

Proyecto Final

Raya Pérez José Luis

Muñoz Sandoval Alan Sebastián

Cruz Martinez Giovanni

Diciembre 2023

Base de Datos para tienda BlockBuster (alquiler de películas)

1. Requerimientos

1. Gestión de Inventarios:
Manejo de películas disponibles para alquiler. Detalles como título, género, director, año de lanzamiento, formato (DVD, Blu-Ray, VHS)
2. Registro de Clientes:
Información de los clientes como nombre y dirección. Historial de alquileres.
3. Transacciones:
Procesos de alquiler de películas. Gestión de tarifas, fechas de alquiler, fechas de devolución.
4. Gestión de Empleados:
Información del personal incluyendo turnos, y detalles de contacto.

2. Modelo Conceptual

Articulo

- IDarticulo: Identificador único (Llave primaria).
- Título: Nombre de la película
- Género: Género de la película
- Director: Director de la película
- Año de Lanzamiento: Año en que se lanzó la película
- Formato/Plataforma: Formato (DVD, Blu-Ray, VHS) para películas.

Cliente

- IDcliente: Identificador único (Llave primaria).
- Nombre: Nombre del cliente.
- Dirección: Dirección del cliente.
- Email: Dirección de correo electrónico.

Ticket

- IDticket: Identificador único (Llave primaria).
- Fecha Alquiler/Venta: Fecha de la transacción.
- Fecha Devolución: Fecha de devolución (para alquileres).
- Total: Monto total de la transacción.
- IDcliente: Identificador del cliente (Llave foránea).
- IDempleado: Identificador del empleado (Llave foránea).

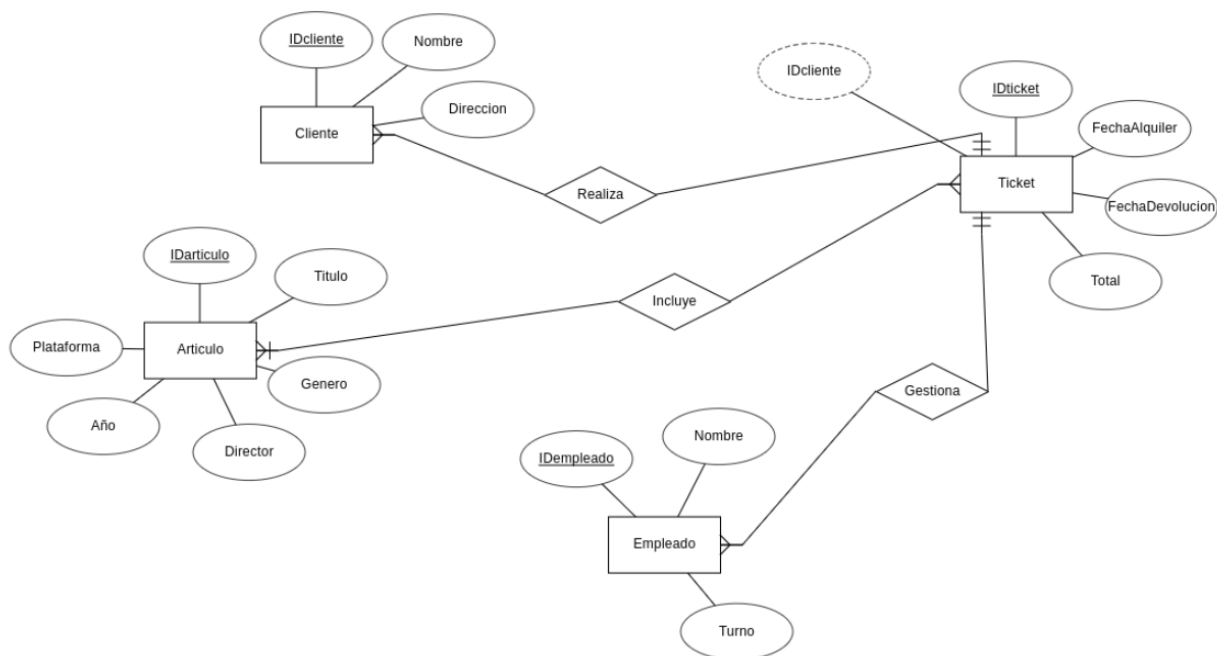
Empleado

- IDempleado: Identificador único (Llave primaria).
- Nombre: Nombre del empleado.
- Turno: Turno del empleado.

