WeTune: Automatic Discovery and Verification of Query Rewrite Rules

Zhaoguo Wang[1,2], Zhou Zhou[1,2], Yicun Yang[1,2], Haoran Ding[1,2]
Gansen Hu[1,2], Ding Ding[3], Chuzhe Tang[1,2], Haibo Chen[1,2], Jinyang Li[3]
[1]Institute of Parallel and Distributed Systems, Shanghai Jiao Tong University
[2]Engineering Research Center for Domain-specific Operating Systems, Ministry of Education, China
[3]Department of Computer Science, New York University

Abstract

Query rewriting transforms a relational database query into an equivalent but more efficient one, which is crucial for the performance of database-backed applications. Such rewriting relies on pre-specified rewrite rules. In existing systems, these rewrite rules are discovered through manual insights and accumulate slowly over the years.

In this paper, we present WeTune, a rule generator that automatically discovers new rewrite rules. Inspired by compiler superoptimization, WeTune enumerates all valid logical query plans up to a certain size and tries to discover equivalent plans that could potentially lead to more efficient rewrites. The core challenge is to determine which set of conditions (aka constraints) allows one to prove the equivalence between a pair of query plans. We address this challenge by enumerating combinations of "interesting" constraints that relate tables and their attributes between each pair of queries. We also propose a new SMT-based verifier to verify the equivalence of a query pair under different enumerated constraints. To evaluate the usefulness of rewrite rules discovered by WeTune, we apply them on the SQL queries collected from the 20 most popular open-source web applications on GitHub. WeTune successfully optimizes 247 queries that existing databases cannot optimize, resulting in substantial performance improvements.

CCS Concepts

• **Information systems → Query optimization**; • **Theory of computation → Program verification**.

Keywords

Query Rewriting, Rewrite Rule Discovery, SQL Solver

ACM Reference Format:
Zhaoguo Wang, Zhou Zhou, Yicun Yang, Haoran Ding, Gansen Hu, Ding Ding, Chuzhe Tang, Haibo Chen, and Jinyang Li. 2022. WeTune: Automatic Discovery and Verification of Query Rewrite Rules. In *Proceedings of the 2022 International Conference on Management of Data (SIGMOD '22), June 12–17, 2022, Philadelphia, PA, USA.* ACM, New York, NY, USA, 59 pages. https://doi.org/10.1145/3514221.3526125

1 Introduction

Database-backed web applications have been the backbone of Internet applications from online shopping to banking. For many web applications, the database query latency is critical for the user experience. For example, it has been reported that an increase of 500ms in latency can reduce the traffic of a website by 20% [28], and users often give up a website when the loading time takes more than three seconds [41].

Query rewriting, which transforms an original query into a semantically equivalent alternative query [16–18], is an important step in query optimization. Effective rewrites can accelerate the execution time of input queries by orders of magnitude [6]. Rewriting relies on rules that specify the equivalence relations between queries. Existing rules are typically crafted by human experts and can take decades to accumulate [16–18, 27, 36, 37, 42, 44].

However, it is insufficient to rely on manual efforts to discover rewrite rules. The rich feature and subtle semantics of the query language make it challenging to prove equivalence [7, 10, 20] and to craft rules. As a result, the set of hand-written rewrite rules grows very slowly and misses many rewrite opportunities. The situation is made worse by the prevalent use of object-relational-mapping (ORM) frameworks in web application development. ORM frees programmers from explicitly constructing SQL queries but also results in non-intuitive queries whose patterns evade rules crafted by humans. To understand the impact of missed rewrites, we studied 50 real-world queries in several popular open-source web applications on Github. All of these queries have been rewritten by the developers to fix their performance issues. Even the latest version of SQL Server fails to rewrite 27 of these queries (54%) to a more efficient form as fixed manually by the developers. One such query incurs latency up to 37 seconds, while its equivalent rewritten one only takes 0.3 seconds [21] (details in §2.2).

In this paper, we propose WeTune, a rule generator that can automatically discover new rewrite rules without any human effort. Drawing inspiration from compiler superoptimization [2, 33], which finds a semantically equivalent optimal code sequence through the exhaustive search, WeTune aims to discover rewrite rules automatically via brute-force enumeration of all potential rules followed by a correctness check of each generated rule. During this discovery process, WeTune relies on heuristics to filter out those rules that are unlikely to improve performance, aka rules whose rewritten query contains more operators of each type than the original query. The remaining rewrite rules are deemed promising. We empirically determine the usefulness of these promising rules by using them to rewrite real world queries and measure the performance benefits of rewritten queries over synthetically generated database tables.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
SIGMOD '22, June 12–17, 2022, Philadelphia, PA, USA
© 2022 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-9249-5/22/06…$15.00
https://doi.org/10.1145/3514221.3526125

Those that result in beneficial rewrites are useful rules discovered by WeTune.

Although the high-level approach is simple, there are several challenges. First, how to represent rewrite rules in a general form that allows enumeration? Second, how to automatically verify the correctness of enumerated rules without human effort? To address these challenges, WeTune represents a rewrite rule as a pair of query plan templates together with a set of constraints that relate the templates to each other. It enumerates all possible query plan templates up to a threshold number of operators. A query plan template is generic in that it uses symbols instead of concrete names to represent tables, columns and predicates. WeTune further enumerates all constraints, which are conditions that could potentially make a pair of enumerated plan templates semantically equivalent. For instance, specific input relations of the two queries could be constrained to be the same, or the attributes used in a projection are restricted to be a subset of attributes in a certain relation. WeTune verifies the correctness of each rewrite rule using the SQL verifiers. It includes a built-in verifier, which provides a formal way of modeling rewrite rules as SMT formulas. Then the correctness problem can be automatically solved with an SMT solver. Besides the built-in verifier, WeTune also can support using existing SQL verifiers such as SPES to prove the correctness.

We have evaluated the effectiveness of WeTune on real-world database-backed applications using the 20 most popular web applications on GitHub. WeTune outputs 1106 promising rewrite rules, 35 of which are used to optimize queries of these applications. Furthermore, our results show that WeTune can successfully optimize 247 queries that are missed by existing systems, resulting in a latency reduction of up to 99%. Such optimization is due to WeTune's ability to discover new rewrite rules not known to any of the existing systems. WeTune can successfully verify the discovered rewrite rules.

In summary, our work makes the following contributions:

- A study showing that existing manually-discovered rewrite rules are insufficient for real-world queries in popular web applications.
- A demonstration that the enumeration approach introduced in compiler superoptimization can be applied in databases to generate query rewrite rules automatically.
- The formal modeling of database query rewrite rules to allow encoding to SMT formulas, which allows WeTune to verify the correctness of new rewrite rules.
- An evaluation of WeTune on a variety of real-world applications, which shows that it can successfully optimize 247 queries with substantial performance improvement.

2 Motivation

In this section, we examine the rewriting opportunities that existing commercial and open-source databases miss. Then, we study the impact of missed rewrites which cause developers to change a query into a more efficient form manually.

2.1 Insufficiency of Existing Rewrites

Existing databases already use a large number of rules to perform rewrites [16–18, 27, 36, 37, 42, 44]. Nevertheless, we have found that many queries still fail to be rewritten into a more efficient form by existing rules. This finding might come as a surprise: after all, decades of efforts have been spent on crafting rewrite rules, should not most—if not all—rules have been discovered already?

Original Query	Opt. By Existing DB	Ideal (WeTune)

q0: SELECT * FROM labels
WHERE id IN (
SELECT id FROM labels
WHERE id IN (
SELECT id FROM labels
WHERE project_id=10
) ORDER BY title ASC)
	q1: SELECT *
	FROM labels
	WHERE id IN (
	SELECT id
	FROM labels
	WHERE
	project_id=10)
		q2: SELECT *
		FROM labels
		WHERE
		project_id=10

q3: SELECT id FROM notes
WHERE type='D'
AND id IN (
SELECT id FROM notes
WHERE commit_id=7)
	Unchanged	q4: SELECT id
		FROM notes
		WHERE type='D'
		AND commit_id=7

Table 1: Examples of the counter-intuitive queries generated by the ORM framework found in GitLab. The first column lists the original queries. The second one lists the best optimization results of the existing DB systems. The third one lists the ideal results, which can be achieved with the rules generated by WeTune. labels.id and notes.id are the primary keys of the tables, respectively.

To see why manual rewrite rules are insufficient, we want to keep in mind that these rules, more or less, are designed for SQL queries written directly by programmers. Human-written queries usually follow intuitive patterns which can be manually analyzed to distill useful rules. However, in modern web development, programmers no longer explicitly construct SQL queries. Rather, they typically make use of an *object-relational-mapping* (ORM) framework, which allows them to write object-oriented code to manipulate contents in the database. The underlying framework automatically generates SQL queries based on application logic. Not only are the resulting SQL queries generated by ORM opaque to programmers, but also they can be *counter-intuitive* to human rule developers. Table 1 shows two examples from the web application GitLab [15], a popular open-source version management website. Both SQL queries were generated by the ORM framework (ActiveRecord) as the result of running developers' Ruby code.

The first query $q0$ aims to select all the git merge requests whose project_id is "10". Specifically, it uses a subquery to compute the set of id values whose project_id is "10" according to the labels table. It then selects all rows from the labels table whose id falls within this set of subquery values. This query is counter-intuitive and inefficient in two places. First, the subquery to compute matching ids contains another inner subquery and the two subqueries are almost *identical*. Second, the ORDER BY clause of the inner subquery is unnecessary because the outermost IN operator treats the subquery SELECT...ORDER BY as an unordered list [1]. Ideally, the redundancies in the query should be identified by the query optimizer via a rewrite rule, so that the resulting optimized query resembles $q2$ as shown in Table 1. However, among MySQL, PostgreSQL, and MS SQL Server, only PostgreSQL and MS SQL Server can partially rewrite the query to $q1$ which removes ORDER BY and one of the two subqueries.

[1] Such behavior follows the SQL standard [22], and is confirmed in MySQL, PostgreSQL and Oracle DB. MS SQL Server explicitly denies such a query and reports the error "ORDER BY is disallowed in subquery".

Another query in Table 1 (q3) fetches id values from the notes table whose type is "D" and column_id is "7". For this query, the IN-selection is redundant because (1) the table used in the subquery is identical to the table used outside, which is the notes table; (2) The column projected by the subquery is the same as the column used in IN-selection, which is the primary key of the notes table. Hence, the subquery can be eliminated and the query could be transformed into a simple query as $q4$. Unfortunately, all three existing databases (MySQL, PostgreSQL and MS SQL Server) miss such opportunities and keep the query unchanged.

The above inefficient structures are unlikely seen in human-generated queries but are common for queries generated by ORM. Since the ORM-generated query results from running application code in different program locations or even third-party libraries, developers are agnostic to potential redundancies (e.g., duplicate subqueries) and inefficiency (e.g., unnecessary ORDER BY). Therefore, it is very difficult for developers to identify and fix the resulting performance issues.

2.2 Impact and Scope of Missed Rewrites

To better understand the impact of missed rewrites on real-world queries, we studied GitHub issues related to query performance in several popular web applications, including Discourse (discussion forum), GitLab (code management) and Spree (e-commerce) etc. Some are written in Ruby, while others are in Java (the full list can be found in our extended version [49]). The applications are chosen based on popularity, judged by the number of stars on GitHub [11, 13, 15, 25, 39, 40, 43, 46].

We manually inspected 50 GitHub issues related to query performance, with 15 from Discourse [25], 25 from GitLab [15], 4 from Spree [11], 2 from Redmine [43], and 4 from others [13, 39, 40, 46]. For all the 50 queries in our study, developers have fixed them by manually rewriting the original SQL query into a more efficient form, as the databases used by the application failed to rewrite these queries in the same efficient way. We have investigated whether state-of-the-art optimizers in different databases can rewrite these queries. Among these 50 queries, 27 queries (54%) cannot be rewritten into the desired forms in issues by the *latest* version of SQL Server (7 of them are similar to the examples in Table 1). The rewriters in MySQL, PostgreSQL, and Apache Calcite (including both Hep and Volcano Planner) perform even worse, failing to rewrite 38 (76%), 41 (82%), 47 (94%) and 46 (92%) of these queries, respectively. Our study shows that, although opportunities exist for many existing queries to be rewritten to a more efficient form, state-of-the-art manually curated rules miss such rewrites.

Among the Github issues in our study, a few of them [4, 21, 35, 48] give concrete numbers on the performance impact after the manual rewrite. For the example in [21], the original query latency can be up to 37 seconds, while the manually rewritten query only takes 0.3 seconds. Such latency difference is due to the optimizer failing to replace an IN-subquery with an INNER JOIN, which prevents the optimizer from selecting a better access path. The other issues [4, 35, 48] also lead to the 75%-99% latency reduction for their respective applications. Unfortunately, it is difficult to diagnose and resolve these performance problems. In particular, for these 50 issues, it took 13 months on average to fix one (via manual query rewrite). As developers do not directly write SQL but access the database via an ORM, they have less visibility and control over the final queries.

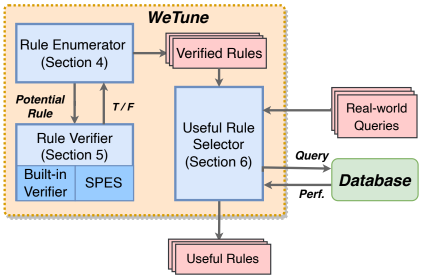


Figure 1: The architecture of WeTune.

3 Our Approach

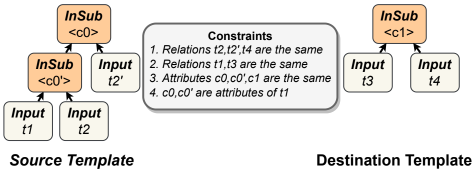
Manually crafted rewrite rules are no longer sufficient in an era where queries are automatically generated by web frameworks. To optimize these auto-generated queries, we need an automated approach to discover rewrite rules.

Basic Idea. WeTune aims to automatically discover new useful rewrite rules without any human effort. It is inspired by compiler superoptimization, especially peephole optimizer [2]. The peephole optimizer aims to transform a sequence of machine instructions into another equivalent but faster sequence and thus has a similar high-level goal as a database query optimizer. Peephole optimizers can automatically discover optimization rules via some form of brute-force search for the instruction sequences [2]. Inspired by this approach, we propose to automatically discover promising query rewrite rules through simple brute-force enumeration and to ensure the correctness of discovered rules through verification.

More concretely, WeTune's search for useful rewrite rules proceeds in two stages. In the first stage, WeTune discovers promising rules by enumerating the potential rewrite rules with the **Rule Enumerator** (§ 4) and verifying their correctness using the **Rule Verifier** (§ 5). We propose a new SMT-based verifier but WeTune can also use other verifiers (e.g. SPES [50]). In this stage, WeTune uses simple heuristics to filter out those rules that are unlikely to bring performance improvement; only promising rules are kept. In the second stage, WeTune empirically determines the usefulness of promising rules by using them to rewrite real-world queries and measuring the performance of the rewritten queries (§ 6). Figure 1 shows the overall architecture of WeTune.

Our high-level approach is straightforward. However, to make it work, we must resolve several technical challenges that face rule enumeration and verification. These challenges are unique to query rewriting and not present in compiler optimization.

(1) How to represent a rewrite rule to make it amicable to enumeration? A rule consists of a pair of queries, which must be generic and not bound to concrete tables and columns. How to enumerate generic queries and make a source query equivalent to a destination query? (Section 4)
(2) How to determine whether an enumerated rewrite rule is correct? Can we adopt an existing query equivalence checker that requires concrete queries to work with generic queries? Can


InSub
<c0>
InSub
<c0'>
Input
t2'
Input
t1
Input
t2
Constraints
1. Relations t2,t2',t4 are the same
2. Relations t1,t3 are the same
3. Attributes c0,c0',c1 are the same
4. c0,c0' are attributes of t1
InSub
<c1>
Input
t3
Input
t4
Source Template
Destination Template

```
q5: ... FROM T WHERE T.x IN (SELECT R.y FROM R)
                  AND T.x IN (SELECT R.y FROM R)
q6: ... FROM T WHERE T.x IN (SELECT R.y FROM R)
```

Figure 2: An example rule found by WeTune (No.4 in Table 7). It can eliminate redundant IN-subquery operator of a SQL query such as $q5$, and rewrite it into $q6$. Existing databases miss the opportunity to rewrite such a counter-intuitive queries.

we develop a new verifier to address the limitations of existing checkers? (Section 5)

4 Rule Enumerator

WeTune models the rewrite rule as a triple $<q_{src}, q_{dest}, C>$, where q_{src} is a source query plan template, q_{dest} is a destination query plan template and C is a set of constraints. A query plan template is a fragment of the logical query plan tree whose operators include selection, projection, etc. Unlike those in a concrete query, the table names, attributes and predicates in a query plan template are symbolic. The constraint set C consists of a set of predicates, each of which describes some relationship between the symbols from the source and destination query plan templates. The rule specifies that if all constraints in C are satisfied, then q_{src} and q_{dest} are semantically equivalent. Given a SQL query q, if some fragment in q matches q_{src}, the matched fragment can be replaced with the corresponding fragment q_{dest} that satisfies C.

Figure 2 shows an example rewrite rule which can eliminate redundant IN-subquery in a SQL query. The source template q_{src} is represented as $\mathsf{InSub}_{c0}(\mathsf{InSub}_{c0'}(\mathsf{t1}, \mathsf{t2}), \mathsf{t2'})$. The operator InSub_{c0} has a left child $\mathsf{InSub}_{c0'}$ and a right child t2'. InSub is an operator in the query plan template which represents IN-subquery. It represents the queries (e.g., $q5$) with two IN-subquery operators, and these two IN-subquery are connected by AND. The destination template q_{dest} is $\mathsf{InSub}_{c1}(\mathsf{t3}, \mathsf{t4})$. The constraint set C specifies the following constraints: $t2$, $t2'$ and $t4$ are the same relations; $t1$ and $t3$ are the same relations; $c0$, $c0'$ and $c1$ are the same attributes. The figure also shows a SQL query $q5$ derived from Gitlab [15]. This query matches q_{src}. Thus, it can be replaced by a better query $q6$ which follows the pattern specified by q_{dest} under the constraints in C. This inefficiency pattern is quite counter-intuitive as its two inner subqueries are almost identical. However, none of our studied DBs can successfully optimize this query.

The Rule Enumerator enumerates potential rewrite rules. To do so, it first enumerates all possible plan templates (Section 4.1). To restrict the search space, it bounds the template size so that the number of operators in a template is within some small threshold. Then, for every pair of plan templates, it enumerates all potential constraints (Section 4.2). Last, it selects the promising rules which are likely to improve the query performance (Section 4.3).

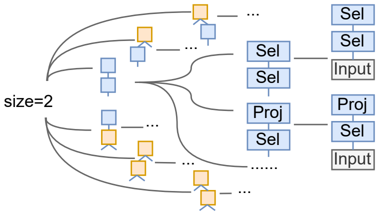


Figure 3: The example of enumerating query plan templates. A blue block denotes an operator with one input, while a yellow block denotes an operator with two inputs.

4.1 Plan Template Enumerator

The query plan template is a tree whose nodes are relational algebra operators with symbolic inputs or parameters [16, 18]:

Operator. Each operator takes one or two relations as input (except Input operator itself), performs algebraic computation according to its semantic, and outputs a single relation. Currently, WeTune only supports the operators in Table 2.

Symbol. In a concrete query plan, operators can be parameterized by concrete schema information such as column names, etc. In a query plan template, such concrete parameters are replaced with symbolic ones. There are three kinds of symbols:

- *Relation Symbol.* A relation symbol *rel* (r for short) represents a relation. It is used to parameterize the input relation of a plan template. e.g., t1, t2, t2', t3 and t4 in Figure 2.
- *Attribute list Symbol.* An attribute list symbol *attrs* (a for short) represents a sequence of attributes. In Figure 2, c0, c0' and c1 are attribute lists. Each of them contains at least one attribute. Additionally, each relation symbol r is associated with an attribute list symbol a_r that represents all the attributes in r.
- *Predicate Symbol.* A predicate symbol *pred* (p for short) represents a predicate, which takes zero or more values as input and yields a boolean value. It is used to parameterize the predicate expression of the selection operator.

The enumeration strategy of WeTune separately enumerates a query plan's tree structure and the operator types for each tree node. More concretely, the enumeration is done in three steps: first, WeTune constructs all possible tree structures with two kinds of internal tree nodes: one type is the node having one child, and another type is the node having two children; Second, for each tree structure, it exhaustively assigns the operators listed in Table 2 to every node to enumerate concrete trees. The number of operator's inputs should match the number of the node's children; Last, it adds Input nodes as the leaf nodes' children. Figure 3 shows the process of enumerating query plan templates having two operators. To reduce the enumeration space, WeTune only enumerates templates up to 4 operators excluding the Input nodes. Furthermore, it filters out those templates that lead to an invalid SQL query, such as misplacing the Deduplication operator.

4.2 Constraint Enumerator

WeTune pairs the enumerated templates as $<q_{src}, q_{dest}>$, and searches for the constraint set that would turn the pair of templates into a valid rewrite rule. A constraint is a predicate that specifies the relationship between symbols in q_{src} and q_{dest}. To bound the search

Operator Name	Symbol	#Input	Description
Input	Input$_r$	0	Input$_r$() represents an initial input relation specified by r.
Projection	Proj$_a$	1	Proj$_a(R)$ projects its input relation R on attributes specified by a.
Selection	Sel$_{p,a}$	1	Sel$_{p,a}(R)$ discards tuples in its input relation R that do not satisfy the predicate p. Values on attributes a from R are used to evaluate the predicate p.
In-Sub Selection	InSub$_a$	2	InSub$_a(R_l, R_r)$ discards tuples in the left input R_l that are absent in the right input R_r. Values on attributes a from R_l are used for the presence check.
(Inner/Left/Right) Join	(I/L/R)Join$_{a_l,a_r}$	2	IJoin$_{a_l,a_r}(R_l, R_r)$ Cartesian products its input relations R_l and R_r, then discards the tuples that have mismatched values on attributes a_l and a_r. (L/R)Join additionally keeps the mismatched tuples and fills NULL on the right/left-side attributes.
Deduplication	Dedup	1	Dedup(R) discards duplication of tuples in its input relation R.

Table 2: SQL operators supported by WeTune.

space, we consider the following limited set of constraints, drawn from our experience of studying existing rewrite rules and examining developers' manual query rewrites.

- *RelEq*(*rel$_1$*, *rel$_2$*). This constraint indicates that two relation symbols, *rel$_1$* and *rel$_2$*, are equivalent (i.e., contain the same tuples).
- *AttrsEq*(*attrs$_1$*, *attrs$_2$*). This constraint indicates that two attribute list symbols, *attrs$_1$* and *attrs$_2$*, have the same sequence of attributes.
- *PredEq*(*pred$_1$*, *pred$_2$*). This constraint indicates that two predicate symbols, *pred$_1$* and *pred$_2$*, are equivalent (i.e. *pred$_1$* ⇔ *pred$_2$*).
- *SubAttrs*(*attrs$_1$*, *attrs$_2$*). This constraint indicates that each attribute in *attrs$_1$* is also in *attrs$_2$*. It can be used to express which relations an attribute list is from. For example, in Figure 2, *SubAttrs*(c_0, a_{t1}) indicates that each attribute in c_0 corresponds to some column from table $t1$.
- *RefAttrs*(*rel$_1$*, *attrs$_1$*, *rel$_2$*, *attrs$_2$*). This constraint indicates that any value in the relation *rel$_1$* on the attribute *attrs$_1$* is also in *rel$_2$* on *attrs$_2$*.
- *Unique*(*rel*, *attrs*). This constraint indicates that every value in *rel* on *attrs* is unique.
- *NotNull*(*rel*, *attrs*). This constraint indicates that every value in *rel* on *attrs* is not NULL.

Given a pair of plan templates <q_{src}, q_{dest}>, constraint enumeration generates the set C^*, which contains all possible constraints related to q_{src} and q_{dest}. This is done by exhaustively filling in the parameters of constraints above with the symbols in q_{src} and q_{dest}. We use C^* later in the search for promising rules.

4.3 Searching for Promising Rules

Given a pair of query templates <q_{src}, q_{dest}>, and constraint set C^* which includes all constraints related to q_{src} and q_{dest}, WeTune needs to search for some subset C of C^* which makes q_{src} and q_{dest} semantically equivalent. Furthermore, we keep only those valid rules which are deemed promising.

A rule of <q_{src}, q_{dest}, C> is promising if it satisfies the following two requirements: first, C is the most relaxed constraint set, such that the removal of any constraint in C compromises the correctness of the rule. In other words, C is the minimal constraint set that enables the equivalence between q_{src} and q_{dest}; Second, q_{dest} does not have more operators of each type than q_{src}. With this heuristic, rewrite rules will simplify rather than complicate the source query, thus are more likely to improve the query performance.

Algorithm 1: Search for Promising Rules

```
1  EnumerateRules(k):
2    T := EnumerateTemplates(k)
3    R := ∅
4    foreach <$q_{src}$, $q_{dest}$> ∈ T × T do
5      if $q_{dest}$ is not simpler than $q_{src}$ then continue
6      $C^*$ := EnumerateConstraints($q_{src}$, $q_{dest}$)
7      $\mathbb{C}$ := SearchRelaxed($q_{src}$, $q_{dest}$, $C^*$)
8      R := R ∪ {<$q_{src}$, $q_{dest}$, C> | C ∈ $\mathbb{C}$}
9    return R
10 SearchRelaxed($q_{src}$, $q_{dest}$, $C^*$):
11   if ¬ProveEq($q_{src}$, $q_{dest}$, $C^*$) then return ∅
12   $\mathbb{C}$ := ∅
13   foreach c ∈ $C^*$ do
14     $\mathbb{C}$ := $\mathbb{C}$ ∪ SearchRelaxed($q_{src}$, $q_{dest}$, $C^*$ − {c})
15   if $\mathbb{C}$ = ∅ then return {$C^*$}
16   else return $\mathbb{C}$
```

Algorithm 1 shows the basic algorithm to search for promising rules. It first enumerates all query templates, as described in Section 4.1. Then, it pairs the enumerated templates as <q_{src}, q_{dest}> and keeps those whose q_{dest} has the same or fewer operators of each type than q_{src} (Line 5). For each pair <q_{src}, q_{dest}>, it generates constraint set C^* by enumerating all possible constraints related to q_{src} and q_{dest}. Last, it invokes SearchRelaxed to recursively search for the subsets of C^* to form the promising rules.

The function SearchRelaxed starts with C^*, and iteratively relaxes the constraint set by removing one constraint (Line 14) and verifies the resulting rule correctness (Line 11). Specifically, it uses an underlying verifier to prove the equivalence between q_{src} and q_{dest} under the constraints in C (Section 5). If the verification fails, we know the constraint set is too relaxed to imply the equivalence. In this case, we stop further relaxing and traceback (Line 11). If no constraint can be further removed, we have found the most relaxed constraint set (s) (Line 15). Note that there could be multiple most relaxed sets, and SearchRelaxed tries to find all of them. This is why it returns a set of sets, and each member is a most relaxed set.

To reduce the search cost, WeTune introduces the following methods: first, it excludes the useless constraints from C^*. A constraint is considered useless if it only involves the symbols in q_{dest} or leads to an illegal query plan; Second, instead of examining every subset of C^*, it only checks the subsets which are both closures and non-conflicting. A subset is a closure if it cannot imply any constraint absent from the set. Meanwhile, a subset is non-conflicting if no constraints in the subset conflict with each other. Two constraints have a conflict if putting them together will introduce an illegal plan. Last, WeTune skips checking the constraint set C if it can be implied by a constraint set C', and C' has already been proved to make q_{src} and q_{dest} equivalent.

5 Rule Verifier

WeTune depends on the rule verifier to ensure correctness. A rule $<q_{src}, q_{dest}, C>$ is correct if the source and destination query plan templates are semantically equivalent when the constraints hold. We design a rule verifier based on FOL (first-order logic) formulation (Section 5.1). WeTune can also use an existing SQL equivalence checker such as SPES [50] to verify rules (Section 5.2).

5.1 Built-in Rule Verifier

At the high level, WeTune's built-in verifier works by first representing a rule $<q_{src}, q_{dest}, C>$ as a U-semiring expression [8], and then converting the expression into FOL formulas. Finally, the FOL formulas are verified using an SMT solver.

5.1.1 Formal Representation of Rules

Given a rewrite rule, we use U-expression [8] to represent q_{src} and q_{dest}, and use FOL formulas to specify the constraint set C.

U-expression. Inspired by UDP [8], we also use U-expressions to model SQL queries under the bag semantics, which capture the multiplicity of a tuple in the relation. Under this representation, a query is viewed as operations on a semiring of natural numbers [8, 19]. We adopt the terms defined in UDP [8], which are summarized below:

- $[\![R]\!](x)$ returns the multiplicity of the tuple x in the relation R.
- $[b] \triangleq$ if b then 1 else 0. Since this expression converts a boolean value to an integer, it can be used to turn a predicate into a U-expression.
- $||e|| \triangleq$ if $e > 0$ then 1 else 0, where e is a U-expression. It models `Deduplication`.
- $not(e) \triangleq$ if $e > 0$ then 0 else 1, where e is a U-expression. It models the negation of a predicate.
- $\sum_{t \in D} f(t) \triangleq f(t_0) + f(t_1) + \cdots$, for all $t_i \in D$, where D is a tuple set called *summation domain*, and f is a function $D \longrightarrow N$. By default, D is an infinite set containing all possible tuples. $\sum_{t \in D} f(t)$ models `Projection`.

In order to model a query plan made out of different operators, these terms are connected by "+" and "∗", which have the same meaning as that of natural numbers. For example, we can use $||\sum_x [t = x.k] \times [\![R]\!](x) \times [x.a > 12]||$ to denote the multiplicity of the tuple t in the output relation of "SELECT DISTINCT x.k FROM R AS x WHERE x.a>12". The summation can be omitted if the projection retains all attributes. For example, $||[\![R]\!](t) \times [t.a > 12]||$ can represent the multiplicity of the tuple t in the output relation of "SELECT DISTINCT * FROM R AS x WHERE x.a > 12". In the following paragraphs, we omit D in summation and just write $\sum_t f(t)$, where t is the *summed variable* of the summation.

Operator	Expression
Input_r	$f(t) \coloneqq r(t)$
Proj_a	$f(t) \coloneqq \sum_{t_l}(f_l(t_l) \times [t = a(t_l)])$
$\text{Sel}_{p,a}$	$f(t) \coloneqq f_l(t) \times [p(a(t))]$
InSubSel_a	$f(t) \coloneqq f_l(t) \times \|\|f_r(a(t))\|\| \times not([IsNull(a(t))])$
IJoin_{a_l,a_r}	$f(t) \coloneqq \sum_{t_l,t_r}([t = t_l \cdot t_r] \times f_l(t_l) \times f_r(t_r) \times\ [a_l(t_l) = a_r(t_r)]$ $\times\ not([IsNull(a_l(t_l))]))$
LJoin_{a_l,a_r}	$f(t) \coloneqq$ (IJoin Expr.) + $\sum_{t_l,t_r}([t = t_l \cdot t_r] \times f_l(t_l) \times [IsNull(t_r)] \times$ $not(\sum_{t'_r}(f_r(t'_r) \times [a_l(t_l) = a_r(t'_r)] \times not([IsNull(a_l(t_l))])))$
RJoin_{a_l,a_r}	(symmetric to LJoin)
Dedup	$f(t) \coloneqq \|\|f_l(t)\|\|$

Table 3: The rules to translate the SQL operator into U-expression. Each U-expression is a function taking a tuple t and returning its multiplicity in the relation produced by the operator. f_l and f_r represent the U-expressions of the operator's left and right children, respectively. $t = t_l \cdot t_r$ is a predicate requiring t is the concatenation of t_l and t_r.

```
Example SQL  q5: ... FROM T WHERE T.x IN (S) AND T.c IN (S)
             q6: ... FROM T WHERE T.x IN (S)
Templates    qsrc:  InSuba(InSuba(r0, r1), r1)
             qdest: InSuba(r0, r1)
```

$$
\begin{aligned}
[\![q_{src}]\!](t) \coloneqq\, & r_0(t) \times not([IsNull(a(t))]) \times ||\sum_x r_1(x) \times [x = a(t)]|| \\
& \times not([IsNull(a(t))]) \times ||\sum_x r_1(x) \times [x = a(t)]|| \\
[\![q_{dest}]\!](t) \coloneqq\, & r_0(t) \times not([IsNull(a(t))]) \times ||\sum_x r_1(x) \times [x = a(t)]||
\end{aligned}
$$

Figure 4: The U-expressions of the rewrite rule in Figure 2. S denotes the entire subquery of "SELECT R.y FROM R". q_{src} and q_{dest} are the source and destination templates. The symbols of a, r_0 and r_1 in the templates represent T.x, T, and the relation produced by S in the example SQL accordingly.

Converting the query template into U-expression. WeTune translates each query template q to a function $[\![q]\!](t) : Tuple \longrightarrow \mathbb{N}$, which takes a tuple t as input and returns its multiplicity in the output relation of the query template. The multiplicity is represented as a U-expression. Unlike UDP [8] which performs the translation to U-expression for concrete queries, WeTune translates for symbolic query templates. The translation involves two steps:

Step 1. Translating the symbols in the query template. We use *uninterpreted* functions to enable the translation:

- Each relation symbol *rel* corresponds to a function $[\![r]\!](t) : Tuple \longrightarrow \mathbb{N}$ that takes a tuple t as input and returns its multiplicity in *rel*.
- Each attribute list symbol *attrs* corresponds to a function $[\![a]\!](t) : Tuple \longrightarrow Tuple$ that takes a tuple t as input, projects it on the attributes in *attrs* and returns the projected tuple.
- Each predicate symbol *pred* corresponds to a function $[\![p]\!](t) : Tuple \longrightarrow Bool$ that takes a tuple t as input and returns whether the tuple satisfies *pred*.

For brevity, we omit $[\![\,]\!]$ whenever there is no ambiguity. For example, $r(t)$ denotes the application of a relation function.

Step 2. translating the plan structure. This is done by recursion on the tree structure, as depicted by Algorithm 2. The function `ToUExpr` takes a (sub-)plan template as input. It returns the translated expression and a representative tuple of the output relation. For each operator, the algorithm recursively calculates the expressions of its children, then looks up in Table 3 to build its own expression based on its children's expressions. Figure 4 shows the translated U-expression of the example in Figure 2. We will discuss the *IsNull* predicate next.

Algorithm 2: Translate Plan to U-expression

```
ToUExpr($q$):
    <$f_l, t_l$> := ToUExpr(q.child[0]) //None if no child
    <$f_r, t_r$> := ToUExpr(q.child[1]) //None if single child
    return TranslateByTable3($q, f_l, t_l, f_r, t_r$)
```

Handling NULL. One of the biggest limitation of UDP's modeling of SQL query is its assumption that none of the attributes in a relation is NULL. Consequently, UDP cannot support the OUTER JOIN operator. According to our study, more than half of SQL queries collected from the web application involve such operators. To handle both NULL and OUTER JOIN, WeTune's translation of U-expression takes into consideration the impact of NULL on the operators, as shown in Table 3. Supporting other operators such as aggregation is trickier and left as future work.

For operator Input_r, the expression $r(\mathsf{NULL})$ returns the multiplicity of NULL tuples [2] in the input relation; For Proj_a, $f(\mathsf{NULL}) = \sum_{t_l}(f_l(t_l) \times [\mathsf{NULL} = a(t_l)])$ will return the multiplicity of tuples from t_l whose attribute a is NULL; For $\mathsf{Sel}_{p,a}$, some predicate p such as ">" and "<" will return unknown on evaluating NULL. When $a(t)$ is NULL and $p(a(t))$ returns *unknown*, $[p(a(t))]$ will return 0, which is the same as $[false]$. Here, we are able to treat the *unknown* in the three-valued logic as the *false* in two-valued logic, as $\mathsf{Sel}_{p,a}$ only evaluates the tuple that makes the predicate *true*; For Dedup, it returns 1 if there is at least one NULL tuple in the relation, otherwise 0.

To model the impact of NULL on the other operators in Table 3, WeTune introduces a new predicate *IsNull* to U-expression. When x is NULL, $IsNull(x)$ returns true and $[IsNull(x)]$ is 1. With such predicate, WeTune is able to filter out the NULL tuples. In detail, for $\mathsf{InSubSel}_a$ (IN-subquery), it uses the *IsNull* predicate to filter out the NULL tuple from the outer query. For INNER JOIN, it uses the *IsNull* to filter out the cases that left or right relation has NULL tuples. We will discuss how to handle NULL for OUTER JOIN in the next paragraph.

Supporting Outer Join operators. WeTune supports the OUTER JOIN operator by using the specific rule in Table 3 based on the modeling of NULL. Unlike INNER JOIN, OUTER JOIN keeps the rows that do not have a matching row on the other side and fills the void with NULL. For example, "x LEFT JOIN y ON x.a = y.b" keeps all rows from the left table x. For a left row that does not match any right row on x.a = y.b, NULL is appended as the right row. Hence, as shown in Table 3, the LEFT JOIN is the addition of two parts: (1) for those matched rows, the same as INNER JOIN; (2) for those non-matched rows, a product of three terms: "$f_l(t_l)$" describes the left rows being kept; "$[IsNull(t_r)]$" describes that NULL is appended as the right row; "$not(\sum_{t'_r}(\cdots))$" describes the non-matching condition. Figure 5 shows an example of translating a LEFT JOIN.

Representing constraints with FOL formulas. Each constraint is directly translated to a FOL formula according to Table 4. A set of constraints C is translated to the conjunction of its members:

$$\mathrm{ToFOL}(C) \triangleq \bigwedge_{c \in C} \mathrm{ToFOL}(c)$$

Constraint	Expression
$RelEq(r_1, r_2)$	$\forall t.r_1(t) = r_2(t)$
$AttrsEq(a_1, a_2)$	$\forall t.a_1(t) = a_2(t)$
$PredEq(p_1, p_2)$	$\forall t.p_1(t) = p_2(t)$
$SubAttrs(a_1, a_2)$	$\forall t.a_1(t) = a_1(a_2(t))$
$RefAttrs(r_1, a_1, r_2, a_2)$	$\forall t_1.((r_1(t_1) > 0 \land \neg(IsNull(a_1(t_1))))$ $\Rightarrow \exists t_2.(r_2(t_2) > 0 \land \neg(IsNull(a_2(t_2)))$ $\land[a_1(t_1) = a_2(t_2)]))$
$Unique(r, a)$	$(\forall t.r(t) \leq 1) \land (\forall t, t'.r(t) > 0 \land r(t') > 0 \land a(t) = a(t')$ $\Rightarrow t = t')$
$NotNull(r, a)$	$\forall t.r(t) > 0 \Rightarrow \neg(IsNull(a(t)))$

Table 4: Translation table from constraint to FOL formulas.

5.1.2 *Verification of the Rule Correctness*

After formalizing the query templates with U-expressions and the constraints with FOL formulas, the rule verifier will check a rule's correctness using the SMT solver. To do so, we need to formalize the correctness of the rule with FOL formulas.

Defining a rule's correctness. To formalize correctness, we need to first introduce the concept of *interpretation*, which specifies the meaning of the relation, predicate and attribute list symbols.

Definition 1 (Interpretation)**.** Given a query plan template q represented as a U-expression, an interpretation is an assignment of meaning to all symbols in q. We denote the concrete query plans under the interpretation I by q^I. Similarly, the truth value of a constraint set C under I is denoted by C^I.

Next, we define the correctness of a rule. Intuitively, if a rule is correct, its source and destination query templates should be equivalent under the rule's constraint set for any interpretations.

Definition 2 (Correctness of a rewrite rule)**.** Given a rule with two query plan templates $<q_{src}, q_{dest}>$ and a constraint set C, q_{src} and q_{dest} are equivalent under C iff the following formula holds.

$$\forall I.C^I \Rightarrow \forall t.q^I_{src}(t) = q^I_{dest}(t)$$

The formula "$\forall t.q^I_{src}(t) = q^I_{dest}(t)$" is consistent with the definition of bag equivalence [19]: two bags are equivalent iff. every tuple has the same multiplicity on both sides. Moreover, the outer quantifier "$\forall I$" requires the proposition to hold under any interpretation.

To prove query equivalence, UDP [8] relies on converting two U-expressions to their normalized forms and then establishing syntactic isomorphism between them. However, such syntactic isomorphism requires establishing a one-to-one equivalent relationship between the summations in the U-expressions, which can not be guaranteed for queries with operators like OUTER JOIN. Figure 5 shows an example. Since the two normalized expressions $[\![q_{src}]\!](t)$ and $[\![q_{dest}]\!](t)$ have different numbers of summations, UDP cannot establish the isomorphism needed for proving equivalence.

Logic-based decision procedure. Unlike UDP, WeTune uses a logic-based decision procedure, which translates the correctness definition (Definition 2) to a FOL formula and verifies it with the SMT solver. There are two challenges in realizing this approach.

[2] A tuple is NULL if all its attributes are NULL. A NULL attribute can be considered as a NULL tuple with only one attribute.

```
Example SQL  q7: SELECT T.* FROM T LEFT JOIN S ON T.k=S.k'
             q8: SELECT T.* FROM T
             Integrity Constraint: S.k' is unique key
```

Templates q_{src}: $Proj(LJoin_{a_0,a_1}(r_0,\ r_1))$

q_{dest}: $Proj(r_0)$

$$\begin{aligned}\llbracket q_{src} \rrbracket(t) &:= \sum_{x,y}([t = x] \times r_0(x) \times r_1(y) \times [a_0(x) = a_1(y)] \times NonNull(a_0(x))) \\ &\quad + \sum_{x,y}([t = x] \times r_0(x) \times [IsNull(y)] \\ &\quad \times not(\sum_{y'} r_1(y') \times [a_0(x) = a_1(y')] \times NonNull(a_0(x)))) \\ \llbracket q_{dest} \rrbracket(t) &:= \sum_{x}([t = x] \times r_0(x))\end{aligned}$$

Figure 5: A pair of equivalent queries that cannot be proved by UDP. The SQL query of $q7$ is collected from an open sourced web application Discourse [25]. q_{src} and q_{dest} are the templates. The symbols of r_0, r_1, a_0 and a_1 in the template can represent the relations of T and S, and the attributes of T.k and S.k' in $q7$ accordingly. $NonNull(\cdot)$ is an abbreviation of $not([IsNull(\cdot)])$ for simplicity.

The first challenge is how to translate the U-expression $q_{src}(t) = q_{dest}(t)$ to a FOL formula. WeTune performs the translation according to Table 5. The table shows the basic U-expressions used by the translation and their corresponding FOL formulas. These FOL formulas ensure the sufficient condition, i.e. for any interpretation that satisfies the FOL formula, then it also satisfies the U-expression. For a compound U-expression, WeTune performs recursive translation. Starting from $q_{src}(t) = q_{dest}(t)$, which defines the correctness of a rewrite rule, WeTune individually translates $q_{src}(t)$ and $q_{dest}(t)$ into FOL formulas. When performing the translation, it will find the matched form in Table 5 and replace it with the FOL formula[3] For example, Figure 6 shows the translated FOL formula when proving the equivalence of two queries in Figure 2.

$$\begin{aligned}\llbracket q_{src} \rrbracket(t) &= r_0(t) \times [\neg(IsNull(a(t))] \times [\exists x. r_1(x) \times [x = a(t)] > 0] \\ &\qquad \times [\exists x. r_1(x) \times [x = a(t)] > 0] \\ \llbracket q_{dest} \rrbracket(t) &= r_0(t) \times [\neg(IsNull(a(t))] \times [\exists x. r_1(x) \times [x = a(t)] > 0]\end{aligned}$$

Figure 6: The first-order logic formula of the example in Figure 2. $[\cdot]$ denotes the transformation from bool to natural number: $[b] := ite(b, 1, 0)$.

When encoding the FOL formulas for the SMT solver, we represent the tuple as an object with uninterpreted sort in SMTLIB; The relation, represented as $\llbracket R \rrbracket(t)$ in U-expression, is encoded as an uninterpreted function $R(t) : Tuple \longrightarrow \mathbb{N}$; the predicate is encoded as an uninterpreted function $P(t) : Tuple \longrightarrow Bool$.

When translating U-expressions to FOL formulas, the most difficult part is the translation of summation (the last two rows in Table 5). The unbounded summation domain makes it difficult to represent the value of a summation in a FOL formula. We address the problem based on the following insight. Since what matters is the equivalence relation, it is unnecessary to explicitly represent the value of a summation. Therefore, when proving $\sum_t f(t) = \sum_t f'(t)$, we aim to find the sufficient condition P such that $P \Rightarrow \sum_t f(t) = \sum_t f'(t)$. When P is proved to be true, then $\sum_t f(t) = \sum_t f'(t)$ must also hold. Furthermore, if such P does not involve summation, we can instead translate P into a FOL formula and prove it, effectively eliminating the summation.

U-expression	FOL formula
$f_1(t) = f_2(t)$	$\mathrm{Tr}(f_1(t)) = \mathrm{Tr}(f_2(t))$
$f_1(t) + f_2(t)$	$\mathrm{Tr}(f_1(t)) + \mathrm{Tr}(f_2(t))$
$f_1(t) \times f_2(t)$	$\mathrm{Tr}(f_1(t)) \times \mathrm{Tr}(f_2(t))$
$\|\|f(t)\|\|$	$ite(\mathrm{Tr}(f(t)) > 0, 1, 0)$
$not(f(t))$	$ite(\mathrm{Tr}(f(t)) > 0, 0, 1)$
$[p]$	$ite(\mathrm{Tr}(p), 1, 0)$
$\|\|\sum_t f(t)\|\|$	$ite(\exists t.\mathrm{Tr}(f(t)) > 0, 1, 0)$
$not(\sum_t f(t))$	$ite(\exists t.\mathrm{Tr}(f(t)) > 0, 0, 1)$
$\sum_t f(t) = 0$	$\forall t. f(t) = 0$
$\sum_t f(t) = 1$	$\exists t.(f(t) = 1 \wedge (\forall t'.t' \neq t \Rightarrow f(t') = 0))$
$\sum_t r(t) \times f(t)$ $= \sum_t r(t) \times g(t)$	$\forall t. r(t) \times \mathrm{Tr}(f(t)) = r(t) \times \mathrm{Tr}(g(t))$
$\sum_t r(t) \times f(t)$ $= \sum_{t,s} r(t) \times g(t) \times h(t, s)$	$\forall t.\ ((r(t) \times \mathrm{Tr}(f(t)) \neq r(t) \times \mathrm{Tr}(g(t))$ $\wedge r(t) \times \mathrm{Tr}(f(t)) = 0 \wedge \mathrm{Tr}(\sum_s h(t, s) = 0))$ $\vee\ (r(t) \times \mathrm{Tr}(f(t)) = r(t) \times \mathrm{Tr}(g(t))$ $\wedge (r(t) \times \mathrm{Tr}(f(t)) = 0 \vee \mathrm{Tr}(\sum_s h(t, s) = 1))))$

Table 5: Translation table from U-expression to FOL formulas. Function Tr recursively translates sub-expressions according to this table. The $ite(p, 0, 1)$ means if p is true, then the formula returns 0. Otherwise, the formula returns 1.

When P is not true, the verification fails and we consider the rewrite rule to be incorrect, which can prevent an incorrect rule from passing the verification. Specifically, we propose Theorem 5.1 and Theorem 5.2, corresponding to the last two rows in Table 5.

Theorem 5.1[4] eliminates the summation when the summed variables of two summations are aligned. It can be generalized to multiple summed variables. The proof can be found in our extended version [49].

Theorem 5.1.

$$\left(\forall I \forall t. r^I(t) \times f^I(t) = r^I(t) \times g^I(t)\right)$$
$$\Rightarrow \left(\forall I. \sum_t \left(r^I(t) \times f^I(t)\right) = \sum_t \left(r^I(t) \times g^I(t)\right)\right)$$

where r is a function that denotes a relation, f(t) and g(t) are arbitrary expressions. The superscript I indicates the interpretation of symbols under I.

Theorem 5.2 generalizes Theorem 5.1 to scenarios where the summed variables are not aligned. The proof can be found in our extended version [49].

Theorem 5.2.

$$\left(\forall I \forall t. \left(r^I(t) \times f^I(t) \neq r^I(t) \times g^I(t) \wedge r^I(t) \times f^I(t) = 0 \wedge \sum_s h^I(t, s) = 0\right)\right.$$
$$\left.\vee \left(r^I(t) \times f^I(t) = r^I(t) \times g^I(t) \wedge \left(r^I(t) \times f^I(t) = 0 \vee \sum_s h^I(t, s) = 1\right)\right)\right)$$
$$\Rightarrow \left(\forall I. \sum_t \left(r^I(t) \times f^I(t)\right) = \sum_{t,s} \left(r^I(t) \times g^I(t) \times h^I(t, s)\right)\right)$$

[3] Occasionally, WeTune can not find any match in Table 5. For example, if both the left and right child of a left join operator are an IN-subquery, then the U-expression contains two sums (aka $\sum$), which cannot be converted into FOL by Table 5. In this case, the verifier cannot prove the rule's correctness.

[4] This theorem fixes a mistake in our SIGMOD'22 paper. The theorem 5.1 in the SIGMOD'22 paper uses ⇔ rather than ⇒. However, this mistake does not corrupt the soundness of WeTune because soundness is guaranteed by ⇒.

where $h(t, s)$ is an arbitrary expression.

The second challenge is that universal quantifiers may make the proof undecidable and cause the SMT solver to timeout. When proving a FOL formula is a tautology, the SMT solver needs to exhaustively check all cases. For example, to prove $q_{src}(t)$ is always equivalent to $q_{dest}(t)$, it needs to check all possible interpretations, and under each interpretation it needs to further check every tuple t. In contrast, it is much easier proving a FOL formula is unsatisfiable (UNSAT), as the SMT solver will stop as soon as it finds a contradiction implying UNSAT, which can avoid exhaustive reasoning.

Therefore, given a rewrite rule of $<q_{src}, q_{dest}, C>$, WeTune verifies its correctness by proving that $\neg(C \Rightarrow \forall t.q_{src}(t) = q_{dest}(t))$ is UNSAT. As a result, instead of exhaustively checking all possible interpretations and tuples, the SMT solver only needs to find a contradiction that implies the formula above is UNSAT to prove rule correctness. Nevertheless, timeouts still occur because the SMT solver may fail to find the contradiction when the rule is either incorrect or too complicated. To evaluate the effect of the timeout, we test 232 rewrite rules from Calcite test suite which are already known to be correct. WeTune can successfully prove 73 rules without timeout. The others cannot be proved because they involve the operators or features that WeTune does not support. We also generate 100 incorrect rules by randomly selecting rules of Calcite and mutating their constraints to make them incorrect. WeTune encounters timeout for 96 of them, and only 4 rules are successfully proved to be incorrect.

In summary, by converting the correctness reasoning to be the UNSAT problem, WeTune is likely to perform the reasoning without timeout when the rules are correct. Our empirical evidence suggests that, for incorrect rules, WeTune tends to encounter timeout instead of giving a counterexample. Thus, WeTune conservatively considers those rules which cause timeout to be incorrect. Currently, we only focus on finding the correct rules and leave checking the incorrect ones without timeout as future work.

5.2 Integrating SPES

WeTune can also use an existing query equivalence checker like SPES [50] to further improve rule discovery in scenarios when its built-in verifier in Section 5 cannot prove a rule's correctness.

Compared to the built-in verifier, SPES additionally supports UNION and Aggregation operators. Therefore, we extend the rule enumerator in Section 4.1 to enumerate plan templates containing these two operators. The Aggregation operator is parameterized with 4 symbols: an attribute list symbol a_{group} for attributes used in the GROUP BY clause; another attribute list symbol a_{agg} for attributes used in the aggregate function; an uninterpreted function symbol f for the aggregate function; a predicate symbol p for the predicate in HAVING clause. For example, the SQL query "SELECT a_{group}, $f(a_{agg})$ FROM ... GROUP BY a_{group} HAVING $p(a_{group})$" is represented as a plan template $Agg_{a_{group}, a_{agg}, f, p}(...)$. WeTune also adds a new constraint $AggrEq(f_1, f_2)$ to indicate that two aggregate functions are equivalent. For the UNION operator, WeTune does not introduce any new symbols or constraints.

Given a rewrite rule $<q_{src}, q_{dest}, C>$, WeTune needs to convert it into inputs accepted by SPES. As SPES only takes the concrete SQL queries and does not recognize the constraint set C, WeTune concretizes the q_{src} and q_{dest} according to the constraint C with the following three steps: First, we assign names to each symbol in q_{src} and q_{dest} according to those equivalence constraints including *RelEq*, *AttrsEq*, *PredEq* and *AggrEq*. Specifically, it puts the equivalent symbols into the same set, and all symbols in the same set will share a randomly generated name. For example, in Figure 2, $t2$, $t2'$ and $t4$ could be assigned with the name "T2". $c0$, $c0'$ and $c1$ could be assigned with the name $C1$. Second, for each attribute, we find the relation it belongs to according to the *SubAttrs* constraints. If an attribute with the name c belongs to a relation with the name t, then we change the attribute name from c to $t.c$. For the example in Figure 2, the name of attribute list $c0$ will be changed to $T1.C1$. Third, we construct the schema definition according to the attributes of relations. In Figure 2, T1's schema has 1 column $C1$.

Features	SPES	Built-in
Aggregation	✓	✗
UNION	✓	✗
NULL	✓	✓
OUTER JOIN	✓	✓
Complex Predicate	✓	✗
Predicate with NOT/XOR/OR	✓	✗
Integrity Constraint	✗	✓
Different # of input tables	✗	✓

Table 6: Comparison of the capabilities of SPES and WeTune's built-in verifier. ✓ indicates a feature is supported or partially supported. Complex predicates refer to predicates with arithmetic operations and CASE.

Table 6 compares different features supported by SPES and the built-in verifier. Compared with SPES, the built-in verifier does not support Aggregate and predicate with NOT/XOR/OR due to our implementation restriction. It cannot support UNION because the U-expression of UNION with Projection, which is in the form "$\sum + \sum$", cannot be converted into the FOL formula. Similarly, the other set operators, such as INTERSECT and DIFFERENCE, also cannot be supported by the built-in verifier. It is unnecessary for the built-in verifier to support complex predicates, because its enumerated query templates do not have concrete predicates. Compared with the built-in verifier, SPES cannot handle SQL query with integrity constraints. Furthermore, SPES cannot prove the equivalence of two queries if they have different input tables, as these queries can not be normalized to the same algebraic representation which is necessary for the proof. For example, SPES cannot prove the equivalence between "SELECT DISTINCT T.* FROM T" with "SELECT DISTINCT T.* FROM T LEFT JOIN R On T.k = R.k", which can be proved by the built-in verifier. However, WeTune does not fully utilize SPES because the current rule enumerator can not enumerate plan templates having concrete aggregation functions, complex predicates and predicates connected by XOR, OR and NOT. This is considered as future work. A detailed comparison between the built-in verifier and SPES can be found in Section 8.5 and Table 7.

6 Selecting Useful Rules

After generating the promising rules, WeTune empirically evaluates their usefulness. The basic idea is to collect queries from real-world applications and evaluate which rules can rewrite those queries into a more efficient form. Ideally, rewrites should be done

$$\begin{aligned} q_{src} &: Proj_{a_0}(LJoin_{a_1,a_2}(Input_{r_0}, Input_{r_1})) \\ C &= \{SubAttrs(a_0, a_{r_0}), SubAttrs(a_1, a_{r_0}), \\ &\quad SubAttrs(a_2, a_{r_1}), Unique(r_1, a_2), \cdots\} \end{aligned}$$

```
q̂: SELECT T.a AS k FROM T LEFT JOIN S ON T.b=S.c
integrity constraint: S.c is unique key
```

Figure 7: An example of generated probing query $\hat{q}$.

by the database optimizer using existing rewriting techniques. However, to work with non-open-source databases, WeTune performs rewrites outside of the database.

WeTune's rewriting logic is based on simple greedy search. Given a query, it iteratively applies the rule that results in the most simplified target query (aka one with the fewest relational operators of each type). There can be more than one such rule at each iterative step. The iterative process terminates when no rewrites are possible. WeTune then obtains the cost estimate of each rewritten query from the existing database using the database's cost estimator, e.g., MySQL supports retrieving estimated cost by EXPLAIN EXTENDED command, MS SQL Server supports the same function by turning on SHOW_PLAN_ALL option. WeTune measures the actual performance of the most cost-efficient version of the query. To run the query and its rewritten version, we populate the database tables according to the schema and integrity constraints using randomly generated data. If the performance is improved by rewriting, then the corresponding rewrite rules are considered useful.

7 Additional Optimization

WeTune proposes two extra optimization strategies to reduce the redundant rules and eliminate ORDER BY in SQL statements.

Reducing redundant rules. Multiple rules can be composed to rewrite a query. For example, consider a query q and three rules R_1, R_2, R_3, we may get the same query p after rewriting q by (1) consecutively applying R_1, R_2 or (2) applying R_3. Thus, R_3 is redundant and can be replaced by the composite of R_1 and R_2. During rule discovery, it is desirable to reduce such redundant rules. Formally, given a set of rules $\mathbb{R}$ and a rule $R \in \mathbb{R}$, R is *reducible* under $\mathbb{R}$ if

$$\forall q.(Rewrite(\mathbb{R}, q) = Rewrite(\mathbb{R} - \{R\}, q))$$

It is impossible to check all queries. Instead, WeTune generates a concrete probing query $\hat{q}$ and concrete constraints according to R's source plan template and constraints. First, WeTune concretizes q_{src} to be $\hat{q}$ according to the steps of concretizing q_{src} for SPES (Section 5.2). Second, WeTune adds concrete integrity constraints for $\hat{q}$ according to *NotNull*, *Unique* and *RefAttrs* constraints in the rule. Figure 7 shows an example. $\hat{q}$ must be the minimal pattern that R is applicable to. i.e., any query that R is applicable to must contain the pattern $\hat{q}$. Thus, to decide the reducibility, it is sufficient to check whether the following condition is true: $Rewrite(\mathbb{R}, \hat{q}) = Rewrite(\mathbb{R} - \{R\}, \hat{q})$

Eliminating ORDER BY. Although WeTune does not support ORDER BY, WeTune can remove it from the query when it does not affect the query semantic. This is based on the insight that in SQL, an "ORDER BY" operator in the subquery can be useless when the outer query does not perform computations that can be affected by the order of tuples in the subquery result (e.g., aggregates a constant value from the subquery). In such cases, WeTune will directly eliminate ORDER BY in the statements.

8 Evaluation

The evaluation aims to answer the following questions:

Q1. How many new useful rules can WeTune discover?

Q2. How many new queries can WeTune optimize over existing systems for real-world applications?

Q3. How does WeTune's built-in verifier compare with SPES?

8.1 Experimental Setup

Implementation. We have built WeTune from scratch, which has about 40k lines of Java code. It takes the max plan template size as the parameter and outputs a set of non-reducible and promising rules. WeTune can also automatically check the usefulness of these rules by cooperating with existing database systems, including MySQL, PostgreSQL and MS SQL Server. Thus, besides the rule enumerator and verifier, it also contains a SQL parser, a query plan builder and a benchmark framework that evaluates SQL performance. Specifically, the built-in verifier is based on the SMT solver Z3 [12].

Generating Rules. WeTune enumerates all query plan templates up to size 4, yielding 3113 distinct templates. WeTune finds 1106 promising and non-reducible rules in 36 hours (on 120 CPU cores in total), among which 32 hours were spent in verification. Each potential rule takes about 50 ms on average to verify. For each rule, WeTune invokes the SMT solver 383 times on average to search for the most relaxed constraint set.

Workload. We use two workloads for the evaluation: one is a real-world workload, another is the Apache Calcite test suite [5]. For the real-world workload, we collect SQL queries from 20 open-source web applications on GitHub with the most stars for evaluation (the full list is included in our extended version [49]). These applications come from various genres, such as e-commerce, content management, discussion forum and social network. The number of contributors varies from 1 (1,902 stars) to 2,007 (22,203 stars). We collected 8,518 unique queries by running unit tests bundled with the source code. The Calcite test suite comprises 232 pairs of queries (464 individual queries) that are known equivalent, and all these queries can be rewritten by the rules in Calcite.

Evaluating Rules. When selecting useful rules, all rules are evaluated based on MS SQL Server 2019. The queries used to evaluate rules include both workloads described above. When executing queries on the database to evaluate latency, we populate four different tables. Two tables have 1K rows, while the other two have 100K rows. For every two tables with the same number of rows, one of them is populated with random data generated according to the uniform distribution, while the other one is populated with random data generated according to the Zipfian distribution with a skewed parameter of 1.25.

Testbed. All experiments are run on a server with a 20-core (2 sockets) Intel E5-2650 v3 CPU, 126 GB DRAM, and 1 TB SSD. The end-to-end latency of every query is evaluated on MS SQL Server 2019. We implement a dedicated client program that issues database queries and resides on the same machine as the database to eliminate network communication overhead. For a given query, the client randomizes the parameters in the query with extra care to avoid that every execution always directly fetches results from the database cache and to prevent the output result set is always empty. Each query is repeatedly executed 200 times in a closed loop (the first 100 times serve as warmup and are not counted into the result). When comparing WeTune with the rewriter in existing databases, we use the rewriter in Microsoft SQL Server 2019.

8.2 New Rewrite Rules

Table 7 shows the rules found by WeTune. There are 35 distinct rules which are useful for the evaluated queries. Among these rules, 9 rules are missing in MS SQL Server, 22 are missing in Calcite, and 5 are missing in both systems. 34 rules are discovered with the 8518 queries collected from the web application; only rule 35 is discovered with the queries in the Calcite test suite.

For the used verifier, 15 rules can be proved by both the built-in verifier and SPES. 16 rules can only be proved by the built-in verifier. SPES fails to prove these rules because 10 cases involve integrity constraints, 4 have mismatched input tables between the q_{src} and q_{dest}, and 2 cases (Rule 19 & 21) are related to SPES implementation. Taking rule 19 as an example, SPES fails to prove its correctness because we replace the predicate symbols in the query templates with the user-defined function. However, SPES does not consider that two user-defined functions are equivalent even if they have the same function name [5]. Compared with the built-in verifier, 4 rules can only be proved by SPES, as these rules have certain features only supported by SPES.

8.3 Queries Optimized by WeTune

We try to use generated rules of WeTune to rewrite both the queries studied in Section 2.2 and collected from real-world applications to see how many of them can be optimized by our discovered rules but cannot be optimized by existing systems.

The number of queries rewritten. For the 50 issues we have studied, WeTune can optimize 76% (38) of them, while MS SQL Server and Calcite can only optimize 46% (23) and 8%(4) of them. WeTune is unable to rewrite the remaining 12 queries due to two reasons. First, 9 of them need to rewrite the predicate expression or add a new predicate that does not equal to predicates in the original query (e.g., rewrite the predicate from "id IS NULL" to " project_id IS NULL" [1]). It requires finer-grained modeling and reasoning of the predicate expressions. The rest 3 of the queries need explicitly model the semantics of operators that WeTune currently does not support, including Aggregate and GROUP BY.

For 8518 queries collected from 20 real-world applications, WeTune can successfully rewrite 674 queries, among which 247 queries SQL Server fails to optimize (the other 427 queries can be effectively optimized). We manually check the remaining 7844 queries to investigate why WeTune cannot rewrite them. The main reason is that most queries (4251) only consist of SELECT-clause and WHERE-clause, without JOIN, subquery, Aggregate or any other clause. Optimizing such queries usually requires transformation at the physical execution level (e.g., index choice), which is beyond the scope of WeTune. The result shows WeTune can optimize more queries over existing databases.

We try to rewrite all 464 queries in the Calcite test suite. WeTune can rewrite 120 queries, among which 26 cannot be effectively optimized by MS SQL Server. For 23 queries of them, the rewriting performed by WeTune can achieve a 23.8% - 95.2% latency reduction than the rewriting of Calcite itself.

Latency reduction. To show the effectiveness of the optimizations found by WeTune, we compare the latency of the rewritten query with the original one on the same database for each of 273 queries that cannot be optimized by MS SQL Server (247 from applications plus 26 from Calcite test suite).

To know whether these rewrites are specific to certain workloads, we synthesized four workloads with varying table sizes (number of rows in the table) and data distribution, as summarized below:

	# of rows=10K	# of rows=1M
uniform dist.	workload A	workload B
zipfian dist. (θ = 1.5)	workload C	workload D

We implement a data generator inserting randomly generated rows into tables, which carefully maintains integrity constraints.

For workload A, WeTune can optimize at least 50% of the queries with more than 10% latency reduction and 17%, 18%, 30% reduction for workload B, C and D, respectively. WeTune can also optimize 13%-21% queries with at least a 90% latency reduction for all four workloads. This demonstrates that the rewrites are not specific to a certain table size or data distribution.

8.4 Case Study

Take the second query in Table 1 as an example of finding sequences of useful rules to optimize a query. First, WeTune iteratively generates new queries via rewrite rules, which takes 1.5s. Second, it consumes 5.3s to use the cost estimator in MS SQL Server to evaluate generated new queries. Then, we evaluate the end-to-end latency of every generated query by issuing it to the database. This step takes 12s, which indicates that we can find the sequence of rules that can produce the query with better performance within a reasonable amount of time.

Figure 8 shows each step of the best sequence of rewrite rules for the example above. First, the IN-selection is transformed to INNER JOIN in (2). Then, the predicate below the INNER JOIN is pulled up above it in (3). Usually, pushing down predicate below a JOIN is a standard optimization technique that can eagerly reduce the number of rows. However, in this case, pulling the predicate up enables new optimization opportunities that lead to a more efficient query. Next, the column "m.commit_id" used in the predicate is replaced by "n.commit_id" in (4). This replacement is guaranteed correct because the ON-condition "n.id=m.id" and the uniqueness property of primary key collectively imply that "m.commit_id=n.commit_id" holds for each row in the result set of the JOIN. Last, the table source t1 is eliminated by applying the JOIN-elimination rule.

Some rules that rewrite the source query plan to a similar plan are still useful, such as rule 17 and rule 18. For example, the query "Select T.y From R Inner Join T On R.x=T.y" will become "Select R.x From R Inner Join T On R.x=T.y" after applying rule 17. This rewrite allows WeTune to further apply rule 7 to eliminate the join when rule 7's constraints are met. Similarly, Rule 18 is useful because it might enable the subsequent application of rule 8.

