

Article management system

User Story

David wants to have a platform where he can record and publish his thoughts. He can record the knowledge he learns in his life, the news he hears about and his views on these things in this article management system. The article management system needs to have the function of classification retrieval. David can also add new categories he wants so that he can choose to publish the article to the appropriate category.

Business Requirements

Users can register a new account.

The user should log in to the account.

Users can add the article category they want to publish.

Users can post articles.

Users can view their basic information.

The user can set up and change the profile photo.

Users can reset their own password.

Article needs to have an author

Articles need to have at least one classification

Article needs to have a title

The article needs to have a body

Articles can be published

The article can have a cover image

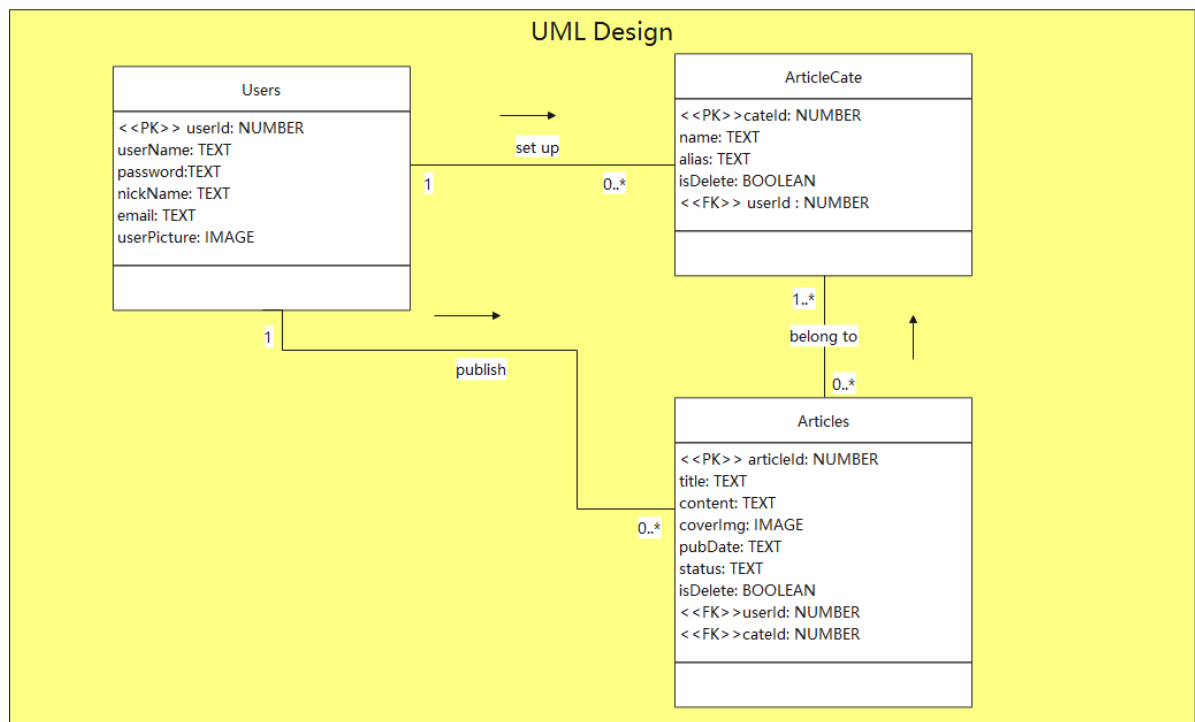
The article category needs to have an alias

Nouns: Users , Articles, Article Category

Verbs: record, publish

Attribs: Users(username, password, nickname, email, profile photo), Article Category(name, alias), Articles(title, content, cover image, publication date, status)

