Atitit jpql ast总结v2 t025.docx

目录

[1.1. 多select字段 1](#_Toc27322)

[1.2. 多个and条件 （ok） 2](#_Toc4049)

[1.3. Select 字段函数（聚合等） 2](#_Toc15831)

[1.4. [ORDER BY 子句] 排序 2](#_Toc19733)

[1.5. 翻页语句 貌似不支持 3](#_Toc16345)

[1.6. 运算符 》 《 》 like 3](#_Toc193)

[2. 相对复杂的表达式 4](#_Toc16992)

[2.1. 别名 as （实现解析） 4](#_Toc16559)

[2.2. JOIN关键字可以省略，上式等同於： 5](#_Toc19370)

[2.3. HAVING 子句 5](#_Toc27311)

[2.4. Select 聚合函数 （实现解析） 5](#_Toc6320)

[2.5. Udf （实现解析） 6](#_Toc19635)

[2.6. Groupby （实现了） 8](#_Toc6406)

## 多select字段

\-[QUERY] Node: 'query'

+-[SELECT\_FROM] Node: 'SELECT\_FROM'

| +-[FROM] Node: 'from'

| | \-[RANGE] Node: 'RANGE'

| | \-[IDENT] Node: '\u8868\u683c1'

| \-[SELECT] Node: 'select'

| +-[IDENT] Node: '\u5b57\u6bb52'

| \-[IDENT] Node: '\u5b57\u6bb53'

\-[WHERE] Node: 'where'

\-[EQ] Node: '='

+-[IDENT] Node: '\u5b57\u6bb51'

\-[NUM\_INT] Node: '123'

SELECT \* FROM 字段3 WHERE 字段1 = 123

## 多个and条件 （ok）

## Select 字段函数（聚合等）

## [ORDER BY 子句] 排序

jpql = "select a,b from 表格1 where c=123 and d =456 order by c1 desc ,c2 ,c3 desc";

多个orderby

\-[ORDER] Node: 'order'

+-[IDENT] Node: 'c1'

+-[DESCENDING] Node: 'desc'

+-[IDENT] Node: 'c2'

+-[IDENT] Node: 'c3'

\-[DESCENDING] Node: 'desc'

+-[SELECT\_FROM] Node: 'SELECT\_FROM'

| +-[FROM] Node: 'from'

| | \-[RANGE] Node: 'RANGE'

| | \-[IDENT] Node: '\u8868\u683c1'

| \-[SELECT] Node: 'select'

| +-[IDENT] Node: '\u5b57\u6bb52'

| \-[IDENT] Node: '\u5b57\u6bb53'

+-[WHERE] Node: 'where'

| \-[EQ] Node: '='

| +-[IDENT] Node: '\u5b57\u6bb51'

| \-[NUM\_INT] Node: '123'

\-[ORDER] Node: 'order'

\-[IDENT] Node: 'c1'

\-[QUERY] Node: 'query'

+-[SELECT\_FROM] Node: 'SELECT\_FROM'

| +-[FROM] Node: 'from'

| | \-[RANGE] Node: 'RANGE'

| | \-[IDENT] Node: '\u8868\u683c1'

| \-[SELECT] Node: 'select'

| +-[IDENT] Node: '\u5b57\u6bb52'

| \-[IDENT] Node: '\u5b57\u6bb53'

+-[WHERE] Node: 'where'

| \-[EQ] Node: '='

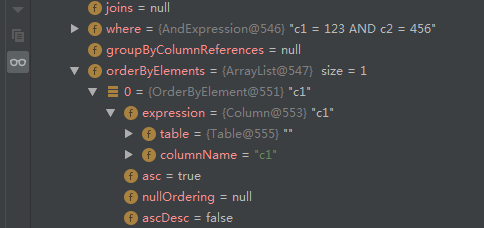
| +-[IDENT] Node: '\u5b57\u6bb51'

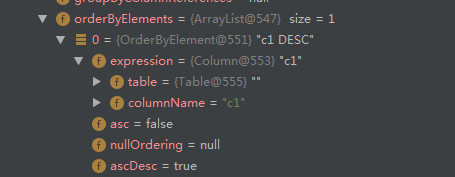
| \-[NUM\_INT] Node: '123'

\-[ORDER] Node: 'order'

+-[IDENT] Node: 'c1'

\-[DESCENDING] Node: 'desc'





## 翻页语句 貌似不支持

直接后面耕者mysql 翻页表达式得了

System.*out*.println(JqplUtil.*Jqpl2sql*(hql,"limit 1"));

