

## Projet Birdy

### Tables

#### Tables SQLite

```
const createUserTable = `CREATE TABLE IF NOT EXISTS "users" (  
  "id" INTEGER,  
  "login" VARCHAR(50) NOT NULL,  
  "password" VARCHAR(50) NOT NULL,  
  "lastname" VARCHAR(50) NOT NULL,  
  "firstname" VARCHAR(50) NOT NULL,  
  PRIMARY KEY("id" AUTOINCREMENT)  
)`;  
db.exec(createUserTable, function(err){  
  if(err){  
    throw err;  
  }  
  console.log('User table ready');  
})
```

Tabella: users Filtro in qualsiasi colonna

	id	login	password	lastname	firstname
	Filtro	Filtro	Filtro	Filtro	Filtro
1	1	sdemanteau	B71A3	de Manteau	Sébastien
2	2	benelapens	blueInk	Lapensée	Bénédicte
3	3	estbon2001	indices	Bonisac	Esther
4	4	ismatuil	nuagesBlanches	Tuil	Ismael
5	5	maitreadele	jesuislareine	Richard	Adèle
6	6	amelicor	champagne	Corcos	Amélie
7	7	alexlamont	été86	Lamont	Alexandre
8	8	cesarine21	lavieenrose	Vergnaud	Césarine
9	9	beniciosetubal	boanoite	Cupertino	Diogo

```

const createNotificationsTable = `CREATE TABLE IF NOT EXISTS "notifications" (
  "id" INTEGER,
  "userId" INTEGER NOT NULL,
  "text" VARCHAR(50) NOT NULL,
  "date" TEXT,
  PRIMARY KEY("id" AUTOINCREMENT)
)`;
db.exec(createNotificationsTable, function(err){
  if(err){
    throw err;
  }
  console.log('Notifications table ready');
})
}

```

Tabella: notifications

	id	userId	text	date
	Filtro	Filtro	Filtro	Filtro
1	2	3	Nouveau rdvLecture conseillé de la part de ...	2021-05-02 22:36:38
2	3	6	Nouveau rdvLecture conseillé de la part de ...	2021-05-02 22:37:24
3	4	6	Nouveau rdvLecture conseillé de la part de ...	2021-05-02 22:38:09
4	5	3	Nouveau rdvLecture conseillé de la part de ...	2021-05-02 22:38:25
5	6	3	Lecture conseillée : "Exercices de style"	2021-05-02 22:39:09
6	7	2	benelapens te propose une demande d'amitié	2021-05-02 22:42:13
7	8	3	ismatuil te propose une demande d'amitié	2021-05-02 22:42:48

```

const createFriendsTable = `CREATE TABLE IF NOT EXISTS "friends" (
  "id1" INTEGER NOT NULL,
  "id2" INTEGER NOT NULL,
  "demanding" INTEGER NOT NULL,
  "accepting" INTEGER NOT NULL,
  PRIMARY KEY("id1" , "id2")
)`;
db.exec(createFriendsTable, function(err){
  if(err){
    throw err;
  }
  console.log('Friends table ready');
})

```

Tabella: friends

	id1	id2	demanding	accepting
	Filtro	Filtro	Filtro	Filtro
1	9	2	1	0
2	7	2	1	1
3	8	5	1	0
4	3	1	1	0
5	3	9	1	0

id1 (user) **demande** l'amitié à id2 (user) :  
demanding = 1 ; accepting = 0.  
**acceptation de id2** : demanding = accepting = 1  
**rejet de id2** : élimination ligne

Tabella: followers

	id1	id2	entity
	Filtro	Filtro	Filtro
1	3	1	authors
2	3	8	books
3	1	4	books
4	1	5	books

id1(user) **follows** id2 (entity : livre ou auteur ou rdvLecture)

```
const createFollowersTable = `CREATE TABLE IF NOT EXISTS "followers" (
  "id1" INTEGER NOT NULL,
  "id2" INTEGER NOT NULL,
  "entity" VARCHAR(50) NOT NULL,
  PRIMARY KEY ("id1" , "id2", "entity")
  FOREIGN KEY ("id1") REFERENCES "users" ("id") ON DELETE CASCADE
);`;
db.exec(createFollowersTable, function(err){
  if(err){
    throw err;
  }
  console.log('Followers table ready');
})
```

```
const createAuthorsTable = `CREATE TABLE IF NOT EXISTS "authors" (
  "id" INTEGER,
  "lastname" VARCHAR(50),
  "firstname" VARCHAR(50),
  "alias" VARCHAR(50),
  "biography" VARCHAR(1000),
  "image" VARCHAR(50),
  PRIMARY KEY("id" AUTOINCREMENT)
);`;
db.exec(createAuthorsTable, function(err){
  if(err){
    throw err;
  }
  console.log('Authors table ready');
})
```