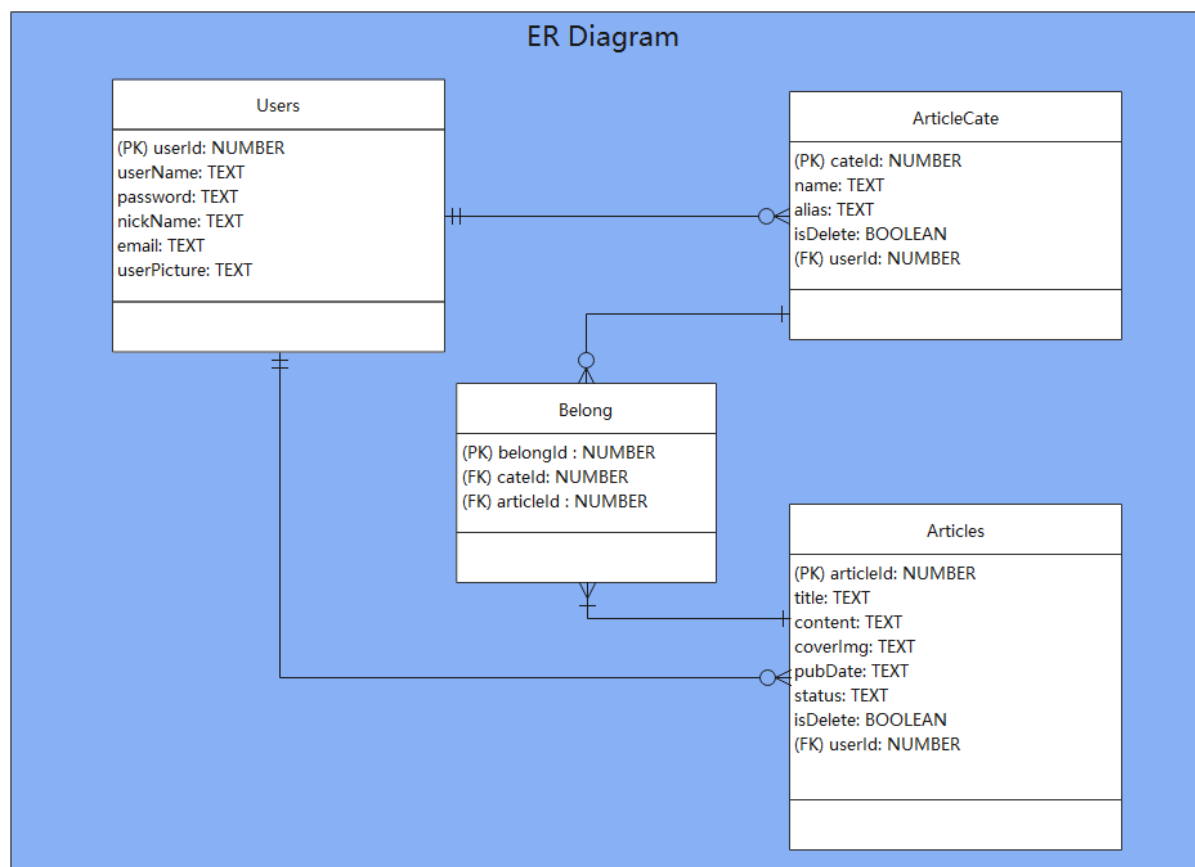
Conceptual Model(UML)



Logical Data Model

ER Diagram

<https://www.edrawmax.cn/online/share.html?code=b80d0af6362911ec90b29d87e1af6f2f>



Relational Schema

Users (userId, userName, nickName, email, userPicture);

ArticleCate (cateId, name, alias, isDelete, userId)

Articles (articleId, title, content, coverImg, pubDate, status, isDelete, userId, cateId)

Belong (belongId, cateId, articleId)

(Notes: PK = PK; FK = FK)

Proof:

Each column of our database table is an indivisible basic data item, so it satisfies the first paradigm of BCNF.

There is no partial functional dependence of non-key fields on any candidate key in the database table, that is, all non-key fields are completely dependent on any set of candidate keys.

