



Modelat conceptual, modelat relacional i SQL

# PRIMERA PRÀCTICA

Harpo Joan Alberola

21716 – Base de Dades I, grup 1



## Índex

<b>INTRODUCCIÓ I ANÀLISI DEL PROBLEMA .....</b>	<b>2</b>
<b>MODELAT CONCEPTUAL I JUSTIFICACIÓ.....</b>	<b>2</b>
<b>TRADUCCIÓ A RELACIONAL.....</b>	<b>6</b>
SELECCIÓ DE CLAUS PRIMÀRIES DE LES TAULES .....	6
SELECCIÓ DE LES CLAUS PRIMÀRIES DE LES RELACIONS .....	7
FUSIÓ DE LES TAULES I RELACIONS .....	8
<b>IMPLEMENTACIÓ EN CODI SQL .....</b>	<b>9</b>
<b>CONCLUSIONS .....</b>	<b>13</b>
<b>REFERÈNCIES BIBLIOGRÀFIQUES I RECURSOS.....</b>	<b>13</b>

## Introducció i anàlisi del problema

El desenvolupament d'una base de dades que emmagatzemi de forma eficaç i eficient la informació necessària per a tenir un registre fiable i consistent requereix d'una bona reflexió de totes les fases involucrades durant el disseny i implementació. És per això que al llarg d'aquesta memòria s'exposen de manera concisa la problemàtica de la situació presentada; les conclusions a les que s'han arribat per a generar un primer model conceptual que il·lustri i representi les relacions entre distintes entitats que ens interessa manipular; la fase corresponent al modelat relacional fruit de la traducció del diagrama conceptual; i una final implementació a codi SQL.

En primer lloc, és fonamental comprendre perfectament quin és el problema a resoldre i quines són les necessitats que s'han de satisfer. Com s'exposa a l'enunciat, s'ha de generar una base de dades per a una empresa dedicada a la gestió i organització d'esdeveniments culturals i d'oci. La idea principal que es pot extreure sobre què ha de gestionar la base de dades es fonamenta en els següents aspectes:

- Tenir un registre de companyies que produeixen espectacles.
- Registrar les sessions i representacions d'aquests espectacles.
- Controlar els clients i el sistema de reserva d'entrades que adquireixen per cada sessió.

Partint d'aquests tres conceptes, es podrà assolir un major nivell de detall que s'ajusti a les especificacions de l'enunciat, com es descriu en l'apartat de modelat conceptual. Tot i això, és important saber quin és l'objectiu principal de la base de dades, per no centrar-nos en aspectes intranscendents o que ens desvien de la solució del problema.

## Modelat conceptual i justificació

El disseny del model conceptual ha requerit de varies iteracions que han dut al model final. Aquest s'adjunta al final de la secció, de manera que es pot comprovar en tot moment de què s'està parlant. Per a una major claredat, se n'expliquen les seves peculiaritats centrant-se en els tres elements comentats a la introducció anterior. A més, s'indiquen amb **negreta** les entitats que són una classe i en *cursiva* altres aspectes rellevants com els atributs o el nom de relacions si és necessari.

En primer lloc, començaré parlant del concepte d'**espectacle**. Cada spectacle es caracteritza per tenir un *nom\_espectacle*, una *descripció*, una *durada*, un *públic* al que va dirigit i una *classificació\_edat* recomanada. A més, cada spectacle pertany a un **gènere** concret. Una opció era implementar-ho com a una classe per poder obtenir amb major facilitat, si es volgués, tots els espectacles d'un gènere concret si el sistema escala en un futur. En canvi, com s'ha deixat constància que només hi ha 4 gèneres disponibles, la decisió final passa per deixar-ho com a atribut i simplificar el model.

D'altra banda, cada espectacle l'organitza una determinada **companyia**, de la que s'emmagatzema el seu *nom\_companyia* i *disciplina\_artística*. Aquestes companyies contracten a varis **artistes** per a representar un **paper** en els espectacles. El paper és una classe associativa degut a que hem de fer una distinció entre els distints parells d'artista-espectacle. S'ha assumit que només interessa saber el paper d'un artista en un moment determinat, sense haver de mantenir un registre a mode d'historial. D'un artista es té la informació bàsica que hereta d'una classe **persona** (posteriorment veurem perquè és útil aquesta classe pare).

