

Anchoverri, Julian – 8

Araquistain, Maria Luz – 7

Arozamena, Zoe – 8

Augusto, Luciano Tomás – 9

Babbicola, Franco Matias – 7

Barbé, Alan –

Beltran Apacani, Alan Flavio – 9

Bercovsky, Juan Pablo – 6

Brunini, Axel – 6

Danilovich, Camila Belen – 9

Defederico, Benjamin Luciano – 7

Delgado, Sebastian Gabriel – 1

Desimone, Oscar – 9

Dreon, Giuliano –

Etchepare, Yanella Ileana –

Ferreyra, Nicolás Benjamin – 7

Fueyo Vuillermoz, Tiago – 4

Gallego, Juan Pablo – 6

Garcia, Maximo Manuel – 4

Gonzalez Puleo, Valentin – 4

Jimenez, Valentina Silvia – 6

Kornhauser, Luna Catalina –

Marcos Pomponio, Victoriano –

Marzoni, Juan Francisco – 9

Mastromarino, Lucio – 6

Paez Spognardi, Joaquin Emanuel – 9

Paez, German – 9

Pascuan, Joel – 7

Provoost Stosich, Aaron – 6

Rehbani, Elián Ezequiel – 9

Rodriguez, Nahuel – 9

Roldan, Lautaro Nicolas – 7

Roselli, Tomas – 7

Schot, Ciro – 8

Silvestre, Geronimo – 9

Valotto, Ana – 6

Zubillaga, Daiana Denise – 6

1 Encontrar el cliente con el depósito más alto en cada ciudad

```
SELECT c.IdCliente, c.Nombre, c.Ciudad, d.Monto
FROM clientes c
WHERE d.Monto = (
    SELECT MAX(d2.Monto)
    FROM depositos d2
    WHERE d2.IdCliente = c.IdCliente
);
```

2 Obtener todas las cuentas que NO tienen movimientos

```
SELECT c.IdCuenta, c.NumeroCuenta
FROM cuentas c
WHERE c.IdCuenta NOT IN (
    SELECT m.IdCuenta
    FROM movimientos m
);
```

- 3 Identificar los clientes que nunca han realizado una extracción

```
SELECT c.IdCliente, c.Nombre
FROM clientes c
WHERE c.IdCliente NOT IN (
    SELECT m.IdCliente
    FROM movimientos m
    WHERE m.Tipo = 'Extracción'
);
```

Ejercicios de Procedimientos

- 4 Crear un procedimiento para insertar un registro en la tabla **depositos** y devolver el Id generado

```
DELIMITER //
CREATE PROCEDURE InsertarDeposito(
    IN p_IdCliente INT,
    IN p_Monto DECIMAL(10,2),
    OUT p_IdGenerado INT
)
BEGIN
    INSERT INTO depositos (IdCliente, Monto, Fecha)
    VALUES (p_IdCliente, p_Monto, NOW());

    SET p_IdGenerado = LAST_INSERT_ID();
END //
DELIMITER ;
```

- 5 Crear un procedimiento que actualice la ciudad de un cliente y también las cuentas asociadas

```
DELIMITER //
CREATE PROCEDURE ActualizarCiudadCliente(
    IN p_IdCliente INT,
    IN p_NuevaCiudad VARCHAR(100)
)
BEGIN
    UPDATE clientes
    SET Ciudad = p_NuevaCiudad
    WHERE IdCliente = p_IdCliente;

    UPDATE cuentas
    SET Ciudad = p_NuevaCiudad
    WHERE IdCliente = p_IdCliente;
END //
DELIMITER ;
```