



# 基幹システムでも使える MySQL EE(商用版) 活用方法 第1回 セキュリティ対策編

MySQL Global Business Unit  
MySQL Principal Sales Consult, MySQL Global Business Unit

# SAFE HARBOR STATEMENT

以下の事項は、弊社の一般的な製品の方向性に関する概要を説明するものです。  
また、情報提供を唯一の目的とするものであり、いかなる契約にも組み込むことはできません。  
以下の事項は、マテリアルやコード、機能を提供することをコミットメントするものではない為、  
購買決定を行う際の判断材料になさらないで下さい。

オラクル製品に関して記載されている機能の開発、リリースおよび時期については、  
弊社の裁量により決定されます。

# MySQL におけるセキュリティ対策

Basic

➡ 1 MySQL 概要

➡ 2 データベースセキュリティ概要

Community

➡ 3 MySQLセキュリティ基本設定

Enterprise

➡ 4 MySQL Enterpriseによるセキュリティ拡張機能

➡ 5 技術サポート& オラクル製品との動作保証

# MySQL 概要



# MySQLのコミュニティ版と商用版

Commercial Editionをご利用頂く事でツールやサポートも利用可能です。

Community Edition (GPL)	Commercial Edition
<ul style="list-style-type: none"><li>• MySQL Community Server</li><li>• MySQL Cluster</li><li>• MySQL GUI管理ツール</li><li>• MySQLコネクタ (JDBC, ODBC, etc.)</li><li>• ドキュメント</li><li>• フォーラム</li></ul>	<ul style="list-style-type: none"><li>• Standard Edition</li><li>• <b>Enterprise Edition</b></li><li>• MySQL Cluster Carrier Grade Edition</li><li>• 商用ライセンス (組み込み用)</li><li>• プロフェッショナルサービス<ul style="list-style-type: none"><li>- トレーニング、コンサルティング、サポート</li></ul></li></ul>

- コミュニティ版ソフトウェアはGPLでソースコードも公開し提供
- 商用版は、付加価値として技術サポートや管理機能、拡張機能を有償で提供

参照: [MySQL Downloads](#)



# MySQL Enterprise Edition のサービスカテゴリ

拡張機能でセキュリティ対応をサポート



## 拡張機能

- 拡張性
- 高可用性
- セキュリティ
- 監査
- 暗号化



## 管理ツール

- 監視
- バックアップ
- 開発
- 管理
- マイグレーション



## サポート

- 技術サポート
- コンサルティングサポート
- オラクル製品との動作保証



	MySQL Editions		
	Standard Edition	Enterprise Edition	Cluster CGE
<b>機能概要</b>			
MySQL Database	✓	✓	✓
MySQL Connectors	✓	✓	✓
MySQL Replication	✓	✓	✓
MySQL Fabric		✓	✓
MySQL Partitioning		✓	✓
MySQL Utilities		✓	✓
Storage Engine: MyISAM, InnoDB	✓	✓	✓
Storage Engine: NDB (ndbcluster)			✓
MySQL Workbench SE/EE*	✓	✓	✓
MySQL Enterprise Monitor*		✓	✓
MySQL Enterprise Backup*		✓	✓
MySQL Enterprise Authentication (外部認証サポート) *		✓	✓
MySQL Enterprise Audit (ポリシーベース監査機能) *		✓	✓
MySQL Enterprise Encryption (非対称暗号化)*		✓	✓
MySQL Enterprise Firewall (SQLインジェクション対策)*		✓	✓
MySQL Enterprise Scalability (スレッドプール) *		✓	✓
MySQL Enterprise High Availability (HAサポート) *		✓	✓
Oracle Enterprise Manager for MySQL*		✓	✓
MySQL Cluster Manager (MySQL Cluster管理) *			✓
MySQL Cluster Geo-Replication			✓

	MySQL Editions		
	Standard SE	Enterprise EE	Cluster CGE
<b>Oracle Premium Support</b>			
24時間365日サポート	✓	✓	✓
インシデント数無制限	✓	✓	✓
ナレッジベース	✓	✓	✓
バグ修正&パッチ提供	✓	✓	✓
コンサルティングサポート	✓	✓	✓
<b>オラクル製品との動作保証</b>			
Oracle Linux	✓	✓	✓
Oracle VM	✓	✓	✓
Oracle Solaris	✓	✓	✓
Oracle Enterprise Manager		✓	✓
Oracle GoldenGate		✓	✓
Oracle Data Integrator		✓	✓
Oracle Fusion Middleware		✓	✓
Oracle Secure Backup		✓	✓
Oracle Audit Vault and Database Firewall		✓	✓

※最新の対比表は、[MySQL Editionsのサイト](#)を参照下さい。



# MySQL Enterprise Edition 管理ツールと拡張機能概要

## MySQL Enterprise Edition

MySQL Enterprise Monitor	複数サーバの一括管理、クエリ性能分析
MySQL Enterprise Backup	高速なオンラインバックアップ、ポイントインタイムリカバリ
MySQL Enterprise Scalability	Thread Poolプラグインによる性能拡張性の向上
MySQL Enterprise Authentication	LDAPやWindows Active Directoryとの外部認証と統合管理
MySQL Enterprise Audit	ユーザ処理の監査、Oracle DBと同じツールで管理も可能
MySQL Enterprise Encryption	非対称暗号化( <a href="#">公開鍵暗号</a> )の業界標準機能を提供
MySQL Enterprise Firewall	SQLインジェクション対策 / オペレーションミス回避
Oracle Enterprise Manager for MySQL	Oracle Enterprise ManagerからMySQLを統合管理可能
Oracle Premier Support	24x7, インシデント無制限、コンサルティングサポート

# データベースセキュリティ概要

# データベースにおける脆弱性



- **設定の不備**

- コントロール設定とデフォルト設定の変更

- **過剰な特権アカウント**

- 権限に関するポリシー設定

- **弱いアクセス制御**

- 管理者専用アカウントの分離

- **弱い認証**

- 強力なパスワードの強制

- **弱い監査**

- コンプライアンス & 監査ポリシー実装

- **暗号化の欠如**

- データ, バックアップ, & ネットワーク暗号化

- **適切な資格情報やキー管理**

- [mysql\\_config\\_editor](#), [Key Vaults](#)の利用

- **安全でないバックアップ**

- バックアップの暗号化

- **モニタリングの欠如**

- ユーザー、オブジェクト、セキュリティ監視

- **アプリケーションのエスケープ処理不備**

- [Database Firewall](#)

# データベースへの攻撃

- **SQLインジェクション**

→対策: DB Firewall, White List, Input Validation

- **バッファオーバーフロー**

→対策: Frequently apply Database Software updates, DB Firewall, White List, Input Validation

- **ブルートフォースアタック**

→対策: lock out accounts after a defined number of incorrect attempts.

- **ネットワーク盗聴**

→対策: Require SSL/TLS for all Connections and Transport

- **マルウェア**

→対策: Tight Access Controls, Limited Network IP access, Change default settings

# データベースへの悪意のある行為

- **情報漏えい**: Obtain credit card and other personal information
  - 防御: Encryption – Data and Network, Tighter Access Controls
- **DoS攻撃**: Run resource intensive queries
  - 防御: Resource Usage Limits – Set various limits – Max Connections, Sessions, Timeouts, ...
- **特権の昇格**: Retrieve and use administrator credentials
  - 防御: Stronger authentication, Access Controls, Auditing
- **なりすまし**: Retrieve and use other credentials
  - 防御: Stronger account and password policies
- **改竄**: Change data in the database, Delete transaction records
  - 防御: Tighter Access Controls, Auditing, Monitoring, Backups

# DBAによる対策

- アクセス権を得る必要のあるユーザーのみが、権限を取得出来るよう確実にする。
- ユーザーおよびアプリケーションの権限を適切にコントロールする。
- ユーザーおよびアプリケーションが、データにアクセスできる場所を制限する。
- 何が、いつ発生したかを適切にモニタリングしておく。
- バックアップが安全でセキュアに取得されている事を確実にしておく。
- 攻撃面を最小化しておく。







# MySQLセキュリティ基本設定

- データベースを安全に利用するためのセキュリティ対策 -

# MySQLセキュリティ概要

MySQL Security

Authentication  
(認証)

