

Hito Individual

Bases de datos

Indice

1. Propuesta comercial	1
2. Diseño conceptual y lógico	3
Justificación de entidades y atributos	4
3. Diseño físico	5
Creación de tablas, objetos y métodos	5
Introducción de datos de prueba (insert into)	8
Validación de métodos	11
Consultas (selects)	15
4. Plan de pruebas	25
Integridad de los datos	26
Prueba de registro de un pedido completo	26
Insert into fallidos	27
Cambiar detalles en tabla Recogidas (update)	27
5. Evaluación crítica	28

1. Propuesta comercial

Cine Arte Atrezzo es una empresa que se dedica al **alquiler de objetos** que aparecen en escenas de cine o teatro. Los clientes habituales son compañías de teatro o productoras cinematográficas. Esta empresa necesita una aplicación para gestionar su negocio en todo lo relacionado con la gestión de clientes y alquileres de atrezzo.

Tras analizar los requisitos del usuario, se presenta una propuesta de diseño de base de datos objeto-relacional. Esta propuesta tiene como objetivo diseñar, desarrollar e implementar una aplicación web para Cine Arte Atrezzo, una empresa dedicada al alquiler de objetos utilizados en escenas de cine y teatro. La aplicación web proporcionará una plataforma eficiente para gestionar clientes, productos disponibles, pedidos, entregas, devoluciones y facturación. Además, se creará una base de datos objeto-relacional que respaldará todas las funcionalidades requeridas.

Esta propuesta comercial busca optimizar la gestión de los procesos internos y facilitar el seguimiento de las transacciones comerciales de esta empresa. El objetivo principal es brindar una solución eficiente y robusta para gestionar la información relacionada con clientes, productos, pedidos, alquileres, facturas, entregas y recogidas.

A continuación, se detallan los **requisitos de usuario**:

1. Registro de pedidos: La base de datos permite registrar los pedidos realizados por los clientes, incluyendo todos los datos pertinentes de los mismos.
2. Catálogo de Productos: La aplicación permitirá a los usuarios consultar los productos disponibles para alquiler en tiempo real. Se incluirá información detallada de cada producto, como descripción y disponibilidad actual.
3. Realización de Pedidos: Los clientes podrán seleccionar los productos deseados y realizar pedidos a través de la aplicación web. Se solicitará al cliente que especifique la fecha de devolución de los productos alquilados.
4. Órdenes de Entrega y Recogida: Al finalizar un pedido, se generará automáticamente una orden de entrega para el transporte de Cine Arte Atrezzo. Asimismo, se creará una orden de recogida para la fecha de devolución especificada por el cliente.

5. Facturación y Pago: Se generará una factura detallada para cada pedido, la cual se entregará al cliente junto con los productos. El pago de la factura se realizará a través del número de cuenta bancaria registrada previamente por el cliente.
6. Seguimiento de Pedidos: La aplicación permitirá a los usuarios consultar en cualquier momento el estado de un pedido, indicando si está pendiente, entregado o devuelto.

El diseño planteado es acorde con los **requisitos del usuario** puesto que se cumplen las siguientes tareas:

- La base de datos propuesta permite gestionar información de los clientes y almacenar sus datos.
- Permite almacenar información de los productos disponibles para alquiler.
- Permite registrar los pedidos realizados por los clientes, incluyendo todos los datos pertinentes de los mismos.
- Genera facturas para los pedidos, incluyendo el pedido asociado, la fecha de la factura y el importe.
- Registrar las entregas y recogidas de los pedidos, incluyendo el pedido asociado, la fecha, la matrícula del transporte y los detalles.

Se ha elaborado un **diseño conceptual y lógico** para la base de datos que abarca las entidades y relaciones clave necesarias para satisfacer las necesidades de este negocio. Las entidades propuestas para gestionar la información solicitada son:

1. Clientes
2. Productos
3. Pedidos
4. Alquileres
5. Facturas
6. Entregas
7. Recogidas

Estas entidades constituyen tablas dentro de la base de datos. En el siguiente apartado se explicarán las relaciones y asociaciones entre cada una de ellas.

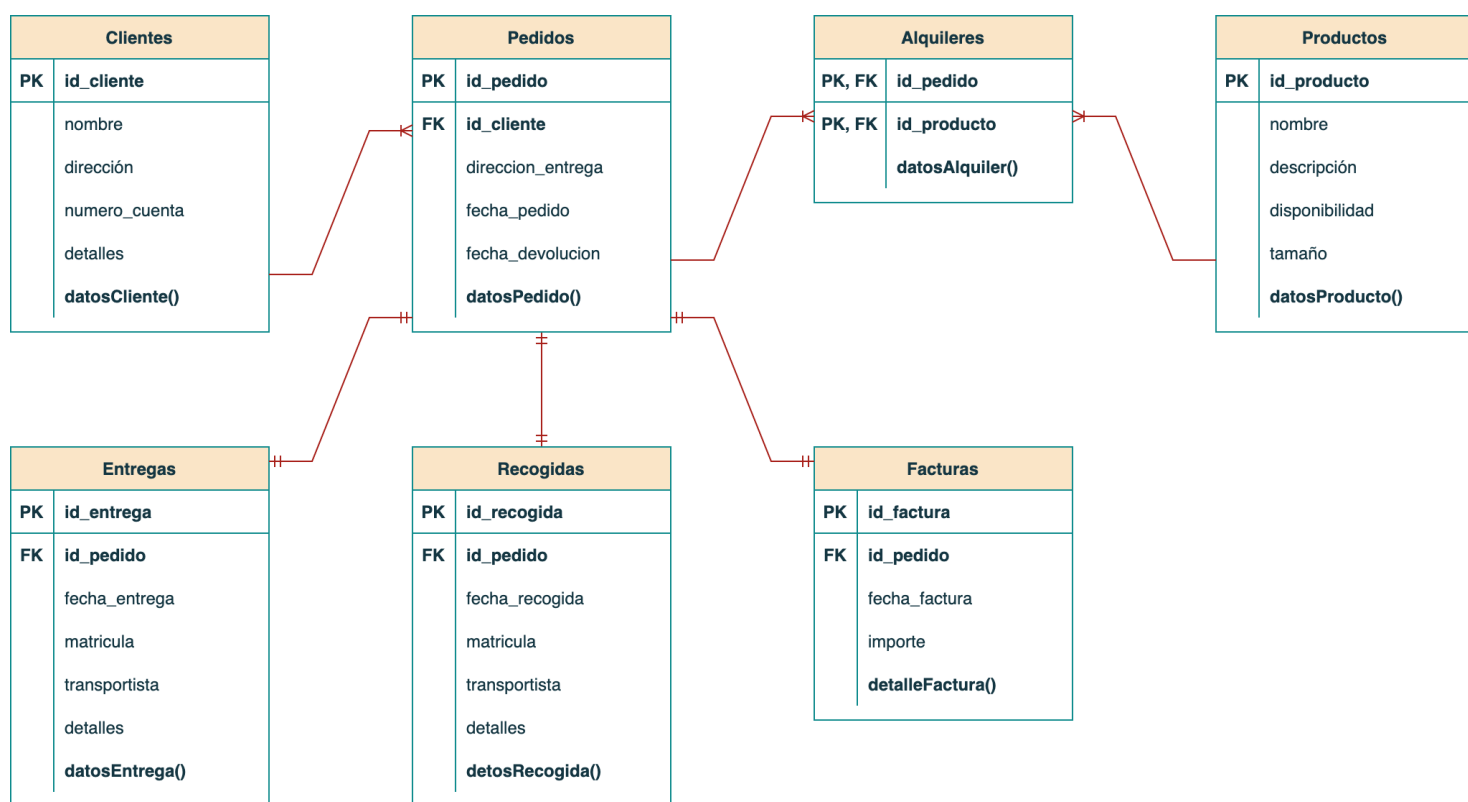
Las asociaciones entre estas entidades son:

- 1 Clientes N Pedidos
- N Pedidos M Productos (N a M) -> Se crea Entidad Asociativa Alquileres
- 1 Pedido N Alquileres
- 1 Producto N Alquileres
- 1 Entregas 1 Pedidos
- 1 Recogidas 1 Pedidos
- 1 Facturas 1 Pedidos

2. Diseño conceptual y lógico

En este apartado, se presenta una **representación visual** del diseño conceptual y lógico propuesto para la base de datos. El diseño conceptual se enfoca en identificar las entidades principales y las relaciones entre ellas, brindando una visión global de la estructura y funcionalidad del sistema.

El diseño lógico se centra en la **traducción** del diseño conceptual a un modelo relacional, definiendo las tablas, atributos y restricciones específicas.



La representación gráfica del diseño conceptual y lógico proporciona una **visión clara** de la estructura de la base de datos, permitiendo comprender rápidamente cómo se organizan los datos y cómo interactúan entre sí. Además, sirve como punto de referencia para el diseño físico posterior, donde se implementará la base de datos en un sistema de gestión de bases de datos relacional.

