CONSULTAS

Consultas #1

```
-- Consulta simple
SELECT * FROM Cliente WHERE Cliente.ClienteID = 1;

-- Consulta avanzada
SELECT * FROM Prestamo WHERE Prestamo.PrestamoID <> 0
AND Prestamo.Monto > 1000 AND Prestamo.Estado = "Activo";

-- LIKE
SELECT * FROM Cliente
WHERE LOWER(Cliente.Nombre) LIKE '_o%';

-- BETWEEN
SELECT * FROM Prestamo WHERE Prestamo.PrestamoID IS NOT NULL
AND Prestamo.Saldo BETWEEN 1000 AND 3000;

-- ORDER BY
SELECT *
FROM loan
ORDER BY start_date ASC;
```

Consultas #2

```
-- 1. CONCAT
SELECT CONCAT(first_name, ' ', last_name) AS full_name
FROM person;

-- 2. UPPER
SELECT UPPER(name) AS business_name
FROM business;

-- 3. LOWER
SELECT LOWER(mail) AS email_lowercase
FROM person;

-- 4. SUBSTRING
SELECT SUBSTRING(document, 1, 4) AS partial_document
FROM person;

-- 5. LENGTH
SELECT first_name, LENGTH(first_name) AS name_length
FROM person;

-- 6. ROUND
SELECT amount, ROUND(amount, 0) AS rounded_amount
FROM loan;

-- 7. ABS
```

```
SELECT loan_id, ABS(estimated_value) AS absolute_value
FROM guarantee;

-- 8. COUNT
SELECT COUNT(*) AS total_loans
FROM loan;

-- 9. SUM
SELECT SUM(amount) AS total_amount
FROM loan;

-- 10. AVG
SELECT AVG(amount) AS average_loan_amount
FROM loan;

-- 11. MIN
SELECT MIN(amount) AS minimum_loan_amount
FROM loan;

-- 12. MAX
SELECT MAX(amount) AS maximum_loan_amount
FROM loan;

-- 13. GROUP BY
SELECT status, COUNT(*) AS count_loans
```

```
FROM loan
GROUP BY status;

-- 14. GROUP BY con HAVING
SELECT client_id, COUNT(*) AS loan_count
FROM loan
GROUP BY client_id
HAVING COUNT(*) > 1;
```

Consultas #3

```
-- 1. Obtener los nombres de los clientes y sus direcciones
SELECT p.first_name, p.last_name, a.description
FROM person p
INNER JOIN address a ON p.address_id = a.id;

-- 2. Obtener los préstamos junto con los nombres de los clientes
SELECT l.amount, p.first_name, p.last_name
FROM loan l
INNER JOIN person p ON l.client_id = p.id;

-- 3. Obtener los pagos y la información del préstamo
SELECT pay.amount, l.amount AS loan_amount
FROM payment pay
INNER JOIN loan l ON pay.loan_id = l.id;

-- 4. Obtener los negocios y sus direcciones
SELECT b.name, a.description
```

```
FROM business b
INNER JOIN address a ON b.address_id = a.id;

-- 5. Obtener las garantías y la información del préstamo correspondiente
SELECT g.type, g.description, l.amount
FROM guarantee g
INNER JOIN loan l ON g.loan_id = l.id;

-- 6. Obtener los validadores de pagos y su información
SELECT pv.comment, p.first_name, p.last_name
FROM payment_validation pv
INNER JOIN person p ON pv.administrator_id = p.id;

-- 7. Obtener los cobradores y los pagos que han manejado
SELECT p.first_name, p.last_name, pay.amount
FROM payment_collector pc
INNER JOIN payment pay ON pc.payment_id = pay.id
INNER JOIN person p ON pc.collector_id = p.id;

-- 8. Obtener información de los préstamos con sus garantes
SELECT l.amount, p.first_name AS client_name, pg.first_name AS guarantor_name
FROM loan l
INNER JOIN person p ON l.client_id = p.id
LEFT JOIN person pg ON l.guarantor_id = pg.id;

-- 9. Obtener los pagos de los préstamos que están pendientes
SELECT pay.amount, l.status
FROM payment pay
INNER JOIN loan l ON pay.loan_id = l.id
```

```
WHERE pay.status = 'Pending';


-- 10. Obtener el total de préstamos por cliente
SELECT p.first_name, p.last_name, COUNT(l.id) AS total_loans
FROM person p
INNER JOIN loan l ON p.id = l.client_id
GROUP BY p.id;
```

