



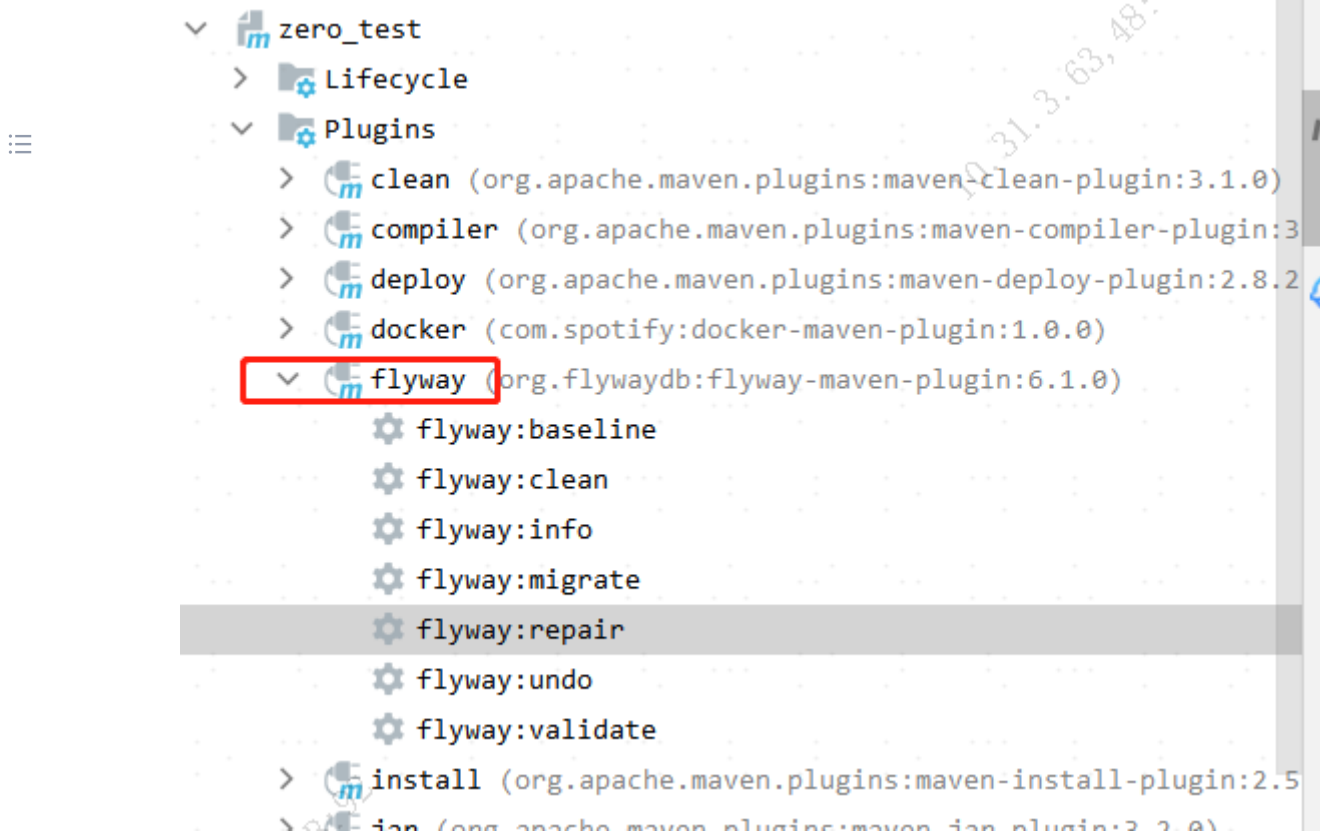
简介

Flyway 是一款开源的数据库版本管理工具。它可以很方便的在命令行中使用， 或者在Java应用程序中引入， 用于管理我们的数据库版本。

- * 自己写的SQL忘了在所有环境执行。
- * 别人写的SQL我们不能确定是否都在所有环境执行过了。
- * 有人修改了已经执行过的SQL， 期望再次执行。
- * 需要新增环境做数据迁移。
- * 每次发版需要手动控制先发DB版本， 再发布应用版本。