Justificación de entidades y atributos

- **Cliente**: Representa la información de los clientes, incluyendo nombre, dirección, número de cuenta y detalles adicionales. Se podría entender el ID de esta tabla cliente como el NIF de la empresa cliente.
- **Producto**: Almacena la información de los productos disponibles para alquiler. Se almacena el nombre, su descripción, disponibilidad y tamaño.
- **Pedido**: Registra los pedidos realizados por los clientes, incluyendo el cliente asociado que ha realizado el pedido, la dirección de entrega, la fecha de pedido y la fecha de devolución. La fecha de devolución es la fecha en la que el cliente solicita devolver el producto, esta fecha será diferente de la fecha de recogida, que será la fecha en la que se lleve a cabo la recogida efectiva.
Por ejemplo, un cliente solicita la devolución de un producto que ha alquilado. Esta fecha se registra como 3 de abril, sería la fecha de devolución. Sin embargo, la recogida de dicho producto no se realiza hasta el día siguiente, el 4 de abril. Sería esta última la fecha de recogida.
- **Alquiler**: Vincula los pedidos y los productos alquilados, estableciendo la relación entre ellos mediante *foreign keys*.
- **Factura**: Genera facturas para los pedidos, incluyendo el ID de pedido asociado mediante *foreign keys*, la fecha de la factura y el importe total.
- **Entrega**: Registra las entregas de los pedidos, incluyendo el pedido asociado, la fecha de entrega, la matrícula del transporte, el transportista que realiza la entrega y los detalles de la entrega.
- **Recogida**: Registra las recogidas de los pedidos, incluyendo el pedido asociado, la fecha de recogida, la matrícula del transporte, el transportista que realiza la recogida y los detalles de la recogida.

3. Diseño físico

En esta etapa del proceso de diseño de la base de datos objeto-relacional para Cine Arte Atrezzo, nos centraremos en el diseño físico de la base de datos utilizando el lenguaje de consulta estructurado (SQL) de Oracle.

El diseño físico se refiere a la implementación concreta de la estructura de la base de datos, incluyendo la definición de tablas, objetos y métodos. El objetivo principal de esta fase es traducir el diseño conceptual y lógico en un conjunto de instrucciones SQL que permitan crear y manipular la base de datos en un entorno de producción.

Se presentarán las sentencias SQL necesarias para crear las tablas, definir los campos y establecer las relaciones entre ellas, así como también cargar datos de prueba para verificar el correcto funcionamiento de la base de datos.

Creación de tablas, objetos y métodos

```
-- 1. Clientes
-- Definición de objeto para la tabla Clientes
CREATE TYPE Cliente_obj AS OBJECT (
    id_cliente INT,
    nombre VARCHAR2(100),
    direccion VARCHAR2(200),
    numero_cuenta VARCHAR2(50),
    detalles VARCHAR2(200),

    -- Método para obtener los datos del cliente
    MEMBER FUNCTION datos_cliente RETURN VARCHAR2
);

-- Tabla Clientes basada en el objeto Cliente
CREATE TABLE Clientes OF Cliente_obj (
    PRIMARY KEY (id_cliente)
);

-- 2. Productos
-- Definición de objeto para la tabla Productos
CREATE TYPE Producto_obj AS OBJECT (
    id_producto INT,
    nombre VARCHAR2(100),
    descripcion VARCHAR2(200),
    disponibilidad CHAR(1),
    tamaño INT,
```

```

    -- Método para obtener los datos del producto
    MEMBER FUNCTION datos_producto RETURN VARCHAR2
);

-- Tabla Productos basada en el objeto Producto
CREATE TABLE Productos OF Producto_obj (
    PRIMARY KEY (id_producto)
);

-- 3. Pedidos
-- Definición de objeto Pedido
CREATE OR REPLACE TYPE Pedido_obj AS OBJECT (
    id_pedido INT,
    cliente REF Cliente_obj,
    direccion_entrega VARCHAR2(200),
    fecha_pedido DATE,
    fecha_devolucion DATE,

    -- Método para obtener los datos del pedido
    MEMBER FUNCTION datos_pedido RETURN VARCHAR2
);

-- Tabla pedidos
CREATE TABLE Pedidos OF Pedido_obj (
    PRIMARY KEY (id_pedido),
    FOREIGN KEY (cliente) REFERENCES Clientes
);

-- 4. Alquileres
-- Definición de objeto Alquiler
CREATE TYPE Alquiler_obj AS OBJECT (
    pedido REF Pedido_obj,
    producto REF Producto_obj,

    -- Método para obtener los datos del alquiler
    MEMBER FUNCTION datos_alquiler RETURN VARCHAR2
);

-- Tabla Alquileres
CREATE TABLE Alquileres OF Alquiler_obj (
    FOREIGN KEY (pedido) REFERENCES Pedidos,
    FOREIGN KEY (producto) REFERENCES Productos
);

-- 5. Facturas
-- Objeto Facturas
CREATE TYPE Factura_obj AS OBJECT (
    id_factura INT,
    pedido REF Pedido_obj,
    fecha_factura DATE,
    importe FLOAT,

```



```

    -- Método para obtener los datos de la factura
    MEMBER FUNCTION detalle_factura RETURN VARCHAR2
);

-- Tabla Facturas
CREATE TABLE Facturas OF Factura_obj (
    PRIMARY KEY (id_factura),
    FOREIGN KEY (pedido) REFERENCES Pedidos
);

-- 6. Entregas
-- Objeto Entregas
CREATE TYPE Entrega_obj AS OBJECT (
    id_entrega INT,
    pedido REF Pedido_obj,
    fecha_entrega DATE,
    matricula VARCHAR2(100),
    transportista VARCHAR2(100),
    detalles VARCHAR2(200),

    -- Método para obtener los datos de la entrega
    MEMBER FUNCTION datos_entrega RETURN VARCHAR2
);

-- Tabla Entregas
CREATE TABLE Entregas OF Entrega_obj (
    PRIMARY KEY (id_entrega),
    FOREIGN KEY (pedido) REFERENCES Pedidos
);

-- 7. Recogidas
-- Objeto Recogidas
CREATE TYPE Recogida_obj AS OBJECT (
    id_recogida INT,
    pedido REF Pedido_obj,
    fecha_recogida DATE,
    matricula VARCHAR2(100),
    transportista VARCHAR2(100),
    detalles VARCHAR2(200),

    -- Método para obtener los datos de la recogida
    MEMBER FUNCTION datos_recogida RETURN VARCHAR2
);

-- Tabla Recogidas
CREATE TABLE Recogidas OF Recogida_obj (
    PRIMARY KEY (id_recogida),
    FOREIGN KEY (pedido) REFERENCES Pedidos
);

```

En las sentencias anteriormente aportadas, se está utilizando el lenguaje de programación PL/SQL de Oracle para definir objetos y tablas objeto.

Por ejemplo, en el primero bloque de código relacionado con "Clientes", se define un **objeto** llamado "Cliente_obj", que contiene los atributos necesarios para representar un cliente, como su ID, nombre, dirección, número de cuenta y detalles adicionales.

Además, se incluye un método llamado "datos_cliente" que permite obtener una representación en formato de cadena de los datos del cliente. Luego se crea una **tabla** llamada "Clientes" basada en el objeto "Cliente_obj", donde se especifica que la clave primaria es el ID del cliente.

Se sigue este mismo proceso con las 7 entidades planteadas para esta base de datos, que constituyen 7 objetos y a su vez 7 tablas objeto.

En resumen, estas sentencias definen objetos y tablas objeto en Oracle, permitiendo la creación de **estructuras personalizadas** y **métodos asociados** a esos objetos para un manejo más eficiente y flexible de los datos en la base de datos. Los objetos y tablas objeto permiten una mayor abstracción y encapsulación de los datos, facilitando la implementación de funcionalidades específicas y mejorando la organización y mantenibilidad del código.

Introducción de datos de prueba (INSERT INTO)

-- 1. Datos de prueba para Clientes

```
INSERT INTO Clientes VALUES (1, 'Estudios Pantalla Plateada', 'Calle Principal 123', '1234567890', 'Este cliente trabaja en televisión.');
```

```
INSERT INTO Clientes VALUES (2, 'Atrezzo Estelar', 'Calle del Olmo 45', '0987654321', 'Este cliente necesita eficacia y rapidez.');
```

```
INSERT INTO Clientes VALUES (3, 'Producciones CineMágico', 'Avenida del Roble 78', '9876543210', 'Este cliente busca atrezzo original y creativo.');
```

```
INSERT INTO Clientes VALUES (4, 'Films de Fantasía S.A.', 'Calle del Pino 31', '5678901234', 'Este cliente necesita decorados y atrezzo realistas.');
```

```
INSERT INTO Clientes VALUES (5, 'Soluciones VestuarioCine', 'Calle del Cedro 87', '4321098765', 'Este cliente requiere un amplio catálogo de opciones de vestuario para cine y televisión.');
```

-- 2. Datos de prueba para Productos

```
INSERT INTO Productos VALUES (1, 'Sillón Chesterfield', 'Un elegante sillón de cuero de estilo clásico, perfecto para ambientar escenas de época.', 'S', 10);
INSERT INTO Productos VALUES (2, 'Maleta Vintage', 'Una maleta antigua de aspecto retro, ideal para viajes en el tiempo o escenarios de época.', 'N', 5);
INSERT INTO Productos VALUES (3, 'Teléfono de Dial', 'Un teléfono de disco auténtico de los años 70, ideal para ambientar escenas de nostalgia o detectivescas.', 'S', 8);
INSERT INTO Productos VALUES (4, 'Reloj de Péndulo', 'Un reloj de péndulo clásico que añade un toque de elegancia y sofisticación a cualquier decorado.', 'S', 12);
INSERT INTO Productos VALUES (5, 'Candelabro Gótico', 'Un candelabro imponente con diseño gótico, perfecto para escenas de misterio y suspense.', 'N', 7);
```