8.5 Built-in Verifier vs. SPES

SPES is the state-of-the-art SQL equivalence verifier [50]. We try to compare the built-in verifier with SPES via two workloads: one

[5] A hypothesis of such design is SPES may aim to support some functions like RANDOM, which is not considered by WeTune.

No.	Source Plan Template	Destination Plan Template	Extra Constraints	Verifier	Calcite	MS
1	$\mathrm{Sel}_{p,r.a_0}(\mathrm{Proj}_{r.a_1}(r))$	$\mathrm{Proj}_{r.a_1}(\mathrm{Sel}_{p,r.a_0}(r))$	SubAttrs(a_0, a_1)	B	Y	Y
2	$\mathrm{Dedup}(\mathrm{Proj}_{r.a}(r))$	$\mathrm{Proj}_{r.a}(r)$	Unique(r, a)	W	N	Y
3	$\mathrm{Sel}_{p,r.a}(\mathrm{Sel}_{p,r.a}(r))$	$\mathrm{Sel}_{p,r.a}(r)$		B	Y	Y
4	$\mathrm{InSub}_{r_0.a_0}(\mathrm{InSub}_{r_0.a_0}(r_0, r_1), r_1)$	$\mathrm{InSub}_{r_0.a_0}(r_0, r_1)$		W	N	N
5	$\mathrm{Proj}_{r.a_0}(\mathrm{Sel}_{p,r.a_1}(\mathrm{Proj}_{r.a_2}(r)))$	$\mathrm{Proj}_{r.a_0}(\mathrm{Sel}_{p,r.a_1}(r))$	SubAttrs(a_0, a_2),SubAttrs(a_1, a_2)	B	Y	Y
6	$\mathrm{LJoin}_{r_0.a_0,r_1.a_1}(r_0, r_1)$	$\mathrm{IJoin}_{r_0.a_0,r_1.a_1}(r_0, r_1)$	RefAttrs(r_0, a_0, r_1, a_1),NotNull(r_0, a_0)	W	N	Y
7	$\mathrm{Proj}_{r_0.a_2}(\mathrm{IJoin}_{r_0.a_0,r_1.a_1}(r_0, r_1))$	$\mathrm{Proj}_{r_0.a_2}(r_0)$	RefAttrs(r_0, a_0, r_1, a_1),NotNull(r_0, a_0),Unique(r_1, a_1)	W	N	Y
8	$\mathrm{Proj}_{r_0.a_2}(\mathrm{Sel}_{p,r_0.a_3}(\mathrm{IJoin}_{r_0.a_0,r_1.a_1}(r_0, r_1)))$	$\mathrm{Proj}_{r_0.a_2}(\mathrm{Sel}_{p,r_0.a_3}(r_0))$	RefAttrs(r_0, a_0, r_1, a_1),NotNull(r_0, a_0),Unique(r_1, a_1)	W	N	C
9	$\mathrm{Dedup}(\mathrm{Proj}_{r_0.a_2}(\mathrm{IJoin}_{r_0.a_0,r_1.a_1}(r_0, r_1)))$	$\mathrm{Dedup}(\mathrm{Proj}_{r_0.a_2}(r_0))$	RefAttrs(r_0, a_0, r_1, a_1),NotNull(r_0, a_0)	W	N	Y
10	$\mathrm{Dedup}(\mathrm{Proj}_{r_0.a_2}(\mathrm{Sel}_{p,r_0.a_3}(\mathrm{IJoin}_{r_0.a_0,r_1.a_1}(r_0, r_1))))$	$\mathrm{Dedup}(\mathrm{Proj}_{r_0.a_2}(\mathrm{Sel}_{p,r_0.a_3}(r_0)))$	RefAttrs(r_0, a_0, r_1, a_1),NotNull(t_0, a_0)	W	N	C
11	$\mathrm{Proj}_{r_0.a_2}(\mathrm{LJoin}_{r_0.a_0,r_1.a_1}(r_0, r_1))$	$\mathrm{Proj}_{r_0.a_2}(a_0)$	Unique(r_1, a_1)	W	N	Y
12	$\mathrm{Proj}_{r_0.a_3}(\mathrm{Sel}_{p,r_0.a_2}(\mathrm{LJoin}_{r_0.a_0,r_1.a_1}(r_0, r_1)))$	$\mathrm{Proj}_{r_0.a_3}(\mathrm{Sel}_{p,r_0.a_2}(r_0))$	Unique(r1, a1)	W	N	Y
13	$\mathrm{Dedup}(\mathrm{Proj}_{r_0.a_2}(\mathrm{LJoin}_{r_0.a_0,r_1.a_1}(r_0, r_1)))$	$\mathrm{Dedup}(\mathrm{Proj}_{r_0.a_2}(a_0))$		W	N	Y
14	$\mathrm{Dedup}(\mathrm{Proj}_{r_0.a_3}(\mathrm{Sel}_{p,r_0.a_2}(\mathrm{LJoin}_{r_0.a_0,r_1.a_1}(r_0, r_1))))$	$\mathrm{Dedup}(\mathrm{Proj}_{r_0.a_3}(\mathrm{Sel}_{p,r_0.a_2}(r_0)))$		W	N	Y
15	$\mathrm{InSub}_{r.a}(r, \mathrm{Proj}_{r'.a}(r'))$	r	NotNull(r, a)	W	Y	N
16	$\mathrm{Proj}_{r.a_1}(\mathrm{IJoin}_{r.a_0,r'.a_0}(r, r')$	$\mathrm{Proj}_{r.a_1}(r)$	NotNull(r, a_0), Unique(r, a_0)	W	N	N
17	$\mathrm{Proj}_{r_1.a_1}(\mathrm{IJoin}_{r_0.a_0,r_1.a_1}(r_0, r_1))$	$\mathrm{Proj}_{r_0.a_0}(\mathrm{IJoin}_{r_0.a_0,r_1.a_1}(r_0, r_1))$		B	N	N
18	$\mathrm{Proj}_{r_1.a_1}(\mathrm{Sel}_{p,r_0.a_2}(\mathrm{IJoin}_{r_0.a_0,r_1.a_1}(r_0, r_1)))$	$\mathrm{Proj}_{r_0.a_0}(\mathrm{Sel}_{p,r_0.a_2}(\mathrm{IJoin}_{r_0.a_0,r_1.a_1}(r_0, r_1)))$		B	N	N
19	$\mathrm{Sel}_{p,r_1.a_1}(\mathrm{IJoin}_{r_0.a_0,r_1.a_1}(r_0, r_1))$	$\mathrm{Sel}_{p,r_0.a_0}(\mathrm{IJoin}_{r_0.a_0,r_1.a_1}(r_0, r_1))$		W	N	Y
20	$\mathrm{IJoin}_{r_1.a_1,r_2.a_2}(\mathrm{IJoin}_{r_0.a_0,r_1.a_1}(r_0, r_1), r_2)$	$\mathrm{IJoin}_{r_0.a_0,r_2.a_2}(\mathrm{IJoin}_{r_0.a_0,r_1.a_1}(r_0, r_1), r_2)$		B	N	Y
21	$\mathrm{LJoin}_{r_1.a_1,r_2.a_2}(\mathrm{IJoin}_{r_0.a_0,r_1.a_1}(r_0, r_1), r_2)$	$\mathrm{LJoin}_{r_0.a_0,r_2.a_2}(\mathrm{IJoin}_{r_0.a_0,r_1.a_1}(r_0, r_1), r_2)$		W	N	Y
22	$\mathrm{Proj}_{r_0.a_2}(\mathrm{IJoin}_{r_0.a_0,r_1.a_1}(r_0, r_1))$	$\mathrm{Proj}_{r_0.a_2}(\mathrm{IJoin}_{r_1.a_1,r_0.a_0}(r_1, r_0))$		B	Y	Y
23	$\mathrm{IJoin}_{r_0.a_0,r_1.a_1}(r_0, \mathrm{IJoin}_{r_1.a_2,r_2.a_3}(r_1, r_2))$	$\mathrm{IJoin}_{r_1.a_2,r_2.a_3}(\mathrm{IJoin}_{r_0.a_0,r_1.a_1}(r_0, r_1), r_2)$		B	Y	Y
24	$\mathrm{Proj}_{r_0.a_2}(\mathrm{InSub}_{r_0.a_0}(r_0, \mathrm{Proj}_{r_1.a_1}(r_1)))$	$\mathrm{Proj}_{r_0.a_2}(\mathrm{IJoin}_{r_0.a_0,r_1.a_1}(r_0, r_1))$	Unique(r_1, a_1)	B	Y	Y
25	$\mathrm{Proj}_{r_0.a_2}(\mathrm{IJoin}_{r_0.a_0,r_1.a_1}(r_0, \mathrm{Dedup}(\mathrm{Proj}_{r_1.a_1}(r_1))))$	$\mathrm{Proj}_{r_0.a_2}(\mathrm{InSub}_{r_0.a_0}(r_0, \mathrm{Proj}_{r_1.a_1}(r_1)))$		B	N	Y
26	$\mathrm{Dedup}(\mathrm{Proj}_{r_0.a_2}(\mathrm{IJoin}_{r_0.a_0,r_1.a_1}(r_0, \mathrm{Dedup}(r_1))))$	$\mathrm{Dedup}(\mathrm{Proj}_{r_0.a_2}(\mathrm{IJoin}_{r_0.a_0,r_1.a_1}(r_0, r_1)))$		W	N	Y
27	$\mathrm{IJoin}_{r_0.a_0,r_1.a_1}(r_0, \mathrm{Sel}_{p,r_1.a_2}(r_1))$	$\mathrm{Sel}_{p,r_1.a_2}(\mathrm{IJoin}_{r_0.a_0,r_1.a_1}(r_0, r_1))$		B	Y	Y
28	$\mathrm{Sel}_{p,r_1.a_2}(\mathrm{IJoin}_{r_0.a_0,r_1.a_1}(r_0, r_1))$	$\mathrm{IJoin}_{r_0.a_0,r_1.a_1}(r_0, \mathrm{Sel}_{p,r_1.a_2}(r_1))$		B	Y	Y
29	$\mathrm{Proj}_{r_0.a_2}(\mathrm{IJoin}_{r_0.a_0,r_1.a_1}(r_0, \mathrm{Proj}_{r_1.a_1}(r_1))))$	$\mathrm{Proj}_{r_0.a_2}(\mathrm{IJoin}_{r_0.a_0,r_1.a_1}(r_0, r_1)))$		B	N	Y
30	$\mathrm{Sel}_{p,r.a_0}(\mathrm{IJoin}_{r.a_1,r'.a_1}(r, r'))$	$\mathrm{Sel}_{p,r'.a_0}(\mathrm{IJoin}_{r.a_1,r'.a_1}(r, r'))$	Unique(r, a_1)	B	N	N
31	$\mathrm{Proj}_{r_0.a_0}(\mathrm{LJoin}_{r_0.a_1,r_1.a_2}(\mathrm{Proj}_{r_0.a_3}(r_0), r_1))$	$\mathrm{Proj}_{r_0.a_0}(\mathrm{LJoin}_{r_0.a_1,r_1.a_2}(r_0, r_1))$	SubAttrs(a_0, a_3), SubAttrs(a_1, a_3)	B	Y	Y
32	$\mathrm{Proj}_{r_0.a_0}(\mathrm{LJoin}_{r_0.a_1,r_1.a_2}(r_0, \mathrm{Proj}_{r_1.a_3}(r_1)))$	$\mathrm{Proj}_{r_0.a_0}(\mathrm{LJoin}_{r_0.a_1,r_1.a_2}(r_0, r_1))$	SubAttrs(a_2, a_3)	S	Y	Y
33	$\mathrm{Agg}_{r.a_0,r.a_1,f,p_0}(\mathrm{Filter}_{p_1,r.a_2}(\mathrm{Proj}_{r.a_3}(r)))$	$\mathrm{Agg}_{r.a_0,r.a_1,f,p_0}(\mathrm{Filter}_{p_1,r.a_2}(r))$	SubAttrs(a_0, a_3), SubAttrs(a_1, a_3), SubAttrs(a_2, a_3)	S	Y	Y
34	$\mathrm{Agg}_{r_0.a_0,r_0.a_1,f,p}(\mathrm{IJoin}_{r_0.a_2,r_1.a_3}(\mathrm{Proj}_{r_0.a_4}(r_0), r_1))$	$\mathrm{Agg}_{r_0.a_0,r_0.a_1,f,p}(\mathrm{IJoin}_{r_0.a_2,r_1.a_3}(r_0, r_1))$	SubAttrs(a_0, a_4), SubAttrs(a_1, a_4), SubAttrs(a_2, a_4)	S	N	Y
35	$\mathrm{Agg}_{r.a_0,r.a_1,f,p_0}(\mathrm{Filter}_{p_0,r.a_0}(r))$	$\mathrm{Agg}_{r.a_0,r.a_1,f,p_0}(r)$		S	Y	N

Table 7: Useful rewrite rules found by WeTune. The Verifier column indicates which verifier can prove the rule. *W* **means the built-in verifier,** *S* **means SPES, and** *B* **means both. The Calcite and MS columns indicate whether Calcite and MS SQL Server support these rules. The tree structure of the plan template is flattened by pre-order traversal. Each r_i represents an input table. Each a_i represents an attribute list. Each p represents a predicate.** IJoin **is the abbreviation for** InnerJoin, **and** LJoin **is for** LeftJoin. **Multiple occurrences of the same symbol (i.e., r_i, a_i, p) depict the equivalence constraint. Each $r_i.a_j$ stands for a constraint** SubAttrs(a_j, a_{r_i})**. Other types of constraints are listed in the column Extra Constraints. For rule 15 and 16,** r **and** r′ **denote two distinct occurrences of the same relation (e.g., "SELECT r.* FROM tbl AS r INNER JOIN tbl AS r' ON r.k=r'.k"). For rule 8 and 10, SQL Server can conditionally (C) eliminate the JOIN only if the attributes a_3 is different from a_0.**

is the 861 rules generated using the built-in verifier. Another is the 232 pairs of equivalent SQL in Calcite test suite.

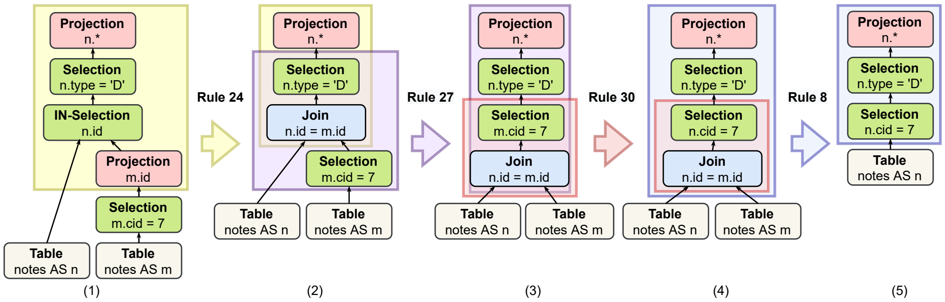
Rules generated by the built-in verifier. With the built-in verifier, WeTune is able to enumerate 861 promising and non-reducible rules. Among these rules, SPES successfully verifies 41 rules. Among the 820 that are not verified, 725 are due to that SPES's current implementation does not support integrity constraints and 95 are due to mismatched numbers of input tables on both sides.

Calcite Test Suite. The Calcite test suite comprises 232 pairs of queries. Each pair includes two equivalent SQL. SPES can successfully verify the equivalence of 95 query pairs of them, while the built-in verifier can prove the equivalence of 73 query pairs. Specifically, 55 pairs can be proved by both the built-in verifier and SPES. The number of pairs that the built-in verifier can prove is less than that can be proved by SPES because most rewrite rules in the test suite involve unsupported features of the built-in verifier, such as complex predicate. However, these features are supported by SPES.

9 Related Work

Query equivalence verification. Recently, researchers have proposed several systems [8–10, 51] to prove the equivalence of SQL queries formally. There are two approaches: some are based on proof assistants [8–10] while others are based on SMT [50, 51]. For the former approach, the state-of-the-art checker [8] uses an algebraic approach to verify the correctness of rules. Although their algebraic approach can model complex query structures based on the bag semantics, it lacks support for three-value-logic reasoning. For the SMT-based approach, recent work [50, 51] proposed *symbolic representation* of the query and leveraged the SMT solver to efficiently prove the equivalence of queries. But these systems lack the support of integrity constraints. WeTune overcomes some of their disadvantages by extending the algebraic approach with three-value-logic reasoning and supporting features such as integrity constraints.

Superoptimization. As a compiler optimization technique, superoptimization [2, 34] aims to find the optimal code sequence of a



```
Original    SELECT n.* FROM notes AS n WHERE n.type = 'D'
              AND n.id IN (SELECT m.id FROM notes AS m WHERE m.commit_id = 7)
Optimized   SELECT n.* FROM notes AS n WHERE n.type = 'D' AND n.commit_id = 7
```

Figure 8: An example sequence of rules discovered by WeTune to optimize a SQL query of a real-world query. The notes.id is the primary key of table notes. The cid in the figure is the abbreviation of commit_id. Each colored arrow in the figure represents a rule with the rewrite rule index above the arrow. The sub-plan in the same color is the source and destination plan template of the rule. The corresponding constraints are omitted.

set of instructions, which inspires WeTune. TASO [23] leverages superoptimization to find rewrite rules to rewrite deep neural networks (DNN). However, these works target different scenarios from query rewrite. WeTune needs to adopt different enumeration and verification methods. For example, in terms of enumeration, WeTune considers the relations of symbols (constraint) in SQL rewrite rules and enumerates all possible constraints. DNN operators have simpler parameters, and TASO only considers the relation between input/output operators. Ruler [38] has proposed a framework that abstracts the "search + verification" methods based on equality saturation to reduce candidate generation and selection cost. It is a general approach instead of specifically targeting SQL queries. WeTune could potentially use this framework to further improve the speed of discovery.

Query optimization. There has been a long line of work for query optimization, roughly divided into two categories depending on the search strategy of query plans. One is through a stratified approach [29, 42, 45], which first rewrites the logical query plan using transformation rules and then performs a cost-based search to map the logical plan to a physical plan. The other is through a unified approach [3, 14, 16, 18, 37], which unifies the logical to logical and logical to physical transformation into one stage. Recently, there has been a trend in adopting deep learning to query optimization [24, 26, 30–32, 47]. Given a set of rewrite rules, LearnedRewrite [52] is able to find the optimal rewrite order by using Monte Carlo Tree with learned cost models. However, these methods require manually written transformation rules and are orthogonal with the goal of WeTune.

10 Limitations

WeTune has the following two major limitations.

Incompleteness. One limitation is the incompleteness of the built-in verifier. First, due to the unbounded nature of the $\sum$ operator, U-expression fundamentally exceeds the expressiveness of FOL. Currently, only cases listed in table 5 can be translated to FOL and automatically verified by the SMT solver. How to automatically transform any U-expression into FOL formulas is left as future work. Second, the translated formula does not always fall into a decidable fragment of the SMT solver; thus may lead to timeout and consequently miss useful rules.

Unsupported SQL features. Another limitation is that the built-in verifier currently only supports rules containing operators listed in table 2. Furthermore, WeTune does not support recursive queries. As described in Section 5.2, some features are unsupported, such as UNION. Some features are partially supported, such as NULL. As described in Section 5.1.1, WeTune currently only considers the impact of NULL on operators in Table 3. Supporting more features is left as future work. Although some SQL features are unsupported, the soundness of WeTune holds for non-recursive queries. In other words, rewriting a query plan with rules obtained by WeTune can guarantee equivalent semantics. This is because, for every non-recursive query, even if it contains unsupported features, replacing its sub-plan without such features with another equivalent plan will not alter its original semantics.

11 Conclusion

This paper presents WeTune, which can automatically discover the rewrite rules for SQL queries. It enumerates all valid logical query plans up to a certain size to discover equivalent plans based on a new SMT-based verifier. We apply the rules discovered by WeTune on SQL queries collected from the 20 most popular open-source web applications on GitHub. WeTune successfully optimizes 247 queries that existing databases cannot optimize, resulting in substantial performance improvements.

Acknowledgments

We thank all anonymous reviewers for their constructive feedback and suggestions. This work is supported in part by National Natural Science Foundation of China (No. 62132014, 61902242, 62172272), the HighTech Support Program from Shanghai Committee of Science and Technology (No. 20ZR1428100). Ding Ding is supported by a DeepMind fellowship. Zhaoguo Wang (zhaoguowang@sjtu.edu.cn) is the corresponding author.

References

[1] Douglas Barbosa Alexandre. 2018. Improve the query performance to find unverified projects. https://gitlab.com/gitlab-org/gitlab/-/commit/11e93a9a4c2ac1b5bd4d32a93a949fc8afbcc449?merge_request_iid=5348.

[2] Sorav Bansal and Alex Aiken. 2006. Automatic generation of peephole superoptimizers. *ACM SIGARCH Computer Architecture News* 34, 5 (2006), 394–403.

[3] Edmon Begoli, Jesús Camacho-Rodríguez, Julian Hyde, Michael J. Mior, and Daniel Lemire. 2018. Apache Calcite: A Foundational Framework for Optimized Query Processing Over Heterogeneous Data Sources. In *Proceedings of the 2018 International Conference on Management of Data* (Houston, TX, USA) *(SIGMOD '18)*. Association for Computing Machinery, New York, NY, USA, 221–230. https://doi.org/10.1145/3183713.3190662

[4] Andreas Brandl. 2018. Replace OR clause with UNION. https://gitlab.com/gitlab-org/gitlab-foss/-/merge_requests/17088#note_59749778

[5] Apache Calcite. 2021. Calcite Test Suite. https://github.com/georgia-tech-db/spes/blob/main/testData/calcite_tests.json.

[6] Jack Chen, Samir Jindel, Robert Walzer, Rajkumar Sen, Nika Jimsheleishvilli, and Michael Andrews. 2016. The MemSQL Query Optimizer: A modern optimizer for real-time analytics in a distributed database. *Proceedings of the VLDB Endowment* 9, 13 (2016), 1401–1412.

[7] Hugh Darwen Chris J Date. 1996. *A Guide to the SQL Standard, Forth Edition*. Addison-Wesley Professional. https://www.amazon.com/Guide-SQL-Standard-4th/dp/0201964260

[8] Shumo Chu, Brendan Murphy, Jared Roesch, Alvin Cheung, and Dan Suciu. 2018. Axiomatic Foundations and Algorithms for Deciding Semantic Equivalences of SQL Queries. *Proc. VLDB Endow.* 11, 11 (July 2018), 1482–1495. https://doi.org/10.14778/3236187.3236200

[9] Shumo Chu, Chenglong Wang, Konstantin Weitz, and Alvin Cheung. 2017. Cosette: An Automated Prover for SQL.. In *Proceedings of the 8th Biennial Conference on Innovative Data Systems Research* (Chaminade, California, USA) *(CIDR '17)*.

[10] Shumo Chu, Konstantin Weitz, Alvin Cheung, and Dan Suciu. 2017. HoTTSQL: Proving Query Rewrites with Univalent SQL Semantics. *SIGPLAN Not.* 52, 6 (June 2017), 510–524. https://doi.org/10.1145/3140587.3062348

[11] Spree Commerce. 2021. Spree. https://github.com/spree/spree.

[12] Leonardo De Moura and Nikolaj Bjørner. 2008. Z3: An Efficient SMT Solver. In *Proceedings of the Theory and Practice of Software, 14th International Conference on Tools and Algorithms for the Construction and Analysis of Systems* (Budapest, Hungary) *(TACAS'08/ETAPS'08)*. Springer-Verlag, Berlin, Heidelberg, 337–340.

[13] Diaspora. 2021. Diaspora. https://github.com/diaspora/diaspora.

[14] Visweswara Sai Prashanth Dintyala, Arpit Narechania, and Joy Arulraj. to appear. SQLCheck: Automated Detection and Diagnosis of SQL Anti-Patterns. (to appear).

[15] GitLab. 2021. GitLab. https://gitlab.com/gitlab-org/gitlab.

[16] Goetz Graefe. 1995. The cascades framework for query optimization. *IEEE Data Eng. Bull.* 18, 3 (1995), 19–29.

[17] Goetz Graefe and David J DeWitt. 1987. The EXODUS optimizer generator. In *Proceedings of the 1987 ACM SIGMOD international conference on Management of data*. 160–172.

[18] Goetz Graefe and William J McKenna. 1993. The volcano optimizer generator: Extensibility and efficient search. In *Proceedings of IEEE 9th International Conference on Data Engineering*. IEEE, 209–218.

[19] Todd J Green, Grigoris Karvounarakis, and Val Tannen. 2007. Provenance semirings. In *Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. 31–40.

[20] Paolo Guagliardo and Leonid Libkin. 2017. A formal semantics of SQL queries, its validation, and applications. *Proceedings of the VLDB Endowment* 11, 1 (2017), 27–39.

[21] Adam Hegyi. 2020. Suboptimal Query in Gitlab. https://gitlab.com/gitlab-org/gitlab/-/merge_requests/34364.

[22] ISO/IEC 9075-1:2003 2003. *Information technology – Database languages – SQL – Part 1: Framework (SQL/Framework)*. Standard. International Organization for Standardization, Geneva, CH.

[23] Zhihao Jia, Oded Padon, James Thomas, Todd Warszawski, Matei Zaharia, and Alex Aiken. 2019. TASO: Optimizing Deep Learning Computation with Automatic Generation of Graph Substitutions. In *Proceedings of the 27th ACM Symposium on Operating Systems Principles* (Huntsville, Ontario, Canada) *(SOSP '19)*. Association for Computing Machinery, New York, NY, USA, 47–62. https://doi.org/10.1145/3341301.3359630

[24] Andreas Kipf, Thomas Kipf, Bernhard Radke, Viktor Leis, Peter Boncz, and Alfons Kemper. 2019. Learned cardinalities: Estimating correlated joins with deep learning. In *Proceedings of the 9th Biennial Conference on Innovative Data Systems Research* (Asilomar, California, USA) *(CIDR '19)*.

[25] Civilized Discourse Construction Kit. 2021. Discourse. https://github.com/discourse/discourse.

[26] Sanjay Krishnan, Zongheng Yang, Ken Goldberg, Joseph Hellerstein, and Ion Stoica. 2018. Learning to optimize join queries with deep reinforcement learning. *arXiv preprint arXiv:1808.03196* (2018).

[27] Alon Y Levy, Inderpal Singh Mumick, and Yehoshua Sagiv. 1994. Query optimization by predicate move-around. In *VLDB*. 96–107.

[28] G. Linden. 2006. Marissa Mayer at Web 2.0. http://glinden.blogspot.com/2006/11/marissa-mayer-at-web-20.html/.

[29] Guy M Lohman. 1988. Grammar-like functional rules for representing query optimization alternatives. *ACM SIGMOD Record* 17, 3 (1988), 18–27.

[30] Ryan Marcus, Parimarjan Negi, Hongzi Mao, Chi Zhang, Mohammad Alizadeh, Tim Kraska, Olga Papaemmanouil, and Nesime Tatbul. 2019. Neo: A learned query optimizer. *arXiv preprint arXiv:1904.03711* (2019).

[31] Ryan Marcus and Olga Papaemmanouil. 2018. Deep reinforcement learning for join order enumeration. In *Proceedings of the First International Workshop on Exploiting Artificial Intelligence Techniques for Data Management*. 1–4.

[32] Ryan Marcus and Olga Papaemmanouil. 2019. Towards a Hands-Free Query Optimizer through Deep Learning. In *Proceedings of the 9th Biennial Conference on Innovative Data Systems Research* (Asilomar, California, USA) *(CIDR '19)*.

[33] Henry Massalin. 1987. Superoptimizer: A Look at the Smallest Program. In *Proceedings of the Second International Conference on Architectual Support for Programming Languages and Operating Systems* (Palo Alto, California, USA) *(ASPLOS II)*. IEEE Computer Society Press, Washington, DC, USA, 122–126. https://doi.org/10.1145/36206.36194

[34] Henry Massalin. 1987. Superoptimizer: a look at the smallest program. *ACM SIGARCH Computer Architecture News* 15, 5 (1987), 122–126.

[35] Sean McGivern. 2017. Speed up counting approvers when some are specified. https://gitlab.com/gitlab-org/gitlab/-/merge_requests/2196.

[36] Inderpal Singh Mumick, Sheldon J Finkelstein, Hamid Pirahesh, and Raghu Ramakrishnan. 1990. Magic is relevant. *ACM SIGMOD Record* 19, 2 (1990), 247–258.

[37] M. Muralikrishna. 1992. Improved Unnesting Algorithms for Join Aggregate SQL Queries. In *Proceedings of the 18th International Conference on Very Large Data Bases (VLDB '92)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 91–102.

[38] Chandrakana Nandi, Max Willsey, Amy Zhu, Yisu Remy Wang, Brett Saiki, Adam Anderson, Adriana Schulz, Dan Grossman, and Zachary Tatlock. 2021. Rewrite Rule Inference Using Equality Saturation. *CoRR* abs/2108.10436 (2021). arXiv:2108.10436 https://arxiv.org/abs/2108.10436

[39] Nebulab. 2021. Solidus. https://github.com/solidusio/solidus.

[40] OpenProject. 2021. OpenProject. https://github.com/opf/openproject.

[41] Neil Patel. 2018. How Loading Time Affects Your Bottom Line. https://neilpatel.com/blog/speed-is-a-killer/.

[42] Hamid Pirahesh, Joseph M. Hellerstein, and Waqar Hasan. 1992. Extensible/Rule Based Query Rewrite Optimization in Starburst. (1992), 39–48. https://doi.org/10.1145/130283.130294

[43] Redmine. 2021. Redmine. https://github.com/redmine/redmine.

[44] Praveen Seshadri, Joseph M Hellerstein, Hamid Pirahesh, TY Cliff Leung, Raghu Ramakrishnan, Divesh Srivastava, Peter J Stuckey, and S Sudarshan. 1996. Cost-based optimization for magic: Algebra and implementation. In *Proceedings of the 1996 ACM SIGMOD international conference on Management of data*. 435–446.

[45] Praveen Seshadri, Joseph M. Hellerstein, Hamid Pirahesh, T. Y. Cliff Leung, Raghu Ramakrishnan, Divesh Srivastava, Peter J. Stuckey, and S. Sudarshan. 1996. Cost-Based Optimization for Magic: Algebra and Implementation. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data* (Montreal, Quebec, Canada) *(SIGMOD '96)*. Association for Computing Machinery, New York, NY, USA, 435–446. https://doi.org/10.1145/233269.233360

[46] Joshua Stein. 2021. Lobster. https://github.com/lobsters/lobsters.

[47] Michael Stillger, Guy M. Lohman, Volker Markl, and Mokhtar Kandil. 2001. LEO - DB2's LEarning Optimizer. In *Proceedings of the 27th International Conference on Very Large Data Bases (VLDB '01)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 19–28.

[48] Guoxiang Tan. 2017. PERF: Avoid 'NOT IN (<subquery>)' which can get really slow. https://github.com/discourse/discourse/commit/28148197d6467cdc7469409f961c00d4e32f4c41.

[49] Zhaoguo Wang, Zhou Zhou, et al. 2022. WeTune: Automatic Discovery and Verification of Query Rewrite Rules (The Extended Version). https://ipads.se.sjtu.edu.cn/_media/publications/wtune_extend.pdf.

[50] Qi Zhou, Joy Arulraj, Shamkant Navathe, William Harris, and Jinpeng Wu. 2020. SPES: A Two-Stage Query Equivalence Verifier. *arXiv preprint arXiv:2004.00481* (2020).

[51] Qi Zhou, Joy Arulraj, Shamkant Navathe, William Harris, and Dong Xu. 2019. Automated verification of query equivalence using satisfiability modulo theories. *Proceedings of the VLDB Endowment* 12, 11 (2019), 1276–1288.

[52] Xuanhe Zhou, Guoliang Li, Chengliang Chai, and Jianhua Feng. 2021. A Learned Query Rewrite System Using Monte Carlo Tree Search. *Proc. VLDB Endow.* 15, 1 (sep 2021), 46–58. https://doi.org/10.14778/3485450.3485456

Application		Genre	Stars	Contri-butors
Ruby	Discourse	Community discussion	33.2k	814
	GitLab	Code management	22.6k	1971
	diaspora	Social network	12.7k	367
	Spree	eCommerce	11.2k	850
	Refinery CMS	Content management	3.8k	419
	Redmine	Project management	4.1k	7
	Fat Free CRM	Customer relationship	3.3k	105
	Homeland	Community discussion	3.4k	134
	Solidus	eCommerce	3.7k	251
	Lobsters	Link sharing	2.9k	138
Java	Halo	Blogging	18.8k	61
	EL-ADMIN	Project management	15.4k	35
	FEBS-Shiro	Project management	5.4k	10
	Guns	Project management	3.2k	14
	Sagan	Blogging	2.7k	53
	ForestBlog	Blogging	3.1k	1
	Shopizer	eCommerce	2.1k	21
	PublicCMS	Content management	1.7k	9
	Broadleaf	eCommerce	1.5k	72
	pybss	Forum	1.3k	11

Table 8: Application corpus for evaluation

Appendix A Application Corpus

Table 8 lists the applications we used for evaluation. For each row, the column **Application** is the programming language and the name of the application. The column **Genre** is its category, summarized from the usage of the application. The column **Contributors** and **Stars** are the number of contributors and stars, accessed on May 7, 2021.

Appendix B Proof of Theorem 5.1

Theorem.

$$\left(\forall I \forall t. r^I(t) \times f^I(t) = r^I(t) \times g^I(t)\right) \tag{1}$$

$$\Rightarrow \left(\forall I. \sum_t \left(r^I(t) \times f^I(t)\right) = \sum_t \left(r^I(t) \times g^I(t)\right)\right)$$

where r is a function that denotes a relation, $f(x)$ and $g(y)$ are arbitrary expressions that do not involve r. The superscript I indicates the interpretation of symbols under I.

Proof. From $r^I(t) \times f^I(t) = r^I(t) \times g^I(t)$ holds for any tuple t and interpretation I, we can infer that $\sum_t \left(r^I(t) \times f^I(t)\right) = \sum_t \left(r^I(t) \times g^I(t)\right)$ holds for any interpretation I, which is apparent. □

Appendix C Proof of Theorem 5.2

Theorem.

$$\left(\forall I \forall t. \left(r^I(t) \times f^I(t) \neq r^I(t) \times g^I(t) \wedge r^I(t) \times f^I(t) = 0 \wedge \sum_s h^I(t,s) = 0\right)\right.$$

$$\left.\vee \left(r^I(t) \times f^I(t) = r^I(t) \times g^I(t) \wedge \left(r^I(t) \times f^I(t) = 0 \vee \sum_s h^I(t,s) = 1\right)\right)\right)$$

$$\Rightarrow \left(\forall I. \sum_t \left(r^I(t) \times f^I(t)\right) = \sum_{t,s} \left(r^I(t) \times g^I(t) \times h^I(t,s)\right)\right)$$

Proof sketch

Proof. $\sum_{x,y}$ is abbreviation of $\sum_x \sum_y$. Then, by theorem 5.1,

$$\forall I. \sum_t \left(r^I(t) \times f^I(t)\right) = \sum_{t,s} \left(r^I(t) \times g^I(t) \times h^I(t,s)\right)$$

is equivalent to

$$\forall I. \forall t.\ r^I(t) \times f(t) = r^I(t) \times g^I(t) \times \sum_s h^I(t,s)$$

Then, for any I_0 and t_0:

$$r^{I_0}(t0) \times f^{I_0}(t_0) = r^{I_0}(t0) \times g^{I_0}(t_0) \times \sum_s h^{I_0}(t_0,s) \tag{2}$$

We denote $r^{I_0}(t_0) \times f^{I_0}(t_0)$ by $E(t_0)$, $r^{I_0}(t_0) \times g^{I_0}(t_0)$ by $D(t_0)$. We discuss the condition where eq. (2) is true by cases of the relationship between $D(t_0)$ and $E(t_0)$:

(1) $E(t_0) = D(t_0)$
  (a) $E(t_0) = 0 \wedge D(t_0) = 0$
    eq. (2) is always true.
  (b) $E(t_0) \neq 0 \wedge D(t_0) \neq 0$
    eq. (2) is true if $\sum_s h^{I_0}(t_0,s) = 1$.
(2) $E(t_0) \neq D(t_0)$
  (a) $E(t_0) = 0 \wedge D(t_0) \neq 0$
    eq. (2) is true if $\sum_s h^{I_0}(t_0,s) = 0$.
  (b) $E(t_0) \neq 0 \wedge D(t_0) = 0$
    eq. (2) is never true.
  (c) $E(t_0) \neq 0 \wedge D(t_0) \neq 0$
    eq. (2) is true if the following proposition holds

$$\exists i \neq 0, j \neq 0. \left(D(t_0) = i \wedge \sum_s h^I(t_0,s) = j \wedge E(t_0) = i \times j\right)$$

Put the cases (1a)-(1b) and (2a)-(2b) together, we have got theorem. □

Appendix D List of Optimizations

Table 9 lists the 245 optimizations found by WeTune, which is missed by existing system. For each row, the column **App** is the application where the SQL statements was found. The column **Stmt Id** is a number used internally by WeTune to identify a unique statement. The column **Original** contains the original statement, while the column **Opt** contains the optimized statement.

Table 9: Optimized Statements List

App	Stmt Id	Original	Opt
broadleaf	90	select adminpermi0_.ADMIN_PERMISSION_ID as ADMIN_PE1_4_, adminpermi0_.DESCRIPTION as DESCRIPT2_4_, adminpermi0_.IS_FRIENDLY as IS_FRIEN3_4_, adminpermi0_.NAME as NAME4_4_, adminpermi0_.PERMISSION_TYPE as PERMISSI5_4_ from BLC_ADMIN_PERMISSION adminpermi0_ inner join BLC_ADMIN_ROLE_PERMISSION_XREF allroles1_ on adminpermi0_.ADMIN_PERMISSION_ID=allroles1_.ADMIN_PERMISSION_ID inner join BLC_ADMIN_ROLE adminrolei2_ on allroles1_.ADMIN_ROLE_ID=adminrolei2_.ADMIN_ROLE_ID where adminpermi0_.IS_FRIENDLY=1 and adminrolei2_.ADMIN_ROLE_ID=1 order by adminpermi0_.DESCRIPTION asc limit 50	SELECT * FROM (SELECT ‘adminpermi0_‘.‘admin_permission_id‘ AS ‘admin_pe1_4_‘, ‘adminpermi0_‘.‘description‘ AS ‘descript2_4_‘, ‘adminpermi0_‘.‘is_friendly‘ AS ‘is_frien3_4_‘, ‘adminpermi0_‘.‘name‘ AS ‘name4_4_‘, ‘adminpermi0_‘.‘permission_type‘ AS ‘permissi5_4_‘ FROM ‘blc_admin_permission‘ AS ‘adminpermi0_‘ INNER JOIN ‘blc_admin_role_permission_xref‘ AS ‘allroles1_‘ ON ‘adminpermi0_‘.‘admin_permission_id‘ = ‘allroles1_‘.‘admin_permission_id‘ WHERE ‘allroles1_‘.‘admin_role_id‘ = 1) AS ‘sub_0‘ WHERE ‘sub_0‘.‘is_frien3_4_‘ = 1 ORDER BY ‘sub_0‘.‘descript2_4_‘ ASC LIMIT 50
broadleaf	91	select count(adminpermi0_.ADMIN_PERMISSION_ID) as col_0_0_ from BLC_ADMIN_PERMISSION adminpermi0_ inner join BLC_ADMIN_ROLE_PERMISSION_XREF allroles1_ on adminpermi0_.ADMIN_PERMISSION_ID=allroles1_.ADMIN_PERMISSION_ID inner join BLC_ADMIN_ROLE adminrolei2_ on allroles1_.ADMIN_ROLE_ID=adminrolei2_.ADMIN_ROLE_ID where adminpermi0_.IS_FRIENDLY=1 and adminrolei2_.ADMIN_ROLE_ID=1	SELECT COUNT(‘adminpermi0_‘.‘admin_permission_id‘) AS ‘col_0_0_‘ FROM ‘blc_admin_permission‘ AS ‘adminpermi0_‘ INNER JOIN ‘blc_admin_role_permission_xref‘ AS ‘allroles1_‘ ON ‘adminpermi0_‘.‘admin_permission_id‘ = ‘allroles1_‘.‘admin_permission_id‘ WHERE ‘allroles1_‘.‘admin_role_id‘ = 1 AND ‘adminpermi0_‘.‘is_friendly‘ = 1
broadleaf	472	select staticasse0_.STATIC_ASSET_ID as STATIC_A1_171_, staticasse0_.ALT_TEXT as ALT_TEXT2_171_, staticasse0_.CREATED_BY as CREATED_3_171_, staticasse0_.DATE_CREATED as DATE_CRE4_171_, staticasse0_.DATE_UPDATED as DATE_UPD5_171_, staticasse0_.UPDATED_BY as UPDATED_6_171_, staticasse0_.FILE_EXTENSION as FILE_EXT7_171_, staticasse0_.FILE_SIZE as FILE_SIZ8_171_, staticasse0_.FULL_URL as FULL_URL9_171_, staticasse0_.MIME_TYPE as MIME_TY10_171_, staticasse0_.NAME as NAME11_171_, staticasse0_.STORAGE_TYPE as STORAGE12_171_, staticasse0_.TITLE as TITLE13_171_, staticasse0_1_.HEIGHT as HEIGHT1_77_, staticasse0_1_.WIDTH as WIDTH2_77_, case when staticasse0_1_.STATIC_ASSET_ID is not null then 1 when staticasse0_.STATIC_ASSET_ID is not null then 0 end as clazz_ from BLC_STATIC_ASSET staticasse0_ left outer join BLC_IMG_STATIC_ASSET staticasse0_1_ on staticasse0_.STATIC_ASSET_ID=staticasse0_1_.STATIC_ASSET_ID where 1=1 order by staticasse0_.STATIC_ASSET_ID asc limit 50	SELECT * FROM ‘blc_static_asset‘ AS ‘staticasse0_‘ LEFT JOIN ‘blc_img_static_asset‘ AS ‘staticasse0_1_‘ ON ‘staticasse0_‘.‘static_asset_id‘ = ‘staticasse0_1_‘.‘static_asset_id‘ ORDER BY ‘staticasse0_‘.‘static_asset_id‘ ASC LIMIT 50
diaspora	74	SELECT ‘o_auth_applications‘.‘id‘ FROM ‘o_auth_applications‘ INNER JOIN ‘authorizations‘ ON ‘o_auth_applications‘.‘id‘ = ‘authorizations‘.‘o_auth_application_id‘ WHERE ‘authorizations‘.‘user_id‘ = 1465	SELECT ‘authorizations‘.‘o_auth_application_id‘ AS ‘o_auth_application_id‘ FROM ‘authorizations‘ AS ‘authorizations‘ WHERE ‘authorizations‘.‘user_id‘ = 1465
diaspora	95	SELECT 1 AS one FROM ‘people‘ INNER JOIN ‘blocks‘ ON ‘people‘.‘id‘ = ‘blocks‘.‘person_id‘ WHERE ‘blocks‘.‘user_id‘ = 1 AND ‘people‘.‘id‘ = 1 LIMIT 1	SELECT 1 AS ‘one‘ FROM ‘blocks‘ AS ‘blocks‘ WHERE ‘blocks‘.‘person_id‘ = 1 AND ‘blocks‘.‘user_id‘ = 1 LIMIT 1
diaspora	182	SELECT DISTINCT ‘people‘.* FROM ‘people‘ INNER JOIN ‘profiles‘ ON ‘profiles‘.‘person_id‘ = ‘people‘.‘id‘ INNER JOIN ‘contacts‘ ‘contacts_people‘ ON ‘contacts_people‘.‘person_id‘ = ‘people‘.‘id‘ LEFT OUTER JOIN contacts ON contacts.user_id = 515 AND contacts.person_id = people.id WHERE (profiles.searchable = true OR contacts.user_id = 515) AND (profiles.full_name LIKE ’%a% contact%’ OR people.diaspora_handle LIKE ’acontact%’) AND ‘people‘.‘closed_account‘ = FALSE AND ‘contacts‘.‘user_id‘ = 515 AND ‘contacts‘.‘receiving‘ = TRUE LIMIT 15 OFFSET 0	SELECT * FROM (SELECT DISTINCT ‘people‘.‘id‘ AS ‘id‘, ‘people‘.‘guid‘ AS ‘guid‘, ‘people‘.‘diaspora_handle‘ AS ‘diaspora_handle‘, ‘people‘.‘serialized_public_key‘ AS ‘serialized_public_key‘, ‘people‘.‘owner_id‘ AS ‘owner_id‘, ‘people‘.‘created_at‘ AS ‘created_at‘, ‘people‘.‘updated_at‘ AS ‘updated_at‘, ‘people‘.‘closed_account‘ AS ‘closed_account‘, ‘people‘.‘fetch_status‘ AS ‘fetch_status‘, ‘people‘.‘pod_id‘ AS ‘pod_id‘ FROM ‘people‘ AS ‘people‘ INNER JOIN ‘contacts‘ AS ‘contacts‘ ON ‘contacts‘.‘user_id‘ = 515 AND ‘contacts‘.‘person_id‘ = ‘people‘.‘id‘ INNER JOIN ‘profiles‘ AS ‘profiles‘ ON ‘people‘.‘id‘ = ‘profiles‘.‘person_id‘ INNER JOIN ‘contacts‘ AS ‘contacts_people‘ ON ‘people‘.‘id‘ = ‘contacts_people‘.‘person_id‘ WHERE (‘profiles‘.‘searchable‘ = TRUE OR ‘contacts‘.‘user_id‘ = 515) AND (‘profiles‘.‘full_name‘ LIKE ’%a% contact%’ OR ‘people‘.‘diaspora_handle‘ LIKE ’acontact%’) AND ‘contacts‘.‘user_id‘ = 515 AND ‘contacts‘.‘receiving‘ = TRUE) AS ‘sub_0‘ WHERE ‘sub_0‘.‘closed_account‘ = FALSE LIMIT 15 OFFSET 0

diaspora	202	SELECT COUNT(DISTINCT ‘contacts‘.‘id‘) FROM ‘contacts‘ LEFT OUTER JOIN ‘people‘ ON ‘people‘.‘id‘ = ‘contacts‘.‘person_id‘ LEFT OUTER JOIN ‘profiles‘ ON ‘profiles‘.‘person_id‘ = ‘people‘.‘id‘ INNER JOIN ‘aspect_memberships‘ ON ‘aspect_memberships‘.‘aspect_id‘ = 250 AND ‘aspect_memberships‘.‘contact_id‘ = ‘contacts‘.‘id‘ WHERE ‘contacts‘.‘user_id‘ = 332	SELECT COUNT(‘contacts‘.‘id‘) FROM ‘contacts‘ AS ‘contacts‘ INNER JOIN ‘aspect_memberships‘ AS ‘aspect_memberships‘ ON ‘contacts‘.‘id‘ = ‘aspect_memberships‘.‘contact_id‘ WHERE ‘contacts‘.‘user_id‘ = 332 AND ‘aspect_memberships‘.‘aspect_id‘ = 250
diaspora	208	SELECT COUNT(*) FROM ‘tags‘ INNER JOIN ‘tag_followings‘ ON ‘tags‘.‘id‘ = ‘tag_followings‘.‘tag_id‘ WHERE ‘tag_followings‘.‘user_id‘ = 1	SELECT COUNT(*) FROM ‘tag_followings‘ AS ‘tag_followings‘ WHERE ‘tag_followings‘.‘user_id‘ = 1
diaspora	293	SELECT COUNT(*) FROM ‘contacts‘ INNER JOIN ‘aspect_memberships‘ ON ‘contacts‘.‘id‘ = ‘aspect_memberships‘.‘contact_id‘ WHERE ‘aspect_memberships‘.‘aspect_id‘ = 3	SELECT COUNT(*) FROM ‘aspect_memberships‘ AS ‘aspect_memberships‘ WHERE ‘aspect_memberships‘.‘aspect_id‘ = 3
diaspora	295	SELECT COUNT(DISTINCT ‘contacts‘.‘id‘) FROM ‘contacts‘ LEFT OUTER JOIN ‘people‘ ON ‘people‘.‘id‘ = ‘contacts‘.‘person_id‘ LEFT OUTER JOIN ‘profiles‘ ON ‘profiles‘.‘person_id‘ = ‘people‘.‘id‘ WHERE ‘contacts‘.‘user_id‘ = 1945	SELECT COUNT(‘contacts‘.‘id‘) FROM ‘contacts‘ AS ‘contacts‘ WHERE ‘contacts‘.‘user_id‘ = 1945
diaspora	299	SELECT COUNT(DISTINCT ‘people‘.‘id‘) FROM ‘people‘ INNER JOIN ‘profiles‘ ON ‘profiles‘.‘person_id‘ = ‘people‘.‘id‘ LEFT OUTER JOIN contacts ON contacts.user_id = 452 AND contacts.person_id = people.id WHERE (profiles.searchable = true OR contacts.user_id = 452) AND (profiles.full_name LIKE ’%closed%’ OR people.diaspora_handle LIKE ’closed%’) AND ‘people‘.‘closed_account‘ = FALSE	SELECT COUNT(DISTINCT ‘people‘.‘id‘) FROM ‘people‘ AS ‘people‘ INNER JOIN ‘profiles‘ AS ‘profiles‘ ON ‘people‘.‘id‘ = ‘profiles‘.‘person_id‘ INNER JOIN ‘contacts‘ AS ‘contacts‘ ON ‘contacts‘.‘user_id‘ = 452 AND ‘contacts‘.‘person_id‘ = ‘people‘.‘id‘ WHERE (‘profiles‘.‘searchable‘ = TRUE OR ‘contacts‘.‘user_id‘ = 452) AND (‘profiles‘.‘full_name‘ LIKE ’%closed%’ OR ‘people‘.‘diaspora_handle‘ LIKE ’closed%’) AND ‘people‘.‘closed_account‘ = FALSE
diaspora	301	SELECT COUNT(*) FROM ‘people‘ INNER JOIN ‘conversation_visibilities‘ ON ‘people‘.‘id‘ = ‘conversation_visibilities‘.‘person_id‘ WHERE ‘conversation_visibilities‘.‘conversation_id‘ = 239	SELECT COUNT(*) FROM ‘conversation_visibilities‘ AS ‘conversation_visibilities‘ WHERE ‘conversation_visibilities‘.‘conversation_id‘ = 239
diaspora	355	SELECT COUNT(*) FROM ‘people‘ INNER JOIN ‘profiles‘ ON ‘profiles‘.‘person_id‘ = ‘people‘.‘id‘ INNER JOIN ‘taggings‘ ON ‘taggings‘.‘taggable_id‘ = ‘profiles‘.‘id‘ AND ‘taggings‘.‘taggable_type‘ = ’Profile’ AND ‘taggings‘.‘context‘ = ’tags’ INNER JOIN ‘tags‘ ON ‘tags‘.‘id‘ = ‘taggings‘.‘tag_id‘ WHERE ‘tags‘.‘name‘ = ’cats’ AND (profiles.searchable IS TRUE)	SELECT COUNT(*) FROM ‘profiles‘ AS ‘profiles‘ INNER JOIN ‘taggings‘ AS ‘taggings‘ ON ‘profiles‘.‘id‘ = ‘taggings‘.‘taggable_id‘ INNER JOIN ‘tags‘ AS ‘tags‘ ON ‘taggings‘.‘tag_id‘ = ‘tags‘.‘id‘ WHERE ‘tags‘.‘name‘ = ’cats’ AND ‘taggings‘.‘taggable_type‘ = ’Profile’ AND ‘taggings‘.‘context‘ = ’tags’ AND ‘profiles‘.‘searchable‘ = TRUE
diaspora	399	SELECT people.id FROM ‘people‘ INNER JOIN ‘roles‘ ON ‘roles‘.‘person_id‘ = ‘people‘.‘id‘ WHERE ‘roles‘.‘name‘ = ’spotlight’	SELECT ‘roles‘.‘person_id‘ AS ‘person_id‘ FROM ‘roles‘ AS ‘roles‘ WHERE ‘roles‘.‘name‘ = ’spotlight’
diaspora	442	SELECT 1 AS one FROM ‘people‘ INNER JOIN ‘conversation_visibilities‘ ON ‘people‘.‘id‘ = ‘conversation_visibilities‘.‘person_id‘ WHERE ‘conversation_visibilities‘.‘conversation_id‘ = 98 AND ‘people‘.‘id‘ = 3 LIMIT 1	SELECT 1 AS ‘one‘ FROM ‘conversation_visibilities‘ AS ‘conversation_visibilities‘ WHERE ‘conversation_visibilities‘.‘person_id‘ = 3 AND ‘conversation_visibilities‘.‘conversation_id‘ = 98 LIMIT 1
diaspora	443	SELECT DISTINCT ‘posts‘.* FROM ‘posts‘ LEFT OUTER JOIN participations ON participations.target_id = posts.id AND participations.target_type = ’Post’ WHERE (‘participations‘.‘author_id‘ = 1047 OR ‘posts‘.‘author_id‘ = 1047) AND (posts.interacted_at <’2020-04-28 06:05:28’) AND ‘posts‘.‘type‘ IN (’StatusMessage’, ’Reshare’) ORDER BY posts.interacted_at DESC, posts.id DESC LIMIT 15	SELECT DISTINCT ‘posts‘.‘id‘ AS ‘id‘, ‘posts‘.‘author_id‘ AS ‘author_id‘, ‘posts‘.‘public‘ AS ‘public‘, ‘posts‘.‘guid‘ AS ‘guid‘, ‘posts‘.‘type‘ AS ‘type‘, ‘posts‘.‘text‘ AS ‘text‘, ‘posts‘.‘created_at‘ AS ‘created_at‘, ‘posts‘.‘updated_at‘ AS ‘updated_at‘, ‘posts‘.‘provider_display_name‘ AS ‘provider_display_name‘, ‘posts‘.‘root_guid‘ AS ‘root_guid‘, ‘posts‘.‘likes_count‘ AS ‘likes_count‘, ‘posts‘.‘comments_count‘ AS ‘comments_count‘, ‘posts‘.‘o_embed_cache_id‘ AS ‘o_embed_cache_id‘, ‘posts‘.‘reshares_count‘ AS ‘reshares_count‘, ‘posts‘.‘interacted_at‘ AS ‘interacted_at‘, ‘posts‘.‘tweet_id‘ AS ‘tweet_id‘, ‘posts‘.‘open_graph_cache_id‘ AS ‘open_graph_cache_id‘, ‘posts‘.‘tumblr_ids‘ AS ‘tumblr_ids‘ FROM ‘posts‘ AS ‘posts‘ INNER JOIN ‘participations‘ AS ‘participations‘ ON ‘participations‘.‘target_id‘ = ‘posts‘.‘id‘ AND ‘participations‘.‘target_type‘ = ’Post’ WHERE (‘participations‘.‘author_id‘ = 1047 OR ‘posts‘.‘author_id‘ = 1047) AND ‘posts‘.‘type‘ IN (?) AND ‘posts‘.‘interacted_at‘ <’2020-04-28 06:05:28’ ORDER BY ‘posts‘.‘interacted_at‘ DESC, ‘posts‘.‘id‘ DESC LIMIT 15

diaspora	444	SELECT DISTINCT ‘posts‘.* FROM ‘posts‘ LEFT OUTER JOIN participations ON participations.target_id = posts.id AND participations.target_type = ’Post’ WHERE (‘participations‘.‘author_id‘ = 1047 OR ‘posts‘.‘author_id‘ = 1047) AND (posts.interacted_at <’2020-04-28 06:05:28’) AND ‘posts‘.‘type‘ IN (’StatusMessage’, ’Reshare’) AND (posts.interacted_at <’2021-04-28T06:05:27.000Z’) ORDER BY posts.interacted_at DESC, posts.id DESC LIMIT 15	SELECT DISTINCT ‘posts‘.‘id‘ AS ‘id‘, ‘posts‘.‘author_id‘ AS ‘author_id‘, ‘posts‘.‘public‘ AS ‘public‘, ‘posts‘.‘guid‘ AS ‘guid‘, ‘posts‘.‘type‘ AS ‘type‘, ‘posts‘.‘text‘ AS ‘text‘, ‘posts‘.‘created_at‘ AS ‘created_at‘, ‘posts‘.‘updated_at‘ AS ‘updated_at‘, ‘posts‘.‘provider_display_name‘ AS ‘provider_display_name‘, ‘posts‘.‘root_guid‘ AS ‘root_guid‘, ‘posts‘.‘likes_count‘ AS ‘likes_count‘, ‘posts‘.‘comments_count‘ AS ‘comments_count‘, ‘posts‘.‘o_embed_cache_id‘ AS ‘o_embed_cache_id‘, ‘posts‘.‘reshares_count‘ AS ‘reshares_count‘, ‘posts‘.‘interacted_at‘ AS ‘interacted_at‘, ‘posts‘.‘tweet_id‘ AS ‘tweet_id‘, ‘posts‘.‘open_graph_cache_id‘ AS ‘open_graph_cache_id‘, ‘posts‘.‘tumblr_ids‘ AS ‘tumblr_ids‘ FROM ‘posts‘ AS ‘posts‘ INNER JOIN ‘participations‘ AS ‘participations‘ ON ‘participations‘.‘target_id‘ = ‘posts‘.‘id‘ AND ‘participations‘.‘target_type‘ = ’Post’ WHERE (‘participations‘.‘author_id‘ = 1047 OR ‘posts‘.‘author_id‘ = 1047) AND ‘posts‘.‘type‘ IN (?) AND ‘posts‘.‘interacted_at‘ <’2021-04-28T06:05:27.000Z’ AND ‘posts‘.‘interacted_at‘ <’2020-04-28 06:05:28’ ORDER BY ‘posts‘.‘interacted_at‘ DESC, ‘posts‘.‘id‘ DESC LIMIT 15
diaspora	453	SELECT DISTINCT ‘posts‘.* FROM ‘posts‘ LEFT OUTER JOIN participations ON participations.target_id = posts.id AND participations.target_type = ’Post’ WHERE (‘participations‘.‘author_id‘ = 1069 OR ‘posts‘.‘author_id‘ = 1069) AND (posts.interacted_at <’2020-04-28 06:05:33’) AND ‘posts‘.‘type‘ IN (’StatusMessage’, ’Reshare’) AND ‘posts‘.‘public‘ = TRUE AND (posts.interacted_at <’2021-04-28T06:05:32.000Z’) ORDER BY posts.interacted_at DESC, posts.id DESC LIMIT 15	SELECT DISTINCT ‘posts‘.‘id‘ AS ‘id‘, ‘posts‘.‘author_id‘ AS ‘author_id‘, ‘posts‘.‘public‘ AS ‘public‘, ‘posts‘.‘guid‘ AS ‘guid‘, ‘posts‘.‘type‘ AS ‘type‘, ‘posts‘.‘text‘ AS ‘text‘, ‘posts‘.‘created_at‘ AS ‘created_at‘, ‘posts‘.‘updated_at‘ AS ‘updated_at‘, ‘posts‘.‘provider_display_name‘ AS ‘provider_display_name‘, ‘posts‘.‘root_guid‘ AS ‘root_guid‘, ‘posts‘.‘likes_count‘ AS ‘likes_count‘, ‘posts‘.‘comments_count‘ AS ‘comments_count‘, ‘posts‘.‘o_embed_cache_id‘ AS ‘o_embed_cache_id‘, ‘posts‘.‘reshares_count‘ AS ‘reshares_count‘, ‘posts‘.‘interacted_at‘ AS ‘interacted_at‘, ‘posts‘.‘tweet_id‘ AS ‘tweet_id‘, ‘posts‘.‘open_graph_cache_id‘ AS ‘open_graph_cache_id‘, ‘posts‘.‘tumblr_ids‘ AS ‘tumblr_ids‘ FROM ‘posts‘ AS ‘posts‘ INNER JOIN ‘participations‘ AS ‘participations‘ ON ‘participations‘.‘target_id‘ = ‘posts‘.‘id‘ AND ‘participations‘.‘target_type‘ = ’Post’ WHERE (‘participations‘.‘author_id‘ = 1069 OR ‘posts‘.‘author_id‘ = 1069) AND ‘posts‘.‘type‘ IN (?) AND ‘posts‘.‘public‘ = TRUE AND ‘posts‘.‘interacted_at‘ <’2021-04-28T06:05:32.000Z’ AND ‘posts‘.‘interacted_at‘ <’2020-04-28 06:05:33’ ORDER BY ‘posts‘.‘interacted_at‘ DESC, ‘posts‘.‘id‘ DESC LIMIT 15
diaspora	460	SELECT DISTINCT people.id FROM ‘people‘ INNER JOIN ‘contacts‘ ON ‘contacts‘.‘person_id‘ = ‘people‘.‘id‘ INNER JOIN ‘aspect_memberships‘ ON ‘aspect_memberships‘.‘contact_id‘ = ‘contacts‘.‘id‘ WHERE 1=0	SELECT DISTINCT ‘contacts‘.‘person_id‘ AS ‘person_id‘ FROM ‘contacts‘ AS ‘contacts‘ INNER JOIN ‘aspect_memberships‘ AS ‘aspect_memberships‘ ON ‘contacts‘.‘id‘ = ‘aspect_memberships‘.‘contact_id‘
diaspora	551	SELECT COUNT(*) FROM ‘users‘ INNER JOIN ‘people‘ ON ‘people‘.‘owner_id‘ = ‘users‘.‘id‘ INNER JOIN ‘profiles‘ ON ‘profiles‘.‘person_id‘ = ‘people‘.‘id‘ WHERE ‘profiles‘.‘birthday‘ >’2007-04-29’	SELECT COUNT(*) FROM ‘profiles‘ AS ‘profiles‘ WHERE ‘profiles‘.‘birthday‘ >’2007-04-29’
diaspora	574	SELECT ‘people‘.* FROM ‘people‘ INNER JOIN ‘profiles‘ ON ‘profiles‘.‘person_id‘ = ‘people‘.‘id‘ LEFT OUTER JOIN contacts ON contacts.user_id = 1 AND contacts.person_id = people.id WHERE (profiles.full_name LIKE ’%korth%’ OR people.diaspora_handle LIKE ’korth%’) AND (profiles.searchable = true OR contacts.user_id = 1) AND ‘people‘.‘closed_account‘ = FALSE ORDER BY contacts.user_id IS NULL, profiles.last_name ASC, profiles.first_name ASC LIMIT 15	SELECT ‘people‘.‘id‘ AS ‘id‘, ‘people‘.‘guid‘ AS ‘guid‘, ‘people‘.‘diaspora_handle‘ AS ‘diaspora_handle‘, ‘people‘.‘serialized_public_key‘ AS ‘serialized_public_key‘, ‘people‘.‘owner_id‘ AS ‘owner_id‘, ‘people‘.‘created_at‘ AS ‘created_at‘, ‘people‘.‘updated_at‘ AS ‘updated_at‘, ‘people‘.‘closed_account‘ AS ‘closed_account‘, ‘people‘.‘fetch_status‘ AS ‘fetch_status‘, ‘people‘.‘pod_id‘ AS ‘pod_id‘ FROM ‘people‘ AS ‘people‘ INNER JOIN ‘profiles‘ AS ‘profiles‘ ON ‘people‘.‘id‘ = ‘profiles‘.‘person_id‘ INNER JOIN ‘contacts‘ AS ‘contacts‘ ON ‘contacts‘.‘user_id‘ = 1 AND ‘contacts‘.‘person_id‘ = ‘people‘.‘id‘ WHERE (‘profiles‘.‘searchable‘ = TRUE OR ‘contacts‘.‘user_id‘ = 1) AND (‘profiles‘.‘full_name‘ LIKE ’%korth%’ OR ‘people‘.‘diaspora_handle‘ LIKE ’korth%’) AND ‘people‘.‘closed_account‘ = FALSE ORDER BY ‘contacts‘.‘user_id‘ IS NULL, ‘profiles‘.‘last_name‘ ASC, ‘profiles‘.‘first_name‘ ASC LIMIT 15

