

MySQL Queries How To Guide

Contents

- 1 Overview
- 2 Prerequisites
- 3 Performing MySQL Queries
 - ◆ 3.1 MySQL Queries on a Swivel appliance using Webmin
 - ◆ 3.2 Querying individual Repositories
- 4 MySQL Queries
 - ◆ 4.1 List users who never have performed a successful login
 - ◆ 4.2 List users who have not had a login in a defined time period
 - ◆ 4.3 List users who have logged in over the last 30 days
 - ◆ 4.4 List users who have logged in over the last 3 months
 - ◆ 4.5 List users who have logged in over the last 3 months and show alert transport email address
 - ◆ 4.6 List users who have logged in over the last 3 months and show strings transport email address
 - ◆ 4.7 List users who have logged in over the last 3 months and show email address (version 3.9.1 onwards)
 - ◆ 4.8 List last login date and time
 - ◆ 4.9 List last login date and time by repository
 - ◆ 4.10 List username and repository
 - ◆ 4.11 Count number of users in each group
 - ◆ 4.12 List number of users in each group
 - ◆ 4.13 List last login date and time and show users who have never logged in
 - ◆ 4.14 List login times for a given user
 - ◆ 4.15 List unsuccessful login times for a given user
 - ◆ 4.16 List login times for all users
 - ◆ 4.17 List unsuccessful login times for all users
 - ◆ 4.18 Count the number of Logins
 - ◆ 4.19 Count the number of Failed Logins
 - ◆ 4.20 Count the number of successful logins over last 30 days
 - ◆ 4.21 Count the number of failed logins over last 30 days
 - ◆ 4.22 Count the number of logins and login failures per day over the last 30 days
 - ◆ 4.23 Count the number of users who have logged in over the last 30 days
 - ◆ 4.24 Count the number of times each user has logged in
 - ◆ 4.25 Count the users number of unsuccessful logins and self resets
 - ◆ 4.26 List the users who have reset their PIN from a particular date
 - ◆ 4.27 List the users who have used ChangePIN from a particular date
 - ◆ 4.28 List the users who have used Self Reset from a particular date
 - ◆ 4.29 List the users who have had an Admin reset from a particular date
 - ◆ 4.30 List user date of creation and last login
 - ◆ 4.31 List users who have been created in the last 30 days
 - ◆ 4.32 List users who have been created in the last month
 - ◆ 4.33 Count the number of users who have never logged in
 - ◆ 4.34 Count the number of users who have never logged in over the last 30 days
 - ◆ 4.35 Count the number of users who have never logged in over the last 180 days but created more than 30 days ago
 - ◆ 4.36 List the number of users who have never logged in over the last 180 days but created more than 30 days ago
 - ◆ 4.37 List users connected today who have not previously logged in
 - ◆ 4.38 lists all the Users Ids, email address, created date, last login, last pin change and repository name
 - ◆ 4.39 List PIN expiry for users
 - ◆ 4.40 List all Disabled accounts
 - ◆ 4.41 List all users with inactive, locked, disabled and deleted state
 - ◆ 4.42 List all users with inactive, locked, disabled and deleted state and show creation, login and PIN change date
- 5 Queries on versions prior to Swivel 3.4
- 6 Known Issues
- 7 Troubleshooting

Overview

The MySQL Swivel database contains some information that is not available through the Swivel Administration Console. This document outlines some of the information that can be queried.

The **Audit Log** is where Swivel maintains an activity log for users. For more information on the Audit Log feature within Swivel, see the [Audit Log How To Guide](#). For information on producing reports through the Swivel Administration console see [Reporting How to guide](#)

For information on querying the Internal Swivel database see [Reporting Using Agent-XML How to Guide](#)

For information on querying a MS SQL database see [MS SQL Queries How To Guide](#)

For database schema information, in order to write your own queries, see [Database Schema](#)

Prerequisites

- SQL Database;
- Audit log requires Swivel 3.4 or higher (for SQL queries on Swivele versions prior to 3.4 see the note at the bottom of the document).

