PERCONA LIVE ONLINE MAY 12 - 13th 2021

Hello!

I am Dongchan, Sung

I am a DBA who also develops, and I am here to share my experiences.

- MySQL DBA, @KakaoBank
- https://github.com/gywndi
- https://gywn.net, https://gywn.blog
- https://www.facebook.com/dongchan.sung

Flexible data transfer

uldra-replicator

Instructions

1. Data transfer technique

2. Binary log description and structure

3. Uldra-Replicator

4. Uldra-Replicator use case
   4-1. Data transfer to Shard DB in real time
   4-2. Merge or separate data for delivery.
   4-3. Change contents while passing data

5. Conclusion

1.
Data transfer technique

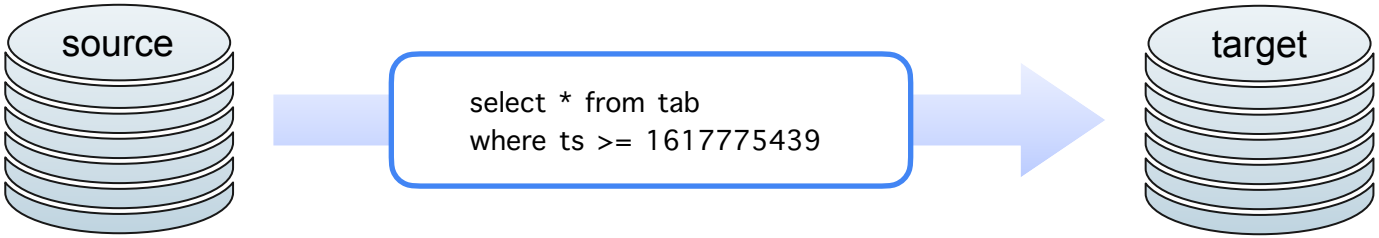
How to transfer data in real time for use in services

Data transfer technique (1/4)

- Real-time migration using SQL
- Real-time migration using CDC

Data transfer technique (2/4)

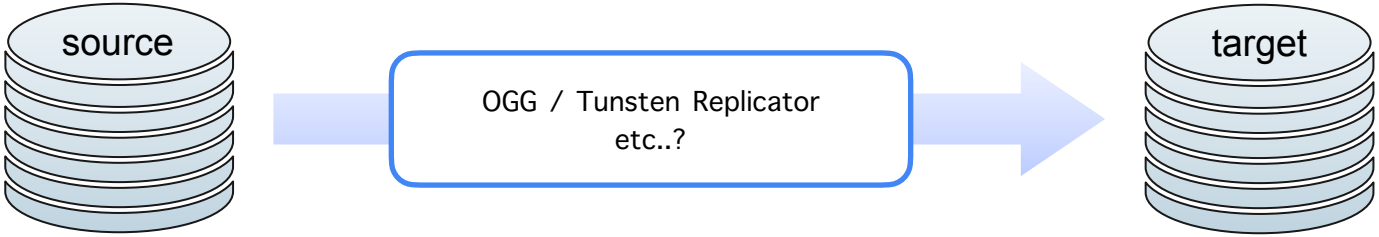
- Real-time migration using SQL



pros.	Easy to control and almost realtime data.
cons.	Possible data loss due to delayed commit

Data transfer technique (3/4)

- Real-time migration using CDC(Changed Data Capture)



pros.	No data loss even from delayed commit
cons.	Replication position management is complicated. Limitation to flexible schema conversion, if CDC not support.

Data transfer technique (4/4)

- Easy to operate and flexible schema conversion plan is required

Idea's been started

uldra-replicator 

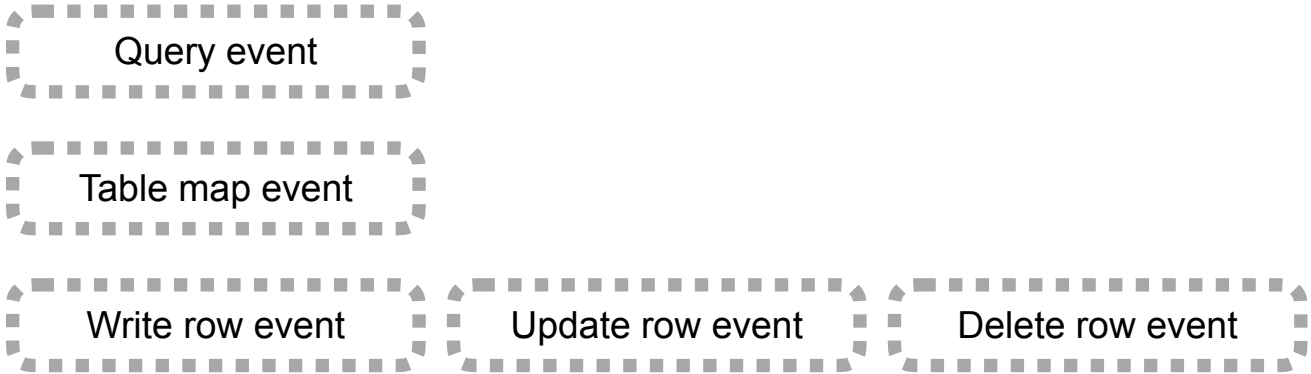
2.

Binary log description and structure

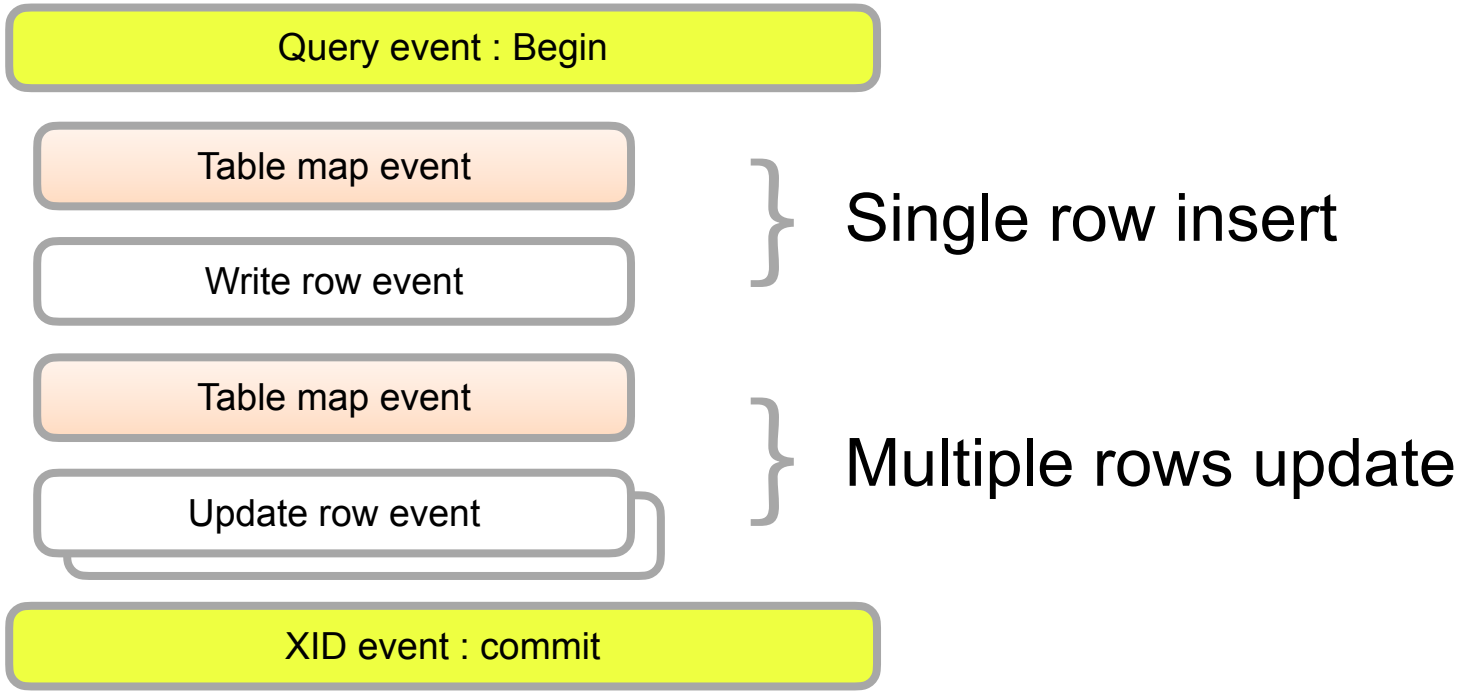
Before talking about Uldra, let's look at the Binlog row format.

Binary log (1/13)

- Logs when a transaction is committed
- Guaranteed transaction order
- Changed data history can be trust



Binary log - row format (2/13)