一、依赖及插件

```
1  <!-- flyway数据库版本控制 -->
2  <dependency>
3      <groupId>org.flywaydb</groupId>
4      <artifactId>flyway-core</artifactId>
5      <version>6.1.0</version>
6  </dependency>
7
8  <!-- flyway插件： 不需要启动项目就可执行， 右侧工具类添加flyway方便操作 -->
9  <plugin>
10     <groupId>org.flywaydb</groupId>
11     <artifactId>flyway-maven-plugin</artifactId>
12     <version>6.1.0</version>
13     <configuration>
14         <url>jdbc:mysql://10.207.0.164:3307/sf_mall?
            useUnicode=true&characterEncoding=utf8&useSSL=false&serverTimezone=Asia/Shanghai&allowMultiQueries=tru
15         <user>root</user>
16         <password>root</password>
17         <driver>com.mysql.cj.jdbc.Driver</driver>
18         <locations>
19             <!--数据库sql文件目录，也可以省略${basedir}-->
20             <location>filesystem:${basedir}/src/main/resources/db/migration/</location>
21         </locations>
22         <table>flyway_schema_history_zero</table>
23         <!--社区版不支持undo-->
24         <!--<undoSqlMigrationPrefix>U</undoSqlMigrationPrefix>-->
25         <!--<sqlMigrationPrefix>V</sqlMigrationPrefix>
26         <repeatableSqlMigrationPrefix>R</repeatableSqlMigrationPrefix>
27         <sqlMigrationSeparator>__</sqlMigrationSeparator>
28         <sqlMigrationSuffixes>.sql</sqlMigrationSuffixes>
29         <encoding>UTF-8</encoding>-->
30     </configuration>
31 </plugin>
```



1. baseline

对已经存在数据库Schema结构的数据库一种解决方案。实现在非空数据库新建MetaData表，并把Migrations应用到该数据库；也可以在已有表结构的数据库中实现添加Metadata表。

2. clean

清除掉对应数据库Schema中所有的对象，包括表结构，视图，存储过程等，clean操作在dev和test阶段很好用，但在生产环境务必禁用。

3. info

用于打印所有的Migrations的详细和状态信息，也是通过MetaData和Migrations完成的，可以快速定位当前的数据库版本。

4. repair

repair操作能够修复metaData表，该操作在metadata出现错误时很有用。

5. undo

撤销操作，社区版不支持。

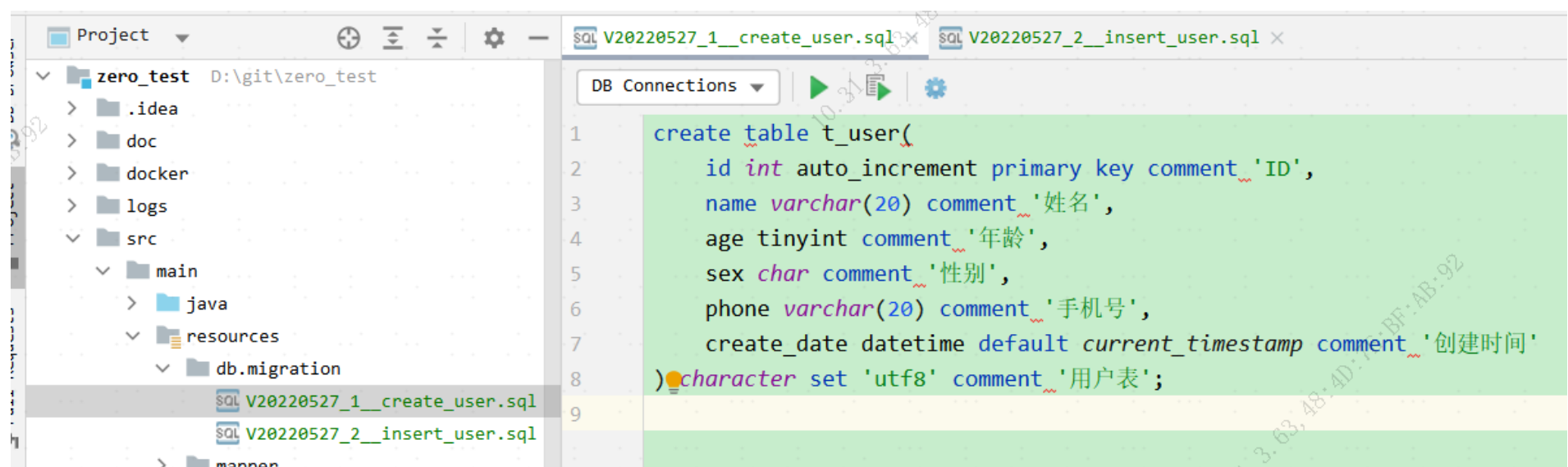
6. validate

验证已经apply的Migrations是否有变更，默认开启的，原理是对比MetaData表与本地Migrations的checkNum值，如果值相同则验证通过，否则失败。

