II. SQL 写入链路实现

写入链路

本节，我们将讲述简单的 INSERT 语句是如何运行的。

```
create table t(id int primary key, val int);
insert into t values(1, 1), (10, 10), (100, 100); -- simple insert
insert into t(id, val) select id + 1, val + 1 from t; -- insert from select
```

这是一条 INSERT 语句，会通过 `executor/insert.go` 中的 `InsertExec` 来执行。

- 1. `executor/builder.go`， `executorBuilder.buildInsert` 函数会构造 `InsertExec`， `InsertExec` 的结构体定义中组合了 `InsertValues`。在构造时，会通过 `InsertValues.initInsertColumns` 生成执行所需要涉及到的 Columns 信息。
- 2. `executor/insert.go`， `InsertExec.Open` 方法会被调用，有的 Insert 是根据 Select 的结果写入的（如上面的第二条 Insert），这种情况下 Insert 中嵌入了一条 Select 语句， `InsertExec` 中也嵌入了一个 `SelectionExec`，在 `Open` 的时候也需要通过 `SelectionExec.Open` 初始化 `SelectionExec`。
- 3. `executor/insert.go`， `InsertExec.Next` 中对普通的 Insert 和根据 Select 的 Insert 会调用不同的函数。
    - 3.1 `executor/insert.go`，普通的 Insert 会使用 `insertRows` 函数进行处理（例子中第一条 Insert）。
    - 3.2 `executor/insert.go`，根据 Select 的 Insert 会使用 `insertRowsFromSelect` 函数进行处理（例子中第二条 Insert）。
- 4. `executor/insert.go`， `insertRows` 和 `insertRowsFromSelect` 都会使用 `InsertExec.exec` 来处理实际写入的数据， `InsertExec.exec` 中，每行数据都会使用被组合的 `InsertValues.addRecord` 进行写入。
- 5. `executor/insert_common.go`， `InsertValues.addRecord` 会将输入的一行数据通过 `table/tables/tables.go` 中的 `TableCommon.AddRecord` 函数写入到 membuffer 当中。

以上是写入的关键路径，这个调用链路中的关键函数也被移除了，你需要根据调用链路的描述进行填充。