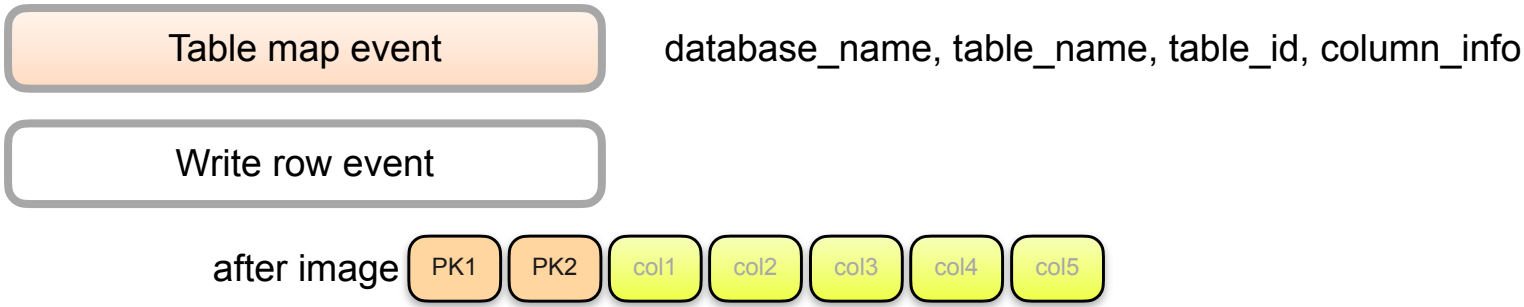


Binary log - table map event (3/13)

```
TableMapEventData{
  tableId=271,
  database='database_name',
  table='table_name',
  columnTypes=15, 15, 15, 15, 15, 15, 3,
  columnMetadata=40, 80, 200, 80, 400, 4, 0,
  columnNullability={1, 2, 3, 4, 5, 6, 7},
  eventMetadata=TableMapEventMetadata{
    signedness={},
    .. skip ..
    enumAndSetColumnCharsets=null
  }
}
```

tableId changed, if schema altered

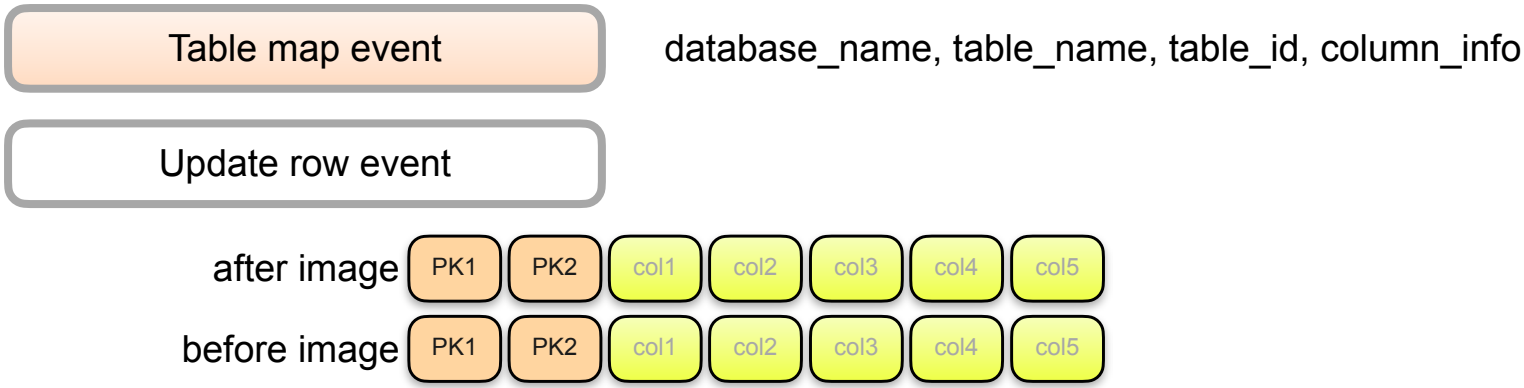
Binary log - write row event (4/13)



Binary log - write row event (5/13)

```
WriteRowsEventData{
  tableId=271,
  includedColumns={0, 1, 2, 3, 4, 5, 6},
  rows=[
    [pk11, pk12, col1, col2, col3, col4, 0],
  ]
}
```

Binary log - update row event (6/13)



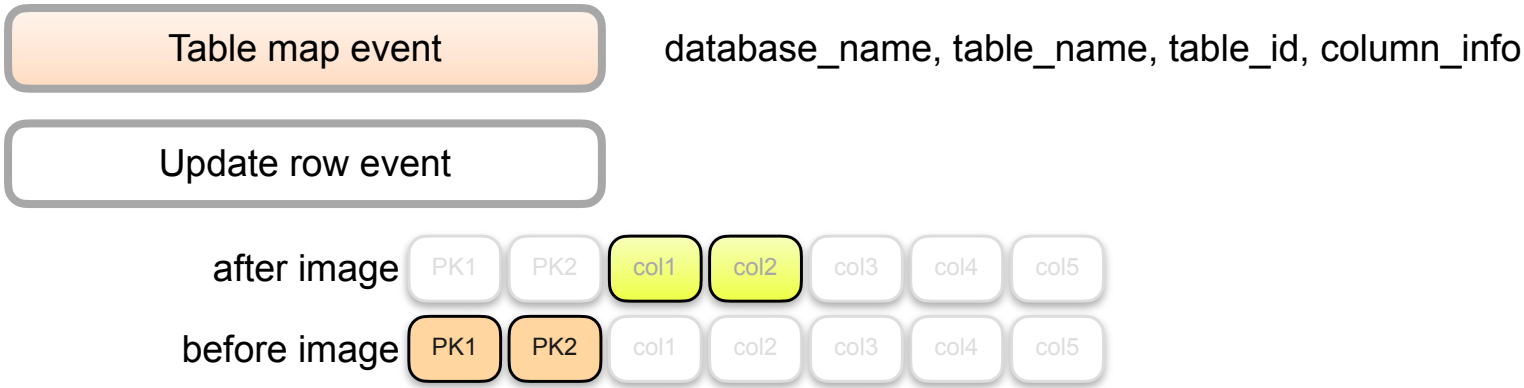
binlog_row_image: full

Binary log - update row event (7/13)

```
UpdateRowsEventData{
  tableId=271,
  includedColumnsBeforeUpdate={0, 1, 2, 3, 4, 5, 6},
  includedColumns={0, 1, 2, 3, 4, 5, 6},
  rows=[
    {before=[pk11, pk12, col1, col2, col3, col4, 0], after=[pk11, pk12, col1, col2, col3, col4_new, 1]},
    {before=[pk21, pk22, col1, col2, col3, col4, 1], after=[pk21, pk22, col1, col2, col3, col4_new, 2]},
    {before=[pk31, pk32, col1, col2, col3, col4, 2], after=[pk31, pk32, col1, col2, col3, col4_new, 3]}
  ]
}
```

binlog_row_image: full

Binary log - update row event (8/13)



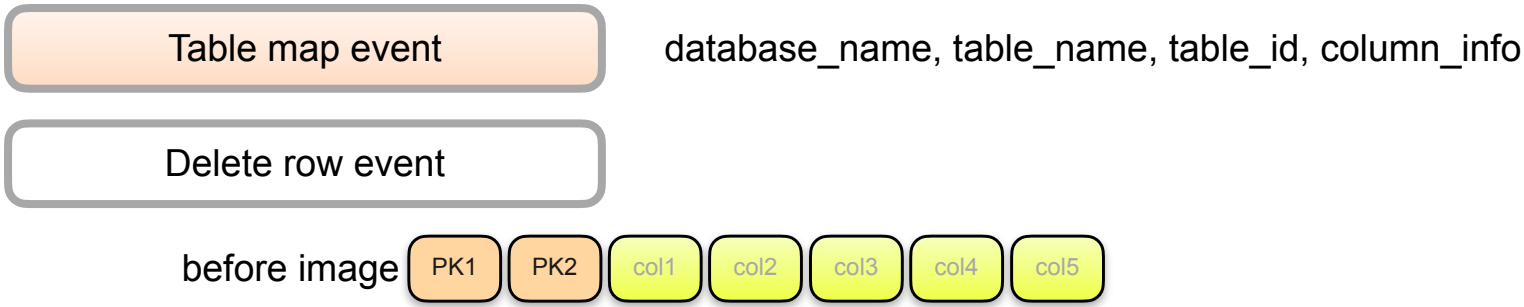
binlog_row_image: minimal

Binary log - update row event (9/13)

```
UpdateRowsEventData{
  tableId=271,
  includedColumnsBeforeUpdate={0, 1},
  includedColumns={6},
  rows=[
    {before=[pk11, pk12], after=[1]},
    {before=[pk21, pk22], after=[2]},
    {before=[pk31, pk32], after=[3]}
  ]
}
```

binlog_row_image: minimal

Binary log - delete row event (10/13)



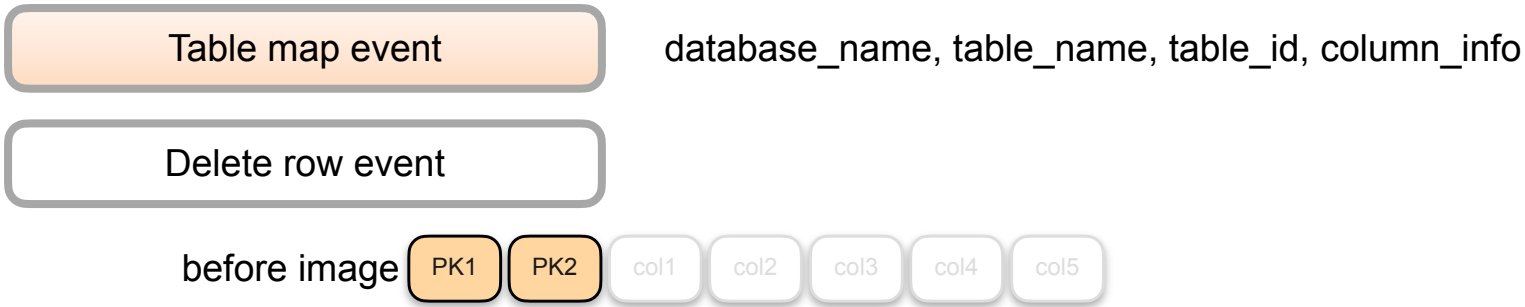
binlog_row_image: full

Binary log - delete row event (11/13)

```
DeleteRowsEventData{
  tableId=271,
  includedColumns={0, 1, 2, 3, 4, 5, 6},
  rows=[
    [pk11, pk12, col1, col2, col3, col4, 0],
  ]
}
```

binlog_row_image: full

Binary log - delete row event (12/13)



binlog_row_image: minimal

Binary log - delete row event (13/13)