二、yml配置

```
1 # flyway
2 spring:
3   flyway:
4     # 是否启用flyway
5     enabled: true
6     # 编码格式，默认UTF-8
7     encoding: UTF-8
8     # 修改默认表名，可实现多个模块分别通过Flyway进行数据库版本控制
9     #table: zero_t_user
10    # 迁移sql脚本文件存放路径，默认db/migration
11    locations: classpath:db/migration
12    # 迁移sql脚本文件名称的前缀，默认V
13    sql-migration-prefix: V
14    # 迁移sql脚本文件名称的分隔符，默认2个下划线__
15    sql-migration-separator: __
16    # 迁移sql脚本文件名称的后缀
17    sql-migration-suffixes: .sql
18    # 迁移时是否进行校验，默认true
19    validate-on-migrate: true
20    # 当迁移发现数据库非空且存在没有元数据的表时，自动执行基准迁移，新建schema_version表
21    baseline-on-migrate: true
22    # 指定基准版本，忽略20210809版本以及之前的所有migration
23    #baseline-version=20210809
24    # 是否禁用清库
25    #clean-disabled: true
```

三、migration目录



四、文件命名规则

V_ 执行一次，用于初始化或版本升级

R_ 重复执行，命名不能加版本号

U_ 执行回滚，过于机械粗暴，建议使用V_

Flyway 是如何比较两个 SQL 文件的先后顺序呢？它采用 采用左对齐原则, 缺位用 0 代替。举几个例子：

1.0.1.1 比 1.0.1 版本高。

1.0.10 比 1.0.9.4 版本高。

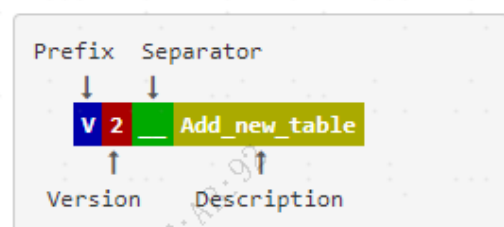
1.0.10 和 1.0.010 版本号一样高, 每个版本号部分的前导 0 会被忽略。

Flyway 将 SQL 文件分为 **Versioned**、**Repeatable** 和 **Undo** 三种：

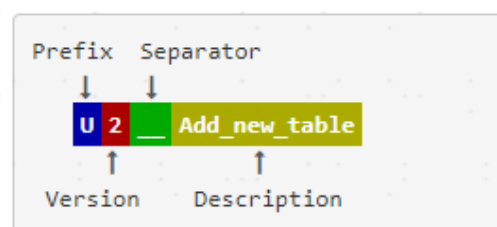
- **Versioned** 用于版本升级, 每个版本有唯一的版本号并只能执行一次。
- **Repeatable** 可重复执行, 当 Flyway 检测到 **Repeatable** 类型的 SQL 脚本的 checksum 有变动, Flyway 就会重新应用该脚本. 它并不用于版本更新, 这类的 migration 总是在 **Versioned** 执行之后才被执行。
- **Undo** 用于撤销具有相同版本的版本化迁移带来的影响。但是该回滚过于粗暴，过于机械化，一般不推荐使用。一般建议使用 **Versioned** 模式来解决。

这三种的命名规则如下图：

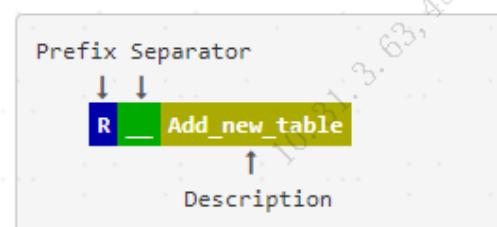
Versioned Migrations



Undo Migrations



Repeatable Migrations



五、测试

1.初次执行

```
[ ] 11:01:32.620 [restartedMain] INFO o.f.c.i.l.VersionPrinter - [info,49] - Flyway Community Edition 6.1.0 by Redgate
[ ] 11:01:33.692 [restartedMain] INFO c.a.d.p.DruidDataSource - [init,998] - {dataSource-1} initied
[ ] 11:01:34.334 [restartedMain] INFO o.f.c.i.d.DatabaseFactory - [info,49] - Database: jdbc:mysql://10.207.0.164:3307/sf_mall (MySQL 5.7)
[ ] 11:01:34.381 [restartedMain] INFO o.f.c.i.c.DbValidate - [info,49] - Successfully validated 2 migrations (execution time 00:00.022s)
[ ] 11:01:34.428 [restartedMain] INFO o.f.c.i.s.JdbcTableSchemaHistory - [info,49] - Creating Schema History table `sf_mall`.`flyway_schema_history` ...
[ ] 11:01:34.766 [restartedMain] INFO o.f.c.i.c.DbMigrate - [info,49] - Current version of schema `sf_mall`: << Empty Schema >>
[ ] 11:01:34.781 [restartedMain] INFO o.f.c.i.c.DbMigrate - [info,49] - Migrating schema `sf_mall` to version 20220527.1 - create user
[ ] 11:01:34.938 [restartedMain] INFO o.f.c.i.c.DbMigrate - [info,49] - Migrating schema `sf_mall` to version 20220527.2 - insert user
[ ] 11:01:34.984 [restartedMain] INFO o.f.c.i.c.DbMigrate - [info,49] - Successfully applied 2 migrations to schema `sf_mall` (execution time 00:00.231s)
[ ] 11:01:35.160 [restartedMain] INFO o.a.c.h.Http11NioProtocol - [log,173] - Starting ProtocolHandler ["http-nio-8088"]
[ ] 11:01:35.191 [restartedMain] INFO com.sf.App - [onApplicationEvent,21] - App is started ok.
[ ] 11:01:35.191 [restartedMain] INFO com.sf.App - [logStarted,61] - Started App in 14.064 seconds (JVM running for 19.09)
```