-- 3. Datos de prueba para Pedidos

```
INSERT INTO Pedidos VALUES (1, (SELECT REF(c) FROM Clientes c WHERE c.id_cliente = 1), 'Calle Principal 123', SYSDATE, SYSDATE+7);
INSERT INTO Pedidos VALUES (2, (SELECT REF(c) FROM Clientes c WHERE c.id_cliente = 2), 'Calle del Olmo 456', SYSDATE, SYSDATE+5);
INSERT INTO Pedidos VALUES (3, (SELECT REF(c) FROM Clientes c WHERE c.id_cliente = 3), 'Avenida del Roble 789', SYSDATE, SYSDATE+3);
INSERT INTO Pedidos VALUES (4, (SELECT REF(c) FROM Clientes c WHERE c.id_cliente = 4), 'Calle del Pino 321', SYSDATE, SYSDATE+2);
INSERT INTO Pedidos VALUES (5, (SELECT REF(c) FROM Clientes c WHERE c.id_cliente = 5), 'Calle del Cedro 987', SYSDATE, SYSDATE+10);
INSERT INTO Pedidos VALUES (6, (SELECT REF(c) FROM Clientes c WHERE c.id_cliente = 5), 'Calle del Cubo 987', SYSDATE, SYSDATE+10);
INSERT INTO Pedidos VALUES (7, (SELECT REF(c) FROM Clientes c WHERE c.id_cliente = 2), 'Calle del Alce 987', SYSDATE, SYSDATE+10);
```

-- 4. Datos de prueba para Alquileres

```
INSERT INTO Alquileres VALUES ((SELECT REF(p) FROM Pedidos p WHERE p.id_pedido = 1), (SELECT REF(pr) FROM Productos pr WHERE pr.id_producto = 1));
INSERT INTO Alquileres VALUES ((SELECT REF(p) FROM Pedidos p WHERE p.id_pedido = 2), (SELECT REF(pr) FROM Productos pr WHERE pr.id_producto = 2));
INSERT INTO Alquileres VALUES ((SELECT REF(p) FROM Pedidos p WHERE p.id_pedido = 3), (SELECT REF(pr) FROM Productos pr WHERE pr.id_producto = 3));
INSERT INTO Alquileres VALUES ((SELECT REF(p) FROM Pedidos p WHERE p.id_pedido = 4), (SELECT REF(pr) FROM Productos pr WHERE pr.id_producto = 4));
INSERT INTO Alquileres VALUES ((SELECT REF(p) FROM Pedidos p WHERE p.id_pedido = 5), (SELECT REF(pr) FROM Productos pr WHERE pr.id_producto = 5));
INSERT INTO Alquileres VALUES ((SELECT REF(p) FROM Pedidos p WHERE p.id_pedido = 6), (SELECT REF(pr) FROM Productos pr WHERE pr.id_producto = 2));
```

```
INSERT INTO Alquileres VALUES ((SELECT REF(p) FROM Pedidos p WHERE
p.id_pedido = 7), (SELECT REF(pr) FROM Productos pr WHERE pr.id_producto =
1));
```

-- 5. Datos de prueba para Facturas

```
INSERT INTO Facturas (id_factura, pedido, fecha_factura, importe)
VALUES (1, (SELECT REF(p) FROM Pedidos p WHERE p.id_pedido = 1), SYSDATE,
100.00);
INSERT INTO Facturas (id_factura, pedido, fecha_factura, importe)
VALUES (2, (SELECT REF(p) FROM Pedidos p WHERE p.id_pedido = 2), SYSDATE,
150.50);
INSERT INTO Facturas (id_factura, pedido, fecha_factura, importe)
VALUES (3, (SELECT REF(p) FROM Pedidos p WHERE p.id_pedido = 3), SYSDATE,
200.00);
INSERT INTO Facturas (id_factura, pedido, fecha_factura, importe)
VALUES (4, (SELECT REF(p) FROM Pedidos p WHERE p.id_pedido = 4), SYSDATE,
120.75);
INSERT INTO Facturas (id_factura, pedido, fecha_factura, importe)
VALUES (5, (SELECT REF(p) FROM Pedidos p WHERE p.id_pedido = 5), SYSDATE,
180.25);
```

-- 6. Datos de prueba para Entregas

```
INSERT INTO Entregas (id_entrega, pedido, fecha_entrega, matricula,
transportista, detalles)
VALUES (1, (SELECT REF(p) FROM Pedidos p WHERE p.id_pedido = 1), SYSDATE +
1, 'ABC123', 'Transportista 1', 'Entrega programada para el set de
filmación "Aventuras en la Selva".');
```

```
INSERT INTO Entregas (id_entrega, pedido, fecha_entrega, matricula,
transportista, detalles)
VALUES (2, (SELECT REF(p) FROM Pedidos p WHERE p.id_pedido = 2), SYSDATE +
2, 'DEF456', 'Transportista 2', 'Entrega urgente para el rodaje de la
película de acción "Explosión Final".');
```

```
INSERT INTO Entregas (id_entrega, pedido, fecha_entrega, matricula,
transportista, detalles)
VALUES (3, (SELECT REF(p) FROM Pedidos p WHERE p.id_pedido = 3), SYSDATE +
3, 'GHI789', 'Transportista 3', 'Entrega especial de atrezzo para el
escenario de la obra de teatro "Sueños de Shakespeare".');
```

```
INSERT INTO Entregas (id_entrega, pedido, fecha_entrega, matricula,
transportista, detalles)
VALUES (4, (SELECT REF(p) FROM Pedidos p WHERE p.id_pedido = 4), SYSDATE +
4, 'JKL012', 'Transportista 4', 'Entrega programada para el set de
grabación de la serie de televisión "Misterios Sin Resolver".');
```

```
INSERT INTO Entregas (id_entrega, pedido, fecha_entrega, matricula,
transportista, detalles)
VALUES (5, (SELECT REF(p) FROM Pedidos p WHERE p.id_pedido = 5), SYSDATE +
5, 'MN0345', 'Transportista 5', 'Entrega de vestuario para el rodaje de la
película histórica "El Tiempo Dorado".');
```

-- 7. Datos de prueba para Recogidas

```
INSERT INTO Recogidas (id_recogida, pedido, fecha_recogida, matricula,
transportista, detalles)
VALUES (1, (SELECT REF(p) FROM Pedidos p WHERE p.id_pedido = 1), SYSDATE +
6, 'PQR678', 'Transportista 6', 'Recogida en la dirección principal');
INSERT INTO Recogidas (id_recogida, pedido, fecha_recogida, matricula,
transportista, detalles)
VALUES (2, (SELECT REF(p) FROM Pedidos p WHERE p.id_pedido = 2), SYSDATE +
7, 'STU901', 'Transportista 7', 'Recogida en la dirección de la
sucursal');
INSERT INTO Recogidas (id_recogida, pedido, fecha_recogida, matricula,
transportista, detalles)
VALUES (3, (SELECT REF(p) FROM Pedidos p WHERE p.id_pedido = 3), SYSDATE +
8, 'VWX234', 'Transportista 8', 'Recogida en la dirección residencial');
INSERT INTO Recogidas (id_recogida, pedido, fecha_recogida, matricula,
transportista, detalles)
VALUES (4, (SELECT REF(p) FROM Pedidos p WHERE p.id_pedido = 4), SYSDATE +
9, 'YZA567', 'Transportista 9', 'Recogida en la dirección comercial');
INSERT INTO Recogidas (id_recogida, pedido, fecha_recogida, matricula,
transportista, detalles)
VALUES (5, (SELECT REF(p) FROM Pedidos p WHERE p.id_pedido = 5), SYSDATE +
10, 'BCD890', 'Transportista 10', 'Recogida en la dirección de entrega
alternativa');
```

Se ha realizado una cantidad adecuada de insert into para garantizar una cantidad suficiente de datos de prueba y poder evaluar y comprobar la eficiencia del sistema.

Validación de métodos

A continuación, procedemos a validar e implementar los métodos. Un método MEMBER FUNCTION, en el contexto de la programación orientada a objetos, es una función o procedimiento que se define dentro del cuerpo de una clase o un tipo de objeto. Este tipo de método está asociado directamente con los objetos de esa clase y puede acceder a los datos y atributos específicos de cada instancia.

-- Implementación método para obtener los datos del cliente

```
CREATE OR REPLACE TYPE BODY Cliente_obj AS
  MEMBER FUNCTION datos_cliente RETURN VARCHAR2 IS
  BEGIN
    RETURN 'Cliente: Nombre: ' || nombre || ' / Dirección: ' || direccion
    || ' / Número de cuenta: ' || numero_cuenta;
  END;
END;
```

-- Ejemplo de uso del método en una consulta

```
SELECT c.datos_cliente() AS datos
FROM Clientes c;
```

DATOS
1 Cliente: Nombre: Estudios Pantalla Plateada / Dirección: Calle Principal 123/ Número de cuenta: 1234567890
2 Cliente: Nombre: Atrezzo Estelar / Dirección: Calle del Olmo 45/ Número de cuenta: 0987654321
3 Cliente: Nombre: Producciones CineMágico / Dirección: Avenida del Roble 78/ Número de cuenta: 9876543210
4 Cliente: Nombre: Films de Fantasía S.A. / Dirección: Calle del Pino 31/ Número de cuenta: 5678901234
5 Cliente: Nombre: Soluciones VestuarioCine / Dirección: Calle del Cedro 87/ Número de cuenta: 4321098765

-- Implementación método para obtener los datos del producto

```
CREATE OR REPLACE TYPE BODY Producto_obj AS
  MEMBER FUNCTION datos_producto RETURN VARCHAR2 IS
  BEGIN
    RETURN 'Producto: Nombre: ' || nombre || ' / Descripción: ' ||
descripcion || ' / Disponibilidad: ' || disponibilidad ;
  END;
END;
```