diaspora	577	SELECT COUNT(DISTINCT ‘people‘.‘id‘) FROM ‘people‘ WHERE ‘people‘.‘id‘ IN (SELECT ‘people‘.‘id‘ FROM ‘people‘ INNER JOIN ‘contacts‘ ON ‘contacts‘.‘person_id‘ = ‘people‘.‘id‘ INNER JOIN ‘aspect_memberships‘ ON ‘aspect_memberships‘.‘contact_id‘ = ‘contacts‘.‘id‘ WHERE ‘people‘.‘id‘ IN (SELECT ‘people‘.‘id‘ FROM ‘people‘ INNER JOIN ‘profiles‘ ON ‘profiles‘.‘person_id‘ = ‘people‘.‘id‘ INNER JOIN ‘contacts‘ ‘contacts_people‘ ON ‘contacts_people‘.‘person_id‘ = ‘people‘.‘id‘ INNER JOIN ‘aspect_memberships‘ ON ‘aspect_memberships‘.‘contact_id‘ = ‘contacts_people‘.‘id‘ LEFT OUTER JOIN contacts ON contacts.user_id = 488 AND contacts.person_id = people.id WHERE (profiles.searchable = true OR contacts.user_id = 488) AND (profiles.full_name LIKE ’%my% aspect% contact%’ OR people.diaspora_handle LIKE ’myaspectcontact%’) AND ‘people‘.‘closed_account‘ = FALSE AND ‘contacts‘.‘user_id‘ = 488 AND ‘aspect_memberships‘.‘aspect_id‘ = 321) AND ‘contacts‘.‘user_id‘ = 488 AND ‘aspect_memberships‘.‘aspect_id‘ = 322)	SELECT COUNT(DISTINCT ‘sub_0‘.‘id‘) FROM ‘people‘ AS ‘people‘ INNER JOIN (SELECT ‘people_0_0‘.‘id‘ AS ‘id‘, ‘people_0_0‘.‘guid‘ AS ‘guid‘, ‘people_0_0‘.‘diaspora_handle‘ AS ‘diaspora_handle‘, ‘people_0_0‘.‘serialized_public_key‘ AS ‘serialized_public_key‘, ‘people_0_0‘.‘owner_id‘ AS ‘owner_id‘, ‘people_0_0‘.‘created_at‘ AS ‘created_at‘, ‘people_0_0‘.‘updated_at‘ AS ‘updated_at‘, ‘people_0_0‘.‘closed_account‘ AS ‘closed_account‘, ‘people_0_0‘.‘fetch_status‘ AS ‘fetch_status‘, ‘people_0_0‘.‘pod_id‘ AS ‘pod_id‘, ‘contacts‘.‘id‘ AS ‘id_0‘, ‘contacts‘.‘user_id‘ AS ‘user_id‘, ‘contacts‘.‘person_id‘ AS ‘person_id‘, ‘contacts‘.‘created_at‘ AS ‘created_at_0‘, ‘contacts‘.‘updated_at‘ AS ‘updated_at_0‘, ‘contacts‘.‘sharing‘ AS ‘sharing‘, ‘contacts‘.‘receiving‘ AS ‘receiving‘, ‘aspect_memberships‘.‘id‘ AS ‘id_1‘, ‘aspect_memberships‘.‘aspect_id‘ AS ‘aspect_id‘, ‘aspect_memberships‘.‘contact_id‘ AS ‘contact_id‘, ‘aspect_memberships‘.‘created_at‘ AS ‘created_at_1‘, ‘aspect_memberships‘.‘updated_at‘ AS ‘updated_at_1‘ FROM ‘people‘ AS ‘people_0_0‘ INNER JOIN ‘contacts‘ AS ‘contacts‘ ON ‘people_0_0‘.‘id‘ = ‘contacts‘.‘person_id‘ INNER JOIN ‘aspect_memberships‘ AS ‘aspect_memberships‘ ON ‘contacts‘.‘id‘ = ‘aspect_memberships‘.‘contact_id‘ WHERE ‘people_0_0‘.‘id‘ IN (SELECT ‘people_1‘.‘id‘ AS ‘id‘ FROM ‘people‘ AS ‘people_1‘ INNER JOIN ‘profiles‘ AS ‘profiles‘ ON ‘people_1‘.‘id‘ = ‘profiles‘.‘person_id‘ INNER JOIN ‘contacts‘ AS ‘contacts_people‘ ON ‘people_1‘.‘id‘ = ‘contacts_people‘.‘person_id‘ INNER JOIN ‘aspect_memberships‘ AS ‘aspect_memberships_0‘ ON ‘contacts_people‘.‘id‘ = ‘aspect_memberships_0‘.‘contact_id‘ INNER JOIN ‘contacts‘ AS ‘contacts_0‘ ON ‘contacts_0‘.‘user_id‘ = 488 AND ‘contacts_0‘.‘person_id‘ = ‘people_1‘.‘id‘ WHERE (‘profiles‘.‘searchable‘ = TRUE OR ‘contacts_0‘.‘user_id‘ = 488) AND (‘profiles‘.‘full_name‘ LIKE ’%my% aspect% contact%’ OR ‘people_1‘.‘diaspora_handle‘ LIKE ’myaspectcontact%’) AND ‘people_1‘.‘closed_account‘ = FALSE AND ‘contacts_0‘.‘user_id‘ = 488 AND ‘aspect_memberships_0‘.‘aspect_id‘ = 321) AND ‘contacts‘.‘user_id‘ = 488 AND ‘aspect_memberships‘.‘aspect_id‘ = 322) AS ‘sub_0‘ ON ‘people‘.‘id‘ = ‘sub_0‘.‘id‘

diaspora	583	SELECT ‘people‘.* FROM ‘people‘ INNER JOIN ‘profiles‘ ON ‘profiles‘.‘person_id‘ = ‘people‘.‘id‘ LEFT OUTER JOIN contacts ON contacts.user_id = 1 AND contacts.person_id = people.id WHERE (profiles.full_name LIKE ’%korth%’ OR people.diaspora_handle LIKE ’korth%’) AND (profiles.searchable = true OR contacts.user_id = 1) AND ‘people‘.‘closed_account‘ = FALSE ORDER BY contacts.user_id IS NULL, profiles.last_name ASC, profiles.first_name ASC LIMIT 15 OFFSET 0	SELECT ‘people‘.‘id‘ AS ‘id‘, ‘people‘.‘guid‘ AS ‘guid‘, ‘people‘.‘diaspora_handle‘ AS ‘diaspora_handle‘, ‘people‘.‘serialized_public_key‘ AS ‘serialized_public_key‘, ‘people‘.‘owner_id‘ AS ‘owner_id‘, ‘people‘.‘created_at‘ AS ‘created_at‘, ‘people‘.‘updated_at‘ AS ‘updated_at‘, ‘people‘.‘closed_account‘ AS ‘closed_account‘, ‘people‘.‘fetch_status‘ AS ‘fetch_status‘, ‘people‘.‘pod_id‘ AS ‘pod_id‘ FROM ‘people‘ AS ‘people‘ INNER JOIN ‘profiles‘ AS ‘profiles‘ ON ‘people‘.‘id‘ = ‘profiles‘.‘person_id‘ INNER JOIN ‘contacts‘ AS ‘contacts‘ ON ‘contacts‘.‘user_id‘ = 1 AND ‘contacts‘.‘person_id‘ = ‘people‘.‘id‘ WHERE (‘profiles‘.‘searchable‘ = TRUE OR ‘contacts‘.‘user_id‘ = 1) AND (‘profiles‘.‘full_name‘ LIKE ’%korth%’ OR ‘people‘.‘diaspora_handle‘ LIKE ’korth%’) AND ‘people‘.‘closed_account‘ = FALSE ORDER BY ‘contacts‘.‘user_id‘ IS NULL, ‘profiles‘.‘last_name‘ ASC, ‘profiles‘.‘first_name‘ ASC LIMIT 15 OFFSET 0
diaspora	585	SELECT COUNT(DISTINCT ‘people‘.‘id‘) FROM ‘people‘ INNER JOIN ‘profiles‘ ON ‘profiles‘.‘person_id‘ = ‘people‘.‘id‘ INNER JOIN ‘contacts‘ ‘contacts_people‘ ON ‘contacts_people‘.‘person_id‘ = ‘people‘.‘id‘ LEFT OUTER JOIN contacts ON contacts.user_id = 491 AND contacts.person_id = people.id WHERE (profiles.searchable = true OR contacts.user_id = 491) AND (profiles.full_name LIKE ’%a% contact%’ OR people.diaspora_handle LIKE ’acontact%’) AND ‘people‘.‘closed_account‘ = FALSE AND ‘contacts‘.‘user_id‘ = 491	SELECT COUNT(DISTINCT ‘people‘.‘id‘) FROM ‘people‘ AS ‘people‘ INNER JOIN ‘contacts‘ AS ‘contacts‘ ON ‘contacts‘.‘user_id‘ = 491 AND ‘contacts‘.‘person_id‘ = ‘people‘.‘id‘ INNER JOIN ‘contacts‘ AS ‘contacts_people‘ ON ‘people‘.‘id‘ = ‘contacts_people‘.‘person_id‘ INNER JOIN ‘profiles‘ AS ‘profiles‘ ON ‘people‘.‘id‘ = ‘profiles‘.‘person_id‘ WHERE (‘profiles‘.‘searchable‘ = TRUE OR ‘contacts‘.‘user_id‘ = 491) AND (‘profiles‘.‘full_name‘ LIKE ’%a% contact%’ OR ‘people‘.‘diaspora_handle‘ LIKE ’acontact%’) AND ‘people‘.‘closed_account‘ = FALSE AND ‘contacts‘.‘user_id‘ = 491
discourse	58	SELECT "groups"."id" FROM "groups" INNER JOIN "invited_groups" ON "groups"."id" = "invited_groups"."group_id" WHERE "invited_groups"."invite_id" = 1318	SELECT "invited_groups"."group_id" AS "group_id" FROM "invited_groups" AS "invited_groups" WHERE "invited_groups"."invite_id" = 1318
discourse	118	SELECT "groups"."id" FROM "groups" INNER JOIN "category_groups" ON "groups"."id" = "category_groups"."group_id" WHERE "category_groups"."category_id" = 3086	SELECT "category_groups"."group_id" AS "group_id" FROM "category_groups" AS "category_groups" WHERE "category_groups"."category_id" = 3086
discourse	123	SELECT distinct categories.id FROM "categories" INNER JOIN "category_groups" ON "categories"."id" = "category_groups"."category_id" INNER JOIN "groups" ON "category_groups"."group_id" = "groups"."id" INNER JOIN "group_users" ON "groups"."id" = "group_users"."group_id" WHERE "group_users"."user_id" = 86	SELECT DISTINCT "category_groups"."category_id" AS "category_id" FROM "category_groups" AS "category_groups" INNER JOIN "group_users" AS "group_users" ON "category_groups"."group_id" = "group_users"."group_id" WHERE "group_users"."user_id" = 86
discourse	181	SELECT "tag_users"."user_id" FROM "tag_users" INNER JOIN "tags" ON "tag_users"."tag_id" = "tags"."id" INNER JOIN "topic_tags" ON "tags"."id" = "topic_tags"."tag_id" WHERE "topic_tags"."topic_id" = 15596 AND "tag_users"."notification_level" = 4	SELECT "tag_users"."user_id" AS "user_id" FROM "tag_users" AS "tag_users" INNER JOIN "topic_tags" AS "topic_tags" ON "tag_users"."tag_id" = "topic_tags"."tag_id" WHERE "topic_tags"."topic_id" = 15596 AND "tag_users"."notification_level" = 4
discourse	276	SELECT "users"."id" FROM "users" INNER JOIN "ignored_users" ON "users"."id" = "ignored_users"."ignored_user_id" WHERE "ignored_users"."user_id" = 155	SELECT "ignored_users"."ignored_user_id" AS "ignored_user_id" FROM "ignored_users" AS "ignored_users" WHERE "ignored_users"."user_id" = 155
discourse	373	SELECT "users"."id" FROM "users" INNER JOIN "group_users" ON "users"."id" = "group_users"."user_id" WHERE "group_users"."group_id" = 2	SELECT "group_users"."user_id" AS "user_id" FROM "group_users" AS "group_users" WHERE "group_users"."group_id" = 2
discourse	449	SELECT "parent_theme_id" FROM "themes" INNER JOIN "child_themes" ON "child_themes"."child_theme_id" = "themes"."id" INNER JOIN "themes" "parent_themes_themes" ON "parent_themes_themes"."id" = "child_themes"."parent_theme_id" WHERE "themes"."id" = 1017	SELECT "child_themes"."parent_theme_id" AS "parent_theme_id" FROM "child_themes" AS "child_themes" WHERE "child_themes"."child_theme_id" = 1017
discourse	456	SELECT "theme_fields".* FROM "theme_fields" INNER JOIN "uploads" ON "uploads"."id" = "theme_fields"."upload_id" WHERE "theme_fields"."theme_id" = 1017 AND "theme_fields"."type_id" = 2	SELECT * FROM "theme_fields" AS "theme_fields" WHERE "theme_fields"."type_id" = 2 AND "theme_fields"."theme_id" = 1017
discourse	624	SELECT "groups"."id" FROM "groups" INNER JOIN "group_users" ON "groups"."id" = "group_users"."group_id" WHERE "group_users"."user_id" = 247	SELECT "group_users"."group_id" AS "group_id" FROM "group_users" AS "group_users" WHERE "group_users"."user_id" = 247
discourse	660	SELECT COUNT(*) FROM "categories" INNER JOIN "category_groups" ON "categories"."id" = "category_groups"."category_id" WHERE "category_groups"."group_id" = 2378	SELECT COUNT(*) FROM "category_groups" AS "category_groups" WHERE "category_groups"."group_id" = 2378
discourse	833	SELECT COUNT(*) FROM "users" INNER JOIN "group_users" ON "users"."id" = "group_users"."user_id" WHERE "group_users"."group_id" = 2397	SELECT COUNT(*) FROM "group_users" AS "group_users" WHERE "group_users"."group_id" = 2397
discourse	932	SELECT "group_users"."group_id" FROM "group_users" WHERE "group_users"."group_id" IN (SELECT "groups"."id" FROM "groups" WHERE (groups.id >0) ORDER BY name ASC) AND "group_users"."user_id" = 762	SELECT "group_users"."group_id" AS "group_id" FROM "group_users" AS "group_users" WHERE "group_users"."user_id" = 762 AND "group_users"."group_id" >0

discourse	933	SELECT "group_users"."group_id" FROM "group_users" WHERE "group_users"."group_id" IN (SELECT "groups"."id" FROM "groups" WHERE (groups.id >0) ORDER BY name ASC) AND "group_users"."user_id" = 762 AND "group_users"."owner" = TRUE	SELECT "group_users"."group_id" AS "group_id" FROM "group_users" AS "group_users" WHERE "group_users"."user_id" = 762 AND "group_users"."owner" = TRUE AND "group_users"."group_id" >0
discourse	940	SELECT "group_users"."group_id" FROM "group_users" WHERE "group_users"."group_id" IN (SELECT "groups"."id" FROM "groups" LEFT JOIN group_users gu ON gu.group_id = groups.id WHERE (groups.id >0) AND (gu.user_id = 779) AND (gu.owner) ORDER BY name ASC) AND "group_users"."user_id" = 779	SELECT "group_users"."group_id" AS "group_id" FROM "group_users" AS "group_users" INNER JOIN "group_users" AS "gu" ON "group_users"."group_id" = "gu"."group_id" WHERE "group_users"."group_id" >0 AND "gu"."owner" = TRUE AND "group_users"."user_id" = 779
discourse	941	SELECT "group_users"."group_id" FROM "group_users" WHERE "group_users"."group_id" IN (SELECT "groups"."id" FROM "groups" LEFT JOIN group_users gu ON gu.group_id = groups.id WHERE (groups.id >0) AND (gu.user_id = 779) AND (gu.owner) ORDER BY name ASC) AND "group_users"."user_id" = 779 AND "group_users"."owner" = TRUE	SELECT "group_users"."group_id" AS "group_id" FROM "group_users" AS "group_users" INNER JOIN "group_users" AS "gu" ON "group_users"."group_id" = "gu"."group_id" WHERE "gu"."group_id" >0 AND "group_users"."user_id" = 779 AND "group_users"."owner" = TRUE
discourse	944	SELECT "group_users"."group_id" FROM "group_users" WHERE "group_users"."group_id" IN (SELECT "groups"."id" FROM "groups" LEFT JOIN group_users gu ON gu.group_id = groups.id WHERE (groups.id >0) AND (gu.user_id = 779) ORDER BY name ASC) AND "group_users"."user_id" = 779	SELECT "group_users"."group_id" AS "group_id" FROM "group_users" AS "gu" INNER JOIN "group_users" AS "group_users" ON "gu"."group_id" = "group_users"."group_id" WHERE "gu"."group_id" >0 AND "group_users"."user_id" = 779
discourse	945	SELECT "group_users"."group_id" FROM "group_users" WHERE "group_users"."group_id" IN (SELECT "groups"."id" FROM "groups" LEFT JOIN group_users gu ON gu.group_id = groups.id WHERE (groups.id >0) AND (gu.user_id = 779) ORDER BY name ASC) AND "group_users"."user_id" = 779 AND "group_users"."owner" = TRUE	SELECT "group_users"."group_id" AS "group_id" FROM "group_users" AS "group_users" INNER JOIN "group_users" AS "gu" ON "group_users"."group_id" = "gu"."group_id" WHERE "gu"."group_id" >0 AND "group_users"."owner" = TRUE AND "group_users"."user_id" = 779
discourse	952	SELECT "group_users"."group_id" FROM "group_users" WHERE "group_users"."group_id" IN (SELECT "groups"."id" FROM "groups" WHERE (groups.id >0) AND (groups.id IN (SELECT id FROM groups WHERE visibility_level = 0 UNION ALL SELECT id FROM groups WHERE visibility_level = 1 AND 761 IS NOT NULL UNION ALL SELECT g.id FROM groups g JOIN group_users gu ON gu.group_id = g.id AND gu.user_id = 761 WHERE g.visibility_level = 2 UNION ALL SELECT g.id FROM groups g LEFT JOIN group_users gu ON gu.group_id = g.id AND gu.user_id = 761 AND gu.owner WHERE g.visibility_level = 3 AND (gu.id IS NOT NULL OR FALSE) UNION ALL SELECT g.id FROM groups g JOIN group_users gu ON gu.group_id = g.id AND gu.user_id = 761 AND gu.owner WHERE g.visibility_level = 4)) AND (automatic IS FALSE OR groups.id = 2) ORDER BY name DESC) AND "group_users"."user_id" = 761	SELECT "group_users"."group_id" AS "group_id" FROM (SELECT "g"."id" AS "id", "g"."name" AS "name", "g"."created_at" AS "created_at", "g"."updated_at" AS "updated_at", "g"."automatic" AS "automatic", "g"."user_count" AS "user_count", "g"."automatic_membership_email_domains" AS "automatic_membership_email_domains", "g"."automatic_membership_retroactive" AS "automatic_membership_retroactive", "g"."primary_group" AS "primary_group", "g"."title" AS "title", "g"."grant_trust_level" AS "grant_trust_level", "g"."incoming_email" AS "incoming_email", "g"."has_messages" AS "has_messages", "g"."flair_url" AS "flair_url", "g"."flair_bg_color" AS "flair_bg_color", "g"."flair_color" AS "flair_color", "g"."bio_raw" AS "bio_raw", "g"."bio_cooked" AS "bio_cooked", "g"."allow_membership_requests" AS "allow_membership_requests", "g"."full_name" AS "full_name", "g"."default_notification_level" AS "default_notification_level", "g"."visibility_level" AS "visibility_level", "g"."public_exit" AS "public_exit", "g"."public_admission" AS "public_admission", "g"."membership_request_template" AS "membership_request_template", "g"."messageable_level" AS "messageable_level", "g"."mentionable_level" AS "mentionable_level", "g"."publish_read_state" AS "publish_read_state", "g"."members_visibility_level" AS "members_visibility_level", "gu"."id" AS "id_0", "gu"."group_id" AS "group_id", "gu"."user_id" AS "user_id", "gu"."created_at" AS "created_at_0", "gu"."updated_at" AS "updated_at_0", "gu"."owner" AS "owner", "gu"."notification_level" AS "notification_level" FROM "groups" AS "g" INNER JOIN "group_users" AS "gu" ON "g"."id" = "gu"."group_id" WHERE "g"."visibility_level" = 4) AS "sub_0_0" INNER JOIN "groups" AS "groups" ON "sub_0_0"."id" = "groups"."id" INNER JOIN "group_users" AS "group_users" ON "sub_0_0"."id" = "group_users"."group_id" WHERE "sub_0_0"."owner" = TRUE AND "group_users"."group_id" >0 AND (NOT "groups"."automatic" = TRUE OR "groups"."id" = 2) AND "group_users"."user_id" = 761

discourse	959	SELECT "group_users"."group_id" FROM "group_users" WHERE "group_users"."group_id" IN (SELECT "groups"."id" FROM "groups" WHERE (groups.id >0) AND (groups.id IN (SELECT id FROM groups WHERE visibility_level = 0 UNION ALL SELECT id FROM groups WHERE visibility_level = 1 AND 761 IS NOT NULL UNION ALL SELECT g.id FROM groups g JOIN group_users gu ON gu.group_id = g.id AND gu.user_id = 761 WHERE g.visibility_level = 2 UNION ALL SELECT g.id FROM groups g LEFT JOIN group_users gu ON gu.group_id = g.id AND gu.user_id = 761 AND gu.owner WHERE g.visibility_level = 3 AND (gu.id IS NOT NULL OR FALSE) UNION ALL SELECT g.id FROM groups g JOIN group_users gu ON gu.group_id = g.id AND gu.user_id = 761 AND gu.owner WHERE g.visibility_level = 4)) AND (automatic IS FALSE OR groups.id = 2) ORDER BY name ASC) AND "group_users"."user_id" = 761	SELECT "group_users"."group_id" AS "group_id" FROM "groups" AS "groups" INNER JOIN (SELECT "g"."id" AS "id", "g"."name" AS "name", "g"."created_at" AS "created_at", "g"."updated_at" AS "updated_at", "g"."automatic" AS "automatic", "g"."user_count" AS "user_count", "g"."automatic_membership_email_domains" AS "automatic_membership_email_domains", "g"."automatic_membership_retroactive" AS "automatic_membership_retroactive", "g"."primary_group" AS "primary_group", "g"."title" AS "title", "g"."grant_trust_level" AS "grant_trust_level", "g"."incoming_email" AS "incoming_email", "g"."has_messages" AS "has_messages", "g"."flair_url" AS "flair_url", "g"."flair_bg_color" AS "flair_bg_color", "g"."flair_color" AS "flair_color", "g"."bio_raw" AS "bio_raw", "g"."bio_cooked" AS "bio_cooked", "g"."allow_membership_requests" AS "allow_membership_requests", "g"."full_name" AS "full_name", "g"."default_notification_level" AS "default_notification_level", "g"."visibility_level" AS "visibility_level", "g"."public_exit" AS "public_exit", "g"."public_admission" AS "public_admission", "g"."membership_request_template" AS "membership_request_template", "g"."messageable_level" AS "messageable_level", "g"."mentionable_level" AS "mentionable_level", "g"."publish_read_state" AS "publish_read_state", "g"."members_visibility_level" AS "members_visibility_level", "gu"."id" AS "id_0", "gu"."group_id" AS "group_id", "gu"."user_id" AS "user_id", "gu"."created_at" AS "created_at_0", "gu"."updated_at" AS "updated_at_0", "gu"."owner" AS "owner", "gu"."notification_level" AS "notification_level" FROM "groups" AS "g" INNER JOIN "group_users" AS "gu" ON "g"."id" = "gu"."group_id" WHERE "g"."visibility_level" = 4) AS "sub_0_0" ON "groups"."id" = "sub_0_0"."id" INNER JOIN "group_users" AS "group_users" ON "groups"."id" = "group_users"."group_id" WHERE "group_users"."group_id" >0 AND "sub_0_0"."owner" = TRUE AND (NOT "groups"."automatic" = TRUE OR "groups"."id" = 2) AND "group_users"."user_id" = 761
discourse	992	SELECT "group_users"."group_id" FROM "group_users" WHERE "group_users"."group_id" IN (SELECT "groups"."id" FROM "groups" LEFT JOIN group_users gu ON gu.group_id = groups.id WHERE (groups.id >0) AND (groups.id >0) AND (gu.user_id = 780) ORDER BY name ASC) AND "group_users"."user_id" = 762	SELECT "group_users"."group_id" AS "group_id" FROM "group_users" AS "group_users" INNER JOIN "group_users" AS "gu" ON "group_users"."group_id" = "gu"."group_id" WHERE "gu"."user_id" = 780 AND "group_users"."user_id" = 762 AND "group_users"."group_id" >0
discourse	993	SELECT "group_users"."group_id" FROM "group_users" WHERE "group_users"."group_id" IN (SELECT "groups"."id" FROM "groups" LEFT JOIN group_users gu ON gu.group_id = groups.id WHERE (groups.id >0) AND (groups.id >0) AND (gu.user_id = 780) ORDER BY name ASC) AND "group_users"."user_id" = 762 AND "group_users"."owner" = TRUE	SELECT "group_users"."group_id" AS "group_id" FROM "group_users" AS "gu" INNER JOIN "group_users" AS "group_users" ON "gu"."group_id" = "group_users"."group_id" WHERE "gu"."group_id" >0 AND "gu"."user_id" = 780 AND "group_users"."user_id" = 762 AND "group_users"."owner" = TRUE
discourse	1006	SELECT COUNT(*) FROM "users" INNER JOIN "group_users" ON "users"."id" = "group_users"."user_id" WHERE "group_users"."group_id" = 2564 AND (users.id >0)	SELECT COUNT(*) FROM "group_users" AS "group_users" WHERE "group_users"."user_id" >0 AND "group_users"."group_id" = 2564
discourse	1008	SELECT users.*, user_options.timezone, group_users.created_at as added_at FROM "users" INNER JOIN "group_users" ON "users"."id" = "group_users"."user_id" INNER JOIN "user_options" ON "user_options"."user_id" = "users"."id" WHERE "group_users"."group_id" = 2564 AND (users.id >0) AND (group_users.owner) ORDER BY last_seen_at DESC NULLS LAST, "users"."username_lower" DESC	SELECT "users"."id" AS "id", "users"."username" AS "username", "users"."created_at" AS "created_at", "users"."updated_at" AS "updated_at", "users"."name" AS "name", "users"."seen_notification_id" AS "seen_notification_id", "users"."last_posted_at" AS "last_posted_at", "users"."password_hash" AS "password_hash", "users"."salt" AS "salt", "users"."active" AS "active", "users"."username_lower" AS "username_lower", "users"."last_seen_at" AS "last_seen_at", "users"."admin" AS "admin", "users"."last_emailed_at" AS "last_emailed_at", "users"."trust_level" AS "trust_level", "users"."approved" AS "approved", "users"."approved_by_id" AS "approved_by_id", "users"."approved_at" AS "approved_at", "users"."previous_visit_at" AS "previous_visit_at", "users"."suspended_at" AS "suspended_at", "users"."suspended_till" AS "suspended_till", "users"."date_of_birth" AS "date_of_birth", "users"."views" AS "views", "users"."flag_level" AS "flag_level", "users"."ip_address" AS "ip_address", "users"."moderator" AS "moderator", "users"."title" AS "title", "users"."uploaded_avatar_id" AS "uploaded_avatar_id", "users"."locale" AS "locale", "users"."primary_group_id" AS "primary_group_id", "users"."registration_ip_address" AS "registration_ip_address", "users"."staged" AS "staged", "users"."first_seen_at" AS "first_seen_at", "users"."silenced_till" AS "silenced_till", "users"."group_locked_trust_level" AS "group_locked_trust_level", "users"."manual_locked_trust_level" AS "manual_locked_trust_level", "users"."secure_identifier" AS "secure_identifier", "user_options"."timezone" AS "timezone", "group_users"."created_at" AS "added_at" FROM "user_options" AS "user_options" INNER JOIN "users" AS "users" ON "user_options"."user_id" = "users"."id" INNER JOIN "group_users" AS "group_users" ON "users"."id" = "group_users"."user_id" WHERE "user_options"."user_id" >0 AND "group_users"."owner" = TRUE AND "group_users"."group_id" = 2564 ORDER BY "last_seen_at" DESC, "users"."username_lower" DESC

discourse	1010	SELECT users.*, user_options.timezone, group_users.created_at as added_at FROM "users" INNER JOIN "group_users" ON "users"."id" = "group_users"."user_id" INNER JOIN "user_options" ON "user_options"."user_id" = "users"."id" WHERE "group_users"."group_id" = 2564 AND (users.id >0) AND (group_users.owner) ORDER BY last_seen_at ASC NULLS LAST, "users"."username_lower" ASC	SELECT "users"."id" AS "id", "users"."username" AS "username", "users"."created_at" AS "created_at", "users"."updated_at" AS "updated_at", "users"."name" AS "name", "users"."seen_notification_id" AS "seen_notification_id", "users"."last_posted_at" AS "last_posted_at", "users"."password_hash" AS "password_hash", "users"."salt" AS "salt", "users"."active" AS "active", "users"."username_lower" AS "username_lower", "users"."last_seen_at" AS "last_seen_at", "users"."admin" AS "admin", "users"."last_emailed_at" AS "last_emailed_at", "users"."trust_level" AS "trust_level", "users"."approved" AS "approved", "users"."approved_by_id" AS "approved_by_id", "users"."approved_at" AS "approved_at", "users"."previous_visit_at" AS "previous_visit_at", "users"."suspended_at" AS "suspended_at", "users"."suspended_till" AS "suspended_till", "users"."date_of_birth" AS "date_of_birth", "users"."views" AS "views", "users"."flag_level" AS "flag_level", "users"."ip_address" AS "ip_address", "users"."moderator" AS "moderator", "users"."title" AS "title", "users"."uploaded_avatar_id" AS "uploaded_avatar_id", "users"."locale" AS "locale", "users"."primary_group_id" AS "primary_group_id", "users"."registration_ip_address" AS "registration_ip_address", "users"."staged" AS "staged", "users"."first_seen_at" AS "first_seen_at", "users"."silenced_till" AS "silenced_till", "users"."group_locked_trust_level" AS "group_locked_trust_level", "users"."manual_locked_trust_level" AS "manual_locked_trust_level", "users"."secure_identifier" AS "secure_identifier", "user_options"."timezone" AS "timezone", "group_users"."created_at" AS "added_at" FROM "user_options" AS "user_options" INNER JOIN "group_users" AS "group_users" ON "user_options"."user_id" = "group_users"."user_id" INNER JOIN "users" AS "users" ON "group_users"."user_id" = "users"."id" WHERE "group_users"."owner" = TRUE AND "group_users"."group_id" = 2564 AND "users"."id" >0 ORDER BY "users"."last_seen_at" ASC, "users"."username_lower" ASC
discourse	1012	SELECT users.*, user_options.timezone, group_users.created_at as added_at FROM "users" INNER JOIN "group_users" ON "users"."id" = "group_users"."user_id" INNER JOIN "user_options" ON "user_options"."user_id" = "users"."id" WHERE "group_users"."group_id" = 2564 AND (users.id >0) AND (group_users.owner) ORDER BY last_posted_at DESC NULLS LAST, "users"."username_lower" DESC	SELECT "users"."id" AS "id", "users"."username" AS "username", "users"."created_at" AS "created_at", "users"."updated_at" AS "updated_at", "users"."name" AS "name", "users"."seen_notification_id" AS "seen_notification_id", "users"."last_posted_at" AS "last_posted_at", "users"."password_hash" AS "password_hash", "users"."salt" AS "salt", "users"."active" AS "active", "users"."username_lower" AS "username_lower", "users"."last_seen_at" AS "last_seen_at", "users"."admin" AS "admin", "users"."last_emailed_at" AS "last_emailed_at", "users"."trust_level" AS "trust_level", "users"."approved" AS "approved", "users"."approved_by_id" AS "approved_by_id", "users"."approved_at" AS "approved_at", "users"."previous_visit_at" AS "previous_visit_at", "users"."suspended_at" AS "suspended_at", "users"."suspended_till" AS "suspended_till", "users"."date_of_birth" AS "date_of_birth", "users"."views" AS "views", "users"."flag_level" AS "flag_level", "users"."ip_address" AS "ip_address", "users"."moderator" AS "moderator", "users"."title" AS "title", "users"."uploaded_avatar_id" AS "uploaded_avatar_id", "users"."locale" AS "locale", "users"."primary_group_id" AS "primary_group_id", "users"."registration_ip_address" AS "registration_ip_address", "users"."staged" AS "staged", "users"."first_seen_at" AS "first_seen_at", "users"."silenced_till" AS "silenced_till", "users"."group_locked_trust_level" AS "group_locked_trust_level", "users"."manual_locked_trust_level" AS "manual_locked_trust_level", "users"."secure_identifier" AS "secure_identifier", "user_options"."timezone" AS "timezone", "group_users"."created_at" AS "added_at" FROM "user_options" AS "user_options" INNER JOIN "group_users" AS "group_users" ON "user_options"."user_id" = "group_users"."user_id" INNER JOIN "users" AS "users" ON "group_users"."user_id" = "users"."id" WHERE "users"."id" >0 AND "group_users"."owner" = TRUE AND "group_users"."group_id" = 2564 ORDER BY "last_posted_at" DESC, "users"."username_lower" DESC

discourse	1015	SELECT users.*, user_options.timezone, group_users.created_at as added_at FROM "users" INNER JOIN "group_users" ON "users"."id" = "group_users"."user_id" INNER JOIN "user_options" ON "user_options"."user_id" = "users"."id" WHERE "group_users"."group_id" = 2565 AND (users.id >0) AND (group_users.owner) ORDER BY "users"."username_lower" ASC	SELECT "users"."id" AS "id", "users"."username" AS "username", "users"."created_at" AS "created_at", "users"."updated_at" AS "updated_at", "users"."name" AS "name", "users"."seen_notification_id" AS "seen_notification_id", "users"."last_posted_at" AS "last_posted_at", "users"."password_hash" AS "password_hash", "users"."salt" AS "salt", "users"."active" AS "active", "users"."username_lower" AS "username_lower", "users"."last_seen_at" AS "last_seen_at", "users"."admin" AS "admin", "users"."last_emailed_at" AS "last_emailed_at", "users"."trust_level" AS "trust_level", "users"."approved" AS "approved", "users"."approved_by_id" AS "approved_by_id", "users"."approved_at" AS "approved_at", "users"."previous_visit_at" AS "previous_visit_at", "users"."suspended_at" AS "suspended_at", "users"."suspended_till" AS "suspended_till", "users"."date_of_birth" AS "date_of_birth", "users"."views" AS "views", "users"."flag_level" AS "flag_level", "users"."ip_address" AS "ip_address", "users"."moderator" AS "moderator", "users"."title" AS "title", "users"."uploaded_avatar_id" AS "uploaded_avatar_id", "users"."locale" AS "locale", "users"."primary_group_id" AS "primary_group_id", "users"."registration_ip_address" AS "registration_ip_address", "users"."staged" AS "staged", "users"."first_seen_at" AS "first_seen_at", "users"."silenced_till" AS "silenced_till", "users"."group_locked_trust_level" AS "group_locked_trust_level", "users"."manual_locked_trust_level" AS "manual_locked_trust_level", "users"."secure_identifier" AS "secure_identifier", "user_options"."timezone" AS "timezone", "group_users"."created_at" AS "added_at" FROM "user_options" AS "user_options" INNER JOIN "group_users" AS "group_users" ON "user_options"."user_id" = "group_users"."user_id" INNER JOIN "users" AS "users" ON "user_options"."user_id" = "users"."id" WHERE "users"."id" >0 AND "group_users"."owner" = TRUE AND "group_users"."group_id" = 2565 ORDER BY "users"."username_lower" ASC
discourse	1044	SELECT users.*, user_options.timezone, group_users.created_at as added_at FROM "users" INNER JOIN "group_users" ON "users"."id" = "group_users"."user_id" INNER JOIN "user_options" ON "user_options"."user_id" = "users"."id" WHERE "group_users"."group_id" = 2574 AND (users.id >0) AND (group_users.owner) ORDER BY group_users.created_at DESC, "users"."username_lower" DESC	SELECT "users"."id" AS "id", "users"."username" AS "username", "users"."created_at" AS "created_at", "users"."updated_at" AS "updated_at", "users"."name" AS "name", "users"."seen_notification_id" AS "seen_notification_id", "users"."last_posted_at" AS "last_posted_at", "users"."password_hash" AS "password_hash", "users"."salt" AS "salt", "users"."active" AS "active", "users"."username_lower" AS "username_lower", "users"."last_seen_at" AS "last_seen_at", "users"."admin" AS "admin", "users"."last_emailed_at" AS "last_emailed_at", "users"."trust_level" AS "trust_level", "users"."approved" AS "approved", "users"."approved_by_id" AS "approved_by_id", "users"."approved_at" AS "approved_at", "users"."previous_visit_at" AS "previous_visit_at", "users"."suspended_at" AS "suspended_at", "users"."suspended_till" AS "suspended_till", "users"."date_of_birth" AS "date_of_birth", "users"."views" AS "views", "users"."flag_level" AS "flag_level", "users"."ip_address" AS "ip_address", "users"."moderator" AS "moderator", "users"."title" AS "title", "users"."uploaded_avatar_id" AS "uploaded_avatar_id", "users"."locale" AS "locale", "users"."primary_group_id" AS "primary_group_id", "users"."registration_ip_address" AS "registration_ip_address", "users"."staged" AS "staged", "users"."first_seen_at" AS "first_seen_at", "users"."silenced_till" AS "silenced_till", "users"."group_locked_trust_level" AS "group_locked_trust_level", "users"."manual_locked_trust_level" AS "manual_locked_trust_level", "users"."secure_identifier" AS "secure_identifier", "user_options"."timezone" AS "timezone", "group_users"."created_at" AS "added_at" FROM "user_options" AS "user_options" INNER JOIN "group_users" AS "group_users" ON "user_options"."user_id" = "group_users"."user_id" INNER JOIN "users" AS "users" ON "group_users"."user_id" = "users"."id" WHERE "group_users"."owner" = TRUE AND "group_users"."group_id" = 2574 AND "users"."id" >0 ORDER BY "group_users"."created_at" DESC, "users"."username_lower" DESC

discourse	1048	SELECT COUNT(*) FROM (SELECT 1 AS one FROM "group_histories" WHERE "group_histories"."group_id" = 2576 ORDER BY group_histories.created_at DESC LIMIT 25 OFFSET 0) subquery_for_count	SELECT COUNT(*) FROM (SELECT 1 AS "one" FROM "group_histories" AS "group_histories" WHERE "group_histories"."group_id" = 2576 LIMIT 25 OFFSET 0) AS "subquery_for_count"
discourse	1052	SELECT COUNT(*) FROM (SELECT 1 AS one FROM "group_histories" WHERE "group_histories"."group_id" = 2578 AND "group_histories"."action" = 2 ORDER BY group_histories.created_at DESC LIMIT 25 OFFSET 0) subquery_for_count	SELECT COUNT(*) FROM (SELECT 1 AS "one" FROM "group_histories" AS "group_histories" WHERE "group_histories"."group_id" = 2578 AND "group_histories"."action" = 2 LIMIT 25 OFFSET 0) AS "subquery_for_count"
discourse	1178	SELECT distinct "posts"."id", "posts"."post_number" FROM (SELECT "posts".* FROM "posts" INNER JOIN "users" ON "users"."id" = "posts"."user_id" WHERE ("posts"."deleted_at" IS NOT NULL) AND "posts"."topic_id" = 15986 AND (posts.user_id = 915 OR post_type IN (1,2,3)) AND (posts.post_type = 1) AND (post_number >1) ORDER BY percent_rank asc, sort_order asc LIMIT 2) posts WHERE ("posts"."deleted_at" IS NOT NULL) ORDER BY "posts"."post_number" ASC	SELECT "posts"."id" AS "id", "posts"."post_number" AS "post_number" FROM "posts" AS "posts" WHERE ("posts"."user_id" = 915 OR "posts"."post_type" IN ($1)) AND "posts"."topic_id" = 15986 AND "posts"."post_type" = 1 AND NOT "posts"."deleted_at" IS NULL AND "posts"."post_number" >1 AND NOT "posts"."deleted_at" IS NULL ORDER BY "posts"."post_number" ASC LIMIT 2
discourse	1181	SELECT "posts"."id" FROM (SELECT "posts".* FROM "posts" INNER JOIN "users" ON "users"."id" = "posts"."user_id" WHERE ("posts"."deleted_at" IS NOT NULL) AND "posts"."topic_id" = 15986 AND "posts"."post_type" IN (1, 2, 3) AND (posts.post_type = 1) AND (post_number >1) AND (COALESCE(users.trust_level,0) >= 2) ORDER BY percent_rank asc, sort_order asc LIMIT 99) posts WHERE ("posts"."deleted_at" IS NOT NULL) ORDER BY "posts"."post_number" ASC	SELECT "posts_0"."id" AS "id" FROM (SELECT "posts"."id" AS "id", "posts"."user_id" AS "user_id", "posts"."topic_id" AS "topic_id", "posts"."post_number" AS "post_number", "posts"."raw" AS "raw", "posts"."cooked" AS "cooked", "posts"."created_at" AS "created_at", "posts"."updated_at" AS "updated_at", "posts"."reply_to_post_number" AS "reply_to_post_number", "posts"."reply_count" AS "reply_count", "posts"."quote_count" AS "quote_count", "posts"."deleted_at" AS "deleted_at", "posts"."off_topic_count" AS "off_topic_count", "posts"."like_count" AS "like_count", "posts"."incoming_link_count" AS "incoming_link_count", "posts"."bookmark_count" AS "bookmark_count", "posts"."avg_time" AS "avg_time", "posts"."score" AS "score", "posts"."reads" AS "reads", "posts"."post_type" AS "post_type", "posts"."sort_order" AS "sort_order", "posts"."last_editor_id" AS "last_editor_id", "posts"."hidden" AS "hidden", "posts"."hidden_reason_id" AS "hidden_reason_id", "posts"."notify_moderators_count" AS "notify_moderators_count", "posts"."spam_count" AS "spam_count", "posts"."illegal_count" AS "illegal_count", "posts"."inappropriate_count" AS "inappropriate_count", "posts"."last_version_at" AS "last_version_at", "posts"."user_deleted" AS "user_deleted", "posts"."reply_to_user_id" AS "reply_to_user_id", "posts"."percent_rank" AS "percent_rank", "posts"."notify_user_count" AS "notify_user_count", "posts"."like_score" AS "like_score", "posts"."deleted_by_id" AS "deleted_by_id", "posts"."edit_reason" AS "edit_reason", "posts"."word_count" AS "word_count", "posts"."version" AS "version", "posts"."cook_method" AS "cook_method", "posts"."wiki" AS "wiki", "posts"."baked_at" AS "baked_at", "posts"."baked_version" AS "baked_version", "posts"."hidden_at" AS "hidden_at", "posts"."self_edits" AS "self_edits", "posts"."reply_quoted" AS "reply_quoted", "posts"."via_email" AS "via_email", "posts"."raw_email" AS "raw_email", "posts"."public_version" AS "public_version", "posts"."action_code" AS "action_code", "posts"."image_url" AS "image_url", "posts"."locked_by_id" AS "locked_by_id" FROM "posts" AS "posts" INNER JOIN "users" AS "users" ON "posts"."user_id" = "users"."id" WHERE COALESCE("users"."trust_level", 0) >= 2) AS "posts_0" WHERE "posts_0"."topic_id" = 15986 AND "posts_0"."post_type" = 1 AND "posts_0"."post_type" IN ($1) AND NOT "posts_0"."deleted_at" IS NULL AND "posts_0"."post_number" >1 AND NOT "posts_0"."deleted_at" IS NULL ORDER BY "posts_0"."post_number" ASC LIMIT 99

discourse	1214	SELECT "posts"."id" FROM (SELECT "posts".* FROM "posts" INNER JOIN "users" ON "users"."id" = "posts"."user_id" WHERE ("posts"."deleted_at" IS NOT NULL) AND "posts"."topic_id" = 15986 AND (posts.user_id = 915 OR post_type IN (1,2,3)) AND (posts.post_type = 1) AND (post_number >1) ORDER BY percent_rank asc, sort_order asc LIMIT 5) posts WHERE ("posts"."deleted_at" IS NOT NULL) ORDER BY "posts"."post_number" ASC	SELECT "posts"."id" AS "id" FROM "posts" AS "posts" WHERE ("posts"."user_id" = 915 OR "posts"."post_type" IN ($1)) AND "posts"."topic_id" = 15986 AND "posts"."post_type" = 1 AND NOT "posts"."deleted_at" IS NULL AND "posts"."post_number" >1 AND NOT "posts"."deleted_at" IS NULL ORDER BY "posts"."post_number" ASC LIMIT 5
discourse	1291	SELECT DISTINCT "parent_theme_id" FROM "themes" INNER JOIN "child_themes" ON "child_themes"."child_theme_id" = "themes"."id" INNER JOIN "themes" "parent_themes_themes" ON "parent_themes_themes"."id" = "child_themes"."parent_theme_id" WHERE "themes"."id" IN (1029, 1028)	SELECT DISTINCT "child_themes"."parent_theme_id" AS "parent_theme_id" FROM "child_themes" AS "child_themes" WHERE "child_themes"."child_theme_id" IN ($1)
discourse	1666	SELECT "users"."id" FROM "users" LEFT JOIN user_options ON user_options.user_id = users.id LEFT JOIN muted_users ON muted_users.user_id = users.id AND muted_users.muted_user_id = 1342 LEFT JOIN ignored_users ON ignored_users.user_id = users.id AND ignored_users.ignored_user_id = 1342 WHERE (user_options.user_id IS NOT NULL) AND ((user_options.user_id IN (1343) AND NOT user_options.allow_private_messages) OR muted_users.user_id IN (1343) OR ignored_users.user_id IN (1343))	SELECT "users"."id" AS "id" FROM "users" AS "users" INNER JOIN "muted_users" AS "muted_users" ON "muted_users"."user_id" = "users"."id" AND "muted_users"."muted_user_id" = 1342 INNER JOIN "ignored_users" AS "ignored_users" ON "ignored_users"."user_id" = "users"."id" AND "ignored_users"."ignored_user_id" = 1342 INNER JOIN "user_options" AS "user_options" ON "users"."id" = "user_options"."user_id" WHERE NOT "user_options"."user_id" IS NULL AND ("user_options"."user_id" IN ($1) AND NOT "user_options"."allow_private_messages" = TRUE OR "muted_users"."user_id" IN ($1) OR "ignored_users"."user_id" IN ($1))
discourse	1957	SELECT "user_emails"."email" FROM "users" INNER JOIN "topic_allowed_users" ON "users"."id" = "topic_allowed_users"."user_id" INNER JOIN "user_emails" ON "user_emails"."user_id" = "users"."id" WHERE "topic_allowed_users"."topic_id" = 16471	SELECT "user_emails"."email" AS "email" FROM "topic_allowed_users" AS "topic_allowed_users" INNER JOIN "user_emails" AS "user_emails" ON "topic_allowed_users"."user_id" = "user_emails"."user_id" WHERE "topic_allowed_users"."topic_id" = 16471
discourse	2239	SELECT "groups"."name" FROM "groups" INNER JOIN "category_groups" ON "groups"."id" = "category_groups"."group_id" WHERE "category_groups"."category_id" = 3276 AND (groups.id >0) AND (groups.id IN (SELECT id FROM groups WHERE visibility_level = 0 UNION ALL SELECT id FROM groups WHERE visibility_level = 1 AND 2112 IS NOT NULL UNION ALL SELECT g.id FROM groups g JOIN group_users gu ON gu.group_id = g.id AND gu.user_id = 2112 WHERE g.visibility_level = 2 UNION ALL SELECT g.id FROM groups g LEFT JOIN group_users gu ON gu.group_id = g.id AND gu.user_id = 2112 AND gu.owner WHERE g.visibility_level = 3 AND (gu.id IS NOT NULL OR FALSE) UNION ALL SELECT g.id FROM groups g JOIN group_users gu ON gu.group_id = g.id AND gu.user_id = 2112 AND gu.owner WHERE g.visibility_level = 4)) AND "groups"."automatic" = FALSE ORDER BY name ASC	SELECT "sub_1"."group_id" AS "name" FROM (SELECT "gu"."id" AS "id", "gu"."group_id" AS "group_id", "gu"."user_id" AS "user_id", "gu"."created_at" AS "created_at", "gu"."updated_at" AS "updated_at", "gu"."owner" AS "owner", "gu"."notification_level" AS "notification_level", "g"."id" AS "id_0", "g"."name" AS "name", "g"."created_at" AS "created_at_0", "g"."updated_at" AS "updated_at_0", "g"."automatic" AS "automatic", "g"."user_count" AS "user_count", "g"."automatic_membership_email_domains" AS "automatic_membership_email_domains", "g"."automatic_membership_retroactive" AS "automatic_membership_retroactive", "g"."primary_group" AS "primary_group", "g"."title" AS "title", "g"."grant_trust_level" AS "grant_trust_level", "g"."incoming_email" AS "incoming_email", "g"."has_messages" AS "has_messages", "g"."flair_url" AS "flair_url", "g"."flair_bg_color" AS "flair_bg_color", "g"."flair_color" AS "flair_color", "g"."bio_raw" AS "bio_raw", "g"."bio_cooked" AS "bio_cooked", "g"."allow_membership_requests" AS "allow_membership_requests", "g"."full_name" AS "full_name", "g"."default_notification_level" AS "default_notification_level", "g"."visibility_level" AS "visibility_level", "g"."public_exit" AS "public_exit", "g"."public_admission" AS "public_admission", "g"."membership_request_template" AS "membership_request_template", "g"."messageable_level" AS "messageable_level", "g"."mentionable_level" AS "mentionable_level", "g"."publish_read_state" AS "publish_read_state", "g"."members_visibility_level" AS "members_visibility_level" FROM "group_users" AS "gu" INNER JOIN "groups" AS "g" ON "gu"."group_id" = "g"."id" WHERE "g"."visibility_level" = 4) AS "sub_1" INNER JOIN (SELECT "groups"."id" AS "id", "groups"."name" AS "name", "groups"."created_at" AS "created_at", "groups"."updated_at" AS "updated_at", "groups"."automatic" AS "automatic", "groups"."user_count" AS "user_count", "groups"."automatic_membership_email_domains" AS "automatic_membership_email_domains", "groups"."automatic_membership_retroactive" AS "automatic_membership_retroactive", "groups"."primary_group" AS "primary_group", "groups"."title" AS "title", "groups"."grant_trust_level" AS "grant_trust_level", "groups"."incoming_email" AS "incoming_email", "groups"."has_messages" AS "has_messages", "groups"."flair_url" AS "flair_url", "groups"."flair_bg_color" AS "flair_bg_color", "groups"."flair_color" AS "flair_color", "groups"."bio_raw" AS "bio_raw", "groups"."bio_cooked" AS "bio_cooked", "groups"."allow_membership_requests" AS "allow_membership_requests", "groups"."full_name" AS "full_name", "groups"."default_notification_level" AS "default_notification_level", "groups"."visibility_level" AS "visibility_level", "groups"."public_exit" AS "public_exit", "groups"."public_admission" AS "public_admission", "groups"."membership_request_template" AS "membership_request_template", "groups"."messageable_level" AS "messageable_level", "groups"."mentionable_level" AS "mentionable_level", "groups"."publish_read_state" AS "publish_read_state", "groups"."members_visibility_level" AS "members_visibility_level", "category_groups"."id" AS "id_0", "category_groups"."category_id" AS "category_id", "category_groups"."group_id" AS "group_id", "category_groups"."created_at" AS "created_at_0", "category_groups"."updated_at" AS "updated_at_0", "category_groups"."permission_type" AS "permission_type" FROM "groups" AS "groups" INNER JOIN "category_groups" AS "category_groups" ON "groups"."id" = "category_groups"."group_id" WHERE "category_groups"."category_id" = 3276) AS "sub_0" ON "sub_1"."id_0" = "sub_0"."id" WHERE "sub_0"."id" >0 AND "sub_1"."user_id" = 2112 AND "sub_1"."owner" = TRUE AND "sub_0"."automatic" = FALSE ORDER BY "sub_1"."group_id" ASC

discourse	2407	SELECT DISTINCT "parent_theme_id" FROM "themes" INNER JOIN "child_themes" ON "child_themes"."child_theme_id" = "themes"."id" INNER JOIN "themes" "parent_themes_themes" ON "parent_themes_themes"."id" = "child_themes"."parent_theme_id" WHERE 1=0	SELECT DISTINCT "child_themes"."parent_theme_id" AS "parent_theme_id" FROM "child_themes" AS "child_themes"
discourse	2832	SELECT distinct "categories"."name" FROM "categories" INNER JOIN "category_tag_groups" ON "category_tag_groups"."category_id" = "categories"."id" INNER JOIN "tag_groups" ON "tag_groups"."id" = "category_tag_groups"."tag_group_id" INNER JOIN "tag_group_memberships" ON "tag_group_memberships"."tag_group_id" = "tag_groups"."id" INNER JOIN "tags" ON "tags"."id" = "tag_group_memberships"."tag_id" WHERE "categories"."id" IN (1, 3398, 3399, 3402) AND "tags"."id" = 1771	SELECT DISTINCT "categories"."name" AS "name" FROM "categories" AS "categories" INNER JOIN "category_tag_groups" AS "category_tag_groups" ON "categories"."id" = "category_tag_groups"."category_id" INNER JOIN "tag_group_memberships" AS "tag_group_memberships" ON "category_tag_groups"."tag_group_id" = "tag_group_memberships"."tag_group_id" WHERE "tag_group_memberships"."tag_id" = 1771 AND "categories"."id" IN ($1)
discourse	3131	SELECT "notifications".* FROM "notifications" LEFT JOIN topics ON notifications.topic_id = topics.id WHERE "notifications"."user_id" = 5191 AND (topics.id IS NULL OR topics.deleted_at IS NOT NULL) ORDER BY notifications.created_at desc LIMIT 3	SELECT * FROM (SELECT "notifications"."id" AS "id", "notifications"."notification_type" AS "notification_type", "notifications"."user_id" AS "user_id", "notifications"."data" AS "data", "notifications"."read" AS "read", "notifications"."created_at" AS "created_at", "notifications"."updated_at" AS "updated_at", "notifications"."topic_id" AS "topic_id", "notifications"."post_number" AS "post_number", "notifications"."post_action_id" AS "post_action_id" FROM "notifications" AS "notifications" INNER JOIN "topics" AS "topics" ON "notifications"."topic_id" = "topics"."id" WHERE "topics"."id" IS NULL OR NOT "topics"."deleted_at" IS NULL) AS "sub_0" WHERE "sub_0"."user_id" = 5191 ORDER BY "sub_0"."created_at" DESC LIMIT 3
discourse	3685	SELECT COUNT(post_actions.id) FROM "post_actions" INNER JOIN "posts" ON "posts"."id" = "post_actions"."post_id" AND ("posts"."deleted_at" IS NOT NULL) LEFT JOIN bookmarks ON bookmarks.post_id = post_actions.post_id AND bookmarks.user_id = post_actions.user_id INNER JOIN topics ON topics.id = posts.topic_id WHERE ("post_actions"."deleted_at" IS NOT NULL) AND "post_actions"."post_action_type_id" = 1 AND "post_actions"."deleted_at" IS NOT NULL AND (bookmarks.id IS NULL)	SELECT COUNT("post_actions"."id") FROM "post_actions" AS "post_actions" INNER JOIN "posts" AS "posts" ON "post_actions"."post_id" = "posts"."id" LEFT JOIN "bookmarks" AS "bookmarks" ON "post_actions"."post_id" = "bookmarks"."post_id" AND "post_actions"."user_id" = "bookmarks"."user_id" WHERE NOT "posts"."deleted_at" IS NULL AND "post_actions"."post_action_type_id" = 1 AND "bookmarks"."id" IS NULL AND NOT "post_actions"."deleted_at" IS NULL
discourse	3689	SELECT post_actions.id, post_actions.post_id, posts.topic_id, post_actions.user_id FROM "post_actions" INNER JOIN "posts" ON "posts"."id" = "post_actions"."post_id" AND ("posts"."deleted_at" IS NOT NULL) LEFT JOIN bookmarks ON bookmarks.post_id = post_actions.post_id AND bookmarks.user_id = post_actions.user_id INNER JOIN topics ON topics.id = posts.topic_id WHERE ("post_actions"."deleted_at" IS NOT NULL) AND "post_actions"."post_action_type_id" = 1 AND "post_actions"."deleted_at" IS NOT NULL AND (bookmarks.id IS NULL) ORDER BY "post_actions"."id" ASC LIMIT 1	SELECT "post_actions"."id" AS "id", "post_actions"."post_id" AS "post_id", "posts"."topic_id" AS "topic_id", "post_actions"."user_id" AS "user_id" FROM "post_actions" AS "post_actions" INNER JOIN "posts" AS "posts" ON "post_actions"."post_id" = "posts"."id" LEFT JOIN "bookmarks" AS "bookmarks" ON "post_actions"."post_id" = "bookmarks"."post_id" AND "post_actions"."user_id" = "bookmarks"."user_id" WHERE NOT "posts"."deleted_at" IS NULL AND "post_actions"."post_action_type_id" = 1 AND "bookmarks"."id" IS NULL AND NOT "post_actions"."deleted_at" IS NULL ORDER BY "post_actions"."id" ASC LIMIT 1
discourse	3690	SELECT COUNT(DISTINCT "directory_items"."id") FROM "directory_items" LEFT OUTER JOIN "users" ON "users"."id" = "directory_items"."user_id" LEFT OUTER JOIN "group_users" ON "group_users"."user_id" = "users"."id" LEFT OUTER JOIN "groups" ON "groups"."id" = "group_users"."group_id" LEFT OUTER JOIN "user_stats" ON "user_stats"."user_id" = "directory_items"."user_id" WHERE "directory_items"."period_type" = 1 AND "groups"."id" = 2898	SELECT COUNT(DISTINCT "directory_items"."id") FROM "group_users" AS "group_users" INNER JOIN "directory_items" AS "directory_items" ON "group_users"."user_id" = "directory_items"."user_id" WHERE "group_users"."group_id" = 2898 AND "directory_items"."period_type" = 1
discourse	3691	SELECT DISTINCT directory_items.likes_received AS alias_0, "directory_items"."id" FROM "directory_items" LEFT OUTER JOIN "users" ON "users"."id" = "directory_items"."user_id" LEFT OUTER JOIN "group_users" ON "group_users"."user_id" = "users"."id" LEFT OUTER JOIN "groups" ON "groups"."id" = "group_users"."group_id" LEFT OUTER JOIN "user_stats" ON "user_stats"."user_id" = "directory_items"."user_id" WHERE "directory_items"."period_type" = 1 AND "groups"."id" = 2898 ORDER BY directory_items.likes_received DESC LIMIT 50 OFFSET 0	SELECT DISTINCT "directory_items"."likes_received" AS "alias_0", "directory_items"."id" AS "id" FROM "group_users" AS "group_users" INNER JOIN "directory_items" AS "directory_items" ON "group_users"."user_id" = "directory_items"."user_id" WHERE "group_users"."group_id" = 2898 AND "directory_items"."period_type" = 1 ORDER BY "directory_items"."likes_received" DESC LIMIT 50 OFFSET 0
discourse	5044	SELECT "categories"."id" FROM "categories" INNER JOIN "category_search_data" ON "category_search_data"."category_id" = "categories"."id" WHERE (category_search_data.locale != ’fr’ OR category_search_data.version != 3) ORDER BY categories.id asc LIMIT 500	SELECT "category_search_data"."category_id" AS "category_id" FROM "category_search_data" AS "category_search_data" WHERE "category_search_data"."locale" <>’fr’ OR "category_search_data"."version" <>3 ORDER BY "category_search_data"."category_id" ASC LIMIT 500

discourse	5189	SELECT directory_items.likes_received AS alias_0, "directory_items"."id" FROM "directory_items" LEFT OUTER JOIN "users" ON "users"."id" = "directory_items"."user_id" LEFT OUTER JOIN "group_users" ON "group_users"."user_id" = "users"."id" WHERE "directory_items"."period_type" = 1 AND "group_users"."group_id" = 2898 ORDER BY directory_items.likes_received DESC LIMIT 50 OFFSET 0	SELECT "directory_items"."likes_received" AS "alias_0", "directory_items"."id" AS "id" FROM "group_users" AS "group_users" INNER JOIN "directory_items" AS "directory_items" ON "group_users"."user_id" = "directory_items"."user_id" WHERE "group_users"."group_id" = 2898 AND "directory_items"."period_type" = 1 ORDER BY "directory_items"."likes_received" DESC LIMIT 50 OFFSET 0
eladmin	28	select role0_.id as id1_16_, role0_.create_time as create_t2_16_, role0_.data_scope as data_sco3_16_, role0_.level as level4_16_, role0_.name as name5_16_, role0_.remark as remark6_16_ from role role0_ left outer join users_roles users1_ on role0_.id=users1_.role_id left outer join user user2_ on users1_.user_id=user2_.id where user2_.id=1	SELECT ‘role0_‘.‘id‘ AS ‘id1_16_‘, ‘role0_‘.‘create_time‘ AS ‘create_t2_16_‘, ‘role0_‘.‘data_scope‘ AS ‘data_sco3_16_‘, ‘role0_‘.‘level‘ AS ‘level4_16_‘, ‘role0_‘.‘name‘ AS ‘name5_16_‘, ‘role0_‘.‘remark‘ AS ‘remark6_16_‘ FROM ‘role‘ AS ‘role0_‘ INNER JOIN ‘users_roles‘ AS ‘users1_‘ ON ‘role0_‘.‘id‘ = ‘users1_‘.‘role_id‘ WHERE ‘users1_‘.‘user_id‘ = 1
eladmin	58	select user0_.id as id1_20_, user0_.create_time as create_t2_20_, user0_.dept_id as dept_id9_20_, user0_.email as email3_20_, user0_.enabled as enabled4_20_, user0_.job_id as job_id10_20_, user0_.last_password_reset_time as last_pas5_20_, user0_.password as password6_20_, user0_.phone as phone7_20_, user0_.avatar_id as avatar_11_20_, user0_.username as username8_20_ from user user0_ left outer join dept dept1_ on user0_.dept_id=dept1_.id where dept1_.id in (2) order by user0_.id desc limit 10	SELECT ‘user0_‘.‘id‘ AS ‘id1_20_‘, ‘user0_‘.‘create_time‘ AS ‘create_t2_20_‘, ‘user0_‘.‘dept_id‘ AS ‘dept_id9_20_‘, ‘user0_‘.‘email‘ AS ‘email3_20_‘, ‘user0_‘.‘enabled‘ AS ‘enabled4_20_‘, ‘user0_‘.‘job_id‘ AS ‘job_id10_20_‘, ‘user0_‘.‘last_password_reset_time‘ AS ‘last_pas5_20_‘, ‘user0_‘.‘password‘ AS ‘password6_20_‘, ‘user0_‘.‘phone‘ AS ‘phone7_20_‘, ‘user0_‘.‘avatar_id‘ AS ‘avatar_11_20_‘, ‘user0_‘.‘username‘ AS ‘username8_20_‘ FROM ‘user‘ AS ‘user0_‘ WHERE ‘user0_‘.‘dept_id‘ IN (?) ORDER BY ‘user0_‘.‘id‘ DESC LIMIT 10
fatfreecrm	16	SELECT ‘groups‘.‘id‘ FROM ‘groups‘ INNER JOIN ‘groups_users‘ ON ‘groups‘.‘id‘ = ‘groups_users‘.‘group_id‘ WHERE ‘groups_users‘.‘user_id‘ = 3056	SELECT ‘groups_users‘.‘group_id‘ AS ‘group_id‘ FROM ‘groups_users‘ AS ‘groups_users‘ WHERE ‘groups_users‘.‘user_id‘ = 3056
fatfreecrm	99	SELECT COUNT(*) FROM ‘groups‘ INNER JOIN ‘groups_users‘ ON ‘groups‘.‘id‘ = ‘groups_users‘.‘group_id‘ WHERE ‘groups_users‘.‘user_id‘ = 1086	SELECT COUNT(*) FROM ‘groups_users‘ AS ‘groups_users‘ WHERE ‘groups_users‘.‘user_id‘ = 1086
fatfreecrm	141	SELECT DISTINCT ‘opportunities‘.‘id‘ FROM ‘opportunities‘ INNER JOIN ‘contact_opportunities‘ ON ‘opportunities‘.‘id‘ = ‘contact_opportunities‘.‘opportunity_id‘ WHERE ‘contact_opportunities‘.‘contact_id‘ = 2076 ORDER BY opportunities.id DESC	SELECT DISTINCT ‘contact_opportunities‘.‘opportunity_id‘ AS ‘opportunity_id‘ FROM ‘contact_opportunities‘ AS ‘contact_opportunities‘ WHERE ‘contact_opportunities‘.‘contact_id‘ = 2076 ORDER BY ‘contact_opportunities‘.‘opportunity_id‘ DESC
fatfreecrm	155	SELECT DISTINCT ‘opportunities‘.‘id‘ FROM ‘opportunities‘ INNER JOIN ‘account_opportunities‘ ON ‘opportunities‘.‘id‘ = ‘account_opportunities‘.‘opportunity_id‘ WHERE ‘account_opportunities‘.‘account_id‘ = 1106 ORDER BY opportunities.id DESC	SELECT DISTINCT ‘account_opportunities‘.‘opportunity_id‘ AS ‘opportunity_id‘ FROM ‘account_opportunities‘ AS ‘account_opportunities‘ WHERE ‘account_opportunities‘.‘account_id‘ = 1106 ORDER BY ‘account_opportunities‘.‘opportunity_id‘ DESC
fatfreecrm	156	SELECT DISTINCT ‘contacts‘.‘id‘ FROM ‘contacts‘ INNER JOIN ‘account_contacts‘ ON ‘contacts‘.‘id‘ = ‘account_contacts‘.‘contact_id‘ WHERE ‘account_contacts‘.‘account_id‘ = 1106	SELECT DISTINCT ‘account_contacts‘.‘contact_id‘ AS ‘contact_id‘ FROM ‘account_contacts‘ AS ‘account_contacts‘ WHERE ‘account_contacts‘.‘account_id‘ = 1106
fatfreecrm	191	SELECT DISTINCT ‘contacts‘.‘id‘ FROM ‘contacts‘ INNER JOIN ‘contact_opportunities‘ ON ‘contacts‘.‘id‘ = ‘contact_opportunities‘.‘contact_id‘ WHERE ‘contact_opportunities‘.‘opportunity_id‘ = 1128 ORDER BY contacts.id DESC	SELECT DISTINCT ‘contact_opportunities‘.‘contact_id‘ AS ‘contact_id‘ FROM ‘contact_opportunities‘ AS ‘contact_opportunities‘ WHERE ‘contact_opportunities‘.‘opportunity_id‘ = 1128 ORDER BY ‘contact_opportunities‘.‘contact_id‘ DESC
gitlab	808	SELECT MIN(users.id) FROM "users" INNER JOIN "identities" ON "identities"."user_id" = "users"."id" WHERE (identities.provider LIKE ’ldap%’) /*application:test,controller:application_settings,action:usage_data,correlation_id:6a0c0fa1ca2505711e59296537dcdff3*/	SELECT MIN("identities"."user_id") FROM "identities" AS "identities" WHERE "identities"."provider" LIKE ’ldap%’
gitlab	809	SELECT MAX(users.id) FROM "users" INNER JOIN "identities" ON "identities"."user_id" = "users"."id" WHERE (identities.provider LIKE ’ldap%’) /*application:test,controller:application_settings,action:usage_data,correlation_id:6a0c0fa1ca2505711e59296537dcdff3*/	SELECT MAX("identities"."user_id") FROM "identities" AS "identities" WHERE "identities"."provider" LIKE ’ldap%’
gitlab	810	SELECT COUNT(users.id) FROM "users" INNER JOIN "identities" ON "identities"."user_id" = "users"."id" WHERE (identities.provider LIKE ’ldap%’) AND "users"."id" BETWEEN $1 AND $2 /*application:test,controller:application_settings,action:usage_data,correlation_id:6a0c0fa1ca2505711e59296537dcdff3*/	SELECT COUNT("identities"."user_id") FROM "identities" AS "identities" WHERE "identities"."user_id" BETWEEN $1 AND $2 AND "identities"."provider" LIKE ’ldap%’
homeland	12	SELECT "users"."id" FROM "users" INNER JOIN "actions" ON "users"."id" = "actions"."user_id" WHERE "actions"."target_id" = $1 AND "actions"."action_type" = $2 AND "actions"."target_type" = $3 AND "actions"."user_type" = $4 AND "actions"."user_type" = $5	SELECT "actions"."user_id" AS "user_id" FROM "actions" AS "actions" WHERE "actions"."user_type" = $5 AND "actions"."user_type" = $4 AND "actions"."target_type" = $3 AND "actions"."target_id" = $1 AND "actions"."action_type" = $2
homeland	72	SELECT "nodes"."id" FROM "nodes" INNER JOIN "actions" ON "nodes"."id" = "actions"."target_id" WHERE "actions"."user_id" = $1 AND "actions"."action_type" = $2 AND "actions"."target_type" = $3 AND "actions"."user_type" = $4 AND "actions"."target_type" = $5	SELECT "actions"."target_id" AS "target_id" FROM "actions" AS "actions" WHERE "actions"."user_type" = $4 AND "actions"."user_id" = $1 AND "actions"."target_type" = $5 AND "actions"."target_type" = $3 AND "actions"."action_type" = $2
homeland	73	SELECT "users"."id" FROM "users" INNER JOIN "actions" ON "users"."id" = "actions"."target_id" WHERE "actions"."user_id" = $1 AND "actions"."action_type" = $2 AND "actions"."target_type" = $3 AND "actions"."user_type" = $4 AND "actions"."target_type" = $5	SELECT "actions"."target_id" AS "target_id" FROM "actions" AS "actions" WHERE "actions"."user_type" = $4 AND "actions"."user_id" = $1 AND "actions"."target_type" = $5 AND "actions"."target_type" = $3 AND "actions"."action_type" = $2