Relaciones

- **Realiza (Cliente - Ticket):** Un cliente puede realizar múltiples transacciones (alquiler/venta).
- **Incluye (Ticket -Artículo):** Una transacción puede incluir varios títulos de películas o videojuegos.
- **Gestiona (Empleado - Ticket):** Un empleado es responsable de gestionar las transacciones.

3. Modelo E/R




```
FOREIGN KEY (IDcliente) REFERENCES Cliente(IDcliente) ON DELETE SET NULL,
FOREIGN KEY (IDempleado) REFERENCES Empleado(IDempleado) ON DELETE SET NULL ON UPDATE CASCADE
);

CREATE TABLE Incluye
(
    IDticket INT NOT NULL,
    IDarticulo INT NOT NULL,
    FOREIGN KEY (IDticket) REFERENCES Ticket(IDticket) ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (IDarticulo) REFERENCES Articulo(IDarticulo) ON DELETE CASCADE ON UPDATE CASCADE,
    PRIMARY KEY (IDticket, IDarticulo)
);

-- Check
ALTER TABLE Ticket
ADD CONSTRAINT check_total_nonnegative
CHECK (Total >= 0);

ALTER TABLE Articulo
ADD CONSTRAINT check_year_valid
CHECK (Año > 1800);

ALTER TABLE Empleado
ADD CONSTRAINT check_turno_valid
CHECK (Turno IN ('Mañana', 'Tarde', 'Noche'));

-- Crear dominios
CREATE DOMAIN total_as_money AS NUMERIC
CHECK (VALUE >= 0 AND VALUE <= 1000);

CREATE DOMAIN year_as_movie_year AS INT
CHECK (VALUE > 1895);

CREATE DOMAIN shift_as_work_shift AS VARCHAR
CHECK (VALUE IN ('Mañana', 'Tarde', 'Noche'));

-- Modificación de las columnas de las tablas para usar los nuevos dominios
ALTER TABLE Ticket
ALTER COLUMN Total TYPE total_as_money
USING Total::total_as_money;

ALTER TABLE Articulo
ALTER COLUMN Año TYPE year_as_movie_year
USING Año::year_as_movie_year;

ALTER TABLE Empleado
ALTER COLUMN Turno TYPE shift_as_work_shift
USING Turno::shift_as_work_shift;

-- Restricción Tuplas para la tabla Ticket
ALTER TABLE Ticket
ADD CONSTRAINT check_fechas_validas
CHECK (FechaDevolucion IS NULL OR FechaDevolucion >= FechaAlquiler);

-- Restricción Tuplas para la tabla Articulo
ALTER TABLE Articulo
```

```
ADD CONSTRAINT check_antiguo_valido  
CHECK (NOT (Genero = 'Antiguo' AND Año >= 1950));
```

Notas:

Para el ID de las tablas, se usó SERIAL para que automáticamente se generaran el id uno por uno
Se crean solo 15 empleados, pues no es lógico crear más de 100 empleados para administrar el negocio con 100 clientes.

6. Restricciones de integridad referencial

1. Ejemplo 1

Tablas involucradas en la restricción:

- Incluye
- Ticket

FK de la tabla que referencia y PK de la tabla referenciada:

- FK: Incluye.IDticket
- PK: Ticket.IDticket

Justificación del trigger de integridad referencial elegido:

- Elegimos el trigger *CASCADE* para la llave foránea porque si un ticket es eliminado, todas las referencias a ese ticket en la tabla 'Incluye' también deben eliminarse automáticamente. Esto asegura que no tengamos registros huérfanos en 'Incluye' que hagan referencia a tickets inexistentes.

