

Proyecto de Bases de datos para modelo perfilamiento de canales de comunicación para los clientes del Banco Falabella

Natalia González Palacios¹, Miguel Angel Niño Vargas²

¹Ingeniera Financiera, ²Ingeniero Mecatrónico,
Universidad Central

Maestría en Analítica de Datos

Curso de Bases de Datos

Bogotá, Colombia

{¹ngonzalezp3@ucentral.edu.co, ²mminov1@ucentral.edu.co

May 26, 2023

Contents

1	Introducción (Max 250 Palabras) - (<i>Primera entrega</i>)	3
2	Características del proyecto de investigación que hace uso de Bases de Datos (Max 500 Palabras) - (<i>Primera entrega</i>)	3
2.1	Titulo del proyecto de investigación (Max 100 Palabras) - (<i>Primera entrega</i>)	4
2.2	Objetivo general (Max 100 Palabras) - (<i>Primera entrega</i>)	4
2.2.1	Objetivos especificos (Max 100 Palabras) - (<i>Primera entrega</i>)	4
2.3	Alcance (Max 200 Palabras) - (<i>Primera entrega</i>)	5
2.4	Pregunta de investigación (Max 100 Palabras) - (<i>Primera entrega</i>) .	5
2.5	Hipotesis (Max 100 Palabras) - (<i>Primera entrega</i>)	5
3	Reflexiones sobre el origen de datos e información (Max 400 Palabras) - (<i>Primera entrega</i>)	7
3.1	¿Cual es el origen de los datos e información ? (Max 100 Palabras) - (<i>Primera entrega</i>)	7
3.2	¿Cuales son las consideraciones legales o eticas del uso de la información? (Max 100 Palabras) - (<i>Primera entrega</i>)	7
3.3	¿Cuales son los retos de la información y los datos que utilizara en la base de datos en terminos de la calidad y la consolidación? (Max 100 Palabras) - (<i>Primera entrega</i>)	8

3.4	¿Que espera de la utilización de un sistema de Bases de Datos para su proyecto? (Max 100 Palabras) - (<i>Primera entrega</i>)	8
4	Diseño del Modelo de Datos del SMBD (Sistema Manejador de Bases de Datos)(<i>Primera entrega</i>)	9
4.1	Características del SMBD (Sistema Manejador de Bases de Datos) para el proyecto (<i>Primera entrega</i>)	10
4.2	Diagrama modelo de datos (<i>Primera entrega</i>)	11
4.3	Imágenes de la Base de Datos (<i>Primera entrega</i>)	12
4.4	Código SQL - lenguaje de definición de datos (DDL) (<i>Primera entrega</i>)	18
4.5	Código SQL - Manipulación de datos (DML) (<i>Primera entrega</i>) . .	20
4.6	Código SQL + Resultados: Vistas (<i>Primera entrega</i>)	22
4.7	Código SQL + Resultados: Triggers (<i>Primera entrega</i>)	24
4.8	Código SQL + Resultados: Funciones (<i>Primera entrega</i>)	24
4.9	Código SQL + Resultados: procedimientos almacenados (<i>Primera entrega</i>)	25
5	Bases de Datos No-SQL (<i>Segunda entrega</i>)	27
5.1	Diagrama Bases de Datos No-SQL (<i>Segunda entrega</i>)	27
5.2	SMBD utilizado para la Base de Datos No-SQL (<i>Segunda entrega</i>)	28
6	Aplicación de ETL (Extract, Transform, Load) y Bodega de Datos (<i>Tercera entrega</i>)	28
6.1	Ejemplo de aplicación de ETL y Bodega de Datos (<i>Tercera entrega</i>)	29
6.2	Bodega de Datos	30
7	Proximos pasos (<i>Tercera entrega</i>)	32
8	Lecciones aprendidas (<i>Tercera entrega</i>)	32
9	Bibliografía	32

1 Introducción (Max 250 Palabras) - (*Primera entrega*)

Actualmente en el Banco Falabella más específicamente en la Gerencia de Prevención de Fraudes, se identificó junto con el área de analítica de fraudes una oportunidad de mejora en la forma o canal de comunicación que se utiliza para contactar a los clientes cuando se quiere comprobar o validar la identidad de estos. Para dar un poco más de contexto a la problemática, como medida de prevención del fraude, el banco al identificar una transacción catalogada por medidas internas como sospechosa, en dirección a no afectar la experiencia del cliente y evitar un posible fraude, se envía una notificación o alerta por diferentes canales de comunicación como los son: WhatsApp, SMS, Correo electrónico o llamada telefónica al cliente solicitándole confirmación con un “sí” o un “no” si efectivamente esta realizando la transacción o si la desconoce y por tanto está siendo posiblemente víctima de fraude.

Sin embargo, aunque es una medida efectiva en cuanto a la detección del fraude, también es cierto que no todos los clientes están atendiendo a esta notificación y quedan transacciones sin respuesta, y se llegó a pensar que esto puede ser debido a que no se está enviando la notificación por el canal correcto de comunicación o el canal que más atiende el cliente. Algunos supuestos planteados inicialmente, pueden ser por ejemplo la edad del cliente, personas de mayor edad puede ser mejor realizarles una llamada antes que enviarle un mensaje de texto que probablemente nunca va a contestar, o si se identifica que un cliente utiliza constantemente la aplicación del banco, se puede determinar que es un cliente digital y el mejor canal para este perfil es vía WhatsApp.

Por lo anterior, con este proyecto se quiere desarrollar un modelo de Machine Learning que nos ayude a identificar el perfil del cliente, por medio de ciertas variables que nos ayuden a clasificar los hábitos del cliente encaminado a determinar si este es un cliente digital o no y basado en ello que nos ayude a clasificar el canal de comunicación más acertado para cada perfil de cliente. Para ello, contamos con una muestra de clientes del año 2022 con variables de hábito de compra, pagos, uso de la aplicación, canal que atendió o utilizo para comunicarse con el banco, entre otros; y con esto llegar a ser más asertivos y mejorar el nivel de respuesta de estas notificaciones con la confirmación o validación de las transacciones con sospechosa de fraude.

2 Características del proyecto de investigación que hace uso de Bases de Datos (Max 500 Palabras) - (*Primera entrega*)

La fuente de datos que vamos a utilizar proviene de clientes del Banco Falabella, teniendo en cuenta que por seguridad de la información no se contarán con datos sensibles de los clientes como su información personal. Por otro lado toda información que el banco considere que no puede ser mostrada en claro, se le realizara un proceso de encriptación o se utilizarán diccionarios de datos con el

fin de no revelar información relevante.

El set de datos se compone de cincuenta y cuatro variables con quinientos mil registros aproximadamente, que hace parte de una muestra recopilada de clientes activos en el año 2022 con sus principales hábitos en cuanto uso de aplicación, principales hábitos de compra (Presencial y No Presencial) y canales de comunicación con mayor respuesta.

El reto a trabajar es consolidar múltiples fuentes de información de manera que se organicen las diferentes variables categóricas y cuantitativas a nivel de cliente y generar un modelo de bases de datos relacional con el mayor desempeño posible al momento de realizar consultas sobre las tablas.

2.1 Título del proyecto de investigación (Max 100 Palabras) - (Primera entrega)

Modelo de predicción canales de comunicación para los clientes del Banco Falabella

2.2 Objetivo general (Max 100 Palabras) - (Primera entrega)

Desarrollar un modelo de Machine Learning que permita identificar el perfil del cliente en función de sus hábitos de compra, pagos, uso de la aplicación entre otros, con el fin de clasificar el canal de comunicación más adecuado para cada perfil de cliente y mejorar la tasa de respuesta de las notificaciones de validación de transacciones sospechosas de fraude.