Database

2.再次执行

```
[ ] 11:10:04.081 [restartedMain] INFO o.f.c.i.l.VersionPrinter - [info,49] - Flyway Community Edition 6.1.0 by Redgate
[ ] 11:10:04.703 [restartedMain] INFO c.a.d.p.DruidDataSource - [init,998] - {dataSource-1} initied
[ ] 11:10:04.944 [restartedMain] INFO o.f.c.i.d.DatabaseFactory - [info,49] - Database: jdbc:mysql://10.207.0.164:3307/sf_mall (MySQL 5.7)
[ ] 11:10:05.007 [restartedMain] INFO o.f.c.i.c.DbValidate - [info,49] - Successfully validated 2 migrations (execution time 00:00.023s)
[ ] 11:10:05.007 [restartedMain] INFO o.f.c.i.c.DbMigrate - [info,49] - Current version of schema `sf_mall`: 20220527.2
[ ] 11:10:05.007 [restartedMain] INFO o.f.c.i.c.DbMigrate - [info,49] - Schema `sf_mall` is up to date. No migration necessary.
[ ] 11:10:05.085 [restartedMain] INFO o.a.c.h.Http11NioProtocol - [log,173] - Starting ProtocolHandler ["http-nio-8088"]
[ ] 11:10:05.116 [restartedMain] INFO com.sf.App - [onApplicationEvent,21] - App is started ok.
[ ] 11:10:05.132 [restartedMain] INFO com.sf.App - [logStarted,61] - Started App in 9.868 seconds (JVM running for 12.399)
Disconnected from the target VM, address: '127.0.0.1:51938', transport: 'socket'
[ ] 11:10:44.483 [SpringContextShutdownHook] INFO c.a.d.p.DruidDataSource - [close,2071] - {dataSource-1} closing ...
[ ] 11:10:44.483 [SpringContextShutdownHook] INFO c.a.d.p.DruidDataSource - [close,2144] - {dataSource-1} closed
```

无更新

需要删除 flyway_schema_history 表里的记录，才可以再次执行 V_ 开头的文件

3.R脚本第一次执行

Project		SQL R_insert_order.sql
util		
App		
resources		
db.migration		
R_insert_order.sql		
U1.0.0__redo_order.sql		
V1.0.0__create_order.sql		
V20220527_1__create_user.sql		
V20220527_2__insert_user.sql		
mapper		
static		
templates		

DB Connections		DB Connections
1		insert into t_order(order_no, amount, count, addr)
2		values
3		('202205270001', 120, 1, '深圳市南山区高新园104栋A座603'),
4		('202205270002', 50, 1, '上海市浦东区圣约华庭C座705'),
5		('202205270003', 66, 3, '珠海市珠江区格力工业园A1门保卫科室');
6		

Database		console	t_order	flyway_schema_history_zero	t_user						
[10.207.0.164 - self] 2 of 5		WHERE ORDER BY									
information_schema											
sf_mall											
tables 3											
flyway_schema_history_zero											
t_order											
t_user											
Server Objects											
		installed_rank	version	description	type	script	checksum	installed_by	installed_on	execution_time	success
1		1	1.0.0	create order	SQL	V1.0.0__create_order.sql	945553215	root	2022-05-27 04:18:19	114	1
2		2	20220527.1	create user	SQL	V20220527_1__create_user.sql	135699064	root	2022-05-27 04:18:19	133	1
3		3	20220527.2	insert user	SQL	V20220527_2__insert_user.sql	639839965	root	2022-05-27 04:18:20	14	1
4		5	<null>	insert order	SQL	R__insert_order.sql	1011766812	root	2022-05-27 05:01:47	6	1