Tabella: authors

	id	lastname	firstname	alias	biography	image
	Filtro	Filtro	Filtro	Filtro	Filtro	Filtro
1	1	Flaubert	Gustave	NULL	Madame Bovary c est moi	flaubert.jpeg
2	2	Baudelaire	Charles	NULL	Luxe, calme et volupté	baudelaire.jpeg
3	3	Rimbaud	Arthur	NULL	La jambe me fait mal	rimbaud.jpeg
4	4	Hugo	Victor	NULL	Misérable à place des Vosges	hugo.jpeg
5	5	NULL	NULL	Colette	Je suis devenue un magasin	colette.jpeg

```
const createBooksTable = `CREATE TABLE IF NOT EXISTS "books" (
  "id" INTEGER,
  "id_author" INTEGER NOT NULL,
  "title" VARCHAR(50) NOT NULL,
  "image" VARCHAR(50),
  PRIMARY KEY("id" AUTOINCREMENT)
)`;
db.exec(createBooksTable, function(err){
  if(err){
    throw err;
  }
  console.log('Books table ready');
})
```

Tabella: books

	id	id_author	title	image
	Filtro	Filtro	Filtro	Filtro
1	1	1	Madame Bovary	emma.jpeg
2	2	3	Le bateau ivre	bateau.jpeg
3	3	4	Les misérables	cosette.jpeg
4	4	4	Notre-Dame de Paris	quasimodo.jpeg
5	5	4	La Légende des siècles	legende.jpeg
6	6	5	La maison de Claudine	claudine.jpeg
7	7	6	Adieux aux armes	catherine.jpeg
8	8	6	Le vieil homme et la mer	lions.jpeg

## Tables MongoDB

```
plumes
```

FILTER {"filter": "example"}

QUERY RESULTS 1-2 OF 2

```
_id: ObjectId("608ee5bbcebc7225943031c0")  
✓ comments: Array  
  ✓ 0: Object  
    userId: "1"  
    text: "Car tu dois etudier un peu plus"  
    spoiler: false  
spoiler: false  
userId: "5"  
typeText: "plume"  
text: "J ai lu Les misérables et je n ai rien compris"  
image: ""  
date: 2021-05-02T17:47:39.142+00:00  
entity_id: "3"  
typeEntity: "books"  
__v: 0
```

```
_id: ObjectId("608ee699cebc7225943031c1")  
> comments: Array  
spoiler: false  
userId: "9"  
typeText: "plume"  
text: "Moi aussi je veux rever des lions"  
image: ""  
date: 2021-05-02T17:51:21.407+00:00  
entity_id: "8"  
typeEntity: "books"  
__v: 0
```

```

const rdvLecturesSchema = db.Schema({
  // id genere automatiquement par mongoose
  speaker : {type : String, required : true}, //nom, prenom ou alias
  title : {type : String, required : true},
  text : {type : String, default : "Rendez-vous lecture en programme!"},
  bookId : {type : String, required : true},
  authorId : {type : String, required : true},
  image : {type : String},
  dateStart : {type : Date, required : true}, // new Date("2016-05-18T16:00:00Z");
  dateStop : {type : Date, required : true},
  createdBy : {type : String, required : true}, //userId
  createdOn : {type : Date, default: Date.now},
  link : {type : String, required : true}
});
// Compilation du schéma en modèle
this.rdvLecture = db.model('rdvLecture', rdvLecturesSchema);
console.log("rdvLecture table ready");

```

BirdyDB

plumes

rdvlectures

Find

Indexes

Schema Anti-Pattern

FILTER {"filter":"example"}

QUERY RESULTS 1-1 OF 1

```

_id: ObjectId("608eeefa63dfdf32858e7bed8")
text: ""
speaker: "Mme Corcos"
title: "Les rêves du vieil homme"
bookId: "8"
authorId: "6"
image: "lions.jpeg"
dateStart: 2021-05-18T16:00:00.000+00:00
dateStop: 2021-05-18T19:00:00.000+00:00
createdBy: "6"
link: "www.zoom.com"
createdOn: 2021-05-02T18:29:58.903+00:00
__v: 0

```