一、环境

1.linux版本：CentOS release 6.8 (Final)

2.mysql版本：mysql-5.7.30-1.el6.x86_64.rpm-bundle.tar

3.架构

6台主机，双主master0和master1，分别对master0和master1搭建一主二从架构

master0:192.168.238.153

slave01:192.168.238.154

slave02:192.168.238.155

master1:192.168.238.156

slave11: 192.168.238.157

slave12: 192.168.238.158

二、设置主从

1.slave01:192.168.238.154和slave02:192.168.238.155

```
    change master to master_host='192.168.238.153',master_port=3306,master_user='root',master_password='root',master_log_file='mysql-bin.000009',master_log_pos=2414;
```

2.slave11: 192.168.238.157和slave12: 192.168.238.158

```
    change master to master_host='192.168.238.156',master_port=3306,master_user='root',master_password='root',master_log_file='mysql-bin.000006',master_log_pos=2360;
```

三、数据库表结构

```
CREATE TABLE `c_order`(
 `id` bigint(20) NOT NULL AUTO_INCREMENT,
 `is_del` bit(1) NOT NULL DEFAULT 0 COMMENT '是否被删除',
 `user_id` int(11) NOT NULL COMMENT '用户id',
 `company_id` int(11) NOT NULL COMMENT '公司id',
 `publish_user_id` int(11) NOT NULL COMMENT 'B端用户id',
 `position_id` int(11) NOT NULL COMMENT '职位ID',
 `resume_type` int(2) NOT NULL DEFAULT 0 COMMENT '简历类型：0附件 1在线',
 `status` varchar(256) NOT NULL COMMENT '投递状态 投递状态WAIT-待处理 AUTO_FILTER-自动过滤 PREPARE_CONTACT-待沟通 REFUSE-拒绝 ARRANGE_INTERVIEW-通知面试',
 `create_time` datetime NOT NULL COMMENT '创建时间',
 `update_time` datetime NOT NULL COMMENT '处理时间',
 PRIMARY KEY (`id`),
 KEY `index_userId_positionId` (`user_id`, `position_id`),
 KEY `idx_userId_operateTime` (`user_id`, `update_time`)
) ENGINE = InnoDB DEFAULT CHARSET = utf8mb4;
```

四、springboot项目

1.实体类COrder

```
@Entity
@Table(name = "c_order")
public class COrder implements Serializable {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "id")
    private Long id;
    @Column(name = "is_del")
    private Boolean isDel;
    @Column(name = "user_id")
    private Integer userId;
    @Column(name = "company_id")
    private Integer companyId;
    @Column(name = "publish_user_id")
    private Integer publishUserId;
    @Column(name = "position_id")
    private Integer positionId;
    @Column(name = "resume_type")
    private Integer resumeType;
    @Column(name = "status")
    private String status;
    @Column(name = "create_time")
    private Date createTime;
    @Column(name = "update_time")
    private Date updateTime;
}
```

2.jpa repository类COrderRepository

```
public interface COrderRepository extends JpaRepository<COrder, Long> {
}
```

3.配置文件application.properties

```
spring.profiles.active=sharding-database
spring.shardingsphere.props.sql.show=true
```

