

Data: 06/02/2018

Autor: JRVG

## Contingut:

1	Exercicis .....	1
1.1	1) Vueling ( <u>mydb01plane</u> ) .....	1
1.2	2) botiga de quadres ( <u>mydb02botigaquadres</u> ) .....	2
1.3	3) Youtube – Stube ( <u>mydb03swideos</u> ) .....	3
1.4	4) Catàleg de llibres ( <u>mydb04catalegllibres</u> ) .....	4
1.5	5) Xarxa social, fotografies ( <u>mydb05xsfotos</u> ) .....	5

## 1 Exercicis

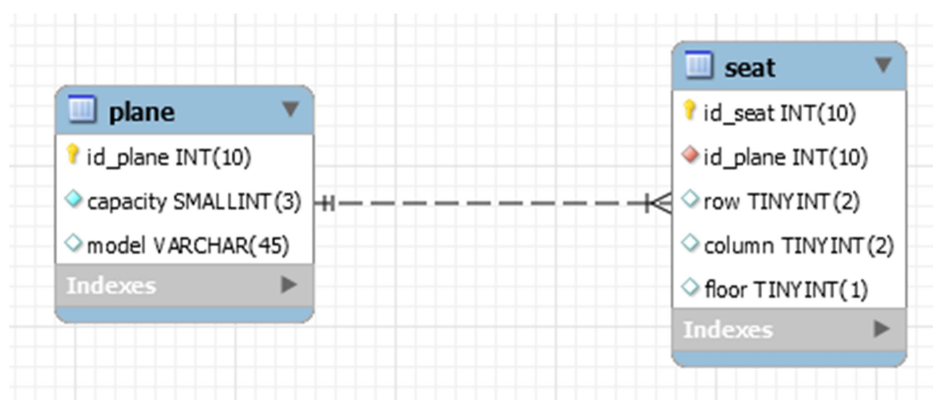
### Exercicis creació estructura de dades

**Objectiu:** Crea el model de base de dades pels següents enunciats. Crear el model amb algun editor de diagrames (per exemple draw.io, cacao.com/es/) i entregar-ho en un PDF anomenat "itacademy\_bbdd\_[nomcognom alumne].pdf".

### **1.1 1) Vueling (mydb01plane)**

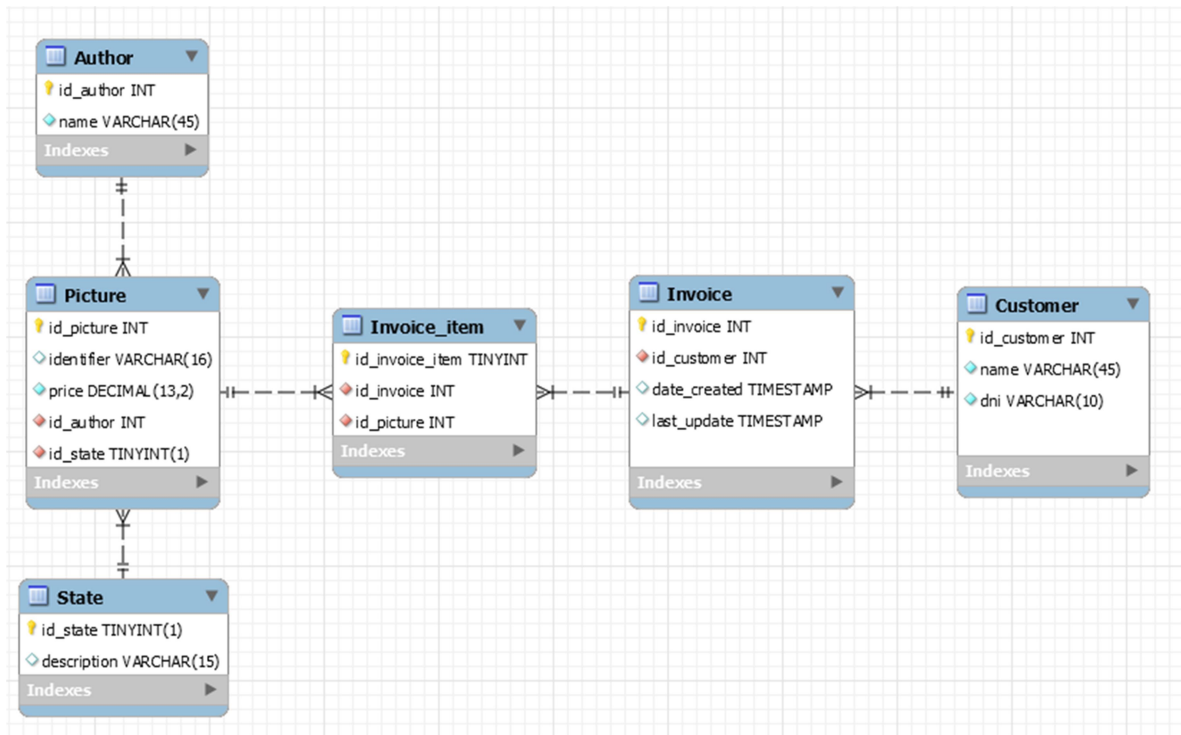
Vueling ens demana digitalitzar el seu sistema ja que fins ara ho fèiem tot a ma. (Per això hi havia tants retards).