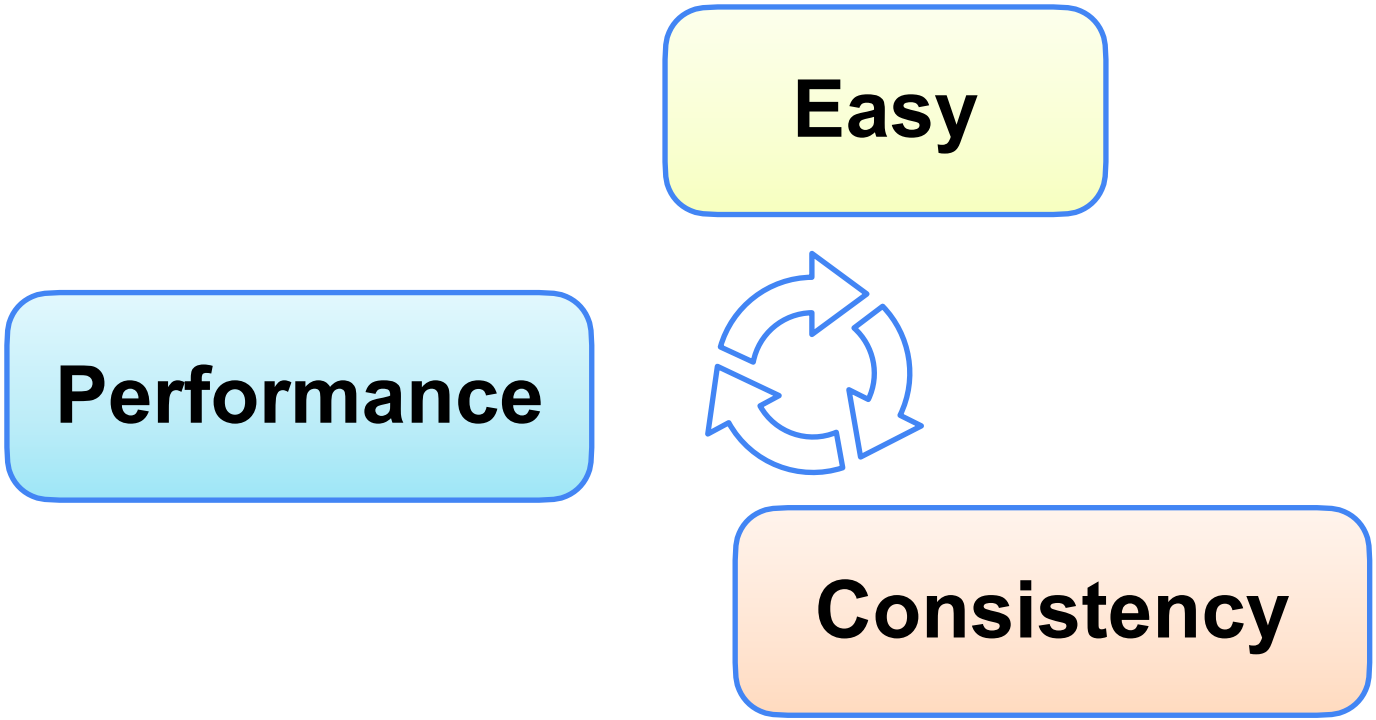
```
DeleteRowsEventData{
  tableId=271,
  includedColumns={0, 1},
  rows=[
    [pk11, pk12],
  ]
}
```

binlog_row_image: minimal

3.
Uldra-Replicator

Let me intruduce flexible data trasfer, uldra-replicator

Uldra-Replicator (1/10)



Uldra-Replicator (2/10)

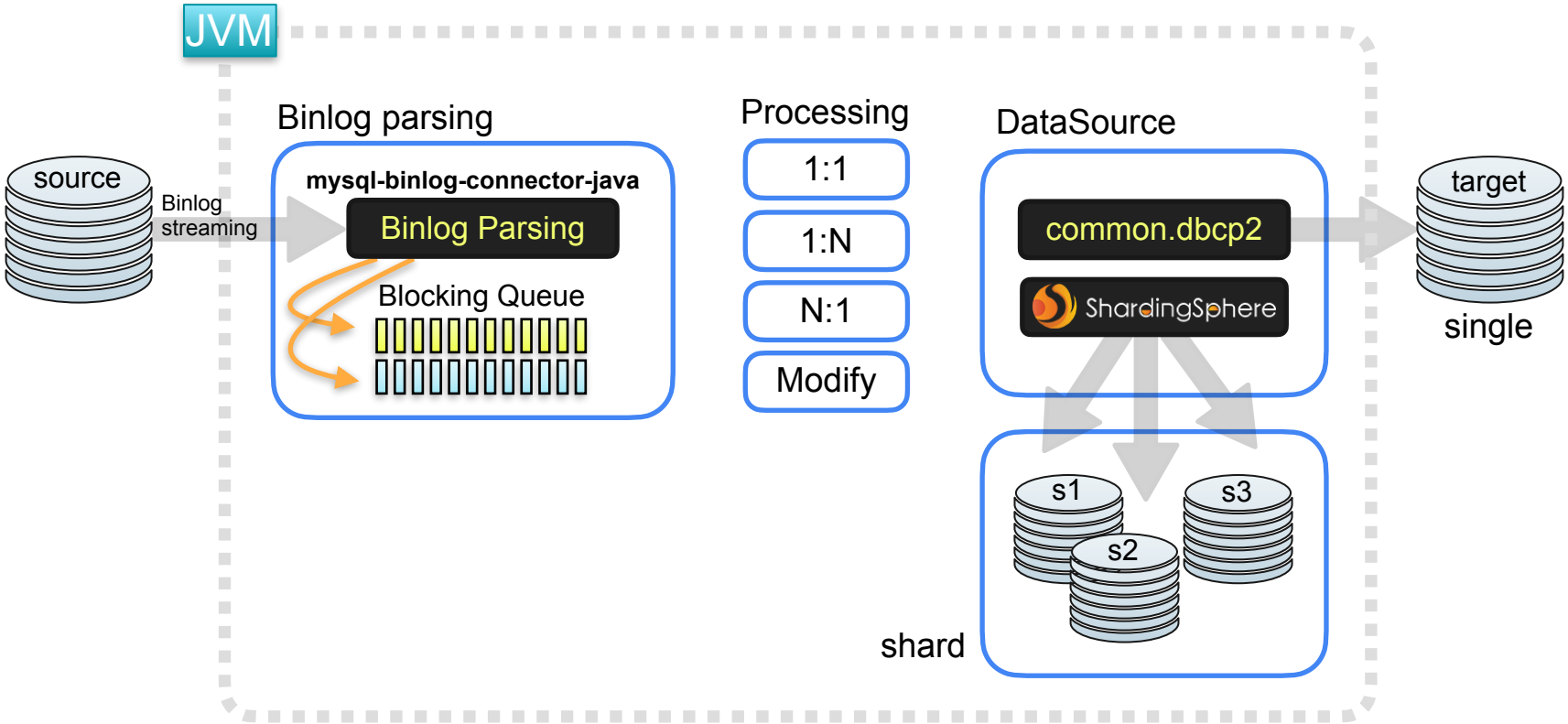
https://shardingsphere.apache.org/

jdk1.8

mysql-binlog-connector-java

https://github.com/shyiko/mysql-binlog-connector-java

Uldra-Replicator (3/10)



Uldra-Replicator (4/10)

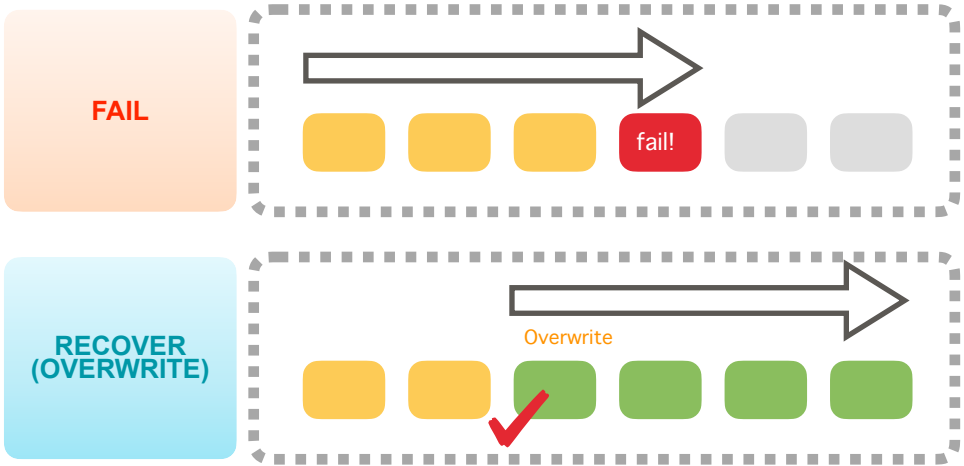
must **ROW format**

Group key

FULL image

3-1. Easy - overwrite (5/10)

Finally, it becomes the correct data.



Loose Replication Position Management

3-1. Easy - overwrite (6/10)

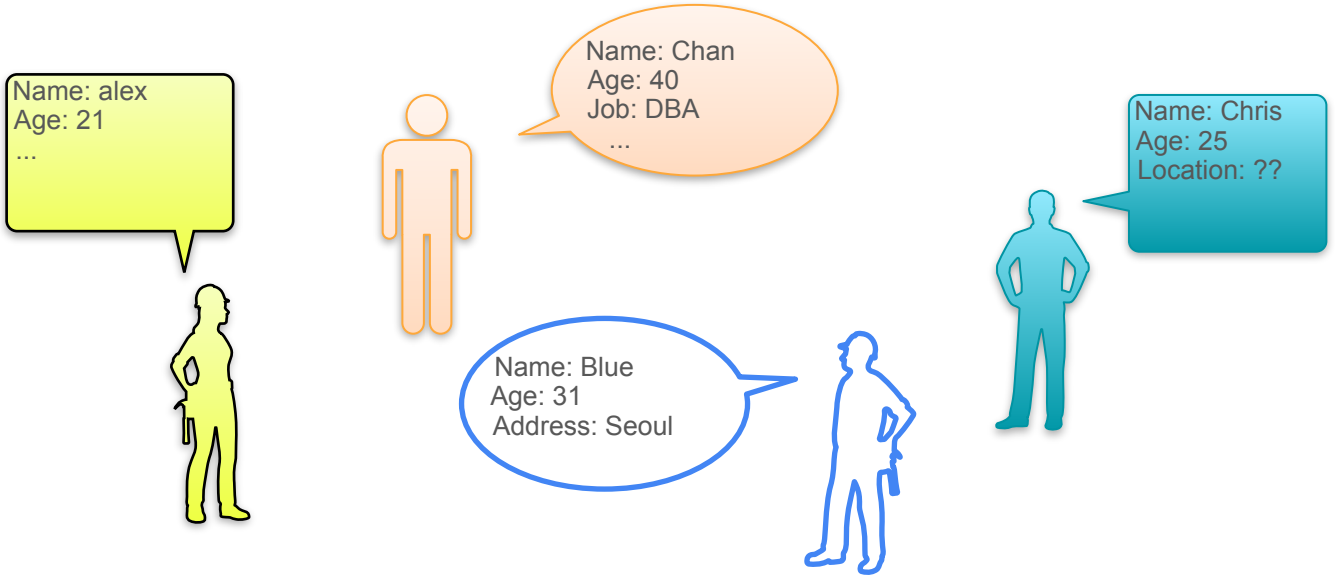
	normal	recover
Write row event	Insert into .. values ..	Insert ignore into .. on duplicate key update ..
Delete row event	delete from ..	delete ignore from ..
Update row event	update .. set ..	update ignore .. set ..

