

MÓDULO PROGRAMADOR

Proyecto Final

Equipo: Tuple-Developers

INTEGRANTES:

- JUAN IGNACIO MORENO (JUANIGNACIOM03)
- JORGE ANTONIO MARCHISONE (JMARCHISONE) (SCRUMMASTER)
- MARTIN ERNESTO OLMEDO (MARTIN-OLMEDO98)
- FABIÁN RICARDO PERISSET (FABIANRICARDOPERISSET)
- SABRINA LOZA (SALOZA14)
- FACUNDO LOZANO (FACUNDOLOZANO0)

BASE DE DATOS:

<https://onecompiler.com/mysql>

Dentro de lo posible, utilizar el código de los archivos SQL que están dentro de la carpeta Base de Datos, ya que puede perder el formato en el PDF.

DDL:

```
CREATE TABLE USUARIO (  
    email VARCHAR(100) PRIMARY KEY,  
    cuil VARCHAR(20) UNIQUE NOT NULL,  
    nombre VARCHAR(40) NOT NULL,  
    apellido VARCHAR(40) NOT NULL,  
    password VARCHAR(255) NOT NULL,  
    saldo DECIMAL(15, 2) NOT NULL DEFAULT 1000000  
);
```

```
CREATE TABLE PORTAFOLIO (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    usuario_email VARCHAR(100) UNIQUE NOT NULL,
```

```
total_invertido DECIMAL(15, 2) NOT NULL DEFAULT 0,  
rendimiento_total DECIMAL(15, 2) NOT NULL DEFAULT 0,  
FOREIGN KEY (usuario_email) REFERENCES USUARIO(email)  
);
```

```
CREATE TABLE ACTIVO (  
    id INT PRIMARY KEY auto_increment,  
    nombre VARCHAR(100) NOT NULL,  
    simbolo VARCHAR(10) NOT NULL,  
    descripcion TEXT,  
    precio_compra DECIMAL(12,2) NOT NULL,  
    precio_venta DECIMAL(12,2) NOT NULL,  
    volumen_disponible INTEGER NOT NULL,  
    ultima_actualizacion TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,  
    activo BOOLEAN DEFAULT true  
);
```

```
CREATE TABLE PORTAFOLIO_ACTIVO (  
    portafolio_id INT NOT NULL,  
    activo_id INT NOT NULL,  
    cantidad INT NOT NULL,  
    PRIMARY KEY (portafolio_id, activo_id),  
    FOREIGN KEY (portafolio_id) REFERENCES PORTAFOLIO(id),  
    FOREIGN KEY (activo_id) REFERENCES ACTIVO(id)  
);
```

-- Cuidado al crear, tipo solo puede ser compra o venta

```
CREATE TABLE TRANSACCION (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    usuario_email VARCHAR(100) NOT NULL,  
    activo_id INT NOT NULL,  
    tipo ENUM('compra', 'venta') NOT NULL,  
    cantidad INT NOT NULL,  
    precio DECIMAL(15, 2) NOT NULL,  
    comision DECIMAL(15, 2) NOT NULL,
```

```
fecha DATETIME NOT NULL,  
  
FOREIGN KEY (usuario_email) REFERENCES USUARIO(email),  
  
FOREIGN KEY (activo_id) REFERENCES ACTIVO(id)  
  
);
```

DML:

--Inserción de datos

insert into

usuario (email, cuil, nombre, apellido, password, saldo)