Volen registrar cada avió que té la companyia segons un codi únic, número de persones que hi caben i el seu model. Cada avió té un número de seients identificats per un número únic.



## 1.2 2) botiga de quadres (mydb02botigaquadres)

Tenim una botiga de quadres. Volem emmagatzemar el stock de tots els quadres que hi ha a la venda. Un quadre té un identificador, un preu i un autor, tot i que hi ha quadres d'autor anònim. S'ha de deixar registrat els quadres que s'han venut, identificant el comprador amb un nom i un dni.



L'autor 'Anonymous' és un autor més de la llista de la taula 'Autor'.

Author - Table											
Table Name: Author		Schema: mydb02botigaquadres									
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression	
id_author	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
name	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'Anonymous'	

Una factura conté els items d'una venda a un client en una sola acció de venda. Cada quadre es detalla en una línia diferent de la factura.

Els quadres d'una factura es consideren venuts en una sola acció. Per aquest motiu he usat el tipus de dada TIMESTAMP per identificar la data i hora de la venda.

Invoice - Table											
Table Name: Invoice		Schema: mydb02botigaquadres									
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression	
id_invoice	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
id_customer	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
date_created	TIMESTAMP	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	DEFAULT CURRENT_TIMESTAMP	
last_update	TIMESTAMP	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP	

Column Name: last_update	Data Type: TIMESTAMP
Charset/Collation: Default Charset	Default: ON UPDATE CURRENT_TIMESTAMP
Comments:	Storage: <input type="radio"/> Virtual <input type="radio"/> Stored
	<input type="checkbox"/> Primary Key <input type="checkbox"/> Not Null <input type="checkbox"/> Unique
	<input type="checkbox"/> Binary <input type="checkbox"/> Unsigned <input type="checkbox"/> Zero Fill
	<input type="checkbox"/> Auto Increment <input type="checkbox"/> Generated

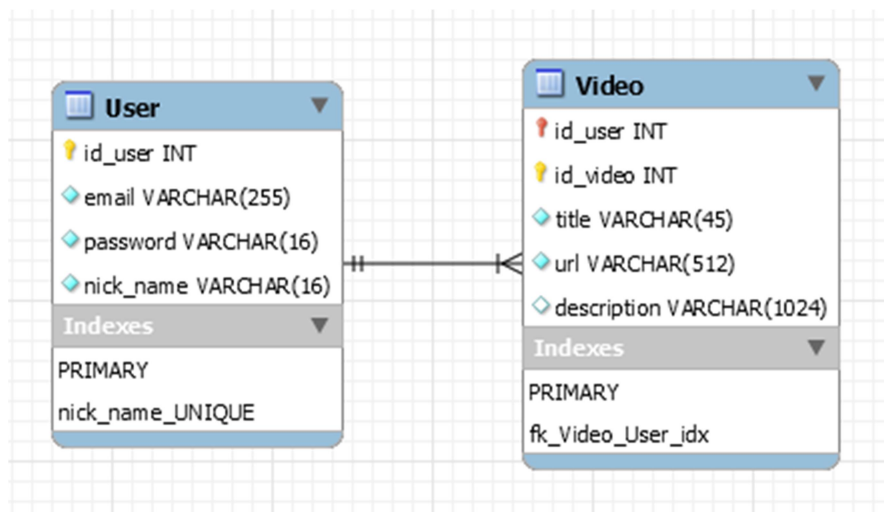
### 1.3 3) Youtube – Stube (mydb03swideos)

Youtube ha vist que som molt bons i ens vol contractar per fer un software.