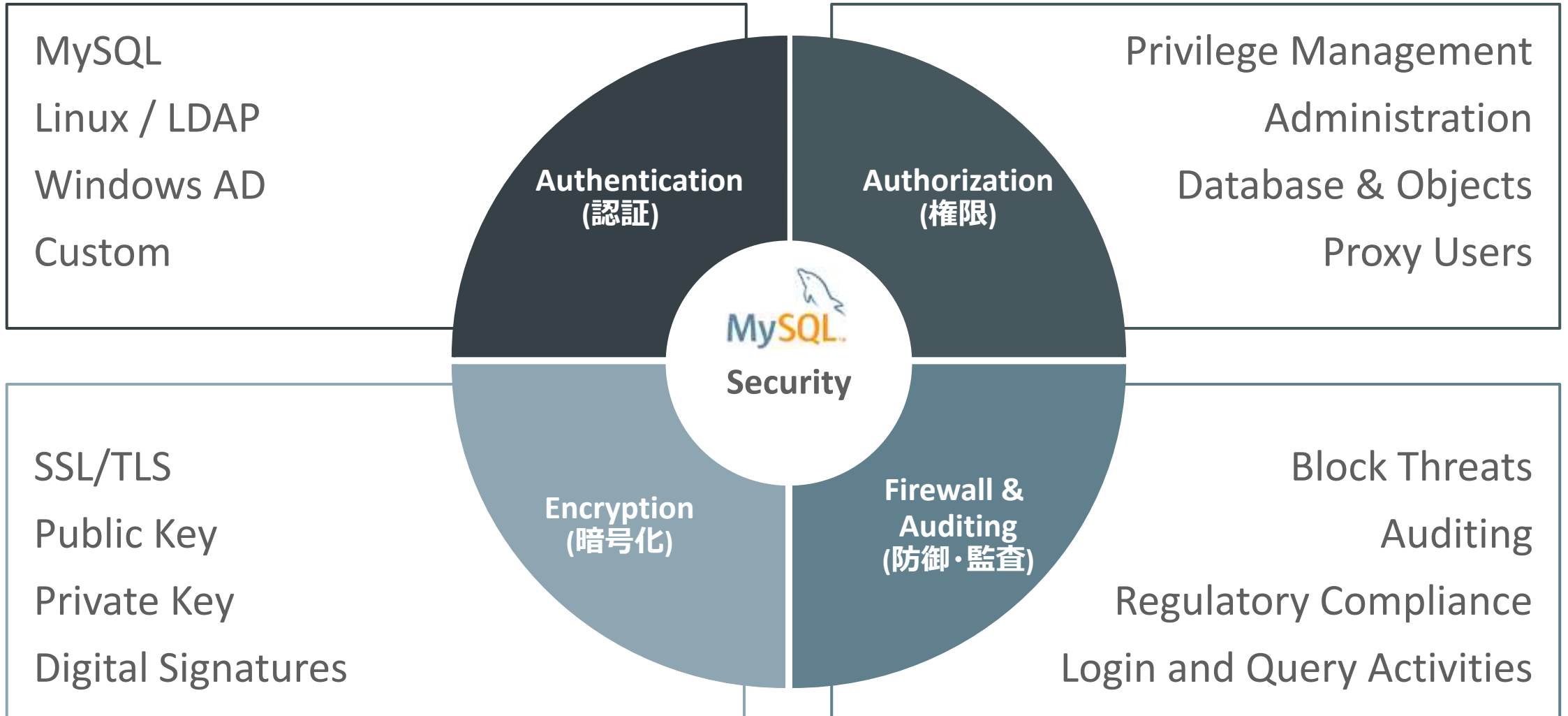
Authorization  
(権限)

Encryption  
(暗号化)

Firewall  
(ファイヤーウォール)

Auditing  
(監査)

# MySQLセキュリティ概要



# MySQLにおけるユーザーアカウントの付与

ユーザー名とパスワードのみを使用してユーザーを認証する他のほとんどのデータベースとは異なり、MySQLでは、ユーザーを認証する際に、追加のlocationパラメータを使用します。このlocationパラメータは、通常、ホスト名、IPアドレスまたはワイルドカード (%) です。MySQLでは、この追加のパラメータを使用して、データベースへのユーザー・アクセスをドメイン内の特定のホストに制限する場合があります。また、これによって、接続元のホストに応じて、ユーザーに異なるパスワードと一連の権限を適用できます。

```
root@localhost [mysql]>select user,host,password from mysql.user;
```

user	host	password
root	localhost	*A41ECFBE1191DDE4713F2B6F5A6CD5D0D0D5DC35
root	centos01	*A41ECFBE1191DDE4713F2B6F5A6CD5D0D0D5DC35
GTID_SSL_USER	192.168.56.%	*84AE91CE95DAA59A02F658041290FEECF1BEE392
admin	%	*A41ECFBE1191DDE4713F2B6F5A6CD5D0D0D5DC35
root	192.168.56.%	*A41ECFBE1191DDE4713F2B6F5A6CD5D0D0D5DC35
audited_user	192.168.56.0/255.255.255.0	*CED32AC7E202DBC0858C9E8935348E7ED3E083D4
GTID_USER	192.168.56.%	*A41ECFBE1191DDE4713F2B6F5A6CD5D0D0D5DC35
sec_password	192.168.56.%	*6FDB5274DE9FC54EDC81BD5909DEBE93CF739D48

ユーザー名	アクセス元 Host / IP	パスワード
-------	-----------------	-------

# 不要なアカウントの削除

~MySQL 5.6x

[Drop the Extra Roots] root@localhost [mysql]>DROP USER root@'127.0.0.1', root@':::1';

```
root@localhost [mysql]>SELECT user,host,plugin,password,password_expired FROM mysql.user;
```

user	host	plugin	password	password_expired
root	localhost		*A41ECFBE1191DDE4713F2B6F5A6CD5D0D0D5DC35	N
root	centos01		*A41ECFBE1191DDE4713F2B6F5A6CD5D0D0D5DC35	N
root	127.0.0.1			N
root	:::1			N
admin	%	mysql_native_password	*A41ECFBE1191DDE4713F2B6F5A6CD5D0D0D5DC35	N
root	192.168.56.%	mysql_native_password	*A41ECFBE1191DDE4713F2B6F5A6CD5D0D0D5DC35	N
audited_user	192.168.56.0/255.255.255.0	mysql_native_password	*CED32AC7E202DBC0858C9E8935348E7ED3E083D4	N
GTID_USER	192.168.56.%	mysql_native_password	*A41ECFBE1191DDE4713F2B6F5A6CD5D0D0D5DC35	N

8 rows in set (0.00 sec)

```
root@localhost [mysql]>DROP USER root@'127.0.0.1', root@':::1';
Query OK, 0 rows affected (0.01 sec)
```

```
root@localhost [mysql]>SELECT user,host,plugin,password,password_expired FROM mysql.user;
```

user	host	plugin	password	password_expired
root	localhost		*A41ECFBE1191DDE4713F2B6F5A6CD5D0D0D5DC35	N
root	centos01		*A41ECFBE1191DDE4713F2B6F5A6CD5D0D0D5DC35	N
admin	%	mysql_native_password	*A41ECFBE1191DDE4713F2B6F5A6CD5D0D0D5DC35	N
root	192.168.56.%	mysql_native_password	*A41ECFBE1191DDE4713F2B6F5A6CD5D0D0D5DC35	N
audited_user	192.168.56.0/255.255.255.0	mysql_native_password	*CED32AC7E202DBC0858C9E8935348E7ED3E083D4	N
GTID_USER	192.168.56.%	mysql_native_password	*A41ECFBE1191DDE4713F2B6F5A6CD5D0D0D5DC35	N

6 rows in set (0.00 sec)

## 不要なアカウントの削除



# mysql\_secure\_installationを利用した不要アカウント削除

- 1) ルートアカウントのパスワードを設定することができます。
- 2) ローカルホスト以外からアクセス可能なルートアカウントを削除する事が出来ます。
- 3) 匿名のユーザーアカウントを削除することができます。
- 4) testデータベースを削除する事が出来ます。(Defaultで全てのユーザーがアクセス可能)

```
[root@Centos03 mysql]# ./bin/mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MySQL
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MySQL to secure it, we'll need the current
password for the root user. If you've just installed MySQL, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

Enter current password for root (enter for none):
OK, success! (by setting password, moving on...)

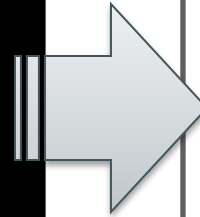
Setting the root password ensures that nobody can log into the MySQL
root user without the proper authorisation.

Set root password? [Y/n] Y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables:
... Success!

By default, a MySQL installation has an anonymous user, allowing anyone
to log into MySQL without having to have a user account created for
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] Y
... Success!
```

**rootパスワード設定**  
**不要なアカウントの削除**  
**Rootのリモートログイン不可**  
**Testデータベースの削除**



```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
+-----+
3 rows in set (0.00 sec)
```

```
mysql> select user,host,password from mysql.user;
+-----+-----+-----+
| user | host | password |
+-----+-----+-----+
| root | localhost | *A41ECFBE1191DDE4713F2B6F5A6CD5D0D0D5DC35 |
| root | 127.0.0.1 | *A41ECFBE1191DDE4713F2B6F5A6CD5D0D0D5DC35 |
| root | ::1 | *A41ECFBE1191DDE4713F2B6F5A6CD5D0D0D5DC35 |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

参照: [4.4.5 mysql\\_secure\\_installation](#)



# Automatic Password Expiration

Since: MySQL 5.6.6~