-- Ejemplo de uso del método en una consulta

```
SELECT p.datos_producto() AS datos
FROM Productos p;
```

DATOS
1 Producto: Nombre: Sillón Chesterfield / Descripción: Un elegante sillón de cuero de estilo clásico, perfecto para ambientar escenas de época. / Disponibilidad: S
2 Producto: Nombre: Maleta Vintage / Descripción: Una maleta antigua de aspecto retro, ideal para viajes en el tiempo o escenarios de época. / Disponibilidad: N
3 Producto: Nombre: Teléfono de Dial / Descripción: Un teléfono de disco auténtico de los años 70, ideal para ambientar escenas de nostalgia o detectivescas. / Disponibilidad: S
4 Producto: Nombre: Reloj de Péndulo / Descripción: Un reloj de péndulo clásico que añade un toque de elegancia y sofisticación a cualquier decorado. / Disponibilidad: S
5 Producto: Nombre: Candelabro Gótico / Descripción: Un candelabro imponente con diseño gótico, perfecto para escenas de misterio y suspense. / Disponibilidad: N

-- Implementación del método para obtener los datos del pedido

```
CREATE OR REPLACE TYPE BODY Pedido_obj AS
  MEMBER FUNCTION datos_pedido RETURN VARCHAR2 IS
  v_cliente Cliente_obj;
  BEGIN
    SELECT Deref(cliente) INTO v_cliente FROM dual;
    RETURN 'Pedido, Cliente: ' || v_cliente.datos_cliente() || ',
Dirección de entrega: ' || direccion_entrega;
  END;
END;
```

-- Ejemplo de uso del método en una consulta

```
SELECT p.datos_pedido() AS datos
FROM Pedidos p;
```

DATOS
1 Pedido, Cliente: Cliente: Nombre: Estudios Pantalla Plateada / Dirección: Calle Principal 123/ Número de cuenta: 1234567890, Dirección de entrega: Calle Principal 123
2 Pedido, Cliente: Cliente: Nombre: Atrezzo Estelar / Dirección: Calle del Olmo 45/ Número de cuenta: 0987654321, Dirección de entrega: Calle del Olmo 456
3 Pedido, Cliente: Cliente: Nombre: Producciones CineMágico / Dirección: Avenida del Roble 78/ Número de cuenta: 9876543210, Dirección de entrega: Avenida del Roble 789
4 Pedido, Cliente: Cliente: Nombre: Films de Fantasía S.A. / Dirección: Calle del Pino 31/ Número de cuenta: 5678901234, Dirección de entrega: Calle del Pino 321
5 Pedido, Cliente: Cliente: Nombre: Soluciones VestuarioCine / Dirección: Calle del Cedro 87/ Número de cuenta: 4321098765, Dirección de entrega: Calle del Cedro 987

-- Implementación del método para obtener los datos del alquiler

```
CREATE OR REPLACE TYPE BODY Alquiler_obj AS
  MEMBER FUNCTION datos_alquiler RETURN VARCHAR2 IS
    v_pedido Pedido_obj;
    v_producto Producto_obj;
  BEGIN
    SELECT Deref(pedido) INTO v_pedido FROM dual;
    SELECT Deref(producto) INTO v_producto FROM dual;
    RETURN 'Alquiler: Pedido: ' || v_pedido.datos_pedido() || ', Producto: ' || v_producto.datos_producto();
  END;
END;
```

-- Ejemplo de uso del método en una consulta

```
SELECT a.datos_alquiler() AS datos
FROM Alquileres a;
```

⚙️ DATOS

1	Alquiler: Pedido: Pedido, Cliente: Cliente: Nombre: Estudios Pantalla Plateada / Dirección: Calle Principal 123/ Número de cuenta: 1234567890, Dirección de entrega: Calle Principal 123, Producto: Producto: Nombre: Ma
2	Alquiler: Pedido: Pedido, Cliente: Cliente: Nombre: Atrezzo Estelar / Dirección: Calle del Olmo 45/ Número de cuenta: 0987654321, Dirección de entrega: Calle del Olmo 456, Producto: Producto: Nombre: Ma
3	Alquiler: Pedido: Pedido, Cliente: Cliente: Nombre: Producciones CineMágico / Dirección: Avenida del Roble 78/ Número de cuenta: 9876543210, Dirección de entrega: Avenida del Roble 789, Producto: Producto: Nombre: Ma
4	Alquiler: Pedido: Pedido, Cliente: Cliente: Nombre: Films de Fantasía S.A. / Dirección: Calle del Pino 31/ Número de cuenta: 5678901234, Dirección de entrega: Calle del Pino 321, Producto: Producto: Nombre: Ma
5	Alquiler: Pedido: Pedido, Cliente: Cliente: Nombre: Soluciones VestuarioCine / Dirección: Calle del Cedro 87/ Número de cuenta: 4321098765, Dirección de entrega: Calle del Cedro 987, Producto: Producto: Nombre: Ma

-- Implementación del método para obtener los datos de la factura

```
CREATE OR REPLACE TYPE BODY Factura_obj AS
  MEMBER FUNCTION detalle_factura RETURN VARCHAR2 IS
    v_pedido Pedido_obj;
  BEGIN
    SELECT Deref(pedido) INTO v_pedido FROM dual;
    RETURN 'Factura: ID: ' || id_factura || ', Pedido: ' || v_pedido.datos_pedido() || ' / Fecha: ' || TO_CHAR(fecha_factura, 'DD-MON-YYYY') || ' / Importe: ' || importe;
  END;
END;
```

-- Ejemplo de uso del método en una consulta

```
SELECT f.detalle_factura() AS datos
FROM Facturas f;
```

⚙️ DATOS

1	Factura: ID: 1, Pedido: Pedido, Cliente: Cliente: Nombre: Estudios Pantalla Plateada / Dirección: Calle Principal 123/ Número de cuenta: 1234567890, Dirección de entrega: Calle Principal 123 / Fecha: 19-MAY-2023 / Importe: 10
2	Factura: ID: 2, Pedido: Pedido, Cliente: Cliente: Nombre: Atrezzo Estelar / Dirección: Calle del Olmo 45/ Número de cuenta: 0987654321, Dirección de entrega: Calle del Olmo 456 / Fecha: 19-MAY-2023 / Importe: 150,5
3	Factura: ID: 3, Pedido: Pedido, Cliente: Cliente: Nombre: Producciones CineMágico / Dirección: Avenida del Roble 78/ Número de cuenta: 9876543210, Dirección de entrega: Avenida del Roble 789 / Fecha: 19-MAY-2023 / Importe: 20
4	Factura: ID: 4, Pedido: Pedido, Cliente: Cliente: Nombre: Films de Fantasía S.A. / Dirección: Calle del Pino 31/ Número de cuenta: 5678901234, Dirección de entrega: Calle del Pino 321 / Fecha: 19-MAY-2023 / Importe: 120,75
5	Factura: ID: 5, Pedido: Pedido, Cliente: Cliente: Nombre: Soluciones VestuarioCine / Dirección: Calle del Cedro 87/ Número de cuenta: 4321098765, Dirección de entrega: Calle del Cedro 987 / Fecha: 19-MAY-2023 / Importe: 180,2

-- Implementación del método para obtener los datos de la entrega

```
CREATE OR REPLACE TYPE BODY Entrega_obj AS
  MEMBER FUNCTION datos_entrega RETURN VARCHAR2 IS
    v_pedido Pedido_obj;
  BEGIN
    SELECT Deref(pedido) INTO v_pedido FROM dual;
    RETURN 'Entrega: ID: ' || id_entrega || ', Pedido: ' || v_pedido.datos_pedido() || ' / Fecha: ' || TO_CHAR(fecha_entrega, 'DD-MON-
```



```
YYYY') || ' / Matrícula: ' || matricula || ' / Transportista: ' ||
transportista || ' / Detalles: ' || detalles;
END;
END;
```

-- Ejemplo de uso del método en una consulta

```
SELECT e.datos_entrega() AS datos
FROM Entregas e;
```

⚙️ DATOS

1	Entrega: ID: 1 / Fecha: 20-MAY-2023 / Matrícula: ABC123 / Transportista: Transportista 1 / Detalles: Entrega programada para el set de filmación "Aventuras en la Selva".
2	Entrega: ID: 2 / Fecha: 21-MAY-2023 / Matrícula: DEF456 / Transportista: Transportista 2 / Detalles: Entrega urgente para el rodaje de la película de acción "Explosión Final".
3	Entrega: ID: 3 / Fecha: 22-MAY-2023 / Matrícula: GHI789 / Transportista: Transportista 3 / Detalles: Entrega especial de atrezzo para el escenario de la obra de teatro "Sueños de Shakespeare".
4	Entrega: ID: 4 / Fecha: 23-MAY-2023 / Matrícula: JKL012 / Transportista: Transportista 4 / Detalles: Entrega programada para el set de grabación de la serie de televisión "Misterios Sin Resolver".
5	Entrega: ID: 5 / Fecha: 24-MAY-2023 / Matrícula: MNO345 / Transportista: Transportista 5 / Detalles: Entrega de vestuario para el rodaje de la película histórica "El Tiempo Dorado".