Insert ignore into .. on duplicate key update ..
Delete ignore from ..

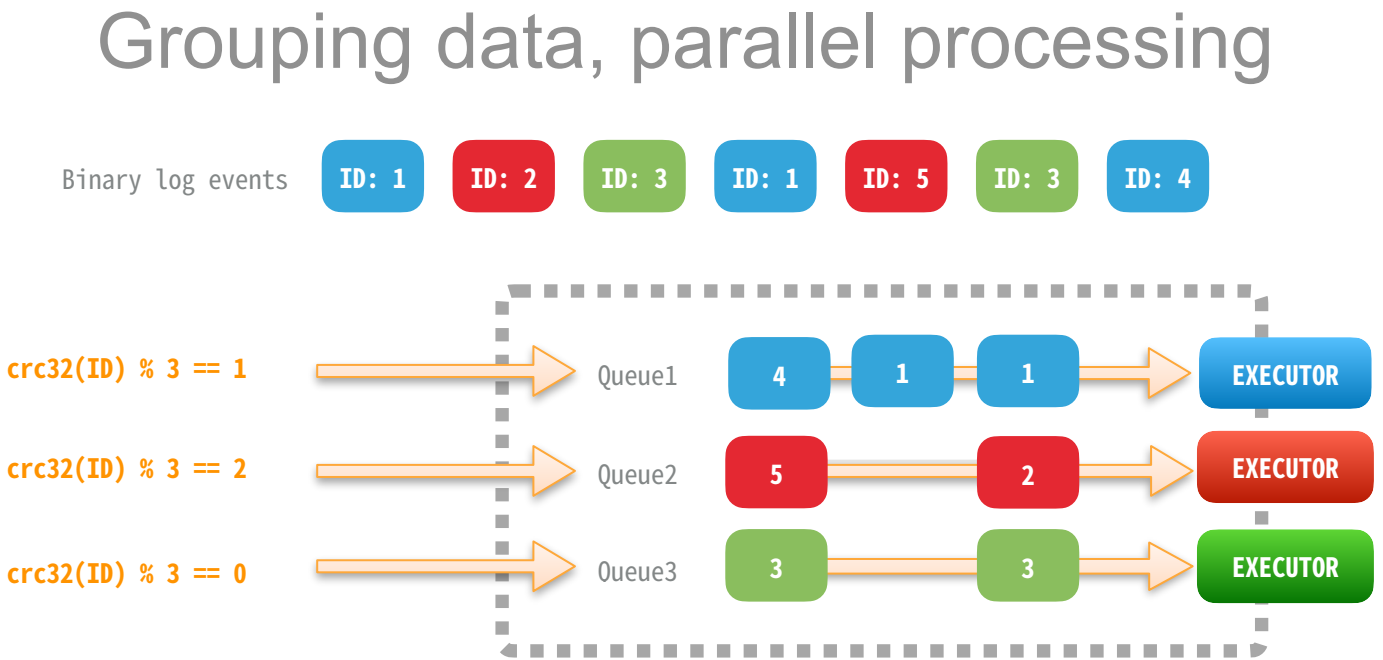
if group key changed

3-2. Performance - parallel processing (7/10)

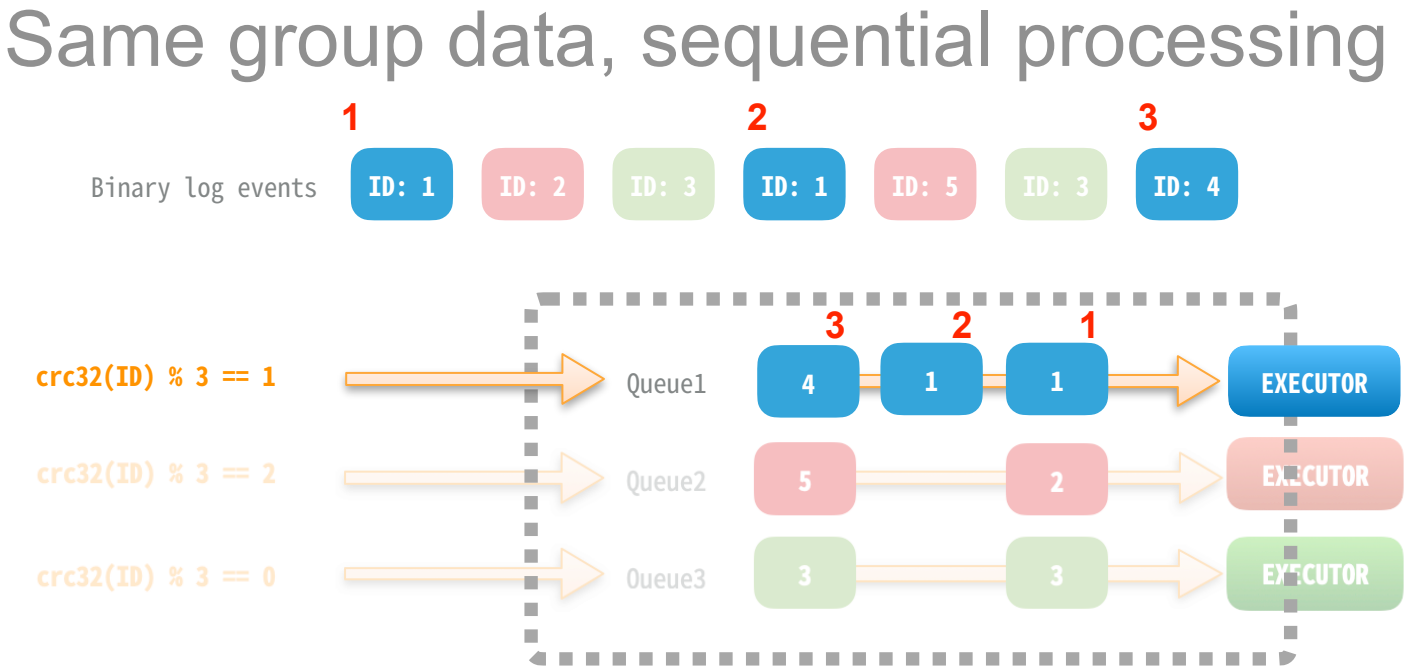
Personal data is isolated from others.



3-2. Performance - parallel processing (8/10)

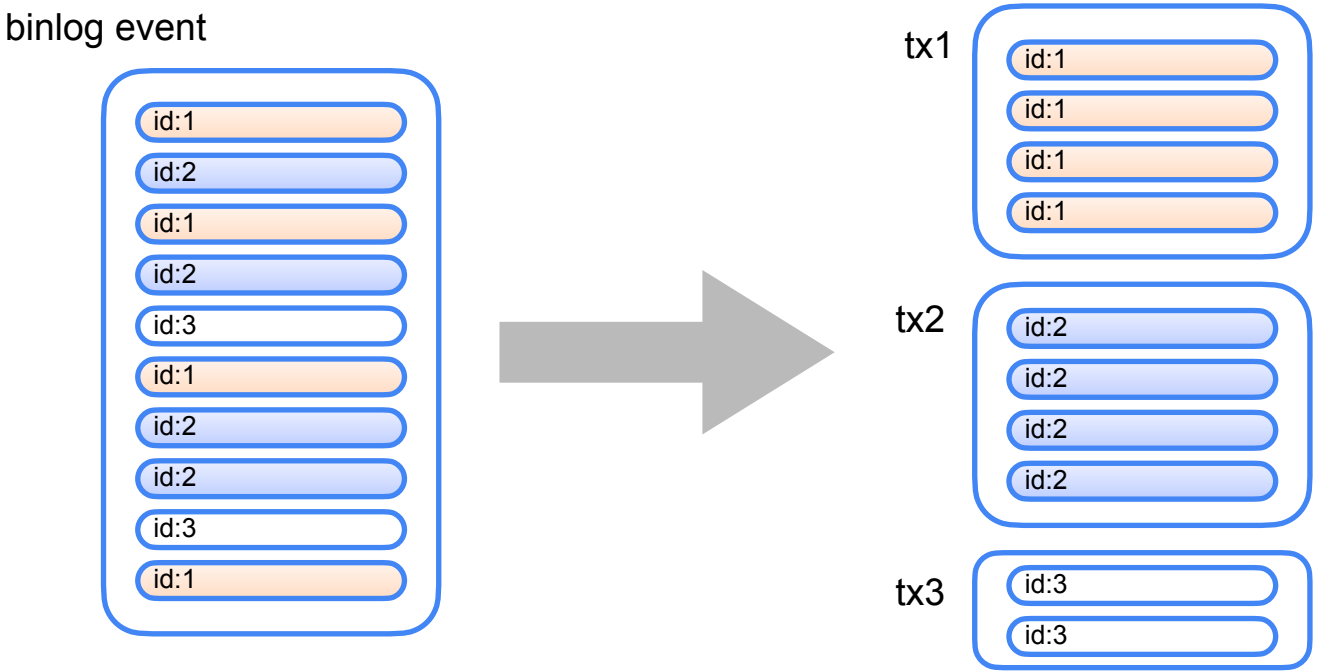


3-3. Consistency - sequential processing (9/10)



3-4. Transctions (10/10)

Divide transactions by user



4.
Uldra-Replicator use case

- Data transfer to Shard DB in real time
- Merge or separate data for delivery
- Change contents while passing data

