

Modulo Programador

TSDS - 2024

Proyecto Final

Docentes :

Ivana Soledad ROJAS CÓRSICO
Maximiliano GIRAUDO
Martín Alejandro GERLERO

Integrantes

Cabrera Julieta Victoria
Chavez Cuffa Rodrigo Mateo
Carnero Penelas Denise Leticia
Casarino Mariano Ezequiel
Corvalán Alejandro Matías
Barrera Lautaro Ezequiel
Bordon Rios Matías Leon

Link al motor de bases de datos online:

<https://onecompiler.com/mysql>

Consultas:

```
CREATE TABLE Inversor (  
    id_inversor INT AUTO_INCREMENT PRIMARY KEY,  
    nombre VARCHAR(20) NOT NULL,  
    apellido VARCHAR(20) NOT NULL,  
    cuil VARCHAR(13) NOT NULL,  
    correo VARCHAR(100) NOT NULL,  
    contrasenia VARCHAR(50) NOT NULL  
);  
  
CREATE TABLE Portafolio (  
    id_portafolio INT AUTO_INCREMENT PRIMARY KEY,  
    id_inversor INT NOT NULL,  
    saldo FLOAT DEFAULT 1000000.0,  
    total_invertido FLOAT,  
    rendimiento FLOAT,  
    FOREIGN KEY (id_inversor) REFERENCES Inversor(id_inversor)  
);  
  
CREATE TABLE Accion (  
    id_accion INT AUTO_INCREMENT PRIMARY KEY,  
    simbolo VARCHAR(10) NOT NULL,  
    nombre_empresa VARCHAR(100) NOT NULL,  
    precio_venta FLOAT,  
    precio_compra FLOAT  
);  
  
CREATE TABLE Operacion (  
    id_operacion INT AUTO_INCREMENT PRIMARY KEY,  
    id_portafolio INT NOT NULL,  
    id_accion INT NOT NULL,  
    id_tipo BOOLEAN,  
    fecha_operacion DATE NOT NULL,  
    precio FLOAT,  
    cantidad INT,  
    total_accion FLOAT,
```

```

        comision FLOAT,
        FOREIGN KEY (id_portafolio) REFERENCES
Portafolio(id_portafolio),
        FOREIGN KEY (id_accion) REFERENCES Accion(id_accion)
);

CREATE TABLE PortafolioAccion (
    id_portafolio INT NOT NULL,
    id_accion INT NOT NULL,
    cantidad INT NOT NULL,
    FOREIGN KEY (id_portafolio) REFERENCES
Portafolio(id_portafolio),
    FOREIGN KEY (id_accion) REFERENCES Accion(id_accion)
);

-- Conjunto de Sentencias DML tipo INSERT para insertar datos
iniciales a la base de datos

INSERT INTO Inversor (nombre, apellido, cuil, correo, contrasenia)
VALUES
('Santiago', 'Lopez', '20412345678', 'santiago.lopez@mail.com',
'passSanti2024'),
('Camila', 'Fernandez', '27234567891', 'camila.fernandez@mail.com',
'camilaf_1234'),
('Diego', 'Martinez', '20387654321', 'diego.martinez@mail.com',
'martinezDiego2023');

INSERT INTO Portafolio (id_inversor, saldo, total_invertido,
rendimiento) VALUES
(1, 1100000.0, 600000.0, 0.07),
(2, 1250000.0, 400000.0, 0.03),
(3, 950000.0, 200000.0, 0.04);

INSERT INTO Accion (simbolo, nombre_empresa, precio_venta,
precio_compra) VALUES
('AAPL', 'Apple Inc.', 190.0, 185.0),
('MSFT', 'Microsoft Corp.', 260.0, 250.5),
('GOOGL', 'Alphabet Inc.', 2850.0, 2800.0);

```

```

INSERT INTO Operacion (id_portafolio, id_accion, id_tipo,
fecha_operacion, precio, cantidad, total_accion, comision) VALUES
(1, 1, 1, '2024-08-10', 185.0, 50, 9250.0, 22.5), -- compra
(1, 2, 0, '2024-09-15', 260.0, 15, 3900.0, 12.5), -- venta
(2, 1, 1, '2024-07-05', 185.0, 40, 7400.0, 18.0), -- compra
(3, 3, 1, '2024-08-21', 2800.0, 6, 16800.0, 35.0); -- compra

INSERT INTO PortafolioAccion (id_portafolio, id_accion, cantidad)
VALUES
(1, 1, 50),
(1, 2, 15),
(2, 1, 40),
(3, 3, 6);

-- 4) Consultas de tipo UPDATE para actualizar datos de los datos
ya insertados

-- 1. Actualiza el saldo de un portafolio después de una inversión.
UPDATE Portafolio
SET saldo = saldo - 10000.0, total_invertido = total_invertido +
10000.0
WHERE id_portafolio = 1;

-- 2. Modifica el precio de venta de una acción.
UPDATE Accion
SET precio_venta = 195.0
WHERE simbolo = 'AAPL';

-- 3. Actualiza el correo y contraseña de un inversor.
UPDATE Inversor
SET correo = 'santi.lopez.new@mail.com', contrasenia =
'nuevaPass2024'
WHERE id_inversor = 1;

-- 4. Modifica la cantidad de acciones en un portafolio específico.
UPDATE PortafolioAccion
SET cantidad = cantidad + 10
WHERE id_portafolio = 1 AND id_accion = 1;

