

INTRODUCTION AU SGBD

CONSOLE.LOG("BIENVENUE À TOUS");

LET'S BEGIN....

MODÈLE CONCEPTUEL DE DONNÉES (MCD)

UN MCD TE PERMET DE CONCEVOIR DE MANIÈRE VISUELLE TA BASE DE DONNÉES ET LES RELATIONS ENTRE LES TABLES.

LOGICIELS

MYSQL WORKBENCH
QUICKEN
dbeaver
ER DIAGRAM DESIGNER
PONYORM
WWW SQL DESIGNER

WORKBENCH

MYSQL WORKBENCH EST UN LOGICIEL DE GESTION ET D'ADMINISTRATION DE BASES DE DONNÉES MYSQL CRÉÉ EN 2004.

UNIFIED MODELING LANGUAGE

LE LANGAGE UML (UNIFIED MODELING LANGUAGE, OU LANGAGE DE MODÉLISATION UNIFIÉ) A ÉTÉ PENSÉ POUR ÊTRE UN LANGAGE DE MODÉLISATION VISUELLE COMMUN, ET RICHE SÉMANTIQUEMENT ET SYNTAXIQUEMENT. IL EST DESTINÉ À L'ARCHITECTURE, LA CONCEPTION ET LA MISE EN ŒUVRE DE SYSTÈMES LOGICIELS COMPLEXES PAR LEUR STRUCTURE AUSSI BIEN QUE LEUR COMPORTEMENT. L'UML A DES APPLICATIONS QUI VONT AU-DELÀ DU DÉVELOPPEMENT LOGICIEL, NOTAMMENT POUR LES FLUX DE PROCESSUS DANS L'INDUSTRIE.

UML

IL RESSEMBLE AUX PLANS UTILISÉS DANS D'AUTRES DOMAINES ET SE COMPOSE DE DIFFÉRENTS TYPES DE DIAGRAMMES. DANS L'ENSEMBLE, LES DIAGRAMMES UML DÉCRIVENT LA LIMITÉ, LA STRUCTURE ET LE COMPORTEMENT DU SYSTÈME ET DES OBJETS QUI S'Y TROUVENT.

LIEN ENTRE LES TABLES

IL EXISTE PLUSIEURS FAÇON DE LIER LES TABLES ENTRE ELLES:

1 ARTICLE LIÉ À UNE SEULE CATÉGORIE

1 ARTICLE LIÉ À PLUSIEURS CATÉGORIES

1 CATÉGORIE EST LIÉE À PLUSIEURS ARTICLES

CETTE RELATION SE FERA AU TRAVERS D'UNE CLÉ ÉTRANGÈRE APPELÉE **FOREIGN KEY**

CARDINALITÉ

Cardinalité	Symbole	Description
Un-Un	<1..1>	Une occurrence de la première entité peut correspondre à une et une seule occurrence de la seconde entité
Un-Plusieurs	<1..n>	Une occurrence de la première entité peut correspondre à plusieurs occurrences de la seconde entité
Plusieurs-Un	<n..1>	Plusieurs occurrences de la première entité peuvent correspondre à la même occurrence de la seconde entité
Plusieurs-Plusieurs	<n..n>	Plusieurs occurrences de la première entité peuvent correspondre à plusieurs occurrences de la seconde entité

PROPRIÉTÉS DES COLUMNS

PK – PRIMARY KEY

NN – NOT NULL

BIN – BINARY

UN – UNSIGNED (STOCK SEULEMENT DES VALEURS POSITIVES)

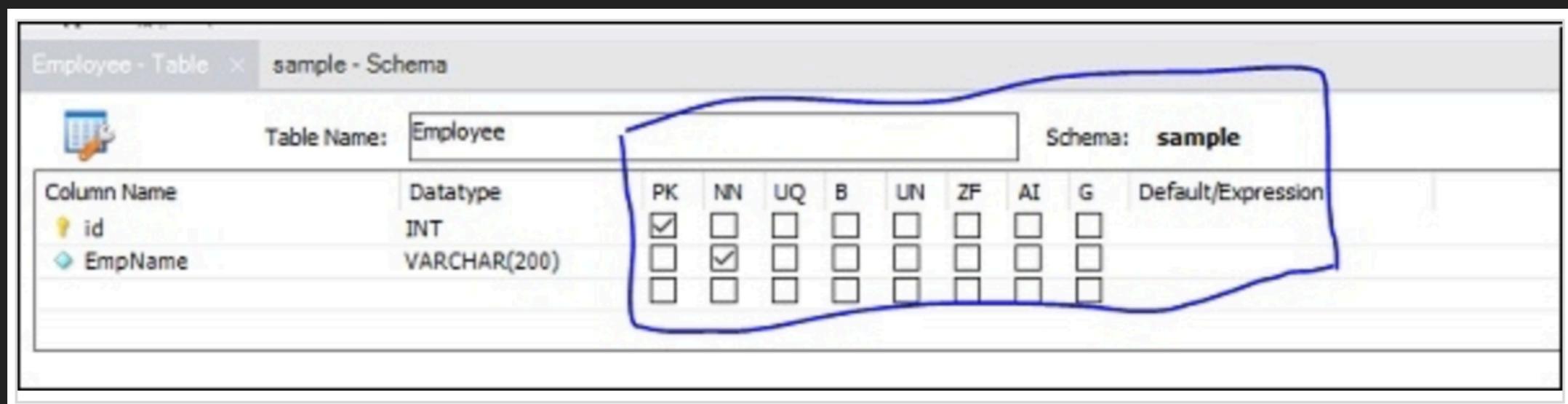
UQ – UNIQUE

ZF – ZERO FILLED

G – GENERATE COLUMN

AI – AUTO INCREMENT

PROPRIÉTÉS



TYPE DE DONNÉES

bigint	Grand numéro	Consultez Utilisation du type de données Grand nombre.
binaire (taille de fichier)	Binaire	Identique à la taille de champ SQL Server
bit	Oui/Non	
caractère (taille du champ), où la taille est inférieure ou égale à 255	Texte	Identique à la taille de champ SQL Server
caractère (taille du champ), dont la taille est supérieure à 255	Mémo	
datetime	Date/Heure	
décimale (précision, échelle)	Nombre	Décimale (les propriétés Précision et Échelle d'Access correspondent à la précision et l'échelle de SQL Server).

TYPE DE DONNÉES

float	Nombre	Double
image	Objet OLE	
Int	Nombre	Entier long
money	Devise	
nchar (taille du champ), où la taille est inférieure ou égale à 255	Texte	Identique à la taille de champ SQL Server
nchar (taille du champ), dont la taille est supérieure à 255	Mémo	
ntext	Mémo	

TYPE DE DONNÉES

numérique (précision, échelle)	Nombre	Décimale (les propriétés Précision et Échelle d'Access correspondent à la précision et l'échelle de SQL Server).
nvarchar (taille du champ), où la taille est inférieure ou égale à 255	Texte	Identique à la taille de champ SQL Server
nvarchar (taille du champ), dont la taille est supérieure à 255	Mémo	
nvarchar(MAX)	Mémo	
real	Nombre	Unique
smalldatetime	Date/Heure	