2.2.1 Objetivos específicos (Max 100 Palabras) - (Primera entrega)

- Recolectar, limpiar y preparar los datos de los clientes del Banco Falabella del año 2022, incluyendo variables de hábitos de compra, pagos, uso de la aplicación y canal de comunicación utilizado.
- Realizar un análisis factorial exploratorio para identificar las variables más significativas y reducir la dimensionalidad del conjunto de datos, con el fin de optimizar los recursos de entrenamiento del modelo de Machine Learning.
- Realizar una técnica de balanceo de los datos para abordar posibles desequilibrios en la distribución de las clases objetivo y mejorar la calidad de la clasificación.
- Entrenar y evaluar dos modelos de Machine Learning: Máquinas de vectores multiclase y Random Forest, utilizando la librería Scikit-Learn de Python, con el fin de seleccionar el modelo más eficaz para clasificar el canal de comunicación adecuado para cada perfil de cliente.
- Implementar el modelo de Machine Learning seleccionado y evaluar su desempeño en la clasificación de los clientes en diferentes perfiles y canales de comunicación, con el fin de mejorar la tasa de respuesta de las notificaciones de validación de transacciones sospechosas de fraude.

2.3 Alcance (Max 200 Palabras) - (*Primera entrega*)

Los principales retos en la creación de una base de datos relacional para este proyecto, pueden incluir:

- Integración de datos: Uno de los mayores desafíos en la creación de una base de datos relacional es la integración de datos de diferentes fuentes. Es probable que los datos del Banco Falabella se almacenen en diferentes sistemas y bases de datos, lo que puede dificultar la integración y la limpieza de los datos para su análisis.
- Diseño de la base de datos: La creación de una base de datos relacional requiere un diseño cuidadoso que tome en cuenta las relaciones entre diferentes entidades y atributos. Para este proyecto, será importante diseñar una base de datos que permita la clasificación y selección de los canales de comunicación más adecuados para cada perfil de cliente.
- Normalización de datos: La normalización es un proceso crítico en la creación de una base de datos relacional, que implica la eliminación de redundancias y la organización de los datos en tablas separadas. Sin embargo, este proceso puede ser desafiante si los datos no están estructurados adecuadamente o si hay errores o inconsistencias en los datos.
- Escalabilidad: Dado que el modelo de Machine Learning se entrenará utilizando grandes conjuntos de datos, la base de datos debe ser escalable para manejar grandes volúmenes de datos y permitir un entrenamiento eficiente del modelo.
- Seguridad de datos: La seguridad de los datos es una consideración crítica en cualquier proyecto de base de datos, especialmente cuando se trata de datos sensibles de los clientes, como información de transacciones financieras. La base de datos debe ser diseñada con medidas de seguridad adecuadas para garantizar la privacidad y protección de los datos de los clientes.

2.4 Pregunta de investigación (Max 100 Palabras) - (*Primera entrega*)

¿Cómo se puede mejorar la selección del canal de comunicación para la validación de transacciones sospechosas de fraude en clientes bancarios mediante el uso de técnicas de análisis de datos y modelos de Machine Learning?

2.5 Hipotesis (Max 100 Palabras) - (*Primera entrega*)

Al desarrollar un modelo de Machine Learning utilizando técnicas de análisis de datos para identificar el perfil del cliente y el canal de comunicación más adecuado para la validación de transacciones sospechosas de fraude en clientes bancarios, se logrará mejorar la eficacia en la respuesta del cliente, reducir el

riesgo de fraude y aumentar la satisfacción del cliente al utilizar un canal de comunicación personalizado y adecuado a sus hábitos y preferencias.

3 Reflexiones sobre el origen de datos e información

(Max 400 Palabras) - (*Primera entrega*)

Para el desarrollo de este proyecto se tuvo que realizar una reunión con la Gerencia de Prevención de Fraudes, en primer lugar para determinar la necesidad actual del área y por otro lado reflexionar acerca de los datos que se podían utilizar para avanzar con la idea planteada. Teniendo en cuenta estas consideraciones, se llegó a la conclusión de utilizar diversas fuentes de información con una muestra de datos representativa de clientes activos en el año 2022 con información relevante sobre sus hábitos de compra y comunicación con el banco.

Sin embargo, se debe tener en cuenta, la importancia de la protección de la privacidad y seguridad de la información de los clientes. Es por ello que se tomaron medidas para garantizar la confidencialidad de los datos sensibles utilizando técnicas de encriptación o diccionarios de datos para proteger información relevante que no debe ser revelada. Por lo anterior, es indispensable tener en cuenta que la obtención de datos personales de los clientes puede generar preocupaciones éticas y legales si no se toman las medidas adecuadas para su protección y uso responsable. Es necesario seguir reflexionando y trabajando en el desarrollo de prácticas éticas y responsables en el manejo de datos en el contexto actual de la era digital.

3.1 ¿Cual es el origen de los datos e información ? (Max 100 Palabras) - (*Primera entrega*)

La fuente de datos que vamos a utilizar proviene de clientes del Banco Falabella, teniendo en cuenta que por seguridad de la información no se contarán con datos sensibles de los clientes como su información personal. Por otro lado toda información que el banco considere que no puede ser mostrada en claro, se le realizará un proceso de encriptación o se utilizarán diccionarios de datos con el fin de no revelar información relevante. El set de datos se compone de cincuenta y cuatro variables con quinientos mil registros aproximadamente, que hace parte de una muestra recopilada de clientes activos en el año 2022 con sus principales hábitos en cuanto uso de aplicación, principales hábitos de compra (Presencial y No Presencial) y canales de comunicación con mayor respuesta. Esta información se encuentra con disponibilidad dentro de la organización en cuatro fuentes: En primer lugar dos tablas (Notificaciones, Transacción Financiera) alojadas en el servicio de Google Cloud Platform GCP; en segundo lugar la tabla del Onboarding alojada en SQL Server y por último la tabla de comunicaciones disponibilizada en un archivo de excel. Para la consolidación y cruce de estas bases se utilizará el lenguaje de programación Python y se cargarán los datos desde un archivo plano en formato csv.

3.2 ¿Cuales son las consideraciones legales o eticas del uso de la información? (Max 100 Palabras) - (*Primera entrega*)

En este caso, se pueden considerar las siguientes consideraciones legales y éticas:

- **Protección de datos personales:** Es importante garantizar que la información de los clientes sea tratada de manera confidencial y se cumplan con las leyes y regulaciones de protección de datos personales. Es necesario asegurar que la información sea utilizada solamente para los fines establecidos en el proyecto y que no se utilice para otros fines sin el consentimiento del titular de los datos.
- **No discriminación:** Al utilizar los datos para determinar el canal de comunicación más adecuado para cada cliente, se debe asegurar que no se realice discriminación por razones de género, edad, etnia o cualquier otro factor. El modelo de Machine Learning debe ser justo y no sesgado.
- **Precisión del modelo:** Es importante que el modelo de Machine Learning utilizado para la clasificación de clientes sea preciso y justo. Debe evitarse que el modelo clasifique erróneamente a un cliente como un posible fraude, lo que podría afectar negativamente la experiencia del cliente.
- **Respeto a la privacidad:** Se debe respetar la privacidad de los clientes y no utilizar la información obtenida para otros fines diferentes al proyecto. También se debe tener en cuenta que el análisis de los datos debe ser realizado por personal autorizado y capacitado en el manejo de información confidencial.

3.3 ¿Cuales son los retos de la información y los datos que utilizara en la base de datos en terminos de la calidad y la consolidación? (Max 100 Palabras) - (Primera entrega)

Uno de los retos principales en este caso es la integración de los datos provenientes de las cuatro fuentes de información que están alojadas en diferentes servicios. Dado que cada servicio puede tener su propia estructura y formato de datos, es necesario llevar a cabo un proceso de consolidación y homogenización para garantizar la calidad de los datos y su correcta integración en la base de datos relacional. Además, puede haber problemas de redundancia y duplicación de datos, lo que podría afectar la integridad y coherencia de la información. Por lo tanto, el reto consiste en establecer un proceso robusto de integración y limpieza de los datos para garantizar la calidad y consistencia de la información.

3.4 ¿Que espera de la utilización de un sistema de Bases de Datos para su proyecto? (Max 100 Palabras) - (Primera entrega)

En primer lugar, un sistema de bases de datos puede ayudar a gestionar grandes cantidades de información de manera organizada y estructurada. Esto facilita la búsqueda y el acceso a la información relevante, así como la eliminación de datos innecesarios.

En segundo lugar, un sistema de bases de datos puede mejorar la seguridad

y privacidad de la información, ya que puede implementar mecanismos de autenticación y autorización para limitar el acceso a los datos sensibles.

En tercer lugar, un sistema de bases de datos puede mejorar la eficiencia y velocidad del procesamiento de datos, lo que puede ser importante para aplicaciones que necesiten acceder a grandes cantidades de información en tiempo real.