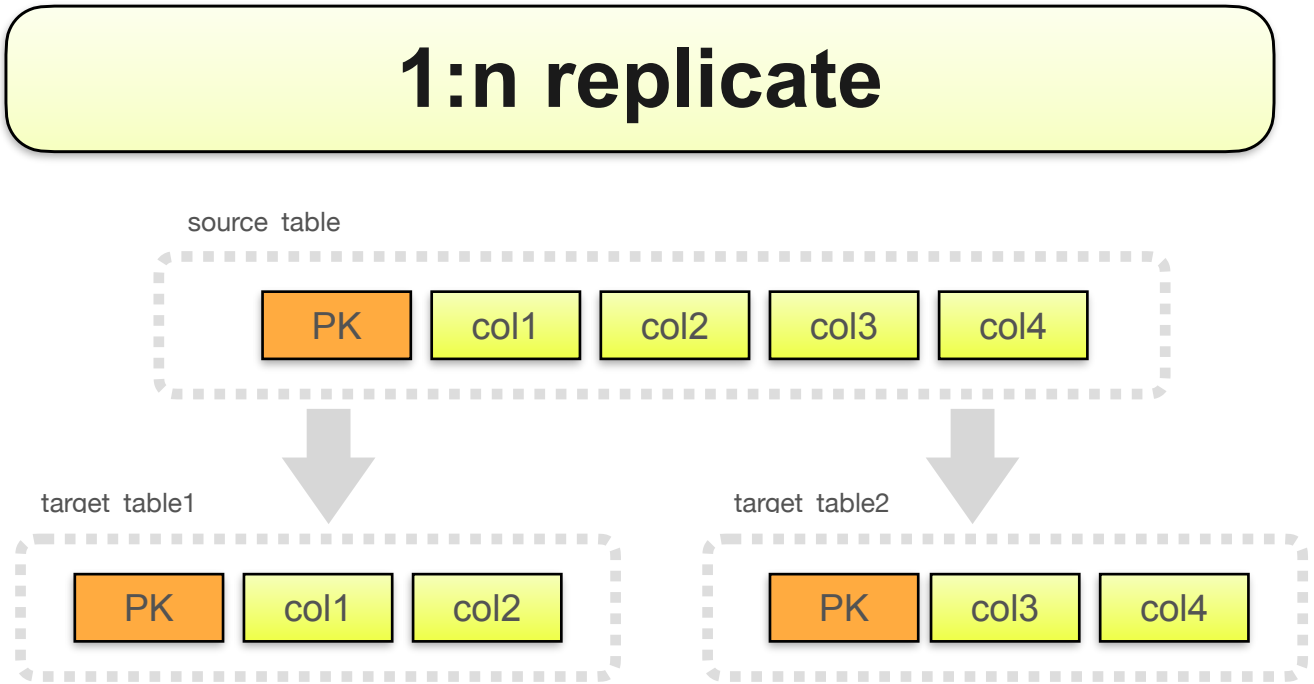
Start uldra-replicator

```
$ mvn install
$ java -jar target/uldra-replicator-0.0.1.jar \
      --config-file="uldra-config.yml"
```

Separate data for delivery

https://git.io/ULDR01

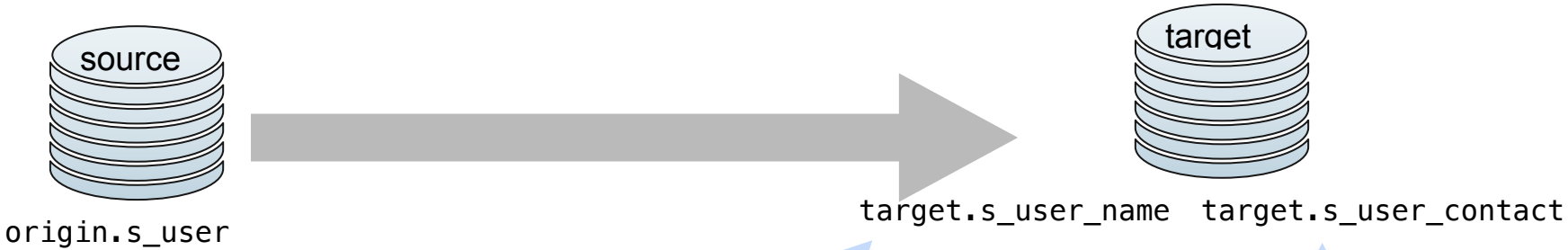
Separate data for delivery (1/4)



Separate data for delivery (2/4)

```
binlogPolicies:
- name: origin.s_user
  groupKey: userid
  replicatPolicies:
  - name: "t_user_name"
    colums: ["userid", "name", "reg_dttm", "mod_tmsp"]
  - name: "t_user_contact"
    colums: ["userid", "email", "phone", "addr", "reg_dttm", "mod_tmsp"]
dataSources:
- !!org.apache.commons.dbcp2.BasicDataSource
  url: jdbc:mysql://127.0.0.1:3306/target
  username: target
  password: target
  maxTotal: 30
```

Separate data for delivery (3/4)




```
mysql> insert into origin.s_user values ('user01', 'user01-name', 'user01@email.com', '00000-01', 'street01', now(), now());
mysql> insert into origin.s_user values ('user02', 'user02-name', 'user02@email.com', '00000-02', 'street02', now(), now());
mysql> insert into origin.s_user values ('user03', 'user03-name', 'user03@email.com', '00000-03', 'street03', now(), now());
mysql> insert into origin.s_user values ('user04', 'user04-name', 'user04@email.com', '00000-04', 'street04', now(), now());
mysql> insert into origin.s_user values ('user05', 'user05-name', 'user05@email.com', '00000-05', 'street05', now(), now());
mysql> insert into origin.s_user values ('user06', 'user06-name', 'user06@email.com', '00000-06', 'street06', now(), now());
mysql> insert into origin.s_user values ('user07', 'user07-name', 'user07@email.com', '00000-07', 'street07', now(), now());
mysql> insert into origin.s_user values ('user08', 'user08-name', 'user08@email.com', '00000-08', 'street08', now(), now());
mysql> insert into origin.s_user values ('user09', 'user09-name', 'user09@email.com', '00000-09', 'street09', now(), now());
```

Separate data for delivery (4/4)

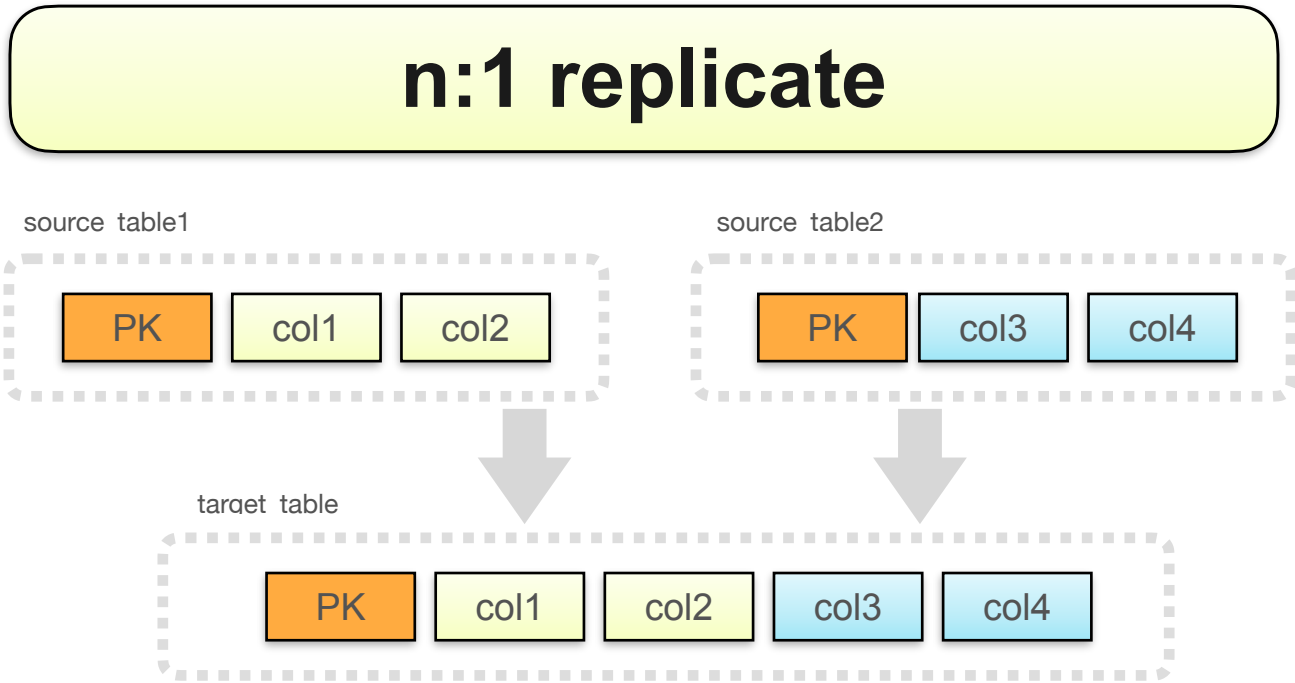
```
mysql> select * from target.t_user_name;
+--------+-------------+---------------------+---------------------+
| userid | name        | reg_dttm            | mod_tmsp            |
+--------+-------------+---------------------+---------------------+
| user01 | user01-name | 2021-04-14 17:02:39 | 2021-04-14 17:02:39 |
| user02 | user02-name | 2021-04-14 17:02:39 | 2021-04-14 17:02:39 |
| user03 | user03-name | 2021-04-14 17:02:39 | 2021-04-14 17:02:39 |
| user04 | user04-name | 2021-04-14 17:02:39 | 2021-04-14 17:02:39 |
| user05 | user05-name | 2021-04-14 17:02:39 | 2021-04-14 17:02:39 |
| user06 | user06-name | 2021-04-14 17:02:39 | 2021-04-14 17:02:39 |
| user07 | user07-name | 2021-04-14 17:02:39 | 2021-04-14 17:02:39 |
| user08 | user08-name | 2021-04-14 17:02:39 | 2021-04-14 17:02:39 |
| user09 | user09-name | 2021-04-14 14:13:22 | 2021-04-14 14:13:22 |
+--------+-------------+---------------------+---------------------+

mysql> select * from target.t_user_contact;
+--------+-----------------+----------+----------+---------------------+---------------------+
| userid | email           | phone    | addr     | reg_dttm            | mod_tmsp            |
+--------+-----------------+----------+----------+---------------------+---------------------+
| user01 | user01@email.com | 00000-01 | street01 | 2021-04-14 17:02:39 | 2021-04-14 17:02:39 |
| user02 | user02@email.com | 00000-02 | street02 | 2021-04-14 17:02:39 | 2021-04-14 17:02:39 |
| user03 | user03@email.com | 00000-03 | street03 | 2021-04-14 17:02:39 | 2021-04-14 17:02:39 |
| user04 | user04@email.com | 00000-04 | street04 | 2021-04-14 17:02:39 | 2021-04-14 17:02:39 |
| user05 | user05@email.com | 00000-05 | street05 | 2021-04-14 17:02:39 | 2021-04-14 17:02:39 |
| user06 | user06@email.com | 00000-06 | street06 | 2021-04-14 17:02:39 | 2021-04-14 17:02:39 |
| user07 | user07@email.com | 00000-07 | street07 | 2021-04-14 17:02:39 | 2021-04-14 17:02:39 |
| user08 | user08@email.com | 00000-08 | street08 | 2021-04-14 17:02:39 | 2021-04-14 17:02:39 |
| user09 | user09@email.com | 00000-09 | street09 | 2021-04-14 14:13:22 | 2021-04-14 14:13:22 |
+--------+-----------------+----------+----------+---------------------+---------------------+
```