Performing MySQL Queries

The MySQL database can be queried through the MySQL command line or through the Webmin interface. depending on the installation, the database name may be pinsafe or pinsafe_rep.

MySQL Queries on a Swivel appliance using Webmin

To view the MySQL database on an appliance, log into webmin (https://<pinsafe_server>:10000). Go to Servers -> MySQL Database Server and select the Swivel database by clicking on the icon, this will either be named pinsafe or pinsafe_rep. If you have both, pinsafe_rep will be the active one.

Webmin Others Cluster System Hardware Servers Networking

MySQL Database Server

MySQL version 5.0.22

MySQL Database Server
PINsafe
Sendmail Mail Server

MySQL Databases

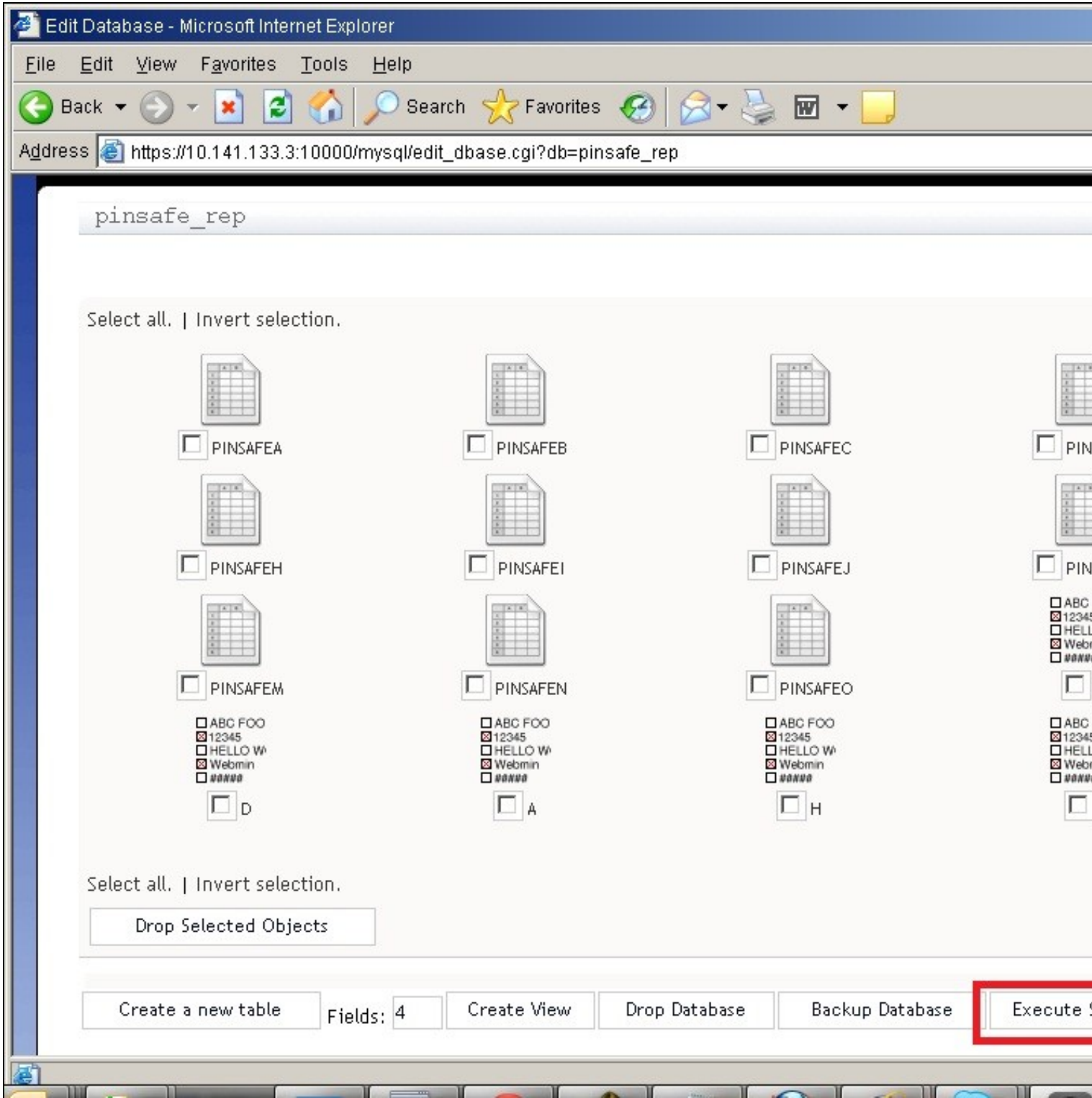
Select all. | Invert selection. | Create a new database.

