

FLINK SQL REDIS Connector

The Redis connector allows for reading data from and writing data into Redis table.

1. How to create a Redis table

The example below shows how to create a Redis table:

```
CREATE TABLE orders (  
  `order_id` STRING,  
  `price` STRING,  
  `order_time` STRING,  
  PRIMARY KEY(order_id) NOT ENFORCED  
) WITH (  
  'connector' = 'redis',  
  'mode' = 'single',  
  'single.host' = '192.168.10.101',  
  'single.port' = '6379',  
  'password' = 'xxxxxx',  
  'command' = 'hmset',  
  'key' = 'orders'  
);
```

Note:

1. Redis table must be define primary key
2. Redis table can not save data type and you must be cast data type to string

2. Writing data example

1. generate source data

```
CREATE TABLE order_source (  
  `order_number` BIGINT,  
  `price` DECIMAL(32,2),  
  `order_time` TIMESTAMP(3),  
  PRIMARY KEY(order_id) NOT ENFORCED  
) WITH (  
  'connector' = 'datagen',  
  'number-of-rows' = '5',  
  'fields.order_number.min' = '1',  
  'fields.order_number.max' = '20',  
  'fields.price.min' = '1001',  
  'fields.price.max' = '1100'  
);
```

2. define redis sink table

```
CREATE TABLE orders (  
  `order_number` STRING,  
  `price` STRING,
```

```

`order_time` STRING,
PRIMARY KEY(order_id) NOT ENFORCED
) WITH (
  'connector' = 'redis',
  'mode' = 'single',
  'single.host' = '192.168.10.101',
  'single.port' = '6379',
  'password' = 'xxxxxx',
  'command' = 'hmset',
  'key' = 'orders'
);

```

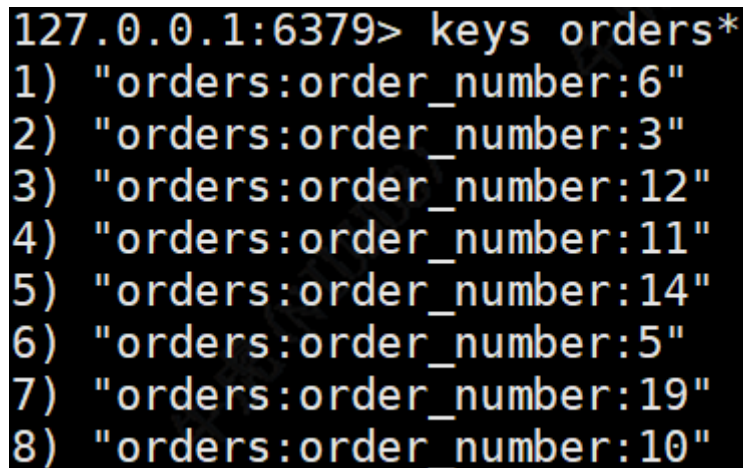
3. insert data to redis sink table (cast data type to string)

```

insert into redis_sink
select
  cast(order_number as STRING) order_number,
  cast(price as STRING) price,
  cast(order_time as STRING) order_time
from orders

```

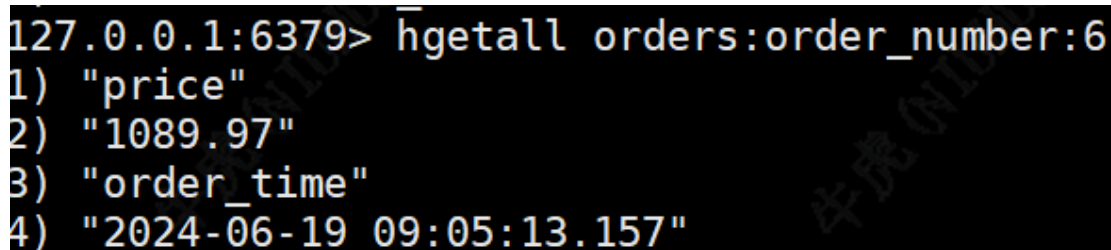
Example data save in Redis as picture:



```

127.0.0.1:6379> keys orders*
1) "orders:order_number:6"
2) "orders:order_number:3"
3) "orders:order_number:12"
4) "orders:order_number:11"
5) "orders:order_number:14"
6) "orders:order_number:5"
7) "orders:order_number:19"
8) "orders:order_number:10"

