

Desenvolupament d'aplicacions amb tecnologies web.

MF0492_3:

UF1845- Accés a dades en aplicacions web de l'entorn servidor.

Introducció a les bases de dades

Índex

- Introducció a les Bbd
 - Bases de dades
 - Tipus de bases de dades
 - Anàlisi, disseny i implementació de *BDD*.



Bases de dades

- Es coneixen com a **base de dades** els bancs d'informació que contenen dades que comparteixen algun tipus de relació, i que busquen ordenar-les i classificar-les en conjunt.
 - Bàsicament, són conjunts de dades pertanyents a un mateix context i emmagatzemats sistemàticament per al seu ús posterior.
- Aplicacions anomenades **sistemes gestors de bases de dades (SGDB)** permeten emmagatzemar i accedir a les dades de forma ràpida i estructurada.



Bases de dades

Mini glossari

- **BDD:** Base de dades
- **SGBD:** Sistema Gestor de Base de Dades
- **DBMS:** *Data Base Management System* = SGBD



Gestors Bases de dades





Base de dades

Les aplicacions de qualsevol tipus necessiten desar, recuperar, actualitzar i esborrar la informació que gestionen en un sistema d'emmagatzematge estable i segur.

Per tant han de disposar d'una base de dades vinculada com a sistema d'emmagatzematge i gestió de la informació.

Tipus de BDDs



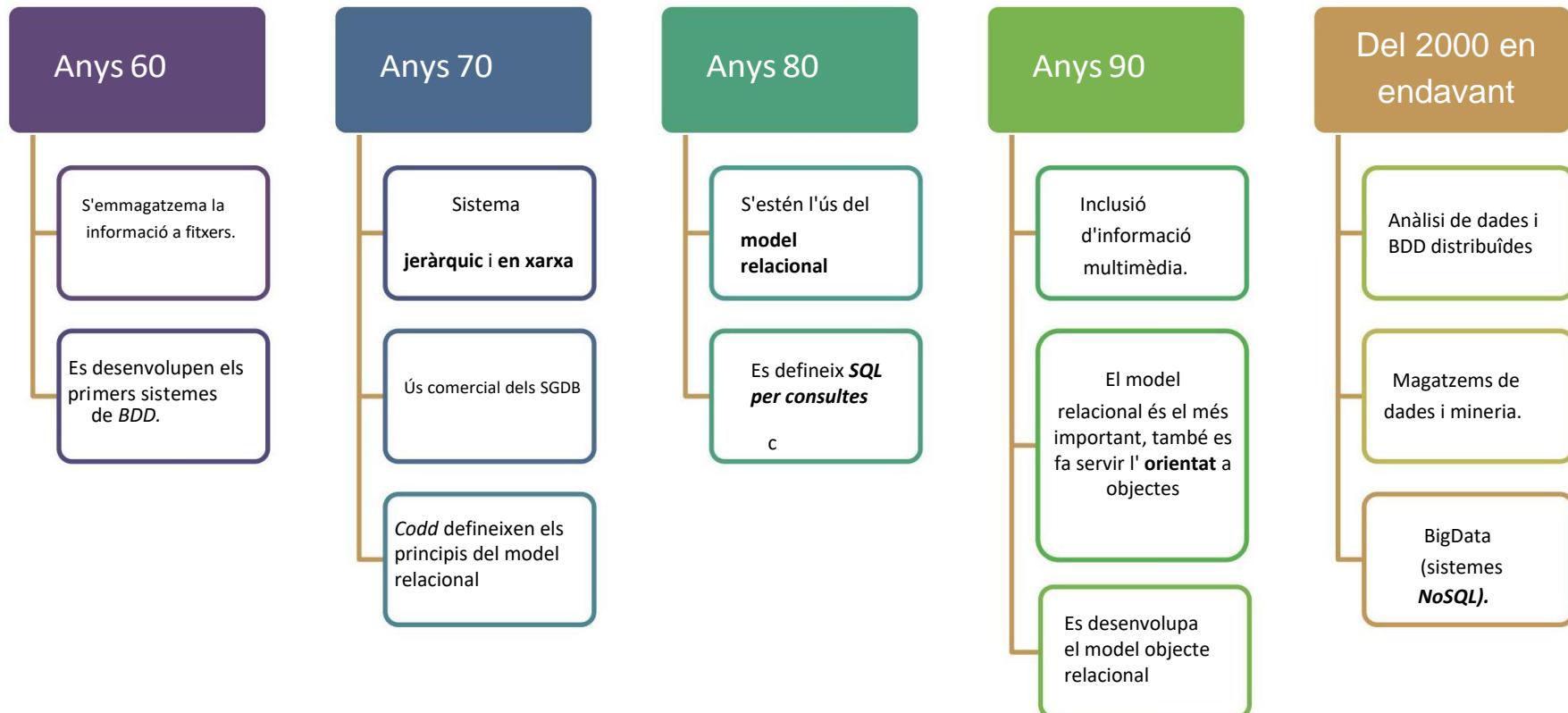
Tipus de bases de dades

Hi ha múltiples paradigmes per a bases de dades, Un dels mes utilitzats és el **relacional**).

