

已批准
ska 26-07-19, 01:39

外键: integer("user_id").reference () => users.id);

在 drizzle 中, query 语法使用 V2, 本心得中所有均用最新的 V2 规范

define Relations :

```
export const relations = defineRelations (
  { table1, table2, ... }, 有哪些表与关联
  (r) => ({  r包含了所有表的引用
    ★ 表名: {
      ★ 关系名: r.one. 另外一张表名 (
        from: r.表名1.字段名
        to: r.表名2.字段名
      })
    }, ...
  })
```

★ 值得注意: drizzle 并不要求 from、to 之间是外键关系

因此, from : xx
to : xx } 这俩可以任意指定, 随心所欲

即 relation 是属于应用层,
用于 query 自动 JOIN

but 为了工程数量, 仍建议同时用 relation、外键

假如表A定义了关联表B的 relation

这个 relation 只能在表A query 的时候用

表B需自己定义一个 relation、不可共享

drizzle 的类型

方式一: 表名.\$infer Select 查询得的数据的类型, 全部可选

表名.\$infer Insert 有默认值的可选

方式二: InferSelectModel <表名>

query 语法

=、AND、OR、NOT、都可在 where、with 里用

where : {

id : 1

}

with : {

关系名 : {

where : {

另一张表的字段名 : {

xxx : xxx

}

}

}

和上面一样

建议在 schema 中一并导出方便使用

export type ... = typeof 表名.\$infer ;

where 里可用的操作符

直接操作 : { OR : [0, 1, ...]
AND
NOT : { }
RAW : (表) => sql` \${table.id} = 1 ` 即能 NJS

字段名操作 :

XXX : {

两种

①

②

OR: [], AND: [], NOT: {}, RAW: (table) => sql`\${table.id} = 1`

字段名: {

OR: [], AND: [], NOT: {}, eq: 1, ne: 1, gt: 1, gte: 1, lt: 1, lte: 1,

in: [1], notIn: [1],

like: "", notLike: "",

isNull: true, isNotNull: true,

}

}

如:

where : {

name : {

like : "John%"

}

}

据文档所述

在没有 with 的时候,

where 中可使用 关系名 : { 操作

可实现根据另一张表的值进行过滤

但可不把另一张表数据放入结果, 即不with

1. 官方建议 query 多表关系, 双表的键要加索引

2. (1) 多对多索引: 建议中间表用复合键 + 索引也加索引, (即三个索引, 中间表)

3. 外键在 query 中不是必须的, 因为 query 的 relation 是软件层的