



UNIVERSIDADE
DE VIGO

ESCOLA SUPERIOR DE ENXEÑERÍA INFORMÁTICA

PRÁCTICA 1: EER y DDL

GRUPO: BDII1_1

TEMÁTICA: LIBR

DNI: 49705010D NOMBRE: ALONSO IGLESIAS, VICTOR
DNI: 76736578E NOMBRE: BLANCO NESPEREIRA, CLAUDIA YU
DNI: 54233422C NOMBRE: CANCELAS GARCIA, DANIEL
DNI: 44661806S NOMBRE: CARBAJO PAULA, JUAN ANTONIO



BASES DE DATOS II 2024-2025

CALIFICACIÓN

ANÁLISIS DE REQUERIMIENTOS (DESCRIPCIÓN)

La librería busca digitalizar su sistema de gestión de libros y operaciones relacionadas.

El nuevo sistema debe poder registrar y gestionar la información sobre los libros, autores, usuarios y ventas, así como las relaciones entre ellos.

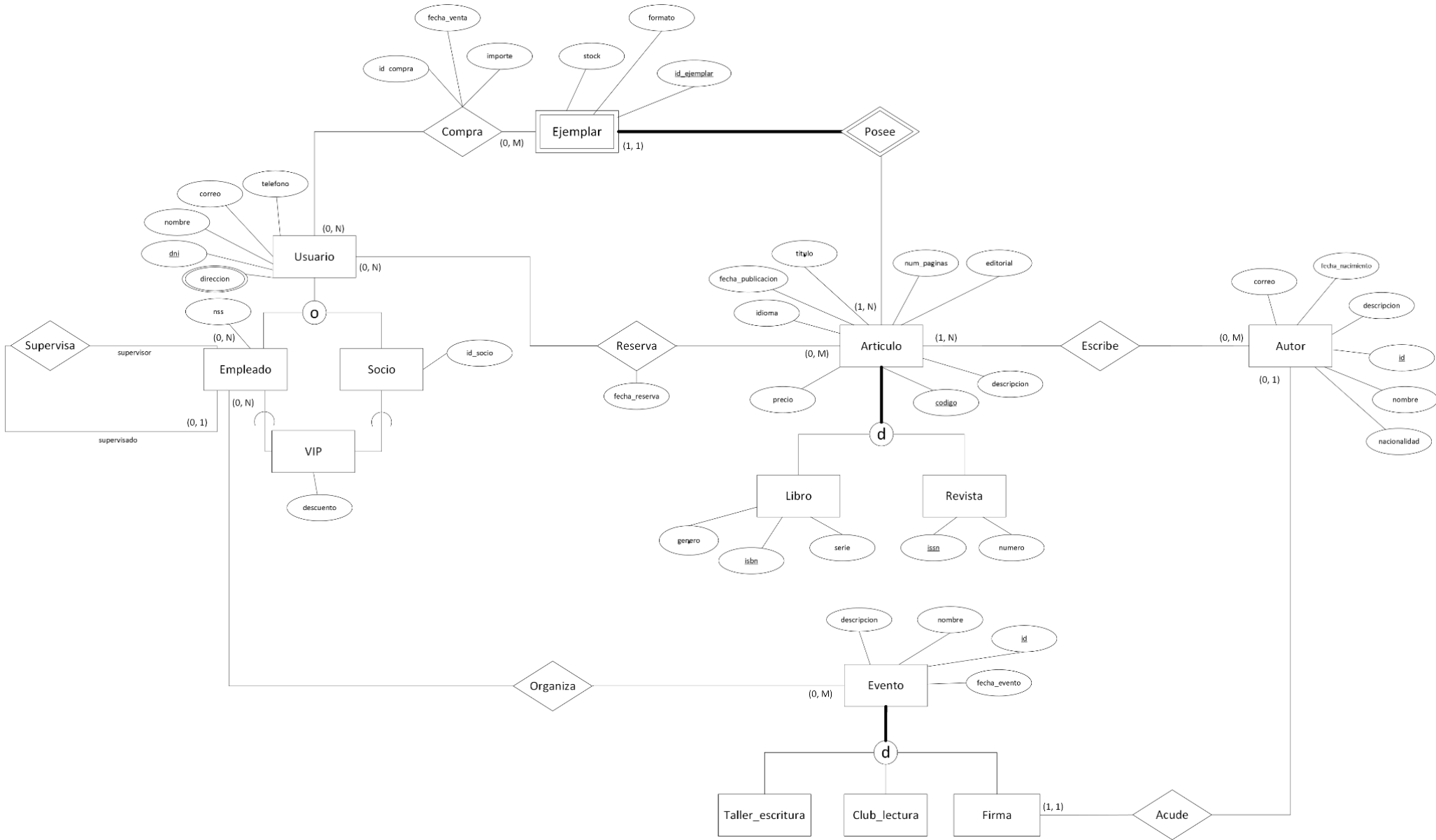
La biblioteca organiza sus materiales en distintas categorías y tiene varios tipos de libros, además de contar con una estructura de empleados para gestionar el servicio.