```
root@localhost [mysql]>ALTER USER 'sec_password'@'192.168.56.%' PASSWORD EXPIRE;
```

```
root@localhost [mysql]>select user,host,password,password_expired from user;
```

user	host	password	password_expired
root	localhost	*A41ECFBE1191DDE4713F2B6F5A6CD5D0D0D5DC35	N
root	centos01	*A41ECFBE1191DDE4713F2B6F5A6CD5D0D0D5DC35	N
admin	%	*A41ECFBE1191DDE4713F2B6F5A6CD5D0D0D5DC35	N
root	192.168.56.%	*A41ECFBE1191DDE4713F2B6F5A6CD5D0D0D5DC35	N
audited_user	192.168.56.0/255.255.255.0	*CED32AC7E202DBC0858C9E8935348E7ED3E083D4	N
GTID_USER	192.168.56.%	*A41ECFBE1191DDE4713F2B6F5A6CD5D0D0D5DC35	N
sec_password	192.168.56.%	*BEA2508E06B227E173EAA0F221C2C838C7545E41	Y

7 rows in set (0.00 sec)

MySQL 5.6 introduces password-expiration capability, to enable database administrators to expire account passwords and **require users to reset their password.**

```
[admin@CentOS02 ~]$ mysql -h 192.168.56.101 -u sec_password -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 5.6.21-enterprise-commercial-advanced-log

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.
```

Oracle is a registered trademark of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

```
SET PASSWORD FOR sec_password@"192.168.56.%"=PASSWORD('Password_2014-');
```

※ MySQL 5.7.6 以降は、ALTER USER がパスワード設定で推奨されます。

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

```
sec_password@192.168.56.101 [(none)]> use mysql
ERROR 1820 (HY000): You must SET PASSWORD before executing this statement
sec_password@192.168.56.101 [(none)]>
```

# Account Management (File\_privの設定)

Since: MySQL 5.0.38~

インストール後の作業：OS上のファイルへのアクセスを制限

- 1) 専用のディレクトリを作成
- 2) mysqlをディレクトリのオーナーに設定
- 3) my.cnfの[mysqld]セクションに"secure\_file\_priv"を追加
- 4) mysqlを再起動

```
admin@192.168.56.201 [mysql]> SELECT @@global.secure_file_priv;
+-----+
| @@global.secure_file_priv |
+-----+
| NULL                       |
+-----+
1 row in set (0.00 sec)
```

**[設定前]**

```
admin@192.168.56.201 [mysql]> SELECT LOAD_FILE('/etc/passwd')\G
***** 1. row *****
LOAD_FILE('/etc/passwd'): root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
```

```
[root@GA01 mysql_share]# cat /etc/my.cnf | grep secure_file
secure_file_priv = /home/mysql/mysql_share
[root@GA01 mysql_share]# /etc/init.d/mysql.server restart
Shutting down MySQL.. SUCCESS!
Starting MySQL. SUCCESS!
[root@GA01 mysql_share]#
```

**[設定後]**

```
admin@192.168.56.201 [(none)]> SELECT @@global.secure_file_priv;
+-----+
| @@global.secure_file_priv |
+-----+
| /home/mysql/mysql_share/  |
+-----+
1 row in set (0.00 sec)
```

```
admin@192.168.56.201 [(none)]> SELECT LOAD_FILE('/etc/passwd')\G
***** 1. row *****
LOAD_FILE('/etc/passwd'): NULL
1 row in set (0.01 sec)
```

# Account Management (Password Validation Plugin)

Since: MySQL 5.6.6~

インストール後の作業：パスワードポリシーのインストール

英数字の混在を強制する、文字数をN文字以上にする、特定のキーワードはパスワードに指定できなくする、といった対応が可能

```
[mysql]> install plugin validate_password soname 'validate_password.so';
```

```
root@localhost [mysql]>SHOW VARIABLES LIKE 'validate_password%';
```

Variable_name	Value
validate_password_dictionary_file	/usr/local/mysql/data/band_dictionary.txt
validate_password_length	8
validate_password_mixed_case_count	1
validate_password_number_count	1
validate_password_policy	STRONG
validate_password_special_char_count	1

```
6 rows in set (0.00 sec)
```

```
root@localhost [mysql]>
```

**例) 8文字以下、小文字のみ、数字未入力、辞書に登録済みの文字はパスワードとしては不適切な為、ERROR 1819で拒否される。**

```
[admin@CentOS01 ~]$ perror 1819;
```

```
MySQL error code 1819 (ER_NOT_VALID_PASSWORD): Your password does not satisfy the current policy requirements
```

```
[admin@CentOS01 ~]$
```

# SSL による通信の暗号化

## インストール後の作業：SSLの設定

MySQL 5.0.10 is bundled with yaSSL for enabling SSL.  
MySQL 5.6.6, SSL support is included by default

```
[admin@CentOS02 ~]$ mysql -h 192.168.56.101 -u admin -p --ssl-ca=/usr/local/mysql/ssl/sql-ssl-cert.pem
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 20
Server version: 5.6.21-enterprise-commercial-advanced-log MySQL Enterprise Server - Advanced Edition (Commercial)

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
```

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

```
admin@192.168.56.101 [(none)]> show status like 'SSL_version';
```

Variable_name	Value
Ssl_version	TLSv1

```
root@localhost [(none)]> SHOW VARIABLES LIKE '%ssl%';
```

Variable_name	Value
have_openssl	YES
have_ssl	YES
ssl_ca	/usr/local/mysql/ssl/sql-ssl-cert.pem
ssl_capath	
ssl_cert	/usr/local/mysql/ssl/sql-gtid-cert.pem
ssl_cipher	
ssl_cr1	
ssl_cr1path	
ssl_key	/usr/local/mysql/ssl/sql-ssl-repl-key.pem

```
GTID_SSL_USER@192.168.56.101 [(none)]> show status like 'ssl_cipher';
```

Variable_name	Value
ssl_cipher	DHE-RSA-AES256-SHA

1 row in set (0.00 sec)

オンプレミスとパブリッククラウド間でデータベース操作やレプリケーションなどをSSLを使い安全に行う事が可能。



# ホストアクセスブロック

```
root@localhost [performance_schema]>desc host_cache;
```

Field	Type	Null	Key	Default
IP	varchar(64)	NO		NULL
HOST	varchar(255)	YES		NULL
HOST_VALIDATED	enum('YES','NO')	NO		NULL
SUM_CONNECT_ERRORS	bigint(20)	NO		NULL
COUNT_HOST_BLOCKED_ERRORS	bigint(20)	NO		NULL
COUNT_NAMEINFO_TRANSIENT_ERRORS	bigint(20)	NO		NULL
COUNT_NAMEINFO_PERMANENT_ERRORS	bigint(20)	NO		NULL
COUNT_FORMAT_ERRORS	bigint(20)	NO		NULL
COUNT_ADDRINFO_TRANSIENT_ERRORS	bigint(20)	NO		NULL
COUNT_ADDRINFO_PERMANENT_ERRORS	bigint(20)	NO		NULL
COUNT_FCRDNS_ERRORS	bigint(20)	NO		NULL
COUNT_HOST_ACL_ERRORS	bigint(20)	NO		NULL
COUNT_NO_AUTH_PLUGIN_ERRORS	bigint(20)	NO		NULL
COUNT_AUTH_PLUGIN_ERRORS	bigint(20)	NO		NULL
COUNT_HANDSHAKE_ERRORS	bigint(20)	NO		NULL
COUNT_PROXY_USER_ERRORS	bigint(20)	NO		NULL
COUNT_PROXY_USER_ACL_ERRORS	bigint(20)	NO		NULL
COUNT_AUTHENTICATION_ERRORS	bigint(20)	NO		NULL
COUNT_SSL_ERRORS	bigint(20)	NO		NULL
COUNT_MAX_USER_CONNECTIONS_ERRORS	bigint(20)	NO		NULL
COUNT_MAX_USER_CONNECTIONS_PER_HOUR_ERRORS	bigint(20)	NO		NULL
COUNT_DEFAULT_DATABASE_ERRORS	bigint(20)	NO		NULL
COUNT_INIT_CONNECT_ERRORS	bigint(20)	NO		NULL
COUNT_LOCAL_ERRORS	bigint(20)	NO		NULL
COUNT_UNKNOWN_ERRORS	bigint(20)	NO		NULL
FIRST_SEEN	timestamp	NO		NULL
LAST_SEEN	timestamp	NO		NULL
FIRST_ERROR_SEEN	timestamp	NO		NULL
LAST_ERROR_SEEN	timestamp	NO		NULL

1 row in set (0.00 sec)