## 运算符 》 《 》 like

　其中IN中指定的，就是群集成员路径表示，而>、<、AND、IN、LIKE等都是WHERE子句中条件表示式，简单列出一些条件表示式如下：

|  |  |
| --- | --- |
| 比较陈述 | =、>、>=、<、<=、<> |
| BETWEEN 陈述 | [NOT BETWEEN |
| LIKE 陈述 | [NOT] LIKE |
| IN 陈述 | [NOT] IN |
| NULL 陈述 | IS [NOT] NULL |
| EMPTY 陈述 | IS [NOT] EMPTY |
| EXISTS 陈述 | [NOT] EXISTS |

# 相对复杂的表达式

## 别名 as （实现解析）

\-[QUERY] Node: 'query'

+-[SELECT\_FROM] Node: 'SELECT\_FROM'

| +-[FROM] Node: 'from'

| | \-[RANGE] Node: 'RANGE'

| | \-[IDENT] Node: '\u8868\u683c1'

| \-[SELECT] Node: 'select'

| +-[IDENT] Node: '\u5b57\u6bb52'

| +-[IDENT] Node: '\u5b57\u6bb53'

| +-[AS] Node: 'as'

| | +-[AGGREGATE] Node: 'sum'

| | | \-[IDENT] Node: 'id'

| | \-[IDENT] Node: 'idAlias1'

| +-[AS] Node: 'as'

| | +-[METHOD\_CALL] Node: '('

| | | +-[IDENT] Node: 'kk'

| | | \-[EXPR\_LIST] Node: 'exprList'

| | | \-[NUM\_INT] Node: '5'

| | \-[IDENT] Node: 'alias2'

| \-[COUNT] Node: 'count'

| \-[ROW\_STAR] Node: '\*'

+-[WHERE] Node: 'where'

| \-[EQ] Node: '='

| +-[IDENT] Node: '\u5b57\u6bb51'

| \-[NUM\_INT] Node: '123'

\-[ORDER] Node: 'order'

+-[IDENT] Node: 'c1'

\-[DESCENDING] Node: 'desc'

\-[QUERY] Node: 'query'

+-[SELECT\_FROM] Node: 'SELECT\_FROM'

| +-[FROM] Node: 'from'

| | \-[RANGE] Node: 'RANGE'

| | \-[IDENT] Node: '\u8868\u683c1'

| \-[SELECT] Node: 'select'

| +-[IDENT] Node: '\u5b57\u6bb52'

| +-[IDENT] Node: '\u5b57\u6bb53'

| +-[AGGREGATE] Node: 'sum'

| | \-[IDENT] Node: 'id'

| +-[AS] Node: 'as'

| | +-[METHOD\_CALL] Node: '('

| | | +-[IDENT] Node: 'kk'

| | | \-[EXPR\_LIST] Node: 'exprList'

| | | \-[NUM\_INT] Node: '5'

| | \-[IDENT] Node: 'alias2'

| \-[COUNT] Node: 'count'

| \-[ROW\_STAR] Node: '\*'

+-[WHERE] Node: 'where'

| \-[EQ] Node: '='

| +-[IDENT] Node: '\u5b57\u6bb51'

| \-[NUM\_INT] Node: '123'

\-[ORDER] Node: 'order'

+-[IDENT] Node: 'c1'

\-[DESCENDING] Node: 'desc'

Exception in thread "main" java.lang.ClassCastExcepti

## JOIN关键字可以省略，上式等同於：

## HAVING 子句

## Select 聚合函数 （实现解析）

\-[QUERY] Node: 'query'

+-[SELECT\_FROM] Node: 'SELECT\_FROM'

| +-[FROM] Node: 'from'

| | \-[RANGE] Node: 'RANGE'

| | \-[IDENT] Node: '\u8868\u683c1'

| \-[SELECT] Node: 'select'

| \-[AGGREGATE] Node: 'sum'

| \-[IDENT] Node: '\u6570\u91cf'

\-[WHERE] Node: 'where'

\-[EQ] Node: '='

+-[IDENT] Node: '\u5b57\u6bb51'

\-[NUM\_INT] Node: '123'

\-[QUERY] Node: 'query'

+-[SELECT\_FROM] Node: 'SELECT\_FROM'

| +-[FROM] Node: 'from'

| | \-[RANGE] Node: 'RANGE'

| | \-[IDENT] Node: '\u8868\u683c1'

| \-[SELECT] Node: 'select'

| +-[IDENT] Node: '\u5b57\u6bb52'

| +-[IDENT] Node: '\u5b57\u6bb53'

| +-[AGGREGATE] Node: 'sum'

| | \-[IDENT] Node: 'id'

| +-[AS] Node: 'as'

| | +-[METHOD\_CALL] Node: '('

| | | +-[IDENT] Node: 'kk'

| | | \-[EXPR\_LIST] Node: 'exprList'

| | | \-[NUM\_INT] Node: '5'

| | \-[IDENT] Node: 'alias2'

| \-[COUNT] Node: 'count'

| \-[ROW\_STAR] Node: '\*'

