**Code Reviewer 1.0**

**Contents**

**1.Objective**

**2.Tools installation**

**3.Introduction of Tools**

**4.Description of Code Reviewer**

**5.Results**

**6.References**

**1.Objective:** To make a code reviewer such that without executing the sql statement it should give Notification about block keywords and also certain mysql statements.

Main Objectives :

(1) wherever where clause is present check indexing on associated columns.

(1) Select ,Delete,Update statements should appear with where clause

(2) Join statement shouldn’t contain Left|Right Join

(3) There Shouldn’t be any Join on Big Tables List of Big tables will be provided.

(4) If there are More than 4 Join in a select statement Then log Warnings.

(5) Insert statements Contains columns for making entries of respective values.

(6) Drop,Truncate statements on allowed when Tables are Temporary

(7) If size of any Declared column is meaningless than give warnings.Examples:varchar(200000),Decimal(20,0)

(8) Alter statement is not permitted to add column,drop column

(9) Delete statements should appear with where clause

(10) Update statements should appear with where clause

Block Keywords are :

"master","slave","alter","show" "kill","start","stop","grant","purge","shutdown","flush","lock","restart","start"," “stop","revoke"

Note:These block keywords are Detected At the starting of code.And appearance of any out of them will Lead to abort the code with list of all block keywords that are present in sql procedure/function.

That’s why at the starting of code we are splitting the code and checking if block keyword is present or not.

**2. Tools Installation :**To meet above objective and Requirements Parsing based approach is used.

This is Python Implementation using **Pyparser**

**python setup.py install**

**From pyparsing import \***

**3.Introduction to Tools:** pyparsing contains functions and their attributes to develop complex parsers.

Two methods are used for parsing input string **setParseAction()** and **parseString().**

Above methods gives different different parsing structure after scanning input string.

**SetResultsName** is used for naming the grammar statements.

**SetDebug()** is used for identify which token is matching with given grammar.

**Dump()** is used to show structure of parse string with label.

**4. Description of Code Reviewer:**

**Grammar Type:**

1. Select

**ColumnS=delimited List(ZeroOrMore(((lpar+casecolumn+rpar)|casecolumn)|functionbody|identifier|Word(nums) | quotedString|oneOf("= <= >= < > != + - \* /")))**

**reserved\_words=(\_inner|\_left|\_right|\_join|\_left\_outer|\_right\_outer|where|\_on|\_into|\_from|\_in)**

**identifier = ~reserved\_words + column**

**whereCond = (((expression)+ Comparator +((lpar+(caseStmt|selectStmt)+rpar)|expression))|(expression+\_between+expression+\_and +expression) |(expression+(\_in|not\_in)+Range) |(column.setResultsName("WhereColumn",listAllMatches=True)+**

**(\_is\_null|\_is\_not\_null)) |((column.setResultsName("WhereColumn",listAllMatches=True))**

**+\_like+columnVal))**

**whereExpr=Forward()**

**T = ((lpar+whereExpr+rpar)|whereCond)**

**E1 =Forward()**

**whereExpr <<= T + E1**

**E1 <<= Optional((\_and|\_or)+T+E1,'')**

**functionbody<<=Functionname+(lpar+Optional(delimitedList(ZeroOrMore(functionbody|identifier|Optional('-','')+Word(nums) | quotedString)|expression),'')+rpar)**

**columnVal = (Optional('-','')+Word(nums) | quotedString)**

**table=column=StoredProcedure=Database=Functionname=Word(alphanums+'\_.`')**

**Above Variables are used in below select grammer**

**selectStmt<<=((select+**

**ColumnS.setResultsName("Columns") +**

**Optional(\_into+ ('\*' |columns).setResultsName("Columns"),'')+**

**\_from + Optional(lpar+selectStmt+rpar,'')+**

**table.setResultsName("Table",listAllMatches=True)+Optional((\_as+table)|identifier,'')+**

**Optional(\_into+column,'')+**

**ZeroOrMore(Optional((\_inner|\_left\_outer|\_right\_outer|\_left|\_right),'').setResultsName("JoinType",listAllMatches=True)+\_join+**

