

# 数据脱敏实训

## 数据脱敏

数据脱敏（Data Masking）是一种[信息安全技术，旨在保护敏感信息和隐私数据，防止未经授权的访问或泄露。它通过对原始数据进行有策略的修改或替换，创建一个看上去与原数据相似但不含真正敏感细节的数据副本，以供非生产环境如开发、测试、分析或培训等用途中安全使用。

## 静态脱敏

### 方式原理

1. 替换：使用相似业务特征数据对敏感原始数据替代
2. 仿真:仿造原始数据敏感内容生成符合敏感数据原始内容语意和句意的新数据替换原来的敏感数据
3. 加密:利用加密算法进行加密
4. 遮掩:利用敏感符号中的部分内容掩饰敏感数据
5. 混淆:打乱敏感数据
6. 偏移主要是通过对敏感数据内容进行随机移位来改变数据内容，偏移算法一般适用于数 值型数据。（注意数值合理）
7. 均值化:对数值求平均值后在偏移
8. 重采样:重采样是指从原始数据集中重新采集数据样本，然后将采集到的样本代替原始位置上的 数据。
9. 取整：取整是指将原始数据中的数值进行四舍五入取整，取整的上限以数据的最高位为度量。

### 静态脱敏实训

#### 1.创建数据表

	first_name	phone	email	web	salary	age	birthday
<input type="checkbox"/> 编辑 <input type="checkbox"/> 复制 <input type="checkbox"/> 删除	hanli	10086	fox@qq.com	https://www.bing.com/search	16000	21	2004-12-23

#### 2. 测试替换

```
1 UPDATE userinfo SET phone= CONCAT(  
2     '1',  
3     FLOOR(RAND() * (9 - 3 + 1)) + 3,  
4     FLOOR(RAND() * 10),  
5     FLOOR(RAND() * 10),  
6     FLOOR(RAND() * 10),  
7     FLOOR(RAND() * 10),  
8     FLOOR(RAND() * 10),  
9     FLOOR(RAND() * 10),  
10    FLOOR(RAND() * 10),  
11    FLOOR(RAND() * 10)  
12 ) WHERE first_name='hanli';  
13
```

+ 选项	first_name	phone	email	web	salary	age	birthday
<input type="checkbox"/> 编辑 <input type="checkbox"/> 复制 <input type="checkbox"/> 删除	hanli	1785125074	fox@qq.com	https://www.bing.com/search	16000	21	2004-12-23

改变了phone，但是随机替换不可逆不利于后续数据的使用。

2. 仿真

--仿真first\_name:

```
1 CREATE TABLE name_mapping(  
2   real_name VARCHAR(255),  
3   fake_name VARCHAR(255)  
4 );
```

创建first\_name表。

显示全部 | 行数: 25

+ 选项

real_name	fake_name
hanli	lifeiyu
Evil	Eail
France	Fall

创建表name表。

```
1 UPDATE userinfo  
2 JOIN name_mapping ON userinfo.first_name=name_mapping.real_name  
3 SET userinfo.first_name=name_mapping.fake_name;
```

改变first\_name

选项

▼

	first_name	phone	email	web
编辑 复制 删除	lifeiyu	1785125074	fox@qq.com	https://www.bing.com

3. 加密

加密web

```
1 UPDATE userinfo SET web=MD5(web) WHERE first_name='hanli'
```

得到结果

phone	email	web	salary	age	birthday
1785125074	fox@qq.com	f195bb9287c77909fc92c44288193a5b	16000	21	2004-12-23

4. 遮掩

以email为例子。

```
1 UPDATE userinfo SET email=CONCAT(REPEAT('*',LOCATE('@',email)-1),
2 SUBSTRING(email,locate('@',email))) WHERE first_name='hanli';
```

将@前面的部分用\*\*\*遮掩。

+ 选项						
↔ T ↔		▼	first_name	phone	email	web
<input type="checkbox"/>	编辑	复制	删除	hanli	1785125074	***@qq.com f195bb9287c77909fc9

## 5. 混淆

以phone字段为例子。

```
1 UPDATE userinfo SET phone = CONCAT( LEFT(phone, 3), LPAD(FLOOR(RAND() *
2 1000000000),8, '0'))where first_name='hanli'
```

选项						
-T→			first_name	phone	email	web
	编辑		复制		删除	hanli 10085903139 fox@qq.com f195bb9287c77909

**LPAD(..., 8, '0')**

- 确保随机数长度为 8 位：
  - 如果生成的随机数不足 8 位，用 0 补齐。

## 6. 偏移

DATADIFF()计算到birthday的日期间的差，然后生成一个随机生成到今天的生日。INTERVAL计算日期和时间

```
1 UPDATE userinfo
2 SET birthday = LEAST(
3     DATE_ADD(birthday, INTERVAL FLOOR(RAND() * DATEDIFF(CURRENT_DATE(),
4     birthday)) DAY),
5     CURRENT_DATE()
6 )
7 WHERE first_name = 'hanli';
```

选项						
-T→			first_name	phone	email	web
	编辑		复制		删除	hanli 10085903139 fox@qq.com f195bb9287c77909

## 7. 均值化

插入两个新的数据

```

1 INSERT INTO userinfo (first_name, phone, email, web, salary, age, birthday)
VALUES ('sam', '12301', 'donk@qq.com', 'https://www.bing.com/', '12020', '22',
'2003-01-01'), ('jack', '91145', 'niko@qq.com', 'https://www.bing.com/',
'14320', '22', '2003-01-01')

