

# Pràctica



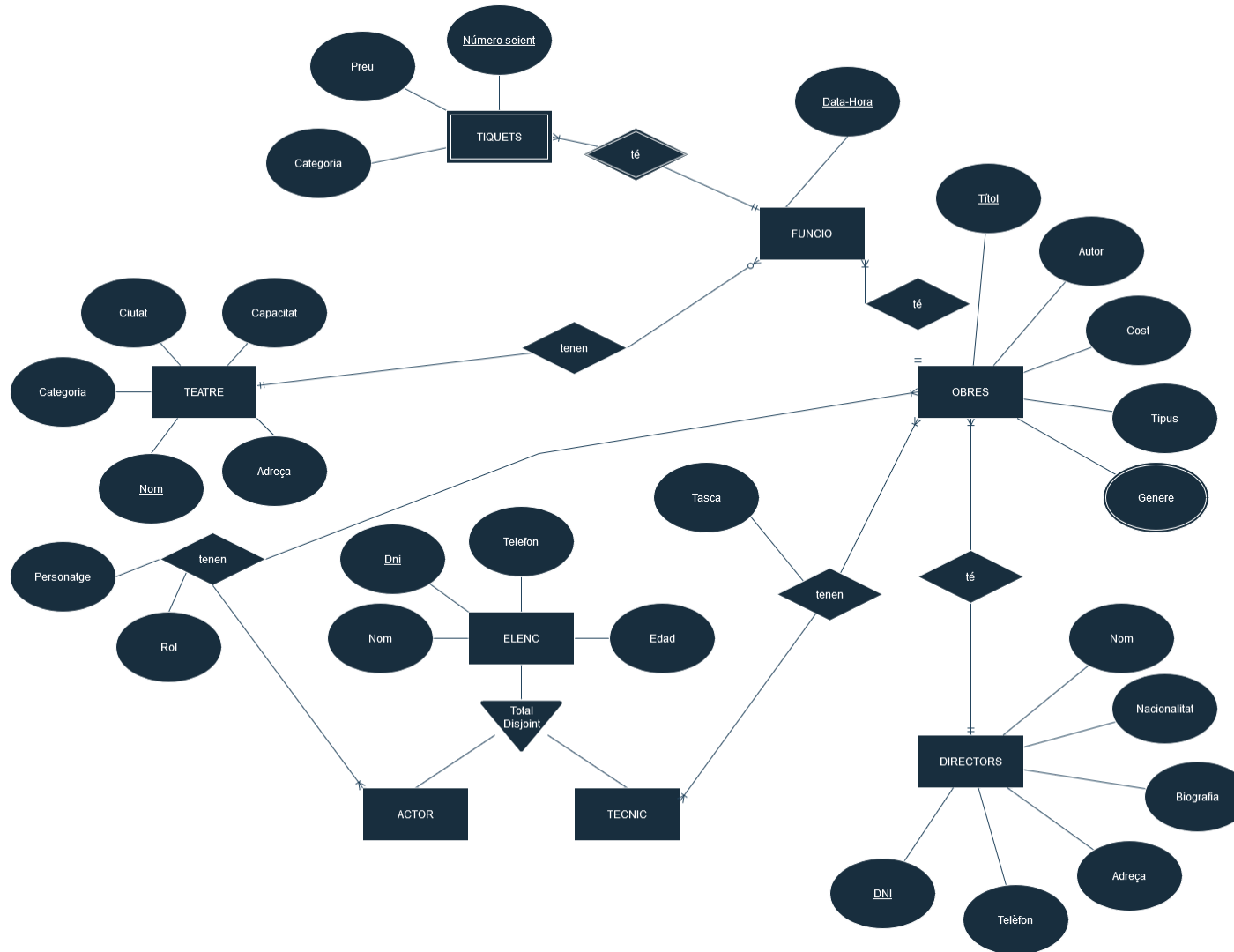
Joan Sánchez Sastre

**Mòdul:** Base de Dades

# Índex:

<b>1. Model Entitat-Relació.....</b>	<b>2</b>
<b>2. Model Relacional.....</b>	<b>3</b>
<b>3. Model físic en MySQL.....</b>	<b>4</b>
Teatres (Base de dades).....	4
Teatre.....	4
Directors.....	4
Obres.....	4
Obres_genere.....	5
Funcio.....	5
Tiquets.....	5
Elenc.....	5
Actor.....	6
Tecnic.....	6
Obra_Actor.....	6
Obra_Tecnic.....	6
<b>4. Model Relacional (Intelj i Workbench).....</b>	<b>7</b>
<b>5. Inserts.....</b>	<b>8</b>
Teatre.....	8
Directors.....	8
Obres.....	9
Obres_genere.....	9
Funcio.....	10
Tiquets.....	10
Elenc.....	11
Actor.....	11
Tecnic.....	12
Obra_Actor.....	12
Obra_Tecnic.....	12
<b>6. Consultes.....</b>	<b>13</b>
<b>7. Vista oferta_teatral.....</b>	<b>16</b>
<b>8. Funció ROL.....</b>	<b>17</b>
<b>9. Funció MILLOR.....</b>	<b>18</b>
<b>10. Procediment GENERES.....</b>	<b>19</b>
<b>11. Procediment PARTICIPANTS.....</b>	<b>20</b>

## 1. Model Entitat-Relació



## 2. Model Relacional

**TEATRE**(Nom, Adreça, Ciutat, Capacitat, Categoria)

**FUNCIO**(DataHora, Teatre, Obra)

**ON** Teatre REFERENCIA TEATRE(Nom),

**ON** Obra REFERENCIA OBRES(Titol)

**TIQUETS**(DataHora, Número, seient, Preu, Categoria)

**ON** Data REFERENCIA FUNCIO(DataHora

**OBRES**(Titol, Autor, Cost, Tipus, Director)

**ON** Director REFERENCIA DIRECTORS(DNI)

**OBRES\_GENERES**(Obra, Genere)

**ON** Obra REFERENCIA OBRES(Titol)

**DIRECTORS**(DNI, Nom, Nacionalitat, Biografia, Telèfon, Adreça)

**ELENC**(DNI, Nom, Telèfon, Edad)

**ACTOR**(DNI)

**ON** DNI REFERENCIA ELENC(DNI)

**TECNIC**(DNI)

**ON** DNI REFERENCIA ELENC(DNI)

**OBRA\_ACTOR**(DNI, Obra, Rol, Personatge)

