# first - second

# yyyy-mm-dd

WITH YourTable AS(

SELECT TIMESTAMP '2001-1-1' AS first, TIMESTAMP '2002-1-1' as second

UNION ALL SELECT TIMESTAMP '2002-1-1' AS first, TIMESTAMP '2001-1-1'

UNION ALL SELECT TIMESTAMP '2001-1-1', TIMESTAMP '2001-2-1'

UNION ALL SELECT TIMESTAMP '2001-2-1', TIMESTAMP '2001-1-1'

UNION ALL SELECT TIMESTAMP '2001-1-1', TIMESTAMP '2001-1-2'

UNION ALL SELECT TIMESTAMP '2001-1-2' , TIMESTAMP '2001-1-1'

)

SELECT TIMESTAMP\_DIFF(first, second, DAY)

FROM YourTable

# Into refactor\_stats

# Standard SQl, large results

# Into refactor\_stats

# Standard SQl, large results

Select

# anchor commit details

Refactor.commit as refactor\_commit

, refactor.repo\_name as refactor\_repo\_name

, refactor.difference\_old\_path as file

# Typo

#, max(refactor.author\_email) as refctor\_email

#, max(refactor.author\_date) as refctor\_date

, max(refactor.author\_email) as refactor\_email

, max(refactor.author\_date) as refactor\_date

, max(refactor.message) as refactor\_message

, max(if(refactor.commit = context\_size.commit, context\_size.files , null)) as refactor\_commit\_files

, max(if(refactor.commit = context\_size.commit, context\_size.non\_test\_files , null)) as refactor\_commit\_non\_test\_files

, count(distinct if(refactor.commit = context.commit and refactor.repo\_name = context.repo\_name, context.difference\_old\_path, null) ) as refactor\_file\_num

, max(refactor.bq\_classification) as is\_refactor\_commit\_also\_fix

, max(refactor.adaptive\_matches > 0) as is\_refactor\_commit\_also\_adaptive

, max((LENGTH(REGEXP\_REPLACE(lower(refactor.difference\_old\_path), 'test', '#'))

- LENGTH(REGEXP\_REPLACE(lower(refactor.difference\_old\_path), 'test', ''))) > 0) as refactor\_file\_test

, max(lower(reverse(substr(reverse(refactor.difference\_old\_path), 0, strpos(reverse(refactor.difference\_old\_path),'.'))))) as refactor\_file\_extension

, max(lower(reverse(substr(reverse(refactor.difference\_old\_path), 0, strpos(reverse(refactor.difference\_old\_path),'.')))) In

('.bat', '.c', '.cc', '.coffee', '.cpp', '.cs', '.cxx', '.go',

'.groovy', '.hs', '.java', '.js', '.lua', '.m',

'.module', '.php', '.pl', '.pm', '.py', '.rb', '.s', '.scala',

'.sh', '.swift', '.tpl', '.twig')) as major\_source\_extension

, count(distinct if(refactor.commit = context.commit and refactor.repo\_name = context.repo\_name

And

(LENGTH(REGEXP\_REPLACE(lower(context.difference\_old\_path), 'test', '#'))

- LENGTH(REGEXP\_REPLACE(lower(context.difference\_old\_path), 'test', ''))) = 0

, context.difference\_old\_path, null) ) as refactor\_non\_test\_file\_num

######## After 3 months

, count(distinct IF(TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) < 3\*30 and TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) > 1

, context.commit, null)) as commits\_after\_3\_months

, count(distinct IF(TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) < 3\*30 and TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) > 1 and context.bq\_classification

, context.commit, null)) as bug\_hits\_after\_3\_months

, if( count(distinct IF(TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) < 3\*30 and TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) > 1

, context.commit, null)) > 0

, 1.0\*count(distinct IF(TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) < 3\*30 and TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) > 1 and context.bq\_classification

, context.commit, null)) / count(distinct IF(TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) < 3\*30 and TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) > 1

, context.commit, null))

