s l	180000.0
1211b bikini/ bottom/ tina provenza talla l m s	100000.0
1211t bikini/ top/ bimori provenza talla l m s	100000.0

Cantidad de tiendas por tipo de comercio

```
SELECT VERTICAL_SUB_GROUP as Tipo_Comercio, COUNT(STORE_ID) as Canitdad_Tiendas  
From store_entity  
GROUP BY 1;
```

	Tipo_Comercio	Canitdad_Tiendas
1	NULL	16035
2	BABIES_KIDS	10
3	EXPRESS	7748
4	FASHION	146
5	GIFTS	28421
6	HEALTH_BEAUTY	1161
7	HOME	364
8	PETS	84700
9	RESTAURANTS	1198150
10	SUPER	503312
11	TECH	1159953

4 MONGODB

4.1 SENTENCIA O CONSULTA DE CREACIÓN DEL DOCUMENTO(s)

Observación: Escriba el código en MongoDB para crear al menos 20 documentos que correspondan a su conjunto de datos específico. Almacene en el repositorio **(REPO EN GITHUB)** el script con el nombre de T1.4.1.Creacion_Documentos.sql

Departamento de Ciencias de la Computación y la Decisión

```
T1.4.1.Creacion_Documentos.mongodb 9+ x
practica_mongo > T1.4.1.Creacion_Documentos.mongodb
Currently connected to cluster0.fdwyd.mongodb.net. Click here to change connection.

1
2
3 use clientes
4 show dbs
   Run Selected Lines From Playground
5 db.clientes.insertOne(
6   {
7     application_user_id : 19182478,
8     gender : "M",
9     register_device : "LG-M400",
10    segment_rfm : "Low Value",
11    is_prime : false
12  }
13 )

TERMINAL  PROBLEMS 30  OUTPUT  DEBUG CONSOLE
MongoDB Enterprise atlas-bedbvn-shard-0:PRIMARY> use clientes
switched to db clientes
MongoDB Enterprise atlas-bedbvn-shard-0:PRIMARY> show dbs
admin 0.000GB
local 5.291GB
myFirstDatabase 0.000GB
MongoDB Enterprise atlas-bedbvn-shard-0:PRIMARY> db.clientes.insertOne(
... {
...   application_user_id : 19182478,
...   gender : "M",
...   register_device : "LG-M400",
...   segment_rfm : "Low Value",
...   is_prime : false
... }
... )
{
  "acknowledged" : true,
  "insertedId" : ObjectId("60a927171a15b482c813ccaa")
}
MongoDB Enterprise atlas-bedbvn-shard-0:PRIMARY>
```