Merge data for delivery

https://git.io/ULDR02

Merge data for delivery (1/7)



Merge data for delivery (2/7)

```
binlogPolicies:
- name: origin.s_user
  groupKey: userid
  replicatPolicies:
  - name: "t_user_all_info"
    upsertMode: true
    colums: ["userid", "name", "email", "phone", "addr", "reg_dttm", "mod_tmsp"]
- name: origin.s_user_status
  groupKey: userid
  replicatPolicies:
  - name: "t_user_all_info"
    upsertMode: true
    colums: ["userid", "vaild_yn", "login_dttm", "login_cnt", "mod_tmsp"]
dataSources:
- !!org.apache.commons.dbcp2.BasicDataSource
  url: jdbc:mysql://127.0.0.1:3306/target
  username: target
  password: target
  maxTotal: 30
```

Merge data for delivery (3/7)


```
mysql> insert into origin.s_user values ('user01', 'user01-name', 'user01@email.com', '00000-01', 'street01', now(), now());
mysql> insert into origin.s_user values ('user02', 'user02-name', 'user02@email.com', '00000-02', 'street02', now(), now());
mysql> insert into origin.s_user values ('user03', 'user03-name', 'user03@email.com', '00000-03', 'street03', now(), now());
mysql> insert into origin.s_user values ('user04', 'user04-name', 'user04@email.com', '00000-04', 'street04', now(), now());
mysql> insert into origin.s_user values ('user05', 'user05-name', 'user05@email.com', '00000-05', 'street05', now(), now());
mysql> insert into origin.s_user values ('user06', 'user06-name', 'user06@email.com', '00000-06', 'street06', now(), now());
mysql> insert into origin.s_user values ('user07', 'user07-name', 'user07@email.com', '00000-07', 'street07', now(), now());
mysql> insert into origin.s_user values ('user08', 'user08-name', 'user08@email.com', '00000-08', 'street08', now(), now());
mysql> insert into origin.s_user values ('user09', 'user09-name', 'user09@email.com', '00000-09', 'street09', now(), now());
```

Merge data for delivery (4/7)

```
mysql> select * from target.t_user_all_info;
+--------+-------------+------------------+----------+----------+----------+------------+-----------+---------------------+---------------------+
| userid | name        | email            | phone    | addr     | vaild_yn | login_dttm | login_cnt | reg_dttm            | mod_tmsp            |
+--------+-------------+------------------+----------+----------+----------+------------+-----------+---------------------+---------------------+
| user01 | user01-name | user01@email.com | 00000-01 | street01 | Y        |            |         0 | 2021-04-24 17:24:01 | 2021-04-24 17:24:01 |
| user02 | user02-name | user02@email.com | 00000-02 | street02 | Y        |            |         0 | 2021-04-24 17:24:01 | 2021-04-24 17:24:01 |
| user03 | user03-name | user03@email.com | 00000-03 | street03 | Y        |            |         0 | 2021-04-24 17:24:01 | 2021-04-24 17:24:01 |
| user04 | user04-name | user04@email.com | 00000-04 | street04 | Y        |            |         0 | 2021-04-24 17:24:01 | 2021-04-24 17:24:01 |
| user05 | user05-name | user05@email.com | 00000-05 | street05 | Y        |            |         0 | 2021-04-24 17:24:01 | 2021-04-24 17:24:01 |
| user06 | user06-name | user06@email.com | 00000-06 | street06 | Y        |            |         0 | 2021-04-24 17:24:01 | 2021-04-24 17:24:01 |
| user07 | user07-name | user07@email.com | 00000-07 | street07 | Y        |            |         0 | 2021-04-24 17:24:01 | 2021-04-24 17:24:01 |
| user08 | user08-name | user08@email.com | 00000-08 | street08 | Y        |            |         0 | 2021-04-24 17:24:01 | 2021-04-24 17:24:01 |
| user09 | user09-name | user09@email.com | 00000-09 | street09 | Y        |            |         0 | 2021-04-24 17:24:02 | 2021-04-24 17:24:02 |
+--------+-------------+------------------+----------+----------+----------+------------+-----------+---------------------+---------------------+
```

target query

```
insert ignore into t_user_all_info (phone,name,mod_tmsp,addr,reg_dttm,userid,email) values ('00000-04','user04-name','2021-04-24 17:24:01.000000','street04','2021-04-24 17:24:01.000000','user04','user04@email.com')
 on duplicate key update
    phone=values(phone),
    name=values(name),
    mod_tmsp=values(mod_tmsp),
    addr=values(addr),
    reg_dttm=values(reg_dttm),
    email=values(email);
```

Merge data for delivery (5/7)


```
mysql> insert into origin.s_user_status values ('user01', 'Y', '201901010714', '0', now(), now());
mysql> insert into origin.s_user_status values ('user02', 'Y', '201902120924', '0', now(), now());
mysql> insert into origin.s_user_status values ('user03', 'Y', '202003230134', '0', now(), now());
mysql> insert into origin.s_user_status values ('user04', 'Y', '202004040744', '0', now(), now());
mysql> insert into origin.s_user_status values ('user05', 'Y', '202005150854', '0', now(), now());
mysql> insert into origin.s_user_status values ('user06', 'Y', '202006261904', '0', now(), now());
mysql> insert into origin.s_user_status values ('user07', 'Y', '202007041014', '0', now(), now());
mysql> insert into origin.s_user_status values ('user08', 'Y', '202101151124', '0', now(), now());
mysql> insert into origin.s_user_status values ('user09', 'Y', '202102271234', '0', now(), now());
```

Merge data for delivery (6/7)

```
mysql> select * from target.t_user_all_info;
+--------+-------------+------------------+----------+----------+-----------+--------------+-----------+---------------------+---------------------+
| userid | name        | email            | phone    | addr     | vaild_yn  | login_dttm   | login_cnt | reg_dttm            | mod_tmsp            |
+--------+-------------+------------------+----------+----------+-----------+--------------+-----------+---------------------+---------------------+
| user01 | user01-name | user01@email.com | 00000-01 | street01 | Y         | 201901010714 |         0 | 2021-04-24 17:24:01 | 2021-04-24 17:30:25 |
| user02 | user02-name | user02@email.com | 00000-02 | street02 | Y         | 201902120924 |         0 | 2021-04-24 17:24:01 | 2021-04-24 17:30:25 |
| user03 | user03-name | user03@email.com | 00000-03 | street03 | Y         | 202003230134 |         0 | 2021-04-24 17:24:01 | 2021-04-24 17:30:25 |
| user04 | user04-name | user04@email.com | 00000-04 | street04 | Y         | 202004040744 |         0 | 2021-04-24 17:24:01 | 2021-04-24 17:30:25 |
| user05 | user05-name | user05@email.com | 00000-05 | street05 | Y         | 202005150854 |         0 | 2021-04-24 17:24:01 | 2021-04-24 17:30:25 |
| user06 | user06-name | user06@email.com | 00000-06 | street06 | Y         | 202006261904 |         0 | 2021-04-24 17:24:01 | 2021-04-24 17:30:25 |
| user07 | user07-name | user07@email.com | 00000-07 | street07 | Y         | 202007041014 |         0 | 2021-04-24 17:24:01 | 2021-04-24 17:30:25 |
| user08 | user08-name | user08@email.com | 00000-08 | street08 | Y         | 202101151124 |         0 | 2021-04-24 17:24:01 | 2021-04-24 17:30:25 |
| user09 | user09-name | user09@email.com | 00000-09 | street09 | Y         | 202102271234 |         0 | 2021-04-24 17:24:02 | 2021-04-24 17:30:25 |
+--------+-------------+------------------+----------+----------+-----------+--------------+-----------+---------------------+---------------------+
```

target query

```
insert ignore into t_user_all_info (login_dttm,login_cnt,vaild_yn,mod_tmsp,userid) values ('20210115112402','0','Y','2021-04-24
17:30:25.000000','user08')
 on duplicate key update
     login_dttm=values(login_dttm),
     login_cnt=values(login_cnt),
     vaild_yn=values(vaild_yn),
     mod_tmsp=values(mod_tmsp);
```

Merge data for delivery (7/7)