homeland	79	SELECT "users"."id" FROM "users" INNER JOIN "team_users" ON "users"."id" = "team_users"."user_id" WHERE "team_users"."team_id" = 1	SELECT "team_users"."user_id" AS "user_id" FROM "team_users" AS "team_users" WHERE "team_users"."team_id" = 1
lobsters	134	SELECT ‘stories‘.‘id‘ FROM ‘stories‘ INNER JOIN ‘taggings‘ ON ‘taggings‘.‘story_id‘ = ‘stories‘.‘id‘ INNER JOIN ‘tags‘ ON ‘tags‘.‘id‘ = ‘taggings‘.‘tag_id‘ INNER JOIN ‘users‘ ON ‘users‘.‘id‘ = ‘stories‘.‘user_id‘ WHERE ‘stories‘.‘merged_story_id‘ IS NULL AND ‘stories‘.‘is_expired‘ = FALSE AND ‘tags‘.‘tag‘ IN (’tag1’, ’tag2’) GROUP BY stories.id HAVING (COUNT(stories.id) = 2)	SELECT ‘stories‘.‘id‘ AS ‘id‘ FROM ‘stories‘ AS ‘stories‘ INNER JOIN ‘taggings‘ AS ‘taggings‘ ON ‘stories‘.‘id‘ = ‘taggings‘.‘story_id‘ INNER JOIN ‘tags‘ AS ‘tags‘ ON ‘taggings‘.‘tag_id‘ = ‘tags‘.‘id‘ WHERE ‘tags‘.‘tag‘ IN (?) AND ‘stories‘.‘merged_story_id‘ IS NULL AND ‘stories‘.‘is_expired‘ = FALSE
redmine	182	SELECT DISTINCT ‘projects‘.* FROM ‘projects‘ LEFT JOIN projects child ON projects.lft <= child.lft AND projects.rgt >= child.rgt WHERE (projects.status <>9) AND (child.id IN (1957)) ORDER BY projects.lft ASC	SELECT * FROM (SELECT DISTINCT ‘projects‘.‘id‘ AS ‘id‘, ‘projects‘.‘name‘ AS ‘name‘, ‘projects‘.‘description‘ AS ‘description‘, ‘projects‘.‘homepage‘ AS ‘homepage‘, ‘projects‘.‘is_public‘ AS ‘is_public‘, ‘projects‘.‘parent_id‘ AS ‘parent_id‘, ‘projects‘.‘created_on‘ AS ‘created_on‘, ‘projects‘.‘updated_on‘ AS ‘updated_on‘, ‘projects‘.‘identifier‘ AS ‘identifier‘, ‘projects‘.‘status‘ AS ‘status‘, ‘projects‘.‘lft‘ AS ‘lft‘, ‘projects‘.‘rgt‘ AS ‘rgt‘, ‘projects‘.‘inherit_members‘ AS ‘inherit_members‘, ‘projects‘.‘default_version_id‘ AS ‘default_version_id‘, ‘projects‘.‘default_assigned_to_id‘ AS ‘default_assigned_to_id‘ FROM ‘projects‘ AS ‘projects‘ INNER JOIN ‘projects‘ AS ‘child‘ ON ‘projects‘.‘lft‘ <= ‘child‘.‘lft‘ AND ‘projects‘.‘rgt‘ >= ‘child‘.‘rgt‘ WHERE ‘projects‘.‘status‘ <>9) AS ‘sub_0‘ WHERE ‘sub_0‘.‘id‘ IN (?) ORDER BY ‘sub_0‘.‘lft‘ ASC
redmine	207	SELECT ‘issues‘.‘id‘ AS t0_r0, ‘issues‘.‘tracker_id‘ AS t0_r1, ‘issues‘.‘project_id‘ AS t0_r2, ‘issues‘.‘subject‘ AS t0_r3, ‘issues‘.‘description‘ AS t0_r4, ‘issues‘.‘due_date‘ AS t0_r5, ‘issues‘.‘category_id‘ AS t0_r6, ‘issues‘.‘status_id‘ AS t0_r7, ‘issues‘.‘assigned_to_id‘ AS t0_r8, ‘issues‘.‘priority_id‘ AS t0_r9, ‘issues‘.‘fixed_version_id‘ AS t0_r10, ‘issues‘.‘author_id‘ AS t0_r11, ‘issues‘.‘lock_version‘ AS t0_r12, ‘issues‘.‘created_on‘ AS t0_r13, ‘issues‘.‘updated_on‘ AS t0_r14, ‘issues‘.‘start_date‘ AS t0_r15, ‘issues‘.‘done_ratio‘ AS t0_r16, ‘issues‘.‘estimated_hours‘ AS t0_r17, ‘issues‘.‘parent_id‘ AS t0_r18, ‘issues‘.‘root_id‘ AS t0_r19, ‘issues‘.‘lft‘ AS t0_r20, ‘issues‘.‘rgt‘ AS t0_r21, ‘issues‘.‘is_private‘ AS t0_r22, ‘issues‘.‘closed_on‘ AS t0_r23, ‘issue_statuses‘.‘id‘ AS t1_r0, ‘issue_statuses‘.‘name‘ AS t1_r1, ‘issue_statuses‘.‘is_closed‘ AS t1_r2, ‘issue_statuses‘.‘position‘ AS t1_r3, ‘issue_statuses‘.‘default_done_ratio‘ AS t1_r4, ‘projects‘.‘id‘ AS t2_r0, ‘projects‘.‘name‘ AS t2_r1, ‘projects‘.‘description‘ AS t2_r2, ‘projects‘.‘homepage‘ AS t2_r3, ‘projects‘.‘is_public‘ AS t2_r4, ‘projects‘.‘parent_id‘ AS t2_r5, ‘projects‘.‘created_on‘ AS t2_r6, ‘projects‘.‘updated_on‘ AS t2_r7, ‘projects‘.‘identifier‘ AS t2_r8, ‘projects‘.‘status‘ AS t2_r9, ‘projects‘.‘lft‘ AS t2_r10, ‘projects‘.‘rgt‘ AS t2_r11, ‘projects‘.‘inherit_members‘ AS t2_r12, ‘projects‘.‘default_version_id‘ AS t2_r13, ‘projects‘.‘default_assigned_to_id‘ AS t2_r14, ‘users‘.‘id‘ AS t3_r0, ‘users‘.‘login‘ AS t3_r1, ‘users‘.‘hashed_password‘ AS t3_r2, ‘users‘.‘firstname‘ AS t3_r3, ‘users‘.‘lastname‘ AS t3_r4, ‘users‘.‘admin‘ AS t3_r5, ‘users‘.‘status‘ AS t3_r6, ‘users‘.‘last_login_on‘ AS t3_r7, ‘users‘.‘language‘ AS t3_r8, ‘users‘.‘auth_source_id‘ AS t3_r9, ‘users‘.‘created_on‘ AS t3_r10, ‘users‘.‘updated_on‘ AS t3_r11, ‘users‘.‘type‘ AS t3_r12, ‘users‘.‘identity_url‘ AS t3_r13, ‘users‘.‘mail_notification‘ AS t3_r14, ‘users‘.‘salt‘ AS t3_r15, ‘users‘.‘must_change_passwd‘ AS t3_r16, ‘users‘.‘passwd_changed_on‘ AS t3_r17, ‘trackers‘.‘id‘ AS t4_r0, ‘trackers‘.‘name‘ AS t4_r1, ‘trackers‘.‘description‘ AS t4_r2, ‘trackers‘.‘is_in_chlog‘ AS t4_r3, ‘trackers‘.‘position‘ AS t4_r4, ‘trackers‘.‘is_in_roadmap‘ AS t4_r5, ‘trackers‘.‘fields_bits‘ AS t4_r6, ‘trackers‘.‘default_status_id‘ AS t4_r7, ‘enumerations‘.‘id‘ AS t5_r0, ‘enumerations‘.‘name‘ AS t5_r1, ‘enumerations‘.‘position‘ AS t5_r2, ‘enumerations‘.‘is_default‘ AS t5_r3, ‘enumerations‘.‘type‘ AS t5_r4, ‘enumerations‘.‘active‘ AS t5_r5, ‘enumerations‘.‘project_id‘ AS t5_r6, ‘enumerations‘.‘parent_id‘ AS t5_r7, ‘enumerations‘.‘position_name‘ AS t5_r8, ‘issue_categories‘.‘id‘ AS t6_r0, ‘issue_categories‘.‘project_id‘ AS t6_r1, ‘issue_categories‘.‘name‘ AS t6_r2, ‘issue_categories‘.‘assigned_to_id‘ AS t6_r3, ‘versions‘.‘id‘ AS t7_r0, ‘versions‘.‘project_id‘ AS t7_r1, ‘versions‘.‘name‘ AS t7_r2, ‘versions‘.‘description‘ AS t7_r3, ‘versions‘.‘effective_date‘ AS t7_r4, ‘versions‘.‘created_on‘ AS t7_r5, ‘versions‘.‘updated_on‘ AS t7_r6, ‘versions‘.‘wiki_page_title‘ AS t7_r7, ‘versions‘.‘status‘ AS t7_r8, ‘versions‘.‘sharing‘ AS t7_r9 FROM ‘issues‘ INNER JOIN ‘projects‘ ON ‘projects‘.‘id‘ = ‘issues‘.‘project_id‘ INNER JOIN ‘issue_statuses‘ ON ‘issue_statuses‘.‘id‘ = ‘issues‘.‘status_id‘ LEFT OUTER JOIN ‘users‘ ON ‘users‘.‘id‘ = ‘issues‘.‘assigned_to_id‘ LEFT OUTER JOIN ‘trackers‘ ON ‘trackers‘.‘id‘ = ‘issues‘.‘tracker_id‘ LEFT OUTER JOIN ‘enumerations‘ ON ‘enumerations‘.‘id‘ = ‘issues‘.‘priority_id‘ AND ‘enumerations‘.‘type‘ IN (’IssuePriority’) LEFT OUTER JOIN ‘issue_categories‘ ON ‘issue_categories‘.‘id‘ = ‘issues‘.‘category_id‘ LEFT OUTER JOIN ‘versions‘ ON ‘versions‘.‘id‘ = ‘issues‘.‘fixed_version_id‘ WHERE (projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’issue_tracking’)) AND ((issues.status_id IN (SELECT id FROM issue_statuses WHERE is_closed=FALSE)) AND projects.id = 1957) ORDER BY projects.lft ASC, issues.id ASC LIMIT 500	SELECT * FROM (SELECT ‘issues‘.‘id‘ AS ‘t0_r0‘, ‘issues‘.‘tracker_id‘ AS ‘t0_r1‘, ‘issues‘.‘project_id‘ AS ‘t0_r2‘, ‘issues‘.‘subject‘ AS ‘t0_r3‘, ‘issues‘.‘description‘ AS ‘t0_r4‘, ‘issues‘.‘due_date‘ AS ‘t0_r5‘, ‘issues‘.‘category_id‘ AS ‘t0_r6‘, ‘issues‘.‘status_id‘ AS ‘t0_r7‘, ‘issues‘.‘assigned_to_id‘ AS ‘t0_r8‘, ‘issues‘.‘priority_id‘ AS ‘t0_r9‘, ‘issues‘.‘fixed_version_id‘ AS ‘t0_r10‘, ‘issues‘.‘author_id‘ AS ‘t0_r11‘, ‘issues‘.‘lock_version‘ AS ‘t0_r12‘, ‘issues‘.‘created_on‘ AS ‘t0_r13‘, ‘issues‘.‘updated_on‘ AS ‘t0_r14‘, ‘issues‘.‘start_date‘ AS ‘t0_r15‘, ‘issues‘.‘done_ratio‘ AS ‘t0_r16‘, ‘issues‘.‘estimated_hours‘ AS ‘t0_r17‘, ‘issues‘.‘parent_id‘ AS ‘t0_r18‘, ‘issues‘.‘root_id‘ AS ‘t0_r19‘, ‘issues‘.‘lft‘ AS ‘t0_r20‘, ‘issues‘.‘rgt‘ AS ‘t0_r21‘, ‘issues‘.‘is_private‘ AS ‘t0_r22‘, ‘issues‘.‘closed_on‘ AS ‘t0_r23‘, ‘issue_statuses‘.‘id‘ AS ‘t1_r0‘, ‘issue_statuses‘.‘name‘ AS ‘t1_r1‘, ‘issue_statuses‘.‘is_closed‘ AS ‘t1_r2‘, ‘issue_statuses‘.‘position‘ AS ‘t1_r3‘, ‘issue_statuses‘.‘default_done_ratio‘ AS ‘t1_r4‘, ‘projects‘.‘id‘ AS ‘t2_r0‘, ‘projects‘.‘name‘ AS ‘t2_r1‘, ‘projects‘.‘description‘ AS ‘t2_r2‘, ‘projects‘.‘homepage‘ AS ‘t2_r3‘, ‘projects‘.‘is_public‘ AS ‘t2_r4‘, ‘projects‘.‘parent_id‘ AS ‘t2_r5‘, ‘projects‘.‘created_on‘ AS ‘t2_r6‘, ‘projects‘.‘updated_on‘ AS ‘t2_r7‘, ‘projects‘.‘identifier‘ AS ‘t2_r8‘, ‘projects‘.‘status‘ AS ‘t2_r9‘, ‘projects‘.‘lft‘ AS ‘t2_r10‘, ‘projects‘.‘rgt‘ AS ‘t2_r11‘, ‘projects‘.‘inherit_members‘ AS ‘t2_r12‘, ‘projects‘.‘default_version_id‘ AS ‘t2_r13‘, ‘projects‘.‘default_assigned_to_id‘ AS ‘t2_r14‘, ‘users‘.‘id‘ AS ‘t3_r0‘, ‘users‘.‘login‘ AS ‘t3_r1‘, ‘users‘.‘hashed_password‘ AS ‘t3_r2‘, ‘users‘.‘firstname‘ AS ‘t3_r3‘, ‘users‘.‘lastname‘ AS ‘t3_r4‘, ‘users‘.‘admin‘ AS ‘t3_r5‘, ‘users‘.‘status‘ AS ‘t3_r6‘, ‘users‘.‘last_login_on‘ AS ‘t3_r7‘, ‘users‘.‘language‘ AS ‘t3_r8‘, ‘users‘.‘auth_source_id‘ AS ‘t3_r9‘, ‘users‘.‘created_on‘ AS ‘t3_r10‘, ‘users‘.‘updated_on‘ AS ‘t3_r11‘, ‘users‘.‘type‘ AS ‘t3_r12‘, ‘users‘.‘identity_url‘ AS ‘t3_r13‘, ‘users‘.‘mail_notification‘ AS ‘t3_r14‘, ‘users‘.‘salt‘ AS ‘t3_r15‘, ‘users‘.‘must_change_passwd‘ AS ‘t3_r16‘, ‘users‘.‘passwd_changed_on‘ AS ‘t3_r17‘, ‘trackers‘.‘id‘ AS ‘t4_r0‘, ‘trackers‘.‘name‘ AS ‘t4_r1‘, ‘trackers‘.‘description‘ AS ‘t4_r2‘, ‘trackers‘.‘is_in_chlog‘ AS ‘t4_r3‘, ‘trackers‘.‘position‘ AS ‘t4_r4‘, ‘trackers‘.‘is_in_roadmap‘ AS ‘t4_r5‘, ‘trackers‘.‘fields_bits‘ AS ‘t4_r6‘, ‘trackers‘.‘default_status_id‘ AS ‘t4_r7‘, ‘enumerations‘.‘id‘ AS ‘t5_r0‘, ‘enumerations‘.‘name‘ AS ‘t5_r1‘, ‘enumerations‘.‘position‘ AS ‘t5_r2‘, ‘enumerations‘.‘is_default‘ AS ‘t5_r3‘, ‘enumerations‘.‘type‘ AS ‘t5_r4‘, ‘enumerations‘.‘active‘ AS ‘t5_r5‘, ‘enumerations‘.‘project_id‘ AS ‘t5_r6‘, ‘enumerations‘.‘parent_id‘ AS ‘t5_r7‘, ‘enumerations‘.‘position_name‘ AS ‘t5_r8‘, ‘issue_categories‘.‘id‘ AS ‘t6_r0‘, ‘issue_categories‘.‘project_id‘ AS ‘t6_r1‘, ‘issue_categories‘.‘name‘ AS ‘t6_r2‘, ‘issue_categories‘.‘assigned_to_id‘ AS ‘t6_r3‘, ‘versions‘.‘id‘ AS ‘t7_r0‘, ‘versions‘.‘project_id‘ AS ‘t7_r1‘, ‘versions‘.‘name‘ AS ‘t7_r2‘, ‘versions‘.‘description‘ AS ‘t7_r3‘, ‘versions‘.‘effective_date‘ AS ‘t7_r4‘, ‘versions‘.‘created_on‘ AS ‘t7_r5‘, ‘versions‘.‘updated_on‘ AS ‘t7_r6‘, ‘versions‘.‘wiki_page_title‘ AS ‘t7_r7‘, ‘versions‘.‘status‘ AS ‘t7_r8‘, ‘versions‘.‘sharing‘ AS ‘t7_r9‘ FROM ‘issues‘ AS ‘issues‘ INNER JOIN ‘issue_statuses‘ AS ‘issue_statuses‘ ON ‘issues‘.‘status_id‘ = ‘issue_statuses‘.‘id‘ LEFT JOIN ‘enumerations‘ AS ‘enumerations‘ ON ‘enumerations‘.‘id‘ = ‘issues‘.‘priority_id‘ AND ‘enumerations‘.‘type‘ IN (?) INNER JOIN ‘versions‘ AS ‘versions‘ ON ‘issues‘.‘fixed_version_id‘ = ‘versions‘.‘id‘ INNER JOIN ‘issue_categories‘ AS ‘issue_categories‘ ON ‘issues‘.‘category_id‘ = ‘issue_categories‘.‘id‘ INNER JOIN ‘users‘ AS ‘users‘ ON ‘issues‘.‘assigned_to_id‘ = ‘users‘.‘id‘ INNER JOIN ‘trackers‘ AS ‘trackers‘ ON ‘issues‘.‘tracker_id‘ = ‘trackers‘.‘id‘ INNER JOIN ‘projects‘ AS ‘projects‘ ON ‘issues‘.‘project_id‘ = ‘projects‘.‘id‘ WHERE EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘projects‘.‘id‘ AND ‘em‘.‘name‘ = ’issue_tracking’) AND ‘projects‘.‘status‘ <>9) AS ‘sub_0‘ WHERE ‘sub_0‘.‘t0_r2‘ = 1957 AND ‘sub_0‘.‘t0_r7‘ IN (SELECT ‘issue_statuses‘.‘id‘ AS ‘id‘ FROM ‘issue_statuses‘ AS ‘issue_statuses‘ WHERE ‘issue_statuses‘.‘is_closed‘ = FALSE) ORDER BY ‘sub_0‘.‘t2_r10‘ ASC, ‘sub_0‘.‘t0_r0‘ ASC LIMIT 500

redmine	216	SELECT ‘custom_fields‘.‘id‘, role_id FROM ‘custom_fields‘ INNER JOIN ‘custom_fields_roles‘ ON ‘custom_fields_roles‘.‘custom_field_id‘ = ‘custom_fields‘.‘id‘ INNER JOIN ‘roles‘ ON ‘roles‘.‘id‘ = ‘custom_fields_roles‘.‘role_id‘ WHERE ‘custom_fields‘.‘type‘ IN (’IssueCustomField’) AND ‘custom_fields‘.‘visible‘ = FALSE	SELECT ‘custom_fields‘.‘id‘ AS ‘id‘, ‘custom_fields_roles‘.‘role_id‘ AS ‘role_id‘ FROM ‘custom_fields‘ AS ‘custom_fields‘ INNER JOIN ‘custom_fields_roles‘ AS ‘custom_fields_roles‘ ON ‘custom_fields‘.‘id‘ = ‘custom_fields_roles‘.‘custom_field_id‘ WHERE ‘custom_fields‘.‘visible‘ = FALSE AND ‘custom_fields‘.‘type‘ IN (?)
redmine	240	SELECT SUM(‘issues‘.‘estimated_hours‘) FROM ‘issues‘ INNER JOIN ‘projects‘ ON ‘projects‘.‘id‘ = ‘issues‘.‘project_id‘ INNER JOIN ‘issue_statuses‘ ON ‘issue_statuses‘.‘id‘ = ‘issues‘.‘status_id‘ WHERE (((projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’issue_tracking’)) AND (((projects.is_public = TRUE AND projects.id NOT IN (SELECT project_id FROM members WHERE user_id IN (6,13))) AND ((issues.is_private = FALSE))))))	SELECT SUM(‘issues‘.‘estimated_hours‘) FROM ‘projects‘ AS ‘projects‘ INNER JOIN ‘issues‘ AS ‘issues‘ ON ‘projects‘.‘id‘ = ‘issues‘.‘project_id‘ WHERE NOT ‘issues‘.‘project_id‘ IN (SELECT ‘project_id‘ FROM ‘members‘ WHERE ‘user_id‘ IN (?)) AND EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘issues‘.‘project_id‘ AND ‘em‘.‘name‘ = ’issue_tracking’) AND ‘projects‘.‘status‘ <>9 AND ‘projects‘.‘is_public‘ = TRUE AND ‘issues‘.‘is_private‘ = FALSE
redmine	252	SELECT SUM(time_entries.hours) FROM ‘issues‘ INNER JOIN ‘projects‘ ON ‘projects‘.‘id‘ = ‘issues‘.‘project_id‘ INNER JOIN ‘issue_statuses‘ ON ‘issue_statuses‘.‘id‘ = ‘issues‘.‘status_id‘ INNER JOIN ‘time_entries‘ ON ‘time_entries‘.‘issue_id‘ = ‘issues‘.‘id‘ WHERE (((projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’issue_tracking’)) AND (((projects.is_public = TRUE AND projects.id NOT IN (SELECT project_id FROM members WHERE user_id IN (6,13))) AND ((issues.is_private = FALSE)))))) AND (((projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’time_tracking’)) AND ((projects.is_public = TRUE AND projects.id NOT IN (SELECT project_id FROM members WHERE user_id IN (6,13))))))	SELECT SUM(‘time_entries‘.‘hours‘) FROM ‘issues‘ AS ‘issues‘ INNER JOIN ‘time_entries‘ AS ‘time_entries‘ ON ‘issues‘.‘id‘ = ‘time_entries‘.‘issue_id‘ INNER JOIN ‘projects‘ AS ‘projects‘ ON ‘issues‘.‘project_id‘ = ‘projects‘.‘id‘ WHERE EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘issues‘.‘project_id‘ AND ‘em‘.‘name‘ = ’issue_tracking’) AND EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘issues‘.‘project_id‘ AND ‘em‘.‘name‘ = ’time_tracking’) AND NOT ‘issues‘.‘project_id‘ IN (SELECT ‘project_id‘ FROM ‘members‘ WHERE ‘user_id‘ IN (?)) AND ‘issues‘.‘is_private‘ = FALSE AND ‘projects‘.‘is_public‘ = TRUE AND ‘projects‘.‘status‘ <>9
redmine	284	SELECT 1 AS one FROM ‘trackers‘ INNER JOIN ‘projects_trackers‘ ON ‘trackers‘.‘id‘ = ‘projects_trackers‘.‘tracker_id‘ WHERE ‘projects_trackers‘.‘project_id‘ = 1 LIMIT 1	SELECT 1 AS ‘one‘ FROM ‘projects_trackers‘ AS ‘projects_trackers‘ WHERE ‘projects_trackers‘.‘project_id‘ = 1 LIMIT 1
redmine	323	SELECT ‘roles‘.‘id‘ FROM ‘roles‘ INNER JOIN ‘custom_fields_roles‘ ON ‘roles‘.‘id‘ = ‘custom_fields_roles‘.‘role_id‘ WHERE ‘custom_fields_roles‘.‘custom_field_id‘ = 1	SELECT ‘custom_fields_roles‘.‘role_id‘ AS ‘role_id‘ FROM ‘custom_fields_roles‘ AS ‘custom_fields_roles‘ WHERE ‘custom_fields_roles‘.‘custom_field_id‘ = 1
redmine	324	SELECT ‘trackers‘.‘id‘ FROM ‘trackers‘ INNER JOIN ‘custom_fields_trackers‘ ON ‘trackers‘.‘id‘ = ‘custom_fields_trackers‘.‘tracker_id‘ WHERE ‘custom_fields_trackers‘.‘custom_field_id‘ = 1	SELECT ‘custom_fields_trackers‘.‘tracker_id‘ AS ‘tracker_id‘ FROM ‘custom_fields_trackers‘ AS ‘custom_fields_trackers‘ WHERE ‘custom_fields_trackers‘.‘custom_field_id‘ = 1
redmine	325	SELECT ‘projects‘.‘id‘ FROM ‘projects‘ INNER JOIN ‘custom_fields_projects‘ ON ‘projects‘.‘id‘ = ‘custom_fields_projects‘.‘project_id‘ WHERE ‘custom_fields_projects‘.‘custom_field_id‘ = 1	SELECT ‘custom_fields_projects‘.‘project_id‘ AS ‘project_id‘ FROM ‘custom_fields_projects‘ AS ‘custom_fields_projects‘ WHERE ‘custom_fields_projects‘.‘custom_field_id‘ = 1
redmine	341	SELECT ‘issues‘.‘id‘ FROM ‘issues‘ INNER JOIN ‘changesets_issues‘ ON ‘issues‘.‘id‘ = ‘changesets_issues‘.‘issue_id‘ WHERE ‘changesets_issues‘.‘changeset_id‘ = 339	SELECT ‘changesets_issues‘.‘issue_id‘ AS ‘issue_id‘ FROM ‘changesets_issues‘ AS ‘changesets_issues‘ WHERE ‘changesets_issues‘.‘changeset_id‘ = 339
redmine	344	SELECT ‘issues‘.* FROM ‘issues‘ INNER JOIN ‘projects‘ ON ‘projects‘.‘id‘ = ‘issues‘.‘project_id‘ INNER JOIN ‘versions‘ ON ‘versions‘.‘id‘ = ‘issues‘.‘fixed_version_id‘ WHERE (issues.fixed_version_id IS NOT NULL AND issues.project_id <>versions.project_id AND versions.sharing <>’system’) AND (versions.project_id IN (1969) OR issues.project_id IN (1969))	SELECT ‘issues‘.‘id‘ AS ‘id‘, ‘issues‘.‘tracker_id‘ AS ‘tracker_id‘, ‘issues‘.‘project_id‘ AS ‘project_id‘, ‘issues‘.‘subject‘ AS ‘subject‘, ‘issues‘.‘description‘ AS ‘description‘, ‘issues‘.‘due_date‘ AS ‘due_date‘, ‘issues‘.‘category_id‘ AS ‘category_id‘, ‘issues‘.‘status_id‘ AS ‘status_id‘, ‘issues‘.‘assigned_to_id‘ AS ‘assigned_to_id‘, ‘issues‘.‘priority_id‘ AS ‘priority_id‘, ‘issues‘.‘fixed_version_id‘ AS ‘fixed_version_id‘, ‘issues‘.‘author_id‘ AS ‘author_id‘, ‘issues‘.‘lock_version‘ AS ‘lock_version‘, ‘issues‘.‘created_on‘ AS ‘created_on‘, ‘issues‘.‘updated_on‘ AS ‘updated_on‘, ‘issues‘.‘start_date‘ AS ‘start_date‘, ‘issues‘.‘done_ratio‘ AS ‘done_ratio‘, ‘issues‘.‘estimated_hours‘ AS ‘estimated_hours‘, ‘issues‘.‘parent_id‘ AS ‘parent_id‘, ‘issues‘.‘root_id‘ AS ‘root_id‘, ‘issues‘.‘lft‘ AS ‘lft‘, ‘issues‘.‘rgt‘ AS ‘rgt‘, ‘issues‘.‘is_private‘ AS ‘is_private‘, ‘issues‘.‘closed_on‘ AS ‘closed_on‘ FROM ‘issues‘ AS ‘issues‘ INNER JOIN ‘versions‘ AS ‘versions‘ ON ‘issues‘.‘fixed_version_id‘ = ‘versions‘.‘id‘ WHERE ‘versions‘.‘sharing‘ <>’system’ AND (‘versions‘.‘project_id‘ IN (?) OR ‘issues‘.‘project_id‘ IN (?)) AND ‘issues‘.‘project_id‘ <>‘versions‘.‘project_id‘ AND NOT ‘issues‘.‘fixed_version_id‘ IS NULL

redmine	345	SELECT 1 AS one FROM ‘roles‘ INNER JOIN ‘member_roles‘ ON ‘roles‘.‘id‘ = ‘member_roles‘.‘role_id‘ WHERE ‘member_roles‘.‘member_id‘ = 1088 LIMIT 1	SELECT 1 AS ‘one‘ FROM ‘member_roles‘ AS ‘member_roles‘ WHERE ‘member_roles‘.‘member_id‘ = 1088 LIMIT 1
redmine	396	SELECT ‘queries‘.* FROM ‘queries‘ LEFT OUTER JOIN projects ON queries.project_id = projects.id WHERE ‘queries‘.‘type‘ IN (’TimeEntryQuery’) AND (queries.project_id IS NULL OR (projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’time_tracking’))) AND (queries.visibility <>0 OR queries.user_id = 1) AND (‘queries‘.‘project_id‘ = 1 OR ‘queries‘.‘project_id‘ IS NULL) ORDER BY ‘queries‘.‘name‘ ASC, ‘queries‘.‘id‘ ASC	SELECT * FROM (SELECT ‘queries‘.‘id‘ AS ‘id‘, ‘queries‘.‘project_id‘ AS ‘project_id‘, ‘queries‘.‘name‘ AS ‘name‘, ‘queries‘.‘filters‘ AS ‘filters‘, ‘queries‘.‘user_id‘ AS ‘user_id‘, ‘queries‘.‘column_names‘ AS ‘column_names‘, ‘queries‘.‘sort_criteria‘ AS ‘sort_criteria‘, ‘queries‘.‘group_by‘ AS ‘group_by‘, ‘queries‘.‘type‘ AS ‘type‘, ‘queries‘.‘visibility‘ AS ‘visibility‘, ‘queries‘.‘options‘ AS ‘options‘ FROM ‘queries‘ AS ‘queries‘ INNER JOIN ‘projects‘ AS ‘projects‘ ON ‘queries‘.‘project_id‘ = ‘projects‘.‘id‘ WHERE ‘queries‘.‘project_id‘ IS NULL OR ‘projects‘.‘status‘ <>9 AND EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘projects‘.‘id‘ AND ‘em‘.‘name‘ = ’time_tracking’)) AS ‘sub_0‘ WHERE (‘sub_0‘.‘visibility‘ <>0 OR ‘sub_0‘.‘user_id‘ = 1) AND ‘sub_0‘.‘type‘ IN (?) AND (‘sub_0‘.‘project_id‘ = 1 OR ‘sub_0‘.‘project_id‘ IS NULL) ORDER BY ‘sub_0‘.‘name‘ ASC, ‘sub_0‘.‘id‘ ASC
redmine	401	SELECT ‘queries‘.* FROM ‘queries‘ LEFT OUTER JOIN projects ON queries.project_id = projects.id WHERE ‘queries‘.‘type‘ IN (’IssueQuery’) AND (queries.project_id IS NULL OR (((projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’issue_tracking’)) AND ((projects.is_public = TRUE AND projects.id NOT IN (SELECT project_id FROM members WHERE user_id IN (2,12))) OR projects.id IN (1,5) OR projects.id IN (2))))) AND (queries.visibility = 2 OR (queries.visibility = 1 AND queries.id IN (SELECT DISTINCT q.id FROM queries q INNER JOIN queries_roles qr on qr.query_id = q.id INNER JOIN member_roles mr ON mr.role_id = qr.role_id INNER JOIN members m ON m.id = mr.member_id AND m.user_id = 2 INNER JOIN projects p ON p.id = m.project_id AND p.status <>9 WHERE q.project_id IS NULL OR q.project_id = m.project_id)) OR queries.user_id = 2) AND ‘queries‘.‘project_id‘ IS NULL ORDER BY ‘queries‘.‘name‘ ASC, ‘queries‘.‘id‘ ASC	SELECT * FROM (SELECT ‘queries‘.‘id‘ AS ‘id‘, ‘queries‘.‘project_id‘ AS ‘project_id‘, ‘queries‘.‘name‘ AS ‘name‘, ‘queries‘.‘filters‘ AS ‘filters‘, ‘queries‘.‘user_id‘ AS ‘user_id‘, ‘queries‘.‘column_names‘ AS ‘column_names‘, ‘queries‘.‘sort_criteria‘ AS ‘sort_criteria‘, ‘queries‘.‘group_by‘ AS ‘group_by‘, ‘queries‘.‘type‘ AS ‘type‘, ‘queries‘.‘visibility‘ AS ‘visibility‘, ‘queries‘.‘options‘ AS ‘options‘ FROM ‘projects‘ AS ‘projects‘ INNER JOIN ‘queries‘ AS ‘queries‘ ON ‘projects‘.‘id‘ = ‘queries‘.‘project_id‘ WHERE ‘queries‘.‘project_id‘ IS NULL OR ‘projects‘.‘status‘ <>9 AND EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘projects‘.‘id‘ AND ‘em‘.‘name‘ = ’issue_tracking’) AND (‘projects‘.‘is_public‘ = TRUE AND NOT ‘projects‘.‘id‘ IN (SELECT ‘project_id‘ FROM ‘members‘ WHERE ‘user_id‘ IN (?)) OR ‘projects‘.‘id‘ IN (?) OR ‘projects‘.‘id‘ IN (?))) AS ‘sub_0‘ WHERE ‘sub_0‘.‘project_id‘ IS NULL AND (‘sub_0‘.‘visibility‘ = 2 OR ‘sub_0‘.‘visibility‘ = 1 AND ‘sub_0‘.‘id‘ IN (SELECT DISTINCT ‘q‘.‘id‘ FROM ‘queries‘ AS ‘q‘ INNER JOIN ‘queries_roles‘ AS ‘qr‘ ON ‘qr‘.‘query_id‘ = ‘q‘.‘id‘ INNER JOIN ‘member_roles‘ AS ‘mr‘ ON ‘mr‘.‘role_id‘ = ‘qr‘.‘role_id‘ INNER JOIN ‘members‘ AS ‘m‘ ON ‘m‘.‘id‘ = ‘mr‘.‘member_id‘ INNER JOIN ‘projects‘ AS ‘p‘ ON ‘p‘.‘id‘ = ‘m‘.‘project_id‘ WHERE (‘q‘.‘project_id‘ IS NULL OR ‘q‘.‘project_id‘ = ‘m‘.‘project_id‘) AND ‘m‘.‘user_id‘ = 2 AND ‘p‘.‘status‘ <>9) OR ‘sub_0‘.‘user_id‘ = 2) AND ‘sub_0‘.‘type‘ IN (?) ORDER BY ‘sub_0‘.‘name‘ ASC, ‘sub_0‘.‘id‘ ASC
redmine	440	SELECT COUNT(*) FROM ‘issues‘ INNER JOIN ‘projects‘ ON ‘projects‘.‘id‘ = ‘issues‘.‘project_id‘ INNER JOIN ‘issue_statuses‘ ON ‘issue_statuses‘.‘id‘ = ‘issues‘.‘status_id‘ WHERE (((projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’issue_tracking’)) AND (((projects.is_public = TRUE AND projects.id NOT IN (SELECT project_id FROM members WHERE user_id IN (2,12))) AND ((issues.is_private = FALSE OR issues.author_id = 2 OR issues.assigned_to_id IN (2)))) OR (projects.id IN (1,5) AND (1=1)) OR (projects.id IN (2) AND ((issues.is_private = FALSE OR issues.author_id = 2 OR issues.assigned_to_id IN (2))))))) AND ((issues.status_id IN (SELECT id FROM issue_statuses WHERE is_closed=FALSE)) AND projects.lft >= 1 AND projects.rgt <= 10)	SELECT COUNT(*) FROM ‘issues‘ AS ‘issues‘ INNER JOIN ‘projects‘ AS ‘projects‘ ON ‘issues‘.‘project_id‘ = ‘projects‘.‘id‘ WHERE EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘issues‘.‘project_id‘ AND ‘em‘.‘name‘ = ’issue_tracking’) AND ‘issues‘.‘status_id‘ IN (SELECT ‘issue_statuses‘.‘id‘ AS ‘id‘ FROM ‘issue_statuses‘ AS ‘issue_statuses‘ WHERE ‘issue_statuses‘.‘is_closed‘ = FALSE) AND ‘projects‘.‘status‘ <>9 AND ‘projects‘.‘rgt‘ <= 10 AND ‘projects‘.‘lft‘ >= 1 AND (‘projects‘.‘is_public‘ = TRUE AND NOT ‘projects‘.‘id‘ IN (SELECT ‘project_id‘ FROM ‘members‘ WHERE ‘user_id‘ IN (?)) AND (‘issues‘.‘is_private‘ = FALSE OR ‘issues‘.‘author_id‘ = 2 OR ‘issues‘.‘assigned_to_id‘ IN (?)) OR ‘projects‘.‘id‘ IN (?) OR ‘projects‘.‘id‘ IN (?) AND (‘issues‘.‘is_private‘ = FALSE OR ‘issues‘.‘author_id‘ = 2 OR ‘issues‘.‘assigned_to_id‘ IN (?)))
redmine	442	SELECT ‘queries‘.* FROM ‘queries‘ LEFT OUTER JOIN projects ON queries.project_id = projects.id WHERE ‘queries‘.‘type‘ IN (’IssueQuery’) AND (queries.project_id IS NULL OR (((projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’issue_tracking’)) AND ((projects.is_public = TRUE AND projects.id NOT IN (SELECT project_id FROM members WHERE user_id IN (2,12))) OR projects.id IN (1,5) OR projects.id IN (2))))) AND (queries.visibility = 2 OR (queries.visibility = 1 AND queries.id IN (SELECT DISTINCT q.id FROM queries q INNER JOIN queries_roles qr on qr.query_id = q.id INNER JOIN member_roles mr ON mr.role_id = qr.role_id INNER JOIN members m ON m.id = mr.member_id AND m.user_id = 2 INNER JOIN projects p ON p.id = m.project_id AND p.status <>9 WHERE q.project_id IS NULL OR q.project_id = m.project_id)) OR queries.user_id = 2) AND (‘queries‘.‘project_id‘ = 1 OR ‘queries‘.‘project_id‘ IS NULL) ORDER BY ‘queries‘.‘name‘ ASC, ‘queries‘.‘id‘ ASC	SELECT * FROM (SELECT ‘queries‘.‘id‘ AS ‘id‘, ‘queries‘.‘project_id‘ AS ‘project_id‘, ‘queries‘.‘name‘ AS ‘name‘, ‘queries‘.‘filters‘ AS ‘filters‘, ‘queries‘.‘user_id‘ AS ‘user_id‘, ‘queries‘.‘column_names‘ AS ‘column_names‘, ‘queries‘.‘sort_criteria‘ AS ‘sort_criteria‘, ‘queries‘.‘group_by‘ AS ‘group_by‘, ‘queries‘.‘type‘ AS ‘type‘, ‘queries‘.‘visibility‘ AS ‘visibility‘, ‘queries‘.‘options‘ AS ‘options‘ FROM ‘queries‘ AS ‘queries‘ INNER JOIN ‘projects‘ AS ‘projects‘ ON ‘queries‘.‘project_id‘ = ‘projects‘.‘id‘ WHERE ‘queries‘.‘project_id‘ IS NULL OR ‘projects‘.‘status‘ <>9 AND EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘projects‘.‘id‘ AND ‘em‘.‘name‘ = ’issue_tracking’) AND (‘projects‘.‘is_public‘ = TRUE AND NOT ‘projects‘.‘id‘ IN (SELECT ‘project_id‘ FROM ‘members‘ WHERE ‘user_id‘ IN (?)) OR ‘projects‘.‘id‘ IN (?) OR ‘projects‘.‘id‘ IN (?))) AS ‘sub_0‘ WHERE (‘sub_0‘.‘visibility‘ = 2 OR ‘sub_0‘.‘visibility‘ = 1 AND ‘sub_0‘.‘id‘ IN (SELECT DISTINCT ‘q‘.‘id‘ FROM ‘queries‘ AS ‘q‘ INNER JOIN ‘queries_roles‘ AS ‘qr‘ ON ‘qr‘.‘query_id‘ = ‘q‘.‘id‘ INNER JOIN ‘member_roles‘ AS ‘mr‘ ON ‘mr‘.‘role_id‘ = ‘qr‘.‘role_id‘ INNER JOIN ‘members‘ AS ‘m‘ ON ‘m‘.‘id‘ = ‘mr‘.‘member_id‘ INNER JOIN ‘projects‘ AS ‘p‘ ON ‘p‘.‘id‘ = ‘m‘.‘project_id‘ WHERE (‘q‘.‘project_id‘ IS NULL OR ‘q‘.‘project_id‘ = ‘m‘.‘project_id‘) AND ‘m‘.‘user_id‘ = 2 AND ‘p‘.‘status‘ <>9) OR ‘sub_0‘.‘user_id‘ = 2) AND ‘sub_0‘.‘type‘ IN (?) AND (‘sub_0‘.‘project_id‘ = 1 OR ‘sub_0‘.‘project_id‘ IS NULL) ORDER BY ‘sub_0‘.‘name‘ ASC, ‘sub_0‘.‘id‘ ASC

redmine	482	SELECT ‘issues‘.‘id‘ AS t0_r0, ‘issues‘.‘tracker_id‘ AS t0_r1, ‘issues‘.‘project_id‘ AS t0_r2, ‘issues‘.‘subject‘ AS t0_r3, ‘issues‘.‘description‘ AS t0_r4, ‘issues‘.‘due_date‘ AS t0_r5, ‘issues‘.‘category_id‘ AS t0_r6, ‘issues‘.‘status_id‘ AS t0_r7, ‘issues‘.‘assigned_to_id‘ AS t0_r8, ‘issues‘.‘priority_id‘ AS t0_r9, ‘issues‘.‘fixed_version_id‘ AS t0_r10, ‘issues‘.‘author_id‘ AS t0_r11, ‘issues‘.‘lock_version‘ AS t0_r12, ‘issues‘.‘created_on‘ AS t0_r13, ‘issues‘.‘updated_on‘ AS t0_r14, ‘issues‘.‘start_date‘ AS t0_r15, ‘issues‘.‘done_ratio‘ AS t0_r16, ‘issues‘.‘estimated_hours‘ AS t0_r17, ‘issues‘.‘parent_id‘ AS t0_r18, ‘issues‘.‘root_id‘ AS t0_r19, ‘issues‘.‘lft‘ AS t0_r20, ‘issues‘.‘rgt‘ AS t0_r21, ‘issues‘.‘is_private‘ AS t0_r22, ‘issues‘.‘closed_on‘ AS t0_r23, ‘issue_statuses‘.‘id‘ AS t1_r0, ‘issue_statuses‘.‘name‘ AS t1_r1, ‘issue_statuses‘.‘is_closed‘ AS t1_r2, ‘issue_statuses‘.‘position‘ AS t1_r3, ‘issue_statuses‘.‘default_done_ratio‘ AS t1_r4, ‘projects‘.‘id‘ AS t2_r0, ‘projects‘.‘name‘ AS t2_r1, ‘projects‘.‘description‘ AS t2_r2, ‘projects‘.‘homepage‘ AS t2_r3, ‘projects‘.‘is_public‘ AS t2_r4, ‘projects‘.‘parent_id‘ AS t2_r5, ‘projects‘.‘created_on‘ AS t2_r6, ‘projects‘.‘updated_on‘ AS t2_r7, ‘projects‘.‘identifier‘ AS t2_r8, ‘projects‘.‘status‘ AS t2_r9, ‘projects‘.‘lft‘ AS t2_r10, ‘projects‘.‘rgt‘ AS t2_r11, ‘projects‘.‘inherit_members‘ AS t2_r12, ‘projects‘.‘default_version_id‘ AS t2_r13, ‘projects‘.‘default_assigned_to_id‘ AS t2_r14 FROM ‘issues‘ INNER JOIN ‘projects‘ ON ‘projects‘.‘id‘ = ‘issues‘.‘project_id‘ INNER JOIN ‘issue_statuses‘ ON ‘issue_statuses‘.‘id‘ = ‘issues‘.‘status_id‘ WHERE (((projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’issue_tracking’)) AND (((projects.is_public = TRUE AND projects.id NOT IN (SELECT project_id FROM members WHERE user_id IN (6,13))) AND ((issues.is_private = FALSE)))))) AND ((issues.status_id IN (SELECT id FROM issue_statuses WHERE is_closed=FALSE)) AND (issues.subject IN (’d’)) AND projects.lft >= 1 AND projects.rgt <= 10) ORDER BY issues.id DESC LIMIT 500	SELECT ‘issues‘.‘id‘ AS ‘t0_r0‘, ‘issues‘.‘tracker_id‘ AS ‘t0_r1‘, ‘issues‘.‘project_id‘ AS ‘t0_r2‘, ‘issues‘.‘subject‘ AS ‘t0_r3‘, ‘issues‘.‘description‘ AS ‘t0_r4‘, ‘issues‘.‘due_date‘ AS ‘t0_r5‘, ‘issues‘.‘category_id‘ AS ‘t0_r6‘, ‘issues‘.‘status_id‘ AS ‘t0_r7‘, ‘issues‘.‘assigned_to_id‘ AS ‘t0_r8‘, ‘issues‘.‘priority_id‘ AS ‘t0_r9‘, ‘issues‘.‘fixed_version_id‘ AS ‘t0_r10‘, ‘issues‘.‘author_id‘ AS ‘t0_r11‘, ‘issues‘.‘lock_version‘ AS ‘t0_r12‘, ‘issues‘.‘created_on‘ AS ‘t0_r13‘, ‘issues‘.‘updated_on‘ AS ‘t0_r14‘, ‘issues‘.‘start_date‘ AS ‘t0_r15‘, ‘issues‘.‘done_ratio‘ AS ‘t0_r16‘, ‘issues‘.‘estimated_hours‘ AS ‘t0_r17‘, ‘issues‘.‘parent_id‘ AS ‘t0_r18‘, ‘issues‘.‘root_id‘ AS ‘t0_r19‘, ‘issues‘.‘lft‘ AS ‘t0_r20‘, ‘issues‘.‘rgt‘ AS ‘t0_r21‘, ‘issues‘.‘is_private‘ AS ‘t0_r22‘, ‘issues‘.‘closed_on‘ AS ‘t0_r23‘, ‘issue_statuses‘.‘id‘ AS ‘t1_r0‘, ‘issue_statuses‘.‘name‘ AS ‘t1_r1‘, ‘issue_statuses‘.‘is_closed‘ AS ‘t1_r2‘, ‘issue_statuses‘.‘position‘ AS ‘t1_r3‘, ‘issue_statuses‘.‘default_done_ratio‘ AS ‘t1_r4‘, ‘projects‘.‘id‘ AS ‘t2_r0‘, ‘projects‘.‘name‘ AS ‘t2_r1‘, ‘projects‘.‘description‘ AS ‘t2_r2‘, ‘projects‘.‘homepage‘ AS ‘t2_r3‘, ‘projects‘.‘is_public‘ AS ‘t2_r4‘, ‘projects‘.‘parent_id‘ AS ‘t2_r5‘, ‘projects‘.‘created_on‘ AS ‘t2_r6‘, ‘projects‘.‘updated_on‘ AS ‘t2_r7‘, ‘projects‘.‘identifier‘ AS ‘t2_r8‘, ‘projects‘.‘status‘ AS ‘t2_r9‘, ‘projects‘.‘lft‘ AS ‘t2_r10‘, ‘projects‘.‘rgt‘ AS ‘t2_r11‘, ‘projects‘.‘inherit_members‘ AS ‘t2_r12‘, ‘projects‘.‘default_version_id‘ AS ‘t2_r13‘, ‘projects‘.‘default_assigned_to_id‘ AS ‘t2_r14‘ FROM ‘issues‘ AS ‘issues‘ INNER JOIN ‘issue_statuses‘ AS ‘issue_statuses‘ ON ‘issues‘.‘status_id‘ = ‘issue_statuses‘.‘id‘ INNER JOIN ‘projects‘ AS ‘projects‘ ON ‘issues‘.‘project_id‘ = ‘projects‘.‘id‘ WHERE EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘projects‘.‘id‘ AND ‘em‘.‘name‘ = ’issue_tracking’) AND ‘projects‘.‘status‘ <>9 AND ‘projects‘.‘rgt‘ <= 10 AND ‘projects‘.‘lft‘ >= 1 AND ‘projects‘.‘is_public‘ = TRUE AND ‘issues‘.‘status_id‘ IN (SELECT ‘issue_statuses‘.‘id‘ AS ‘id‘ FROM ‘issue_statuses‘ AS ‘issue_statuses‘ WHERE ‘issue_statuses‘.‘is_closed‘ = FALSE) AND ‘issues‘.‘subject‘ IN (?) AND ‘issues‘.‘is_private‘ = FALSE ORDER BY ‘issues‘.‘id‘ DESC LIMIT 500
redmine	517	SELECT ‘users‘.‘id‘ FROM ‘users‘ INNER JOIN ‘watchers‘ ON ‘users‘.‘id‘ = ‘watchers‘.‘user_id‘ WHERE ‘watchers‘.‘watchable_id‘ = 1 AND ‘watchers‘.‘watchable_type‘ = ’Issue’	SELECT ‘watchers‘.‘user_id‘ AS ‘user_id‘ FROM ‘watchers‘ AS ‘watchers‘ WHERE ‘watchers‘.‘watchable_type‘ = ’Issue’ AND ‘watchers‘.‘watchable_id‘ = 1
redmine	524	SELECT COUNT(*) FROM ‘issues‘ INNER JOIN ‘projects‘ ON ‘projects‘.‘id‘ = ‘issues‘.‘project_id‘ INNER JOIN ‘issue_statuses‘ ON ‘issue_statuses‘.‘id‘ = ‘issues‘.‘status_id‘ WHERE (((projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’issue_tracking’)) AND (((projects.is_public = TRUE AND projects.id NOT IN (SELECT project_id FROM members WHERE user_id IN (6,13))) AND ((issues.is_private = FALSE)))))) AND (issues.updated_on >’2014-01-02 19:59:59’)	SELECT COUNT(*) FROM ‘projects‘ AS ‘projects‘ INNER JOIN ‘issues‘ AS ‘issues‘ ON ‘projects‘.‘id‘ = ‘issues‘.‘project_id‘ WHERE NOT ‘projects‘.‘id‘ IN (SELECT ‘project_id‘ FROM ‘members‘ WHERE ‘user_id‘ IN (?)) AND EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘projects‘.‘id‘ AND ‘em‘.‘name‘ = ’issue_tracking’) AND ‘projects‘.‘status‘ <>9 AND ‘projects‘.‘is_public‘ = TRUE AND ‘issues‘.‘updated_on‘ >’2014-01-02 19:59:59’ AND ‘issues‘.‘is_private‘ = FALSE
redmine	540	SELECT ‘issues‘.‘id‘ AS t0_r0, ‘issues‘.‘tracker_id‘ AS t0_r1, ‘issues‘.‘project_id‘ AS t0_r2, ‘issues‘.‘subject‘ AS t0_r3, ‘issues‘.‘description‘ AS t0_r4, ‘issues‘.‘due_date‘ AS t0_r5, ‘issues‘.‘category_id‘ AS t0_r6, ‘issues‘.‘status_id‘ AS t0_r7, ‘issues‘.‘assigned_to_id‘ AS t0_r8, ‘issues‘.‘priority_id‘ AS t0_r9, ‘issues‘.‘fixed_version_id‘ AS t0_r10, ‘issues‘.‘author_id‘ AS t0_r11, ‘issues‘.‘lock_version‘ AS t0_r12, ‘issues‘.‘created_on‘ AS t0_r13, ‘issues‘.‘updated_on‘ AS t0_r14, ‘issues‘.‘start_date‘ AS t0_r15, ‘issues‘.‘done_ratio‘ AS t0_r16, ‘issues‘.‘estimated_hours‘ AS t0_r17, ‘issues‘.‘parent_id‘ AS t0_r18, ‘issues‘.‘root_id‘ AS t0_r19, ‘issues‘.‘lft‘ AS t0_r20, ‘issues‘.‘rgt‘ AS t0_r21, ‘issues‘.‘is_private‘ AS t0_r22, ‘issues‘.‘closed_on‘ AS t0_r23, ‘issue_statuses‘.‘id‘ AS t1_r0, ‘issue_statuses‘.‘name‘ AS t1_r1, ‘issue_statuses‘.‘is_closed‘ AS t1_r2, ‘issue_statuses‘.‘position‘ AS t1_r3, ‘issue_statuses‘.‘default_done_ratio‘ AS t1_r4, ‘projects‘.‘id‘ AS t2_r0, ‘projects‘.‘name‘ AS t2_r1, ‘projects‘.‘description‘ AS t2_r2, ‘projects‘.‘homepage‘ AS t2_r3, ‘projects‘.‘is_public‘ AS t2_r4, ‘projects‘.‘parent_id‘ AS t2_r5, ‘projects‘.‘created_on‘ AS t2_r6, ‘projects‘.‘updated_on‘ AS t2_r7, ‘projects‘.‘identifier‘ AS t2_r8, ‘projects‘.‘status‘ AS t2_r9, ‘projects‘.‘lft‘ AS t2_r10, ‘projects‘.‘rgt‘ AS t2_r11, ‘projects‘.‘inherit_members‘ AS t2_r12, ‘projects‘.‘default_version_id‘ AS t2_r13, ‘projects‘.‘default_assigned_to_id‘ AS t2_r14 FROM ‘issues‘ INNER JOIN ‘projects‘ ON ‘projects‘.‘id‘ = ‘issues‘.‘project_id‘ INNER JOIN ‘issue_statuses‘ ON ‘issue_statuses‘.‘id‘ = ‘issues‘.‘status_id‘ WHERE (((projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’issue_tracking’)) AND (((projects.is_public = TRUE AND projects.id NOT IN (SELECT project_id FROM members WHERE user_id IN (6,13))) AND ((issues.is_private = FALSE)))))) AND ((issues.status_id IN (SELECT id FROM issue_statuses WHERE is_closed=FALSE)) AND issues.id IN (SELECT issues.id FROM issues LEFT OUTER JOIN custom_values ON custom_values.customized_type=’Issue’ AND custom_values.customized_id=issues.id AND custom_values.custom_field_id=1 WHERE (custom_values.value IN (’MySQL’)) AND (((1=1) AND (issues.tracker_id IN (SELECT tracker_id FROM custom_fields_trackers WHERE custom_field_id = 1)) AND (EXISTS (SELECT 1 FROM custom_fields ifa WHERE ifa.is_for_all = TRUE AND ifa.id = 1) OR issues.project_id IN (SELECT project_id FROM custom_fields_projects WHERE custom_field_id = 1)))))) ORDER BY issues.id DESC LIMIT 25 OFFSET 0	SELECT * FROM ‘issues‘ AS ‘issues‘ INNER JOIN ‘issue_statuses‘ AS ‘issue_statuses‘ ON ‘issues‘.‘status_id‘ = ‘issue_statuses‘.‘id‘ INNER JOIN ‘projects‘ AS ‘projects‘ ON ‘issues‘.‘project_id‘ = ‘projects‘.‘id‘ WHERE NOT ‘projects‘.‘id‘ IN (SELECT ‘project_id‘ FROM ‘members‘ WHERE ‘user_id‘ IN (?)) AND ‘issues‘.‘status_id‘ IN (SELECT ‘issue_statuses‘.‘id‘ AS ‘id‘ FROM ‘issue_statuses‘ AS ‘issue_statuses‘ WHERE ‘issue_statuses‘.‘is_closed‘ = FALSE) AND EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘issues‘.‘project_id‘ AND ‘em‘.‘name‘ = ’issue_tracking’) AND ‘projects‘.‘status‘ <>9 AND ‘projects‘.‘is_public‘ = TRUE AND ‘issues‘.‘is_private‘ = FALSE AND ‘issues‘.‘id‘ IN (SELECT ‘issues‘.‘id‘ AS ‘id‘ FROM ‘issues‘ AS ‘issues‘ LEFT JOIN ‘custom_values‘ AS ‘custom_values‘ ON ‘custom_values‘.‘customized_type‘ = ’Issue’ AND ‘custom_values‘.‘customized_id‘ = ‘issues‘.‘id‘ AND ‘custom_values‘.‘custom_field_id‘ = 1 WHERE ‘issues‘.‘tracker_id‘ IN (SELECT ‘custom_fields_trackers‘.‘tracker_id‘ AS ‘tracker_id‘ FROM ‘custom_fields_trackers‘ AS ‘custom_fields_trackers‘ WHERE ‘custom_fields_trackers‘.‘custom_field_id‘ = 1) AND (EXISTS (SELECT 1 FROM ‘custom_fields‘ AS ‘ifa‘ WHERE ‘ifa‘.‘is_for_all‘ = TRUE AND ‘ifa‘.‘id‘ = 1) OR ‘issues‘.‘project_id‘ IN (SELECT ‘project_id‘ FROM ‘custom_fields_projects‘ WHERE ‘custom_field_id‘ = 1)) AND ‘custom_values‘.‘value‘ IN (?)) ORDER BY ‘issues‘.‘id‘ DESC LIMIT 25 OFFSET 0

redmine	548	SELECT ‘issues‘.‘id‘ AS t0_r0, ‘issues‘.‘tracker_id‘ AS t0_r1, ‘issues‘.‘project_id‘ AS t0_r2, ‘issues‘.‘subject‘ AS t0_r3, ‘issues‘.‘description‘ AS t0_r4, ‘issues‘.‘due_date‘ AS t0_r5, ‘issues‘.‘category_id‘ AS t0_r6, ‘issues‘.‘status_id‘ AS t0_r7, ‘issues‘.‘assigned_to_id‘ AS t0_r8, ‘issues‘.‘priority_id‘ AS t0_r9, ‘issues‘.‘fixed_version_id‘ AS t0_r10, ‘issues‘.‘author_id‘ AS t0_r11, ‘issues‘.‘lock_version‘ AS t0_r12, ‘issues‘.‘created_on‘ AS t0_r13, ‘issues‘.‘updated_on‘ AS t0_r14, ‘issues‘.‘start_date‘ AS t0_r15, ‘issues‘.‘done_ratio‘ AS t0_r16, ‘issues‘.‘estimated_hours‘ AS t0_r17, ‘issues‘.‘parent_id‘ AS t0_r18, ‘issues‘.‘root_id‘ AS t0_r19, ‘issues‘.‘lft‘ AS t0_r20, ‘issues‘.‘rgt‘ AS t0_r21, ‘issues‘.‘is_private‘ AS t0_r22, ‘issues‘.‘closed_on‘ AS t0_r23, ‘issue_statuses‘.‘id‘ AS t1_r0, ‘issue_statuses‘.‘name‘ AS t1_r1, ‘issue_statuses‘.‘is_closed‘ AS t1_r2, ‘issue_statuses‘.‘position‘ AS t1_r3, ‘issue_statuses‘.‘default_done_ratio‘ AS t1_r4, ‘projects‘.‘id‘ AS t2_r0, ‘projects‘.‘name‘ AS t2_r1, ‘projects‘.‘description‘ AS t2_r2, ‘projects‘.‘homepage‘ AS t2_r3, ‘projects‘.‘is_public‘ AS t2_r4, ‘projects‘.‘parent_id‘ AS t2_r5, ‘projects‘.‘created_on‘ AS t2_r6, ‘projects‘.‘updated_on‘ AS t2_r7, ‘projects‘.‘identifier‘ AS t2_r8, ‘projects‘.‘status‘ AS t2_r9, ‘projects‘.‘lft‘ AS t2_r10, ‘projects‘.‘rgt‘ AS t2_r11, ‘projects‘.‘inherit_members‘ AS t2_r12, ‘projects‘.‘default_version_id‘ AS t2_r13, ‘projects‘.‘default_assigned_to_id‘ AS t2_r14 FROM ‘issues‘ INNER JOIN ‘projects‘ ON ‘projects‘.‘id‘ = ‘issues‘.‘project_id‘ INNER JOIN ‘issue_statuses‘ ON ‘issue_statuses‘.‘id‘ = ‘issues‘.‘status_id‘ WHERE (((projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’issue_tracking’)) AND (((projects.is_public = TRUE AND projects.id NOT IN (SELECT project_id FROM members WHERE user_id IN (3,12))) AND ((issues.is_private = FALSE OR issues.author_id = 3 OR issues.assigned_to_id IN (3)))) OR (projects.id IN (1) AND ((issues.is_private = FALSE OR issues.author_id = 3 OR issues.assigned_to_id IN (3))))))) AND ((issues.status_id IN (SELECT id FROM issue_statuses WHERE is_closed=FALSE))) ORDER BY issues.id DESC LIMIT 500	SELECT ‘issues‘.‘id‘ AS ‘t0_r0‘, ‘issues‘.‘tracker_id‘ AS ‘t0_r1‘, ‘issues‘.‘project_id‘ AS ‘t0_r2‘, ‘issues‘.‘subject‘ AS ‘t0_r3‘, ‘issues‘.‘description‘ AS ‘t0_r4‘, ‘issues‘.‘due_date‘ AS ‘t0_r5‘, ‘issues‘.‘category_id‘ AS ‘t0_r6‘, ‘issues‘.‘status_id‘ AS ‘t0_r7‘, ‘issues‘.‘assigned_to_id‘ AS ‘t0_r8‘, ‘issues‘.‘priority_id‘ AS ‘t0_r9‘, ‘issues‘.‘fixed_version_id‘ AS ‘t0_r10‘, ‘issues‘.‘author_id‘ AS ‘t0_r11‘, ‘issues‘.‘lock_version‘ AS ‘t0_r12‘, ‘issues‘.‘created_on‘ AS ‘t0_r13‘, ‘issues‘.‘updated_on‘ AS ‘t0_r14‘, ‘issues‘.‘start_date‘ AS ‘t0_r15‘, ‘issues‘.‘done_ratio‘ AS ‘t0_r16‘, ‘issues‘.‘estimated_hours‘ AS ‘t0_r17‘, ‘issues‘.‘parent_id‘ AS ‘t0_r18‘, ‘issues‘.‘root_id‘ AS ‘t0_r19‘, ‘issues‘.‘lft‘ AS ‘t0_r20‘, ‘issues‘.‘rgt‘ AS ‘t0_r21‘, ‘issues‘.‘is_private‘ AS ‘t0_r22‘, ‘issues‘.‘closed_on‘ AS ‘t0_r23‘, ‘issue_statuses‘.‘id‘ AS ‘t1_r0‘, ‘issue_statuses‘.‘name‘ AS ‘t1_r1‘, ‘issue_statuses‘.‘is_closed‘ AS ‘t1_r2‘, ‘issue_statuses‘.‘position‘ AS ‘t1_r3‘, ‘issue_statuses‘.‘default_done_ratio‘ AS ‘t1_r4‘, ‘projects‘.‘id‘ AS ‘t2_r0‘, ‘projects‘.‘name‘ AS ‘t2_r1‘, ‘projects‘.‘description‘ AS ‘t2_r2‘, ‘projects‘.‘homepage‘ AS ‘t2_r3‘, ‘projects‘.‘is_public‘ AS ‘t2_r4‘, ‘projects‘.‘parent_id‘ AS ‘t2_r5‘, ‘projects‘.‘created_on‘ AS ‘t2_r6‘, ‘projects‘.‘updated_on‘ AS ‘t2_r7‘, ‘projects‘.‘identifier‘ AS ‘t2_r8‘, ‘projects‘.‘status‘ AS ‘t2_r9‘, ‘projects‘.‘lft‘ AS ‘t2_r10‘, ‘projects‘.‘rgt‘ AS ‘t2_r11‘, ‘projects‘.‘inherit_members‘ AS ‘t2_r12‘, ‘projects‘.‘default_version_id‘ AS ‘t2_r13‘, ‘projects‘.‘default_assigned_to_id‘ AS ‘t2_r14‘ FROM ‘issues‘ AS ‘issues‘ INNER JOIN ‘issue_statuses‘ AS ‘issue_statuses‘ ON ‘issues‘.‘status_id‘ = ‘issue_statuses‘.‘id‘ INNER JOIN ‘projects‘ AS ‘projects‘ ON ‘issues‘.‘project_id‘ = ‘projects‘.‘id‘ WHERE (‘projects‘.‘is_public‘ = TRUE AND NOT ‘projects‘.‘id‘ IN (SELECT ‘project_id‘ FROM ‘members‘ WHERE ‘user_id‘ IN (?)) AND (‘issues‘.‘is_private‘ = FALSE OR ‘issues‘.‘author_id‘ = 3 OR ‘issues‘.‘assigned_to_id‘ IN (?)) OR ‘projects‘.‘id‘ IN (?) AND (‘issues‘.‘is_private‘ = FALSE OR ‘issues‘.‘author_id‘ = 3 OR ‘issues‘.‘assigned_to_id‘ IN (?))) AND EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘issues‘.‘project_id‘ AND ‘em‘.‘name‘ = ’issue_tracking’) AND ‘issues‘.‘status_id‘ IN (SELECT ‘issue_statuses‘.‘id‘ AS ‘id‘ FROM ‘issue_statuses‘ AS ‘issue_statuses‘ WHERE ‘issue_statuses‘.‘is_closed‘ = FALSE) AND ‘projects‘.‘status‘ <>9 ORDER BY ‘issues‘.‘id‘ DESC LIMIT 500