**ON** DNI REFERENCIA ACTOR(DNI),

**ON** Obra REFERENCIA OBRES(Titol)

**OBRA\_Tecnic**(DNI, Obra, Tasca)

**ON** DNI REFERENCIA TECNIC(DNI),

**ON** Obra REFERENCIA OBRES(Titol)

### 3. Model físic en MySQL

#### Teatres (Base de dades)

```
drop database if exists teatres;  
Create database teatres;  
use teatres;
```

#### Teatre

```
create table TEATRE(  
    Nom varchar(30) PRIMARY KEY,  
    Adreça varchar(100) not null,  
    Ciutat VARCHAR(50) not null,  
    Capacitat int not null,  
    Categoria varchar(15) not null,  
    constraint check (Categoria in ('De luxe', 'Gran', 'Mitja',  
'Petit'))  
);
```

#### Directors

```
CREATE TABLE DIRECTORS(  
    DNI VARCHAR(9) PRIMARY KEY UNIQUE,  
    Nom VARCHAR(30) not null,  
    Nacionalitat VARCHAR(30),  
    Biografia VARCHAR(200),  
    Telefon INT not null,  
    Adreça VARCHAR(100) not null  
);
```

#### Obres

```
CREATE TABLE OBRES(  
    Titol varchar(30) primary key,  
    Autor varchar(30) not null,  
    Cost double not null,  
    Tipus varchar(15) not null,  
    Director varchar(9) not null,  
    foreign key (Director) REFERENCES DIRECTORS(DNI),  
    constraint check(Tipus in ('Clàssic', 'Modern', 'Absurd', 'Negre'))  
);
```

## Obres\_genere

```
CREATE TABLE OBRES_GENERE (  
    Obra varchar(30),  
    Genere varchar(50),  
    primary key (Obra, Genere),  
    foreign key (Obra) REFERENCES OBRES(Titol)  
);
```

## Funcio

```
CREATE TABLE FUNCIO(  
    DataHora DATETIME,  
    Teatre VARCHAR(30),  
    Obra VARCHAR(30),  
    PRIMARY KEY (DataHora, Teatre, Obra),  
    FOREIGN KEY (Teatre) REFERENCES TEATRE(Nom),  
    FOREIGN KEY (Obra) REFERENCES OBRES(Titol)  
);
```