**(table.setResultsName("JoinTables",listAllMatches=True)|(lpar+selectStmt+rpar))+**

**Optional((\_as+table)|(identifier),'')+**

**Optional(\_on+ColumnS,''))+**

**Optional(where + Group(whereExpr), '').setResultsName("where",listAllMatches=True) +**

**Each([Optional(groupby + (column|functionbody)+Optional(\_asc|\_desc,''),'').setDebug(False),Optional(\_having +Group(whereExpr) ,'').setDebug(False),**

**Optional(orderby + (column|functionbody),'').setDebug(False),Optional(\_desc|\_asc,''),Optional(\_limit+(columnVal|column),'') ]))+**

**Optional((\_union\_all|\_union)+selectStmt,'')).setResultsName("SelectJoinUnion",listAllMatches=True)**

**Statement Types :**

(1**) Statement and matching part of the select statement grammar**

**Examples**

**select+**

**delimitedList(column)+**

**Optional(\_into+columns).setResultsName("Columns"),'')+**

**\_from+**

**table.setResultsName("Table",listAllMatches=True)+**

**Optional(identifier,'')+**

**Optional((\_inner),'').setResultsName("JoinType",listAllMatches=True)+**

**\_join+**

**(table.setResultsName("JoinTables",listAllMatches=True))+**

**Optional((identifier),'')+**

**Optional(\_on+ColumnS,'')+**

**Optional(where + Group(whereExpr), '').setResultsName("where",listAllMatches=True) +Optional(\_limit+(columnVa,'') ]))**

**Examples**

select

p.inqptcategoryid ,

cwctmap.packageenddate,

i.ispremium

into

v\_packagetype,

v\_packageexpirydate ,

v\_ispremium

from cwctdealermapping cwctmap inner join packages p on cwctmap.packageid=p.id

inner join inquirypointcategory i on p.inqptcategoryid=i.id where cwdealerid=v\_dealerid limit 1;

(2**) Statement and matching part of the select statement grammar**

**Examples**

select concat(hosturl , directorypath , logourl) into v\_certifiedlogourl

from classified\_certifiedorg

where id = v\_certificationid;

(3)

**Examples**

select s.id into v\_inquiryid

from sellinquiries s

inner join carphotos c

on c.inquiryid = s.id

where s.tc\_stockid = v\_stockid

and s.sourceid = v\_sourceid

and c.tc\_carphotoid = v\_tc\_carphotoid;

(4) **Case statement appearing inside select query**

**Examples**

select count(id)

into v\_photocount

from carphotos

where inquiryid = v\_inquiryid

and isdealer = (

case

when v\_sellertype = 1 then 1

else 0

end )

and isactive = 1

and isapproved = 1;

(5) **Function for returning value and putting back into a variable**

**Examples**

select `CalculateSortScoreForIndividual`(v\_ispremium, v\_score, null, v\_photocount) into v\_SortScoreNew;

(6)  **If Inside function body neither number,identifier,functionname,and neither a expression occurs Than**

**functionbody<<=Functionname+(**

**Lpar+**

**Optional(delimitedList**

**(ZeroOrMore(**

**Functionbody**

**|**

**Identifier**

**|**

**Optional('-','')**

**+**

**Word(nums)),'')**

**+rpar)**

**Examples**

select distinct cfl.custemail, cfl.custmobile, cfl.inquiryid, cfl.custname, cfl.issenttosource, si.sourceid, cfl.isverified, cfl.id, si.dealerid, si.tc\_stockid

from classifiedleads cfl

inner join sellinquiries si on cfl.inquiryid = si.id and cfl.sellertype = '1' -- sellertype is varchar

inner join ct\_addonpackages cap on si.dealerid = cap.cwdealerid

where cap.addonpackageid = 101 and cfl.issenttosource = 0 and date\_add(now(),interval -15 minute) > entrydatetime

and cap.isactive=1;

