LABORATORIO 6 - CONTROL DE CONCURRENCIA DISTRIBUIDA

PARTE A: SQL PURO CON MÚLTIPLES TERMINALES

PASO 1: PREPARACIÓN DEL ENTORNO

1.1 Crear las bases de datos

Abrir Terminal 1 (como superusuario):

```
6  CREATE DATABASE banco_lima;
7  CREATE DATABASE banco_cusco;
8  CREATE DATABASE banco_arequipa;
```

Data Output Messages Notifications

```
CREATE DATABASE

Query returned successfully in 213 msec.
```

```
7  CREATE DATABASE banco_cusco;
8  CREATE DATABASE banco_arequipa;
```

Data Output Messages Notifications

```
CREATE DATABASE

Query returned successfully in 226 msec.
```

```
8  CREATE DATABASE banco_arequipa;
```

Data Output Messages Notifications

```
CREATE DATABASE

Query returned successfully in 216 msec.
```

```
10 CREATE USER estudiante WITH PASSWORD 'lab2024';
11 GRANT ALL PRIVILEGES ON DATABASE banco_lima TO estudiante;
12 GRANT ALL PRIVILEGES ON DATABASE banco_cusco TO estudiante;
13 GRANT ALL PRIVILEGES ON DATABASE banco_arequipa TO estudiante;
```

Data Output Messages Notifications

```
GRANT

Query returned successfully in 40 msec.
```

1.2 Estructura de tablas para BANCO_LIMA

Conectar a banco_lima:

```sql
-- Conectar a banco_lima:
CREATE TABLE cuentas (
    id SERIAL PRIMARY KEY,
    numero_cuenta VARCHAR(20) UNIQUE NOT NULL,
    titular VARCHAR(100) NOT NULL,
    saldo NUMERIC(15,2) NOT NULL CHECK (saldo >= 0),
    sucursal VARCHAR(50) DEFAULT 'Lima',
    fecha_creacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    ultima_modificacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    version INTEGER DEFAULT 1
);
```

Data Output | Messages | Notifications

```
CREATE TABLE

Query returned successfully in 41 msec.
```

```sql
CREATE TABLE transacciones_log (
    id SERIAL PRIMARY KEY,
    transaccion_id VARCHAR(50) NOT NULL,
    cuenta_id INTEGER REFERENCES cuentas(id),
    tipo_operacion VARCHAR(20),
    monto NUMERIC(15,2),
    estado VARCHAR(20),
    timestamp_inicio TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    timestamp_prepare TIMESTAMP,
    timestamp_final TIMESTAMP,
    descripcion TEXT
);
```

Data Output | Messages | Notifications

```
CREATE TABLE

Query returned successfully in 42 msec.
```

```sql
CREATE TABLE control_2pc (
    transaccion_id VARCHAR(58) PRIMARY KEY,
    estado_global VARCHAR(28),
    participantes TEXT[],
    votos_commit INTEGER DEFAULT 0,
    votos_abort INTEGER DEFAULT 0,
    timestamp_inicio TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    timestamp_decision TIMESTAMP,
    coordinador VARCHAR(50)
);
```

Data Output | Messages | Notifications

```
CREATE TABLE

Query returned successfully in 42 msec.
```

```sql
INSERT INTO cuentas (numero_cuenta, titular, saldo) VALUES
('LIMA-001', 'Juan Pérez Rodríguez', 5000.00),
('LIMA-002', 'María García Flores', 3000.00),
('LIMA-003', 'Carlos López Mendoza', 7500.00),
('LIMA-004', 'Ana Torres Vargas', 2800.00),
('LIMA-005', 'Pedro Ramírez Castro', 6200.00);
```

Data Output | Messages | Notifications

```
INSERT 0 5

Query returned successfully in 41 msec.
```

1.3 Estructura para BANCO_CUSCO

Abrir Terminal 2 y conectar:

```sql
CREATE TABLE cuentas (
    id SERIAL PRIMARY KEY,
    numero_cuenta VARCHAR(20) UNIQUE NOT NULL,
    titular VARCHAR(100) NOT NULL,
    saldo NUMERIC(15,2) NOT NULL CHECK (saldo >= 0),
    sucursal VARCHAR(50) DEFAULT 'Cusco',
    fecha_creacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    ultima_modificacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    version INTEGER DEFAULT 1
);
```

Data Output | Messages | Notifications

```
CREATE TABLE

Query returned successfully in 44 msec.
```

```sql
CREATE TABLE transacciones_log (
    id SERIAL PRIMARY KEY,
    transaccion_id VARCHAR(50) NOT NULL,
    cuenta_id INTEGER REFERENCES cuentas(id),
    tipo_operacion VARCHAR(20),
    monto NUMERIC(15,2),
    estado VARCHAR(20),
    timestamp_inicio TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    timestamp_prepare TIMESTAMP,
    timestamp_final TIMESTAMP,
    descripcion TEXT
);
```

Data Output | Messages | Notifications

```
CREATE TABLE

Query returned successfully in 43 msec.
```

```sql
CREATE TABLE control_2pc (
    transaccion_id VARCHAR(58) PRIMARY KEY,
    estado_global VARCHAR(28),
    participantes TEXT[],
    votos_commit INTEGER DEFAULT 0,
    votos_abort INTEGER DEFAULT 0,
    timestamp_inicio TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    timestamp_decision TIMESTAMP,
    coordinador VARCHAR(50)
);
```

