

---

**M6.UF3.A5.P3**  
**DISSENY BASE DE DADES**  
**MYSQL Y MONGO**

**Eduard Lara**

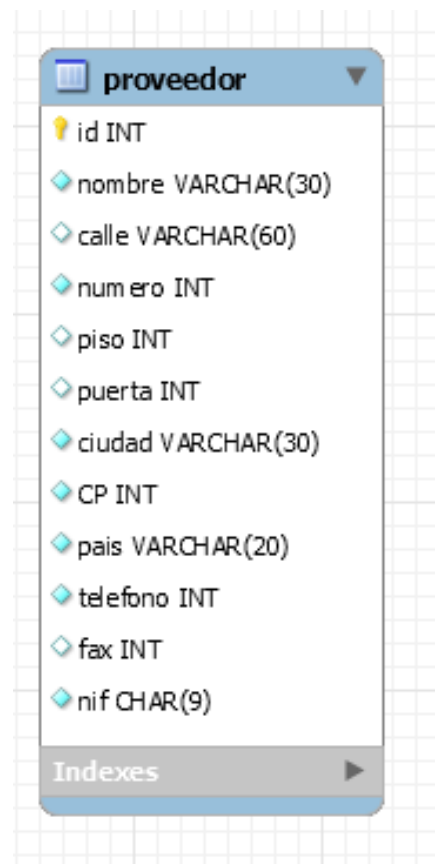
# INDICE

---

1. Exemple disseny de base de dades relacionals. Optica
2. Exemple disseny de base de dades relacionals. Pizzeria
3. Modelat amb dades Mongo
4. Exemple disseny de base de dades Mongo. Optica
5. Exemple disseny de base de dades Mongo. Pizzeria
6. Practica disseny Youtube
7. Practica disseny Spotify

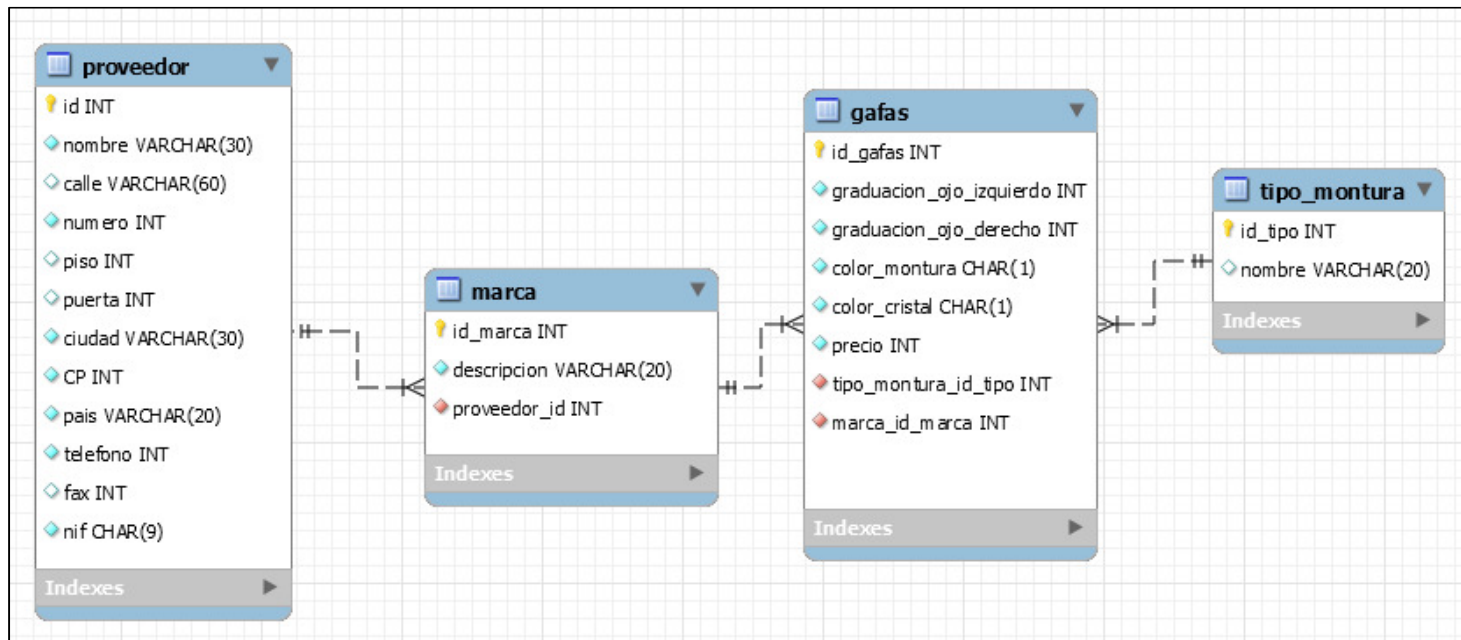
# 1. EXEMPLE OPTICA MYSQL

**Pas 1.** En concret vol saber de cada proveïdor el nom, l'adreça (carrer, número, pis, porta, ciutat, codi postal i país), telèfon, fax, NIF.



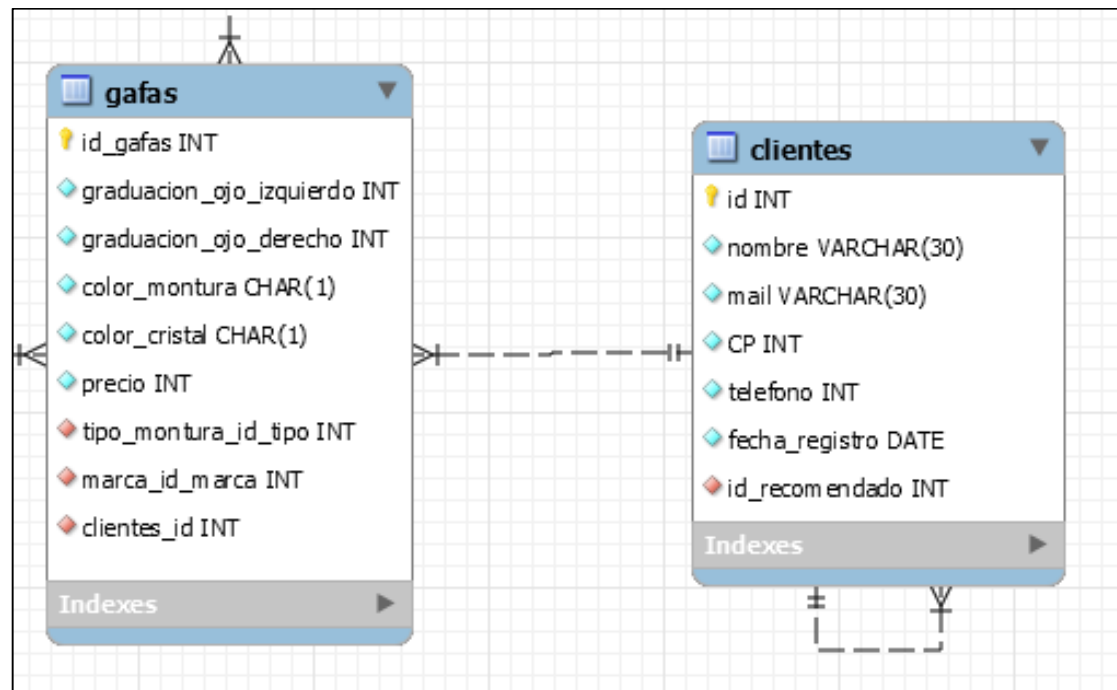
# 1. EXEMPLE OPTICA MYSQL

**Pas 2.** La política de compres de l'òptica es basa en que les ulleres d'una marca es compraran a un únic proveïdor (així en podrà treure més bons preus), però poden comprar ulleres de diverses marques a un proveïdor. De les ulleres vol saber, la marca, la graduació de cadascun dels vidres, el tipus de muntura (flotant, pasta o metàl·lica), el color de la muntura, el color de cada vidre i el preu.



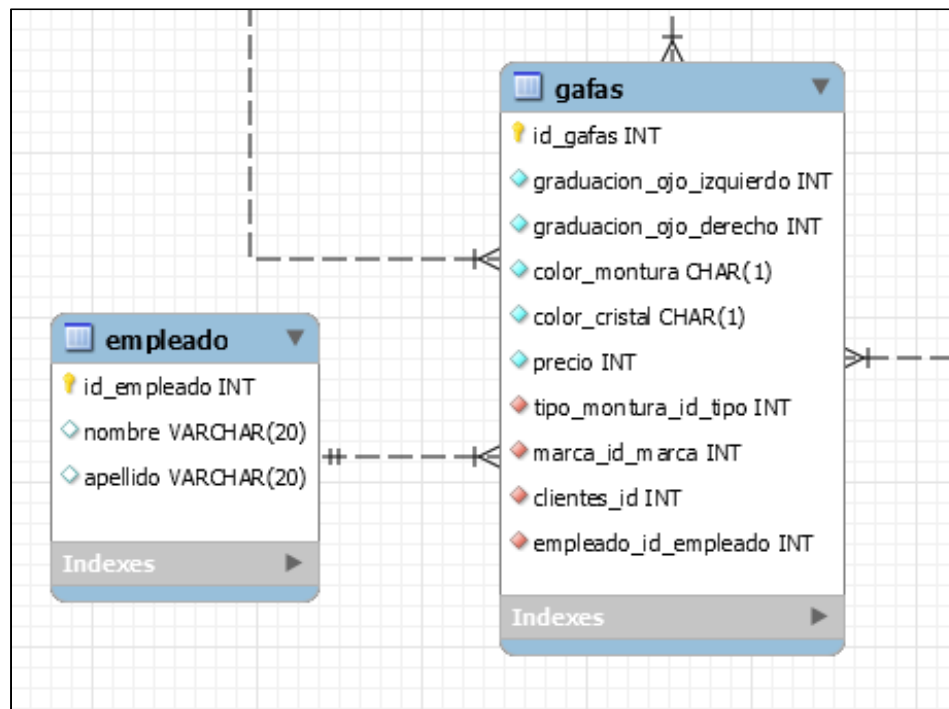
# 1. EXEMPLE OPTICA MYSQL

**Pas 3.** Dels clients vol emmagatzemar el nom, l'adreça postal, el telèfon, el correu electrònic i la data de registre. També ens demanen, quan arriba un client nou, d'emmagatzemar el client que li ha recomanat l'establiment (sempre i quan algú li hagi recomanat).