, 0 ) as bug\_ratio\_after\_3\_months

, count(distinct IF(TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) < 3\*30 and TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) > 1 and context.refactor\_matches > 0

, context.commit, null)) as refactor\_after\_3\_months

# Avg File per commit

, avg(IF(TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) < 3\*30 and TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) > 1

, context\_size.files, null)) as avg\_file\_per\_commit\_after\_3\_months

, avg(IF(TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) < 3\*30 and TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) > 1

, context\_size.non\_test\_files, null)) as avg\_non\_test\_file\_per\_commit\_after\_3\_months

# capped Avg File per commit

, avg(IF(TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) < 3\*30 and TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) > 1

, if(context\_size.files > 21, 21,context\_size.files) , null)) as cap\_avg\_file\_per\_commit\_after\_3\_months

, avg(IF(TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) < 3\*30 and TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) > 1

, if(context\_size.non\_test\_files > 17, 17,context\_size.non\_test\_files), null)) as cap\_avg\_non\_test\_file\_per\_commit\_after\_3\_months

####### Before 3 months

, count(distinct IF(TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) > - 3\*30 and TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) < -1

, context.commit, null)) as commits\_before\_3\_months

, count(distinct IF(TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) > -3\*30 and TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) < -1 and context.bq\_classification

, context.commit, null)) as bug\_hits\_before\_3\_months

, if( count(distinct IF(TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) > -3\*30 and TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) < -1

, context.commit, null) )> 0

,1.0\*count(distinct IF(TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) > -3\*30 and TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) < -1 and context.bq\_classification

, context.commit, null)) / count(distinct IF(TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) > -3\*30 and TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) < -1

, context.commit, null))

, 0) as bug\_ratio\_before\_3\_months

, count(distinct IF(TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) > -3\*30 and TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) < -1 and context.refactor\_matches > 0

, context.commit, null)) as refactor\_before\_3\_months

, avg(IF(TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) > -3\*30 and TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) < -1

, context\_size.files, null)) as avg\_file\_per\_commit\_before\_3\_months

, avg(IF(TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) > -3\*30 and TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) < -1

, context\_size.non\_test\_files, null)) as avg\_non\_test\_file\_per\_commit\_before\_3\_months

# capped Avg File per commit

, avg(IF(TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) > -3\*30 and TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) < -1

, if(context\_size.files > 21, 21,context\_size.files), null)) as cap\_avg\_file\_per\_commit\_before\_3\_months

, avg(IF(TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) > -3\*30 and TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) < -1

, if(context\_size.non\_test\_files > 17, 17,context\_size.non\_test\_files), null)) as cap\_avg\_non\_test\_file\_per\_commit\_before\_3\_months

##### 6 months

######## After 6 months

, count(distinct IF(TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) < 6\*30 and TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) > 1

, context.commit, null)) as commits\_after\_6\_months

, count(distinct IF(TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) < 6\*30 and TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) > 1 and context.bq\_classification

, context.commit, null)) as bug\_hits\_after\_6\_months

, if (count(distinct IF(TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) < 6\*30 and TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) > 1

, context.commit, null)) > 0

,1.0\*count(distinct IF(TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) < 6\*30 and TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) > 1 and context.bq\_classification

, context.commit, null)) / count(distinct IF(TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) < 6\*30 and TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) > 1

, context.commit, null))

, 0) as bug\_ratio\_after\_6\_months

, count(distinct IF(TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) < 6\*30 and TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) > 1 and context.refactor\_matches > 0

, context.commit, null)) as refactor\_after\_6\_months

, avg(IF(TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) < 6\*30 and TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) > 1

, context\_size.files, null)) as avg\_file\_per\_commit\_after\_6\_months

, avg(IF(TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) < 6\*30 and TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) > 1

, context\_size.non\_test\_files, null)) as avg\_non\_test\_file\_per\_commit\_after\_6\_months

# cap avg

, avg(IF(TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) < 6\*30 and TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) > 1

, if(context\_size.files > 21, 21,context\_size.files), null)) as cap\_avg\_file\_per\_commit\_after\_6\_months