Data Output | Messages | Notifications

```
CREATE TABLE

Query returned successfully in 42 msec.
```

```sql
INSERT INTO cuentas (numero_cuenta, titular, saldo) VALUES
('CUSCO-001', 'Rosa Quispe Huamán', 2000.00),
('CUSCO-002', 'Pedro Mamani Condori', 4500.00),
('CUSCO-003', 'Carmen Ccoa Flores', 1800.00),
('CUSCO-004', 'Luis Apaza Choque', 5300.00),
('CUSCO-005', 'Elena Puma Quispe', 3700.00);
```

Data Output | Messages | Notifications

```
INSERT 0 5

Query returned successfully in 44 msec.
```

1.4 Estructura para BANCO_AREQUIPA

Abrir Terminal 3 y conectar:

```sql
CREATE TABLE cuentas (
    id SERIAL PRIMARY KEY,
    numero_cuenta VARCHAR(20) UNIQUE NOT NULL,
    titular VARCHAR(100) NOT NULL,
    saldo NUMERIC(15,2) NOT NULL CHECK (saldo >= 0),
    sucursal VARCHAR(50) DEFAULT 'Arequipa',
    fecha_creacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    ultima_modificacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    version INTEGER DEFAULT 1
);
```

Data Output | Messages | Notifications

```
CREATE TABLE

Query returned successfully in 48 msec.
```

```sql
CREATE TABLE transacciones_log (
    id SERIAL PRIMARY KEY,
    transaccion_id VARCHAR(50) NOT NULL,
    cuenta_id INTEGER REFERENCES cuentas(id),
    tipo_operacion VARCHAR(20),
    monto NUMERIC(15,2),
    estado VARCHAR(20),
    timestamp_inicio TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    timestamp_prepare TIMESTAMP,
    timestamp_final TIMESTAMP,
    descripcion TEXT
);
```

Data Output | Messages | Notifications

```
CREATE TABLE

Query returned successfully in 47 msec.
```

```sql
CREATE TABLE control_2pc (
    transaccion_id VARCHAR(58) PRIMARY KEY,
    estado_global VARCHAR(28),
    participantes TEXT[],
    votos_commit INTEGER DEFAULT 0,
    votos_abort INTEGER DEFAULT 0,
    timestamp_inicio TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    timestamp_decision TIMESTAMP,
    coordinador VARCHAR(50)
);
```

Data Output | Messages | Notifications

```
CREATE TABLE

Query returned successfully in 50 msec.
```

```sql
INSERT INTO cuentas (numero_cuenta, titular, saldo) VALUES
('AQP-001', 'Luis Vargas Bellido', 6000.00),
('AQP-002', 'Carmen Silva Medina', 2800.00),
('AQP-003', 'Roberto Mendoza Pinto', 9200.00),
('AQP-004', 'Isabel Díaz Salazar', 4100.00),
('AQP-005', 'Jorge Paredes Ramos', 7000.00);
```

Data Output | Messages | Notifications

```
INSERT 0 5

Query returned successfully in 43 msec.
```

EJERCICIO 1: TWO-PHASE COMMIT MANUAL PASO A PASO

Escenario:

Transferir $1,000 de LIMA-001 (Lima) a CUSCO-001 (Cusco)

Generar ID de transacción único:

```sql
SELECT 'TXN-' || to_char(now(), 'YYYYMMDD-HH24MISS') AS transaccion_i
```

Data Output | Messages | Notifications

Showing rows: 1 to 1 | Page No: 1

	transaccion_id text
1	TXN-20251107-232913

Usar este ID en todos los siguientes comandos (reemplazar 'TXN-20250928-143022' con tu ID generado)

FASE 0: INICIAR TRANSACCIÓN EN TODOS LOS NODOS

Terminal 1 (Lima):

```
47  BEGIN;
48  INSERT INTO control_2pc (transaccion_id, estado_globa
49  VALUES ('TXN-20251107-232913', 'INICIADA', 'LIMA');
50  SELECT * FROM control_2pc WHERE transaccion_id = 'TXN
51
```

Data Output | Messages | Notifications

Showing rows: 1 to 1 | Page No: 1 of 1

	transaccion_id [PK] character varying (58)	estado_global character varying (28)	participantes text[]	votos_commit integer	votos_abort integer
1	TXN-20251107-232913	INICIADA	[null]	0	0

Terminal 2 (Cusco):

```
44  BEGIN;
45  INSERT INTO control_2pc (transaccion_id, estado_globa
46  VALUES ('TXN-20251107-232913', 'INICIADA', 'LIMA');
47
```

Data Output | Messages | Notifications

```
INSERT 0 1

Query returned successfully in 40 msec.
```

FASE 1: PREPARE (Preparación)