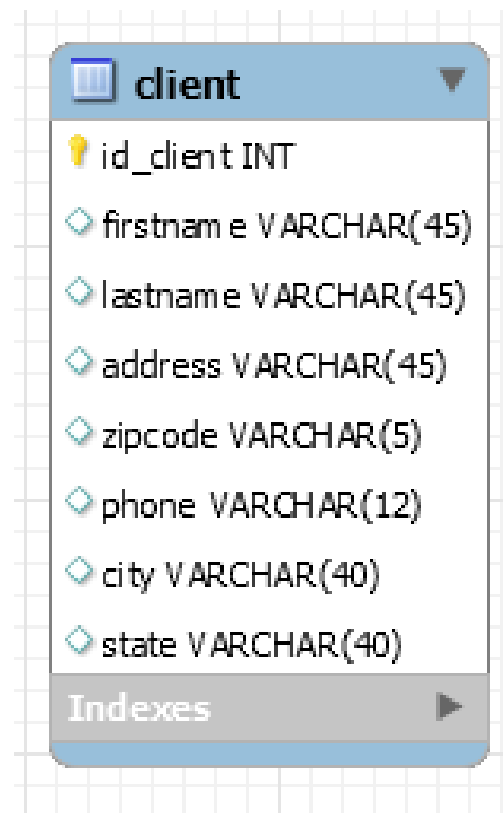
# 1. EXEMPLE OPTICA MYSQL

**Pas 4.** El nostre sistema haurà d'indicar qui ha sigut l'empleat que ha venut cada ullera.



## 2. EXEMPLE PIZZERIA MYSQL

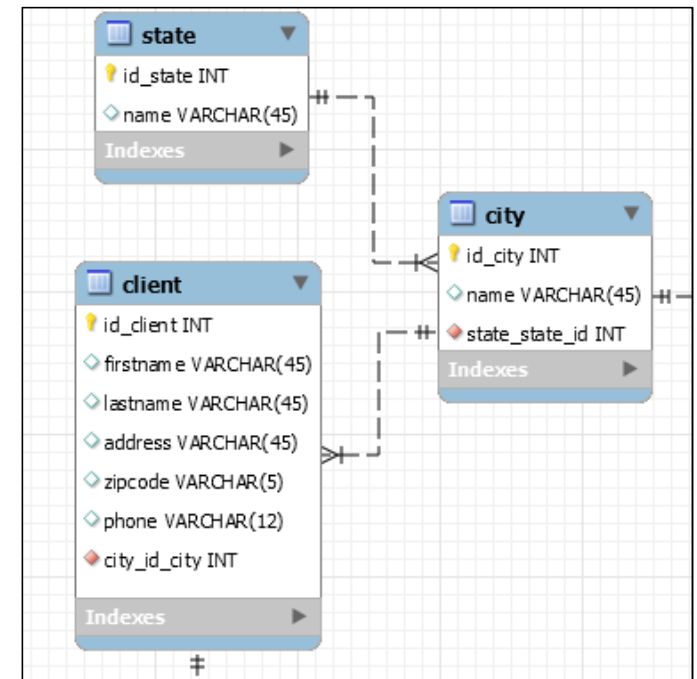
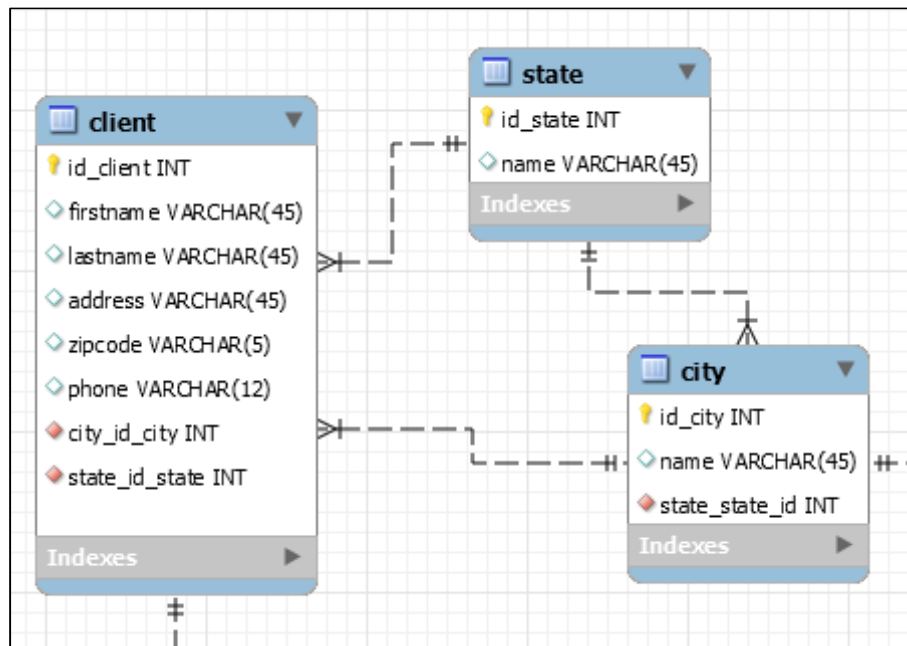
**Pas 1.** Per a cada client emmagatzemem un identificador únic, nom, cognoms, adreça, codi postal, localitat, província i número de telèfon.



## 2. EXEMPLE PIZZERIA MYSQL

**Pas 2.** Les dades de localitat i província estaran emmagatzemats en taules separades. Sabem que una localitat pertany a una única província, i que una província pot tenir moltes localitats.

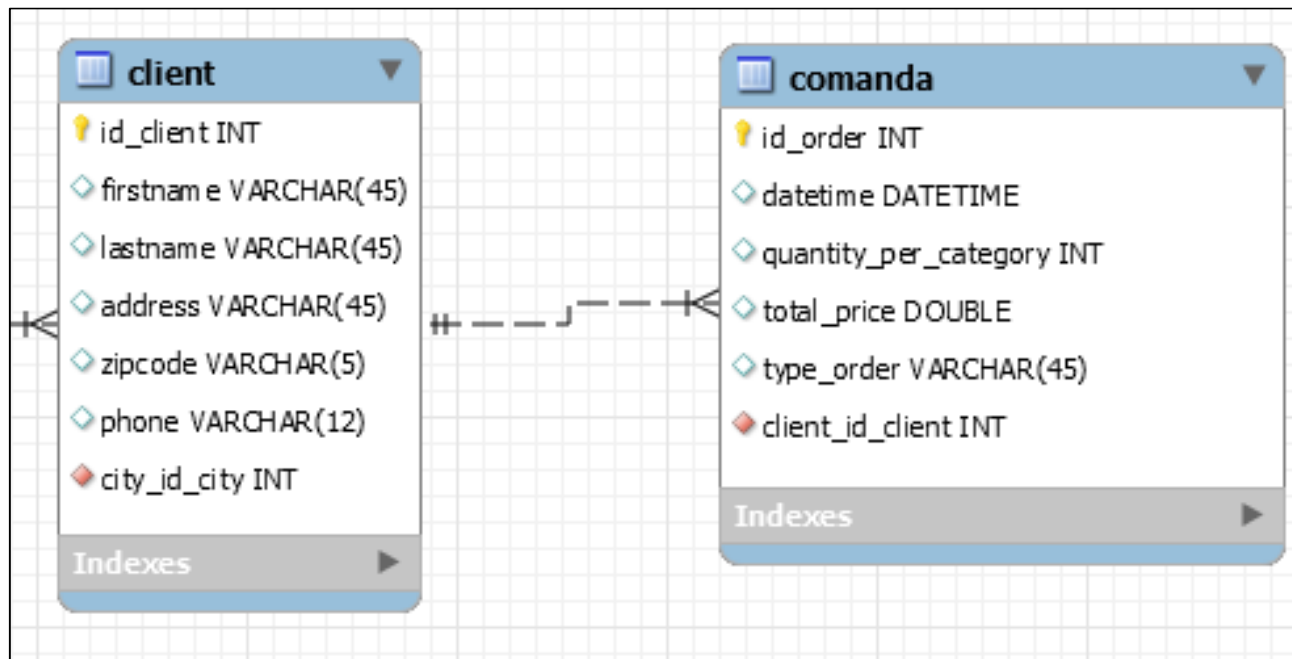
**Pas 3.** Per a cada localitat emmagatzemem un identificador únic i un nom. Per a cada província emmagatzemem un identificador únic i un nom.





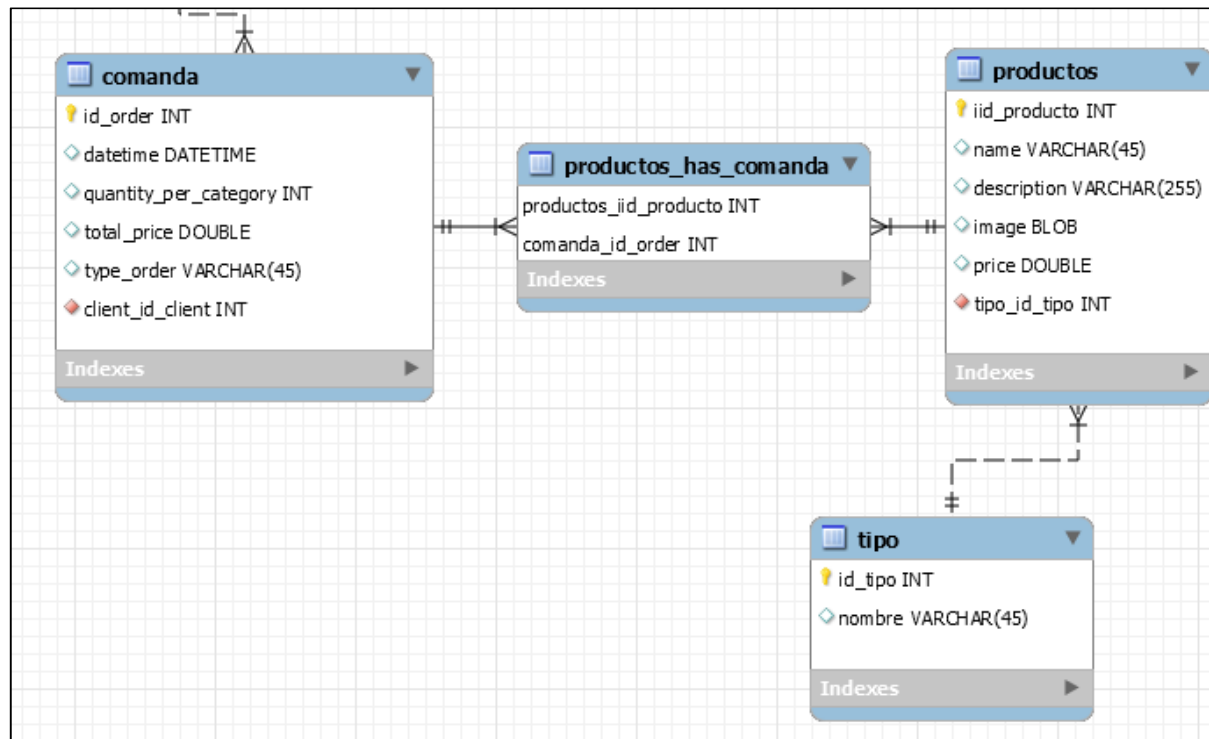
## 2. EXEMPLE PIZZERIA MYSQL

**Pas 4.** Un client pot realitzar moltes comandes, però una única comanda només pot ser realitzat per un únic client. De cada comanda s'emmagatzema un identificador únic, data/hora, si la comanda és per a repartiment a domicili o per a recollir en botiga, la quantitat de productes que s'han seleccionat de cada tipus i el preu total.