(userId) -> (userName, password, nickName, email, userPicture)

(cateId) -> (name, alias, isDelete. (FK) userId: NUMBER)

(articleId) -> (title, content, coverImg, pubDate, status, isDelete, (FK) userId)

In our data table, there is no transfer function dependence of non-key fields on any candidate key field, which conforms to the third normal form.

Within the scope of functional dependency, our data table has been completely separated, and the exception of insertion and deletion has been eliminated.

Physical Model

Create database with SQLite

```
--This table is for article category.
CREATE TABLE ArticleCate (
  cateId INTEGER PRIMARY KEY ON CONFLICT ROLLBACK AUTOINCREMENT
  NOT NULL
  UNIQUE,
  name TEXT UNIQUE ON CONFLICT ROLLBACK
  NOT NULL,
  alias TEXT UNIQUE ON CONFLICT ROLLBACK
  NOT NULL,
  isDelete BOOLEAN NOT NULL
  DEFAULT (0),
  userId REFERENCES Users (userId)
  NOT NULL
);
--This table is for saving articles detail information.
CREATE TABLE Articles (
  articleId INTEGER PRIMARY KEY AUTOINCREMENT
  UNIQUE
  NOT NULL,
  title TEXT NOT NULL,
  content TEXT NOT NULL,
  coverImg TEXT,
```

```

pubDate DATETIME NOT NULL,
status TEXT NOT NULL,
isDelete BOOLEAN NOT NULL
DEFAULT (0),
userId INTEGER REFERENCES Users (userId)
NOT NULL
);
--This table is connecting Articles table and ArticleCate table which is many to
many relationship.
CREATE TABLE Belong (
belongId INTEGER PRIMARY KEY AUTOINCREMENT
NOT NULL
UNIQUE,
cateId INTEGER REFERENCES ArticleCate (cateId)
NOT NULL,
articleId INTEGER REFERENCES Articles (articleId)
NOT NULL
);
--This table is for saving user information.
CREATE TABLE Users (
userId INTEGER PRIMARY KEY ON CONFLICT ROLLBACK AUTOINCREMENT
NOT NULL
UNIQUE,
password TEXT (225) NOT NULL,
userName TEXT UNIQUE
NOT NULL,
nickName TEXT,
email TEXT,
userPicture TEXT
);

```

Successful create table

Figure 1: Creating tables in the database. The figure shows four screenshots of the DDL (Data Definition Language) editor for the 'project01' database. The tables being created are:

- ArticleCate (project01):** A table with columns: catelid (INTEGER, PRIMARY KEY, AUTOINCREMENT), name (TEXT, NOT NULL, UNIQUE ON CONFLICT ROLLBACK), alias (TEXT, NOT NULL, UNIQUE ON CONFLICT ROLLBACK), isDelete (BOOLEAN, NOT NULL, DEFAULT (0)), and userId (INTEGER, REFERENCES Users (userId), NOT NULL).
- Articles (project01):** A table with columns: articleId (INTEGER, PRIMARY KEY, AUTOINCREMENT), title (TEXT, NOT NULL), content (TEXT, NOT NULL), coverImg (TEXT, NOT NULL), pubDate (DATETIME, NOT NULL), status (TEXT, NOT NULL), isDelete (BOOLEAN, NOT NULL, DEFAULT (0)), and userId (INTEGER, REFERENCES Users (userId), NOT NULL).
- Belong (project01):** A table with columns: belongId (INTEGER, PRIMARY KEY, AUTOINCREMENT), catelid (INTEGER, REFERENCES ArticleCate (catelid), NOT NULL), articleId (INTEGER, REFERENCES Articles (articleId), NOT NULL), and isDelete (BOOLEAN, NOT NULL, DEFAULT (0)).
- Users (project01):** A table with columns: userId (INTEGER, PRIMARY KEY, AUTOINCREMENT), password (TEXT (225), NOT NULL), userName (TEXT, UNIQUE, NOT NULL), nickName (TEXT), email (TEXT), and userPicture (TEXT).