Terminal 1 (Lima) - Participante ORIGEN:

```
52  SELECT numero_cuenta, titular, saldo
53  FROM cuentas
54  WHERE numero_cuenta = 'LIMA-001' FOR UPDATE;
55
56  INSERT INTO transacciones_log
57  (transaccion_id, cuenta_id, tipo_operacion, monto, es
58  SELECT 'TXN-20251107-232913', id, 'DEBITO', 1000.00,
59  'Transferencia a CUSCO-001'
60  FROM cuentas WHERE numero_cuenta = 'LIMA-001';
61
62  UPDATE transacciones_log
63  SET estado = 'PREPARED', timestamp_prepare = CURRENT_
64  WHERE transaccion_id = 'TXN-20251107-232913' AND tipo
65
66  UPDATE control_2pc
67  SET votos_commit = votos_commit + 1, estado_global =
68  WHERE transaccion_id = 'TXN-20251107-232913';
69
70  SELECT * FROM transacciones_log WHERE transaccion_id
71  SELECT * FROM control_2pc WHERE transaccion_id = 'TXN
```

Data Output | Messages | Notifications

SQL

Showing rows: 1 to 1 | Page No: 1 of 1

	transaccion_id [PK] character varying (58)	estado_global character varying (28)	participantes text[]	votos_commit integer	votos_abort integer
1	TXN-20251107-232913	PREPARANDO	[null]	1	0

Terminal 2 (Cusco) - Participante DESTINO:

```
48  SELECT numero_cuenta, titular, saldo
49  FROM cuentas
50  WHERE numero_cuenta = 'CUSCO-001' FOR UPDATE;
51
52  INSERT INTO transacciones_log
53  (transaccion_id, cuenta_id, tipo_operacion, monto, es
54  SELECT 'TXN-20251107-232913', id, 'CREDITO', 1000.00,
55  'Transferencia desde LIMA-001'
56  FROM cuentas WHERE numero_cuenta = 'CUSCO-001';
57
58  UPDATE transacciones_log
59  SET estado = 'PREPARED', timestamp_prepare = CURRENT_
60  WHERE transaccion_id = 'TXN-20251107-232913' AND tipo
61
62  UPDATE control_2pc
63  SET votos_commit = votos_commit + 1
64  WHERE transaccion_id = 'TXN-20251107-232913';
65
66  SELECT * FROM transacciones_log WHERE transaccion_id
67  SELECT * FROM control_2pc WHERE transaccion_id = 'TXN
68
```

Data Output | Messages | Notifications

Showing rows: 1 to 1 | Page No: 1 of 1

	transaccion_id [PK] character varying (58)	estado_global character varying (28)	participantes text[]	votos_commit integer	votos_abort integer
1	TXN-20251107-232913	INICIADA	[null]	1	0

FASE 2: DECISIÓN (Commit o Abort)

Terminal 4 (Monitor/Coordinador):

```
74  SELECT transaccion_id, estado_global, votos_commit, v
75  CASE
76    WHEN votos_commit = 2 THEN 'TODOS VOTARON COMMIT -
77    WHEN votos_abort > 0 THEN 'HAY VOTOS ABORT - PROCED
78    ELSE 'ESPERANDO VOTOS'
79  END AS decision
80  FROM control_2pc
81  WHERE transaccion_id = 'TXN-20251107-232913';
82
```

Data Output | Messages | Notifications

SQL

Showing rows: 1 to 1 | Page No: 1 of 1

	transaccion_id [PK] character varying (58)	estado_global character varying (28)	votos_commit integer	votos_abort integer	decision text
1	TXN-20251107-232913	PREPARANDO	1	0	ESPERANDO VOT

Si todos votaron COMMIT (votos_commit = 2):

Terminal 1 (Lima):

```sql
UPDATE cuentas
SET saldo = saldo - 1000.00,
    ultima_modificacion = CURRENT_TIMESTAMP,
    version = version + 1
WHERE numero_cuenta = 'LIMA-001';

UPDATE transacciones_log
SET estado = 'COMMITTED', timestamp_final = CURRENT_
WHERE transaccion_id = 'TXN-20251107-232913' AND tip

UPDATE control_2pc
SET estado_global = 'CONFIRMADA', timestamp_decision
WHERE transaccion_id = 'TXN-20251107-232913';

COMMIT;

SELECT numero_cuenta, titular, saldo FROM cuentas WH
```

Data Output | Messages | Notifications

Showing rows: 1 to 1 | Page No: 1 of 1

	numero_cuenta character varying (20)	titular character varying (100)	saldo numeric (15,2)
1	LIMA-001	Juan Pérez Rodríguez	4000.00

Terminal 2 (Cusco):

```
70  UPDATE cuentas
71  SET saldo = saldo + 1000.00,
72      ultima_modificacion = CURRENT_TIMESTAMP,
73      version = version + 1
74  WHERE numero_cuenta = 'CUSCO-001';
75
76  UPDATE transacciones_log
77  SET estado = 'COMMITTED', timestamp_final = CURRENT_T
78  WHERE transaccion_id = 'TXN-20251107-232913' AND tipo
79
80  UPDATE control_2pc
81  SET estado_global = 'CONFIRMADA', timestamp_decision
82  WHERE transaccion_id = 'TXN-20251107-232913';
83
84  COMMIT;
85
86  SELECT numero_cuenta, titular, saldo FROM cuentas WHE
87
88
```

Data Output | Messages | Notifications

Showing rows: 1 to 1 | Page No: 1 of 1

	numero_cuenta character varying (20)	titular character varying (100)	saldo numeric (15,2)
1	CUSCO-001	Rosa Quispe Huamán	3000.00