redmine	599	SELECT COUNT(*) AS count_all, ‘issues‘.‘category_id‘ AS issues_category_id FROM ‘issues‘ INNER JOIN ‘projects‘ ON ‘projects‘.‘id‘ = ‘issues‘.‘project_id‘ INNER JOIN ‘issue_statuses‘ ON ‘issue_statuses‘.‘id‘ = ‘issues‘.‘status_id‘ WHERE (((projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’issue_tracking’)) AND (((projects.is_public = TRUE AND projects.id NOT IN (SELECT project_id FROM members WHERE user_id IN (6,13))) AND ((issues.is_private = FALSE)))))) AND ((issues.status_id IN (SELECT id FROM issue_statuses WHERE is_closed=FALSE))) GROUP BY ‘issues‘.‘category_id‘	SELECT COUNT(*) AS ‘count_all‘, ‘issues‘.‘category_id‘ AS ‘issues_category_id‘ FROM ‘issues‘ AS ‘issues‘ INNER JOIN ‘projects‘ AS ‘projects‘ ON ‘issues‘.‘project_id‘ = ‘projects‘.‘id‘ WHERE EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘projects‘.‘id‘ AND ‘em‘.‘name‘ = ’issue_tracking’) AND NOT ‘issues‘.‘project_id‘ IN (SELECT ‘project_id‘ FROM ‘members‘ WHERE ‘user_id‘ IN (?)) AND ‘issues‘.‘status_id‘ IN (SELECT ‘issue_statuses‘.‘id‘ AS ‘id‘ FROM ‘issue_statuses‘ AS ‘issue_statuses‘ WHERE ‘issue_statuses‘.‘is_closed‘ = FALSE) AND ‘projects‘.‘status‘ <>9 AND ‘projects‘.‘is_public‘ = TRUE AND ‘issues‘.‘is_private‘ = FALSE GROUP BY ‘issues‘.‘category_id‘
redmine	612	SELECT SUM(time_entries.hours) FROM ‘issues‘ INNER JOIN ‘projects‘ ON ‘projects‘.‘id‘ = ‘issues‘.‘project_id‘ INNER JOIN ‘issue_statuses‘ ON ‘issue_statuses‘.‘id‘ = ‘issues‘.‘status_id‘ INNER JOIN ‘time_entries‘ ON ‘time_entries‘.‘issue_id‘ = ‘issues‘.‘id‘ WHERE (((projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’issue_tracking’)) AND (((projects.is_public = TRUE AND projects.id NOT IN (SELECT project_id FROM members WHERE user_id IN (6,13))) AND ((issues.is_private = FALSE)))))) AND ((issues.status_id IN (SELECT id FROM issue_statuses WHERE is_closed=FALSE))) AND (((projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’time_tracking’)) AND ((projects.is_public = TRUE AND projects.id NOT IN (SELECT project_id FROM members WHERE user_id IN (6,13))))))	SELECT SUM(‘time_entries‘.‘hours‘) FROM ‘issues‘ AS ‘issues‘ INNER JOIN ‘time_entries‘ AS ‘time_entries‘ ON ‘issues‘.‘id‘ = ‘time_entries‘.‘issue_id‘ INNER JOIN ‘projects‘ AS ‘projects‘ ON ‘issues‘.‘project_id‘ = ‘projects‘.‘id‘ WHERE NOT ‘projects‘.‘id‘ IN (SELECT ‘project_id‘ FROM ‘members‘ WHERE ‘user_id‘ IN (?)) AND EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘issues‘.‘project_id‘ AND ‘em‘.‘name‘ = ’time_tracking’) AND EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘issues‘.‘project_id‘ AND ‘em‘.‘name‘ = ’issue_tracking’) AND ‘issues‘.‘status_id‘ IN (SELECT ‘issue_statuses‘.‘id‘ AS ‘id‘ FROM ‘issue_statuses‘ AS ‘issue_statuses‘ WHERE ‘issue_statuses‘.‘is_closed‘ = FALSE) AND ‘issues‘.‘is_private‘ = FALSE AND ‘projects‘.‘is_public‘ = TRUE AND ‘projects‘.‘status‘ <>9
redmine	619	SELECT ‘queries‘.* FROM ‘queries‘ LEFT OUTER JOIN projects ON queries.project_id = projects.id WHERE ‘queries‘.‘type‘ IN (’IssueQuery’) AND (queries.project_id IS NULL OR (projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’issue_tracking’))) AND (queries.visibility <>0 OR queries.user_id = 1) AND ‘queries‘.‘project_id‘ IS NULL ORDER BY ‘queries‘.‘name‘ ASC, ‘queries‘.‘id‘ ASC	SELECT ‘queries‘.‘id‘ AS ‘id‘, ‘queries‘.‘project_id‘ AS ‘project_id‘, ‘queries‘.‘name‘ AS ‘name‘, ‘queries‘.‘filters‘ AS ‘filters‘, ‘queries‘.‘user_id‘ AS ‘user_id‘, ‘queries‘.‘column_names‘ AS ‘column_names‘, ‘queries‘.‘sort_criteria‘ AS ‘sort_criteria‘, ‘queries‘.‘group_by‘ AS ‘group_by‘, ‘queries‘.‘type‘ AS ‘type‘, ‘queries‘.‘visibility‘ AS ‘visibility‘, ‘queries‘.‘options‘ AS ‘options‘ FROM ‘queries‘ AS ‘queries‘ INNER JOIN ‘projects‘ AS ‘projects‘ ON ‘queries‘.‘project_id‘ = ‘projects‘.‘id‘ WHERE ‘projects‘.‘id‘ IS NULL AND (‘queries‘.‘project_id‘ IS NULL OR ‘projects‘.‘status‘ <>9 AND EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘projects‘.‘id‘ AND ‘em‘.‘name‘ = ’issue_tracking’)) AND (‘queries‘.‘visibility‘ <>0 OR ‘queries‘.‘user_id‘ = 1) AND ‘queries‘.‘type‘ IN (?) ORDER BY ‘queries‘.‘name‘ ASC, ‘queries‘.‘id‘ ASC
redmine	620	SELECT COUNT(*) FROM ‘issues‘ INNER JOIN ‘projects‘ ON ‘projects‘.‘id‘ = ‘issues‘.‘project_id‘ INNER JOIN ‘issue_statuses‘ ON ‘issue_statuses‘.‘id‘ = ‘issues‘.‘status_id‘ WHERE (((projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’issue_tracking’)) AND (((projects.is_public = TRUE AND projects.id NOT IN (SELECT project_id FROM members WHERE user_id IN (3,12))) AND ((issues.is_private = FALSE OR issues.author_id = 3 OR issues.assigned_to_id IN (3)))) OR (projects.id IN (1) AND ((issues.is_private = FALSE OR issues.author_id = 3 OR issues.assigned_to_id IN (3))))))) AND ((issues.status_id IN (SELECT id FROM issue_statuses WHERE is_closed=FALSE)))	SELECT COUNT(*) FROM ‘projects‘ AS ‘projects‘ INNER JOIN ‘issues‘ AS ‘issues‘ ON ‘projects‘.‘id‘ = ‘issues‘.‘project_id‘ WHERE EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘issues‘.‘project_id‘ AND ‘em‘.‘name‘ = ’issue_tracking’) AND ‘issues‘.‘status_id‘ IN (SELECT ‘issue_statuses‘.‘id‘ AS ‘id‘ FROM ‘issue_statuses‘ AS ‘issue_statuses‘ WHERE ‘issue_statuses‘.‘is_closed‘ = FALSE) AND ‘projects‘.‘status‘ <>9 AND (‘projects‘.‘is_public‘ = TRUE AND NOT ‘projects‘.‘id‘ IN (SELECT ‘project_id‘ FROM ‘members‘ WHERE ‘user_id‘ IN (?)) AND (‘issues‘.‘is_private‘ = FALSE OR ‘issues‘.‘author_id‘ = 3 OR ‘issues‘.‘assigned_to_id‘ IN (?)) OR ‘projects‘.‘id‘ IN (?) AND (‘issues‘.‘is_private‘ = FALSE OR ‘issues‘.‘author_id‘ = 3 OR ‘issues‘.‘assigned_to_id‘ IN (?)))
redmine	623	SELECT COUNT(*) FROM ‘issues‘ INNER JOIN ‘projects‘ ON ‘projects‘.‘id‘ = ‘issues‘.‘project_id‘ INNER JOIN ‘issue_statuses‘ ON ‘issue_statuses‘.‘id‘ = ‘issues‘.‘status_id‘ WHERE (((projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’issue_tracking’)) AND (((projects.is_public = TRUE AND projects.id NOT IN (SELECT project_id FROM members WHERE user_id IN (898,12))) AND ((issues.is_private = FALSE OR issues.author_id = 898 OR issues.assigned_to_id IN (898))))))) AND ((issues.status_id IN (SELECT id FROM issue_statuses WHERE is_closed=FALSE)))	SELECT COUNT(*) FROM ‘issues‘ AS ‘issues‘ INNER JOIN ‘projects‘ AS ‘projects‘ ON ‘issues‘.‘project_id‘ = ‘projects‘.‘id‘ WHERE EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘issues‘.‘project_id‘ AND ‘em‘.‘name‘ = ’issue_tracking’) AND NOT ‘projects‘.‘id‘ IN (SELECT ‘project_id‘ FROM ‘members‘ WHERE ‘user_id‘ IN (?)) AND ‘issues‘.‘status_id‘ IN (SELECT ‘issue_statuses‘.‘id‘ AS ‘id‘ FROM ‘issue_statuses‘ AS ‘issue_statuses‘ WHERE ‘issue_statuses‘.‘is_closed‘ = FALSE) AND ‘projects‘.‘status‘ <>9 AND ‘projects‘.‘is_public‘ = TRUE AND (‘issues‘.‘is_private‘ = FALSE OR ‘issues‘.‘author_id‘ = 898 OR ‘issues‘.‘assigned_to_id‘ IN (?))

redmine	625	SELECT ‘queries‘.* FROM ‘queries‘ LEFT OUTER JOIN projects ON queries.project_id = projects.id WHERE ‘queries‘.‘type‘ IN (’IssueQuery’) AND (queries.project_id IS NULL OR (((projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’issue_tracking’)) AND ((projects.is_public = TRUE AND projects.id NOT IN (SELECT project_id FROM members WHERE user_id IN (898,12))))))) AND (queries.visibility = 2 OR queries.user_id = 898) AND ‘queries‘.‘project_id‘ IS NULL ORDER BY ‘queries‘.‘name‘ ASC, ‘queries‘.‘id‘ ASC	SELECT ‘queries‘.‘id‘ AS ‘id‘, ‘queries‘.‘project_id‘ AS ‘project_id‘, ‘queries‘.‘name‘ AS ‘name‘, ‘queries‘.‘filters‘ AS ‘filters‘, ‘queries‘.‘user_id‘ AS ‘user_id‘, ‘queries‘.‘column_names‘ AS ‘column_names‘, ‘queries‘.‘sort_criteria‘ AS ‘sort_criteria‘, ‘queries‘.‘group_by‘ AS ‘group_by‘, ‘queries‘.‘type‘ AS ‘type‘, ‘queries‘.‘visibility‘ AS ‘visibility‘, ‘queries‘.‘options‘ AS ‘options‘ FROM ‘queries‘ AS ‘queries‘ INNER JOIN ‘projects‘ AS ‘projects‘ ON ‘queries‘.‘project_id‘ = ‘projects‘.‘id‘ WHERE ‘queries‘.‘project_id‘ IS NULL AND (‘queries‘.‘visibility‘ = 2 OR ‘queries‘.‘user_id‘ = 898) AND ‘queries‘.‘type‘ IN (?) AND (‘queries‘.‘project_id‘ IS NULL OR ‘projects‘.‘status‘ <>9 AND EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘projects‘.‘id‘ AND ‘em‘.‘name‘ = ’issue_tracking’) AND (‘projects‘.‘is_public‘ = TRUE AND NOT ‘projects‘.‘id‘ IN (SELECT ‘project_id‘ FROM ‘members‘ WHERE ‘user_id‘ IN (?)))) ORDER BY ‘queries‘.‘name‘ ASC, ‘queries‘.‘id‘ ASC
redmine	629	SELECT COUNT(*) FROM ‘issues‘ INNER JOIN ‘projects‘ ON ‘projects‘.‘id‘ = ‘issues‘.‘project_id‘ INNER JOIN ‘issue_statuses‘ ON ‘issue_statuses‘.‘id‘ = ‘issues‘.‘status_id‘ WHERE (projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’issue_tracking’)) AND (issues.project_id IN (SELECT projects.id FROM projects LEFT OUTER JOIN custom_values ON custom_values.customized_type=’Project’ AND custom_values.customized_id=projects.id AND custom_values.custom_field_id=1279 WHERE (custom_values.value IN (’Foo’)) AND (1=1)))	SELECT COUNT(*) FROM ‘issues‘ AS ‘issues‘ INNER JOIN ‘projects‘ AS ‘projects‘ ON ‘issues‘.‘project_id‘ = ‘projects‘.‘id‘ WHERE ‘projects‘.‘status‘ <>9 AND ‘issues‘.‘project_id‘ IN (SELECT ‘projects_0‘.‘id‘ AS ‘id‘ FROM ‘projects‘ AS ‘projects_0‘ INNER JOIN ‘custom_values‘ AS ‘custom_values‘ ON ‘custom_values‘.‘customized_type‘ = ’Project’ AND ‘custom_values‘.‘customized_id‘ = ‘projects_0‘.‘id‘ AND ‘custom_values‘.‘custom_field_id‘ = 1279 WHERE ‘custom_values‘.‘value‘ IN (?) AND EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘projects_0‘.‘id‘ AND ‘em‘.‘name‘ = ’issue_tracking’))
redmine	633	SELECT ‘issues‘.‘id‘ AS t0_r0, ‘issues‘.‘tracker_id‘ AS t0_r1, ‘issues‘.‘project_id‘ AS t0_r2, ‘issues‘.‘subject‘ AS t0_r3, ‘issues‘.‘description‘ AS t0_r4, ‘issues‘.‘due_date‘ AS t0_r5, ‘issues‘.‘category_id‘ AS t0_r6, ‘issues‘.‘status_id‘ AS t0_r7, ‘issues‘.‘assigned_to_id‘ AS t0_r8, ‘issues‘.‘priority_id‘ AS t0_r9, ‘issues‘.‘fixed_version_id‘ AS t0_r10, ‘issues‘.‘author_id‘ AS t0_r11, ‘issues‘.‘lock_version‘ AS t0_r12, ‘issues‘.‘created_on‘ AS t0_r13, ‘issues‘.‘updated_on‘ AS t0_r14, ‘issues‘.‘start_date‘ AS t0_r15, ‘issues‘.‘done_ratio‘ AS t0_r16, ‘issues‘.‘estimated_hours‘ AS t0_r17, ‘issues‘.‘parent_id‘ AS t0_r18, ‘issues‘.‘root_id‘ AS t0_r19, ‘issues‘.‘lft‘ AS t0_r20, ‘issues‘.‘rgt‘ AS t0_r21, ‘issues‘.‘is_private‘ AS t0_r22, ‘issues‘.‘closed_on‘ AS t0_r23, ‘issue_statuses‘.‘id‘ AS t1_r0, ‘issue_statuses‘.‘name‘ AS t1_r1, ‘issue_statuses‘.‘is_closed‘ AS t1_r2, ‘issue_statuses‘.‘position‘ AS t1_r3, ‘issue_statuses‘.‘default_done_ratio‘ AS t1_r4, ‘trackers‘.‘id‘ AS t2_r0, ‘trackers‘.‘name‘ AS t2_r1, ‘trackers‘.‘description‘ AS t2_r2, ‘trackers‘.‘is_in_chlog‘ AS t2_r3, ‘trackers‘.‘position‘ AS t2_r4, ‘trackers‘.‘is_in_roadmap‘ AS t2_r5, ‘trackers‘.‘fields_bits‘ AS t2_r6, ‘trackers‘.‘default_status_id‘ AS t2_r7, ‘enumerations‘.‘id‘ AS t3_r0, ‘enumerations‘.‘name‘ AS t3_r1, ‘enumerations‘.‘position‘ AS t3_r2, ‘enumerations‘.‘is_default‘ AS t3_r3, ‘enumerations‘.‘type‘ AS t3_r4, ‘enumerations‘.‘active‘ AS t3_r5, ‘enumerations‘.‘project_id‘ AS t3_r6, ‘enumerations‘.‘parent_id‘ AS t3_r7, ‘enumerations‘.‘position_name‘ AS t3_r8 FROM ‘issues‘ INNER JOIN ‘projects‘ ON ‘projects‘.‘id‘ = ‘issues‘.‘project_id‘ LEFT OUTER JOIN ‘issue_statuses‘ ON ‘issue_statuses‘.‘id‘ = ‘issues‘.‘status_id‘ LEFT OUTER JOIN ‘trackers‘ ON ‘trackers‘.‘id‘ = ‘issues‘.‘tracker_id‘ LEFT OUTER JOIN ‘enumerations‘ ON ‘enumerations‘.‘id‘ = ‘issues‘.‘priority_id‘ AND ‘enumerations‘.‘type‘ IN (’IssuePriority’) WHERE ‘issues‘.‘fixed_version_id‘ = 2 AND (((projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’issue_tracking’)) AND (((projects.is_public = TRUE AND projects.id NOT IN (SELECT project_id FROM members WHERE user_id IN (6,13))) AND ((issues.is_private = FALSE)))))) ORDER BY trackers.position, issues.id	SELECT * FROM (SELECT ‘issues‘.‘id‘ AS ‘t0_r0‘, ‘issues‘.‘tracker_id‘ AS ‘t0_r1‘, ‘issues‘.‘project_id‘ AS ‘t0_r2‘, ‘issues‘.‘subject‘ AS ‘t0_r3‘, ‘issues‘.‘description‘ AS ‘t0_r4‘, ‘issues‘.‘due_date‘ AS ‘t0_r5‘, ‘issues‘.‘category_id‘ AS ‘t0_r6‘, ‘issues‘.‘status_id‘ AS ‘t0_r7‘, ‘issues‘.‘assigned_to_id‘ AS ‘t0_r8‘, ‘issues‘.‘priority_id‘ AS ‘t0_r9‘, ‘issues‘.‘fixed_version_id‘ AS ‘t0_r10‘, ‘issues‘.‘author_id‘ AS ‘t0_r11‘, ‘issues‘.‘lock_version‘ AS ‘t0_r12‘, ‘issues‘.‘created_on‘ AS ‘t0_r13‘, ‘issues‘.‘updated_on‘ AS ‘t0_r14‘, ‘issues‘.‘start_date‘ AS ‘t0_r15‘, ‘issues‘.‘done_ratio‘ AS ‘t0_r16‘, ‘issues‘.‘estimated_hours‘ AS ‘t0_r17‘, ‘issues‘.‘parent_id‘ AS ‘t0_r18‘, ‘issues‘.‘root_id‘ AS ‘t0_r19‘, ‘issues‘.‘lft‘ AS ‘t0_r20‘, ‘issues‘.‘rgt‘ AS ‘t0_r21‘, ‘issues‘.‘is_private‘ AS ‘t0_r22‘, ‘issues‘.‘closed_on‘ AS ‘t0_r23‘, ‘issue_statuses‘.‘id‘ AS ‘t1_r0‘, ‘issue_statuses‘.‘name‘ AS ‘t1_r1‘, ‘issue_statuses‘.‘is_closed‘ AS ‘t1_r2‘, ‘issue_statuses‘.‘position‘ AS ‘t1_r3‘, ‘issue_statuses‘.‘default_done_ratio‘ AS ‘t1_r4‘, ‘trackers‘.‘id‘ AS ‘t2_r0‘, ‘trackers‘.‘name‘ AS ‘t2_r1‘, ‘trackers‘.‘description‘ AS ‘t2_r2‘, ‘trackers‘.‘is_in_chlog‘ AS ‘t2_r3‘, ‘trackers‘.‘position‘ AS ‘t2_r4‘, ‘trackers‘.‘is_in_roadmap‘ AS ‘t2_r5‘, ‘trackers‘.‘fields_bits‘ AS ‘t2_r6‘, ‘trackers‘.‘default_status_id‘ AS ‘t2_r7‘, ‘enumerations‘.‘id‘ AS ‘t3_r0‘, ‘enumerations‘.‘name‘ AS ‘t3_r1‘, ‘enumerations‘.‘position‘ AS ‘t3_r2‘, ‘enumerations‘.‘is_default‘ AS ‘t3_r3‘, ‘enumerations‘.‘type‘ AS ‘t3_r4‘, ‘enumerations‘.‘active‘ AS ‘t3_r5‘, ‘enumerations‘.‘project_id‘ AS ‘t3_r6‘, ‘enumerations‘.‘parent_id‘ AS ‘t3_r7‘, ‘enumerations‘.‘position_name‘ AS ‘t3_r8‘ FROM ‘issues‘ AS ‘issues‘ LEFT JOIN ‘enumerations‘ AS ‘enumerations‘ ON ‘enumerations‘.‘id‘ = ‘issues‘.‘priority_id‘ AND ‘enumerations‘.‘type‘ IN (?) INNER JOIN ‘trackers‘ AS ‘trackers‘ ON ‘issues‘.‘tracker_id‘ = ‘trackers‘.‘id‘ INNER JOIN ‘issue_statuses‘ AS ‘issue_statuses‘ ON ‘issues‘.‘status_id‘ = ‘issue_statuses‘.‘id‘ INNER JOIN ‘projects‘ AS ‘projects‘ ON ‘issues‘.‘project_id‘ = ‘projects‘.‘id‘ WHERE EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘projects‘.‘id‘ AND ‘em‘.‘name‘ = ’issue_tracking’) AND NOT ‘issues‘.‘project_id‘ IN (SELECT ‘project_id‘ FROM ‘members‘ WHERE ‘user_id‘ IN (?)) AND ‘projects‘.‘status‘ <>9 AND ‘projects‘.‘is_public‘ = TRUE) AS ‘sub_0‘ WHERE ‘sub_0‘.‘t0_r22‘ = FALSE AND ‘sub_0‘.‘t0_r10‘ = 2 ORDER BY ‘sub_0‘.‘t2_r4‘, ‘sub_0‘.‘t0_r0‘

redmine	643	SELECT COUNT(*) FROM ‘issues‘ INNER JOIN ‘projects‘ ON ‘projects‘.‘id‘ = ‘issues‘.‘project_id‘ INNER JOIN ‘issue_statuses‘ ON ‘issue_statuses‘.‘id‘ = ‘issues‘.‘status_id‘ WHERE (((projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’issue_tracking’)) AND (((projects.is_public = TRUE AND projects.id NOT IN (SELECT project_id FROM members WHERE user_id IN (6,13))) AND ((issues.is_private = FALSE)))))) AND (projects.lft >= 1 AND projects.rgt <= 10)	SELECT COUNT(*) FROM ‘issues‘ AS ‘issues‘ INNER JOIN ‘projects‘ AS ‘projects‘ ON ‘issues‘.‘project_id‘ = ‘projects‘.‘id‘ WHERE NOT ‘issues‘.‘project_id‘ IN (SELECT ‘project_id‘ FROM ‘members‘ WHERE ‘user_id‘ IN (?)) AND EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘issues‘.‘project_id‘ AND ‘em‘.‘name‘ = ’issue_tracking’) AND ‘projects‘.‘status‘ <>9 AND ‘projects‘.‘rgt‘ <= 10 AND ‘projects‘.‘lft‘ >= 1 AND ‘projects‘.‘is_public‘ = TRUE AND ‘issues‘.‘is_private‘ = FALSE
redmine	658	SELECT COUNT(*) FROM ‘issues‘ INNER JOIN ‘projects‘ ON ‘projects‘.‘id‘ = ‘issues‘.‘project_id‘ INNER JOIN ‘issue_statuses‘ ON ‘issue_statuses‘.‘id‘ = ‘issues‘.‘status_id‘ WHERE (((projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’issue_tracking’)) AND (((projects.is_public = TRUE AND projects.id NOT IN (SELECT project_id FROM members WHERE user_id IN (6,13))) AND ((issues.is_private = FALSE)))))) AND ((issues.status_id IN (SELECT id FROM issue_statuses WHERE is_closed=FALSE)) AND (issues.start_date >’2011-10-11 23:59:59.999999’ AND issues.start_date <= ’2011-10-12 23:59:59.999999’))	SELECT COUNT(*) FROM ‘projects‘ AS ‘projects‘ INNER JOIN ‘issues‘ AS ‘issues‘ ON ‘projects‘.‘id‘ = ‘issues‘.‘project_id‘ WHERE EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘projects‘.‘id‘ AND ‘em‘.‘name‘ = ’issue_tracking’) AND NOT ‘projects‘.‘id‘ IN (SELECT ‘project_id‘ FROM ‘members‘ WHERE ‘user_id‘ IN (?)) AND ‘issues‘.‘status_id‘ IN (SELECT ‘issue_statuses‘.‘id‘ AS ‘id‘ FROM ‘issue_statuses‘ AS ‘issue_statuses‘ WHERE ‘issue_statuses‘.‘is_closed‘ = FALSE) AND ‘projects‘.‘status‘ <>9 AND ‘projects‘.‘is_public‘ = TRUE AND ‘issues‘.‘start_date‘ <= ’2011-10-12 23:59:59.999999’ AND ‘issues‘.‘start_date‘ >’2011-10-11 23:59:59.999999’ AND ‘issues‘.‘is_private‘ = FALSE
redmine	670	SELECT COUNT(*) FROM ‘issues‘ INNER JOIN ‘projects‘ ON ‘projects‘.‘id‘ = ‘issues‘.‘project_id‘ INNER JOIN ‘issue_statuses‘ ON ‘issue_statuses‘.‘id‘ = ‘issues‘.‘status_id‘ WHERE (((projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’issue_tracking’)) AND (((projects.is_public = TRUE AND projects.id NOT IN (SELECT project_id FROM members WHERE user_id IN (6,13))) AND ((issues.is_private = FALSE)))))) AND ((issues.status_id IN (SELECT id FROM issue_statuses WHERE is_closed=FALSE)) AND issues.id NOT IN (SELECT issues.id FROM issues LEFT OUTER JOIN custom_values ON custom_values.customized_type=’Issue’ AND custom_values.customized_id=issues.id AND custom_values.custom_field_id=1 WHERE (custom_values.value IN (’c’)) AND (((1=1) AND (issues.tracker_id IN (SELECT tracker_id FROM custom_fields_trackers WHERE custom_field_id = 1)) AND (EXISTS (SELECT 1 FROM custom_fields ifa WHERE ifa.is_for_all = TRUE AND ifa.id = 1) OR issues.project_id IN (SELECT project_id FROM custom_fields_projects WHERE custom_field_id = 1))))))	SELECT COUNT(*) FROM ‘projects‘ AS ‘projects‘ INNER JOIN ‘issues‘ AS ‘issues‘ ON ‘projects‘.‘id‘ = ‘issues‘.‘project_id‘ WHERE NOT ‘projects‘.‘id‘ IN (SELECT ‘project_id‘ FROM ‘members‘ WHERE ‘user_id‘ IN (?)) AND EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘issues‘.‘project_id‘ AND ‘em‘.‘name‘ = ’issue_tracking’) AND ‘issues‘.‘status_id‘ IN (SELECT ‘issue_statuses‘.‘id‘ AS ‘id‘ FROM ‘issue_statuses‘ AS ‘issue_statuses‘ WHERE ‘issue_statuses‘.‘is_closed‘ = FALSE) AND ‘projects‘.‘status‘ <>9 AND ‘projects‘.‘is_public‘ = TRUE AND ‘issues‘.‘is_private‘ = FALSE AND NOT ‘issues‘.‘id‘ IN (SELECT ‘issues‘.‘id‘ FROM ‘issues‘ LEFT JOIN ‘custom_values‘ ON ‘custom_values‘.‘customized_type‘ = ’Issue’ AND ‘custom_values‘.‘customized_id‘ = ‘issues‘.‘id‘ AND ‘custom_values‘.‘custom_field_id‘ = 1 WHERE ‘custom_values‘.‘value‘ IN (?) AND (‘issues‘.‘tracker_id‘ IN (SELECT ‘tracker_id‘ FROM ‘custom_fields_trackers‘ WHERE ‘custom_field_id‘ = 1) AND (EXISTS (SELECT 1 FROM ‘custom_fields‘ AS ‘ifa‘ WHERE ‘ifa‘.‘is_for_all‘ = TRUE AND ‘ifa‘.‘id‘ = 1) OR ‘issues‘.‘project_id‘ IN (SELECT ‘project_id‘ FROM ‘custom_fields_projects‘ WHERE ‘custom_field_id‘ = 1))))
redmine	680	SELECT COUNT(*) FROM ‘issues‘ INNER JOIN ‘projects‘ ON ‘projects‘.‘id‘ = ‘issues‘.‘project_id‘ INNER JOIN ‘issue_statuses‘ ON ‘issue_statuses‘.‘id‘ = ‘issues‘.‘status_id‘ WHERE (((projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’issue_tracking’)) AND (((projects.is_public = TRUE AND projects.id NOT IN (SELECT project_id FROM members WHERE user_id IN (6,13))) AND ((issues.is_private = FALSE)))))) AND ((issues.status_id IN (SELECT id FROM issue_statuses WHERE is_closed=FALSE)) AND (issues.estimated_hours IS NOT NULL))	SELECT COUNT(*) FROM ‘issues‘ AS ‘issues‘ INNER JOIN ‘projects‘ AS ‘projects‘ ON ‘issues‘.‘project_id‘ = ‘projects‘.‘id‘ WHERE EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘issues‘.‘project_id‘ AND ‘em‘.‘name‘ = ’issue_tracking’) AND NOT ‘issues‘.‘project_id‘ IN (SELECT ‘project_id‘ FROM ‘members‘ WHERE ‘user_id‘ IN (?)) AND ‘issues‘.‘status_id‘ IN (SELECT ‘issue_statuses‘.‘id‘ AS ‘id‘ FROM ‘issue_statuses‘ AS ‘issue_statuses‘ WHERE ‘issue_statuses‘.‘is_closed‘ = FALSE) AND ‘projects‘.‘status‘ <>9 AND ‘projects‘.‘is_public‘ = TRUE AND ‘issues‘.‘is_private‘ = FALSE AND NOT ‘issues‘.‘estimated_hours‘ IS NULL
redmine	706	SELECT SUM(‘issues‘.‘estimated_hours‘) FROM ‘issues‘ INNER JOIN ‘projects‘ ON ‘projects‘.‘id‘ = ‘issues‘.‘project_id‘ WHERE (((projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’issue_tracking’)) AND (((projects.is_public = TRUE AND projects.id NOT IN (SELECT project_id FROM members WHERE user_id IN (2,12))) AND ((issues.is_private = FALSE OR issues.author_id = 2 OR issues.assigned_to_id IN (2)))) OR (projects.id IN (1,5) AND (1=1)) OR (projects.id IN (2) AND ((issues.is_private = FALSE OR issues.author_id = 2 OR issues.assigned_to_id IN (2))))))) AND ((projects.id = 1 OR (projects.lft >1 AND projects.rgt <10)))	SELECT SUM(‘issues‘.‘estimated_hours‘) FROM ‘projects‘ AS ‘projects‘ INNER JOIN ‘issues‘ AS ‘issues‘ ON ‘projects‘.‘id‘ = ‘issues‘.‘project_id‘ WHERE EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘issues‘.‘project_id‘ AND ‘em‘.‘name‘ = ’issue_tracking’) AND ‘projects‘.‘status‘ <>9 AND (‘projects‘.‘is_public‘ = TRUE AND NOT ‘projects‘.‘id‘ IN (SELECT ‘project_id‘ FROM ‘members‘ WHERE ‘user_id‘ IN (?)) AND (‘issues‘.‘is_private‘ = FALSE OR ‘issues‘.‘author_id‘ = 2 OR ‘issues‘.‘assigned_to_id‘ IN (?)) OR ‘projects‘.‘id‘ IN (?) OR ‘projects‘.‘id‘ IN (?) AND (‘issues‘.‘is_private‘ = FALSE OR ‘issues‘.‘author_id‘ = 2 OR ‘issues‘.‘assigned_to_id‘ IN (?))) AND (‘projects‘.‘id‘ = 1 OR ‘projects‘.‘lft‘ >1 AND ‘projects‘.‘rgt‘ <10)

redmine	712	SELECT ‘issues‘.‘id‘ AS t0_r0, ‘issues‘.‘tracker_id‘ AS t0_r1, ‘issues‘.‘project_id‘ AS t0_r2, ‘issues‘.‘subject‘ AS t0_r3, ‘issues‘.‘description‘ AS t0_r4, ‘issues‘.‘due_date‘ AS t0_r5, ‘issues‘.‘category_id‘ AS t0_r6, ‘issues‘.‘status_id‘ AS t0_r7, ‘issues‘.‘assigned_to_id‘ AS t0_r8, ‘issues‘.‘priority_id‘ AS t0_r9, ‘issues‘.‘fixed_version_id‘ AS t0_r10, ‘issues‘.‘author_id‘ AS t0_r11, ‘issues‘.‘lock_version‘ AS t0_r12, ‘issues‘.‘created_on‘ AS t0_r13, ‘issues‘.‘updated_on‘ AS t0_r14, ‘issues‘.‘start_date‘ AS t0_r15, ‘issues‘.‘done_ratio‘ AS t0_r16, ‘issues‘.‘estimated_hours‘ AS t0_r17, ‘issues‘.‘parent_id‘ AS t0_r18, ‘issues‘.‘root_id‘ AS t0_r19, ‘issues‘.‘lft‘ AS t0_r20, ‘issues‘.‘rgt‘ AS t0_r21, ‘issues‘.‘is_private‘ AS t0_r22, ‘issues‘.‘closed_on‘ AS t0_r23, ‘issue_statuses‘.‘id‘ AS t1_r0, ‘issue_statuses‘.‘name‘ AS t1_r1, ‘issue_statuses‘.‘is_closed‘ AS t1_r2, ‘issue_statuses‘.‘position‘ AS t1_r3, ‘issue_statuses‘.‘default_done_ratio‘ AS t1_r4, ‘trackers‘.‘id‘ AS t2_r0, ‘trackers‘.‘name‘ AS t2_r1, ‘trackers‘.‘description‘ AS t2_r2, ‘trackers‘.‘is_in_chlog‘ AS t2_r3, ‘trackers‘.‘position‘ AS t2_r4, ‘trackers‘.‘is_in_roadmap‘ AS t2_r5, ‘trackers‘.‘fields_bits‘ AS t2_r6, ‘trackers‘.‘default_status_id‘ AS t2_r7, ‘enumerations‘.‘id‘ AS t3_r0, ‘enumerations‘.‘name‘ AS t3_r1, ‘enumerations‘.‘position‘ AS t3_r2, ‘enumerations‘.‘is_default‘ AS t3_r3, ‘enumerations‘.‘type‘ AS t3_r4, ‘enumerations‘.‘active‘ AS t3_r5, ‘enumerations‘.‘project_id‘ AS t3_r6, ‘enumerations‘.‘parent_id‘ AS t3_r7, ‘enumerations‘.‘position_name‘ AS t3_r8 FROM ‘issues‘ INNER JOIN ‘projects‘ ON ‘projects‘.‘id‘ = ‘issues‘.‘project_id‘ LEFT OUTER JOIN ‘issue_statuses‘ ON ‘issue_statuses‘.‘id‘ = ‘issues‘.‘status_id‘ LEFT OUTER JOIN ‘trackers‘ ON ‘trackers‘.‘id‘ = ‘issues‘.‘tracker_id‘ LEFT OUTER JOIN ‘enumerations‘ ON ‘enumerations‘.‘id‘ = ‘issues‘.‘priority_id‘ AND ‘enumerations‘.‘type‘ IN (’IssuePriority’) WHERE ‘issues‘.‘fixed_version_id‘ = 4 AND (((projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’issue_tracking’)) AND (((projects.is_public = TRUE AND projects.id NOT IN (SELECT project_id FROM members WHERE user_id IN (7,12))) AND ((issues.is_private = FALSE OR issues.author_id = 7 OR issues.assigned_to_id IN (7))))))) ORDER BY trackers.position, issues.id	SELECT * FROM (SELECT ‘issues‘.‘id‘ AS ‘t0_r0‘, ‘issues‘.‘tracker_id‘ AS ‘t0_r1‘, ‘issues‘.‘project_id‘ AS ‘t0_r2‘, ‘issues‘.‘subject‘ AS ‘t0_r3‘, ‘issues‘.‘description‘ AS ‘t0_r4‘, ‘issues‘.‘due_date‘ AS ‘t0_r5‘, ‘issues‘.‘category_id‘ AS ‘t0_r6‘, ‘issues‘.‘status_id‘ AS ‘t0_r7‘, ‘issues‘.‘assigned_to_id‘ AS ‘t0_r8‘, ‘issues‘.‘priority_id‘ AS ‘t0_r9‘, ‘issues‘.‘fixed_version_id‘ AS ‘t0_r10‘, ‘issues‘.‘author_id‘ AS ‘t0_r11‘, ‘issues‘.‘lock_version‘ AS ‘t0_r12‘, ‘issues‘.‘created_on‘ AS ‘t0_r13‘, ‘issues‘.‘updated_on‘ AS ‘t0_r14‘, ‘issues‘.‘start_date‘ AS ‘t0_r15‘, ‘issues‘.‘done_ratio‘ AS ‘t0_r16‘, ‘issues‘.‘estimated_hours‘ AS ‘t0_r17‘, ‘issues‘.‘parent_id‘ AS ‘t0_r18‘, ‘issues‘.‘root_id‘ AS ‘t0_r19‘, ‘issues‘.‘lft‘ AS ‘t0_r20‘, ‘issues‘.‘rgt‘ AS ‘t0_r21‘, ‘issues‘.‘is_private‘ AS ‘t0_r22‘, ‘issues‘.‘closed_on‘ AS ‘t0_r23‘, ‘issue_statuses‘.‘id‘ AS ‘t1_r0‘, ‘issue_statuses‘.‘name‘ AS ‘t1_r1‘, ‘issue_statuses‘.‘is_closed‘ AS ‘t1_r2‘, ‘issue_statuses‘.‘position‘ AS ‘t1_r3‘, ‘issue_statuses‘.‘default_done_ratio‘ AS ‘t1_r4‘, ‘trackers‘.‘id‘ AS ‘t2_r0‘, ‘trackers‘.‘name‘ AS ‘t2_r1‘, ‘trackers‘.‘description‘ AS ‘t2_r2‘, ‘trackers‘.‘is_in_chlog‘ AS ‘t2_r3‘, ‘trackers‘.‘position‘ AS ‘t2_r4‘, ‘trackers‘.‘is_in_roadmap‘ AS ‘t2_r5‘, ‘trackers‘.‘fields_bits‘ AS ‘t2_r6‘, ‘trackers‘.‘default_status_id‘ AS ‘t2_r7‘, ‘enumerations‘.‘id‘ AS ‘t3_r0‘, ‘enumerations‘.‘name‘ AS ‘t3_r1‘, ‘enumerations‘.‘position‘ AS ‘t3_r2‘, ‘enumerations‘.‘is_default‘ AS ‘t3_r3‘, ‘enumerations‘.‘type‘ AS ‘t3_r4‘, ‘enumerations‘.‘active‘ AS ‘t3_r5‘, ‘enumerations‘.‘project_id‘ AS ‘t3_r6‘, ‘enumerations‘.‘parent_id‘ AS ‘t3_r7‘, ‘enumerations‘.‘position_name‘ AS ‘t3_r8‘ FROM ‘issues‘ AS ‘issues‘ LEFT JOIN ‘enumerations‘ AS ‘enumerations‘ ON ‘enumerations‘.‘id‘ = ‘issues‘.‘priority_id‘ AND ‘enumerations‘.‘type‘ IN (?) INNER JOIN ‘issue_statuses‘ AS ‘issue_statuses‘ ON ‘issues‘.‘status_id‘ = ‘issue_statuses‘.‘id‘ INNER JOIN ‘trackers‘ AS ‘trackers‘ ON ‘issues‘.‘tracker_id‘ = ‘trackers‘.‘id‘ INNER JOIN ‘projects‘ AS ‘projects‘ ON ‘issues‘.‘project_id‘ = ‘projects‘.‘id‘ WHERE EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘projects‘.‘id‘ AND ‘em‘.‘name‘ = ’issue_tracking’) AND NOT ‘projects‘.‘id‘ IN (SELECT ‘project_id‘ FROM ‘members‘ WHERE ‘user_id‘ IN (?)) AND ‘projects‘.‘status‘ <>9 AND ‘projects‘.‘is_public‘ = TRUE) AS ‘sub_0‘ WHERE (‘sub_0‘.‘t0_r22‘ = FALSE OR ‘sub_0‘.‘t0_r11‘ = 7 OR ‘sub_0‘.‘t0_r8‘ IN (?)) AND ‘sub_0‘.‘t0_r10‘ = 4 ORDER BY ‘sub_0‘.‘t2_r4‘, ‘sub_0‘.‘t0_r0‘

redmine	715	SELECT SUM(COALESCE(estimated_hours, 2.0) * 100) FROM ‘issues‘ INNER JOIN ‘projects‘ ON ‘projects‘.‘id‘ = ‘issues‘.‘project_id‘ INNER JOIN ‘issue_statuses‘ ON ‘issue_statuses‘.‘id‘ = ‘issues‘.‘status_id‘ WHERE ‘issues‘.‘fixed_version_id‘ = 4 AND (((projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’issue_tracking’)) AND (((projects.is_public = TRUE AND projects.id NOT IN (SELECT project_id FROM members WHERE user_id IN (7,12))) AND ((issues.is_private = FALSE OR issues.author_id = 7 OR issues.assigned_to_id IN (7))))))) AND ‘issue_statuses‘.‘is_closed‘ = TRUE	SELECT SUM(COALESCE(‘issues‘.‘estimated_hours‘, 2.0) * 100) FROM ‘issues‘ AS ‘issues‘ INNER JOIN ‘issue_statuses‘ AS ‘issue_statuses‘ ON ‘issues‘.‘status_id‘ = ‘issue_statuses‘.‘id‘ INNER JOIN ‘projects‘ AS ‘projects‘ ON ‘issues‘.‘project_id‘ = ‘projects‘.‘id‘ WHERE NOT ‘issues‘.‘project_id‘ IN (SELECT ‘project_id‘ FROM ‘members‘ WHERE ‘user_id‘ IN (?)) AND EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘projects‘.‘id‘ AND ‘em‘.‘name‘ = ’issue_tracking’) AND ‘projects‘.‘status‘ <>9 AND ‘projects‘.‘is_public‘ = TRUE AND (‘issues‘.‘is_private‘ = FALSE OR ‘issues‘.‘author_id‘ = 7 OR ‘issues‘.‘assigned_to_id‘ IN (?)) AND ‘issues‘.‘fixed_version_id‘ = 4 AND ‘issue_statuses‘.‘is_closed‘ = TRUE
redmine	748	SELECT COUNT(*) FROM ‘issues‘ INNER JOIN ‘projects‘ ON ‘projects‘.‘id‘ = ‘issues‘.‘project_id‘ INNER JOIN ‘issue_statuses‘ ON ‘issue_statuses‘.‘id‘ = ‘issues‘.‘status_id‘ WHERE (((projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’issue_tracking’)) AND (((projects.is_public = TRUE AND projects.id NOT IN (SELECT project_id FROM members WHERE user_id IN (6,13))) AND ((issues.is_private = FALSE)))))) AND (projects.status IN (’1’))	SELECT COUNT(*) FROM ‘projects‘ AS ‘projects‘ INNER JOIN ‘issues‘ AS ‘issues‘ ON ‘projects‘.‘id‘ = ‘issues‘.‘project_id‘ WHERE EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘issues‘.‘project_id‘ AND ‘em‘.‘name‘ = ’issue_tracking’) AND ‘projects‘.‘status‘ IN (?) AND ‘projects‘.‘status‘ <>9 AND ‘projects‘.‘is_public‘ = TRUE AND NOT ‘projects‘.‘id‘ IN (SELECT ‘project_id‘ FROM ‘members‘ WHERE ‘user_id‘ IN (?)) AND ‘issues‘.‘is_private‘ = FALSE
redmine	808	SELECT ‘versions‘.* FROM ‘versions‘ INNER JOIN ‘projects‘ ON ‘projects‘.‘id‘ = ‘versions‘.‘project_id‘ WHERE (projects.lft >= 6 AND projects.rgt <= 7 AND projects.status <>9) AND (((projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’issue_tracking’)) AND ((projects.is_public = TRUE AND projects.id NOT IN (SELECT project_id FROM members WHERE user_id IN (6,13))))))	SELECT ‘versions‘.‘id‘ AS ‘id‘, ‘versions‘.‘project_id‘ AS ‘project_id‘, ‘versions‘.‘name‘ AS ‘name‘, ‘versions‘.‘description‘ AS ‘description‘, ‘versions‘.‘effective_date‘ AS ‘effective_date‘, ‘versions‘.‘created_on‘ AS ‘created_on‘, ‘versions‘.‘updated_on‘ AS ‘updated_on‘, ‘versions‘.‘wiki_page_title‘ AS ‘wiki_page_title‘, ‘versions‘.‘status‘ AS ‘status‘, ‘versions‘.‘sharing‘ AS ‘sharing‘ FROM ‘projects‘ AS ‘projects‘ INNER JOIN ‘versions‘ AS ‘versions‘ ON ‘projects‘.‘id‘ = ‘versions‘.‘project_id‘ WHERE EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘versions‘.‘project_id‘ AND ‘em‘.‘name‘ = ’issue_tracking’) AND ‘projects‘.‘rgt‘ <= 7 AND ‘projects‘.‘lft‘ >= 6 AND ‘projects‘.‘is_public‘ = TRUE AND NOT ‘projects‘.‘id‘ IN (SELECT ‘project_id‘ FROM ‘members‘ WHERE ‘user_id‘ IN (?)) AND ‘projects‘.‘status‘ <>9
redmine	810	SELECT ‘issues‘.‘id‘ AS t0_r0, ‘issues‘.‘tracker_id‘ AS t0_r1, ‘issues‘.‘project_id‘ AS t0_r2, ‘issues‘.‘subject‘ AS t0_r3, ‘issues‘.‘description‘ AS t0_r4, ‘issues‘.‘due_date‘ AS t0_r5, ‘issues‘.‘category_id‘ AS t0_r6, ‘issues‘.‘status_id‘ AS t0_r7, ‘issues‘.‘assigned_to_id‘ AS t0_r8, ‘issues‘.‘priority_id‘ AS t0_r9, ‘issues‘.‘fixed_version_id‘ AS t0_r10, ‘issues‘.‘author_id‘ AS t0_r11, ‘issues‘.‘lock_version‘ AS t0_r12, ‘issues‘.‘created_on‘ AS t0_r13, ‘issues‘.‘updated_on‘ AS t0_r14, ‘issues‘.‘start_date‘ AS t0_r15, ‘issues‘.‘done_ratio‘ AS t0_r16, ‘issues‘.‘estimated_hours‘ AS t0_r17, ‘issues‘.‘parent_id‘ AS t0_r18, ‘issues‘.‘root_id‘ AS t0_r19, ‘issues‘.‘lft‘ AS t0_r20, ‘issues‘.‘rgt‘ AS t0_r21, ‘issues‘.‘is_private‘ AS t0_r22, ‘issues‘.‘closed_on‘ AS t0_r23, ‘issue_statuses‘.‘id‘ AS t1_r0, ‘issue_statuses‘.‘name‘ AS t1_r1, ‘issue_statuses‘.‘is_closed‘ AS t1_r2, ‘issue_statuses‘.‘position‘ AS t1_r3, ‘issue_statuses‘.‘default_done_ratio‘ AS t1_r4, ‘trackers‘.‘id‘ AS t2_r0, ‘trackers‘.‘name‘ AS t2_r1, ‘trackers‘.‘description‘ AS t2_r2, ‘trackers‘.‘is_in_chlog‘ AS t2_r3, ‘trackers‘.‘position‘ AS t2_r4, ‘trackers‘.‘is_in_roadmap‘ AS t2_r5, ‘trackers‘.‘fields_bits‘ AS t2_r6, ‘trackers‘.‘default_status_id‘ AS t2_r7, ‘enumerations‘.‘id‘ AS t3_r0, ‘enumerations‘.‘name‘ AS t3_r1, ‘enumerations‘.‘position‘ AS t3_r2, ‘enumerations‘.‘is_default‘ AS t3_r3, ‘enumerations‘.‘type‘ AS t3_r4, ‘enumerations‘.‘active‘ AS t3_r5, ‘enumerations‘.‘project_id‘ AS t3_r6, ‘enumerations‘.‘parent_id‘ AS t3_r7, ‘enumerations‘.‘position_name‘ AS t3_r8 FROM ‘issues‘ INNER JOIN ‘projects‘ ON ‘projects‘.‘id‘ = ‘issues‘.‘project_id‘ LEFT OUTER JOIN ‘issue_statuses‘ ON ‘issue_statuses‘.‘id‘ = ‘issues‘.‘status_id‘ LEFT OUTER JOIN ‘trackers‘ ON ‘trackers‘.‘id‘ = ‘issues‘.‘tracker_id‘ LEFT OUTER JOIN ‘enumerations‘ ON ‘enumerations‘.‘id‘ = ‘issues‘.‘priority_id‘ AND ‘enumerations‘.‘type‘ IN (’IssuePriority’) WHERE ‘issues‘.‘fixed_version_id‘ = 3 AND (projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’issue_tracking’)) ORDER BY trackers.position, issues.id	SELECT ‘issues‘.‘id‘ AS ‘t0_r0‘, ‘issues‘.‘tracker_id‘ AS ‘t0_r1‘, ‘issues‘.‘project_id‘ AS ‘t0_r2‘, ‘issues‘.‘subject‘ AS ‘t0_r3‘, ‘issues‘.‘description‘ AS ‘t0_r4‘, ‘issues‘.‘due_date‘ AS ‘t0_r5‘, ‘issues‘.‘category_id‘ AS ‘t0_r6‘, ‘issues‘.‘status_id‘ AS ‘t0_r7‘, ‘issues‘.‘assigned_to_id‘ AS ‘t0_r8‘, ‘issues‘.‘priority_id‘ AS ‘t0_r9‘, ‘issues‘.‘fixed_version_id‘ AS ‘t0_r10‘, ‘issues‘.‘author_id‘ AS ‘t0_r11‘, ‘issues‘.‘lock_version‘ AS ‘t0_r12‘, ‘issues‘.‘created_on‘ AS ‘t0_r13‘, ‘issues‘.‘updated_on‘ AS ‘t0_r14‘, ‘issues‘.‘start_date‘ AS ‘t0_r15‘, ‘issues‘.‘done_ratio‘ AS ‘t0_r16‘, ‘issues‘.‘estimated_hours‘ AS ‘t0_r17‘, ‘issues‘.‘parent_id‘ AS ‘t0_r18‘, ‘issues‘.‘root_id‘ AS ‘t0_r19‘, ‘issues‘.‘lft‘ AS ‘t0_r20‘, ‘issues‘.‘rgt‘ AS ‘t0_r21‘, ‘issues‘.‘is_private‘ AS ‘t0_r22‘, ‘issues‘.‘closed_on‘ AS ‘t0_r23‘, ‘issue_statuses‘.‘id‘ AS ‘t1_r0‘, ‘issue_statuses‘.‘name‘ AS ‘t1_r1‘, ‘issue_statuses‘.‘is_closed‘ AS ‘t1_r2‘, ‘issue_statuses‘.‘position‘ AS ‘t1_r3‘, ‘issue_statuses‘.‘default_done_ratio‘ AS ‘t1_r4‘, ‘trackers‘.‘id‘ AS ‘t2_r0‘, ‘trackers‘.‘name‘ AS ‘t2_r1‘, ‘trackers‘.‘description‘ AS ‘t2_r2‘, ‘trackers‘.‘is_in_chlog‘ AS ‘t2_r3‘, ‘trackers‘.‘position‘ AS ‘t2_r4‘, ‘trackers‘.‘is_in_roadmap‘ AS ‘t2_r5‘, ‘trackers‘.‘fields_bits‘ AS ‘t2_r6‘, ‘trackers‘.‘default_status_id‘ AS ‘t2_r7‘, ‘enumerations‘.‘id‘ AS ‘t3_r0‘, ‘enumerations‘.‘name‘ AS ‘t3_r1‘, ‘enumerations‘.‘position‘ AS ‘t3_r2‘, ‘enumerations‘.‘is_default‘ AS ‘t3_r3‘, ‘enumerations‘.‘type‘ AS ‘t3_r4‘, ‘enumerations‘.‘active‘ AS ‘t3_r5‘, ‘enumerations‘.‘project_id‘ AS ‘t3_r6‘, ‘enumerations‘.‘parent_id‘ AS ‘t3_r7‘, ‘enumerations‘.‘position_name‘ AS ‘t3_r8‘ FROM ‘issues‘ AS ‘issues‘ LEFT JOIN ‘enumerations‘ AS ‘enumerations‘ ON ‘enumerations‘.‘id‘ = ‘issues‘.‘priority_id‘ AND ‘enumerations‘.‘type‘ IN (?) INNER JOIN ‘trackers‘ AS ‘trackers‘ ON ‘issues‘.‘tracker_id‘ = ‘trackers‘.‘id‘ INNER JOIN ‘issue_statuses‘ AS ‘issue_statuses‘ ON ‘issues‘.‘status_id‘ = ‘issue_statuses‘.‘id‘ INNER JOIN ‘projects‘ AS ‘projects‘ ON ‘issues‘.‘project_id‘ = ‘projects‘.‘id‘ WHERE EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘issues‘.‘project_id‘ AND ‘em‘.‘name‘ = ’issue_tracking’) AND ‘projects‘.‘status‘ <>9 AND ‘issues‘.‘fixed_version_id‘ = 3 ORDER BY ‘trackers‘.‘position‘, ‘issues‘.‘id‘

redmine	869	SELECT ‘roles‘.‘id‘ FROM ‘roles‘ INNER JOIN ‘roles_managed_roles‘ ON ‘roles‘.‘id‘ = ‘roles_managed_roles‘.‘managed_role_id‘ WHERE ‘roles_managed_roles‘.‘role_id‘ = 2	SELECT ‘roles_managed_roles‘.‘managed_role_id‘ AS ‘managed_role_id‘ FROM ‘roles_managed_roles‘ AS ‘roles_managed_roles‘ WHERE ‘roles_managed_roles‘.‘role_id‘ = 2
redmine	918	SELECT ‘issues‘.‘id‘ FROM ‘issues‘ INNER JOIN ‘projects‘ ON ‘projects‘.‘id‘ = ‘issues‘.‘project_id‘ INNER JOIN ‘issue_statuses‘ ON ‘issue_statuses‘.‘id‘ = ‘issues‘.‘status_id‘ WHERE (((projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’issue_tracking’)) AND (((projects.is_public = TRUE AND projects.id NOT IN (SELECT project_id FROM members WHERE user_id IN (6,13))) AND ((issues.is_private = FALSE)))))) AND ((issues.tracker_id IN (’1’)) AND projects.lft >= 1 AND projects.rgt <= 10) ORDER BY issues.id DESC LIMIT 501	SELECT ‘issues‘.‘id‘ AS ‘id‘ FROM ‘projects‘ AS ‘projects‘ INNER JOIN ‘issues‘ AS ‘issues‘ ON ‘projects‘.‘id‘ = ‘issues‘.‘project_id‘ WHERE NOT ‘issues‘.‘project_id‘ IN (SELECT ‘project_id‘ FROM ‘members‘ WHERE ‘user_id‘ IN (?)) AND EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘projects‘.‘id‘ AND ‘em‘.‘name‘ = ’issue_tracking’) AND ‘projects‘.‘status‘ <>9 AND ‘projects‘.‘rgt‘ <= 10 AND ‘projects‘.‘lft‘ >= 1 AND ‘projects‘.‘is_public‘ = TRUE AND ‘issues‘.‘tracker_id‘ IN (?) AND ‘issues‘.‘is_private‘ = FALSE ORDER BY ‘issues‘.‘id‘ DESC LIMIT 501
redmine	992	SELECT ‘issues‘.* FROM ‘issues‘ INNER JOIN ‘projects‘ ON ‘projects‘.‘id‘ = ‘issues‘.‘project_id‘ INNER JOIN ‘versions‘ ON ‘versions‘.‘id‘ = ‘issues‘.‘fixed_version_id‘ WHERE (issues.fixed_version_id IS NOT NULL AND issues.project_id <>versions.project_id AND versions.sharing <>’system’) AND (issues.fixed_version_id = 7)	SELECT ‘issues‘.‘id‘ AS ‘id‘, ‘issues‘.‘tracker_id‘ AS ‘tracker_id‘, ‘issues‘.‘project_id‘ AS ‘project_id‘, ‘issues‘.‘subject‘ AS ‘subject‘, ‘issues‘.‘description‘ AS ‘description‘, ‘issues‘.‘due_date‘ AS ‘due_date‘, ‘issues‘.‘category_id‘ AS ‘category_id‘, ‘issues‘.‘status_id‘ AS ‘status_id‘, ‘issues‘.‘assigned_to_id‘ AS ‘assigned_to_id‘, ‘issues‘.‘priority_id‘ AS ‘priority_id‘, ‘issues‘.‘fixed_version_id‘ AS ‘fixed_version_id‘, ‘issues‘.‘author_id‘ AS ‘author_id‘, ‘issues‘.‘lock_version‘ AS ‘lock_version‘, ‘issues‘.‘created_on‘ AS ‘created_on‘, ‘issues‘.‘updated_on‘ AS ‘updated_on‘, ‘issues‘.‘start_date‘ AS ‘start_date‘, ‘issues‘.‘done_ratio‘ AS ‘done_ratio‘, ‘issues‘.‘estimated_hours‘ AS ‘estimated_hours‘, ‘issues‘.‘parent_id‘ AS ‘parent_id‘, ‘issues‘.‘root_id‘ AS ‘root_id‘, ‘issues‘.‘lft‘ AS ‘lft‘, ‘issues‘.‘rgt‘ AS ‘rgt‘, ‘issues‘.‘is_private‘ AS ‘is_private‘, ‘issues‘.‘closed_on‘ AS ‘closed_on‘ FROM ‘versions‘ AS ‘versions‘ INNER JOIN ‘issues‘ AS ‘issues‘ ON ‘versions‘.‘id‘ = ‘issues‘.‘fixed_version_id‘ WHERE NOT ‘versions‘.‘id‘ IS NULL AND ‘versions‘.‘id‘ = 7 AND ‘versions‘.‘sharing‘ <>’system’ AND ‘issues‘.‘project_id‘ <>‘versions‘.‘project_id‘
redmine	1050	SELECT ‘issues‘.‘id‘ FROM ‘issues‘ INNER JOIN ‘projects‘ ON ‘projects‘.‘id‘ = ‘issues‘.‘project_id‘ INNER JOIN ‘issue_statuses‘ ON ‘issue_statuses‘.‘id‘ = ‘issues‘.‘status_id‘ WHERE (((projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’issue_tracking’)) AND (((projects.is_public = TRUE AND projects.id NOT IN (SELECT project_id FROM members WHERE user_id IN (6,13))) AND ((issues.is_private = FALSE)))))) AND (issues.id IN (1,7)) ORDER BY COALESCE((SELECT SUM(estimated_hours) FROM issues subtasks WHERE ((projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’issue_tracking’)) AND (((projects.is_public = TRUE AND projects.id NOT IN (SELECT project_id FROM members WHERE user_id IN (6,13))) AND ((subtasks.is_private = FALSE))))) AND subtasks.root_id = issues.root_id AND subtasks.lft >= issues.lft AND subtasks.rgt <= issues.rgt), 0) ASC, issues.id DESC	SELECT ‘issues‘.‘id‘ AS ‘id‘ FROM ‘projects‘ AS ‘projects‘ INNER JOIN ‘issues‘ AS ‘issues‘ ON ‘projects‘.‘id‘ = ‘issues‘.‘project_id‘ WHERE NOT ‘projects‘.‘id‘ IN (SELECT ‘project_id‘ FROM ‘members‘ WHERE ‘user_id‘ IN (?)) AND EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘projects‘.‘id‘ AND ‘em‘.‘name‘ = ’issue_tracking’) AND ‘projects‘.‘status‘ <>9 AND ‘projects‘.‘is_public‘ = TRUE AND ‘issues‘.‘is_private‘ = FALSE AND ‘issues‘.‘id‘ IN (?) ORDER BY COALESCE((SELECT SUM(‘estimated_hours‘) FROM ‘issues‘ AS ‘subtasks‘ WHERE ‘projects‘.‘status‘ <>9 AND EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘projects‘.‘id‘ AND ‘em‘.‘name‘ = ’issue_tracking’) AND (‘projects‘.‘is_public‘ = TRUE AND NOT ‘projects‘.‘id‘ IN (SELECT ‘project_id‘ FROM ‘members‘ WHERE ‘user_id‘ IN (?)) AND ‘subtasks‘.‘is_private‘ = FALSE) AND ‘subtasks‘.‘root_id‘ = ‘issues‘.‘root_id‘ AND ‘subtasks‘.‘lft‘ >= ‘issues‘.‘lft‘ AND ‘subtasks‘.‘rgt‘ <= ‘issues‘.‘rgt‘), 0) ASC, ‘issues‘.‘id‘ DESC
redmine	1101	SELECT 1 AS one FROM ‘issues‘ INNER JOIN ‘changesets_issues‘ ON ‘issues‘.‘id‘ = ‘changesets_issues‘.‘issue_id‘ WHERE ‘changesets_issues‘.‘changeset_id‘ = 103 AND ‘issues‘.‘id‘ = 2 LIMIT 1	SELECT 1 AS ‘one‘ FROM ‘changesets_issues‘ AS ‘changesets_issues‘ WHERE ‘changesets_issues‘.‘issue_id‘ = 2 AND ‘changesets_issues‘.‘changeset_id‘ = 103 LIMIT 1
redmine	1121	SELECT 1 AS one FROM ‘members‘ INNER JOIN ‘member_roles‘ ON ‘members‘.‘id‘ = ‘member_roles‘.‘member_id‘ WHERE ‘member_roles‘.‘role_id‘ = 914 LIMIT 1	SELECT 1 AS ‘one‘ FROM ‘member_roles‘ AS ‘member_roles‘ WHERE ‘member_roles‘.‘role_id‘ = 914 LIMIT 1
redmine	1131	SELECT DISTINCT ‘wiki_pages‘.‘created_on‘, ‘wiki_pages‘.‘id‘ FROM ‘wiki_pages‘ INNER JOIN ‘wiki_contents‘ ON ‘wiki_contents‘.‘page_id‘ = ‘wiki_pages‘.‘id‘ INNER JOIN ‘wikis‘ ON ‘wikis‘.‘id‘ = ‘wiki_pages‘.‘wiki_id‘ INNER JOIN ‘projects‘ ON ‘projects‘.‘id‘ = ‘wikis‘.‘project_id‘ WHERE (((projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’wiki’)) AND ((projects.is_public = TRUE AND projects.id NOT IN (SELECT project_id FROM members WHERE user_id IN (6,13)))))) AND (((title LIKE ’%highlighted%’) OR (wiki_contents.text LIKE ’%highlighted%’))) ORDER BY ‘wiki_pages‘.‘created_on‘ DESC, ‘wiki_pages‘.‘id‘ DESC	SELECT DISTINCT ‘wiki_pages‘.‘created_on‘ AS ‘created_on‘, ‘wiki_pages‘.‘id‘ AS ‘id‘ FROM ‘wiki_pages‘ AS ‘wiki_pages‘ INNER JOIN ‘wiki_contents‘ AS ‘wiki_contents‘ ON ‘wiki_pages‘.‘id‘ = ‘wiki_contents‘.‘page_id‘ INNER JOIN ‘wikis‘ AS ‘wikis‘ ON ‘wiki_pages‘.‘wiki_id‘ = ‘wikis‘.‘id‘ INNER JOIN ‘projects‘ AS ‘projects‘ ON ‘wikis‘.‘project_id‘ = ‘projects‘.‘id‘ WHERE EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘wikis‘.‘project_id‘ AND ‘em‘.‘name‘ = ’wiki’) AND (‘wiki_pages‘.‘title‘ LIKE ’%highlighted%’ OR ‘wiki_contents‘.‘text‘ LIKE ’%highlighted%’) AND ‘projects‘.‘status‘ <>9 AND ‘projects‘.‘is_public‘ = TRUE AND NOT ‘projects‘.‘id‘ IN (SELECT ‘project_id‘ FROM ‘members‘ WHERE ‘user_id‘ IN (?)) ORDER BY ‘wiki_pages‘.‘created_on‘ DESC, ‘wiki_pages‘.‘id‘ DESC

