

به نام خدا

# پایگاه داده بلاگ

## درس آزمایشگاه پایگاه داده

استاد: سید حامد صالح

دانشجو:

علی مانیان ۹۹۱۱۰۰۲۲۳۰۲۰۱۲

دانشکده فنی شهید مهاجر اصفهان

نیمسال اول ۱۴۰۰

# فهرست

2	نمودار ERD
3	توضیح پایگاه داده
3	جدول User
3	جدول Article
3	جدول Category
3	جدول Tag
4	جدول Comment
4	کوئری ها (Queries)
4	ایجاد جدول Articles
4	ایجاد جدول Categories
4	ایجاد جدول Comments
5	ایجاد جدول Tags
5	ایجاد جدول Users
5	Stored Procedure ها
5	ایجاد کاربر
6	دریافت مقاله
6	حذف دسته بندی
6	ویرایش برچسب
7	گزارشات خروجی
7	لیست همه ی کاربرانی که مقاله دارند
7	تعداد مقاله های هر کاربر
7	مرتب سازی لیست مقاله ها با عنوان صعودی و متن نزولی
7	مقاله های کاربری با آیدی یک که در دسته بندی یک نیستند
7	عنوان دسته بندی هایی که کاربر یک در مقالاتش استفاده نکرده

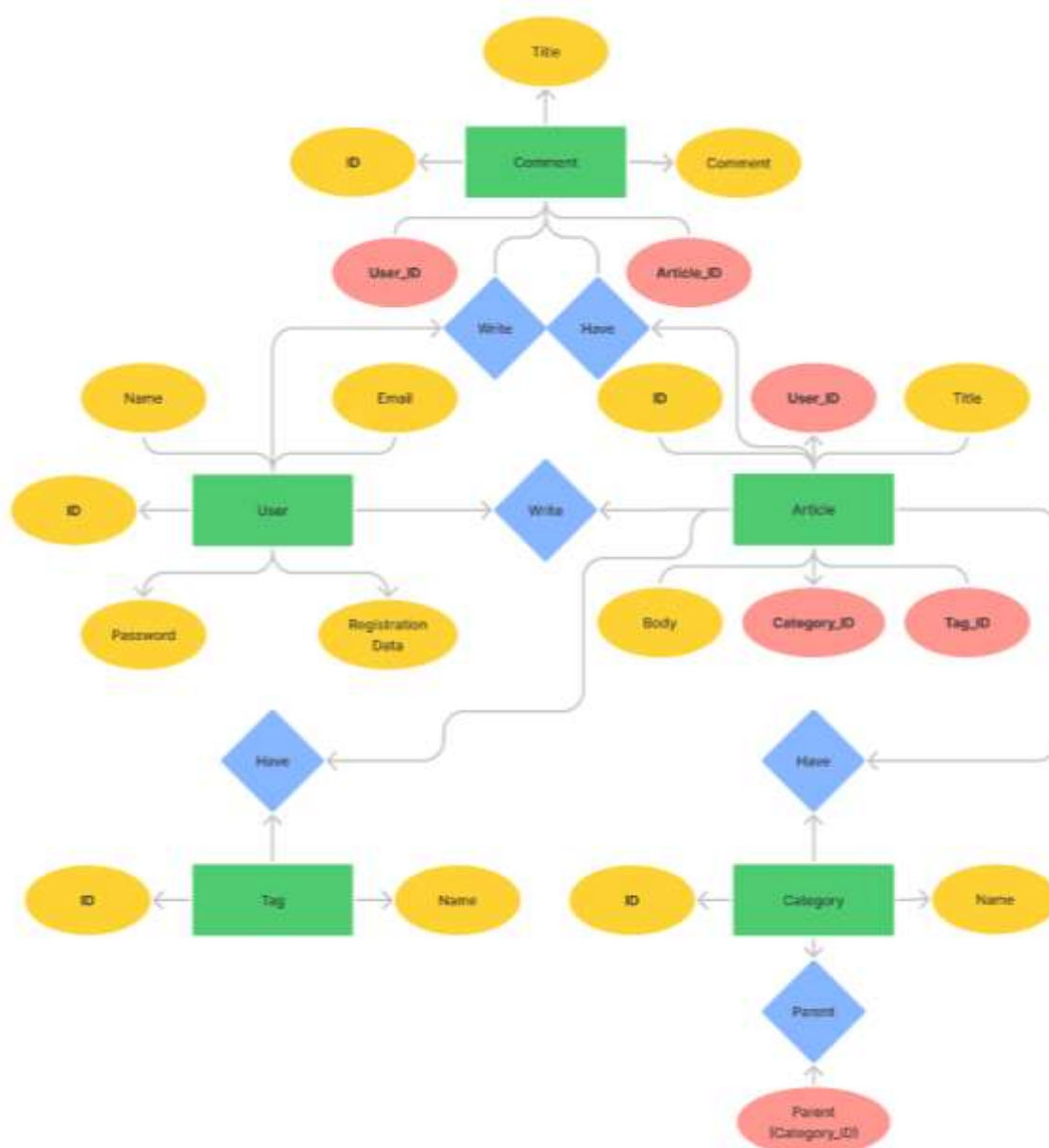
## نمودار ERD

نکته: با توجه به عدم پشتیبانی نرم افزار استفاده شده (FigJam) برای طراحی نمودار ER با استانداردها مطابقت ندارد اما سعی شده با رنگ تمایزها مشخص شود.

رنگ سبز: موجودیت‌ها

رنگ زرد: صفت‌های معمولی (در صورتی که BOLD باشد کلید اصلی است)

رنگ قرمز: صفت‌های وابسته به فیلدی در جداول دیگر



## توضیح پایگاه داده

### User جدول

این جدول شامل اطلاعات کاربران می‌باشد که می‌تواند آیدی کاربر که به صورت Auto Increment است، نام، ایمیل، رمزعبور به صورت هش شده، تاریخ ثبت‌نام و حتی فیلدهایی مانند آخرین ورود، شماره موبایل و... باشد.

