

When You Connected As Labuser Find All Objects Which Names Starts With Db_ . Connect As System Find All

When You Connected As Labuser Find All Objects Which Names Starts With Db_ . Connect As System Find All

Understanding how to efficiently locate database objects based on naming conventions is fundamental for database administrators and developers. Whether you're managing a development environment or handling a production database, being able to quickly identify objects such as tables, views, procedures, or other database objects that follow specific naming patterns can streamline troubleshooting, auditing, and maintenance tasks. In particular, when you connect as a user like Labuser or as System, the methods to find objects starting with a specific prefix like 'Db_' become essential tools in your arsenal.

This article explores the techniques and best practices for finding all objects whose names start with 'Db_' when connected as different user roles, specifically Labuser and System. We will delve into the SQL queries, system views, and scripts that enable efficient searches, along with tips for managing large and complex databases.

Understanding User Roles: Labuser and System

Who Is Labuser?

Labuser is typically a user account used for development, testing, or learning purposes. It often has limited privileges compared to administrative accounts and may only have access to specific schemas or objects.

Who Is System?

The System user in Oracle or similar database systems generally refers to a highly privileged account, often with DBA rights. Connecting as System allows complete access to all database objects and metadata, making it ideal for comprehensive searches across schemas.

Finding Database Objects with Names Starting with 'Db_'

Why Search for Objects Starting with 'Db_'?

Naming conventions like 'Db_' often indicate objects related to database administration, such as tables, views, procedures, or packages. Identifying these objects helps in:

- Auditing administrative objects
- Managing security and permissions
- Cleaning up deprecated or unnecessary objects
- Ensuring compliance with naming standards

Methods Overview

Depending on your user role—Labuser or System—the approach to find objects varies. The key is to

utilize system views that store metadata about database objects.

Connecting as Labuser: Finding Objects Starting With 'Dba_'

Limitations and Considerations

Since Labuser may have restricted privileges, the scope of objects you can see is limited to schemas and objects you have access to. Ensure your user has SELECT privileges on the relevant system views.

Using Data Dictionary Views

Common views include:

- ALL_OBJECTS: Shows objects accessible to the current user
- USER_OBJECTS: Shows objects owned by the current user
- DBA_OBJECTS: Shows all objects in the database (requires DBA privilege)

Sample Query Using ALL_OBJECTS:

```
```sql
SELECT owner, object_name, object_type
FROM all_objects
WHERE object_name LIKE 'DBA_%' ESCAPE '\'
ORDER BY owner, object_name;
```
```

Notes:

- The `LIKE 'DBA\\_%' ESCAPE '\` clause searches for object names starting with 'Dbas_' (case-sensitive depending on the database).
- Adjust the query if your environment is case-insensitive.

Example Output:

```
OWNER	OBJECT_NAME	OBJECT_TYPE
HR	Dbas_Users	TABLE
ADMIN	Dbas_Policies	VIEW
```

Using USER_OBJECTS for Owned Objects

If your user owns some objects starting with 'Dbas_', you can run:

```
```sql
SELECT object_name, object_type
FROM user_objects
WHERE object_name LIKE 'Dbas_%' ESCAPE '\
ORDER BY object_name;
```

---
```

Connecting as System: Finding All Objects Starting With

'Dba_'

Full Access to Database Metadata

Connecting as System grants the ability to search across all schemas. This is ideal for comprehensive audits and management.

Using DBA_OBJECTS View

The DBA_OBJECTS view contains metadata about every object in the database.

```
```sql
SELECT owner, object_name, object_type
FROM dba_objects
WHERE object_name LIKE 'Dba_%' ESCAPE '\'
ORDER BY owner, object_name;
```
```

Additional Tips:

- For case-insensitive searches, consider using the `UPPER()` function:

```
```sql
SELECT owner, object_name, object_type
FROM dba_objects
WHERE UPPER(object_name) LIKE 'DBA_%' ESCAPE '\'
ORDER BY owner, object_name;
```
```

- To retrieve objects of a specific type, add `AND object\_type = 'TABLE'` or similar.

Finding Objects Across Multiple Schemas

If you want to find all objects starting with 'Dba_' across the entire database:

```
```sql
SELECT owner, object_name, object_type
FROM dba_objects
WHERE object_name LIKE 'Dba_%' ESCAPE '\'
ORDER BY owner, object_name;
```
```

This query provides a comprehensive view, which is valuable during audits or cleanup procedures.

Advanced Techniques for Object Searches

Using Regular Expressions

Some databases support REGEXP_LIKE for more flexible pattern matching:

```
```sql
SELECT owner, object_name, object_type
FROM dba_objects
WHERE REGEXP_LIKE(object_name, '^Dba_')
ORDER BY owner, object_name;
```
```

Automating Searches with Scripts

Creating scripts or stored procedures can automate frequent searches, especially in environments with many objects.