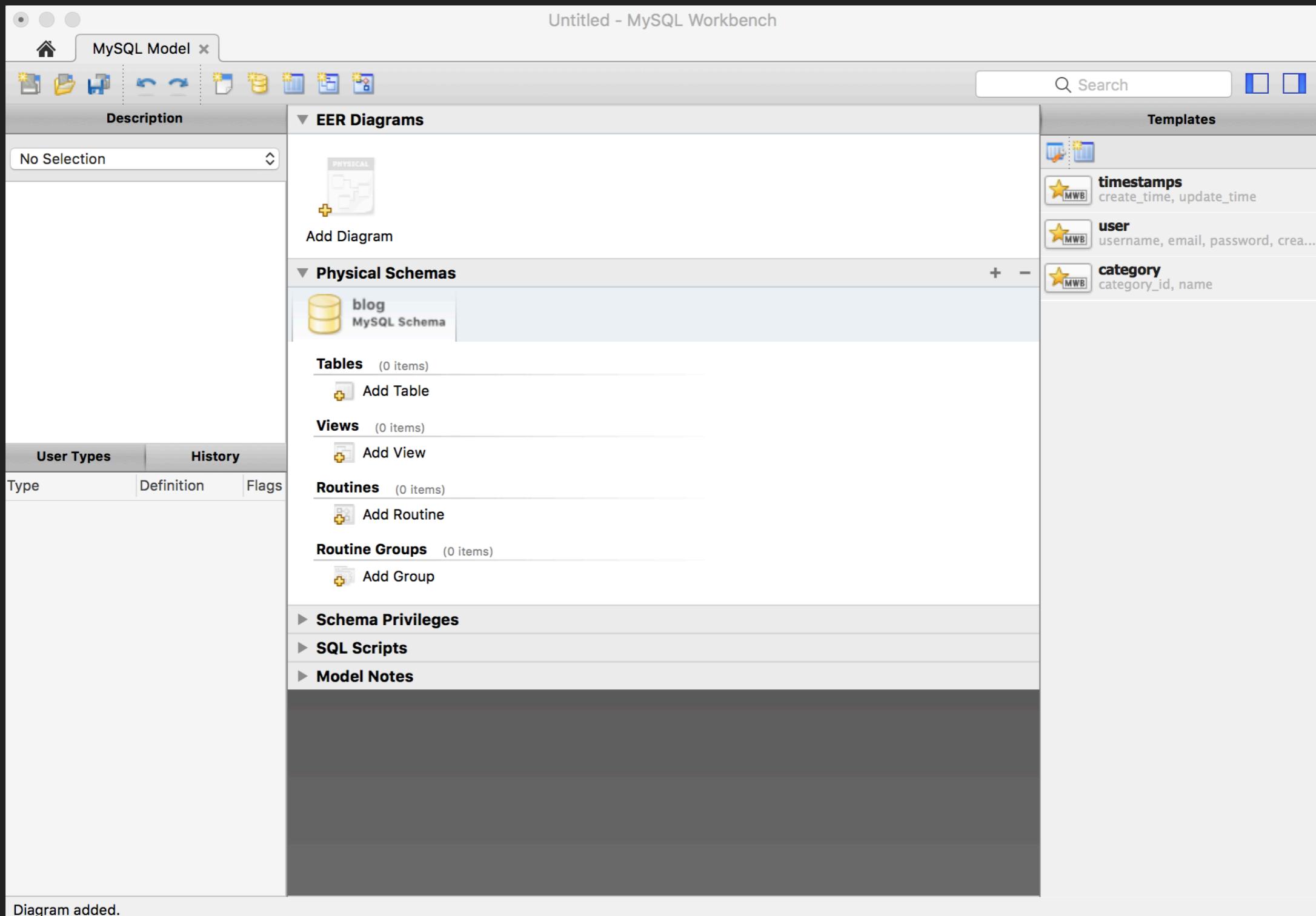
TYPE DE DONNÉES

smallint	Nombre	Entier
smallmoney	Devise	
sql_variant	Texte	255
text	Mémo	
timestamp	Binaire	8
tinyint	Nombre	Byte
uniqueidentifier	Nombre	N° de réplication

TYPE DE DONNÉES

varbinary	Binaire	Identique à la taille de champ SQL Server
varbinary (MAX)	Objet OLE	
varchar (taille du champ), où la taille est inférieure ou égale à 255	Texte	Identique à la taille du champ SQL Server
varchar (taille du champ), dont la taille est supérieure à 255	Mémo	
varchar(MAX)	Mémo	
xml	Mémo	

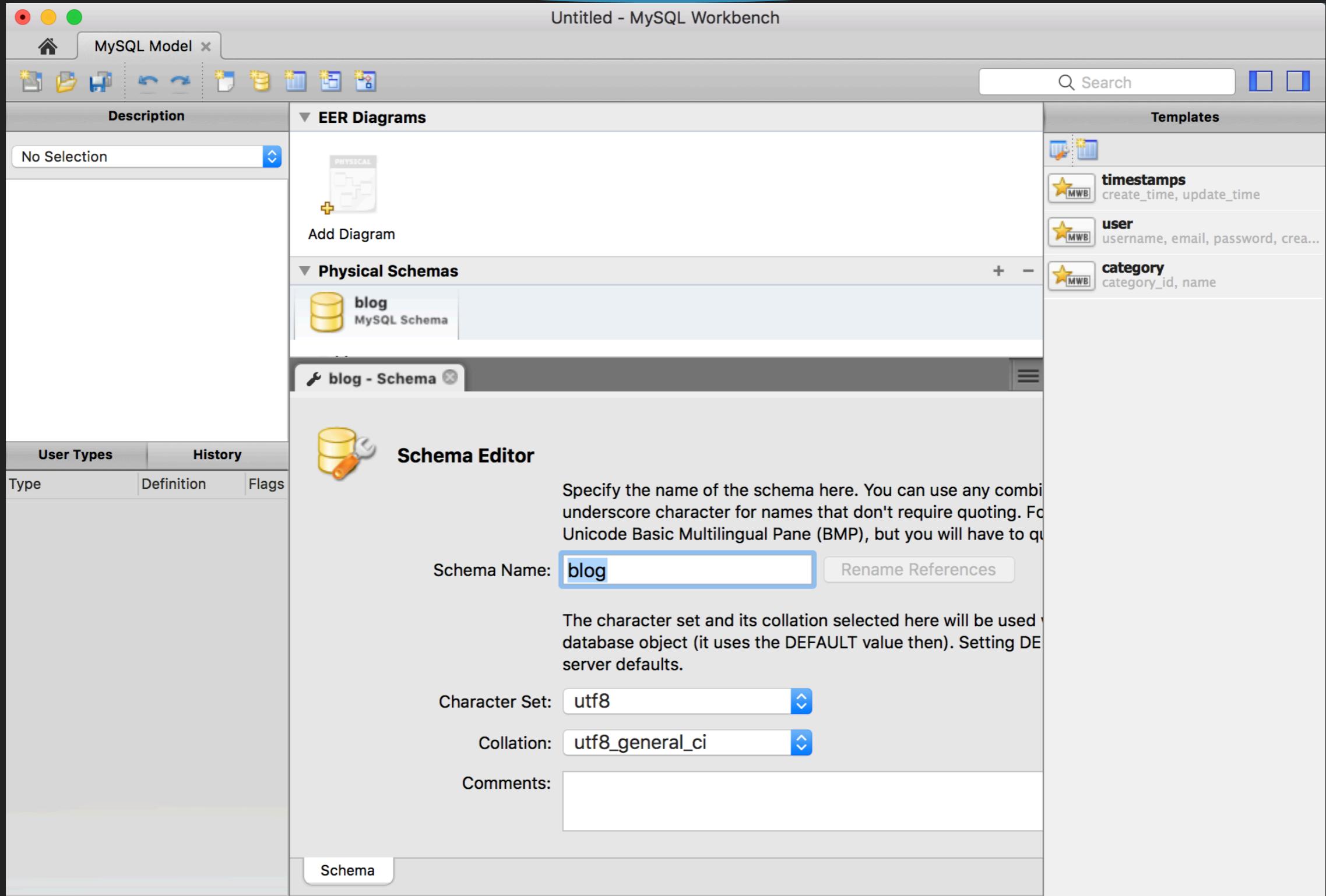
MODÉLISONS !