```

生成一个临时的表

```

1 CREATE TEMPORARY TABLE temp_average_salary AS SELECT ROUND(AVG(salary)) AS
average_salary FROM userinfo;

```

用工资平均值然后加上范围内的随机的偏移量

```

1 UPDATE userinfo
2 SET salary = (
3     SELECT ROUND(average_salary + (RAND() * 100 - 50))
4     FROM temp_average_salary
5 )
6 WHERE salary IS NOT NULL;

```

两个代码一起运行。因为临时表在结束时候会被自动删除。

	salary	age	birthday
7909fc92c44288193a5b	14129	21	2008-09-10
ing.com/	14128	22	2003-01-01
ing.com/	14089	22	2003-01-01

## 8. 重采样

从字段中的web随机选取去一个给所有行中的Web赋值为此随机的网址

```

1 UPDATE userinfo
2 SET web = (SELECT web FROM (SELECT web FROM userinfo ORDER BY RAND() LIMIT 1)
AS subquery);

```

first_name	phone	email	web	salary	age	birthday
ali	10085903139	fox@qq.com	https://www.bing.com/	14129	21	2008-09-10
n	12301	donk@qq.com	https://www.bing.com/	14128	22	2003-01-01
c	91145	niko@qq.com	https://www.bing.com/	14089	22	2003-01-01

## 9. 取整

对工资取整。

```
1 UPDATE userinfo SET salary = ROUND(salary,-1 * FLOOR(LOG10(salary))) where
2 first_name='hanli';
```



## 动态脱敏实训

### 1. 制作表并插入数据

```
1 CREATE TABLE userinfo_dynamic (
2     id_number VARCHAR(20),
3     password VARCHAR(15),
4     postcode VARCHAR(10),
5     banknumber VARCHAR(20),
6     phone VARCHAR(12),
7     email VARCHAR(30)
8 );
9
```

```
1 INSERT INTO userinfo_dynamic (id_number, password, postcode, banknumber,
2 phone, email) VALUES
3 ('320125198901011234', 'Abc123456', '210000', '6225880112345678',
4 '13812345678', 'example1_123@gmail.com'),
5 ('330326199012122345', 'Passw0rd', '210000', '6225880112345678', '215300',
6 'example1_123@163.com'),
7 ('440308199012133456', 'StrongPwd456', '518000', '6229123456789012',
8 '17334567890', 'example1_123@qq.com'),
9 ('510106199012144567', 'Passw0rd789', '610000', '6228123456789012',
10 '13645678901', 'example1_123@yahoo.com'),
11 ('610112199012155678', 'P4ssw0rd123', '710000', '6227123456789012',
12 '15356789012', 'example1_123@qq.com');
```

进行数据库连接，代码见项目github源码

```
C:\Python3.11\python.exe D:\wdnmd\Crypto\pythonProject\mysql.py
('320125198901011234', 'Abc123456', '210000', '6225880112345678', '13812345678', 'example1_123@gmail.com')
('330326199012122345', 'Passw0rd', '210000', '6225880112345678', '215300', 'example1_123@163.com')
('440308199012133456', 'StrongPwd456', '518000', '6229123456789012', '17334567890', 'example1_123@qq.com')
('510106199012144567', 'Passw0rd789', '610000', '6228123456789012', '13645678901', 'example1_123@yahoo.com')
('610112199012155678', 'P4ssw0rd123', '710000', '6227123456789012', '15356789012', 'example1_123@qq.com')
```

。接下来进行动态脱敏，显识别敏感内容，在对敏感内容脱敏

```
1 ('320125198901011234', 'Abc123456', '210000', '6225880112345678',
  '13812345678', 'example1_123@gmail.com')
2 ('330326199012122345', 'Passw0rd', '210000', '6225880112345678', '215300',
  'example1_123@163.com')
3 ('440308199012133456', 'StrongPwd456', '518000', '6229123456789012',
  '17334567890', 'example1_123@qq.com')
4 ('510106199012144567', 'Passw0rd789', '610000', '6228123456789012',
  '13645678901', 'example1_123@yahoo.com')
5 ('610112199012155678', 'P4ssw0rd123', '710000', '6227123456789012',
  '15356789012', 'example1_123@qq.com')
```

6  
7 检测到的敏感数据:

```
8 (0, 0)
9 (0, 1)
10 (0, 2)
11 (0, 3)
12 (0, 4)
13 (0, 5)
14 (1, 0)
15 (1, 1)
16 (1, 2)
17 (1, 3)
18 (1, 4)
19 (1, 5)
20 (2, 0)
21 (2, 1)
22 (2, 2)
23 (2, 3)
24 (2, 4)
25 (2, 5)
26 (3, 0)
27 (3, 1)
28 (3, 2)
29 (3, 3)
30 (3, 4)
31 (3, 5)
32 (4, 0)
33 (4, 1)
34 (4, 2)
35 (4, 3)
36 (4, 4)
37 (4, 5)
```

38  
39 脱敏后的数据查询结果:

```
40 ['3201*****11234', 'Ab****456', '2***00', '6225*****5678',
  '138*****678', 'examp*****il.com']
41 ['3303*****22345', 'Pa****rd', '2***00', '6225*****5678', '2***00',
  'examp*****3.com']
42 ['4403*****33456', 'Str*****456', '5***00', '6229*****9012',
  '173*****890', 'examp*****q.com']
43 ['5101*****44567', 'Pas*****789', '6***00', '6228*****9012',
  '136*****901', 'examp*****oo.com']
44 ['6101*****55678', 'P4s*****123', '7***00', '6227*****9012',
  '153*****012', 'examp*****q.com']
45
```

棚山结果。