4.R脚本重复执行（不修改内容）

无变化

4 rows

<

	id	order_no	amount	count	addr	phone	create_date
1	1	202205270001	120	1	深圳市南山区高新园104栋A座603	<null>	2022-05-27 05:01:47
2	2	202205270002	50	1	上海市浦东区圣约华庭C座705	<null>	2022-05-27 05:01:47
3	3	202205270003	66	3	珠海市珠江区格力工业园A1门保卫科室2	<null>	2022-05-27 05:01:47

5.R脚本重复执行（修改内容）

- util
- App
- resources
 - db.migration
 - R_insert_order.sql
 - U1.0.0__redo_order.sql
 - V1.0.0__create_order.sql
 - V20220527_1__create_user.sql
 - V20220527_2__insert_user.sql
 - mapper
 - static

DB Connections

```

1 insert into t_order(order_no, amount, count, addr)
2 values
3 ('202205270001', 120, 1, '深圳市南山区高新园104栋A座603'),
4 ('202205270002', 50, 1, '上海市浦东区圣约华庭C座705'),
5 ('202205270003', 66, 3, '珠海市珠江区格力工业园A1门保卫科室2'),
6 ('202205270004', 1000, 1, '广州市番禺区星海学院');
7

```

新增一条。修改一条

	installed_rank	version	description	type	script	checksum	installed_by	installed_on	execution_time	success
1	1	1.0.0	create order	SQL	V1.0.0__create_order.sql	945553215	root	2022-05-27 04:18:19	114	1
2	2	20220527.1	create user	SQL	V20220527_1__create_user.sql	135699064	root	2022-05-27 04:18:19	133	1
3	3	20220527.2	insert user	SQL	V20220527_2__insert_user.sql	639839965	root	2022-05-27 04:18:20	14	1
4	5	<null>	insert order	SQL	R__insert_order.sql	1011766812	root	2022-05-27 05:01:47	6	1
5	6	<null>	insert order	SQL	R__insert_order.sql	-2073399762	root	2022-05-27 05:08:23	10	1
6	7	<null>	insert order	SQL	R__insert_order.sql	-1857080679	root	2022-05-27 05:08:45	6	1

	id	order_no	amount	count	addr	phone	create_date
1	1	202205270001	120	1	深圳市南山区高新园104栋A座603	<null>	2022-05-27 05:01:47
2	2	202205270002	50	1	上海市浦东区圣约华庭C座705	<null>	2022-05-27 05:01:47
3	3	202205270003	66	3	珠海市珠江区格力工业园A1门保卫科室	<null>	2022-05-27 05:01:47
4	4	202205270001	120	1	深圳市南山区高新园104栋A座603	<null>	2022-05-27 05:08:23
5	5	202205270002	50	1	上海市浦东区圣约华庭C座705	<null>	2022-05-27 05:08:23
6	6	202205270003	66	3	珠海市珠江区格力工业园A1门保卫科室	<null>	2022-05-27 05:08:23
7	7	202205270004	1000	1	广州市番禺区星海学院	<null>	2022-05-27 05:08:23
8	8	202205270001	120	1	深圳市南山区高新园104栋A座603	<null>	2022-05-27 05:08:45
9	9	202205270002	50	1	上海市浦东区圣约华庭C座705	<null>	2022-05-27 05:08:45
10	10	202205270003	66	3	珠海市珠江区格力工业园A1门保卫科室2	<null>	2022-05-27 05:08:45
11	11	202205270004	1000	1	广州市番禺区星海学院	<null>	2022-05-27 05:08:45

6.R脚本重复执行（修改位置，脚本内容未变）

SQL R_insert_order.sql

DB Connections

```

1 insert into t_order(order_no, amount, count, addr) values
2 ('202205270001', 120, 1, '深圳市南山区高新园104栋A座603'),
3 ('202205270002', 50, 1, '上海市浦东区圣约华庭C座705'),
4 ('202205270003', 66, 3, '珠海市珠江区格力工业园A1门保卫科室2'),
5 ('202205270004', 1000, 1, '广州市番禺区星海学院');
6

```

WHERE

ORDER BY

	installed_rank	version	description	type	script	checksum	installed_by	installed_on	execution_time	success
1	1	1.0.0	create order	SQL	V1.0.0__create_order.sql	945553215	root	2022-05-27 04:18:19	114	1
2	2	20220527.1	create user	SQL	V20220527_1__create_user.sql	135699064	root	2022-05-27 04:18:19	133	1
3	3	20220527.2	insert user	SQL	V20220527_2__insert_user.sql	639839965	root	2022-05-27 04:18:20	14	1
4	5	<null>	insert order	SQL	R__insert_order.sql	1011766812	root	2022-05-27 05:01:47	6	1
5	6	<null>	insert order	SQL	R__insert_order.sql	-2073399762	root	2022-05-27 05:08:23	10	1
6	7	<null>	insert order	SQL	R__insert_order.sql	-1857080679	root	2022-05-27 05:08:45	6	1
7	8	<null>	insert order	SQL	R__insert_order.sql	909179804	root	2022-05-27 05:12:41	9	1