+-[WHERE] Node: 'where'

| \-[EQ] Node: '='

| +-[IDENT] Node: '\u5b57\u6bb51'

| \-[NUM\_INT] Node: '123'

\-[ORDER] Node: 'order'

+-[IDENT] Node: 'c1'

\-[DESCENDING] Node: 'desc'

## Udf （实现解析）

\-[QUERY] Node: 'query'

+-[SELECT\_FROM] Node: 'SELECT\_FROM'

| +-[FROM] Node: 'from'

| | \-[RANGE] Node: 'RANGE'

| | \-[IDENT] Node: '\u8868\u683c1'

| \-[SELECT] Node: 'select'

| +-[IDENT] Node: '\u5b57\u6bb52'

| +-[IDENT] Node: '\u5b57\u6bb53'

| +-[AGGREGATE] Node: 'sum'

| | \-[IDENT] Node: 'id'

| +-[AS] Node: 'as'

| | +-[COUNT] Node: 'count'

| | | \-[ROW\_STAR] Node: '\*'

| | \-[IDENT] Node: 'cntAlias'

| \-[AS] Node: 'as'

| +-[METHOD\_CALL] Node: '('

| | +-[IDENT] Node: 'Udf'

| | \-[EXPR\_LIST] Node: 'exprList'

| | \-[IDENT] Node: 'c11'

| \-[IDENT] Node: 'udfC11As'

+-[WHERE] Node: 'where'

| \-[EQ] Node: '='

| +-[IDENT] Node: '\u5b57\u6bb51'

| \-[NUM\_INT] Node: '123'

\-[ORDER] Node: 'order'

+-[IDENT] Node: 'c1'

\-[DESCENDING] Node: 'desc'

\-[QUERY] Node: 'query'

+-[SELECT\_FROM] Node: 'SELECT\_FROM'

| +-[FROM] Node: 'from'

| | \-[RANGE] Node: 'RANGE'

| | \-[IDENT] Node: '\u8868\u683c1'

| \-[SELECT] Node: 'select'

| +-[IDENT] Node: '\u5b57\u6bb52'

| +-[IDENT] Node: '\u5b57\u6bb53'

| +-[AGGREGATE] Node: 'sum'

| | \-[IDENT] Node: 'id'

| +-[AS] Node: 'as'

| | +-[COUNT] Node: 'count'

| | | \-[ROW\_STAR] Node: '\*'

| | \-[IDENT] Node: 'cntAlias'

| \-[METHOD\_CALL] Node: '('

| +-[IDENT] Node: 'Udf'

| \-[EXPR\_LIST] Node: 'exprList'

| \-[IDENT] Node: 'c11'

+-[WHERE] Node: 'where'

| \-[EQ] Node: '='

| +-[IDENT] Node: '\u5b57\u6bb51'

| \-[NUM\_INT] Node: '123'

\-[ORDER] Node: 'order'

+-[IDENT] Node: 'c1'

\-[DESCENDING] Node: 'desc'

Dbg

-[QUERY] Node: 'query'

+-[SELECT\_FROM] Node: 'SELECT\_FROM'

| +-[FROM] Node: 'from'

| | \-[RANGE] Node: 'RANGE'

| | \-[IDENT] Node: '\u8868\u683c1'

| \-[SELECT] Node: 'select'

| +-[IDENT] Node: '\u5b57\u6bb52'

| +-[IDENT] Node: '\u5b57\u6bb53'

| +-[AGGREGATE] Node: 'sum'

| | \-[IDENT] Node: 'id'

| +-[AS] Node: 'as'

| | +-[COUNT] Node: 'count'

| | | \-[ROW\_STAR] Node: '\*'

| | \-[IDENT] Node: 'cntAlias'

| \-[METHOD\_CALL] Node: '('

| +-[IDENT] Node: 'Udf'

| \-[EXPR\_LIST] Node: 'exprList'

| \-[IDENT] Node: 'c11'

+-[WHERE] Node: 'where'

| \-[EQ] Node: '='

| +-[IDENT] Node: '\u5b57\u6bb51'

| \-[NUM\_INT] Node: '123'

\-[ORDER] Node: 'order'

+-[IDENT] Node: 'c1'

\-[DESCENDING] Node: 'desc'

## Groupby （实现了）

Atitit 不错的ql选型 jpql

Atitit 常见数据查询语言 GraphQL sql