15、数据库字段对应

## 1、主键对应

|  |  |  |
| --- | --- | --- |
| 主键Id | Long | bigint(20) 不要看到数字就设置为long，long也是有字数限制的 |
| 字符串 | String | varcahr(20) |
| status(0,1,2,) | Integer | int(11) |
| 金额 | BigDecimal | decimal(19,2) ,保留2位小数  mysql中的decimal字段，声明语法为DECIMAL(M,D)。 DECIMAL(M,D)中D值的是小数部分的位数，若插入的值未指定小数部分或者小数部分不足D位则会自动补到D位小数，若插入的值小数部分超过了D为则会发生截断，截取前D位小数(四舍五入截取)。M值得是整数部分加小数部分的总长度，也即插入的数字整数部分不能超过M-D位，否则不能成功插入，会报超出范围的错误。  先保证小数点，再保证整数 decimal精度类型，函数完整格式DECIMAL(M,D) M是是只的最大精度数位，1-65 D是小数点右侧数位0-30 举例说明，11615.23653234568这个数存你说的三个格式 decimal：11615 decimal(3)：999 decimal(3,2)：9.99 decimal(10,5)11615.23653 超出精度范围的数会被强制进位并只显示数据类型定义的格式 |
| 日期 | Date | timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP, (自动创建) |
|  |  | timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,自动更新 |
|  | Date | datetime |
|  | Date | date 只有日期没有时分秒 |

### 1、索引

@Entity  
@Table(name = "advertiser\_cash\_flow",  
 indexes = {@Index(name = "index\_userId",columnList = "userId")})  
public class AdvertiserCashFlow implements Serializable {

CREATE TABLE `advertiser\_cash\_flow` (

`id` bigint(20) NOT NULL AUTO\_INCREMENT,

`balance` decimal(19,2) DEFAULT NULL,

`cdate` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP,

PRIMARY KEY (`id`),

KEY `index\_userId` (`userId`)

)

## 2、举例说明

@Id  
@GeneratedValue(strategy = GenerationType.***IDENTITY***)  
private Long id;  
  
private Long adminId; //创建者id  
  
private String userName ; //新客户名称【id】邮箱  
  
private BigDecimal balance; //老客户金额

private Integer auditStatus ; //审核状态 0 待审核 1 审核通过

@Temporal(TemporalType.***TIMESTAMP***)  
@Column(columnDefinition="TIMESTAMP DEFAULT CURRENT\_TIMESTAMP",insertable = true,updatable = false)  
private Date cdate;

private Date udate;

create table tableTest(

id BIGINT(20) not null auto\_increment,

adminId BIGINT(20) default null,

userName varchar(200) default null,

auditStatus int(11) not NULL default 1,

balance DECIMAL(19,2),

cdate timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP,

udate timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

unique key key\_ adminId(adminId ),

unique index index\_ userName (userName),

PRIMARY key (id));

# 警告

## 1、timestamp 多个日期，如果可能为空，则建议使用datetime（默认值建议设置为）

0001-01-01 00:00:00，因为0000-00-00 00:00:00mysql不能保存，而且会报错，

普通字段不要设置为timestamp，timestamp列必须有默认值，默认值可以为“0000-00-00 00:00:00”，但不能为null。如果我们在save实体的时候，没有给相关timestamp设置值，那么他就会自动由mysql将当前时间设置进去，

@Temporal(TemporalType.***TIMESTAMP***)  
@Column(columnDefinition="TIMESTAMP DEFAULT CURRENT\_TIMESTAMP",insertable = true,updatable = false)  
@ApiModelProperty(hidden = true)  
private Date cdate;  
  
@UpdateTimestamp  
@Temporal(TemporalType.***TIMESTAMP***)  
@ApiModelProperty(hidden = true)  
private Date udate;

## 2、数据库设计尽量使用 user\_id 下划线加小写，不要使用userId

## 3、字段尽量不能设置null，这样不方便建立索引，为了防止为null，我们可以给赋予初始值，今后建议 default '',给一个空串，空串不占内存空间，NULL是占内存空间的

### 1、举例

BigDecimal totalBalance = BigDecimal.***ZERO***

## 2、使用，columnDefinition="TIMESTAMP DEFAULT CURRENT\_TIMESTAMP"