(7) **Select column are like this than :**

**Examples**

**ColumnS=delimitedList(**

**ZeroOrMore(((**

**lpar+**

**Casecolumn+**

**rpar)|casecolumn)**

**|**

**Functionbody**

**|**

**identifier|**

**Word(nums)**

**|**

**quotedString**

**|**

**oneOf("= <= >= < > != + - \* /")))**

SELECT cma.Name + ' '+ cmo.Name+ ' '+cv.Name CarName

**(8)**

**casecolumn=\_case**

**+**

**(functionbody**

**|**

**column)**

**+OneOrMore(\_when**

**+columnVal**

**+\_then+**

**(columnVal**

**|**

**\_null))+**

**Optional(\_else**

**+(columnVal**

**|**

**Functionbody**

**|**

**column),'')**

**+\_end**

**Examples**

**select lst.dealerid,**

**lst.owners as noowners,**

**lst.sortscore,**

**Lst.videocount,**

**lst.additionalfuel,**

**case lst.originalimgpath when null then '' else concat(lst.hosturl , '310x174' , lst.originalimgpath end as imageurlmedium,**

**case vs.carfueltype when 1 then 'Petrol' when 2 then 'Diesel' when 3 then 'CNG' when 4 then 'LPG' when 5 then 'Electric' end as carfueltype,**

**lst.rootid, lst.rootname**

**from livelistings lst**

**inner join cwmasterdb.carversions vs on lst.versionid=vs.id**

**where lst.sellertype=1 and lst.dealerid not in (select listmember from fnsplitcsv)**

**and 12\*(year(now())-year(lst.makeyear)) + month(now()) - month(lst.makeyear) <= 96**

**and lst.cityid in(select id from cwmasterdb.cities where name in ('Agra', 'gurgaon') and isdeleted=0);**

**(9) In Declare statement we have select statement like this**

**DeclarativeSyntax=\_declare + DataType + ';'**

**DataType=((**

**\_int**

**|**

**\_varchar**

**|**

**\_numeric**

**|**

**\_decimal**

**|**

**\_float**

**'|**

**\_double**

**|**

**\_integer**

**|**

**\_datetime**

**|**

**\_tinyint**

**|**

**\_smallint**

**|**

**\_bigint**

**|**

**\_date**

**|**

**\_time**

**|**

**\_year**

**|**

**\_char**

**|**

**\_timestamp**

**|**

**\_bit**

**|**

**\_nvarchar).setResultsName("Data",listAllMatches=True)**

**+DataTypeSize.setResultsName("DataSize",listAllMatches=True))**

**+Optional(\_unsigned,'')**

**+Optional(**

**Optional(\_default,'')**

**+(\_not\_null|\_null**

**|**

**(Optional('-','')**

**+Word(nums)**

**|**

**(lpar**

**+selectStmt**

**+rpar))**

**|**

**(Combine(Word(nums) + "." + Word(nums)))),'')+**

**Optional(\_auto\_incr,'')**

**DataTypeSize=Optional(**

**Lpar+**

**(SizeInt.setResultsName("IntegerPartSize")**

**+**

**','+SizeFraction.setResultsName("FractionPartSize")**

**|**

**Size.setResultsName("DataTypeSize"))**

**+rpar,'')**

**Example**

**declare v\_mobile varchar(20) default (**

**select customermobile from customersellinquiries where id = v\_carid limit 1**

**);**

2.Delete:

deleteStmt=(\_delete+

\_from+

table.setResultsName("Table")+

Optional(where + Group(whereExpr), '').setResultsName("where")).setResultsName("Delete",listAllMatches=True)

**Examples**

Delete from mytable;

Delete from mytable where (a>=b) and (b=c or b!=d)

3.Update:

updateStmt=(\_update+

table.setResultsName("Table")+

\_set+(SetColumnValues)+

Optional(where+ Group(whereExpr), '').setResultsName("where")).setResultsName("Update",listAllMatches=True)

**Examples**

Update table set a=b,b=d,a=now(12)\*d, where a=d and b=c or v=i;

update carphotos

set ismain = 0

where tc\_carphotoid <> v\_tc\_carphotoid and inquiryid=v\_inquiryid and isactive = 1 and isapproved = 1;

update carphotos

set ismain = v\_ismain,

description = v\_description,

originalimgpath = v\_originalimgpath,

title=v\_title

where tc\_carphotoid=v\_tc\_carphotoid and inquiryid=v\_inquiryid;

4.Create:

ColumnsDefine = delimitedList (column.setResultsName(" Column ",listAllMatches=True ) + DataType ))

createStmt=(\_create+

Optional(\_temporary,'').setResultsName("Temporary")+

\_table+

table.setResultsName("Table")+

((lpar + ( selectStmt | ColumnsDefine.setResultsName( " ColumnsDetails " )) + rpar)

|

(( selectStmt | ColumnsDefine )))).setResultsName( " Create",listAllMatches=True)

**Example**

create temporary table tempindividualresponseforct

select carprice,

cityname

from

(select

c.id buyerid,

c.name buyername,

llc.cityname cityname

from classifiedrequests cr

inner join ll\_cities llc on csi.cityid=llc.cityid -- added by afrose

where cr.requestdatetime >= date\_add(now(), interval v\_individualresponseend minute) and cr.requestdatetime <= date\_add(now(), interval v\_individualresponsestart minute)

union

select

-1 buyerid,

cl.custname buyername,

llc.cityname cityname

from classifiedleads cl

inner join ll\_cities llc on csi.cityid=llc.cityid -- added by afrose

where cl.isverified = 0 and cl.sellertype =2

and cl.entrydatetime >= date\_add(now(), interval v\_individualresponseend minute) and cl.entrydatetime <= date\_add(now(), interval v\_individualresponsestart minute)

) a

left join tc\_customerdetails tcd on a.buyermobile = tcd.mobile

and tcd.entrydate >= date\_add(now(), interval v\_dealerresponsedays day) and tcd.entrydate<= now()

where tcd.mobile is null;

5.Insert:

**InsertIntoColumns=Optional((**

**Lpar**

**+**

**delimitedList(column).setResultsName("InsertColumns",listAllMatches=True)**

**+**

**rpar),'')**

**InsertColumnValues=lpar**

**+**

**delimitedList((**

**columnVal**

**|**

**Column**

**|**

**functionbody).setResultsName("ColumnValue"))**

**+**

**Rpar**

insertStmt=(\_insert\_into+

((table.setResultsName("Table")+

InsertIntoColumns+

\_values+

InsertColumnValues.setResultsName("ColumnValues"))

|

(functionbody

+

selectStmt))).setResultsName("Insert",listAllMatches=True)

**Examples**

insert into sellinquiries

(

dealerid ,

carversionid ,

Sourceid

)

values

(

v\_emi ,

v\_sourceid

);

6.Truncate:

**option=Optional(\_if\_exists,'')**

truncateStmt=(

\_truncate

+

Optional(\_temporary,'').setResultsName("Temporary")

+

\_table

+

option

+

table.setResultsName("Table")).setResultsName("Truncate",listAllMatche=True)

**Example**

truncate table if exists tempindividualresponseforct;

7.Drop

dropStmt=(\_drop

+Optional(\_temporary,'').setResultsName("Temporary")

+(\_table

+

Option

+

tables.setResultsName("Table"))).setResultsName("Drop",listAllMatches=True)

**Example**

drop table if exists tempindividualresponseforct;

8.Call Procedure statement:

callStmt=

\_call

+StoredProcedure

+lpar

+delimitedList(columnVal

|

column)

+rpar

**Example**

9.Set Statement:

setStmt = \_set + column.setResultsName("ValueName") + Literal(' = ')+( caseStmt | Value )

Value=expression

caseStmt = \_case + ZeroOrMore( \_when + column

+ oneOf(" = != >= <= <> < > ''") +

columnVal +

\_ then + columnVal)

+ \_else + columnVal + \_end

**Example**

set v\_lg = ifnull(v\_longitude,0);

Value includes cobination of functions/columns/columnvalues and arthimatic operations

set v\_currentdistance= sqrt( power(((v\_lt - v\_ltvaluation) \* v\_constlt), 2)

+ power(((v\_lg - v\_lgvaluation) \* v\_constlg), 2) );

SET LASTNAME =CASE WHEN LASTNAME = 'AAA' THEN 'BBB'   
 WHEN LASTNAME = 'CCC' THEN 'DDD'   
 WHEN LASTNAME = 'EEE' THEN 'FFF'   
 ELSE LASTNAME  
 END

10.If ( condition , Yes ,No ) is Like Condition?True:False

IF = \_if + lpar + cond + ',' + yes + ',' + no + rpar

cond = ( Value ) + oneOf(" = != <= >= < > <> ") + (Value)

yes = ( Value )

no = ( Value )

