**Analítica de Grandes Datos Facultad de Minas**

**Universidad Nacional de Colombia**

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*](https://www.kaggle.com/datasets) *o* [*https://arxiv.org/*](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, envío por*** [***https://forms.gle/h7ty3yZykaUq5m7y6***](https://forms.gle/h7ty3yZykaUq5m7y6)

1. **Comprensión del negocio**
   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 rappitendero***, ***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. **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. **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.

1. **Comprensión de los datos**
   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 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.

* 1. **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:**

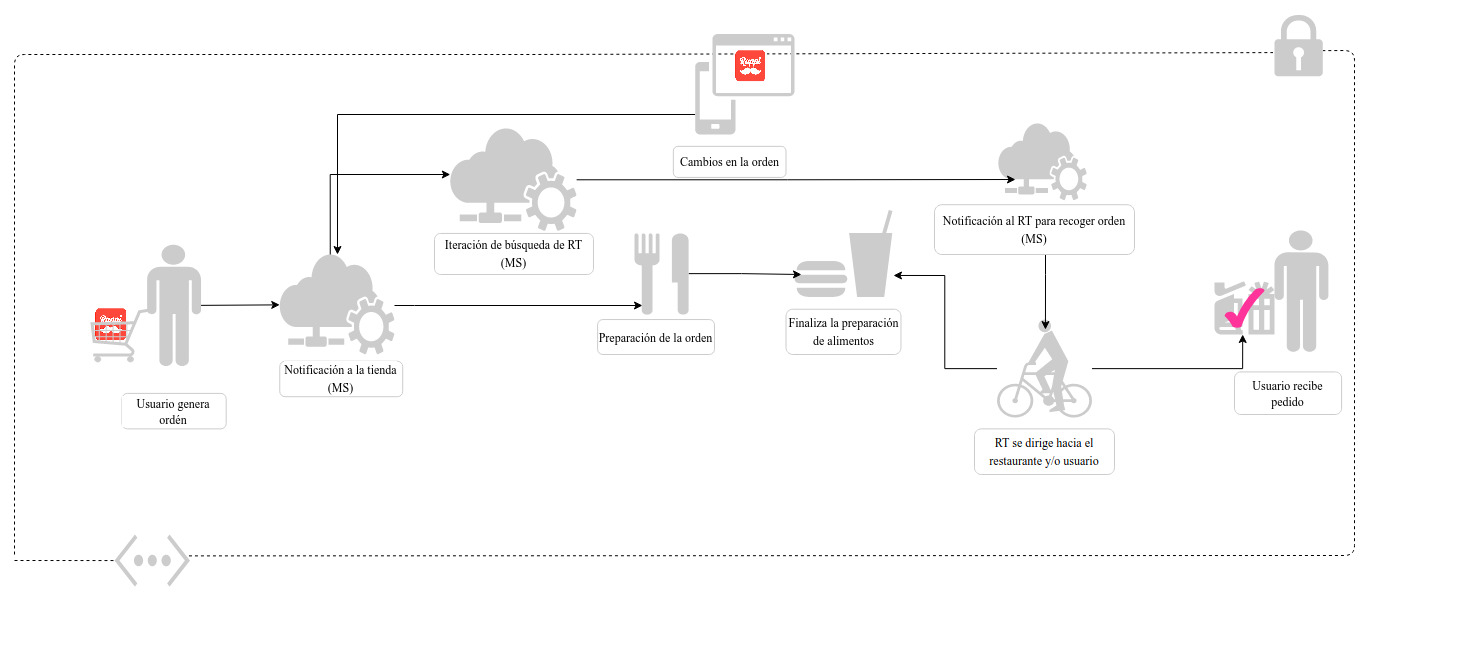
|  |  |  |
| --- | --- | --- |
| **Nombre del atributo / variable** | **Formato o Tipo de Dato** | **Descripción** |
| STORE\_ID | INTEGER | Identificador de la tienda |
| ORDER\_ID | INTEGER | Identificador de la orden |
| CITY | TEXT | Ciudad |
| PRICE | INTEGER | Precio de la orden |
| PRODUCT\_NAME | TEXT | Nombre del producto |
| VERTICAL\_SUB\_GROUP | TEXT | Tipo de establecimiento |
| 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 |

