



PLANNER PRO

Proyecto Final

DESCRIPCIÓN BREVE

Especificación funcional y técnica de proyecto SQL
Planner Pro, sistema de gestión de finanzas
personales

Gonzalo Araya

SQL – 10-07-22

Contenido

1	Especificación Funcional	2
1.1	Descripción y objetivo del proyecto	2
1.2	Situación problemática	2
1.3	Modelo de negocio.	2
1.4	Diagrama de entidad relación.....	3
1.5	Explicación del funcionamiento del sistema.....	5
1.6	Glosario de términos del sistema	6
2	Diseño de datos.....	8
2.1	Tablas.....	8
2.2	Vistas.....	13
2.2.1	Vista transactions_summary	13
2.2.2	Vista transferencias	13
2.2.3	Vista gastos.....	14
2.2.4	Vista ingresos.....	14
2.3	Funciones	15
2.3.1	Función calcular IVA:	15
2.3.2	Función ingresos vs gastos:	15
2.3.3	Función capacidad de ahorro:	15
2.4	Triggers	16
2.4.1	Ingreso de dinero:	16
2.4.2	Gasto con dinero:	16
2.4.3	auditory_create_transactions:	16
2.4.4	auditory_update_transactions:	16
2.5	Stored procedures	17
2.5.1	getExpensesByCategoryYear:	17
2.5.2	getExpensesByCategories:.....	17

1 Especificación Funcional

1.1 Descripción y objetivo del proyecto

Planner-pro es un sistema pensado en la organización de las finanzas personales, que permite monitorear y planificar los movimientos de dinero. El presente proyecto describe el modelado e implementación de la base de datos de este sistema, desarrollado con el sistema gestor de bases de datos MySQL

1.2 Situación problemática

En la actualidad, muchas personas se ven afectadas por no tener educación financiera, lo cual genera que serie de problemas financieros y emocionales que pueden afectar la calidad de vida. Si no se tiene el control de los ingresos y gastos, se pueden generar graves consecuencias al largo plazo.

El hecho de no monitorear conlleva a caer en la tentación de utilizar tarjetas de créditos y préstamos para financiar gastos diarios, lo que puede traducirse en una acumulación de deudas y en pagos mensuales insostenibles.

Una mala planificación y distribución del dinero disponible provoca que no se tengan los recursos necesarios ante contingencias de la vida diaria o de gastos recurrentes que se saben con anterioridad. Esto también provoca incertidumbre y preocupación constante, afectando la salud física y mental

1.3 Modelo de negocio.

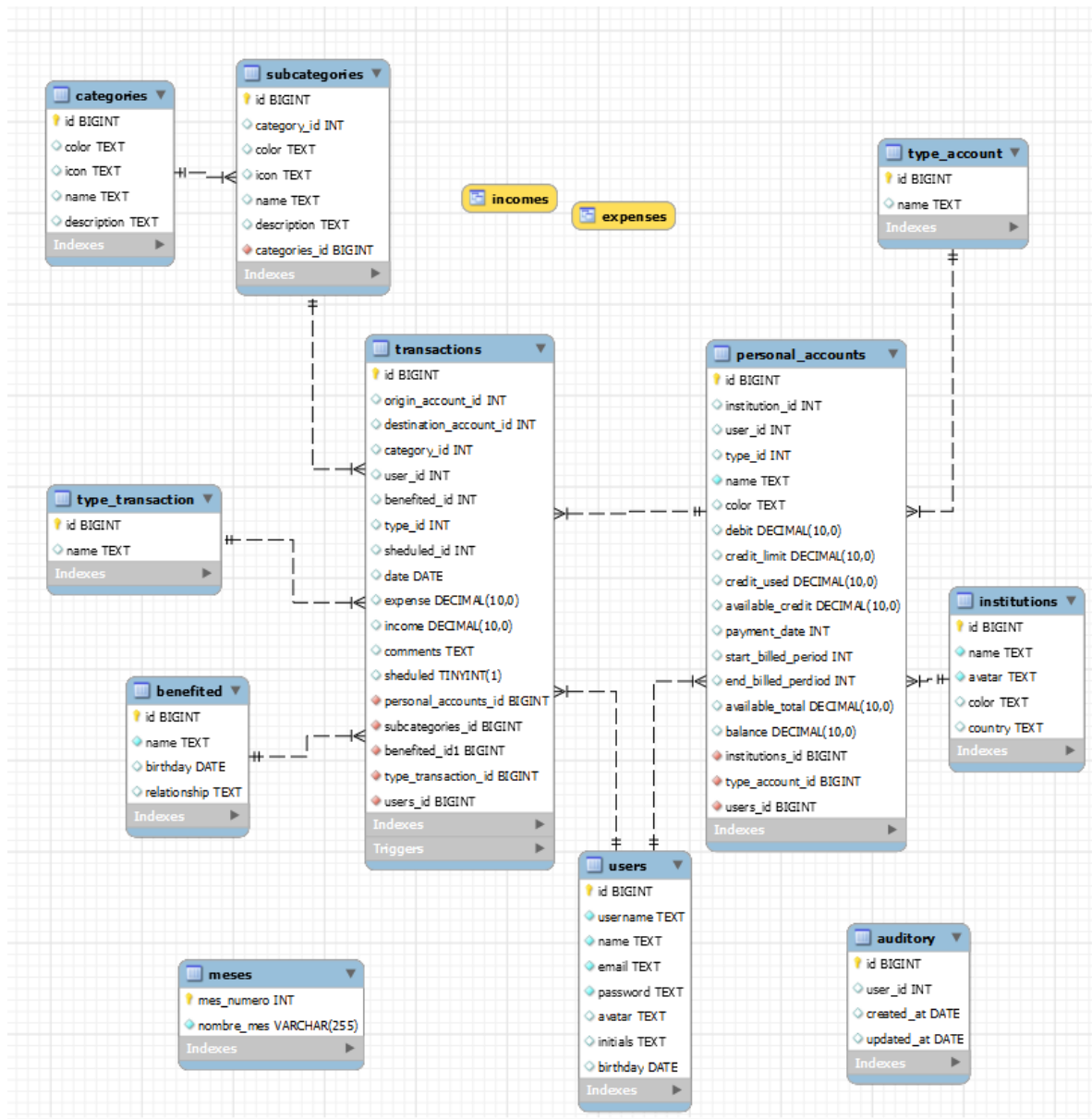
El sistema de gestión de finanzas personales ofrece una solución a esta problemática al proporcionar a los usuarios herramientas que les permiten:

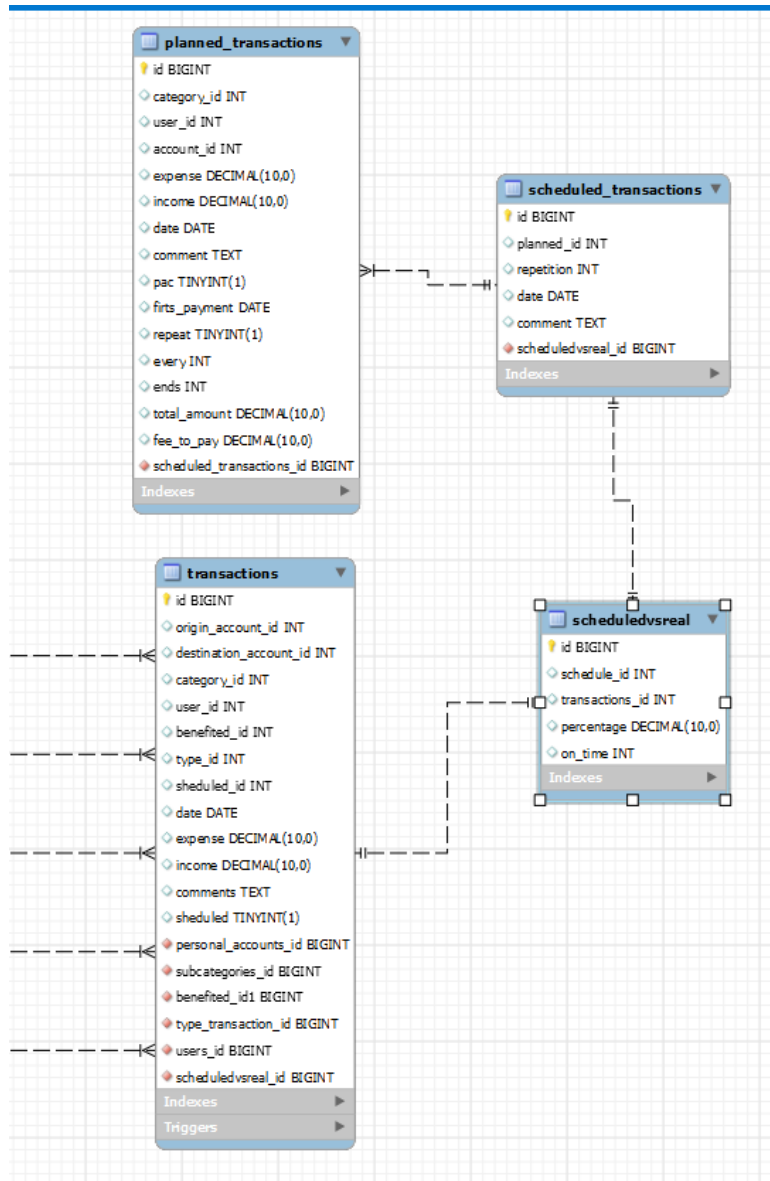
- Monitorear sus ingresos y gastos: El sistema permite registrar las transacciones como ingresos, gastos y transferencias entre cuentas. Esto permite al usuario tener una visión completa de su situación financiera en un solo lugar.
- Crear presupuestos: El sistema puede ayudar al usuario a crear y seguir un presupuesto, lo que le permite controlar sus gastos y asegurarse de no gastar más de lo que gana.
- Establecer metas financieras: El sistema puede ayudar al usuario a establecer metas financieras, como ahorrar para una casa o un automóvil, y realizar un seguimiento de su progreso hacia el logro de estas metas.

Para poder organizar el dinero necesitamos saber en qué estamos gastando dinero, para poder predecir patrones a futuro. Para ello se debe registrar los gastos e ingresos, almacenándolos en una tabla y consultándolos para poder analizarlos.

Junto con lo anterior, también es clave distribuir correctamente el dinero disponible proyectando gastos planificados en el futuro, por lo cual también se crea una tabla de transacciones planificadas.