Instrucción DELETE para evidenciar que la restricción está funcionando:

```
1  -- Primero insertamos un ticket y un artículo de prueba  
2  INSERT INTO Ticket (FechaAlquiler, FechaDevolucion, Total, IDcliente, IDempleado)  
3  VALUES ('2023-11-17', '2023-11-27', 99.72, 37, 14);  
4  
5  -- Obtenemos el ID del ticket insertado  
6  SELECT LASTVAL();  
7  
8  -- Suponiendo que el ID obtenido es 101, insertamos una entrada en Incluye  
9  INSERT INTO Incluye (IDticket, IDarticulo) VALUES (101, 89);  
10  
11 -- Ahora eliminamos el ticket  
12 DELETE FROM Ticket WHERE IDticket = 101;  
13  
14 -- Verificamos si la entrada en Incluye asociada ha sido eliminada  
15 SELECT * FROM Incluye WHERE IDticket = 101;
```

Data Output Messages Notifications

Idticket [PK] integer	Idarticulo [PK] integer	

Vemos que no se nos devuelve ningún dato, indicando que la restricción funciona

2. Ejemplo 2

Tablas involucradas en la restricción:

- Ticket
- Cliente

FK de la tabla que referencia y PK de la tabla referenciada:

- FK: Ticket.IDcliente
- PK: Cliente.IDcliente

Justificación del trigger de integridad referencial elegido:

- Elegimos el trigger *SET NULL* para la llave foránea porque si un cliente es eliminado, no queremos perder los registros de los tickets que compró, pero debemos indicar que el cliente ya no existe en nuestra base de datos. Por lo tanto, establecemos el valor de IDcliente en los tickets relacionados como NULL.

Instrucción DELETE para evidenciar que la restricción está funcionando:

```

1      -- Primero insertamos un cliente de prueba
2      INSERT INTO Cliente (Nombre, Direccion) VALUES ('Cliente Prueba', 'Dirección Prueba');
3
4      -- Obtenemos el ID del cliente insertado
5      SELECT LASTVAL();
6
7      -- Como el ID obtenido es 113, insertamos un ticket asociado a este cliente
8      INSERT INTO Ticket (FechaAlquiler, FechaDevolucion, Total, IDcliente, IDempleado)
9      VALUES ('2023-11-17', '2023-11-27', 50.00, 113, 14);
10
11     -- Ahora eliminamos el cliente
12     DELETE FROM Cliente WHERE IDcliente = 113;
13
14     -- Verificamos si el IDcliente en el ticket asociado se ha establecido como NULL
15     SELECT * FROM Ticket WHERE IDcliente IS NULL;

```

Data Output Messages Notifications

	idticket [PK] integer	fechaalquiler date	fechadevolucion date	total double precision	idcliente integer	idempleado integer
1	113	2023-11-17	2023-11-27	50	[null]	14

Vemos que aparece el cliente como null, indicando que la restricción funciona

3. Ejemplo3

Tablas involucradas en la restricción:

- Ticket
- Empleado

FK de la tabla que referencia y PK de la tabla referenciada:

- FK: Ticket.IDempleado
- PK: Empleado.IDempleado

Justificación del trigger de integridad referencial elegido:

- Elegimos el trigger *SET NULL* para la llave foránea porque si un empleado es eliminado, no queremos perder los registros de los tickets que gestionó, pero necesitamos reflejar que el empleado ya no está en la empresa. Por lo tanto, establecemos el valor de IDempleado en los tickets relacionados como NULL.

Instrucción DELETE para evidenciar que la restricción está funcionando:

```

1  -- Primero insertamos un empleado de prueba
2  INSERT INTO Empleado (Nombre, Turno) VALUES ('Empleado Prueba', 'Matutino');
3
4  -- Obtenemos el ID del empleado insertado
5  SELECT LASTVAL();
6
7  -- Como ID obtenido es 21, insertamos un ticket asociado a este empleado
8  INSERT INTO Ticket (FechaAlquiler, FechaDevolucion, Total, IDcliente, IDempleado)
9  VALUES ('2023-11-17', '2023-11-27', 75.00, 37, 21);
10
11 -- Ahora eliminamos el empleado
12 DELETE FROM Empleado WHERE IDempleado = 21;
13
14 -- Verificamos si el IDempleado en el ticket asociado se ha establecido como NULL
15 SELECT * FROM Ticket WHERE IDempleado IS NULL;

```

Data Output Messages Notifications

	Idticket [PK] integer	fechaalquiler date	fechadevolucion date	total double precision	Idcliente integer	IDempleado integer
1	108	2023-11-17	2023-11-27	75	37	[null]

Vemos que aparece el empleado como null, indicando que la restricción funciona

4. Ejemplo 4**Tablas involucradas en la restricción:**

- Incluye
- Artículo

FK de la tabla que referencia y PK de la tabla referenciada:

- FK: Incluye.IDArticulo
- PK: Artículo.IDArticulo

Justificación del trigger de integridad referencial elegido:

- Elegimos el trigger *CASCADE* para la llave foránea porque si un artículo es eliminado, es lógico eliminar todas las referencias de ese artículo en la tabla 'Incluye', ya que el artículo ya no existe y no puede ser parte de ninguna transacción.