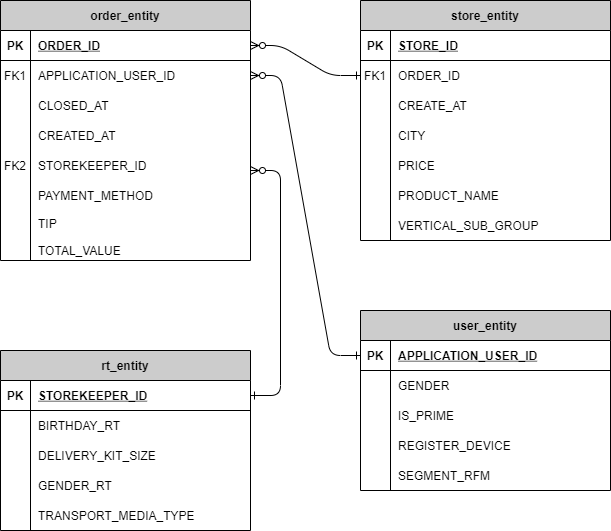
* 1. **Modelo del dominio**

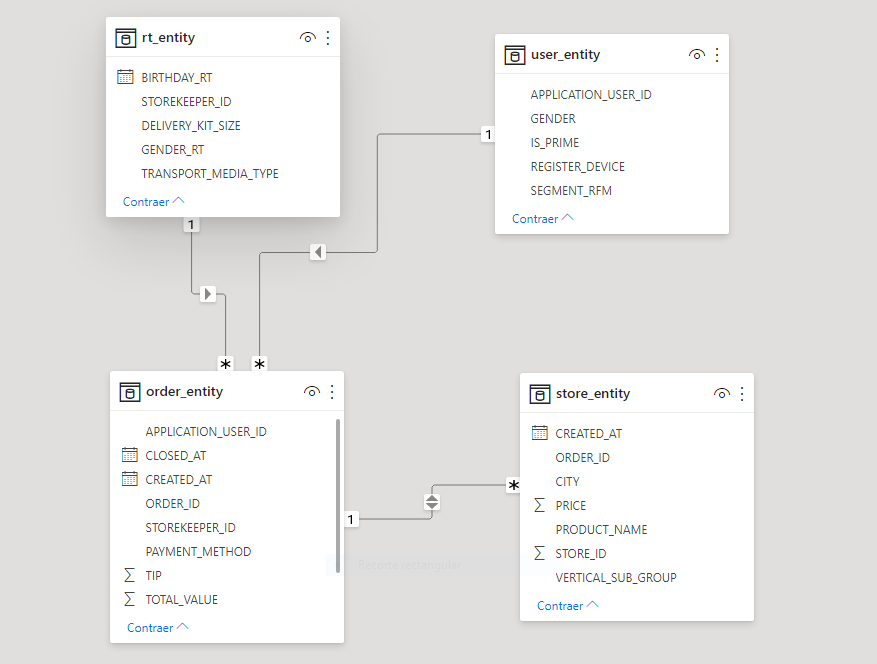
La forma en que se realiza extracción de los datos está soportada sobre la aplicación móvil de Rappi basado en una infraestructura de apis que permiten levantar la información correspondiente. A continuación, se explicará a grandes rasgos cómo es el flujo de ejecución de una orden: Una vez el usuario 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.



1. **Modelo Entidad-Relación**
   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.*

**

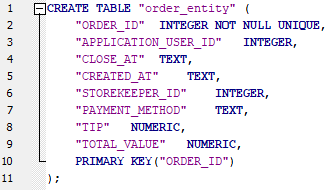
**

* 1. **Sentencia o consulta de creación del tabla(s)**

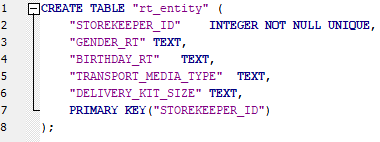
***Observación****: Escriba el código en el Sistema de Gestión de Bases de Datos Relacionales de su elección (se recomienda SQLite por simplicidad, mediante https://sqlitebrowser.org/) para crear las tablas que corresponda con su conjunto de datos específico. Almacene en el repositorio* **(REPO EN GITHUB)**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

***Se recomienda repasar SQL en*** [***https://www.w3schools.com/sql/default.asp***](https://www.w3schools.com/sql/default.asp)