Se busca diseñar una base de datos que permita registrar y gestionar lo siguiente:

- En la librería se venden artículos, de los cuales se almacenan datos como el título, idioma, editorial, descripción, fecha de publicación, número de páginas y precio. Estos artículos pueden ser libros o revistas. En el caso de los libros, se registra información adicional como el ISBN, los géneros a los que pertenece y si es parte de una serie. Para las revistas, se guardaría el ISSN y el número de la revista.
- De los usuarios se almacenará información como el nombre, teléfono, correo, DNI y dirección. Los usuarios pueden ser empleados, socios, o ambos. Para los empleados, se registrará el número de seguridad social (NSS), mientras que para los socios se guardará un ID de socio. Un usuario que sea VIP será alguien que es empleado y socio al mismo tiempo, y además tendrá asignado un descuento especial por su membresía.
- Los empleados también pueden supervisar a otros empleados, guardando una relación de supervisión entre ellos.
- La librería dispondrá de un número limitado de ejemplares de los artículos disponibles en tienda, cuyo formato podrá ser tapa blanda, tapa dura o bolsillo.
- Los usuarios podrán realizar compras, de las cuales se guardará información como un ID, fecha de compra, método de pago y un importe. Cada compra estará asociada a los ejemplares específicos adquiridos. Además, si un artículo no está disponible en stock, los usuarios tendrán la opción de reservarlo para adquirirlo posteriormente.
- Los artículos pueden ser escritos por uno o varios autores. De cada autor, se almacenará información como el correo, fecha de nacimiento, una descripción, un ID, nombre y nacionalidad. Estos datos se utilizarán para crear una ficha de autor, donde se registrarán todos los artículos que esa persona ha escrito, permitiendo un seguimiento completo de su obra en la librería.

- En la librería se organizan distintos tipos de eventos, como firmas de libros, clubes de lectura y talleres de escritura, los cuales son organizados por los empleados. De cada evento, se registrará el nombre, la fecha y una descripción. Además, los autores son llamados por la librería a realizar firmas, por lo que también se debe guardar dicha información en nuestra base de datos.

DIAGRAMA EER



SENTENCIAS DE INICIALIZACIÓN (DROP'S, ETC.)

SET AUTOCOMMIT on;

SET SERVEROUTPUT on;

/** Eliminar tablas si existen**/

DROP TABLE USUARIO CASCADE CONSTRAINTS;

DROP TABLE USUARIO_DIRECCIONES CASCADE CONSTRAINTS;

DROP TABLE EMPLEADO CASCADE CONSTRAINTS;

DROP TABLE SOCIO CASCADE CONSTRAINTS;

DROP TABLE VIP CASCADE CONSTRAINTS;

DROP TABLE ARTICULO CASCADE CONSTRAINTS;

DROP TABLE LIBRO CASCADE CONSTRAINTS;

DROP TABLE REVISTA CASCADE CONSTRAINTS;

DROP TABLE COMPRA CASCADE CONSTRAINTS;

DROP TABLE EJEMPLAR CASCADE CONSTRAINTS;

DROP TABLE AUTOR CASCADE CONSTRAINTS;

DROP TABLE EVENTO CASCADE CONSTRAINTS;

DROP TABLE FIRMA CASCADE CONSTRAINTS;

DROP TABLE CLUB_LECTURA CASCADE CONSTRAINTS;

DROP TABLE TALLER_ESCRITURA CASCADE CONSTRAINTS;

DROP TABLE RESERVA CASCADE CONSTRAINTS;

DROP TABLE ORGANIZA CASCADE CONSTRAINTS;

DROP TABLE ESCRIBE CASCADE CONSTRAINTS;