4 Diseño del Modelo de Datos del SMBD (Sistema Manejador de Bases de Datos) (*Primera entrega*)

A la hora de seleccionar una base de datos pequeña, existen dos opciones populares: MySQL y Oracle. Ambas son bases de datos relacionales y ofrecen características similares, pero hay diferencias clave que pueden influir en la elección según las necesidades específicas.

En primer lugar, MySQL es una opción más económica, ya que Oracle tiene un alto costo de licencia y mantenimiento, mientras que MySQL es de código abierto y gratuita. Esto puede ser una ventaja para nosotros lo que nos facilita su uso y no requerimos de las características más avanzadas de Oracle.

En segundo lugar, MySQL se destaca por su facilidad de uso y capacidad de escalabilidad. Es una base de datos muy fácil de instalar y configurar, y puede manejar grandes volúmenes de datos y usuarios simultáneos. Oracle, por otro lado, puede requerir más conocimientos técnicos para su instalación y configuración, y es menos escalable que MySQL en algunas situaciones.

En tercer lugar, MySQL cuenta con una gran comunidad de desarrolladores y una amplia gama de herramientas de soporte, además de que se contribuye activamente al desarrollo de nuevas herramientas y aplicaciones para la base de datos. Además, existen muchos recursos de soporte disponibles en línea para MySQL, como documentación detallada, foros de discusión y tutoriales en línea.

Por último, MySQL es conocida por su compatibilidad con una amplia gama de lenguajes de programación. Tiene una API que permite la conexión con varios lenguajes de programación, lo que facilita el trabajo. Oracle, aunque también tiene una comunidad activa, no es tan amplia como la de MySQL y puede ser más difícil encontrar recursos de soporte en línea.

En conclusión y para nuestro caso, nuestra elección será MySQL, gracias a las facilidades que está nos ofrece.

4.1 Características del SMBD (Sistema Manejador de Bases de Datos) para el proyecto (*Primera entrega*)

MySQL es un sistema de gestión de bases de datos (SGBD) muy popular y ampliamente utilizado en todo el mundo. Se caracteriza por ser una base de datos relacional de código abierto, lo que significa que es gratuita para descargar y usar. Además, MySQL es conocida por su facilidad de uso y su capacidad para escalar y manejar grandes volúmenes de datos y usuarios simultáneos. También cuenta con una gran comunidad de desarrolladores y una amplia gama de herramientas de soporte, lo que la hace una opción atractiva para pequeñas y medianas empresas.

Entre las características más destacadas de MySQL, podemos mencionar su velocidad y eficiencia en la gestión de grandes cantidades de datos, su capacidad para trabajar con una amplia variedad de lenguajes de programación, su alta disponibilidad y escalabilidad, y su compatibilidad con diferentes sistemas operativos.

En cuanto a las características específicas de las bases de datos MySQL, podemos destacar su capacidad para soportar múltiples transacciones concurrentes y su alta seguridad en el manejo de datos. También cuenta con una amplia variedad de herramientas y plugins para facilitar la gestión y el mantenimiento de la base de datos. Otras características incluyen la posibilidad de realizar copias de seguridad y restauraciones de bases de datos, la compatibilidad con diferentes motores de almacenamiento, y la capacidad de realizar consultas complejas en grandes conjuntos de datos de manera rápida y eficiente.

4.2 Diagrama modelo de datos (*Primera entrega*)

El diagrama modelo de datos es una representación visual que muestra la estructura de una base de datos y las relaciones entre las diferentes tablas que la componen. En el caso de las tablas mencionadas anteriormente (User, Transaction, Fraud, Commerce, Country y Genre), el diagrama modelo de datos permitiría visualizar cómo se relacionan las diferentes entidades, sus atributos y cómo se organizan los datos dentro de ellas. Este tipo de diagrama es una herramienta importante para diseñar una base de datos y garantizar su integridad y eficiencia.

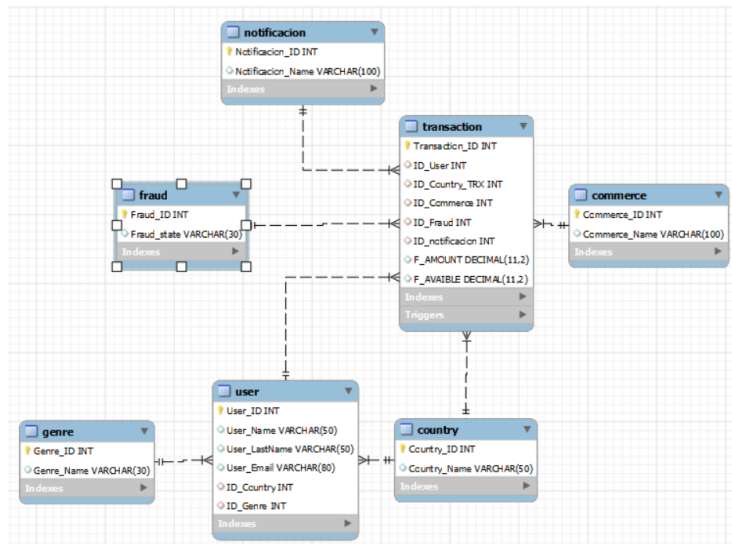
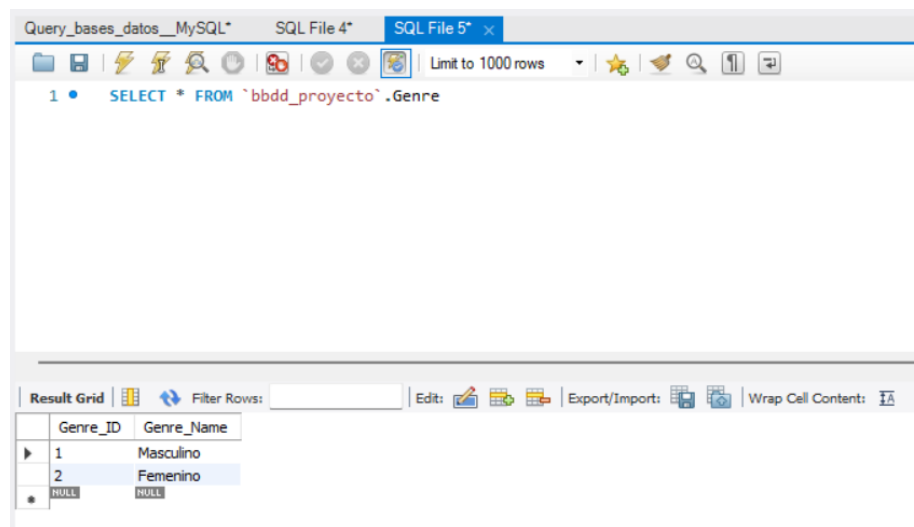


Figure 1: EER Diagram

4.3 Imágenes de la Base de Datos (*Primera entrega*)

En cualquier sistema de gestión de bases de datos (DBMS), la creación del esquema de la base de datos es uno de los primeros pasos críticos en el desarrollo de cualquier aplicación. El esquema de la base de datos define la estructura de las tablas y cómo se relacionan entre sí. En este sentido, se presentan las tablas creadas en MySQL para la gestión de fraudes para un reconocido banco. Las tablas presentadas son User, Transaction, Commerce, Fraud, Country y Genre, y se incluye una imagen de cada una de ellas para ilustrar su estructura.