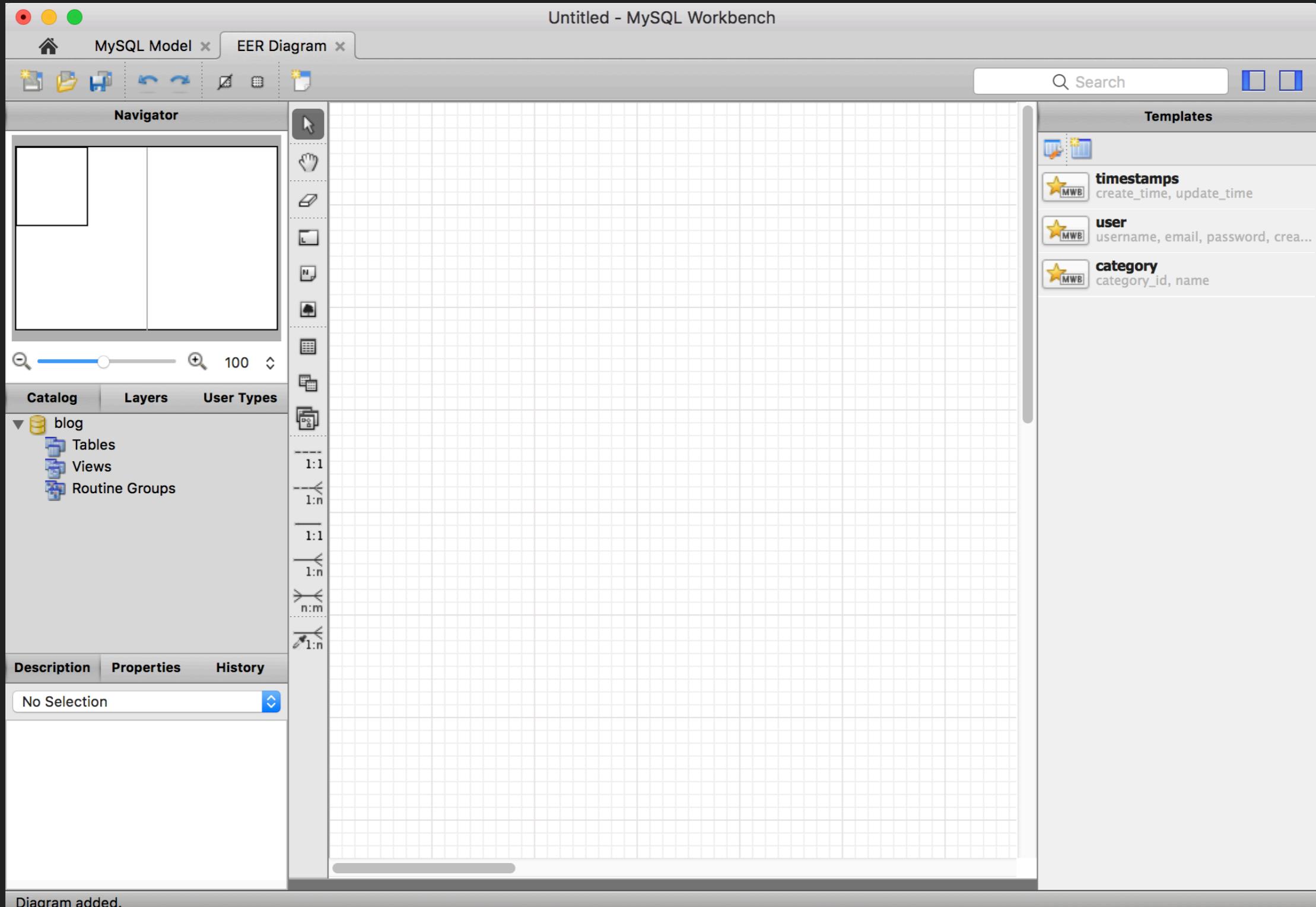
MODÉLISONS !

CRÉER NOTRE SCHÉMA

MODÉLISONS !



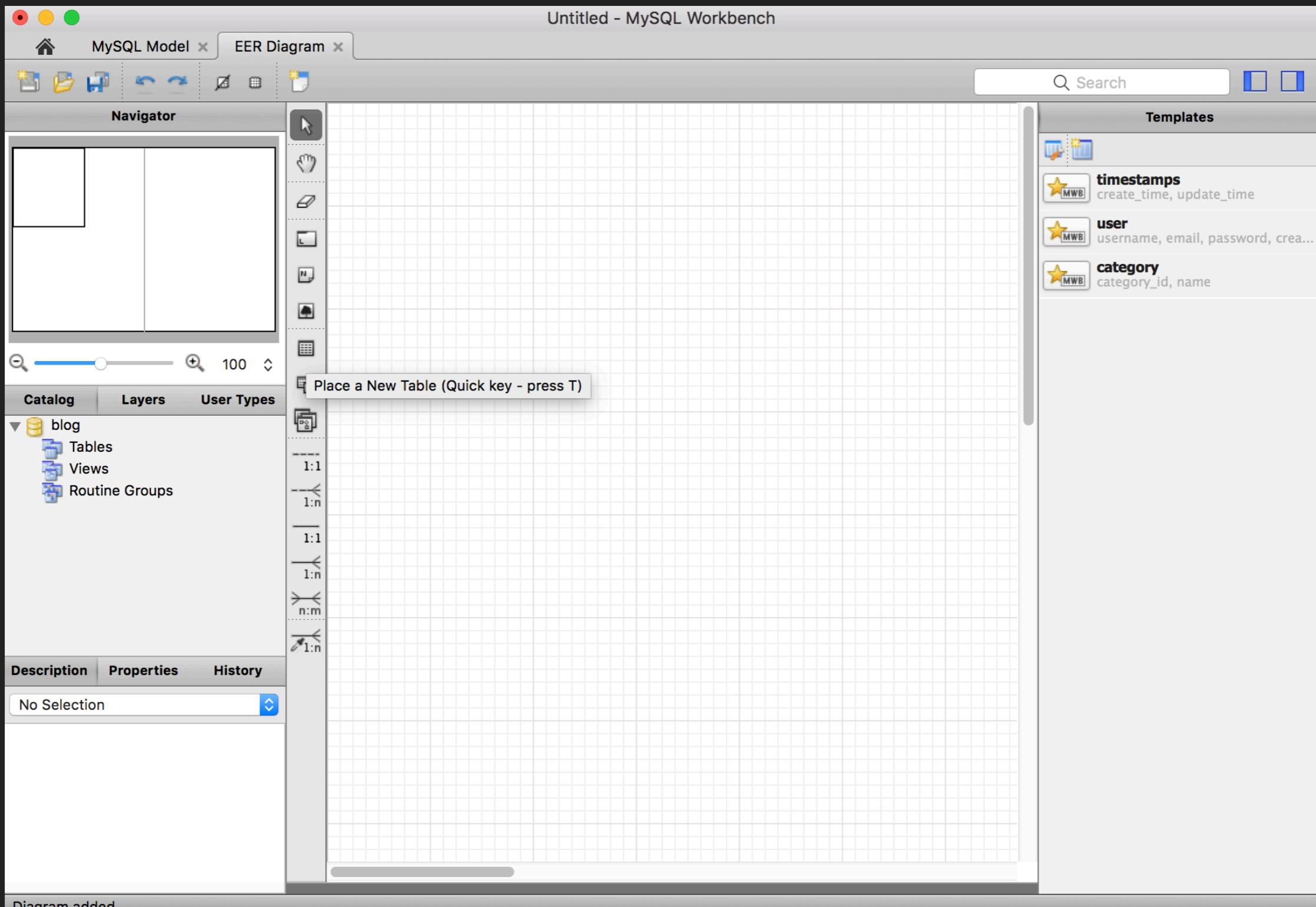
MODÉLISONS !



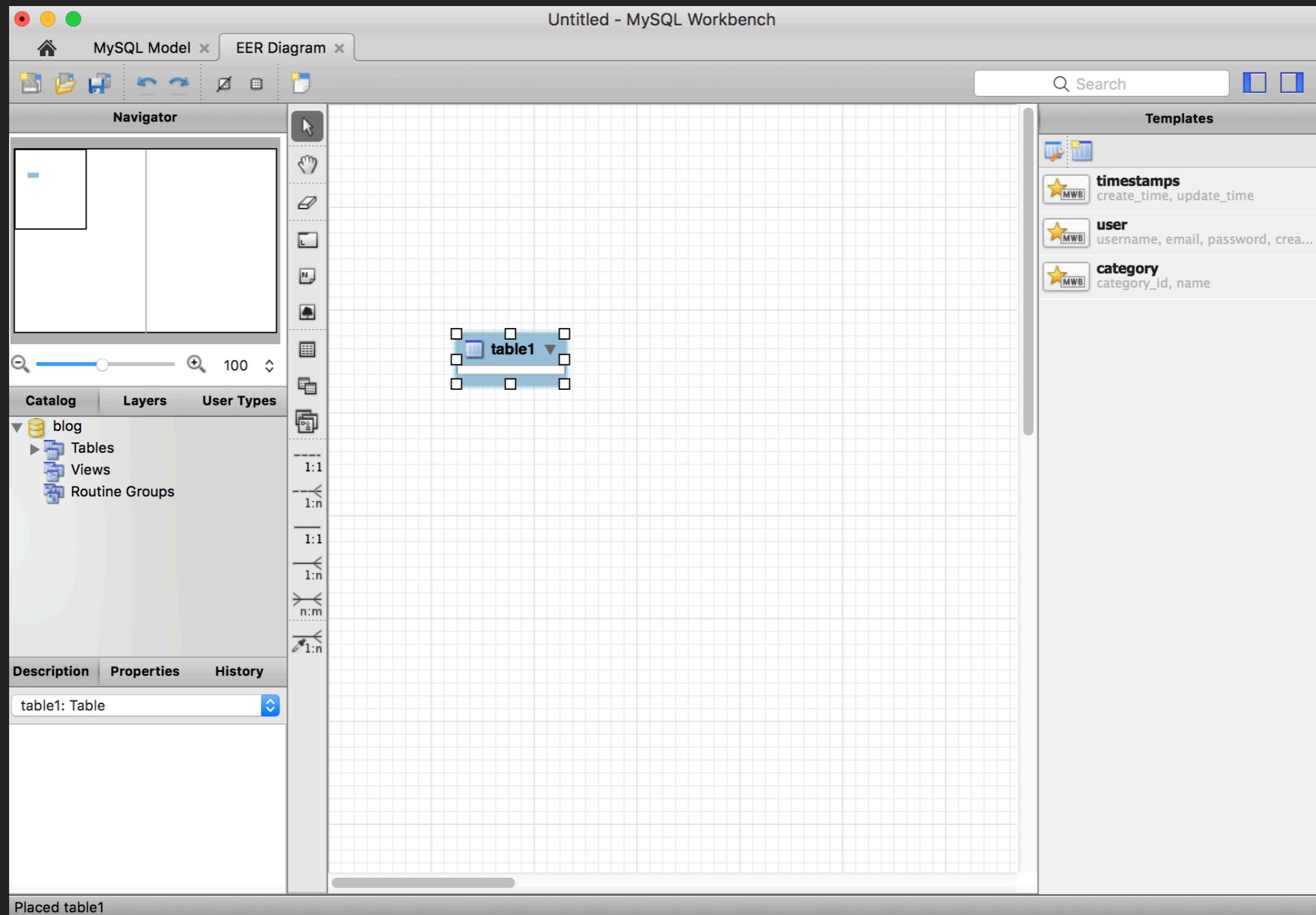
MODÉLISONS !