Des dels primers que van aparèixer, com els models **jeràrquic** o en **xarxa**, al model més estès i popular actualment que és el **relacional** i que treballarem en aquest curs.

Hi ha altres models importants, com els **orientats a objectes** i els sistemes **NoSQL** més actuals (entre els quals s'inclouen les **BDD documentals**) que permeten tractar grans quantitats d' informació ràpidament.

Tipus i evolució



Tipus de bases de dades

Primera generació

Jeràrquiques. En desús

En xarxa. En desús

Segona generació:

Relacionals. Model més actiu

Tercera generació

Orientades a objecte

Objecte-relacionals

Nova generació: *NoSQL(BigData)*

BDDs jeràrquiques

A les **bases de dades jeràrquiques**:

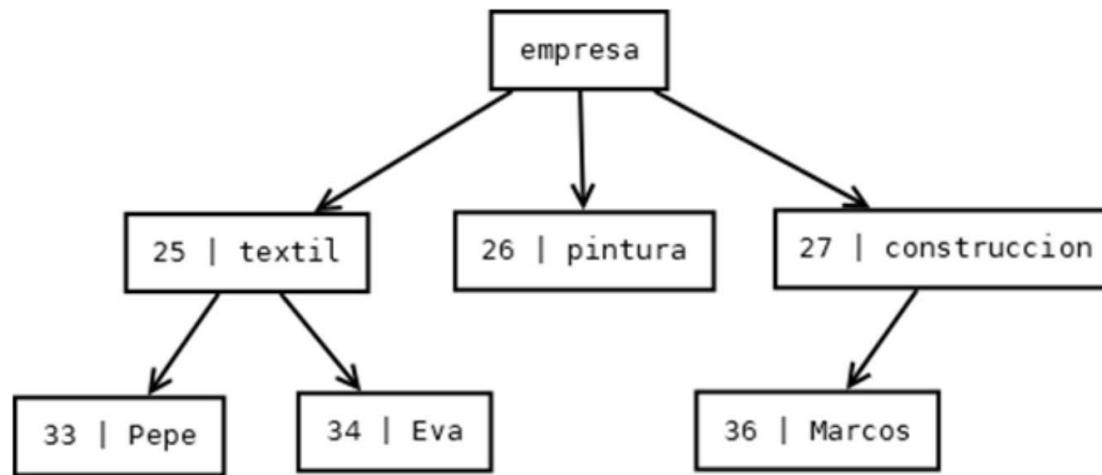
S'organitza la informació amb un jerarquia on la relació entre les entitats d'aquest model sempre és del tipus pare/fill.

Hi ha una sèrie de nodes que contindran atributs i que es relacionaran amb nodes fills, de manera que hi pot haver més d'un fill per al pare, però un fill només té un pare.

Les entitats d'aquest model s'anomenen segments i els atributs camps.

La forma visual d'aquest model és d'arbre invertit, a la part superior hi ha els pares i a la inferior els fills.

Mostra BDDs jeràrquiques



BDDs jeràrquiques

Aquest tipus de bases de dades presenten els problemes següents:

Duplicitat de registres i inconsistència de dades, a més no es poden representar bé les relacions n a n.

El sistema no garanteix la integritat referencial.



BDDs en xarxa

A les **bases de dades en xarxa**:

S'organitza la informació a registres i enllaços.