information_schema mysql pinsafe_rep

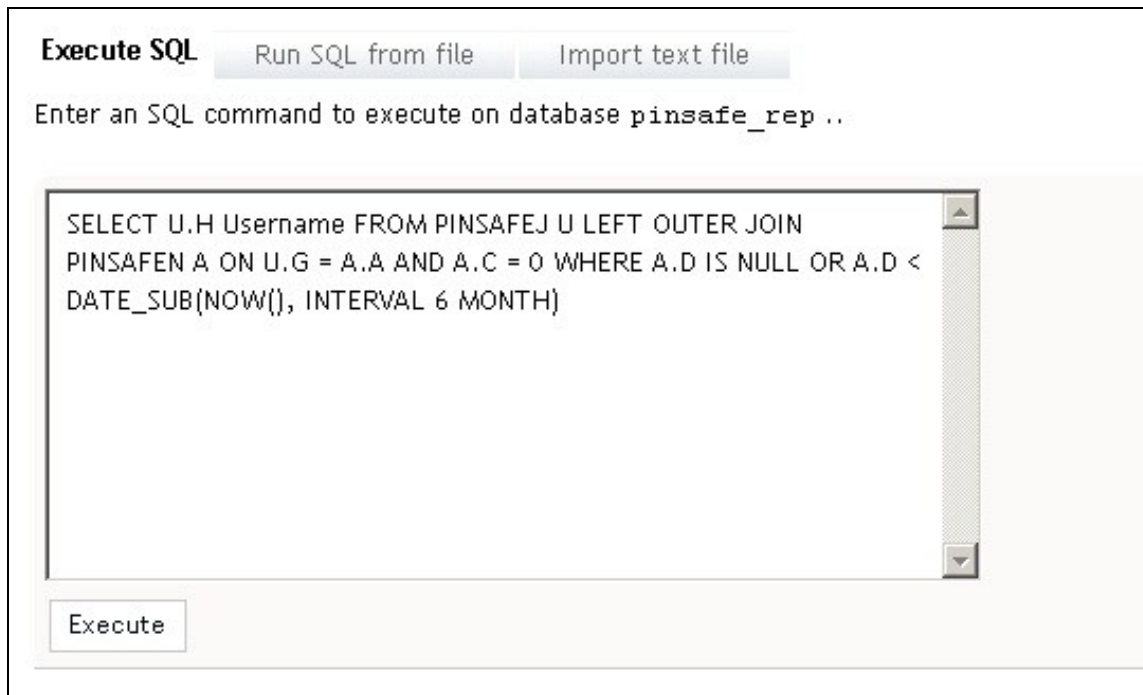
Select all. | Invert selection. | Create a new database.

Drop Selected Databases

Click on the Execute SQL button.