redmine	1144	SELECT ‘issues‘.‘id‘ AS t0_r0, ‘issues‘.‘tracker_id‘ AS t0_r1, ‘issues‘.‘project_id‘ AS t0_r2, ‘issues‘.‘subject‘ AS t0_r3, ‘issues‘.‘description‘ AS t0_r4, ‘issues‘.‘due_date‘ AS t0_r5, ‘issues‘.‘category_id‘ AS t0_r6, ‘issues‘.‘status_id‘ AS t0_r7, ‘issues‘.‘assigned_to_id‘ AS t0_r8, ‘issues‘.‘priority_id‘ AS t0_r9, ‘issues‘.‘fixed_version_id‘ AS t0_r10, ‘issues‘.‘author_id‘ AS t0_r11, ‘issues‘.‘lock_version‘ AS t0_r12, ‘issues‘.‘created_on‘ AS t0_r13, ‘issues‘.‘updated_on‘ AS t0_r14, ‘issues‘.‘start_date‘ AS t0_r15, ‘issues‘.‘done_ratio‘ AS t0_r16, ‘issues‘.‘estimated_hours‘ AS t0_r17, ‘issues‘.‘parent_id‘ AS t0_r18, ‘issues‘.‘root_id‘ AS t0_r19, ‘issues‘.‘lft‘ AS t0_r20, ‘issues‘.‘rgt‘ AS t0_r21, ‘issues‘.‘is_private‘ AS t0_r22, ‘issues‘.‘closed_on‘ AS t0_r23, ‘issue_statuses‘.‘id‘ AS t1_r0, ‘issue_statuses‘.‘name‘ AS t1_r1, ‘issue_statuses‘.‘is_closed‘ AS t1_r2, ‘issue_statuses‘.‘position‘ AS t1_r3, ‘issue_statuses‘.‘default_done_ratio‘ AS t1_r4, ‘projects‘.‘id‘ AS t2_r0, ‘projects‘.‘name‘ AS t2_r1, ‘projects‘.‘description‘ AS t2_r2, ‘projects‘.‘homepage‘ AS t2_r3, ‘projects‘.‘is_public‘ AS t2_r4, ‘projects‘.‘parent_id‘ AS t2_r5, ‘projects‘.‘created_on‘ AS t2_r6, ‘projects‘.‘updated_on‘ AS t2_r7, ‘projects‘.‘identifier‘ AS t2_r8, ‘projects‘.‘status‘ AS t2_r9, ‘projects‘.‘lft‘ AS t2_r10, ‘projects‘.‘rgt‘ AS t2_r11, ‘projects‘.‘inherit_members‘ AS t2_r12, ‘projects‘.‘default_version_id‘ AS t2_r13, ‘projects‘.‘default_assigned_to_id‘ AS t2_r14 FROM ‘issues‘ INNER JOIN ‘projects‘ ON ‘projects‘.‘id‘ = ‘issues‘.‘project_id‘ INNER JOIN ‘issue_statuses‘ ON ‘issue_statuses‘.‘id‘ = ‘issues‘.‘status_id‘ WHERE (((projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’issue_tracking’)) AND (((projects.is_public = TRUE AND projects.id NOT IN (SELECT project_id FROM members WHERE user_id IN (2,12))) AND ((issues.is_private = FALSE OR issues.author_id = 2 OR issues.assigned_to_id IN (2)))) OR (projects.id IN (1,5) AND (1=1)) OR (projects.id IN (2) AND ((issues.is_private = FALSE OR issues.author_id = 2 OR issues.assigned_to_id IN (2))))))) AND ((issues.project_id IN (’1’,’2’,’5’))) ORDER BY issues.id DESC	SELECT * FROM ‘issues‘ AS ‘issues‘ INNER JOIN ‘issue_statuses‘ AS ‘issue_statuses‘ ON ‘issues‘.‘status_id‘ = ‘issue_statuses‘.‘id‘ INNER JOIN ‘projects‘ AS ‘projects‘ ON ‘issues‘.‘project_id‘ = ‘projects‘.‘id‘ WHERE (‘projects‘.‘is_public‘ = TRUE AND NOT ‘projects‘.‘id‘ IN (SELECT ‘project_id‘ FROM ‘members‘ WHERE ‘user_id‘ IN (?)) AND (‘issues‘.‘is_private‘ = FALSE OR ‘issues‘.‘author_id‘ = 2 OR ‘issues‘.‘assigned_to_id‘ IN (?)) OR ‘projects‘.‘id‘ IN (?) OR ‘projects‘.‘id‘ IN (?) AND (‘issues‘.‘is_private‘ = FALSE OR ‘issues‘.‘author_id‘ = 2 OR ‘issues‘.‘assigned_to_id‘ IN (?))) AND ‘projects‘.‘id‘ IN (?) AND ‘projects‘.‘status‘ <>9 AND EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘projects‘.‘id‘ AND ‘em‘.‘name‘ = ’issue_tracking’) ORDER BY ‘issues‘.‘id‘ DESC
redmine	1163	SELECT ‘roles‘.‘id‘ FROM ‘roles‘ INNER JOIN ‘queries_roles‘ ON ‘roles‘.‘id‘ = ‘queries_roles‘.‘role_id‘ WHERE ‘queries_roles‘.‘query_id‘ = 2068	SELECT ‘queries_roles‘.‘role_id‘ AS ‘role_id‘ FROM ‘queries_roles‘ AS ‘queries_roles‘ WHERE ‘queries_roles‘.‘query_id‘ = 2068
redmine	1178	SELECT COUNT(*) FROM ‘news‘ INNER JOIN ‘projects‘ ON ‘projects‘.‘id‘ = ‘news‘.‘project_id‘ WHERE (((projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’news’)) AND ((projects.is_public = TRUE AND projects.id NOT IN (SELECT project_id FROM members WHERE user_id IN (6,13))))))	SELECT COUNT(*) FROM ‘projects‘ AS ‘projects‘ INNER JOIN ‘news‘ AS ‘news‘ ON ‘projects‘.‘id‘ = ‘news‘.‘project_id‘ WHERE NOT ‘projects‘.‘id‘ IN (SELECT ‘project_id‘ FROM ‘members‘ WHERE ‘user_id‘ IN (?)) AND EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘news‘.‘project_id‘ AND ‘em‘.‘name‘ = ’news’) AND ‘projects‘.‘status‘ <>9 AND ‘projects‘.‘is_public‘ = TRUE
redmine	1211	SELECT attachments.* FROM ‘attachments‘ LEFT JOIN versions ON attachments.container_type=’Version’ AND versions.id = attachments.container_id LEFT JOIN projects ON versions.project_id = projects.id OR (attachments.container_type=’Project’ AND attachments.container_id = projects.id) WHERE (attachments.author_id = 2) AND (((projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’files’)) AND ((projects.is_public = TRUE AND projects.id NOT IN (SELECT project_id FROM members WHERE user_id IN (2,12))) OR projects.id IN (1,5) OR projects.id IN (2)))) ORDER BY attachments.id DESC LIMIT 10	SELECT ‘attachments‘.‘id‘ AS ‘id‘, ‘attachments‘.‘container_id‘ AS ‘container_id‘, ‘attachments‘.‘container_type‘ AS ‘container_type‘, ‘attachments‘.‘filename‘ AS ‘filename‘, ‘attachments‘.‘disk_filename‘ AS ‘disk_filename‘, ‘attachments‘.‘filesize‘ AS ‘filesize‘, ‘attachments‘.‘content_type‘ AS ‘content_type‘, ‘attachments‘.‘digest‘ AS ‘digest‘, ‘attachments‘.‘downloads‘ AS ‘downloads‘, ‘attachments‘.‘author_id‘ AS ‘author_id‘, ‘attachments‘.‘created_on‘ AS ‘created_on‘, ‘attachments‘.‘description‘ AS ‘description‘, ‘attachments‘.‘disk_directory‘ AS ‘disk_directory‘ FROM ‘attachments‘ AS ‘attachments‘ INNER JOIN ‘versions‘ AS ‘versions‘ ON ‘attachments‘.‘container_type‘ = ’Version’ AND ‘versions‘.‘id‘ = ‘attachments‘.‘container_id‘ INNER JOIN ‘projects‘ AS ‘projects‘ ON ‘versions‘.‘project_id‘ = ‘projects‘.‘id‘ OR ‘attachments‘.‘container_type‘ = ’Project’ AND ‘attachments‘.‘container_id‘ = ‘projects‘.‘id‘ WHERE EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘projects‘.‘id‘ AND ‘em‘.‘name‘ = ’files’) AND ‘projects‘.‘status‘ <>9 AND (‘projects‘.‘is_public‘ = TRUE AND NOT ‘projects‘.‘id‘ IN (SELECT ‘project_id‘ FROM ‘members‘ WHERE ‘user_id‘ IN (?)) OR ‘projects‘.‘id‘ IN (?) OR ‘projects‘.‘id‘ IN (?)) AND ‘attachments‘.‘author_id‘ = 2 ORDER BY ‘attachments‘.‘id‘ DESC LIMIT 10

redmine	1231	SELECT wiki_content_versions.updated_on, wiki_content_versions.comments, wiki_content_versions.version, wiki_pages.title, wiki_content_versions.page_id, wiki_content_versions.author_id, wiki_content_versions.id FROM ‘wiki_content_versions‘ LEFT JOIN wiki_pages ON wiki_pages.id = wiki_content_versions.page_id LEFT JOIN wikis ON wikis.id = wiki_pages.wiki_id LEFT JOIN projects ON projects.id = wikis.project_id WHERE (wiki_content_versions.updated_on BETWEEN ’2020-04-16’ AND ’2020-05-16’) AND (((projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’wiki’)) AND ((projects.is_public = TRUE AND projects.id NOT IN (SELECT project_id FROM members WHERE user_id IN (6,13))))))	SELECT ‘wiki_content_versions‘.‘updated_on‘ AS ‘updated_on‘, ‘wiki_content_versions‘.‘comments‘ AS ‘comments‘, ‘wiki_content_versions‘.‘version‘ AS ‘version‘, ‘wiki_pages‘.‘title‘ AS ‘title‘, ‘wiki_content_versions‘.‘page_id‘ AS ‘page_id‘, ‘wiki_content_versions‘.‘author_id‘ AS ‘author_id‘, ‘wiki_content_versions‘.‘id‘ AS ‘id‘ FROM ‘wiki_content_versions‘ AS ‘wiki_content_versions‘ INNER JOIN ‘wiki_pages‘ AS ‘wiki_pages‘ ON ‘wiki_content_versions‘.‘page_id‘ = ‘wiki_pages‘.‘id‘ INNER JOIN ‘wikis‘ AS ‘wikis‘ ON ‘wiki_pages‘.‘wiki_id‘ = ‘wikis‘.‘id‘ INNER JOIN ‘projects‘ AS ‘projects‘ ON ‘wikis‘.‘project_id‘ = ‘projects‘.‘id‘ WHERE EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘projects‘.‘id‘ AND ‘em‘.‘name‘ = ’wiki’) AND ‘wiki_content_versions‘.‘updated_on‘ BETWEEN ’2020-04-16’ AND ’2020-05-16’ AND ‘projects‘.‘status‘ <>9 AND ‘projects‘.‘is_public‘ = TRUE AND NOT ‘projects‘.‘id‘ IN (SELECT ‘project_id‘ FROM ‘members‘ WHERE ‘user_id‘ IN (?))
redmine	1232	SELECT ‘messages‘.* FROM ‘messages‘ INNER JOIN ‘boards‘ ON ‘boards‘.‘id‘ = ‘messages‘.‘board_id‘ INNER JOIN ‘projects‘ ON ‘projects‘.‘id‘ = ‘boards‘.‘project_id‘ WHERE (messages.created_on BETWEEN ’2020-04-16’ AND ’2020-05-16’) AND (((projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’boards’)) AND ((projects.is_public = TRUE AND projects.id NOT IN (SELECT project_id FROM members WHERE user_id IN (6,13))))))	SELECT ‘messages‘.‘id‘ AS ‘id‘, ‘messages‘.‘board_id‘ AS ‘board_id‘, ‘messages‘.‘parent_id‘ AS ‘parent_id‘, ‘messages‘.‘subject‘ AS ‘subject‘, ‘messages‘.‘content‘ AS ‘content‘, ‘messages‘.‘author_id‘ AS ‘author_id‘, ‘messages‘.‘replies_count‘ AS ‘replies_count‘, ‘messages‘.‘last_reply_id‘ AS ‘last_reply_id‘, ‘messages‘.‘created_on‘ AS ‘created_on‘, ‘messages‘.‘updated_on‘ AS ‘updated_on‘, ‘messages‘.‘locked‘ AS ‘locked‘, ‘messages‘.‘sticky‘ AS ‘sticky‘ FROM ‘messages‘ AS ‘messages‘ INNER JOIN ‘boards‘ AS ‘boards‘ ON ‘messages‘.‘board_id‘ = ‘boards‘.‘id‘ INNER JOIN ‘projects‘ AS ‘projects‘ ON ‘boards‘.‘project_id‘ = ‘projects‘.‘id‘ WHERE NOT ‘projects‘.‘id‘ IN (SELECT ‘project_id‘ FROM ‘members‘ WHERE ‘user_id‘ IN (?)) AND EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘boards‘.‘project_id‘ AND ‘em‘.‘name‘ = ’boards’) AND ‘projects‘.‘status‘ <>9 AND ‘projects‘.‘is_public‘ = TRUE AND ‘messages‘.‘created_on‘ BETWEEN ’2020-04-16’ AND ’2020-05-16’
redmine	1284	SELECT ‘issues‘.* FROM ‘issues‘ INNER JOIN ‘issue_statuses‘ ON ‘issue_statuses‘.‘id‘ = ‘issues‘.‘status_id‘ INNER JOIN ‘trackers‘ ON ‘trackers‘.‘id‘ = ‘issues‘.‘tracker_id‘ INNER JOIN ‘projects‘ ON ‘projects‘.‘id‘ = ‘issues‘.‘project_id‘ INNER JOIN ‘enumerations‘ ON ‘enumerations‘.‘id‘ = ‘issues‘.‘priority_id‘ AND ‘enumerations‘.‘type‘ IN (’IssuePriority’) WHERE (EXISTS (SELECT 1 FROM journals WHERE journals.journalized_type=’Issue’ AND journals.journalized_id=issues.id AND (journals.user_id IN (944)) AND ((journals.private_notes = FALSE OR journals.user_id = 6 OR (1=0)))))	SELECT ‘issues‘.‘id‘ AS ‘id‘, ‘issues‘.‘tracker_id‘ AS ‘tracker_id‘, ‘issues‘.‘project_id‘ AS ‘project_id‘, ‘issues‘.‘subject‘ AS ‘subject‘, ‘issues‘.‘description‘ AS ‘description‘, ‘issues‘.‘due_date‘ AS ‘due_date‘, ‘issues‘.‘category_id‘ AS ‘category_id‘, ‘issues‘.‘status_id‘ AS ‘status_id‘, ‘issues‘.‘assigned_to_id‘ AS ‘assigned_to_id‘, ‘issues‘.‘priority_id‘ AS ‘priority_id‘, ‘issues‘.‘fixed_version_id‘ AS ‘fixed_version_id‘, ‘issues‘.‘author_id‘ AS ‘author_id‘, ‘issues‘.‘lock_version‘ AS ‘lock_version‘, ‘issues‘.‘created_on‘ AS ‘created_on‘, ‘issues‘.‘updated_on‘ AS ‘updated_on‘, ‘issues‘.‘start_date‘ AS ‘start_date‘, ‘issues‘.‘done_ratio‘ AS ‘done_ratio‘, ‘issues‘.‘estimated_hours‘ AS ‘estimated_hours‘, ‘issues‘.‘parent_id‘ AS ‘parent_id‘, ‘issues‘.‘root_id‘ AS ‘root_id‘, ‘issues‘.‘lft‘ AS ‘lft‘, ‘issues‘.‘rgt‘ AS ‘rgt‘, ‘issues‘.‘is_private‘ AS ‘is_private‘, ‘issues‘.‘closed_on‘ AS ‘closed_on‘ FROM ‘issues‘ AS ‘issues‘ INNER JOIN ‘enumerations‘ AS ‘enumerations‘ ON ‘issues‘.‘priority_id‘ = ‘enumerations‘.‘id‘ WHERE EXISTS (SELECT 1 FROM ‘journals‘ AS ‘journals‘ WHERE ‘journals‘.‘user_id‘ IN (?) AND (‘journals‘.‘private_notes‘ = FALSE OR ‘journals‘.‘user_id‘ = 6) AND ‘journals‘.‘journalized_type‘ = ’Issue’ AND ‘journals‘.‘journalized_id‘ = ‘issues‘.‘id‘) AND ‘enumerations‘.‘type‘ IN (?)

redmine	1298	SELECT ‘messages‘.* FROM ‘messages‘ INNER JOIN ‘boards‘ ON ‘boards‘.‘id‘ = ‘messages‘.‘board_id‘ INNER JOIN ‘projects‘ ON ‘projects‘.‘id‘ = ‘boards‘.‘project_id‘ WHERE (messages.created_on BETWEEN ’2020-04-16’ AND ’2020-05-16’) AND (((projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’boards’)) AND ((projects.is_public = TRUE AND projects.id NOT IN (SELECT project_id FROM members WHERE user_id IN (2,12))) OR projects.id IN (1,5) OR projects.id IN (2))))	SELECT ‘messages‘.‘id‘ AS ‘id‘, ‘messages‘.‘board_id‘ AS ‘board_id‘, ‘messages‘.‘parent_id‘ AS ‘parent_id‘, ‘messages‘.‘subject‘ AS ‘subject‘, ‘messages‘.‘content‘ AS ‘content‘, ‘messages‘.‘author_id‘ AS ‘author_id‘, ‘messages‘.‘replies_count‘ AS ‘replies_count‘, ‘messages‘.‘last_reply_id‘ AS ‘last_reply_id‘, ‘messages‘.‘created_on‘ AS ‘created_on‘, ‘messages‘.‘updated_on‘ AS ‘updated_on‘, ‘messages‘.‘locked‘ AS ‘locked‘, ‘messages‘.‘sticky‘ AS ‘sticky‘ FROM ‘messages‘ AS ‘messages‘ INNER JOIN ‘boards‘ AS ‘boards‘ ON ‘messages‘.‘board_id‘ = ‘boards‘.‘id‘ INNER JOIN ‘projects‘ AS ‘projects‘ ON ‘boards‘.‘project_id‘ = ‘projects‘.‘id‘ WHERE EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘boards‘.‘project_id‘ AND ‘em‘.‘name‘ = ’boards’) AND ‘projects‘.‘status‘ <>9 AND (‘projects‘.‘is_public‘ = TRUE AND NOT ‘projects‘.‘id‘ IN (SELECT ‘project_id‘ FROM ‘members‘ WHERE ‘user_id‘ IN (?)) OR ‘projects‘.‘id‘ IN (?) OR ‘projects‘.‘id‘ IN (?)) AND ‘messages‘.‘created_on‘ BETWEEN ’2020-04-16’ AND ’2020-05-16’
redmine	1438	SELECT ‘issues‘.‘id‘ AS t0_r0, ‘issues‘.‘tracker_id‘ AS t0_r1, ‘issues‘.‘project_id‘ AS t0_r2, ‘issues‘.‘subject‘ AS t0_r3, ‘issues‘.‘description‘ AS t0_r4, ‘issues‘.‘due_date‘ AS t0_r5, ‘issues‘.‘category_id‘ AS t0_r6, ‘issues‘.‘status_id‘ AS t0_r7, ‘issues‘.‘assigned_to_id‘ AS t0_r8, ‘issues‘.‘priority_id‘ AS t0_r9, ‘issues‘.‘fixed_version_id‘ AS t0_r10, ‘issues‘.‘author_id‘ AS t0_r11, ‘issues‘.‘lock_version‘ AS t0_r12, ‘issues‘.‘created_on‘ AS t0_r13, ‘issues‘.‘updated_on‘ AS t0_r14, ‘issues‘.‘start_date‘ AS t0_r15, ‘issues‘.‘done_ratio‘ AS t0_r16, ‘issues‘.‘estimated_hours‘ AS t0_r17, ‘issues‘.‘parent_id‘ AS t0_r18, ‘issues‘.‘root_id‘ AS t0_r19, ‘issues‘.‘lft‘ AS t0_r20, ‘issues‘.‘rgt‘ AS t0_r21, ‘issues‘.‘is_private‘ AS t0_r22, ‘issues‘.‘closed_on‘ AS t0_r23, ‘issue_statuses‘.‘id‘ AS t1_r0, ‘issue_statuses‘.‘name‘ AS t1_r1, ‘issue_statuses‘.‘is_closed‘ AS t1_r2, ‘issue_statuses‘.‘position‘ AS t1_r3, ‘issue_statuses‘.‘default_done_ratio‘ AS t1_r4, ‘projects‘.‘id‘ AS t2_r0, ‘projects‘.‘name‘ AS t2_r1, ‘projects‘.‘description‘ AS t2_r2, ‘projects‘.‘homepage‘ AS t2_r3, ‘projects‘.‘is_public‘ AS t2_r4, ‘projects‘.‘parent_id‘ AS t2_r5, ‘projects‘.‘created_on‘ AS t2_r6, ‘projects‘.‘updated_on‘ AS t2_r7, ‘projects‘.‘identifier‘ AS t2_r8, ‘projects‘.‘status‘ AS t2_r9, ‘projects‘.‘lft‘ AS t2_r10, ‘projects‘.‘rgt‘ AS t2_r11, ‘projects‘.‘inherit_members‘ AS t2_r12, ‘projects‘.‘default_version_id‘ AS t2_r13, ‘projects‘.‘default_assigned_to_id‘ AS t2_r14 FROM ‘issues‘ INNER JOIN ‘projects‘ ON ‘projects‘.‘id‘ = ‘issues‘.‘project_id‘ INNER JOIN ‘issue_statuses‘ ON ‘issue_statuses‘.‘id‘ = ‘issues‘.‘status_id‘ WHERE (((projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’issue_tracking’)) AND (((projects.is_public = TRUE AND projects.id NOT IN (SELECT project_id FROM members WHERE user_id IN (2,12))) AND ((issues.is_private = FALSE OR issues.author_id = 2 OR issues.assigned_to_id IN (2)))) OR (projects.id IN (1,5) AND (1=1)) OR (projects.id IN (2) AND ((issues.is_private = FALSE OR issues.author_id = 2 OR issues.assigned_to_id IN (2))))))) AND ((issues.status_id IN (SELECT id FROM issue_statuses WHERE is_closed=FALSE)) AND issues.id IN (SELECT watchers.watchable_id FROM watchers WHERE watchers.watchable_type=’Issue’ AND watchers.user_id IN (’2’)) AND projects.status IN (’1’)) ORDER BY issues.updated_on DESC, issues.id DESC LIMIT 10	SELECT * FROM (SELECT ‘issues‘.‘id‘ AS ‘t0_r0‘, ‘issues‘.‘tracker_id‘ AS ‘t0_r1‘, ‘issues‘.‘project_id‘ AS ‘t0_r2‘, ‘issues‘.‘subject‘ AS ‘t0_r3‘, ‘issues‘.‘description‘ AS ‘t0_r4‘, ‘issues‘.‘due_date‘ AS ‘t0_r5‘, ‘issues‘.‘category_id‘ AS ‘t0_r6‘, ‘issues‘.‘status_id‘ AS ‘t0_r7‘, ‘issues‘.‘assigned_to_id‘ AS ‘t0_r8‘, ‘issues‘.‘priority_id‘ AS ‘t0_r9‘, ‘issues‘.‘fixed_version_id‘ AS ‘t0_r10‘, ‘issues‘.‘author_id‘ AS ‘t0_r11‘, ‘issues‘.‘lock_version‘ AS ‘t0_r12‘, ‘issues‘.‘created_on‘ AS ‘t0_r13‘, ‘issues‘.‘updated_on‘ AS ‘t0_r14‘, ‘issues‘.‘start_date‘ AS ‘t0_r15‘, ‘issues‘.‘done_ratio‘ AS ‘t0_r16‘, ‘issues‘.‘estimated_hours‘ AS ‘t0_r17‘, ‘issues‘.‘parent_id‘ AS ‘t0_r18‘, ‘issues‘.‘root_id‘ AS ‘t0_r19‘, ‘issues‘.‘lft‘ AS ‘t0_r20‘, ‘issues‘.‘rgt‘ AS ‘t0_r21‘, ‘issues‘.‘is_private‘ AS ‘t0_r22‘, ‘issues‘.‘closed_on‘ AS ‘t0_r23‘, ‘issue_statuses‘.‘id‘ AS ‘t1_r0‘, ‘issue_statuses‘.‘name‘ AS ‘t1_r1‘, ‘issue_statuses‘.‘is_closed‘ AS ‘t1_r2‘, ‘issue_statuses‘.‘position‘ AS ‘t1_r3‘, ‘issue_statuses‘.‘default_done_ratio‘ AS ‘t1_r4‘, ‘projects‘.‘id‘ AS ‘t2_r0‘, ‘projects‘.‘name‘ AS ‘t2_r1‘, ‘projects‘.‘description‘ AS ‘t2_r2‘, ‘projects‘.‘homepage‘ AS ‘t2_r3‘, ‘projects‘.‘is_public‘ AS ‘t2_r4‘, ‘projects‘.‘parent_id‘ AS ‘t2_r5‘, ‘projects‘.‘created_on‘ AS ‘t2_r6‘, ‘projects‘.‘updated_on‘ AS ‘t2_r7‘, ‘projects‘.‘identifier‘ AS ‘t2_r8‘, ‘projects‘.‘status‘ AS ‘t2_r9‘, ‘projects‘.‘lft‘ AS ‘t2_r10‘, ‘projects‘.‘rgt‘ AS ‘t2_r11‘, ‘projects‘.‘inherit_members‘ AS ‘t2_r12‘, ‘projects‘.‘default_version_id‘ AS ‘t2_r13‘, ‘projects‘.‘default_assigned_to_id‘ AS ‘t2_r14‘ FROM ‘issues‘ AS ‘issues‘ INNER JOIN ‘projects‘ AS ‘projects‘ ON ‘issues‘.‘project_id‘ = ‘projects‘.‘id‘ INNER JOIN ‘issue_statuses‘ AS ‘issue_statuses‘ ON ‘issues‘.‘status_id‘ = ‘issue_statuses‘.‘id‘ WHERE EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘projects‘.‘id‘ AND ‘em‘.‘name‘ = ’issue_tracking’) AND ‘projects‘.‘status‘ IN (?) AND ‘projects‘.‘status‘ <>9 AND ‘issue_statuses‘.‘id‘ IN (SELECT ‘issue_statuses‘.‘id‘ AS ‘id‘ FROM ‘issue_statuses‘ AS ‘issue_statuses‘ WHERE ‘issue_statuses‘.‘is_closed‘ = FALSE) AND (‘projects‘.‘is_public‘ = TRUE AND NOT ‘projects‘.‘id‘ IN (SELECT ‘project_id‘ FROM ‘members‘ WHERE ‘user_id‘ IN (?)) AND (‘issues‘.‘is_private‘ = FALSE OR ‘issues‘.‘author_id‘ = 2 OR ‘issues‘.‘assigned_to_id‘ IN (?)) OR ‘projects‘.‘id‘ IN (?) OR ‘projects‘.‘id‘ IN (?) AND (‘issues‘.‘is_private‘ = FALSE OR ‘issues‘.‘author_id‘ = 2 OR ‘issues‘.‘assigned_to_id‘ IN (?)))) AS ‘sub_0‘ WHERE ‘sub_0‘.‘t0_r0‘ IN (SELECT ‘watchers‘.‘watchable_id‘ AS ‘watchable_id‘ FROM ‘watchers‘ AS ‘watchers‘ WHERE ‘watchers‘.‘watchable_type‘ = ’Issue’ AND ‘watchers‘.‘user_id‘ IN (?)) ORDER BY ‘sub_0‘.‘t0_r14‘ DESC, ‘sub_0‘.‘t0_r0‘ DESC LIMIT 10

redmine	1439	SELECT COUNT(*) FROM ‘issues‘ INNER JOIN ‘projects‘ ON ‘projects‘.‘id‘ = ‘issues‘.‘project_id‘ INNER JOIN ‘issue_statuses‘ ON ‘issue_statuses‘.‘id‘ = ‘issues‘.‘status_id‘ WHERE (((projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’issue_tracking’)) AND (((projects.is_public = TRUE AND projects.id NOT IN (SELECT project_id FROM members WHERE user_id IN (2,12))) AND ((issues.is_private = FALSE OR issues.author_id = 2 OR issues.assigned_to_id IN (2)))) OR (projects.id IN (1,5) AND (1=1)) OR (projects.id IN (2) AND ((issues.is_private = FALSE OR issues.author_id = 2 OR issues.assigned_to_id IN (2))))))) AND ((issues.status_id IN (SELECT id FROM issue_statuses WHERE is_closed=FALSE)) AND issues.id IN (SELECT watchers.watchable_id FROM watchers WHERE watchers.watchable_type=’Issue’ AND watchers.user_id IN (’2’)) AND projects.status IN (’1’))	SELECT COUNT(*) FROM ‘issues‘ AS ‘issues‘ INNER JOIN ‘projects‘ AS ‘projects‘ ON ‘issues‘.‘project_id‘ = ‘projects‘.‘id‘ WHERE EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘issues‘.‘project_id‘ AND ‘em‘.‘name‘ = ’issue_tracking’) AND ‘issues‘.‘status_id‘ IN (SELECT ‘issue_statuses‘.‘id‘ AS ‘id‘ FROM ‘issue_statuses‘ AS ‘issue_statuses‘ WHERE ‘issue_statuses‘.‘is_closed‘ = FALSE) AND ‘issues‘.‘id‘ IN (SELECT ‘watchers‘.‘watchable_id‘ AS ‘watchable_id‘ FROM ‘watchers‘ AS ‘watchers‘ WHERE ‘watchers‘.‘watchable_type‘ = ’Issue’ AND ‘watchers‘.‘user_id‘ IN (?)) AND ‘projects‘.‘status‘ IN (?) AND ‘projects‘.‘status‘ <>9 AND (‘projects‘.‘is_public‘ = TRUE AND NOT ‘projects‘.‘id‘ IN (SELECT ‘project_id‘ FROM ‘members‘ WHERE ‘user_id‘ IN (?)) AND (‘issues‘.‘is_private‘ = FALSE OR ‘issues‘.‘author_id‘ = 2 OR ‘issues‘.‘assigned_to_id‘ IN (?)) OR ‘projects‘.‘id‘ IN (?) OR ‘projects‘.‘id‘ IN (?) AND (‘issues‘.‘is_private‘ = FALSE OR ‘issues‘.‘author_id‘ = 2 OR ‘issues‘.‘assigned_to_id‘ IN (?)))
redmine	1466	SELECT COUNT(*) AS count_all, ‘issues‘.‘tracker_id‘ AS issues_tracker_id FROM ‘issues‘ INNER JOIN ‘projects‘ ON ‘projects‘.‘id‘ = ‘issues‘.‘project_id‘ INNER JOIN ‘issue_statuses‘ ON ‘issue_statuses‘.‘id‘ = ‘issues‘.‘status_id‘ WHERE (((projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’issue_tracking’)) AND (((projects.is_public = TRUE AND projects.id NOT IN (SELECT project_id FROM members WHERE user_id IN (6,13))) AND ((issues.is_private = FALSE)))))) AND ((issues.status_id IN (SELECT id FROM issue_statuses WHERE is_closed=FALSE)) AND projects.lft >= 1 AND projects.rgt <= 10) GROUP BY ‘issues‘.‘tracker_id‘	SELECT COUNT(*) AS ‘count_all‘, ‘issues‘.‘tracker_id‘ AS ‘issues_tracker_id‘ FROM ‘issues‘ AS ‘issues‘ INNER JOIN ‘projects‘ AS ‘projects‘ ON ‘issues‘.‘project_id‘ = ‘projects‘.‘id‘ WHERE EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘projects‘.‘id‘ AND ‘em‘.‘name‘ = ’issue_tracking’) AND ‘issues‘.‘status_id‘ IN (SELECT ‘issue_statuses‘.‘id‘ AS ‘id‘ FROM ‘issue_statuses‘ AS ‘issue_statuses‘ WHERE ‘issue_statuses‘.‘is_closed‘ = FALSE) AND ‘projects‘.‘status‘ <>9 AND ‘projects‘.‘rgt‘ <= 10 AND ‘projects‘.‘lft‘ >= 1 AND ‘projects‘.‘is_public‘ = TRUE AND NOT ‘projects‘.‘id‘ IN (SELECT ‘project_id‘ FROM ‘members‘ WHERE ‘user_id‘ IN (?)) AND ‘issues‘.‘is_private‘ = FALSE GROUP BY ‘issues‘.‘tracker_id‘
redmine	1473	SELECT ‘issues‘.‘id‘ AS t0_r0, ‘issues‘.‘tracker_id‘ AS t0_r1, ‘issues‘.‘project_id‘ AS t0_r2, ‘issues‘.‘subject‘ AS t0_r3, ‘issues‘.‘description‘ AS t0_r4, ‘issues‘.‘due_date‘ AS t0_r5, ‘issues‘.‘category_id‘ AS t0_r6, ‘issues‘.‘status_id‘ AS t0_r7, ‘issues‘.‘assigned_to_id‘ AS t0_r8, ‘issues‘.‘priority_id‘ AS t0_r9, ‘issues‘.‘fixed_version_id‘ AS t0_r10, ‘issues‘.‘author_id‘ AS t0_r11, ‘issues‘.‘lock_version‘ AS t0_r12, ‘issues‘.‘created_on‘ AS t0_r13, ‘issues‘.‘updated_on‘ AS t0_r14, ‘issues‘.‘start_date‘ AS t0_r15, ‘issues‘.‘done_ratio‘ AS t0_r16, ‘issues‘.‘estimated_hours‘ AS t0_r17, ‘issues‘.‘parent_id‘ AS t0_r18, ‘issues‘.‘root_id‘ AS t0_r19, ‘issues‘.‘lft‘ AS t0_r20, ‘issues‘.‘rgt‘ AS t0_r21, ‘issues‘.‘is_private‘ AS t0_r22, ‘issues‘.‘closed_on‘ AS t0_r23, ‘issue_statuses‘.‘id‘ AS t1_r0, ‘issue_statuses‘.‘name‘ AS t1_r1, ‘issue_statuses‘.‘is_closed‘ AS t1_r2, ‘issue_statuses‘.‘position‘ AS t1_r3, ‘issue_statuses‘.‘default_done_ratio‘ AS t1_r4, ‘trackers‘.‘id‘ AS t2_r0, ‘trackers‘.‘name‘ AS t2_r1, ‘trackers‘.‘description‘ AS t2_r2, ‘trackers‘.‘is_in_chlog‘ AS t2_r3, ‘trackers‘.‘position‘ AS t2_r4, ‘trackers‘.‘is_in_roadmap‘ AS t2_r5, ‘trackers‘.‘fields_bits‘ AS t2_r6, ‘trackers‘.‘default_status_id‘ AS t2_r7, ‘enumerations‘.‘id‘ AS t3_r0, ‘enumerations‘.‘name‘ AS t3_r1, ‘enumerations‘.‘position‘ AS t3_r2, ‘enumerations‘.‘is_default‘ AS t3_r3, ‘enumerations‘.‘type‘ AS t3_r4, ‘enumerations‘.‘active‘ AS t3_r5, ‘enumerations‘.‘project_id‘ AS t3_r6, ‘enumerations‘.‘parent_id‘ AS t3_r7, ‘enumerations‘.‘position_name‘ AS t3_r8 FROM ‘issues‘ INNER JOIN ‘projects‘ ON ‘projects‘.‘id‘ = ‘issues‘.‘project_id‘ LEFT OUTER JOIN ‘issue_statuses‘ ON ‘issue_statuses‘.‘id‘ = ‘issues‘.‘status_id‘ LEFT OUTER JOIN ‘trackers‘ ON ‘trackers‘.‘id‘ = ‘issues‘.‘tracker_id‘ LEFT OUTER JOIN ‘enumerations‘ ON ‘enumerations‘.‘id‘ = ‘issues‘.‘priority_id‘ AND ‘enumerations‘.‘type‘ IN (’IssuePriority’) WHERE ‘issues‘.‘fixed_version_id‘ = 2 AND (((projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’issue_tracking’)) AND (((projects.is_public = TRUE AND projects.id NOT IN (SELECT project_id FROM members WHERE user_id IN (2,12))) AND ((issues.is_private = FALSE OR issues.author_id = 2 OR issues.assigned_to_id IN (2)))) OR (projects.id IN (1,5) AND (1=1)) OR (projects.id IN (2) AND ((issues.is_private = FALSE OR issues.author_id = 2 OR issues.assigned_to_id IN (2))))))) ORDER BY trackers.position, issues.id	SELECT ‘issues‘.‘id‘ AS ‘t0_r0‘, ‘issues‘.‘tracker_id‘ AS ‘t0_r1‘, ‘issues‘.‘project_id‘ AS ‘t0_r2‘, ‘issues‘.‘subject‘ AS ‘t0_r3‘, ‘issues‘.‘description‘ AS ‘t0_r4‘, ‘issues‘.‘due_date‘ AS ‘t0_r5‘, ‘issues‘.‘category_id‘ AS ‘t0_r6‘, ‘issues‘.‘status_id‘ AS ‘t0_r7‘, ‘issues‘.‘assigned_to_id‘ AS ‘t0_r8‘, ‘issues‘.‘priority_id‘ AS ‘t0_r9‘, ‘issues‘.‘fixed_version_id‘ AS ‘t0_r10‘, ‘issues‘.‘author_id‘ AS ‘t0_r11‘, ‘issues‘.‘lock_version‘ AS ‘t0_r12‘, ‘issues‘.‘created_on‘ AS ‘t0_r13‘, ‘issues‘.‘updated_on‘ AS ‘t0_r14‘, ‘issues‘.‘start_date‘ AS ‘t0_r15‘, ‘issues‘.‘done_ratio‘ AS ‘t0_r16‘, ‘issues‘.‘estimated_hours‘ AS ‘t0_r17‘, ‘issues‘.‘parent_id‘ AS ‘t0_r18‘, ‘issues‘.‘root_id‘ AS ‘t0_r19‘, ‘issues‘.‘lft‘ AS ‘t0_r20‘, ‘issues‘.‘rgt‘ AS ‘t0_r21‘, ‘issues‘.‘is_private‘ AS ‘t0_r22‘, ‘issues‘.‘closed_on‘ AS ‘t0_r23‘, ‘issue_statuses‘.‘id‘ AS ‘t1_r0‘, ‘issue_statuses‘.‘name‘ AS ‘t1_r1‘, ‘issue_statuses‘.‘is_closed‘ AS ‘t1_r2‘, ‘issue_statuses‘.‘position‘ AS ‘t1_r3‘, ‘issue_statuses‘.‘default_done_ratio‘ AS ‘t1_r4‘, ‘trackers‘.‘id‘ AS ‘t2_r0‘, ‘trackers‘.‘name‘ AS ‘t2_r1‘, ‘trackers‘.‘description‘ AS ‘t2_r2‘, ‘trackers‘.‘is_in_chlog‘ AS ‘t2_r3‘, ‘trackers‘.‘position‘ AS ‘t2_r4‘, ‘trackers‘.‘is_in_roadmap‘ AS ‘t2_r5‘, ‘trackers‘.‘fields_bits‘ AS ‘t2_r6‘, ‘trackers‘.‘default_status_id‘ AS ‘t2_r7‘, ‘enumerations‘.‘id‘ AS ‘t3_r0‘, ‘enumerations‘.‘name‘ AS ‘t3_r1‘, ‘enumerations‘.‘position‘ AS ‘t3_r2‘, ‘enumerations‘.‘is_default‘ AS ‘t3_r3‘, ‘enumerations‘.‘type‘ AS ‘t3_r4‘, ‘enumerations‘.‘active‘ AS ‘t3_r5‘, ‘enumerations‘.‘project_id‘ AS ‘t3_r6‘, ‘enumerations‘.‘parent_id‘ AS ‘t3_r7‘, ‘enumerations‘.‘position_name‘ AS ‘t3_r8‘ FROM ‘issues‘ AS ‘issues‘ INNER JOIN ‘projects‘ AS ‘projects‘ ON ‘issues‘.‘project_id‘ = ‘projects‘.‘id‘ LEFT JOIN ‘enumerations‘ AS ‘enumerations‘ ON ‘enumerations‘.‘id‘ = ‘issues‘.‘priority_id‘ AND ‘enumerations‘.‘type‘ IN (?) INNER JOIN ‘trackers‘ AS ‘trackers‘ ON ‘issues‘.‘tracker_id‘ = ‘trackers‘.‘id‘ INNER JOIN ‘issue_statuses‘ AS ‘issue_statuses‘ ON ‘issues‘.‘status_id‘ = ‘issue_statuses‘.‘id‘ WHERE EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘projects‘.‘id‘ AND ‘em‘.‘name‘ = ’issue_tracking’) AND ‘projects‘.‘status‘ <>9 AND (‘projects‘.‘is_public‘ = TRUE AND NOT ‘projects‘.‘id‘ IN (SELECT ‘project_id‘ FROM ‘members‘ WHERE ‘user_id‘ IN (?)) AND (‘issues‘.‘is_private‘ = FALSE OR ‘issues‘.‘author_id‘ = 2 OR ‘issues‘.‘assigned_to_id‘ IN (?)) OR ‘projects‘.‘id‘ IN (?) OR ‘projects‘.‘id‘ IN (?) AND (‘issues‘.‘is_private‘ = FALSE OR ‘issues‘.‘author_id‘ = 2 OR ‘issues‘.‘assigned_to_id‘ IN (?))) AND ‘issues‘.‘fixed_version_id‘ = 2 ORDER BY ‘trackers‘.‘position‘, ‘issues‘.‘id‘

redmine	1499	SELECT ‘issues‘.‘id‘ FROM ‘issues‘ INNER JOIN ‘projects‘ ON ‘projects‘.‘id‘ = ‘issues‘.‘project_id‘ INNER JOIN ‘issue_statuses‘ ON ‘issue_statuses‘.‘id‘ = ‘issues‘.‘status_id‘ WHERE (((projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’issue_tracking’)) AND (((projects.is_public = TRUE AND projects.id NOT IN (SELECT project_id FROM members WHERE user_id IN (6,13))) AND ((issues.is_private = FALSE)))))) AND ((issues.status_id IN (’5’))) ORDER BY issues.id ASC LIMIT 501	SELECT ‘issues‘.‘id‘ AS ‘id‘ FROM ‘projects‘ AS ‘projects‘ INNER JOIN ‘issues‘ AS ‘issues‘ ON ‘projects‘.‘id‘ = ‘issues‘.‘project_id‘ WHERE NOT ‘projects‘.‘id‘ IN (SELECT ‘project_id‘ FROM ‘members‘ WHERE ‘user_id‘ IN (?)) AND EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘projects‘.‘id‘ AND ‘em‘.‘name‘ = ’issue_tracking’) AND ‘projects‘.‘status‘ <>9 AND ‘projects‘.‘is_public‘ = TRUE AND ‘issues‘.‘status_id‘ IN (?) AND ‘issues‘.‘is_private‘ = FALSE ORDER BY ‘issues‘.‘id‘ ASC LIMIT 501
redmine	1510	SELECT COUNT(*) FROM ‘issues‘ INNER JOIN ‘projects‘ ON ‘projects‘.‘id‘ = ‘issues‘.‘project_id‘ INNER JOIN ‘issue_statuses‘ ON ‘issue_statuses‘.‘id‘ = ‘issues‘.‘status_id‘ WHERE (((projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’issue_tracking’)) AND (((projects.is_public = TRUE AND projects.id NOT IN (SELECT project_id FROM members WHERE user_id IN (6,13))) AND ((issues.is_private = FALSE)))))) AND ((issues.status_id IN (SELECT id FROM issue_statuses WHERE is_closed=FALSE)) AND (issues.assigned_to_id IS NULL OR issues.assigned_to_id NOT IN (SELECT DISTINCT members.user_id FROM members WHERE members.project_id = issues.project_id)) AND projects.id = 2017)	SELECT COUNT(*) FROM ‘projects‘ AS ‘projects‘ INNER JOIN ‘issues‘ AS ‘issues‘ ON ‘projects‘.‘id‘ = ‘issues‘.‘project_id‘ WHERE ‘projects‘.‘id‘ = 2017 AND EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘projects‘.‘id‘ AND ‘em‘.‘name‘ = ’issue_tracking’) AND ‘issues‘.‘status_id‘ IN (SELECT ‘issue_statuses‘.‘id‘ AS ‘id‘ FROM ‘issue_statuses‘ AS ‘issue_statuses‘ WHERE ‘issue_statuses‘.‘is_closed‘ = FALSE) AND ‘projects‘.‘status‘ <>9 AND ‘projects‘.‘is_public‘ = TRUE AND NOT ‘projects‘.‘id‘ IN (SELECT ‘project_id‘ FROM ‘members‘ WHERE ‘user_id‘ IN (?)) AND ‘issues‘.‘is_private‘ = FALSE AND (‘issues‘.‘assigned_to_id‘ IS NULL OR NOT ‘issues‘.‘assigned_to_id‘ IN (SELECT DISTINCT ‘members‘.‘user_id‘ FROM ‘members‘ WHERE ‘members‘.‘project_id‘ = ‘issues‘.‘project_id‘))
redmine	1554	SELECT wiki_content_versions.updated_on, wiki_content_versions.comments, wiki_content_versions.version, wiki_pages.title, wiki_content_versions.page_id, wiki_content_versions.author_id, wiki_content_versions.id FROM ‘wiki_content_versions‘ LEFT JOIN wiki_pages ON wiki_pages.id = wiki_content_versions.page_id LEFT JOIN wikis ON wikis.id = wiki_pages.wiki_id LEFT JOIN projects ON projects.id = wikis.project_id WHERE (wiki_content_versions.updated_on BETWEEN ’2020-04-15’ AND ’2020-05-16’) AND (((projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’wiki’)) AND ((projects.is_public = TRUE AND projects.id NOT IN (SELECT project_id FROM members WHERE user_id IN (2,12))) OR projects.id IN (1,5) OR projects.id IN (2))))	SELECT ‘wiki_content_versions‘.‘updated_on‘ AS ‘updated_on‘, ‘wiki_content_versions‘.‘comments‘ AS ‘comments‘, ‘wiki_content_versions‘.‘version‘ AS ‘version‘, ‘wiki_pages‘.‘title‘ AS ‘title‘, ‘wiki_content_versions‘.‘page_id‘ AS ‘page_id‘, ‘wiki_content_versions‘.‘author_id‘ AS ‘author_id‘, ‘wiki_content_versions‘.‘id‘ AS ‘id‘ FROM ‘wiki_content_versions‘ AS ‘wiki_content_versions‘ INNER JOIN ‘wiki_pages‘ AS ‘wiki_pages‘ ON ‘wiki_content_versions‘.‘page_id‘ = ‘wiki_pages‘.‘id‘ INNER JOIN ‘wikis‘ AS ‘wikis‘ ON ‘wiki_pages‘.‘wiki_id‘ = ‘wikis‘.‘id‘ INNER JOIN ‘projects‘ AS ‘projects‘ ON ‘wikis‘.‘project_id‘ = ‘projects‘.‘id‘ WHERE EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘projects‘.‘id‘ AND ‘em‘.‘name‘ = ’wiki’) AND ‘wiki_content_versions‘.‘updated_on‘ BETWEEN ’2020-04-15’ AND ’2020-05-16’ AND ‘projects‘.‘status‘ <>9 AND (‘projects‘.‘is_public‘ = TRUE AND NOT ‘projects‘.‘id‘ IN (SELECT ‘project_id‘ FROM ‘members‘ WHERE ‘user_id‘ IN (?)) OR ‘projects‘.‘id‘ IN (?) OR ‘projects‘.‘id‘ IN (?))
redmine	1565	SELECT wiki_content_versions.updated_on, wiki_content_versions.comments, wiki_content_versions.version, wiki_pages.title, wiki_content_versions.page_id, wiki_content_versions.author_id, wiki_content_versions.id FROM ‘wiki_content_versions‘ LEFT JOIN wiki_pages ON wiki_pages.id = wiki_content_versions.page_id LEFT JOIN wikis ON wikis.id = wiki_pages.wiki_id LEFT JOIN projects ON projects.id = wikis.project_id WHERE (wiki_content_versions.updated_on BETWEEN ’2020-04-15’ AND ’2020-05-16’) AND (((((projects.id = 1 OR (projects.lft >1 AND projects.rgt <10))) AND (projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’wiki’))) AND ((projects.is_public = TRUE AND projects.id NOT IN (SELECT project_id FROM members WHERE user_id IN (6,13))))))	SELECT ‘wiki_content_versions‘.‘updated_on‘ AS ‘updated_on‘, ‘wiki_content_versions‘.‘comments‘ AS ‘comments‘, ‘wiki_content_versions‘.‘version‘ AS ‘version‘, ‘wiki_pages‘.‘title‘ AS ‘title‘, ‘wiki_content_versions‘.‘page_id‘ AS ‘page_id‘, ‘wiki_content_versions‘.‘author_id‘ AS ‘author_id‘, ‘wiki_content_versions‘.‘id‘ AS ‘id‘ FROM ‘wiki_content_versions‘ AS ‘wiki_content_versions‘ INNER JOIN ‘wiki_pages‘ AS ‘wiki_pages‘ ON ‘wiki_content_versions‘.‘page_id‘ = ‘wiki_pages‘.‘id‘ INNER JOIN ‘wikis‘ AS ‘wikis‘ ON ‘wiki_pages‘.‘wiki_id‘ = ‘wikis‘.‘id‘ INNER JOIN ‘projects‘ AS ‘projects‘ ON ‘wikis‘.‘project_id‘ = ‘projects‘.‘id‘ WHERE NOT ‘projects‘.‘id‘ IN (SELECT ‘project_id‘ FROM ‘members‘ WHERE ‘user_id‘ IN (?)) AND EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘projects‘.‘id‘ AND ‘em‘.‘name‘ = ’wiki’) AND ‘wiki_content_versions‘.‘updated_on‘ BETWEEN ’2020-04-15’ AND ’2020-05-16’ AND ‘projects‘.‘status‘ <>9 AND ‘projects‘.‘is_public‘ = TRUE AND (‘projects‘.‘id‘ = 1 OR ‘projects‘.‘lft‘ >1 AND ‘projects‘.‘rgt‘ <10)

redmine	1580	SELECT DISTINCT ‘changesets‘.‘committed_on‘, ‘changesets‘.‘id‘ FROM ‘changesets‘ INNER JOIN ‘repositories‘ ON ‘repositories‘.‘id‘ = ‘changesets‘.‘repository_id‘ INNER JOIN ‘projects‘ ON ‘projects‘.‘id‘ = ‘repositories‘.‘project_id‘ WHERE (1=0) AND (((comments LIKE ’%%very first commit%%’))) ORDER BY ‘changesets‘.‘committed_on‘ DESC, ‘changesets‘.‘id‘ DESC	SELECT ‘changesets‘.‘committed_on‘ AS ‘committed_on‘, ‘changesets‘.‘id‘ AS ‘id‘ FROM ‘changesets‘ AS ‘changesets‘ WHERE ‘changesets‘.‘comments‘ LIKE ’%%very first commit%%’ ORDER BY ‘changesets‘.‘committed_on‘ DESC, ‘changesets‘.‘id‘ DESC
redmine	1608	SELECT COUNT(*) FROM ‘issues‘ INNER JOIN ‘projects‘ ON ‘projects‘.‘id‘ = ‘issues‘.‘project_id‘ INNER JOIN ‘issue_statuses‘ ON ‘issue_statuses‘.‘id‘ = ‘issues‘.‘status_id‘ WHERE (((projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’issue_tracking’)) AND (((projects.is_public = TRUE AND projects.id NOT IN (SELECT project_id FROM members WHERE user_id IN (6,13))) AND ((issues.is_private = FALSE))))))	SELECT COUNT(*) FROM ‘projects‘ AS ‘projects‘ INNER JOIN ‘issues‘ AS ‘issues‘ ON ‘projects‘.‘id‘ = ‘issues‘.‘project_id‘ WHERE NOT ‘issues‘.‘project_id‘ IN (SELECT ‘project_id‘ FROM ‘members‘ WHERE ‘user_id‘ IN (?)) AND EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘issues‘.‘project_id‘ AND ‘em‘.‘name‘ = ’issue_tracking’) AND ‘projects‘.‘status‘ <>9 AND ‘projects‘.‘is_public‘ = TRUE AND ‘issues‘.‘is_private‘ = FALSE
redmine	1648	SELECT ‘issues‘.‘id‘ FROM ‘issues‘ INNER JOIN ‘projects‘ ON ‘projects‘.‘id‘ = ‘issues‘.‘project_id‘ INNER JOIN ‘issue_statuses‘ ON ‘issue_statuses‘.‘id‘ = ‘issues‘.‘status_id‘ WHERE (((projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’issue_tracking’)) AND (((projects.is_public = TRUE AND projects.id NOT IN (SELECT project_id FROM members WHERE user_id IN (6,13))) AND ((issues.is_private = FALSE)))))) AND ((issues.status_id IN (SELECT id FROM issue_statuses WHERE is_closed=FALSE))) ORDER BY issues.id ASC LIMIT 501	SELECT ‘issues‘.‘id‘ AS ‘id‘ FROM ‘issues‘ AS ‘issues‘ INNER JOIN ‘projects‘ AS ‘projects‘ ON ‘issues‘.‘project_id‘ = ‘projects‘.‘id‘ WHERE EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘issues‘.‘project_id‘ AND ‘em‘.‘name‘ = ’issue_tracking’) AND NOT ‘projects‘.‘id‘ IN (SELECT ‘project_id‘ FROM ‘members‘ WHERE ‘user_id‘ IN (?)) AND ‘issues‘.‘status_id‘ IN (SELECT ‘issue_statuses‘.‘id‘ AS ‘id‘ FROM ‘issue_statuses‘ AS ‘issue_statuses‘ WHERE ‘issue_statuses‘.‘is_closed‘ = FALSE) AND ‘projects‘.‘status‘ <>9 AND ‘projects‘.‘is_public‘ = TRUE AND ‘issues‘.‘is_private‘ = FALSE ORDER BY ‘issues‘.‘id‘ ASC LIMIT 501
redmine	1654	SELECT DISTINCT ‘changesets‘.‘committed_on‘, ‘changesets‘.‘id‘ FROM ‘changesets‘ INNER JOIN ‘repositories‘ ON ‘repositories‘.‘id‘ = ‘changesets‘.‘repository_id‘ INNER JOIN ‘projects‘ ON ‘projects‘.‘id‘ = ‘repositories‘.‘project_id‘ WHERE (((projects.status <>9 AND EXISTS (SELECT 1 AS one FROM enabled_modules em WHERE em.project_id = projects.id AND em.name=’repository’)) AND ((projects.is_public = TRUE AND projects.id NOT IN (SELECT project_id FROM members WHERE user_id IN (2,12))) OR projects.id IN (1,5) OR projects.id IN (2)))) AND (repositories.project_id IN (1,2,5)) AND (((comments LIKE ’%recipe%’)) OR ((comments LIKE ’%subproject%’))) ORDER BY ‘changesets‘.‘committed_on‘ DESC, ‘changesets‘.‘id‘ DESC	SELECT DISTINCT ‘changesets‘.‘committed_on‘ AS ‘committed_on‘, ‘changesets‘.‘id‘ AS ‘id‘ FROM ‘changesets‘ AS ‘changesets‘ INNER JOIN ‘repositories‘ AS ‘repositories‘ ON ‘changesets‘.‘repository_id‘ = ‘repositories‘.‘id‘ INNER JOIN ‘projects‘ AS ‘projects‘ ON ‘repositories‘.‘project_id‘ = ‘projects‘.‘id‘ WHERE ‘projects‘.‘id‘ IN (?) AND EXISTS (SELECT 1 AS ‘one‘ FROM ‘enabled_modules‘ AS ‘em‘ WHERE ‘em‘.‘project_id‘ = ‘repositories‘.‘project_id‘ AND ‘em‘.‘name‘ = ’repository’) AND ‘projects‘.‘status‘ <>9 AND (‘projects‘.‘is_public‘ = TRUE AND NOT ‘projects‘.‘id‘ IN (SELECT ‘project_id‘ FROM ‘members‘ WHERE ‘user_id‘ IN (?)) OR ‘projects‘.‘id‘ IN (?) OR ‘projects‘.‘id‘ IN (?)) AND (‘changesets‘.‘comments‘ LIKE ’%recipe%’ OR ‘changesets‘.‘comments‘ LIKE ’%subproject%’) ORDER BY ‘changesets‘.‘committed_on‘ DESC, ‘changesets‘.‘id‘ DESC
refinerycms	42	SELECT COUNT(*) FROM ‘refinery_pages‘ LEFT OUTER JOIN ‘refinery_pages‘ ‘parent‘ ON ‘parent‘.‘id‘ = ‘refinery_pages‘.‘parent_id‘ WHERE (((‘refinery_pages‘.‘lft‘ IS NULL OR ‘refinery_pages‘.‘rgt‘ IS NULL) OR ‘refinery_pages‘.‘lft‘ >= ‘refinery_pages‘.‘rgt‘) OR ‘refinery_pages‘.‘parent_id‘ IS NOT NULL AND (‘refinery_pages‘.‘lft‘ <= ‘parent‘.‘lft‘ OR ‘refinery_pages‘.‘rgt‘ >= ‘parent‘.‘rgt‘))	SELECT COUNT(*) FROM ‘refinery_pages‘ AS ‘refinery_pages‘ INNER JOIN ‘refinery_pages‘ AS ‘parent‘ ON ‘refinery_pages‘.‘parent_id‘ = ‘parent‘.‘id‘ WHERE ‘refinery_pages‘.‘lft‘ IS NULL OR ‘refinery_pages‘.‘rgt‘ IS NULL OR ‘refinery_pages‘.‘lft‘ >= ‘refinery_pages‘.‘rgt‘ OR NOT ‘refinery_pages‘.‘parent_id‘ IS NULL AND (‘refinery_pages‘.‘lft‘ <= ‘parent‘.‘lft‘ OR ‘refinery_pages‘.‘rgt‘ >= ‘parent‘.‘rgt‘)
shopizer	5	select distinct group0_.GROUP_ID as GROUP_ID1_65_0_, permission2_.PERMISSION_ID as PERMISSI1_40_1_, group0_.DATE_CREATED as DATE_CRE2_65_0_, group0_.DATE_MODIFIED as DATE_MOD3_65_0_, group0_.UPDT_ID as UPDT_ID4_65_0_, group0_.GROUP_NAME as GROUP_NA5_65_0_, group0_.GROUP_TYPE as GROUP_TY6_65_0_, permission2_.DATE_CREATED as DATE_CRE2_40_1_, permission2_.DATE_MODIFIED as DATE_MOD3_40_1_, permission2_.UPDT_ID as UPDT_ID4_40_1_, permission2_.PERMISSION_NAME as PERMISSI5_40_1_, permission1_.GROUP_ID as GROUP_ID2_41_0__, permission1_.PERMISSION_ID as PERMISSI1_41_0__ from SM_GROUP group0_ left outer join PERMISSION_GROUP permission1_ on group0_.GROUP_ID=permission1_.GROUP_ID left outer join PERMISSION permission2_ on permission1_.PERMISSION_ID=permission2_.PERMISSION_ID where group0_.GROUP_TYPE=’ADMIN’	SELECT ‘group0_‘.‘group_id‘ AS ‘group_id1_65_0_‘, ‘permission2_‘.‘permission_id‘ AS ‘permissi1_40_1_‘, ‘group0_‘.‘date_created‘ AS ‘date_cre2_65_0_‘, ‘group0_‘.‘date_modified‘ AS ‘date_mod3_65_0_‘, ‘group0_‘.‘updt_id‘ AS ‘updt_id4_65_0_‘, ‘group0_‘.‘group_name‘ AS ‘group_na5_65_0_‘, ‘group0_‘.‘group_type‘ AS ‘group_ty6_65_0_‘, ‘permission2_‘.‘date_created‘ AS ‘date_cre2_40_1_‘, ‘permission2_‘.‘date_modified‘ AS ‘date_mod3_40_1_‘, ‘permission2_‘.‘updt_id‘ AS ‘updt_id4_40_1_‘, ‘permission2_‘.‘permission_name‘ AS ‘permissi5_40_1_‘, ‘permission1_‘.‘group_id‘ AS ‘group_id2_41_0__‘, ‘permission1_‘.‘permission_id‘ AS ‘permissi1_41_0__‘ FROM ‘sm_group‘ AS ‘group0_‘ LEFT JOIN ‘permission_group‘ AS ‘permission1_‘ ON ‘group0_‘.‘group_id‘ = ‘permission1_‘.‘group_id‘ LEFT JOIN ‘permission‘ AS ‘permission2_‘ ON ‘permission1_‘.‘permission_id‘ = ‘permission2_‘.‘permission_id‘ WHERE ‘group0_‘.‘group_type‘ = ’ADMIN’