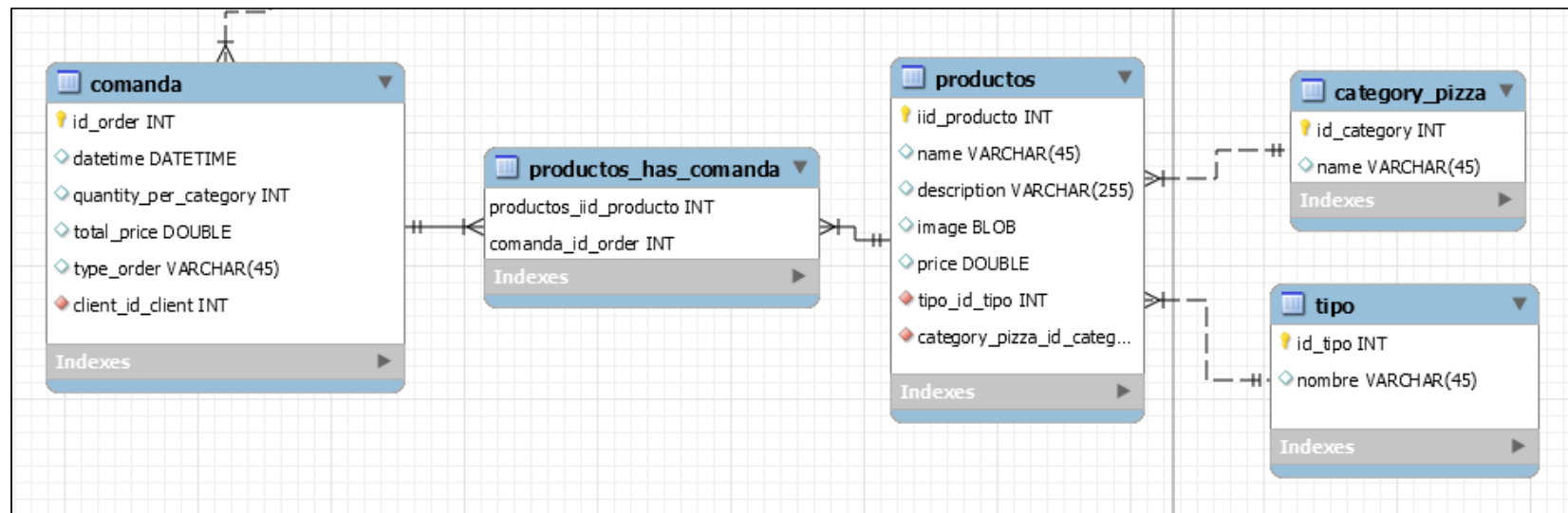
## 2. EXEMPLE PIZZERIA MYSQL

**Pas 5.** Una comanda pot constar d'un o diversos productes. Els productes poden ser pizzas, hamburgueses i begudes. De cada producte s'emmagatzema: un identificador únic, nom, descripció, imatge i preu.



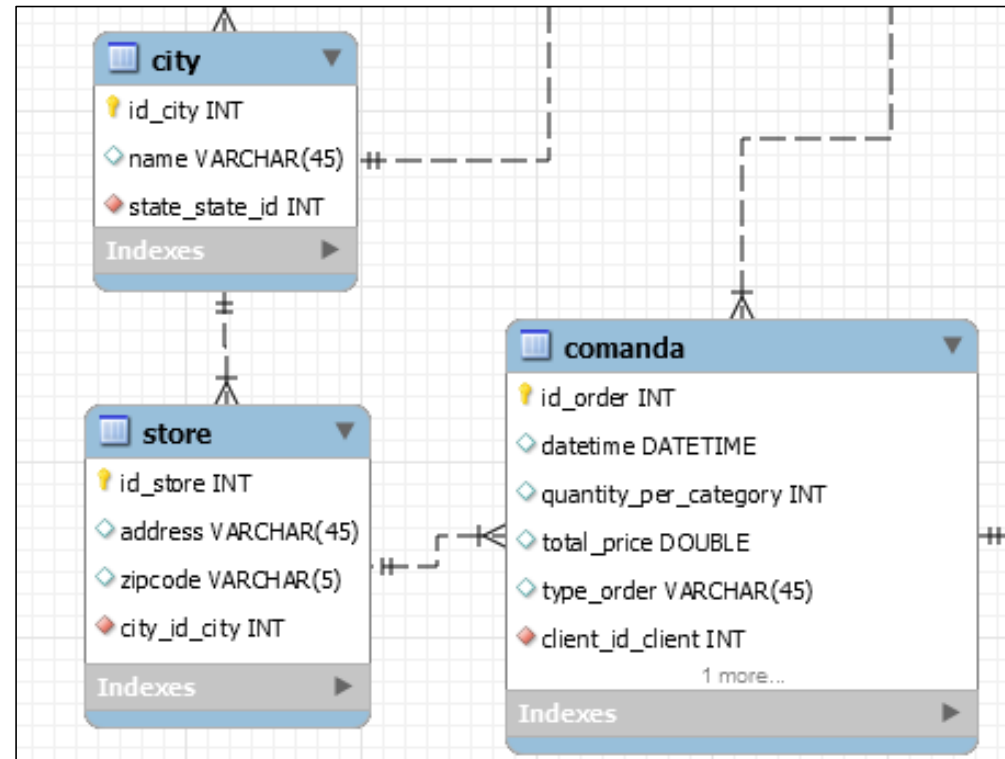
## 2. EXEMPLE PIZZERIA MYSQL

**Pas 6.** En el cas de les pizzes existeixen diverses categories que poden anar canviant de nom al llarg de l'any. Una pizza només pot estar dins d'una categoria, però una categoria pot tenir moltes pizzes. De cada categoria s'emmagatzema un identificador únic i un nom.



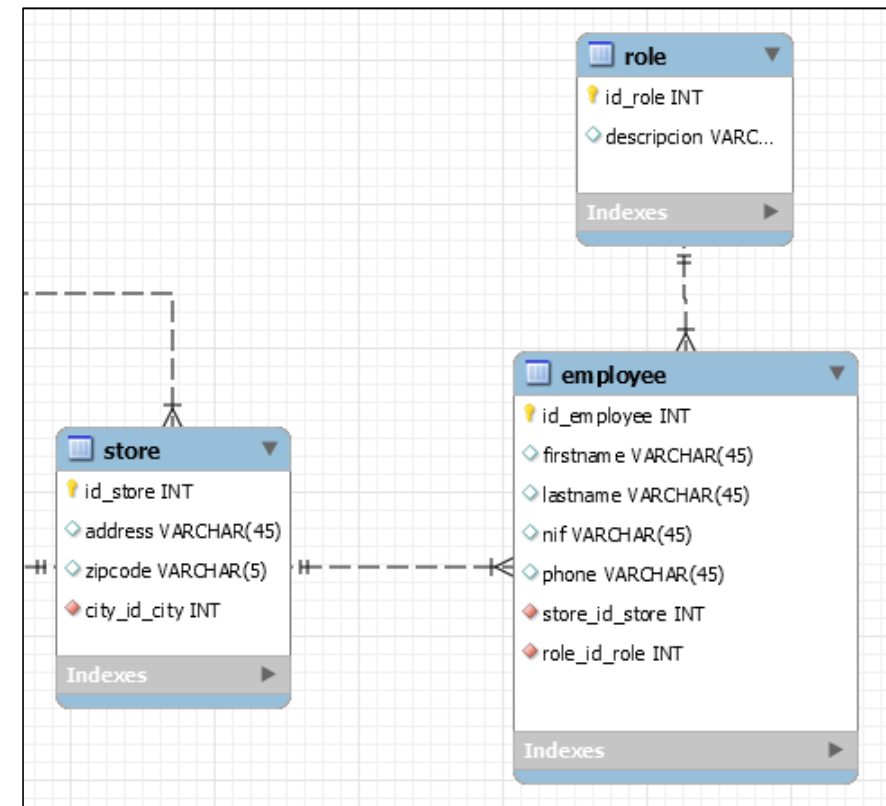
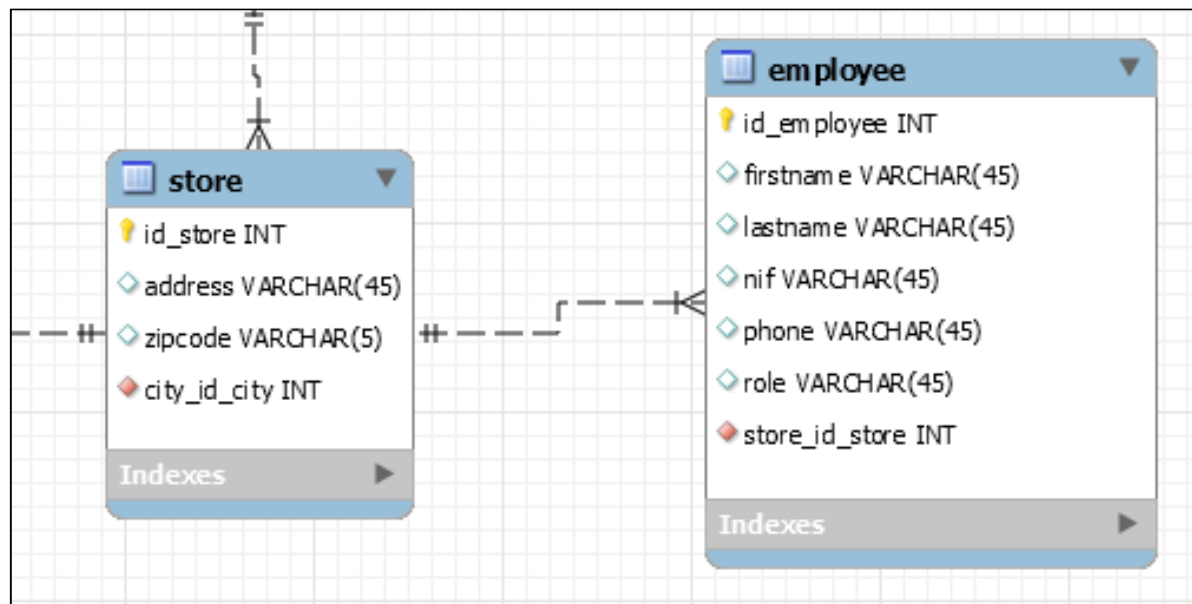
## 2. EXEMPLE PIZZERIA MYSQL

**Pas 7.** Una comanda és gestionada per una única botiga i una botiga pot gestionar moltes comandes. De cada botiga s'emmagatzema un identificador únic, adreça, codi postal, localitat i província.