**Example**

Set column\_name=If( expression,expression,expression )

11.If...Else if …..Else….End if :

ifStmt << = (( \_if\_exists | \_if\_not\_exists | \_if ) +

(( lpar + ifCondition + rpar)

| ifCondition) + \_then

+( ZeroOrMore ((

setStmt | updateStmt | selectStmt | callWithSelect|insertStmt | truncateStmt | deleteStmt | alterStmt | createStmt | callStmt|dropStmt|ifStmt)+';')|

(Return+';'))

+ZeroOrMore(\_else\_if+((lpar+ifCondition+rpar)|ifCondition)+\_then

+(ZeroOrMore(( setStmt | updateStmt | selectStmt | callWithSelect|

insertStmt| truncateStmt| deleteStmt| alterStmt|

createStmt

callStmt|

dropStmt|

ifStmt)+';')|

(Return+';')))+

Optional(\_else+(ZeroOrMore((setStmt|updateStmt| selectStmt|callWithSelect|

insertStmt

|truncateStmt|

deleteStmt|alterStmt|createStmt|callStmt|dropStmt|ifStmt)+';'))

|(Return+';'),'')+

\_end\_if).setResultsName("IfStmt")

Other = ZeroOrMore( functionbody | identifier | columnVal )

ifStmt = Forward( )

ifCondition = ( ( Group ( whereExpr ) ) |

( column + ( \_is\_null | \_is\_not\_null ))|

(selectStmt) | Other )

**Example**

(1)

IF EXISTS (SELECT InquiryId FROM livelistings WHERE Inquiryid = v\_CarId AND SellerType = 2) THEN

SET v\_IsLive = 1;

ELSE

SET v\_IsLive = 0;

END IF;

(2)

IF v\_SellInquiryId = -1 THEN

SELECT ID INTO v\_AreaId

FROM cwmasterdb.areas

WHERE Name = v\_AreaName AND PinCode = v\_PinCode limit 1;

VALUES(v\_CustomerId, v\_ShareToCT);

SET v\_ID = LAST\_INSERT\_ID();

if(v\_ID is not null and v\_Owners is not null) then

insert into customersellinquirydetails(InquiryId,Owners) values (v\_ID,v\_Owners);

end if;

ELSe

UPDATE customersellinquiries

SET CityId = v\_CityId,

Price = v\_Price,

PinCode = v\_PinCode,

PackageId = IF(v\_PackageId > 0, v\_PackageId, PackageId),

ShareToCT = IF(v\_ShareToCT != -1, v\_ShareToCT, ShareToCT),

CustomerName = ifnull(v\_CustomerName,CustomerName),

CustomerEmail = ifnull(v\_CustomerEmail,CustomerEmail),

CustomerMobile = ifnull(v\_CustomerMobile,CustomerMobile)

WHERE ID = v\_SellInquiryId;

SET v\_ID = v\_SellInquiryId;

update customersellinquirydetails set Owners = ifnull(v\_Owners,Owners) where InquiryId = v\_SellInquiryId;

END IF;

12.While…..End While:

WhileStmt=Forward()

WhileStmt<<=\_while + Group(whereExpr) + \_do + ZeroOrMore((setStmt

|ifStmt.setResultsName("IfStmt")

|callStmt

|insertStmt

|createStmt

|dropStmt|

deleteStmt|

alterStmt|

truncateStmt

|selectStmt|

updateStmt

|WhileStmt)+';')+\_end\_while

**Example**

while v\_p <= v\_l do

set v\_c = ascii(substring(v\_s, v\_p, 1));

if (v\_c between 48 and 57 or v\_c between 65 and 90 or v\_c between 97 and 122)

then

set v\_s2 = concat(v\_s2,char(v\_c));

end if;

set v\_p = v\_p + 1;

end while ;

13.Create Procedure Standard statement:

InputParameters=delimitedList(

Optional((\_in | \_out | \_inout),'') + column + DataType

)

DeclarativeSyntax=( \_declare + column + DataType + ';'