### Article جدول

این جدول مربوط به مقاله‌های وبلاگ می‌باشد. شامل آیدی، آیدی کاربر به عنوان کلید خارجی جدول کاربران به عنوان نویسنده مقاله، عنوان مقاله، متن اصلی مقاله، آیدی دسته‌بندی که کلید خارجی از جدول دسته‌بندی و آیدی برچسب که کلید خارجی از جدول برچسب است می‌باشد.

**نکته:** در این نوع ساختار تنها یک دسته‌بندی و یک برچسب می‌توان برای یک مقاله تعیین کرد. برای اینکه بتوانیم بیشتر از یک دسته بندی یا بیشتر از یک برچسب برای یک مقاله تعیین کنیم می‌توانیم یک جدول واسطه مثلاً با نام Article\_Tags ایجاد کنیم و در آن جدول آیدی، آیدی مقاله، آیدی برچسب را قرار دهیم تا بتوانیم برای یک مقاله چندین برچسب انتخاب کنیم در این صورت دیگر نیازی به فیلد برچسب در جدول مقاله نیست. همین کار را می‌توان برای دسته‌بندی نیز انجام داد.

### Category جدول

این جدول که برای دسته‌بندی استفاده می‌شود شامل آیدی، نام دسته بندی و پدر یا همان والد دسته بندی است که والد در صورتی که NULL یا صفر باشد به این معنی است که والد ندارد و خود می‌تواند والد باشد.

**نکته:** بین والد جدول دسته بندی و آیدی جدول دسته‌بندی ارتباط وجود دارد به این صورت که در قسمت والد آیدی دسته بندی والد قرار داده می‌شود.

### Tag جدول

این جدول برای برچسب می‌باشد و شامل آیدی که کلید اصلی است و نام برچسب می‌باشد.

## جدول Comment

جدول دیدگاه شامل فیلدهای آیدی، عنوان دیدگاه، متن اصلی دیدگاه، آیدی کاربر به عنوان آیدی کاربردی که دیدگاه را نوشته است و آیدی مقاله به عنوان مقاله‌ای که برای آن دیدگاه نوشته شده می‌باشد.