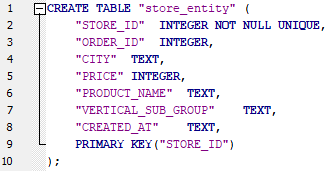
***order\_entity:***

******

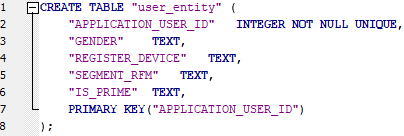
***rt\_entity:***

******

***store\_entity:***

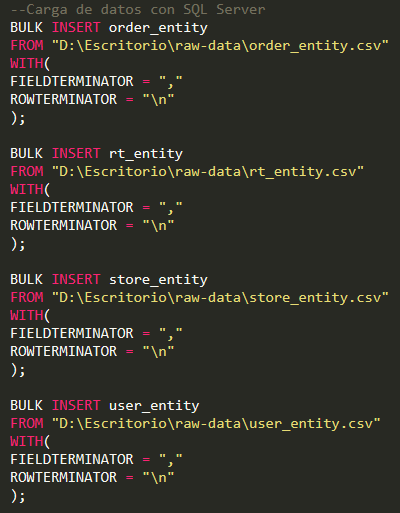
******

***user\_entity:***

******

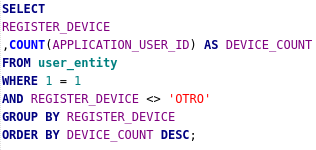
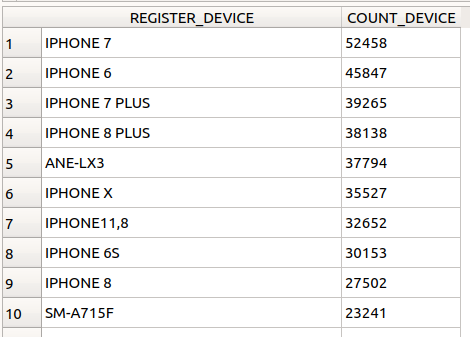
* 1. **Sentencias para Insertar datos**

***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*

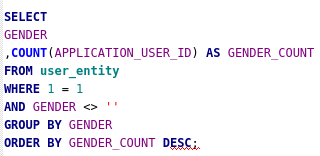
**

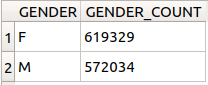
* 1. **Sentencia de consulta**

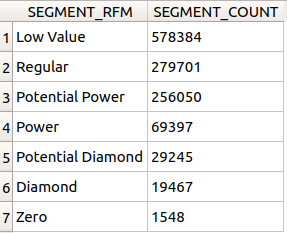
***Observación****: realice la exploración básica de los datos, conteos totales y por categorías, máximos, promedio y mínimos.* Es decir, aplique estadística descriptiva con el fin de conocer las propiedades de los datos y entenderlos lo mejor posible. Use solamente sentencias SQL. Anexe las tomas de 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*

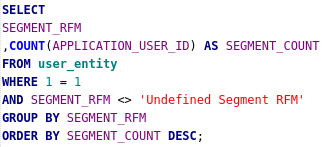
**Conteo de device por application\_user\_id**

**Conteo de gender por application user\_id**

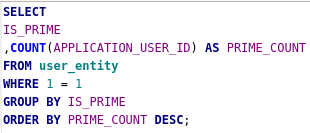
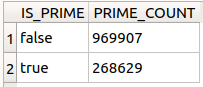




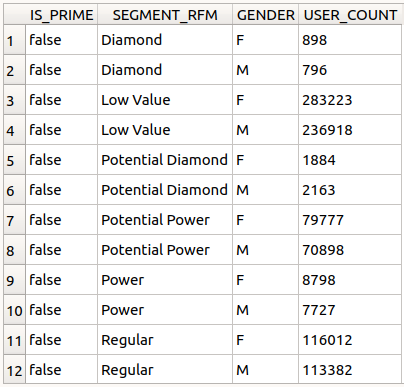
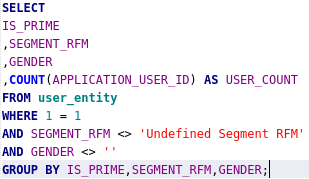
**Conteo de segmento rfm por application user\_id**



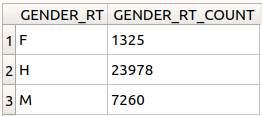
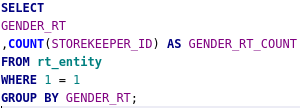
**Conteo de prime por application user\_id**

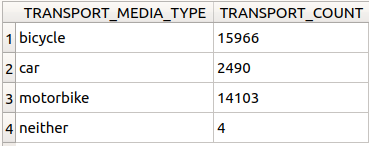
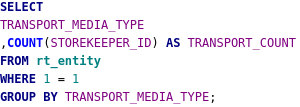


**Conteo de genero, segmento y prime de usuarios**

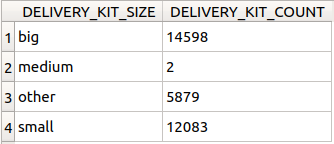


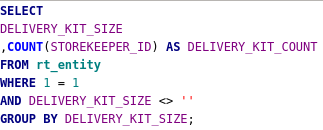
**Conteo de gener de rt por storekeeper\_id**