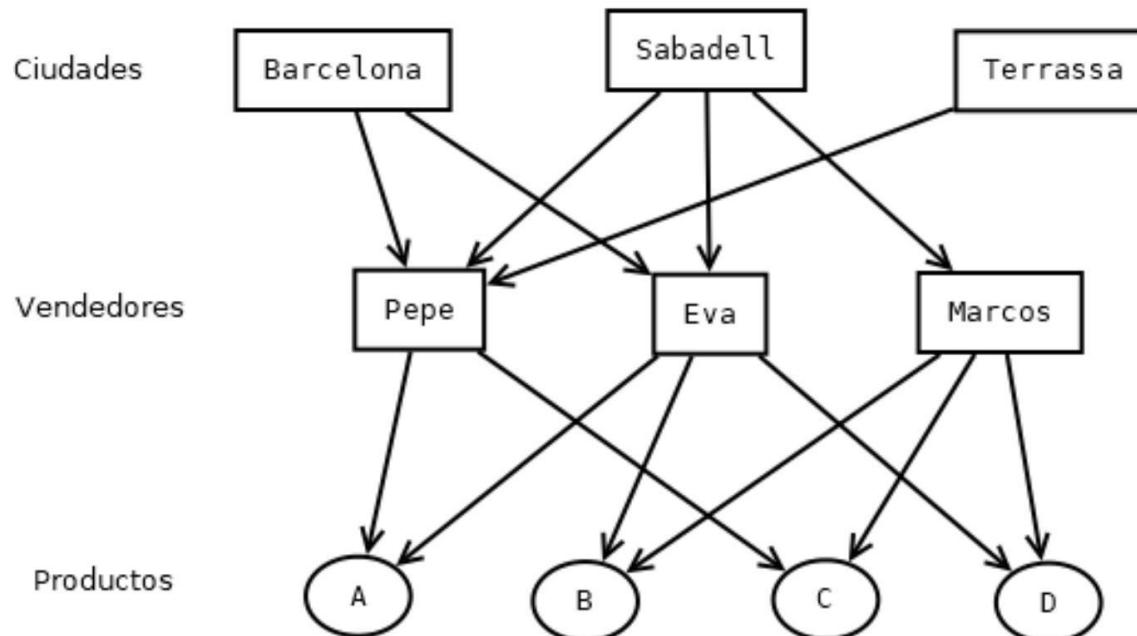
Als registres s'emmagatzemen les dades utilitzant atributs, mentre que els enllaços permeten relacionar els registres de la base de dades.

El model en xarxa més acceptat és l'anomenat *codasyl*, que durant molt de temps ha esdevingut un estàndard.

Les bases de dades en xarxa són semblants a les jeràrquiques només que pot haver-hi més d'un pare per cada fill.

En aquest model es poden representar perfectament relacions diverses a diverses

Mostra bases de dades en xarxa



BDDs en xarxa

Aquest tipus de bases de dades també tenen problemes:

La seva dificultat de gestió per l'accés i complexitat fa que estiguin abandonant-se completament.



BDDs relacionals

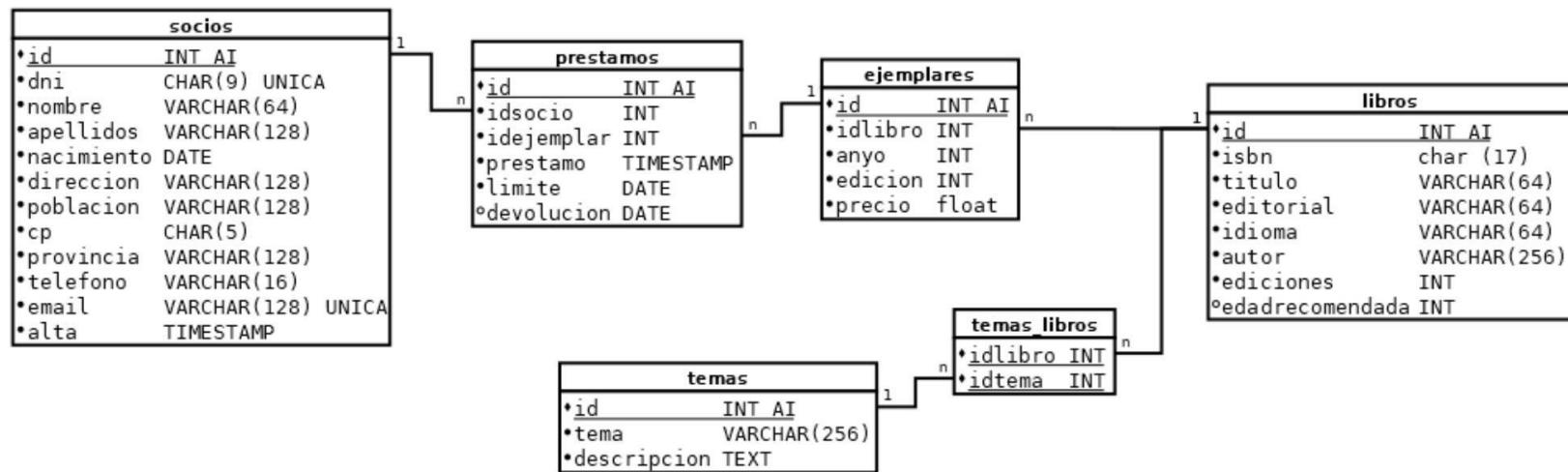
Una **Base de Dades Relacional** és una base de dades que compleix el model relacional, que és el model **més utilitzat actualment**.