```



```

127.0.0.1:6379> hgetall orders:order_number:6
1) "price"
2) "1089.97"
3) "order_time"
4) "2024-06-19 09:05:13.157"

```

Note:

1. Redis key = key : primary key: (primary key) value
2. the default is comma separated

3. Reading data example

1. define redis read table

```
CREATE TABLE orders (
  `order_number` STRING,
  `price` STRING,
  `order_time` STRING,
  PRIMARY KEY(order_id) NOT ENFORCED
) WITH (
  'connector' = 'redis',
  'mode' = 'single',
  'single.host' = '192.168.10.101',
  'single.port' = '6379',
  'password' = 'xxxxxx',
  'command' = 'hgetall',
  'key' = 'orders'
);
```

2. query data from redis table

```
select * from orders
```

```
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
```

op	order_number	price	order_time
+I	14	1007.95	2024-06-19 09:05:13.137
+I	12	1026.44	2024-06-19 09:04:35.970
+I	10	1011.41	2024-06-19 09:04:36.111
+I	3	1095.39	2024-06-19 09:04:36.133
+I	11	1044.10	2024-06-19 09:05:13.117
+I	19	1002.48	2024-06-19 09:05:13.097
+I	6	1089.97	2024-06-19 09:05:13.157
+I	5	1073.50	2024-06-19 09:05:13.024

4. Cluster Mode

Redis connector also support cluster mode: cluster mode need to use 'cluster.nodes' option to describe redis cluster hosts and ports and also mode need to change 'cluster' mode

```
create table redis_sink (
  site_id STRING,
  inverter_id STRING,
  start_time STRING,
  PRIMARY KEY(site_id) NOT ENFORCED
) WITH (
  'connector' = 'redis',
  'mode' = 'cluster',
  'cluster.nodes' =
  'test3:7001,test3:7002,test3:7003,test3:8001,test3:8002,test3:8003',
  'password' = '123123',
  'command' = 'hmset',
  'key' = 'site_inverter'
);
```

5. Connector Options

Option	Required	Default	Type	Description
connector	required	no	String	connector name

Option	Required	Default	Type	Description
mode	required	no	String	redis cluster mode (single or cluster)
single.host	optional	no	String	redis single mode machine host
single.port	optional	no	int	redis single mode running port
password	optional	no	String	redis database password
command	required	no	String	redis write data or read data command
key	required	no	String	redis key
expire	optional	no	Int	set key ttl
field	optional	no	String	get a value with field when using hget command
cursor	optional	no	Int	using hscan command(e.g:1,2)
start	optional	0	Int	read data when using lrange command
end	optional	10	Int	read data when using lrange command
connection.max.wait-mills	optional	no	Int	redis connection parameter
connection.timeout-ms	optional	no	Int	redis connection parameter
connection.max-total	optional	no	Int	redis connection parameter
connection.max-idle	optional	no	Int	redis connection parameter
connection.test-on-borrow	optional	no	Boolean	redis connection parameter
connection.test-on-return	optional	no	Boolean	redis connection parameter
connection.test-while-idle	optional	no	Boolean	redis connection parameter
so.timeout-ms	optional	no	Int	redis connection parameter
max.attempts	optional	no	Int	redis connection parameter

6.Fuzzy matching

Hash Type: support fuzzy matching

```
create table redis_sink (  
  order_number STRING,  
  price STRING,  
  order_time STRING,  
  PRIMARY KEY(order_number) NOT ENFORCED  
) WITH (  
  'connector' = 'redis',  
  'mode' = 'single',  
  'single.host' = 'test1',  
  'single.port' = '6379',  
  'password' = 'xxxxxxxx',  
  'command' = 'hgetall',  
  'key' = 'orders'  
);
```

Query result:

op	order_number	price	order_time
+I	14	1007.95	2024-06-19 09:05:13.137
+I	12	1026.44	2024-06-19 09:04:35.970
+I	10	1011.41	2024-06-19 09:04:36.111
+I	11	1044.10	2024-06-19 09:05:13.117
+I	3	1095.39	2024-06-19 09:04:36.133
+I	19	1002.48	2024-06-19 09:05:13.097
+I	6	1089.97	2024-06-19 09:05:13.157
+I	5	1073.50	2024-06-19 09:05:13.024

In example we use "key = orders" and find the whole table of orders, and also we can use "orders:orders_number". the connector still recognize.