



```
PARSERCONTEXT
typedef struct ParserContext {
    Query * ssql;
    size_t select_length;
    size_t condition_length;
    size_t from_length;
    size_t value_length;
    Value values[MAX_NUM];
    Condition conditions[MAX_NUM];
    CompOp comp;
    char id[MAX_NUM];
} ParserContext;
```

```
QUERY
typedef struct Query {
    enum SqlCommandFlag flag;
    union Queries sstr;
} Query;
```

```
QUERIES
union Queries {
    Selects selection; // SELECT
    Inserts insertion; // INSERT
    Deletes deletion; // DELETE
    Updates update; // UPDATE
    CreateTable create_table; // CREATE
    DropTable drop_table; // DROP
    CreateIndex create_index; // CREATE INDEX
    DropIndex drop_index; // DROP INDEX
    DescTable desc_table; // DESC (utf?)
    LoadData load_data; // LOAD (utf?)
    char *errors;
};
```

```
SELECTS
typedef struct {
    size_t attr_num; // Length of attrs in Select clause
    RelAttr attributes[MAX_NUM]; // attrs in Select clause
    size_t relation_num; // Length of relations in Fro
    clause
    char * relations[MAX_NUM]; // relations in From clause
    size_t condition_num; // Length of conditions in Where
    clause
    Condition conditions[MAX_NUM]; // conditions in Where clause
} Selects;
```

```
DELETES
typedef struct {
    char *relation_name; // Relation to delete from
    size_t condition_num; // Length of conditions in Where
    clause
    Condition conditions[MAX_NUM]; // conditions in Where clause
} Deletes;
```

```
UPDATES
typedef struct {
    char *relation_name; // Relation to update
    char *attribute_name; // Attribute to update
    Value value; // update value
    size_t condition_num; // Length of conditions in Where
    clause
    Condition conditions[MAX_NUM]; // conditions in Where clause
} Updates;
```

```
DROPTABLE
typedef struct {
    char *relation_name; // Relation name
} DropTable;
```

```
CREATEINDEX
typedef struct {
    char *index_name; // Index name
    char *relation_name; // Relation name
    char *attribute_name; // Attribute name
} CreateIndex;
```

```
DROPTABLE
typedef struct {
    const char *index_name; // Index name
} DropIndex;
```

```
DESCSTABLE
typedef struct {
    const char *relation_name;
    const char *file_name;
} DescTable;
```

```
LOADDATA
typedef struct {
    const char *relation_name;
    const char *file_name;
    LoadData;
}
```

```
CONDITION
typedef struct _Condition {
    int blhsIsAttr; // TRUE if left-hand side is an attribute
    // 1时, 操作符右边是属性, 0时, 是属性值
    Value lhsValue; // left-hand side value if blhsIsAttr = FALSE
    RelAttr lhsAttr; // left-hand side attribute
    CompOp op; // comparison operator
    int brhsIsAttr; // TRUE if right-hand side is an attribute
    // 1时, 操作符右边是属性, 0时, 是属性值
    // and not a value
    RelAttr rhsAttr; // right-hand side attribute if brhsIsAttr =
    TRUE ?????
    Value rhsValue; // right-hand side value if brhsIsAttr = FALSE
} Condition;
```

```
INSERTS
typedef struct {
    char *relation_name; // Relation to insert into
    size_t value_num; // Length of values
    Value values[MAX_NUM]; // values to insert
} Inserts;
```

```
CREATETABLE
typedef struct {
    char *relation_name; // Relation name
    size_t attribute_count; // Length of attribute
    AttrInfo attributes[MAX_NUM]; // attributes
} CreateTable;
```

```
RELATTR
typedef struct {
    char *relName; // relation name (may be NULL) 表名
    char *attrName; // attribute name 属性名
} RelAttr;
```

```
COMPOP
typedef enum {
    Equal, // "=" 1
    LEqual, // "<=" 2
    NEqual, // "<>" 3
    Less1, // "<" 4
    GEqual, // ">=" 5
    Great1, // ">" 6
    NO_OP
} CompOp;
```

```
VALUE
struct _Value {
    AttrType type; // type of value
    void *data; // value
};
```

```
ATTRINFO
typedef struct {
    char *name; // Attribute name
    AttrType type; // type of attribute
    size_t length; // Length of attribute
} AttrInfo;
```

```
ATTRTYPE
typedef enum { chars, ints, floats } AttrType;
```