Figure 2: Viewing the table structure. The figure shows four screenshots of the 'Structure' tab for the same four tables. The columns and their data types are listed below:

Table	Column	Data type	Primary Key	Foreign Key	Unique
ArticleCate (project01)	1 catelid	INTEGER	Yes	No	No
	2 name	TEXT	No	No	Yes
	3 alias	TEXT	No	No	Yes
	4 isDelete	BOOLEAN	No	No	No
	5 userId	INTEGER	No	Yes	No
	6				
Articles (project01)	1 articleId	INTEGER	Yes	No	No
	2 title	TEXT	No	No	No
	3 content	TEXT	No	No	No
	4 coverImg	TEXT	No	No	No
	5 pubDate	DATETIME	No	No	No
	6 status	TEXT	No	No	No
	7 isDelete	BOOLEAN	No	No	No
	8 userId	INTEGER	No	Yes	No
Belong (project01)	1 belongId	INTEGER	Yes	No	No
	2 catelid	INTEGER	No	Yes	No
	3 articleId	INTEGER	No	Yes	No
Users (project01)	1 userId	INTEGER	Yes	No	No
	2 password	TEXT (225)	No	No	No
	3 userName	TEXT	No	No	Yes
	4 nickName	TEXT	No	No	No
	5 email	TEXT	No	No	No
	6 userPicture	TEXT	No	No	No

Import data into database

Figure 3: Importing data into the database. The figure shows four screenshots of the 'Data' tab for the same four tables. The data is imported from a CSV file. The columns and their values are listed below:

Table	Column	Value		
ArticleCate (project01)	catelid	1, 2, 3, 4, 5, 6, 7, 8		
	name	computer scient, finance, car, politic, cool things, study, Stock, life		
	alias	cs, fin, che, zhengche, nice thing, go to sch, GuShi, have fun		
	isDelete			
	userId			
Articles (project01)	articleId	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20		
	title	DonecVitaeNisi.avi, NonMauris.ppt, MalesuadaIn.gif, DiamCras.png, Ridiculus.xls, NonVelitNec.mp3, LiberoConvallisEget.mor, Justo.mpeg, EstLaciniaNisi.mp3, MaurisEgetMassa.tiff, Column: title, Data type: TEXT, Table: Articles, ROWID: 9, Constraints: NOT NULL, HendreritAt.jpeg, SedTristique.ppt, CongueEgetSemper.mp, ConvallisNuncPain.xls		
	Belong (project01)	belongId	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20	
		catelid	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20	
		articleId	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20	
		Users (project01)	userId	1, 2, 3, 4, 5, 6, 7, 8, 9, 10
			password	vb9aj9qeO, blheeP, cnm2Kr, dbT5JUQsm, sOk0eg9, FzdjYc, PTvh7vQ, gfY27n, I76iif, ZY1yKU
			userName	rgregory0, rtrowill1, kcherryman2, tanfusso3, drobbie4, tdowell5, dezzell6, clamie7, plintott8, mkiping9
			nickName	
			email	
			userPicture	

