#Into valid\_2018\_files

# Standard Sql

Select

repo\_name

, difference\_old\_path as file

# For each file, the extension will be unique.

# Max is used for aggregation

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

, count(distinct commit) as commits

, count(distinct case when (

(((LENGTH(REGEXP\_REPLACE(lower(message), '(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")((choose|take|set|use)\\s\*(the|a)?\\s\*correct|(not|isn\'t|doesn\'t)\\s+work(s|ing)?|bad initialization|buffer overflow|bug(s|z)?|bugfix(es)?|correct\\s\*(a|the|some|few|this)|correct(ed|ion|ly|s)?|dangling pointer|deadlock|defect|double free|error|fail(ed|s)?|failure(s)?|fault(s)?|faulty initialization|fix(ed|es)?|fixin(s)?|fixing(s)?|fixup(s)?|flaw(s)?|hang|heap overflow|incorrect(ly)?|memory leak|missing\\s(default value|initialization|switch case)|mistake(s|n|nly)?|null pointer|overrun|problem(s)|race condition|resource leak|revert|segmentation fault|workaround|wrong(nly)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")', '#'))

-

LENGTH(REGEXP\_REPLACE(lower(message), '(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")((choose|take|set|use)\\s\*(the|a)?\\s\*correct|(not|isn\'t|doesn\'t)\\s+work(s|ing)?|bad initialization|buffer overflow|bug(s|z)?|bugfix(es)?|correct\\s\*(a|the|some|few|this)|correct(ed|ion|ly|s)?|dangling pointer|deadlock|defect|double free|error|fail(ed|s)?|failure(s)?|fault(s)?|faulty initialization|fix(ed|es)?|fixin(s)?|fixing(s)?|fixup(s)?|flaw(s)?|hang|heap overflow|incorrect(ly)?|memory leak|missing\\s(default value|initialization|switch case)|mistake(s|n|nly)?|null pointer|overrun|problem(s)|race condition|resource leak|revert|segmentation fault|workaround|wrong(nly)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")', ''))))

-

((LENGTH(REGEXP\_REPLACE(lower(message), '(((\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")fix(ed|s|es|ing)?.{1,40}(#|(camel|snake|kebab|flat|lower|upper)\\s\*case|code review|coding style|comment(s)?|cosmetic|cr(s)?(-)?|documentation|format(s|ing)?|help|remark(s)|space(s)?|style|styling|typo(s)?|warning(s)?|whitespace(s)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|"))|((#|(camel|snake|kebab|flat|lower|upper)\\s\*case|code review|coding style|comment(s)?|cosmetic|cr(s)?(-)?|documentation|format(s|ing)?|help|remark(s)|space(s)?|style|styling|typo(s)?|warning(s)?|whitespace(s)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|").{1,40}(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")fix(ed|s|es|ing)?(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|"))|((error check(ing)?|error handling|error message(s)?|error report(s|ing)?|exception handling|fixed point)))', '#'))

-

LENGTH(REGEXP\_REPLACE(lower(message), '(((\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")fix(ed|s|es|ing)?.{1,40}(#|(camel|snake|kebab|flat|lower|upper)\\s\*case|code review|coding style|comment(s)?|cosmetic|cr(s)?(-)?|documentation|format(s|ing)?|help|remark(s)|space(s)?|style|styling|typo(s)?|warning(s)?|whitespace(s)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|"))|((#|(camel|snake|kebab|flat|lower|upper)\\s\*case|code review|coding style|comment(s)?|cosmetic|cr(s)?(-)?|documentation|format(s|ing)?|help|remark(s)|space(s)?|style|styling|typo(s)?|warning(s)?|whitespace(s)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|").{1,40}(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")fix(ed|s|es|ing)?(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|"))|((error check(ing)?|error handling|error message(s)?|error report(s|ing)?|exception handling|fixed point)))', ''))))

-

((LENGTH(REGEXP\_REPLACE(lower(message), '(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")(lack|n\'t|never|nobody|none|not|nothing|without)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|").{0,20}(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")((choose|take|set|use)\\s\*(the|a)?\\s\*correct|(not|isn\'t|doesn\'t)\\s+work(s|ing)?|bad initialization|buffer overflow|bug(s|z)?|bugfix(es)?|correct\\s\*(a|the|some|few|this)|correct(ed|ion|ly|s)?|dangling pointer|deadlock|defect|double free|error|fail(ed|s)?|failure(s)?|fault(s)?|faulty initialization|fix(ed|es)?|fixin(s)?|fixing(s)?|fixup(s)?|flaw(s)?|hang|heap overflow|incorrect(ly)?|memory leak|missing\\s(default value|initialization|switch case)|mistake(s|n|nly)?|null pointer|overrun|problem(s)|race condition|resource leak|revert|segmentation fault|workaround|wrong(nly)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")', '#'))