```
root@localhost [performance_schema]>select @@global.max_connect_errors;
```

```
-----+-----+
| @@global.max_connect_errors |
+-----+-----+
| 100 |
+-----+-----+
1 row in set (0.00 sec)
```

The default is 100 as of MySQL 5.6.6, 10 before that.

- ※ skip\_name\_resolveが設定されている場合は有効にならない。
- ※ Unlock:は、[FLUSH HOSTS](#) か [mysqladmin flush-hosts](#) コマンドにて実行
- ※ [FLUSH HOSTS](#) and [TRUNCATE TABLE host\\_cache](#) は同じ結果になります。
- ※ 基本的にはローカルネットワークアクセスコントロール用途



# MySQLの機能を利用した暗号化(共通鍵暗号) Since: MySQL 4.0.2~

```
root@localhost > select (HEX(AES_ENCRYPT("AES暗号化-個人情報01", 'password'))) into @AES_ENC;
Query OK, 1 row affected (0.00 sec)

root@localhost > select @AES_ENC;
+-----+
| @AES_ENC |
+-----+
| D8F991170C3468696E4D963AE4A8E2A1D6404D7066F30A5D0419C0EC80D0602B |
+-----+
1 row in set (0.00 sec)

root@localhost > select AES_DECRYPT(UNHEX(@AES_ENC), 'password');
+-----+
| AES_DECRYPT(UNHEX(@AES_ENC), "password") |
+-----+
| AES暗号化-個人情報01 |
+-----+
1 row in set (0.00 sec)

root@localhost >
```

AES\_ENCRYPT(str,key\_str),  
AES\_DECRYPT(crypt\_str,key\_str)が  
MySQL標準で使用可能なものの中で、  
暗号的に最も安全な暗号化関数でした。

標準128ビットのキーの長さを使用したエンコード  
暗号化・復号化文字列は共に同じものを利用

AES暗号化は128ビットのキーの長さを使用したエンコードを行いますが、256ビットまで延長する事が出来ます。





# MySQLの機能を利用した暗号化(共通鍵暗号) Since: MySQL 5.6.17~

```
root@localhost > SELECT @@session.block_encryption_mode;
+-----+
| @@session.block_encryption_mode |
+-----+
| aes-128-ecb                       |
+-----+
1 row in set (0.00 sec)

root@localhost > SET block_encryption_mode = 'aes-256-cbc';
Query OK, 0 rows affected (0.00 sec)

root@localhost > SELECT @@session.block_encryption_mode;
+-----+
| @@session.block_encryption_mode |
+-----+
| aes-256-cbc                       |
+-----+
1 row in set (0.00 sec)

root@localhost > SET @key_str = SHA2('password',512);
Query OK, 0 rows affected (0.00 sec)

root@localhost > SELECT @key_str;
+-----+
| @key_str                           |
+-----+
| b109f3bbbc244eb82441917ed06d618b9008dd09b3befd1b5e07394c706a8bb980b1d7785e5976ec049b46df5f13 |
+-----+
1 row in set (0.00 sec)

root@localhost > SET @init_vector = RANDOM_BYTES(16);
Query OK, 0 rows affected (0.00 sec)

root@localhost > SET @crypt_str = HEX(AES_ENCRYPT("AES暗号化-個人情報01",@key_str,@init_vector));
Query OK, 0 rows affected (0.00 sec)

root@localhost > SELECT @crypt_str;
+-----+
| @crypt_str                           |
+-----+
| 6BFA4275B9167BC976C13CDE295FDF7EADB41DB27C6BD75CFA6F44255422AB28 |
+-----+
1 row in set (0.00 sec)

root@localhost > SELECT AES_DECRYPT(UNHEX(@crypt_str),@key_str,@init_vector);
+-----+
| AES_DECRYPT(UNHEX(@crypt_str),@key_str,@init_vector) |
+-----+
| AES暗号化-個人情報01                               |
+-----+
1 row in set (0.00 sec)

root@localhost >
```

AES\_ENCRYPT(str,key\_str[,init\_vector])  
AES\_DECRYPT(crypt\_str,key\_str[,init\_vector])

128, 192 or 256 bitの暗号化方法が設定可能  
暗号化の必要性和暗号化によるオーバーヘッドにより選択

標準256ビットのキーの長さを使用したエンコード  
暗号化・復号化文字列は共に同じものを利用

# MySQL 5.7.7 セキュリティの強化

MySQL 5.7

## ユーザ管理とセキュリティ

- mysql\_install\_db コマンド非推奨
  - mysqld の --initialize または --initialize-insecure オプションで初期化
- CREATE USER 文と ALTER USER 文にオプション追加
  - SSL, PASSWORD EXPIRE, ACCOUNT [LOCK | UNLOCK]
- mysql.user テーブルの password 列が authentication\_string に変更
- SET PASSWORD 文および PASSWORD() 関数が非推奨
  - ALTER USER 文での設定を推奨
- ENCRYPT, DES\_ENCRYPT, DES\_DECRYPT 関数非推奨 AES 推奨

```
mysql_install_db --user=mysql
```



```
mysqld --initialize --user=mysql
```

# Security - Encryption, Passwords, Installation

MySQL 5.7.xx

- AES 256 Encryption (Default)
  - パスワードローテーションポリシー
    - インスタンス全体、ユーザー単位で設定可能
    - 未使用時に、アカウントを無効化
- ※ 5.7 RCでは、Default設定が360日

## [ Global Configuration ]

```
SET GLOBAL default_password_lifetime = 180;
```

## [ Individual user accounts ]

```
ALTER USER joro@localhost PASSWORD EXPIRE INTERVAL 90 DAY;
```

```
ALTER USER joro@localhost PASSWORD EXPIRE NEVER;
```

## [ Account Lock ]

```
ALTER USER joro@localhost ACCOUNT LOCK;
```

## [ SSL & Limit Connection ]

```
ALTER USER joro@localhost REQUIRE SSL WITH  
MAX_CONNECTIONS_PER_HOUR 20;
```

max_user_connections	int(11) unsigned	NO	0
plugin	char(64)	NO	mysql_native_password
authentication_string	text	YES	NULL
password_expired	enum('N', 'Y')	NO	N
password_last_changed	timestamp	YES	NULL
password_lifetime	smallint(5) unsigned	YES	NULL
account_locked	enum('N', 'Y')	NO	N

- Deployment: デフォルトで安全に無人インストールを行う事が可能  
インストール時にランダムなパスワードを設定/匿名のアカウントを削除  
テストアカウント, スキーマ(test), デモファイルは基本インストールにおいても作成されなくなりました

# Security – SSL, Proxy User

- SSL

- Enabled by default
- Auto-detection of existing keys and certs
- Auto generation of keys and certs when needed
- New helper utility: mysql\_ssl\_rsa\_setup

- Extended Proxy User Support

- Added Built-in Authentication Plugins support for Proxy Users
- Allows multiple users to share a single set of managed privileges

Proxyユーザーは、MySQL5.5からサポートされている機能ですが、追加の認証プラグインが必要でした。MySQL5.7.7RCからは、MySQL標準のmysql\_native\_password とsha256\_password認証をサポートする事で、より柔軟にアカウント管理を行う事が出来るようになりました。

```
SET @@global.check_proxy_users = ON;
```

```
SET @@global.mysql_native_password_proxy_users = ON;
```

MySQL 5.7.xx

```
root@localhost [mysql]> CREATE USER proxy_base@localhost;
Query OK, 0 rows affected (0.00 sec)

root@localhost [mysql]> CREATE USER admin_1@localhost;
Query OK, 0 rows affected (0.00 sec)

root@localhost [mysql]> CREATE USER admin_2@localhost;
Query OK, 0 rows affected (0.00 sec)

root@localhost [mysql]> GRANT PROXY ON proxy_base@localhost TO admin_1@localhost;
Query OK, 0 rows affected (0.07 sec)

root@localhost [mysql]> GRANT PROXY ON proxy_base@localhost TO admin_2@localhost;
Query OK, 0 rows affected (0.00 sec)

root@localhost [mysql]> GRANT SELECT ON USER01.* TO proxy_base@localhost;
Query OK, 0 rows affected (0.00 sec)
```

```
[admin@misc01 ~]$ /usr/local/mysql/bin/mysql -u admin_1 -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6
Server version: 5.7.7-rc-log MySQL Community Server (GPL)

Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.


Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

admin_1@localhost [(none)]> SELECT USER(), CURRENT_USER(), @@session.proxy_user;
+-----+-----+-----+
| USER() | CURRENT_USER() | @@session.proxy_user |
+-----+-----+-----+
| admin_1@localhost | proxy_base@localhost | 'admin_1'@'localhost' |
+-----+-----+-----+
1 row in set (0.00 sec)

admin_1@localhost [(none)]> SHOW GRANTS;
+-----+
| Grants for proxy_base@localhost |
+-----+
| GRANT USAGE ON *.* TO 'proxy_base'@'localhost' |
| GRANT SELECT ON USER01.* TO 'proxy_base'@'localhost' |
+-----+
2 rows in set (0.00 sec)

admin_1@localhost [(none)]>
```

A dolphin is captured in mid-leap, emerging from the surface of a clear blue ocean. The dolphin's body is sleek and dark, with water droplets glistening on its skin. The background is a deep, vibrant blue, and the dolphin's reflection is visible in the water below. The overall scene is dynamic and energetic.

# MySQL Enterpriseによるセキュリティ拡張機能

- データ保護とコンプライアンス対応の強化 -



# MySQL Enterprise Editionによるデータ保護



## MySQL Enterprise Backup

- オンラインバックアップ/リカバリ
- クラウドストレージへバックアップ
- 差分バックアップ & ポイントインタイムリカバリ
- バックアップデータ暗号化



## MySQL Enterprise Security

- 外部認証との統合 (PAM, Windows, LDAP, etc.)
- MySQL Enterprise Monitorでのセキュリティアドバイザ



## MySQL Enterprise Encryption

- 最少1024Bitからの暗号化
- 公開鍵方式 / 非対称暗号
- 暗号的ハッシュによる電子署名、照合および妥当性確認



## MySQL Enterprise Audit

- 接続, ログイン, SQL実行の記録
- ポリシーベースのフィルタリングおよびログ切り替え
- オラクルの監査仕様に準拠したXMLベースの出力



# MySQL Enterprise Backup



バックアップ&リストアオプション：AES暗号化 (64 hexadecimal digit)

- 暗号化キーを直接指定する場合

```
shell# mysqlbackup --backup-image=/backups/image.enc --encrypt ¥  
> --key=23D987F3A047B475C900127148F9E0394857983645192874A2B3049570C12A34 ¥  
> --backup-dir=/var/tmp/backup backup-to-image
```

- ファイルに格納した暗号化キーを使用する場合

```
shell# mysqlbackup --backup-image=/backups/image.enc --encrypt ¥  
> --key-file=/meb/key --backup-dir=/var/tmp/backup backup-to-image
```

- ファイルに格納した復号化キーを直接指定する場合

```
shell# mysqlbackup --backup-image=/backups/image.enc --decrypt ¥  
> --key-file=/meb/key --backup-dir=/backups/extract-dir extract
```

参照: [Chapter 8 Encryption for Backups](#)

# MySQL Enterprise Security

EE

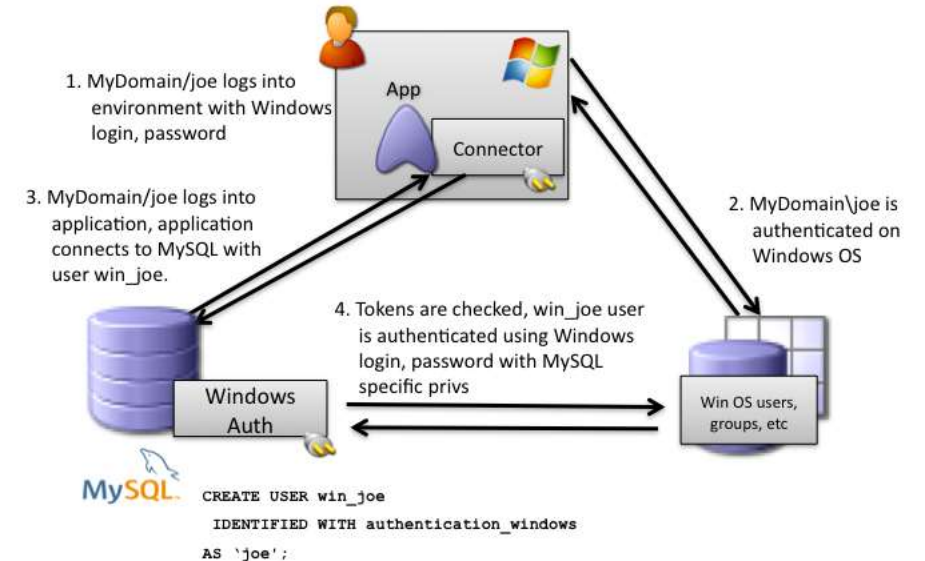
- SSLに対応した通信
- アクセスコントロール
  - 外部認証方式へのアクセス
  - 標準認証インターフェース対応 (Unix、LDAP、Kerberosなど)
  - プロキシ/非プロキシユーザー
- 監査と監視
  - MySQLのセキュリティアドバイザ
  - Oracle Audit Vaultとの互換性 (ログフォーマット)
- Oracle Database FirewallによるFirewallサポートも可能



# MySQL Enterprise Authentication

## 外部認証のサポート

- PAM (Pluggable Authentication Modules)
  - 外部認証方式へのアクセス
  - 標準のインタフェース (Unix, LDAP, Kerberosなど)
  - プロキシ/非プロキシユーザー
- Windows
  - ネイティブWindowsサービス (WAD) へのアクセス
  - Windowsにログイン済みユーザを認証
- プラガブル認証API



MySQLアプリケーションを既存のセキュリティ・インフラストラクチャ/SOPと統合

# MySQL Enterprise Audit

## ポリシーベースの監査機能を提供

- ログオン、クエリーの情報監査可能
- ユーザーがポリシーを設定可能：フィルタリング、ログローテーション
- 動的に設定を変更可能：Audit設定時にサーバの再起動が不要
- Oracleの仕様に合わせXMLベースの監査ログを出力
- サイズに基づいた監査ログファイルの自動ローテーション
- XML ベースの監査ログストリーム
- MySQL 5.5のAudit APIを使って実装 / MySQL 5.5.28 以上で使用可能

**コンプライアンス対応等で監査が必要なアプリケーションでもMySQLを利用可能**

# MySQL Enterprise Audit

## 管理者



```
mysql> INSTALL PLUGIN audit_log SONAME 'audit_log.so';

mysql> SHOW VARIABLES LIKE 'audit_log%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| audit_log_buffer_size | 1048576 |
| audit_log_file | audit.log |
| audit_log_flush | OFF |
| audit_log_policy | ALL |
| audit_log_rotate_on_size | 1044480 |
| audit_log_strategy | SYNCHRONOUS |
+-----+-----+
```

1. DBA enables Audit plugin



## Joe (ユーザー)



```
shell> mysql -h joeshost -u joe -p
Enter password: *****
```

```
mysql> SELECT * FROM joes_table;
+-----+-----+
| FIRST_NAME | LAST_NAME |
+-----+-----+
| Joe | User |
+-----+-----+
```

2. User Joe connects and runs a query

## 3. Joe's connection & query logged

EE

```
<?xml version="1.0" encoding="UTF-8"?>
<AUDIT>
  <AUDIT_RECORD
    TIMESTAMP="2012-08-02T14:52:12"
    NAME="Audit"
    SERVER_ID="1"
    VERSION="1"
    STARTUP_OPTIONS="--port=3306"
    OS_VERSION="i686-Linux"
    MYSQL_VERSION="5.5.28-debug-log"/>
  <AUDIT_RECORD
    TIMESTAMP="2012-08-02T14:52:41"
    NAME="Connect"
    CONNECTION_ID="1"
    STATUS="0"
    USER="joe"
    PRIV_USER="root"
    OS_LOGIN=""
    PROXY_USER=""
    HOST="SERVER1"
    IP="127.0.0.1"
    DB="joes_db"/>
  <AUDIT_RECORD
    TIMESTAMP="2012-08-02T14:53:45"
    NAME="Query"
    CONNECTION_ID="1"
    STATUS="0"
    SQLTEXT="SELECT * FROM joes_table;"/>
</AUDIT>
```

WHO

WHERE

WHEN

WHAT

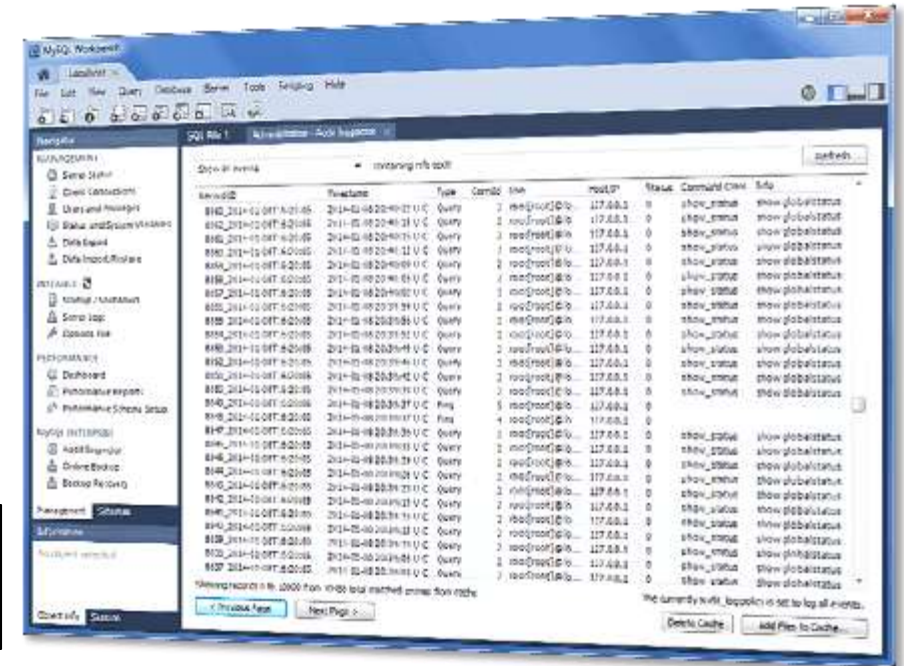
# Audit Log Filtering

## 監査ログのフィルタリング

- Event filtering by account name
  - SET GLOBAL audit\_log\_include\_accounts='root@localhost';
  - SET GLOBAL audit\_log\_exclude\_accounts='secure@localhost';
- Event filtering by status
  - SET GLOBAL audit\_log\_statement\_policy=ALL;
  - SET GLOBAL audit\_log\_connection\_policy=ERRORS;
- Better instrumentation and statistics
  - SHOW STATUS LIKE 'Audit\_log\_events\_filtered';

### MySQL Utilities

mysqlauditadmin      audit log maintenance utility  
 mysqlauditgrep      audit log search utility





# MySQL Enterprise Encryption

EE

- MySQLの暗号化ライブラリ
  - AES256による対称鍵暗号
  - 公開鍵 / 非対称鍵暗号
- キーの管理
  - 公開鍵および秘密鍵の生成
  - 鍵交換方式: RSA, DSA, DH
- 署名とデータの検証
  - 電子署名、検証、妥当性確認のための暗号学的ハッシュ関数
- Oracle Key Vaultとの統合



## The maximum key length (OpenSSLによる制約)

RSA	16,384
DSA	10,000
DH	10,000

# MySQL Enterprise Encryption



非対称暗号: RSA, DS