-- Implementación del método para obtener los datos de la recogida

```
CREATE OR REPLACE TYPE BODY Recogida_obj AS
  MEMBER FUNCTION datos_recogida RETURN VARCHAR2 IS
    v_pedido Pedido_obj;
  BEGIN
    SELECT Deref(pedido) INTO v_pedido FROM dual;
    RETURN 'Recogida: ID: ' || id_recogida || ', Pedido: ' ||
v_pedido.datos_pedido() || ' / Fecha: ' || TO_CHAR(fecha_recogida, 'DD-
MON-YYYY') || ' / Matrícula: ' || matricula || ' / Transportista: ' ||
transportista || ' / Detalles: ' || detalles;
  END;
END;
```

-- Ejemplo de uso del método en una consulta

```
SELECT r.datos_recogida() AS datos
FROM Recogidas r;
```

⚙️ DATOS

1	Recogida: ID: 1, Pedido: Pedido, Cliente: Cliente: Nombre: Estudios Pantalla Plateada / Dirección: Calle Principal 123/ Número de cuenta: 1234567890, Dirección de entrega: Calle Principal 123 / Fecha: 25-MAY-2023 / Matrícula: PQR678 / Transportista: Transp
2	Recogida: ID: 2, Pedido: Pedido, Cliente: Cliente: Nombre: Atrezzo Estelar / Dirección: Calle del Olmo 45/ Número de cuenta: 0987654321, Dirección de entrega: Calle del Olmo 456 / Fecha: 26-MAY-2023 / Matrícula: STU901 / Transportista: Transportista 7 / De
3	Recogida: ID: 3, Pedido: Pedido, Cliente: Cliente: Nombre: Producciones Cinemágico / Dirección: Avenida del Roble 78/ Número de cuenta: 9876543210, Dirección de entrega: Avenida del Roble 789 / Fecha: 27-MAY-2023 / Matrícula: VWX234 / Transportista: Transp
4	Recogida: ID: 4, Pedido: Pedido, Cliente: Cliente: Nombre: Filma de Fantasía S.A. / Dirección: Calle del Pino 31/ Número de cuenta: 5678901234, Dirección de entrega: Calle del Pino 321 / Fecha: 28-MAY-2023 / Matrícula: YZA567 / Transportista: Transportista
5	Recogida: ID: 5, Pedido: Pedido, Cliente: Cliente: Nombre: Soluciones VestuarioCine / Dirección: Calle del Cedro 87/ Número de cuenta: 4321098765, Dirección de entrega: Calle del Cedro 987 / Fecha: 29-MAY-2023 / Matrícula: BCD890 / Transportista: Transport

Se han presentado consultas y resultados de todos los métodos de la base de datos.

Consultas (SELECTS)

En este apartado realizaremos diferentes consultas relevantes a la base de datos en múltiples tablas. Se indicará qué objetivo se logra con cada una de ellas.

1. Obtener todos los productos disponibles en la base de datos:

```
SELECT p.id_producto, p.nombre, p.descripcion, p.tamaño
FROM Productos p
WHERE p.disponibilidad = 'S';
```

ID_PRODUCTO	NOMBRE	DESCRIPCION	TAMAÑO
1	1 Sillón Chesterfield	Un elegante sillón de cuero de estilo clásico, perfecto para ambientar escenas de época.	10
2	3 Teléfono de Dial	Un teléfono de disco auténtico de los años 70, ideal para ambientar escenas de nostalgia o detectivescas.	8
3	4 Reloj de Péndulo	Un reloj de péndulo clásico que añade un toque de elegancia y sofisticación a cualquier decorado.	12

2. Obtener todos los pedidos junto con los datos del cliente y la dirección de entrega:

```
SELECT p.id_pedido, c.nombre AS nombre_cliente, c.direccion AS
direccion_entrega
FROM Pedidos p
JOIN Clientes c ON p.cliente = REF(c);
```

ID_PEDIDO	NOMBRE_CLIENTE	DIRECCION_ENTREGA
1	1 Estudios Pantalla Plateada	Calle Principal 123
2	2 Atrezzo Estelar	Calle del Olmo 45
3	3 Producciones CineMágico	Avenida del Roble 78
4	4 Films de Fantasía S.A.	Calle del Pino 31
5	5 Soluciones VestuarioCine	Calle del Cedro 87

3. Obtener todos los clientes con los datos de los mismos.

```
SELECT c.id_cliente, c.nombre, c.direccion, c.numero_cuenta, c.detalles
FROM Clientes c;
```

ID_CLIENTE	NOMBRE	DIRECCION	NUMERO_CUENTA	DETALLES
1	1 Estudios Pantalla Plateada	Calle Principal 123	1234567890	Este cliente trabaja en televisión.
2	2 Atrezzo Estelar	Calle del Olmo 45	0987654321	Este cliente necesita eficacia y rapidez.
3	3 Producciones CineMágico	Avenida del Roble 78	9876543210	Este cliente busca atrezzo original y creativo.
4	4 Films de Fantasía S.A.	Calle del Pino 31	5678901234	Este cliente necesita decorados y atrezzo realistas.
5	5 Soluciones VestuarioCine	Calle del Cedro 87	4321098765	Este cliente requiere un amplio catálogo de opciones de vestuario para cine y televisión.

4. Obtener todas las facturas junto con los datos del pedido y el importe:

```
SELECT f.id_factura, p.id_pedido, p.cliente.nombre AS nombre_cliente,
p.direccion_entrega, p.fecha_pedido, f.importe, f.fecha_factura
FROM Facturas f
JOIN Pedidos p ON f.pedido = REF(p);
```

	ID_FACTURA	ID_PEDIDO	NOMBRE_CLIENTE	DIRECCION_ENTREGA	FECHA_PEDIDO	IMPORTE	FECHA_FACTURA
1	1	1	Estudios Pantalla Plateada	Calle Principal 123	19/05/23	100	19/05/23
2	2	2	Atrezzo Estelar	Calle del Olmo 456	19/05/23	150,5	19/05/23
3	3	3	Producciones CineMágico	Avenida del Roble 789	19/05/23	200	19/05/23
4	4	4	Films de Fantasía S.A.	Calle del Pino 321	19/05/23	120,75	19/05/23
5	5	5	Soluciones VestuarioCine	Calle del Cedro 987	19/05/23	180,25	19/05/23

5. Obtener el promedio del importe de las facturas:

```
SELECT AVG(importe) AS promedio_importe
FROM Facturas;
```

	PROMEDIO_IMPORTE
1	150,3

Esta consulta puede ser útil para obtener información estadística sobre el importe promedio de las facturas emitidas. Esto puede servir en un caso real para evaluar la rentabilidad y conocer el nivel de ingresos que está teniendo la empresa Cine Arte Atrezzo, en este caso.

6. Obtener los datos de las entregas junto con los datos de los pedidos y clientes asociados:

```
SELECT e.id_entrega, p.id_pedido, p.direccion_entrega, p.fecha_pedido,
c.nombre AS nombre_cliente, e.fecha_entrega, e.matricula, e.transportista
FROM Entregas e
JOIN Pedidos p ON e.pedido = REF(p)
JOIN Clientes c ON p.cliente = REF(c);
```

	ID_ENTREGA	ID_PEDIDO	DIRECCION_ENTREGA	FECHA_PEDIDO	NOMBRE_CLIENTE	FECHA_ENTREGA	MATRICULA	TRANSPORTISTA
1	1	1	Calle Principal 123	19/05/23	Estudios Pantalla Plateada	20/05/23	ABC123	Transportista 1
2	2	2	Calle del Olmo 456	19/05/23	Atrezzo Estelar	21/05/23	DEF456	Transportista 2
3	3	3	Avenida del Roble 789	19/05/23	Producciones CineMágico	22/05/23	GHI789	Transportista 3
4	4	4	Calle del Pino 321	19/05/23	Films de Fantasía S.A.	23/05/23	JKL012	Transportista 4
5	5	5	Calle del Cedro 987	19/05/23	Soluciones VestuarioCine	24/05/23	MNO345	Transportista 5

Esta consulta resulta muy útil para realizar un seguimiento exhaustivo de las entregas. Permite obtener una visión general de las entregas realizadas por la empresa, además puede ser útil para brindar un servicio más eficiente y personalizado al cliente.

7. Obtener los datos de las recogidas junto con los datos de los pedidos y clientes asociados:

```
SELECT r.id_recogida, p.id_pedido, p.direccion_entrega, p.fecha_pedido,
c.nombre AS nombre_cliente, r.fecha_recogida, r.matricula, r.transportista
FROM Recogidas r
JOIN Pedidos p ON r.pedido = REF(p)
JOIN Clientes c ON p.cliente = REF(c);
```

	ID_RECOGIDA	ID_PEDIDO	DIRECCION_ENTREGA	FECHA_PEDIDO	NOMBRE_CLIENTE	FECHA_RECOGIDA	MATRICULA	TRANSPORTISTA
1	1	1	Calle Principal 123	19/05/23	Estudios Pantalla Plateada	25/05/23	PQR678	Transportista 6
2	2	2	Calle del Olmo 456	19/05/23	Atrezzo Estelar	26/05/23	STU901	Transportista 7
3	3	3	Avenida del Roble 789	19/05/23	Producciones CineMágico	27/05/23	VWX234	Transportista 8
4	4	4	Calle del Pino 321	19/05/23	Films de Fantasía S.A.	28/05/23	YZA567	Transportista 9
5	5	5	Calle del Cedro 987	19/05/23	Soluciones VestuarioCine	29/05/23	BCD890	Transportista 10

Similar a la consulta anterior, al combinar los datos de recogida con los datos de los pedidos y los clientes, esta consulta permite realizar un análisis de rendimiento logístico y seguimiento de recogidos. Se puede evaluar la eficiencia en la gestión de las entregas, identificar posibles retrasos o problemas recurrentes.