Enter one of the below queries then click on Execute. For further information on see [SQL commands in Webmin](#)



Querying individual Repositories

Each Repository may be queried by name or Repository ID. To find a Repository ID user:

```
SELECT A ID, B NAME FROM PINSAFEL
```

To query by NAME the following is used:

```
AND PINSAFEL.B = <NAME>
```

To query by ID the following is used:

```
AND I = <ID>
```

Example

Users who have never logged on (Successfully), for all repositories

```
USE pinsafe_rep; SELECT H Username FROM PINSAFEJ WHERE G NOT IN (SELECT DISTINCT A FROM PINSAFEN WHERE C=0)
```

Users who have never logged on (Successfully), for repositories <NAME>, Replace <NAME> with the name of the repository.

```
SELECT H Username FROM PINSAFEJ INNER JOIN PINSAFEL WHERE PINSAFEJ.I = PINSAFEL.A WHERE PINSAFEJ.G NOT IN (SELECT DISTINCT A FROM PINSAFEN WHERE C=0) AND PINSAFEL.B = <NAME>
```

Users who have never logged on (Successfully), for repository <ID>, Replace <ID> with the ID of the repository.

```
SELECT H Username FROM PINSAFEJ WHERE G NOT IN (SELECT DISTINCT A FROM PINSAFEN WHERE C=0) AND I = <ID>
```

MySQL Queries

List users who never have performed a successful login

The following query lists all users that have never (successfully) logged on to PINsafe:

```
SELECT H Username FROM PINSAFEJ WHERE G NOT IN (SELECT DISTINCT A FROM PINSAFEN WHERE C=0)
```

List users who have not had a login in a defined time period

The following query lists all users that have not logged in in the last 3 months from the current date and time (obviously, the period can be varied), The query NOW() specifies the current time going back the required time period, to query going back to the start of the day use DATE():

```
SELECT U.H Username FROM PINSAFEJ U LEFT OUTER JOIN PINSAFEN A ON U.G = A.A AND A.C = 0 WHERE A.D IS NULL OR A.D < DATE_SUB(NOW(), INTERVAL 3 MONTH)
```

List users who have logged in over the last 30 days

```
SELECT U.H Username, A1.D CreationTime, A2.D LoginTime FROM PINSAFEJ U INNER JOIN PINSAFEN A1 ON U.G = A1.A AND A1.C = 3 LEFT OUTER JOIN PINSAFEN A2 ON U.G = A2.A AND A2.C = 0 WHERE A2.D > DATE_SUB(NOW(), INTERVAL 30 DAY);
```

List users who have logged in over the last 3 months

```
SELECT U.H Username, R.B Repository FROM PINSAFEJ U INNER JOIN PINSAFEL R ON U.I = R.A LEFT OUTER JOIN PINSAFEN A ON U.G = A.A AND A.C = 0 WHERE A.D IS NULL OR A.D < DATE_SUB(NOW(), INTERVAL 3 MONTH)
```

List users who have logged in over the last 3 months and show alert transport email address

```
SELECT U.H Username, R.B Repository, T.A Email FROM PINSAFEJ U INNER JOIN PINSAFEL R ON U.I = R.A LEFT OUTER JOIN PINSAFEA T ON U.G = T.C LEFT OUTER JOIN PINSAFEN A ON U.G = A.A AND A.C = 0 WHERE A.D IS NULL OR A.D < DATE_SUB(NOW(), INTERVAL 3 MONTH)
```

List users who have logged in over the last 3 months and show strings transport email address

```
SELECT U.H Username, R.B Repository, T.A Email FROM PINSAFEJ U INNER JOIN PINSAFEL R ON U.I = R.A LEFT OUTER JOIN PINSAFEH T ON U.G = T.C LEFT OUTER JOIN PINSAFEN A ON U.G = A.A AND A.C = 0 WHERE A.D IS NULL OR A.D < DATE_SUB(NOW(), INTERVAL 3 MONTH)
```

List users who have logged in over the last 3 months and show email address (version 3.9.1 onwards)

```
SELECT U.H Username, R.B Repository, A.C Email FROM PINSAFEJ U INNER JOIN PINSAFEL R ON U.I = R.A LEFT OUTER JOIN PINSAFEP A ON U.G = A.A AND A.B = 'email' LEFT OUTER JOIN PINSAFEN A ON U.G = A.A AND A.C = 0 WHERE A.D IS NULL OR A.D < DATE_SUB(NOW(), INTERVAL 3 MONTH)
```