-

LENGTH(REGEXP\_REPLACE(lower(message), '(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")(lack|n\'t|never|nobody|none|not|nothing|without)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|").{0,20}(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")((choose|take|set|use)\\s\*(the|a)?\\s\*correct|(not|isn\'t|doesn\'t)\\s+work(s|ing)?|bad initialization|buffer overflow|bug(s|z)?|bugfix(es)?|correct\\s\*(a|the|some|few|this)|correct(ed|ion|ly|s)?|dangling pointer|deadlock|defect|double free|error|fail(ed|s)?|failure(s)?|fault(s)?|faulty initialization|fix(ed|es)?|fixin(s)?|fixing(s)?|fixup(s)?|flaw(s)?|hang|heap overflow|incorrect(ly)?|memory leak|missing\\s(default value|initialization|switch case)|mistake(s|n|nly)?|null pointer|overrun|problem(s)|race condition|resource leak|revert|segmentation fault|workaround|wrong(nly)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")', ''))))) > 0

) then commit else null end) as hits

, 1.0\*count(distinct case when

(

(((LENGTH(REGEXP\_REPLACE(lower(message), '(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")((choose|take|set|use)\\s\*(the|a)?\\s\*correct|(not|isn\'t|doesn\'t)\\s+work(s|ing)?|bad initialization|buffer overflow|bug(s|z)?|bugfix(es)?|correct\\s\*(a|the|some|few|this)|correct(ed|ion|ly|s)?|dangling pointer|deadlock|defect|double free|error|fail(ed|s)?|failure(s)?|fault(s)?|faulty initialization|fix(ed|es)?|fixin(s)?|fixing(s)?|fixup(s)?|flaw(s)?|hang|heap overflow|incorrect(ly)?|memory leak|missing\\s(default value|initialization|switch case)|mistake(s|n|nly)?|null pointer|overrun|problem(s)|race condition|resource leak|revert|segmentation fault|workaround|wrong(nly)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")', '#'))

-

LENGTH(REGEXP\_REPLACE(lower(message), '(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")((choose|take|set|use)\\s\*(the|a)?\\s\*correct|(not|isn\'t|doesn\'t)\\s+work(s|ing)?|bad initialization|buffer overflow|bug(s|z)?|bugfix(es)?|correct\\s\*(a|the|some|few|this)|correct(ed|ion|ly|s)?|dangling pointer|deadlock|defect|double free|error|fail(ed|s)?|failure(s)?|fault(s)?|faulty initialization|fix(ed|es)?|fixin(s)?|fixing(s)?|fixup(s)?|flaw(s)?|hang|heap overflow|incorrect(ly)?|memory leak|missing\\s(default value|initialization|switch case)|mistake(s|n|nly)?|null pointer|overrun|problem(s)|race condition|resource leak|revert|segmentation fault|workaround|wrong(nly)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")', ''))))

-

((LENGTH(REGEXP\_REPLACE(lower(message), '(((\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")fix(ed|s|es|ing)?.{1,40}(#|(camel|snake|kebab|flat|lower|upper)\\s\*case|code review|coding style|comment(s)?|cosmetic|cr(s)?(-)?|documentation|format(s|ing)?|help|remark(s)|space(s)?|style|styling|typo(s)?|warning(s)?|whitespace(s)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|"))|((#|(camel|snake|kebab|flat|lower|upper)\\s\*case|code review|coding style|comment(s)?|cosmetic|cr(s)?(-)?|documentation|format(s|ing)?|help|remark(s)|space(s)?|style|styling|typo(s)?|warning(s)?|whitespace(s)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|").{1,40}(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")fix(ed|s|es|ing)?(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|"))|((error check(ing)?|error handling|error message(s)?|error report(s|ing)?|exception handling|fixed point)))', '#'))

-

LENGTH(REGEXP\_REPLACE(lower(message), '(((\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")fix(ed|s|es|ing)?.{1,40}(#|(camel|snake|kebab|flat|lower|upper)\\s\*case|code review|coding style|comment(s)?|cosmetic|cr(s)?(-)?|documentation|format(s|ing)?|help|remark(s)|space(s)?|style|styling|typo(s)?|warning(s)?|whitespace(s)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|"))|((#|(camel|snake|kebab|flat|lower|upper)\\s\*case|code review|coding style|comment(s)?|cosmetic|cr(s)?(-)?|documentation|format(s|ing)?|help|remark(s)|space(s)?|style|styling|typo(s)?|warning(s)?|whitespace(s)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|").{1,40}(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")fix(ed|s|es|ing)?(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|"))|((error check(ing)?|error handling|error message(s)?|error report(s|ing)?|exception handling|fixed point)))', ''))))