8. Obtener el total de importes facturados por cliente y el número de facturas emitidas:

```
SELECT c.nombre AS nombre_cliente, COUNT(f.id_factura) AS numero_facturas,
SUM(f.importe) AS total_importe
FROM Clientes c
LEFT JOIN Pedidos p ON c.id_cliente = p.cliente.id_cliente
LEFT JOIN Facturas f ON p.id_pedido = f.pedido.id_pedido
GROUP BY c.nombre;
```

	NOMBRE_CLIENTE	NUMERO_FACTURAS	TOTAL_IMPORTE
1	Producciones CineMágico	1	200
2	Films de Fantasía S.A.	1	120,75
3	Soluciones VestuarioCine	1	180,25
4	Atrezzo Estelar	1	150,5
5	Estudios Pantalla Plateada	1	100

Esta consulta a nivel empresarial puede ser útil para obtener información agregada sobre los clientes y su relación con los pedidos y facturas. La consulta proporciona el número de facturas asociadas a cada cliente y el total del importe de esas facturas. Esto permite realizar un análisis de ventas por cliente, identificando aquellos clientes que generan más facturas y contribuyen en mayor medida al total de ventas de la empresa. Esta consulta permite a la empresa, por tanto, identificar a los clientes valiosos.

9. Obtener el número total de pedidos realizados por un cliente específico:

```
SELECT c.nombre AS nombre_cliente, COUNT(p.id_pedido) AS total_pedidos
FROM Clientes c
LEFT JOIN Pedidos p ON c.id_cliente = p.cliente.id_cliente
WHERE c.id_cliente = 4
GROUP BY c.nombre;
```

	NOMBRE_CLIENTE	TOTAL_PEDIDOS
1	Films de Fantasía S.A.	1

Esta consulta permite el seguimiento del historial de pedidos de un cliente concreto. Al filtrar la consulta por un cliente específico, esta consulta puede ayudarnos a realizar una evaluación de la fidelidad del cliente, observando la cantidad total de pedidos realizados por ese cliente. Esto puede ser útil para tener una visión general de su actividad comercial.

10. Obtener el detalle de un pedido en concreto junto con el importe total de la factura correspondiente:

```
SELECT p.id_pedido, p.cliente.nombre AS nombre_cliente,
p.direccion_entrega, p.fecha_pedido, f.importe
FROM Pedidos p
JOIN Facturas f ON p.id_pedido = f.pedido.id_pedido
WHERE p.id_pedido = 3;
```

	ID_PEDIDO	NOMBRE_CLIENTE	DIRECCION_ENTREGA	FECHA_PEDIDO	IMPORTE
1	3	Producciones CineMágico	Avenida del Roble 789	19/05/23	200

Esta consulta permite realizar un seguimiento específico de un pedido en particular y obtener información detallada sobre él. Al unir la tabla de pedidos con la tabla de facturas, podemos asegurarnos de que solo se obtengan los pedidos que tienen una factura asociada. Esto es útil para verificar y garantizar la consistencia y la integridad de los datos entre estas dos entidades relacionadas. Esta consulta también permitirá a la empresa la resolución de consultas o problemas relacionados con un pedido en específico.

11. Obtener el cliente con el gasto total más alto:

```
SELECT c.id_cliente, c.nombre, SUM(f.importe) AS gasto_total
FROM Clientes c
JOIN Pedidos p ON c.id_cliente = p.cliente.id_cliente
JOIN Facturas f ON p.id_pedido = f.pedido.id_pedido
GROUP BY c.id_cliente, c.nombre
ORDER BY gasto_total DESC
FETCH FIRST 1 ROWS ONLY;
```

	ID_CLIENTE	NOMBRE	GASTO_TOTAL
1	3	Producciones CineMágico	200

Esta consulta es útil para identificar al cliente que ha tenido el mayor gasto total en términos de importe de facturas. Esto permite identificar al cliente más valioso en términos de volumen de compras y contribución a los ingresos de la empresa. Al conocer al cliente de mayor gasto total, la empresa puede enfocar sus esfuerzos de marketing, ventas y atención al cliente en esta persona o empresa. Esto implica brindar un servicio personalizado, ofrecer servicios exclusivos, y establecer relaciones sólidas para fomentar su lealtad y aumentar el valor de su negocio.

12. Obtener el producto más alquilado:

```
SELECT p.id_producto, p.nombre, COUNT(*) AS total_alquileres
FROM Productos p
JOIN Alquileres a ON p.id_producto = a.producto.id_producto
GROUP BY p.id_producto, p.nombre
ORDER BY total_alquileres DESC
FETCH FIRST 1 ROWS ONLY;
```

	ID_PRODUCTO	NOMBRE	TOTAL_ALQUILERES
1	2	Maleta Vintage	2

Esta consulta es útil para identificar el producto más popular o con mayor cantidad de alquileres. Será el producto más demandado por los clientes. Al conocer este importante dato, la empresa puede desarrollar estrategias específicas para promocionar y destacar este producto, aumentar los precios o desarrollar complementos y accesorios relacionados para maximizar los ingresos generados por este.

13. Obtener los productos con más alquileres ordenados:

```
SELECT p.id_producto, p.nombre, COUNT(*) AS total_alquileres
FROM Productos p
JOIN Alquileres a ON p.id_producto = a.producto.id_producto
GROUP BY p.id_producto, p.nombre
ORDER BY total_alquileres DESC;
```

	ID_PRODUCTO	NOMBRE	TOTAL_ALQUILERES
1	2	Maleta Vintage	2
2	3	Teléfono de Dial	1
3	5	Candelabro Gótico	1
4	1	Sillón Chesterfield	1
5	4	Reloj de Péndulo	1

Esta consulta permite identificar los productos más populares y demandados por los clientes. Al conocer los productos más populares en términos de alquileres, la empresa puede gestionar mejor su inventario y asegurarse de tener suficiente disponibilidad de estos productos para satisfacer la demanda. Esto implica mantener un stock adecuado, y la adquisición de los productos en función de su popularidad. Además, al conocer la cantidad de alquileres de cada producto, la empresa puede realizar un análisis de tendencias y preferencias del cliente. Puede identificar los productos más solicitados y comprender mejor las necesidades y gustos de sus clientes. Esta información puede ayudar a tomar decisiones informadas sobre la gestión de inventario, desarrollo de productos y estrategias de marketing.

14. Obtener los clientes en orden descendente según el total gastado:

```
SELECT c.nombre AS nombre_cliente, SUM(f.importe) AS total_gastado
FROM Clientes c
INNER JOIN Pedidos p ON c.id_cliente = p.cliente.id_cliente
INNER JOIN Facturas f ON p.id_pedido = f.pedido.id_pedido
GROUP BY c.nombre
ORDER BY SUM(f.importe) DESC;
```

	NOMBRE_CLIENTE	TOTAL_GASTADO
1	Producciones CineMágico	200
2	Soluciones VestuarioCine	180,25
3	Atrezzo Estelar	150,5
4	Films de Fantasía S.A.	120,75
5	Estudios Pantalla Plateada	100

La consulta proporciona información sobre el total gastado por cada cliente, lo que permite realizar análisis comparativos de la rentabilidad entre diferentes clientes. Esto es útil para evaluar la rentabilidad de los clientes individuales, identificar aquellos que generan mayores ingresos y tomar decisiones basadas en datos sobre cómo asignar recursos y esfuerzos de marketing. Además, la empresa puede segmentar a los clientes en función de su nivel de gasto. Esto permite adaptar las estrategias de marketing y ventas a cada segmento, ofrecer promociones y ofertas específicas para incentivar un mayor gasto, y diseñar campañas personalizadas para satisfacer las necesidades y preferencias de cada grupo de clientes.

15. Obtener el nombre y la cuenta bancaria de los clientes que tienen pedidos asociados

```
SELECT c.nombre, c.numero_cuenta
FROM Clientes c
WHERE c.id_cliente IN (SELECT p.cliente.id_cliente FROM Pedidos p);
```

	NOMBRE	NUMERO_CUENTA
1	Films de Fantasía S.A.	5678901234
2	Producciones CineMágico	9876543210
3	Atrezzo Estelar	0987654321
4	Soluciones VestuarioCine	4321098765
5	Estudios Pantalla Plateada	1234567890

La consulta proporciona información específica sobre los nombres y números de cuenta de los clientes que han realizado pedidos. Esta información es útil para mantener actualizados los registros de clientes, gestionar las cuentas de los clientes de manera efectiva y garantizar una comunicación precisa y oportuna. Además, la empresa puede identificar a los clientes activos y concretar estrategias de retención y fidelización específicas para ellos.

16. Obtener los clientes junto con la información de las facturas correspondientes, ordenados por nombre de cliente:

```
SELECT c.nombre, c.numero_cuenta, f.id_factura, f.fecha_factura, f.importe
FROM Clientes c
INNER JOIN Facturas f ON c.id_cliente = f.pedido.cliente.id_cliente
ORDER BY c.nombre;
```

	NOMBRE	NUMERO_CUENTA	ID_FACTURA	FECHA_FACTURA	IMPORTE
1	Atrezzo Estelar	0987654321	2	19/05/23	150,5
2	Estudios Pantalla Plateada	1234567890	1	19/05/23	100
3	Films de Fantasía S.A.	5678901234	4	19/05/23	120,75
4	Producciones CineMágico	9876543210	3	19/05/23	200
5	Soluciones VestuarioCine	4321098765	5	19/05/23	180,25

Esta consulta proporciona información sobre los nombres y números de cuenta de los clientes, lo que es útil para mantener registros precisos y actualizados de los clientes. Facilita la gestión de cuentas de clientes y garantiza una comunicación efectiva y personalizada con ellos.