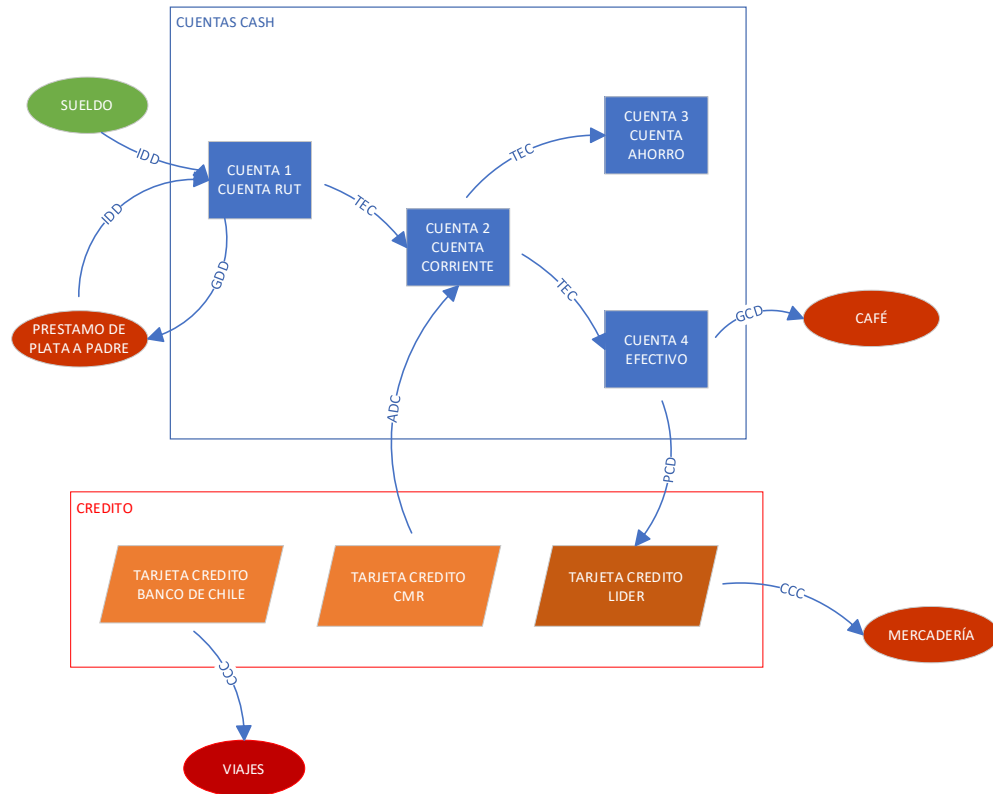
1.4 Diagrama de entidad relación





1.5 Explicación del funcionamiento del sistema

En el siguiente diagrama se intenta explicar cómo funcionaría el sistema, y que acciones se deben ejecutar dentro del sistema, dependiendo de las acciones que realice el usuario, por ejemplo, si existe una transferencia entre cuentas (TEC), Se descuenta el monto en la cuenta de origen y se suma a la cuenta de destino, además se debe registrar este movimiento en la tabla de transacciones. Estas acciones se podrían realizar a través de triggers en el sistema gestor de bases de datos o desde el backend.



ID	INGRESO DE DINERO	Sumar monto a la cuenta de destino en la tabla CUENTAS, registrar en movimiento en la tabla TRANSACCIONES
ADC	ADELANTO DESDE CRÉDITO	Sumar monto a la cuenta de destino en la tabla CUENTAS, registrar en movimiento en la tabla TRANSACCIONES, descontar cupo de la tarjeta de crédito en la tabla CUENTAS
GCC	GASTO CON CRÉDITO	Registrar en movimiento en la tabla TRANSACCIONES PANIFICADAS, descontar cupo de la tarjeta de crédito en la tabla CUENTAS, agregar registros de pagos en la tabla TRANSACCIONES PROGRAMADAS.
GCD	GASTO CON DINERO	Descontar monto a la cuenta de origen en la tabla CUENTAS, registrar en movimiento en la tabla TRANSACCIONES
PCD	PAGO DE CRÉDITO CON DINERO	Descontar monto a la cuenta de origen en la tabla CUENTAS, registrar en movimiento en la tabla TRANSACCIONES, aumentar el cupo de la tarjeta en la tabla CUENTAS
TEC	TRANSFERENCIA ENTRE CUENTAS	Registrar movimiento en la tabla TRANSACCIONES, aumentar el monto en la cuenta de destino y descontarlo en la cuenta de origen de la tabla CUENTAS.

1.6 Glosario de términos del sistema

Este sistema fue desarrollado en Chile por lo que los conceptos pueden variar dependiendo de la ubicación geográfica, de todas formas, para poder ayudar a entender el planteamiento del sistema, a continuación, se definen los conceptos claves:

Transacción: Todo movimiento de dinero real (debito) en el sistema, desde o hacia el exterior (gastos e ingresos) o dentro (transferencias entre cuentas) del sistema.

Registro de transacciones:

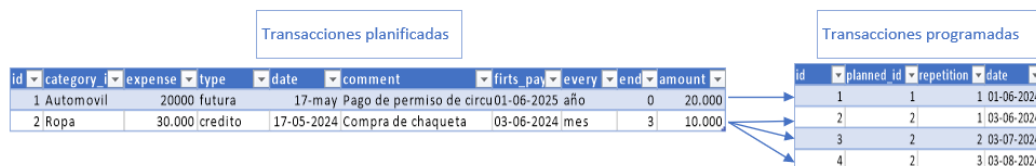
- Registra las transacciones (gastos, ingresos, transferencias)
- Busca obtener la verdad sobre los gastos realizados.
- Enfoca el presente al momento de ingresarlos tomando conciencia de los gastos.
- Enfoca el pasado al obtener la data de los gastos, poder analizarlos y gestionarlos.

Transacciones planificadas:

- Proyecta las transacciones futuras
- Se puede determinar presupuestos para no gastar más de lo que se gana
- Permite ver la variación de los distintos pagos recurrentes en los meses para sumar a los gastos fijos.