VERIFICACIÓN FINAL

Terminal 4 (Monitor):

```
103 SELECT * FROM cuentas WHERE numero_cuenta = 'LIMA-001';
104 SELECT * FROM transacciones_log WHERE transaccion_id = 'TXN-20251107-2
105 SELECT * FROM control_2pc WHERE transaccion_id = 'TXN-20251107-232913'
106
```

Data Output | Messages | Notifications

Showing rows: 1 to 1 | Page No: 1 of 1

	transaccion_id [PK] character varying (58)	estado_global character varying (28)	participantes text[]	votos_commit integer	votos_abort integer	timestamp_inicio timestamp without time zone
1	TXN-20251107-232913	CONFIRMADA	[null]	1	0	2025-11-07 23:35:37.119256

```
89  SELECT * FROM cuentas WHERE numero_cuenta = 'CUSCO-001';
90  SELECT * FROM transacciones_log WHERE transaccion_id = 'TXN-2025110
91
```

Data Output | Messages | Notifications

Showing rows: 1 to 1 | Page No: 1 of 1

	id [PK] integer	transaccion_id character varying (50)	cuenta_id integer	tipo_operacion character varying (20)	monto numeric (15,2)	estado character varying (20)	timestamp_ini timestamp wit
1	1	TXN-20251107-232913	1	CREDITO	1000.00	COMMITTED	2025-11-07 23

```
114  SELECT 'LIMA' AS sucursal,
115         COALESCE(SUM(CASE WHEN tipo_operacion = 'DEBITO' THEN -monto EL
116  FROM transacciones_log
117  WHERE transaccion_id = 'TXN-20251107-232913'
118
119  UNION ALL
120
121  SELECT 'CUSCO' AS sucursal,
122         COALESCE(resultado.neto, 0) AS neto
123  FROM dblink('conn_cusco',
124      'SELECT SUM(CASE WHEN tipo_operacion = ''CREDITO'' THEN monto ELSE
125       FROM transacciones_log
126       WHERE transaccion_id = ''TXN-20251107-232913''')
127      AS resultado(neto NUMERIC);
```

Data Output | Messages | Notifications

Showing rows: 1 to 2 | Page No: 1

	sucursal text	neto numeric
1	LIMA	-1000.00
2	CUSCO	1000.00

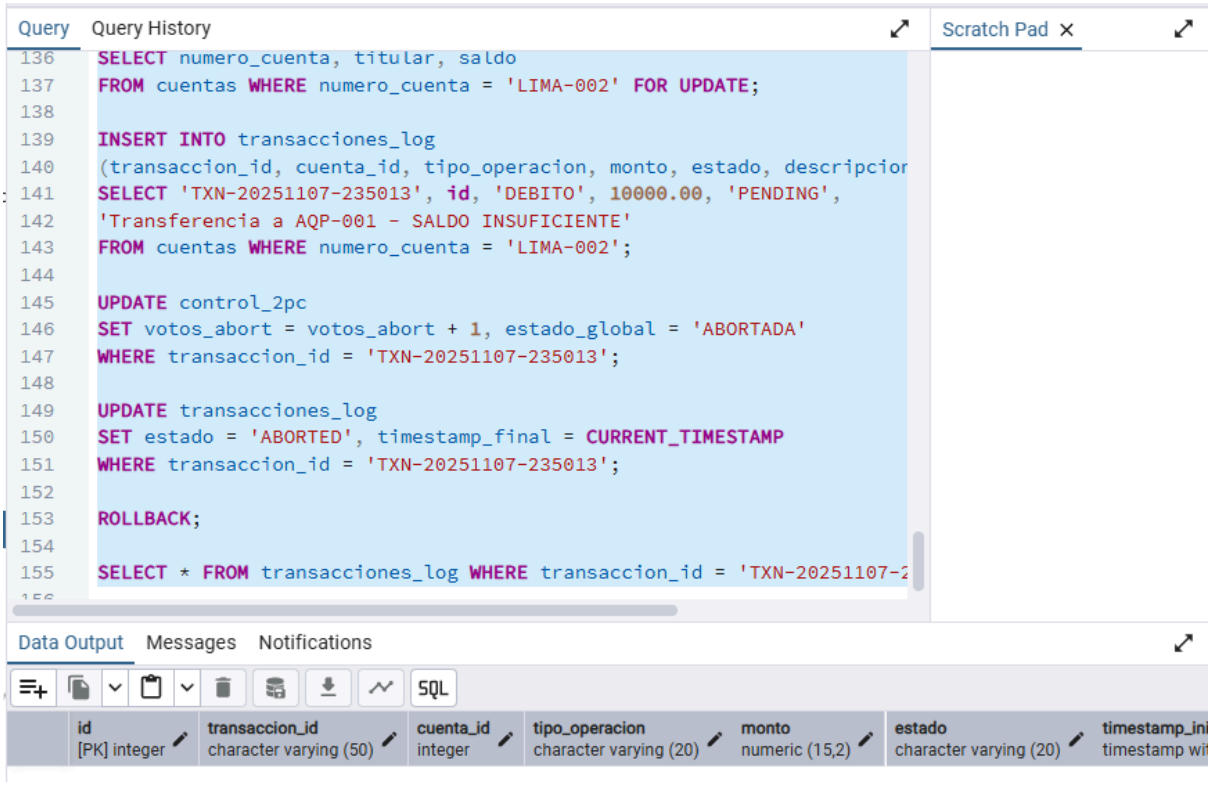
EJERCICIO 2: SIMULACIÓN DE ABORT (Saldo Insuficiente) Escenario: Intentar transferir $10,000 de LIMA-002 (saldo: $3,000) a AQP-001 **Terminal 1 (Lima):**

```
129  SELECT 'TXN-' || to_char(now(), 'YYYYMMDD-HH24MISS') AS transaccion_ic
```