17. Obtener el detalle de los pedidos realizados por un cliente en particular.

```
SELECT c.nombre AS NOMBRE_CLIENTE, p.datos_pedido() AS detalle_pedido
FROM Pedidos p
INNER JOIN Clientes c ON p.cliente.id_cliente = c.id_cliente
WHERE c.id_cliente = 2;
```

	NOMBRE_CLIENTE	DETALLE_PEDIDO
1	Atrezzo Estelar Pedido, Cliente: Cliente: Nombre: Atrezzo Estelar / Dirección: Calle del Olmo 45/ Número de cuenta: 0987654321, Dirección de entrega: Calle del Olmo 456	
2	Atrezzo Estelar Pedido, Cliente: Cliente: Nombre: Atrezzo Estelar / Dirección: Calle del Olmo 45/ Número de cuenta: 0987654321, Dirección de entrega: Calle del Alce 987	

La consulta utiliza la función "datos_pedido()" para obtener información detallada sobre los pedidos del cliente. Esta información es útil para tener un registro completo de los pedidos de un cliente en particular.

18. Obtener el nombre de los productos alquilados por cada cliente:

```
SELECT c.nombre, p.nombre AS producto_alquilado
FROM Clientes c
INNER JOIN Pedidos pe ON c.id_cliente = pe.cliente.id_cliente
INNER JOIN Alquileres a ON pe.id_pedido = a.pedido.id_pedido
INNER JOIN Productos p ON a.producto.id_producto = p.id_producto;
```

	NOMBRE	PRODUCTO_ALQUILADO
1	Estudios Pantalla Plateada	Sillón Chesterfield
2	Atrezzo Estelar	Maleta Vintage
3	Producciones CineMágico	Teléfono de Dial
4	Films de Fantasía S.A.	Reloj de Péndulo
5	Soluciones VestuarioCine	Candelabro Gótico
6	Soluciones VestuarioCine	Maleta Vintage

Esta consulta permite realizar un seguimiento de los productos que están en manos de los clientes y tener un registro actualizado de los alquileres.

19. Obtener la información completa del pedido, con diferentes detalles

```
SELECT p.id_pedido, c.nombre AS nombre_cliente, pr.nombre AS
nombre_producto, p.direccion_entrega, c.numero_cuenta, c.detalles,
p.fecha_pedido, p.fecha_devolucion, f.importe
FROM Pedidos p
INNER JOIN Clientes c ON p.cliente.id_cliente = c.id_cliente
INNER JOIN Alquileres a ON p.id_pedido = a.pedido.id_pedido
INNER JOIN Productos pr ON a.producto.id_producto = pr.id_producto
INNER JOIN Facturas f ON p.id_pedido = f.pedido.id_pedido;
```

ID_PEDIDO	NOMBRE_CLIENTE	NOMBRE_PRODUCTO	DIRECCION_ENTREGA	NUMERO_CUENTA	DETALLES	FECHA_PEDIDO	FECHA_DEVOLUCION	IMPORTE
1	Estudios Pantalla Plateada	Sillón Chesterfield	Calle Principal 123	1234567890	Este cliente trabaja en televisión.	19/05/23	26/05/23	100
2	Atrezzo Estelar	Maleta Vintage	Calle del Olmo 456	0987654321	Este cliente necesita eficacia y rapidez.	19/05/23	24/05/23	150,5
3	Producciones CineMágico	Teléfono de Dial	Avenida del Roble 789	9876543210	Este cliente busca atrezzo original y creativo.	19/05/23	22/05/23	200
4	Films de Fantasía S.A.	Reloj de Péndulo	Calle del Pino 321	5678901234	Este cliente necesita decorados y atrezzo realistas.	19/05/23	21/05/23	120,75
5	Soluciones VestuarioCine	Candelabro Gótico	Calle del Cedro 987	4321098765	Este cliente requiere un amplio catálogo de opciones de vestuario para cine y televisión.	19/05/23	29/05/23	180,25

La consulta proporciona una visión completa del ciclo de pedido, desde la información del pedido (ID de pedido, fechas de pedido y devolución) hasta los detalles del cliente (nombre, número de cuenta) y los detalles del producto (nombre). Esto facilita el seguimiento y la gestión de cada etapa del proceso de pedido. Además, ayuda a mantener registros precisos de las transacciones, realizar conciliaciones y gestionar las cuentas de los clientes de manera efectiva.

20. Obtener los datos del pedido con los detalles del producto, la entrega y la recogida correspondientes:

```
SELECT p.id_pedido, c.nombre AS nombre_cliente, pr.nombre AS
nombre_producto, pr.descripcion AS descripcion_producto,
      e.fecha_entrega, e.matricula AS matricula_entrega, e.transportista
AS transportista_entrega,
      r.fecha_recogida, r.matricula AS matricula_recogida,
r.transportista AS transportista_recogida
FROM Pedidos p
INNER JOIN Clientes c ON p.cliente.id_cliente = c.id_cliente
INNER JOIN Alquileres a ON p.id_pedido = a.pedido.id_pedido
INNER JOIN Productos pr ON a.producto.id_producto = pr.id_producto
LEFT JOIN Entregas e ON p.id_pedido = e.pedido.id_pedido
LEFT JOIN Recogidas r ON p.id_pedido = r.pedido.id_pedido;
```

ID_PEDIDO	NOMBRE_CLIENTE	NOMBRE_PRODUCTO	DESCRIPCION_PRODUCTO	FECHA_ENTREGA	MATRICULA_ENTREGA	TRANSPORTISTA_ENTREGA	FECHA_RECOGIDA	MATRICULA_RECOGIDA
1	Estudios Pantalla Plateada	Sillón Chesterfield	Un elegante sillón de cuero de estilo clásico, perfecto para ambientar escenas de época.	20/05/23	ABC123	Transportista 1	25/05/23	PQR678
2	Atrezzo Estelar	Maleta Vintage	Una maleta antigua de aspecto retro, ideal para viajes en el tiempo o escenarios de época.	21/05/23	DEF456	Transportista 2	26/05/23	STU901
3	Producciones CineMágico	Teléfono de Dial	Un teléfono de disco auténtico de los años 70, ideal para ambientar escenas de nostalgia o detectivescas.	22/05/23	GHI789	Transportista 3	27/05/23	VWX234
4	Films de Fantasía S.A.	Reloj de Péndulo	Un reloj de péndulo clásico que añade un toque de elegancia y sofisticación a cualquier decorado.	23/05/23	JKL012	Transportista 4	28/05/23	YZA567
5	Soluciones VestuarioCine	Candelabro Gótico	Un candelabro imponente con diseño gótico, perfecto para escenas de misterio y suspense.	24/05/23	MNO345	Transportista 5	29/05/23	BCD890
6	Soluciones VestuarioCine	Maleta Vintage	Una maleta antigua de aspecto retro, ideal para viajes en el tiempo o escenarios de época.	(null)	(null)	(null)	(null)	(null)
7	Atrezzo Estelar	Sillón Chesterfield	Un elegante sillón de cuero de estilo clásico, perfecto para ambientar escenas de época.	(null)	(null)	(null)	(null)	(null)

Esta consulta obtiene información detallada sobre los pedidos, incluyendo el nombre del cliente que realizó el pedido, el producto y sus detalles, la fecha de entrega y la recogida correspondientes y la matrícula del vehículo que realizó estas acciones. Con esta información, la empresa puede realizar un seguimiento detallado de los pedidos, conocer qué productos se han entregado y recogido, así como los detalles logísticos asociados a dichas operaciones. Esto puede ser útil para el control y la gestión de inventario, la planificación de rutas de entrega y recogida, y el análisis de la eficiencia operativa de la empresa.

21. Obtener los datos y detalles de pedidos, entregas y recogidas

```
SELECT p.id_pedido, c.nombre AS nombre_cliente, p.fecha_pedido,
      e.fecha_entrega, e.matricula AS matricula_entrega, e.detalles AS
DETALLES_ENTREGA,
      r.fecha_recogida, r.matricula AS matricula_recogida, r.detalles AS
DETALLES_RECOGIDA
FROM Pedidos p
JOIN Clientes c ON p.cliente.id_cliente = c.id_cliente
LEFT JOIN Entregas e ON p.id_pedido = Deref(e.pedido).id_pedido
LEFT JOIN Recogidas r ON p.id_pedido = Deref(r.pedido).id_pedido;
```

O	NOMBRE_CLIENTE	FECHA_PEDIDO	FECHA_ENTREGA	MATRICULA_ENTREGA	DETALLES_ENTREGA	FECHA_RECOGIDA	MATRICULA_RECOGIDA	DETALLES_RECOGIDA
1	Estudios Pantalla Plateada	19/05/23	20/05/23	ABC123	Entrega programada para el set de filmación "Aventuras en la Selva".	25/05/23	PQR678	Recogida en la dirección principal
2	Atrezzo Estelar	19/05/23	21/05/23	DEF456	Entrega urgente para el rodaje de la película de acción "Explosión Final".	26/05/23	STU901	Recogida en la dirección de la sucursal
3	Producciones CineMágico	19/05/23	22/05/23	GHI789	Entrega especial de atrezzo para el escenario de la obra de teatro "Sueños de Shakespeare".	27/05/23	VWX234	Recogida en la dirección residencial
4	Films de Fantasía S.A.	19/05/23	23/05/23	JKL012	Entrega programada para el set de grabación de la serie de televisión "Misterios Sin Resolver".	28/05/23	YZA567	Recogida en la dirección comercial
5	Soluciones VestuarioCine	19/05/23	24/05/23	MNO345	Entrega de vestuario para el rodaje de la película histórica "El Tiempo Dorado".	29/05/23	BCD890	Recogida en la dirección de entrega alternativa
6	Soluciones VestuarioCine	19/05/23	(null)	(null)	(null)	(null)	(null)	(null)
7	Atrezzo Estelar	19/05/23	(null)	(null)	(null)	(null)	(null)	(null)

Esta consulta ofrece una variante a la consulta anterior. La empresa puede obtener información detallada sobre el proceso de entrega y recogida de productos. Esto facilita la gestión logística y permite un seguimiento preciso de cuándo se entregaron y se recogieron los productos. La variante con el Select 20 es que este último incluye el campo Detalles de entrega y Detalles de recogida.