## کوئری‌ها (Queries)

### ایجاد جدول Articles

```
DROP TABLE IF EXISTS [articles];
CREATE TABLE articles (
  [id] int NOT NULL IDENTITY,
  [title] varchar(255) NOT NULL,
  [body] varchar(max) NOT NULL,
  [user_id] int NOT NULL,
  [category_id] int NOT NULL,
  [tag_id] int NOT NULL,
  PRIMARY KEY ([id])
);
```

### ایجاد جدول Categories

```
DROP TABLE IF EXISTS [categories];
CREATE TABLE categories (
  [id] int NOT NULL IDENTITY,
  [name] varchar(255) NOT NULL,
  [parent] int NULL,
  PRIMARY KEY ([id])
);
```

### ایجاد جدول Comments

```
DROP TABLE IF EXISTS [comments];
CREATE TABLE comments (
  [id] int NOT NULL IDENTITY,
  [title] varchar(255) NOT NULL,
  [comment] varchar(max) NOT NULL,
  [user_id] int NOT NULL,
  [article_id] int NOT NULL,
  PRIMARY KEY ([id])
);
```

## ایجاد جدول Tags

```
DROP TABLE IF EXISTS [tags];
CREATE TABLE tags (
  [id] int NOT NULL IDENTITY,
  [name] varchar(255) NOT NULL,
  PRIMARY KEY ([id])
);
```

## ایجاد جدول Users

```
DROP TABLE IF EXISTS [users];
CREATE TABLE users (
  [id] int NOT NULL IDENTITY,
  [name] varchar(255) NOT NULL,
  [email] varchar(255) NOT NULL,
  [password] varchar(255) NOT NULL,
  [registration_date] date NOT NULL,
  PRIMARY KEY ([id])
);
```

## Stored Procedure ها

### ایجاد کاربر

```
CREATE PROCEDURE USP_CreateUser
(
  @Name NVARCHAR(255),
  @Email NVARCHAR(255),
  @Password NVARCHAR(255)
)
AS
BEGIN
  INSERT INTO [dbo].[users]
    ([name]
    ,[email]
    ,[password]
    ,[registration_date])
  VALUES
    (@Name
    ,@Email
    ,HASHBYTES('SHA2_512', @Password)
    ,GETDATE())
END
GO
```

## دریافت مقاله

```
CREATE PROCEDURE USP_GetArticle
(
    @ID INT
)
AS
BEGIN
    SELECT * FROM [dbo].[articles] WHERE id = @ID
END
GO
```

## حذف دسته‌بندی

```
CREATE PROCEDURE USP_DeleteCategory
(
    @ID INT
)
AS
BEGIN
    DELETE FROM [dbo].[categories]
    WHERE id = @ID
END
GO
```

## ویرایش برچسب

```
CREATE PROCEDURE USP_EditTag
(
    @ID INT,
    @Name NVARCHAR(255)
)
AS
BEGIN
    UPDATE [dbo].[tags]
    SET [name] = @Name
    WHERE id = @ID
END
GO
```

## گزارشات خروجی

لیست همه‌ی کاربرانی که مقاله دارند.

```
SELECT users.name, articles.title
FROM [users]
INNER JOIN [articles]
ON users.id = articles.user_id;
```

## تعداد مقاله‌های هر کاربر

```
SELECT users.name as 'نام', COUNT(articles.title) as 'تعداد مقاله'
FROM [users]
FULL JOIN [articles]
ON users.id = articles.user_id
GROUP BY users.name;
```

## مرتب‌سازی لیست مقاله‌ها با عنوان صعودی و متن نزولی

```
SELECT *
FROM [articles]
ORDER BY articles.title, articles.body DESC
```

## مقاله‌های کاربری با آیدی یک که در دسته بندی یک نیستند.

```
SELECT *
FROM [articles]
WHERE articles.user_id = 1 AND articles.category_id <> 1
```

## عنوان دسته‌بندی‌هایی که کاربر یک در مقالاتش استفاده نکرده

```
LECT categories.name FROM categories WHERE categories.id
NOT IN(SELECT articles.category_id
FROM [articles]
WHERE articles.user_id = 1)
```