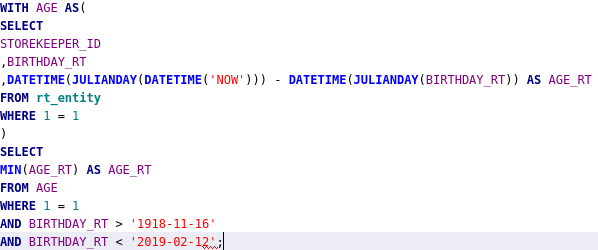
**Conteo de transporte de rt por storekeeper\_id**

**Conteo de kit size de rt por storekeeper\_id**

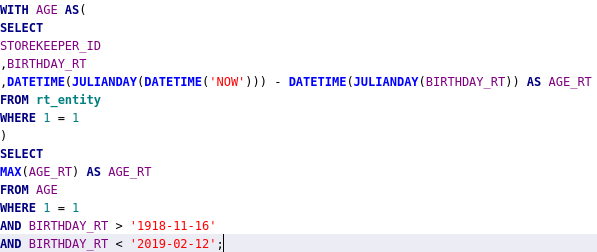


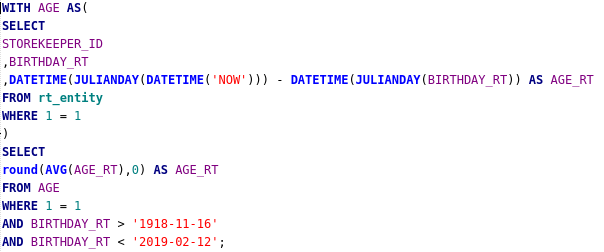


**Edad mínima de rt**

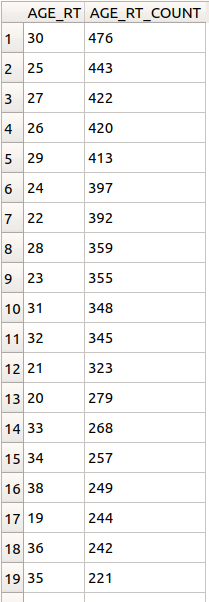


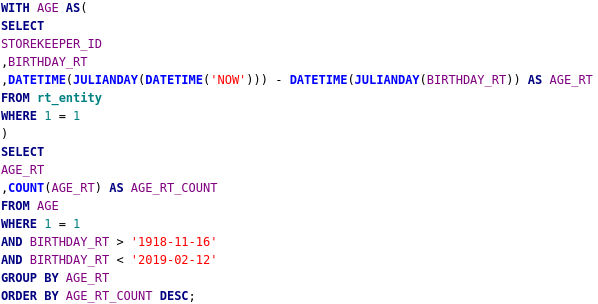
**Edad máxima de rt**

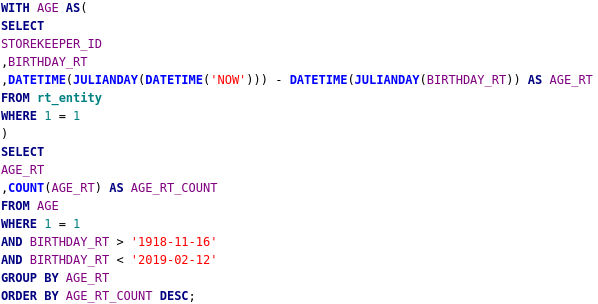


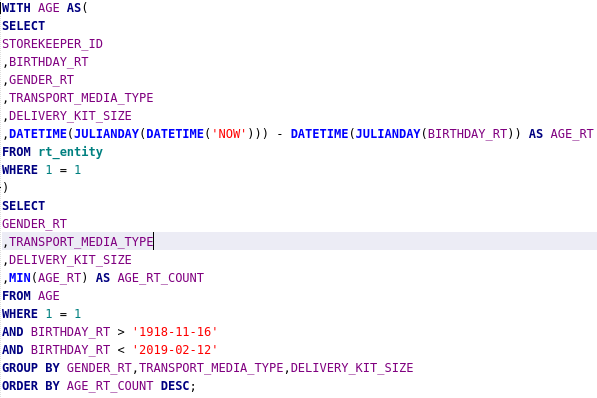
**Edad promedio de rt**

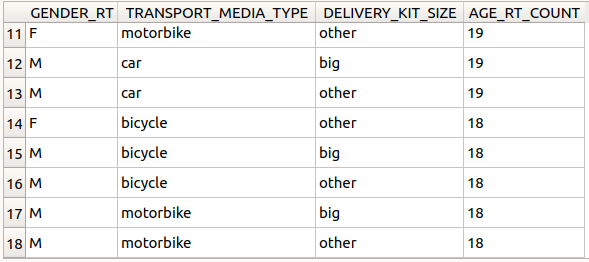


**Conteo edades del rt**

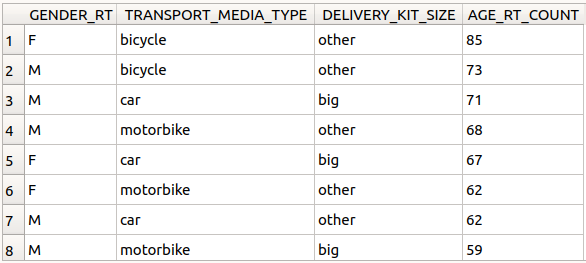
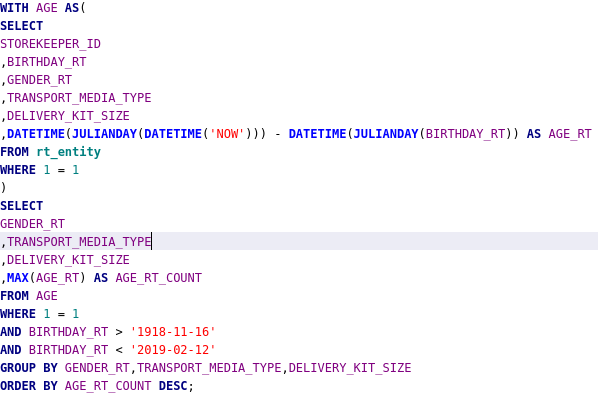


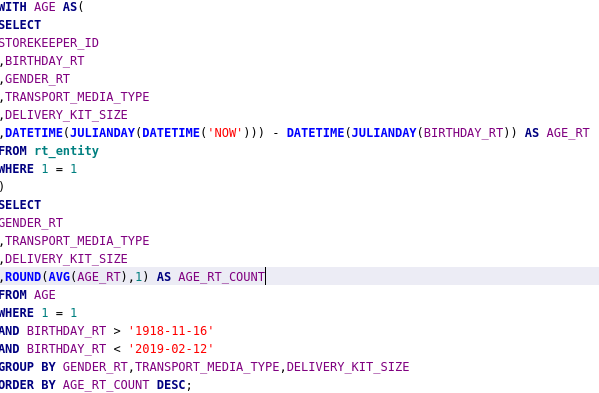


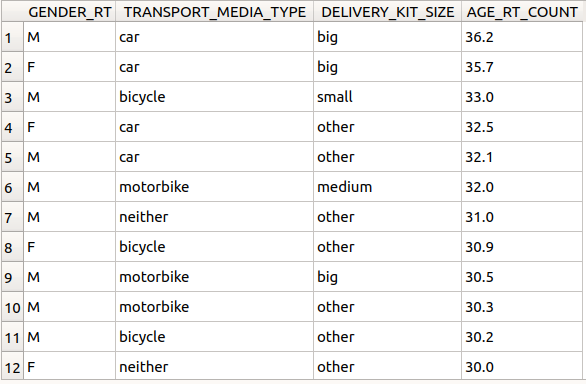