-

((LENGTH(REGEXP\_REPLACE(lower(message), '(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")(lack|n\'t|never|nobody|none|not|nothing|without)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|").{0,20}(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")((choose|take|set|use)\\s\*(the|a)?\\s\*correct|(not|isn\'t|doesn\'t)\\s+work(s|ing)?|bad initialization|buffer overflow|bug(s|z)?|bugfix(es)?|correct\\s\*(a|the|some|few|this)|correct(ed|ion|ly|s)?|dangling pointer|deadlock|defect|double free|error|fail(ed|s)?|failure(s)?|fault(s)?|faulty initialization|fix(ed|es)?|fixin(s)?|fixing(s)?|fixup(s)?|flaw(s)?|hang|heap overflow|incorrect(ly)?|memory leak|missing\\s(default value|initialization|switch case)|mistake(s|n|nly)?|null pointer|overrun|problem(s)|race condition|resource leak|revert|segmentation fault|workaround|wrong(nly)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")', '#'))

-

LENGTH(REGEXP\_REPLACE(lower(message), '(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")(lack|n\'t|never|nobody|none|not|nothing|without)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|").{0,20}(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")((choose|take|set|use)\\s\*(the|a)?\\s\*correct|(not|isn\'t|doesn\'t)\\s+work(s|ing)?|bad initialization|buffer overflow|bug(s|z)?|bugfix(es)?|correct\\s\*(a|the|some|few|this)|correct(ed|ion|ly|s)?|dangling pointer|deadlock|defect|double free|error|fail(ed|s)?|failure(s)?|fault(s)?|faulty initialization|fix(ed|es)?|fixin(s)?|fixing(s)?|fixup(s)?|flaw(s)?|hang|heap overflow|incorrect(ly)?|memory leak|missing\\s(default value|initialization|switch case)|mistake(s|n|nly)?|null pointer|overrun|problem(s)|race condition|resource leak|revert|segmentation fault|workaround|wrong(nly)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")', ''))))) > 0

) then commit else null end) /count(distinct commit) as hit\_ratio

, count(distinct case when extract(month from author\_date) in (1,2,3) then commit else null end) as commits\_q1

, count(distinct case when extract(month from author\_date) in (4, 5 ,6) then commit else null end) as commits\_q2

, count(distinct case when (extract(month from author\_date) in (1,2,3))

And

(

(((LENGTH(REGEXP\_REPLACE(lower(message), '(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")((choose|take|set|use)\\s\*(the|a)?\\s\*correct|(not|isn\'t|doesn\'t)\\s+work(s|ing)?|bad initialization|buffer overflow|bug(s|z)?|bugfix(es)?|correct\\s\*(a|the|some|few|this)|correct(ed|ion|ly|s)?|dangling pointer|deadlock|defect|double free|error|fail(ed|s)?|failure(s)?|fault(s)?|faulty initialization|fix(ed|es)?|fixin(s)?|fixing(s)?|fixup(s)?|flaw(s)?|hang|heap overflow|incorrect(ly)?|memory leak|missing\\s(default value|initialization|switch case)|mistake(s|n|nly)?|null pointer|overrun|problem(s)|race condition|resource leak|revert|segmentation fault|workaround|wrong(nly)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")', '#'))

-

LENGTH(REGEXP\_REPLACE(lower(message), '(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")((choose|take|set|use)\\s\*(the|a)?\\s\*correct|(not|isn\'t|doesn\'t)\\s+work(s|ing)?|bad initialization|buffer overflow|bug(s|z)?|bugfix(es)?|correct\\s\*(a|the|some|few|this)|correct(ed|ion|ly|s)?|dangling pointer|deadlock|defect|double free|error|fail(ed|s)?|failure(s)?|fault(s)?|faulty initialization|fix(ed|es)?|fixin(s)?|fixing(s)?|fixup(s)?|flaw(s)?|hang|heap overflow|incorrect(ly)?|memory leak|missing\\s(default value|initialization|switch case)|mistake(s|n|nly)?|null pointer|overrun|problem(s)|race condition|resource leak|revert|segmentation fault|workaround|wrong(nly)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")', ''))))