CRÉER NOTRE PREMIÈRE TABLE

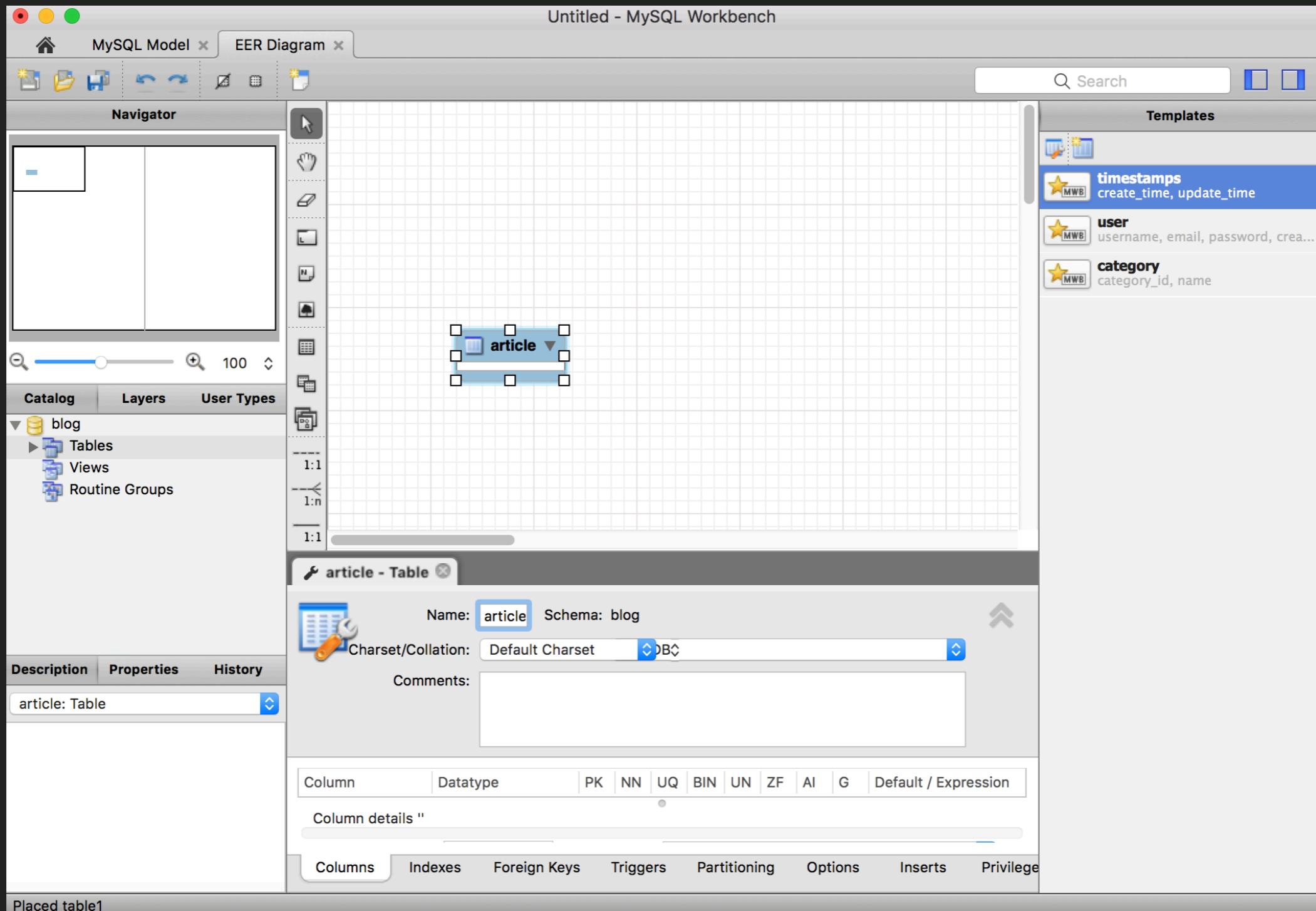
MODÉLISONS !



MODÉLISONS !



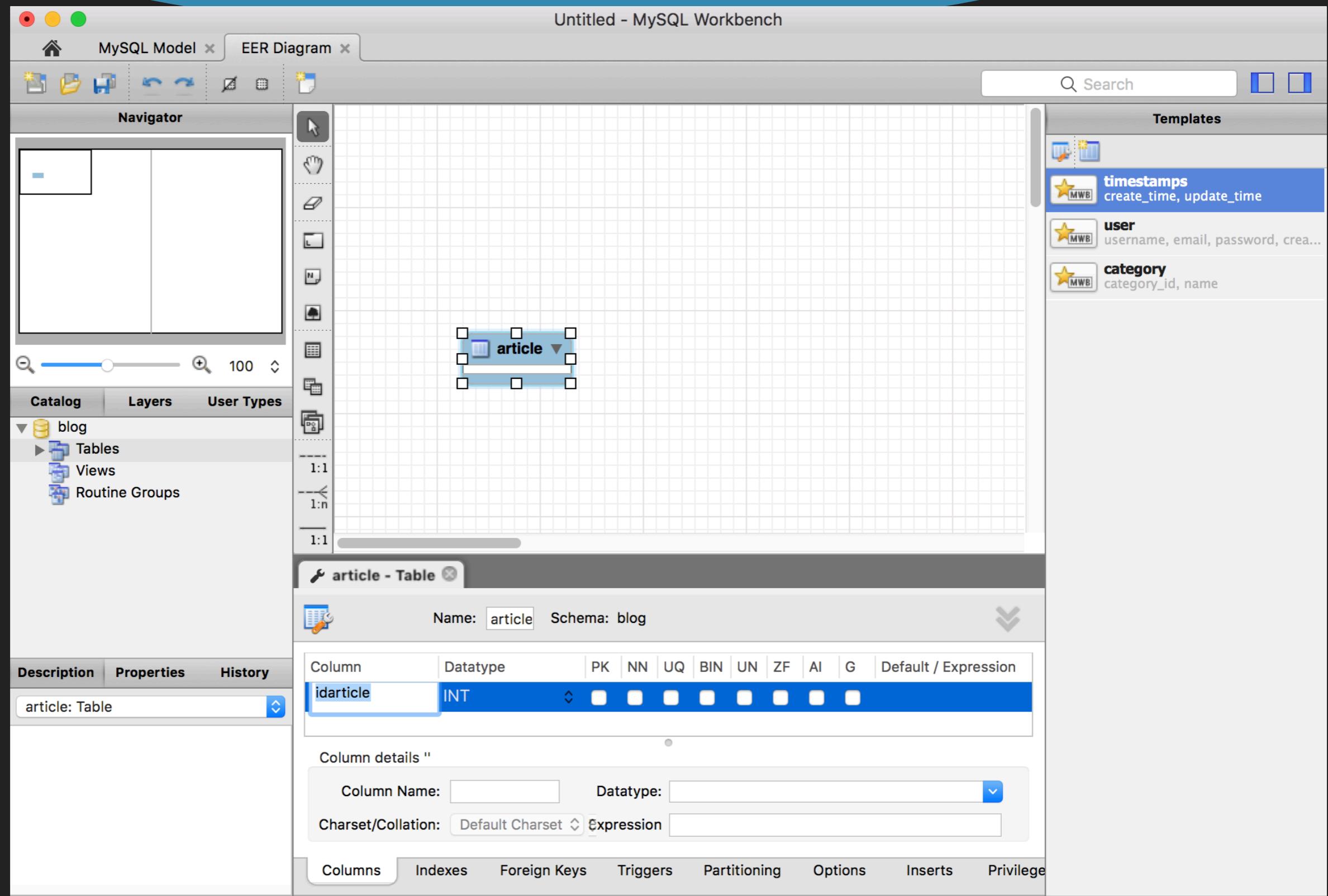
MODÉLISONS !