**edad mínima por género edades del rt**

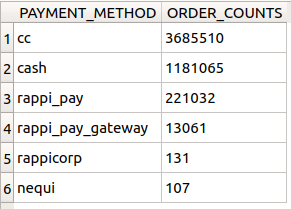


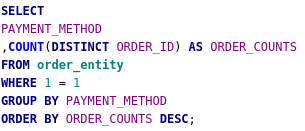
**Edad máxima por género edades del rt**

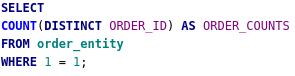


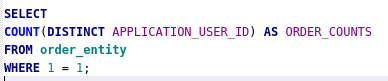
**Edad promedio por género edades del rt**



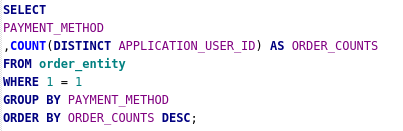
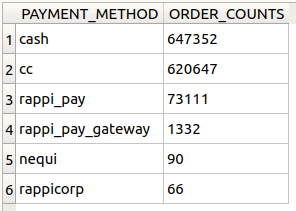
**Conteo de métodos de pago por order\_id**

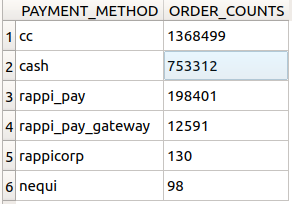


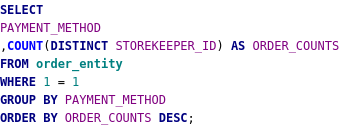
**Conteo de total órdenes** 

**Conteo total de órdenes por usuario**

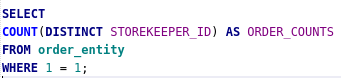
**Conteo de métodos de pago por usuario**