Instrucción DELETE para evidenciar que la restricción está funcionando:

```

1  -- Primero insertamos un artículo de prueba
2  INSERT INTO Articulo (Titulo, Genero, Director, Año, Plataforma)
3  VALUES ('Artículo Prueba', 'Género Prueba', 'Director Prueba', 2021, 'DVD');
4
5  -- Obtenemos el ID del artículo insertado
6  SELECT LASTVAL();
7
8  -- Como el ID obtenido es 101, insertamos una entrada en Incluye
9  INSERT INTO Incluye (IDticket, IDarticulo) VALUES (57, 101);
10
11 -- Ahora eliminamos el artículo
12 DELETE FROM Articulo WHERE IDarticulo = 101;
13
14 -- Verificamos si las entradas en Incluye asociadas han sido eliminadas
15 SELECT * FROM Incluye WHERE IDarticulo = 101;

```

Data Output Messages Notifications

Idticket	IDarticulo
[PK] integer	[PK] integer

Vemos que no se nos devuelve ningún dato, indicando que la restricción funciona

7. Restricciones Check

1. Ejemplo 1

Tabla elegida: Ticket

Atributo elegido: Total

Breve descripción de la restricción: El total no puede ser negativo, debe ser mayor o igual a cero.

Instrucción para la creación de la restricción:

```

ALTER TABLE Ticket
ADD CONSTRAINT check_total_nonnegative
CHECK (Total >= 0);

```

Instrucción que permita evidenciar que la restricción está funcionando:

```

1
2  INSERT INTO Ticket (FechaAlquiler, FechaDevolucion, Total, IDcliente, IDempleado)
3  VALUES ('2023-12-05', '2023-12-15', -10.00, 37, 14);
4

```

Data Output Messages Notifications

ERROR: La fila que falla contiene (114, 2023-12-05, 2023-12-15, -10, 37, 14).el nuevo registro para la relación «ticket» viola la restricción «check» «check_total_nonnegative»

ERROR: el nuevo registro para la relación «ticket» viola la restricción «check» «check_total_nonnegative»

SQL state: 23514

Detail: La fila que falla contiene (114, 2023-12-05, 2023-12-15, -10, 37, 14).

2. Ejemplo 2

Tabla elegida: Artículo

Atributo elegido: Año

Breve descripción de la restricción: El año del artículo debe ser posterior a 1800.

Instrucción para la creación de la restricción:

```
ALTER TABLE Artículo
```

```
ADD CONSTRAINT check_year_valid
```

```
CHECK (Año > 1800);
```

Instrucción que permita evidenciar que la restricción está funcionando:

```
1 INSERT INTO Artículo (Título, Género, Director, Año, Plataforma)
2 VALUES ('Artículo Prueba', 'Género Prueba', 'Director Prueba', 1700, 'DVD');
3
```

Data Output Messages Notifications

ERROR: La fila que falla contiene (102, Artículo Prueba, Género Prueba, Director Prueba, 1700, DVD).el nuevo registro para la relación «artículo» viola la restricción «check» «artículo_año_check»
ERROR: el nuevo registro para la relación «artículo» viola la restricción «check» «artículo_año_check»
SQL state: 23514
Detail: La fila que falla contiene (102, Artículo Prueba, Género Prueba, Director Prueba, 1700, DVD).

3. Ejemplo 3

Tabla elegida: Empleado

Atributo elegido: Turno

Breve descripción de la restricción: El turno debe ser 'Mañana', 'Tarde' o 'Noche'.

Instrucción para la creación de la restricción:

```
ALTER TABLE Empleado
```

```
ADD CONSTRAINT check_turno_valid
```

```
CHECK (Turno IN ('Mañana', 'Tarde', 'Noche'));
```

Instrucción que permita evidenciar que la restricción está funcionando:

```
1 -- Intento de insertar un empleado con un turno no válido
2 INSERT INTO Empleado (Nombre, Turno) VALUES ('Empleado Prueba', 'Madrugada');
3
```

Data Output Messages Notifications

ERROR: La fila que falla contiene (22, Empleado Prueba, Madrugada).el nuevo registro para la relación «empleado» viola la restricción «check» «check_turno_valid»
ERROR: el nuevo registro para la relación «empleado» viola la restricción «check» «check_turno_valid»
SQL state: 23514
Detail: La fila que falla contiene (22, Empleado Prueba, Madrugada).