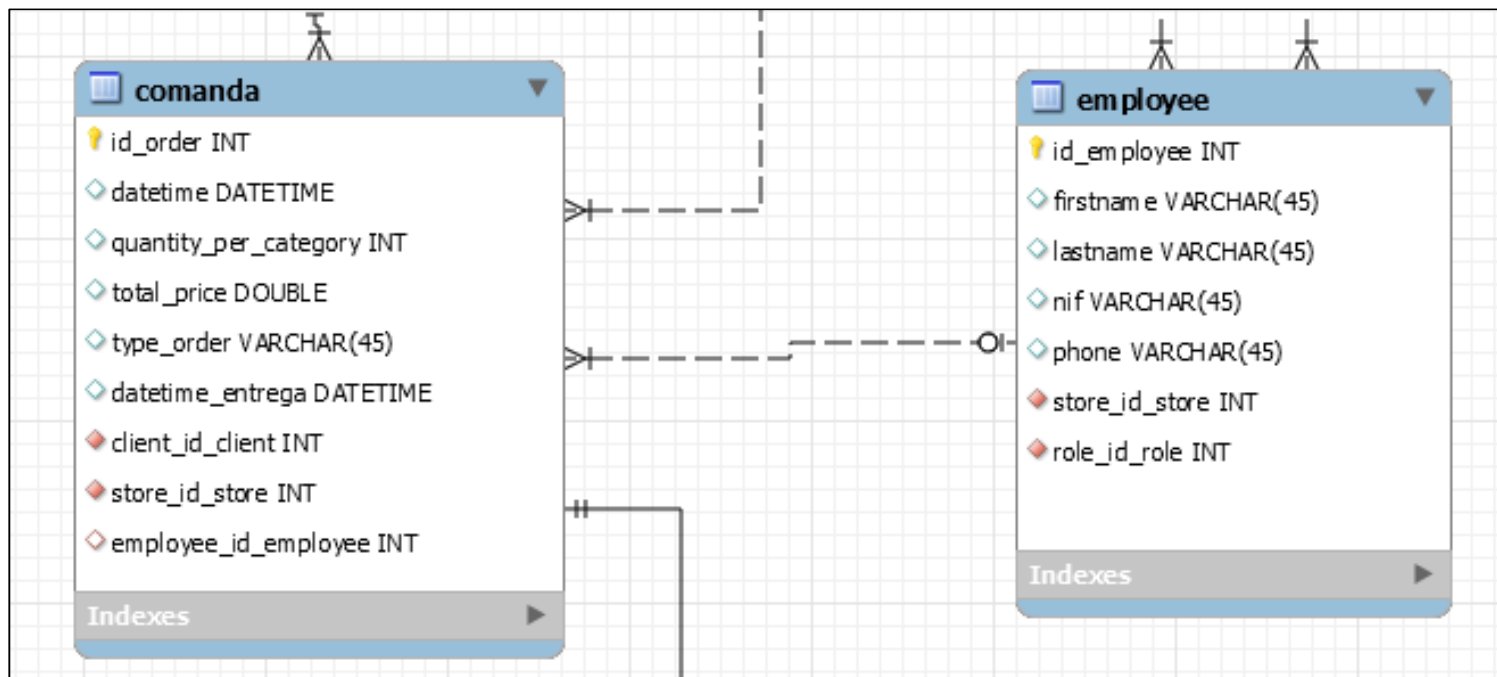
## 2. EXEMPLE PIZZERIA MYSQL

**Pas 8.** En una botiga poden treballar molts empleats i un empleat només pot treballar en una botiga. De cada empleat s'emmagatzema un identificador únic, nom, cognoms, nif, telèfon i si treballa com a cuiner o repartidor.



## 2. EXEMPLE PIZZERIA MYSQL

**Pas 9.** Per a les comandes de repartiment a domicili interessa guardar qui és el repartidor que realitza el lliurament de la comanda i la data/hora del moment del lliurament.



# 3. MODELAT DADES MONGO

## Format JSON

### Cells in a Table

#### Cars

id	owner	make	model	year
007	Daniel	Ferrari	GTS	1982
008	Daniel	Fiat	500S	2013

### Fields as Keys/Values

Prefix (keys) with column names

```
{
  "_id": 007,
  "owner": "Daniel",
  "make": "Ferrari",
  "model": "GTS",
  "year": 1982
}
```

# 3. MODELAT DADES MONGO

## Differences: Tabular vs Document

	Tabular	MongoDB
Steps to create the model	1 – define schema 2 – develop app and queries	1 – identifying the queries 2 – define schema
Initial schema	<ul style="list-style-type: none"><li>• 3<sup>rd</sup> normal form</li><li>• one possible solution</li></ul>	<ul style="list-style-type: none"><li>• many possible solutions</li></ul>
Final schema	<ul style="list-style-type: none"><li>• likely denormalized</li></ul>	<ul style="list-style-type: none"><li>• few changes</li></ul>
Schema evolution	<ul style="list-style-type: none"><li>• difficult and not optimal</li><li>• likely downtime</li></ul>	<ul style="list-style-type: none"><li>• easy</li><li>• no downtime</li></ul>
Query performance	<ul style="list-style-type: none"><li>• mediocre</li></ul>	<ul style="list-style-type: none"><li>• optimized</li></ul>



# 3. MODELAT DADES MONGO

---

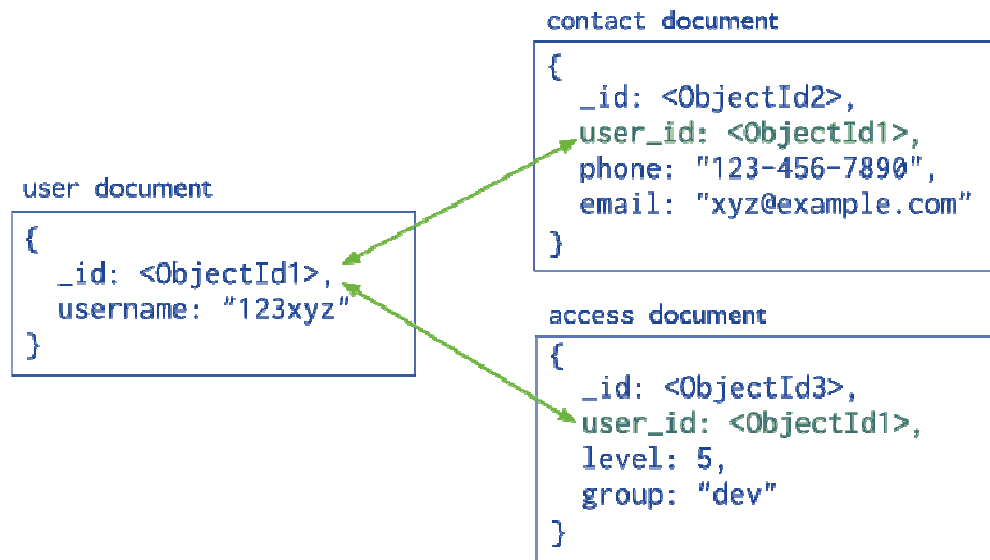
## Disseny de l'esquema

- Enfoc diferent al relacional
- No 3FN → tendència a denormalitzar
- *MongoDB* no suporta transaccions
  - Assegura que les operacions son atòmiques
  - Solució:
    1. Reestructurar el codi per que tota la informació estigui continguda en un únic document.
    2. Implementar un sistema de bloqueig per software (semàfor, etc... ).
    3. Tolerar un grau de inconsistència en el sistema.
- **Denormalitzar** les dades per minimitzar la redundància però facilitant que mitjançant operacions atòmiques es mantingui la integritat de les dades

# 3. MODELAT DADES MONGO

## Referències manuals

- Emmagatzemar el camp `_id` com clau aliena
- La aplicació realitza una 2<sup>a</sup> consulta per obtenir les dades relacionades.
- Son senzilles i suficients para la majoria dels casos d'us



```
var idUsuario = ObjectId();

db.usuario.insert({
  _id: idUsuario,
  nombre: "123xyz"
});

db.contacto.insert({
  usuario_id:idUsuario,
  telefono: "123-456-7890",
  email: "xyz@ejemplo.com"
});
```

# 3. MODELAT DADES MONGO

## DBRef

- Objecte que representa una referencia d'un document a un altre mitjançant el valor del camp `_id`, el nombre de la col·lecció i, opcionalment, el nombre de la base de dades
- `{"$ref" : <nombreColeccion>, "$id" : <valorCampo_id>, "$db" : <nombreBaseDatos> }`
- Permet referenciar documents localitzats en diferents col·leccions.
- En *Java*, mitjançant la classe `DBRef`

```
db.contacto.insert({
  usuario_id: new DBRef("usuario", idUsuario),
  telefono:  "123 456 7890",
  email:  "xyz@ejemplo.com"
});
```

# 3. MODELAT DADES MONGO

## Dades embegudes

- Mitjançant sub-documents
- Dintre d'un atribut o un array
- Permet obtenir totes les dades amb un accés
- Es recomana el seu ús en relacions:
  - 1:1
  - 1 a pocs
- Un documento BSON pot contenir un màxim de 16MB
- Si un documento creix molt → fer servir referències o GridFS

```
{
  _id: <ObjectId>,
  username: "123xyz",
  contact: {
    phone: "123-456-7890",
    email: "xyz@example.com"
  },
  access: {
    level: 5,
    group: "dev"
  }
}
```

Embedded sub-document

Embedded sub-document

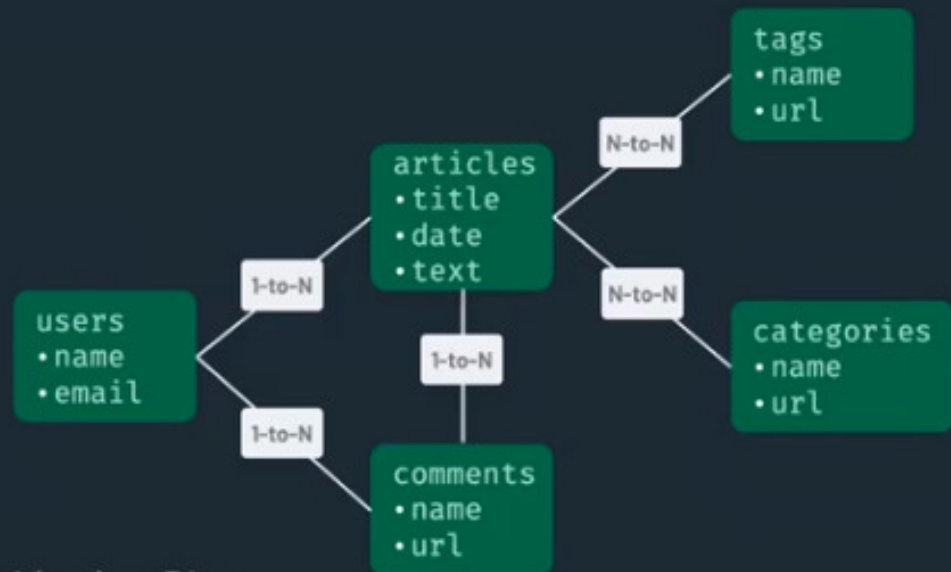
# 3. MODELAT DADES MONGO

N'hi ha 3 tipus de relacions:

- 1 a 1
- 1 a N
- N a M

Relationships:

- One-to-one
- One-to-many
- Many-to-many

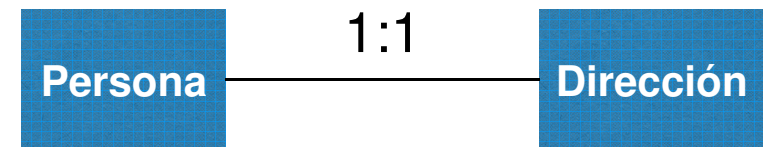


Example: Entities and relationships in a Blog

# 3. MODELAT DADES MONGO

## Relacions 1 : 1

- Embeure un documento dintre d'un altre
- Motius per no embeure:
  - Freqüència d'accés.
    - Si a un d'ells s'accedeix molt poc
    - Al separar-los → s'allibera memòria
- Mida dels elements.
  - Si hi ha un que es molt mes gran que l'altre
  - O si un es modifica moltes més vegades que l'altre



```
{
  nombre: "Aitor",
  edad: 38,
  direccion: {
    calle: "Mayor",
    ciudad: "Elx"
  }
}
```

## 3. MODELAT DADES MONGO

### Relacions 1 : 1 embeded

#### One-to-One Embedded

```
Book = {  
  "_id": 1,  
  "title": "Harry Potter and the Methods of Rationality",  
  "slug": "9781857150193-hpmor",  
  "author": {  
    "firstName": "Eliezer",  
    "lastName": "Yudkowsky"  
  },  
  // more fields follow...  
}
```

### 3. MODELAT DADES MONGO

## Relacions 1 : 1 enllaçades

## One-to-One Linked

```
Book = { // either side can track
  "_id": 1,
  "title": "Harry Potter and the Methods of Rationality",
  "slug": "9781857150193-hpmor",
  "author": 1, // more fields follow...
}

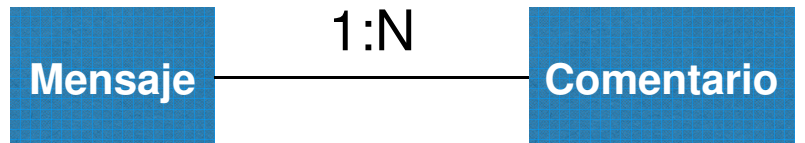
Author = {
  "_id": 1,
  "firstName": "Eliezer",
  "lastName": "Yudkowsky"
  "book": 1, // more fields follow...
}
```



# 3. MODELAT DADES MONGO

## Relacions 1 : N, 1 a pocs

- Embeure les dades
- Crear un array dintre de l'entitat 1
- El missatge conte un array de comentari



```
{
  titulo: "La broma asesina",
  url: "http://es.wikipedia.org/wiki/Batman:_The_Killing_Joke",
  text: "La dualidad de Batman y Joker",
  comentarios: [
    {
      autor: "Bruce Wayne",
      fecha: ISODate("2015-04-01T09:31:32Z"),
      comentario: "A mi me encantó"
    }, {
      autor: "Bruno Díaz",
      fecha: ISODate("2015-04-03T10:07:28Z"),
      comentario: "El mejor"
    }
  ]
}
```

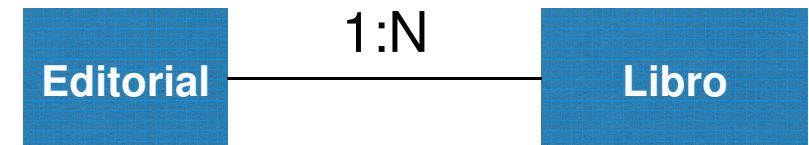
# 3. MODELAT DADES MONGO

## Relacions 1 : N (1 a molts)

- Referencia de N a 1
  - Igual que clau aliena
- Restricció 16MB BSON
- Es poden fer servir documents embeguts amb redundància de dades
  - Productes (pvp) d'una comanda
  - Adreça d'enviament d'un client

```
{
  _id: 1,
  nombre: "O'Reilly",
  pais: "EE.UU."
}
```

Editorial



```
{
  _id: 1234,
  titulo: "MongoDB: The Definitive Guide",
  autor: ["Kristina Chodorow", "Mike Dirolf"],
  numPaginas: 216,
  editorial_id: 1,
}
{
  _id: 1235,
  titulo: "50 Tips and Tricks for MongoDB Developer",
  autor: "Kristina Chodorow",
  numPaginas: 68,
  editorial_id: 1,
}
```

Libro

# 3. MODELAT DADES MONGO

## Relacions 1 : N

### One-to-Many: Array in Parent

```
Author= {
  "_id": 1,
  "firstName": "Eliezer",
  "lastName": "Yudkowsky",
  "books": [1, 5, 17],
  // more fields follow...
}
```

### One-to-Many: Scalar in Child

```
Book1= {
  "_id": 1,
  "title": "Harry Potter and the Methods of Rationality",
  "slug": "9781857150193-hpmor",
  "author": 1,           // more fields follow...
}

Book2= {
  "_id": 5,
  "title": "How to Actually Change Your Mind",
  "slug": "1939311179490-how-to-change",
  "author": 1,           // more fields follow...
}
```

# 3. MODELAT DADES MONGO

---

## Relacions N : M

- Solen ser relaciones pocs a pocs
- 3 possibilitats
  1. Enfoc relacional con col·lecció intermèdia
    - Desaconsellat → 3 consultes
  2. Dos documents, cada un amb un array que contingui els ids de l'altre document (*2 Way Embedding*).
    - Vigilar la inconsistència de datos
  3. Embeure un document dintre d'un altre (*One Way Embedding*)
    - No se recomana si algun de los documents pot créixer molt
    - Revisar si un document depèn de l'altre per a la seva creació

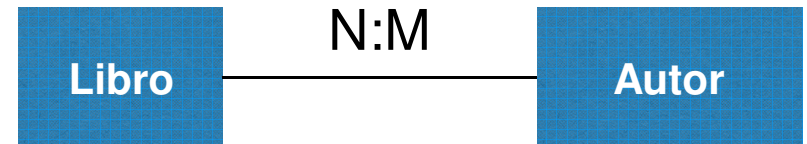
### 3. MODELAT DADES MONGO

#### Relacions N : M.

#### Many-to-Many: Arrays on either side

```
Book = {                                     //either side can track
  "_id": 5,
  "title": "Harry Potter and the Methods of Rationality",
  "slug": "9781857150193-hpmor",
  "authors": [1, 3],                        // more fields follow...
}

Author = {
  "_id": 1,
  "firstName": "Eliezer",
  "lastName": "Yudkowsky",
  "books": [5, 7],                          // more fields follow...
}
```

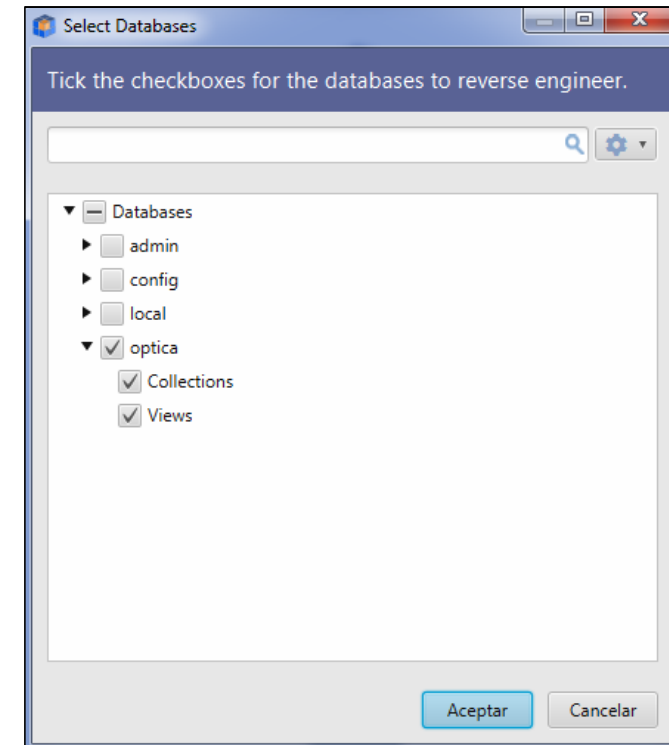
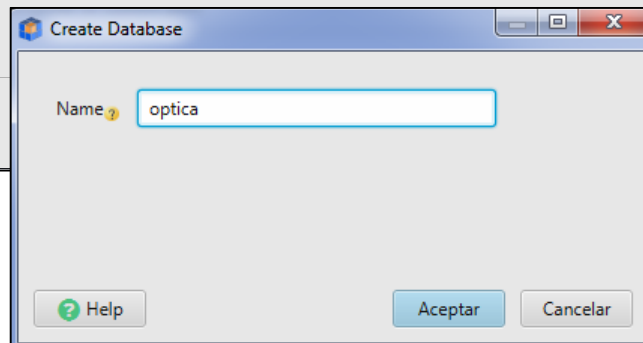
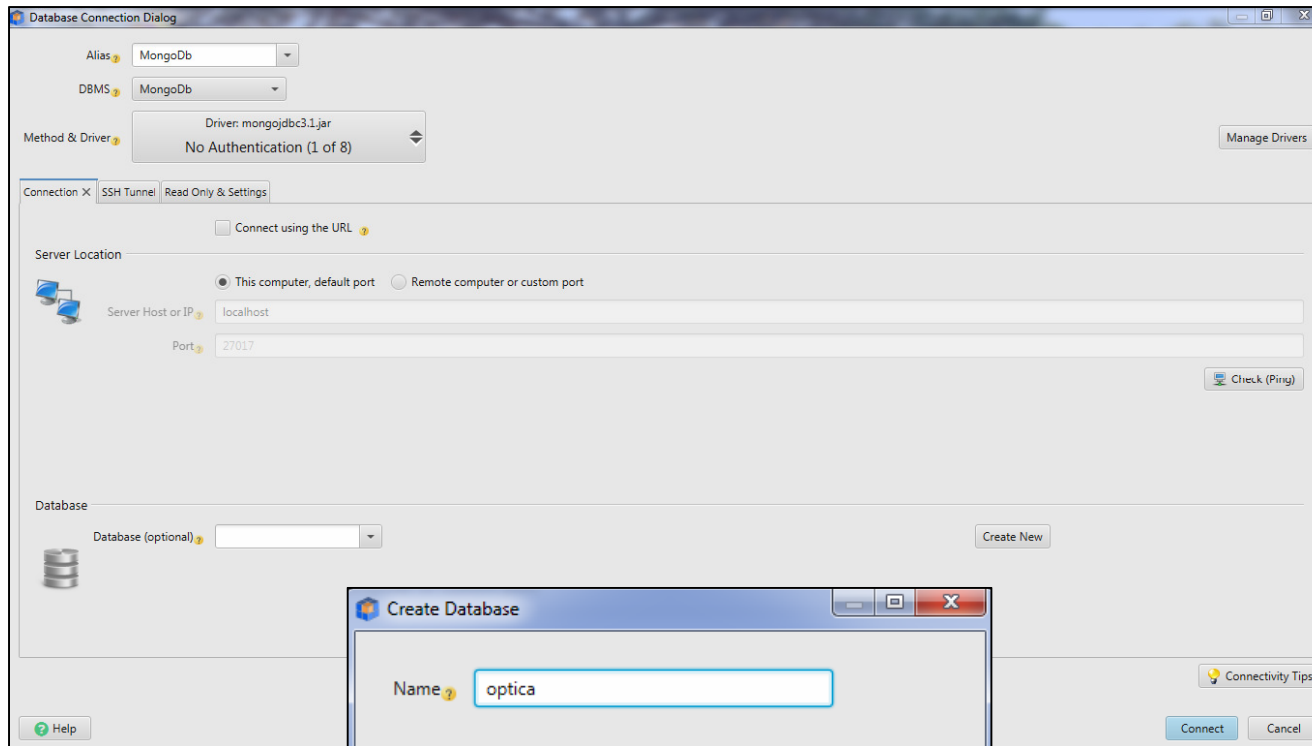


```
{
  _id: 1,
  titulo: "La historia interminable",
  anyo: 1979,
  autores: [1]
},{
  _id: 2,
  titulo: "Momo",
  anyo: 1973,
  autores: [1]
}, {
  _id: 1,
  nombre: "Michael Ende",
  pais: "Alemania",
  libros: [1,2]
}
```