values

```
(  
  
    'juan.perez@example.com',  
  
    '20-12345678-9',  
  
    'Juan',  
  
    'Perez',  
  
    'password123',  
  
    1000000  
  
,  
  
(  
  
    'maria.gomez@example.com',  
  
    '27-87654321-0',  
  
    'Maria',  
  
    'Gomez',  
  
    'password456',  
  
    1000000  
  
,  
  
(  
  
    'carlos.lopez@example.com',  
  
    '23-23456789-1',  
  
    'Carlos',  
  
    'Lopez',  
  
    'password789',  
  
    1000000
```

```
),  
(  
  'ana.martinez@example.com',  
  '24-34567890-2',  
  'Ana',  
  'Martinez',  
  'password012',  
  1000000  
)  
(  
  'luis.sanchez@example.com',  
  '25-45678901-3',  
  'Luis',  
  'Sanchez',  
  'password345',  
  1000000  
)  
(  
  'laura.diaz@example.com',  
  '26-56789012-4',  
  'Laura',  
  'Diaz',  
  'password678',  
  1000000  
)  
(  
  'jose.fernandez@example.com',  
  '27-67890123-5',  
  'Jose',  
  'Fernandez',  
  'password901',  
  1000000  
)  
(  
  'marta.garcia@example.com',
```

```
'28-78901234-6',  
  
'Marta',  
  
'Garcia',  
  
'password234',  
  
1000000  
,  
(  
  
'pedro.ramirez@example.com',  
  
'29-89012345-7',  
  
'Pedro',  
  
'Ramirez',  
  
'password567',  
  
1000000  
,  
(  
  
'sofia.torres@example.com',  
  
'30-90123456-8',  
  
'Sofia',  
  
'Torres',  
  
'password890',  
  
1000000  
);
```

insert into

portafolio (usuario_email, total_invertido, rendimiento_total)

values

```
('juan.perez@example.com', 500000, 50000),  
( 'maria.gomez@example.com', 300000, 30000),  
( 'carlos.lopez@example.com', 200000, 20000),  
( 'ana.martinez@example.com', 400000, 40000),  
( 'luis.sanchez@example.com', 100000, 10000),  
( 'laura.diaz@example.com', 600000, 60000),  
( 'jose.fernandez@example.com', 700000, 70000),  
( 'marta.garcia@example.com', 800000, 80000),  
( 'pedro.ramirez@example.com', 900000, 90000),
```

('sofia.torres@example.com', 1000000, 100000);

insert into

activo (

nombre,

simbolo,

descripcion,

precio_compra,

precio_venta,

volumen_disponible

)

values

('Grupo Financiero Galicia', 'GGAL', 'Banco líder en Argentina', 5800, 5900, 2880),

('Ternium Argentina', 'TXAR', 'Productor de acero', 790, 800, 1020),

('Pampa Energía', 'PAMP', 'Empresa de energía', 3020, 3100, 1990),

('Telecom Argentina', 'TECO2', 'Proveedor de telecomunicaciones', 1820, 1920, 419),

('Aluar', 'ALUA', 'Productor de aluminio', 843, 849, 810),

('BBVA Banco Francés', 'BBAR', 'Banco comercial', 4650, 4715, 222),

('Banco Macro', 'BMA', 'Banco comercial', 8650, 8850, 318),

('Edenor', 'EDN', 'Distribuidor de electricidad', 1340, 1375, 4722),

('Mirgor', 'MIRG', 'Fabricante de productos electrónicos', 20425, 21800, 5740),

('Central Puerto', 'CEPU', 'Generador de energía eléctrica', 1205, 1250, 1270);

INSERT INTO transaccion (

usuario_email,

activo_id,

tipo,

cantidad,

precio,

comision,

fecha

)

VALUES

('juan.perez@example.com', 1, 'compra', 10, 1500.00, 15.00, '2023-10-01 10:00:00'),

('maria.gomez@example.com', 2, 'venta', 5, 1250.00, 12.50, '2023-10-02 11:00:00'),

('carlos.lopez@example.com', 3, 'compra', 2, 6600.00, 66.00, '2023-10-03 12:00:00'),