shopizer	41	select count(distinct product0_.PRODUCT_ID) as col_0_0_ from PRODUCT product0_ inner join PRODUCT_DESCRIPTION descriptio1_ on product0_.PRODUCT_ID=descriptio1_.PRODUCT_ID inner join PRODUCT_CATEGORY categories2_ on product0_.PRODUCT_ID=categories2_.PRODUCT_ID inner join CATEGORY category3_ on categories2_.CATEGORY_ID=category3_.CATEGORY_ID where product0_.MERCHANT_ID=1 and descriptio1_.LANGUAGE_ID=1 and (category3_.CATEGORY_ID in (2)) and product0_.AVAILABLE=1 and product0_.DATE_AVAILABLE<='2019-10-21 21:11:10.902'	SELECT COUNT(DISTINCT ‘product0_‘.‘product_id‘) AS ‘col_0_0_‘ FROM ‘product_category‘ AS ‘categories2_‘ INNER JOIN ‘product_description‘ AS ‘descriptio1_‘ ON ‘categories2_‘.‘product_id‘ = ‘descriptio1_‘.‘product_id‘ INNER JOIN ‘product‘ AS ‘product0_‘ ON ‘categories2_‘.‘product_id‘ = ‘product0_‘.‘product_id‘ WHERE ‘categories2_‘.‘category_id‘ IN (?) AND ‘product0_‘.‘merchant_id‘ = 1 AND ‘product0_‘.‘date_available‘ <= '2019-10-21 21:11:10.902' AND ‘product0_‘.‘available‘ = 1 AND ‘descriptio1_‘.‘language_id‘ = 1
shopizer	59	select count(distinct product0_.PRODUCT_ID) as col_0_0_ from PRODUCT product0_ inner join PRODUCT_DESCRIPTION descriptio1_ on product0_.PRODUCT_ID=descriptio1_.PRODUCT_ID inner join PRODUCT_CATEGORY categories2_ on product0_.PRODUCT_ID=categories2_.PRODUCT_ID inner join CATEGORY category3_ on categories2_.CATEGORY_ID=category3_.CATEGORY_ID where product0_.MERCHANT_ID=1 and descriptio1_.LANGUAGE_ID=1 and (category3_.CATEGORY_ID in (1))	SELECT COUNT(DISTINCT ‘product0_‘.‘product_id‘) AS ‘col_0_0_‘ FROM ‘product_category‘ AS ‘categories2_‘ INNER JOIN ‘product‘ AS ‘product0_‘ ON ‘categories2_‘.‘product_id‘ = ‘product0_‘.‘product_id‘ INNER JOIN ‘product_description‘ AS ‘descriptio1_‘ ON ‘categories2_‘.‘product_id‘ = ‘descriptio1_‘.‘product_id‘ WHERE ‘categories2_‘.‘category_id‘ IN (?) AND ‘product0_‘.‘merchant_id‘ = 1 AND ‘descriptio1_‘.‘language_id‘ = 1
shopizer	67	select count(distinct product0_.PRODUCT_ID) as col_0_0_ from PRODUCT product0_ inner join PRODUCT_DESCRIPTION descriptio1_ on product0_.PRODUCT_ID=descriptio1_.PRODUCT_ID inner join PRODUCT_CATEGORY categories2_ on product0_.PRODUCT_ID=categories2_.PRODUCT_ID inner join CATEGORY category3_ on categories2_.CATEGORY_ID=category3_.CATEGORY_ID inner join MANUFACTURER manufactur4_ on product0_.MANUFACTURER_ID=manufactur4_.MANUFACTURER_ID where product0_.MERCHANT_ID=1 and descriptio1_.LANGUAGE_ID=1 and (category3_.CATEGORY_ID in (1)) and manufactur4_.MANUFACTURER_ID=1 and product0_.AVAILABLE=1 and product0_.DATE_AVAILABLE<='2019-10-21 21:17:32.7'	SELECT COUNT(DISTINCT ‘product0_‘.‘product_id‘) AS ‘col_0_0_‘ FROM ‘product_category‘ AS ‘categories2_‘ INNER JOIN ‘product‘ AS ‘product0_‘ ON ‘categories2_‘.‘product_id‘ = ‘product0_‘.‘product_id‘ INNER JOIN ‘product_description‘ AS ‘descriptio1_‘ ON ‘product0_‘.‘product_id‘ = ‘descriptio1_‘.‘product_id‘ WHERE ‘categories2_‘.‘category_id‘ IN (?) AND ‘product0_‘.‘manufacturer_id‘ = 1 AND ‘product0_‘.‘merchant_id‘ = 1 AND ‘product0_‘.‘date_available‘ <= '2019-10-21 21:17:32.7' AND ‘product0_‘.‘available‘ = 1 AND ‘descriptio1_‘.‘language_id‘ = 1
shopizer	72	select distinct user0_.USER_ID as USER_ID1_72_0_, group2_.GROUP_ID as GROUP_ID1_65_1_, merchantst3_.MERCHANT_ID as MERCHANT1_27_2_, language4_.LANGUAGE_ID as LANGUAGE1_21_3_, user0_.ACTIVE as ACTIVE2_72_0_, user0_.ADMIN_EMAIL as ADMIN_EM3_72_0_, user0_.ADMIN_NAME as ADMIN_NA4_72_0_, user0_.ADMIN_PASSWORD as ADMIN_PA5_72_0_, user0_.ADMIN_A1 as ADMIN_A6_72_0_, user0_.ADMIN_A2 as ADMIN_A7_72_0_, user0_.ADMIN_A3 as ADMIN_A8_72_0_, user0_.DATE_CREATED as DATE_CRE9_72_0_, user0_.DATE_MODIFIED as DATE_MO10_72_0_, user0_.UPDT_ID as UPDT_ID11_72_0_, user0_.LANGUAGE_ID as LANGUAG19_72_0_, user0_.ADMIN_FIRST_NAME as ADMIN_F12_72_0_, user0_.LAST_ACCESS as LAST_AC13_72_0_, user0_.ADMIN_LAST_NAME as ADMIN_L14_72_0_, user0_.LOGIN_ACCESS as LOGIN_A15_72_0_, user0_.MERCHANT_ID as MERCHAN20_72_0_, user0_.ADMIN_Q1 as ADMIN_Q16_72_0_, user0_.ADMIN_Q2 as ADMIN_Q17_72_0_, user0_.ADMIN_Q3 as ADMIN_Q18_72_0_, group2_.DATE_CREATED as DATE_CRE2_65_1_, group2_.DATE_MODIFIED as DATE_MOD3_65_1_, group2_.UPDT_ID as UPDT_ID4_65_1_, group2_.GROUP_NAME as GROUP_NA5_65_1_, group2_.GROUP_TYPE as GROUP_TY6_65_1_, groups1_.USER_ID as USER_ID1_73_0__, groups1_.GROUP_ID as GROUP_ID2_73_0__, merchantst3_.DATE_CREATED as DATE_CRE2_27_2_, merchantst3_.DATE_MODIFIED as DATE_MOD3_27_2_, merchantst3_.UPDT_ID as UPDT_ID4_27_2_, merchantst3_.STORE_CODE as STORE_CO5_27_2_, merchantst3_.CONTINUESHOPPINGURL as CONTINUE6_27_2_, merchantst3_.COUNTRY_ID as COUNTRY23_27_2_, merchantst3_.CURRENCY_ID as CURRENC24_27_2_, merchantst3_.CURRENCY_FORMAT_NATIONAL as CURRENCY7_27_2_, merchantst3_.LANGUAGE_ID as LANGUAG25_27_2_, merchantst3_.DOMAIN_NAME as DOMAIN_N8_27_2_, merchantst3_.IN_BUSINESS_SINCE as IN_BUSIN9_27_2_, merchantst3_.INVOICE_TEMPLATE as INVOICE10_27_2_, merchantst3_.SEIZEUNITCODE as SEIZEUN11_27_2_, merchantst3_.STORE_EMAIL as STORE_E12_27_2_, merchantst3_.STORE_LOGO as STORE_L13_27_2_, merchantst3_.STORE_TEMPLATE as STORE_T14_27_2_, merchantst3_.STORE_ADDRESS as STORE_A15_27_2_, merchantst3_.STORE_CITY as STORE_C16_27_2_, merchantst3_.STORE_NAME as STORE_N17_27_2_, merchantst3_.STORE_PHONE as STORE_P18_27_2_, merchantst3_.STORE_POSTAL_CODE as STORE_P19_27_2_, merchantst3_.STORE_STATE_PROV as STORE_S20_27_2_, merchantst3_.USE_CACHE as USE_CAC21_27_2_, merchantst3_.WEIGHTUNITCODE as WEIGHTU22_27_2_, merchantst3_.ZONE_ID as ZONE_ID26_27_2_, language4_.DATE_CREATED as DATE_CRE2_21_3_, language4_.DATE_MODIFIED as DATE_MOD3_21_3_, language4_.UPDT_ID as UPDT_ID4_21_3_, language4_.CODE as CODE5_21_3_, language4_.SORT_ORDER as SORT_ORD6_21_3_ from USER user0_ left outer join USER_GROUP groups1_ on user0_.USER_ID=groups1_.USER_ID left outer join SM_GROUP group2_ on groups1_.GROUP_ID=group2_.GROUP_ID inner join MERCHANT_STORE merchantst3_ on user0_.MERCHANT_ID=merchantst3_.MERCHANT_ID	SELECT ‘user0_‘.‘user_id‘ AS ‘user_id1_72_0_‘, ‘group2_‘.‘group_id‘ AS ‘group_id1_65_1_‘, ‘merchantst3_‘.‘merchant_id‘ AS ‘merchant1_27_2_‘, ‘language4_‘.‘language_id‘ AS ‘language1_21_3_‘, ‘user0_‘.‘active‘ AS ‘active2_72_0_‘, ‘user0_‘.‘admin_email‘ AS ‘admin_em3_72_0_‘, ‘user0_‘.‘admin_name‘ AS ‘admin_na4_72_0_‘, ‘user0_‘.‘admin_password‘ AS ‘admin_pa5_72_0_‘, ‘user0_‘.‘admin_a1‘ AS ‘admin_a6_72_0_‘, ‘user0_‘.‘admin_a2‘ AS ‘admin_a7_72_0_‘, ‘user0_‘.‘admin_a3‘ AS ‘admin_a8_72_0_‘, ‘user0_‘.‘date_created‘ AS ‘date_cre9_72_0_‘, ‘user0_‘.‘date_modified‘ AS ‘date_mo10_72_0_‘, ‘user0_‘.‘updt_id‘ AS ‘updt_id11_72_0_‘, ‘user0_‘.‘language_id‘ AS ‘languag19_72_0_‘, ‘user0_‘.‘admin_first_name‘ AS ‘admin_f12_72_0_‘, ‘user0_‘.‘last_access‘ AS ‘last_ac13_72_0_‘, ‘user0_‘.‘admin_last_name‘ AS ‘admin_l14_72_0_‘, ‘user0_‘.‘login_access‘ AS ‘login_a15_72_0_‘, ‘user0_‘.‘merchant_id‘ AS ‘merchan20_72_0_‘, ‘user0_‘.‘admin_q1‘ AS ‘admin_q16_72_0_‘, ‘user0_‘.‘admin_q2‘ AS ‘admin_q17_72_0_‘, ‘user0_‘.‘admin_q3‘ AS ‘admin_q18_72_0_‘, ‘group2_‘.‘date_created‘ AS ‘date_cre2_65_1_‘, ‘group2_‘.‘date_modified‘ AS ‘date_mod3_65_1_‘, ‘group2_‘.‘updt_id‘ AS ‘updt_id4_65_1_‘, ‘group2_‘.‘group_name‘ AS ‘group_na5_65_1_‘, ‘group2_‘.‘group_type‘ AS ‘group_ty6_65_1_‘, ‘groups1_‘.‘user_id‘ AS ‘user_id1_73_0__‘, ‘groups1_‘.‘group_id‘ AS ‘group_id2_73_0__‘, ‘merchantst3_‘.‘date_created‘ AS ‘date_cre2_27_2_‘, ‘merchantst3_‘.‘date_modified‘ AS ‘date_mod3_27_2_‘, ‘merchantst3_‘.‘updt_id‘ AS ‘updt_id4_27_2_‘, ‘merchantst3_‘.‘store_code‘ AS ‘store_co5_27_2_‘, ‘merchantst3_‘.‘continueshoppingurl‘ AS ‘continue6_27_2_‘, ‘merchantst3_‘.‘country_id‘ AS ‘country23_27_2_‘, ‘merchantst3_‘.‘currency_id‘ AS ‘currenc24_27_2_‘, ‘merchantst3_‘.‘currency_format_national‘ AS ‘currency7_27_2_‘, ‘merchantst3_‘.‘language_id‘ AS ‘languag25_27_2_‘, ‘merchantst3_‘.‘domain_name‘ AS ‘domain_n8_27_2_‘, ‘merchantst3_‘.‘in_business_since‘ AS ‘in_busin9_27_2_‘, ‘merchantst3_‘.‘invoice_template‘ AS ‘invoice10_27_2_‘, ‘merchantst3_‘.‘seizeunitcode‘ AS ‘seizeun11_27_2_‘, ‘merchantst3_‘.‘store_email‘ AS ‘store_e12_27_2_‘, ‘merchantst3_‘.‘store_logo‘ AS ‘store_l13_27_2_‘, ‘merchantst3_‘.‘store_template‘ AS ‘store_t14_27_2_‘, ‘merchantst3_‘.‘store_address‘ AS ‘store_a15_27_2_‘, ‘merchantst3_‘.‘store_city‘ AS ‘store_c16_27_2_‘, ‘merchantst3_‘.‘store_name‘ AS ‘store_n17_27_2_‘, ‘merchantst3_‘.‘store_phone‘ AS ‘store_p18_27_2_‘, ‘merchantst3_‘.‘store_postal_code‘ AS ‘store_p19_27_2_‘, ‘merchantst3_‘.‘store_state_prov‘ AS ‘store_s20_27_2_‘, ‘merchantst3_‘.‘use_cache‘ AS ‘use_cac21_27_2_‘, ‘merchantst3_‘.‘weightunitcode‘ AS ‘weightu22_27_2_‘, ‘merchantst3_‘.‘zone_id‘ AS ‘zone_id26_27_2_‘, ‘language4_‘.‘date_created‘ AS ‘date_cre2_21_3_‘, ‘language4_‘.‘date_modified‘ AS ‘date_mod3_21_3_‘, ‘language4_‘.‘updt_id‘ AS ‘updt_id4_21_3_‘, ‘language4_‘.‘code‘ AS ‘code5_21_3_‘, ‘language4_‘.‘sort_order‘ AS ‘sort_ord6_21_3_‘ FROM ‘user‘ AS ‘user0_‘ LEFT JOIN ‘user_group‘ AS ‘groups1_‘ ON ‘user0_‘.‘user_id‘ = ‘groups1_‘.‘user_id‘ LEFT JOIN ‘sm_group‘ AS ‘group2_‘ ON ‘groups1_‘.‘group_id‘ = ‘group2_‘.‘group_id‘ INNER JOIN ‘merchant_store‘ AS ‘merchantst3_‘ ON ‘user0_‘.‘merchant_id‘ = ‘merchantst3_‘.‘merchant_id‘ INNER JOIN ‘language‘ AS ‘language4_‘ ON ‘user0_‘.‘language_id‘ = ‘language4_‘.‘language_id‘ WHERE ‘user0_‘.‘admin_name‘ = 'admin'

shopizer	126	select distinct group0_.GROUP_ID as GROUP_ID1_65_0_, permission2_.PERMISSION_ID as PERMISSI1_40_1_, group0_.DATE_CREATED as DATE_CRE2_65_0_, group0_.DATE_MODIFIED as DATE_MOD3_65_0_, group0_.UPDT_ID as UPDT_ID4_65_0_, group0_.GROUP_NAME as GROUP_NA5_65_0_, group0_.GROUP_TYPE as GROUP_TY6_65_0_, permission2_.DATE_CREATED as DATE_CRE2_40_1_, permission2_.DATE_MODIFIED as DATE_MOD3_40_1_, permission2_.UPDT_ID as UPDT_ID4_40_1_, permission2_.PERMISSION_NAME as PERMISSI5_40_1_, permission1_.GROUP_ID as GROUP_ID2_41_0__, permission1_.PERMISSION_ID as PERMISSI1_41_0__ from SM_GROUP group0_ left outer join PERMISSION_GROUP permission1_ on group0_.GROUP_ID=permission1_.GROUP_ID left outer join PERMISSION permission2_ on permission1_.PERMISSION_ID=permission2_.PERMISSION_ID order by group0_.GROUP_ID	SELECT ‘group0_‘.‘group_id‘ AS ‘group_id1_65_0_‘, ‘permission2_‘.‘permission_id‘ AS ‘permissi1_40_1_‘, ‘group0_‘.‘date_created‘ AS ‘date_cre2_65_0_‘, ‘group0_‘.‘date_modified‘ AS ‘date_mod3_65_0_‘, ‘group0_‘.‘updt_id‘ AS ‘updt_id4_65_0_‘, ‘group0_‘.‘group_name‘ AS ‘group_na5_65_0_‘, ‘group0_‘.‘group_type‘ AS ‘group_ty6_65_0_‘, ‘permission2_‘.‘date_created‘ AS ‘date_cre2_40_1_‘, ‘permission2_‘.‘date_modified‘ AS ‘date_mod3_40_1_‘, ‘permission2_‘.‘updt_id‘ AS ‘updt_id4_40_1_‘, ‘permission2_‘.‘permission_name‘ AS ‘permissi5_40_1_‘, ‘permission1_‘.‘group_id‘ AS ‘group_id2_41_0__‘, ‘permission1_‘.‘permission_id‘ AS ‘permissi1_41_0__‘ FROM ‘sm_group‘ AS ‘group0_‘ LEFT JOIN ‘permission_group‘ AS ‘permission1_‘ ON ‘group0_‘.‘group_id‘ = ‘permission1_‘.‘group_id‘ LEFT JOIN ‘permission‘ AS ‘permission2_‘ ON ‘permission1_‘.‘permission_id‘ = ‘permission2_‘.‘permission_id‘ ORDER BY ‘group0_‘.‘group_id‘
solidus	162	SELECT DISTINCT ‘spree_zones‘.‘id‘ FROM ‘spree_zones‘ INNER JOIN ‘spree_zone_members‘ ON ‘spree_zone_members‘.‘zone_id‘ = ‘spree_zones‘.‘id‘ WHERE (‘spree_zone_members‘.‘zoneable_type‘ = ’Spree::State’ AND 1=0 OR ‘spree_zone_members‘.‘zoneable_type‘ = ’Spree::Country’ AND ‘spree_zone_members‘.‘zoneable_id‘ IN (29))	SELECT DISTINCT ‘spree_zone_members‘.‘zone_id‘ AS ‘zone_id‘ FROM ‘spree_zone_members‘ AS ‘spree_zone_members‘ WHERE ‘spree_zone_members‘.‘zoneable_type‘ = ’Spree::State’ OR ‘spree_zone_members‘.‘zoneable_type‘ = ’Spree::Country’ AND ‘spree_zone_members‘.‘zoneable_id‘ IN (?)
solidus	230	SELECT COUNT(*) FROM (SELECT 1 AS one FROM ‘spree_taxonomies‘ WHERE ‘spree_taxonomies‘.‘name‘ LIKE ’%style%’ ORDER BY ‘spree_taxonomies‘.‘position‘ ASC, name LIMIT 25 OFFSET 0) subquery_for_count	SELECT COUNT(*) FROM (SELECT 1 AS ‘one‘ FROM ‘spree_taxonomies‘ AS ‘spree_taxonomies‘ WHERE ‘spree_taxonomies‘.‘name‘ LIKE ’%style%’ LIMIT 25 OFFSET 0) AS ‘subquery_for_count‘
solidus	277	SELECT COUNT(*) FROM (SELECT 1 AS one FROM ‘spree_taxonomies‘ ORDER BY ‘spree_taxonomies‘.‘position‘ ASC, name LIMIT 25 OFFSET 0) subquery_for_count	SELECT COUNT(*) FROM (SELECT 1 AS ‘one‘ FROM ‘spree_taxonomies‘ AS ‘spree_taxonomies‘ LIMIT 25 OFFSET 0) AS ‘subquery_for_count‘
solidus	279	SELECT COUNT(*) FROM (SELECT 1 AS one FROM ‘spree_taxonomies‘ ORDER BY ‘spree_taxonomies‘.‘position‘ ASC, name LIMIT 1 OFFSET 0) subquery_for_count	SELECT COUNT(*) FROM (SELECT 1 AS ‘one‘ FROM ‘spree_taxonomies‘ AS ‘spree_taxonomies‘ LIMIT 1 OFFSET 0) AS ‘subquery_for_count‘
solidus	292	SELECT COUNT(count_column) FROM (SELECT DISTINCT ‘spree_products‘.‘id‘ AS count_column FROM ‘spree_products‘ INNER JOIN ‘spree_variants‘ ON ‘spree_variants‘.‘is_master‘ = TRUE AND ‘spree_variants‘.‘product_id‘ = ‘spree_products‘.‘id‘ INNER JOIN ‘spree_variants‘ ‘variants_including_masters_spree_products_join‘ ON ‘variants_including_masters_spree_products_join‘.‘deleted_at‘ IS NULL AND ‘variants_including_masters_spree_products_join‘.‘product_id‘ = ‘spree_products‘.‘id‘ INNER JOIN ‘spree_prices‘ ON ‘spree_prices‘.‘deleted_at‘ IS NULL AND ‘spree_prices‘.‘variant_id‘ = ‘variants_including_masters_spree_products_join‘.‘id‘ WHERE ‘spree_products‘.‘deleted_at‘ IS NULL AND EXISTS (SELECT ‘spree_prices‘.* FROM ‘spree_prices‘ WHERE ‘spree_prices‘.‘deleted_at‘ IS NULL AND ‘spree_variants‘.‘id‘ = ‘spree_prices‘.‘variant_id‘) AND (‘spree_products‘.available_on <= ’2020-05-16 05:22:46.685012’) AND ‘spree_prices‘.‘deleted_at‘ IS NULL AND ‘spree_prices‘.‘currency‘ = ’USD’ AND ‘spree_prices‘.‘country_iso‘ IS NULL LIMIT 12 OFFSET 0) subquery_for_count	SELECT COUNT(‘spree_products‘.‘id‘) FROM ‘spree_products‘ AS ‘spree_products‘ INNER JOIN ‘spree_variants‘ AS ‘spree_variants‘ ON ‘spree_products‘.‘id‘ = ‘spree_variants‘.‘product_id‘ INNER JOIN ‘spree_variants‘ AS ‘variants_including_masters_spree_products_join‘ ON ‘spree_products‘.‘id‘ = ‘variants_including_masters_spree_products_join‘.‘product_id‘ INNER JOIN ‘spree_prices‘ AS ‘spree_prices‘ ON ‘variants_including_masters_spree_products_join‘.‘id‘ = ‘spree_prices‘.‘variant_id‘ WHERE EXISTS (SELECT ‘spree_prices‘.‘id‘ AS ‘id‘, ‘spree_prices‘.‘variant_id‘ AS ‘variant_id‘, ‘spree_prices‘.‘amount‘ AS ‘amount‘, ‘spree_prices‘.‘currency‘ AS ‘currency‘, ‘spree_prices‘.‘deleted_at‘ AS ‘deleted_at‘, ‘spree_prices‘.‘created_at‘ AS ‘created_at‘, ‘spree_prices‘.‘updated_at‘ AS ‘updated_at‘, ‘spree_prices‘.‘country_iso‘ AS ‘country_iso‘ FROM ‘spree_prices‘ AS ‘spree_prices‘ WHERE ‘spree_variants‘.‘id‘ = ‘spree_prices‘.‘variant_id‘ AND ‘spree_prices‘.‘deleted_at‘ IS NULL) AND ‘variants_including_masters_spree_products_join‘.‘deleted_at‘ IS NULL AND ‘spree_variants‘.‘is_master‘ = TRUE AND ‘spree_products‘.‘deleted_at‘ IS NULL AND ‘spree_products‘.‘available_on‘ <= ’2020-05-16 05:22:46.685012’ AND ‘spree_prices‘.‘deleted_at‘ IS NULL AND ‘spree_prices‘.‘deleted_at‘ IS NULL AND ‘spree_prices‘.‘currency‘ = ’USD’ AND ‘spree_prices‘.‘country_iso‘ IS NULL LIMIT 12 OFFSET 0

solidus	294	SELECT COUNT(DISTINCT ‘spree_products‘.‘id‘) FROM ‘spree_products‘ INNER JOIN ‘spree_variants‘ ON ‘spree_variants‘.‘is_master‘ = TRUE AND ‘spree_variants‘.‘product_id‘ = ‘spree_products‘.‘id‘ INNER JOIN ‘spree_variants‘ ‘variants_including_masters_spree_products_join‘ ON ‘variants_including_masters_spree_products_join‘.‘deleted_at‘ IS NULL AND ‘variants_including_masters_spree_products_join‘.‘product_id‘ = ‘spree_products‘.‘id‘ INNER JOIN ‘spree_prices‘ ON ‘spree_prices‘.‘deleted_at‘ IS NULL AND ‘spree_prices‘.‘variant_id‘ = ‘variants_including_masters_spree_products_join‘.‘id‘ WHERE ‘spree_products‘.‘deleted_at‘ IS NULL AND EXISTS (SELECT ‘spree_prices‘.* FROM ‘spree_prices‘ WHERE ‘spree_prices‘.‘deleted_at‘ IS NULL AND ‘spree_variants‘.‘id‘ = ‘spree_prices‘.‘variant_id‘) AND (‘spree_products‘.available_on <= ’2020-05-16 05:22:46.685012’) AND ‘spree_prices‘.‘deleted_at‘ IS NULL AND ‘spree_prices‘.‘currency‘ = ’USD’ AND ‘spree_prices‘.‘country_iso‘ IS NULL	SELECT COUNT(DISTINCT ‘spree_products‘.‘id‘) FROM ‘spree_variants‘ AS ‘variants_including_masters_spree_products_join‘ INNER JOIN ‘spree_prices‘ AS ‘spree_prices‘ ON ‘variants_including_masters_spree_products_join‘.‘id‘ = ‘spree_prices‘.‘variant_id‘ INNER JOIN ‘spree_products‘ AS ‘spree_products‘ ON ‘variants_including_masters_spree_products_join‘.‘product_id‘ = ‘spree_products‘.‘id‘ INNER JOIN ‘spree_variants‘ AS ‘spree_variants‘ ON ‘variants_including_masters_spree_products_join‘.‘product_id‘ = ‘spree_variants‘.‘product_id‘ WHERE EXISTS (SELECT ‘spree_prices‘.‘id‘ AS ‘id‘, ‘spree_prices‘.‘variant_id‘ AS ‘variant_id‘, ‘spree_prices‘.‘amount‘ AS ‘amount‘, ‘spree_prices‘.‘currency‘ AS ‘currency‘, ‘spree_prices‘.‘deleted_at‘ AS ‘deleted_at‘, ‘spree_prices‘.‘created_at‘ AS ‘created_at‘, ‘spree_prices‘.‘updated_at‘ AS ‘updated_at‘, ‘spree_prices‘.‘country_iso‘ AS ‘country_iso‘ FROM ‘spree_prices‘ AS ‘spree_prices‘ WHERE ‘spree_variants‘.‘id‘ = ‘spree_prices‘.‘variant_id‘ AND ‘spree_prices‘.‘deleted_at‘ IS NULL) AND ‘variants_including_masters_spree_products_join‘.‘deleted_at‘ IS NULL AND ‘spree_variants‘.‘is_master‘ = TRUE AND ‘spree_products‘.‘deleted_at‘ IS NULL AND ‘spree_products‘.‘available_on‘ <= ’2020-05-16 05:22:46.685012’ AND ‘spree_prices‘.‘currency‘ = ’USD’ AND ‘spree_prices‘.‘country_iso‘ IS NULL AND ‘spree_prices‘.‘deleted_at‘ IS NULL
solidus	296	SELECT DISTINCT ‘spree_products‘.* FROM ‘spree_products‘ INNER JOIN ‘spree_variants‘ ON ‘spree_variants‘.‘is_master‘ = TRUE AND ‘spree_variants‘.‘product_id‘ = ‘spree_products‘.‘id‘ INNER JOIN ‘spree_variants‘ ‘variants_including_masters_spree_products_join‘ ON ‘variants_including_masters_spree_products_join‘.‘deleted_at‘ IS NULL AND ‘variants_including_masters_spree_products_join‘.‘product_id‘ = ‘spree_products‘.‘id‘ INNER JOIN ‘spree_prices‘ ON ‘spree_prices‘.‘deleted_at‘ IS NULL AND ‘spree_prices‘.‘variant_id‘ = ‘variants_including_masters_spree_products_join‘.‘id‘ WHERE ‘spree_products‘.‘deleted_at‘ IS NULL AND EXISTS (SELECT ‘spree_prices‘.* FROM ‘spree_prices‘ WHERE ‘spree_prices‘.‘deleted_at‘ IS NULL AND ‘spree_variants‘.‘id‘ = ‘spree_prices‘.‘variant_id‘) AND (‘spree_products‘.available_on <= ’2020-05-16 05:22:46.685012’) AND ‘spree_prices‘.‘deleted_at‘ IS NULL AND ‘spree_prices‘.‘currency‘ = ’USD’ AND ‘spree_prices‘.‘country_iso‘ IS NULL LIMIT 12 OFFSET 0	SELECT * FROM (SELECT DISTINCT ‘spree_products‘.‘id‘ AS ‘id‘, ‘spree_products‘.‘name‘ AS ‘name‘, ‘spree_products‘.‘description‘ AS ‘description‘, ‘spree_products‘.‘available_on‘ AS ‘available_on‘, ‘spree_products‘.‘deleted_at‘ AS ‘deleted_at‘, ‘spree_products‘.‘slug‘ AS ‘slug‘, ‘spree_products‘.‘meta_description‘ AS ‘meta_description‘, ‘spree_products‘.‘meta_keywords‘ AS ‘meta_keywords‘, ‘spree_products‘.‘tax_category_id‘ AS ‘tax_category_id‘, ‘spree_products‘.‘shipping_category_id‘ AS ‘shipping_category_id‘, ‘spree_products‘.‘created_at‘ AS ‘created_at‘, ‘spree_products‘.‘updated_at‘ AS ‘updated_at‘, ‘spree_products‘.‘promotionable‘ AS ‘promotionable‘, ‘spree_products‘.‘meta_title‘ AS ‘meta_title‘ FROM ‘spree_variants‘ AS ‘variants_including_masters_spree_products_join‘ INNER JOIN ‘spree_prices‘ AS ‘spree_prices‘ ON ‘variants_including_masters_spree_products_join‘.‘id‘ = ‘spree_prices‘.‘variant_id‘ INNER JOIN ‘spree_products‘ AS ‘spree_products‘ ON ‘variants_including_masters_spree_products_join‘.‘product_id‘ = ‘spree_products‘.‘id‘ INNER JOIN ‘spree_variants‘ AS ‘spree_variants‘ ON ‘spree_products‘.‘id‘ = ‘spree_variants‘.‘product_id‘ WHERE EXISTS (SELECT ‘spree_prices‘.‘id‘ AS ‘id‘, ‘spree_prices‘.‘variant_id‘ AS ‘variant_id‘, ‘spree_prices‘.‘amount‘ AS ‘amount‘, ‘spree_prices‘.‘currency‘ AS ‘currency‘, ‘spree_prices‘.‘deleted_at‘ AS ‘deleted_at‘, ‘spree_prices‘.‘created_at‘ AS ‘created_at‘, ‘spree_prices‘.‘updated_at‘ AS ‘updated_at‘, ‘spree_prices‘.‘country_iso‘ AS ‘country_iso‘ FROM ‘spree_prices‘ AS ‘spree_prices‘ WHERE ‘spree_variants‘.‘id‘ = ‘spree_prices‘.‘variant_id‘ AND ‘spree_prices‘.‘deleted_at‘ IS NULL) AND ‘variants_including_masters_spree_products_join‘.‘deleted_at‘ IS NULL AND ‘spree_variants‘.‘is_master‘ = TRUE AND ‘spree_prices‘.‘currency‘ = ’USD’ AND ‘spree_prices‘.‘country_iso‘ IS NULL AND ‘spree_prices‘.‘deleted_at‘ IS NULL) AS ‘sub_0‘ WHERE ‘sub_0‘.‘deleted_at‘ IS NULL AND ‘sub_0‘.‘available_on‘ <= ’2020-05-16 05:22:46.685012’ LIMIT 12 OFFSET 0

solidus	332	SELECT ‘spree_stock_locations‘.‘id‘ FROM ‘spree_stock_locations‘ INNER JOIN ‘spree_stock_items‘ ON ‘spree_stock_locations‘.‘id‘ = ‘spree_stock_items‘.‘stock_location_id‘ WHERE ‘spree_stock_items‘.‘deleted_at‘ IS NULL AND ‘spree_stock_items‘.‘variant_id‘ = 2789	SELECT ‘spree_stock_items‘.‘stock_location_id‘ AS ‘stock_location_id‘ FROM ‘spree_stock_items‘ AS ‘spree_stock_items‘ WHERE ‘spree_stock_items‘.‘variant_id‘ = 2789 AND ‘spree_stock_items‘.‘deleted_at‘ IS NULL
solidus	382	SELECT ‘spree_orders‘.* FROM ‘spree_orders‘ LEFT OUTER JOIN ‘spree_users‘ ON ‘spree_users‘.‘id‘ = ‘spree_orders‘.‘user_id‘ WHERE ‘spree_users‘.‘id‘ = 339 ORDER BY ‘spree_orders‘.‘total‘ DESC LIMIT 10 OFFSET 0	SELECT * FROM ‘spree_orders‘ AS ‘spree_orders‘ WHERE ‘spree_orders‘.‘user_id‘ = 339 ORDER BY ‘spree_orders‘.‘total‘ DESC LIMIT 10 OFFSET 0
solidus	391	SELECT ‘spree_orders‘.* FROM ‘spree_orders‘ LEFT OUTER JOIN ‘spree_users‘ ON ‘spree_users‘.‘id‘ = ‘spree_orders‘.‘user_id‘ WHERE ‘spree_users‘.‘id‘ = 333 ORDER BY spree_orders.completed_at IS NULL, ‘spree_orders‘.‘completed_at‘ DESC, ‘spree_orders‘.‘created_at‘ DESC LIMIT 10 OFFSET 0	SELECT * FROM ‘spree_orders‘ AS ‘spree_orders‘ WHERE ‘spree_orders‘.‘user_id‘ = 333 ORDER BY ‘spree_orders‘.‘completed_at‘ IS NULL, ‘spree_orders‘.‘completed_at‘ DESC, ‘spree_orders‘.‘created_at‘ DESC LIMIT 10 OFFSET 0
solidus	393	SELECT ‘spree_orders‘.* FROM ‘spree_orders‘ LEFT OUTER JOIN ‘spree_users‘ ON ‘spree_users‘.‘id‘ = ‘spree_orders‘.‘user_id‘ WHERE ‘spree_users‘.‘id‘ = 339 ORDER BY ‘spree_orders‘.‘total‘ ASC LIMIT 10 OFFSET 0	SELECT * FROM ‘spree_orders‘ AS ‘spree_orders‘ WHERE ‘spree_orders‘.‘user_id‘ = 339 ORDER BY ‘spree_orders‘.‘total‘ ASC LIMIT 10 OFFSET 0
solidus	409	SELECT COUNT(*) FROM (SELECT 1 AS one FROM ‘spree_payments‘ WHERE ‘spree_payments‘.‘order_id‘ = 183 ORDER BY ‘spree_payments‘.‘created_at‘ ASC LIMIT 25 OFFSET 0) subquery_for_count	SELECT COUNT(*) FROM (SELECT 1 AS ‘one‘ FROM ‘spree_payments‘ AS ‘spree_payments‘ WHERE ‘spree_payments‘.‘order_id‘ = 183 LIMIT 25 OFFSET 0) AS ‘subquery_for_count‘
solidus	430	SELECT COUNT(*) FROM (SELECT 1 AS one FROM ‘spree_stock_locations‘ WHERE ‘spree_stock_locations‘.‘active‘ = TRUE ORDER BY name ASC LIMIT 25 OFFSET 0) subquery_for_count	SELECT COUNT(*) FROM (SELECT 1 AS ‘one‘ FROM ‘spree_stock_locations‘ AS ‘spree_stock_locations‘ WHERE ‘spree_stock_locations‘.‘active‘ = TRUE LIMIT 25 OFFSET 0) AS ‘subquery_for_count‘
solidus	434	SELECT COUNT(*) FROM (SELECT 1 AS one FROM ‘spree_stock_locations‘ ORDER BY name ASC LIMIT 25 OFFSET 0) subquery_for_count	SELECT COUNT(*) FROM (SELECT 1 AS ‘one‘ FROM ‘spree_stock_locations‘ AS ‘spree_stock_locations‘ LIMIT 25 OFFSET 0) AS ‘subquery_for_count‘
solidus	476	SELECT DISTINCT ‘spree_products‘.* FROM ‘spree_products‘ INNER JOIN ‘spree_variants‘ ON ‘spree_variants‘.‘is_master‘ = TRUE AND ‘spree_variants‘.‘product_id‘ = ‘spree_products‘.‘id‘ INNER JOIN ‘spree_variants‘ ‘variants_including_masters_spree_products_join‘ ON ‘variants_including_masters_spree_products_join‘.‘deleted_at‘ IS NULL AND ‘variants_including_masters_spree_products_join‘.‘product_id‘ = ‘spree_products‘.‘id‘ INNER JOIN ‘spree_prices‘ ON ‘spree_prices‘.‘deleted_at‘ IS NULL AND ‘spree_prices‘.‘variant_id‘ = ‘variants_including_masters_spree_products_join‘.‘id‘ WHERE ‘spree_products‘.‘deleted_at‘ IS NULL AND EXISTS (SELECT ‘spree_prices‘.* FROM ‘spree_prices‘ WHERE ‘spree_prices‘.‘deleted_at‘ IS NULL AND ‘spree_variants‘.‘id‘ = ‘spree_prices‘.‘variant_id‘) AND (‘spree_products‘.available_on <= ’2020-05-16 05:23:15.200456’) AND (‘spree_products‘.‘name‘ LIKE ’%shirt%’ OR ‘spree_products‘.‘description‘ LIKE ’%shirt%’) AND ‘spree_prices‘.‘deleted_at‘ IS NULL AND ‘spree_prices‘.‘currency‘ = ’USD’ AND ‘spree_prices‘.‘country_iso‘ IS NULL LIMIT 1 OFFSET 0	SELECT * FROM (SELECT DISTINCT ‘spree_products‘.‘id‘ AS ‘id‘, ‘spree_products‘.‘name‘ AS ‘name‘, ‘spree_products‘.‘description‘ AS ‘description‘, ‘spree_products‘.‘available_on‘ AS ‘available_on‘, ‘spree_products‘.‘deleted_at‘ AS ‘deleted_at‘, ‘spree_products‘.‘slug‘ AS ‘slug‘, ‘spree_products‘.‘meta_description‘ AS ‘meta_description‘, ‘spree_products‘.‘meta_keywords‘ AS ‘meta_keywords‘, ‘spree_products‘.‘tax_category_id‘ AS ‘tax_category_id‘, ‘spree_products‘.‘shipping_category_id‘ AS ‘shipping_category_id‘, ‘spree_products‘.‘created_at‘ AS ‘created_at‘, ‘spree_products‘.‘updated_at‘ AS ‘updated_at‘, ‘spree_products‘.‘promotionable‘ AS ‘promotionable‘, ‘spree_products‘.‘meta_title‘ AS ‘meta_title‘ FROM ‘spree_variants‘ AS ‘variants_including_masters_spree_products_join‘ INNER JOIN ‘spree_prices‘ AS ‘spree_prices‘ ON ‘variants_including_masters_spree_products_join‘.‘id‘ = ‘spree_prices‘.‘variant_id‘ INNER JOIN ‘spree_products‘ AS ‘spree_products‘ ON ‘variants_including_masters_spree_products_join‘.‘product_id‘ = ‘spree_products‘.‘id‘ INNER JOIN ‘spree_variants‘ AS ‘spree_variants‘ ON ‘variants_including_masters_spree_products_join‘.‘product_id‘ = ‘spree_variants‘.‘product_id‘ WHERE EXISTS (SELECT ‘spree_prices‘.‘id‘ AS ‘id‘, ‘spree_prices‘.‘variant_id‘ AS ‘variant_id‘, ‘spree_prices‘.‘amount‘ AS ‘amount‘, ‘spree_prices‘.‘currency‘ AS ‘currency‘, ‘spree_prices‘.‘deleted_at‘ AS ‘deleted_at‘, ‘spree_prices‘.‘created_at‘ AS ‘created_at‘, ‘spree_prices‘.‘updated_at‘ AS ‘updated_at‘, ‘spree_prices‘.‘country_iso‘ AS ‘country_iso‘ FROM ‘spree_prices‘ AS ‘spree_prices‘ WHERE ‘spree_variants‘.‘id‘ = ‘spree_prices‘.‘variant_id‘ AND ‘spree_prices‘.‘deleted_at‘ IS NULL) AND ‘variants_including_masters_spree_products_join‘.‘deleted_at‘ IS NULL AND ‘spree_variants‘.‘is_master‘ = TRUE AND ‘spree_prices‘.‘currency‘ = ’USD’ AND ‘spree_prices‘.‘country_iso‘ IS NULL AND ‘spree_prices‘.‘deleted_at‘ IS NULL) AS ‘sub_0‘ WHERE (‘sub_0‘.‘name‘ LIKE ’%shirt%’ OR ‘sub_0‘.‘description‘ LIKE ’%shirt%’) AND ‘sub_0‘.‘deleted_at‘ IS NULL AND ‘sub_0‘.‘available_on‘ <= ’2020-05-16 05:23:15.200456’ LIMIT 1 OFFSET 0

solidus	489	SELECT COUNT(*) FROM (SELECT 1 AS one FROM ‘spree_product_properties‘ WHERE ‘spree_product_properties‘.‘product_id‘ = 429 ORDER BY ‘spree_product_properties‘.‘position‘ ASC LIMIT 25 OFFSET 0) subquery_for_count	SELECT COUNT(*) FROM (SELECT 1 AS ‘one‘ FROM ‘spree_product_properties‘ AS ‘spree_product_properties‘ WHERE ‘spree_product_properties‘.‘product_id‘ = 429 LIMIT 25 OFFSET 0) AS ‘subquery_for_count‘
solidus	500	SELECT COUNT(DISTINCT ‘spree_products‘.‘id‘) FROM ‘spree_products‘ INNER JOIN ‘spree_variants‘ ON ‘spree_variants‘.‘is_master‘ = TRUE AND ‘spree_variants‘.‘product_id‘ = ‘spree_products‘.‘id‘ INNER JOIN ‘spree_variants‘ ‘variants_including_masters_spree_products_join‘ ON ‘variants_including_masters_spree_products_join‘.‘deleted_at‘ IS NULL AND ‘variants_including_masters_spree_products_join‘.‘product_id‘ = ‘spree_products‘.‘id‘ INNER JOIN ‘spree_prices‘ ON ‘spree_prices‘.‘deleted_at‘ IS NULL AND ‘spree_prices‘.‘variant_id‘ = ‘variants_including_masters_spree_products_join‘.‘id‘ WHERE ‘spree_products‘.‘deleted_at‘ IS NULL AND EXISTS (SELECT ‘spree_prices‘.* FROM ‘spree_prices‘ WHERE ‘spree_prices‘.‘deleted_at‘ IS NULL AND ‘spree_variants‘.‘id‘ = ‘spree_prices‘.‘variant_id‘) AND (‘spree_products‘.available_on <= ’2020-05-16 05:23:15.214379’) AND (‘spree_products‘.‘name‘ LIKE ’%shirt%’ OR ‘spree_products‘.‘description‘ LIKE ’%shirt%’) AND ‘spree_prices‘.‘deleted_at‘ IS NULL AND ‘spree_prices‘.‘currency‘ = ’USD’ AND ‘spree_prices‘.‘country_iso‘ IS NULL	SELECT COUNT(DISTINCT ‘spree_products‘.‘id‘) FROM ‘spree_variants‘ AS ‘variants_including_masters_spree_products_join‘ INNER JOIN ‘spree_prices‘ AS ‘spree_prices‘ ON ‘variants_including_masters_spree_products_join‘.‘id‘ = ‘spree_prices‘.‘variant_id‘ INNER JOIN ‘spree_products‘ AS ‘spree_products‘ ON ‘variants_including_masters_spree_products_join‘.‘product_id‘ = ‘spree_products‘.‘id‘ INNER JOIN ‘spree_variants‘ AS ‘spree_variants‘ ON ‘variants_including_masters_spree_products_join‘.‘product_id‘ = ‘spree_variants‘.‘product_id‘ WHERE EXISTS (SELECT ‘spree_prices‘.‘id‘ AS ‘id‘, ‘spree_prices‘.‘variant_id‘ AS ‘variant_id‘, ‘spree_prices‘.‘amount‘ AS ‘amount‘, ‘spree_prices‘.‘currency‘ AS ‘currency‘, ‘spree_prices‘.‘deleted_at‘ AS ‘deleted_at‘, ‘spree_prices‘.‘created_at‘ AS ‘created_at‘, ‘spree_prices‘.‘updated_at‘ AS ‘updated_at‘, ‘spree_prices‘.‘country_iso‘ AS ‘country_iso‘ FROM ‘spree_prices‘ AS ‘spree_prices‘ WHERE ‘spree_variants‘.‘id‘ = ‘spree_prices‘.‘variant_id‘ AND ‘spree_prices‘.‘deleted_at‘ IS NULL) AND ‘variants_including_masters_spree_products_join‘.‘deleted_at‘ IS NULL AND ‘spree_variants‘.‘is_master‘ = TRUE AND (‘spree_products‘.‘name‘ LIKE ’%shirt%’ OR ‘spree_products‘.‘description‘ LIKE ’%shirt%’) AND ‘spree_products‘.‘deleted_at‘ IS NULL AND ‘spree_products‘.‘available_on‘ <= ’2020-05-16 05:23:15.214379’ AND ‘spree_prices‘.‘currency‘ = ’USD’ AND ‘spree_prices‘.‘country_iso‘ IS NULL AND ‘spree_prices‘.‘deleted_at‘ IS NULL
solidus	523	SELECT COUNT(*) FROM (SELECT 1 AS one FROM ‘spree_promotion_codes‘ WHERE ‘spree_promotion_codes‘.‘promotion_id‘ = 114 ORDER BY ‘spree_promotion_codes‘.‘value‘ ASC LIMIT 50 OFFSET 0) subquery_for_count	SELECT COUNT(*) FROM (SELECT 1 AS ‘one‘ FROM ‘spree_promotion_codes‘ AS ‘spree_promotion_codes‘ WHERE ‘spree_promotion_codes‘.‘promotion_id‘ = 114 LIMIT 50 OFFSET 0) AS ‘subquery_for_count‘
solidus	560	SELECT ‘spree_orders‘.* FROM ‘spree_orders‘ LEFT OUTER JOIN ‘spree_users‘ ON ‘spree_users‘.‘id‘ = ‘spree_orders‘.‘user_id‘ WHERE ‘spree_users‘.‘id‘ = 297 LIMIT 10 OFFSET 0	SELECT * FROM ‘spree_orders‘ AS ‘spree_orders‘ WHERE ‘spree_orders‘.‘user_id‘ = 297 LIMIT 10 OFFSET 0
solidus	621	SELECT COUNT(*) FROM (SELECT DISTINCT ‘spree_products_taxons‘.‘position‘ FROM ‘spree_products‘ INNER JOIN ‘spree_variants‘ ON ‘spree_variants‘.‘is_master‘ = TRUE AND ‘spree_variants‘.‘product_id‘ = ‘spree_products‘.‘id‘ INNER JOIN ‘spree_prices‘ ON ‘spree_prices‘.‘currency‘ = ’USD’ AND ‘spree_prices‘.‘country_iso‘ IS NULL AND ‘spree_prices‘.‘variant_id‘ = ‘spree_variants‘.‘id‘ INNER JOIN ‘spree_variants‘ ‘variants_including_masters_spree_products_join‘ ON ‘variants_including_masters_spree_products_join‘.‘deleted_at‘ IS NULL AND ‘variants_including_masters_spree_products_join‘.‘product_id‘ = ‘spree_products‘.‘id‘ INNER JOIN ‘spree_prices‘ ‘prices_spree_products‘ ON ‘prices_spree_products‘.‘deleted_at‘ IS NULL AND ‘prices_spree_products‘.‘variant_id‘ = ‘variants_including_masters_spree_products_join‘.‘id‘ LEFT OUTER JOIN ‘spree_products_taxons‘ ON ‘spree_products_taxons‘.‘product_id‘ = ‘spree_products‘.‘id‘ WHERE ‘spree_products‘.‘deleted_at‘ IS NULL AND EXISTS (SELECT ‘spree_prices‘.* FROM ‘spree_prices‘ WHERE ‘spree_prices‘.‘deleted_at‘ IS NULL AND ‘spree_variants‘.‘id‘ = ‘spree_prices‘.‘variant_id‘) AND (‘spree_products‘.available_on <= ’2020-05-16 05:23:22.762367’) AND ‘spree_products_taxons‘.‘taxon_id‘ = 280 AND (‘spree_prices‘.‘amount‘ BETWEEN 18.0 AND 20.0 OR ‘spree_prices‘.‘amount‘ >= 20.0) AND ‘spree_prices‘.‘deleted_at‘ IS NULL AND ‘spree_prices‘.‘currency‘ = ’USD’ AND ‘spree_prices‘.‘country_iso‘ IS NULL) subquery_for_count	SELECT COUNT(*) FROM (SELECT DISTINCT ‘spree_products_taxons‘.‘position‘ AS ‘position‘ FROM ‘spree_products‘ AS ‘spree_products‘ INNER JOIN ‘spree_variants‘ AS ‘spree_variants‘ ON ‘spree_products‘.‘id‘ = ‘spree_variants‘.‘product_id‘ INNER JOIN ‘spree_variants‘ AS ‘variants_including_masters_spree_products_join‘ ON ‘spree_products‘.‘id‘ = ‘variants_including_masters_spree_products_join‘.‘product_id‘ INNER JOIN ‘spree_products_taxons‘ AS ‘spree_products_taxons‘ ON ‘spree_products‘.‘id‘ = ‘spree_products_taxons‘.‘product_id‘ INNER JOIN ‘spree_prices‘ AS ‘spree_prices‘ ON ‘variants_including_masters_spree_products_join‘.‘id‘ = ‘spree_prices‘.‘variant_id‘ INNER JOIN ‘spree_prices‘ AS ‘prices_spree_products‘ ON ‘spree_prices‘.‘variant_id‘ = ‘prices_spree_products‘.‘variant_id‘ WHERE EXISTS (SELECT ‘spree_prices‘.‘id‘ AS ‘id‘, ‘spree_prices‘.‘variant_id‘ AS ‘variant_id‘, ‘spree_prices‘.‘amount‘ AS ‘amount‘, ‘spree_prices‘.‘currency‘ AS ‘currency‘, ‘spree_prices‘.‘deleted_at‘ AS ‘deleted_at‘, ‘spree_prices‘.‘created_at‘ AS ‘created_at‘, ‘spree_prices‘.‘updated_at‘ AS ‘updated_at‘, ‘spree_prices‘.‘country_iso‘ AS ‘country_iso‘ FROM ‘spree_prices‘ AS ‘spree_prices‘ WHERE ‘spree_variants‘.‘id‘ = ‘spree_prices‘.‘variant_id‘ AND ‘spree_prices‘.‘deleted_at‘ IS NULL) AND ‘variants_including_masters_spree_products_join‘.‘deleted_at‘ IS NULL AND ‘spree_variants‘.‘is_master‘ = TRUE AND ‘spree_products_taxons‘.‘taxon_id‘ = 280 AND ‘spree_products‘.‘deleted_at‘ IS NULL AND ‘spree_products‘.‘available_on‘ <= ’2020-05-16 05:23:22.762367’ AND (‘spree_prices‘.‘amount‘ BETWEEN 18.0 AND 20.0 OR ‘spree_prices‘.‘amount‘ >= 20.0) AND ‘spree_prices‘.‘country_iso‘ IS NULL AND ‘prices_spree_products‘.‘deleted_at‘ IS NULL AND ‘spree_prices‘.‘currency‘ = ’USD’) AS ‘subquery_for_count‘

solidus	622	SELECT DISTINCT ‘spree_products_taxons‘.‘position‘ FROM ‘spree_products‘ INNER JOIN ‘spree_variants‘ ON ‘spree_variants‘.‘is_master‘ = TRUE AND ‘spree_variants‘.‘product_id‘ = ‘spree_products‘.‘id‘ INNER JOIN ‘spree_prices‘ ON ‘spree_prices‘.‘currency‘ = ’USD’ AND ‘spree_prices‘.‘country_iso‘ IS NULL AND ‘spree_prices‘.‘variant_id‘ = ‘spree_variants‘.‘id‘ INNER JOIN ‘spree_variants‘ ‘variants_including_masters_spree_products_join‘ ON ‘variants_including_masters_spree_products_join‘.‘deleted_at‘ IS NULL AND ‘variants_including_masters_spree_products_join‘.‘product_id‘ = ‘spree_products‘.‘id‘ INNER JOIN ‘spree_prices‘ ‘prices_spree_products‘ ON ‘prices_spree_products‘.‘deleted_at‘ IS NULL AND ‘prices_spree_products‘.‘variant_id‘ = ‘variants_including_masters_spree_products_join‘.‘id‘ LEFT OUTER JOIN ‘spree_products_taxons‘ ON ‘spree_products_taxons‘.‘product_id‘ = ‘spree_products‘.‘id‘ WHERE ‘spree_products‘.‘deleted_at‘ IS NULL AND EXISTS (SELECT ‘spree_prices‘.* FROM ‘spree_prices‘ WHERE ‘spree_prices‘.‘deleted_at‘ IS NULL AND ‘spree_variants‘.‘id‘ = ‘spree_prices‘.‘variant_id‘) AND (‘spree_products‘.available_on <= ’2020-05-16 05:23:22.762367’) AND ‘spree_products_taxons‘.‘taxon_id‘ = 280 AND (‘spree_prices‘.‘amount‘ BETWEEN 18.0 AND 20.0 OR ‘spree_prices‘.‘amount‘ >= 20.0) AND ‘spree_prices‘.‘deleted_at‘ IS NULL AND ‘spree_prices‘.‘currency‘ = ’USD’ AND ‘spree_prices‘.‘country_iso‘ IS NULL LIMIT 1 OFFSET 0	SELECT DISTINCT ‘spree_products_taxons‘.‘position‘ AS ‘position‘ FROM ‘spree_products‘ AS ‘spree_products‘ INNER JOIN ‘spree_variants‘ AS ‘spree_variants‘ ON ‘spree_products‘.‘id‘ = ‘spree_variants‘.‘product_id‘ INNER JOIN ‘spree_variants‘ AS ‘variants_including_masters_spree_products_join‘ ON ‘spree_products‘.‘id‘ = ‘variants_including_masters_spree_products_join‘.‘product_id‘ INNER JOIN ‘spree_products_taxons‘ AS ‘spree_products_taxons‘ ON ‘spree_products‘.‘id‘ = ‘spree_products_taxons‘.‘product_id‘ INNER JOIN ‘spree_prices‘ AS ‘spree_prices‘ ON ‘variants_including_masters_spree_products_join‘.‘id‘ = ‘spree_prices‘.‘variant_id‘ INNER JOIN ‘spree_prices‘ AS ‘prices_spree_products‘ ON ‘spree_prices‘.‘variant_id‘ = ‘prices_spree_products‘.‘variant_id‘ WHERE EXISTS (SELECT ‘spree_prices‘.‘id‘ AS ‘id‘, ‘spree_prices‘.‘variant_id‘ AS ‘variant_id‘, ‘spree_prices‘.‘amount‘ AS ‘amount‘, ‘spree_prices‘.‘currency‘ AS ‘currency‘, ‘spree_prices‘.‘deleted_at‘ AS ‘deleted_at‘, ‘spree_prices‘.‘created_at‘ AS ‘created_at‘, ‘spree_prices‘.‘updated_at‘ AS ‘updated_at‘, ‘spree_prices‘.‘country_iso‘ AS ‘country_iso‘ FROM ‘spree_prices‘ AS ‘spree_prices‘ WHERE ‘spree_variants‘.‘id‘ = ‘spree_prices‘.‘variant_id‘ AND ‘spree_prices‘.‘deleted_at‘ IS NULL) AND ‘variants_including_masters_spree_products_join‘.‘deleted_at‘ IS NULL AND ‘spree_variants‘.‘is_master‘ = TRUE AND ‘spree_products_taxons‘.‘taxon_id‘ = 280 AND ‘spree_products‘.‘deleted_at‘ IS NULL AND ‘spree_products‘.‘available_on‘ <= ’2020-05-16 05:23:22.762367’ AND (‘spree_prices‘.‘amount‘ BETWEEN 18.0 AND 20.0 OR ‘spree_prices‘.‘amount‘ >= 20.0) AND ‘spree_prices‘.‘country_iso‘ IS NULL AND ‘prices_spree_products‘.‘deleted_at‘ IS NULL AND ‘spree_prices‘.‘currency‘ = ’USD’ LIMIT 1 OFFSET 0
solidus	623	SELECT DISTINCT ‘spree_products_taxons‘.‘position‘ AS alias_0, ‘spree_products‘.‘id‘ FROM ‘spree_products‘ INNER JOIN ‘spree_variants‘ ON ‘spree_variants‘.‘is_master‘ = TRUE AND ‘spree_variants‘.‘product_id‘ = ‘spree_products‘.‘id‘ INNER JOIN ‘spree_prices‘ ON ‘spree_prices‘.‘currency‘ = ’USD’ AND ‘spree_prices‘.‘country_iso‘ IS NULL AND ‘spree_prices‘.‘variant_id‘ = ‘spree_variants‘.‘id‘ INNER JOIN ‘spree_variants‘ ‘variants_including_masters_spree_products_join‘ ON ‘variants_including_masters_spree_products_join‘.‘deleted_at‘ IS NULL AND ‘variants_including_masters_spree_products_join‘.‘product_id‘ = ‘spree_products‘.‘id‘ INNER JOIN ‘spree_prices‘ ‘prices_spree_products‘ ON ‘prices_spree_products‘.‘deleted_at‘ IS NULL AND ‘prices_spree_products‘.‘variant_id‘ = ‘variants_including_masters_spree_products_join‘.‘id‘ LEFT OUTER JOIN ‘spree_products_taxons‘ ON ‘spree_products_taxons‘.‘product_id‘ = ‘spree_products‘.‘id‘ WHERE ‘spree_products‘.‘deleted_at‘ IS NULL AND EXISTS (SELECT ‘spree_prices‘.* FROM ‘spree_prices‘ WHERE ‘spree_prices‘.‘deleted_at‘ IS NULL AND ‘spree_variants‘.‘id‘ = ‘spree_prices‘.‘variant_id‘) AND (‘spree_products‘.available_on <= ’2020-05-16 05:23:22.762367’) AND ‘spree_products_taxons‘.‘taxon_id‘ = 280 AND (‘spree_prices‘.‘amount‘ BETWEEN 18.0 AND 20.0 OR ‘spree_prices‘.‘amount‘ >= 20.0) AND ‘spree_prices‘.‘deleted_at‘ IS NULL AND ‘spree_prices‘.‘currency‘ = ’USD’ AND ‘spree_prices‘.‘country_iso‘ IS NULL ORDER BY ‘spree_products_taxons‘.‘position‘ ASC LIMIT 12 OFFSET 0	SELECT DISTINCT ‘spree_products_taxons‘.‘position‘ AS ‘alias_0‘, ‘spree_products‘.‘id‘ AS ‘id‘ FROM ‘spree_products‘ AS ‘spree_products‘ INNER JOIN ‘spree_variants‘ AS ‘spree_variants‘ ON ‘spree_products‘.‘id‘ = ‘spree_variants‘.‘product_id‘ INNER JOIN ‘spree_prices‘ AS ‘prices_spree_products‘ ON ‘spree_variants‘.‘id‘ = ‘prices_spree_products‘.‘variant_id‘ INNER JOIN ‘spree_prices‘ AS ‘spree_prices‘ ON ‘prices_spree_products‘.‘variant_id‘ = ‘spree_prices‘.‘variant_id‘ INNER JOIN ‘spree_variants‘ AS ‘variants_including_masters_spree_products_join‘ ON ‘spree_products‘.‘id‘ = ‘variants_including_masters_spree_products_join‘.‘product_id‘ INNER JOIN ‘spree_products_taxons‘ AS ‘spree_products_taxons‘ ON ‘spree_products‘.‘id‘ = ‘spree_products_taxons‘.‘product_id‘ WHERE EXISTS (SELECT ‘spree_prices‘.‘id‘ AS ‘id‘, ‘spree_prices‘.‘variant_id‘ AS ‘variant_id‘, ‘spree_prices‘.‘amount‘ AS ‘amount‘, ‘spree_prices‘.‘currency‘ AS ‘currency‘, ‘spree_prices‘.‘deleted_at‘ AS ‘deleted_at‘, ‘spree_prices‘.‘created_at‘ AS ‘created_at‘, ‘spree_prices‘.‘updated_at‘ AS ‘updated_at‘, ‘spree_prices‘.‘country_iso‘ AS ‘country_iso‘ FROM ‘spree_prices‘ AS ‘spree_prices‘ WHERE ‘variants_including_masters_spree_products_join‘.‘id‘ = ‘spree_prices‘.‘variant_id‘ AND ‘spree_prices‘.‘deleted_at‘ IS NULL) AND ‘variants_including_masters_spree_products_join‘.‘deleted_at‘ IS NULL AND ‘spree_variants‘.‘is_master‘ = TRUE AND ‘spree_products_taxons‘.‘taxon_id‘ = 280 AND ‘spree_products‘.‘deleted_at‘ IS NULL AND ‘spree_products‘.‘available_on‘ <= ’2020-05-16 05:23:22.762367’ AND (‘spree_prices‘.‘amount‘ BETWEEN 18.0 AND 20.0 OR ‘spree_prices‘.‘amount‘ >= 20.0) AND ‘spree_prices‘.‘country_iso‘ IS NULL AND ‘prices_spree_products‘.‘deleted_at‘ IS NULL AND ‘spree_prices‘.‘currency‘ = ’USD’ ORDER BY ‘spree_products_taxons‘.‘position‘ ASC LIMIT 12 OFFSET 0

solidus	629	SELECT COUNT(*) FROM (SELECT DISTINCT ‘spree_products_taxons‘.‘position‘ FROM ‘spree_products‘ INNER JOIN ‘spree_variants‘ ON ‘spree_variants‘.‘is_master‘ = TRUE AND ‘spree_variants‘.‘product_id‘ = ‘spree_products‘.‘id‘ INNER JOIN ‘spree_prices‘ ON ‘spree_prices‘.‘currency‘ = ’USD’ AND ‘spree_prices‘.‘country_iso‘ IS NULL AND ‘spree_prices‘.‘variant_id‘ = ‘spree_variants‘.‘id‘ INNER JOIN ‘spree_variants‘ ‘variants_including_masters_spree_products_join‘ ON ‘variants_including_masters_spree_products_join‘.‘deleted_at‘ IS NULL AND ‘variants_including_masters_spree_products_join‘.‘product_id‘ = ‘spree_products‘.‘id‘ INNER JOIN ‘spree_prices‘ ‘prices_spree_products‘ ON ‘prices_spree_products‘.‘deleted_at‘ IS NULL AND ‘prices_spree_products‘.‘variant_id‘ = ‘variants_including_masters_spree_products_join‘.‘id‘ LEFT OUTER JOIN ‘spree_products_taxons‘ ON ‘spree_products_taxons‘.‘product_id‘ = ‘spree_products‘.‘id‘ WHERE ‘spree_products‘.‘deleted_at‘ IS NULL AND EXISTS (SELECT ‘spree_prices‘.* FROM ‘spree_prices‘ WHERE ‘spree_prices‘.‘deleted_at‘ IS NULL AND ‘spree_variants‘.‘id‘ = ‘spree_prices‘.‘variant_id‘) AND (‘spree_products‘.available_on <= ’2020-05-16 05:23:24.103686’) AND ‘spree_products_taxons‘.‘taxon_id‘ = 294 AND ‘spree_prices‘.‘amount‘ BETWEEN 15.0 AND 18.0 AND ‘spree_prices‘.‘deleted_at‘ IS NULL AND ‘spree_prices‘.‘currency‘ = ’USD’ AND ‘spree_prices‘.‘country_iso‘ IS NULL) subquery_for_count	SELECT COUNT(*) FROM (SELECT DISTINCT ‘spree_products_taxons‘.‘position‘ AS ‘position‘ FROM ‘spree_products‘ AS ‘spree_products‘ INNER JOIN ‘spree_variants‘ AS ‘spree_variants‘ ON ‘spree_products‘.‘id‘ = ‘spree_variants‘.‘product_id‘ INNER JOIN ‘spree_variants‘ AS ‘variants_including_masters_spree_products_join‘ ON ‘spree_products‘.‘id‘ = ‘variants_including_masters_spree_products_join‘.‘product_id‘ INNER JOIN ‘spree_products_taxons‘ AS ‘spree_products_taxons‘ ON ‘spree_products‘.‘id‘ = ‘spree_products_taxons‘.‘product_id‘ INNER JOIN ‘spree_prices‘ AS ‘spree_prices‘ ON ‘spree_variants‘.‘id‘ = ‘spree_prices‘.‘variant_id‘ INNER JOIN ‘spree_prices‘ AS ‘prices_spree_products‘ ON ‘spree_prices‘.‘variant_id‘ = ‘prices_spree_products‘.‘variant_id‘ WHERE EXISTS (SELECT ‘spree_prices‘.‘id‘ AS ‘id‘, ‘spree_prices‘.‘variant_id‘ AS ‘variant_id‘, ‘spree_prices‘.‘amount‘ AS ‘amount‘, ‘spree_prices‘.‘currency‘ AS ‘currency‘, ‘spree_prices‘.‘deleted_at‘ AS ‘deleted_at‘, ‘spree_prices‘.‘created_at‘ AS ‘created_at‘, ‘spree_prices‘.‘updated_at‘ AS ‘updated_at‘, ‘spree_prices‘.‘country_iso‘ AS ‘country_iso‘ FROM ‘spree_prices‘ AS ‘spree_prices‘ WHERE ‘variants_including_masters_spree_products_join‘.‘id‘ = ‘spree_prices‘.‘variant_id‘ AND ‘spree_prices‘.‘deleted_at‘ IS NULL) AND ‘variants_including_masters_spree_products_join‘.‘deleted_at‘ IS NULL AND ‘spree_variants‘.‘is_master‘ = TRUE AND ‘spree_products_taxons‘.‘taxon_id‘ = 294 AND ‘spree_products‘.‘deleted_at‘ IS NULL AND ‘spree_products‘.‘available_on‘ <= ’2020-05-16 05:23:24.103686’ AND ‘spree_prices‘.‘amount‘ BETWEEN 15.0 AND 18.0 AND ‘spree_prices‘.‘country_iso‘ IS NULL AND ‘spree_prices‘.‘currency‘ = ’USD’ AND ‘prices_spree_products‘.‘deleted_at‘ IS NULL) AS ‘subquery_for_count‘
solidus	631	SELECT DISTINCT ‘spree_products_taxons‘.‘position‘ AS alias_0, ‘spree_products‘.‘id‘ FROM ‘spree_products‘ INNER JOIN ‘spree_variants‘ ON ‘spree_variants‘.‘is_master‘ = TRUE AND ‘spree_variants‘.‘product_id‘ = ‘spree_products‘.‘id‘ INNER JOIN ‘spree_prices‘ ON ‘spree_prices‘.‘currency‘ = ’USD’ AND ‘spree_prices‘.‘country_iso‘ IS NULL AND ‘spree_prices‘.‘variant_id‘ = ‘spree_variants‘.‘id‘ INNER JOIN ‘spree_variants‘ ‘variants_including_masters_spree_products_join‘ ON ‘variants_including_masters_spree_products_join‘.‘deleted_at‘ IS NULL AND ‘variants_including_masters_spree_products_join‘.‘product_id‘ = ‘spree_products‘.‘id‘ INNER JOIN ‘spree_prices‘ ‘prices_spree_products‘ ON ‘prices_spree_products‘.‘deleted_at‘ IS NULL AND ‘prices_spree_products‘.‘variant_id‘ = ‘variants_including_masters_spree_products_join‘.‘id‘ LEFT OUTER JOIN ‘spree_products_taxons‘ ON ‘spree_products_taxons‘.‘product_id‘ = ‘spree_products‘.‘id‘ WHERE ‘spree_products‘.‘deleted_at‘ IS NULL AND EXISTS (SELECT ‘spree_prices‘.* FROM ‘spree_prices‘ WHERE ‘spree_prices‘.‘deleted_at‘ IS NULL AND ‘spree_variants‘.‘id‘ = ‘spree_prices‘.‘variant_id‘) AND (‘spree_products‘.available_on <= ’2020-05-16 05:23:24.103686’) AND ‘spree_products_taxons‘.‘taxon_id‘ = 294 AND ‘spree_prices‘.‘amount‘ BETWEEN 15.0 AND 18.0 AND ‘spree_prices‘.‘deleted_at‘ IS NULL AND ‘spree_prices‘.‘currency‘ = ’USD’ AND ‘spree_prices‘.‘country_iso‘ IS NULL ORDER BY ‘spree_products_taxons‘.‘position‘ ASC LIMIT 12 OFFSET 0	SELECT DISTINCT ‘spree_products_taxons‘.‘position‘ AS ‘alias_0‘, ‘spree_products‘.‘id‘ AS ‘id‘ FROM ‘spree_products‘ AS ‘spree_products‘ INNER JOIN ‘spree_variants‘ AS ‘spree_variants‘ ON ‘spree_products‘.‘id‘ = ‘spree_variants‘.‘product_id‘ INNER JOIN ‘spree_prices‘ AS ‘spree_prices‘ ON ‘spree_variants‘.‘id‘ = ‘spree_prices‘.‘variant_id‘ INNER JOIN ‘spree_prices‘ AS ‘prices_spree_products‘ ON ‘spree_variants‘.‘id‘ = ‘prices_spree_products‘.‘variant_id‘ INNER JOIN ‘spree_products_taxons‘ AS ‘spree_products_taxons‘ ON ‘spree_products‘.‘id‘ = ‘spree_products_taxons‘.‘product_id‘ INNER JOIN ‘spree_variants‘ AS ‘variants_including_masters_spree_products_join‘ ON ‘spree_products_taxons‘.‘product_id‘ = ‘variants_including_masters_spree_products_join‘.‘product_id‘ WHERE EXISTS (SELECT ‘spree_prices‘.‘id‘ AS ‘id‘, ‘spree_prices‘.‘variant_id‘ AS ‘variant_id‘, ‘spree_prices‘.‘amount‘ AS ‘amount‘, ‘spree_prices‘.‘currency‘ AS ‘currency‘, ‘spree_prices‘.‘deleted_at‘ AS ‘deleted_at‘, ‘spree_prices‘.‘created_at‘ AS ‘created_at‘, ‘spree_prices‘.‘updated_at‘ AS ‘updated_at‘, ‘spree_prices‘.‘country_iso‘ AS ‘country_iso‘ FROM ‘spree_prices‘ AS ‘spree_prices‘ WHERE ‘variants_including_masters_spree_products_join‘.‘id‘ = ‘spree_prices‘.‘variant_id‘ AND ‘spree_prices‘.‘deleted_at‘ IS NULL) AND ‘variants_including_masters_spree_products_join‘.‘deleted_at‘ IS NULL AND ‘spree_variants‘.‘is_master‘ = TRUE AND ‘spree_products_taxons‘.‘taxon_id‘ = 294 AND ‘spree_products‘.‘deleted_at‘ IS NULL AND ‘spree_products‘.‘available_on‘ <= ’2020-05-16 05:23:24.103686’ AND ‘spree_prices‘.‘amount‘ BETWEEN 15.0 AND 18.0 AND ‘prices_spree_products‘.‘deleted_at‘ IS NULL AND ‘spree_prices‘.‘currency‘ = ’USD’ AND ‘spree_prices‘.‘country_iso‘ IS NULL ORDER BY ‘spree_products_taxons‘.‘position‘ ASC LIMIT 12 OFFSET 0

