# Oracle-MySQL兼容改造（网办）

## （一）、to\_date()函数修改

### 1．问题出现形式

to\_date()是Oracle数据库中的日期字符转日期类型的函数，而在MySQL中不存在这样的函数。因此，若在SQL语句中使用，切换到MySQL中导致不兼容。

（1）to\_date()函数使用在查询语句中的where子句中作为日期查询条件。

例如：

select \* from table where createDate >to\_date(?,'yyyy-mm-dd hh24:mi:ss');

注入字符串形式的日期参数: find (sql, new Object[] {dataStr})；

（2）to\_date()函数使用在更新语句/插入语中作为更新/插入日期字段

例如：

String sql = "update table set time=to\_date(?,'yyyy-mm-dd hh24:mi:ss') where id=?";

注入字符串形式的日期参数：update (sql, new Object[] {dataStr})；

### 2．兼容修改方法

注入Timestamp类型参数：去除SQL中的to\_date()函数，然后将字符串日期类型转化为Timestamp类型，注入到SQL中。

例如：

String sql = "select \* from table where createDate >?";

Timestamp d= CommonUtils\_api.getInstance().parseStringToTimestamp(

dateStr,CommonUtils\_api.YYYY\_MM\_DD\_HH\_mm\_SS);

注入Timestamp类型参数find (sql, new Object[] {d})；

## （二）、to\_char()函数修改

to\_char()是Oracle数据库中将日期转为字符格式的函数，而在MySQL数据库中不存在。因此，在使用to\_cahr()函数的SQL语句切换到MySQL时导致不兼容。

### 1．出现形式

在查询语句中使用to\_char函数将日期转为指定类型的日期字符格式，在前台页面直接展示。

例如：

select to\_char(CREATEDATE,'yyyy-mm-dd hh24:mi:ss') CREATEDATE from table。

在SQL Where条件中使用to\_char()，

例如：

Select \* from table where to\_char(cTime, ,'yyyy-mm-dd hh24:mi:ss') > to\_char(?,'yyyy-mm-dd hh24:mi:ss')

### 2．兼容修改方法

在SQL语句中对日期不进行格式转化，将日期格式的字段放到页面上格式化。

SQL语句修改为：select CREATEDATE from table。

页面上格式化日期：

网办管理系统：

var date = new Date(CREATEDATE);

var month=date.getMonth()+1;

var day = date.getDate();

var dateStr = date.getFullYear() + '-' + month + '-' + day;

网上办事系统：

LEx.util.Format.formatDate(CREATEDATE)

若to\_char（）函数使用在查询条件中，作为日期大小比较的情况，我们按照to\_date 的修改方式将日期比较修改为参数注入方式。

例如：

String sql = "select \* from table where cTime >?";

Timestamp d= CommonUtils\_api.getInstance().parseStringToTimestamp(

dateStr,CommonUtils\_api.YYYY\_MM\_DD\_HH\_mm\_SS);

注入Timestamp类型参数find (sql, new Object[] {d})；

## （三）、select \* from table 查询方式修改

### 1．问题出现形式

使用select \* from table 语句在MySQL数据库下查询出的结果集字段名称为小写，在ORACLE数据库下结果集字段名称为大写。这就会使得处理查询结果的json串时由于大小写之分导致获取值不兼容。使用select \* from table 语句在网版管理系统中大量出现。

例如：

String sql = “select \* from table where …”;

Page<Record> list = super.paginate(page, rows, sql+where,param.toArray());

JSONObject json = toJqGrid(list);

此时，在后台或者页面通过json中的键取值就存在大小写之区分的兼容问题

在后台使用情况：Json.get(“key”); Json.get(“KEY”);

在页面使用情况：

jqGrid(colModel:[name:”key”]); jqGrid(colModel:[name:”KEY”])

### 2．兼容修改方法

这个问题我们尝试使用过的方法有两种。

第一种为：在查出的json结果中将json字段键统一转大写或者小写，来兼容不同数据库的情况。

第二种方法是：修改SQL查询语句，将select \* from table 修改为 select 字段名，字段名2 from table。查询我们所需要使用的字段就可以，不用每次都查询出所有字段。

为了提高执行效率我们采用第二种方法进行兼容修改，即SQL语句修改。

将带有select \* 的SQL 查询语句修改为：

Select 字段名1，字段名2，…字段名n from table。

这里，无论字段名大写，还是小写返回结果均大写。系统中我们采用字段名大写的方式。

## （四）、sysdate获取系统时间修改

### 1．问题出现形式

Sysdate 是ORACLE数据库获取系统当前时间的时间函数。网办系统中主要出现在插入语句和更新语句中，来记录插入数据和更新数据的时间。

例如：

insert into table(id,name,parent,ctime) values (?,?,?,sysdate);

update table set name=?, state=?, address=? ,ctime = sysdate where id =?;