console

t_order

flyway_sc...story_zero

t_user

15 rows

Tx Auto

DDL

WHERE

ORDER BY

	id	order_no	amount	count	addr	phone	create_date
1	1	202205270001	120	1	深圳市南山区高新园104栋A座603	<null>	2022-05-27 05:01:47
2	2	202205270002	50	1	上海市浦东区圣约华庭C座705	<null>	2022-05-27 05:01:47
3	3	202205270003	66	3	珠海市珠江区格力工业园A1门保卫科室	<null>	2022-05-27 05:01:47
4	4	202205270001	120	1	深圳市南山区高新园104栋A座603	<null>	2022-05-27 05:08:23
5	5	202205270002	50	1	上海市浦东区圣约华庭C座705	<null>	2022-05-27 05:08:23
6	6	202205270003	66	3	珠海市珠江区格力工业园A1门保卫科室	<null>	2022-05-27 05:08:23
7	7	202205270004	1000	1	广州市番禺区星海学院	<null>	2022-05-27 05:08:23
8	8	202205270001	120	1	深圳市南山区高新园104栋A座603	<null>	2022-05-27 05:08:45
9	9	202205270002	50	1	上海市浦东区圣约华庭C座705	<null>	2022-05-27 05:08:45
10	10	202205270003	66	3	珠海市珠江区格力工业园A1门保卫科室2	<null>	2022-05-27 05:08:45
11	11	202205270004	1000	1	广州市番禺区星海学院	<null>	2022-05-27 05:08:45
12	12	202205270001	120	1	深圳市南山区高新园104栋A座603	<null>	2022-05-27 05:12:41
13	13	202205270002	50	1	上海市浦东区圣约华庭C座705	<null>	2022-05-27 05:12:41
14	14	202205270003	66	3	珠海市珠江区格力工业园A1门保卫科室2	<null>	2022-05-27 05:12:41
15	15	202205270004	1000	1	广州市番禺区星海学院	<null>	2022-05-27 05:12:41

又执行了一次

六、最佳实践

- 1

生产务必禁 `spring.flyway.cleanDisabled=false` 。
- 2

尽量避免使用 `Undo` 模式。
- 3

开发版本号尽量根据团队来进行多层次的命名避免混乱。比如 `V1.0.1__ProjectName_{Feature|fix}_Developer_Description.sql`，这种命名包含开发者和相关功能的信息。
- 4

`spring.flyway.outOfOrder` 取值 生产上使用 `false`，开发中使用 `true`。
- 5

多个系统公用一个 数据库 `schema` 时配置`spring.flyway.table` 为不同的系统设置不同的 `metadata` 表名而不使用缺省值 `flyway_schema_hi`
- ## 异常：
- 1.nested exception is org.flywaydb.core.api.FlywayException: Found more than one migration with version 20220527
- 发现版本为20220527的多个迁移：V20220527__create_user.sql V20220527__insert_user.sql
- 修改为：V20220527_1__create_user.sql V20220527_2__insert_user.sql
- Project