En aquest software, que li direm Stube, un usuari tindrà un llistat de tots els vídeos que ha publicat.

Un usuari té un email, una contrasenya, un nom d'usuari únic.

Els vídeos tindran una descripció, un títol, una url del fitxer i un identificador que serà compost (id usuari, id vídeo).



User - Table x

Table Name:  Schema: **mydb03swideos**

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
id_user	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
email	VARCHAR(255)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
password	VARCHAR(16)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
nick_name	VARCHAR(16)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Column Name:  Data Type:

Charset/Collation:

Comments:

Storage: ☐ Virtual ☐ Stored

☐ Primary Key ☒ Not Null ☒ Unique

☐ Binary ☐ Unsigned ☐ Zero Fill

☐ Auto Increment ☐ Generated

According to RFC 5321, forward and reverse path can be up to 256 chars long, so the email address can be up to 254 characters long. You're safe with using 255 chars.

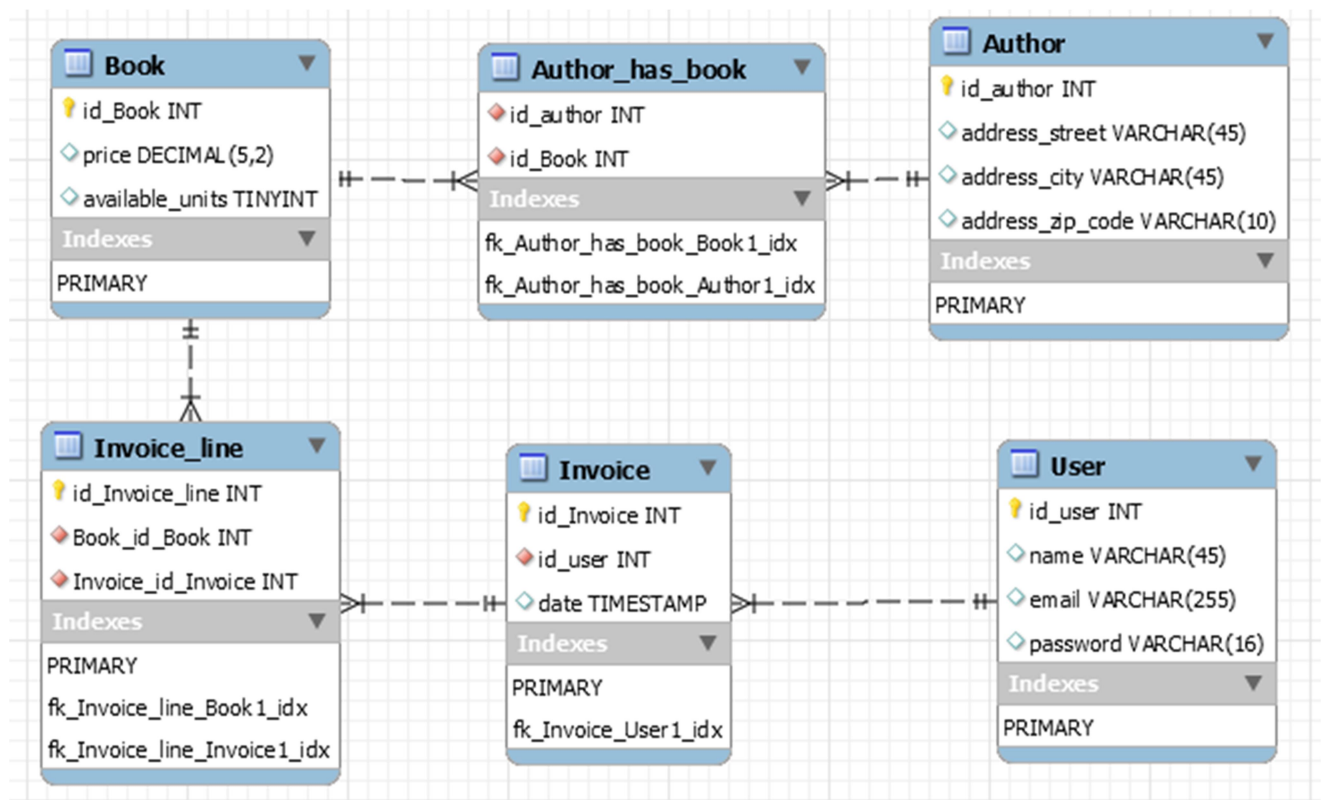
RFC 5321: Simple Mail Transfer Protocol

#### 1.4 4) Catàleg de llibres (*mydb04catalegllibres*)

Com som les feries de la selva, Amazon ens ha demanat que fem un software per ells. (Youtube no pagaven prou bé)

Volem tenir un catàleg de tots els llibres que hi ha a la venta, quantes unitats queden disponibles i el seu preu. Un autor tindrà una direcció i si ha escrit un o més llibres.