8. Dominios Personalizados

1. Ejemplo 1

Tabla elegida: Ticket

Atributo elegido: Total

Breve descripción del dominio y de la restricción check propuesta: El dominio para 'Total' asegura que el monto sea no negativo y no exceda un máximo establecido, por ejemplo, 10000.

Instrucción para la creación del dominio personalizado:

```
CREATE DOMAIN total_as_money AS NUMERIC  
CHECK (VALUE >= 0 AND VALUE <= 10000);
```

```
1  -- Inserciones que cumplen con la restricción del dominio total_as_money  
2  INSERT INTO Ticket (FechaAlquiler, FechaDevolucion, Total, IDcliente, IDempleado)  
3  VALUES ('2023-11-17', '2023-11-27', 100.00, 1, 1);  
4  
5  -- Inserciones que violan la restricción del dominio total_as_money  
6  -- Estas inserciones deben fallar  
7  INSERT INTO Ticket (FechaAlquiler, FechaDevolucion, Total, IDcliente, IDempleado)  
8  VALUES ('2023-11-17', '2023-11-27', -50.00, 1, 1);  
9  
10 INSERT INTO Ticket (FechaAlquiler, FechaDevolucion, Total, IDcliente, IDempleado)  
11 VALUES ('2023-11-17', '2023-11-27', 1500.00, 1, 1);  
12
```

Data Output Messages Notifications

ERROR: el valor para el dominio total_as_money viola la restricción «check» «total_as_money_check»

SQL state: 23514

2. Ejemplo 2

Tabla elegida: Artículo

Atributo elegido: Año

Breve descripción del dominio y de la restricción check propuesta: El dominio para 'Año' garantiza que el año sea razonable para una película, por ejemplo, mayor a 1895 (año de la primera película).

Instrucción para la creación del dominio personalizado:

```
CREATE DOMAIN year_as_movie_year AS INT  
CHECK (VALUE > 1895);
```

```
1 -- Inserciones que cumplen con la restricción del dominio year_as_movie_year
2 INSERT INTO Articulo (Titulo, Genero, Director, Año, Plataforma)
3 VALUES ('Artículo Cumple', 'Género Prueba', 'Director Prueba', 2000, 'DVD');
4
5 -- Inserción que viola la restricción del dominio year_as_movie_year
6 -- Esta inserción debe fallar
7 INSERT INTO Articulo (Titulo, Genero, Director, Año, Plataforma)
8 VALUES ('Artículo No Cumple', 'Género Prueba', 'Director Prueba', 1890, 'DVD');
```

Data Output Messages Notifications

ERROR: el valor para el dominio year_as_movie_year viola la restricción «check» «year_as_movie_year_check»
SQL state: 23514

3. Ejemplo 3

Tabla elegida: Empleado

Atributo elegido: Turno

Breve descripción del dominio y de la restricción check propuesta: El dominio para 'Turno' valida que el turno sea uno de los valores aceptados: 'Mañana', 'Tarde', o 'Noche'.

Instrucción para la creación del dominio personalizado:

```
CREATE DOMAIN shift_as_work_shift AS VARCHAR
CHECK (VALUE IN ('Mañana', 'Tarde', 'Noche'));
```

```
1 -- Inserciones que cumplen con la restricción del dominio shift_as_work_shift
2 INSERT INTO Empleado (Nombre, Turno) VALUES ('Empleado Prueba', 'Mañana');
3
4 -- Inserción que viola la restricción del dominio shift_as_work_shift
5 -- Esta inserción debe fallar
6 INSERT INTO Empleado (Nombre, Turno) VALUES ('Empleado No Cumple', 'Madrugada');
```

Data Output Messages Notifications

ERROR: el valor para el dominio shift_as_work_shift viola la restricción «check» «shift_as_work_shift_check»
SQL state: 23514

9. Restricciones para Tuplas

1. Ejemplo 1

Tabla elegida: Ticket

Breve descripción de la restricción: La fecha de devolución no puede ser anterior a la fecha de alquiler.

