6.3.6 Password Expiration Policy

MySQL enables database administrators to expire account passwords manually, and to establish a policy for automatic password expiration.

To expire a password manually, the database administratior uses the **ALTER USER** statement:

```
ALTER USER 'jeffrey'@'localhost' PASSWORD EXPIRE;
```

This operation marks the password expired in the corresponding **mysql.user** table row.

Automatic password expiration is available in MySQL 5.7.4 and later. The **mysql.user** table indicates for each account when its password was last changed, and the server automatically the server treats the password as expired at client connection time if it is past its permitted lifetime. This works with no explicit manual password expiration.

The **default_password_lifetime** system variable defines the global automatic password expiration policy. It applies to accounts that use MySQL built-in authentication methods (accounts that use an authentication plugin of**mysql_native_password**, **mysql_old_password**, or**sha256_password**).

The default global policy is that passwords have a lifetime of 360 days. To change the policy, change the value of The **default_password_lifetime**. If the value is a positive integer, it indicates the permitted password lifetime in days. A value of 0 disables automatic expiration.

Examples:

- To establish a global policy that passwords have a lifetime of approximately six months, start the server with these lines in an option file:

  ```
  [mysqld]
  default_password_lifetime=180
  ```

- To establish a global policy such that passwords never expire, set **default_password_lifetime** to 0:

  ```
  [mysqld]
  default_password_lifetime=0
  ```

- **default_password_lifetime** can also be changed at runtime (this requires the **SUPER** privilege):

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

No matter the global policy, it can be overridden for individual accounts with **ALTER USER**:

- Require the password to be changed every 90 days:

  ```
  ALTER USER 'jeffrey'@'localhost' PASSWORD EXPIRE INTERVAL 90 DAY;
  ```

- Disable password expiration:

```
ALTER USER 'jeffrey'@'localhost' PASSWORD EXPIRE NEVER;
```

- Defer to the global expiration policy:

```
ALTER USER 'jeffrey'@'localhost' PASSWORD EXPIRE DEFAULT;
```

These `ALTER USER` statements update the corresponding `mysql.user` table row.

When a client successfully connects, the server determines whether the account password is expired:

- The server checks whether the password has been manually expired and, if so, restricts the session.
- Otherwise, the server checks whether the password is past its lifetime according to the automatic password expiration policy. If so, the server considers the password expired and restricts the session.

A client session operates in restricted mode if the account password was expired manually or if the password is considered past its lifetime per the automatic expiration policy. In restricted mode, operations performed within the session result in an error until the user establishes a new account password:

```
mysql> SELECT 1;
ERROR 1820 (HY000): You must SET PASSWORD before executing this statement

mysql> ALTER USER USER() IDENTIFIED BY 'new_password';
Query OK, 0 rows affected (0.01 sec)

mysql> SELECT 1;
+---+
| 1 |
+---+
| 1 |
+---+
1 row in set (0.00 sec)
```

This restricted mode of operation permits `SET` statements, which is useful before MySQL 5.7.6 if `SET PASSWORD` must be used instead of `ALTER USER` and the account password has a hashing format that requires `old_passwords` to be set to a value different from its default.

It is possible for an administrative user to reset the account password, but any existing sessions for the account remain restricted. A client using the account must disconnect and reconnect before statements can be executed successfully.

> **Note**
>
> It is possible to “reset” a password by setting it to its current value. As a matter of good policy, it is preferable to choose a different password.