## Tiquets

```
CREATE TABLE TIQUETS(  
    DataHora DATETIME,  
    Numero_seient INT,  
    Preu DOUBLE NOT NULL,  
    Categoria VARCHAR(30) NOT NULL,  
    PRIMARY KEY (DataHora, Numero_seient),  
    CONSTRAINT CHECK (Categoria IN ('Llotja', 'Platea', 'Amfiteatre')),  
    FOREIGN KEY (DataHora) REFERENCES FUNCIO(DataHora)  
);
```

## Elenc

```
CREATE TABLE ELENC(  
    DNI VARCHAR(9) PRIMARY KEY UNIQUE,  
    Nom VARCHAR(30) not null,  
    Telefon INT not null,  
    Edat INT  
);
```

## Actor

```
CREATE TABLE ACTOR(  
    DNI VARCHAR(9) PRIMARY KEY UNIQUE,  
    FOREIGN KEY (DNI) REFERENCES ELENC(DNI)  
);
```

## Tecnic

```
CREATE TABLE TECNIC(  
    DNI VARCHAR(9) PRIMARY KEY UNIQUE,  
    FOREIGN KEY (DNI) REFERENCES ELENC(DNI)  
);
```

## Obra\_Actor

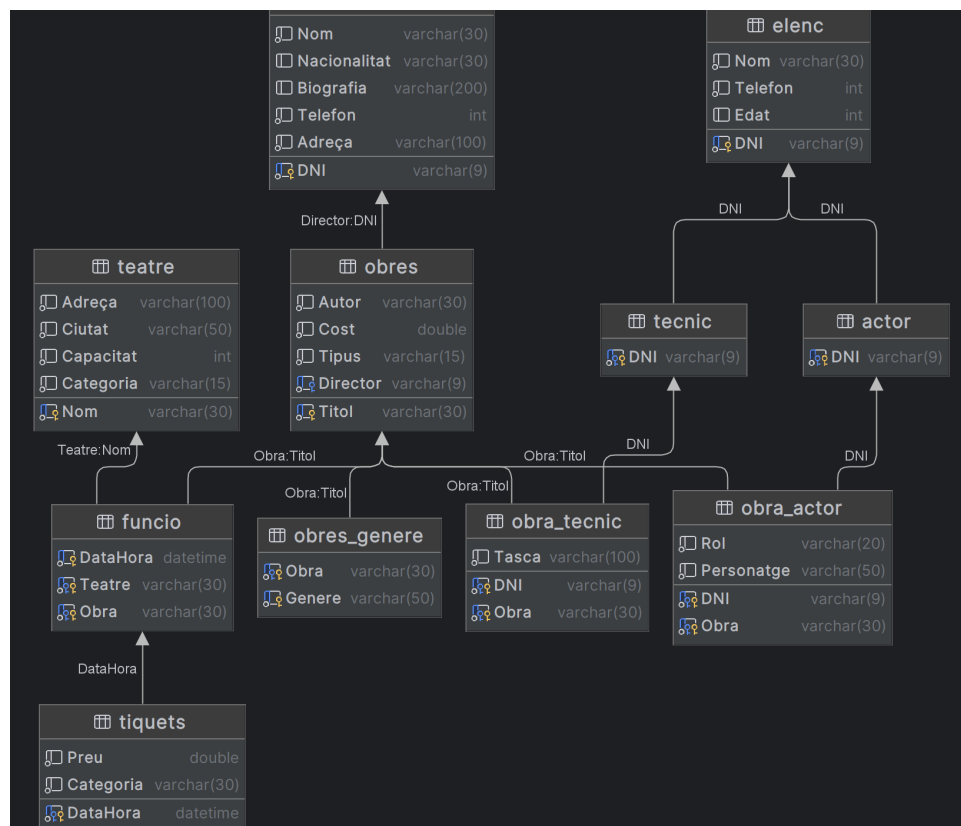
```
CREATE TABLE OBRA_ACTOR(  
    DNI VARCHAR(9),  
    Obra VARCHAR(30),  
    Rol VARCHAR(20) NOT NULL,  
    Personatge VARCHAR(50) NOT NULL,  
    PRIMARY KEY (DNI, Obra),  
    FOREIGN KEY (DNI) REFERENCES ACTOR(DNI),  
    FOREIGN KEY (Obra) REFERENCES OBRES(Titol),  
    CONSTRAINT CHECK ( Rol IN ('Principal', 'Secundari', 'Extra'))  
);
```

