

語法

SQL

```
SELECT * FROM table WHERE NAME=':NAME' [ AND BIRTHDAY=':BIRTHDAY' ] WITH UR
```

Java

```
DataSet ds = Transaction.getDynamicDataSet();  
ds.setField("NAME", "王小明");  
if(BIRTHDAY!=null && BIRTHDAY.trim().length!=0){  
    ds.setField("BIRTHDAY", "2017-09-04");  
}  
ds.searchAndRetrieve(SQL);
```

eBAF SQL 的優點

- 用':KEY'取代 PreparedStatement 中的?
 - 對應方便
 - 重複的 KEY 只要設定一次
- 用[SQL 語法]動態設定 SQL
 - 如果中括號裡面的所有參數都有設定的話，轉為標準 SQL 時，中括號裡面的語法會出現
 - 不需要針對不同條件寫多句 SQL

eBAF SQL 的缺點

- 效能會比較差，因為要將 eBAF SQL 轉為標準 SQL

其它語法

in(A, B, ...)

SQL

```
SELECT * FROM table WHERE TYPE='in:TYPE' WITH UR
```

Java

```
DataSet ds = Transaction.getDynamicDataSet();  
ds.setFieldValues("TYPE", {"日本", "香港", "韓國"});  
ds.searchAndRetrieve(SQL);
```

動態設定

- select 欄位
- Table
- order by 欄位
- group by 欄位

API

```
DataSet.setField( FieldType fieldType, String paramName, String...  
paramValues )
```

FieldType: 設定種類

- FieldType.queryField
- FieldType.table
- FieldType.orderBy
- FieldType.groupBy

paramName: 參數名稱

paramValues: 參數

範例

SQL

```
select 'queryField:QUERY_COL'
from 'table:QUERY_TABLE' A
[ left join 'table:QUERY_TABLE2' B on A.CODE_NAME = B.CODE_NAME ]
[ left join 'table:QUERY_TABLE3' B on A.BATCH_ID = B.BATCH_ID ]
where 1=1
[ and A.BATCH_ID = ':BATCH_ID' ]
[ and A.CODE_NAME = ':CODE_NAME' ]
[ 'groupBy:GROUP_COL' ]
[ 'orderBy:ORDER_COL1' ASC ]
[ 'orderBy:ORDER_COL2' DESC ]
with ur
```

Java

```
if("1".equals(status)){
    ds.setField(FieldType.queryField, "QUERY_COL", new
String[]{"A.CODE_NAME", "A.SQLSTRING"});
    ds.setField(FieldType.table, "QUERY_TABLE", new String[]{"DBCM",
"DTCM04ZZ"});
    ds.setField(FieldType.table, "QUERY_TABLE2", new String[]{"DBCM",
"DTCM05ZZ"});
    ds.setField(FieldType.orderBy, "ORDER_COL1", "A.CODE_NAME");
    ds.setField("CODE_NAME", "test_CODE_NAME");
}
```

轉換後的 SQL

```
select A.CODE_NAME, A.SQLSTRING
from DBCM.DTCM04ZZ A
left join DBCM.DTCM05ZZ B on A.CODE_NAME = B.CODE_NAME
where 1=1 and A.CODE_NAME = ?
order by A.CODE_NAME ASC
with ur
```

[練習]