Departamento de Ciencias de la Computación y la Decisión

```
db.clientes.insertMany( [
  { application_user_id : 188467675 , gender : "F" , register_device : "SM-A207M" , segment_rfm : "Low Value" , is_prime : false },
  { application_user_id : 151403266 , gender : "F" , register_device : "IPHONE 7" , segment_rfm : "Low Value" , is_prime : false },
  { application_user_id : 177440560 , gender : "F" , register_device : "IPHONE11" , segment_rfm : "Low Value" , is_prime : false },
  { application_user_id : 1111162047 , gender : "M" , register_device : "XKI-LX3" , segment_rfm : "Low Value" , is_prime : false },
  { application_user_id : 1911476158 , gender : "F" , register_device : "SM-A315G" , segment_rfm : "Potential Power" , is_prime : false },
  { application_user_id : 13704866 , gender : "F" , register_device : "IPHONE 6" , segment_rfm : "Low Value" , is_prime : false },
  { application_user_id : 142564570 , gender : "M" , register_device : "SM-G570M" , segment_rfm : "Potential Diamond" , is_prime : true },
  { application_user_id : 1911532770 , gender : "M" , register_device : "DUB-LX3" , segment_rfm : "Undefined Segment RFM" , is_prime : false },
  { application_user_id : 1710393562 , gender : "F" , register_device : "MRD-LX3" , segment_rfm : "Regular" , is_prime : false },
  { application_user_id : 199885113 , gender : "M" , register_device : "REDMI NOTE" , segment_rfm : "Potential Power" , is_prime : false },
  { application_user_id : 1111332408 , gender : "" , register_device : "SM-G973F" , segment_rfm : "Regular" , is_prime : false },
  { application_user_id : 1610199741 , gender : "F" , register_device : "IPHONE11" , segment_rfm : "Low Value" , is_prime : false },
  { application_user_id : 156830306 , gender : "F" , register_device : "ANE-LX3" , segment_rfm : "Low Value" , is_prime : false },
  { application_user_id : 162137761 , gender : "F" , register_device : "ANE-LX3" , segment_rfm : "Low Value" , is_prime : false },
  { application_user_id : 1011104563 , gender : "M" , register_device : "REDMI NOTE 8" , segment_rfm : "Low Value" , is_prime : false },
  { application_user_id : 118629996 , gender : "F" , register_device : "IPHONE11" , segment_rfm : "Low Value" , is_prime : false },
  { application_user_id : 193950962 , gender : "F" , register_device : "SM-G935F" , segment_rfm : "Low Value" , is_prime : true },
  { application_user_id : 173239679 , gender : "F" , register_device : "SM-A305G" , segment_rfm : "Potential Power" , is_prime : true },
  { application_user_id : 155241361 , gender : "M" , register_device : "IPHONE X" , segment_rfm : "Potential Power" , is_prime : false },
  { application_user_id : 193386463 , gender : "M" , register_device : "SM-N920G" , segment_rfm : "Low Value" , is_prime : false },
  { application_user_id : 112418674 , gender : "F" , register_device : "SM-A750G" , segment_rfm : "Potential Power" , is_prime : false },
  { application_user_id : 183614295 , gender : "F" , register_device : "SM-J500M" , segment_rfm : "Low Value" , is_prime : false },
  { application_user_id : 185634065 , gender : "M" , register_device : "IPHONE 7" , segment_rfm : "Regular" , is_prime : false },
  { application_user_id : 114185223 , gender : "" , register_device : "IPHONE 7" , segment_rfm : "Regular" , is_prime : false },
  { application_user_id : 1711167156 , gender : "M" , register_device : "SM-A715F" , segment_rfm : "Regular" , is_prime : false },
  { application_user_id : 1111289365 , gender : "F" , register_device : "SM-A305G" , segment_rfm : "Regular" , is_prime : false },
  { application_user_id : 154524779 , gender : "F" , register_device : "ASUS_X00HD" , segment_rfm : "Low Value" , is_prime : false },
  { application_user_id : 153232888 , gender : "F" , register_device : "ANE-LX3" , segment_rfm : "Low Value" , is_prime : false },
  { application_user_id : 193759562 , gender : "M" , register_device : "IPHONE 7" , segment_rfm : "Regular" , is_prime : false },
  { application_user_id : 181821740 , gender : "M" , register_device : "IPHONE 8" , segment_rfm : "Power" , is_prime : true },
] );
```

4.2 SENTENCIA DE CONSULTA

Observación: Realice la exploración básica de los datos, conteos totales y por categorías, máximos, promedio y mínimos. Es decir, aplique estadística descriptiva con el fin de conocer las propiedades de los datos y entenderlos lo mejor posible. Use solamente sentencias SQL. Anexe las tomas de pantalla donde evidencie la sentencia SQL y su correspondiente ejecución. Además, **Almacene en el repositorio (REPO EN GITHUB) el script con el nombre de T1.4.2.Consultar_Datos.sql**