## Obra\_Tecnic

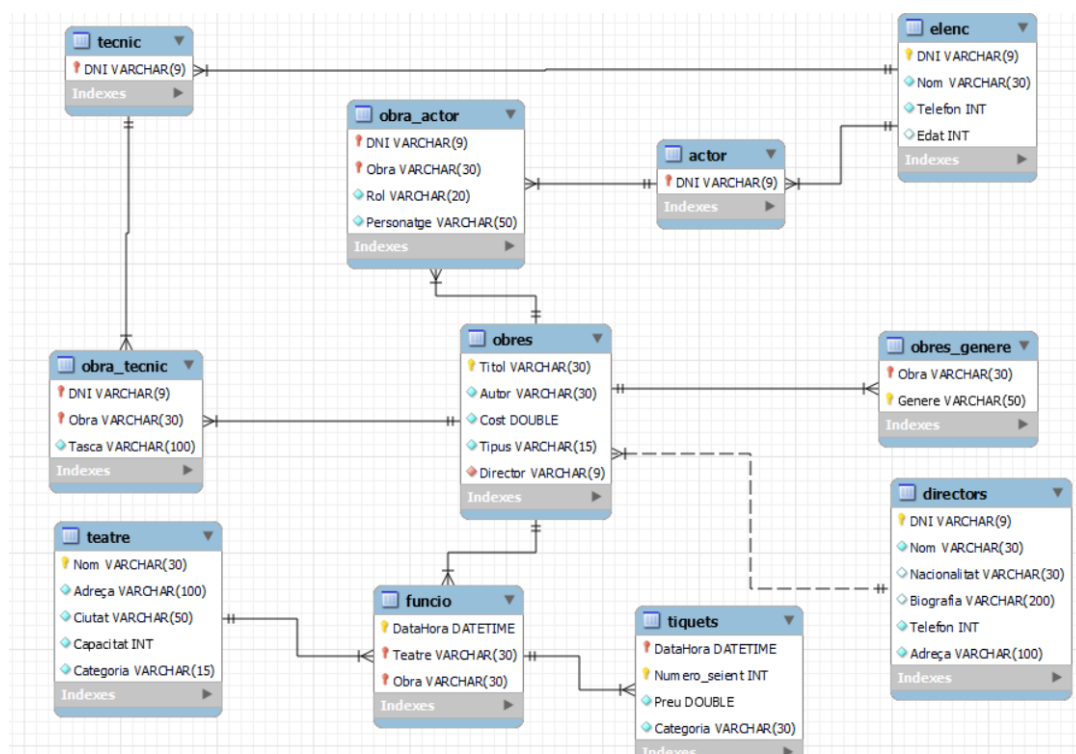
```
CREATE TABLE OBRA_TECNIC(  
    DNI VARCHAR(9),  
    Obra VARCHAR(30),  
    Tasca VARCHAR(100) NOT NULL,  
    PRIMARY KEY (DNI, Obra),  
    FOREIGN KEY (DNI) REFERENCES TECNIC(DNI),  
    FOREIGN KEY (Obra) REFERENCES OBRES(Titol)  
);
```

## 4. Model Relacional (Intelj i Workbench)

Amb **Intelj**:



Amb **Workbench**:





## 5. Inserts

### Teatre

```
INSERT INTO TEATRE VALUES
('Teatre Lliure', 'Carrer de Montseny, 47', 'Barcelona', 500, 'Gran'),
('Teatre Nacional de Catalunya', 'Plaça de les Arts, 1', 'Barcelona',
900, 'De luxe'),
('Sala Beckett', 'Carrer de Pere IV, 228-232', 'Barcelona', 100,
'Petit'),
('Teatre Apolo', 'Carrer de Nou de la Rambla, 113', 'Barcelona', 1500,
'Gran'),
('Teatre Grec', 'Passeig de Santa Madrona, 36', 'Barcelona', 2000,
'Gran'),
('Teatre Poliorama', 'La Rambla, 115', 'Barcelona', 700, 'Mitja'),
('Teatre Tívoli', 'Carrer de Casp, 8', 'Barcelona', 1600, 'Gran'),
('Teatre Romea', 'Carrer Hospital, 51', 'Barcelona', 550, 'Mitja'),
('Mercat de les Flors', 'Carrer de Lleida, 59', 'Barcelona', 500,
'Gran'),
('Teatre Coliseum', 'Gran Via de les Corts Catalanes, 595',
'Barcelona', 2000, 'De luxe');
```