Instrucción para la creación de la restricción:

```
ALTER TABLE Ticket
ADD CONSTRAINT check_dates_valid
CHECK (FechaDevolucion >= FechaAlquiler);
```

Instrucción INSERT que permita evidenciar que la restricción está funcionando:

```
1
2
3 -- Ejemplo de inserción que cumple con las restricciones
4 INSERT INTO Ticket (FechaAlquiler, FechaDevolucion, Total, IDcliente, IDempleado)
5 VALUES ('2023-12-05', '2023-12-10', 50.00, 1, 1);
6
7
8
9 -- Ejemplo de inserciones que violan las restricciones
10 -- Estas inserciones deben fallar
11 INSERT INTO Ticket (FechaAlquiler, FechaDevolucion, Total, IDcliente, IDempleado)
12 VALUES ('2023-12-10', '2023-12-05', 50.00, 1, 1);
13
14
15
```

Data Output Messages Notifications

ERROR: La fila que falla contiene (118, 2023-12-10, 2023-12-05, 50.00, 1, 1).el nuevo registro para la relación «ticket» viola la restricción «check» «check_fechas_validas»

ERROR: el nuevo registro para la relación «ticket» viola la restricción «check» «check_fechas_validas»

SQL state: 23514

Detail: La fila que falla contiene (118, 2023-12-10, 2023-12-05, 50.00, 1, 1).

2. Ejemplo 2

Tabla elegida: Articulo

Breve descripción de la restricción: Que el año de la película concuerde con el género .«Antiguo»

Instrucción para la creación de la restricción:

```
ALTER TABLE Articulo
ADD CONSTRAINT check_antiguo_valido
CHECK (NOT (Genero = 'Antiguo' AND Año >= 1950));
));
```

Instrucción INSERT que permita evidenciar que la restricción está funcionando:

```
1
2
3
4 -- Ejemplo de inserción que cumple con las restricciones
5 INSERT INTO Articulo (Titulo, Genero, Director, Año, Plataforma)
6 VALUES ('Vaso griego', 'Antiguo', 'Desconocido', 1900, 'Físico');
7 -- Ejemplo de inserciones que violan las restricciones
8
9 -- Estas inserciones deben fallar
10 INSERT INTO Articulo (Titulo, Genero, Director, Año, Plataforma)
11 VALUES ('Vaso griego moderno', 'Antiguo', 'Desconocido', 2000, 'Físico');
12
```

Data Output Messages Notifications

ERROR: La fila que falla contiene (186, Vaso griego moderno, Antiguo, Desconocido, 2000, Físico).el nuevo registro para la relación «articulo» viola la restricción «check» «check_antiguo_valido»

ERROR: el nuevo registro para la relación «articulo» viola la restricción «check» «check_antiguo_valido»

SQL state: 23514

Detail: La fila que falla contiene (186, Vaso griego moderno, Antiguo, Desconocido, 2000, Físico).

10. Consultas

1. Consulta 1

Redacción clara de la consulta: Listar id, titulo y año de todos los artículos que están en formato Blu-Ray y que son del año 2000 en adelante.

Código en lenguaje SQL de la consulta:

```
SELECT IDarticulo, Titulo, Año
FROM Articulo
WHERE Plataforma = 'BlueRay' AND Año >= 2000;
```

1 SELECT IDarticulo, Titulo, Año

2 FROM Articulo

3 WHERE Plataforma = 'BlueRay' AND Año >= 2000;

Data Output

Messages

Notifications

	Idarticulo [PK] integer	titulo character varying (100)	año integer
1	2	But I'm a Cheerleader	2004
2	7	Escape from Dartmoor	2004
3	12	Offspring	2011
4	24	Nightmare City	2009
5	36	Glengarry Glen Ross	2003
6	37	When Evening Falls on Bucharest or Metabolism	2003
7	45	Where the Boys Are	2003
8	48	Little Engine That Could, The	2003
9	56	Captain America	2012
10	58	Command Decision	2011
11	60	Close-Up (Nema-ye Nazdik)	2010
12	62	Disappeared, The	2012
13	64	Stationmaster Is on Fire - The Movie, That Is!, The (Stinsen brinner... filmen allt...	2002
14	69	Shadrach	2011
15	73	Command Decision	2001
16	80	Piranha	2001
17	90	Fidanzati, I (Fiances, The)	2002
18	92	Pulling John	2003
19	97	The Gentleman from Epsom	2002
20	98	The Girls	2009
21	99	La La Land	2016
22	100	The Amazing Spider-Man 2	2012

2. Consulta 2

Redacción clara de la consulta: Obtener en orden de mayor a menor el número de tickets que cada empleado ha manejado.