)

createProcedureStmt = createProcedure +

StoredProcedure.setResultsName("Procedure") +

lpar + Optional(InputParameters.setResultsName("Input"),'') + rpar +

Optional( \_sql\_security\_invoker,'').setResultsName("SQLSECURITY")+

\_begin +

ZeroOrMore(DeclarativeSyntax).setResultsName("Declare")+

ZeroOrMore(( selectStmt|setStmt|ifStmt.setResultsName("IfStmt")|callStmt|updateStmt|createStmt|dropStmt|alterStmt|insertStmt|deleteStmt|WhileStmt.setResultsName("WhileStmt")|createStmt)+';'

) +

\_end + Optional(';','')

**Example:**

CREATE PROCEDURE `InsertCustomerSellInquiriesPartial\_v17\_6\_2` (v\_SellInquiryId NUMERIC,

v\_CustomerId DECIMAL,

v\_CityId INT,

v\_Owners decimal)

SQL SECURITY INVOKER

BEGIN

declare column\_name varchar(2000);

update customersellinquirydetails set Owners = ifnull(v\_Owners,Owners) where InquiryId = v\_SellInquiryId;

END ;

14.Create Function Statement(Return)

createfunctionStmt = createFunction + Functionname.setResultsName("Function")+

Lpar +

Optional(InputParameters.setResultsName("Input"),'') +

Rpar +

\_returns+ DataType+ Optional(\_sql\_security\_invoker,'').setResultsName("SQLSECURITY")+

\_begin+

ZeroOrMore(DeclarativeSyntax).setResultsName("Declare")+

ZeroOrMore((selectStmt|setStmt|ifStmt.setResultsName("IfStmt")|callStmt|updateStmt|createStmt|dropStmt|alterStmt|insertStmt|deleteStmt|WhileStmt.setResultsName("WhileStmt")|createStmt|Return)+';') +

\_end+Optional(';','')

**Example:**

CREATE FUNCTION `RemoveSpecialChars`(v\_s varchar(2500)) RETURNS varchar(2500) SQL SECURITY INVOKER

begin

declare v\_s2 varchar(2500);

if v\_s is null then

return null;

end if;

set v\_l = char\_length(v\_s)

while v\_p <= v\_l do

set v\_c = ascii(substring(v\_s, v\_p, 1));

if (v\_c between 48 and 57 or v\_c between 65 and 90 or v\_c between 97 and 122)

then

set v\_s2 = concat(v\_s2,char(v\_c));

end if;

set v\_p = v\_p + 1;

end while ;

if char\_length(v\_s2) = 0

then

return null;

end if;

return v\_s2;

End

15.Alter :

alterStmt=(\_alter +

\_table

+table.setResultsName("Table")+ delimitedList(DropColumnStmt.setResultsName("DropColumnStmt",listAllMatches=True)|

addDropStmt.setResultsName("AddDropStmt",listAllMatches=True)|

ChangeColumnStmt.setResultsName("ChangeColmnStmt",listAllMatches=True)|

RenameTableStmt.setResultsName("RenameTableStmt",listAllMatches=True))).setResultsName("Alter",listAllMatches=True)

DropColumnStmt=\_drop+Optional(\_column,'')+column.setResultsName("column").setResultsName("DropColumn")

ChangeColumnStmt=\_change.setResultsName("Change")+column.setResultsName("ColumnFirst")+column.setResultsName("ColumnSecond")+DataType

RenameTableStmt=\_rename.setResultsName("Rename")+table.setResultsName("Tables")

addDropStmt = (\_add|\_modify)

+Group((\_index.setResultsName("Index")

|\_unique.setResultsName("Unique")

|\_primary\_key.setResultsName("PrimaryKey"))

+lpar +(column+ZeroOrMore(','+column)).setResultsName("Columns") +

rpar|(Optional(\_column,'')+ column.setResultsName("Columns")+DataType))

**Example:**

Alter table table\_name add column column\_name,

drop column column\_name,

add index(p,qa),

add primary key(b),

change bc bg varchar(200);

**5.Results:**

Average 61 % of Accuracy.

Output :Optional(List\_of\_files\_that\_are\_Not\_parsable,’’) + List\_Of\_warnings

Input:Folder containing sql files or a file with database name

6.References :

[Pyparsing](http://pythonhosted.org/pyparsing/#Indices)