Consultas #4

```
-- 1. Obtener todos los clientes y sus préstamos (incluyendo clientes sin préstamos)
SELECT p.first_name, p.last_name, l.amount
FROM person p
LEFT JOIN loan l ON p.id = l.client_id;


-- 2. Obtener todos los pagos y la información de los préstamos (incluyendo pagos sin préstamo asociado)
SELECT pay.amount, l.amount AS loan_amount
FROM payment pay
LEFT JOIN loan l ON pay.loan_id = l.id;


-- 3. Obtener todos los negocios y sus direcciones (incluyendo negocios sin dirección)
SELECT b.name, a.description
FROM business b
LEFT JOIN address a ON b.address_id = a.id;


-- 4. Obtener todos los validadores de pagos y sus comentarios (incluyendo validadores sin comentarios)
```

```
SELECT p.first_name, p.last_name, pv.comment
FROM person p
LEFT JOIN payment_validation pv ON p.id = pv.administrator_id;

-- 5. Obtener todos los cobradores y los pagos que han manejado (incluyendo cobradores sin pagos)
SELECT p.first_name, p.last_name, pay.amount
FROM payment_collector pc
LEFT JOIN payment pay ON pc.payment_id = pay.id
LEFT JOIN person p ON pc.collector_id = p.id;

-- Consultas con RIGHT JOIN

-- 6. Obtener todos los préstamos y sus garantes (incluyendo préstamos sin garantes)
SELECT l.amount, p.first_name AS client_name, pg.first_name AS guarantor_name
FROM loan l
RIGHT JOIN person pg ON l.guarantor_id = pg.id
LEFT JOIN person p ON l.client_id = p.id;

-- 7. Obtener todas las garantías y la información de los préstamos (incluyendo garantías sin préstamos)
SELECT g.type, g.description, l.amount
FROM guarantee g
RIGHT JOIN loan l ON g.loan_id = l.id;

-- 8. Obtener todos los pagos y sus validaciones (incluyendo validaciones sin pagos)
SELECT pay.amount, pv.comment
FROM payment pay
RIGHT JOIN payment_validation pv ON pay.id = pv.payment_id;
```

-- Consulta con CROSS JOIN

-- 9. Obtener todas las combinaciones de personas y negocios

```
SELECT p.first_name, b.name
FROM person p
CROSS JOIN business b;
```

-- 10. Obtener todas las combinaciones de préstamos y garantías

```
SELECT l.amount AS loan_amount, g.type
FROM loan l
CROSS JOIN guarantee g;
```

Consultas #5

-- IN

-- 1. Encontrar todos los clientes que tienen préstamos asociados.

```
SELECT first_name, last_name
FROM person
WHERE id IN (SELECT client_id FROM loan);
```

-- 2. Mostrar los cobradores que tienen alguna ruta asignada.

```
SELECT first_name, last_name
FROM person
WHERE id IN (SELECT collector_id FROM route);
```

-- 3. Mostrar los clientes cuyo préstamo ha sido aprobado.

```
SELECT first_name, last_name

FROM person

WHERE id IN (SELECT client_id FROM loan WHERE status_id IN (SELECT id
FROM status WHERE status = 'Aprobado'));


-- 4. Encontrar las garantías relacionadas con préstamos cuyo cliente sea "María
Gómez".

SELECT description, estimated_value

FROM guarantee

WHERE loan_id IN (SELECT id FROM loan WHERE client_id IN (SELECT id
FROM person WHERE first_name = 'María' AND last_name = 'Gómez'));


-- 5. Listar las notificaciones enviadas a usuarios que son administradores.

SELECT message, type_id

FROM notification

WHERE user_id IN (SELECT id FROM user WHERE id IN (SELECT user_id
FROM user_role WHERE role_id = (SELECT id FROM role WHERE role =
'Administrador')));


-- ANY


-- 6. Listar los clientes cuyo teléfono coincide con algún teléfono registrado en la base
de datos.

SELECT first_name, last_name

FROM person

WHERE phone = ANY (SELECT phone FROM person);


-- 7. Encontrar los préstamos cuyo monto sea mayor que el monto de algún préstamo
otorgado a Carlos López.

SELECT amount

FROM loan
```