-

((LENGTH(REGEXP\_REPLACE(lower(message), '(((\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")fix(ed|s|es|ing)?.{1,40}(#|(camel|snake|kebab|flat|lower|upper)\\s\*case|code review|coding style|comment(s)?|cosmetic|cr(s)?(-)?|documentation|format(s|ing)?|help|remark(s)|space(s)?|style|styling|typo(s)?|warning(s)?|whitespace(s)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|"))|((#|(camel|snake|kebab|flat|lower|upper)\\s\*case|code review|coding style|comment(s)?|cosmetic|cr(s)?(-)?|documentation|format(s|ing)?|help|remark(s)|space(s)?|style|styling|typo(s)?|warning(s)?|whitespace(s)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|").{1,40}(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")fix(ed|s|es|ing)?(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|"))|((error check(ing)?|error handling|error message(s)?|error report(s|ing)?|exception handling|fixed point)))', '#'))

-

LENGTH(REGEXP\_REPLACE(lower(message), '(((\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")fix(ed|s|es|ing)?.{1,40}(#|(camel|snake|kebab|flat|lower|upper)\\s\*case|code review|coding style|comment(s)?|cosmetic|cr(s)?(-)?|documentation|format(s|ing)?|help|remark(s)|space(s)?|style|styling|typo(s)?|warning(s)?|whitespace(s)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|"))|((#|(camel|snake|kebab|flat|lower|upper)\\s\*case|code review|coding style|comment(s)?|cosmetic|cr(s)?(-)?|documentation|format(s|ing)?|help|remark(s)|space(s)?|style|styling|typo(s)?|warning(s)?|whitespace(s)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|").{1,40}(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")fix(ed|s|es|ing)?(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|"))|((error check(ing)?|error handling|error message(s)?|error report(s|ing)?|exception handling|fixed point)))', ''))))

-

((LENGTH(REGEXP\_REPLACE(lower(message), '(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")(lack|n\'t|never|nobody|none|not|nothing|without)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|").{0,20}(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")((choose|take|set|use)\\s\*(the|a)?\\s\*correct|(not|isn\'t|doesn\'t)\\s+work(s|ing)?|bad initialization|buffer overflow|bug(s|z)?|bugfix(es)?|correct\\s\*(a|the|some|few|this)|correct(ed|ion|ly|s)?|dangling pointer|deadlock|defect|double free|error|fail(ed|s)?|failure(s)?|fault(s)?|faulty initialization|fix(ed|es)?|fixin(s)?|fixing(s)?|fixup(s)?|flaw(s)?|hang|heap overflow|incorrect(ly)?|memory leak|missing\\s(default value|initialization|switch case)|mistake(s|n|nly)?|null pointer|overrun|problem(s)|race condition|resource leak|revert|segmentation fault|workaround|wrong(nly)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")', '#'))

-

LENGTH(REGEXP\_REPLACE(lower(message), '(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")(lack|n\'t|never|nobody|none|not|nothing|without)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|").{0,20}(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")((choose|take|set|use)\\s\*(the|a)?\\s\*correct|(not|isn\'t|doesn\'t)\\s+work(s|ing)?|bad initialization|buffer overflow|bug(s|z)?|bugfix(es)?|correct\\s\*(a|the|some|few|this)|correct(ed|ion|ly|s)?|dangling pointer|deadlock|defect|double free|error|fail(ed|s)?|failure(s)?|fault(s)?|faulty initialization|fix(ed|es)?|fixin(s)?|fixing(s)?|fixup(s)?|flaw(s)?|hang|heap overflow|incorrect(ly)?|memory leak|missing\\s(default value|initialization|switch case)|mistake(s|n|nly)?|null pointer|overrun|problem(s)|race condition|resource leak|revert|segmentation fault|workaround|wrong(nly)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")', ''))))) > 0

)

then commit else null end) as hits\_q1

, count(distinct case when (extract(month from author\_date) in (4,5,6))

And

(

(((LENGTH(REGEXP\_REPLACE(lower(message), '(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")((choose|take|set|use)\\s\*(the|a)?\\s\*correct|(not|isn\'t|doesn\'t)\\s+work(s|ing)?|bad initialization|buffer overflow|bug(s|z)?|bugfix(es)?|correct\\s\*(a|the|some|few|this)|correct(ed|ion|ly|s)?|dangling pointer|deadlock|defect|double free|error|fail(ed|s)?|failure(s)?|fault(s)?|faulty initialization|fix(ed|es)?|fixin(s)?|fixing(s)?|fixup(s)?|flaw(s)?|hang|heap overflow|incorrect(ly)?|memory leak|missing\\s(default value|initialization|switch case)|mistake(s|n|nly)?|null pointer|overrun|problem(s)|race condition|resource leak|revert|segmentation fault|workaround|wrong(nly)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")', '#'))