SENTENCIAS DE CREACIÓN DE TABLAS

`/** Crear tablas **/`

`CREATE TABLE USUARIO (`

`DNI VARCHAR2(10),`

`NOMBRE VARCHAR2(30),`

`CORREO VARCHAR2(30),`

`TELF VARCHAR2(9),`

`PRIMARY KEY (DNI)`

`);`

`CREATE TABLE USUARIO_DIRECCIONES (`

`DNI VARCHAR2(10),`

`DIRECCION VARCHAR2(100),`

`PRIMARY KEY (DNI, DIRECCION),`

`FOREIGN KEY (DNI) REFERENCES USUARIO`

`);`

`CREATE TABLE EMPLEADO (`

`DNI VARCHAR2(10),`

`NSS VARCHAR2(12) UNIQUE,`

`PRIMARY KEY (DNI),`

`FOREIGN KEY (DNI) REFERENCES USUARIO ON DELETE CASCADE`

`);`

```
ALTER TABLE EMPLEADO ADD CONSTRAINT SUPERVISOR FOREIGN KEY (DNI)
REFERENCES EMPLEADO;
```

```
CREATE TABLE SOCIO (  
    DNI VARCHAR2(10),  
    ID_SOCIO VARCHAR2(12),  
    PRIMARY KEY (DNI),  
    FOREIGN KEY (DNI) REFERENCES USUARIO ON DELETE CASCADE  
);
```

```
CREATE TABLE VIP (  
    DNI VARCHAR2(10),  
    DESCUENTO INTEGER NOT NULL,  
    PRIMARY KEY (DNI),  
    FOREIGN KEY (DNI) REFERENCES EMPLEADO ON DELETE CASCADE,  
    FOREIGN KEY (DNI) REFERENCES SOCIO ON DELETE CASCADE,  
    CHECK (DESCUENTO > 0 AND DESCUENTO <= 50)  
);
```

```
CREATE TABLE ARTICULO (  
    ID_ARTICULO INTEGER PRIMARY KEY,  
    IDIOMA VARCHAR2(20),  
    FECHA_PUBLICACION DATE,  
    TITULO VARCHAR2(30) NOT NULL,  
    NUM_PAGINAS INTEGER,  
    EDITORIAL VARCHAR2(30),
```

```
DESCRIPCION VARCHAR2(200),  
PRECIO DECIMAL(10, 2) NOT NULL,  
CHECK (PRECIO > 0),  
CHECK (NUM_PAGINAS > 0)  
);
```

```
CREATE TABLE LIBRO (  
    ID_ARTICULO INTEGER,  
    ISBN VARCHAR2(25),  
    SERIE VARCHAR2(30),  
    GENERO VARCHAR2(20),  
    PRIMARY KEY (ISBN, ID_ARTICULO),  
    FOREIGN KEY (ID_ARTICULO) REFERENCES ARTICULO  
);
```

```
CREATE TABLE REVISTA (  
    ID_ARTICULO INTEGER,  
    ISSN VARCHAR2(17),  
    NUM_REVISTA VARCHAR2(30),  
    PRIMARY KEY (ISSN, ID_ARTICULO),  
    FOREIGN KEY (ID_ARTICULO) REFERENCES ARTICULO  
);
```

```
CREATE TABLE EJEMPLAR (  
    ID_EJEMPLAR INTEGER,
```

```

ID_ARTICULO INTEGER,
STOCK INTEGER NOT NULL,
FORMATO VARCHAR2(15),
PRIMARY KEY (ID_EJEMPLAR, ID_ARTICULO),
FOREIGN KEY (ID_ARTICULO) REFERENCES ARTICULO,
CHECK (STOCK >= 0),
CHECK (FORMATO IN ('TAPA DURA', 'TAPA BLANDA', 'BOLSILLO'))
);

```

```

CREATE TABLE COMPRA (
    DNI VARCHAR2(10),
    ID_COMPRA INTEGER,
    ID_EJEMPLAR INTEGER,
    ID_ARTICULO INTEGER,
    FECHA_VENTA DATE NOT NULL,
    IMPORTE DECIMAL(10, 2) NOT NULL,
    CANTIDAD INTEGER NOT NULL,
    PRIMARY KEY (ID_EJEMPLAR, ID_ARTICULO, ID_COMPRA),
    FOREIGN KEY (ID_EJEMPLAR, ID_ARTICULO) REFERENCES EJEMPLAR,
    FOREIGN KEY (DNI) REFERENCES USUARIO,
    CHECK (IMPORTE > 0)
);

