

1) Введем обозначения:

LS = <Long-слагаемое>

LOGS = <логическое слагаемое>

LOGM = <логический множитель>

LM = <Long-множитель>

LOGEXPR = <логическое выражение>

LEXPR = <Long-выражение>

TEXPR = <Text-выражение>

EXPR = <выражение>

TREL = <Text-отношение>

LREL = <Long-отношение>

REL = <отношение>

LVAL = <Long-величина>

CMP = <операция сравнения>

STRE = <строка-образец>

CONST = <список констант>

2) Перепишем грамматику в этих обозначениях:

<WHERE-клауза> ::=

WHERE <имя поля типа TEXT> [**NOT**] **LIKE** STRE |

WHERE EXPR [**NOT**] **IN** (CONST) |

WHERE LOGEXPR |

WHERE ALL

STRE ::= <строка>

EXPR ::= LEXPR | TEXPR

CONST ::= <строка> { , <строка> } | <длинное целое> { , <длинное целое> }

LEXPR ::= LS { <+|-> LS }

<+|-> ::= + | -

LS ::= LM { <*/|%> LM }

<*/|%> ::= * | / | %

LM ::= LVAL | (LEXPR)

LVAL ::= <имя поля типа LONG> | <длинное целое>

TEXPR ::= <имя поля типа TEXT> | <строка>

LOGEXPR ::= LOGS { **OR** LOGS }

LOGS ::= LOGM { **AND** LOGM }

LOGM ::= **NOT** LOGM | (LOGEXPR) | (REL)

REL ::= TREL | LREL

TREL ::= TEXPR CMP TEXPR

LREL ::= LEXPR CMP LEXPR

CMP ::= = | > | < | >= | <= | !=

3) Избавимся от пересечений FIRST:

<WHERE-клауза> ::= **WHERE** WHEREEXPR

WHEREEXPR ::= <имя поля типа TEXT> LOGINLIKE | <строка> LOGIN | LOGEXPR

[[**NOT**] **IN** (CONST)] | **ALL**

LOGINLIKE ::= NOTINLIKE | LOGTREL

LOGIN ::= [**NOT**] **IN** (CONST) | LOGTREL

LOGTREL ::= CMP TEXPR LOGEXPR

NOTINLIKE ::= [**NOT**] INLIKE

INLIKE = **LIKE** <строка> | **IN** (CONST)

```

CONST ::= <строка> { , <строка> } | <длинное целое> { , <длинное целое> }
LEXPR ::= LS { <+|-> LS }
LS ::= LM { <*/|%> LM }
LM ::= LVAL | ( LOGEXPR )
LVAL ::= <имя поля типа LONG> | <длинное целое>
TEXPR ::= <имя поля типа TEXT> | <строка>
LOGEXPR ::= LOGS { OR LOGS }
LOGS ::= LOGM { AND LOGM }
LOGM ::= NOT LOGM | REL
REL ::= TREL | LREL
TREL ::= TEXPR CMP TEXPR
LREL ::= LEXPR [CMP LEXPR]
CMP ::= = | > | < | >= | <= | !=

```

Пришлось расширить грамматику, избавившись от семантических правил. Теперь необходима дополнительная проверка для логических и арифметических выражений, а также для WHERE-IN запроса с арифметическим выражением и WHERE запроса с логическим выражением.

4) Проверим цикл:

```

LEXPR ::= LS { <+|-> LS }
LS ::= LM { <*/|%> LM }
LM ::= LVAL | ( LOGEXPR )

```

Преобразуем:

```

LEXPR ::= LS LEXPR'
LEXPR' ::= <+|-> LS LEXPR' | eps
LS ::= LM LS'
LS' ::= <*/|%> LM LS' | eps
LM ::= LVAL | ( LOGEXPR )

```

follow(LEXPR') =), NOT, IN, =, >, <, >=, <=, !=

first(LEXPR') = +, -

follow(LS') =), NOT, IN, =, >, <, >=, <=, !=

first(LS') = *, /, %

Итог: PC применим

5) Проверим цикл:

```

LOGEXPR ::= LOGS { OR LOGS }
LOGS ::= LOGM { AND LOGM }
LOGM ::= NOT LOGM | REL

```

Преобразуем:

```

LOGEXPR ::= LOGS LOGEXPR'
LOGEXPR' ::= OR LOGS LOGEXPR' | eps
LOGS ::= LOGM LOGS'
LOGS' ::= AND LOGM LOGS' | eps
LOGM ::= NOT LOGM | REL

```

follow(LOGEXPR') =)

first(LOGEXPR') = OR

follow(LOGS') =)

first(LOGS') = AND

Итог: PC применим