```
個人情報管理者>SELECT "=== [RSA] Private and Public鍵の復号化"
+-----+
| STEP1 |
+-----+
| === [RSA] Private and Public鍵の復号化 |
+-----+
1 row in set (0.00 sec)

個人情報管理者>CREATE TABLE priv_key ('key' VARCHAR(2048))
Query OK, 1 row affected (0.04 sec)
Records: 1 Duplicates: 0 Warning: 0

個人情報管理者>SELECT * FROM priv_key
Query OK, 1 row affected (0.01 sec)

個人情報管理者>CREATE TABLE pub_key ('key' VARCHAR(2048))
Query OK, 1 row affected (0.02 sec)
Records: 1 Duplicates: 0 Warning: 0

個人情報管理者>SELECT * FROM pub_key
Query OK, 1 row affected (0.00 sec)

個人情報管理者>
```

```
個人情報管理者>SELECT @priv_key\G
***** 1. row *****
@priv_key: -----BEGIN RSA PRIVATE KEY-----
MIICXAIBAAKBgQDX7aUxdyXb0mw2HtBhFyUskdcG/eLbEKU6Uz0xux61wMHkRJ+t
OHRYbGSV1tRAYQt0TxXGwbg17kFpGU6oYT1484EpHRUrX0v4SRAPsa9aC/pq00UC
XsoYPJBBkMYE96hJsKpTd8Mo5RPCvRQK/rJEHNIS8SwPL7drCpLmXHJEdwIDAQAB
AoGAXq9E2vYGUaXCwdCtS4XctTiWc+hsy+b2rSbHFMGa69REsZYt9sVkr0mIqfP0
Su7DGRN81xUnc8gZkr6YMVnA2yxjLrDdoy7rvWQTMLE0e3DVxHU5Xwefbczo9R65
b0t+YUJrLL5Ywys3/Y1yh767gmEoLD1VYoHZDhF51PBudHECQQD314h62kkwNx+P
0i4foKkw+oAcM3rARpyzZKVvSyRy3ZZY1K1judHzEwBAkkGntY6CqfahWmTseAKk
V1Q51AXjAkeA30LXT50ad/ire0cJ2cHBoGRC4+uh8U0TxYrfWpfz6YbwaOF34qPv
Pcz5Ve84aYE3QxwWUje6Fm0bYCD4T8UrXQJAS1SA0bkUvdfer to/qxkvkjGyIkVG
NdE9HBI8JFRfxehGSbbXsxfHMv1iVwBRm6LC/PE/rKMxp1hEGsgcEwkgVwJARg3f
KagOKh7pDyLPwHg/nwhYZNQHGIIQq9A1DUFx1vx0MSpyU1ZTC+Q1ch07U0KYvB0m9
JUU1CNxrfppZ0A36MQJBAKnhG++SWwx20TsBAz8TpYE8sES0QyPevHD/XY7wisce
cGE7x28G0QMoonXf1MtqUT//kGGpLdkTpzjHE/tq3as=
-----END RSA PRIVATE KEY-----

個人情報管理者>SELECT @pub_key\G
***** 1. row *****
@pub_key: -----BEGIN RSA PUBLIC KEY-----
MIGJAoGBANftptF3Jds6bDYe0GEXJSyR1wb94tsQpTpTPTG7HrXAweREn604dFhs
ZJXW1EBhC3RPFcbBuDWWQwkZTqhhOXjzGskdFstc6/hJEA+xr1oL+mo7RQJeyhg8
kEGQxgT3qEmwq1N3wyj1E9y9FAR+skQc0hLxLckvt2sKkuZccKR3AgMBAAE=
-----END RSA PUBLIC KEY-----
```

復号化用

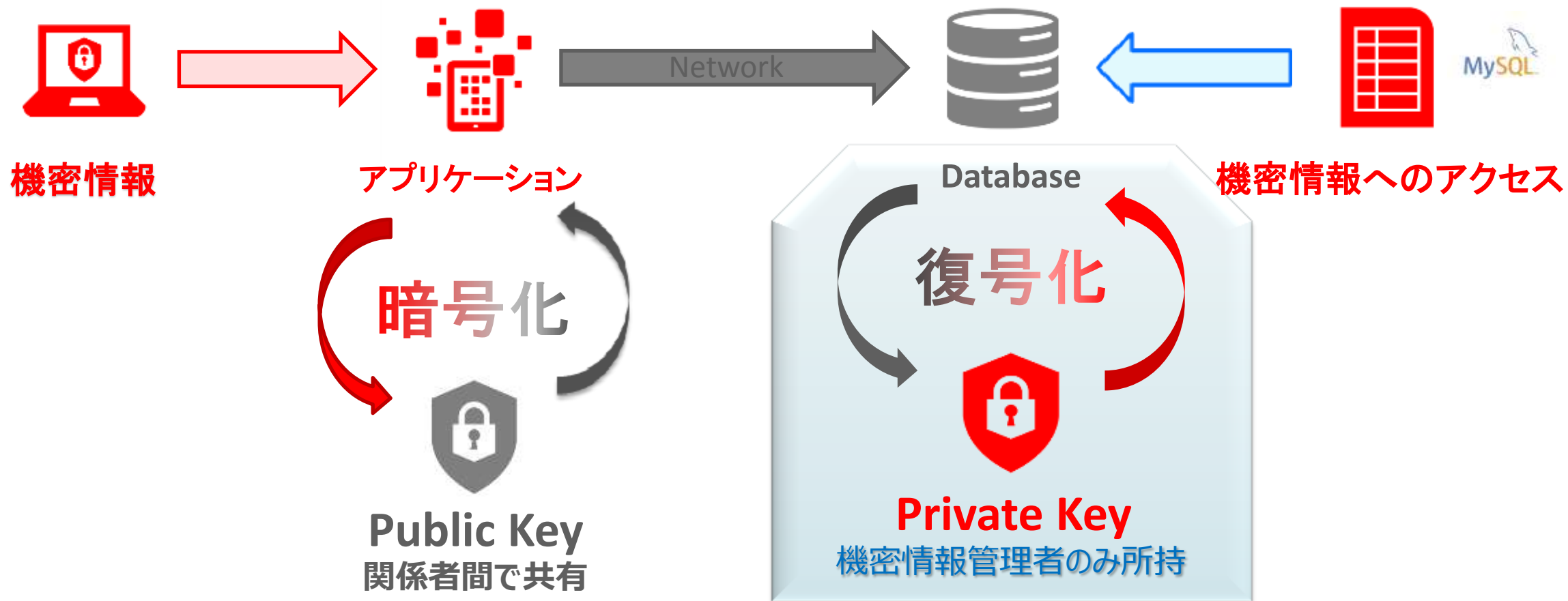
暗号化用



# 例：機密情報の取り扱いフロー

Handling Sensitive Information

EE



Private, Public鍵を分けて管理出来る環境においては、よりセキュアにデータを保護をする事が可能。

# MySQL Enterprise Encryption



MySQL Enterprise Encryption は非対称暗号化(公開鍵暗号)の業界標準機能を提供  
機密データの保護や HIPAA, SOX 法, PCI DSSなどの規制要件の遵守に役立てる事が可能。

```
開発チーム>CREATE TABLE enc_pub SELECT HEX(ASYMMETRIC_ENCRYPT('RSA','MASAカード オラ次郎 1234-5678-9012-3456 99/99', @pub_key)) as 'Card_Info';  
Query OK, 1 row affected (0.07 sec)  
Records: 1 Duplicates: 0 Warnings: 0
```

Public鍵による暗号化