La informació es desa en **taules**, on cada fila és un **registre** (tupla) i cada columna representa un **camp** (atribut).

Permet establir **relacions** entre les dades de les diferents taules.

Després de ser postulades les seves bases el 1970 per *Edgar Frank Codd*, dels laboratoris *IBM* a San José (Califòrnia), no va trigar a consolidar-se com a nou paradigma.

Mostra base de dades relacional



BDDs relacionals

Any	Fet
1970	Codd publica les bases del model relacional
1971-72	Primers desenvolupaments teòrics
1973-78	Primers prototips
1978	Apareix el llenguatge <i>QBE (Query By Example)</i>
1979	Apareix <i>Oracle</i>
1980	Apareix <i>Ingres</i>
1981	Apareix <i>SQL</i>
1982	Apareix <i>DB2</i> (d' <i>IBM</i>)
1986	<i>ANSI</i> normalitza l' <i>SQL (SQL/ANSI)</i>
1987	<i>SQL</i> d' <i>ISO</i>
1990	Versió dos del model relacional (<i>RM/ V2</i>)
1992	<i>SQL92</i>
1998	<i>SQL3</i>

BDDs orientades a objecte

Des de l'aparició de la **programació orientada a objectes** es va començar a pensar en **bases de dades adaptades** a aquests llenguatges, de manera que dades i els procediments es puguin emmagatzemar junts.

S'intenta corregir les limitacions de les relacionals: herència, tipus definits per l'usuari, *triggers*, suport multimèdia...

Estan pensades per treballar en conjunció amb llenguatges de programació OO com *Java*, *C#*, *Visual Basic.NET* i *C++*.

Exemples: *ObjectDatabase++*, *ObjectStone*, *GemStone*, *Perst*, *Wakanda*, *ObjectDB*...

El problema dels basis de dades orientades a objectes és que no tenen paradigmes unificats, ni estandardització

BDDs objecte-relacionals

Les **objecte-relacionals** són un híbrid entre el model relacional i l'orientat a objectes.

- S'intenta una compatibilitat relacional donant la possibilitat d'integrar millors de l'orientació a objectes.
- Es basen en l'estàndard **SQL99**, que afegeix la possibilitat d'emmagatzemar procediments, *triggers*, tipus de l'usuari, consultes recursives...
- Les darreres versions de la majoria de les grans bases de dades relacionals (*Oracle Database*, *SQL Server*, *MySQL*, *MariaDB*...) són objecte relacionals.



NoSQL (BigData)

NoSQL és una àmplia classe de *SGDB* que difereixen del model clàssic en aspectes importants com que no usen *SQL* com a llenguatge principal de consultes.

Les dades emmagatzemades no requereixen estructures fixes com taules, normalment no suporten operacions *JOIN*, ni garanteixen completament *ACID* (atomicitat, consistència, aïllament i durabilitat).

Els sistemes *NoSQL* s'anomenen de vegades "no només *SQL*" per subratllar el fet que també poden suportar llenguatges de consulta de tipus *SQL*.

NoSQL (BigData)

S'utilitzen en xarxes socials, com *Google*, *Amazon*, *Twitter* i *Facebook* per tractar grans volums de dades, on és més important tenir la dada que la coherència.

Normalment, no ofereixen gaire més que la possibilitat d'emmagatzemar registres, estant optimitzades per a les operacions d'afegir i recuperar.

No existeix un sol model de bases de dades *NoSQL*, si us interessa el tema trobareu cursos a *CIFO Violeta* i documentació i recursos a la xarxa: <https://es.wikipedia.org/wiki/NoSQL>.

NoSQL documentals

Les **bases de dades documentals** són una de les més importants dins dels models no relacionals o *NoSQL*.

Una **base de dades** d'aquestes característiques es recolza en la utilització de documents per a l'estructuració de l'emmagatzematge i les consultes.

Aquests documents són compostos per tots i cadascun dels registres i dades associades.

Els **documents** que contenen aquestes bases de dades poden referir-se a documents de text, fitxers **XML** o **JSON**.



NoSQL documentals

- La conformació d'aquestes **bases de dades** en lloc d'estar integrada per columnes amb noms i tipus de dades, com passa a les *BBDD* relacionals, té dades semiestructurades en forma de documents i valors.