4.配置文件application-sharding-master-slaves.properties

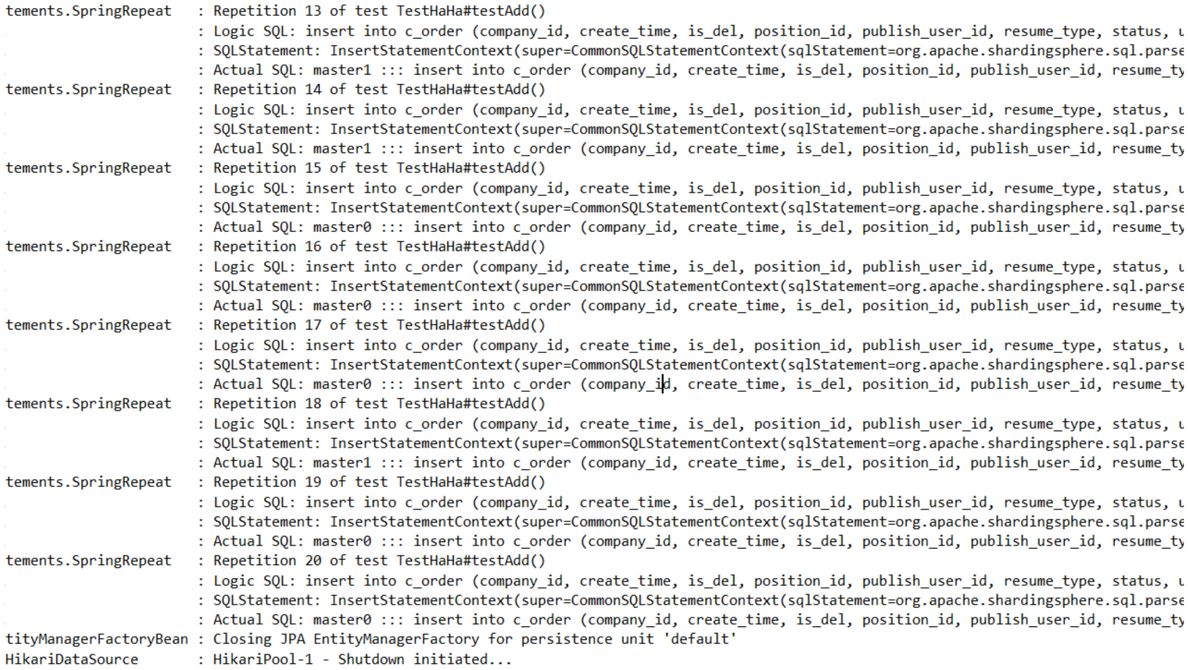
```
#数据源
spring.shardingsphere.datasource.names=master0,slave01,slave02,master1,slave11,slave12

spring.shardingsphere.datasource.master0.type=com.zaxxer.hikari.HikariDataSource
spring.shardingsphere.datasource.master0.driver-class-name=com.mysql.jdbc.Driver
spring.shardingsphere.datasource.master0.jdbc-
url=jdbc:mysql://192.168.238.153:3306/lagou?
useUnicode=true&characterEncoding=utf-8&useSSL=false
spring.shardingsphere.datasource.master0.username=root
spring.shardingsphere.datasource.master0.password=root

spring.shardingsphere.datasource.slave01.type=com.zaxxer.hikari.HikariDataSource
spring.shardingsphere.datasource.slave01.driver-class-name=com.mysql.jdbc.Driver
spring.shardingsphere.datasource.slave01.jdbc-
url=jdbc:mysql://192.168.238.154:3306/lagou?useSSL=false
spring.shardingsphere.datasource.slave01.username=root
spring.shardingsphere.datasource.slave01.password=root

spring.shardingsphere.datasource.slave02.type=com.zaxxer.hikari.HikariDataSource
spring.shardingsphere.datasource.slave02.driver-class-name=com.mysql.jdbc.Driver
spring.shardingsphere.datasource.slave02.jdbc-
url=jdbc:mysql://192.168.238.155:3306/lagou?useSSL=false
spring.shardingsphere.datasource.slave02.username=root
spring.shardingsphere.datasource.slave02.password=root

spring.shardingsphere.datasource.master1.type=com.zaxxer.hikari.HikariDataSource
spring.shardingsphere.datasource.master1.driver-class-name=com.mysql.jdbc.Driver
spring.shardingsphere.datasource.master1.jdbc-
url=jdbc:mysql://192.168.238.156:3306/lagou?
useUnicode=true&characterEncoding=utf-8&useSSL=false
spring.shardingsphere.datasource.master1.username=root
spring.shardingsphere.datasource.master1.password=root

spring.shardingsphere.datasource.slave11.type=com.zaxxer.hikari.HikariDataSource
spring.shardingsphere.datasource.slave11.driver-class-name=com.mysql.jdbc.Driver
spring.shardingsphere.datasource.slave11.jdbc-
url=jdbc:mysql://192.168.238.157:3306/lagou?useSSL=false
spring.shardingsphere.datasource.slave11.username=root
spring.shardingsphere.datasource.slave11.password=root

spring.shardingsphere.datasource.slave12.type=com.zaxxer.hikari.HikariDataSource
spring.shardingsphere.datasource.slave12.driver-class-name=com.mysql.jdbc.Driver
spring.shardingsphere.datasource.slave12.jdbc-
url=jdbc:mysql://192.168.238.158:3306/lagou?useSSL=false
spring.shardingsphere.datasource.slave12.username=root
spring.shardingsphere.datasource.slave12.password=root

#sharding-database-table  基于user_id对c_order表进行数据分片
spring.shardingsphere.sharding.tables.c_order.database-strategy.inline.sharding-
column=user_id
spring.shardingsphere.sharding.tables.c_order.database-
strategy.inline.algorithm-expression=master$->{user_id % 2}
spring.shardingsphere.sharding.tables.c_order.actual-data-nodes=master$->
{0..1}.c_order
spring.shardingsphere.sharding.tables.c_order.key-generator.column=id
spring.shardingsphere.sharding.tables.c_order.key-generator.type=SNOWFLAKE

#读写分离
spring.shardingsphere.sharding.master-slave-rules.master0.master-data-source-
name=master0
spring.shardingsphere.sharding.master-slave-rules.master0.slave-data-source-
names=slave01, slave02
```

```
spring.shardingsphere.sharding.master-slave-rules.master1.master-data-source-name=master1
spring.shardingsphere.sharding.master-slave-rules.master1.slave-data-source-names=slave11, slave12
```