### Directors

```
INSERT INTO DIRECTORS VALUES
('12345678A', 'Pedro Almodóvar', 'Española', 'Director de cine y
guionista', 931234567, 'Calle Almagro, 28, Madrid'),
('23456789B', 'Guillermo del Toro', null, 'Director de cine y
escritor', 932345678, 'Calle Acacias, 10, Guadalajara'),
('34567890C', 'Christopher Nolan', 'Británica', 'Director de cine y
guionista', 934567890, 'Avenida Lexington, 115, Los Ángeles'),
('45678901D', 'Sofia Coppola', 'Estadounidense', 'Directora de cine y
guionista', 935678901, 'Beverly Boulevard, 5200, Los Ángeles'),
('56789012E', 'Alejandro González Iñárritu', 'Mexicana', null,
936789012, 'Calle Veracruz, 6, Ciudad de México'),
('67890123F', 'Wes Anderson', 'Estadounidense', 'Director de cine y
guionista', 937890123, 'Broadway, 234, Nueva York'),
('78901234G', 'Tim Burton', 'Estadounidense', 'Director de cine y
productor', 938901234, 'Laurel Canyon Boulevard, 1400, Los Ángeles'),
('89012345H', 'Quentin Tarantino', 'Estadounidense', 'Director de cine,
guionista y actor', 939012345, 'Calle Vista del Mar, 1600, Los
Ángeles'),
```

```
('90123456I', 'Park Chan-wook', 'Coreana', null, 940123456,
'Gangnam-daero, 42-gil, Seúl'),
('01234567J', 'Pedro Costa', null, 'Director de cine', 941234567, 'Rua
do Benfornoso, 120, Lisboa');
```

## Obres

```
INSERT INTO OBRES VALUES
('Hamlet', 'William Shakespeare', 35.50, 'Clàssic', '12345678A'),
('La Casa de Bernarda Alba', 'Federico García Lorca', 25.00, 'Clàssic',
'34567890C'),
('La Cabra o qui és Sylvia?', 'Edward Albee', 30.00, 'Absurd',
'23456789B'),
('Cyrano de Bergerac', 'Edmond Rostand', 28.50, 'Clàssic',
'45678901D'),
('La Rattonera', 'Agatha Christie', 20.00, 'Negre', '56789012E'),
('La Tempesta', 'William Shakespeare', 33.00, 'Clàssic', '67890123F'),
('La Gavina', 'Anton Chekhov', 22.00, 'Clàssic', '01234567J'),
('Qui té por de Virginia Woolf?', 'Edward Albee', 27.00, 'Modern',
'78901234G'),
('Hort dels cirerers', 'Anton Chekhov', 26.50, 'Clàssic', '89012345H'),
('Equus', 'Peter Shaffer', 31.00, 'Modern', '90123456I');
```

## Obres\_genere

```
INSERT INTO OBRES_GENERE VALUES
('Hamlet', 'Tragèdia'),
('Hamlet', 'Drama'),
('La Casa de Bernarda Alba', 'Drama'),
('La Casa de Bernarda Alba', 'Realisme'),
('La Cabra o qui és Sylvia?', 'Drama'),
('Cyrano de Bergerac', 'Comèdia'),
('Cyrano de Bergerac', 'Romàntic'),
('La Rattonera', 'Misteri'),
('La Tempesta', 'Comèdia'),
('La Gavina', 'Drama'),
('Qui té por de Virginia Woolf?', 'Drama'),
('Qui té por de Virginia Woolf?', 'Psicològic'),
('Equus', 'Drama'),
('Equus', 'Psicològic'),
('Equus', 'Realisme psicològic'),
('Hamlet', 'Teatre del Renaixement'),
('Cyrano de Bergerac', 'Teatre de l''absurd');
```

## Funcio

```
INSERT INTO FUNCIO VALUES
('2023-05-01 20:00:00', 'Teatre Lliure', 'Hamlet'),
('2023-05-02 19:00:00', 'Teatre Nacional de Catalunya', 'La Casa de
Bernarda Alba'),
('2023-05-03 21:00:00', 'Sala Beckett', 'La Cabra o qui és Sylvia?'),
('2023-05-04 20:30:00', 'Teatre Apolo', 'Cyrano de Bergerac'),
('2023-05-05 18:00:00', 'Teatre Grec', 'La Rattonera'),
('2023-05-03 20:00:00', 'Teatre Poliorama', 'La Tempesta'),
('2023-05-07 19:30:00', 'Teatre Tívoli', 'La Gavina'),
('2023-05-02 21:00:00', 'Teatre Romea', 'Qui té por de Virginia
Woolf?'),
('2023-05-09 20:00:00', 'Mercat de les Flors', 'Hort dels cirerers'),
('2023-06-01 19:00:00', 'Teatre Coliseum', 'Equus');
```