-

LENGTH(REGEXP\_REPLACE(lower(message), '(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")((choose|take|set|use)\\s\*(the|a)?\\s\*correct|(not|isn\'t|doesn\'t)\\s+work(s|ing)?|bad initialization|buffer overflow|bug(s|z)?|bugfix(es)?|correct\\s\*(a|the|some|few|this)|correct(ed|ion|ly|s)?|dangling pointer|deadlock|defect|double free|error|fail(ed|s)?|failure(s)?|fault(s)?|faulty initialization|fix(ed|es)?|fixin(s)?|fixing(s)?|fixup(s)?|flaw(s)?|hang|heap overflow|incorrect(ly)?|memory leak|missing\\s(default value|initialization|switch case)|mistake(s|n|nly)?|null pointer|overrun|problem(s)|race condition|resource leak|revert|segmentation fault|workaround|wrong(nly)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")', ''))))

-

((LENGTH(REGEXP\_REPLACE(lower(message), '(((\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")fix(ed|s|es|ing)?.{1,40}(#|(camel|snake|kebab|flat|lower|upper)\\s\*case|code review|coding style|comment(s)?|cosmetic|cr(s)?(-)?|documentation|format(s|ing)?|help|remark(s)|space(s)?|style|styling|typo(s)?|warning(s)?|whitespace(s)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|"))|((#|(camel|snake|kebab|flat|lower|upper)\\s\*case|code review|coding style|comment(s)?|cosmetic|cr(s)?(-)?|documentation|format(s|ing)?|help|remark(s)|space(s)?|style|styling|typo(s)?|warning(s)?|whitespace(s)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|").{1,40}(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")fix(ed|s|es|ing)?(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|"))|((error check(ing)?|error handling|error message(s)?|error report(s|ing)?|exception handling|fixed point)))', '#'))

-

LENGTH(REGEXP\_REPLACE(lower(message), '(((\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")fix(ed|s|es|ing)?.{1,40}(#|(camel|snake|kebab|flat|lower|upper)\\s\*case|code review|coding style|comment(s)?|cosmetic|cr(s)?(-)?|documentation|format(s|ing)?|help|remark(s)|space(s)?|style|styling|typo(s)?|warning(s)?|whitespace(s)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|"))|((#|(camel|snake|kebab|flat|lower|upper)\\s\*case|code review|coding style|comment(s)?|cosmetic|cr(s)?(-)?|documentation|format(s|ing)?|help|remark(s)|space(s)?|style|styling|typo(s)?|warning(s)?|whitespace(s)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|").{1,40}(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")fix(ed|s|es|ing)?(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|"))|((error check(ing)?|error handling|error message(s)?|error report(s|ing)?|exception handling|fixed point)))', ''))))

-

((LENGTH(REGEXP\_REPLACE(lower(message), '(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")(lack|n\'t|never|nobody|none|not|nothing|without)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|").{0,20}(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")((choose|take|set|use)\\s\*(the|a)?\\s\*correct|(not|isn\'t|doesn\'t)\\s+work(s|ing)?|bad initialization|buffer overflow|bug(s|z)?|bugfix(es)?|correct\\s\*(a|the|some|few|this)|correct(ed|ion|ly|s)?|dangling pointer|deadlock|defect|double free|error|fail(ed|s)?|failure(s)?|fault(s)?|faulty initialization|fix(ed|es)?|fixin(s)?|fixing(s)?|fixup(s)?|flaw(s)?|hang|heap overflow|incorrect(ly)?|memory leak|missing\\s(default value|initialization|switch case)|mistake(s|n|nly)?|null pointer|overrun|problem(s)|race condition|resource leak|revert|segmentation fault|workaround|wrong(nly)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")', '#'))

-

LENGTH(REGEXP\_REPLACE(lower(message), '(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")(lack|n\'t|never|nobody|none|not|nothing|without)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|").{0,20}(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")((choose|take|set|use)\\s\*(the|a)?\\s\*correct|(not|isn\'t|doesn\'t)\\s+work(s|ing)?|bad initialization|buffer overflow|bug(s|z)?|bugfix(es)?|correct\\s\*(a|the|some|few|this)|correct(ed|ion|ly|s)?|dangling pointer|deadlock|defect|double free|error|fail(ed|s)?|failure(s)?|fault(s)?|faulty initialization|fix(ed|es)?|fixin(s)?|fixing(s)?|fixup(s)?|flaw(s)?|hang|heap overflow|incorrect(ly)?|memory leak|missing\\s(default value|initialization|switch case)|mistake(s|n|nly)?|null pointer|overrun|problem(s)|race condition|resource leak|revert|segmentation fault|workaround|wrong(nly)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")', ''))))) > 0