2 way embedding

## 4. EXAMPLE OPTICA MONGO

### Pas 0. Creació base de dades òptica a traves de DBSchema



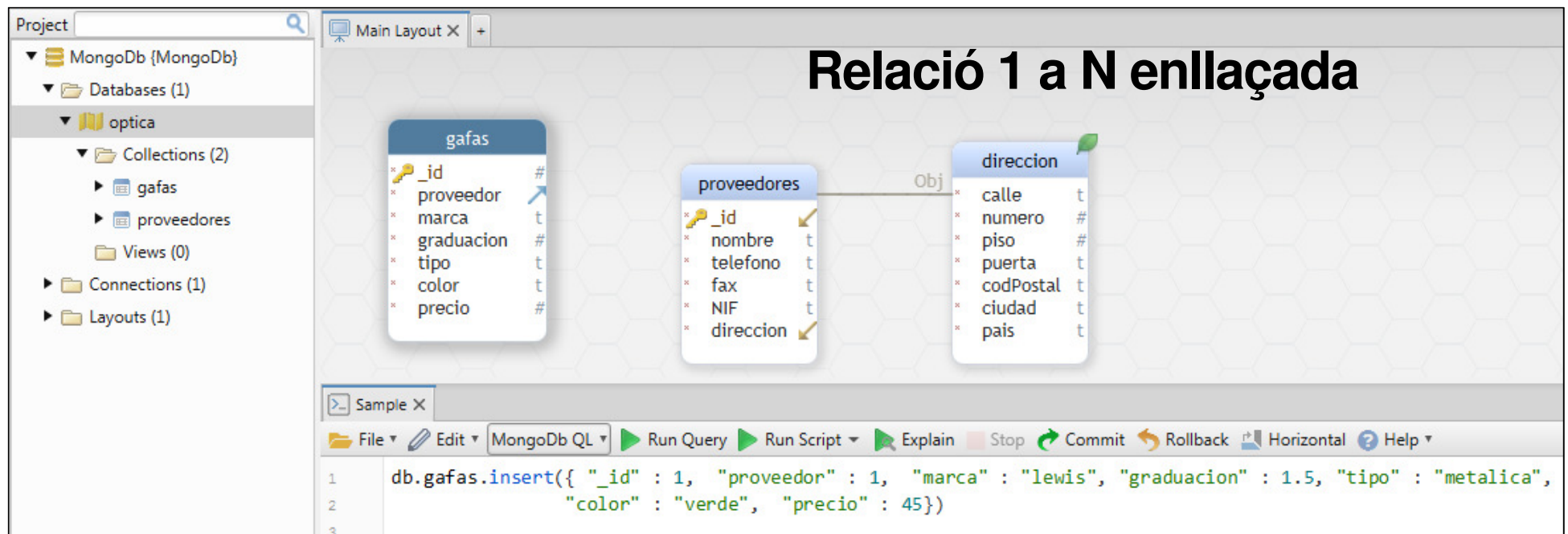
## 4. EXAMPLE OPTICA MONGO

**Pas 1.** En primer lloc l'òptica vol saber quin és el proveïdor de cadascuna de les ulleres. En concret vol saber de cada proveïdor el nom, l'adreça (carrer, número, pis, porta, ciutat, codi postal i país), telèfon, fax, NIF.

The diagram shows two collections: 'proveedores' and 'direccion'. The 'proveedores' collection has fields: \_id (key), nombre, telefono, fax, NIF, and direccion. The 'direccion' collection has fields: calle, numero, piso, puerta, codPostal, ciudad, and pais. A green arrow points from the 'direccion' field in 'proveedores' to the 'direccion' collection, indicating an embedded relationship. The text 'Relació 1 a 1 Embbbeded' is displayed in the center.

## 4. EXAMPLE OPTICA MONGO

**Pas 2.** La política de compres de l'òptica es basa en que les ulleres d'una marca es compraran a un únic proveïdor (així en podrà treure més bons preus), però poden comprar ulleres de diverses marques a un proveïdor. De les ulleres vol saber, la marca, la graduació de cadascun dels vidres, el tipus de muntura (flotant, pasta o metàl·lica), el color de la muntura, el color de cada vidre i el preu.





## 4. EXAMPLE OPTICA MONGO

**Pas 3.** Es pot establir una relació entre proveedor de gafas i el id de proveedores però es merament descriptiva: No evita que es puguin insertar elements referenciats que no existeixen

New Foreign Key

Name (mixed)

Description

Definition X Optimizations

Collection optica.gafas references

Database  Collection

From Field	To Field
proveedor	_id

+ Add Remove Up Down

Type 1:1 or many

On Delete  On Update

☒ Virtual (only in DbSchema)

Help Aceptar Cancelar



### How to Create Foreign Key



Create Foreign Key by drag and drop of one column over the PK (referred) column.  
Clone columns by drop over the table header.  
Using drag and drop you can also re-order the columns.

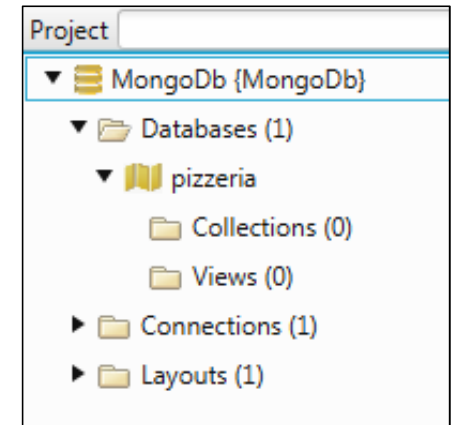
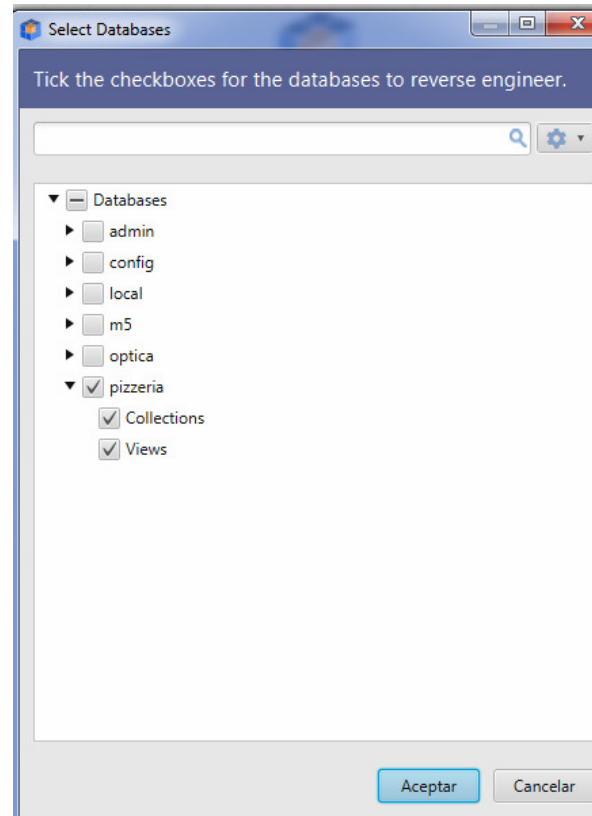
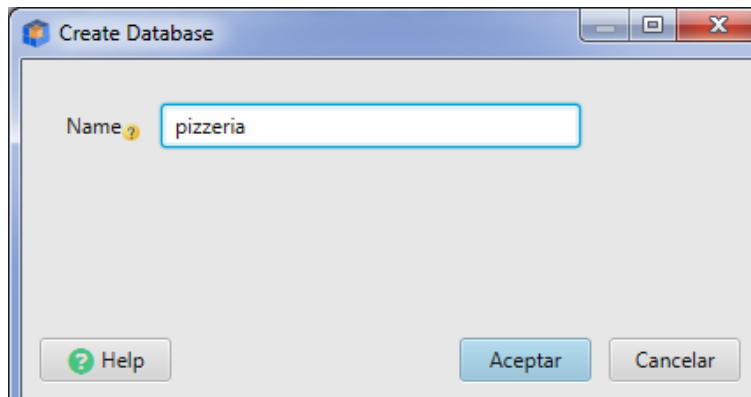
## 4. EXAMPLE OPTICA MONGO

**Pas 4.** Dels clients vol emmagatzemar el nom, l'adreça postal, el telèfon, el correu electrònic i la data de registre. També ens demanen, quan arriba un client nou, d'emmagatzemar el client que li ha recomanat l'establiment (sempre i quan algú li hagi recomanat). El nostre sistema haurà d'indicar qui ha sigut l'empleat que ha venut cada ullera.