```

```

CREATE TABLE AUTOR (
    ID_AUTOR INTEGER PRIMARY KEY,
    NOMBRE VARCHAR2(30),

```

```
CORREO VARCHAR2(20),  
FECHA_NACIMIENTO DATE,  
NACIONALIDAD VARCHAR2(20),  
DESCRIPCION VARCHAR2(200)  
);
```

```
CREATE TABLE EVENTO (  
    ID_EVENTO INTEGER PRIMARY KEY,  
    NOMBRE VARCHAR2(20),  
    DESCRIPCION VARCHAR2(200),  
    FECHA_EVENTO DATE  
);
```

```
CREATE TABLE FIRMA (  
    ID_EVENTO INTEGER,  
    ID_FIRMA INTEGER,  
    ID_AUTOR INTEGER NOT NULL,  
    PRIMARY KEY(ID_FIRMA, ID_EVENTO),  
    FOREIGN KEY(ID_EVENTO) REFERENCES EVENTO ON DELETE CASCADE,  
    FOREIGN KEY(ID_AUTOR) REFERENCES AUTOR ON DELETE SET NULL  
);
```

```
CREATE TABLE CLUB_LECTURA (  
    ID_EVENTO INTEGER,  
    ID_LECTURA INTEGER,
```

```
PRIMARY KEY(ID_LECTURA, ID_EVENTO),  
FOREIGN KEY(ID_EVENTO) REFERENCES EVENTO ON DELETE CASCADE  
);
```

```
CREATE TABLE TALLER_ESCRITURA (  
    ID_EVENTO INTEGER,  
    ID_ESCRITURA INTEGER,  
    PRIMARY KEY(ID_ESCRITURA, ID_EVENTO),  
    FOREIGN KEY(ID_EVENTO) REFERENCES EVENTO(ID_EVENTO) ON DELETE CASCADE  
);
```

```
CREATE TABLE RESERVA (  
    DNI VARCHAR2(10),  
    ID_ARTICULO INTEGER,  
    FECHA_RESERVA DATE NOT NULL,  
    CANTIDAD INTEGER NOT NULL,  
    PRIMARY KEY(DNI, ID_ARTICULO),  
    FOREIGN KEY(DNI) REFERENCES USUARIO(DNI),  
    FOREIGN KEY(ID_ARTICULO) REFERENCES ARTICULO(ID_ARTICULO),  
    CHECK (CANTIDAD >0)  
);
```

```
CREATE TABLE ORGANIZA (  
    DNI VARCHAR2(10),  
    ID_EVENTO INTEGER,
```

```
PRIMARY KEY(DNI, ID_EVENTO),  
FOREIGN KEY(DNI) REFERENCES EMPLEADO(DNI),  
FOREIGN KEY(ID_EVENTO) REFERENCES EVENTO(ID_EVENTO)  
);
```

```
CREATE TABLE ESCRIBE (  
    ID_ARTICULO INTEGER,  
    ID_AUTOR INTEGER,  
    PRIMARY KEY(ID_ARTICULO, ID_AUTOR),  
    FOREIGN KEY(ID_ARTICULO) REFERENCES ARTICULO(ID_ARTICULO),  
    FOREIGN KEY(ID_AUTOR) REFERENCES AUTOR(ID_AUTOR)  
);
```

SENTENCIAS DE CREACIÓN DE ÍNDICES

/** Crear Índices **/

/**

Los índices que se crean a continuación tienen como objeto hacer más eficientes las creaciones de las vistas, de esta forma solo se cargarían en memoria los atributos necesarios.

/**

Este índice se crea sobre los atributos ID_ARTICULO, CANTIDAD de la tabla COMPRA.

**/

```
CREATE INDEX INDICE_COMPRA_ARTICULO_CANTIDAD ON COMPRA (ID_ARTICULO, CANTIDAD);
```

/**

Este índice se crea sobre los atributos ID_ARTICULO, STOCK de la tabla EJEMPLAR.

**/

```
CREATE INDEX INDICE_EJEMPLAR_ARTICULO_STOCK ON EJEMPLAR (ID_ARTICULO, STOCK);
```

/**

Este índice se crea sobre los atributos PRECIO de la tabla ARTICULO.

**/

```
CREATE INDEX INDICE_ARTICULO_PRECIO ON ARTICULO(PRECIO);
```

SENTENCIAS DE CREACIÓN DE VISTAS

/** Crear vistas **/

/**

Se crea una vista no actualizable que mostrará los artículos que están disponibles en la tienda en un momento determinado.

Al formarse a partir de dos tablas, según Oracle la modificación es imposible, por lo que la vista no es actualizable.

**/

```
CREATE OR REPLACE VIEW ARTICULOS_DISPONIBLES AS
```

```
SELECT E.ID_ARTICULO, A.TITULO, A.IDIOMA, A.PRECIO, (E.STOCK - SUM(C.CANTIDAD))  
AS DISPONIBLE
```

```
FROM COMPRA C, EJEMPLAR E, ARTICULO A
```

```
WHERE A.ID_ARTICULO = E.ID_ARTICULO AND E.ID_ARTICULO = C.ID_ARTICULO
```

```
GROUP BY E.ID_ARTICULO, A.TITULO, A.IDIOMA, A.PRECIO, E.STOCK
```

```
HAVING (E.STOCK - SUM(C.CANTIDAD)) > 0;
```

/**

Se crea una vista no actualizable que filtrará los libros por autor.

Al formarse a partir de dos tablas, según Oracle la modificación es imposible, por lo que la vista no es actualizable

**/

```
CREATE OR REPLACE VIEW ARTICULOS_POR_AUTOR AS
```

```
SELECT AU.NOMBRE AS AUTOR, AU.CORREO, A.ID_ARTICULO, A.TITULO, A.IDIOMA,  
A.PRECIO
```

```
FROM AUTOR AU, ARTICULO A, ESCRIBE E
```

```
WHERE AU.ID_AUTOR = E.ID_AUTOR AND E.ID_ARTICULO = A.ID_ARTICULO;
```

/**

Se crea una vista actualizable que muestra los artículos por precio.