List last login date and time

The following query lists the last date/time that each user logged in (the oldest login time first - add DESC at the end for most recent first):

```
SELECT U.H Username, A.D LoginTime FROM PINSAFEJ U INNER JOIN PINSAFEN A ON U.G = A.A AND A.C = 0 ORDER BY LoginTime
```

List last login date and time by repository

The following query lists the last date/time that each user logged in from a specified repository listed by login time.

```
SELECT U.H Username, A.D LoginTime FROM PINSAFEJ U INNER JOIN PINSAFEN A ON U.G = A.A AND A.C = 0 INNER JOIN PINSAFEL R ON U.I = R.A AND R.B = '<repos_name>' ORDER BY LoginTime
```

Replace <repos_name> in the above query with the name of the repository.

List username and repository

```
SELECT U.H Username, R.B FROM PINSAFEJ U JOIN PINSAFEN A1 ON U.G = A1.A AND A1.C = 3 LEFT OUTER JOIN PINSAFEN A2 ON U.G = A2.A AND A2.C=0 JOIN PINSAFEL R ON U.I = R.A WHERE ( A1.D < DATE_SUB( NOW(), INTERVAL 30 DAY ) ) AND ( A2.D IS NULL OR A2.D < DATE_SUB( NOW(), INTERVAL 180 DAY ) )
```

Count number of users in each group

```
SELECT R.B, G.A, COUNT(U.G)
```

```
FROM PINSAFEJ U
JOIN PINSAFEL R ON U.I=R.A
JOIN PINSAFEI G ON U.G=G.B
GROUP BY R.B, G.A
ORDER BY R.B, G.A
```

List number of users in each group

```
SELECT R.B, G.A, U.C
```

```
FROM PINSAFEJ U
JOIN PINSAFEL R ON U.I=R.A
JOIN PINSAFEI G ON U.G=G.B
ORDER BY R.B, G.A, U.C
```

List last login date and time and show users who have never logged in

The following query shows all users and their last login date. Any dates listed as NULL means that the user has never logged in, or never changed their PIN.

```
SELECT U.H Username, A1.D LastLogin, A2.D PINChange FROM PINSAFEJ U LEFT OUTER JOIN PINSAFEN A1 ON U.G = A1.A AND A1.C = 0 LEFT OUTER JOIN PINSAFEN A2 ON U.G = A2.A AND A2.C = 1
```

List login times for a given user

```
SELECT I Username, E DateTime FROM PINSAFEM WHERE A = 0 AND I = ? ORDER BY E
```

Replace the ? with the username in quotes.

List unsuccessful login times for a given user

```
SELECT I Username, E DateTime FROM PINSAFEM WHERE A = 14 AND I = ? ORDER BY E
```

Replace the ? with the username in quotes.

List login times for all users

```
SELECT I Username, E DateTime FROM PINSAFEM WHERE A = 0 ORDER BY I, E
```

List unsuccessful login times for all users

```
SELECT I Username, E DateTime FROM PINSAFEM WHERE A = 14 ORDER BY I, E
```

Count the number of Logins

This gives the total number of authentications in the audit table. (see note above for Audit Log):

```
SELECT COUNT(*) FROM PINSAFEM WHERE A=0
```

Count the number of Failed Logins

This gives the total number of failed authentications in the audit table. (see note above for Audit Log):

```
SELECT COUNT(*) FROM PINSAFEM WHERE A=14
```

Count the number of successful logins over last 30 days

The following query shows the number of successful login attempts over the last 30 days (see note above for Audit Log):

```
SELECT COUNT(*) FROM PINSAFEM WHERE A=0 AND E > DATE_SUB(NOW(), INTERVAL 30 DAY)
```