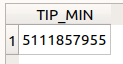
**Conteo de métodos de pago por rt**

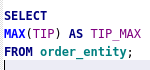
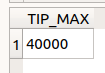


**Conteo total de órdenes por rt**

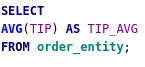


**Propina mínima**

**Suma propina**

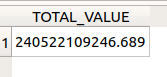
**Propina máxima**

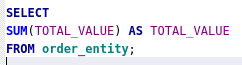
**Propina promedio**

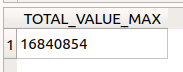
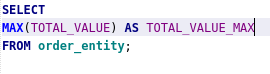


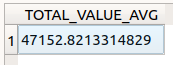
**Valor mínimo de la orden**

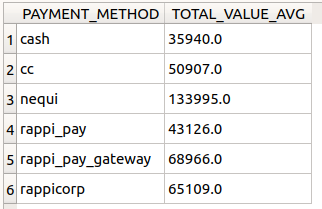
****

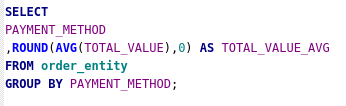
**Suma de valor de todas las órdenes**

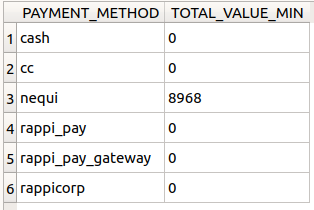


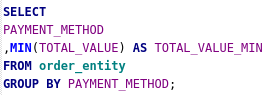
**Valor máximo de la orden**

**Valor promedio de la orden**

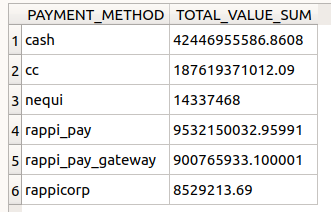
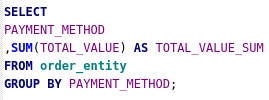
**Valor promedio de la orden por método de pago**



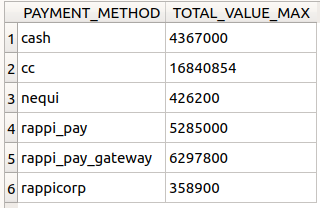
**Valor mínimo de la orden por método de pago**

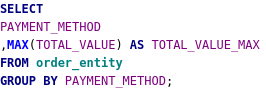


**Suma total de las órdenes por método de pago**

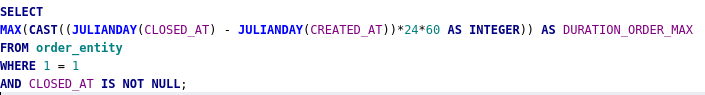


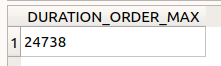
**Valor máximo de la orden por método de pago**



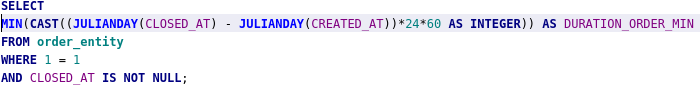


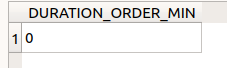
**Duración máxima de una orden (en minutos)**



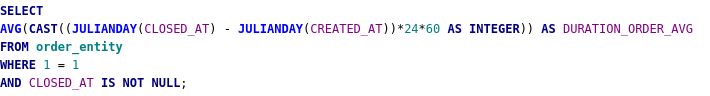
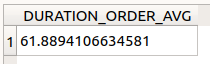


**Duración mínima de una orden (en minutos)**

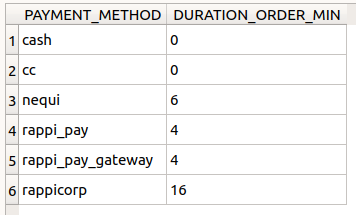
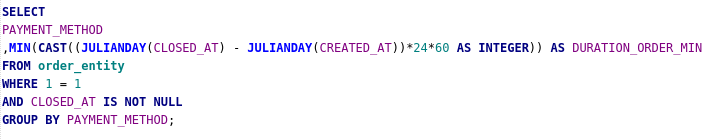




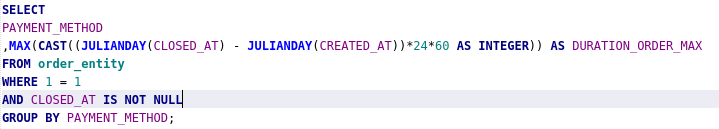
**Duración promedio de una orden (en minutos)**