Transacciones programadas:

- Son las repeticiones proyectadas de las transacciones planificadas



Presupuesto: Predicción de las transacciones del futuro. Se componen de transacciones periódicas, transacciones recurrentes y transacciones planificadas.

Ingreso: Todo dinero que entra al sistema

- Pago de sueldos
- Pago de aguinaldos
- Pago de bonos
- Regalos en giftcard

Gasto: Todo dinero que sale

- Pago de tarjetas de créditos

- Pago de productos y servicios no presupuestados
- Pago de transacciones presupuestadas

Transferencia entre cuentas: Transacción en la cual el dinero no sale del sistema.

- Transferencia de débito a debito
- Transferencia de débito a cuenta de ahorro
- Retiro de débito a efectivo
- Depósito de efectivo a debito
- Transferencia de cuenta de ahorro a debito
- Transferencia de cuenta de ahorro a efectivo

2 Diseño de datos

2.1 Tablas

users: Registra los usuarios del sistema

Campos	tipo de dato	Restricciones	Descripción
id	SERIAL	PK	identificador auto incremental
username	TEXT	NOT NULL	Nombre del usuario
email	TEXT	NOT NULL, UNIQUE	email del usuario
password	TEXT	NOT NULL	Contraseña
avatar			Foto de perfil
initials	TEXT		iniciales del nombre, en caso de que no tenga foto en avatar
birthday	DATE		Cumpleaños del usuario

categories: Categorías de los registros de transacciones, Ej: Automóvil.

Campos	tipo de dato	Restricciones	Descripción
id	SERIAL	PK	identificador auto incremental
color	TEXT		Color de la categoría, con fines visuales para la app.
icon	TEXT		Icono de la categoría, con fines visuales para la app.
name	TEXT	NOT NULL	Nombre de la categoría, Ej: Vivienda
description	TEXT		Descripción de la categoría

subcategories: Subcategoría derivada de las categorías. En el ejemplo del automóvil, una subcategoría sería las mantenciones o la renovación de los documentos.

Campos	tipo de dato	Restricciones	Descripción
id	SERIAL	PK	identificador auto incremental
category_id	INTEGER	FK	id de categoría a la que pertenece la subcategoría.
color	TEXT		Color de la subcategoría, con fines visuales para la app.
icon	TEXT		Icono de la subcategoría, con fines visuales para la app.
name	TEXT	NOT NULL	Nombre de la subcategoría, Ej: Agua (Vivienda)
description	TEXT		Descripción de la subcategoría

Institutions: Registra las instituciones bancarias para normalizar el campo institution_id de la tabla anterior.

Campos	tipo de dato	Restricciones	Descripción
id	SERIAL	PK	identificador auto incremental
name	TEXT	NOT NULL	Nombre de la institución. Ejemplo: Banco Estado de Chile
avatar	TEXT		Avatar con fines visuales para la app.
color			Color con fines visuales para la app.

Personal_accounts: Son las cuentas personales que posee el usuario. Aquí se consideran la fuente de donde se saca y se guarda dinero como por ejemplo dinero en efectivo, tarjetas de débito, tarjetas de crédito, cuentas de ahorro, etc.

Campos	tipo de dato	Restricciones	Descrpción
id	SERIAL	PK	identificador autoincremental
institution_id	INTEGER	FK	ID de la institución bancaria
user_id	INTEGER	FK	ID de usuario al que pertenece la cuenta
type_id	TEXT	FK	ID tipo de cuenta: Ej: Cuenta vista, efectivo, cuenta corriente.
name	TEXT	NOT NULL	Nombre de la cuenta, Ej: Cuenta Rut, efectivo, Mastercard Falabella
color	TEXT		Color de la categoría, con fines visuales para la app
debit	NUMERIC		Debito de la cuenta
credit_limit	NUMERIC		Limite de credito en la cuenta, Ej: Las cuentas corrientes tiene un cierto cupo de crédito
credit_used	NUMERIC		Crédito usado
available_credit	NUMERIC		Crédito disponible
payment_date	INTEGER		Día de pago de la cuenta
start_billed_period	INTEGER		Inicio del periodo de facturación de la cuenta
end_billed_period	INTEGER		Fin del periodo de facturación de la cuenta
billing_date	INTEGER		Día de pago de la cuenta
available_total	NUMERIC		Dinero disponible = debito + credito disponible
balance	NUMERIC		Balance general = debito - credito usado

Benefited: Son los beneficiarios por los cuales se realiza un gasto. Ej: Si el gasto es para un regalo de cumpleaños, aquí se especifica quién obtendrá ese regalo y así se podrá analizar cuanto dinero se ha gastado en esa persona.

Beneficiados	tipo de dato	Restricciones	Descripción
id	SERIAL	PK	identificador autoincremental
name	TEXT	NOT NULL	Nombre del beneficiado
birthday	DATE		Cumpleaños
relationship	TEXT		Relación con el beneficiado, Ejemplo: Padre, amigo, etc.

Transactions: Registra las transacciones que se realizan como gastos, ingresos y transferencias entre cuentas.