Count the number of failed logins over last 30 days

The following query shows the number of successful login attempts over the last 30 days (see note above for Audit Log):

```
SELECT COUNT(*) FROM PINSAFEM WHERE A=14 AND E > DATE_SUB(NOW(), INTERVAL 30 DAY)
```

Count the number of logins and login failures per day over the last 30 days

```
select D1 AS Date, L AS Logins, F AS Failures from (select Date(e) D1, count(*) L from pinsafem where a=0 group by date(e)) AS Logins left outer join (select Date(e) D2, count(*) F from pinsafem where a=14 group by date(e)) AS Failures on d1=d2;
```

Count the number of users who have logged in over the last 30 days

```
SELECT COUNT(*) FROM PINSAFEJ U INNER JOIN PINSAFEN A1 ON U.G = A1.A AND A1.C = 3 LEFT OUTER JOIN PINSAFEN A2 ON U.G = A2.A AND A2.C = 0 WHERE A2.D > DATE_SUB(NOW(), INTERVAL 30 DAY);
```

Count the number of times each user has logged in

The following query counts the number of times each user has logged in (see Audit Log note above):

```
SELECT I Username, Count(*) Count FROM PINSAFEM WHERE A = ? GROUP BY I
```

Replace ? by 4 for unlocked, 5 for locked, 2 for self-reset, 6 for admin reset and 0 for successful login.

To look for a specific user, change "GROUP BY I" to "WHERE I = 'name'"

Count the users number of unsuccessful logins and self resets

The following query will give the number of self-reset and unsuccessful logins since the last successful login:

```
SELECT H Username, B FailCount, F ResetCount FROM PINSAFEJ
```

to look for a particular user, add "WHERE H = 'name'"

List the users who have reset their PIN from a particular date

(Includes ChangePIN, ResetPIN and admin reset/resend of the PIN)

```
SELECT DISTINCT PINSAFEJ.H FROM PINSAFEJ INNER JOIN PINSAFEN ON PINSAFEJ.G = PINSAFEN.A WHERE (PINSAFEN.C = 1 OR PINSAFEN.C = 2 OR PINSAFEN.C=6) AND PINSAFEN.D > '2011-11-18 10:30'
```

The report can be customised as follows

The **DISTINCT** option allows each user to be listed only once for the combination of resets.

ChangePIN **PINSAFEN.C = 1**

Self Reset **PINSAFEN.C = 2**

Admin reset **PINSAFEN.C = 6**

List the users who have used ChangePIN from a particular date

```
SELECT PINSAFEJ.H FROM PINSAFEJ INNER JOIN PINSAFEN ON PINSAFEJ.G = PINSAFEN.A WHERE PINSAFEN.C = 1 AND PINSAFEN.D > '2013-12-23 16:30'
```

List the users who have used Self Reset from a particular date

```
SELECT PINSAFEJ.H FROM PINSAFEJ INNER JOIN PINSAFEN ON PINSAFEJ.G = PINSAFEN.A WHERE PINSAFEN.C = 2 AND PINSAFEN.D > '2013-12-23 16:30'
```

List the users who have had an Admin reset from a particular date

```
SELECT PINSAFEJ.H FROM PINSAFEJ INNER JOIN PINSAFEN ON PINSAFEJ.G = PINSAFEN.A WHERE PINSAFEN.C = 6 AND PINSAFEN.D > '2013-12-23 16:30'
```

List user date of creation and last login

The following query will show when the account was created and when the user last connected per day:

```
SELECT U.H Username, A1.D CreationTime, A2.D LoginTime FROM PINSAFEJ U INNER JOIN PINSAFEN A1 ON U.G = A1.A AND A1.C = 3 LEFT OUTER JOIN PINSAFEN A2 ON U.G = A2.A AND A2.C = 0
```

