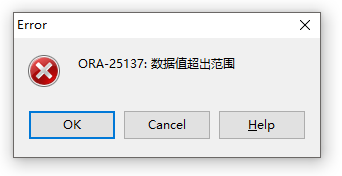
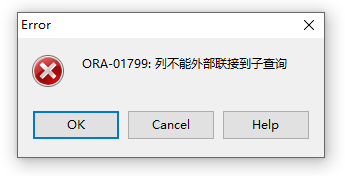
SQL适配文档(兼容MySql和Oracle)

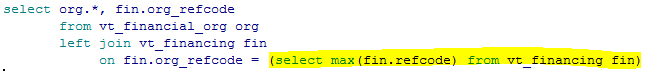
正常

* + **COALESCE(exp……)**
    - 返回第一个不为空的值，如果全为NULL，则返回空字符串。
      * 例1：SELECT COALESCE(NULL,NULL,’q’) FROM DUAL;，执行后的结果为q。
      * 例2：SELECT COALESCE(NULL,NULL,NULL) FROM DUAL;，执行后的结果为空字符串。
      * 例3：SELECT COALESCE(NULL,NULL,’q’,NULL,NULL) FROM DUAL;，执行后的结果为q。
  + **TO\_CHAR(date, exp)**
    - YYYY 年份
      * 例1：SELECT TO\_CHAR(NOW(), ‘YYYY’) FROM DUAL;，假设NOW()的值为2017-05-26 14:22:10，则执行后的结果为2017。
    - Q 季度
      * 例1：SELECT TO\_CHAR(NOW(), ‘Q’) FROM DUAL;，假设NOW()的值为2017-05-26 14:22:10，则执行后的结果为2，表示第二季度。
    - MM 月份
      * 例1：SELECT TO\_CHAR(NOW(), ‘MM’) FROM DUAL;，假设NOW()的值为2017-05-26 14:22:10，则执行后的结果为05。
    - D 星期(每周的星期天为第一天)
      * 例1：SELECT TO\_CHAR(NOW(), ‘D’) FROM DUAL;，假设NOW()的值为2017-05-26 14:22:10，则执行后的结果为6。
    - DD 日期
      * 例1：SELECT TO\_CHAR(NOW(), ‘DD’) FROM DUAL;，假设NOW()的值为2017-05-26 14:22:10，则执行后的结果为26。
    - yyyy-MM-dd 年月日
      * 例1：SELECT TO\_CHAR(NOW(), ‘yyyy-MM-dd’) FROM DUAL;，假设NOW()的值为2017-05-26 14:22:10，则执行后的结果为2017-05-26。
  + **TO\_DATE(dateStr, exp)**
    - yyyy-MM-dd 年月日
      * 例1：SELECT TO\_DATE (‘2017-05-26 14:22:10’, ‘yyyy-MM-dd’) FROM DUAL;，执行后的结果为DATE类型2017-05-26。
    - yyyy-MM-dd hh:mi:ss 年月日时分秒
      * 例1：SELECT TO\_DATE(‘2017-05-26 14:22:10’, ‘yyyy-MM-dd hh:mi:ss’) FROM DUAL;，执行后的结果为DATE类型2017-05-26 2:22:10。
    - yyyy-MM-dd hh24:mi:ss 年月日时分秒
      * 例1：SELECT TO\_DATE(‘2017-05-26 14:22:10’, ‘yyyy-MM-dd hh24:mi:ss’) FROM DUAL;，执行后的结果为DATE类型2017-05-26 14:22:10。
  + **ADDDATE(date, num)**
    - 日期运算，如果为正数，则添加，如果为负数，则减少
      * 例1：SELECT ADDDATE(SYSDATE, -6) FROM DUAL;，假设SYSDATE的值为2017-05-26，则执行运算后的值为2017-05-20。
      * 例2：SELECT ADDDATE(SYSDATE, 4) FROM DUAL;，假设SYSDATE的值为2017-05-26，则执行运算后的值为2017-05-30。
  + **NOW()**
    - 获取当前时间(含时分秒)
      * 例1：SELECT NOW() FROM DUAL;，执行后将得到值2017-05-26 14:22:10。
  + **CAST(exp as type)**
    - 格式化值为指定类型
      * 例1：SELECT CAST(‘800’ as CHAR);，执行后得到字符800。建议写成AS CHAR。
  + **LEFT(exp, num)**
    - 取exp下标0到num之间的字符串。
      * 例1：SELECT LEFT(‘ABCDE’, 3) FROM DUAL;，执行后将得到值ABC。
  + **RAND()**
    - 取随机数
      * 例1：SELECT RAND() FROM DUAL;，执行后将获取一个0到1之间的随机数。
  + **GROUP\_CONCAT()**
    - 该函数返回带有来自一个组的连接的非NULL值的字符串结果。
  + **CONCAT(exp,exp)**
    - 改函数只允许有两个参数，以同时兼容MySQL和Oracle。