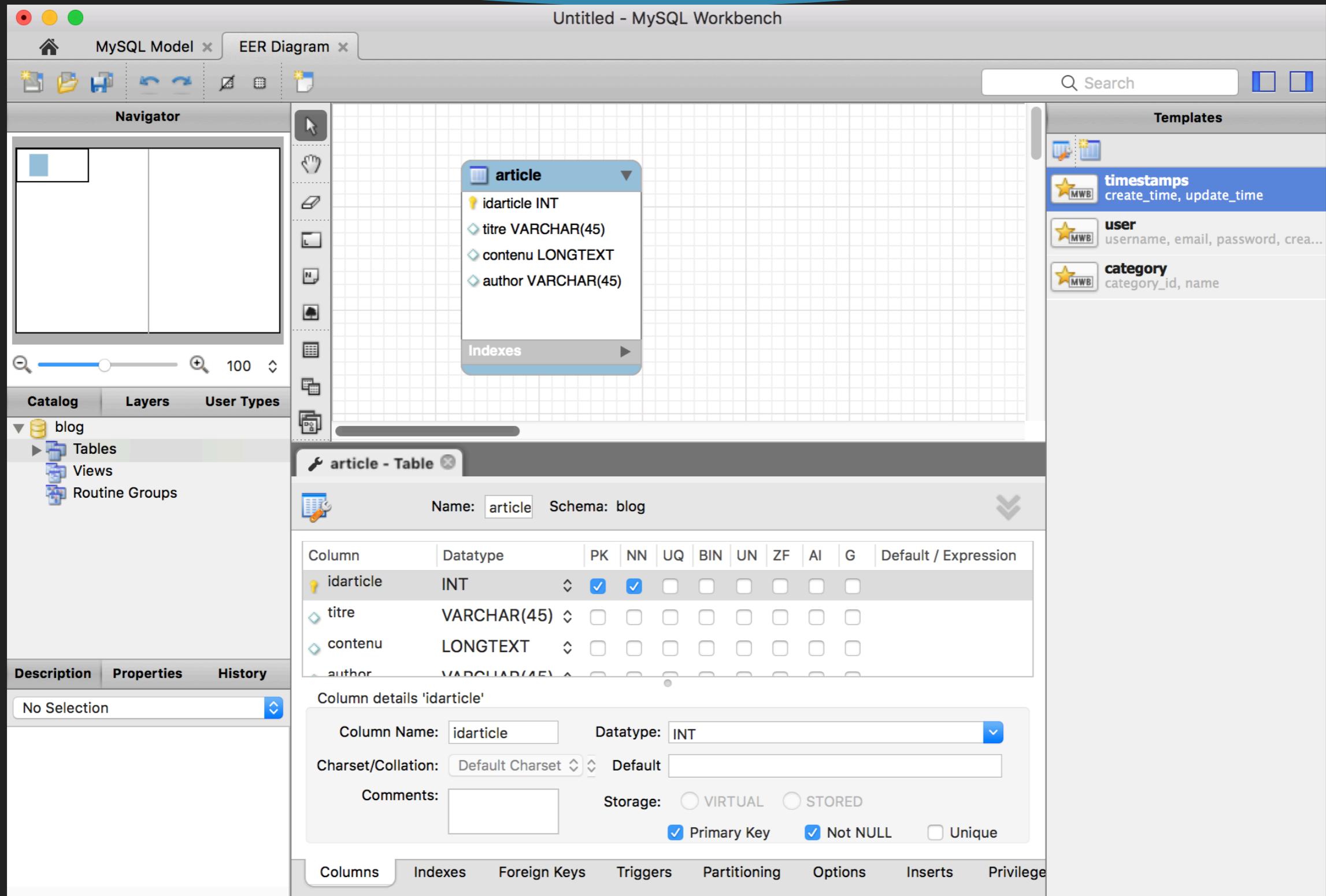
MODÉLISONS !

ALIMENTER NOTRE TABLE

MODÉLISONS !



MODÉLISONS !



MODÉLISONS !

Untitled - MySQL Workbench

Create a UNIQUE index for the column

Navigator

article

- idarticle INT
- title VARCHAR(45)
- content LONGTEXT
- author VARCHAR(45)

Indexes

categories

- idcategories INT
- name VARCHAR(70)

Indexes

Templates

- timestamps** create_time, update_time
- user** username, email, password, crea...
- category** category_id, name

Catalog Layers User Types

blog

Tables

Views

Routine Groups

article - Table

Name: article Schema: blog

Column Datatype PK NN UQ BIN UN ZF AI G Default / Expression

content	LONGTEXT	<input type="checkbox"/>									
author	VARCHAR(45)	<input type="checkbox"/>									

<click to edit>

Column details 'contenu'