Tenir la informació bàsica relativa als espectacles no és suficient. Hem de poder saber a quin dia i hora es programa una funció, o hauríem de poder reservar un lloc si hi volem assistir. Per això existeix la classe **sessió**, que és l'element central que ens relaciona els distints espectacles amb la gestió d'entrades. Aquesta sessió té un *preu\_base* en funció de l'espectacle i unes dades temporals que la serveixen per programar-la. En funció d'aquesta *data* i *hora*, una sessió es troba en un **estat\_sessió**, el qual és una classe degut a que el seu valor anirà variant. També és important informar en quin idioma es desenvoluparà, o si disposa de subtítols o serveis de traducció. Per a evitar la redundància de dades, s'ha creat la classe **idioma**, que té tres relacions distintes amb la sessió, i on les multiplicitats són importants degut a que ens indiquen si disposem de subtítols/traducció o no. Aquesta és la solució òptima per a enregistrar els distints idiomes, degut a que si s'implementés com a diversos atributs que guarden cada un dels idiomes d'una sessió, la repetició d'elements (redundància) seria considerablement elevada.

Cada sessió es desenvolupa en un determinat **lloc**, que pot ser de diversos *tipus\_lloc*, té una *capacitat* limitada i està situat a una *ubicació*. Degut a aspectes de redundància, aquesta ubicació només contindrà informació local, com és el carrer i codi postal on es situa, i la classe **ciutat** ens indicarà el país concret on es troba. El lloc es divideix en **zones**, la qual té un *preu\_zona* determinat i un atribut booleà que ens indica si està *numerada* o no, a més de variables booleanes que indicaran si es disposa de serveis d'accessibilitat especials o no (*ascensor*, *rampa* i *mobilitat\_reduïda*). No s'ha volgut implementar una classe que encapsuli la informació d'accessibilitat degut a que, podent representar la mateixa informació amb només tres atributs booleans, serà més eficient en termes d'espai de memòria. Dins aquesta zona hi pot haver o no **seients**, en funció de si està numerada, els quals tenen un **tipus** associat que n'indica l'extra corresponent al seu *preu\_seient*. El *número\_seient* del seient (que ens servirà per a identificar-lo), no és només un enter. Amb aquest nombre sabrem a quina zona pertany un seient. Per exemple, si es tracta de distingir entre seients de la zona 1 i 2, s'ha assumit que els seients es numeren 101,102,...,1*n* i 201,202,...,2*n*.

Independentment de que la zona sigui numerada o no, es poden adquirir **entrades** que tenen un determinat *preu\_entrada*. Aquest preu sorgeix de la suma del preu base de la sessió, el de la zona concreta, i l'extra per si es tracta d'un seient amb un càrrec addicional

dins la zona. Com l'entrada té un **estat\_entrada** mutable al llarg del temps, s'ha afegit una classe que ens ho indica.

