

OQL From 子句

1 简介

FROM 子句指定必须从中检索数据的实体或其他源。该子句以 FROM 关键字开头，后随实体名称。要同时从其他实体中选择数据，通过 JOIN 关键字添加这些实体。该语法比正式的 SQL FROM 子句语法严格一些。

以下是完整语法的示例：

```
FROM
  {
    entity_name | ( sub_oql_query )
  }
  [ [ AS ] from_alias ]

  {
    { INNER | { { LEFT | RIGHT | FULL } [ OUTER ] } } JOIN
    entity_path [ [ AS ] from_alias ]
    [ ON <constraint> ]
  } [ ,...n ]
```

2 entity_name

这将指定必须从中检索数据的实体。可选择以双引号将实体名称括入。如果实体名称是保留的 OQL 词语（如 Order 或 Group），则必须使用双引号。

3 (sub_oql_query)

这将指定另一个必须从中检索数据的 OQL 查询。这将是当前查询的源。子查询必须放在括号内。

4 JOIN（联接）

这将指定要联接至该查询的实体的路径。支持四种不同类型的联接：

- 内部联接
- 左外联接
- 右外联接

- 完整联接

语法如下：

```
{ INNER | { { LEFT | RIGHT | FULL } [ OUTER ] } } JOIN  
    entity_path [ [ AS ] from_alias ]  
    [ ON <constraint> ]
```

4.1 entity_path

这将指定要联接的实体以及从 FROM 子句中先前定义的实体到该实体的路径。

路径 `Crm.Customer/Crm.Customer_Address/Crm.Address` 定义从先前定义的实体 **Crm.Customer** 到新实体 **Crm.Address** 的路径。

类似于 `entity_name`，可以使用双引号。

4.2 [ON <constraint>]

这将约束 FROM 子句的 JOIN 部分中的指定实体。约束语法类似于 WHERE 子句。只有当前和前面的 JOIN 元素的实体和 from 别名可在约束中使用。

该部分为可选。系统将根据指定的 `entity_path` 生成适当的联接条件。

4.3 联接类型

4.3.1 INNER JOIN (内部联接)

INNER JOIN 是实体之间最常见的联接操作，表示默认联接类型。该查询将实体 A 的每一行与实体 B 的每一行进行比较，以找到具有关联且满足 JOIN 谓词的所有行对。如果关联存在且满足 JOIN 谓词，则将 A 和 B 的每对匹配行的列值组合成一个结果行。

语法如下：

```
[ INNER ] JOIN entity_path [ ON <constraint> ]
```

4.3.2 LEFT OUTER JOIN (左外联接)

使用 LEFT OUTER JOIN 构造，查询将实体 A 的每一行与实体 B 的每一行进行比较，以找到具有关联的所有行对，从而满足 JOIN 谓词。当存在关联且满足 JOIN 谓词时，则将 A 和 B 每对匹配行的列值组合成一个结果行。

但是，与 INNER JOIN 构造不同的是，查询还将返回与实体 B 不匹配的实体 A 的行。指定实体 B 的列时，这些列包含这些行的空值。

语法如下：

```
LEFT [ OUTER ] JOIN entity_path [ ON <constraint> ]
```

4.3.3 RIGHT OUTER JOIN (右外联接)

使用 RIGHT OUTER JOIN 构造，查询将实体 A 的每一行与实体 B 的每一行进行比较，以找到具有关联的所有行对，从而满足 JOIN 谓词。如果关联存在且满足 JOIN 谓词，则将 A 和 B 的每对匹配行的列值组合成一个结果行。

但是，与 INNER JOIN 构造不同的是，还将返回实体 B 中与实体 A 不匹配的行。指定实体 A 的列时，这些列将包含这些行的空值。

语法如下：

```
RIGHT [ OUTER ] JOIN entity_path [ ON <constraint> ]
```

4.3.4 FULL OUTER JOIN (完整外部联接)

使用 FULL OUTER JOIN 构造，查询将实体 A 的每一行与实体 B 的每一行进行比较，以找到具有关联的所有行对，从而满足联接谓词。当存在关联且满足联接谓词时，则将 A 和 B 每对匹配行的列值组合成一个结果行。

但是，与 INNER JOIN 构造不同的是，还将返回不匹配的实体的数据。对于这些行，缺少实体的列将包含空值。

语法如下：

```
FULL [ OUTER ] JOIN entity_path [ ON <constraint> ]
```

4.4 示例

在这种情况下，将使用 LEFT OUTER JOIN 来获取表 A 中与表 B 没有关联的记录。

例如，有两个实体**客户**和**订单**，其中客户可以与多个订单关联。需要检索没有任何订单的所有客户。

```
SELECT
  Customer/Name as Name,
  Customer/<anyotherattribute> as <anyotherattribute>
FROM MyModule.Customer
  LEFT OUTER JOIN Customer/MyModule.Customer_Order/MyModule.Order as Order
WHERE Order/ID IS NULL
```