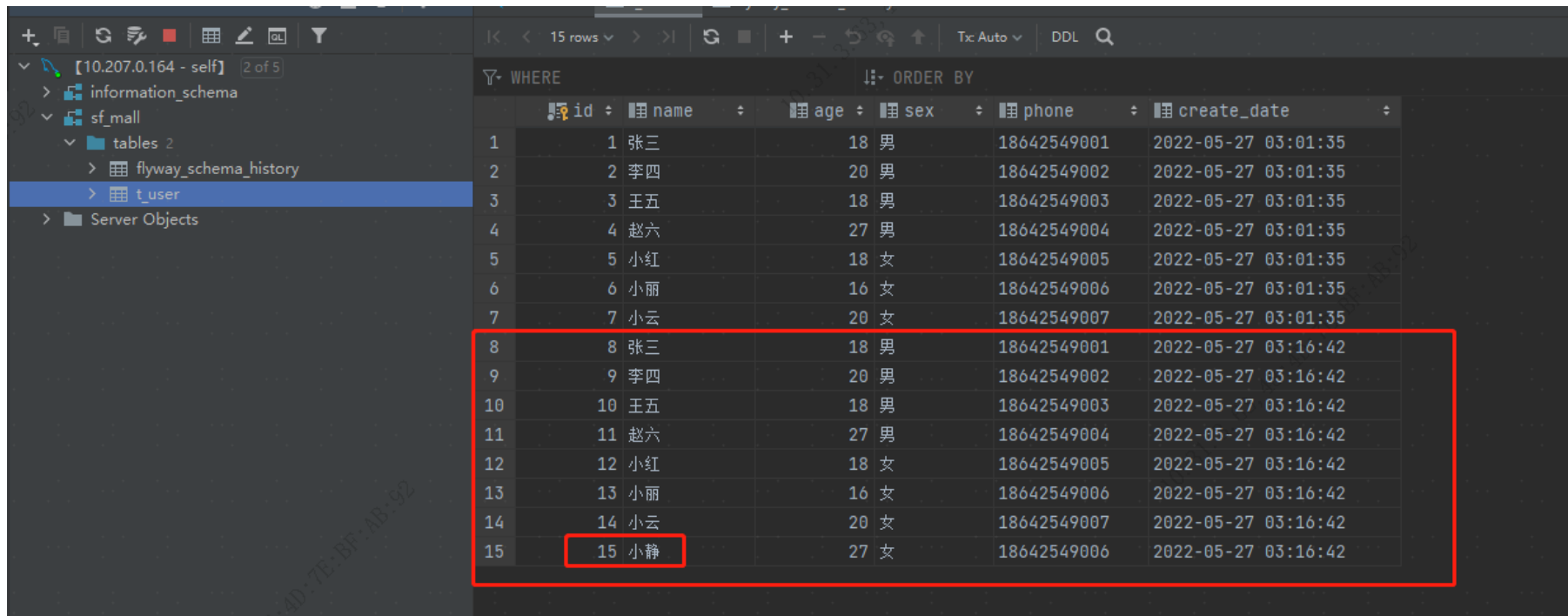
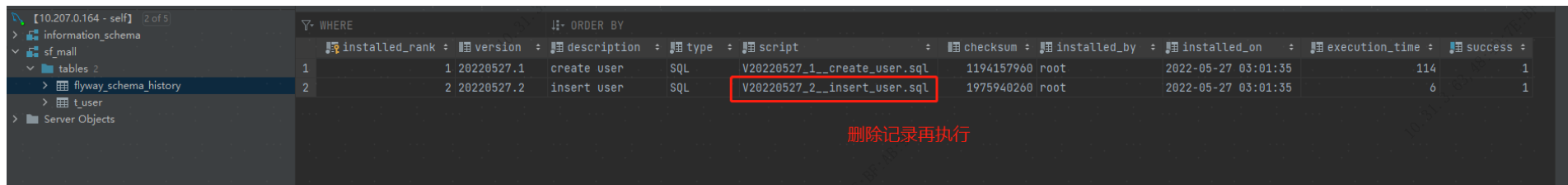
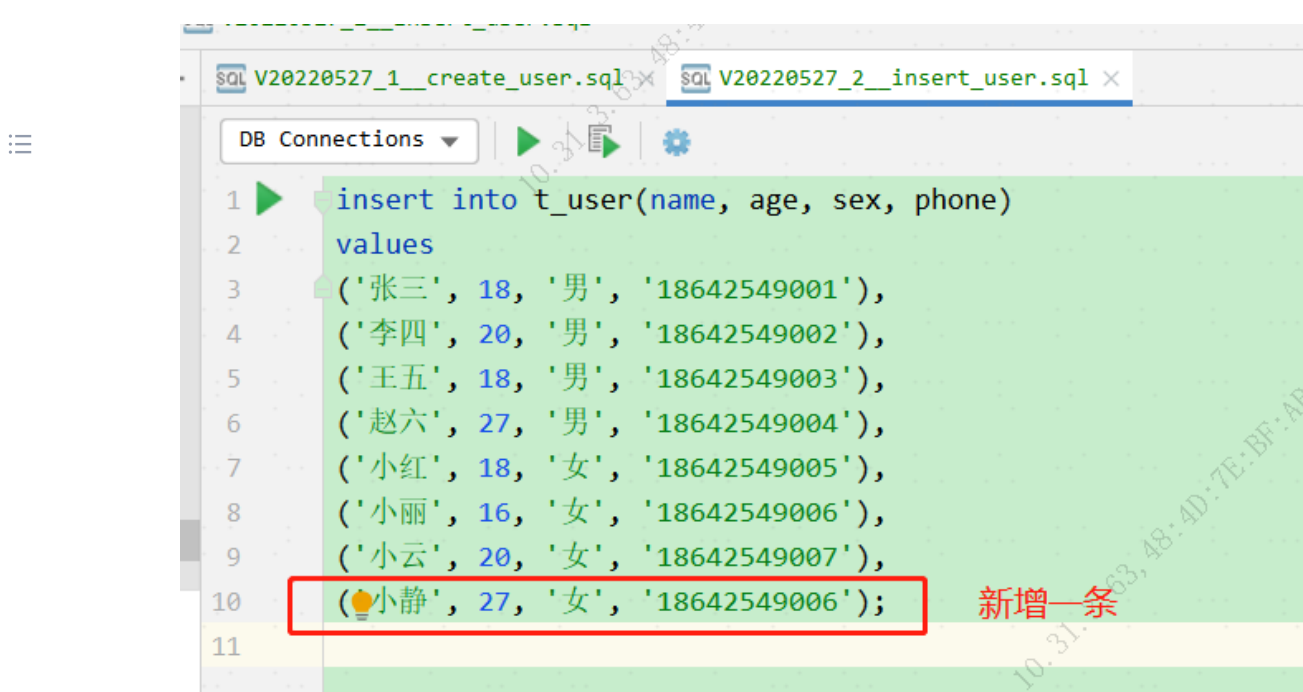
zero_test