)

then commit else null end) as hits\_q2

### hit\_rate\_q1

, 1.0\*count(distinct case when (extract(month from author\_date) in (1,2,3))

And

(

(((LENGTH(REGEXP\_REPLACE(lower(message), '(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")((choose|take|set|use)\\s\*(the|a)?\\s\*correct|(not|isn\'t|doesn\'t)\\s+work(s|ing)?|bad initialization|buffer overflow|bug(s|z)?|bugfix(es)?|correct\\s\*(a|the|some|few|this)|correct(ed|ion|ly|s)?|dangling pointer|deadlock|defect|double free|error|fail(ed|s)?|failure(s)?|fault(s)?|faulty initialization|fix(ed|es)?|fixin(s)?|fixing(s)?|fixup(s)?|flaw(s)?|hang|heap overflow|incorrect(ly)?|memory leak|missing\\s(default value|initialization|switch case)|mistake(s|n|nly)?|null pointer|overrun|problem(s)|race condition|resource leak|revert|segmentation fault|workaround|wrong(nly)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")', '#'))

-

LENGTH(REGEXP\_REPLACE(lower(message), '(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")((choose|take|set|use)\\s\*(the|a)?\\s\*correct|(not|isn\'t|doesn\'t)\\s+work(s|ing)?|bad initialization|buffer overflow|bug(s|z)?|bugfix(es)?|correct\\s\*(a|the|some|few|this)|correct(ed|ion|ly|s)?|dangling pointer|deadlock|defect|double free|error|fail(ed|s)?|failure(s)?|fault(s)?|faulty initialization|fix(ed|es)?|fixin(s)?|fixing(s)?|fixup(s)?|flaw(s)?|hang|heap overflow|incorrect(ly)?|memory leak|missing\\s(default value|initialization|switch case)|mistake(s|n|nly)?|null pointer|overrun|problem(s)|race condition|resource leak|revert|segmentation fault|workaround|wrong(nly)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")', ''))))

-

((LENGTH(REGEXP\_REPLACE(lower(message), '(((\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")fix(ed|s|es|ing)?.{1,40}(#|(camel|snake|kebab|flat|lower|upper)\\s\*case|code review|coding style|comment(s)?|cosmetic|cr(s)?(-)?|documentation|format(s|ing)?|help|remark(s)|space(s)?|style|styling|typo(s)?|warning(s)?|whitespace(s)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|"))|((#|(camel|snake|kebab|flat|lower|upper)\\s\*case|code review|coding style|comment(s)?|cosmetic|cr(s)?(-)?|documentation|format(s|ing)?|help|remark(s)|space(s)?|style|styling|typo(s)?|warning(s)?|whitespace(s)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|").{1,40}(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")fix(ed|s|es|ing)?(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|"))|((error check(ing)?|error handling|error message(s)?|error report(s|ing)?|exception handling|fixed point)))', '#'))

-

LENGTH(REGEXP\_REPLACE(lower(message), '(((\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")fix(ed|s|es|ing)?.{1,40}(#|(camel|snake|kebab|flat|lower|upper)\\s\*case|code review|coding style|comment(s)?|cosmetic|cr(s)?(-)?|documentation|format(s|ing)?|help|remark(s)|space(s)?|style|styling|typo(s)?|warning(s)?|whitespace(s)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|"))|((#|(camel|snake|kebab|flat|lower|upper)\\s\*case|code review|coding style|comment(s)?|cosmetic|cr(s)?(-)?|documentation|format(s|ing)?|help|remark(s)|space(s)?|style|styling|typo(s)?|warning(s)?|whitespace(s)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|").{1,40}(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")fix(ed|s|es|ing)?(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|"))|((error check(ing)?|error handling|error message(s)?|error report(s|ing)?|exception handling|fixed point)))', ''))))

-

((LENGTH(REGEXP\_REPLACE(lower(message), '(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")(lack|n\'t|never|nobody|none|not|nothing|without)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|").{0,20}(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")((choose|take|set|use)\\s\*(the|a)?\\s\*correct|(not|isn\'t|doesn\'t)\\s+work(s|ing)?|bad initialization|buffer overflow|bug(s|z)?|bugfix(es)?|correct\\s\*(a|the|some|few|this)|correct(ed|ion|ly|s)?|dangling pointer|deadlock|defect|double free|error|fail(ed|s)?|failure(s)?|fault(s)?|faulty initialization|fix(ed|es)?|fixin(s)?|fixing(s)?|fixup(s)?|flaw(s)?|hang|heap overflow|incorrect(ly)?|memory leak|missing\\s(default value|initialization|switch case)|mistake(s|n|nly)?|null pointer|overrun|problem(s)|race condition|resource leak|revert|segmentation fault|workaround|wrong(nly)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")', '#'))