```

```

-- 5. Actualiza el rendimiento de un portafolio.
UPDATE Portafolio
SET rendimiento = rendimiento + 0.02
WHERE id_portafolio = 2;

-- 6. Actualiza el precio de venta de una acción.
UPDATE Accion
SET precio_venta = 195.0
WHERE simbolo = 'AAPL';

-- 7. Reduce el precio de compra de una acción.
UPDATE Accion
SET precio_compra = 245.0
WHERE simbolo = 'MSFT';

-- 8. Cambia el nombre de la empresa asociada a una acción despues
de una supuesta fusión.
UPDATE Accion
SET nombre_empresa = 'Alphabet Inc. - Nueva División'
WHERE simbolo = 'GOOGL';

-- 9. Incrementa el precio de venta y compra de una acción.
UPDATE Accion
SET precio_venta = precio_venta * 1.05, precio_compra =
precio_compra * 1.05
WHERE simbolo = 'AAPL';

-- 10. Cambia el símbolo de una acción que hizo un rebranding.
UPDATE Accion
SET simbolo = 'META'
WHERE nombre_empresa = 'Facebook Inc.';

-- 5) Consultas de tipo SELECT que permiten obtener datos de los
datos ya insertados.

-- 1. Obtener el nombre, apellido y saldo de cada inversor con el
total invertido en su portafolio.
SELECT i.nombre, i.apellido, p.saldo, p.total_invertido
FROM Inversor i
JOIN Portafolio p ON i.id_inversor = p.id_inversor;

```

```

-- 2. Listado de todas las acciones disponibles con su símbolo,
nombre de empresa y precio de venta actual.
SELECT simbolo, nombre_empresa, precio_venta
FROM Accion;

-- 3. Ver el historial de operaciones realizadas en un portafolio
específico, mostrando fecha, tipo de operación (compra o venta),
cantidad y precio.
SELECT o.fecha_operacion,
       CASE WHEN o.id_tipo = 1 THEN 'Compra' ELSE 'Venta' END AS
tipo_operacion,
       o.cantidad, o.precio, o.total_accion, o.comision
FROM Operacion o
WHERE o.id_portafolio = 1; -- Podemos cambiar el "1" por el ID que
nos interese

-- 4. Ver el rendimiento actual de cada portafolio, con su ID y el
rendimiento.
SELECT id_portafolio, rendimiento
FROM Portafolio;

-- 5. Obtener un resumen de cada inversor, con su nombre, apellido,
cantidad total de acciones en su portafolio y el saldo disponible.
SELECT i.nombre, i.apellido, SUM(pa.cantidad) AS total_acciones,
p.saldo
FROM Inversor i
JOIN Portafolio p ON i.id_inversor = p.id_inversor
JOIN PortafolioAccion pa ON p.id_portafolio = pa.id_portafolio
GROUP BY i.id_inversor, p.saldo;

-- 6) Consultas multitabla que permiten obtener datos de interés
para el caso de estudio.

-- 1. Consulta de Portafolios con el Valor Total de Acciones
Actuales
-- Calcula el valor actual de todas las acciones que posee cada
portafolio. Multiplicamos el precio de venta actual por la cantidad
de acciones para obtener el valor total de cada tipo de acción en

```

el portafolio y lo sumamos. Nos ayuda a entender el valor total de las inversiones activas de cada inversor.

```
SELECT p.id_portafolio, i.nombre, i.apellido, SUM(a.precio_venta *  
pa.cantidad) AS valor_total_acciones  
FROM Portafolio p  
JOIN Inversor i ON p.id_inversor = i.id_inversor  
JOIN PortafolioAccion pa ON p.id_portafolio = pa.id_portafolio  
JOIN Accion a ON pa.id_accion = a.id_accion  
GROUP BY p.id_portafolio;
```

-- 2. Consulta de Inversores con el Total de Comisiones Pagadas en Operaciones

-- Sirve para conocer cuánto pago cada inversor en comisiones a lo largo de todas sus operaciones. Esto es ser útil para analizar los costos de transacción acumulados por cada inversor y ver qué tan frecuentemente están operando en el mercado.

```
SELECT i.nombre, i.apellido, SUM(o.comision) AS total_comisiones  
FROM Inversor i  
JOIN Portafolio p ON i.id_inversor = p.id_inversor  
JOIN Operacion o ON p.id_portafolio = o.id_portafolio  
GROUP BY i.id_inversor;
```

-- 3. Consulta de Acciones con el Rendimiento Promedio por Operación

-- Queremos ver el rendimiento promedio de cada acción en función de las operaciones realizadas. Calculamos el promedio de las diferencias entre los precios de compra y venta de cada operación. Esto nos sirve para identificar cuáles acciones fueron más rentables en promedio para los inversores.

```
SELECT a.simbolo, a.nombre_empresa,  
       AVG(CASE WHEN o.id_tipo = 1 THEN o.precio ELSE -o.precio  
END) AS rendimiento_promedio  
FROM Accion a  
JOIN Operacion o ON a.id_accion = o.id_accion  
GROUP BY a.id_accion;
```