5.测试类

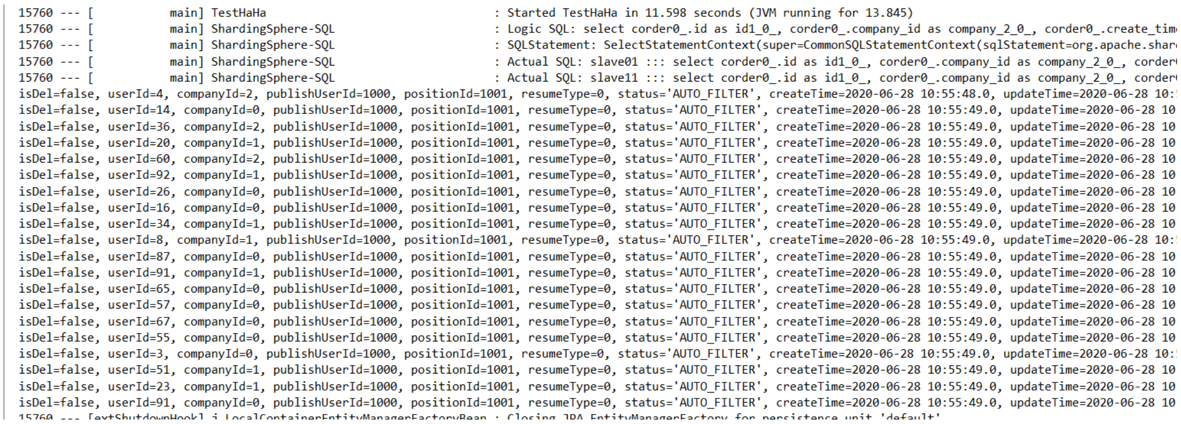
```
@RunWith(SpringRunner.class)
@SpringBootTest(classes = ShardingJdbcApplication.class)
public class TestHaHa {
    @Resource
    private COrderRepository cOrderRepository;

    @Test
    @Repeat(20)
    public void testAdd(){
        Random random = new Random();
        int userId = random.nextInt(100);
        int companyId = random.nextInt(3);
        COrder cOrder = new COrder();
        cOrder.setDel(false);
        cOrder.setUserId(userId);
        cOrder.setCompanyId(companyId);
        cOrder.setPublishUserId(1000);
        cOrder.setPositionId(1001);
        cOrder.setResumeType(0);
        cOrder.setStatus("AUTO_FILTER");
        cOrder.setCreateTime(new Date());
        cOrder.setUpdateTime(new Date());
        cOrderRepository.save(cOrder);
    }

    @Test
    public void testFind(){
        List<COrder> cOrders = cOrderRepository.findAll();
        cOrders.forEach(cOrder->{
            System.out.println(cOrder);
        });
    }
}
```