('ana.martinez@example.com', 4, 'venta', 3, 2100.00, 21.00, '2023-10-04 13:00:00'),
('luis.sanchez@example.com', 5, 'compra', 1, 2800.00, 28.00, '2023-10-05 14:00:00'),
('laura.diaz@example.com', 6, 'venta', 4, 1400.00, 14.00, '2023-10-06 15:00:00'),
('jose.fernandez@example.com', 7, 'compra', 6, 1200.00, 12.00, '2023-10-07 16:00:00'),
('marta.garcia@example.com', 8, 'venta', 7, 3500.00, 35.00, '2023-10-08 17:00:00'),
('pedro.ramirez@example.com', 9, 'compra', 8, 4800.00, 48.00, '2023-10-09 18:00:00'),
('sofia.torres@example.com', 10, 'venta', 9, 450.00, 4.50, '2023-10-10 19:00:00'),
('juan.perez@example.com', 2, 'venta', 3, 3750.00, 37.50, '2023-10-11 10:00:00'),
('maria.gomez@example.com', 3, 'compra', 4, 13200.00, 132.00, '2023-10-12 11:00:00'),
('carlos.lopez@example.com', 4, 'venta', 5, 3500.00, 35.00, '2023-10-13 12:00:00'),
('ana.martinez@example.com', 5, 'compra', 2, 5600.00, 56.00, '2023-10-14 13:00:00'),
('luis.sanchez@example.com', 6, 'venta', 1, 1400.00, 14.00, '2023-10-15 14:00:00'),
('laura.diaz@example.com', 7, 'compra', 3, 3600.00, 36.00, '2023-10-16 15:00:00'),
('jose.fernandez@example.com', 8, 'venta', 2, 7000.00, 70.00, '2023-10-17 16:00:00'),
('marta.garcia@example.com', 9, 'compra', 1, 4800.00, 48.00, '2023-10-18 17:00:00'),
('pedro.ramirez@example.com', 10, 'venta', 4, 1800.00, 18.00, '2023-10-19 18:00:00'),
('sofia.torres@example.com', 1, 'compra', 5, 7500.00, 75.00, '2023-10-20 19:00:00');

insert into portafolio_activo (portafolio_id, activo_id, cantidad)

values

(1, 1, 20),
(2, 2, 10),
(3, 3, 6),
(4, 4, 8),
(5, 5, 3),
(6, 6, 5),
(7, 7, 9),
(8, 8, 9),
(9, 9, 9),
(10, 10, 13),
(1, 2, 3),
(2, 3, 4),
(3, 4, 5),
(4, 5, 2),
(5, 6, 1),

(6, 7, 3),
(7, 8, 2),
(8, 9, 1),
(9, 10, 4),
(10, 1, 5);

--Consultas de update

UPDATE USUARIO

SET saldo = saldo + 5000

WHERE email = 'juan.perez@example.com';

UPDATE USUARIO

SET apellido = 'Rodriguez'

WHERE email = 'maria.gomez@example.com';

UPDATE USUARIO

SET password = 'newpassword123'

WHERE email = 'carlos.lopez@example.com';

UPDATE ACTIVO

SET precio_compra = precio_compra * 1.05, precio_venta = precio_venta * 1.05

WHERE simbolo = 'TXAR';

UPDATE ACTIVO

SET precio_compra = precio_compra * 0.95, precio_venta = precio_venta * 0.95

WHERE simbolo = 'BBAR';

--Muestra todas las acciones compradas de todos los usuarios, y con que precio.

SELECT u.nombre, u.apellido, p.usuario_email, pa.cantidad, a.id as id_activo, a.nombre, a.simbolo, a.precio_compra,
a.precio_venta

FROM PORTAFOLIO p

LEFT JOIN PORTAFOLIO_ACTIVO pa ON p.id = pa.portafolio_id

LEFT JOIN ACTIVO a ON pa.activo_id = a.id

INNER JOIN usuario u ON p.usuario_email = u.email

--Muestra los usuarios que mas transacciones realizan. Puede usarse para aplicar alguna campaña de marketing.

```
SELECT u.nombre, u.apellido, u.email,  
COUNT(t.id) AS cantidad_transacciones  
FROM USUARIO u  
LEFT JOIN TRANSACCION t ON u.email = t.usuario_email  
GROUP BY u.email, u.nombre, u.apellido  
ORDER BY cantidad_transacciones DESC;
```

--Mostrar la cantidad total de transacciones que se han realizado por cada activo. Permite conocer las acciones más populares.

```
SELECT a.nombre, a.simbolo,  
COUNT(t.id) AS cantidad_transacciones  
FROM ACTIVO a  
LEFT JOIN TRANSACCION t ON a.id = t.activo_id  
GROUP BY a.id, a.nombre, a.simbolo;
```