BDD NoSQL vs BDD relacional



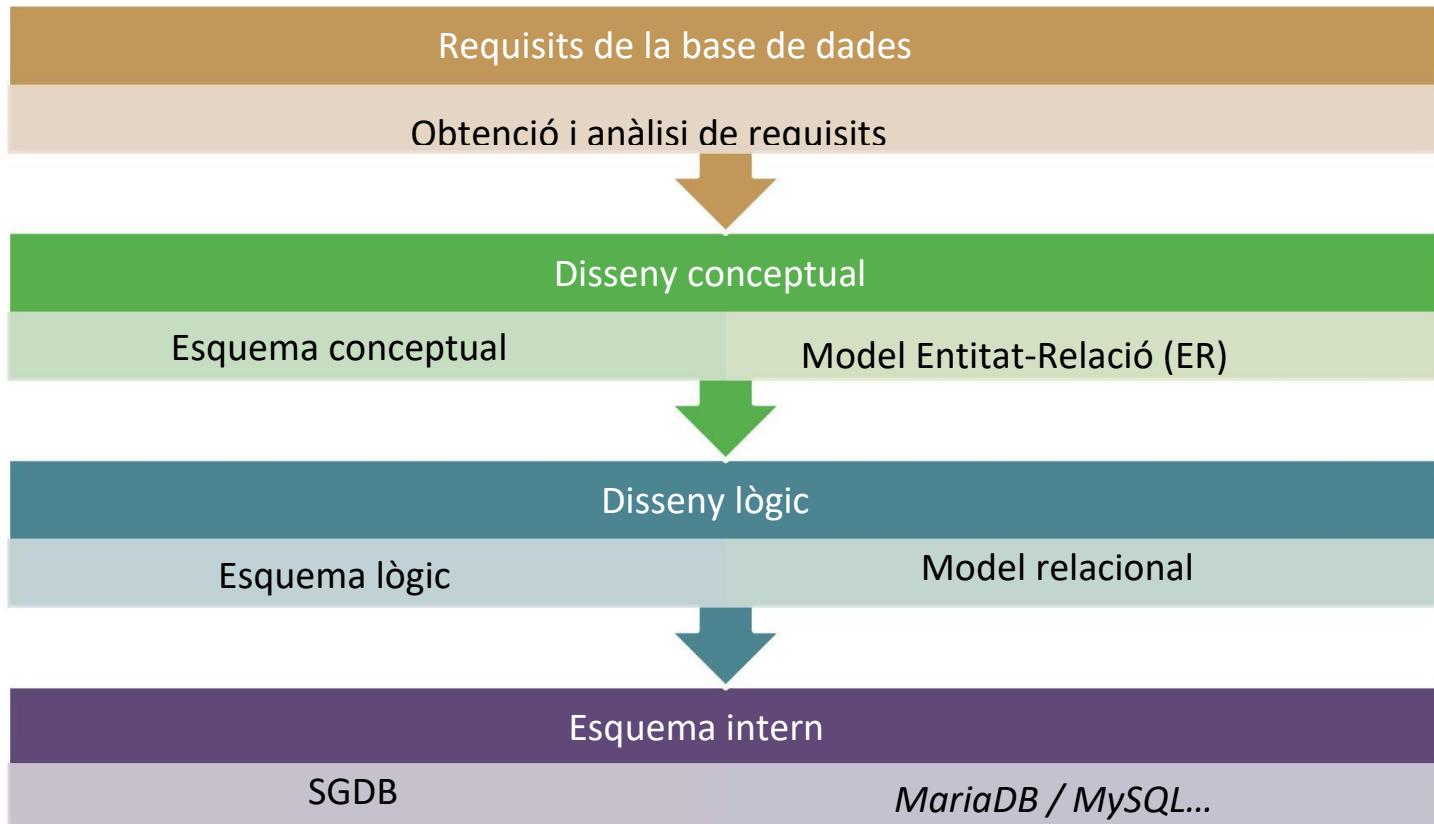
Anàlisi i disseny d'una BDD



Desenvolupament

- Les fases de desenvolupament d'un projecte de programari inclouen: anàlisi, disseny, implementació, prova i manteniment.
- Quan les aplicacions tenen bases de dades, hem de **fer també una anàlisi de la informació a gestionar**, determinant les entitats, els atributs i les seves relacions.
 - Després de l'anàlisi, farem un **disseny de la base de dades**.
 - Les fases esmentades es fan tant a nivell de l'aplicació com per a la BDD i la GUI (interfície gràfica d'usuari).