Column Name: contenu Datatype: LONGTEXT

Charset/Collation: Default Charset Default

Comments: Storage: VIRTUAL STORED

Primary Key Not NULL Unique

Columns Indexes Foreign Keys Triggers Partitioning Options Inserts Privilege

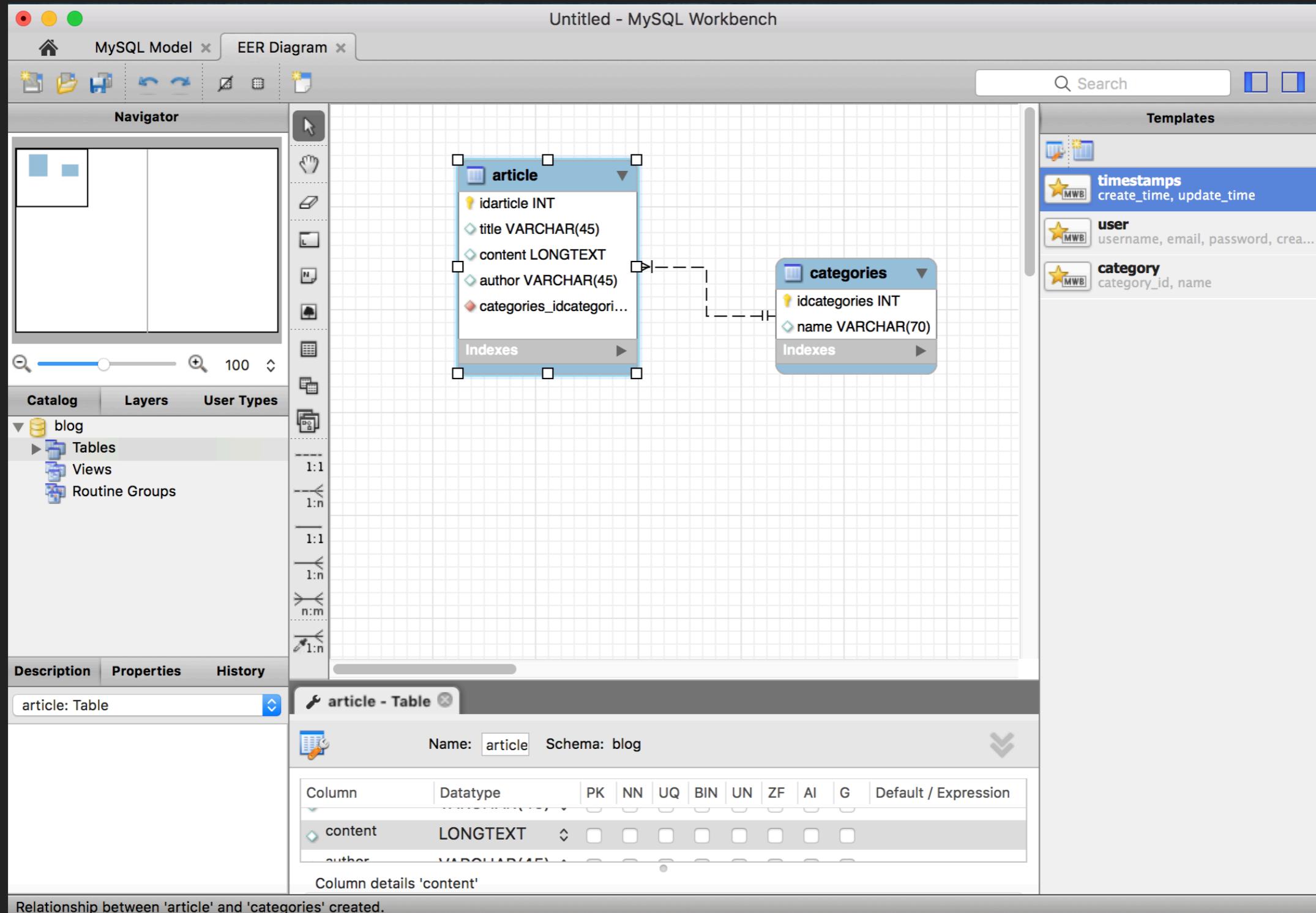
Placed table1

MODÉLISONS !

CRÉER DES LIAISONS ENTRE LES TABLES

MODÉLISONS !

