Необходимо реализовать конструктор Multiple INSERT запросов к БД.

Проблема, следующая: к нам, приходит достаточно большое количество данных, которые нужно вставить в разные базы данных, например - PostgreSQL или MSSQL.

Конструктор запросов должен съесть предположим 2000 вставок, и объединить их в ровно то количество INSERT`ов, сколько удовлетворят входным требованиям, это количество вставок ($1,$2,$3) и количестве параметров ($1) в 1 запросе.

Разные базы данных имеют различные ограничения на количество вставляемых **строк** и **параметров** за 1 раз. Например, MSSQL не позволит вставить более 2100 строк, а PostgreSQL может вставить до 65К параметров, т.е у него нет ограничений на вставку строк, но при этом есть ограничение на параметры. Требуется вынести ограничение максимального количество строк и параметров в константы, либо передавать эти параметры в конструктор какой-то функции.

Если превышена длина вставляемых строк или параметров, необходимо отправить несколько SQL запросов.

Ниже приложил интерфейс конструктора и маленький пример с комментариями:

**package** main  
  
**import** (  
 "database/sql"  
)  
  
**type** BatchQuery **struct** {  
 Query string  
 Args []**interface**{}  
}  
  
**type** InsertBuilder **interface** {  
 Append(args ...**interface**{})

// BatchQuery массив на случай если будет превышено кол-во строк или параметров   
 ToSQL() ([]BatchQuery, error)  
}  
  
**func** BulkDevice(db \*sql.DB) {  
 rows := []**struct** {  
 Name string  
 GroupID uint  
 PlatformID uint  
 }{  
 {Name: "device 1", GroupID: 1, PlatformID: 5281},  
 {Name: "device 2", GroupID: 2, PlatformID: 5281},  
 {Name: "device 3", GroupID: 3, PlatformID: 5281},  
 }  
  
 // Здесь подсовывается имплементация интерфейса InsertBuilder  
 **var** b InsertBuilder  
 **for** \_, r := **range** rows {  
 b.Append(r.Name, r.GroupID, r.PlatformID)  
 }  
  
 batches, err := b.ToSQL()  
 **if** err != nil {  
 // do something  
 }  
 **for** \_, b := **range** batches {  
 \_, err = db.Exec(  
 "INSERT INTO devices ('name', 'group\_id', 'platform\_id') VALUES "+b.Query,  
 b.Args...,  
 )  
 **if** err != nil {  
 // do something  
 }  
  
 // на выходе должно было что-то вроде  
 // INSERT INTO devices ('name', 'group\_id', 'platform\_id') VALUES ($1,$2,$3), ($4,$5,$6), ($7,$8,$9)  
 }  
}

Можно просто реализовать интерфейс InsertBuilder, чтобы не придумывать сигнатуру методов.