WHERE amount > ANY (SELECT amount FROM loan WHERE client_id = (SELECT id FROM person WHERE first_name = 'Carlos' AND last_name = 'López'));

-- 8. Encontrar los préstamos que tienen una tasa de interés menor que cualquier préstamo de "Juan Pérez".

SELECT amount, interest

FROM loan

WHERE interest < ANY (SELECT interest FROM loan WHERE client_id = (SELECT id FROM person WHERE first_name = 'Juan' AND last_name = 'Pérez'));

-- 9. Mostrar todos los préstamos cuyo monto es mayor que cualquier garantía registrada.

SELECT amount

FROM loan

WHERE amount > ANY (SELECT estimated_value FROM guarantee);

-- 10. Encontrar los usuarios cuyo salario es menor que el de algún administrador.

SELECT first_name, last_name

FROM person

WHERE salary < ANY (SELECT salary FROM person WHERE id IN (SELECT person_id FROM user_role WHERE role_id = (SELECT id FROM role WHERE role = 'Administrador')));

-- ALL

-- 11. Listar los préstamos que tienen un monto mayor que todas las garantías registradas.

SELECT amount

FROM loan

WHERE amount > ALL (SELECT estimated_value FROM guarantee);

-- 12. Mostrar los préstamos que tienen una tasa de interés mayor que todos los préstamos de "María Gómez".

SELECT amount, interest

FROM loan

WHERE interest > ALL (SELECT interest FROM loan WHERE client_id = (SELECT id FROM person WHERE first_name = 'María' AND last_name = 'Gómez'));

-- 13. Encontrar todos los cobradores que ganan más que todos los administradores.

SELECT first_name, last_name

FROM person

WHERE salary > ALL (SELECT salary FROM person WHERE id IN (SELECT person_id FROM user_role WHERE role_id = (SELECT id FROM role WHERE role = 'Administrador')))

AND id IN (SELECT person_id FROM user_role WHERE role_id = (SELECT id FROM role WHERE role = 'Cobrador'));

-- 14. Encontrar los clientes cuyo teléfono es diferente de todos los teléfonos de los fiadores.

SELECT first_name, last_name

FROM person

WHERE phone <> ALL (SELECT phone FROM person WHERE id IN (SELECT person_id FROM user_role WHERE role_id = (SELECT id FROM role WHERE role = 'Fiador')));

-- 15. Mostrar los clientes cuyo préstamo es mayor que todos los préstamos de "Carlos López".

SELECT first_name, last_name

FROM person

WHERE id IN (SELECT client_id FROM loan WHERE amount > ALL (SELECT amount FROM loan WHERE client_id = (SELECT id FROM person WHERE first_name = 'Carlos' AND last_name = 'López')));

-- EXISTS

-- 16. Mostrar los clientes que tienen al menos un préstamo.

SELECT first_name, last_name

FROM person

WHERE EXISTS (SELECT 1 FROM loan WHERE client_id = person.id);

-- 17. Listar todos los cobradores que tienen rutas asignadas.

SELECT first_name, last_name

FROM person

WHERE EXISTS (SELECT 1 FROM route WHERE collector_id = person.id);

-- 18. Encontrar los administradores que validaron algún pago.

SELECT first_name, last_name

FROM person

WHERE EXISTS (SELECT 1 FROM payment_validation WHERE administrator_id = person.id);

-- 19. Mostrar todos los préstamos que tienen pagos asociados.

SELECT amount, start_date, end_date

FROM loan

WHERE EXISTS (SELECT 1 FROM payment WHERE loan_id = loan.id);

-- 20. Listar los clientes que han recibido notificaciones.

SELECT first_name, last_name

FROM person

WHERE EXISTS (SELECT 1 FROM notification WHERE user_id = (SELECT id FROM user WHERE person_id = person.id));