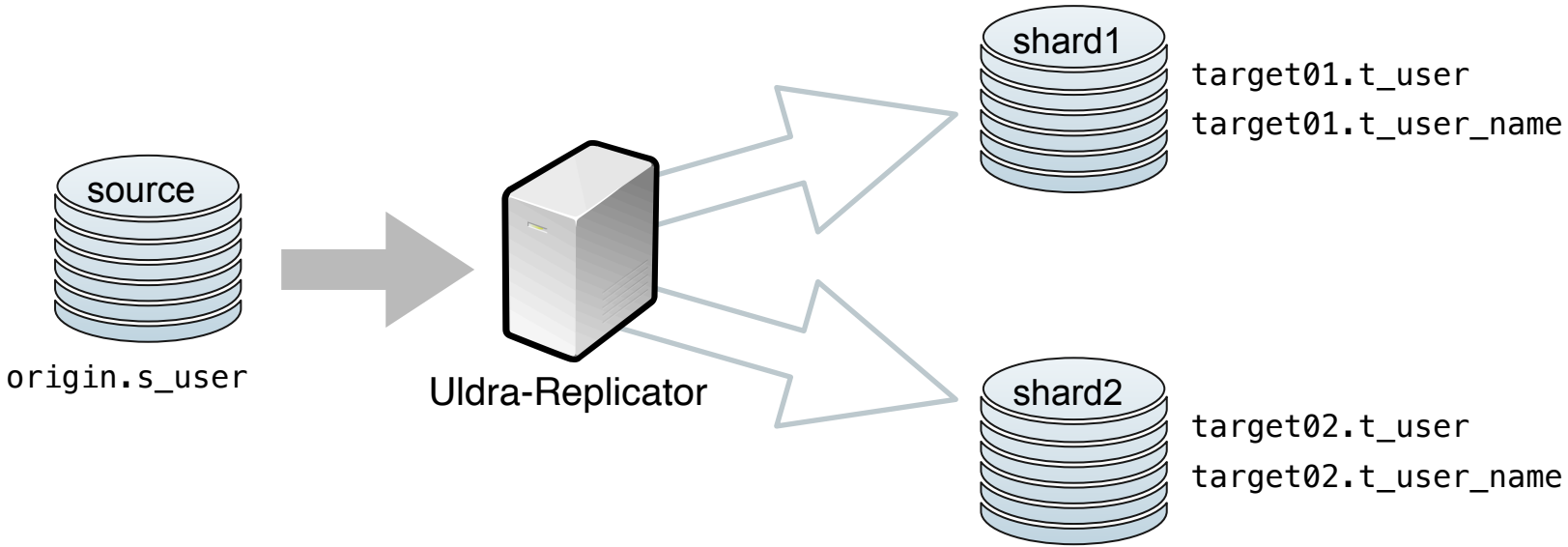
```
mysql> update origin.s_user_status set login_cnt = login_cnt + 1 order by rand() limit 3;
Query OK, 3 rows affected (0.02 sec)
Rows matched: 3  Changed: 3  Warnings: 0

mysql> select * from target.t_user_all_info;
+--------+-------------+------------------+----------+----------+----------+--------------+-----------+---------------------+---------------------+
| userid | name        | email            | phone    | addr     | vaild_yn | login_dttm   | login_cnt | reg_dttm            | mod_tmsp            |
+--------+-------------+------------------+----------+----------+----------+--------------+-----------+---------------------+---------------------+
| user01 | user01-name | user01@email.com | 00000-01 | street01 | Y        | 201901010714 |         1 | 2021-04-24 17:24:01 | 2021-04-24 17:30:25 |
| user02 | user02-name | user02@email.com | 00000-02 | street02 | Y        | 201902120924 |         0 | 2021-04-24 17:24:01 | 2021-04-24 17:30:25 |
| user03 | user03-name | user03@email.com | 00000-03 | street03 | Y        | 202003230134 |         0 | 2021-04-24 17:24:01 | 2021-04-24 17:30:25 |
| user04 | user04-name | user04@email.com | 00000-04 | street04 | Y        | 202004040744 |         0 | 2021-04-24 17:24:01 | 2021-04-24 17:30:25 |
| user05 | user05-name | user05@email.com | 00000-05 | street05 | Y        | 202005150854 |         0 | 2021-04-24 17:24:01 | 2021-04-24 17:30:25 |
| user06 | user06-name | user06@email.com | 00000-06 | street06 | Y        | 202006261904 |         0 | 2021-04-24 17:24:01 | 2021-04-24 17:30:25 |
| user07 | user07-name | user07@email.com | 00000-07 | street07 | Y        | 202007041014 |         0 | 2021-04-24 17:24:01 | 2021-04-24 17:30:25 |
| user08 | user08-name | user08@email.com | 00000-08 | street08 | Y        | 202101151124 |         1 | 2021-04-24 17:24:01 | 2021-04-24 17:30:25 |
| user09 | user09-name | user09@email.com | 00000-09 | street09 | Y        | 202102271234 |         1 | 2021-04-24 17:24:02 | 2021-04-24 17:30:25 |
+--------+-------------+------------------+----------+----------+----------+--------------+-----------+---------------------+---------------------+
```

Data transfer to Shard DB

https://git.io/ULDR03

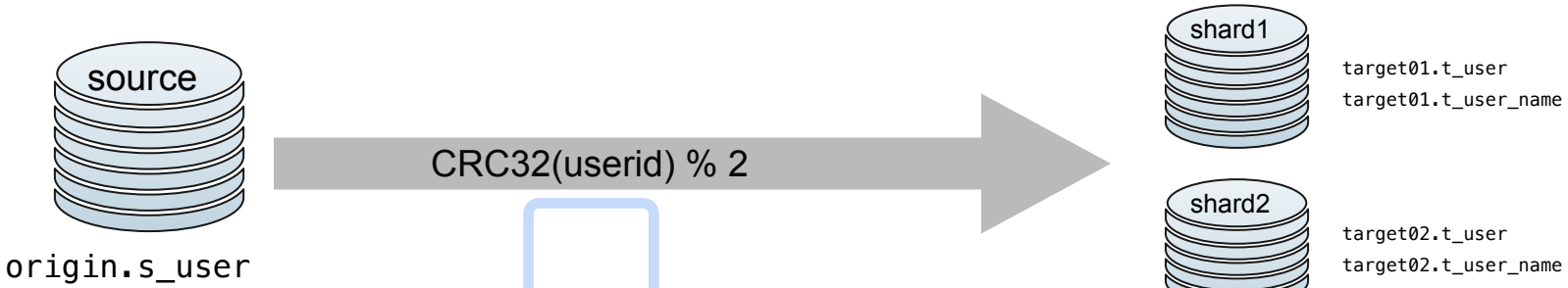
Data transfer to Shard DB in real time



Data transfer to Shard DB in real time (2/4)

```
binlogPolicies:
- name: origin.s_user
  groupKey: userid
  replicatPolicies:
  - name: "t_user"
  - name: "t_user_name"
    softDelete: true
    upsertMode: true
    colums: ["userid", "name", "reg_dttm", "mod_tmsp"]
dataSources:
- !!org.apache.commons.dbcp2.BasicDataSource
  url: jdbc:mysql://127.0.0.1:3306/target01
  username: target01
  password: target01
  maxTotal: 30
- !!org.apache.commons.dbcp2.BasicDataSource
  url: jdbc:mysql://127.0.0.1:3306/target02
  username: target02
  password: target02
  maxTotal: 30
```

Data transfer to Shard DB in real time (3/4)




```
mysql> insert into origin.s_user values ('user01', 'user01-name', 'user01@email.com', '00000-01', 'street01', now(), now());
mysql> insert into origin.s_user values ('user02', 'user02-name', 'user02@email.com', '00000-02', 'street02', now(), now());
mysql> insert into origin.s_user values ('user03', 'user03-name', 'user03@email.com', '00000-03', 'street03', now(), now());
mysql> insert into origin.s_user values ('user04', 'user04-name', 'user04@email.com', '00000-04', 'street04', now(), now());
mysql> insert into origin.s_user values ('user05', 'user05-name', 'user05@email.com', '00000-05', 'street05', now(), now());
mysql> insert into origin.s_user values ('user06', 'user06-name', 'user06@email.com', '00000-06', 'street06', now(), now());
mysql> insert into origin.s_user values ('user07', 'user07-name', 'user07@email.com', '00000-07', 'street07', now(), now());
mysql> insert into origin.s_user values ('user08', 'user08-name', 'user08@email.com', '00000-08', 'street08', now(), now());
mysql> insert into origin.s_user values ('user09', 'user09-name', 'user09@email.com', '00000-09', 'street09', now(), now());
```

Data transfer to Shard DB in real time (4/4)