```
開発チーム>select * from enc_pub\G  
***** 1. row *****  
Card_Info: 0ECFEF85397A941293630ECF40FA2345C66783E4C8131357EE06DBFC4939FB0B85B35E67A33E62F1764CA6158210EFC26B16C094BE92E5B7F11E22FE365E77D96239C0  
AD1AA7AD0AF5792D10A435C73AC65CE36BA200C5A0568757F0A424AEDC9D032FEA3EC6DCED13A5FF85F57E21818606F01437D94240EC4503BB3932B89  
1 row in set (0.00 sec)
```

暗号化されたデータ

暗号化されたデータをテープへの保存

データの暗号化により、  
データをより安全に保管する事が可能



# MySQL Enterprise Firewall

EE



- SQLインジェクション対策、リアルタイムで保護
  - ホワイトリストモデル、実行されるクエリーを分析してホワイトリストと照合
- 学習してホワイトリストを自動作成
  - ユーザー毎に、SQL実行パターンを記録して自動的にホワイトリストを作成
- 不審なアクセスをブロック
  - ポリシーに違反するトランザクションを検知し、ブロック
- 透過的
  - アプリケーションを変更する必要無し

```
fw_user@localhost [test]> select * from FW_DEMO where id = 3;
+-----+-----+
| ID | title
+-----+-----+
| 3 | test firewall35.6.24-enterprise-commercial-advanced-log |
+-----+-----+
1 row in set (0.00 sec)

fw_user@localhost [test]> select * from FW_DEMO;
ERROR 1045 (28000): Statement was blocked by Firewall
fw_user@localhost [test]>
```



Select \*.\* from employee where id=22

Select \*.\* from employee where id=22 or 1=1



✓ Allow & Log

✗ Block & Log



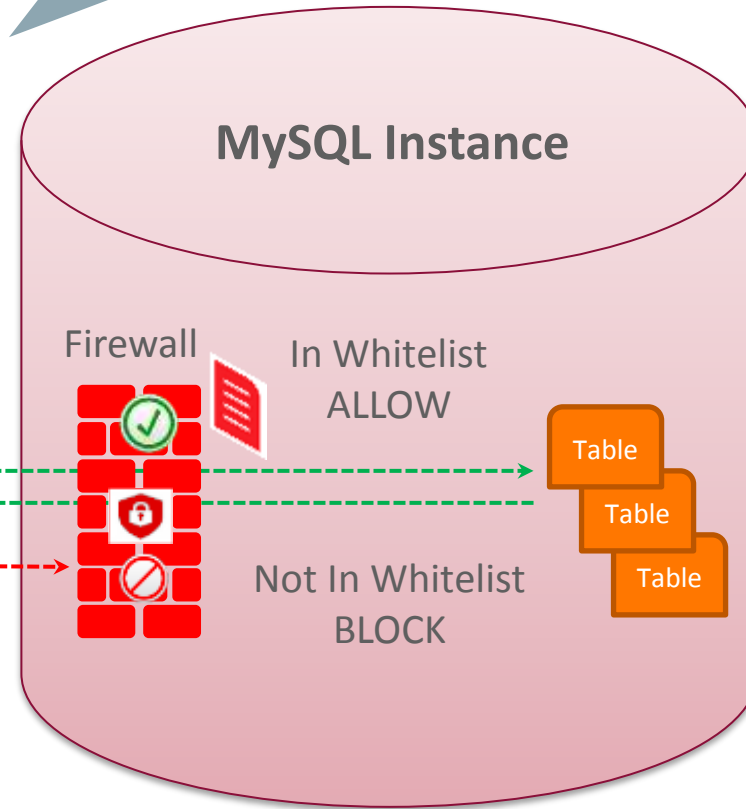
# Firewall Overview

SQL Injection Attack  
Via Browser



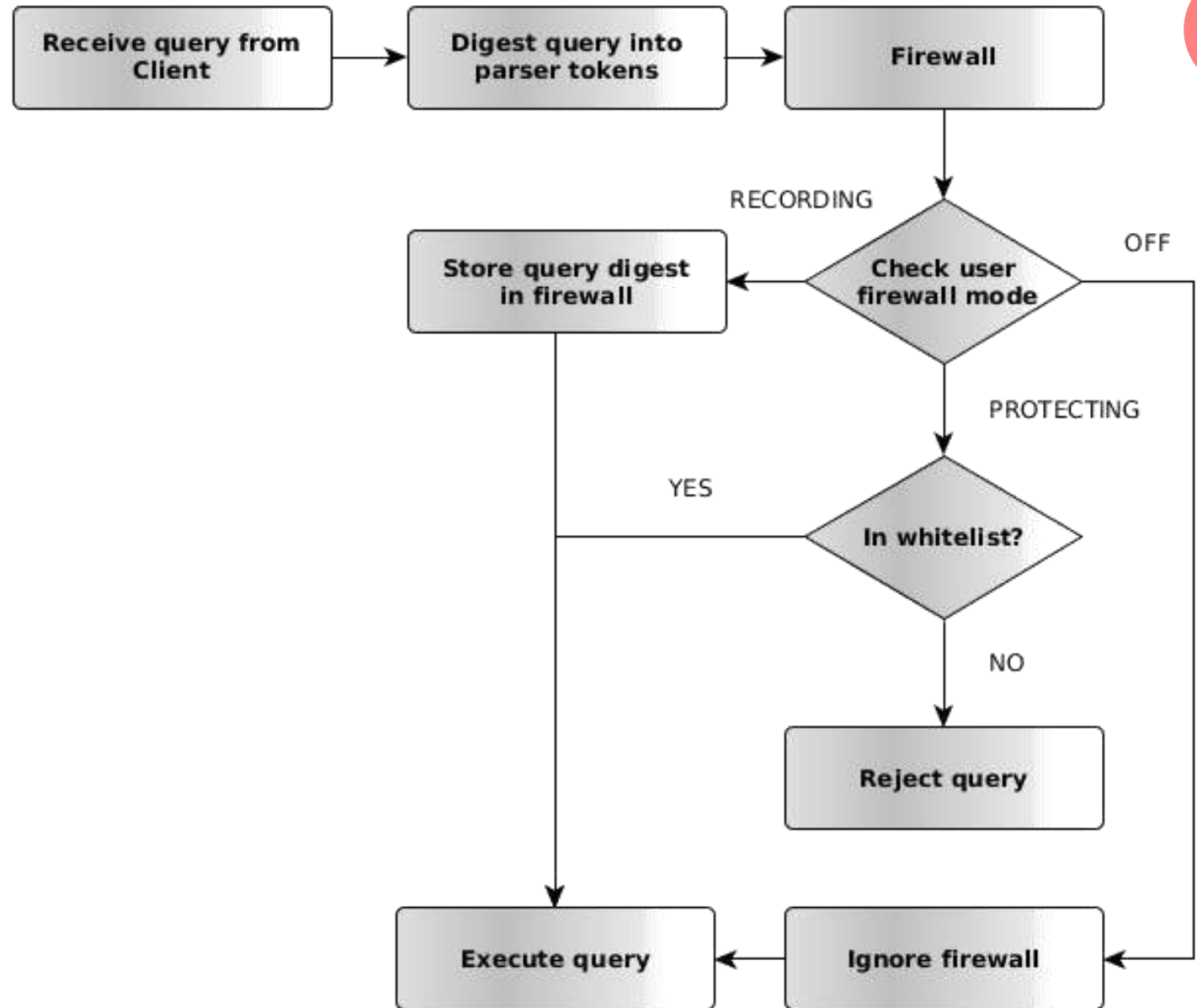
Inbound  
SQL traffic

社内ポータルにおいても、アプリケーションユーザー毎に実行可能なSQLを分ける事で、ユーザー毎に実行可能なSQLを分ける事も可能。





# Firewall Workflow



# MySQL Enterprise Firewall Details

- ユーザー毎にFirewall設定 (アプリケーション、ユーザー、管理者)
- ユーザー毎の設定 (例)

学習

```
CALL sp_set_firewall_mode('fw_user@localhost','RECORDING');
```

防御

```
CALL sp_set_firewall_mode('fw_user@localhost','PROTECTING');
```

無効化

```
CALL sp_set_firewall_mode('fw_user@localhost','OFF');
```

初期化

```
CALL sp_set_firewall_mode('fw_user@localhost', 'RESET');
```

メモ: Replication利用時のSlave側における対応:

```
SELECT read_firewall_whitelist('fw_user@localhost','RECORDING') FROM mysql.firewall_whitelist;
```

# Per User Firewall White Lists

EE

The screenshot shows the Oracle Enterprise Edition (EE) 'Users and Privileges' administration interface. The window title is 'Administration - Users and Privileges'. The interface is for 'Sandbox 5.6.23 (Commercial)'. The main section is 'Users and Privileges', with a sub-section 'User Accounts' on the left and 'Details for account msandbox@%' on the right.

The 'User Accounts' table lists the following users and their firewall rule counts:

User	From Host	State	# Rules
meb_user	localhost	Protecting	1
root_ssl	%	Learning	0
msandbox_ro	127.%	Off	0
msandbox_rw	127.%	Off	0
msandbox_ro	localhost	Off	0
msandbox_rw	localhost	Off	0
rsandbox	127.%	Off	0
msandbox	%	Off	0
msandbox	localhost	Learning	0
root	localhost	Off	0

The 'Details for account msandbox@%' section shows the 'Firewall Rules' tab selected. The 'Firewall Rules' section has a dropdown menu set to 'Off', a 'Manage Rules' button, and a 'Clear' button. Below this, there is a 'Copy queries' section with a dropdown menu set to 'meb\_user', a 'Copy From...' button, and a 'Copy To...' button.

# What's the whitelist look like?