Data Output | Messages | Notifications

Showing rows: 1 to 1 | Page No: 1

	transaccion_id text
1	TXN-20251107-235013



Terminal 3 (Arequipa):

EJERCICIO 3: SIMULACIÓN DE DEADLOCK DISTRIBUIDO

Escenario: Dos transferencias cruzadas simultáneas

Transferencia A: LIMA-003 → CUSCO-002 ($500)
Transferencia B: CUSCO-002 → LIMA-003 ($300)
Ejecutadas simultáneamente

Terminal 1 (Transferencia A - Lima primero):

Terminal 2 (Transferencia B - Cusco primero):

Instalación de dblink para deadlock distribuido

```
1   CREATE EXTENSION IF NOT EXISTS dblink;
2   SELECT dblink_connect('conn_cusco',
3       'host=localhost dbname=banco_cusco user=estudiante password=lab202
4
5
```

Data Output | Messages | Notifications

```
NOTICE:  la extensión «dblink» ya existe, omitiendo
```

```
98  CREATE EXTENSION IF NOT EXISTS dblink;
99  SELECT dblink_connect('conn_lima',
100 'host=localhost dbname=banco_lima user=estudiante password=lab2024
101
```

Data Output | Messages | Notifications

Showing rows: 1 to 1 | Page No: 1

	dblink_connect text
1	OK

Ahora ejecutar deadlock real:

Terminal 1:

```
160 BEGIN;
161 SELECT * FROM cuentas WHERE numero_cuenta = 'LIMA-00
162 SELECT pg_sleep(5);
163 SELECT * FROM dblink('conn_cusco',
164 'SELECT * FROM cuentas WHERE numero_cuenta = ''CUSCO-
165 AS t(id int, numero_cuenta varchar, titular varchar,
166
```

Data Output | Messages | Notifications

```
ERROR:  permiso denegado a la tabla cuentas
CONTEXT:  while executing query on dblink connection named "conn_cusco"

SQL state: 42501
```

Terminal 2 (Cusco):

Resultado esperado:

```
103  BEGIN;
104  SELECT * FROM cuentas WHERE numero_cuenta = 'CUSCO-0(
105  SELECT pg_sleep(2);
106  SELECT * FROM dblink('conn_lima',
107  'SELECT * FROM cuentas WHERE numero_cuenta = ''LIMA-(
108  AS t(id int, numero_cuenta varchar, titular varchar,
109
```

Data Output | Messages | Notifications

```
ERROR:  remote query result rowtype does not match the specified FROM clause rowtype

SQL state: 42804
```

Limpieza:

```
166  ROLLBACK;
```

Data Output | Messages | Notifications

```
ROLLBACK

Query returned successfully in 48 msec.
```

```
109  ROLLBACK;
```

Data Output | Messages | Notifications

```
ROLLBACK

Query returned successfully in 65 msec.
```

PARTE B: AUTOMATIZACIÓN CON PL/pgSQL

PASO 4: CREAR FUNCIONES ALMACENADAS

4.1 Función de preparación (PREPARE)

Terminal 1 (Lima):

```sql
CREATE OR REPLACE FUNCTION preparar_debito(
    p_transaccion_id VARCHAR,
    p_numero_cuenta VARCHAR,
    p_monto NUMERIC
) RETURNS BOOLEAN AS $$
DECLARE
    v_cuenta_id INTEGER;
    v_saldo_actual NUMERIC;
BEGIN
    SELECT id, saldo INTO v_cuenta_id, v_saldo_actua
    FROM cuentas WHERE numero_cuenta = p_numero_cuen
    IF NOT FOUND THEN RAISE NOTICE 'Cuenta % no enco
    IF v_saldo_actual < p_monto THEN RAISE NOTICE 'Sa
    INSERT INTO transacciones_log (transaccion_id, cu
    VALUES (p_transaccion_id, v_cuenta_id, 'DEBITO',
    RAISE NOTICE 'VOTE-COMMIT para cuenta %', p_numer
    RETURN TRUE;
EXCEPTION WHEN OTHERS THEN RAISE NOTICE 'Error: %',
END;
  $$ LANGUAGE plpgsql;
```

Data Output | Messages | Notifications

```
CREATE FUNCTION

Query returned successfully in 44 msec.
```

```sql
BEGIN;
SELECT preparar_debito('TXN-TEST-001', 'LIMA-001', 5(
SELECT preparar_debito('TXN-TEST-002', 'LIMA-001', 5(
ROLLBACK;
```

Data Output | Messages | Notifications

```
NOTICE:  VOTE-COMMIT para cuenta LIMA-001
NOTICE:  Saldo insuficiente
ROLLBACK

Query returned successfully in 41 msec.
```

Probar la función:

4.2 Función de preparación crédito

Terminal 2 (Cusco):

```
112  CREATE OR REPLACE FUNCTION preparar_credito(
113      p_transaccion_id VARCHAR,
114      p_numero_cuenta VARCHAR,
115      p_monto NUMERIC
116  ) RETURNS BOOLEAN AS $$
117  DECLARE v_cuenta_id INTEGER;
118  BEGIN
119      SELECT id INTO v_cuenta_id FROM cuentas WHERE nur
120      IF NOT FOUND THEN RAISE NOTICE 'Cuenta % no encor
121      INSERT INTO transacciones_log (transaccion_id, cu
122      VALUES (p_transaccion_id, v_cuenta_id, 'CREDITO'
123      RAISE NOTICE 'VOTE-COMMIT para cuenta %', p_numer
124      RETURN TRUE;
125  EXCEPTION WHEN OTHERS THEN RAISE NOTICE 'Error: %', S
126  END;
127    $$ LANGUAGE plpgsql;
```

Data Output | Messages | Notifications

```
CREATE FUNCTION
```

4.3 Función de commit

Terminal 1 (Lima):

```
197  CREATE OR REPLACE FUNCTION confirmar_transaccion(p_t
198  DECLARE v_registro RECORD;
199  BEGIN
200      FOR v_registro IN SELECT cuenta_id, tipo_operaci
201                        WHERE transaccion_id = p_trans
202          IF v_registro.tipo_operacion = 'DEBITO' THEN
203              UPDATE cuentas SET saldo = saldo - v_reg
204                                 ultima_modificacion =
205                                 version = version + 1
206          ELSIF v_registro.tipo_operacion = 'CREDITO'
207              UPDATE cuentas SET saldo = saldo + v_reg
208                                 ultima_modificacion =
209                                 version = version + 1
210          END IF;
211          UPDATE transacciones_log SET estado = 'COMMI
212          WHERE transaccion_id = p_transaccion_id AND
213          RAISE NOTICE 'Operación % confirmada', v_reg
214      END LOOP;
215      UPDATE control_2pc SET estado_global = 'CONFIRMA
216      WHERE transaccion_id = p_transaccion_id;
217  END;
218    $$ LANGUAGE plpgsql;
219
```

Data Output | Messages | Notifications

```
CREATE FUNCTION

Query returned successfully in 41 msec.
```

Copiar la misma función en Cusco (Terminal 2)

```
130  CREATE OR REPLACE FUNCTION confirmar_transaccion(p_t
131  DECLARE v_registro RECORD;
132  BEGIN
133      FOR v_registro IN SELECT cuenta_id, tipo_operaci
134                        WHERE transaccion_id = p_trans
135          IF v_registro.tipo_operacion = 'DEBITO' THEN
136              UPDATE cuentas SET saldo = saldo - v_reg
137                                 ultima_modificacion =
138                                 version = version + 1
139          ELSIF v_registro.tipo_operacion = 'CREDITO'
140              UPDATE cuentas SET saldo = saldo + v_reg
141                                 ultima_modificacion =
142                                 version = version + 1
143          END IF;
144          UPDATE transacciones_log SET estado = 'COMMI
145          WHERE transaccion_id = p_transaccion_id AND
146          RAISE NOTICE 'Operación % confirmada', v_reg
147      END LOOP;
148      UPDATE control_2pc SET estado_global = 'CONFIRMA
149      WHERE transaccion_id = p_transaccion_id;
150  END;
151    $$ LANGUAGE plpgsql;
152
```

Data Output | Messages | Notifications

```
CREATE FUNCTION

Query returned successfully in 43 msec.
```

4.4 Función de abort

Terminal 1 (Lima) y Terminal 2 (Cusco):

```
221  CREATE OR REPLACE FUNCTION abortar_transaccion(p_tra
222  BEGIN
223      UPDATE transacciones_log SET estado = 'ABORTED',
224      WHERE transaccion_id = p_transaccion_id;
225      UPDATE control_2pc SET estado_global = 'ABORTADA
226      WHERE transaccion_id = p_transaccion_id;
227      RAISE NOTICE 'Transacción % abortada', p_transac
228  END;
229    $$ LANGUAGE plpgsql;
```

Data Output | Messages | Notifications

```
CREATE FUNCTION

Query returned successfully in 51 msec.
```

```
154  CREATE OR REPLACE FUNCTION abortar_transaccion(p_tra
155  BEGIN
156      UPDATE transacciones_log SET estado = 'ABORTED',
157      WHERE transaccion_id = p_transaccion_id;
158      UPDATE control_2pc SET estado_global = 'ABORTADA
159      WHERE transaccion_id = p_transaccion_id;
160      RAISE NOTICE 'Transacción % abortada', p_transac
161  END;
162    $$ LANGUAGE plpgsql;
```

Data Output | Messages | Notifications

```
CREATE FUNCTION

Query returned successfully in 52 msec.
```

EJERCICIO 4: USAR FUNCIONES PARA 2PC AUTOMATIZADO

```
231
232  SELECT 'TXN-' || to_char(now(), 'YYYYMMDD-HH24MISS')
233
```

Data Output | Messages | Notifications

Showing rows: 1 to 1 | Page No: 1 of 1

	transaccion_id text
1	TXN-20251108-001540

Escenario: Transferir $800 de LIMA-004 a CUSCO-003

Terminal 1 (Lima):

```
233  BEGIN;
234  INSERT INTO control_2pc (transaccion_id, estado_globa
235  VALUES ('TXN-20251108-001540', 'PREPARANDO', 'LIMA')
236  SELECT preparar_debito('TXN-20251108-001540', 'LIMA-
237
```