Es actualizable porque permite realizar operaciones de inserción, modificación y eliminación sobre las filas de la vista.

**/

```
CREATE OR REPLACE VIEW ARTICULOS_POR_PRECIO AS
```

```
SELECT TITULO, IDIOMA, PRECIO
```

```
FROM ARTICULO
```

```
ORDER BY PRECIO;
```

SENTENCIAS DE INSERCIÓN

/**

Insertar datos

**/

```
INSERT INTO USUARIO (DNI, NOMBRE, CORREO, TELF) VALUES ('12345678A', 'Juan Pérez', 'juan.perez@email.com', '612345678');
```

```
INSERT INTO USUARIO (DNI, NOMBRE, CORREO, TELF) VALUES ('23456789B', 'María García', 'maria.garcia@email.com', '623456789');
```

```
INSERT INTO USUARIO (DNI, NOMBRE, CORREO, TELF) VALUES ('34567890C', 'Luis Martínez', 'luis.martinez@email.com', '634567890');
```

```
INSERT INTO USUARIO (DNI, NOMBRE, CORREO, TELF) VALUES ('45678901D', 'Ana López', 'ana.lopez@email.com', '645678901');
```

```
INSERT INTO USUARIO (DNI, NOMBRE, CORREO, TELF) VALUES ('56789012E', 'Pablo Rodríguez', 'pablo.rodriguez@email.com', '656789012');
```

```
INSERT INTO USUARIO_DIRECCIONES (DNI, DIRECCION) VALUES ('12345678A', 'Calle Falsa 123');
```

```
INSERT INTO USUARIO_DIRECCIONES (DNI, DIRECCION) VALUES ('23456789B', 'Avenida Siempre Viva 742');
```

```
INSERT INTO USUARIO_DIRECCIONES (DNI, DIRECCION) VALUES ('34567890C', 'Calle de la Paz 456');
```

```
INSERT INTO USUARIO_DIRECCIONES (DNI, DIRECCION) VALUES ('45678901D', 'Paseo de los Patos 789');
```

```
INSERT INTO USUARIO_DIRECCIONES (DNI, DIRECCION) VALUES ('56789012E', 'Boulevard de los Sueños Rotos 101');
```

```
INSERT INTO EMPLEADO (DNI, NSS) VALUES ('12345678A', 'NSS001234');
```

```
INSERT INTO EMPLEADO (DNI, NSS) VALUES ('23456789B', 'NSS005678');
```

```
INSERT INTO EMPLEADO (DNI, NSS) VALUES ('34567890C', 'NSS007890');
```

INSERT INTO EMPLEADO (DNI, NSS) VALUES ('45678901D', 'NSS009012');

INSERT INTO EMPLEADO (DNI, NSS) VALUES ('56789012E', 'NSS002345');

INSERT INTO SOCIO (DNI, ID_SOCIO) VALUES ('12345678A', 'SOC001');

INSERT INTO SOCIO (DNI, ID_SOCIO) VALUES ('23456789B', 'SOC002');

INSERT INTO SOCIO (DNI, ID_SOCIO) VALUES ('34567890C', 'SOC003');

INSERT INTO SOCIO (DNI, ID_SOCIO) VALUES ('45678901D', 'SOC004');

INSERT INTO SOCIO (DNI, ID_SOCIO) VALUES ('56789012E', 'SOC005');

INSERT INTO VIP (DNI, DESCUENTO) VALUES ('12345678A', 10);

INSERT INTO VIP (DNI, DESCUENTO) VALUES ('23456789B', 15);

INSERT INTO VIP (DNI, DESCUENTO) VALUES ('34567890C', 20);

INSERT INTO VIP (DNI, DESCUENTO) VALUES ('45678901D', 25);

INSERT INTO VIP (DNI, DESCUENTO) VALUES ('56789012E', 30);

INSERT INTO ARTICULO (ID_ARTICULO, IDIOMA, FECHA_PUBLICACION, TITULO, NUM_PAGINAS, EDITORIAL, DESCRIPCION, PRECIO) VALUES (1, 'Español', TO_DATE('2023-01-01', 'YYYY-MM-DD'), 'El Gran Gatsby', 180, 'Editorial 1', 'Una novela clásica.', 15.99);

INSERT INTO ARTICULO (ID_ARTICULO, IDIOMA, FECHA_PUBLICACION, TITULO, NUM_PAGINAS, EDITORIAL, DESCRIPCION, PRECIO) VALUES (2, 'Español', TO_DATE('2023-02-01', 'YYYY-MM-DD'), 'Cien Años de Soledad', 400, 'Editorial 2', 'Una novela mágica.', 22.50);