Exemple passos a seguir



Exemple

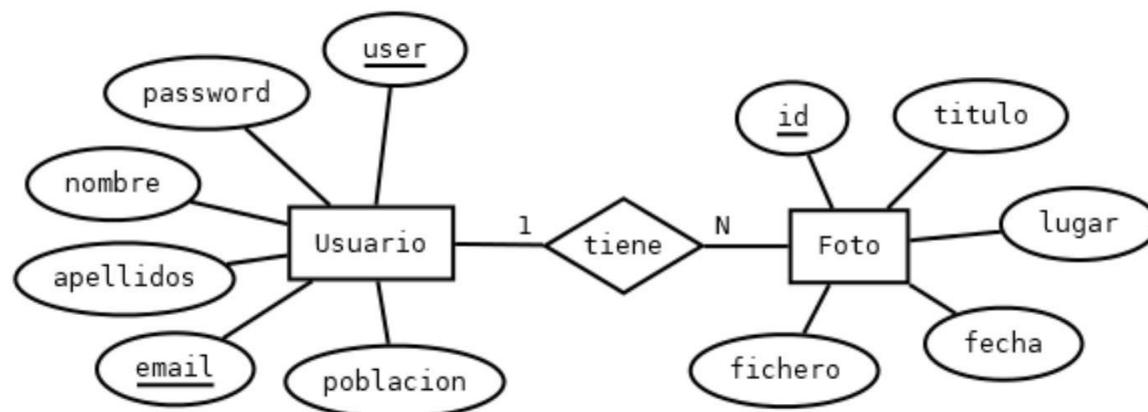
- Per comprendre millor els passos indicats a la diapositiva anterior, dissenyarem una petita *BDD* a partir d'un enunciat.
- ENUNCIAT: volem una aplicació que permeti registrar usuaris i que, un cop es trobin identificats, puguin pujar les seves fotografies.

És evident que la base de dades ha de guardar informació tant d'usuaris com de fotografies (entitats).

El disseny conceptual

- Començarem el disseny de la nostra base de dades elaborant un model conceptual.
- El disseny **conceptual** és una **representació del món real** que consta d'una col·lecció d'objectes bàsics, que seran representats posteriorment en un sistema de base de dades.
- S'elabora a partir de l'anàlisi de requeriments.
- Utilitzarem el **model entitat-relació (ER)** per al nostre disseny

Exemple model entitat-relació (ER)

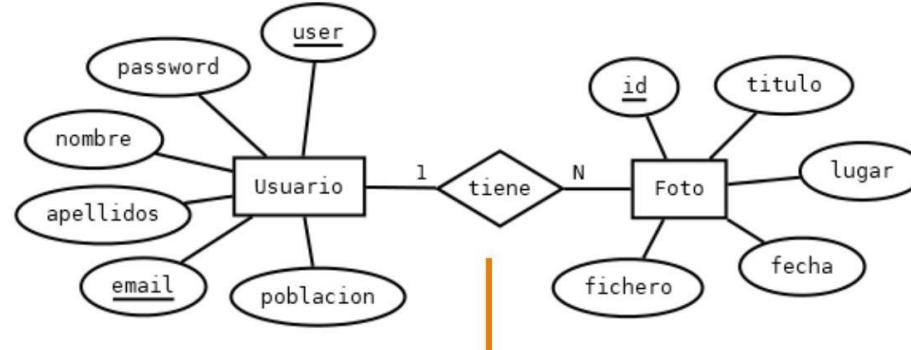


El disseny lògic

- A continuació, passarem el nostre esquema conceptual a un esquema lògic.
- Un esquema **canònic** o **lògic** és un esquema que representa la estructura d'una base de dades.
 - Depèn del tipus de **DBMS** (o *SGBD*) que utilitzarem i es crea a partir del model conceptual.
- Nosaltres treballarem amb el **model relacional**.

Exemple model relacional

ER (disseny conceptual)



El convertim en... Relacional (disseny lògic)

usuarios	
•user	VARCHAR(16)
•password	VARCHAR(32)
•nombre	VARCHAR(64)
•apellido	VARCHAR(256)
•email	VARCHAR(128) UNICA
•poblacion	VARCHAR(128)

fotos	
•id	INTEGER AI
•titulo	VARCHAR(256)
•lugar	VARCHAR (256)
•fecha	DATE
•fichero	VARCHAR(256)
•descripcion	TEXT
•user	VARCHAR (16)

Implementació

- Un cop elaborat l'esquema relacional, podrem convertir-lo en **una sèrie d'instruccions SQL** que crearan la nostra nova base de dades a l' SGDB triat.



- Hi ha eines per generar el codi a partir de l'esquema relacional, sense haver de picar SQL. Òbviament no les farem servir per ara, ja que un dels nostres primers objectius és aprendre SQL.