### 2．兼容修改方法

将sysdate去除，修改为参数注入方式，注入参数为系统当前时间new Date()。

例如：

sql = insert into table(id,name t,ctime) values (?,?,?);

sql =update table set id =?, name =?, ctime = ? where …;

update(sql, new Object[]{id,name,new Date()});

## （五）、rownum分页兼容修改

### 1．出现问题形式

rownum是ORACLE数据库的下使用的函数，系统中使用rownum来进行获取指定数量的数据。在SQL语句中使用后，当切换到MySQL时，就会导致不兼容。

例如：

Select \* from table where 条件 AND rownum<50;

executeDataset(sql, params, dsName)

### 2．兼容修改方式

将rownum去除，然后使用分页查询函数进行查询指定条数的数据。

例如：

Sql = select \* from table where 条件;

executeDataset(sql, start, limit, params);

## （六）、嵌套查询表别名兼容修改

### 1．出现问题形式

这里讨论的嵌套查询是：子查询作为父查询的查询表的嵌套查询。

在Oracle与MySQL中“select \* from (select \* from table )结构”有所区别：

在Oracle中子查询结果作为父查询的表时可以使用别名，也可以不使用别名。在Oracle中表的别名不使用 as。

在MySQL中子查询结果作为父查询的表时必须使用别名，在MySQL中表的别名可以使用 as,也可以不使用as。

**在Oracle 中**：

Select \* from (select id,name from table) (无别名) 正确执行

Select \* from (select id,name from table) tab (有别名) 正确执行

Select \* from (select id,name from table) as tab (使用as取别名) 错误

**在MySQL 中**：

Select \* from (select id,name from table) (无别名) 错误

Select \* from (select id,name from table) tab (有别名) 正确执行

Select \* from (select id,name from table) as tab (使用as取别名) 正确执行

### 2．兼容修改方式

根据上面的结论，为了让Oracle和MySQL兼容，我们采用第中形式的嵌套语句，即使用表别名并不用as：

Select \* from (select id,name from table) tab

## （七）、全连接（full join）兼容修改

### 1．出现问题形式

full join是数据库全连接语句，在Oracle提供这样的语法，而在MySQL中没有支持。使用 full join的sql语句在MySQL环境下就出现了兼容问题。

select \* from t1 full join t2 on t1.id=t2.id

### 2．兼容修改方法

为了使MySQL/Oracle数据库同时兼容，我们将full join 使用  UNION 结合left join right join 的方式实现。

例如：

1. 对于两张表t1,t2:  
   SELECT \* FROM t1  
   LEFT JOIN t2 ON t1.id = t2.id  
   UNION  
   SELECT \* FROM t1  
   RIGHT JOIN t2 ON t1.id = t2.id
2. 对于三张表t1,t2,t3

SELECT \* FROM t1

LEFT JOIN t2 ON t1.id = t2.id

LEFT JOIN t3 ON t2.id = t3.id

UNION

SELECT \* FROM t1

RIGHT JOIN t2 ON t1.id = t2.id

LEFT JOIN t3 ON t2.id = t3.id

UNION

SELECT \* FROM t1

RIGHT JOIN t2 ON t1.id = t2.id

RIGHT JOIN t3 ON t2.id = t3.id

其中，t1,t2,t3不仅可以是一个实际的表，还可以是一个查询结果

## （八）、日期加减天数兼容修改

### 1．出现问题形式

有的需求是通过表中的两个时间字段进行加减比较。

例如：

Select \* from table where date1+2 >date2

date1、date2是数据表中的两个日期类型字段。

在Oracle中日期增加或者减去几天，可以直接使用“+”或“-”运算符直接计算；

在MySQL中操作日期的加减需要使用函数进行:

DATE\_ADD(date,  interval 天数 day),

DATE\_SUB(date,  interval 天数 day)

### 2．兼容修改方法

这里，很难找到一个公共的方法能让MySQL和Oracle同时兼容，因此我们采用了区分MySQL和Oracle的方式来分别处理不同环境下的查询语句。通过读取配置文件数据资源配置的方式进行区分，当前环境采用的数据库。

在Oracle中日期增加或者减去几天，可以直接使用“+”或“-”运算符直接计算；

在MySQL中操作日期的加减使用函数进行:

DATE\_ADD(date,  interval 天数 day),

DATE\_SUB(date,  interval 天数 day)

例如：

ClassPathResource resource =

**new** ClassPathResource("datasource.properties");

Properties props = PropertiesLoaderUtils.*loadProperties*(resource);

String driverFullName =

props.getProperty("jdbc.master.driverClassName");

String sql="";

**if**(driverFullName.contains("oracle")){

sql = "Select \* from table where date1+?(天数) >date2";

} **else if(**driverFullName.contains("mysql")){

sql = "select \* from table where DATE\_ADD(date1, INTERVAL ?(天数) DAY)>date2";

}