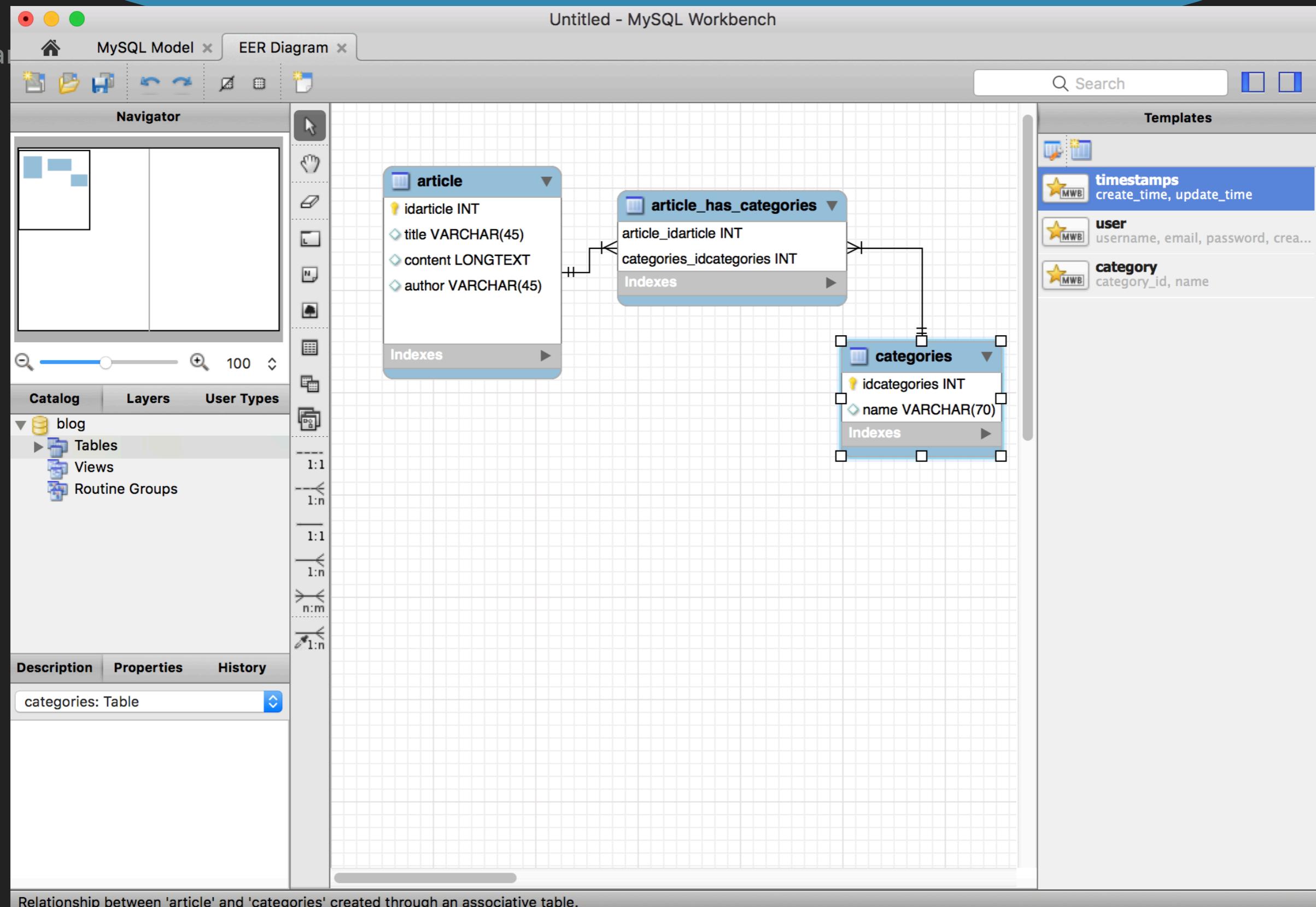
ici article a une catégorie



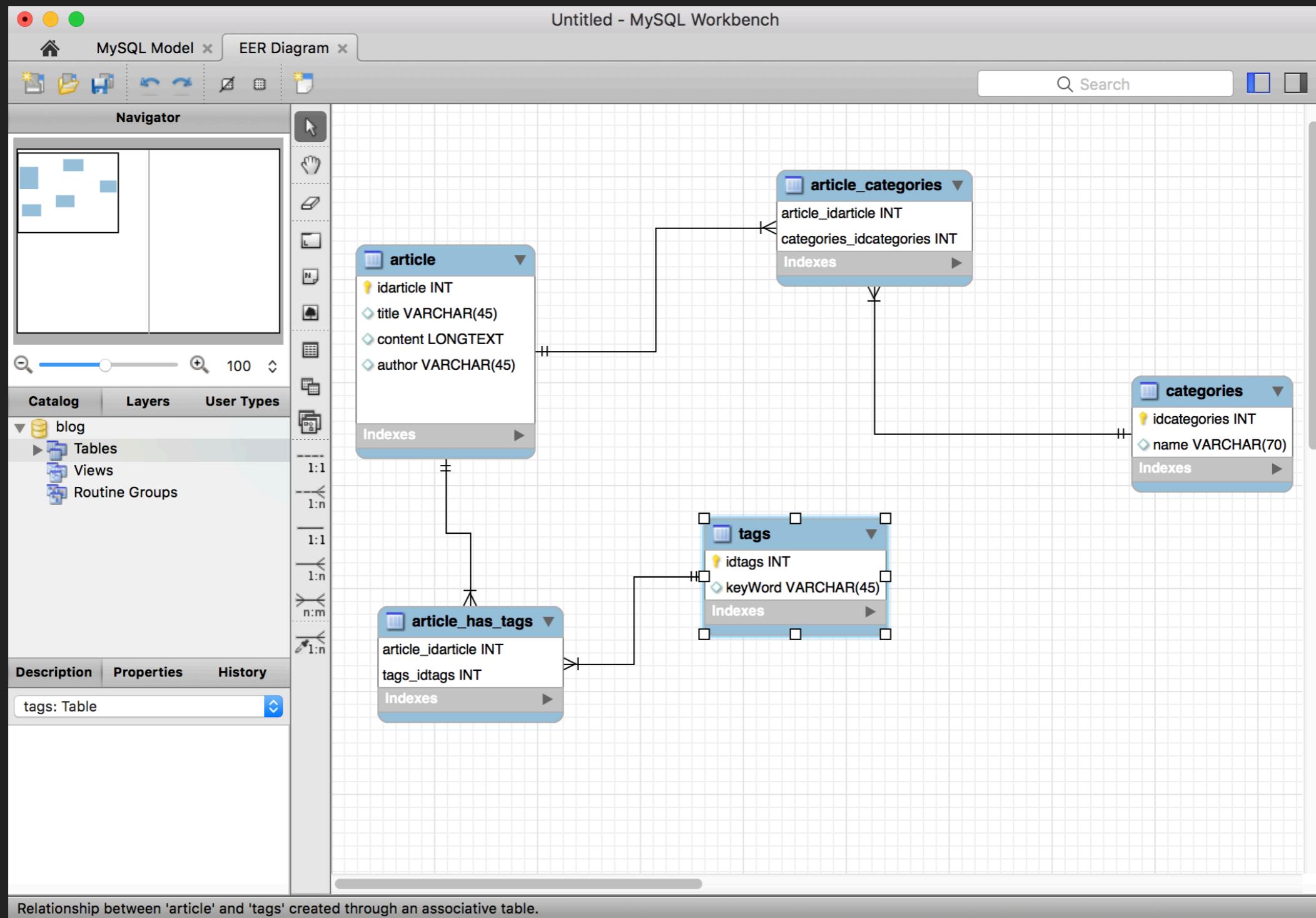
LIAISON ENTRE LES TABLES

SI JE VEUX QU'UN ARTICLE SOIT LIÉ À PLUSIEURS CATÉGORIES ON VA FAIRE UNE TABLE DE LIAISON

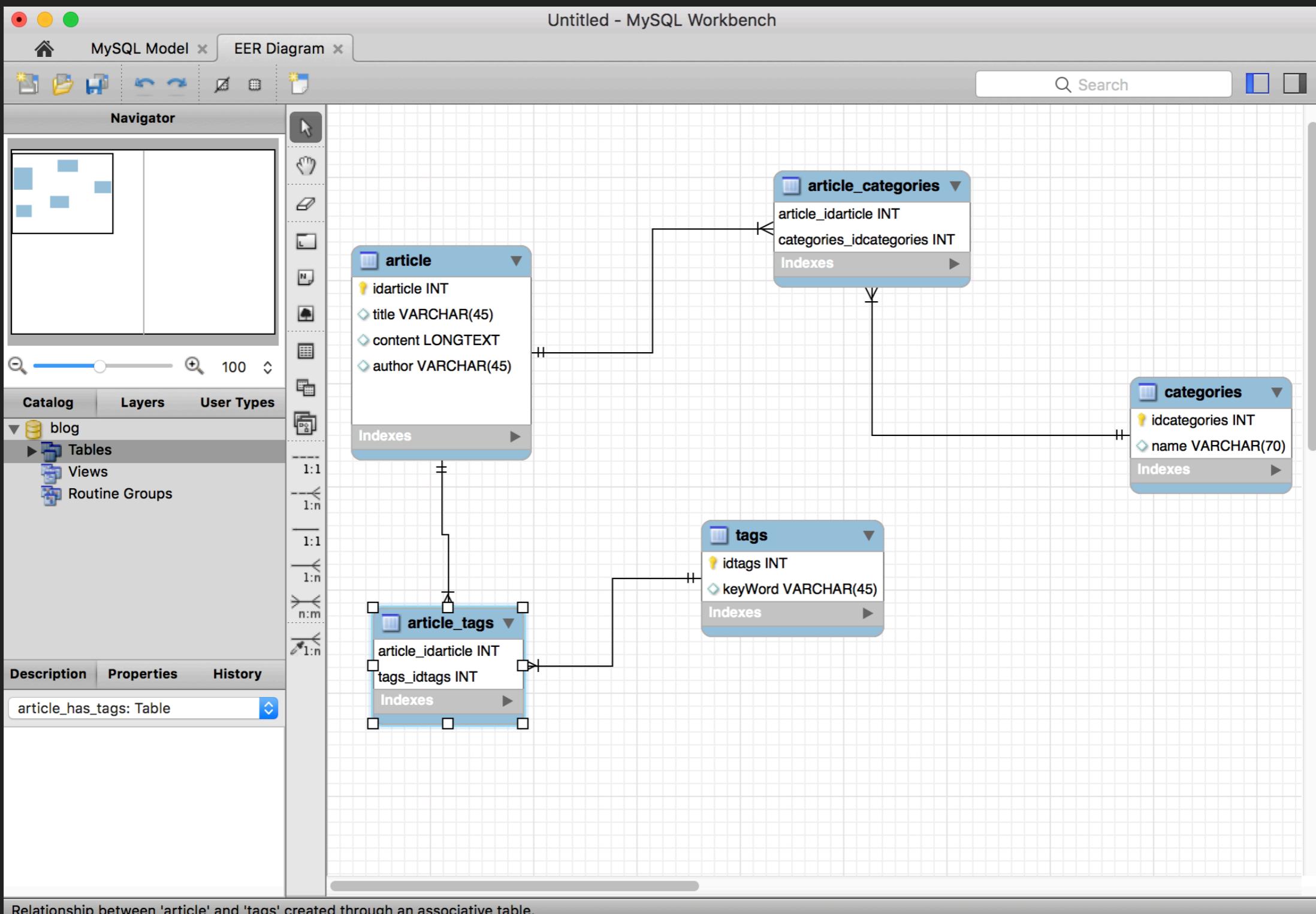
PURPOSE



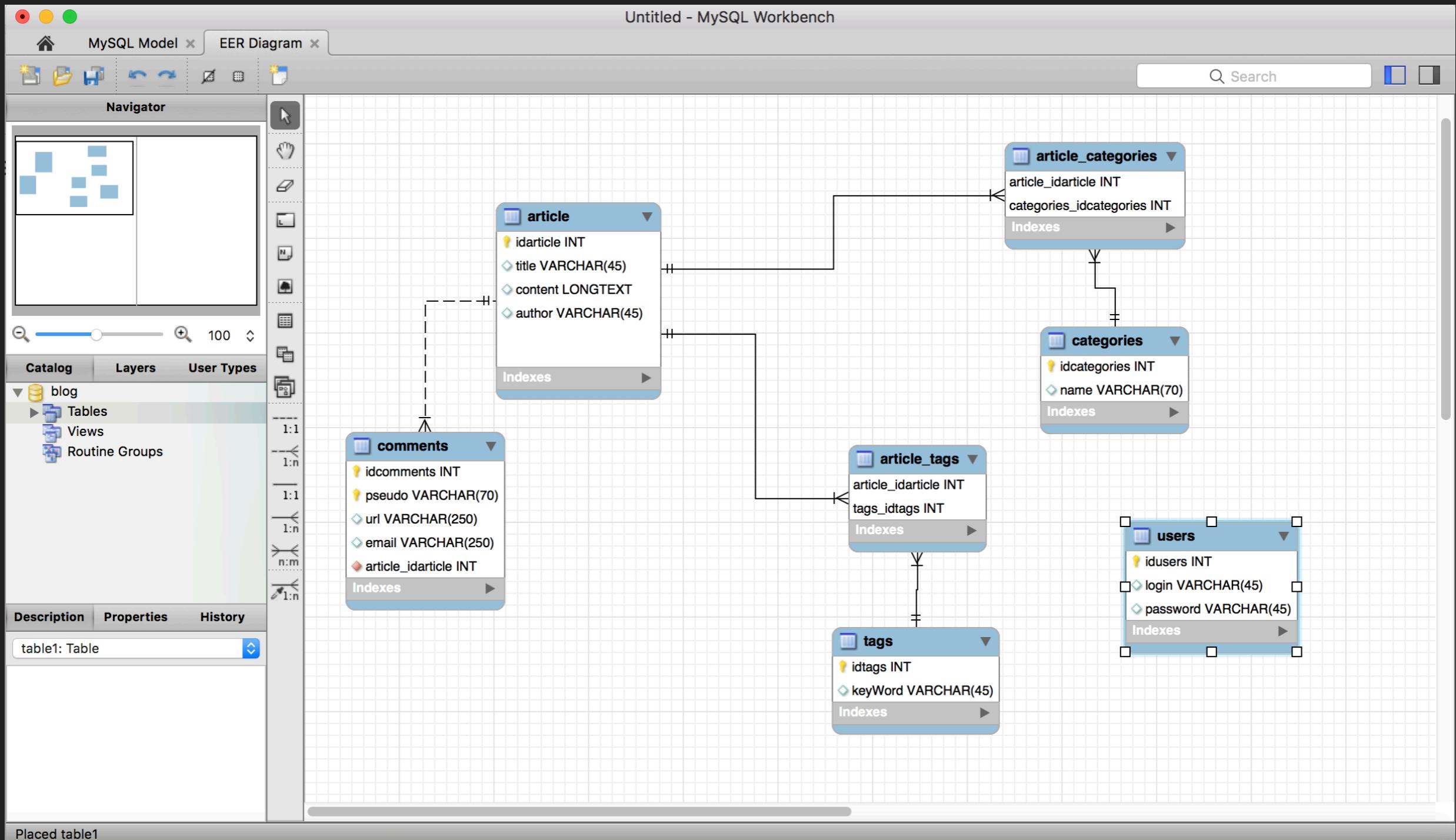
PURPOSE



PURPOSE



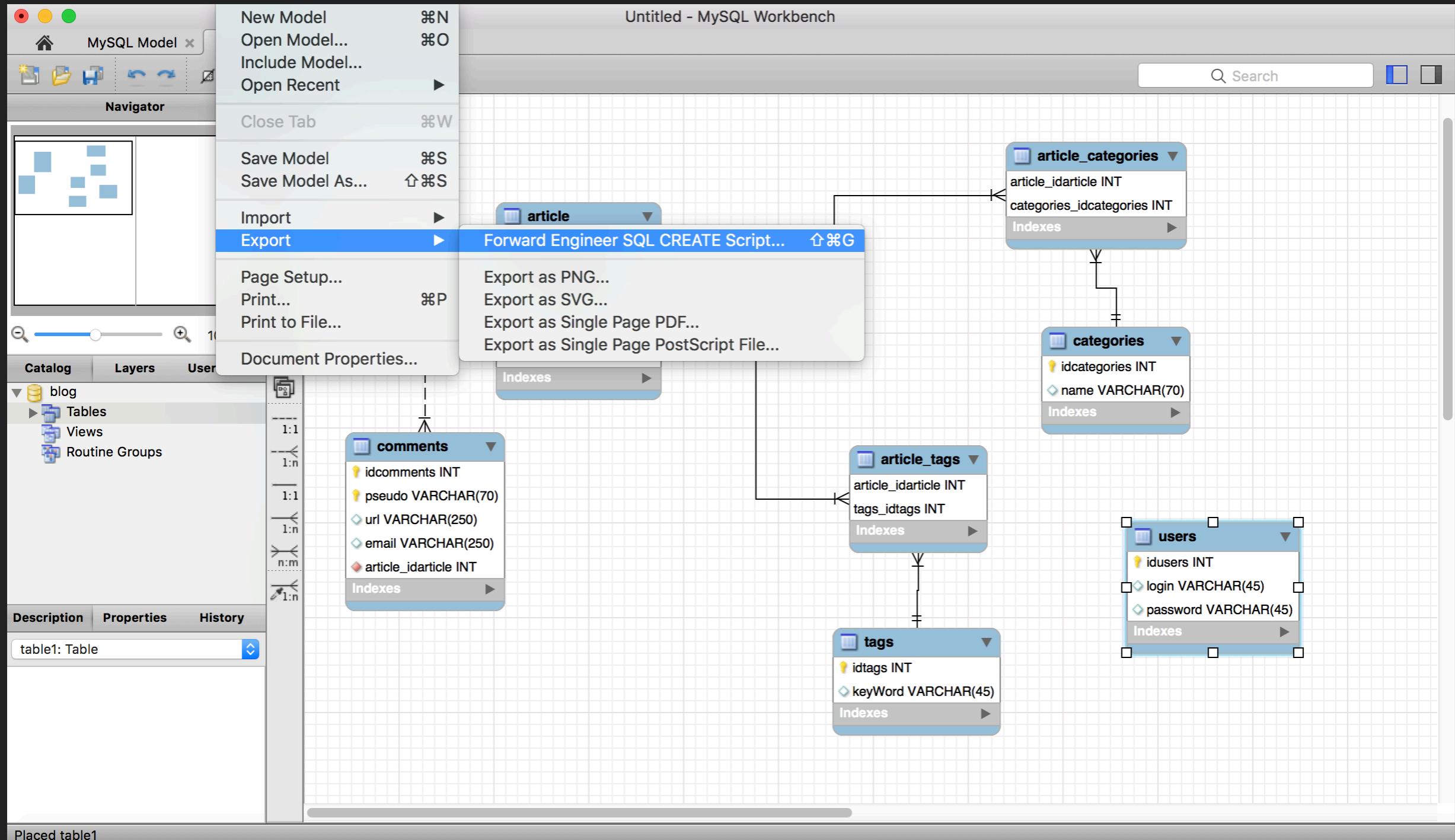
PURPOSE



PURPOSE

EXPORTER LE CODE SQL

PURPOSE



PURPOSE

Forward Engineer SQL Script

SQL Export Options

SQL Export Options

Filter Objects

Review SQL Script

Output SQL Script File: ...

Leave blank to view generated script but not save to a file.

SQL Options

Generate DROP Statements Before Each CREATE Statement

Generate DROP SCHEMA

Sort Tables Alphabetically ⓘ

Skip Creation of FOREIGN KEYS

Skip creation of FK Indexes as well

Omit Schema Qualifier in Object Names

Generate USE statements

Generate Separate CREATE INDEX Statements

Add SHOW WARNINGS After Every DDL Statement

Do Not Create Users. Only Export Privileges

Don't create view placeholder tables.

Generate INSERT Statements for Tables

Disable FK checks for inserts

Create triggers after inserts

Go Back Continue

PURPOSE

Forward Engineer SQL Script

SQL Object Export Filter

To exclude objects of a specific type from the SQL Export, disable the corresponding checkbox. Press Show Filter and add objects or patterns to the ignore list to exclude them from the export.

- Export MySQL Table Objects
7 Total Objects, 7 Selected Show Filter
- Export MySQL View Objects
0 Total Objects, 0 Selected Show Filter
- Export MySQL Routine Objects
0 Total Objects, 0 Selected Show Filter
- Export MySQL Trigger Objects
0 Total Objects, 0 Selected Show Filter
- Export User Objects
0 Total Objects, 0 Selected Show Filter

Go Back Continue

PURPOSE

Forward Engineer SQL Script

Review Generated Script

Review the generated script.

```
1 -- MySQL Script generated by MySQL Workbench
2 -- Sun Jan 26 17:02:20 2020
3 -- Model: New Model Version: 1.0
4 -- MySQL Workbench Forward Engineering
5
6 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