Data Output | Messages | Notifications

Showing rows: 1 to 1 | Page No: 1 of 1

	preparar_debito boolean
1	true

Terminal 2 (Cusco):

```
165   BEGIN;
166 INSERT INTO control_2pc (transaccion_id, estado_globa
167 VALUES ('TXN-20251108-001540', 'PREPARANDO', 'LIMA')
168 SELECT preparar_credito('TXN-20251108-001540', 'CUSC
169
```

Data Output | Messages | Notifications

Showing rows: 1 to 1 | Page No: 1 of 1

	preparar_credito boolean
1	true

Terminal 4 (Monitor - verificar votos):

```
239 SELECT * FROM transacciones_log WHERE transaccion_id
240
```

Data Output | Messages | Notifications

Showing rows: 1 to 1 | Page No: 1 of 1

	id [PK] integer	transaccion_id character varying (50)	cuenta_id integer	tipo_operacion character varying (20)	monto numeric (15,2)	estado character varyi
1	4	TXN-20251108-001540	4	DEBITO	800.00	PREPARED

```
169
170 SELECT * FROM transacciones_log WHERE transaccion_id
171
```

Data Output | Messages | Notifications

Showing rows: 1 to 1 | Page No: 1 of 1

	id [PK] integer	transaccion_id character varying (50)	cuenta_id integer	tipo_operacion character varying (20)	monto numeric (15,2)	estado character varyi
1	2	TXN-20251108-001540	3	CREDITO	800.00	PREPARED

✓ Successfully run. Total query runtime: 68 msec. 1 rows affected.

Total rows: 1 | Query complete 00:00:00.068 | CRLF | Ln 171, Col 1

Si ambos votaron COMMIT, ejecutar FASE 2:

Terminal 1 (Lima):

```
242  SELECT confirmar_transaccion('TXN-20251108-001540');
243  COMMIT;
244  SELECT saldo FROM cuentas WHERE numero_cuenta = 'LIM
245
```

Data Output | Messages | Notifications

Showing rows: 1 to 1 | Page No: 1 of 1

	saldo numeric (15,2)
1	2000.00

Terminal 2 (Cusco):

```
173  SELECT confirmar_transaccion('TXN-20251108-001540');
174  COMMIT;
175  SELECT saldo FROM cuentas WHERE numero_cuenta = 'CUS
176
```

Data Output | Messages | Notifications

Showing rows: 1 to 1 | Page No: 1 of 1

	saldo numeric (15,2)
1	2600.00

PASO 5: FUNCIÓN COORDINADORA COMPLETA

5.1 Crear función coordinadora avanzada

Terminal 1 (Lima):

```
285         PERFORM dblink_exec(v_dblink_name,
286             format('SELECT confirmar_transaccion(%L)
287         );
288         PERFORM dblink_disconnect(v_dblink_name);
289         RETURN QUERY SELECT TRUE, 'Transferencia exi
290     ELSE
291         RAISE NOTICE 'Decisión: GLOBAL-ABORT';
292         PERFORM abortar_transaccion(v_transaccion_id
293         PERFORM dblink_exec(v_dblink_name,
294             format('SELECT abortar_transaccion(%L)',
295         );
296         PERFORM dblink_disconnect(v_dblink_name);
297         RETURN QUERY SELECT FALSE, 'Transferencia ab
298     END IF;
299 EXCEPTION
300     WHEN OTHERS THEN
301         RAISE NOTICE 'Error: %', SQLERRM;
302         BEGIN
303             PERFORM abortar_transaccion(v_transaccio
304             PERFORM dblink_disconnect(v_dblink_name)
305         EXCEPTION WHEN OTHERS THEN NULL;
306         END;
307         RETURN QUERY SELECT FALSE, 'Error: ' || SQLE
308 END;
309   $$ LANGUAGE plpgsql;
310
```

Data Output | Messages | Notifications

```
CREATE FUNCTION

Query returned successfully in 46 msec.
```

5.2 Usar la función coordinadora

Terminal 1 (Lima):

```
312  CREATE EXTENSION IF NOT EXISTS dblink;
313  BEGIN;
314  SELECT * FROM transferencia_distribuida_coordinador
315      'LIMA-005', 'CUSCO-004', 1200.00, 'cusco'
316  );
317  COMMIT;
318  SELECT saldo FROM cuentas WHERE numero_cuenta = 'LIM
319
```

Data Output | Messages | Notifications

Showing rows: 1 to 1 | Page No: 1 of 1

	saldo numeric (15,2)
1	6200.00

Terminal 2 (Cusco):

```
178  SELECT saldo FROM cuentas WHERE numero_cuenta = 'CUS
179  SELECT * FROM transacciones_log ORDER BY timestamp_i
180
```

Data Output | Messages | Notifications

Showing rows: 1 to 2 | Page No: 1 of 1

	id [PK] integer	transaccion_id character varying (50)	cuenta_id integer	tipo_operacion character varying (20)	monto numeric (15,2)	estado character varyi
1	2	TXN-20251108-001540	3	CREDITO	800.00	COMMITTED
2	1	TXN-20251107-232913	1	CREDITO	1000.00	COMMITTED

✓ Successfully run. Total query runtime: 70 msec. 2 rows affected.