Departamento de Ciencias de la Computación y la Decisión

```
Run Selected Lines From Playground
71 db.clientes.aggregate( [
72 {
73   $group: {
74     _id: null,
75     count: { $sum: 1 }
76   }
77 }
78 ] )
79

TERMINAL PROBLEMS 30 OUTPUT DEBUG CONSOLE

Mongo(application_user_id : 181821740 , storekeeper_id : 420356 , created_at : "20
... ] );lication_user_id : 181821740 , storekeeper_id : 420356 , created_at : "20
{
  "acknowledged" : true,
  "insertedIds" : [
    ObjectId("60a927881a15b482c813ccc9"),
    ObjectId("60a927881a15b482c813ccca"),
    ObjectId("60a927881a15b482c813cccb"),
    ObjectId("60a927881a15b482c813cccc"),
    ObjectId("60a927881a15b482c813cccd"),
    ObjectId("60a927881a15b482c813ccce"),
    ObjectId("60a927881a15b482c813cccf")
  ]
}
MongoDB Enterprise atlas-bedbvn-shard-0:PRIMARY> db.clientes.aggregate( [
... {
...   $group: {
...     _id: null,
...     count: { $sum: 1 }
...   }
... }
... ] )
{ "_id" : null, "count" : 31 }
MongoDB Enterprise atlas-bedbvn-shard-0:PRIMARY> |
```

```
82 db.clientes.aggregate( [ { $group :
83   { _id : "$is_prime",
84     cantidad_prime: { $sum: 1 }
85   }
86   },
87   {
88     $sort : {cantidad_prime : -1}
89   }
90 ] )
91
92

TERMINAL PROBLEMS 30 OUTPUT DEBUG CONSOLE

MongoDB Enterprise atlas-bedbvn-shard-0:PRIMARY> db.clientes.aggregate( [ { $group :
...   { _id : "$is_prime"
...     ,cantidad_prime: { $sum: 1 }
...   }
...   },
...   {
...     $sort : {cantidad_prime : -1}
...   }
... ] )
{ "_id" : false, "cantidad_prime" : 28 }
{ "_id" : true, "cantidad_prime" : 3 }
MongoDB Enterprise atlas-bedbvn-shard-0:PRIMARY> |
```

5 ANÁLISIS DE LECTURA

Observación: Considerando el artículo: “The Definitive Guide to Graph Databases for the RDBMS Developer” de Neo4J. Compartido en las carpeta de lecturas recomendadas. Analice y responda cada pregunta en máximo 150 palabras:

1. **¿Cuáles son las limitaciones, que se pueden inferir de la lectura, para migrar los conjuntos de datos relacionales a NoSQL?**
 - La refactorización de la base de datos puede tardar semanas o meses.
 - El problema fundamental con el modelo relacional desnormalizado es su resistencia a la rápida evolución que los negocios actuales desarrollan.
 - Muchas bases de datos relacionales no están diseñadas y optimizadas para exportar grandes cantidades de datos en un corto período de tiempo, entonces el proceso puede detenerse de manera importante.

2. **¿Cuáles limitaciones adicionales se deben considerar, a parte de las mencionadas en el artículo?**
 - Transacciones ACID: Con las bases de datos NoSQL para lograr mayor rendimiento y escalabilidad no cuentan con todas las características de atomicidad, consistencia, aislamiento y durabilidad
 - Algunas soluciones NoSQL no cuentan con transacciones, entonces por ejemplo si voy a insertar dos documentos juntos, tendría que validar en dos pasos para verificar que se hayan insertado con éxito.
 - Las bases de datos SQL cuentan con claves externas con coherencia para garantizar que la base de datos esté siempre en un estado válido, mientras que la mayoría de bases de datos NoSQL no tienen tal garantía.

3. **¿Cuáles son las razones (criterios) que se deben considerar para migrar un conjunto de datos relacionados a NoSQL?**
 - Cuando se requiere hacer consultas sobre relaciones y conexiones entre los datos.
 - Rendimiento y la capacidad de respuesta de las consultas, ya que en una base de datos relacional a medida que aumenta el tamaño del conjunto de datos el rendimiento se deteriora.
 - Ciclos de desarrollo acelerados. Cuando se pueden tener cambios y modificaciones frecuentes en el esquema.
 - Capacidad de respuesta a los cambios internos y externos en las empresas, ya que un conjunto de datos NoSQL permite evolucionar al mismo ritmo que la aplicación, sin poner en peligro la funcionalidad actual.

Departamento de Ciencias de la Computación y la Decisión

- Algunas bases de datos NoSQL brindan transaccionalidad, alta disponibilidad, escalabilidad y almacenamiento de grandes volúmenes de datos.