5-1.testAdd测试结果



```
tements.SpringRepeat    : Repetition 13 of test TestHaHa#testAdd()
                        : Logic SQL: insert into c_order (company_id, create_time, is_del, position_id, publish_user_id, resume_type, status, u
                        : SQLStatement: InsertStatementContext(super=CommonSQLStatementContext(sqlStatement=org.apache.shardingsphere.sql.parse
                        : Actual SQL: master1 ::: insert into c_order (company_id, create_time, is_del, position_id, publish_user_id, resume_ty
tements.SpringRepeat    : Repetition 14 of test TestHaHa#testAdd()
                        : Logic SQL: insert into c_order (company_id, create_time, is_del, position_id, publish_user_id, resume_type, status, u
                        : SQLStatement: InsertStatementContext(super=CommonSQLStatementContext(sqlStatement=org.apache.shardingsphere.sql.parse
                        : Actual SQL: master1 ::: insert into c_order (company_id, create_time, is_del, position_id, publish_user_id, resume_ty
tements.SpringRepeat    : Repetition 15 of test TestHaHa#testAdd()
                        : Logic SQL: insert into c_order (company_id, create_time, is_del, position_id, publish_user_id, resume_type, status, u
                        : SQLStatement: InsertStatementContext(super=CommonSQLStatementContext(sqlStatement=org.apache.shardingsphere.sql.parse
                        : Actual SQL: master0 ::: insert into c_order (company_id, create_time, is_del, position_id, publish_user_id, resume_ty
tements.SpringRepeat    : Repetition 16 of test TestHaHa#testAdd()
                        : Logic SQL: insert into c_order (company_id, create_time, is_del, position_id, publish_user_id, resume_type, status, u
                        : SQLStatement: InsertStatementContext(super=CommonSQLStatementContext(sqlStatement=org.apache.shardingsphere.sql.parse
                        : Actual SQL: master0 ::: insert into c_order (company_id, create_time, is_del, position_id, publish_user_id, resume_ty
tements.SpringRepeat    : Repetition 17 of test TestHaHa#testAdd()
                        : Logic SQL: insert into c_order (company_id, create_time, is_del, position_id, publish_user_id, resume_type, status, u
                        : SQLStatement: InsertStatementContext(super=CommonSQLStatementContext(sqlStatement=org.apache.shardingsphere.sql.parse
                        : Actual SQL: master0 ::: insert into c_order (company_id, create_time, is_del, position_id, publish_user_id, resume_ty
tements.SpringRepeat    : Repetition 18 of test TestHaHa#testAdd()
                        : Logic SQL: insert into c_order (company_id, create_time, is_del, position_id, publish_user_id, resume_type, status, u
                        : SQLStatement: InsertStatementContext(super=CommonSQLStatementContext(sqlStatement=org.apache.shardingsphere.sql.parse
                        : Actual SQL: master1 ::: insert into c_order (company_id, create_time, is_del, position_id, publish_user_id, resume_ty
tements.SpringRepeat    : Repetition 19 of test TestHaHa#testAdd()
                        : Logic SQL: insert into c_order (company_id, create_time, is_del, position_id, publish_user_id, resume_type, status, u
                        : SQLStatement: InsertStatementContext(super=CommonSQLStatementContext(sqlStatement=org.apache.shardingsphere.sql.parse
                        : Actual SQL: master0 ::: insert into c_order (company_id, create_time, is_del, position_id, publish_user_id, resume_ty
tements.SpringRepeat    : Repetition 20 of test TestHaHa#testAdd()
                        : Logic SQL: insert into c_order (company_id, create_time, is_del, position_id, publish_user_id, resume_type, status, u
                        : SQLStatement: InsertStatementContext(super=CommonSQLStatementContext(sqlStatement=org.apache.shardingsphere.sql.parse
                        : Actual SQL: master0 ::: insert into c_order (company_id, create_time, is_del, position_id, publish_user_id, resume_ty
tityManagerFactoryBean : Closing JPA EntityManagerFactory for persistence unit 'default'
HikariDataSource        : HikariPool-1 - Shutdown initiated...
```