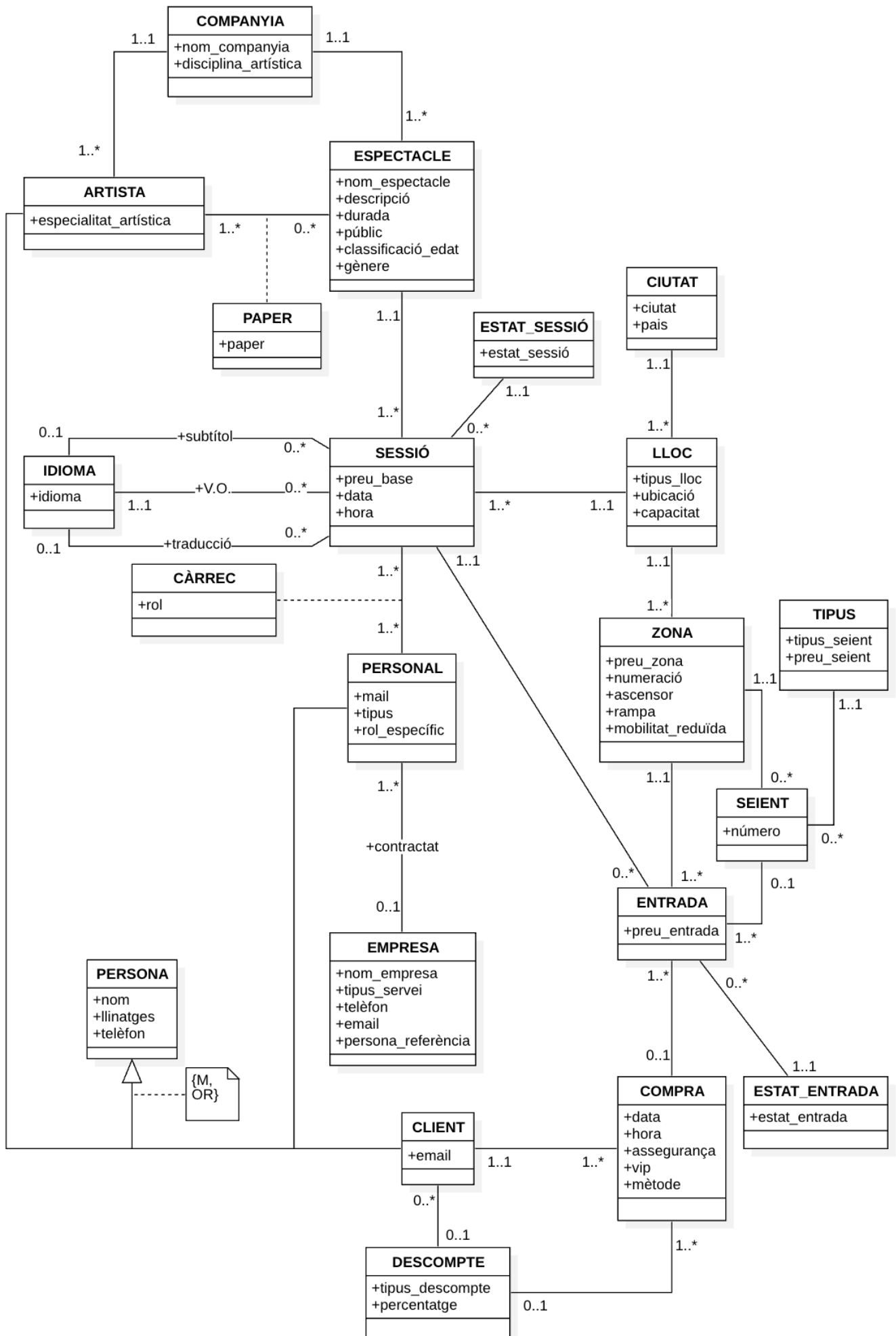
Tot i això, les entrades no ens serveixen de gaire si no hi ha qui les compri. Per això els clients poden reservar-les mitjançant una **compra** on es poden comprar varies en un determinat moment. El pagador ha de poder seleccionar el *mètode\_pagament* que més li convingui. El mètode seleccionat per a realitzar el pagament és un atribut degut a que serà un valor enumerat que no variarà al llarg del temps i només hi ha tres opcions disponibles. Tampoc s'ha implementat tota l'estructura darrere els pagaments ni es guarda informació sobre el possible compte bancari, usuari de Paypal o demés dades bancàries. A més, durant la compra es pot seleccionar una opció addicional (*assegurança* o *serveis VIP* són valors booleans que indiquen si han estat seleccionats o no). Aquesta solució escollida per a les opcions addicionals és similar al cas anterior de les característiques d'accessibilitat d'una zona. Per coherència entre les distintes entitats de la base de dades, s'ha optat un altre pic per implementar-ho com a atributs d'una classe.

El **client** que fa la compra pot tenir accés a un **descompte** si escau, on se li aplicarà un *percentatge* sobre el preu de les entrades. S'indica explícitament mitjançant la multiplicitat 0..1 que només es pot optar a un descompte degut a que no pot pertànyer al col·lectiu d'estudiants, jubilats o grup a la vegada.