Total rows: 2 | Query complete 00:00:00.070 | CRLF | Ln 178, Col 1

PARTE C: SAGA PATTERN CON TRIGGERS

PASO 6: IMPLEMENTAR SAGA CON COMPENSACIONES

6.1 Crear tablas para SAGA

Terminal 1 (Lima):

```sql
CREATE TABLE saga_ordenes (
    orden_id VARCHAR(50) PRIMARY KEY,
    tipo VARCHAR(50),
    estado VARCHAR(20),
    datos JSONB,
    paso_actual INTEGER DEFAULT 0,
    timestamp_inicio TIMESTAMP DEFAULT CURRENT_TIMES
    timestamp_final TIMESTAMP
);
CREATE TABLE saga_pasos (
    id SERIAL PRIMARY KEY,
    orden_id VARCHAR(50) REFERENCES saga_ordenes(orde
    numero_paso INTEGER,
    nombre_paso VARCHAR(100),
    estado VARCHAR(20),
    accion_ejecutada TEXT,
    compensacion_ejecutada TEXT,
    timestamp_ejecucion TIMESTAMP,
    timestamp_compensacion TIMESTAMP,
    error_mensaje TEXT
);
CREATE TABLE saga_eventos (
    id SERIAL PRIMARY KEY,
    orden_id VARCHAR(50) REFERENCES saga_ordenes(orde
    tipo_evento VARCHAR(50),
    descripcion TEXT,
```

Data Output | Messages | Notifications

```
CREATE TABLE

Query returned successfully in 57 msec.
```

Copiar las mismas tablas en Cusco y Arequipa

6.2 Crear función para ejecutar SAGA

Terminal 1 (Lima):

```
352  CREATE OR REPLACE FUNCTION ejecutar_saga_transferenc
353      p_cuenta_origen VARCHAR,
354      p_cuenta_destino VARCHAR,
355      p_monto NUMERIC,
356      p_db_destino VARCHAR
357  ) RETURNS TABLE (
358      exito BOOLEAN,
359      orden_id VARCHAR,
360      mensaje TEXT
361  ) AS $$
362  DECLARE
363      v_orden_id VARCHAR;
364      v_paso1_exito BOOLEAN := FALSE;
365      v_paso2_exito BOOLEAN := FALSE;
366      v_paso3_exito BOOLEAN := FALSE;
367      v_cuenta_origen_id INTEGER;
368      v_saldo_origen NUMERIC;
369  BEGIN
370      v_orden_id := 'SAGA-' || to_char(now(), 'YYYYMMDI
371
372      INSERT INTO saga_ordenes (orden_id, tipo, estado
373      VALUES (
374          v_orden_id,
375          'TRANSFERENCIA',
```

Data Output | Messages | Notifications

```
CREATE FUNCTION

Query returned successfully in 42 msec.
```

6.3 Probar SAGA exitosa

Terminal 1 (Lima):

```
581  BEGIN;
582  SELECT * FROM ejecutar_saga_transferencia(
583      'LIMA-001', 'CUSCO-005', 300.00, 'cusco'
584  );
585  COMMIT;
586  SELECT * FROM saga_ordenes ORDER BY timestamp_inicio
587  SELECT * FROM saga_pasos WHERE orden_id = (SELECT ord
588  SELECT * FROM saga_eventos WHERE orden_id = (SELECT
589
```

Data Output | Messages | Notifications

SQL

Showing rows: 1 to 3 | Page No: 1 of 1

	id [PK] integer	orden_id character varying (50)	tipo_evento character varying (50)	descripcion text
1	2	SAGA-20251108-003931	PASO COMPLETADO	Paso 1: Fondos bloqueados
2	3	SAGA-20251108-003931	PASO_FALLIDO	Paso 2: permiso denegado a la tabla
3	4	SAGA-20251108-003931	COMPENSACION_EJECUTADA	Compensación Paso 1: Fondos desb

6.4 Probar SAGA con fallo y compensación

Terminal 1 (Lima):

```
592  BEGIN;
593  SELECT * FROM ejecutar_saga_transferencia(
594      'LIMA-002', 'CUSCO-999', 500.00, 'cusco'
595  );
596  COMMIT;
597  SELECT numero_paso, nombre_paso, estado, accion_ejecu
598  FROM saga_pasos
599  WHERE orden_id = (SELECT orden_id FROM saga_ordenes (
600  ORDER BY numero_paso;
601  SELECT * FROM saga_eventos
602  WHERE orden_id = (SELECT orden_id FROM saga_ordenes (
603  ORDER BY timestamp_evento;
604
```

Data Output | Messages | Notifications

Showing rows: 1 to 3 | Page No: 1 of 1

	id [PK] integer	orden_id character varying (50)	tipo_evento character varying (50)	descripcion text
1	5	SAGA-20251108-004033	PASO COMPLETADO	Paso 1: Fondos bloqueados
2	6	SAGA-20251108-004033	PASO_FALLIDO	Paso 2: duplicate connection name
3	7	SAGA-20251108-004033	COMPENSACION_EJECUTADA	Compensación Paso 1: Fondos desb