Campos	tipo de dato	Restric.	Descripción
id	SERIAL	PK	identificador autoincremental
origin_account_id	integer	FK	ID de la cuenta de origen
destination_account_id	integer	FK	ID de la cuenta de destino
category_id	integer	FK	ID de la categoría
user_id	integer	FK	ID Usuario que realiza la transacción
benefited_id	integer	FK	ID de Beneficiado con la transacción
type_id	integer	FK	ID de Tipo de transacción: Ingreso, gasto, transferencia entre cuentas
sheduled_id	integer	FK	Si es una transacción planificada, se especifica la id.
date	DATE		Fecha de la transacción
expense	NUMERIC		Gasto
income	NUMERIC		Ingreso
comments	TEXT		Comentarios
sheduled	bool		Es Planificado? True o false.

planned_transactions: Registra las transacciones planificadas, estas transacciones son las que se realizarán en un futuro y según sea su nivel de repetición pueden ser:

- Transacciones periódicas: No tienen fecha de termino, su periodo puede variar, siendo generalmente mensual como el pago de la luz, agua, etc. o anuales como el pago de revisión técnica.
- Transacciones recurrentes: Se ven reflejadas con periodicidad incluso mensual, pero tienen fecha de termino. Ej Pago de mensualidad de universidad, compras con crédito (a cuotas), etc.
- Transacción futura: Es una transacción que se realizará una sola vez en el futuro. Ejemplo, un viaje de vacaciones.

Campos	tipo de dato	Restricc.	Descripción
id	SERIAL	PK	identificador autoincremental
category_id	integer	FK	identificador de la categoría
user_id	integer	FK	id del usuario
benefited_id			id del beneficiario
account_id	integer	FK	id de la cuenta
comment	TEXT		Comentarios
expense	NUMERIC		monto de gasto
income	NUMERIC		monto de ingreso
type	TEXT		Tipo: crédito, transacción recurrente, transacción futura o transacción periódica
first_payment	date		Primera fecha de pago
deadline	date		Fecha límite de pago
repeat	BOOL		Repetir?: True o false
every	INTEGER		Cada cuanto repetir
interval	text		Intervalo: día, mes, año
ends	INTEGER		Terminar después de tantas repeticiones
total_amount	NUMERIC		Monto total = ends * expense
pac	BOOL		¿Pago automático de cuentas? True o false

scheluded_transactions: Registra las transacciones que se crean derivadas de las transacciones planificadas.

Campos	tipo de dato	Restricciones	Descrpción
id	SERIAL	PK	identificador autoincremental
planned_id	integer	FK	id de la transacción planificada
repetition	integer		Cantidad de repticiones de la transacción planificada
date	DATE		fecha
comment	TEXT		comentarios

scheduledVSreal: Es una tabla resumen que se crea para comparar las transacciones programadas contra la tabla de transacciones. Nos sirve para ver si lo programado se cumple, en el tiempo y en el presupuesto dispuesto para tal transacción

Campos	tipo de dato	Restricciones	Descrpción
id	SERIAL	PK	identificador autoincremental
schedule_id	INT	Fk	Id de la transacción programada
transactions_id	INT	Fk	Id de la transacción
percentage	NUMERIC	FK	porcentaje de cumplimiento entre lo programado y lo real
on_time	INT	FK	Días de diferencia entre lo programado y lo real

2.2 Vistas

Una vista es una alternativa que se puede utilizar para almacenar consultas recurrentes en este caso se crearon 4 vistas para tener acceso a la tabla “transacciones” y a sus diferentes campos que tienen relaciones con otras tablas y que de otra forma solo se verían los id de las tablas referencias, haciendo difícil su interpretación, tal como se muestra en la siguiente imagen:

	id	origin_account_id	destination_account_id	category_id	user_id	benefited_id	sheduled_id	date	expense	income	comments	type
▶	14	1	NULL	5	1	6	NULL	2024-06-11	98000	NULL	prueba desde VScode2	gasto
	15	3	NULL	3	1	24	NULL	2024-05-02	147970	NULL		gasto
	16	2	NULL	15	1	10	NULL	2024-03-25	93713	NULL		gasto
	17	2	1	1	1	1	NULL	2024-06-12	112000	NULL	desde base	transferencia
	18	1	NULL	3	1	10	NULL	2023-12-05	157297	NULL		gasto
	19	3	NULL	35	1	14	NULL	2023-04-06	215412	NULL		gasto
	20	3	NULL	11	1	6	NULL	2024-02-08	142127	NULL		gasto
	21	3	NULL	38	1	8	NULL	2023-08-23	111780	NULL		gasto
	22	2	NULL	19	1	8	NULL	2024-04-23	74465	NULL		gasto
	23	3	NULL	23	1	20	NULL	2023-05-08	249714	NULL		gasto
	24	1	NULL	22	1	16	NULL	2023-11-18	161896	NULL		gasto
	25	2	NULL	42	1	11	NULL	2024-02-16	229040	NULL		gasto
	26	1	NULL	12	1	5	NULL	2024-01-11	173564	NULL		gasto
	27	2	NULL	17	1	18	NULL	2024-04-15	148847	NULL		gasto
	28	2	NULL	11	1	6	NULL	2023-06-21	150711	NULL		gasto
	29	2	NULL	1	1	1	NULL	2024-06-13	45000	NULL	prueba desde VScode2	gasto
	30	1	NULL	2	1	2	NULL	2024-06-13	678685	NULL	prueba desde VScode2	gasto
	31	1	2	1	1	1	NULL	2024-06-12	111000	NULL	desde base	transferencia
	32	1	2	1	1	1	NULL	2024-06-12	111000	NULL	desde base	transferencia

2.2.1 Vista transactions_summary

Esta vista tiene como objetivo mostrar los todos los campos de la tabla transacciones, muestra los ingresos, gastos y transferencias, haciendo más legible su respuesta. Se conecta con las tablas personal_accounts, categories, subcategories y benefited

```
24 • select * from transactions_summary;
25
26
```