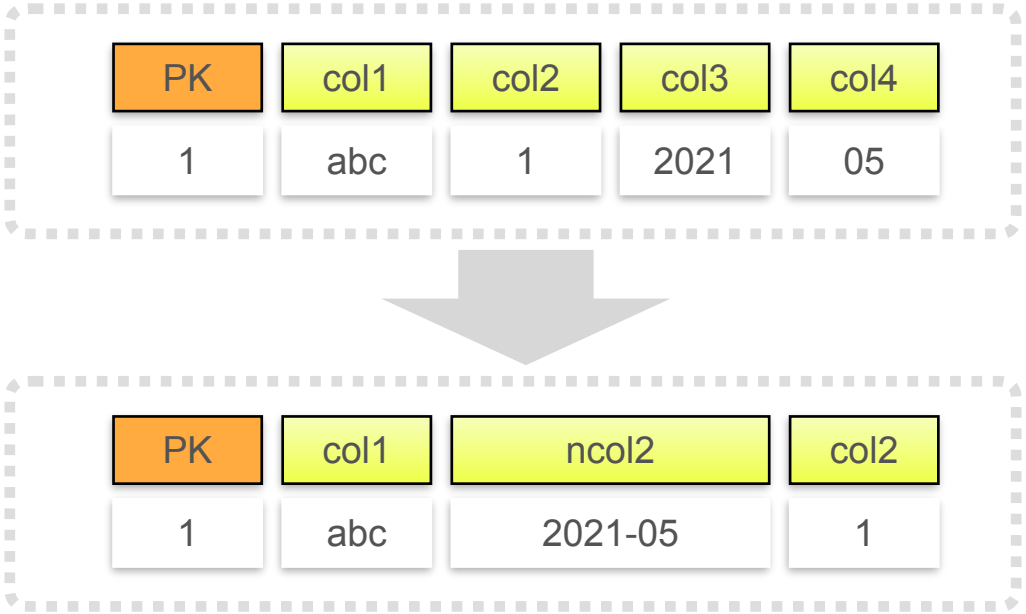
```
mysql> select 'target01' as target, crc32(t1.userid)%2 "crc % 2", t1.* from target01.t_user t1
    -> union all
    -> select 'target02' as target, crc32(t2.userid)%2 "crc % 2", t2.* from target02.t_user t2;
+----------+---------+--------+------------+-------------------+----------+----------+---------------------+---------------------+
| target   | crc % 2 | userid | name       | email             | phone    | addr     | reg_dttm            | mod_tmsp            |
+----------+---------+--------+------------+-------------------+----------+----------+---------------------+---------------------+
| target01 |       0 | user01 | user01-name | user01@email.com | 00000-01 | street01 | 2021-04-24 13:45:27 | 2021-04-24 13:45:27 |
| target01 |       0 | user02 | user02-name | user02@email.com | 00000-02 | street02 | 2021-04-24 13:47:29 | 2021-04-24 13:47:29 |
| target01 |       0 | user03 | user03-name | user03@email.com | 00000-03 | street03 | 2021-04-24 13:47:43 | 2021-04-24 13:47:43 |
| target01 |       0 | user08 | user08-name | user08@email.com | 00000-08 | street08 | 2021-04-24 14:04:46 | 2021-04-24 14:04:46 |
| target01 |       0 | user09 | user09-name | user09@email.com | 00000-09 | street09 | 2021-04-24 14:04:46 | 2021-04-24 14:04:46 |
| target02 |       1 | user04 | user04-name | user04@email.com | 00000-04 | street04 | 2021-04-24 13:48:06 | 2021-04-24 13:48:06 |
| target02 |       1 | user05 | user05-name | user05@email.com | 00000-05 | street05 | 2021-04-24 14:04:46 | 2021-04-24 14:04:46 |
| target02 |       1 | user06 | user06-name | user06@email.com | 00000-06 | street06 | 2021-04-24 14:04:46 | 2021-04-24 14:04:46 |
| target02 |       1 | user07 | user07-name | user07@email.com | 00000-07 | street07 | 2021-04-24 14:04:46 | 2021-04-24 14:04:46 |
+----------+---------+--------+------------+-------------------+----------+----------+---------------------+---------------------+

mysql> select 'target01' as target, crc32(t1.userid)%2 "crc % 2", t1.* from target01.t_user_name t1
    -> union all
    -> select 'target02' as target, crc32(t2.userid)%2 "crc % 2", t2.* from target02.t_user_name t2;
+----------+---------+--------+------------+---------------------+---------------------+
| target   | crc % 2 | userid | name       | reg_dttm            | mod_tmsp            |
+----------+---------+--------+------------+---------------------+---------------------+
| target01 |       0 | user01 | user01-name | 2021-04-24 14:13:21 | 2021-04-24 14:13:21 |
| target01 |       0 | user02 | user02-name | 2021-04-24 14:13:21 | 2021-04-24 14:13:21 |
| target01 |       0 | user03 | user03-name | 2021-04-24 14:13:21 | 2021-04-24 14:13:21 |
| target01 |       0 | user08 | user08-name | 2021-04-24 14:13:21 | 2021-04-24 14:13:21 |
| target01 |       0 | user09 | user09-name | 2021-04-24 14:13:22 | 2021-04-24 14:13:22 |
| target02 |       1 | user04 | user04-name | 2021-04-24 14:13:21 | 2021-04-24 14:13:21 |
| target02 |       1 | user05 | user05-name | 2021-04-24 14:13:21 | 2021-04-24 14:13:21 |
| target02 |       1 | user06 | user06-name | 2021-04-24 14:13:21 | 2021-04-24 14:13:21 |
| target02 |       1 | user07 | user07-name | 2021-04-24 14:13:21 | 2021-04-24 14:13:21 |
+----------+---------+--------+------------+---------------------+---------------------+
```

Change content while passing data

https://git.io/ULDR04

Change content while passing data (1/6)



Change content while passing data (2/6)

Row Handler Interface

```
package net.gywn.binlog.handler;

import net.gywn.binlog.beans.BinlogOperation;

public interface RowHandler {
     public void init();

     public void modify(final BinlogOperation operation);
}
```

Change content while passing data (3/6)

Row Handler Implement

```
package net.gywn.binlog.custom;

import net.gywn.binlog.api.RowHandler;
import net.gywn.binlog.beans.BinlogOperation;

public class RowHandlerCustom implements RowHandler {

	@Override
	public void init() {
		// TODO Auto-generated method stub
	}

	@Override
	public void modify(final BinlogOperation operation) {
		String nameOrigin = operation.getDatMap().get("name");
		String addrOrigin = operation.getDatMap().get("addr");
		String newname = String.format("%s:%s", nameOrigin, addrOrigin);
		operation.getDatMap().put("newname", newname);
	}
}
```

Change content while passing data (4/6)

```
binlogPolicies:
- name: origin.s_user
  groupKey: userid
  rowHandlerClassName: "net.gywn.binlog.custom.RowHandlerCustom"
  replicatPolicies:
  - name: "t_user_newname"
    colums: ["userid", "newname", "reg_dttm", "mod_tmsp"]
dataSources:
- !!org.apache.commons.dbcp2.BasicDataSource
  url: jdbc:mysql://127.0.0.1:3306/target
  username: target
  password: target
  maxTotal: 30
```

Change content while passing data (5/6)


```
mysql> insert into origin.s_user values ('user01', 'user01-name', 'user01@email.com', '00000-01', 'street01', now(), now());
mysql> insert into origin.s_user values ('user02', 'user02-name', 'user02@email.com', '00000-02', 'street02', now(), now());
mysql> insert into origin.s_user values ('user03', 'user03-name', 'user03@email.com', '00000-03', 'street03', now(), now());
mysql> insert into origin.s_user values ('user04', 'user04-name', 'user04@email.com', '00000-04', 'street04', now(), now());
mysql> insert into origin.s_user values ('user05', 'user05-name', 'user05@email.com', '00000-05', 'street05', now(), now());
mysql> insert into origin.s_user values ('user06', 'user06-name', 'user06@email.com', '00000-06', 'street06', now(), now());
mysql> insert into origin.s_user values ('user07', 'user07-name', 'user07@email.com', '00000-07', 'street07', now(), now());
mysql> insert into origin.s_user values ('user08', 'user08-name', 'user08@email.com', '00000-08', 'street08', now(), now());
mysql> insert into origin.s_user values ('user09', 'user09-name', 'user09@email.com', '00000-09', 'street09', now(), now());
```

Change content while passing data (6/6)

```
mysql> select * from target.t_user_newname;
+--------+----------------------+---------------------+---------------------+
| userid | newname              | reg_dttm            | mod_tmsp            |
+--------+----------------------+---------------------+---------------------+
| user01 | user01-name:street01 | 2021-04-24 11:47:43 | 2021-04-24 11:47:43 |
| user02 | user02-name:street02 | 2021-04-24 11:47:43 | 2021-04-24 11:47:43 |
| user03 | user03-name:street03 | 2021-04-24 11:47:43 | 2021-04-24 11:47:43 |
| user04 | user04-name:street04 | 2021-04-24 11:47:43 | 2021-04-24 11:47:43 |
| user05 | user05-name:street05 | 2021-04-24 11:47:43 | 2021-04-24 11:47:43 |
| user06 | user06-name:street06 | 2021-04-24 11:47:43 | 2021-04-24 11:47:43 |
| user07 | user07-name:street07 | 2021-04-24 11:47:43 | 2021-04-24 11:47:43 |
| user08 | user08-name:street08 | 2021-04-24 11:47:43 | 2021-04-24 11:47:43 |
| user09 | user09-name:street09 | 2021-04-24 11:47:44 | 2021-04-24 11:47:44 |
+--------+----------------------+---------------------+---------------------+
```

5.
Conclusion

“

Flexible data transfer

uldra-replicator

Conclusion

- Binlog parsing & convert data
- Easy - overwriting log events
- Performance - grouping data, parallel processing
- Data Consistency - same group data, sequential processing
- Transaction - Dividing transaction by user

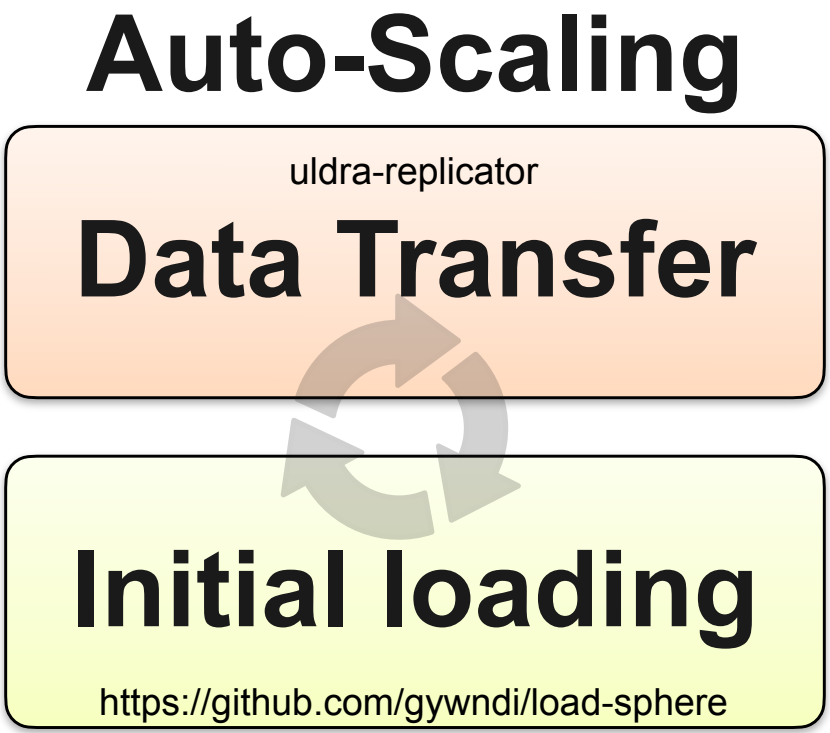
Conclusion

- Merge or separate data for delivery
- Data transfer to Shard DB in real time
- Change contents while passing data

Conclusion

https://github.com/gywndi/uldra-replicator

Final goal


Auto-Scaling
uldra-replicator
Data Transfer
Initial loading
https://github.com/gywndi/load-sphere

THANK YOU !

PERCONA
LIVEONLINE
MAY 12 - 13th
2021