4. Plan de pruebas

El diseño de un plan de pruebas es una parte crucial del proceso de desarrollo y mantenimiento de una base de datos. Es fundamental para asegurar la calidad y la integridad de los datos almacenados en ella, así como verificar el correcto funcionamiento de la base de datos. Es importante considerar estos aspectos clave:

- **Objetivos de las pruebas:** Se deben establecer los objetivos claros de las pruebas. Estos objetivos pueden incluir la validación de la integridad de los datos, entre otros aspectos.
- **Datos de prueba:** Se deben crear conjuntos de datos de prueba realistas que cubran una amplia gama de situaciones. Esto implica generar datos de prueba tanto válidos como inválidos para evaluar el comportamiento de la base de datos en diferentes escenarios.

Las pruebas podrían abarcar las siguientes áreas:

- Pruebas de integridad de datos de entidad: Se verifica que los registros de cada entidad se creen y actualicen correctamente. Esto implica comprobar que se inserten los datos obligatorios en cada registro y que se mantengan consistentes durante las actualizaciones.
- Pruebas de integridad de datos de dominio: Se validan los valores de los atributos para asegurarse de que cumplen con las restricciones y reglas definidas. Esto incluye comprobar rangos de fechas, formatos de datos, valores mínimos y máximos, restricciones de longitud, entre otros.
- Pruebas de integridad de datos referencial: Se verifican las relaciones entre las tablas y se asegura que los datos referenciados existan y sean consistentes. Esto se logra mediante pruebas de inserción, actualización y eliminación, y comprobando que las restricciones de clave foránea se respeten en todo momento.

Integridad de los datos

El diseño propuesto garantiza completamente la integridad de los datos (de entidad, de dominio y referencial). Se han establecido restricciones y reglas, se han definido claves primarias y claves foráneas para mantener la integridad referencial, evitando así inconsistencias en los datos relacionados entre las tablas.

Sin embargo, se pueden realizar pruebas específicas para garantizar la integridad de los datos de entidad, de dominio y referencial.

- La integridad de entidad se puede verificar mediante pruebas que aseguren que no existen registros duplicados en las tablas de clientes, productos, pedidos, facturas, entre otros. Se pueden realizar pruebas de inserción y actualización para validar que los campos obligatorios estén presentes y que no se violen las restricciones de clave primaria.
- Para garantizar la integridad de dominio, se pueden ejecutar pruebas que verifiquen que los valores almacenados en los campos correspondan a los dominios definidos. Por ejemplo, se pueden validar que los tamaños de los productos estén dentro de un rango específico, que las fechas de pedido y factura sean válidas, y que los importes sean valores numéricos adecuados.
- La integridad referencial se puede comprobar mediante pruebas que verifiquen que las relaciones entre las tablas se mantengan correctamente. Esto implica asegurarse de que los valores en los campos de clave externa correspondan a valores existentes en las tablas relacionadas. Por ejemplo, se pueden realizar pruebas para asegurar que los **pedidos estén asociados a clientes válidos** y que las facturas estén vinculadas a pedidos existentes.

Prueba de registro de un pedido completo

-- Insertar un nuevo cliente

```
INSERT INTO Clientes (id_cliente, nombre, direccion, numero_cuenta,
detalles)
VALUES (23, 'Jane Smith', '789 Oak Avenue', '987654321', 'Cliente VIP');
```

-- Insertar un nuevo pedido

```
INSERT INTO Pedidos (id_pedido, cliente, direccion_entrega, fecha_pedido,
fecha_devolucion)
VALUES (23, (SELECT REF(c) FROM Clientes c WHERE c.id_cliente = 3), '987
Maple Street', SYSDATE, SYSDATE + 10);
```

-- Insertar un nuevo alquiler

```
INSERT INTO Alquileres (pedido, producto)
VALUES ((SELECT REF(p) FROM Pedidos p WHERE p.id_pedido = 3), (SELECT
REF(pr) FROM Productos pr WHERE pr.id_producto = 2));
```

```
SELECT c.id_cliente, c.nombre, c.direccion, c.numero_cuenta, c.detalles
FROM Clientes c;
```

ID_CLIENTE	NOMBRE	DIRECCION	NUMERO_CUENTA	DETALLES
1	1 Estudios Pantalla Plateada	Calle Principal 123	1234567890	Este cliente trabaja en televisión.
2	2 Atrezzo Estelar	Calle del Olmo 45	0987654321	Este cliente necesita eficacia y rapidez.
3	3 Producciones CineMágico	Avenida del Roble 78	9876543210	Este cliente busca atrezzo original y creativo.
4	4 Films de Fantasía S.A.	Calle del Pino 31	5678901234	Este cliente necesita decorados y atrezzo realistas.
5	5 Soluciones VestuarioCine	Calle del Cedro 87	4321098765	Este cliente requiere un amplio catálogo de opciones de vestuario para cine y televisión.
6	23 Jane Smith	789 Oak Avenue	987654321	Cliente VIP

Observamos como la prueba de insertar un pedido completo en la base de datos se ha llevado a cabo satisfactoriamente.

INSERT INTO fallidos

Estos fueron algunos de los errores arrojaros cuando no se llevó a cabo una correcta inserción de los datos. El error nos indica que se violaron restricciones.

Error que empieza en la línea: 446 del comando :

```
INSERT INTO Clientes (id_cliente, nombre, direccion, numero_cuenta, detalles)
VALUES (1, 'John Doe', '123 Main Street', '123456789', 'Cliente regular')
Informe de error -
ORA-00001: restricción única (HITO.SYS_C007519) violada
```

Error que empieza en la línea: 449 del comando :

```
INSERT INTO Pedidos (id_pedido, cliente, direccion_entrega, fecha_pedido, fecha_devolucion)
VALUES (1, (SELECT REF(c) FROM Clientes c WHERE c.id_cliente = 1), '456 Elm Street', SYSDATE, SYSDATE + 7)
Informe de error -
ORA-00001: restricción única (HITO.SYS_C007523) violada
```

Cambiar detalles en tabla Recogidas (UPDATE)

```
UPDATE Recogidas
SET detalles = 'La recogida se llevó a cabo de manera satisfactoria.'
WHERE id_recogida = 2;
```

```
SELECT r.id_recogida, p.id_pedido, p.cliente.nombre AS nombre_cliente,
p.direccion_entrega, r.fecha_recogida, r.matricula, r.transportista,
r.detalles
FROM Recogidas r
JOIN Pedidos p ON r.pedido = REF(p);
```

ID_RECOGIDA	ID_PEDIDO	NOMBRE_CLIENTE	DIRECCION_ENTREGA	FECHA_RECOGIDA	MATRICULA	TRANSPORTISTA	DETALLES
1	1	1 Estudios Pantalla Plateada	Calle Principal 123	25/05/23	PQR678	Transportista 6	Recogida en la dirección principal
2	2	2 Atrezzo Estelar	Calle del Olmo 456	26/05/23	STU901	Transportista 7	La recogida se llevó a cabo de manera satisfactoria.

5. Evaluación crítica

En la base de datos propuesta para Cine Arte Atrezzo, se garantiza la integridad de los datos debido a las restricciones definidas en el diseño. Las claves primarias y foráneas se han establecido correctamente, asegurando que los registros estén vinculados de manera adecuada.

Además, se han definido restricciones de dominio para los campos, limitando los valores válidos que pueden ser almacenados. Estas características, combinadas con la implementación de pruebas adecuadas en el plan de pruebas, ayudarán a mantener la integridad de los datos y a evitar inconsistencias en la base de datos.

Por otro lado, el uso del modelo relacional orientado a objetos de Oracle ofrece ventajas significativas en términos de representación precisa de la estructura y comportamiento de los datos del mundo real.

Al combinar objetos y tablas, se logra una mayor flexibilidad y organización en el diseño de la base de datos. Los objetos permiten modelar entidades complejas y encapsular su lógica y comportamiento en métodos asociados. Esto facilita la implementación de funcionalidades específicas y el mantenimiento de la base de datos.

La base de datos propuesta para Cine Arte Atrezzo cumple con todos los requisitos de usuario de manera excepcional y presenta características destacadas que la hacen realmente atractiva. Su estructura, basada en el **modelo relacional orientado a objetos** de Oracle, junto con el uso de objetos, tablas y métodos, garantiza la **integridad de los datos** y brinda flexibilidad y eficiencia en la gestión de la información. La base de datos diseñada satisface las necesidades de la empresa y mejora sus operaciones diarias.