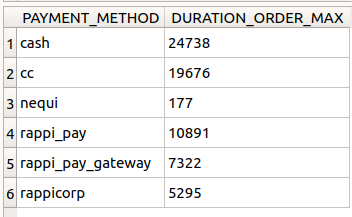


**Duración mínima de una orden por método de pago (en minutos)**

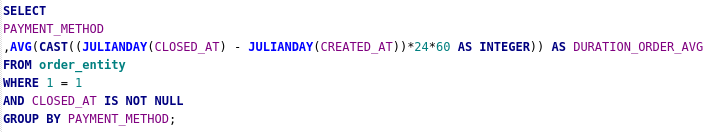


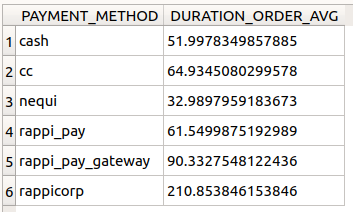
**Duración máxima de una orden por método de pago (en minutos)**



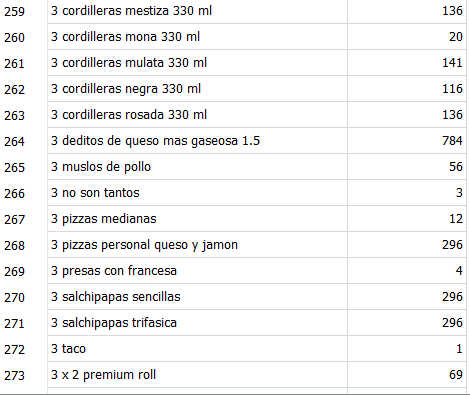


**Duración promedio de una orden por método de pago (en minutos)**

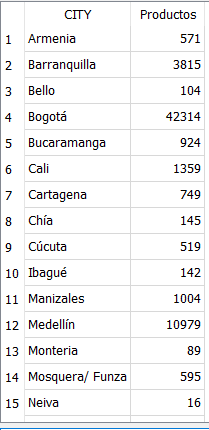




**Conteo de productos ordenados** 



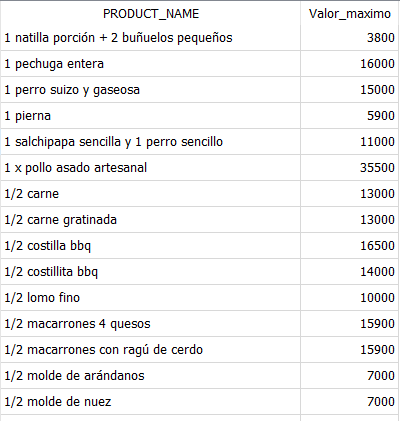
**Conteo de productos diferentes por ciudad**