Del client en volem saber l'*email* on enviar la confirmació de la compra i les seves entrades, a més de les seves dades personals. Degut a l'existència de diversos perfils de persona, s'ha creat aquesta classe que encapsula els atributs de *nom*, *l·linatges* i *telèfon*. Cal indicar que aquesta herència és en forma de {*M,OR*} degut a que, un artista o encarregat del funcionament d'una sessió pot ser espectador ocasionalment d'algun espectacle, trencant així amb l'exclusivitat d'una XOR o AND.

El tercer col·lectiu que hereta aquests atributs anteriors n'és el **personal** encarregat de treballar a les distintes sessions programades. Tot el personal té un **càrrec** concret a cada una de les sessions, i s'ha de poder saber quin és el seu *rol\_específic* associat i el tipus de treball que realitza. Es tracta d'un cas clar de resolució mitjançant una classe associativa, degut a que no es repetiran casos de que un treballador encarregat de realitzar qualsevol tasca, repetirà la tasca en la mateixa sessió (no hi haurà parells sessió – treballador repetits). En cas que el personal sigui extern, ho sabrem gràcies a la multiplicitat que relaciona personal i classe **empresa**. Si es dona aquest fet i un treballador és d'una empresa externa, es poden saber els atributs que ens interessin. Cal dir que per al cas de la *persona\_referència* s'ha assumit que la informació que conté aquest atribut és únicament el nom de la persona amb que es té el contacte directe amb l'empresa, en cas de que s'hagi de contactar amb ella. No és un individu de la que hem de registrar les seves dades personals ni heretar-los de la classe persona.

A continuació s'adjunta el model conceptual que sorgeix com a producte de l'anàlisi de l'enunciat, representant de forma clara totes les classes i relacions anteriorment explicades.



## Traducció a relacional

Posterior a la fase de modelat que ha donat com a resultat el diagrama UML anterior, s'ha d'aplicar l'algorisme de traducció per a obtenir les taules que ens indicaran les claus primàries i claus foranes de les distintes entitats. Aquestes s'indiquen amb **negreta** i subratllades, respectivament. També, els identificadors creats durant aquesta fase s'indiquen amb *cursiva*.

### Selecció de claus primàries de les taules

La selecció de claus primàries com *nom\_companyia* o *nom\_espectacle* s'ha fet per a simplificar els atributs de les classes. Seria més adient generar un identificador concret, però s'ha assumit que els noms de companyies, espectacles o llocs són suficientment descriptius i identificatius per a la situació que estem descrivint.

També s'ha de dir que en el cas de les persones, es crea un identificador intern del sistema. L'identificador no és el DNI o equivalent, degut a que s'ha decidit que no és necessari emmagatzemar el document personal en el cas d'aquesta base de dades del sector oci.

**COMPANYIA** (***nom\_companyia***, disciplina\_artística)

**ARTISTA** (***id\_artista***, especialitat\_artística)

**PAPER** (***id\_paper***, paper)

**ESPECTACLE** (***nom\_espectacle***, descripció, durada, públic, classificació\_edat, gènere)

**SESSIÓ** (***id\_sessió***, preu\_base, data, hora)

**ESTAT\_SESSIÓ** (***estat\_sessió***)

**CIUTAT** (***ciutat***, país)

**LLOC** (***nom\_lloc***, tipus\_lloc, ubicació, capacitat)

**IDIOMA** (***idioma***)

**ZONA** (***id\_zona***, preu\_zona, numeració, ascensor, rampa, mobilitat\_reduïda)

**SEIENT** (***número\_seient***)

**TIPUS** (***tipus\_seient***, preu\_seient)

**ENTRADA** (***id\_entrada***, preu\_entrada)

**ESTAT\_ENTRADA** (***estat\_entrada***)

**COMPRA** (***id\_compra***, data, hora, assegurança, vip)

**CLIENT** (***id\_client***, email)

**DESCOMpte** (***tipus\_descompte***, percentatge)

**PERSONA** (***id\_persona***, nom, llinatges, telèfon)

**EMPRESA** (***nom\_empresa***, tipus\_servei, telèfon, email, persona\_referència)

**PERSONAL** (***id\_personal***, mail, tipus, rol\_específic)

**CÀRREC** (***id\_càrrec***, rol)

## Selecció de les claus primàries de les relacions

Per a seleccionar les *primary keys* de les relacions s'han tingut en compte les multiplicitats del model conceptual adjunt en la secció anterior. Si ens fixem, l'únic potencial punt conflictiu de d'aquesta passa de l'algorisme de traducció té a veure amb la selecció de les claus relacionades amb les classes associatives. Degut a que el model no té connexions  $N..M$  entre taules, no hi ha relacions on la clau primària sigui composta.