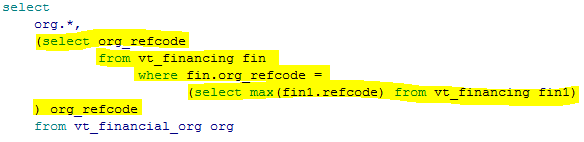
警告

1. SQL入参时请注意大小写，在Oracle中值判断是区分大小写的。
   1. 例如：select columns from table where column1 in **(‘a’, ‘b’)** 和 select columns from table where column1 in **(‘A’, ‘B’)** 是有区别的。
2. 使用CAST函数时类型一定要指定长度，在Oracle中不指定长度会抛出异常。
   1. 例如：select cast(‘1234’ as **char**) from dual; 在Oracle中执行该语句会抛出以下异常信息：
      1. 
         1. 解决方案：应改为 select cast(‘1234’ as **varchar(9)**) from dual;
3. 不要使用 **LIMIT** ，可用物理分页取代。
4. LEFT JOIN .. **ON .. 条件子句**中可以出现已预知参数，但不允许出现子查询，否则在Oracle数据库中会抛出以下异常信息：
   1. 
      1. 解决方案：不要使用LEFT JOIN，在查询列中改用子查询的方式。

例1：



应该替换为以下语句：



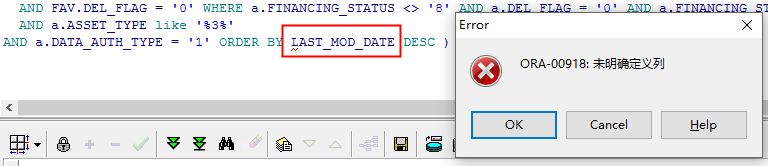
1. 自定义Mapper.xml文件中，如果传入的参数的字段为null对象，此时MyBatis是无法获取对应的jdbcType类型。MyBatis会抛出以下异常：
   1. Java.sql.SQLException:无效的列类型:1111;error code[17004]
      1. 解决方案：手动指定参数类型。
         1. 例1：#{orgRefcode} 应该替换为 #{refcode, jdbcType=INTEGER}

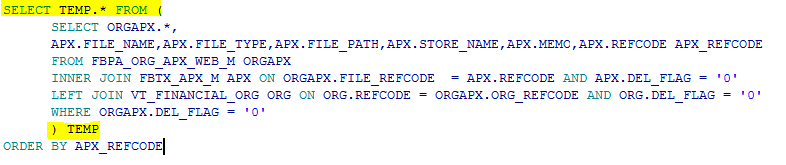


1. 严禁使用**DATE\_ADD(exp,val)**函数，应统一替换为**ADDDATE(exp,val)**。
2. 设计表字段时请注意字段长度，Oracle数据库中字段的长度不能大于30个字符。

错误

1. 如果页面查询出现错误：**无效的列类型1111**，请检查SQL语句中参数占位符是否指定了数据类型。请参考**警告6**。
   1. 错误示例：
2. 如果SQL语句正常，但是在Oracle环境下查询不出记录，请检查SQL语句中是否出现以下情况：
   1. SQL语句中条件子句中的列名没有指定表别名，应在最外层包裹一层。





1. **ORA-00909: 参数个数无效**：通常原因是因为SQL语句中使用了CONCAT函数，并且参数大于2个。
   1. 例1：**SELECT CONCAT(‘%’, ‘1’, ‘%’) FROM DUAL**；在MySql环境中可以正常执行，但是在Oracle环境中执行该语句会抛出ORA-00909异常。建议请在程序中处理这种逻辑性的问题。