

Let's go step by step into **Orphaned Users in SQL Server** — what they are, why they happen, and how to fix them.

◆ What is an Orphaned User?

- In SQL Server, a **Database User** is mapped to a **Login** at the instance level (in sys.server_principals).
- When the login is deleted, or its **SID** (Security Identifier) no longer matches, the database user remains but **loses the link to the login**.
- This "orphaned" user cannot log in to the database, even if the login exists at the instance level.

📁 In short: **Login exists in the instance, User exists in the DB, but their SIDs do not match.**

◆ Common Causes

1. **Database Restore/Attach** on a different SQL Server instance (login SIDs differ).
2. **Login deleted** at the instance level, but DB user still exists.
3. **Migration** of databases across environments without migrating logins.
4. **Manual changes** to logins or mismatched SID creation.

◆ How to Identify Orphaned Users

Run this query inside the database:

-- List orphaned users

```
EXEC sp_change_users_login 'Report';
```

📁 This returns users that have no matching login.

(⚠️ Note: sp_change_users_login is deprecated, but still works until SQL Server 2012+; newer method is below.)

Modern approach (SQL Server 2012 and above):

```
SELECT dp.name AS DBUser, dp.sid AS DBUserSID
FROM sys.database_principals dp
LEFT JOIN sys.server_principals sp ON dp.sid = sp.sid
WHERE dp.type IN ('S','U','G') -- SQL user, Windows user/group
AND sp.sid IS NULL;
```

◆ Fixing Orphaned Users

1. Re-map User to Existing Login

If login exists but SID mismatch:

```
ALTER USER [dbusername] WITH LOGIN = [loginname];
```

Example:

```
ALTER USER [JohnUser] WITH LOGIN = [JohnLogin];
```

2. Create Missing Login for User

If login doesn't exist at all:

```
CREATE LOGIN [JohnLogin] WITH PASSWORD = 'StrongPassword!';
```

```
ALTER USER [JohnUser] WITH LOGIN = [JohnLogin];
```

3. Using sp_change_users_login (Deprecated but handy)

-- Fix single orphaned user by mapping to existing login

```
EXEC sp_change_users_login 'Auto_Fix', 'dbusername', NULL, 'StrongPassword!';
```

-- Example

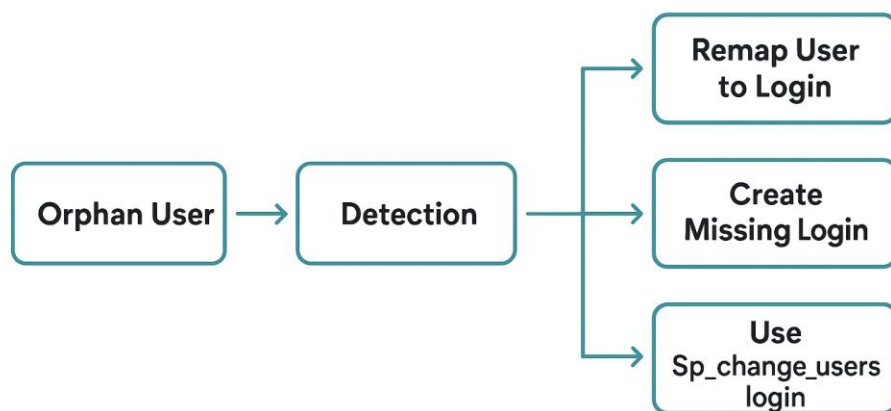
```
EXEC sp_change_users_login 'Auto_Fix', 'JohnUser', NULL, 'MyP@sswOrd!';
```

◆ Best Practices to Avoid Orphaned Users

1. **Backup and restore logins** during migrations:
-- Script out logins with SID and password hash
sp_help_revlogin;
2. Use **contained databases** in SQL Server 2012+ (users don't depend on server-level logins).
3. Always use ALTER USER ... WITH LOGIN in modern versions.
4. Maintain a **login-to-user mapping document** in multi-server setups.

✅ Summary:

An orphaned user occurs when a database user loses its link with a server login (due to SID mismatch or missing login). You can fix it by remapping the user to an existing login, creating a new login, or using sp_change_users_login (deprecated).



mpps.com