

## 三级模式

---

### 模式

介于用户级和物理级之间，是所有用户的公共数据视图(全局的数据视图)，是数据库管理员看到和使用的数据库，一个数据库只存在一个DBA视图，即一个数据库只有一个模式

### 外模式

最接近用户的一级数据库，是用户看到和使用的 局部数据 的逻辑结构和特征的描述，又称为用户视图。外模式是模式的子集，一个数据库可以用多个外模式

### 内模式

内模式也称为存储模式，是数据物理结构和存储方式的藐视，一个数据库只有一个内模式

## 两级映像

---

### 外模式/模式映像

在外模式和模式之间存在外模式/模式映像，用于定义外模式和模式之间的对应关系

一个模式有多个外模式，每一个外模式，数据库系统都有一个外模式/模式映像

外模式/模式映像通常包含在各自外模式的描述中

### 模式/内模式映像

在模式和内模式之间存在模式/内模式映像，用于定义模式和内模式之间的对应性。定义了数据库全局逻辑结构和存储结构之间的对应关系。

数据库只有一个模式，也只有一个内模式，所以模式/内模式映像是唯一的。

模式/内模式映像通常包含在模式描述中

## 数据模式

---

实体型：具有相同属性的实体必然的特征，例如学生（学号，性别，出生年月）

在微信用户及聊天群聊天业务模型中，所涉及到的所有实体型如下

- 用户(账号，昵称，密码，性别，生日，电话，邮箱，头像，年龄)
- 好友(账号，备注，所属好友ID)
- 消息(消息ID，内容，状态，时间，类型，发送者，接收者)
- 群聊(群ID，名称，创建时间，群主，图标)
- 会话(会话ID，发送者ID，接收者ID，类型，最后消息ID)

## 数据建立

使用sql脚本在MySQL中建立实体型所对应的几个表，并向其中插入数据。

```
1  --build&insert.sql
2  create database MyIM;
3  use MyIM;
4  create table Users(
5      U_ID Int not null primary key,
6      U_NickName Varchar(20),
7      U_PassWord Varchar(20),
8      U_Sex Varchar(20) ,
9      U_Birthday Datetime ,
10     U_Telephone Varchar(30),
11     U_Email Varchar(50),
12     U_HeadPortrait Varchar(100),
13     U_Age Int
14
15 );
16 INSERT INTO Users
17 (U_ID,U_NickName ,U_PassWord,U_Sex ,U_Birthday,U_Telephone ,U_Email
18 ,U_Age)
19 VALUES
20 (18221023,'xiehengjian',123456,'male','1999-05-
21 05','13195529212','18221023@bjtu.edu.cn',20),
22 (18221024,'zhangsan',123456,'male','1999-05-
23 06','13195529213','18221024@bjtu.edu.cn',20),
24 (18221025,'lisi',123456,'male','1999-05-
25 07','13195529214','18221025@bjtu.edu.cn',20);
26
27 create table Friends(
28     F_ID Int not null primary key,
29     F_FriendID Int ,
30     F_Name Varchar(30),
31     F_UserID Int
32
33 );
34 INSERT INTO Friends
35 (F_FriendID,F_Name,F_UserID )
36 VALUES(18221024,'zhangsan',18221023);
```

```

33
34 create table Messages(
35     M_ID      Int not null primary key,
36     M_PostMessages  Text,
37     M_status   INT,
38     M_Time     Datetime ,
39     M_MessagesTypeID  Int ,
40     M_FromUserID  Int ,
41     M_ToUserID   Int
42
43 );
44 INSERT INTO Messages
45 (M_ID
46 ,M_PostMessages,M_status,M_Time,M_MessagesTypeID,M_FromUserID,M_ToUse
47 rID)
48 VALUES
49 (0,'hello',0,'2020-10-1',1,18221024,18221024);
50
51 create table User_Groups(
52     UG_ID      Int not null primary key,
53     UG_Name     Varchar(30),
54     UG_CreateTime  Datetime ,
55     UG_AdminID   Int ,
56     UG_ICon      Varchar(30)
57 );
58 INSERT INTO User_Groups
59 (UG_Name,UG_CreateTime,UG_AdminID)
60 VALUES("mygroup",'2020-10-1',18221023);
61
62 create table Session(
63     S_ID Int not null primary key,
64
65     S_FromUserID      Int ,
66     S_TypeID Int ,
67     S_FinalMessageID INT ,
68     S_ToUserID       Int
69 );
70 INSERT INTO Session
71 (S_TypeID,S_FromUserID ,S_ToUserID ,S_FinalMessageID )
72 VALUES(0,18221024,10221023,0);

```

进入MySQL并执行脚本

```
creative — com.docker.cli < docker exec -it mymysql bash — 86x40
[root@f4c960fba1fe:~# mysql -uroot -p
[Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.6.49 MySQL Community Server (GPL)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> source ~/build&insert.sql
Query OK, 1 row affected (0.00 sec)

Database changed
Query OK, 0 rows affected (0.00 sec)

Query OK, 3 rows affected, 3 warnings (0.00 sec)
Records: 3  Duplicates: 0  Warnings: 3

Query OK, 0 rows affected (0.01 sec)

Query OK, 1 row affected, 1 warning (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 1 row affected (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

Query OK, 1 row affected, 1 warning (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 1 row affected, 1 warning (0.00 sec)

mysql>
```