## Tiquets

```
INSERT INTO TIQUETS VALUES
('2023-05-01 20:00:00', 25, 30.00, 'Platea'),
('2023-05-01 20:00:00', 26, 30.00, 'Platea'),
('2023-05-01 20:00:00', 27, 30.00, 'Platea'),
('2023-05-02 19:00:00', 10, 40.00, 'Llotja'),
('2023-05-02 19:00:00', 11, 40.00, 'Llotja'),
('2023-05-02 19:00:00', 12, 40.00, 'Llotja'),
('2023-05-03 21:00:00', 50, 21.00, 'Amfiteatre'),
('2023-05-03 21:00:00', 51, 80.00, 'Llotja'),
('2023-05-03 21:00:00', 1, 19.00, 'Platea'),
('2023-05-04 20:30:00', 1, 35.00, 'Platea'),
('2023-05-04 20:30:00', 2, 35.00, 'Platea');
```

## Elenc

```
INSERT INTO ELENC VALUES
('11111111A', 'Emma Thompson', 931111111, null),
('22222222B', 'Tom Hanks', 932222222, 65),
('33333333C', 'Meryl Streep', 933333333, 72),
('44444444D', 'Leonardo DiCaprio', 934444444, null),
('55555555E', 'Jodie Foster', 935555555, 59),
('66666666F', 'Robert De Niro', 936666666, 78),
('77777777G', 'Kate Winslet', 937777777, 46),
('88888888H', 'Anthony Hopkins', 938888888, 84),
('99999999I', 'Julia Roberts', 939999999, 55),
('00000000J', 'Denzel Washington', 940000000, 67),
('12345678A', 'John Smith', 931234567, 35),
('23456789B', 'Jane Doe', 932345678, 28),
('34567890C', 'Michael Johnson', 933456789, 42),
('45678901D', 'Lisa Rodriguez', 934567890, null),
('56789012E', 'David Lee', 935678901, 47),
('67890123F', 'Sarah Kim', 936789012, 31),
('78901234G', 'Daniel Garcia', 937890123, null),
('89012345H', 'Melissa Davis', 938901234, 22),
('90123456I', 'Kevin Hernandez', 939012345, 43),
('01234567J', 'Stephanie Brown', 940123456, 29);
```

## Actor

```
INSERT INTO ACTOR VALUES
('11111111A'),
('22222222B'),
('33333333C'),
('44444444D'),
('55555555E'),
('66666666F'),
('77777777G'),
('88888888H'),
('99999999I'),
('00000000J');
```

## Tecnic

```
INSERT INTO TECNIC VALUES
('12345678A'),
('23456789B'),
('34567890C'),
('45678901D'),
('56789012E'),
('67890123F'),
('78901234G'),
('89012345H'),
('90123456I'),
('01234567J');
```

## Obra\_Actor

```
INSERT INTO OBRA_ACTOR VALUES
('11111111A', 'Hamlet', 'Principal', 'Elinor Dashwood'),
('22222222B', 'La Casa de Bernarda Alba', 'Principal', 'Forrest Gump'),
('33333333C', 'La Cabra o qui és Sylvia?', 'Principal', 'Kay Graham'),
('44444444D', 'Cyrano de Bergerac', 'Principal', 'Jordan Belfort'),
('55555555E', 'La Ratonera', 'Secundari', 'Clarice Starling'),
('66666666F', 'La Tempesta', 'Secundari', 'Vito Corleone'),
('77777777G', 'La Gavina', 'Secundari', 'Rose DeWitt Bukater'),
('88888888H', 'Qui té por de Virginia Woolf?', 'Extra', 'Dr. Hannibal
Lecter'),
('99999999I', 'Hort dels cirerers', 'Extra', 'Vivian Ward'),
('00000000J', 'Equus', 'Extra', 'Frank Lucas');
```