, avg(IF(TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) < 6\*30 and TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) > 1

, if(context\_size.non\_test\_files > 17, 17,context\_size.non\_test\_files), null)) as cap\_avg\_non\_test\_file\_per\_commit\_after\_6\_months

####### Before 6 months

, count(distinct IF(TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) > - 6\*30 and TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) < -1

, context.commit, null)) as commits\_before\_6\_months

, count(distinct IF(TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) > -6\*30 and TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) < -1 and context.bq\_classification

, context.commit, null)) as bug\_hits\_before\_6\_months

, if (count(distinct IF(TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) > -6\*30 and TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) < -1

, context.commit, null)) > 0

, 1.0\*count(distinct IF(TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) > -6\*30 and TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) < -1 and context.bq\_classification

, context.commit, null)) / count(distinct IF(TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) > -6\*30 and TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) < -1

, context.commit, null))

, 0) as bug\_ratio\_before\_6\_months

, count(distinct IF(TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) > -6\*30 and TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) < -1 and context.refactor\_matches > 0

, context.commit, null)) as refactor\_before\_6\_months

, avg(IF(TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) > -6\*30 and TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) < -1

, context\_size.files, null)) as avg\_file\_per\_commit\_before\_6\_months

, avg(IF(TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) > -6\*30 and TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) < -1

, context\_size.non\_test\_files, null)) as avg\_non\_test\_file\_per\_commit\_before\_6\_months

# cap

, avg(IF(TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) > -6\*30 and TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) < -1

, if(context\_size.files > 21, 21,context\_size.files), null)) as cap\_avg\_file\_per\_commit\_before\_6\_months

, avg(IF(TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) > -6\*30 and TIMESTAMP\_DIFF(context.author\_date, refactor.author\_date, DAY) < -1

, if(context\_size.non\_test\_files > 17, 17,context\_size.non\_test\_files), null)) as cap\_avg\_non\_test\_file\_per\_commit\_before\_6\_months

#### refactor types

# Refactor - rename

, max( (LENGTH(REGEXP\_REPLACE(lower(refactor.message),'(^|$|[^a-z\_])(renam(e|es|ed|ing|ings)|better nam(e|es|ing))(^|$|[^a-z\_])', '@'))-LENGTH(REGEXP\_REPLACE(lower(refactor.message),'(^|$|[^a-z\_])(renam(e|es|ed|ing|ings)|better nam(e|es|ing))(^|$|[^a-z\_])', ''))) )

as rename

# Refactor - whitespace\_wars

, max( (LENGTH(REGEXP\_REPLACE(lower(refactor.message),'(^|$|[^a-z\_])(space(s)?|white(-| )?space(s)?|tab(s)?)(^|$|[^a-z\_])', '@'))-LENGTH(REGEXP\_REPLACE(lower(refactor.message),'(^|$|[^a-z\_])(space(s)?|white(-| )?space(s)?|tab(s)?)(^|$|[^a-z\_])', ''))) )

as whitespace\_wars

# Refactor - style

, max( (LENGTH(REGEXP\_REPLACE(lower(refactor.message),'(^|$|[^a-z\_])(style|styling)(^|$|[^a-z\_])', '@'))-LENGTH(REGEXP\_REPLACE(lower(refactor.message),'(^|$|[^a-z\_])(style|styling)(^|$|[^a-z\_])', ''))) )

as style

# Refactor - upgrade

, max( (LENGTH(REGEXP\_REPLACE(lower(refactor.message),'(^|$|[^a-z\_])(upgrad(e|es|ed|ing))(^|$|[^a-z\_])', '@'))-LENGTH(REGEXP\_REPLACE(lower(refactor.message),'(^|$|[^a-z\_])(upgrad(e|es|ed|ing))(^|$|[^a-z\_])', ''))) )

as upgrade

# Refactor - reuse

, max( (LENGTH(REGEXP\_REPLACE(lower(refactor.message),'(^|$|[^a-z\_])(re(-| )?use|re(-| )?usability)(^|$|[^a-z\_])', '@'))-LENGTH(REGEXP\_REPLACE(lower(refactor.message),'(^|$|[^a-z\_])(re(-| )?use|re(-| )?usability)(^|$|[^a-z\_])', ''))) )