List users who have been created in the last 30 days

The following query lists all users which have been created in the last 30 days:

```
SELECT U.H FROM PINSAFEJ U JOIN PINSAFEN A ON U.G=A.A AND A.C=3 WHERE A.D > DATE_SUB(NOW(), INTERVAL 30 DAY)
```

List users who have been created in the last month

The following query lists all users which have been created in the last month:

```
SELECT U.H FROM PINSAFEJ U JOIN PINSAFEN A ON U.G=A.A AND A.C=3 WHERE A.D > DATE_SUB(NOW(), INTERVAL 1 MONTH)
```

Count the number of users who have never logged in

The following query reports the number of users who have never logged in:

```
SELECT COUNT(*) FROM PINSAFEJ WHERE G NOT IN (SELECT A FROM PINSAFEN WHERE C=0)
```

Count the number of users who have never logged in over the last 30 days

The following query logs the number of users who have not logged in during the last 30 days:

```
SELECT COUNT(*) FROM PINSAFEJ U LEFT OUTER JOIN PINSAFEN A ON U.G = A.A AND A.C=0 WHERE A.D IS NULL OR A.D < DATE_SUB(NOW(), INTERVAL 30 DAY)
```

Count the number of users who have never logged in over the last 180 days but created more than 30 days ago

Since the above query includes new users, who might not have had a chance to log in yet, the following query logs users who were created over 30 days ago, but have not logged in during the last 30 days:

```
SELECT COUNT(*) FROM PINSAFEJ U JOIN PINSAFEN A1 ON U.G = A1.A AND A1.C = 3 LEFT OUTER JOIN PINSAFEN A2 ON U.G = A2.A AND A2.C=0 WHERE (A1.D < DATE_SUB(NOW(), INTERVAL 30 DAY)) AND (A2.D IS NULL OR A2.D < DATE_SUB(NOW(), INTERVAL 180 DAY))
```

List the number of users who have never logged in over the last 180 days but created more than 30 days ago

Since the above query includes new users, who might not have had a chance to log in yet, the following query logs users who were created over 30 days ago, but have not logged in during the last 30 days:

```
SELECT U.H Username FROM PINSAFEJ U JOIN PINSAFEN A1 ON U.G = A1.A AND A1.C = 3 LEFT OUTER JOIN PINSAFEN A2 ON U.G = A2.A AND A2.C=0 WHERE (A1.D < DATE_SUB(NOW(), INTERVAL 30 DAY)) AND (A2.D IS NULL OR A2.D < DATE_SUB(NOW(), INTERVAL 180 DAY))
```

List users connected today who have not previously logged in

This query will show users who connected today, but have not connected before.

```
SELECT U.H Username, A1.D CreationTime, A2.D LoginTime FROM PINSAFEJ U INNER JOIN PINSAFEN A1 ON U.G = A1.A AND A1.C = 3 INNER JOIN PINSAFEN A2 ON U.G = A2.A AND A2.C = 0 WHERE A2.D > DATE_SUB(NOW(), INTERVAL 1 DAY) AND NOT EXISTS(SELECT G FROM PINSAFEM A3 WHERE A3.G=U.H AND A3.E < DATE_SUB(NOW(), INTERVAL 1 DAY))
```

lists all the Users ids, email address, created date, last login, last pin change and repository name

```
SELECT U.H Username, U.E ReposName, T.A Email, A1.D CreateDate, A2.D LoginDate, A3.D PINChange FROM PINSAFEJ U LEFT OUTER JOIN PINSAFEAT ON U.G = T.C LEFT OUTER JOIN PINSAFEN A1 ON U.G = A1.A AND A1.C = 3 LEFT OUTER JOIN PINSAFEN A2 ON U.G = A2.A AND A2.C = 0 LEFT OUTER JOIN PINSAFEN A3 ON U.G = A3.A AND A3.C = 1
```

List PIN expiry for users

```
SELECT U.H USERNAME, ADDDATE(GREATEST(COALESCE(A1.D, '2000-01-01'), COALESCE(A2.D, '2000-01-01'), COALESCE(A3.D, '2000-01-01')), 60) EXPIRY_DATE FROM PINSAFEJ U LEFT OUTER JOIN PINSAFEN A1 ON U.G = A1.A AND A1.C = 1 LEFT OUTER JOIN PINSAFEN A2 ON U.G = A2.A AND A2.C = 6 LEFT OUTER JOIN PINSAFEN A3 ON U.G = A3.A AND A2.C = 3
```

Replace 60 with the actual PIN validity period.