INSERT INTO ARTICULO (ID_ARTICULO, IDIOMA, FECHA_PUBLICACION, TITULO, NUM_PAGINAS, EDITORIAL, DESCRIPCION, PRECIO) VALUES (3, 'Español', TO_DATE('2023-03-01', 'YYYY-MM-DD'), 'Don Quijote de la Mancha', 600, 'Editorial 3', 'Una historia épica.', 30.00);

INSERT INTO ARTICULO (ID_ARTICULO, IDIOMA, FECHA_PUBLICACION, TITULO, NUM_PAGINAS, EDITORIAL, DESCRIPCION, PRECIO) VALUES (4, 'Español', TO_DATE('2023-04-01', 'YYYY-MM-DD'), '1984', 328, 'Editorial 4', 'Una novela distópica.', 12.75);

```
INSERT INTO ARTICULO (ID_ARTICULO, IDIOMA, FECHA_PUBLICACION, TITULO,
NUM_PAGINAS, EDITORIAL, DESCRIPCION, PRECIO) VALUES (5, 'Español',
TO_DATE('2023-05-01', 'YYYY-MM-DD'), 'Orgullo y Prejuicio', 432, 'Editorial 5', 'Una novela
romántica.', 18.20);
```

```
INSERT INTO LIBRO (ID_ARTICULO, ISBN, SERIE, GENERO) VALUES (1,
'978-3-16-148410-0', 'Serie 1', 'Ficción');
```

```
INSERT INTO LIBRO (ID_ARTICULO, ISBN, SERIE, GENERO) VALUES (2,
'978-3-16-148411-7', 'Serie 2', 'Ficción Mágica');
```

```
INSERT INTO LIBRO (ID_ARTICULO, ISBN, SERIE, GENERO) VALUES (3,
'978-3-16-148412-4', 'Serie 3', 'Ficción Clásica');
```

```
INSERT INTO LIBRO (ID_ARTICULO, ISBN, SERIE, GENERO) VALUES (4,
'978-3-16-148413-1', 'Serie 4', 'Distopía');
```

```
INSERT INTO LIBRO (ID_ARTICULO, ISBN, SERIE, GENERO) VALUES (5,
'978-3-16-148414-8', 'Serie 5', 'Romance');
```

```
INSERT INTO REVISTA (ID_ARTICULO, ISSN, NUM_REVISTA) VALUES (1, '1234-5678',
'Revista 1');
```

```
INSERT INTO REVISTA (ID_ARTICULO, ISSN, NUM_REVISTA) VALUES (2, '2345-6789',
'Revista 2');
```

```
INSERT INTO REVISTA (ID_ARTICULO, ISSN, NUM_REVISTA) VALUES (3, '3456-7890',
'Revista 3');
```

```
INSERT INTO REVISTA (ID_ARTICULO, ISSN, NUM_REVISTA) VALUES (4, '4567-8901',
'Revista 4');
```

```
INSERT INTO REVISTA (ID_ARTICULO, ISSN, NUM_REVISTA) VALUES (5, '5678-9012',
'Revista 5');
```

```
INSERT INTO EJEMPLAR (ID_EJEMPLAR, ID_ARTICULO, STOCK, FORMATO) VALUES (1, 1,
10, 'TAPA DURA');
```

```
INSERT INTO EJEMPLAR (ID_EJEMPLAR, ID_ARTICULO, STOCK, FORMATO) VALUES (2, 2,
5, 'TAPA BLANDA');
```

```
INSERT INTO EJEMPLAR (ID_EJEMPLAR, ID_ARTICULO, STOCK, FORMATO) VALUES (3, 3,
15, 'BOLSILLO');
```

INSERT INTO EJEMPLAR (ID_EJEMPLAR, ID_ARTICULO, STOCK, FORMATO) VALUES (4, 4, 7, 'TAPA DURA');

INSERT INTO EJEMPLAR (ID_EJEMPLAR, ID_ARTICULO, STOCK, FORMATO) VALUES (5, 5, 20, 'TAPA BLANDA');

INSERT INTO COMPRA (DNI, ID_COMPRA, ID_EJEMPLAR, ID_ARTICULO, FECHA_VENTA, IMPORTE, CANTIDAD) VALUES ('12345678A', 1, 1, 1, TO_DATE('2023-10-01', 'YYYY-MM-DD'), 15.99, 1);

INSERT INTO COMPRA (DNI, ID_COMPRA, ID_EJEMPLAR, ID_ARTICULO, FECHA_VENTA, IMPORTE, CANTIDAD) VALUES ('23456789B', 2, 2, 2, TO_DATE('2023-10-02', 'YYYY-MM-DD'), 22.50, 1);

INSERT INTO COMPRA (DNI, ID_COMPRA, ID_EJEMPLAR, ID_ARTICULO, FECHA_VENTA, IMPORTE, CANTIDAD) VALUES ('34567890C', 3, 3, 3, TO_DATE('2023-10-03', 'YYYY-MM-DD'), 30.00, 2);