Imágenes:



The screenshot shows a MySQL query editor window with the following components:

- Query Editor:** Contains the SQL query: `SELECT * FROM `bbdd_proyecto`.Genre`
- Result Grid:** Displays the structure of the 'Genre' table with the following data:

Genre_ID	Genre_Name
1	Masculino
2	Femenino
NULL	NULL

Figure 2: Tabla Genero

Query_bases_datos__MySQL* SQL File 4* SQL File 5* x

Limit to 1000 rows

1 • SELECT * FROM `bdd_proyecto`.`User`

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:

	User_ID	User_Name	User_LastName	User_Email	ID_Country	ID_Genre
▶	1	Nicholas	MacCosto	nmaccosto0@cdbaby.com	50	2
	2	Lotte	Coghlan	lcoghlan1@mozilla.com	2	1
	3	Pavlov	Heinonen	pheinonen2@engadget.com	23	2
	4	Anni	Killough	akillough3@godaddy.com	44	1
	5	Artair	Avramovich	aavramovich4@nationalgeographic.com	31	2
	6	Leonard	McAndie	lmcandie5@plala.or.jp	33	2
	7	Sauncho	Cicchitello	scicchitello6@exblog.jp	10	2
	8	Haleigh	McNess	hmcness7@w3.org	22	1
	9	Tressa	Loadsmen	tloadsmen8@goo.gl	11	2
	10	Madeleine	Reyner	mreyner9@nbcnews.com	9	1
	11	Arleen	Litherland	alitherlanda@auda.org.au	22	2
	12	Lilias	Halpeine	lhalpeineb@time.com	23	2
	13	Gerta	Klageman	gklagemanc@sun.com	24	1
	14	Norry	MacIriach	nmaciriachd@rediff.com	33	1
	15	Gleda	Raper	grapere@e-recht24.de	35	2
	16	Marshall	Clampe	mclampef@discuz.net	47	1
	17	Daniele	Cersey	dcerseyg@army.mil	13	2
	18	Cristal	Gutman	cgutmanh@army.mil	6	2
	19	Delano	Ordelt	dordelti@wikipedia.org	20	1
	20	Elvira	Iskower	eiskowerj@pagesperso-orange.fr	39	2
	21	Abby	Glozman	aglozmank@weather.com	31	1
	22	Clarine	Venning	cvenningl@opera.com	46	1
	23	Elga	Crilley	ecrilleym@walmart.com	50	2
	24	Penny	Pray	pprayn@adobe.com	23	1
	25	Cori	Illiston	cillistono@mysql.com	32	1
	26	Wye	Bruton	wbrutonp@wp.com	38	1

User 3 x

Figure 3: Tabla User

Query_bases_datos__MySQL* SQL File 4* SQL File 5*

Limit to 1000 rows

1 • SELECT * FROM `bbdd_proyecto`.Transaction

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Co

	Transaction_ID	ID_User	ID_Country_TRX	ID_Commerce	F_AMOUNT	F_AVAIBLE
▶	1	44	31	4	13994.86	69142.90
	2	4	42	31	12590.08	4397.07
	3	23	22	11	95644.45	46864.40
	4	4	4	31	36019.94	33118.32
	5	6	23	33	2428.16	34770.31
	6	39	37	41	46453.36	76956.06
	7	48	2	33	45521.54	93606.47
	8	5	49	25	37867.41	2190.92
	9	46	17	24	76590.11	79925.29
	10	42	35	30	51213.33	19333.19
	11	50	6	5	38102.87	23012.60
	12	7	35	17	15797.47	61293.84
	13	26	15	48	33785.20	24975.50
	14	17	36	22	88000.29	74198.56
	15	9	14	29	3545.07	5518.42
	16	12	9	38	10438.18	41168.95
	17	19	1	19	54653.70	97589.86
	18	39	24	3	19815.66	90597.41
	19	16	20	44	16503.80	92527.23
	20	15	46	25	6392.62	17234.44
	21	5	14	30	91664.57	94699.98
	22	30	3	41	90708.18	21838.25
	23	29	37	6	10801.35	94834.40
	24	23	11	12	34156.66	13597.83
	25	35	38	6	94069.60	30155.78
	26	32	35	22	68824.54	28027.01

Transaction 4 ×

Figure 4: Tabla Transaction

Query_bases_datos__MySQL* SQL File 4* SQL File 5* x

Limit to 1000 rows

1 • SELECT * FROM `bdd_proyecto`.Country

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content

	Country_ID	Country_Name
▶	1	China
	2	Portugal
	3	Portugal
	4	Cameroon
	5	Argentina
	6	China
	7	Thailand
	8	Czech Republic
	9	China
	10	China
	11	United States
	12	Bolivia
	13	Russia
	14	Greece
	15	Indonesia
	16	Tunisia
	17	Indonesia
	18	China
	19	China
	20	China
	21	Russia
	22	China
	23	Chile
	24	United Kingdom
	25	Russia
	26	Russia

Country 5 x

Figure 5: Tabla Country

Query_bases_datos__MySQL* SQL File 4* SQL File 5*

Limit to 1000 rows

1 • SELECT * FROM `bbdd_proyecto`.Commerce

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:

Commerce_ID	Commerce_Name
1	First Trust Latin America AlphaDEX Fund
2	Corporate Office Properties Trust
3	Centrex Inc.
4	Sotheby's
5	ICON plc
6	First Commonwealth Financial Corporation
7	Lakeland Financial Corporation
8	NovoCure Limited
9	IDEX Corporation
10	CPS Technologies Corp.
11	Trevena, Inc.
12	Tecnoglass Inc.
13	Alliance MMA, Inc.
14	Capstead Mortgage Corporation
15	Western Refining Logistics, LP
16	Sarepta Therapeutics, Inc.
17	DWS High Income Opportunities Fund, Inc.
18	Caesars Acquisition Company
19	The Hanover Insurance Group, Inc.
20	RPC, Inc.
21	Allegheny Technologies Incorporated
22	PowerShares DWA Healthcare Momentu...
23	Quaker Chemical Corporation
24	Continental Resources, Inc.
25	National Bank Holdings Corporation
26	Wilhelmina International, Inc.

Commerce 2 x

Figure 6: Tabla Commerce

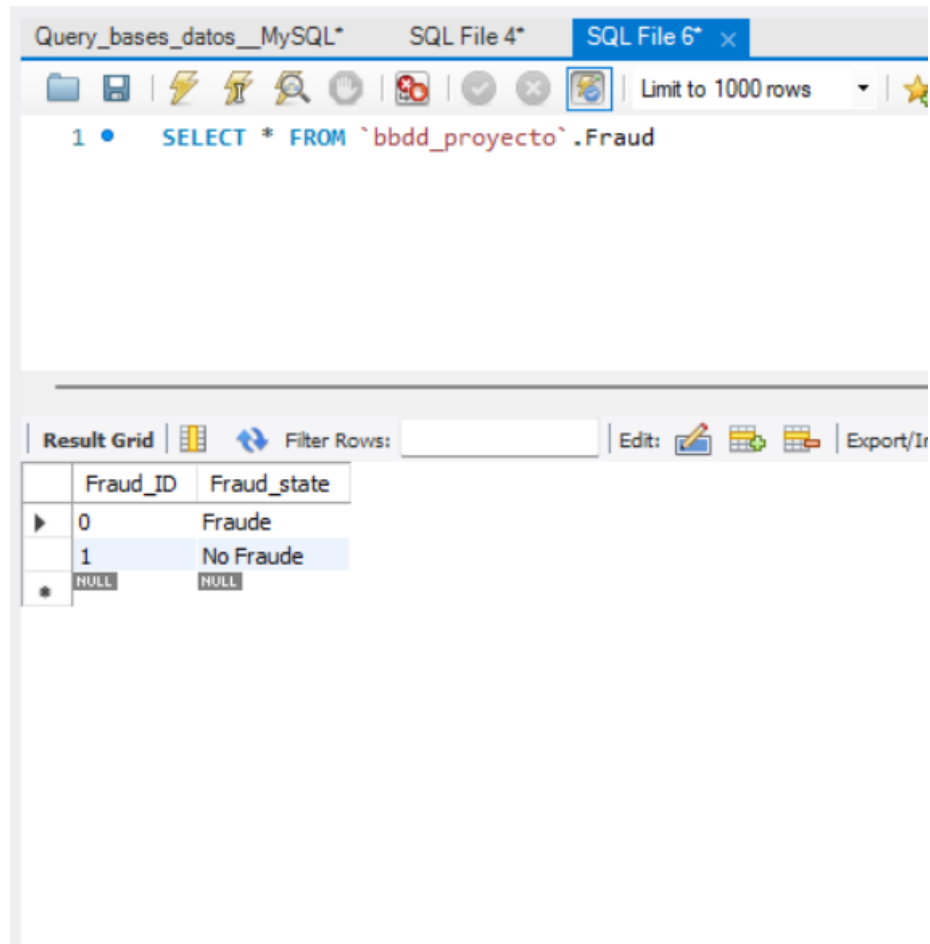


Figure 7: Tabla Fraud

4.4 Código SQL - lenguaje de definición de datos (DDL) (Primera entrega)

El lenguaje de definición de datos (DDL) es un conjunto de comandos utilizados en SQL para definir y manipular la estructura de las tablas y otros objetos de la base de datos. En el caso específico de las tablas User, Transaction, Commerce, Fraud, Country y Genre, creadas en MySQL, se utilizan comandos DDL para definir sus características, como el nombre, el tipo y tamaño de las columnas, las restricciones de integridad y las claves primarias y foráneas.