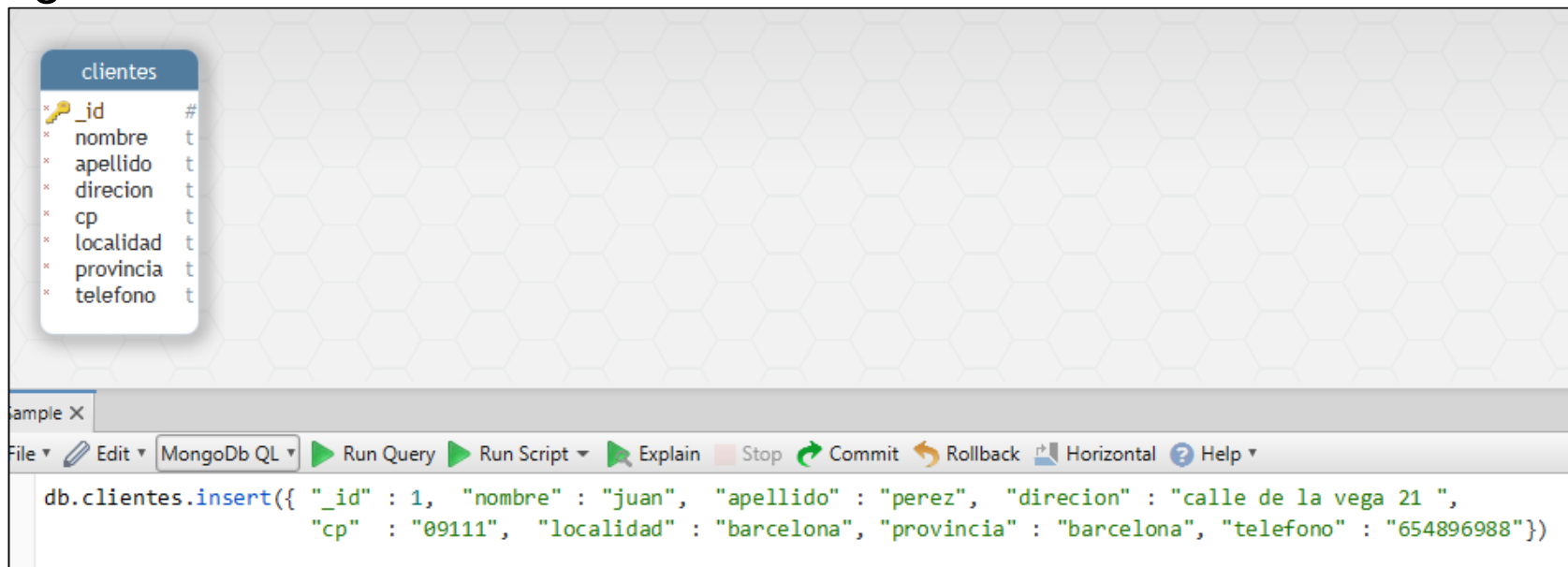
# 5. EXAMPLE PIZZERIA MONGO

**Pas 0.** S'ha de crear la base de dades pizzeria:



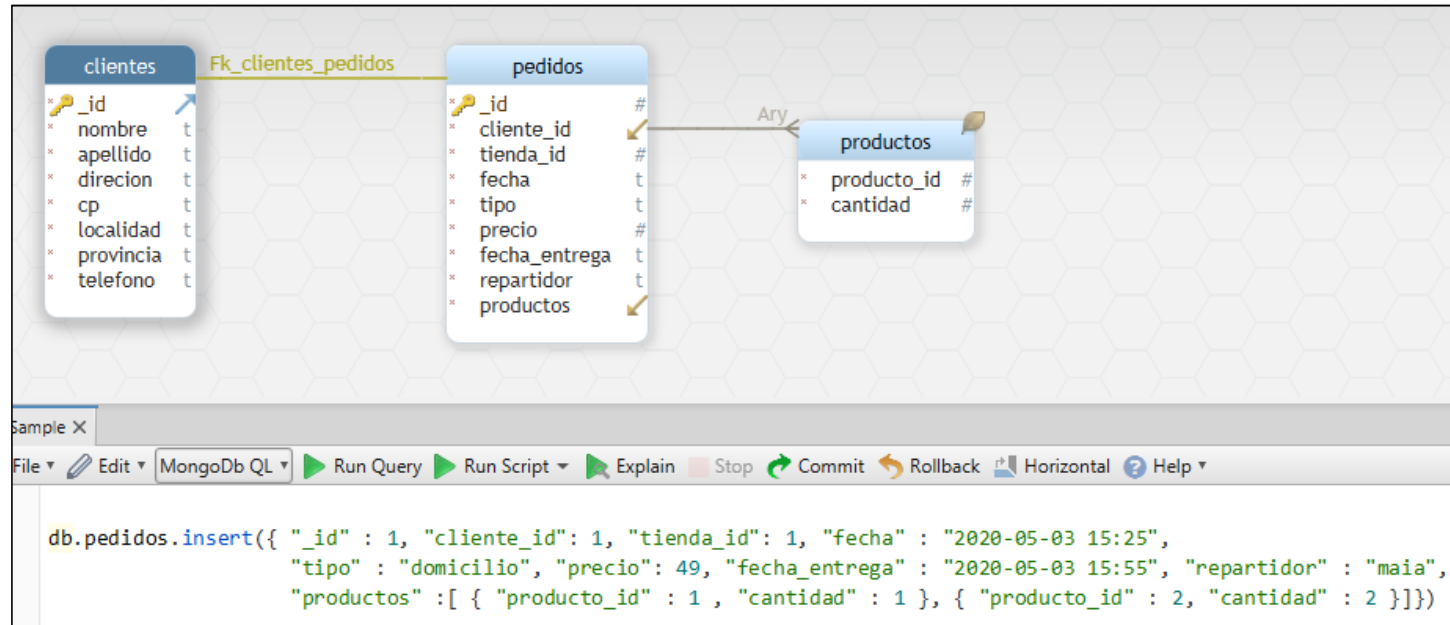
## 5. EXAMPLE PIZZERIA MONGO

**Pas 1.** Per a cada client emmagatzemem un identificador únic, nom, cognoms, adreça, codi postal, localitat, província i número de telèfon. ~~Les dades de localitat i província estaran emmagatzemats en taules separades. Sabem que una localitat pertany a una única província, i que una província pot tenir moltes localitats. Per a cada localitat emmagatzemem un identificador únic i un nom. Per a cada província emmagatzemem un identificador únic i un nom.~~



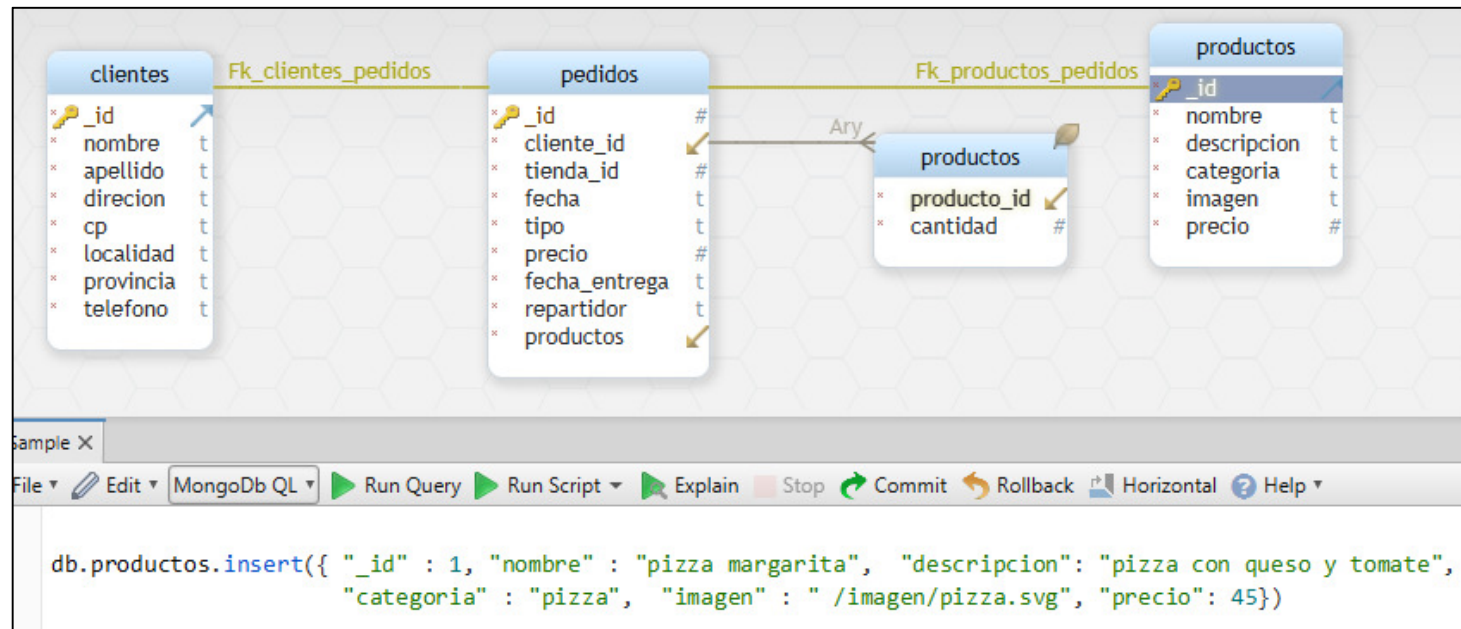
## 5. EXAMPLE PIZZERIA MONGO

**Pas 2.** Un client pot realitzar moltes comandes, però una única comanda només pot ser realitzat per un únic client. De cada comanda s'emmagatzema un identificador únic, data/hora, si la comanda és per a repartiment a domicili o per a recollir en botiga, la quantitat de productes que s'han seleccionat de cada tipus i el preu total. Una comanda pot constar d'un o diversos productes. Una comanda és gestionada per una única botiga i una botiga pot gestionar moltes comandes. Per a les comandes de repartiment a domicili interessa guardar qui és el repartidor que realitza el lliurament de la comanda i la data/hora del moment del lliurament.