```
root@GA01 [mysql]> SELECT userhost, substr(rule,1,80) FROM mysql.firewall_whitelist WHERE userhost= 'fw_user@localhost';
```

userhost	substr(rule,1,80)
fw_user@localhost	SELECT SCHEMA ( )
fw_user@localhost	SHOW TABLES
fw_user@localhost	INSERT INTO `FW_DEMO` ( `title` ) VALUES ( `concat` ( ? , @@version ) )
fw_user@localhost	DESC `FW_DEMO`
fw_user@localhost	SELECT * FROM `FW_DEMO` WHERE `id` = ?
fw_user@localhost	CREATE TABLE `FW_DEMO` ( `ID` INTEGER UNSIGNED AUTO_INCREMENT NOT NULL PRIMARY K
fw_user@localhost	SHOW SCHEMAS
fw_user@localhost	SELECT SYSTEM_USER ( )
fw_user@localhost	SELECT @@version_comment LIMIT ?
fw_user@localhost	SELECT * FROM SYSTEM_USER

```
10 rows in set (0.00 sec)
```

```
root@GA01 [mysql]> SELECT * FROM information_schema.mysql_firewall_users;
```

USERHOST	MODE
fw_user@localhost	PROTECTING

```
1 row in set (0.00 sec)
```

**White Listに登録されたSQLステートメント  
White Listを使ってプロテクトされているアカウントの状態**

# What happens when SQL is blocked?

- クライアントアプリケーションへのエラーレスポンス

```
fw_user@localhost [test]> select * from FW_DEMO where id = 1;
```

```
+----+-----+
| id | text |
+----+-----+
|  1 | Firewall demo - 1 |
+----+-----+
1 row in set (0.00 sec)
```

```
fw_user@localhost [test]> select * from FW_DEMO;
ERROR 1045 (28000): Statement was blocked by Firewall
fw_user@localhost [test]> TRUNCATE TABLE FW_DEMO;
ERROR 1045 (28000): Statement was blocked by Firewall
fw_user@localhost [test]>
```



- エラーログへ記録 (MySQL5.7ではSYSLOG連携も)

```
2015-04-30 11:15:15 'ACCESS DENIED for fw_user@localhost. Reason: No match in whitelist. SELECT SYSTEM_USER ( ) '
2015-04-30 11:15:21 'ACCESS DENIED for fw_user@localhost. Reason: No match in whitelist. SELECT * FROM `FW_DEMO` '
2015-04-30 11:15:21 'ACCESS DENIED for fw_user@localhost. Reason: No match in whitelist. SELECT SYSTEM_USER ( ) '
2015-04-30 11:15:35 'ACCESS DENIED for fw_user@localhost. Reason: No match in whitelist. TRUNCATE TABLE `FW_DEMO` '
```

# Monitoring the Firewall

## インクリメントカウンタ

### Firewall Status Counters

```
admin@localhost [test]> SHOW STATUS LIKE 'Firewall%';
```

variable_name	value
Firewall_access_denied	15
Firewall_access_granted	1
Firewall_cached_entries	2

```
3 rows in set (0.00 sec)
```

```
admin@localhost [test]>
```

ブロックされると、Firewall\_access\_deniedが  
カウントアップされていきます



A high-speed photograph of a dolphin leaping from the water. The dolphin is captured mid-air, its body arched as it moves from the bottom right towards the top left. The water around the dolphin is splashing and creating white foam, contrasting with the deep blue of the ocean. The dolphin's skin appears wet and glistening. The background is a clear, deep blue sky.

技術サポート& オラクル製品との動作保証

# MySQL Enterprise Support

- 最大のMySQLのエンジニアリングおよびサポート組織
- MySQL開発チームによるサポート
- 29言語で世界クラスのサポートを提供
- メンテナンス・リリース、バグ修正、パッチ、アップデートの提供
- 24時間x365日サポート
- 無制限サポート・インシデント
- MySQLコンサルティング・サポート



Get immediate help for any MySQL issue, plus expert advice

# MySQL Supportの特徴

- 「パフォーマンス・チューニング」や「SQLチューニング」まで通常サポートの範囲内
  - コンサルティングサポートが含まれており、「クエリ・レビュー」、「パフォーマンス・チューニング」、「レプリケーション・レビュー」、「パーティショニング・レビュー」などに対応可能
  - 詳細はこちらを参照下さい  
<http://www-jp.mysql.com/support/consultative.html>
- ソースコードレベルでサポート可能
  - ほとんどのサポートエンジニアがソースを読めるため、対応が早い
  - 開発エンジニアとサポートエンジニアも密に連携している

# MySQL Supportの特徴

- **物理サーバー単位課金**

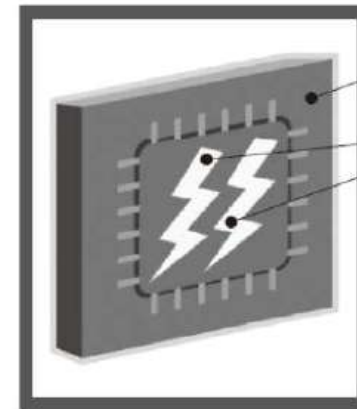
- CPU数、コア数に依存しない価格体系
- ※ 4CPUまで(コア数は制限無し)

- **オラクルのライフタイムサポート**

- 詳細はこちらを参照下さい

<http://www.oracle.com/jp/support/lifetime-support/index.html>

<http://www-jp.mysql.com/support/>



**プロセッサ(CPU/物理CPU)**

「モノ」として存在するプロセッサ・チップを指します。「物理CPU」と表記する場合があります。

**コア(プロセッサ・コア)**

物理的なCPUチップの内部にある演算処理の中核部分(通常は外部から見えませんが、本図では模式的に船隻の絵で表現しています)。「プロセッサ・コア」と呼ぶ場合があります。

**ソケット(スロット)**

コンピュータにプロセッサ(CPU)を搭載するための端子。プロセッサ(CPU)が搭載されていないソケットを空きソケットと呼ぶ場合があります。

# MySQL & オラクル製品との動作保証

- Oracle Linux
- Oracle VM
- Oracle Solaris
- Oracle Clusterware
- Oracle Secure Backup
- Oracle Enterprise Manager
- Oracle Fusion Middleware
- Oracle GoldenGate
- Oracle Audit Vault & Database Firewall
- MyOracle Online Support

**MySQL Integrates into your Oracle Environment**



# Get Started Today!

## MySQL Enterprise Edition Trial



**30日間トライアル**

Oracle Software Delivery Cloud

<http://edelivery.oracle.com/>

製品パックを選択: “MySQL Database”

製品マニュアル: <http://dev.mysql.com/doc/index-enterprise.html>

事例紹介: <http://www.mysql.com/why-mysql/case-studies/#ja-5-0>

## Contact a MySQL Sales Rep



[MySQL お問い合わせ窓口]

電話: 0120-065556

【受付時間】平日 9:00-12:00/13:00-18:00  
(祝日及び年末年始休業日を除きます)

メール: [MySQL-Sales\\_jp\\_grp@oracle.com](mailto:MySQL-Sales_jp_grp@oracle.com)

URL: <http://www.mysql.com/about/contact/>



# まとめ

- 1 MySQLでは、Community Editionにおいても、データベースのセキュリティに必要なアカウントや暗号化オプションが用意されています。  
また、MySQL5.7においては更に初期設定、パスワードローテーションポリシー、Proxyユーザー等の設定がより強化されています。
- 2 MySQL Commercial Editionにおいては、追加のソフトウェアやアプライアンス機器等を利用せずに、MySQLの拡張機能を有効にする事で、セキュリティ対策に必要な「統合認証」、「監査」、「暗号化(1024 bit以上)」、「データベース ファイヤーウォール」機能等を利用する事が可能です。
- 3 運用面においては、サポートチームを利用頂く事により、24x365の無制限サポートやコンサルティング・サポートを受ける事で、DB構成の最適化及び検証や調査にかかる、運用工数を大幅に削減する事が可能です。

有難うございました

ORACLE®