as reuse

# Refactor - simple

, max( (LENGTH(REGEXP\_REPLACE(lower(refactor.message),'(^|$|[^a-z\_])(simplif(y|es|ied|ying|ication)|simplicity)(^|$|[^a-z\_])', '@'))-LENGTH(REGEXP\_REPLACE(lower(refactor.message),'(^|$|[^a-z\_])(simplif(y|es|ied|ying|ication)|simplicity)(^|$|[^a-z\_])', ''))) )

as simple

# Refactor - software\_goals

, max( (LENGTH(REGEXP\_REPLACE(lower(refactor.message),'(((^|$|[^a-z\_])(better|improv(e|es|ed|ing)|increas(e|es|ed|ing)|reduc(e|es|ed|ing)|worse|make|more|less))((^|$|[^a-z\_])|(^|$|[^a-z\_])[\\s\\S]{0,50}(^|$|[^a-z\_]))(abstraction|coherence|cohesion|complexity|correctness|coupling|dependability|duplication|efficiency|extensibility|flexibility|maintainability|naming|performance|portability|quality|readability|reliability|re(-| )?use|re(-| )?usability|security|simplicity|testability|testable|re(-| )?usable|readable|portable|maintainable|flexible|efficient|encapsulation)(^|$|[^a-z\_]))', '@'))-LENGTH(REGEXP\_REPLACE(lower(refactor.message),'(((^|$|[^a-z\_])(better|improv(e|es|ed|ing)|increas(e|es|ed|ing)|reduc(e|es|ed|ing)|worse|make|more|less))((^|$|[^a-z\_])|(^|$|[^a-z\_])[\\s\\S]{0,50}(^|$|[^a-z\_]))(abstraction|coherence|cohesion|complexity|correctness|coupling|dependability|duplication|efficiency|extensibility|flexibility|maintainability|naming|performance|portability|quality|readability|reliability|re(-| )?use|re(-| )?usability|security|simplicity|testability|testable|re(-| )?usable|readable|portable|maintainable|flexible|efficient|encapsulation)(^|$|[^a-z\_]))', ''))) )

as software\_goals

# Refactor - reorganize

, max( (LENGTH(REGEXP\_REPLACE(lower(refactor.message),'(^|$|[^a-z\_])(re(-|)?organiz(e|es|ed|ing)|re(-|)?structur(e|es|ed|ing)|compos(e|es|ed|ing)|de(-| )?compos(e|es|ed|ing)|combin(e|es|ed|ing)|re(-|)?arrang(e|es|ed|ing)|re(-| )?packag(e|es|ed|ing))(^|$|[^a-z\_])', '@'))-LENGTH(REGEXP\_REPLACE(lower(refactor.message),'(^|$|[^a-z\_])(re(-|)?organiz(e|es|ed|ing)|re(-|)?structur(e|es|ed|ing)|compos(e|es|ed|ing)|de(-| )?compos(e|es|ed|ing)|combin(e|es|ed|ing)|re(-|)?arrang(e|es|ed|ing)|re(-| )?packag(e|es|ed|ing))(^|$|[^a-z\_])', ''))) )

as reorganize

# Refactor - feedback

, max( (LENGTH(REGEXP\_REPLACE(lower(refactor.message),'(^|$|[^a-z\_])(py(-| )?lint|lint|review comments(s)?|code review|cr|pep8)(^|$|[^a-z\_])', '@'))-LENGTH(REGEXP\_REPLACE(lower(refactor.message),'(^|$|[^a-z\_])(py(-| )?lint|lint|review comments(s)?|code review|cr|pep8)(^|$|[^a-z\_])', ''))) )