Un usuari té un nom d'usuari, un email i una contrasenya. Quan l'usuari fa una compra d'un o més llibres, s'emmagatzema una factura on queda referenciat cada un dels articles que ha comprat.

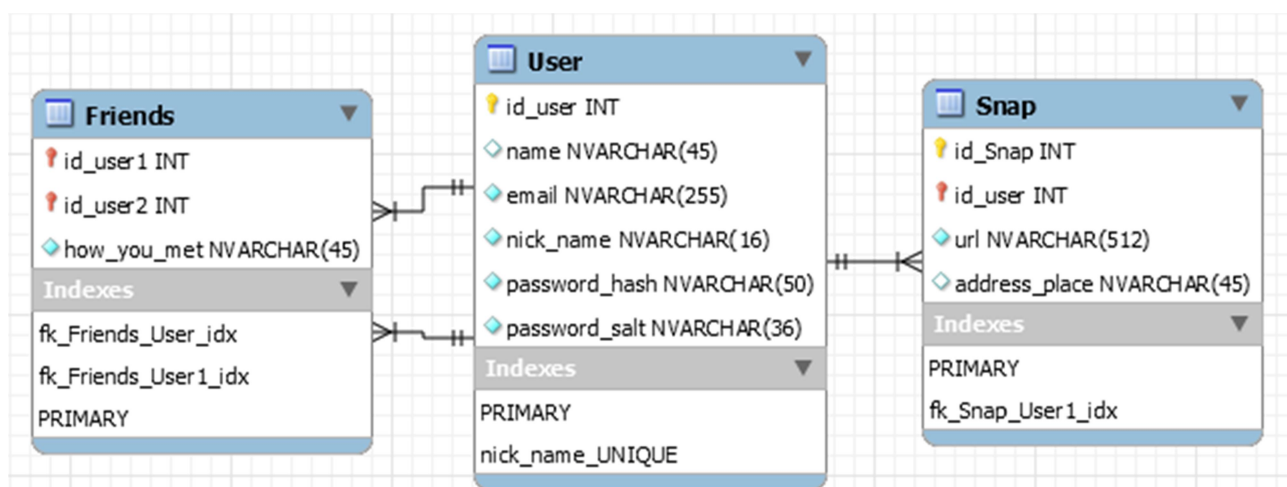


### 1.5 5) Xarxa social, fotografies (mydb05xsfotos)

Tenim una xarxa social. Volem emmagatzemar el nom, email i contrasenya de cada usuari. Cada usuari tindrà un id únic que l'identificarà.

Els usuaris poden ser amics entre ells això si, s'ha de informar de com s'han conegut.

Un usuari pot penjar varies fotografies que estaran fetes en un lloc (adreça) i s'identificaran amb un codi únic (codi del usuari i codi de la fotografia). No s'emmagatzemarà la fotografia en sí, sinó la URL d'on es pot trobar.



Instead the "@pPassword = password in clear text" I recommend to store the hash generated with a secure algorithm to encrypt it.

1) password\_hash = HASHBYTES('SHA2\_512', @pPassword)

2) It is better to add a unique salt per user. The salt is an internal key that is added to the password before calculating the hash.

The sample is valid for SQL Server. For MySQL instead of NEWID() it can be used UUID() to generate the salt.

When doing operations as INSERT or UPDATE the unique salt will be calculated and stored in the table. I show a code sample:

```
# MS SQL: DECLARE @salt UNIQUEIDENTIFIER=NEWID()
```

```
# MySQL version
```

```
DECLARE @salt UNIQUEIDENTIFIER=UUID()
```

```
HASHBYTES('SHA2_512', @pPassword+CAST(@salt AS NVARCHAR(36)))
```

But when authenticating a user it would be better to perform two queries:

1st: Get the password\_salt from the table and with the password from the user calculate the hash.

2nd: Get the password\_hash from the table and compare it against the calculated one.

The reason of these two operations is to avoid having both fields from the data base at the same time.