INSERT INTO COMPRA (DNI, ID_COMPRA, ID_EJEMPLAR, ID_ARTICULO, FECHA_VENTA, IMPORTE, CANTIDAD) VALUES ('45678901D', 4, 4, 4, TO_DATE('2023-10-04', 'YYYY-MM-DD'), 12.75, 3);

INSERT INTO COMPRA (DNI, ID_COMPRA, ID_EJEMPLAR, ID_ARTICULO, FECHA_VENTA, IMPORTE, CANTIDAD) VALUES ('56789012E', 5, 5, 5, TO_DATE('2023-10-05', 'YYYY-MM-DD'), 18.20, 1);

INSERT INTO AUTOR (ID_AUTOR, NOMBRE, CORREO, FECHA_NACIMIENTO, NACIONALIDAD, DESCRIPCION) VALUES (1, 'Gabriel García Márquez', 'garcia.m@autor.com', TO_DATE('1927-03-06', 'YYYY-MM-DD'), 'Colombiano', 'Novelista y escritor.');

INSERT INTO AUTOR (ID_AUTOR, NOMBRE, CORREO, FECHA_NACIMIENTO, NACIONALIDAD, DESCRIPCION) VALUES (2, 'Jane Austen', 'jane.a@autor.com', TO_DATE('1775-12-16', 'YYYY-MM-DD'), 'Británica', 'Escritora y novelista.');

INSERT INTO AUTOR (ID_AUTOR, NOMBRE, CORREO, FECHA_NACIMIENTO, NACIONALIDAD, DESCRIPCION) VALUES (3, 'George Orwell', 'orwell.g@autor.com', TO_DATE('1903-06-25', 'YYYY-MM-DD'), 'Británico', 'Novelista y periodista.');

INSERT INTO AUTOR (ID_AUTOR, NOMBRE, CORREO, FECHA_NACIMIENTO, NACIONALIDAD, DESCRIPCION) VALUES (4, 'Mark Twain', 'twain.m@autor.com', TO_DATE('1835-11-30', 'YYYY-MM-DD'), 'Estadounidense', 'Escritor y humorista.');

INSERT INTO AUTOR (ID_AUTOR, NOMBRE, CORREO, FECHA_NACIMIENTO, NACIONALIDAD, DESCRIPCION) VALUES (5, 'J.K. Rowling', 'rowling.j@autor.com', TO_DATE('1965-07-31', 'YYYY-MM-DD'), 'Británica', 'Escritora de fantasía.');

```
INSERT INTO EVENTO (ID_EVENTO, NOMBRE, DESCRIPCION, FECHA_EVENTO) VALUES
(1, 'Feria del Libro', 'Una feria dedicada a la literatura.', TO_DATE('2023-11-01',
'YYYY-MM-DD'));
```

```
INSERT INTO EVENTO (ID_EVENTO, NOMBRE, DESCRIPCION, FECHA_EVENTO) VALUES
(2, 'Club de Lectura', 'Reunión mensual de lectores.', TO_DATE('2023-11-15',
'YYYY-MM-DD'));
```

```
INSERT INTO EVENTO (ID_EVENTO, NOMBRE, DESCRIPCION, FECHA_EVENTO) VALUES
(3, 'Taller de Escritura', 'Taller para mejorar habilidades de escritura.',
TO_DATE('2023-12-01', 'YYYY-MM-DD'));
```

```
INSERT INTO EVENTO (ID_EVENTO, NOMBRE, DESCRIPCION, FECHA_EVENTO) VALUES
(4, 'Firma de Libros', 'Evento donde los autores firman sus libros.', TO_DATE('2023-12-15',
'YYYY-MM-DD'));
```

```
INSERT INTO EVENTO (ID_EVENTO, NOMBRE, DESCRIPCION, FECHA_EVENTO) VALUES
(5, 'Charla con Autores', 'Conversaciones con escritores contemporáneos.',
TO_DATE('2024-01-10', 'YYYY-MM-DD'));
```

```
INSERT INTO FIRMA (ID_EVENTO, ID_FIRMA, ID_AUTOR) VALUES (4, 1, 1);
```

```
INSERT INTO FIRMA (ID_EVENTO, ID_FIRMA, ID_AUTOR) VALUES (4, 2, 2);
```

```
INSERT INTO FIRMA (ID_EVENTO, ID_FIRMA, ID_AUTOR) VALUES (4, 3, 3);
```

```
INSERT INTO FIRMA (ID_EVENTO, ID_FIRMA, ID_AUTOR) VALUES (4, 4, 4);
```

```
INSERT INTO FIRMA (ID_EVENTO, ID_FIRMA, ID_AUTOR) VALUES (4, 5, 5);
```

```
INSERT INTO CLUB_LECTURA (ID_EVENTO, ID_LECTURA) VALUES (2, 1);
```

```
INSERT INTO CLUB_LECTURA (ID_EVENTO, ID_LECTURA) VALUES (2, 2);
```

```
INSERT INTO CLUB_LECTURA (ID_EVENTO, ID_LECTURA) VALUES (2, 3);
```

```
INSERT INTO CLUB_LECTURA (ID_EVENTO, ID_LECTURA) VALUES (2, 4);
```

```
INSERT INTO CLUB_LECTURA (ID_EVENTO, ID_LECTURA) VALUES (2, 5);
```