Result Grid										
Filter Rows:										
Export: Wrap Cell Contents: <input type="checkbox"/>										
Fecha	Cuenta_origen	Cuenta_destino	categoria	Subcategoria	Gastos	Ingreso	Tipo	Beneficiado	Comentarios	
▶ 2023-01-06	Cuenta Rut	NULL	Casa	Gas	177446	NULL	gasto	Mario Cortez		
2023-01-07	Mastercard BdCh	NULL	Estudios	Full Stack developer	60940	NULL	gasto	Lorenzo Cifuentes		
2023-01-08	Cuenta corriente BdCH	NULL	Transporte	Transporte a trabajo	154041	NULL	gasto	Valentina Alburquenque		
2023-01-08	Mastercard BdCh	NULL	Estudios	Libros	76277	NULL	gasto	Natalia Ruiz		
2023-01-09	Mastercard BdCh	NULL	Casa	Luz	79388	NULL	gasto	Antony Araya		
2023-01-09	Cuenta Rut	NULL	Vida	Snack	75870	NULL	gasto	Valentina Alburquenque		
2023-01-09	Cuenta Rut	NULL	Estudios	Full Stack developer	218356	NULL	gasto	Eduardo Campillay		
2023-01-10	Cuenta corriente BdCH	NULL	Servicios	Netflix	107747	NULL	gasto	Jeremias Araya		
2023-01-12	Mastercard BdCh	NULL	Automovil	Estacionamiento	35989	NULL	gasto	Braian Araya		
2023-01-14	Cuenta Rut	NULL	Vida	Mercadería	116181	NULL	gasto	Eduardo Campillay		
2023-01-15	Cuenta Rut	NULL	Estudios	Libros	39428	NULL	gasto	Claudia Castillo		
2023-01-16	Cuenta corriente BdCH	NULL	Ahorro	Contingencias	191698	NULL	gasto	Lorenzo Cifuentes		
2023-01-18	Cuenta Rut	NULL	Salud	Dentista	33434	NULL	gasto	Natalia Ruiz		
2023-01-19	Cuenta Rut	NULL	Automovil	Permiso de circulación	84305	NULL	gasto	Eduardo Campillay		
2023-01-20	Mastercard BdCh	NULL	Vida	Mercadería	188241	NULL	gasto	Jeremias Araya		
2023-01-20	Cuenta corriente BdCH	Mastercard BdCh	Compras	Otros	138326	138326	transferencia entr...	Javi Duarte		
2023-01-22	Cuenta Rut	Cuenta corrien...	Servicios	Celular	35223	NULL	transferencia entr...	Claudio Paves		
2023-01-23	Cuenta corriente BdCH	NULL	Casa	Alojamiento	233646	NULL	gasto	Gonzalo Aguilera		
2023-01-23	Cuenta Rut	NULL	Automovil	Combustible	97384	NULL	gasto	Mario Cortez		

2.2.2 Vista transferencias

Esta vista tiene como objetivo mostrar solo las transferencias de la tabla transacciones, haciendo más legible su respuesta. Se conecta con las tablas personal_accounts, subcategories y benefited.

fecha	cuenta_origen	cuenta_destino	Gastos	Ingreso	Tipo	Beneficiado	Comentarios
2023-01-20	Cuenta corriente BdCH	Mastercard BdCh	138326	138326	transferencia	Javi Duarte	
2023-01-22	Cuenta Rut	Cuenta corriente BdCH	35223	35223	transferencia	Claudio Paves	
2023-01-23	Cuenta Rut	Mastercard BdCh	170675	170675	transferencia	Claudio Paves	
2023-02-06	Mastercard BdCh	Mastercard BdCh	154035	154035	transferencia	Jorge Julia	
2023-03-11	Cuenta corriente BdCH	Cuenta Rut	70609	70609	transferencia	Javi Duarte	
2023-03-28	Mastercard BdCh	Cuenta corriente BdCH	196795	196795	transferencia	Jorge Julia	
2023-04-24	Mastercard BdCh	Cuenta Rut	35223	35223	transferencia	Jeremias Araya	
2023-04-28	Cuenta corriente BdCH	Mastercard BdCh	96915	96915	transferencia	Claudio Paves	
2023-05-30	Cuenta Rut	Cuenta corriente BdCH	96915	96915	transferencia	Javiera Araya	
2023-05-30	Mastercard BdCh	Mastercard BdCh	96813	96813	transferencia	Valentina Alburquenque	
2023-06-04	Cuenta corriente BdCH	Mastercard BdCh	6635	6635	transferencia	Elias Araya	
2023-06-06	Mastercard BdCh	Cuenta Rut	84537	84537	transferencia	Gonzalo Aguilera	
2023-07-01	Cuenta corriente BdCH	Cuenta Rut	138326	138326	transferencia	Claudio Paves	

2.2.3 Vista gastos

Esta vista tiene como objetivo mostrar los gastos de la tabla transacciones, haciendo más legible su respuesta. Se conecta con las tablas personal_accounts, subcategories y benefited.