R-COMPANYIA-ESPECTACLE (nom\_companyia, **nom\_espectacle**)

R-ARTISTA-COMPANYIA (**id\_artista**, nom\_companyia)

R-ARTISTA-ESPECTACLE (**id\_artista**, **nom\_espectacle**)

R-ESPECTACLE-SESSIÓ (nom\_espectacle, **id\_sessió**)

R-SESSIÓ-ESTAT\_SESSIÓ (**id\_sessió**, estat\_sessió)

SUBTÍTOL (idioma\_subtitol, **id\_sessió**)

V.O. (idioma\_original, **id\_sessió**)

TRADUCCIÓ (idioma\_traducció, **id\_sessió**)

R-SESSIÓ-LLOC (**id\_sessió**, nom\_lloc)

R-LLOC-CIUTAT (**nom\_lloc**, ciutat)

R-LLOC-ZONA (nom\_lloc, **id\_zona**)

R-ZONA-SEIENT (id\_zona, **número\_seient**)

R-SEIENT-TIPUS (**número\_seient**, tipus\_seient)

R-ZONA-ENTRADA (id\_zona, **id\_entrada**)

R-ENTRADA-SEIENT (**id\_entrada**, número\_seient)

R-ENTRADA-ESTAT\_ENTRADA (**id\_entrada**, estat\_entrada)

R-SESSIÓ-ENTRADA (id\_sessió, **id\_entrada**)

R-SESSIÓ-PERSONAL (**id\_sessió**, **id\_personal**)

R-PERSONAL-EMPRESA (**id\_personal**, nom\_empresa)

R-ENTRADA-COMPRA (**id\_entrada**, id\_compra)

R-CLIENT-COMPRA (id\_client, **id\_compra**)

R-CLIENT-DESCOMPTA (**id\_client**, tipus\_descompte)

R-COMPRA-DESCOMPTA (**id\_compra**, tipus\_descompte)

R-PERSONAL-PERSONA (**id\_personal**, **id\_persona**)

R-CLIENT-PERSONA (**id\_client**, **id\_persona**)

R-ARTISTA-PERSONA (**id\_artista**, **id\_persona**)



## Fusió de les taules i relacions

La passa final d'aquest algorisme de traducció es l'estudi de la possibilitat de fusionar les classes amb les relacions. Cal dir que, llevat les classes associatives, el fet de que no hi hagi claus compostes ha simplificat molt aquesta fase, i les fusions no han suposat gaire complicació.

Com a observació, a les classes associatives **paper** i **càrrec** s'han suprimit els identificadors propis de cada un d'ells i la clau primària és la composició dels parells que formen la instància. Això és degut a que, semànticament, els identificadors no aporten cap tipus d'informació, i només ho fan el atributs *paper* i *càrrec*.

**COMPANYIA** (nom\_companyia, disciplina\_artística)

**ARTISTA** (id\_artista, especialitat\_artistica, nom\_companyia)

**PAPER** (id\_artista, nom\_espectacle, paper)

**ESPECTACLE** (nom\_espectacle, descripció, durada, públic, classificació\_edat, gènere)

**SESSIÓ** (id\_sessió, preu\_base, data, hora, nom\_espectacle, estat\_sessió, idioma\_subtitol, idioma\_original, idioma\_traducció, nom\_lloc)