Implementació (parcial)

```
-- Crear la BDD
CREATE DATABASE mini_instagram
    DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;

-- Usar la BDD
USE mini_instagram;

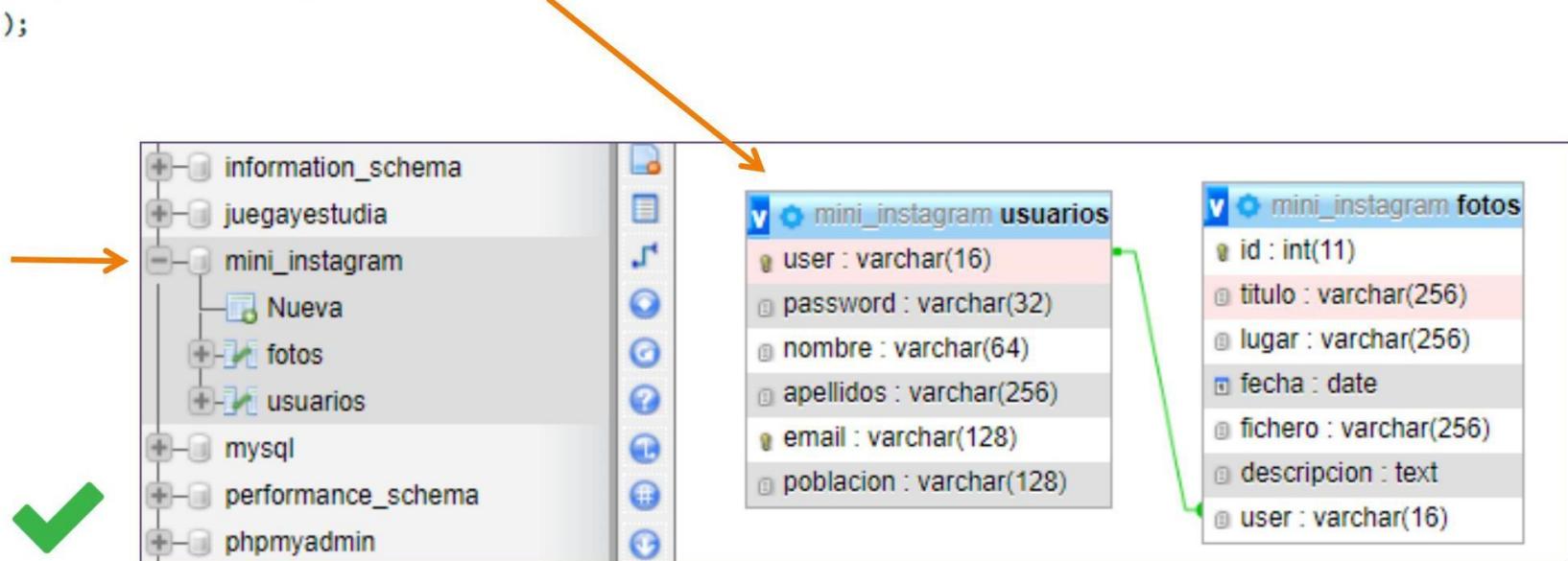
-- CREACIÓN DE LAS TABLAS
-- creación de la tabla usuarios
CREATE TABLE usuarios(
    user VARCHAR(16) PRIMARY KEY,
    password VARCHAR(32) NOT NULL,
    nombre VARCHAR(64) NOT NULL,
    apellidos VARCHAR(256) NOT NULL,
    email VARCHAR(128) NOT NULL UNIQUE KEY,
    poblacion VARCHAR(128) NOT NULL
);
```

Resultat

```
CREATE TABLE usuarios(
    user VARCHAR(16) PRIMARY KEY,
    password VARCHAR(32) NOT NULL,
    nombre VARCHAR(64) NOT NULL,
    apellidos VARCHAR(256) NOT NULL,
    email VARCHAR(128) NOT NULL UNIQUE KEY,
    poblacion VARCHAR(128) NOT NULL
);
```

usuarios	
•user	VARCHAR(16)
•password	VARCHAR(32)
•nombre	VARCHAR(64)
•apellidos	VARCHAR(256)
•email	VARCHAR(128) UNICA
•poblacion	VARCHAR(128)

fotos	
•id	INTEGER AI
•titulo	VARCHAR(256)
•lugar	VARCHAR (256)
◦fecha	DATE
◦fichero	VARCHAR(256)
◦descripcion	TEXT
◦user	VARCHAR (16)



En funcionament...

- Un cop la nostra base de dades estigui operativa, les aplicacions ja podran començar a treballar-hi.
- A continuació es mostren unes captures de com realment aquest disseny funcionarà correctament.
 - Mostra de dades a les taules de la *BDD*.
 - Mosta de consultes en *SQL* que recuperen informació.