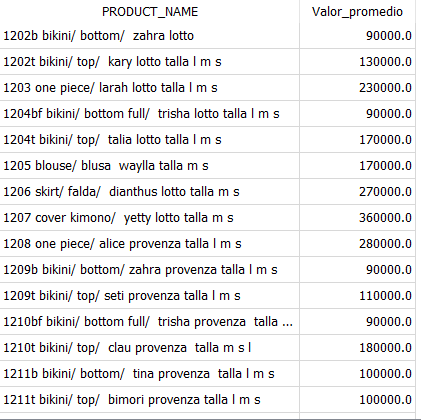
**Valor mínimo de cada producto**



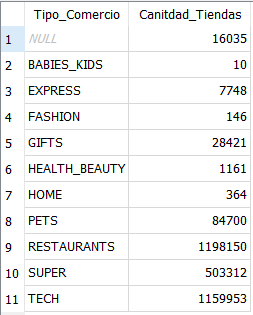
**Valor máximo de cada producto**



**Valor promedio de cada producto**

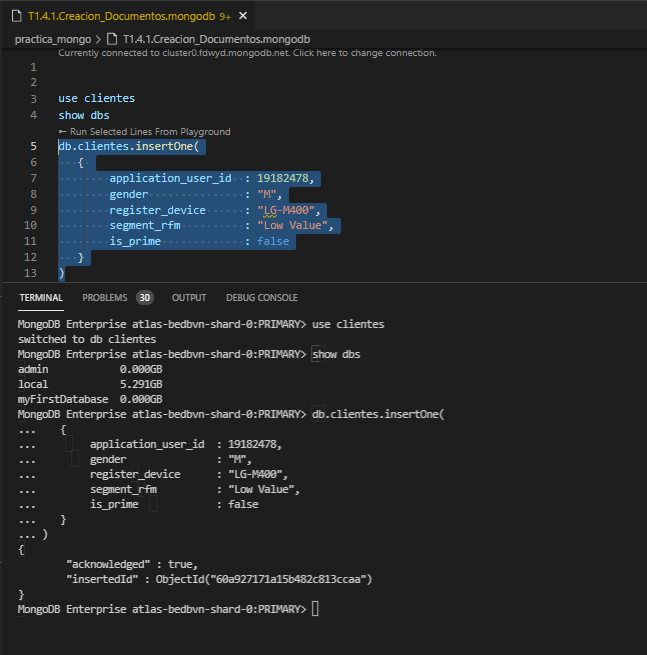


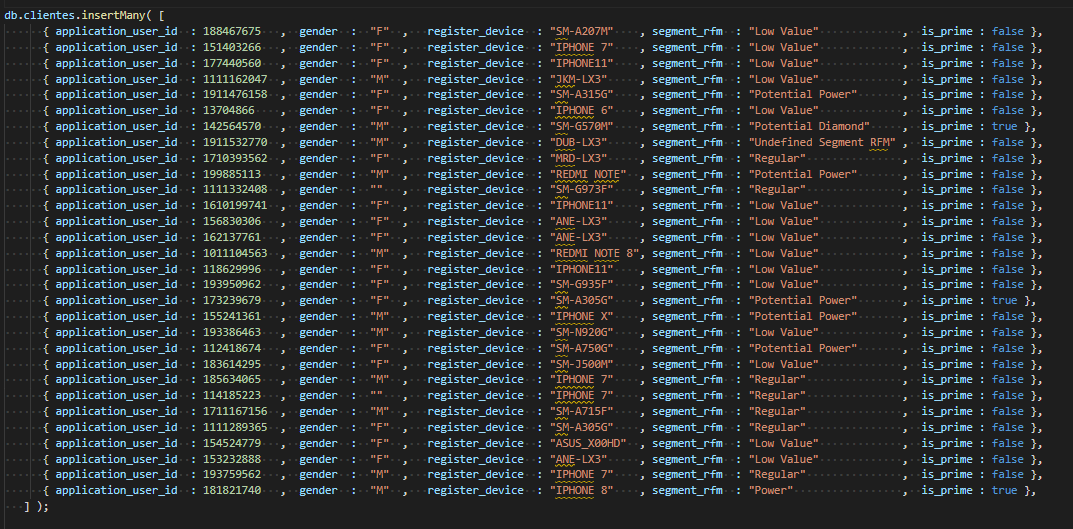
**Cantidad de tiendas por tipo de comercio**



1. **MongoDB**
   1. **Sentencia o consulta de creación del documento(s)**

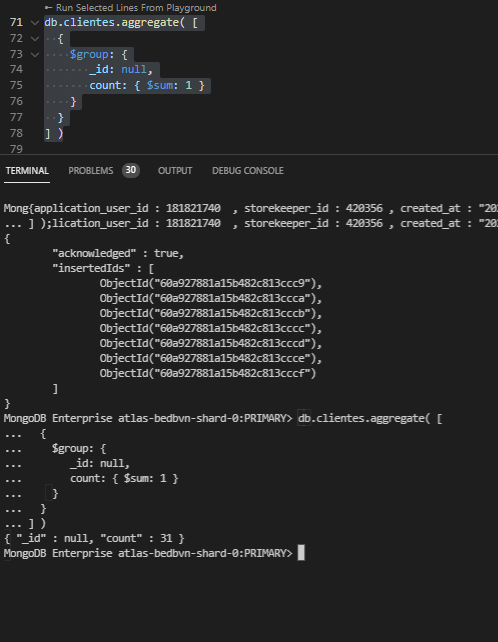
***Observación****: Escriba el código en MongoDB para crear al menos 20 documentos que correspondan a su conjunto de datos específico. Almacene en el repositorio* **(REPO EN GITHUB)** *el script con el nombre de T1.4.1.Creacion\_Documentos.sql*

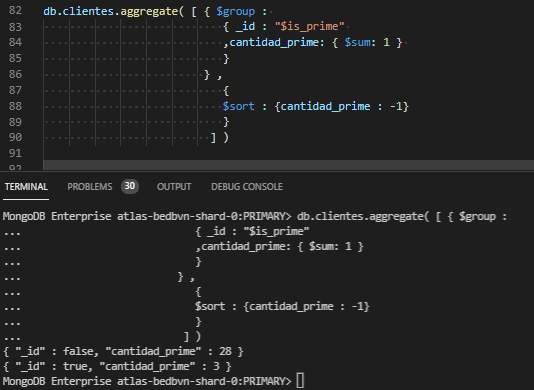
**

**

* 1. **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*





1. **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.

1. **¿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.

1. **¿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 tienen 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.
* Algunas bases de datos NoSQL brindan transaccionalidad, alta disponibilidad, escalabilidad y almacenamiento de grandes volúmenes de datos.