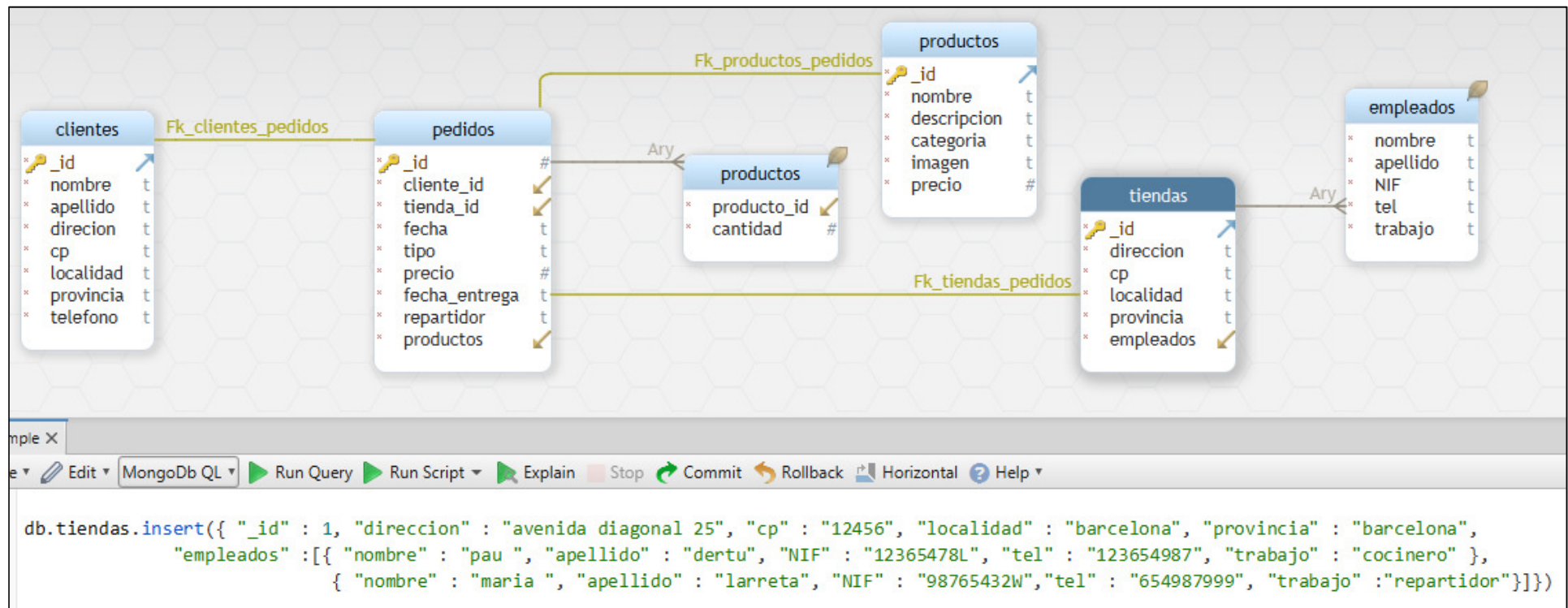
## 5. EXAMPLE PIZZERIA MONGO

**Pas 3.** Els productes poden ser pizzas, hamburgueses i begudes. De cada producte s'emmagatzema: un identificador únic, nom, descripció, imatge i preu. En el cas de les pizzas existeixen diverses categories que poden anar canviant de nom al llarg de l'any. Una pizza només pot estar dins d'una categoria, però una categoria pot tenir moltes pizzas. De cada categoria s'emmagatzema un identificador únic i un nom.



## 5. EXAMPLE PIZZERIA MONGO

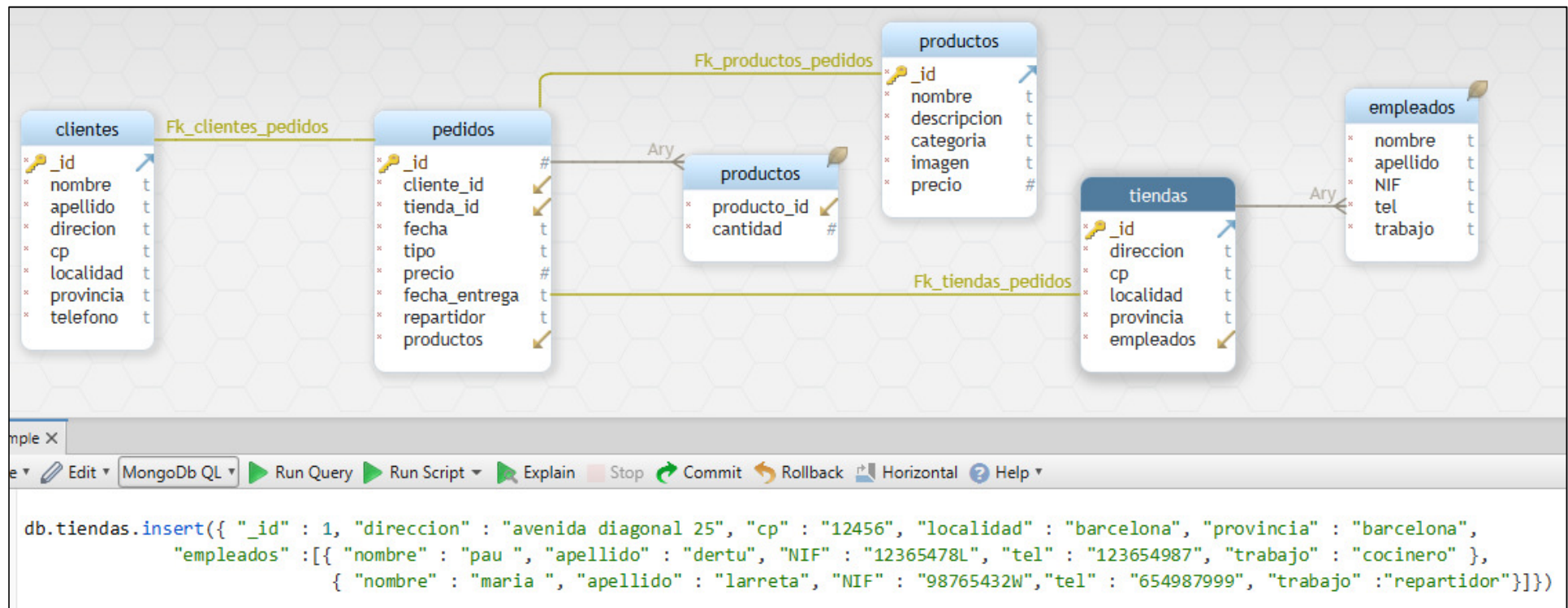
**Pas 4.** De cada botiga s'emmagatzema un identificador únic, adreça, codi postal, localitat i província. En una botiga poden treballar molts empleats i un empleat només pot treballar en una botiga. De cada empleat s'emmagatzema un identificador únic, nom, cognoms, nif, telèfon i si treballa com a cuiner o repartidor.





## 5. EXAMPLE PIZZERIA MONGO

**Pas 4.** De cada botiga s'emmagatzema un identificador únic, adreça, codi postal, localitat i província. En una botiga poden treballar molts empleats i un empleat només pot treballar en una botiga. De cada empleat s'emmagatzema un identificador únic, nom, cognoms, nif, telèfon i si treballa com a cuiner o repartidor.





## 6. PRACTICA DISSENY YOUTUBE

---

- De cada usuari guardem un identificador únic, email, password, nom d'usuari, data de naixement, sexe, país, codi postal.
- Un usuari publica vídeos.
- De cada vídeo guardem un identificador únic, un títol, una descripció, una grandària, el nom de l'arxiu de vídeo, durada del vídeo, un thumbnail, el nombre de reproduccions, el número de likes, el número de dislikes.
- Un vídeo pot tenir tres estats diferents: públic, ocult i privat.
- Un vídeo pot tenir moltes etiquetes. Una etiqueta s'identifica per una Identificador únic i un nom d'etiqueta.
- Interessa guardar qui és l'usuari que publica el vídeo i en quina data/hora el fa.
- Un usuari pot crear un canal. Un canal té un identificador únic, un nom, una descripció i una data de creació.
- Un usuari es pot subscriure als canals d'altres usuaris.

## 6. PRACTICA DISSENY YOUTUBE

---

- Un usuari pot donar-li un like o un dislike a un vídeo una única vegada. Caldrà portar un registre dels usuaris que li han donat like i dislike a un determinat vídeo i en quina data/hora ho van fer.
- Un usuari pot crear playlists amb els vídeos que li agraden. Cada playlist té un identificador únic, un nom, una data de creació, i un estat que indica que pot ser pública o privada.
- Un usuari pot escriure comentaris en un vídeo determinat. Cada comentari està identificat per un identificador únic, el text del comentari i la data/hora en la qual es va realitzar.
- Un usuari pot marcar un comentari com m'agrada o no m'agrada. Caldrà portar un registre dels usuaris que han marcat un comentari com m'agrada/no m'agrada, i en quina data/hora ho van fer.

## 7. PRACTICA DISSENY SPOTIFY

---

- Existeixen dos tipus d'usuaris: usuari free i usuari premium.
- De cada usuari guardem un identificador únic, email, password, nom d'usuari, data de naixement, sexe, país, codi postal.
- Els usuaris premium realitzen subscripcions. Les dades necessàries que caldrà guardar per a cada subscripció són: data d'inici de la subscripció, data de renovació del servei i una forma de pagament, que pot ser mitjançant targeta de crèdit o PayPal.
- De les targetes de crèdit guardem el número de targeta, mes i any de caducitat i el codi de seguretat.
- Dels usuaris que paguen amb PayPal guardem el nom d'usuari de PayPal.
- Ens interessa portar un registre de tots els pagaments que un usuari premium ha anat realitzant durant el període que està subscrit. De cada pagament es guarda la data, un número d'ordre (que és únic) i un total.
- Un usuari pot crear moltes playlists. De cada playlist guardem un títol, el nombre de cançons que conté, un identificador únic i una data de creació.

## 7. PRACTICA DISSENY SPOTIFY

---

- Quan un usuari esborra una playlist no s'esborra del sistema, sinó que es marca com que ha estat eliminada. D'aquesta manera l'usuari pot tornar a recuperar els seus playlists en cas que les hagi eliminat per error. És necessari emmagatzemar la data en la qual uneixi playlist ha estat marcada com eliminada.
- Podem dir que existeixen dos tipus de playlists: actives i esborrades.
- Una playlist que està activa pot ser compartida amb altres usuaris, això vol dir que altres usuaris poden afegir cançons en ella. En una llista compartida ens interessa saber quin usuari ha estat el que ha afegit cada cançó i en quina data ho va fer.
- Una cançó només pot pertànyer a un únic àlbum. Un àlbum pot contenir moltes cançons. Un àlbum ha estat publicat per un únic artista. Un artista pot haver publicat molts àlbums.
- De cada cançó guardem un identificador únic, un títol, una durada i el nombre de vegades que ha estat reproduïda pels usuaris de Spotify.
- De cada àlbum guardem un identificador únic, títol, any de publicació i una imatge amb la portada.

## 7. PRACTICA DISSENY SPOTIFY

---

- De cada artista guardem un identificador únic, nom i una imatge de l'artista.
- Un usuari pot seguir a molts artistes.
- Un artista pot estar relacionat amb altres artistes que facin música semblant. De manera que Spotify pugui mostrar-nos un llistat d'artistes relacionats amb els artistes que ens agraden.
- També ens interessa guardar quins són els àlbums i les cançons favorites d'un usuari. Un usuari pot seleccionar molts àlbums i moltes cançons com a favorites.