as feedback

# Refactor - refactor

, max( (LENGTH(REGEXP\_REPLACE(lower(refactor.message),'(^|$|[^a-z\_])(re(-| )?factor(ed|s|ing|ings)?)(^|$|[^a-z\_])', '@'))-LENGTH(REGEXP\_REPLACE(lower(refactor.message),'(^|$|[^a-z\_])(re(-| )?factor(ed|s|ing|ings)?)(^|$|[^a-z\_])', ''))) )

as refactor

# Refactor - unneeded

, max( (LENGTH(REGEXP\_REPLACE(lower(refactor.message),'(^|$|[^a-z\_])(unnecessary|unneeded|unused|(not|never|no longer) used|no longer needed|redundant|useless|duplicate(d)?|deprecated|obsolete(d)?|commented)(^|$|[^a-z\_])', '@'))-LENGTH(REGEXP\_REPLACE(lower(refactor.message),'(^|$|[^a-z\_])(unnecessary|unneeded|unused|(not|never|no longer) used|no longer needed|redundant|useless|duplicate(d)?|deprecated|obsolete(d)?|commented)(^|$|[^a-z\_])', ''))) )

as unneeded

# Refactor - se\_constructs

, max( (LENGTH(REGEXP\_REPLACE(lower(refactor.message),'(^|$|[^a-z\_])(enum(s)?|names(-| )?pace(s)?)(^|$|[^a-z\_])', '@'))-LENGTH(REGEXP\_REPLACE(lower(refactor.message),'(^|$|[^a-z\_])(enum(s)?|names(-| )?pace(s)?)(^|$|[^a-z\_])', ''))) )

as se\_constructs

# Refactor - optimization

, max( (LENGTH(REGEXP\_REPLACE(lower(refactor.message),'(^|$|[^a-z\_])(optimiz(e|es|ed|ing|ation|ations)|efficient)(^|$|[^a-z\_])', '@'))-LENGTH(REGEXP\_REPLACE(lower(refactor.message),'(^|$|[^a-z\_])(optimiz(e|es|ed|ing|ation|ations)|efficient)(^|$|[^a-z\_])', ''))) )

as optimization

# Refactor - clean

, max( (LENGTH(REGEXP\_REPLACE(lower(refactor.message),'(^|$|[^a-z\_])(clean(ing|s|ed)|cleaner)(^|$|[^a-z\_])', '@'))-LENGTH(REGEXP\_REPLACE(lower(refactor.message),'(^|$|[^a-z\_])(clean(ing|s|ed)|cleaner)(^|$|[^a-z\_])', ''))) )

as clean

# Refactor - rework

, max( (LENGTH(REGEXP\_REPLACE(lower(refactor.message),'(^|$|[^a-z\_])(re(-)?work(ed|s|ing|ings)?|re(-)?(write|write|wrote|writing)|re(-)?cod(e|ed|es|ing))(^|$|[^a-z\_])', '@'))-LENGTH(REGEXP\_REPLACE(lower(refactor.message),'(^|$|[^a-z\_])(re(-)?work(ed|s|ing|ings)?|re(-)?(write|write|wrote|writing)|re(-)?cod(e|ed|es|ing))(^|$|[^a-z\_])', ''))) )

as rework

# Refactor - todo

, max( (LENGTH(REGEXP\_REPLACE(lower(refactor.message),'(^|$|[^a-z\_])(to( |-)?do(s)?)(^|$|[^a-z\_])', '@'))-LENGTH(REGEXP\_REPLACE(lower(refactor.message),'(^|$|[^a-z\_])(to( |-)?do(s)?)(^|$|[^a-z\_])', ''))) )

as todo

# Refactor - spelling

, max( (LENGTH(REGEXP\_REPLACE(lower(refactor.message),'(^|$|[^a-z\_])(typo(s)?|spelling|mis(-| )?sspell(ed|ing)?)(^|$|[^a-z\_])', '@'))-LENGTH(REGEXP\_REPLACE(lower(refactor.message),'(^|$|[^a-z\_])(typo(s)?|spelling|mis(-| )?sspell(ed|ing)?)(^|$|[^a-z\_])', ''))) )