INSERT INTO TALLER_ESCRITURA (ID_EVENTO, ID_ESCRITURA) VALUES (3, 1);
INSERT INTO TALLER_ESCRITURA (ID_EVENTO, ID_ESCRITURA) VALUES (3, 2);
INSERT INTO TALLER_ESCRITURA (ID_EVENTO, ID_ESCRITURA) VALUES (3, 3);
INSERT INTO TALLER_ESCRITURA (ID_EVENTO, ID_ESCRITURA) VALUES (3, 4);
INSERT INTO TALLER_ESCRITURA (ID_EVENTO, ID_ESCRITURA) VALUES (3, 5);

INSERT INTO RESERVA (DNI, ID_ARTICULO, FECHA_RESERVA, CANTIDAD) VALUES ('12345678A', 1, TO_DATE('2023-10-10', 'YYYY-MM-DD'), 2);

INSERT INTO RESERVA (DNI, ID_ARTICULO, FECHA_RESERVA, CANTIDAD) VALUES ('23456789B', 2, TO_DATE('2023-10-11', 'YYYY-MM-DD'), 1);

INSERT INTO RESERVA (DNI, ID_ARTICULO, FECHA_RESERVA, CANTIDAD) VALUES ('34567890C', 3, TO_DATE('2023-10-12', 'YYYY-MM-DD'), 3);

INSERT INTO RESERVA (DNI, ID_ARTICULO, FECHA_RESERVA, CANTIDAD) VALUES ('45678901D', 4, TO_DATE('2023-10-13', 'YYYY-MM-DD'), 1);

INSERT INTO RESERVA (DNI, ID_ARTICULO, FECHA_RESERVA, CANTIDAD) VALUES ('56789012E', 5, TO_DATE('2023-10-14', 'YYYY-MM-DD'), 4);

INSERT INTO ORGANIZA (DNI, ID_EVENTO) VALUES ('12345678A', 1);

INSERT INTO ORGANIZA (DNI, ID_EVENTO) VALUES ('23456789B', 2);

INSERT INTO ORGANIZA (DNI, ID_EVENTO) VALUES ('34567890C', 3);

INSERT INTO ORGANIZA (DNI, ID_EVENTO) VALUES ('45678901D', 4);

INSERT INTO ORGANIZA (DNI, ID_EVENTO) VALUES ('56789012E', 5);

INSERT INTO ESCRIBE (ID_ARTICULO, ID_AUTOR) VALUES (3, 3);

INSERT INTO ESCRIBE (ID_ARTICULO, ID_AUTOR) VALUES (4, 4);

INSERT INTO ESCRIBE (ID_ARTICULO, ID_AUTOR) VALUES (5, 5);

SENTENCIAS SQL DE COMPROBACIÓN

*/** Consultas de comprobación **/*

SELECT * FROM USUARIO;

SELECT * FROM USUARIO_DIRECCIONES;

SELECT * FROM EMPLEADO;

SELECT * FROM SOCIO;

SELECT * FROM VIP;

SELECT * FROM ARTICULO;

SELECT * FROM LIBRO;

SELECT * FROM REVISTA;

SELECT * FROM EJEMPLAR;

SELECT * FROM COMPRA;

SELECT * FROM AUTOR;

SELECT * FROM EVENTO;

SELECT * FROM FIRMA;

SELECT * FROM CLUB_LECTURA;

SELECT * FROM TALLER_ESCRITURA;

SELECT * FROM RESERVA;

SELECT * FROM ORGANIZA;

SELECT * FROM ESCRIBE;

SELECT * FROM ARTICULOS_DISPONIBLES;

SELECT * FROM ARTICULOS_POR_AUTOR;

SELECT * FROM ARTICULOS_POR_PRECIO;

HOJA DE FIRMAS

DNI: 49705010D

NOMBRE: ALONSO IGLESIAS, VICTOR

DNI: 76736578E

NOMBRE: BLANCO NESPEREIRA, CLAUDIA YU

DNI: 54233422C

NOMBRE: CANCELAS GARCIA, DANIEL

DNI: 44661806S

NOMBRE: CARBAJO PAULA, JUAN ANTONIO