Código en lenguaje SQL de la consulta:

```
SELECT e.IDempleado, e.Nombre, COUNT(t.IDticket) as Numero_Tickets
FROM Empleado e
JOIN Ticket t ON e.IDempleado = t.IDempleado
GROUP BY e.IDempleado, e.Nombre
ORDER BY Numero_Tickets DESC;
```

```

1 SELECT e.IDempleado, e.Nombre, COUNT(t.IDticket) as Numero_Tickets
2 FROM Empleado e
3 JOIN Ticket t ON e.IDempleado = t.IDempleado
4 GROUP BY e.IDempleado, e.Nombre
5 ORDER BY Numero_Tickets DESC;

```

Data Output Messages Notifications

	Idempleado [PK] integer	nombre character varying (100)	numero_tickets bigint
1	1	Moise Lequeux	10
2	14	Calypso Kearney	9
3	3	Gibb Berndt	9
4	2	Ryan Kays	9
5	4	Kimbell Minget	8
6	10	Claudian Ramsdale	7
7	15	Jeanie Murrthum	7
8	12	Aurlie Esselin	7
9	8	Sully Jenkin	7
10	11	Krishna Meddows	6
11	9	Hymie Searight	5
12	13	Rita Dayne	5
13	5	Gordy Hanbury	5
14	6	Theodosia Cuthill	4
15	7	Sancho Bushnell	3

3. Consulta 3

Redacción clara de la consulta: Determinar las tres películas más alquiladas.

Código en lenguaje SQL de la consulta:

```

SELECT a.Titulo, COUNT(i.IDarticulo) as Numero_Alquileres
FROM Articulo a
JOIN Incluye i ON a.IDarticulo = i.IDarticulo
GROUP BY a.Titulo
ORDER BY Numero_Alquileres DESC
LIMIT 3;

```

```
1 SELECT a.Titulo, COUNT(i.IDarticulo) as Numero_Alquileres
2 FROM Articulo a
3 JOIN Incluye i ON a.IDarticulo = i.IDarticulo
4 GROUP BY a.Titulo
5 ORDER BY Numero_Alquileres DESC
6 LIMIT 3;
```

Data Output

Messages

Notifications

	titulo character varying (100)	numero_alquileres bigint
1	Regarding Henry	4
2	Ballad of Narayama, The (Narayama bushiko)	4
3	Lipstick	4

11. Vistas

1. Vista 1

Redacción clara de la vista planteada: Esta vista muestra el total de ingresos generados por cada cliente a través de los alquileres.

Código en lenguaje SQL que permita crear la vista solicitada:

```
CREATE VIEW IngresosPorCliente AS
SELECT c.IDcliente, c.Nombre, SUM(t.Total) as Total_Ingresos
FROM Cliente c
JOIN Ticket t ON c.IDcliente = t.IDcliente
GROUP BY c.IDcliente, c.Nombre;
```

```

1 CREATE VIEW IngresosPorCliente AS
2 SELECT c.IDcliente, c.Nombre, SUM(t.Total) as Total_Ingresos
3 FROM Cliente c
4 JOIN Ticket t ON c.IDcliente = t.IDcliente
5 GROUP BY c.IDcliente, c.Nombre;
6 SELECT * FROM IngresosPorCliente;

```

Data Output Messages Notifications

	Idcliente integer	nombre character varying (255)	total_Ingresos numeric
1	87	Jarret	327.42
2	74	Kimberlyn	289.02
3	54	Will	199.12
4	71	Ruthie	439.87
5	34	Ernst	272.84
6	51	Lucais	250.26
7	52	Roobbie	66.39
8	70	Renard	136.42
9	83	Rebbecca	157.54
10	67	Rhea	192.59
11	63	Norene	137.7
12	10	Nappy	64.42
13	35	Kincaid	203.79
14	6	Joachim	95.01
15	89	Ave	208.78
16	36	Laura	102.92
17	31	June	138.03
18	50	Leo	152.13
19	97	Kerwin	65.73
20	14	Jere	64.92
21	66	Michel	468.48
22	68	Samson	148.8
Total rows: 63 of 63			Query complete 00:00:00.426

2. Vista 2

Redacción clara de la vista planteada: Esta vista proporciona detalles de cada alquiler, incluyendo el artículo y el cliente que lo alquiló.