查看表中数据

## 原型设计

参考微信的UI设计，设计了 [会话页](#) 与 [聊天页](#) 两个用户界面。

使用工具为 [Axure Pro 9](#)，所用图标来自于[Iconfont](#)





## 好友请求

7:45

张一鸣请求添加您为好友



## 马云

8:44

钱不够花就跟我说，别客气



## 马化腾

7:56

上次给你的一个亿花完了没



## 李彦宏

昨天8:21

健哥，借我点钱，百度要不行了

8:31



马云

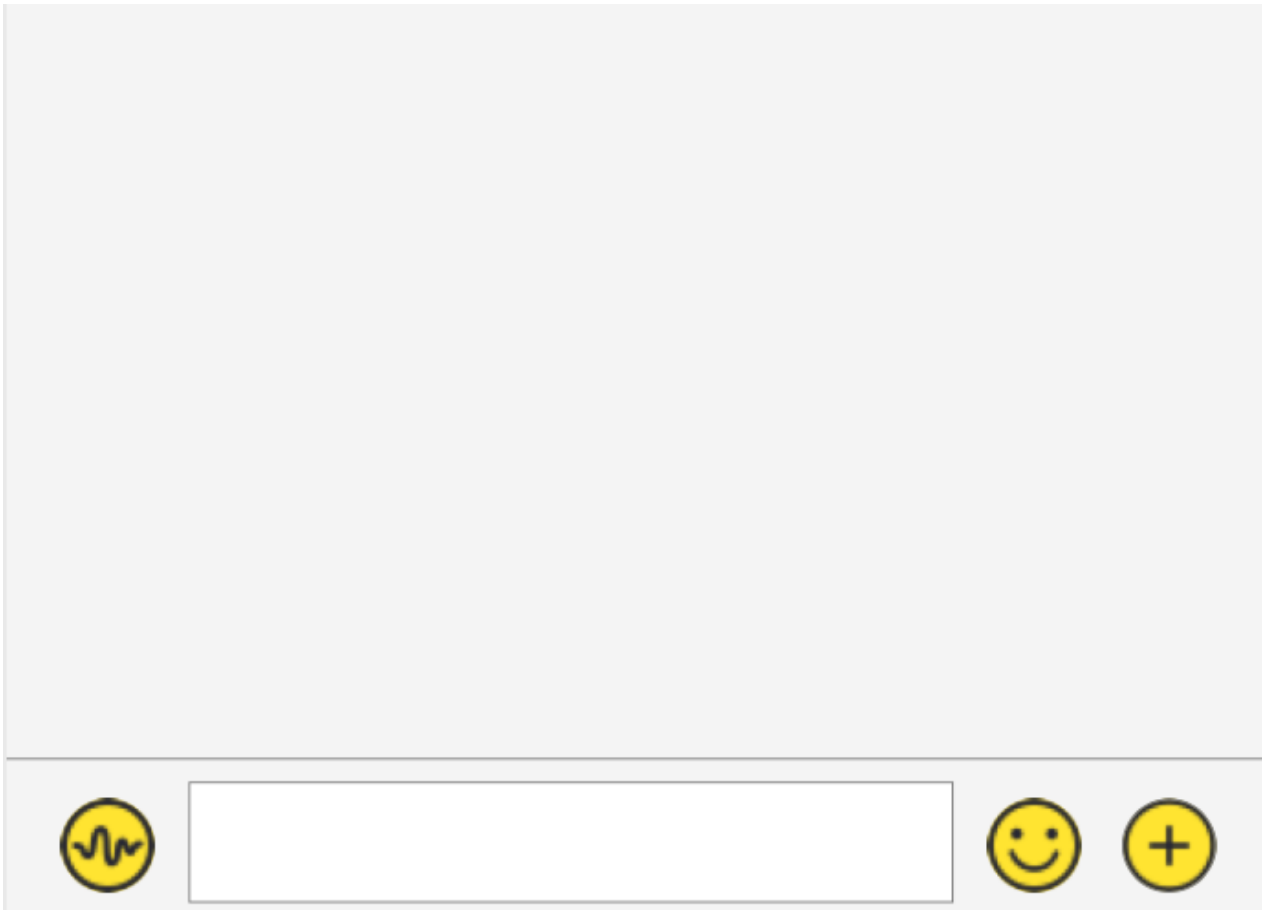
10月1日 13:23



自从辞职CEO以后，钱也挣得少了

钱不够花就跟我说，别客气





## 数据规模估算

---

由于即时通讯业务中存储的如会话和消息等数据的增长速度非常快，相比之下用户与群组等数据可以忽略不计，平均每人发送消息可达100条/天，一百万日活的即时通讯软件每天处理数据记录可达一亿，对于这种大规模的数据存储，对于大型的表，或者包含大量重复文本或数值数据的表，应该考虑使用 COMPRESSED(压缩的) 行格式。这样只需要较少的 I/O 就可以把数据取到缓冲池，或执行全表扫描。考虑使用 OPTIMIZE TABLE 语句重新组织表并压缩浪费的空间。对重新组织后的表进行全表扫描所需要的 I/O 会更少。