Los comandos DDL son esenciales para crear, modificar y eliminar las estructuras de las tablas en la base de datos. Además, son utilizados por los administradores de la base de datos y los desarrolladores para asegurarse de que la estructura de las tablas se ajuste a las necesidades del negocio y sea eficiente en el almacenamiento y acceso a la información.

En el siguiente código SQL se muestra cómo se definen las tablas mencionadas utilizando comandos DDL en MySQL.

```
## SE CREA LA BASE DE DATOS
```

```
DROP DATABASE 'bbdd_proyecto';  
CREATE SCHEMA 'bbdd_proyecto' ;
```

```
## SE CREAN LAS TABLAS
```

```
CREATE TABLE 'bbdd_proyecto'.Country (  
    Country_ID int not null AUTO_INCREMENT,  
    Country_Name VARCHAR(50),  
    PRIMARY KEY (Country_ID)  
);
```

```
CREATE TABLE 'bbdd_proyecto'.Genre (  
    Genre_ID int not null AUTO_INCREMENT,  
    Genre_Name VARCHAR(30),  
    PRIMARY KEY (Genre_ID)  
);
```

```
CREATE TABLE 'bbdd_proyecto'.Commerce (  
    Commerce_ID int not null AUTO_INCREMENT,  
    Commerce_Name VARCHAR(100),  
    PRIMARY KEY (Commerce_ID)  
);
```

```
CREATE TABLE 'bbdd_proyecto'.User (  
    User_ID INT(30) not null AUTO_INCREMENT,  
    User_Name VARCHAR(50),
```

```

        User_LastName VARCHAR(50),
        User_Email VARCHAR(80),
        ID_Country INT(50),
        ID_Genre INT(50),
        PRIMARY KEY (User_ID)
    );

CREATE TABLE 'bbdd_proyecto'.Transaction (
    Transaction_ID INT(30) not null AUTO_INCREMENT,
    ID_User INT(50),
    ID_Country_TRX INT(50),
    ID_Commerce INT(80),
    ID_Fraud INT(50),
    F_AMOUNT DECIMAL(11,2),
    F_AVAIBLE DECIMAL(11,2),
    PRIMARY KEY (Transaction_ID)
);

## SE CONFIGURAN LAS LLAVES FORANEAS

ALTER TABLE 'bbdd_proyecto'.User
ADD FOREIGN KEY (ID_Country) REFERENCES
'bbdd_proyecto'.Country(Country_ID);

ALTER TABLE 'bbdd_proyecto'.User
ADD FOREIGN KEY (ID_Genre) REFERENCES
'bbdd_proyecto'.Genre(Genre_ID);

ALTER TABLE 'bbdd_proyecto'.Transaction
ADD FOREIGN KEY (ID_User) REFERENCES
'bbdd_proyecto'.User(User_ID);

ALTER TABLE 'bbdd_proyecto'.Transaction
ADD FOREIGN KEY (ID_Country_TRX) REFERENCES
'bbdd_proyecto'.Country(Country_ID);

ALTER TABLE 'bbdd_proyecto'.Transaction
ADD FOREIGN KEY (ID_Commerce) REFERENCES
'bbdd_proyecto'.Commerce(Commerce_ID);

ALTER TABLE 'bbdd_proyecto'.Transaction
ADD FOREIGN KEY (ID_Fraud)
REFERENCES 'bbdd_proyecto'.Fraud(Fraud_ID);

```

4.5 Código SQL - Manipulación de datos (DML) (*Primera entrega*)

El lenguaje SQL no solo se utiliza para crear y definir las estructuras de las tablas, sino que también permite realizar la manipulación de los datos almacenados en ellas. Para esto, se utiliza el lenguaje de manipulación de datos (DML), el cual se encarga de agregar, modificar, eliminar y consultar los datos almacenados en las tablas.

En el contexto de las tablas creadas en MySQL como User, Transaction, Commerce, Country y Genre, el uso del lenguaje DML es esencial para realizar operaciones como la inserción de nuevos registros de usuarios o transacciones, la actualización de los datos de un usuario o transacción existente, la eliminación de un registro o conjunto de registros específicos, y la realización de consultas para extraer información relevante de los datos almacenados.

A continuación, se presentan algunos ejemplos de código SQL que ilustran cómo se pueden realizar algunas de estas operaciones utilizando el lenguaje DML en el contexto de estas tablas.

OPERACIÓN PARA INSERTAR DATOS

SE CARGAN LOS DATOS DE LA TABLA GENRE

```
INSERT INTO 'bbdd_proyecto'.'genre' ('Genre_ID', 'Genre_Name')
VALUES ('1', 'Masculino');
```

SE CARGAN LOS DATOS DE LA TABLA COUNTRY

```
insert into 'bbdd_proyecto'.Country ('Country_ID', 'Country_Name')
values (1, 'China');
```

SE CARGAN LOS DATOS DE LA TABLA COMMERCE

```
insert into 'bbdd_proyecto'.Commerce ('Commerce_ID', 'Commerce_Name')
values (1, 'First Trust Latin America AlphaDEX Fund');
```

SE CARGAN LOS DATOS DE LA TABLA USER

```
insert into 'bbdd_proyecto'.User
('User_ID', 'User_Name', 'User_LastName', 'User_Email',
 'ID_Country', 'ID_Genre')
values (1, 'Nicholas', 'MacCosto', 'nmaccosto0@cdbaby.com', 50, 2);
```

```

## SE CARGAN LOS DATOS DE LA TABLA FRAUD

INSERT INTO 'bbdd_proyecto'.'Fraud' ('Fraud_ID', 'Fraud_state')
VALUES ('0', 'Fraude');

## SE CARGAN LOS DATOS DE LA TABLA TRANSACTION

insert into 'bbdd_proyecto'.Transaction ('Transaction_ID', 'ID_User',
'ID_Country_trx', 'ID_Commerce', 'ID_Fraud', 'F_AMOUNT', 'F_AVAIBLE')
values (1, 18, 27, 4, 0, 68067.97, 53412.53);

# OPERACIÓN PARA ELIMINAR DATOS

## SE ELIMINAN LOS DATOS DE LA TABLA GENRE

DELETE FROM 'bbdd_proyecto'.'genre' WHERE ('Genre_ID' = '1');

## SE ELIMINAN LOS DATOS DE LA TABLA FRAUD

DELETE FROM 'bbdd_proyecto'.'Fraud' WHERE ('Fraud_ID' = '1');

## SE ELIMINAN LOS DATOS DE LA TABLA COUNTRY

DELETE FROM 'bbdd_proyecto'.'country' WHERE ('Country_ID' = '1');

## SE ELIMINAN LOS DATOS DE LA TABLA COMMERCE

DELETE FROM 'bbdd_proyecto'.'commerce' WHERE ('Commerce_ID' = '1');

## SE ELIMINAN LOS DATOS DE LA TABLA USER

DELETE FROM 'bbdd_proyecto'.'user' WHERE ('User_ID' = '1');

## SE ELIMINAN LOS DATOS DE LA TABLA TRANSACTION

DELETE FROM 'bbdd_proyecto'.'transaction' WHERE ('Transaction_ID' = '1');

# OPERACIÓN PARA MODIFICAR DATOS

## SE MODIFICAN LOS DATOS DE LA TABLA GENRE

UPDATE 'bbdd_proyecto'.'genre' SET 'Genre_Name' = 'Nuevo valor'

```

```

WHERE ('Genre_ID' = '1');

## SE MODIFICAN LOS DATOS DE LA TABLA FRAUD

UPDATE 'bdd_proyecto'.'Fraud' SET 'Fraud_State' = 'Nuevo valor'
WHERE ('Fraud_ID' = '1');

## SE MODIFICAN LOS DATOS DE LA TABLA COUNTRY

UPDATE 'bdd_proyecto'.'country' SET 'Country_Name' = 'Nuevo valor'
WHERE ('Country_ID' = '1');

## SE MODIFICAN LOS DATOS DE LA TABLA COMMERCE

UPDATE 'bdd_proyecto'.'commerce' SET 'Commerce_Name' = 'Nuevo valor'
WHERE ('Commerce_ID' = '2');

## SE MODIFICAN LOS DATOS DE LA TABLA USER

UPDATE 'bdd_proyecto'.'user' SET 'User_LastName' = 'Nuevo valor'
WHERE ('User_ID' = '1');

## SE MODIFICAN LOS DATOS DE LA TABLA TRANSACTION

UPDATE 'bdd_proyecto'.'transaction' SET 'ID_User' = 'Nuevo valor'
WHERE ('Transaction_ID' = '1');

```

4.6 Código SQL + Resultados: Vistas (*Primera entrega*)

Cuando trabajamos con bases de datos, es común que necesitemos acceder a una vista específica de los datos, en lugar de acceder a la tabla completa. En SQL, podemos crear vistas que nos permitan ver los datos de una o varias tablas de una manera específica. Las vistas son útiles cuando necesitamos ver solo una parte de los datos o cuando queremos combinar datos de varias tablas en una sola vista.