**ESTAT\_SESSIÓ** (estat\_sessió)

**CIUTAT** (ciutat, país)

**LLOC** (nom\_lloc, tipus\_lloc, ubicació, capacitat, ciutat)

**IDIOMA** (idioma)

**ZONA** (id\_zona, preu\_zona, numeració, ascensor, rampa, mobilitat\_reduïda, nom\_lloc)

**SEIENT** (número\_seient, tipus\_seient, id\_zona)

**TIPUS** (tipus\_seient, preu\_seient)

**ENTRADA** (id\_entrada, preu\_entrada, id\_compra, id\_sessió, estat\_entrada, número\_seient, id\_zona)

**ESTAT\_ENTRADA** (estat\_entrada)

**COMPRA** (id\_compra, data, hora, assegurança, vip, id\_client, tipus\_descompte)

**CLIENT** (id\_client, email, tipus\_descompte)

**DESCOMPTE** (tipus\_descompte, percentatge)

**PERSONA** (id\_persona, nom, llinatges, telèfon)

**EMPRESA** (nom\_empresa, tipus\_servei, telèfon, email, persona\_referència)

**PERSONAL** (id\_personal, mail, tipus, rol\_específic, nom\_empresa)

**CÀRREC** (id\_sessió, id\_personal, rol)

Finalment s'ha provat si el procés de traducció compleixi els requeriments de normalització fins a la 3<sup>a</sup> forma normal. Es compleix degut a que no hi ha atributs multi avaluats, no hi ha claus amb dependències parcials i no hi ha dependències transitives entre taules.

## Implementació en codi SQL

En aquesta secció s'adjunta el codi de la creació de les taules que s'han dissenyat fins aquest punt, on s'ha tingut en compte l'ordre de creació de les taules, de tal manera que les *constraints* siguin coherents. També s'han seleccionat tipus de dades específics per a cada atribut a emmagatzemar. Això fa que, en el cas de variables com *gènere* o *estat\_sessió* entre d'altres, s'especifiquin com a tipus enumerat i es categoritzin segons els valors de l'enunciat. Un altre aspecte a tenir en compte és que, en els casos en que sigui possible, s'ha preferit per afegir la sentència **NOT NULL**, excepte en casos com *preu\_seient* (que pot no tenir un extra associat en funció del seient) o *idioma\_subtítol*, que pot no tenir-ne. Tot això fa que el codi exposat sigui completament executable.

```
CREATE TABLE COMPANYIA (  
    nom_companyia VARCHAR(128) PRIMARY KEY,  
    disciplina_artistica VARCHAR(128) NOT NULL  
);  
  
CREATE TABLE ESPECTACLE (  
    nom_espectacle VARCHAR(128) PRIMARY KEY,  
    descripció TEXT NOT NULL,  
    durada TIME NOT NULL,  
    públic VARCHAR(128) NOT NULL,  
    classificació_edat TINYINT NOT NULL,  
    gènere ENUM('Comèdia', 'Drama', 'Musical', 'Òpera') NOT NULL  
);  
  
CREATE TABLE ESTAT_SESSIÓ (  
    estat_sessió ENUM('Programada', 'Cancel·lada', 'Finalitzada', 'En curs') PRIMARY KEY  
);  
  
CREATE TABLE IDIOMA (  
    idioma VARCHAR(32) PRIMARY KEY  
);  
  
CREATE TABLE CIUTAT (  
    ciutat VARCHAR(64) PRIMARY KEY,  
    país VARCHAR(64) NOT NULL  
);  
  
CREATE TABLE TIPUS_SEIENT (  
    tipus_seient VARCHAR(64) PRIMARY KEY,  
    preu_seient DECIMAL(8,2)  
);  
  
CREATE TABLE ESTAT_ENTRADA (  
    estat_entrada ENUM('Activa', 'Cancel·lada', 'Utilitzada') PRIMARY KEY  
);
```

```

CREATE TABLE DESCOMPTE (
    tipus_descompte ENUM('Estudiants', 'Jubilats', 'Grups') PRIMARY KEY,
    percentatge DECIMAL(5,2) NOT NULL
);

CREATE TABLE PERSONA (
    id_persona INT PRIMARY KEY,
    nom VARCHAR(64) NOT NULL,
    llinatges VARCHAR(128) NOT NULL,
    telèfon VARCHAR(16) NOT NULL
);

CREATE TABLE LLOC (
    nom_lloc VARCHAR(128) PRIMARY KEY,
    tipus_lloc VARCHAR(64) NOT NULL,
    ubicació VARCHAR(128) NOT NULL,
    capacitat INT NOT NULL,
    ciutat VARCHAR(64) NOT NULL,
    FOREIGN KEY (ciutat) REFERENCES CIUTAT(ciutat)
);

CREATE TABLE SESSIÓ (
    id_sessió INT PRIMARY KEY,
    preu_base DECIMAL(8,2) NOT NULL,
    data DATE NOT NULL,
    hora TIME NOT NULL,
    nom_espectacle VARCHAR(128) NOT NULL,
    estat_sessió ENUM('Programada', 'Cancel·lada', 'Finalitzada', 'En curs') NOT NULL,
    idioma_subtitol VARCHAR(32),
    idioma_original VARCHAR(32) NOT NULL,
    idioma_traducció VARCHAR(32),
    nom_lloc VARCHAR(128) NOT NULL,
    FOREIGN KEY (nom_espectacle) REFERENCES ESPECTACLE(nom_espectacle),
    FOREIGN KEY (estat_sessió) REFERENCES ESTAT_SESSIÓ(estat_sessió),
    FOREIGN KEY (idioma_subtitol) REFERENCES IDIOMA(idioma),
    FOREIGN KEY (idioma_original) REFERENCES IDIOMA(idioma),
    FOREIGN KEY (idioma_traducció) REFERENCES IDIOMA(idioma),
    FOREIGN KEY (nom_lloc) REFERENCES LLOC(nom_lloc)
);

CREATE TABLE ZONA (
    id_zona INT PRIMARY KEY,
    preu_zona DECIMAL(8,2),
    numeració BOOLEAN NOT NULL,
    ascensor BOOLEAN NOT NULL,
    rampa BOOLEAN NOT NULL,
    mobilitat_reduïda BOOLEAN NOT NULL,
    nom_lloc VARCHAR(128) NOT NULL,
    FOREIGN KEY (nom_lloc) REFERENCES LLOC(nom_lloc)
);