-

LENGTH(REGEXP\_REPLACE(lower(message), '(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")(lack|n\'t|never|nobody|none|not|nothing|without)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|").{0,20}(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")((choose|take|set|use)\\s\*(the|a)?\\s\*correct|(not|isn\'t|doesn\'t)\\s+work(s|ing)?|bad initialization|buffer overflow|bug(s|z)?|bugfix(es)?|correct\\s\*(a|the|some|few|this)|correct(ed|ion|ly|s)?|dangling pointer|deadlock|defect|double free|error|fail(ed|s)?|failure(s)?|fault(s)?|faulty initialization|fix(ed|es)?|fixin(s)?|fixing(s)?|fixup(s)?|flaw(s)?|hang|heap overflow|incorrect(ly)?|memory leak|missing\\s(default value|initialization|switch case)|mistake(s|n|nly)?|null pointer|overrun|problem(s)|race condition|resource leak|revert|segmentation fault|workaround|wrong(nly)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")', ''))))) > 0

)

then commit else null end)

/ count(distinct case when extract(month from author\_date) in (1,2,3) then commit else null end)

As hit\_rate\_q1

### hit\_rate\_q1

, 1.0\*count(distinct case when (extract(month from author\_date) in (4,5,6))

And

(

(((LENGTH(REGEXP\_REPLACE(lower(message), '(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")((choose|take|set|use)\\s\*(the|a)?\\s\*correct|(not|isn\'t|doesn\'t)\\s+work(s|ing)?|bad initialization|buffer overflow|bug(s|z)?|bugfix(es)?|correct\\s\*(a|the|some|few|this)|correct(ed|ion|ly|s)?|dangling pointer|deadlock|defect|double free|error|fail(ed|s)?|failure(s)?|fault(s)?|faulty initialization|fix(ed|es)?|fixin(s)?|fixing(s)?|fixup(s)?|flaw(s)?|hang|heap overflow|incorrect(ly)?|memory leak|missing\\s(default value|initialization|switch case)|mistake(s|n|nly)?|null pointer|overrun|problem(s)|race condition|resource leak|revert|segmentation fault|workaround|wrong(nly)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")', '#'))

-

LENGTH(REGEXP\_REPLACE(lower(message), '(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")((choose|take|set|use)\\s\*(the|a)?\\s\*correct|(not|isn\'t|doesn\'t)\\s+work(s|ing)?|bad initialization|buffer overflow|bug(s|z)?|bugfix(es)?|correct\\s\*(a|the|some|few|this)|correct(ed|ion|ly|s)?|dangling pointer|deadlock|defect|double free|error|fail(ed|s)?|failure(s)?|fault(s)?|faulty initialization|fix(ed|es)?|fixin(s)?|fixing(s)?|fixup(s)?|flaw(s)?|hang|heap overflow|incorrect(ly)?|memory leak|missing\\s(default value|initialization|switch case)|mistake(s|n|nly)?|null pointer|overrun|problem(s)|race condition|resource leak|revert|segmentation fault|workaround|wrong(nly)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")', ''))))

-

((LENGTH(REGEXP\_REPLACE(lower(message), '(((\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")fix(ed|s|es|ing)?.{1,40}(#|(camel|snake|kebab|flat|lower|upper)\\s\*case|code review|coding style|comment(s)?|cosmetic|cr(s)?(-)?|documentation|format(s|ing)?|help|remark(s)|space(s)?|style|styling|typo(s)?|warning(s)?|whitespace(s)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|"))|((#|(camel|snake|kebab|flat|lower|upper)\\s\*case|code review|coding style|comment(s)?|cosmetic|cr(s)?(-)?|documentation|format(s|ing)?|help|remark(s)|space(s)?|style|styling|typo(s)?|warning(s)?|whitespace(s)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|").{1,40}(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")fix(ed|s|es|ing)?(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|"))|((error check(ing)?|error handling|error message(s)?|error report(s|ing)?|exception handling|fixed point)))', '#'))