En el caso de las tablas mencionadas, se pueden crear vistas para obtener información específica de los datos almacenados en ellas. Por ejemplo, se pueden crear vistas para mostrar información resumida de las transacciones realizadas por un usuario en particular, o para mostrar el total de ventas de un comercio en un país específico.

Ejemplo:

```

CREATE VIEW 'bdd_proyecto'.transactions_per_user AS
SELECT ID_User, COUNT(*) AS num_transactions

```

```
FROM 'bbdd_proyecto'.Transaction  
GROUP BY ID_User;
```

The screenshot shows a MySQL query editor with the following query:

```
1 • SELECT * FROM `bbdd_proyecto`.transactions_per_user  
2 ORDER BY num_transactions DESC
```

The results are displayed in a table with the following columns: ID_User and num_transactions. The data is sorted in descending order of num_transactions.

ID_User	num_transactions
39	5
26	3
30	3
3	2
4	2
5	2
23	2
18	2
19	2
12	2
44	2
46	2
15	1
16	1
17	1
22	1
2	1
29	1
6	1
32	1

Figure 8: View Transactions Per User

4.7 Código SQL + Resultados: Triggers (*Primera entrega*)

Los Triggers en SQL son un conjunto de acciones que se activan automáticamente en respuesta a un cambio en una tabla o vista. Es decir, son un tipo de procedimiento almacenado que se ejecuta automáticamente en respuesta a un evento en una tabla, como una inserción, actualización o eliminación de datos. Estos desencadenadores son muy útiles cuando se necesita ejecutar una acción automáticamente en función de ciertas condiciones en la base de datos. En otras palabras, son una forma de automatizar tareas en una base de datos.

En el caso de las tablas User, Transaction, Commerce, Country y Genre mencionadas anteriormente, se pueden crear diferentes tipos de Triggers para realizar diversas acciones, como validar la integridad de los datos antes de su inserción, actualizar automáticamente un registro cuando se modifique otro relacionado, o incluso realizar cálculos complejos y operaciones lógicas en la base de datos.

Ejemplo: Se crea un Trigger para validar que el Cliente exista en la tabla User cuando se ingrese un nuevo registro en la tabla Transaction.

```
DELIMITER $$
CREATE TRIGGER 'bbdd_proyecto'.'check_user_id'
BEFORE INSERT ON 'bbdd_proyecto'.Transaction
FOR EACH ROW
BEGIN
    DECLARE user_count INT;

    SELECT COUNT(*) INTO user_count FROM 'bbdd_proyecto'.
    User WHERE User_ID = NEW.ID_User;

    IF user_count = 0 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Error: User_ID does not exist in User table';
    END IF;
END$$
```

4.8 Código SQL + Resultados: Funciones (*Primera entrega*)

El lenguaje SQL es un estándar utilizado para gestionar y manipular datos en bases de datos relacionales. Las funciones en SQL son herramientas útiles para realizar cálculos y operaciones en los datos almacenados en tablas, y pueden ser usadas en diferentes partes de una consulta. En el contexto de las tablas mencionadas, como Country, Genre, Fraud, Commerce, User y Transaction, las funciones pueden ser empleadas para generar resultados útiles en consultas complejas y para facilitar el análisis de los datos almacenados

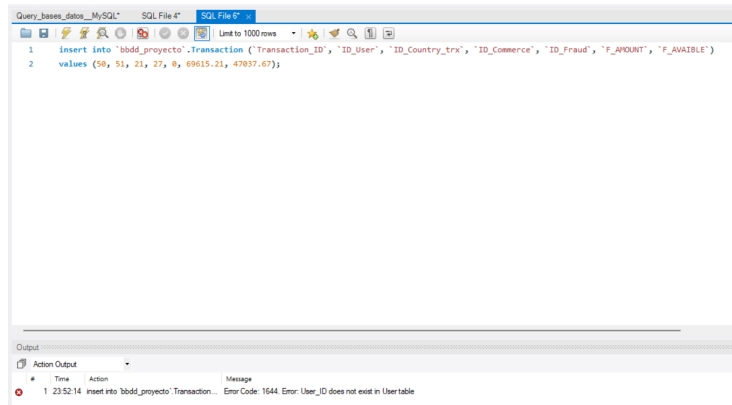


Figure 9: Trigger Check User Id

Ejemplo: Se crea una función que calcula el promedio del monto de las transacciones realizadas por un cliente en específico

```
DELIMITER $$
CREATE FUNCTION `bdd_proyecto`.`average_transaction_amount`(user_id INT)
RETURNS DECIMAL(11,2) DETERMINISTIC READS SQL DATA
BEGIN
    DECLARE total_amount DECIMAL(11,2);
    DECLARE transaction_count INT;

    SELECT SUM(F_AMOUNT) INTO total_amount FROM `bdd_proyecto`.`Transaction`
    WHERE ID_User = user_id;
    SELECT COUNT(*) INTO transaction_count FROM `bdd_proyecto`.`Transaction`
    WHERE ID_User = user_id;

    IF transaction_count = 0 THEN
        RETURN 0;
    ELSE
        RETURN total_amount / transaction_count;
    END IF;
END$$
```

4.9 Código SQL + Resultados: procedimientos almacenados (Primera entrega)

Los procedimientos almacenados son un tipo de código SQL que nos permiten crear una serie de instrucciones que se ejecutarán de manera conjunta cada vez que se llame al procedimiento. Estos procedimientos pueden ser utilizados

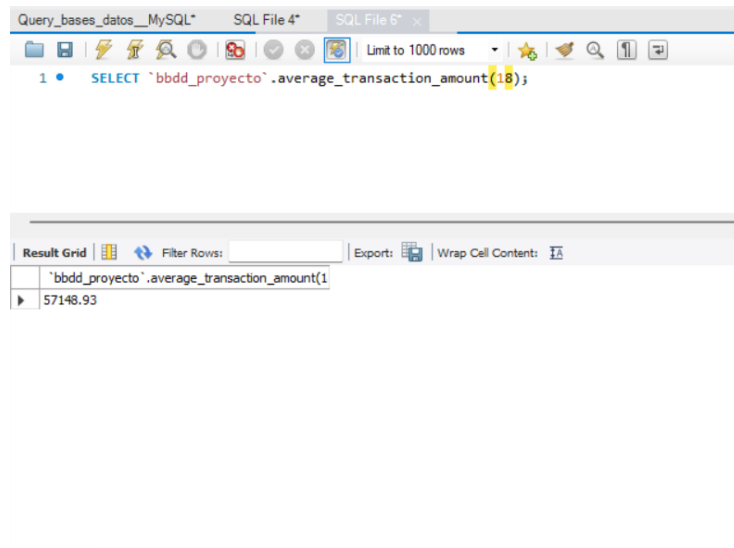


Figure 10: Function Average Transaction Amount

para realizar tareas complejas que involucren varias operaciones en una o varias tablas de una base de datos. En el contexto de las tablas mencionadas, se pueden utilizar procedimientos almacenados para llevar a cabo operaciones como la actualización de varios registros al mismo tiempo o la creación de nuevos registros en una tabla en base a la información de otra tabla. Al utilizar procedimientos almacenados, se obtiene una mayor modularidad en el código, lo que facilita su mantenimiento y reutilización en futuros proyectos.