Testing table is work

```
--Check all articles in database.
select Articles.title from ArticleCate, Articles, Belong, Users
where Users.userId = ArticleCate.userId
and ArticleCate.cateId = Belong.cateId
and Articles.articleId = Belong.articleId;
--Select Articles which belong to study category.
select Articles.articleId,Articles.title,ArticleCate.name
from Articles, Belong, ArticleCate
where Articles.articleId = Belong.articleId
and Belong.cateId = ArticleCate.cateId
and ArticleCate.name = 'study';
--Find out total number of articles and category name of each category.
select ArticleCate.name, count(ArticleCate.name) as numberOfArticle
from Articles, Belong, ArticleCate
where Articles.articleId = Belong.articleId
and Belong.cateId = ArticleCate.cateId
group by ArticleCate.name
having numberOfArticle > 1;
--Find total of articles.
select count(*) from (select ArticleCate.name from Articles, Belong, ArticleCate
where Articles.articleId = Belong.articleId
and Belong.cateId = ArticleCate.cateId);

--Find out total number of articles and category name of each category and show
information whether the number is greater than 10 or not.
select ArticleCate.name, numberOfArticle,
case
    when numberOfArticle > 10 then 'The number Of Articles is greater than 10'
    when numberOfArticle = 10 then 'The number Of Articles is 10'
    else 'The number Of Articles is under 10'
end as numberOfArticleInfo from
(select ArticleCate.name, count(ArticleCate.name)
as numberOfArticle from Articles, Belong, ArticleCate
where Articles.articleId = Belong.articleId
and Belong.cateId = ArticleCate.cateId
group by ArticleCate.name
having numberOfArticle > 1);
```

```
20 select count(*) from (select ArticleCate.name from Articles, Belong, ArticleCate
21 where Articles.articleId = Belong.articleId
22 and Belong.cateId = ArticleCate.cateId);
```

23

24

25

26

Grid view

Form view



Total rows loaded: 1

count(*)

1 160

```

10 select Articles.title from ArticleCate, Articles, Belong, Users
11 where Users.userId = ArticleCate.userId
12 and ArticleCate.cateId = Belong.cateId
13 and Articles.articleId = Belong.articleId;
14
15
16

```

Grid view		Form view
		Total rows loaded: 160
	title	
1	DonecVitaeNisi.avi	
2	NonMauris.ppt	
3	Malesuadaln.gif	
4	DiamCras.png	
5	Ridiculus.xls	
6	NonVelitNec.mp3	
7	LiberoConvallisEget.mov	
8	Justo.mpeg	
9	EstLaciniaNisi.mp3	
10	MaurisEgetMassa.tiff	
11	ConvallisEget.pdf	
12	ProinRisusPraesent.xls	
13	Lacinia.ppt	

```

10 select Articles.articleId,Articles.title,ArticleCate.name
11 from Articles, Belong, ArticleCate
12 where Articles.articleId = Belong.articleId
13 and Belong.cateId = ArticleCate.cateId
14 and ArticleCate.name = 'study';
15
16

```

Grid view

Form view

1

Total rows loaded: 6

	articleId	title	name
1	6	NonVelitNec.mp3	study
2	90	DignissimVestibulumVestibulum.avi	study
3	94	NecNisiVolutpat.ppt	study
4	98	ElitSodales.xls	study
5	89	SedNisiNunc.ppt	study
6	93	MalesuadaInImperdiet.xls	study

```

13 select ArticleCate.name, count(ArticleCate.name) as numberOfArticle
14 from Articles, Belong, ArticleCate
15 where Articles.articleId = Belong.articleId
16 and Belong.cateId = ArticleCate.cateId
17 group by ArticleCate.name
18 having numberOfArticle > 1;

```

Grid view Form view	
1 Total rows loaded: 8	
name	numberOfArticle
1 Stock	6
2 car	34
3 computer scient	2
4 cool things	38
5 finance	34
6 life	6
7 politic	34
8 study	6

```

23 select ArticleCate.name, numberOfArticle,
24 case
25     when numberOfArticle > 10 then 'The number Of Articles is greater than 10'
26     when numberOfArticle = 10 then 'The number Of Articles is 10'
27     else 'The number Of Articles is under 10'
28 end as numberOfArticleInfo from
29 (select ArticleCate.name, count(ArticleCate.name)
30 as numberOfArticle from Articles, Belong, ArticleCate
31 where Articles.articleId = Belong.articleId
32 and Belong.cateId = ArticleCate.cateId
33 group by ArticleCate.name
34 having numberOfArticle > 1);

```

Grid view

Form view

</