solidus	633	SELECT COUNT(*) FROM (SELECT DISTINCT ‘spree_products_taxons‘.‘position‘ FROM ‘spree_products‘ INNER JOIN ‘spree_variants‘ ON ‘spree_variants‘.‘is_master‘ = TRUE AND ‘spree_variants‘.‘product_id‘ = ‘spree_products‘.‘id‘ INNER JOIN ‘spree_prices‘ ON ‘spree_prices‘.‘currency‘ = ’USD’ AND ‘spree_prices‘.‘country_iso‘ IS NULL AND ‘spree_prices‘.‘variant_id‘ = ‘spree_variants‘.‘id‘ INNER JOIN ‘spree_variants‘ ‘variants_including_masters_spree_products_join‘ ON ‘variants_including_masters_spree_products_join‘.‘deleted_at‘ IS NULL AND ‘variants_including_masters_spree_products_join‘.‘product_id‘ = ‘spree_products‘.‘id‘ INNER JOIN ‘spree_prices‘ ‘prices_spree_products‘ ON ‘prices_spree_products‘.‘deleted_at‘ IS NULL AND ‘prices_spree_products‘.‘variant_id‘ = ‘variants_including_masters_spree_products_join‘.‘id‘ LEFT OUTER JOIN ‘spree_products_taxons‘ ON ‘spree_products_taxons‘.‘product_id‘ = ‘spree_products‘.‘id‘ WHERE ‘spree_products‘.‘deleted_at‘ IS NULL AND EXISTS (SELECT ‘spree_prices‘.* FROM ‘spree_prices‘ WHERE ‘spree_prices‘.‘deleted_at‘ IS NULL AND ‘spree_variants‘.‘id‘ = ‘spree_prices‘.‘variant_id‘) AND (‘spree_products‘.available_on <= ’2020-05-16 05:23:13.956474’) AND ‘spree_products_taxons‘.‘taxon_id‘ = 182 AND ‘spree_prices‘.‘amount‘ <= 10.0 AND ‘spree_prices‘.‘deleted_at‘ IS NULL AND ‘spree_prices‘.‘currency‘ = ’USD’ AND ‘spree_prices‘.‘country_iso‘ IS NULL) subquery_for_count	SELECT COUNT(*) FROM (SELECT DISTINCT ‘spree_products_taxons‘.‘position‘ AS ‘position‘ FROM ‘spree_products‘ AS ‘spree_products‘ INNER JOIN ‘spree_variants‘ AS ‘spree_variants‘ ON ‘spree_products‘.‘id‘ = ‘spree_variants‘.‘product_id‘ INNER JOIN ‘spree_prices‘ AS ‘prices_spree_products‘ ON ‘spree_variants‘.‘id‘ = ‘prices_spree_products‘.‘variant_id‘ INNER JOIN ‘spree_prices‘ AS ‘spree_prices‘ ON ‘spree_variants‘.‘id‘ = ‘spree_prices‘.‘variant_id‘ INNER JOIN ‘spree_variants‘ AS ‘variants_including_masters_spree_products_join‘ ON ‘spree_products‘.‘id‘ = ‘variants_including_masters_spree_products_join‘.‘product_id‘ INNER JOIN ‘spree_products_taxons‘ AS ‘spree_products_taxons‘ ON ‘spree_products‘.‘id‘ = ‘spree_products_taxons‘.‘product_id‘ WHERE EXISTS (SELECT ‘spree_prices‘.‘id‘ AS ‘id‘, ‘spree_prices‘.‘variant_id‘ AS ‘variant_id‘, ‘spree_prices‘.‘amount‘ AS ‘amount‘, ‘spree_prices‘.‘currency‘ AS ‘currency‘, ‘spree_prices‘.‘deleted_at‘ AS ‘deleted_at‘, ‘spree_prices‘.‘created_at‘ AS ‘created_at‘, ‘spree_prices‘.‘updated_at‘ AS ‘updated_at‘, ‘spree_prices‘.‘country_iso‘ AS ‘country_iso‘ FROM ‘spree_prices‘ AS ‘spree_prices‘ WHERE ‘variants_including_masters_spree_products_join‘.‘id‘ = ‘spree_prices‘.‘variant_id‘ AND ‘spree_prices‘.‘deleted_at‘ IS NULL) AND ‘variants_including_masters_spree_products_join‘.‘deleted_at‘ IS NULL AND ‘spree_variants‘.‘is_master‘ = TRUE AND ‘spree_products_taxons‘.‘taxon_id‘ = 182 AND ‘spree_products‘.‘deleted_at‘ IS NULL AND ‘spree_products‘.‘available_on‘ <= ’2020-05-16 05:23:13.956474’ AND ‘spree_prices‘.‘amount‘ <= 10.0 AND ‘spree_prices‘.‘country_iso‘ IS NULL AND ‘spree_prices‘.‘currency‘ = ’USD’ AND ‘prices_spree_products‘.‘deleted_at‘ IS NULL) AS ‘subquery_for_count‘
solidus	636	SELECT DISTINCT ‘spree_products_taxons‘.‘position‘ AS alias_0, ‘spree_products‘.‘id‘ FROM ‘spree_products‘ INNER JOIN ‘spree_variants‘ ON ‘spree_variants‘.‘is_master‘ = TRUE AND ‘spree_variants‘.‘product_id‘ = ‘spree_products‘.‘id‘ INNER JOIN ‘spree_prices‘ ON ‘spree_prices‘.‘currency‘ = ’USD’ AND ‘spree_prices‘.‘country_iso‘ IS NULL AND ‘spree_prices‘.‘variant_id‘ = ‘spree_variants‘.‘id‘ INNER JOIN ‘spree_variants‘ ‘variants_including_masters_spree_products_join‘ ON ‘variants_including_masters_spree_products_join‘.‘deleted_at‘ IS NULL AND ‘variants_including_masters_spree_products_join‘.‘product_id‘ = ‘spree_products‘.‘id‘ INNER JOIN ‘spree_prices‘ ‘prices_spree_products‘ ON ‘prices_spree_products‘.‘deleted_at‘ IS NULL AND ‘prices_spree_products‘.‘variant_id‘ = ‘variants_including_masters_spree_products_join‘.‘id‘ LEFT OUTER JOIN ‘spree_products_taxons‘ ON ‘spree_products_taxons‘.‘product_id‘ = ‘spree_products‘.‘id‘ WHERE ‘spree_products‘.‘deleted_at‘ IS NULL AND EXISTS (SELECT ‘spree_prices‘.* FROM ‘spree_prices‘ WHERE ‘spree_prices‘.‘deleted_at‘ IS NULL AND ‘spree_variants‘.‘id‘ = ‘spree_prices‘.‘variant_id‘) AND (‘spree_products‘.available_on <= ’2020-05-16 05:23:18.933826’) AND ‘spree_products_taxons‘.‘taxon_id‘ = 238 AND ‘spree_prices‘.‘amount‘ BETWEEN 15.0 AND 18.0 AND ‘spree_prices‘.‘deleted_at‘ IS NULL AND ‘spree_prices‘.‘currency‘ = ’USD’ AND ‘spree_prices‘.‘country_iso‘ IS NULL ORDER BY ‘spree_products_taxons‘.‘position‘ ASC LIMIT 2 OFFSET 2	SELECT DISTINCT ‘spree_products_taxons‘.‘position‘ AS ‘alias_0‘, ‘spree_products‘.‘id‘ AS ‘id‘ FROM ‘spree_products‘ AS ‘spree_products‘ INNER JOIN ‘spree_variants‘ AS ‘spree_variants‘ ON ‘spree_products‘.‘id‘ = ‘spree_variants‘.‘product_id‘ INNER JOIN ‘spree_variants‘ AS ‘variants_including_masters_spree_products_join‘ ON ‘spree_products‘.‘id‘ = ‘variants_including_masters_spree_products_join‘.‘product_id‘ INNER JOIN ‘spree_products_taxons‘ AS ‘spree_products_taxons‘ ON ‘spree_products‘.‘id‘ = ‘spree_products_taxons‘.‘product_id‘ INNER JOIN ‘spree_prices‘ AS ‘prices_spree_products‘ ON ‘spree_variants‘.‘id‘ = ‘prices_spree_products‘.‘variant_id‘ INNER JOIN ‘spree_prices‘ AS ‘spree_prices‘ ON ‘spree_variants‘.‘id‘ = ‘spree_prices‘.‘variant_id‘ WHERE EXISTS (SELECT ‘spree_prices‘.‘id‘ AS ‘id‘, ‘spree_prices‘.‘variant_id‘ AS ‘variant_id‘, ‘spree_prices‘.‘amount‘ AS ‘amount‘, ‘spree_prices‘.‘currency‘ AS ‘currency‘, ‘spree_prices‘.‘deleted_at‘ AS ‘deleted_at‘, ‘spree_prices‘.‘created_at‘ AS ‘created_at‘, ‘spree_prices‘.‘updated_at‘ AS ‘updated_at‘, ‘spree_prices‘.‘country_iso‘ AS ‘country_iso‘ FROM ‘spree_prices‘ AS ‘spree_prices‘ WHERE ‘variants_including_masters_spree_products_join‘.‘id‘ = ‘spree_prices‘.‘variant_id‘ AND ‘spree_prices‘.‘deleted_at‘ IS NULL) AND ‘variants_including_masters_spree_products_join‘.‘deleted_at‘ IS NULL AND ‘spree_variants‘.‘is_master‘ = TRUE AND ‘spree_products_taxons‘.‘taxon_id‘ = 238 AND ‘spree_products‘.‘deleted_at‘ IS NULL AND ‘spree_products‘.‘available_on‘ <= ’2020-05-16 05:23:18.933826’ AND ‘spree_prices‘.‘amount‘ BETWEEN 15.0 AND 18.0 AND ‘prices_spree_products‘.‘deleted_at‘ IS NULL AND ‘spree_prices‘.‘currency‘ = ’USD’ AND ‘spree_prices‘.‘country_iso‘ IS NULL ORDER BY ‘spree_products_taxons‘.‘position‘ ASC LIMIT 2 OFFSET 2

solidus	643	SELECT ‘spree_option_values‘.‘id‘ FROM ‘spree_option_values‘ INNER JOIN ‘spree_variant_property_rule_conditions‘ ON ‘spree_option_values‘.‘id‘ = ‘spree_variant_property_rule_conditions‘.‘option_value_id‘ WHERE ‘spree_variant_property_rule_conditions‘.‘variant_property_rule_id‘ = 7	SELECT ‘spree_variant_property_rule_conditions‘.‘option_value_id‘ AS ‘option_value_id‘ FROM ‘spree_variant_property_rule_conditions‘ AS ‘spree_variant_property_rule_conditions‘ WHERE ‘spree_variant_property_rule_conditions‘.‘variant_property_rule_id‘ = 7
solidus	646	SELECT COUNT(*) FROM (SELECT 1 AS one FROM ‘spree_product_properties‘ WHERE ‘spree_product_properties‘.‘product_id‘ = 421 AND ‘spree_product_properties‘.‘value‘ LIKE ’%loose%’ ORDER BY ‘spree_product_properties‘.‘position‘ ASC LIMIT 25 OFFSET 0) subquery_for_count	SELECT COUNT(*) FROM (SELECT 1 AS ‘one‘ FROM ‘spree_product_properties‘ AS ‘spree_product_properties‘ WHERE ‘spree_product_properties‘.‘value‘ LIKE ’%loose%’ AND ‘spree_product_properties‘.‘product_id‘ = 421 LIMIT 25 OFFSET 0) AS ‘subquery_for_count‘
solidus	656	SELECT COUNT(*) FROM (SELECT 1 AS one FROM ‘spree_stock_locations‘ WHERE ‘spree_stock_locations‘.‘name‘ LIKE ’%south%’ ORDER BY name ASC LIMIT 25 OFFSET 0) subquery_for_count	SELECT COUNT(*) FROM (SELECT 1 AS ‘one‘ FROM ‘spree_stock_locations‘ AS ‘spree_stock_locations‘ WHERE ‘spree_stock_locations‘.‘name‘ LIKE ’%south%’ LIMIT 25 OFFSET 0) AS ‘subquery_for_count‘
solidus	658	SELECT COUNT(*) FROM (SELECT 1 AS one FROM ‘spree_stock_locations‘ ORDER BY name ASC LIMIT 1 OFFSET 0) subquery_for_count	SELECT COUNT(*) FROM (SELECT 1 AS ‘one‘ FROM ‘spree_stock_locations‘ AS ‘spree_stock_locations‘ LIMIT 1 OFFSET 0) AS ‘subquery_for_count‘
solidus	669	SELECT ‘spree_stock_locations‘.‘id‘ FROM ‘spree_stock_locations‘ INNER JOIN ‘spree_user_stock_locations‘ ON ‘spree_stock_locations‘.‘id‘ = ‘spree_user_stock_locations‘.‘stock_location_id‘ WHERE ‘spree_user_stock_locations‘.‘user_id‘ = 785	SELECT ‘spree_user_stock_locations‘.‘stock_location_id‘ AS ‘stock_location_id‘ FROM ‘spree_user_stock_locations‘ AS ‘spree_user_stock_locations‘ WHERE ‘spree_user_stock_locations‘.‘user_id‘ = 785
solidus	671	SELECT ‘spree_countries‘.‘id‘ FROM ‘spree_countries‘ INNER JOIN ‘spree_zone_members‘ ON ‘spree_countries‘.‘id‘ = ‘spree_zone_members‘.‘zoneable_id‘ WHERE ‘spree_zone_members‘.‘zone_id‘ = 457 AND ‘spree_zone_members‘.‘zoneable_type‘ = ’Spree::Country’	SELECT ‘spree_zone_members‘.‘zoneable_id‘ AS ‘zoneable_id‘ FROM ‘spree_zone_members‘ AS ‘spree_zone_members‘ WHERE ‘spree_zone_members‘.‘zoneable_type‘ = ’Spree::Country’ AND ‘spree_zone_members‘.‘zone_id‘ = 457
solidus	681	SELECT COUNT(*) FROM (SELECT 1 AS one FROM ‘spree_states‘ ORDER BY name ASC LIMIT 1 OFFSET 1) subquery_for_count	SELECT COUNT(*) FROM (SELECT 1 AS ‘one‘ FROM ‘spree_states‘ AS ‘spree_states‘ LIMIT 1 OFFSET 1) AS ‘subquery_for_count‘
solidus	761	SELECT COUNT(*) FROM (SELECT 1 AS one FROM ‘spree_zones‘ WHERE ‘spree_zones‘.‘name‘ LIKE ’%south%’ ORDER BY name ASC LIMIT 25 OFFSET 0) subquery_for_count	SELECT COUNT(*) FROM (SELECT 1 AS ‘one‘ FROM ‘spree_zones‘ AS ‘spree_zones‘ WHERE ‘spree_zones‘.‘name‘ LIKE ’%south%’ LIMIT 25 OFFSET 0) AS ‘subquery_for_count‘
solidus	765	SELECT COUNT(*) FROM (SELECT 1 AS one FROM ‘spree_zones‘ ORDER BY name ASC LIMIT 25 OFFSET 0) subquery_for_count	SELECT COUNT(*) FROM (SELECT 1 AS ‘one‘ FROM ‘spree_zones‘ AS ‘spree_zones‘ LIMIT 25 OFFSET 0) AS ‘subquery_for_count‘
solidus	767	SELECT COUNT(*) FROM (SELECT 1 AS one FROM ‘spree_zones‘ ORDER BY name ASC LIMIT 1 OFFSET 0) subquery_for_count	SELECT COUNT(*) FROM (SELECT 1 AS ‘one‘ FROM ‘spree_zones‘ AS ‘spree_zones‘ LIMIT 1 OFFSET 0) AS ‘subquery_for_count‘
solidus	794	SELECT COUNT(*) FROM (SELECT 1 AS one FROM ‘spree_countries‘ ORDER BY name ASC LIMIT 1 OFFSET 0) subquery_for_count	SELECT COUNT(*) FROM (SELECT 1 AS ‘one‘ FROM ‘spree_countries‘ AS ‘spree_countries‘ LIMIT 1 OFFSET 0) AS ‘subquery_for_count‘
solidus	797	SELECT COUNT(*) FROM (SELECT 1 AS one FROM ‘spree_countries‘ ORDER BY name ASC LIMIT 25 OFFSET 0) subquery_for_count	SELECT COUNT(*) FROM (SELECT 1 AS ‘one‘ FROM ‘spree_countries‘ AS ‘spree_countries‘ LIMIT 25 OFFSET 0) AS ‘subquery_for_count‘
solidus	799	SELECT COUNT(*) FROM (SELECT 1 AS one FROM ‘spree_countries‘ WHERE ‘spree_countries‘.‘name‘ LIKE ’%zam%’ ORDER BY name ASC LIMIT 25 OFFSET 0) subquery_for_count	SELECT COUNT(*) FROM (SELECT 1 AS ‘one‘ FROM ‘spree_countries‘ AS ‘spree_countries‘ WHERE ‘spree_countries‘.‘name‘ LIKE ’%zam%’ LIMIT 25 OFFSET 0) AS ‘subquery_for_count‘
solidus	818	SELECT COUNT(*) FROM ‘spree_roles‘ INNER JOIN ‘spree_roles_users‘ ON ‘spree_roles‘.‘id‘ = ‘spree_roles_users‘.‘role_id‘ WHERE ‘spree_roles_users‘.‘user_id‘ = 2401 AND ‘spree_roles‘.‘id‘ = 27	SELECT COUNT(*) FROM ‘spree_roles_users‘ AS ‘spree_roles_users‘ WHERE ‘spree_roles_users‘.‘role_id‘ = 27 AND ‘spree_roles_users‘.‘user_id‘ = 2401
solidus	834	SELECT COUNT(*) FROM (SELECT 1 AS one FROM ‘spree_taxons‘ ORDER BY ‘spree_taxons‘.‘taxonomy_id‘ ASC, ‘spree_taxons‘.‘lft‘ ASC LIMIT 500 OFFSET 0) subquery_for_count	SELECT COUNT(*) FROM (SELECT 1 AS ‘one‘ FROM ‘spree_taxons‘ AS ‘spree_taxons‘ LIMIT 500 OFFSET 0) AS ‘subquery_for_count‘
solidus	836	SELECT COUNT(*) FROM (SELECT 1 AS one FROM ‘spree_taxons‘ WHERE ‘spree_taxons‘.‘name‘ LIKE ’%Ruby%’ ORDER BY ‘spree_taxons‘.‘taxonomy_id‘ ASC, ‘spree_taxons‘.‘lft‘ ASC LIMIT 500 OFFSET 0) subquery_for_count	SELECT COUNT(*) FROM (SELECT 1 AS ‘one‘ FROM ‘spree_taxons‘ AS ‘spree_taxons‘ WHERE ‘spree_taxons‘.‘name‘ LIKE ’%Ruby%’ LIMIT 500 OFFSET 0) AS ‘subquery_for_count‘
spree	135	SELECT DISTINCT ‘spree_orders‘.* FROM ‘spree_orders‘ LEFT OUTER JOIN ‘spree_order_promotions‘ ON ‘spree_order_promotions‘.‘order_id‘ = ‘spree_orders‘.‘id‘ LEFT OUTER JOIN ‘spree_promotions‘ ON ‘spree_promotions‘.‘id‘ = ‘spree_order_promotions‘.‘promotion_id‘ WHERE (‘spree_promotions‘.‘id‘ IN (1) AND ‘spree_orders‘.‘completed_at‘ IS NOT NULL) LIMIT 1 OFFSET 0	SELECT * FROM (SELECT DISTINCT ‘spree_orders‘.‘id‘ AS ‘id‘, ‘spree_orders‘.‘number‘ AS ‘number‘, ‘spree_orders‘.‘item_total‘ AS ‘item_total‘, ‘spree_orders‘.‘total‘ AS ‘total‘, ‘spree_orders‘.‘state‘ AS ‘state‘, ‘spree_orders‘.‘adjustment_total‘ AS ‘adjustment_total‘, ‘spree_orders‘.‘user_id‘ AS ‘user_id‘, ‘spree_orders‘.‘completed_at‘ AS ‘completed_at‘, ‘spree_orders‘.‘bill_address_id‘ AS ‘bill_address_id‘, ‘spree_orders‘.‘ship_address_id‘ AS ‘ship_address_id‘, ‘spree_orders‘.‘payment_total‘ AS ‘payment_total‘, ‘spree_orders‘.‘shipment_state‘ AS ‘shipment_state‘, ‘spree_orders‘.‘payment_state‘ AS ‘payment_state‘, ‘spree_orders‘.‘email‘ AS ‘email‘, ‘spree_orders‘.‘special_instructions‘ AS ‘special_instructions‘, ‘spree_orders‘.‘created_at‘ AS ‘created_at‘, ‘spree_orders‘.‘updated_at‘ AS ‘updated_at‘, ‘spree_orders‘.‘currency‘ AS ‘currency‘, ‘spree_orders‘.‘last_ip_address‘ AS ‘last_ip_address‘, ‘spree_orders‘.‘created_by_id‘ AS ‘created_by_id‘, ‘spree_orders‘.‘shipment_total‘ AS ‘shipment_total‘, ‘spree_orders‘.‘additional_tax_total‘ AS ‘additional_tax_total‘, ‘spree_orders‘.‘promo_total‘ AS ‘promo_total‘, ‘spree_orders‘.‘channel‘ AS ‘channel‘, ‘spree_orders‘.‘included_tax_total‘ AS ‘included_tax_total‘, ‘spree_orders‘.‘item_count‘ AS ‘item_count‘, ‘spree_orders‘.‘approver_id‘ AS ‘approver_id‘, ‘spree_orders‘.‘approved_at‘ AS ‘approved_at‘, ‘spree_orders‘.‘confirmation_delivered‘ AS ‘confirmation_delivered‘, ‘spree_orders‘.‘considered_risky‘ AS ‘considered_risky‘, ‘spree_orders‘.‘token‘ AS ‘token‘, ‘spree_orders‘.‘canceled_at‘ AS ‘canceled_at‘, ‘spree_orders‘.‘canceler_id‘ AS ‘canceler_id‘, ‘spree_orders‘.‘store_id‘ AS ‘store_id‘, ‘spree_orders‘.‘state_lock_version‘ AS ‘state_lock_version‘, ‘spree_orders‘.‘taxable_adjustment_total‘ AS ‘taxable_adjustment_total‘, ‘spree_orders‘.‘non_taxable_adjustment_total‘ AS ‘non_taxable_adjustment_total‘ FROM ‘spree_orders‘ AS ‘spree_orders‘ INNER JOIN ‘spree_order_promotions‘ AS ‘spree_order_promotions‘ ON ‘spree_orders‘.‘id‘ = ‘spree_order_promotions‘.‘order_id‘ WHERE

spree	169	SELECT ‘spree_stock_locations‘.‘id‘ FROM ‘spree_stock_locations‘ INNER JOIN ‘spree_stock_items‘ ON ‘spree_stock_locations‘.‘id‘ = ‘spree_stock_items‘.‘stock_location_id‘ WHERE ‘spree_stock_items‘.‘deleted_at‘ IS NULL AND ‘spree_stock_items‘.‘variant_id‘ = 60	SELECT ‘spree_stock_items‘.‘stock_location_id‘ AS ‘stock_location_id‘ FROM ‘spree_stock_items‘ AS ‘spree_stock_items‘ WHERE ‘spree_stock_items‘.‘variant_id‘ = 60 AND ‘spree_stock_items‘.‘deleted_at‘ IS NULL
spree	288	SELECT ‘spree_promotions‘.‘id‘ FROM ‘spree_promotions‘ INNER JOIN ‘spree_order_promotions‘ ON ‘spree_promotions‘.‘id‘ = ‘spree_order_promotions‘.‘promotion_id‘ WHERE ‘spree_order_promotions‘.‘order_id‘ = 1	SELECT ‘spree_order_promotions‘.‘promotion_id‘ AS ‘promotion_id‘ FROM ‘spree_order_promotions‘ AS ‘spree_order_promotions‘ WHERE ‘spree_order_promotions‘.‘order_id‘ = 1
spree	315	SELECT ‘spree_roles‘.‘id‘ FROM ‘spree_roles‘ INNER JOIN ‘spree_role_users‘ ON ‘spree_roles‘.‘id‘ = ‘spree_role_users‘.‘role_id‘ WHERE ‘spree_role_users‘.‘user_id‘ = 2	SELECT ‘spree_role_users‘.‘role_id‘ AS ‘role_id‘ FROM ‘spree_role_users‘ AS ‘spree_role_users‘ WHERE ‘spree_role_users‘.‘user_id‘ = 2
spree	373	SELECT COUNT(DISTINCT ‘spree_products‘.‘id‘) FROM ‘spree_products‘ INNER JOIN ‘spree_products_taxons‘ ON ‘spree_products_taxons‘.‘product_id‘ = ‘spree_products‘.‘id‘ INNER JOIN ‘spree_taxons‘ ON ‘spree_taxons‘.‘id‘ = ‘spree_products_taxons‘.‘taxon_id‘ INNER JOIN ‘spree_variants‘ ON ‘spree_variants‘.‘deleted_at‘ IS NULL AND ‘spree_variants‘.‘product_id‘ = ‘spree_products‘.‘id‘ AND ‘spree_variants‘.‘is_master‘ = TRUE INNER JOIN ‘spree_prices‘ ON ‘spree_prices‘.‘deleted_at‘ IS NULL AND ‘spree_prices‘.‘variant_id‘ = ‘spree_variants‘.‘id‘ WHERE ‘spree_products‘.‘deleted_at‘ IS NULL AND ‘spree_taxons‘.‘id‘ = 105 AND (‘spree_products‘.deleted_at IS NULL or ‘spree_products‘.deleted_at >= ’2020-05-01 07:05:49.149607’) AND (‘spree_products‘.discontinue_on IS NULL or ‘spree_products‘.discontinue_on >= ’2020-05-01 07:05:49.149863’) AND (‘spree_products‘.available_on <= ’2020-05-01 07:05:49.149846’)	SELECT COUNT(DISTINCT ‘spree_products‘.‘id‘) FROM ‘spree_variants‘ AS ‘spree_variants‘ INNER JOIN ‘spree_prices‘ AS ‘spree_prices‘ ON ‘spree_variants‘.‘id‘ = ‘spree_prices‘.‘variant_id‘ INNER JOIN ‘spree_products_taxons‘ AS ‘spree_products_taxons‘ ON ‘spree_variants‘.‘product_id‘ = ‘spree_products_taxons‘.‘product_id‘ INNER JOIN ‘spree_products‘ AS ‘spree_products‘ ON ‘spree_variants‘.‘product_id‘ = ‘spree_products‘.‘id‘ WHERE ‘spree_products_taxons‘.‘taxon_id‘ = 105 AND ‘spree_variants‘.‘is_master‘ = TRUE AND ‘spree_variants‘.‘deleted_at‘ IS NULL AND (‘spree_products‘.‘discontinue_on‘ IS NULL OR ‘spree_products‘.‘discontinue_on‘ >= ’2020-05-01 07:05:49.149863’) AND ‘spree_products‘.‘deleted_at‘ IS NULL AND (‘spree_products‘.‘deleted_at‘ IS NULL OR ‘spree_products‘.‘deleted_at‘ >= ’2020-05-01 07:05:49.149607’) AND ‘spree_products‘.‘available_on‘ <= ’2020-05-01 07:05:49.149846’ AND ‘spree_prices‘.‘deleted_at‘ IS NULL
spree	393	SELECT COUNT(*) FROM (SELECT DISTINCT ‘spree_products‘.* FROM ‘spree_products‘ INNER JOIN ‘spree_variants‘ ON ‘spree_variants‘.‘deleted_at‘ IS NULL AND ‘spree_variants‘.‘product_id‘ = ‘spree_products‘.‘id‘ AND ‘spree_variants‘.‘is_master‘ = TRUE INNER JOIN ‘spree_prices‘ ON ‘spree_prices‘.‘deleted_at‘ IS NULL AND ‘spree_prices‘.‘variant_id‘ = ‘spree_variants‘.‘id‘ WHERE ‘spree_products‘.‘deleted_at‘ IS NULL AND (‘spree_products‘.deleted_at IS NULL or ‘spree_products‘.deleted_at >= ’2020-05-01 07:05:53.713734’) AND (‘spree_products‘.discontinue_on IS NULL or ‘spree_products‘.discontinue_on >= ’2020-05-01 07:05:53.714089’) AND (‘spree_products‘.available_on <= ’2020-05-01 07:05:53.714072’) ORDER BY ‘spree_products‘.‘updated_at‘ ASC LIMIT 25 OFFSET 0) subquery_for_count	SELECT COUNT(*) FROM (SELECT DISTINCT ‘spree_products‘.‘id‘ AS ‘id‘, ‘spree_products‘.‘name‘ AS ‘name‘, ‘spree_products‘.‘description‘ AS ‘description‘, ‘spree_products‘.‘available_on‘ AS ‘available_on‘, ‘spree_products‘.‘discontinue_on‘ AS ‘discontinue_on‘, ‘spree_products‘.‘deleted_at‘ AS ‘deleted_at‘, ‘spree_products‘.‘slug‘ AS ‘slug‘, ‘spree_products‘.‘meta_description‘ AS ‘meta_description‘, ‘spree_products‘.‘meta_keywords‘ AS ‘meta_keywords‘, ‘spree_products‘.‘tax_category_id‘ AS ‘tax_category_id‘, ‘spree_products‘.‘shipping_category_id‘ AS ‘shipping_category_id‘, ‘spree_products‘.‘created_at‘ AS ‘created_at‘, ‘spree_products‘.‘updated_at‘ AS ‘updated_at‘, ‘spree_products‘.‘promotionable‘ AS ‘promotionable‘, ‘spree_products‘.‘meta_title‘ AS ‘meta_title‘ FROM ‘spree_products‘ AS ‘spree_products‘ INNER JOIN ‘spree_variants‘ AS ‘spree_variants‘ ON ‘spree_products‘.‘id‘ = ‘spree_variants‘.‘product_id‘ INNER JOIN ‘spree_prices‘ AS ‘spree_prices‘ ON ‘spree_variants‘.‘id‘ = ‘spree_prices‘.‘variant_id‘ WHERE ‘spree_variants‘.‘is_master‘ = TRUE AND ‘spree_variants‘.‘deleted_at‘ IS NULL AND ‘spree_prices‘.‘deleted_at‘ IS NULL AND (‘spree_products‘.‘discontinue_on‘ IS NULL OR ‘spree_products‘.‘discontinue_on‘ >= ’2020-05-01 07:05:53.714089’) AND ‘spree_products‘.‘deleted_at‘ IS NULL AND (‘spree_products‘.‘deleted_at‘ IS NULL OR ‘spree_products‘.‘deleted_at‘ >= ’2020-05-01 07:05:53.713734’) AND ‘spree_products‘.‘available_on‘ <= ’2020-05-01 07:05:53.714072’ LIMIT 25 OFFSET 0) AS ‘subquery_for_count‘

spree	396	SELECT COUNT(*) FROM (SELECT DISTINCT ‘spree_products‘.* FROM ‘spree_products‘ INNER JOIN ‘spree_variants‘ ON ‘spree_variants‘.‘deleted_at‘ IS NULL AND ‘spree_variants‘.‘product_id‘ = ‘spree_products‘.‘id‘ AND ‘spree_variants‘.‘is_master‘ = TRUE INNER JOIN ‘spree_prices‘ ON ‘spree_prices‘.‘deleted_at‘ IS NULL AND ‘spree_prices‘.‘variant_id‘ = ‘spree_variants‘.‘id‘ WHERE ‘spree_products‘.‘deleted_at‘ IS NULL AND (‘spree_products‘.deleted_at IS NULL or ‘spree_products‘.deleted_at >= ’2020-05-01 07:05:55.179498’) AND (‘spree_products‘.discontinue_on IS NULL or ‘spree_products‘.discontinue_on >= ’2020-05-01 07:05:55.179826’) AND (‘spree_products‘.available_on <= ’2020-05-01 07:05:55.179810’) ORDER BY ‘spree_products‘.‘updated_at‘ DESC LIMIT 25 OFFSET 0) subquery_for_count	SELECT COUNT(*) FROM (SELECT DISTINCT ‘spree_products‘.‘id‘ AS ‘id‘, ‘spree_products‘.‘name‘ AS ‘name‘, ‘spree_products‘.‘description‘ AS ‘description‘, ‘spree_products‘.‘available_on‘ AS ‘available_on‘, ‘spree_products‘.‘discontinue_on‘ AS ‘discontinue_on‘, ‘spree_products‘.‘deleted_at‘ AS ‘deleted_at‘, ‘spree_products‘.‘slug‘ AS ‘slug‘, ‘spree_products‘.‘meta_description‘ AS ‘meta_description‘, ‘spree_products‘.‘meta_keywords‘ AS ‘meta_keywords‘, ‘spree_products‘.‘tax_category_id‘ AS ‘tax_category_id‘, ‘spree_products‘.‘shipping_category_id‘ AS ‘shipping_category_id‘, ‘spree_products‘.‘created_at‘ AS ‘created_at‘, ‘spree_products‘.‘updated_at‘ AS ‘updated_at‘, ‘spree_products‘.‘promotionable‘ AS ‘promotionable‘, ‘spree_products‘.‘meta_title‘ AS ‘meta_title‘ FROM ‘spree_products‘ AS ‘spree_products‘ INNER JOIN ‘spree_variants‘ AS ‘spree_variants‘ ON ‘spree_products‘.‘id‘ = ‘spree_variants‘.‘product_id‘ INNER JOIN ‘spree_prices‘ AS ‘spree_prices‘ ON ‘spree_variants‘.‘id‘ = ‘spree_prices‘.‘variant_id‘ WHERE ‘spree_variants‘.‘is_master‘ = TRUE AND ‘spree_variants‘.‘deleted_at‘ IS NULL AND (‘spree_products‘.‘discontinue_on‘ IS NULL OR ‘spree_products‘.‘discontinue_on‘ >= ’2020-05-01 07:05:55.179826’) AND ‘spree_products‘.‘deleted_at‘ IS NULL AND (‘spree_products‘.‘deleted_at‘ IS NULL OR ‘spree_products‘.‘deleted_at‘ >= ’2020-05-01 07:05:55.179498’) AND ‘spree_products‘.‘available_on‘ <= ’2020-05-01 07:05:55.179810’ AND ‘spree_prices‘.‘deleted_at‘ IS NULL LIMIT 25 OFFSET 0) AS ‘subquery_for_count‘
spree	405	SELECT COUNT(DISTINCT ‘spree_products‘.‘id‘) FROM ‘spree_products‘ INNER JOIN ‘spree_variants‘ ON ‘spree_variants‘.‘deleted_at‘ IS NULL AND ‘spree_variants‘.‘product_id‘ = ‘spree_products‘.‘id‘ AND ‘spree_variants‘.‘is_master‘ = TRUE INNER JOIN ‘spree_prices‘ ON ‘spree_prices‘.‘deleted_at‘ IS NULL AND ‘spree_prices‘.‘variant_id‘ = ‘spree_variants‘.‘id‘ AND ‘spree_prices‘.‘currency‘ = ’USD’ INNER JOIN ‘spree_prices‘ ‘prices_spree_variants‘ ON ‘prices_spree_variants‘.‘deleted_at‘ IS NULL AND ‘prices_spree_variants‘.‘variant_id‘ = ‘spree_variants‘.‘id‘ WHERE ‘spree_products‘.‘deleted_at‘ IS NULL AND ‘spree_prices‘.‘amount‘ BETWEEN 19.97 AND 20.01 AND ‘spree_prices‘.‘currency‘ = ’USD’ AND ‘spree_products‘.‘name‘ LIKE ’%Test Product%’ AND (‘spree_products‘.deleted_at IS NULL or ‘spree_products‘.deleted_at >= ’2020-05-01 07:05:59.453732’) AND (‘spree_products‘.discontinue_on IS NULL or ‘spree_products‘.discontinue_on >= ’2020-05-01 07:05:59.454038’) AND (‘spree_products‘.available_on <= ’2020-05-01 07:05:59.454019’)	SELECT COUNT(DISTINCT ‘spree_products‘.‘id‘) FROM ‘spree_products‘ AS ‘spree_products‘ INNER JOIN ‘spree_variants‘ AS ‘spree_variants‘ ON ‘spree_products‘.‘id‘ = ‘spree_variants‘.‘product_id‘ INNER JOIN ‘spree_prices‘ AS ‘prices_spree_variants‘ ON ‘spree_variants‘.‘id‘ = ‘prices_spree_variants‘.‘variant_id‘ INNER JOIN ‘spree_prices‘ AS ‘spree_prices‘ ON ‘prices_spree_variants‘.‘variant_id‘ = ‘spree_prices‘.‘variant_id‘ WHERE ‘spree_variants‘.‘is_master‘ = TRUE AND ‘spree_variants‘.‘deleted_at‘ IS NULL AND ‘spree_products‘.‘name‘ LIKE ’%Test Product%’ AND (‘spree_products‘.‘discontinue_on‘ IS NULL OR ‘spree_products‘.‘discontinue_on‘ >= ’2020-05-01 07:05:59.454038’) AND ‘spree_products‘.‘deleted_at‘ IS NULL AND (‘spree_products‘.‘deleted_at‘ IS NULL OR ‘spree_products‘.‘deleted_at‘ >= ’2020-05-01 07:05:59.453732’) AND ‘spree_products‘.‘available_on‘ <= ’2020-05-01 07:05:59.454019’ AND ‘spree_prices‘.‘amount‘ BETWEEN 19.97 AND 20.01 AND ‘prices_spree_variants‘.‘deleted_at‘ IS NULL AND ‘spree_prices‘.‘currency‘ = ’USD’
spree	447	SELECT COUNT(*) FROM (SELECT 1 AS one FROM ‘spree_taxons‘ WHERE ‘spree_taxons‘.‘parent_id‘ IS NULL ORDER BY ‘spree_taxons‘.‘lft‘ ASC LIMIT 25 OFFSET 0) subquery_for_count	SELECT COUNT(*) FROM (SELECT 1 AS ‘one‘ FROM ‘spree_taxons‘ AS ‘spree_taxons‘ WHERE ‘spree_taxons‘.‘parent_id‘ IS NULL LIMIT 25 OFFSET 0) AS ‘subquery_for_count‘
spree	469	SELECT DISTINCT ‘spree_variants‘.* FROM ‘spree_variants‘ INNER JOIN ‘spree_prices‘ ON ‘spree_prices‘.‘deleted_at‘ IS NULL AND ‘spree_prices‘.‘variant_id‘ = ‘spree_variants‘.‘id‘ LEFT OUTER JOIN ‘spree_products‘ ON ‘spree_products‘.‘deleted_at‘ IS NULL AND ‘spree_products‘.‘id‘ = ‘spree_variants‘.‘product_id‘ WHERE ‘spree_variants‘.‘deleted_at‘ IS NULL AND (‘spree_variants‘.‘is_master‘ = FALSE OR (‘spree_variants‘.id IN (SELECT MIN(‘spree_variants‘.id) FROM ‘spree_variants‘ GROUP BY ‘spree_variants‘.product_id HAVING COUNT(*) = 1))) AND (spree_prices.currency = ’USD’) AND (spree_prices.amount IS NOT NULL) AND (‘spree_products‘.‘name‘ LIKE ’%fritos%’ OR ‘spree_variants‘.‘sku‘ LIKE ’%fritos%’) LIMIT 25 OFFSET 0	SELECT * FROM (SELECT DISTINCT ‘spree_variants‘.‘id‘ AS ‘id‘, ‘spree_variants‘.‘sku‘ AS ‘sku‘, ‘spree_variants‘.‘weight‘ AS ‘weight‘, ‘spree_variants‘.‘height‘ AS ‘height‘, ‘spree_variants‘.‘width‘ AS ‘width‘, ‘spree_variants‘.‘depth‘ AS ‘depth‘, ‘spree_variants‘.‘deleted_at‘ AS ‘deleted_at‘, ‘spree_variants‘.‘discontinue_on‘ AS ‘discontinue_on‘, ‘spree_variants‘.‘is_master‘ AS ‘is_master‘, ‘spree_variants‘.‘product_id‘ AS ‘product_id‘, ‘spree_variants‘.‘cost_price‘ AS ‘cost_price‘, ‘spree_variants‘.‘cost_currency‘ AS ‘cost_currency‘, ‘spree_variants‘.‘position‘ AS ‘position‘, ‘spree_variants‘.‘track_inventory‘ AS ‘track_inventory‘, ‘spree_variants‘.‘tax_category_id‘ AS ‘tax_category_id‘, ‘spree_variants‘.‘updated_at‘ AS ‘updated_at‘, ‘spree_variants‘.‘created_at‘ AS ‘created_at‘, ‘spree_variants‘.‘count_on_hand‘ AS ‘count_on_hand‘ FROM ‘spree_variants‘ AS ‘spree_variants‘ INNER JOIN ‘spree_prices‘ AS ‘spree_prices‘ ON ‘spree_variants‘.‘id‘ = ‘spree_prices‘.‘variant_id‘ INNER JOIN ‘spree_products‘ AS ‘spree_products‘ ON ‘spree_products‘.‘deleted_at‘ IS NULL AND ‘spree_products‘.‘id‘ = ‘spree_variants‘.‘product_id‘ WHERE (‘spree_products‘.‘name‘ LIKE ’%fritos%’ OR ‘spree_variants‘.‘sku‘ LIKE ’%fritos%’) AND ‘spree_prices‘.‘deleted_at‘ IS NULL AND ‘spree_prices‘.‘currency‘ = ’USD’ AND NOT ‘spree_prices‘.‘amount‘ IS NULL) AS ‘sub_0‘ WHERE (‘sub_0‘.‘is_master‘ = FALSE OR ‘sub_0‘.‘id‘ IN (SELECT MIN(‘spree_variants‘.‘id‘) FROM ‘spree_variants‘ GROUP BY ‘spree_variants‘.‘product_id‘ HAVING COUNT(*) = 1)) AND ‘sub_0‘.‘deleted_at‘ IS NULL LIMIT 25 OFFSET 0

spree	528	SELECT COUNT(*) FROM (SELECT 1 AS one FROM ‘spree_orders‘ WHERE ‘spree_orders‘.‘user_id‘ = 660 AND ‘spree_orders‘.‘completed_at‘ IS NOT NULL ORDER BY ‘spree_orders‘.‘completed_at‘ ASC LIMIT 25 OFFSET 0) subquery_for_count	SELECT COUNT(*) FROM (SELECT 1 AS ‘one‘ FROM ‘spree_orders‘ AS ‘spree_orders‘ WHERE ‘spree_orders‘.‘user_id‘ = 660 AND NOT ‘spree_orders‘.‘completed_at‘ IS NULL LIMIT 25 OFFSET 0) AS ‘subquery_for_count‘
spree	587	SELECT COUNT(*) FROM (SELECT 1 AS one FROM ‘spree_taxons‘ WHERE ‘spree_taxons‘.‘parent_id‘ = 17 ORDER BY ‘spree_taxons‘.‘lft‘ ASC LIMIT 25 OFFSET 0) subquery_for_count	SELECT COUNT(*) FROM (SELECT 1 AS ‘one‘ FROM ‘spree_taxons‘ AS ‘spree_taxons‘ WHERE ‘spree_taxons‘.‘parent_id‘ = 17 LIMIT 25 OFFSET 0) AS ‘subquery_for_count‘
spree	589	SELECT COUNT(*) FROM (SELECT 1 AS one FROM ‘spree_taxons‘ WHERE ‘spree_taxons‘.‘name‘ LIKE ’%Imaginary%’ ORDER BY ‘spree_taxons‘.‘taxonomy_id‘ ASC, ‘spree_taxons‘.‘lft‘ ASC LIMIT 25 OFFSET 0) subquery_for_count	SELECT COUNT(*) FROM (SELECT 1 AS ‘one‘ FROM ‘spree_taxons‘ AS ‘spree_taxons‘ WHERE ‘spree_taxons‘.‘name‘ LIKE ’%Imaginary%’ LIMIT 25 OFFSET 0) AS ‘subquery_for_count‘
spree	591	SELECT COUNT(*) FROM (SELECT 1 AS one FROM ‘spree_taxons‘ ORDER BY ‘spree_taxons‘.‘taxonomy_id‘ ASC, ‘spree_taxons‘.‘lft‘ ASC LIMIT 25 OFFSET 0) subquery_for_count	SELECT COUNT(*) FROM (SELECT 1 AS ‘one‘ FROM ‘spree_taxons‘ AS ‘spree_taxons‘ LIMIT 25 OFFSET 0) AS ‘subquery_for_count‘
spree	593	SELECT COUNT(*) FROM (SELECT 1 AS one FROM ‘spree_taxons‘ WHERE ‘spree_taxons‘.‘parent_id‘ = 62 AND ‘spree_taxons‘.‘name‘ LIKE ’%Ruby%’ ORDER BY ‘spree_taxons‘.‘lft‘ ASC LIMIT 25 OFFSET 0) subquery_for_count	SELECT COUNT(*) FROM (SELECT 1 AS ‘one‘ FROM ‘spree_taxons‘ AS ‘spree_taxons‘ WHERE ‘spree_taxons‘.‘parent_id‘ = 62 AND ‘spree_taxons‘.‘name‘ LIKE ’%Ruby%’ LIMIT 25 OFFSET 0) AS ‘subquery_for_count‘
spree	614	SELECT COUNT(*) FROM (SELECT 1 AS one FROM ‘spree_payments‘ WHERE ‘spree_payments‘.‘order_id‘ = 23 ORDER BY ‘spree_payments‘.‘created_at‘ ASC LIMIT 25 OFFSET 0) subquery_for_count	SELECT COUNT(*) FROM (SELECT 1 AS ‘one‘ FROM ‘spree_payments‘ AS ‘spree_payments‘ WHERE ‘spree_payments‘.‘order_id‘ = 23 LIMIT 25 OFFSET 0) AS ‘subquery_for_count‘
spree	634	SELECT COUNT(*) FROM (SELECT 1 AS one FROM ‘spree_orders‘ WHERE ‘spree_orders‘.‘user_id‘ = 277 AND ‘spree_orders‘.‘completed_at‘ IS NOT NULL ORDER BY spree_orders.completed_at IS NULL, ‘spree_orders‘.‘completed_at‘ DESC, ‘spree_orders‘.‘created_at‘ DESC LIMIT 25 OFFSET 0) subquery_for_count	SELECT COUNT(*) FROM (SELECT 1 AS ‘one‘ FROM ‘spree_orders‘ AS ‘spree_orders‘ WHERE ‘spree_orders‘.‘user_id‘ = 277 AND NOT ‘spree_orders‘.‘completed_at‘ IS NULL LIMIT 25 OFFSET 0) AS ‘subquery_for_count‘
spree	657	SELECT COUNT(*) FROM (SELECT 1 AS one FROM ‘spree_stock_locations‘ WHERE ‘spree_stock_locations‘.‘name‘ LIKE ’%south%’ ORDER BY name ASC LIMIT 25 OFFSET 0) subquery_for_count	SELECT COUNT(*) FROM (SELECT 1 AS ‘one‘ FROM ‘spree_stock_locations‘ AS ‘spree_stock_locations‘ WHERE ‘spree_stock_locations‘.‘name‘ LIKE ’%south%’ LIMIT 25 OFFSET 0) AS ‘subquery_for_count‘
spree	659	SELECT COUNT(*) FROM (SELECT 1 AS one FROM ‘spree_stock_locations‘ ORDER BY name ASC LIMIT 1 OFFSET 0) subquery_for_count	SELECT COUNT(*) FROM (SELECT 1 AS ‘one‘ FROM ‘spree_stock_locations‘ AS ‘spree_stock_locations‘ LIMIT 1 OFFSET 0) AS ‘subquery_for_count‘
spree	662	SELECT COUNT(*) FROM (SELECT 1 AS one FROM ‘spree_stock_locations‘ ORDER BY name ASC LIMIT 25 OFFSET 0) subquery_for_count	SELECT COUNT(*) FROM (SELECT 1 AS ‘one‘ FROM ‘spree_stock_locations‘ AS ‘spree_stock_locations‘ LIMIT 25 OFFSET 0) AS ‘subquery_for_count‘
spree	691	SELECT COUNT(*) FROM (SELECT 1 AS one FROM ‘spree_zones‘ ORDER BY name ASC LIMIT 1 OFFSET 0) subquery_for_count	SELECT COUNT(*) FROM (SELECT 1 AS ‘one‘ FROM ‘spree_zones‘ AS ‘spree_zones‘ LIMIT 1 OFFSET 0) AS ‘subquery_for_count‘
spree	693	SELECT COUNT(*) FROM (SELECT 1 AS one FROM ‘spree_zones‘ ORDER BY name ASC LIMIT 25 OFFSET 0) subquery_for_count	SELECT COUNT(*) FROM (SELECT 1 AS ‘one‘ FROM ‘spree_zones‘ AS ‘spree_zones‘ LIMIT 25 OFFSET 0) AS ‘subquery_for_count‘
spree	695	SELECT COUNT(*) FROM (SELECT 1 AS one FROM ‘spree_zones‘ WHERE ‘spree_zones‘.‘name‘ LIKE ’%south%’ ORDER BY name ASC LIMIT 25 OFFSET 0) subquery_for_count	SELECT COUNT(*) FROM (SELECT 1 AS ‘one‘ FROM ‘spree_zones‘ AS ‘spree_zones‘ WHERE ‘spree_zones‘.‘name‘ LIKE ’%south%’ LIMIT 25 OFFSET 0) AS ‘subquery_for_count‘
spree	712	SELECT COUNT(*) FROM (SELECT DISTINCT ‘spree_products‘.* FROM ‘spree_products‘ INNER JOIN ‘spree_variants‘ ON ‘spree_variants‘.‘deleted_at‘ IS NULL AND ‘spree_variants‘.‘product_id‘ = ‘spree_products‘.‘id‘ AND ‘spree_variants‘.‘is_master‘ = TRUE INNER JOIN ‘spree_prices‘ ON ‘spree_prices‘.‘deleted_at‘ IS NULL AND ‘spree_prices‘.‘variant_id‘ = ‘spree_variants‘.‘id‘ WHERE ‘spree_products‘.‘deleted_at‘ IS NULL AND (‘spree_products‘.discontinue_on IS NULL or ‘spree_products‘.discontinue_on >= ’2020-05-01 07:07:42.906418’) AND (‘spree_products‘.available_on <= ’2020-05-01 07:07:42.906389’) ORDER BY ‘spree_products‘.‘created_at‘ ASC LIMIT 25 OFFSET 0) subquery_for_count	SELECT COUNT(*) FROM (SELECT DISTINCT ‘spree_products‘.‘id‘ AS ‘id‘, ‘spree_products‘.‘name‘ AS ‘name‘, ‘spree_products‘.‘description‘ AS ‘description‘, ‘spree_products‘.‘available_on‘ AS ‘available_on‘, ‘spree_products‘.‘discontinue_on‘ AS ‘discontinue_on‘, ‘spree_products‘.‘deleted_at‘ AS ‘deleted_at‘, ‘spree_products‘.‘slug‘ AS ‘slug‘, ‘spree_products‘.‘meta_description‘ AS ‘meta_description‘, ‘spree_products‘.‘meta_keywords‘ AS ‘meta_keywords‘, ‘spree_products‘.‘tax_category_id‘ AS ‘tax_category_id‘, ‘spree_products‘.‘shipping_category_id‘ AS ‘shipping_category_id‘, ‘spree_products‘.‘created_at‘ AS ‘created_at‘, ‘spree_products‘.‘updated_at‘ AS ‘updated_at‘, ‘spree_products‘.‘promotionable‘ AS ‘promotionable‘, ‘spree_products‘.‘meta_title‘ AS ‘meta_title‘ FROM ‘spree_products‘ AS ‘spree_products‘ INNER JOIN ‘spree_variants‘ AS ‘spree_variants‘ ON ‘spree_products‘.‘id‘ = ‘spree_variants‘.‘product_id‘ INNER JOIN ‘spree_prices‘ AS ‘spree_prices‘ ON ‘spree_variants‘.‘id‘ = ‘spree_prices‘.‘variant_id‘ WHERE ‘spree_variants‘.‘is_master‘ = TRUE AND ‘spree_variants‘.‘deleted_at‘ IS NULL AND ‘spree_prices‘.‘deleted_at‘ IS NULL AND (‘spree_products‘.‘discontinue_on‘ IS NULL OR ‘spree_products‘.‘discontinue_on‘ >= ’2020-05-01 07:07:42.906418’) AND ‘spree_products‘.‘deleted_at‘ IS NULL AND ‘spree_products‘.‘available_on‘ <= ’2020-05-01 07:07:42.906389’ LIMIT 25 OFFSET 0) AS ‘subquery_for_count‘

spree	741	SELECT COUNT(*) FROM (SELECT 1 AS one FROM ‘spree_shipments‘ INNER JOIN ‘spree_orders‘ ON ‘spree_orders‘.‘id‘ = ‘spree_shipments‘.‘order_id‘ WHERE ‘spree_orders‘.‘user_id‘ = 548 ORDER BY coalesce(spree_shipments.shipped_at, spree_shipments.created_at) desc, ‘spree_shipments‘.‘id‘ DESC LIMIT 25 OFFSET 0) subquery_for_count	SELECT COUNT(*) FROM (SELECT 1 AS ‘one‘ FROM ‘spree_shipments‘ AS ‘spree_shipments‘ INNER JOIN ‘spree_orders‘ AS ‘spree_orders‘ ON ‘spree_shipments‘.‘order_id‘ = ‘spree_orders‘.‘id‘ WHERE ‘spree_orders‘.‘user_id‘ = 548 LIMIT 25 OFFSET 0) AS ‘subquery_for_count‘
spree	758	SELECT ‘spree_users‘.‘id‘ FROM ‘spree_users‘ LEFT OUTER JOIN ‘spree_addresses‘ ON ‘spree_addresses‘.‘id‘ = ‘spree_users‘.‘ship_address_id‘ WHERE ((spree_addresses.firstname like ’%Result%’) OR (spree_addresses.lastname like ’%Result%’))	SELECT ‘spree_users‘.‘id‘ AS ‘id‘ FROM ‘spree_users‘ AS ‘spree_users‘ INNER JOIN ‘spree_addresses‘ AS ‘spree_addresses‘ ON ‘spree_users‘.‘ship_address_id‘ = ‘spree_addresses‘.‘id‘ WHERE ‘spree_addresses‘.‘firstname‘ LIKE ’%Result%’ OR ‘spree_addresses‘.‘lastname‘ LIKE ’%Result%’
spree	766	SELECT COUNT(*) FROM (SELECT 1 AS one FROM ‘spree_reimbursements‘ ORDER BY ‘spree_reimbursements‘.‘created_at‘ DESC LIMIT 25 OFFSET 0) subquery_for_count	SELECT COUNT(*) FROM (SELECT 1 AS ‘one‘ FROM ‘spree_reimbursements‘ AS ‘spree_reimbursements‘ LIMIT 25 OFFSET 0) AS ‘subquery_for_count‘
spree	781	SELECT COUNT(*) FROM (SELECT 1 AS one FROM ‘spree_customer_returns‘ ORDER BY ‘spree_customer_returns‘.‘created_at‘ DESC LIMIT 25 OFFSET 0) subquery_for_count	SELECT COUNT(*) FROM (SELECT 1 AS ‘one‘ FROM ‘spree_customer_returns‘ AS ‘spree_customer_returns‘ LIMIT 25 OFFSET 0) AS ‘subquery_for_count‘
spree	783	SELECT COUNT(*) FROM (SELECT 1 AS one FROM ‘spree_taxonomies‘ WHERE ‘spree_taxonomies‘.‘name‘ LIKE ’%style%’ ORDER BY spree_taxonomies.position, spree_taxonomies.created_at, name LIMIT 25 OFFSET 0) subquery_for_count	SELECT COUNT(*) FROM (SELECT 1 AS ‘one‘ FROM ‘spree_taxonomies‘ AS ‘spree_taxonomies‘ WHERE ‘spree_taxonomies‘.‘name‘ LIKE ’%style%’ LIMIT 25 OFFSET 0) AS ‘subquery_for_count‘
spree	785	SELECT COUNT(*) FROM (SELECT 1 AS one FROM ‘spree_taxonomies‘ ORDER BY spree_taxonomies.position, spree_taxonomies.created_at, name LIMIT 25 OFFSET 0) subquery_for_count	SELECT COUNT(*) FROM (SELECT 1 AS ‘one‘ FROM ‘spree_taxonomies‘ AS ‘spree_taxonomies‘ LIMIT 25 OFFSET 0) AS ‘subquery_for_count‘
spree	787	SELECT COUNT(*) FROM (SELECT 1 AS one FROM ‘spree_taxonomies‘ ORDER BY spree_taxonomies.position, spree_taxonomies.created_at, name LIMIT 1 OFFSET 0) subquery_for_count	SELECT COUNT(*) FROM (SELECT 1 AS ‘one‘ FROM ‘spree_taxonomies‘ AS ‘spree_taxonomies‘ LIMIT 1 OFFSET 0) AS ‘subquery_for_count‘
spree	789	SELECT ‘spree_option_values‘.‘id‘ FROM ‘spree_option_values‘ INNER JOIN ‘spree_option_value_variants‘ ON ‘spree_option_values‘.‘id‘ = ‘spree_option_value_variants‘.‘option_value_id‘ WHERE ‘spree_option_value_variants‘.‘variant_id‘ = 2	SELECT ‘spree_option_value_variants‘.‘option_value_id‘ AS ‘option_value_id‘ FROM ‘spree_option_value_variants‘ AS ‘spree_option_value_variants‘ WHERE ‘spree_option_value_variants‘.‘variant_id‘ = 2
spree	853	SELECT ‘spree_countries‘.‘id‘ FROM ‘spree_countries‘ INNER JOIN ‘spree_zone_members‘ ON ‘spree_countries‘.‘id‘ = ‘spree_zone_members‘.‘zoneable_id‘ WHERE ‘spree_zone_members‘.‘zone_id‘ = 154 AND ‘spree_zone_members‘.‘zoneable_type‘ = ’Spree::Country’	SELECT ‘spree_zone_members‘.‘zoneable_id‘ AS ‘zoneable_id‘ FROM ‘spree_zone_members‘ AS ‘spree_zone_members‘ WHERE ‘spree_zone_members‘.‘zoneable_type‘ = ’Spree::Country’ AND ‘spree_zone_members‘.‘zone_id‘ = 154
spree	886	SELECT COUNT(*) FROM ‘spree_option_values‘ INNER JOIN ‘spree_option_value_variants‘ ON ‘spree_option_values‘.‘id‘ = ‘spree_option_value_variants‘.‘option_value_id‘ WHERE ‘spree_option_value_variants‘.‘variant_id‘ = 981	SELECT COUNT(*) FROM ‘spree_option_value_variants‘ AS ‘spree_option_value_variants‘ WHERE ‘spree_option_value_variants‘.‘variant_id‘ = 981
spree	1017	SELECT COUNT(*) FROM ‘spree_taxons‘ INNER JOIN ‘spree_products_taxons‘ ON ‘spree_taxons‘.‘id‘ = ‘spree_products_taxons‘.‘taxon_id‘ WHERE ‘spree_products_taxons‘.‘product_id‘ = 1606	SELECT COUNT(*) FROM ‘spree_products_taxons‘ AS ‘spree_products_taxons‘ WHERE ‘spree_products_taxons‘.‘product_id‘ = 1606
spree	1022	SELECT COUNT(*) FROM ‘spree_products‘ INNER JOIN ‘spree_product_properties‘ ON ‘spree_product_properties‘.‘product_id‘ = ‘spree_products‘.‘id‘ INNER JOIN ‘spree_properties‘ ON ‘spree_properties‘.‘id‘ = ‘spree_product_properties‘.‘property_id‘ WHERE ‘spree_products‘.‘deleted_at‘ IS NULL AND ‘spree_properties‘.‘id‘ = 28	SELECT COUNT(*) FROM ‘spree_products‘ AS ‘spree_products‘ INNER JOIN ‘spree_product_properties‘ AS ‘spree_product_properties‘ ON ‘spree_products‘.‘id‘ = ‘spree_product_properties‘.‘product_id‘ WHERE ‘spree_product_properties‘.‘property_id‘ = 28 AND ‘spree_products‘.‘deleted_at‘ IS NULL
spree	1093	SELECT DISTINCT ‘spree_zones‘.‘id‘ FROM ‘spree_zones‘ INNER JOIN ‘spree_zone_members‘ ON ‘spree_zone_members‘.‘zone_id‘ = ‘spree_zones‘.‘id‘ WHERE ((spree_zone_members.zoneable_type = ’Spree::State’ ANDspree_zone_members.zoneable_id IN (3051)) OR (spree_zone_members.zoneable_type = ’Spree::Country’ ANDspree_zone_members.zoneable_id IN (1543)))	SELECT DISTINCT ‘spree_zone_members‘.‘zone_id‘ AS ‘zone_id‘ FROM ‘spree_zone_members‘ AS ‘spree_zone_members‘ WHERE ‘spree_zone_members‘.‘zoneable_type‘ = ’Spree::State’ AND ‘spree_zone_members‘.‘zoneable_id‘ IN (?) OR ‘spree_zone_members‘.‘zoneable_type‘ = ’Spree::Country’ AND ‘spree_zone_members‘.‘zoneable_id‘ IN (?)
spree	1147	SELECT COUNT(*) FROM (SELECT 1 AS one FROM ‘spree_stock_transfers‘ WHERE ‘spree_stock_transfers‘.‘destination_location_id‘ = 4 ORDER BY ‘spree_stock_transfers‘.‘created_at‘ DESC LIMIT 25 OFFSET 0) subquery_for_count	SELECT COUNT(*) FROM (SELECT 1 AS ‘one‘ FROM ‘spree_stock_transfers‘ AS ‘spree_stock_transfers‘ WHERE ‘spree_stock_transfers‘.‘destination_location_id‘ = 4 LIMIT 25 OFFSET 0) AS ‘subquery_for_count‘

spree	1150	SELECT COUNT(*) FROM (SELECT 1 AS one FROM ‘spree_stock_transfers‘ ORDER BY ‘spree_stock_transfers‘.‘created_at‘ DESC LIMIT 25 OFFSET 0) subquery_for_count	SELECT COUNT(*) FROM (SELECT 1 AS ‘one‘ FROM ‘spree_stock_transfers‘ AS ‘spree_stock_transfers‘ LIMIT 25 OFFSET 0) AS ‘subquery_for_count‘
spree	1151	SELECT COUNT(*) FROM (SELECT 1 AS one FROM ‘spree_stock_transfers‘ WHERE ‘spree_stock_transfers‘.‘source_location_id‘ = 1 ORDER BY ‘spree_stock_transfers‘.‘created_at‘ DESC LIMIT 25 OFFSET 0) subquery_for_count	SELECT COUNT(*) FROM (SELECT 1 AS ‘one‘ FROM ‘spree_stock_transfers‘ AS ‘spree_stock_transfers‘ WHERE ‘spree_stock_transfers‘.‘source_location_id‘ = 1 LIMIT 25 OFFSET 0) AS ‘subquery_for_count‘
spree	1165	SELECT ‘spree_orders‘.* FROM ‘spree_orders‘ LEFT OUTER JOIN ‘spree_users‘ ON ‘spree_users‘.‘id‘ = ‘spree_orders‘.‘user_id‘ WHERE ‘spree_users‘.‘id‘ = 26 ORDER BY ‘spree_orders‘.‘state‘ DESC LIMIT 25 OFFSET 0	SELECT * FROM ‘spree_orders‘ AS ‘spree_orders‘ WHERE ‘spree_orders‘.‘user_id‘ = 26 ORDER BY ‘spree_orders‘.‘state‘ DESC LIMIT 25 OFFSET 0
spree	1167	SELECT ‘spree_orders‘.* FROM ‘spree_orders‘ LEFT OUTER JOIN ‘spree_users‘ ON ‘spree_users‘.‘id‘ = ‘spree_orders‘.‘user_id‘ WHERE ‘spree_users‘.‘id‘ = 32 ORDER BY ‘spree_orders‘.‘number‘ DESC LIMIT 25 OFFSET 0	SELECT * FROM ‘spree_orders‘ AS ‘spree_orders‘ WHERE ‘spree_orders‘.‘user_id‘ = 32 ORDER BY ‘spree_orders‘.‘number‘ DESC LIMIT 25 OFFSET 0
spree	1168	SELECT ‘spree_orders‘.* FROM ‘spree_orders‘ LEFT OUTER JOIN ‘spree_users‘ ON ‘spree_users‘.‘id‘ = ‘spree_orders‘.‘user_id‘ WHERE ‘spree_users‘.‘id‘ = 35 ORDER BY ‘spree_orders‘.‘completed_at‘ ASC LIMIT 25 OFFSET 0	SELECT * FROM ‘spree_orders‘ AS ‘spree_orders‘ WHERE ‘spree_orders‘.‘user_id‘ = 35 ORDER BY ‘spree_orders‘.‘completed_at‘ ASC LIMIT 25 OFFSET 0
spree	1169	SELECT ‘spree_orders‘.* FROM ‘spree_orders‘ LEFT OUTER JOIN ‘spree_users‘ ON ‘spree_users‘.‘id‘ = ‘spree_orders‘.‘user_id‘ WHERE ‘spree_users‘.‘id‘ = 35 ORDER BY ‘spree_orders‘.‘completed_at‘ DESC LIMIT 25 OFFSET 0	SELECT * FROM ‘spree_orders‘ AS ‘spree_orders‘ WHERE ‘spree_orders‘.‘user_id‘ = 35 ORDER BY ‘spree_orders‘.‘completed_at‘ DESC LIMIT 25 OFFSET 0
spree	1203	select p.id from spree_products p inner join spree_products_taxons op on p.id = op.product_id	SELECT ‘op‘.‘product_id‘ AS ‘product_id‘ FROM ‘spree_products_taxons‘ AS ‘op‘