Ejemplo: se crea un procedimiento que actualiza el valor de la transacción añadiéndole el IVA a toda la tabla.

```
DELIMITER $$
CREATE PROCEDURE `bbdd_proyecto`.`update_transaction` (IN user_id INT,
IN new_name VARCHAR(50))
BEGIN
    UPDATE `bbdd_proyecto`.`Transaction` SET F_AMOUNT = F_AMOUNT*1.1;
END$$
```

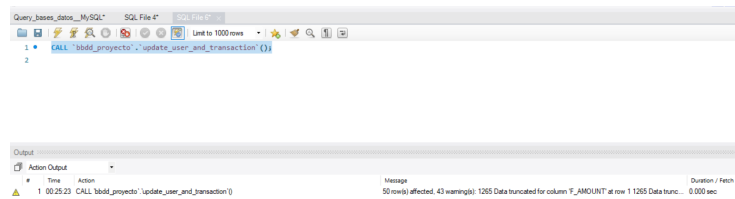


Figure 11: Procedure Update Transaction

5 Bases de Datos No-SQL (*Segunda entrega*)

En el mundo de las bases de datos, MongoDB destaca como una solución popular para el almacenamiento y recuperación de datos de forma eficiente y flexible. En este contexto, se ha desarrollado una base de datos NoSQL utilizando MongoDB, lo cual implica la creación de una estructura de datos no relacional para gestionar y organizar la información. Esta base de datos ofrece ventajas significativas en términos de escalabilidad, rendimiento y adaptabilidad a diferentes tipos de datos. A lo largo del proyecto, se han aplicado diversas estrategias y técnicas para diseñar una base de datos eficaz que satisfaga los requisitos específicos del entorno en el que se implementará.

5.1 Diagrama Bases de Datos No-SQL (*Segunda entrega*)

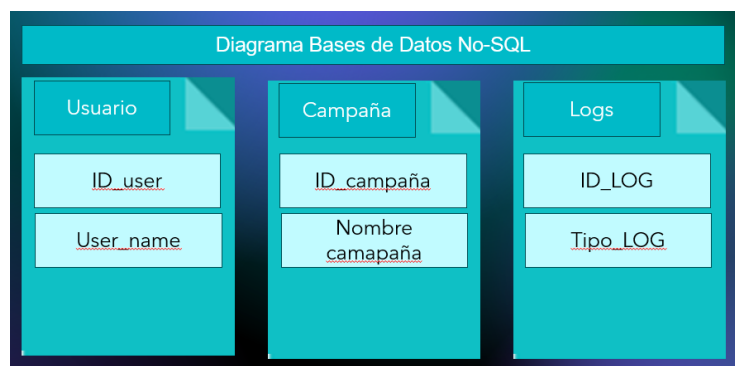


Figure 12: Diaagrama base de datos NoSQL

5.2 SMBD utilizado para la Base de Datos No-SQL (*Segunda entrega*)

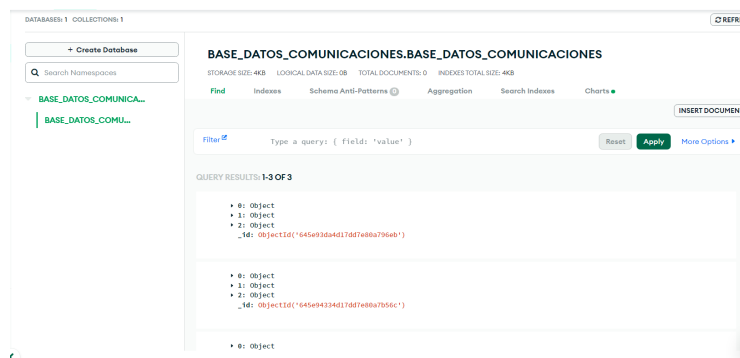


Figure 13: SMBD base de datos NoSQL

Para nuestro proyecto usaremos MongoDB el cuál es un modelo de datos NoSQL basado en documentos, lo que significa que los datos se almacenan en documentos BSON (Binary JSON) en lugar de filas y columnas como en las bases de datos relacionales. Esto proporciona una mayor flexibilidad en la estructura de los datos y permite una fácil escalabilidad horizontal.

Una de las ventajas de utilizar MongoDB como SMBD para bases de datos NoSQL es su capacidad para manejar grandes volúmenes de datos y ofrecer un alto rendimiento.

Además, MongoDB tiene como características como replicación automática para mayor disponibilidad y tolerancia a fallos, indexación eficiente para consultas rápidas y una poderosa capacidad de consulta y agregación, que para nuestro caso es sumamente importante debido al tipo de información que es manejada.

6 Aplicación de ETL (Extract, Transform, Load) y Bodega de Datos (*Tercera entrega*)

La aplicación de ETL (Extract, Transform, Load) y la creación de una Bodega de Datos son elementos fundamentales en el ámbito de la gestión de datos y el análisis empresarial. ETL es un proceso que involucra la extracción de datos de diversas fuentes, la transformación de estos datos para que sean coherentes y útiles, y finalmente, la carga de los datos en un repositorio centralizado.

La aplicación de ETL permite a las organizaciones recopilar información de múltiples fuentes, como bases de datos, sistemas de gestión, archivos planos, aplicaciones web, entre otros. Luego, los datos extraídos se someten a una serie de transformaciones para asegurar que estén limpios, normalizados y estructurados de manera coherente. Estas transformaciones pueden incluir

filtros, cálculos, conversiones de formatos, agregaciones y enriquecimiento con datos adicionales.

Una vez que los datos se han transformado, se cargan en una Bodega de Datos, que actúa como un repositorio centralizado y optimizado para el análisis de datos. Una Bodega de Datos almacena los datos históricos y actuales de la organización, permitiendo realizar consultas y análisis complejos que respalden la toma de decisiones empresariales.

La combinación de la aplicación de ETL y la Bodega de Datos proporciona una base sólida para la gestión eficiente de los datos de una organización y el análisis de negocio. Estas herramientas son esenciales para obtener información valiosa, descubrir patrones, tendencias y relaciones ocultas en los datos, y brindar a las empresas una ventaja competitiva en un entorno cada vez más data-driven.

6.1 Ejemplo de aplicación de ETL y Bodega de Datos (Tercera entrega)

Ejemplo aplicación ETL

```
In [1]: import pyodbc
import os
import pandas as pd

In [2]: os.chdir(r"D:\BACKUP MIGUEL VASQUEZ\ANALÍTICA DATOS\SEGUNDO SEMESTRE BASES DE DATOS")
tabla_transaction = pd.read_csv('Tabla_transaction.csv', encoding='ansi')
tabla_transaction.head()

Out[2]:
  Transaction_ID  ID_User  ID_Country_TRX  ID_Commerce  ID_Fraud  ID_notificacion  F_AMOUNT  F_AVAILABLE
0              1         10             27           4         0             NaN      68067.97      53412.53
1              2         10             22           1         1             NaN      73448.25      53813.38
2              3         43             30           48         1             NaN      42674.21      38269.06
3              5         15             11           21         1             NaN      2237.24      28545.59
4              6         31             19           45         1             NaN      104.70      56670.24

In [3]: tabla_transaction = tabla_transaction.drop_duplicates()

In [6]: tabla_transaction['Transaction_ID'] = tabla_transaction['Transaction_ID'].astype('int64')
tabla_transaction['ID_User'] = tabla_transaction['ID_User'].astype('int64')
tabla_transaction['ID_Country_TRX'] = tabla_transaction['ID_Country_TRX'].astype('int64')
tabla_transaction['ID_Commerce'] = tabla_transaction['ID_Commerce'].astype('int64')
tabla_transaction['ID_Fraud'] = tabla_transaction['ID_Fraud'].astype('int64')

In [7]: tabla_transaction = tabla_transaction.fillna(0)

In [8]: import mysql.connector

# Establish a connection to the MySQL database
cnx = mysql.connector.connect(
    host='localhost',
    user='root',
    password='root123',
    database='bbdd_proyecto'
)

# Create a cursor to interact with the database
cursor = cnx.cursor()

In [9]: PEPE = '?', '*'
PEPE

Out[9]: '?', '?', '?', '?', '?', '?', '?'

In [15]: MY_TABLE = 'bbdd_proyecto.Transaction ("Transaction_ID", "ID_User", "ID_Country_trx", "ID_Commerce", "ID_Fraud", "F_AMOUNT", "F_AVAILABLE")

insert_tb = f"INSERT INTO (MY_TABLE) VALUES ((PEPE(0:1)))" #,?
cursor = cnx.cursor()
cursor.fast_executemany = True #carga rápida no una a una
cursor.executemany(insert_tb, tabla_transaction.values.tolist())
cursor.commit()
cursor.close()
cnx.close()
```