2.testFind测试结果



```
15760 --- [           main] TestHaHa                                 : Started TestHaHa in 11.598 seconds (JVM running for 13.845)
15760 --- [           main] ShardingSphere-SQL                       : Logic SQL: select corder0_.id as id1_0_, corder0_.company_id as company_2_0_, corder0_.create_tim
15760 --- [           main] ShardingSphere-SQL                       : SQLStatement: SelectStatementContext(super=CommonSQLStatementContext(sqlStatement=org.apache.shar
15760 --- [           main] ShardingSphere-SQL                       : Actual SQL: slave01 ::: select corder0_.id as id1_0_, corder0_.company_id as company_2_0_, corder
15760 --- [           main] ShardingSphere-SQL                       : Actual SQL: slave11 ::: select corder0_.id as id1_0_, corder0_.company_id as company_2_0_, corder
isDel=false, userId=4, companyId=2, publishUserId=1000, positionId=1001, resumeType=0, status='AUTO_FILTER', createTime=2020-06-28 10:55:48.0, updateTime=2020-06-28 10:
isDel=false, userId=14, companyId=0, publishUserId=1000, positionId=1001, resumeType=0, status='AUTO_FILTER', createTime=2020-06-28 10:55:49.0, updateTime=2020-06-28 10
isDel=false, userId=36, companyId=2, publishUserId=1000, positionId=1001, resumeType=0, status='AUTO_FILTER', createTime=2020-06-28 10:55:49.0, updateTime=2020-06-28 10
isDel=false, userId=20, companyId=1, publishUserId=1000, positionId=1001, resumeType=0, status='AUTO_FILTER', createTime=2020-06-28 10:55:49.0, updateTime=2020-06-28 10
isDel=false, userId=60, companyId=2, publishUserId=1000, positionId=1001, resumeType=0, status='AUTO_FILTER', createTime=2020-06-28 10:55:49.0, updateTime=2020-06-28 10
isDel=false, userId=92, companyId=1, publishUserId=1000, positionId=1001, resumeType=0, status='AUTO_FILTER', createTime=2020-06-28 10:55:49.0, updateTime=2020-06-28 10
isDel=false, userId=26, companyId=0, publishUserId=1000, positionId=1001, resumeType=0, status='AUTO_FILTER', createTime=2020-06-28 10:55:49.0, updateTime=2020-06-28 10
isDel=false, userId=16, companyId=0, publishUserId=1000, positionId=1001, resumeType=0, status='AUTO_FILTER', createTime=2020-06-28 10:55:49.0, updateTime=2020-06-28 10
isDel=false, userId=34, companyId=1, publishUserId=1000, positionId=1001, resumeType=0, status='AUTO_FILTER', createTime=2020-06-28 10:55:49.0, updateTime=2020-06-28 10
isDel=false, userId=8, companyId=1, publishUserId=1000, positionId=1001, resumeType=0, status='AUTO_FILTER', createTime=2020-06-28 10:55:49.0, updateTime=2020-06-28 10:
isDel=false, userId=87, companyId=0, publishUserId=1000, positionId=1001, resumeType=0, status='AUTO_FILTER', createTime=2020-06-28 10:55:49.0, updateTime=2020-06-28 10
isDel=false, userId=91, companyId=1, publishUserId=1000, positionId=1001, resumeType=0, status='AUTO_FILTER', createTime=2020-06-28 10:55:49.0, updateTime=2020-06-28 10
isDel=false, userId=65, companyId=0, publishUserId=1000, positionId=1001, resumeType=0, status='AUTO_FILTER', createTime=2020-06-28 10:55:49.0, updateTime=2020-06-28 10
isDel=false, userId=57, companyId=0, publishUserId=1000, positionId=1001, resumeType=0, status='AUTO_FILTER', createTime=2020-06-28 10:55:49.0, updateTime=2020-06-28 10
isDel=false, userId=67, companyId=0, publishUserId=1000, positionId=1001, resumeType=0, status='AUTO_FILTER', createTime=2020-06-28 10:55:49.0, updateTime=2020-06-28 10
isDel=false, userId=55, companyId=0, publishUserId=1000, positionId=1001, resumeType=0, status='AUTO_FILTER', createTime=2020-06-28 10:55:49.0, updateTime=2020-06-28 10
isDel=false, userId=3, companyId=0, publishUserId=1000, positionId=1001, resumeType=0, status='AUTO_FILTER', createTime=2020-06-28 10:55:49.0, updateTime=2020-06-28 10:
isDel=false, userId=51, companyId=1, publishUserId=1000, positionId=1001, resumeType=0, status='AUTO_FILTER', createTime=2020-06-28 10:55:49.0, updateTime=2020-06-28 10
isDel=false, userId=23, companyId=1, publishUserId=1000, positionId=1001, resumeType=0, status='AUTO_FILTER', createTime=2020-06-28 10:55:49.0, updateTime=2020-06-28 10
isDel=false, userId=91, companyId=0, publishUserId=1000, positionId=1001, resumeType=0, status='AUTO_FILTER', createTime=2020-06-28 10:55:49.0, updateTime=2020-06-28 10
15760 --- [extShutdownHook] j.LocalContainerEntityManagerFactoryBean : Closing JPA EntityManagerFactory for persistence unit 'default'
```