## Obra\_Tecnic

```
INSERT INTO OBRA_TECNIC VALUES
('12345678A', 'La Cabra o qui és Sylvia?', 'Disseny de vestuari'),
('23456789B', 'La Casa de Bernarda Alba', 'Il·luminació'),
('34567890C', 'Hamlet', 'So'),
('45678901D', 'Cyrano de Bergerac', 'Maquillatge'),
('56789012E', 'La Ratonera', 'Construcció escenografia'),
('67890123F', 'La Tempesta', 'Propulsor'),
('78901234G', 'La Gavina', 'Regidoria'),
('89012345H', 'Qui té por de Virginia Woolf?', 'Direcció
artística'),
('90123456I', 'Hort dels cirerers', 'Coreografia'),
('01234567J', 'Equus', 'Assistent de producció');
```

## 6. Consultes

1. Mostra els actors que han interpretat un personatge principal a una obra de gènere 'Drama' mostra també l'obra i el director corresponent.

```
SELECT a.DNI, o.Titol, d.Nom AS Director
FROM ACTOR a
JOIN OBRA_ACTOR oa ON a.DNI = oa.DNI
JOIN OBRES o ON oa.Obra = o.Titol
JOIN OBRES_GENERE og ON o.Titol = og.Obra
JOIN DIRECTORS d ON o.Director = d.DNI
WHERE oa.Rol = 'Principal' AND og.Genere = 'Drama';
```

2. Quin director ha dirigit més obres?

```
SELECT d.Nom, COUNT(o.Titol) AS "Número d'obres"
FROM DIRECTORS d
JOIN OBRES o ON d.DNI = o.Director
GROUP BY d.DNI
HAVING COUNT(o.Titol) = (SELECT COUNT(*) FROM OBRES GROUP BY
Director ORDER BY COUNT(*) DESC LIMIT 1);
```

3. Quantes obres hi ha de cada categoria de teatre?

```
SELECT t.Categoria, COUNT(*) AS "Número de obres"
FROM TEATRE t
JOIN FUNCIO f ON t.Nom = f.Teatre
JOIN OBRES o ON f.Obra = o.Titol
GROUP BY t.Categoria;
```

4. Mostra els actors i a quantes obres han treballat de manera descendent.

```
SELECT e.Nom, COUNT(*) AS "Número d'obres"
FROM ELENC e
JOIN OBRA_ACTOR oa ON e.DNI = oa.DNI
GROUP BY e.DNI
ORDER BY COUNT(*) DESC;
```

5. Quines obres es representen en el teatre 'Teatre Nacional de Catalunya' en la pròxima setmana tinguent en compte que ens trobem el dia 05/05/2023?

```
SELECT f.Obra, f.DataHora
FROM FUNCIO f
JOIN TEATRE t ON f.Teatre = t.Nom
WHERE t.Nom = 'Teatre Nacional de Catalunya'
AND f.DataHora BETWEEN DATE('2023-05-05') AND DATE('2023-05-05') +
INTERVAL 1 WEEK;
```

6. Llista d'obres i directors ordenats per ciutat.

```
SELECT t.Ciutat, o.Titol, d.Nom AS Director
FROM TEATRE t
JOIN FUNCIO f ON t.Nom = f.Teatre
JOIN OBRES o ON f.Obra = o.Titol
JOIN DIRECTORS d ON o.Director = d.DNI
ORDER BY t.Ciutat;
```

7. Quines obres han generat més beneficis? mostra també el cost d'aquestes i els beneficis nets obtinguts.

```
SELECT o.Titol, o.Cost AS "Cost de l'obra", SUM(t.Preu) AS
"Ingressos", SUM(t.Preu) - o.Cost AS "Beneficis nets"
FROM OBRES o
JOIN FUNCIO f ON o.Titol = f.Obra
JOIN TIQUETS t ON f.DataHora = t.DataHora
GROUP BY o.Titol
HAVING SUM(t.Preu) - o.Cost = (SELECT SUM(t.Preu) - o.Cost FROM
OBRES o
                                JOIN FUNCIO f ON o.Titol = f.Obra
                                JOIN TIQUETS t ON f.DataHora =
t.DataHora
                                GROUP BY o.Titol
                                ORDER BY SUM(t.Preu) - o.Cost DESC
                                LIMIT 1
);
```

8. Mostra els teatres on s'han representat obres dirigides per un director espanyol, indicant la data, l'hora i el nombre d'entrades venudes de cada funció.

```
SELECT f.Teatre, f.DataHora, COUNT(t.Numero_seient) AS "Nombre  
d'entrades venudes"  
FROM FUNCIO f  
JOIN OBRES o ON f.Obra = o.Titol  
JOIN DIRECTORS d ON o.Director = d.DNI  
LEFT JOIN TIQUETS t ON f.DataHora = t.DataHora  
WHERE d.Nacionalitat = 'Espanyola'  
GROUP BY f.Teatre, f.DataHora;
```