```

```

CREATE TABLE SEIENT (
    número_seient INT PRIMARY KEY,
    tipus_seient VARCHAR(64) NOT NULL,
    id_zona INT NOT NULL,
    FOREIGN KEY (tipus_seient) REFERENCES TIPUS_SEIENT(tipus_seient),
    FOREIGN KEY (id_zona) REFERENCES ZONA(id_zona)
);

CREATE TABLE EMPRESA (
    nom_empresa VARCHAR(128) PRIMARY KEY,
    tipus_servei VARCHAR(128) NOT NULL,
    telèfon VARCHAR(16) NOT NULL,
    email VARCHAR(128) NOT NULL,
    persona_referència VARCHAR(128) NOT NULL
);

CREATE TABLE PERSONAL (
    id_personal INT PRIMARY KEY,
    mail VARCHAR(128) NOT NULL,
    tipus VARCHAR(64) NOT NULL,
    rol_especific VARCHAR(128) NOT NULL,
    nom_empresa VARCHAR(128),
    FOREIGN KEY (id_personal) REFERENCES PERSONA(id_persona),
    FOREIGN KEY (nom_empresa) REFERENCES EMPRESA(nom_empresa)
);

CREATE TABLE ARTISTA (
    id_artista INT PRIMARY KEY,
    especialitat_artística VARCHAR(128) NOT NULL,
    nom_companyia VARCHAR(128) NOT NULL,
    FOREIGN KEY (id_artista) REFERENCES PERSONA(id_persona),
    FOREIGN KEY (nom_companyia) REFERENCES COMPANYIA(nom_companyia)
);

CREATE TABLE CLIENT (
    id_client INT PRIMARY KEY,
    email VARCHAR(128) NOT NULL,
    tipus_descompte ENUM('Estudiants', 'Jubilats', 'Grups'),
    FOREIGN KEY (id_client) REFERENCES PERSONA(id_persona),
    FOREIGN KEY (tipus_descompte) REFERENCES DESCOMPTE(tipus_descompte)
);

CREATE TABLE COMPRA (
    id_compra INT PRIMARY KEY,
    data DATE NOT NULL,
    hora TIME NOT NULL,
    assegurança BOOLEAN NOT NULL,
    vip BOOLEAN NOT NULL,
    id_client INT NOT NULL,
    tipus_descompte ENUM('Estudiants', 'Jubilats', 'Grups'),
    FOREIGN KEY (id_client) REFERENCES CLIENT(id_client),