Figure 14: Ejemplo aplicación ETL en Python

Resultado aplicación ETL

Query 1 x Query_bases_datos_MySQL

Limit to 1000 rows

1 • SELECT * FROM `bdd_proyecto`.Transaction

Result Grid

Filter Rows:

Edit

Export/Import:

Wrap Cell Content:

Transaction_ID	ID_User	ID_Country_TRX	ID_Commerce	ID_Fraud	ID_notificacion	F_AMOUNT	F_AVAIBLE
1	18	27	4	0	NULL	68067.97	53412.53
2	16	22	11	1	NULL	73448.25	53813.38
3	43	30	48	1	NULL	42574.21	38269.66
5	15	11	21	1	NULL	2237.24	28545.59
6	31	19	45	1	NULL	104.70	56679.24
7	11	30	46	0	NULL	28151.48	66315.56
8	37	46	7	1	NULL	96636.08	34663.54
9	41	7	23	1	NULL	54477.57	8876.14
10	31	42	5	1	NULL	13989.63	10255.14
11	38	9	9	0	NULL	90439.60	69826.70
12	49	14	39	0	NULL	14941.21	57462.44
13	11	16	43	1	NULL	73909.81	84960.66
14	17	14	24	0	NULL	21153.64	1625.69
15	4	4	29	1	NULL	32056.94	21054.87

Transaction 1 x

Figure 15: Resultado luego de la aplicación ETL en Python

6.2 Bodega de Datos

Una bodega de datos, también conocida como data warehouse, es un repositorio centralizado de datos organizado y diseñado para facilitar la toma de decisiones y el análisis de información en una organización. Se trata de una infraestructura de almacenamiento de datos que recopila, integra y gestiona datos provenientes de diversas fuentes, con el objetivo de proporcionar a los usuarios un acceso rápido y eficiente a información de calidad para su análisis y generación de informes.

La finalidad principal de una bodega de datos es proporcionar un entorno consolidado y estructurado donde los datos se almacenan de manera histórica y se transforman en un formato adecuado para su análisis. Esto implica que los datos de diferentes sistemas operacionales y fuentes de información se extraen, se limpian, se integran y se organizan de manera coherente, permitiendo a los usuarios obtener una visión integral y coherente de los datos en un solo lugar.

Las bodegas de datos se caracterizan por ser orientadas a temas específicos y estar optimizadas para consultas y análisis de datos. Están diseñadas para soportar grandes volúmenes de información y ofrecen capacidades de consulta y generación de informes eficientes y escalables.

Algunos beneficios de implementar una bodega de datos incluyen:

Consolidación de datos: Permite combinar datos de diversas fuentes y sistemas en un único repositorio, lo que facilita el análisis y la generación de informes a partir de una visión global de los datos.

Mejora en la calidad de los datos: Los datos son sometidos a procesos de limpieza, transformación y normalización para asegurar su calidad y coherencia.

Apoyo a la toma de decisiones: Proporciona a los usuarios finales una fuente confiable y consistente de información para respaldar la toma de decisiones basadas en datos.

Análisis de datos avanzado: Permite realizar análisis complejos, como minería de datos, modelado predictivo y segmentación, al disponer de datos históricos y de calidad.

Rendimiento optimizado: Las bodegas de datos están diseñadas para consultas y análisis eficientes, con estructuras de almacenamiento y optimizaciones que aceleran el acceso a los datos.

```
In [16]: import pandas as pd

# Datos de la tabla Transaction
data_transaction = {
    'transaction_id': [1, 2, 3, 4, 5],
    'amount': [100, 250, 150, 300, 200],
    'date': ['2022-01-01', '2022-01-02', '2022-01-03', '2022-01-04', '2022-01-05']
}

df_transaction = pd.DataFrame(data_transaction)

# Datos de la tabla Fraud
data_fraud = {
    'transaction_id': [2, 4],
    'fraud_type': ['Credit Card Fraud', 'Identity Theft'],
    'is_fraudulent': [True, True]
}

df_fraud = pd.DataFrame(data_fraud)

# Datos de la tabla Client
data_client = {
    'client_id': [1, 2, 3, 4, 5],
    'name': ['John', 'Mary', 'Peter', 'Anna', 'Luis'],
    'age': [30, 25, 40, 35, 28]
}

df_client = pd.DataFrame(data_client)

# Combinar los DataFrames en uno solo
df_lago_datos = pd.merge(df_transaction, df_fraud, on='transaction_id', how='left')
df_lago_datos = pd.merge(df_lago_datos, df_client, left_on='transaction_id', right_on='client_id', how='left')

# Guardar el DataFrame en un archivo CSV
df_lago_datos.to_csv('lago_datos.csv', index=False)
```

Figure 16: Ejemplo de una bodega de datos en Python

7 Proximos pasos (*Tercera entrega*)

Los próximos pasos para este proyecto es lograr implemntar el modelo de Machine Learning desarrollado dentro de la organización en el aplicativo Paytrue que es aquel utilizado dentro de la organización para la gestión del fraude. Esto con el fin de mejorar la contactabilidad de los clientes y disminuir el riesgo que un cliente pueda ser victima de fraude.

8 Lecciones aprendidas (*Tercera entrega*)

Una de las lecciones más importantes es la relevancia de la ética de los datos para este tipo de problemas, ya que se manejan datos sensibles de clientes y comportamiento transaccional. Por otro lado, que existen diferentes herramientas para el manejo y cargue de información.

9 Bibliografía

- Cortes, C., Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273-297. <https://doi.org/10.1007/BF00994018>
- Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32. <https://doi.org/10.1023/A:1010933404324>
- Fabrigar, L. R., Wegener, D. T., MacCallum, R. C., Strahan, E. J. (1999). Evaluating the use of exploratory factor analysis in psychological research. Psychological Methods, 4(3), 272-299. <https://doi.org/10.1037/1082-989X.4.3.272>
- Batista, G. E., Prati, R. C., Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD Explorations Newsletter, 6(1), 20- 29. <https://doi.org/10.1145/1007730.1007735>
- Schölkopf, B., Smola, A. J. (2002). Learning with kernels: Support vector machines, regularization, optimization, and beyond. MIT press. <https://doi.org/10.1007/3-540-33486-64>
- Cristianini, N., Shawe-Taylor, J. (2000). An introduction to support vector machines and other kernel-based learning methods. Cambridge University Press. <https://doi.org/10.1017/CBO9780511801389>
- Liaw, A., Wiener, M. (2002). Classification and regression by randomForest. R News, 2(3), 18-22. <https://CRAN.R-project.org/doc/Rnews/>
- Cutler, D. R., Edwards Jr, T. C., Beard, K. H., Cutler, A., Hess, K. T., Gibson, J., Lawler, J.J.(2007). Random forests for classification in ecology. Ecology, 88(11), 2783-2792. <https://doi.org/10.1890/07-0539.1>

- Floyd, F. J., Widaman, K. F. (1995). Factor analysis in the development and refinement of clinical assessment instruments. *Psychological Assessment*, 7(3), 286-299. <https://doi.org/10.1037/10403590.7.3.286>
- Kline, P. (2015). *Principles and practice of structural equation modeling*. Guilford publications.
- Drummond, C., Holte, R. C. (2003). C4.5, class imbalance, and cost sensitivity: Why undersampling beats over-sampling. In *Workshop on Learning from Imbalanced Datasets II* (pp. 1- 8). Citeseer.
- He, H., Bai, Y., Garcia, E. A., Li, S. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)* (pp. 1322-1328). IEEE. <https://doi.org/10.1109/IJCNN.2008.4633969>