	date	cuenta_origen	expense	name	type	Beneficiado
▶	2024-06-11	Cuenta Rut	98000	Insumos hogar	gasto	Braian Araya
	2024-05-02	Mastercard BdCh	147970	Luz	gasto	Felipe Nunez
	2024-03-25	Cuenta corriente BdCH	93713	Medico	gasto	Antony Araya
	2023-12-05	Cuenta Rut	157297	Luz	gasto	Antony Araya
	2023-04-06	Mastercard BdCh	215412	Herramientas	gasto	Valentina Alburquenque
	2024-02-08	Mastercard BdCh	142127	Transporte por viajes	gasto	Braian Araya
	2023-08-23	Mastercard BdCh	111780	Otros	gasto	Mario Cortez
	2024-04-23	Cuenta corriente BdCH	74465	Dentista	gasto	Mario Cortez
	2023-05-08	Mastercard BdCh	249714	Spotify	gasto	Lorenzo Cifuentes
	2023-11-18	Cuenta Rut	161896	Youtube	gasto	Claudia Caceres
	2024-02-16	Cuenta corriente BdCH	229040	Snack	gasto	Jeremias Araya
	2024-01-11	Cuenta Rut	173564	Transporte a trabajo	gasto	Javier Araya
	2024-04-15	Cuenta corriente BdCH	148847	Deporte	gasto	Javi Duarte
	2023-06-21	Cuenta corriente BdCH	150711	Transporte por viajes	gasto	Braian Araya

2.2.4 Vista ingresos

Esta vista tiene como objetivo mostrar solo los ingresos de la tabla transacciones, haciendo más legible su respuesta. Se conecta con la tabla personal_accounts.

date	cuenta_destino	Ingresos	Tipo
2023-01-07	Cuenta Rut	167081	ingreso
2023-01-17	Cuenta corriente BdCH	71652	ingreso
2023-01-17	Cuenta Rut	40462	ingreso
2023-01-20	Cuenta Rut	212454	ingreso
2023-01-20	Cuenta Rut	231064	ingreso
2023-01-22	Mastercard BdCh	44289	ingreso
2023-01-26	Cuenta corriente BdCH	190095	ingreso
2023-01-29	Cuenta Rut	175187	ingreso
2023-01-31	Mastercard BdCh	196330	ingreso
2023-02-04	Mastercard BdCh	211202	ingreso
2023-02-05	Cuenta Rut	72625	ingreso
2023-02-11	Mastercard BdCh	42011	ingreso
2023-02-12	Mastercard BdCh	207510	ingreso

2.3 Funciones

2.3.1 Función calcular IVA:

En Chile y algunos países de Latinoamérica existe el impuesto a las ventas y servicios que equivale a un 19% del total del producto, con esta función podemos calcular cuanto estamos pagando de iva en una transacción.

	<code>calcular_iva(100)</code>
▶	19

2.3.2 Función ingresos vs gastos:

Con esta función podemos calcular el porcentaje de ingresos sobre gastos, recibe dos parámetros: ingresos y gastos y retorna el porcentaje de los ingresos que se utilizaron para cubrir los gastos. La fórmula es $(\text{gastos}/\text{ingresos}) * 100$.

	<code>ingresos_vs_gastos(20000,15000)</code>
	75

2.3.3 Función capacidad de ahorro:

Muestra el porcentaje que se pudo ahorrar: $(\text{ahorro}/\text{ingresos}) * 100$

	<code>capacidad_de_ahorro(100000,1000000)</code>
	10

2.4 Triggers

2.4.1 Ingreso de dinero:

Este trigger detecta una transacción de tipo “ingreso” por lo cual aumenta (actualiza) la columna “debito” en el registro de la cuenta en la tabla personal_account.

2.4.2 Gasto con dinero:

Este trigger detecta una transacción de tipo “gasto” por lo cual descuenta (actualiza) la columna “debito” en el registro de la cuenta en la tabla personal_account.

2.4.3 auditory_create_transactions:

Este trigger se dispara luego de que se cree un registro nuevo en la tabla transacciones, registrando la fecha y el usuario en la tabla auditoria.

2.4.4 auditory_update_transactions:

Este trigger se dispara luego de que se actualice un registro en la tabla transacciones, registrando la fecha y el usuario en la tabla auditoria.

2.5 Stored procedures

2.5.1 getExpensesByCategoryYear:

Ese procedimiento almacenado recoge la suma de todos los gastos por categoría y por año, mostrando los porcentajes de los gastos. Tiene como entrada el año a consultar.

```
12 • call getExpensesByCategoryYear('2024')
13
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
Categoria	total	porcentaje	
Otros	2,415,709	2.8	
Dentista	2,014,817	2.3	
Tecnologicos	1,534,948	1.8	
Transporte por viajes	1,469,110	1.7	
Agua	1,222,590	1.4	
Bar, restaurant	1,149,561	1.3	
Arriendo	872,059	1.0	
Ropa y zapatos	752,623	0.9	
Alojamiento	746,843	0.9	
Netflix	700,617	0.8	

2.5.2 getExpensesByCategories:

Este SP obtiene la suma de los gastos por categoría y por mes. Recibe como entrada el mes y el año.

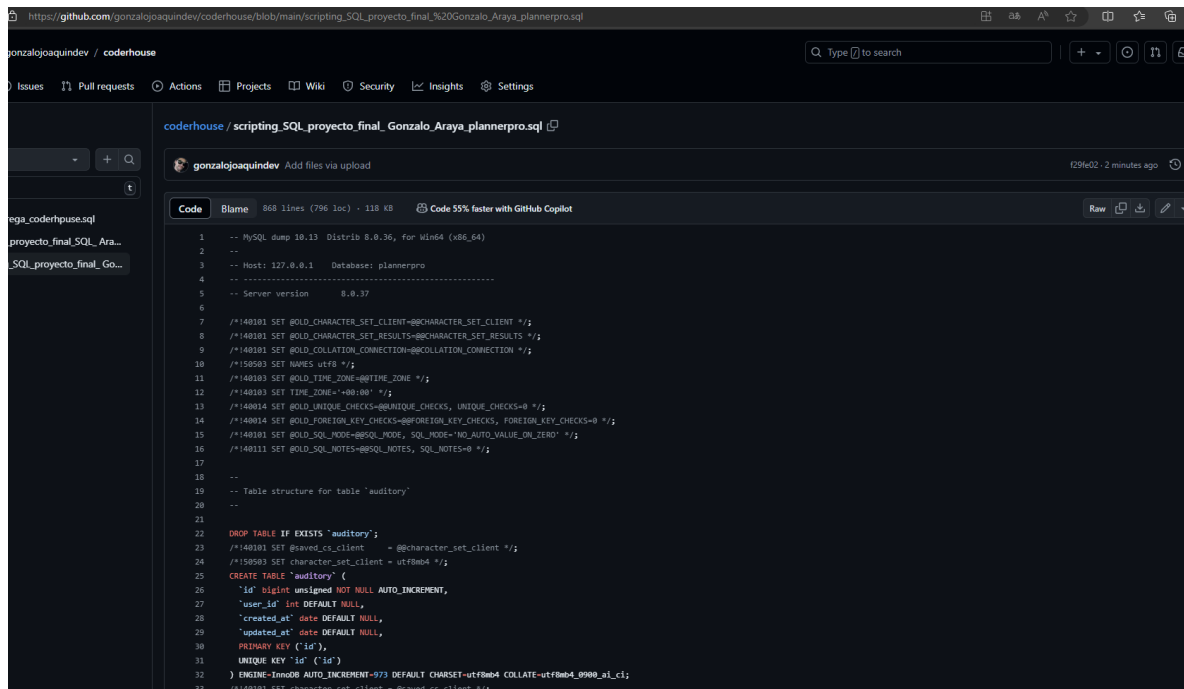
```
26 call getExpensesByCategoriesMonth("Enero", "2024")
27
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
YEAR(Fecha)	Fecha	Categoria	Gastos
2024	2024-01-01	Transporte a trabajo	139689
2024	2024-01-03	Transporte por viajes	222184
2024	2024-01-04	Agua	226436
2024	2024-01-05	Dentista	174050
2024	2024-01-05	Arriendo	7067
2024	2024-01-07	Dentista	125882
2024	2024-01-07	Netflix	19217
2024	2024-01-08	Transporte a trabajo	18366
2024	2024-01-08	Regalo	240559
2024	2024-01-10	SOAP	70239
2024	2024-01-11	Transporte a trabajo	173564

3 Scripting de creación de base de datos

El scripting está alojado en github para lo acceder a el puedes entrar al siguiente link:

[gonzalojoaquindev/coderhouse: SQL preentrega 1 \(github.com\)](https://github.com/gonzalojoaquindev/coderhouse/blob/main/scripting_SQL_proyecto_final_Gonzalo_Araya_plannerpro.sql)
[coderhouse/scripting SQL proyecto final Gonzalo Araya plannerpro.sql at main · gonzalojoaquindev/coderhouse \(github.com\)](https://github.com/gonzalojoaquindev/coderhouse/blob/main/scripting_SQL_proyecto_final_Gonzalo_Araya_plannerpro.sql)

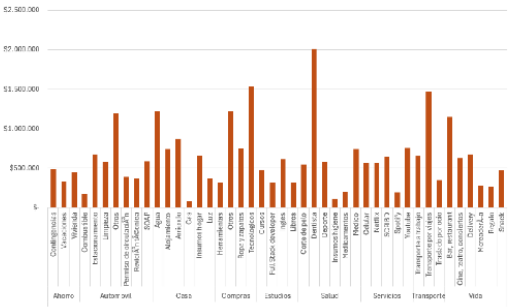


The screenshot shows a GitHub repository page for 'gonzalojoaquindev / coderhouse'. The file 'scripting_SQL_proyecto_final_Gonzalo_Araya_plannerpro.sql' is selected. The script is a MySQL dump for a database named 'plannerpro'. It includes settings for character set, collation, and server version. The main part of the script is the creation of a table named 'auditory'. The table has columns 'id', 'user_id', 'created_at', and 'updated_at'. The 'id' column is the primary key and is unsigned. The 'user_id' column is an integer. The 'created_at' and 'updated_at' columns are dates. The table is created with the 'InnoDB' engine and 'utf8mb4' character set.

```
1 -- MySQL dump 10.13. Distrib 8.0.36, for Win64 (x86_64)
2 --
3 -- Host: 127.0.0.1 Database: plannerpro
4 --
5 -- Server version      8.0.37
6
7 /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
8 /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
9 /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
10 /*!50503 SET NAMES utf8 */;
11 /*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
12 /*!40103 SET TIME_ZONE='+00:00' */;
13 /*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
14 /*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
15 /*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
16 /*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
17
18 --
19 -- Table structure for table 'auditory'
20 --
21
22 DROP TABLE IF EXISTS `auditory`;
23 /*!40101 SET @saved_cs_client      = @@character_set_client */;
24 /*!50503 SET character_set_client = utf8mb4 */;
25 CREATE TABLE `auditory` (
26   `id` bigint unsigned NOT NULL AUTO_INCREMENT,
27   `user_id` int DEFAULT NULL,
28   `created_at` date DEFAULT NULL,
29   `updated_at` date DEFAULT NULL,
30   PRIMARY KEY (`id`),
31   UNIQUE KEY `id` (`id`)
32 ) ENGINE=InnoDB AUTO_INCREMENT=973 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
33 /*!40101 SET @character_set_client = @saved_cs_client */;
```

Además, podrás encontrar el reporte de la información extraída desde el sistema:

1.1.3 Gasto por categorías y subcategorías anual (2024)



1.1.4 Gastos por subcategoría último mes (junio)