as spelling

# Refactor - improve

, max( (LENGTH(REGEXP\_REPLACE(lower(refactor.message),'(^|$|[^a-z\_])(improv(e|es|ed|ing)|better)(^|$|[^a-z\_])', '@'))-LENGTH(REGEXP\_REPLACE(lower(refactor.message),'(^|$|[^a-z\_])(improv(e|es|ed|ing)|better)(^|$|[^a-z\_])', ''))) )

as improve

FROM

`citric-trees-187317.repo\_attributes.valid\_commits\_classified` as refactor

Join

`citric-trees-187317.repo\_attributes.valid\_commits\_classified` as context

On

Refactor.repo\_name = context.repo\_name

And

Refactor.difference\_old\_path = context.difference\_old\_path

Join

`citric-trees-187317.repo\_attributes.valid\_commit\_size` as context\_size

On

Context.commit = context\_size.commit

Where

# Anchor is a refactor

Refactor.refactor\_matches > 0

# date after 2008, GitHub funding year (avoid old /possibly corrupted data)

And

TIMESTAMP\_DIFF( refactor.author\_date, cast( '2008-01-01' as TIMESTAMP), DAY) > 0

# date until 2019 to ease reproducibility

and

TIMESTAMP\_DIFF( refactor.author\_date, cast( '2019-01-01' as TIMESTAMP), DAY) < 0

group by

refactor\_commit

, refactor\_repo\_name

, file

# abstraction

Select

count(\*) as refactors

, avg(bug\_ratio\_after\_6\_months -bug\_ratio\_before\_6\_months) as diff\_6\_month

, avg(bug\_ratio\_after\_3\_months -bug\_ratio\_before\_3\_months) as diff\_3\_month

, sum(if (avg\_file\_per\_commit\_before\_3\_months - avg\_file\_per\_commit\_after\_3\_months > 0.5,1,0)) as coupling\_1\_refactors

, avg(if (avg\_file\_per\_commit\_after\_3\_months - avg\_file\_per\_commit\_before\_3\_months > 1 , bug\_ratio\_after\_3\_months -bug\_ratio\_before\_3\_months, null)) as coupling\_1\_diff\_3\_month

, avg(if (avg\_file\_per\_commit\_after\_6\_months - avg\_file\_per\_commit\_before\_6\_months > 1 , bug\_ratio\_after\_6\_months -bug\_ratio\_before\_6\_months, null)) as coupling\_1\_diff\_6\_month

, sum(if (avg\_file\_per\_commit\_after\_3\_months - avg\_file\_per\_commit\_before\_3\_months > 1

And commits\_after\_3\_months > 10 and commits\_after\_3\_months > 10,1,0)) as coupling\_1\_commits\_10\_refactors

, avg(if (avg\_file\_per\_commit\_after\_3\_months - avg\_file\_per\_commit\_before\_3\_months > 1

And commits\_after\_3\_months > 10 and commits\_after\_3\_months > 10 , bug\_ratio\_after\_3\_months -bug\_ratio\_before\_3\_months, null)) as coupling\_1\_commits\_10\_diff\_3\_month

, avg(if (avg\_file\_per\_commit\_after\_6\_months - avg\_file\_per\_commit\_before\_6\_months > 1

And commits\_after\_6\_months > 10 and commits\_after\_6\_months > 10 , bug\_ratio\_after\_6\_months -bug\_ratio\_before\_6\_months, null)) as coupling\_1\_commits\_10\_diff\_6\_month

from

[citric-trees-187317:repo\_attributes.refactor\_stats]

Select

Bug\_ratio\_after\_3\_months < bug\_ratio\_before\_3\_months as improved\_ccp

, avg\_file\_per\_commit\_after\_3\_months < avg\_file\_per\_commit\_before\_3\_months as improved\_productivity

, count(\*) as cases

from

`citric-trees-187317.repo\_attributes.refactor\_stats`

Where

commits\_after\_3\_months > 4

and commits\_after\_3\_months > 4

Group by

Improved\_ccp

, improved\_productivity

Order by

Improved\_ccp

, improved\_productivity