7 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
8 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DA
9
10
11 -- Schema blog
12
13
14 -- Schema blog
15
16
17 CREATE SCHEMA IF NOT EXISTS `blog` DEFAULT CHARACTER SET utf8 ;
18 USE `blog` ;
19
20
21 -- Table `blog`.`article`
22
23 ┌─ CREATE TABLE IF NOT EXISTS `blog`.`article` (
24   `idarticle` INT NOT NULL,
25   `title` VARCHAR(45) NULL,
26   `content` LONGTEXT NULL,
27   `author` VARCHAR(45) NULL,
28   PRIMARY KEY (`idarticle`)
29   ENGINE = InnoDB;
30
31
32
33 -- Table `blog`.`categories`
34
```

100% 1:1

Save to Other File... Copy to Clipboard

Go Back Finish

PURPOSE

GÉNÉRER NOS TABLES DANS PHP MYADMIN

PURPOSE

phpMyAdmin

Server: localhost:8889 » Database: shauto » Table: factures_vente

Browse Structure SQL Search Insert Export Import Privileges Operations Triggers

Recent Favorites

New entreprise information_schema mysql performance_schema shauto New cars client factures_vente sys

Run SQL query/queries on table shauto.factures_vente:

```
109 PRIMARY KEY (`idcomments`, `pseudo`),
110 CONSTRAINT `fk_comments_article1`
111   FOREIGN KEY (`article_idarticle`)
112     REFERENCES `blog`.`article` (`idarticle`)
113     ON DELETE NO ACTION
114     ON UPDATE NO ACTION)
115 ENGINE = InnoDB;
116
117 CREATE INDEX `fk_comments_article1_idx` ON `blog`.`comments` (`article_idarticle` ASC) VISIBLE;
118
119
120 -- Table `blog`.`users`
121 --
122
123 CREATE TABLE IF NOT EXISTS `blog`.`users` (
124   `idusers` INT NOT NULL,
125   `login` VARCHAR(45) NULL,
126   `password` VARCHAR(45) NULL,
127   PRIMARY KEY (`idusers`))
128 ENGINE = InnoDB;
129
130
131 SET SQL_MODE=@OLD_SQL_MODE;
132 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
133 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

Columns

id_invoice
firstName_client
name_client
address_client
postalCode_client

SELECT * SELECT INSERT UPDATE DELETE Clear Format Get auto-saved query

Bind parameters

[Delimiter ;] Show this query here again Retain query box Rollback when finished Enable foreign key checks Go

Console