Dades a les taules

user	password	nombre	apellidos	email	poblacion
eva	1234	Eva	López	eva@cifo.cat	Terrassa
juan	1234	Juan	Fernández	juan@cifo.cat	Sabadell
marcos	1234	Marcos	Martínez	marcos@cifo.cat	Sant Cugat
marta	1234	Marta	Rodríguez	marta@cifo.cat	Manresa
pepe	1234	Jose	Pérez	pepe@cifo.cat	Sabadell

usuaris



fotos



id	titulo	lugar	fecha	fichero	descripcion	user
1	Montserrat	Manresa	2000-01-01	mon.jpg	NULL	pepe
2	El Pilar	Zaragoza	2001-01-01	pil.jpg	NULL	pepe
3	La Mola	Terrassa	2002-01-01	mol.jpg	NULL	eva
4	Piscina	Sabadell	2003-01-01	pis.jpg	NULL	eva
5	Bloque de pisos	Ripollet	2004-01-01	blo.jpg	NULL	juan
6	Cuñado pesado	Terrassa	2005-01-01	cun.jpg	NULL	juan
7	Mi perro	Sabadell	2006-01-01	per.jpg	NULL	juan
8	Torre de Pisa	Pisa	2007-01-01	tor.jpg	NULL	marta
9	Puerta del Sol	Madrid	2008-01-01	sol.jpg	NULL	marcos
10	Jardín	Barcelona	2009-01-01	jar.jpg	NULL	marcos
11	Cascada	Toronto	2010-01-01	cas.jpg	NULL	eva



Mostra de consultes (una taula)

```
59      -- fotos de Sabadell
60 •  SELECT * FROM fotos WHERE lugar='Sabadell';
61
Result Grid | Filter Rows: [ ] | Edit: [ ] | Export/Import:
+----+----+----+----+----+----+----+
| id | titulo | lugar | fecha | fichero | descripcion | user |
+----+----+----+----+----+----+----+
| 4  | Piscina | Sabadell | 2003-01-01 | pis.jpg | NULL          | eva   |
| 7  | Mi perro | Sabadell | 2006-01-01 | per.jpg | NULL          | juan  |
+----+----+----+----+----+----+----+
```



```
62      -- fotos de Pepe o de Marcos
63 •  SELECT * FROM fotos WHERE user IN ('pepe','marcos');
64
Result Grid | Filter Rows: [ ] | Edit: [ ] | Export/Import: [ ] | Wrap
+----+----+----+----+----+----+----+
| id | titulo | lugar | fecha | fichero | descripcion | user |
+----+----+----+----+----+----+----+
| 1  | Montserrat | Manresa | 2000-01-01 | mon.jpg | NULL          | pepe  |
| 2  | El Pilar | Zaragoza | 2001-01-01 | pil.jpg | NULL          | pepe  |
| 9  | Puerta del Sol | Madrid | 2008-01-01 | sol.jpg | NULL          | marcos |
| 10 | Jardín | Barcelona | 2009-01-01 | jar.jpg | NULL          | marcos |
+----+----+----+----+----+----+----+
```



Mostra de consultes (*outer join*)

```
65    -- lista de usuarios y sus fotos
66 •  SELECT nombre, apellidos, titulo, lugar
67    FROM usuarios LEFT JOIN fotos ON usuarios.user=fotos.user
68    ORDER BY usuarios.user ASC;
69
```



	nombre	apellidos	titulo	lugar
▶	Eva	López	La Mola	Terrassa
	Eva	López	Piscina	Sabadell
	Eva	López	Cascada	Toronto
	Juan	Fernández	Bloque de pisos	Ripollet
	Juan	Fernández	Cuñado pesado	Terrassa
	Juan	Fernández	Mi perro	Sabadell
	Marcos	Martínez	Puerta del Sol	Madrid
	Marcos	Martínez	Jardín	Barcelona
	Marta	Rodríguez	Torre de Pisa	Pisa
	Jose	Pérez	Montserrat	Manresa
	Jose	Pérez	El Pilar	Zaragoza

Mostra de consultes (totals i grups)

```
70      -- los tres lugares con más fotos
71 •  SELECT lugar, COUNT(id) AS total
72      FROM fotos
73      GROUP BY lugar
74      ORDER BY total DESC
75      LIMIT 3;
76
```



Result Grid		Filter Rows:	Export:
	lugar	total	
▶	Terrassa	2	
	Sabadell	2	
	Barcelona	1	



Mostra de consultes (cond. grups)

```
77    -- socios con más de dos fotos
78 •  SELECT nombre, apellidos, COUNT(fotos.id) AS total
79      FROM usuarios LEFT JOIN fotos ON usuarios.user=fotos.user
80      GROUP BY usuarios.user
81      HAVING total>2;
82
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

	nombre	apellidos	total
▶	Eva	López	3
	Juan	Fernández	3



Infografía del procés