-

LENGTH(REGEXP\_REPLACE(lower(message), '(((\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")fix(ed|s|es|ing)?.{1,40}(#|(camel|snake|kebab|flat|lower|upper)\\s\*case|code review|coding style|comment(s)?|cosmetic|cr(s)?(-)?|documentation|format(s|ing)?|help|remark(s)|space(s)?|style|styling|typo(s)?|warning(s)?|whitespace(s)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|"))|((#|(camel|snake|kebab|flat|lower|upper)\\s\*case|code review|coding style|comment(s)?|cosmetic|cr(s)?(-)?|documentation|format(s|ing)?|help|remark(s)|space(s)?|style|styling|typo(s)?|warning(s)?|whitespace(s)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|").{1,40}(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")fix(ed|s|es|ing)?(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|"))|((error check(ing)?|error handling|error message(s)?|error report(s|ing)?|exception handling|fixed point)))', ''))))

-

((LENGTH(REGEXP\_REPLACE(lower(message), '(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")(lack|n\'t|never|nobody|none|not|nothing|without)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|").{0,20}(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")((choose|take|set|use)\\s\*(the|a)?\\s\*correct|(not|isn\'t|doesn\'t)\\s+work(s|ing)?|bad initialization|buffer overflow|bug(s|z)?|bugfix(es)?|correct\\s\*(a|the|some|few|this)|correct(ed|ion|ly|s)?|dangling pointer|deadlock|defect|double free|error|fail(ed|s)?|failure(s)?|fault(s)?|faulty initialization|fix(ed|es)?|fixin(s)?|fixing(s)?|fixup(s)?|flaw(s)?|hang|heap overflow|incorrect(ly)?|memory leak|missing\\s(default value|initialization|switch case)|mistake(s|n|nly)?|null pointer|overrun|problem(s)|race condition|resource leak|revert|segmentation fault|workaround|wrong(nly)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")', '#'))

-

LENGTH(REGEXP\_REPLACE(lower(message), '(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")(lack|n\'t|never|nobody|none|not|nothing|without)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|").{0,20}(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")((choose|take|set|use)\\s\*(the|a)?\\s\*correct|(not|isn\'t|doesn\'t)\\s+work(s|ing)?|bad initialization|buffer overflow|bug(s|z)?|bugfix(es)?|correct\\s\*(a|the|some|few|this)|correct(ed|ion|ly|s)?|dangling pointer|deadlock|defect|double free|error|fail(ed|s)?|failure(s)?|fault(s)?|faulty initialization|fix(ed|es)?|fixin(s)?|fixing(s)?|fixup(s)?|flaw(s)?|hang|heap overflow|incorrect(ly)?|memory leak|missing\\s(default value|initialization|switch case)|mistake(s|n|nly)?|null pointer|overrun|problem(s)|race condition|resource leak|revert|segmentation fault|workaround|wrong(nly)?)(\\s|\\.|\\?|\\!|\\[|\\]|\\)|\\:|^|$|\\,|\'|")', ''))))) > 0

)

then commit else null end)

/ count(distinct case when extract(month from author\_date) in (4,5,6) then commit else null end)

As hit\_rate\_q2

From

`repo\_attributes.valid\_commits\_2018`

Group by

repo\_name

, file

# into file\_commit\_dist

# legacy sql

Select

ascommits as commits

, count(\*) as files

From

[repo\_attributes.valid\_2018\_files]

Where

Extension 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')

Group by

commits

Order by commits

# into file\_hit\_rate\_dist

# legacy sql

Select

round(hit\_ratio,2) as rhit\_rate

, count(\*) as files

, sum(ascommits) as commits

, sum(hits) as hits

From

[repo\_attributes.valid\_2018\_files]

Where

Extension 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')

Group by

rhit\_rate

Order by rhit\_rate

# into file\_hit\_rate\_dist\_atleast\_10

# legacy sql

Select

round(hit\_ratio,2) as rhit\_rate

, count(\*) as files

, sum(ascommits) as commits

, sum(hits) as hits

From

[repo\_attributes.valid\_2018\_files]

Where

Extension 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')

and

Ascommits > 9

Group by

rhit\_rate

Order by rhit\_rate

# into file\_2018\_stability

# legacy\_sql

Select \*

From

[repo\_attributes.valid\_2018\_files\_with\_stability]

Where

Extension 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')

And

Commits\_q1 > 9

And

Commits\_q2 > 9

# into file\_ccp\_stability\_dist

# Legacy sql

Select

round(hit\_rate\_q1,2) as hit\_rate\_q1

, round(hit\_rate\_q2,2) as hit\_rate\_q2

, count(\*) as files

from

[repo\_attributes.file\_2018\_stability]

Group by

Hit\_rate\_q1

, hit\_rate\_q2

Order by

Hit\_rate\_q1

, hit\_rate\_q2