Código en lenguaje SQL que permita crear la vista solicitada:

```

CREATE VIEW DetalleAlquileres AS
SELECT a.Titulo, a.Año, c.Nombre as Cliente, t.FechaAlquiler, t.FechaDevolucion

```



```

FROM Articulo a
JOIN Incluye i ON a.IDarticulo = i.IDarticulo
JOIN Ticket t ON i.IDticket = t.IDticket
JOIN Cliente c ON t.IDcliente = c.IDcliente;

```

```

1 CREATE VIEW DetalleAlquileres AS
2 SELECT a.Titulo, a.Año, c.Nombre as Cliente, t.FechaAlquiler, t.FechaDevolucion
3 FROM Articulo a
4 JOIN Incluye i ON a.IDarticulo = i.IDarticulo
5 JOIN Ticket t ON i.IDticket = t.IDticket
6 JOIN Cliente c ON t.IDcliente = c.IDcliente;
7 SELECT * FROM DetalleAlquileres;
8

```

Data Output Messages Notifications

	titulo character varying (100)	año integer	cliente character varying (255)	fechaalquiler date	fechadevolucion date
1	Stolen Seas	1965	Jerome	2023-11-07	2023-11-09
2	The Legend of Bloody Jack	2009	Jerome	2023-11-07	2023-11-09
3	Bangkok Dangerous	1993	Norina	2023-11-17	2023-11-27
4	Moon	1991	June	2021-08-03	2021-08-17
5	Lipstick	2004	Michel	2021-07-19	2021-08-01
6	Young and Prodigious T.S. Spivet, The (L'extravagant voyage du jeune et prodigieux T.S. Spiv...	2009	Ramona	2022-01-25	2022-02-03
7	Milky Way, The (Vole lactée, La)	1999	Hana	2023-05-30	2023-06-12
8	In a World...	2008	Norene	2023-08-12	2023-08-19
9	Dancing at the Blue Iguana	2010	Lilly	2022-03-20	2022-03-26
10	I Married A Strange Person!	1995	Lilly	2022-05-24	2022-05-29
11	Getaway, The	1961	Emylee	2021-10-19	2021-10-31
12	Command Decision	2011	Harris	2021-02-25	2021-03-07
13	Lipstick	2004	Norene	2023-08-12	2023-08-19
14	How Stella Got Her Groove Back	2002	Kynthia	2022-04-28	2022-04-29
15	Lipstick	2004	Harris	2022-11-07	2022-11-19
16	Sixteen Candles	2002	Will	2022-11-04	2022-11-06
17	Regarding Henry	1990	Nappy	2022-05-31	2022-06-13
18	Going Clear: Scientology and the Prison of Belief	1999	Kylynn	2022-04-29	2022-05-04
19	Regarding Henry	1990	Brandais	2022-08-20	2022-08-31
20	Non-Stop	1998	Alidia	2023-08-27	2023-09-09
21	Bangkok Dangerous	1993	Isiahi	2021-02-12	2021-02-19
22

Total rows: 100 of 100 Query complete 00:00:00.239

Success

3. Vista 3

Redacción clara de la vista planteada: Esta vista muestra una lista de artículos clasificados por su género.

Código en lenguaje SQL que permita crear la vista solicitada:

```

CREATE VIEW ArticulosPorGeneros AS
SELECT Genero, COUNT(IDarticulo) as Numero_Articulos
FROM Articulo
GROUP BY Genero;

```

```
1 CREATE VIEW ArticulosPorGeneros AS
2 SELECT Genero, COUNT(IDarticulo) as Numero_Articul
3 FROM Articulo
4 GROUP BY Genero;
5
6 SELECT * FROM ArticulosPorGeneros;
7
```

Data Output Messages Notifications

	genero character varying (100) 🔒	numero_articulos bigint 🔒
1	Animation	3
2	Crime	2
3	Comedy	1
4	Documentary	7
5	Horror	6
6	Drama	38
7	Action	15
8	Musical	1
9	(no genres listed)	1
10	Western	1
11	Comedy	19
12	Adventure	6