9. Quines obres no han estat representades al teatre 'Teatre Lliure', i el nom del seu director?

```
SELECT o.Titol, d.Nom AS "Nom del director", o.Cost  
FROM OBRES o  
JOIN DIRECTORS d ON o.Director = d.DNI  
WHERE o.Titol NOT IN (SELECT f.Obra FROM FUNCIO f WHERE f.Teatre =  
'Teatre Lliure');
```

10. Llista d'actors i personatges que han participat a l'obra 'Hamlet'.

```
SELECT e.Nom AS "Nom de l'actor", oa.Personatge  
FROM ELENC e  
JOIN OBRA_ACTOR oa ON e.DNI = oa.DNI  
JOIN OBRES o ON oa.Obra = o.Titol  
WHERE o.Titol = 'Hamlet';
```



## 7. Vista oferta\_teatral

```
create or replace view ofetra_teatral as
select Titol, d.Nom as 'Director', group_concat(OG.Genere) as
'Genere', Tipus, (select group_concat(T.Ciutat)

from FUNCIO

join OBRES O on O.Titol = FUNCIO.Obra

join TEATRE T on T.Nom = FUNCIO.Teatre

where o.Titol = OBRES.Titol) as 'Ciutats', CONCAT(Cost, '€') as
'Cost'
from obres
    join DIRECTORS D on D.DNI = OBRES.Director
    join OBRES_GENERE OG on OBRES.Titol = OG.Obra
    join funcio f on OG.Obra = f.Obra
where year(f.DataHora) = year(curdate())
group by Titol;
```

## 8. Funció ROL

```
create function ROL(nomActor varchar(30), titolObra varchar(30))
returns varchar(20)
reads sql data
begin
    if not exists(select * from obra_actor where Obra = titolObra
and DNI = (select DNI from elenc where Nom = nomActor)) then
        return '***';
    end if;
    return (select Rol from obra_actor where Obra = titolObra and
DNI = (select DNI from elenc where Nom = nomActor));
end;
```

## 9. Funció MILLOR

```

create function MILLOR() returns text
reads sql data
begin
    declare result text default '';
    declare fin boolean default false;
    declare titolObra varchar(30);

    declare myCursor cursor for (SELECT o.Titol
                                FROM OBRES o
                                JOIN FUNCIO f ON o.Titol = f.Obra
                                JOIN TIQUETS t ON f.DataHora =
t.DataHora
                                GROUP BY o.Titol
                                HAVING SUM(t.Preu)= (SELECT
SUM(t.Preu) FROM OBRES o
                                JOIN FUNCIO f ON o.Titol = f.Obra
                                JOIN TIQUETS t ON f.DataHora = t.DataHora
                                GROUP BY o.Titol
                                ORDER BY SUM(t.Preu) DESC
                                LIMIT
1
                                ));
    declare continue handler for not found set fin = true;
    open myCursor;
    bucle: loop
        fetch myCursor into titolObra;
        if fin = true then
            leave bucle;
        end if;
        set result = concat(result, titolObra, '\n');
    end loop;
    close myCursor;
    return result;
end;

```

## 10. Procediment GENERES

```
create procedure GENERES()  
reads sql data  
begin  
    select Genere, count(Obra) from obres_genere group by Genere  
order by count(*) desc;  
end;
```

## 11. Procediment PARTICIPANTS

```
create procedure PARTICIPANTS(idObra varchar(30), llista char(1))
reads sql data
begin
    if llista = 't' then
        select E.Nom, E.Edat, OT.Tasca
        from obra_tecnic OT
            join TECNIC T on T.DNI = OT.DNI
            JOIN ELENC E on E.DNI = T.DNI
        where Obra = idObra
        order by E.Nom asc, OT.Tasca asc;

    elseif llista = 'a' then
        SELECT E.Nom, E.Edat, OA.Personatge, OA.Rol
        FROM obra_actor OA
        JOIN ACTOR A ON A.DNI = OA.DNI
        JOIN ELENC E ON E.DNI = A.DNI
        where Obra = idObra
        ORDER BY
            CASE OA.Rol
                WHEN 'Principal' THEN 1
                WHEN 'Secundari' THEN 2
                WHEN 'Extra' THEN 3
                ELSE 4
            END;

    else
        select 'Second parameter not valid';
    end if;
end;
```