DB Connections

SQL V20220527_1__create_user.sql

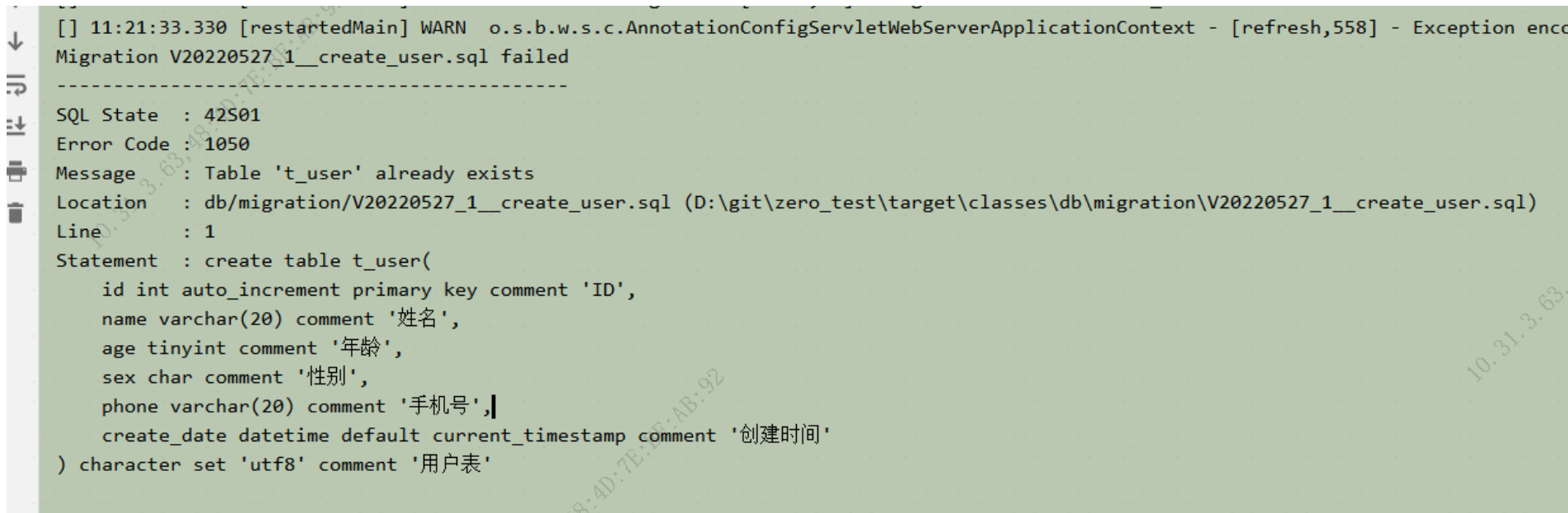
SQL V20220527_2__insert_user.sql

1	create table t_user(
2	id int auto_increment primary key comment 'ID',
3	name varchar(20) comment '姓名',
4	age tinyint comment '年龄',
5	sex char comment '性别',
6	phone varchar(20) comment '手机号',
7	create_date datetime default current_timestamp comment '创建时间'
8) character set 'utf8' comment '用户表';
9	
- 2.nested exception is org.flywaydb.core.api.FlywayException: Validate failed:
- 校验失败Migration checksum mismatch for migration version 20220527.2



3.Message : Table 't_user' already exists

先在库里建表，再执行异常。执行建表脚本时，不允许已经存在该表



4.Wrong repeatable migration name format: R1.0.0__insert_order.sql (It cannot contain a version and should look like this: R__insert order.sql)

R_脚本命名不能带版本号

5.Flyway Pro Edition or Flyway Enterprise Edition upgrade required: undo is not supported by Flyway Community Edition.

社区版IDEA不支持回滚命令