List all Disabled accounts

```
SELECT U.H Name FROM PINSAFEJ U JOIN PINSAFEC S ON U.G = S.C AND S.B = 0 WHERE S.D = 1
```

List all users with inactive, locked, disabled and deleted state

```
SELECT U.H UID, R.B REPOS, S1.D DISABLED, S2.D LOCKED, S3.D DELETED, S4.D INACTIVE FROM PINSAFEJ U JOIN PINSAFEL R ON U.I=R.A LEFT OUTER JOIN PINSAFEC S1 ON U.G=S1.C AND S1.B=0 LEFT OUTER JOIN PINSAFEC S2 ON U.G=S2.C AND S2.B=1 LEFT OUTER JOIN PINSAFEC S3 ON U.G=S3.C AND S3.B=4 LEFT OUTER JOIN PINSAFEC S4 ON U.G=S4.C AND S4.B=5
```

List all users with inactive, locked, disabled and deleted state and show creation, login and PIN change date

```
SELECT U.H UID, R.B REPOS, S1.D DISABLED, S2.D LOCKED, S3.D DELETED, S4.D INACTIVE, A1.D CREATED, A2.D LOGGEDIN, A3.D PINCHANGE FROM PINSAFEJ U JOIN PINSAFEL R ON U.I=R.A LEFT OUTER JOIN PINSAFEC S1 ON U.G=S1.C AND S1.B=0 LEFT OUTER JOIN PINSAFEC S2 ON U.G=S2.C AND S2.B=1 LEFT OUTER JOIN PINSAFEC S3 ON U.G=S3.C AND S3.B=4 LEFT OUTER JOIN PINSAFEC S4 ON U.G=S4.C AND S4.B=5 LEFT OUTER JOIN PINSAFEN A1 ON U.G=A1.A AND A1.C=3 LEFT OUTER JOIN PINSAFEN A2 ON U.G=A2.A AND A2.C=0 LEFT OUTER JOIN PINSAFEN A3 ON U.G=A3.A AND A3.C=1
```

Queries on versions prior to Swivel 3.4

For versions of Swivel earlier than 3.4, the table PINSAFED performed a similar function to that now performed by PINSAFEN, i.e. recording the latest date/time of various activities.

```
SELECT H Username FROM PINSAFEJ WHERE G NOT IN (SELECT DISTINCT A FROM PINSAFED WHERE C=1)
```

Note that the code for login on PINSAFED is C=1, rather than C=0 for PINSAFEN. Therefore, the query to list last login date and time is

```
SELECT U.H Username, A.D LoginTime FROM PINSAFEJ U INNER JOIN PINSAFED A ON U.G = A.A AND A.C = 1 ORDER BY LoginTime
```


Known Issues

When running reports to list locked users, some users may be listed multiple times due to duplicate locks in the database, however the lock count and the number of locks displayed in the Swivel Administration Console will display the correct value for number of locked users.

Troubleshooting

Are we able to isolate which regional Swivel server the user has logged in on from the database?

No not though SQL queries. However this information is available in the Swivel logs and obtainable through a log parser. Depending on how the Swivel agent is configured, there is a 'source' option in the Agent XML. This stores the IP information of the user logging on and there is a field in the database to store this information.

Locked user account list shows some users as not locked

The locked user count will report not only those that are flagged as locked but those accounts that have more than the number of failed authentications. When the user who has exceeded the maximum login attempts but whose account is not marked locked, next tries to login, the account will be marked as locked.