```

```

    FOREIGN KEY (tipus_descompte) REFERENCES DESCOMPTE(tipus_descompte)
);

CREATE TABLE ENTRADA (
    id_entrada INT PRIMARY KEY,
    preu_entrada DECIMAL(8,2) NOT NULL,
    id_compra INT NOT NULL,
    id_sessió INT NOT NULL,
    estat_entrada ENUM('Activa', 'Cancel·lada', 'Utilitzada') NOT NULL,
    número_seient INT NOT NULL,
    id_zona INT NOT NULL,
    FOREIGN KEY (id_compra) REFERENCES COMPRA(id_compra),
    FOREIGN KEY (id_sessió) REFERENCES SESSIÓ(id_sessió),
    FOREIGN KEY (estat_entrada) REFERENCES ESTAT_ENTRADA(estat_entrada),
    FOREIGN KEY (número_seient) REFERENCES SEIENT(número_seient),
    FOREIGN KEY (id_zona) REFERENCES ZONA(id_zona)
);

CREATE TABLE CARREC (
    id_sessió INT NOT NULL,
    id_personal INT NOT NULL,
    rol VARCHAR(128) NOT NULL,
    PRIMARY KEY (id_sessió, id_personal),
    FOREIGN KEY (id_sessió) REFERENCES SESSIÓ(id_sessió),
    FOREIGN KEY (id_personal) REFERENCES PERSONAL(id_personal)
);

CREATE TABLE PAPER (
    id_artista INT NOT NULL,
    nom_espectacle VARCHAR(128) NOT NULL,
    paper VARCHAR(128) NOT NULL,
    PRIMARY KEY (id_artista, nom_espectacle),
    FOREIGN KEY (id_artista) REFERENCES ARTISTA(id_artista),
    FOREIGN KEY (nom_espectacle) REFERENCES ESPECTACLE(nom_espectacle)
);

```

## Conclusions

El desenvolupament d'aquesta base de dades per a la gestió d'esdeveniments culturals i d'oci ha permès abordar un procés complet de disseny i implementació que garanteix la fiabilitat i eficiència en l'emmagatzematge i gestió de la informació. A través de les diverses fases descrites, s'ha aconseguit construir una estructura sòlida i coherent que cobreix totes les necessitats plantejades a l'enunciat.

En primer lloc, l'anàlisi del problema ha estat fonamental per identificar les entitats i relacions clau, així com per comprendre les necessitats específiques del sistema. Això ha facilitat la creació d'un model conceptual detallat que reflecteix de manera clara les interaccions entre espectacles, companyies, artistes, sessions, clients i el sistema de compra d'entrades.

El procés de traducció del model conceptual al model relacional ha permès definir amb precisió les taules, claus primàries i foranes, assegurant la integritat referencial de la base de dades. La selecció acurada de claus primàries, juntament amb la gestió de relacions i classes associatives, ha contribuït a optimitzar el disseny i obtenir un model que complia amb els requeriments de normalització.

La implementació en codi SQL ha culminat el treball, amb la definició de les taules i restriccions adequades per garantir la coherència i consistència de la informació. L'ús de tipus de dades específics, restriccions NOT NULL i valors enumerats ha permès concloure un model completament coherent.

En definitiva, aquest projecte no només ha complert amb els requisits funcionals inicials, sinó que també ha proporcionat una base sòlida per a futures ampliacions o adaptacions del sistema, demostrant la importància d'un bon disseny conceptual i relacional en el desenvolupament de qualsevol base de dades.

## Referències bibliogràfiques i recursos

A continuació s'exposen les fonts i recursos d'informació consultades per al desenvolupament d'aquesta pràctica, junt amb la raó que ho ha motivat:

- [El model relacional, Base de Dades I \(material de l'assignatura\)](#): dubtes relacionats amb la traducció de taules.
- [Normalització, Base de Dades I \(material de l'assignatura\)](#): consulta dels requisits de les distintes formes normals d'un model relacional.
- [ChatGPT, OpenAI](#): discussió del significat d'algunes relacions del model conceptual, per a comparar quina és la millor manera de representar la idea de l'autor. Consultes relacionades amb les especificacions de tipus de SQL en la declaració de taules.