Example Script:

```
```sql
DECLARE
v_object_name VARCHAR2(100);
BEGIN
FOR rec IN (
SELECT owner, object_name, object_type
FROM dba_objects
WHERE object_name LIKE 'Dba_%' ESCAPE '\'
) LOOP
-- Process each object as needed
DBMS_OUTPUT.PUT_LINE('Owner: ' || rec.owner || ', Name: ' || rec.object_name || ', Type: ' ||
rec.object_type);
END LOOP;
END;
```

---
```

Best Practices and Tips

- Verify Privileges: Ensure your user has the necessary privileges to access system views like ALL_OBJECTS, USER_OBJECTS, or DBA_OBJECTS.
- Case Sensitivity: Be aware of how your database handles case sensitivity in object names.

- Use Proper Filtering: Always include specific filters to narrow down results and improve query performance.
- Regular Audits: Schedule regular searches to monitor the creation of new objects matching your naming conventions.
- Documentation: Maintain documentation of naming standards and regularly review object lists for compliance.

Summary

Finding all objects with names starting with 'Db*_ ' when connected as Labuser or System involves understanding user privileges and utilizing the right system views. When connected as Labuser, your search is limited to accessible schemas and objects, typically via ALL_OBJECTS or USER_OBJECTS. Connecting as System provides a global view across the entire database through DBA_OBJECTS.

Using SQL queries with the LIKE operator, combined with case-insensitive functions or regular expressions, allows precise and flexible searches. Automating these queries with scripts enhances efficiency, especially in large environments.

By mastering these techniques, database administrators and developers can effectively manage and audit database objects, ensuring adherence to naming conventions and maintaining the database's integrity and security.

Further Resources:

- Oracle Database Documentation on Data Dictionary Views
- SQL Pattern Matching Techniques

- Best Practices for Database Object Naming Standards
- Automating Database Audits and Reports

Frequently Asked Questions

What is the purpose of connecting as 'Labuser' to find objects starting with 'Dba_'?

Connecting as 'Labuser' allows you to query and identify all database objects whose names begin with 'Dba_', helping in tasks like auditing, cleanup, or management of specific schemas.

How do you find all objects starting with 'Dba_' when connected as 'Labuser'?

You can execute queries against the data dictionary views such as ALL_OBJECTS or USER_OBJECTS, using a WHERE clause like WHERE OBJECT_NAME LIKE 'Dba_%'.

What is the difference between connecting as 'Labuser' and 'System' for object discovery?

Connecting as 'Labuser' limits the scope to objects accessible to that user, whereas connecting as 'System' grants access to all objects in the database, enabling a comprehensive search across schemas.

How can I connect as 'System' to find all objects starting with 'Dba_'?

Use a privileged account with SYSDBA rights to connect as 'System' and run queries on ALL_OBJECTS or DBA_OBJECTS views, e.g., SELECT FROM DBA_OBJECTS WHERE OBJECT_NAME LIKE 'Dba_%'.

What are the security considerations when connecting as 'System' to find objects?

Connecting as 'System' grants high-level privileges, so it should be done carefully to prevent accidental modifications or security breaches; always follow best practices and audit such actions.

Can I automate the process of finding all objects starting with 'Dba_' for both users and system?

Yes, you can create scripts or scheduled jobs that connect to the database and query relevant data dictionary views to list objects matching the 'Dba_' prefix for both user and system levels.

What tools or SQL commands are recommended for locating objects starting with 'Dba_'?

SQL queries on views like DBA_OBJECTS, ALL_OBJECTS, or USER_OBJECTS using the LIKE operator are effective. For example, `SELECT OBJECT_NAME, OWNER FROM DBA_OBJECTS WHERE OBJECT_NAME LIKE 'Dba_%'`.

Is there a difference in the results when connecting as 'Labuser' versus 'System' for these object searches?

Yes, connecting as 'Labuser' may limit results to objects accessible or owned by that user, while 'System' can provide a comprehensive list across all schemas, assuming appropriate privileges.

Additional Resources

When You Connected As Labuser Find All Objects Which Names Starts With Dba_. Connect As System Find All

In the realm of database administration and management, the ability to efficiently locate and manipulate database objects based on naming conventions is a critical skill. Particularly in environments where numerous objects—such as tables, views, procedures, or indexes—are created with systematic prefixes or naming patterns, the capability to filter and identify these objects becomes a cornerstone of effective database maintenance, security audits, and development tasks. This article provides a comprehensive analysis of the procedures, considerations, and best practices involved when connecting as a specific user—such as Labuser—to find all objects whose names start with Db*_ and subsequently connecting as a higher-privileged user—such as System—to find all objects across the database.

Understanding User Roles and Privileges in Database Environments

1. The Importance of User Contexts: Labuser vs. System

In most relational database management systems (RDBMS), users are assigned roles and privileges that define their scope of access. Two common user contexts are:

- Labuser: Typically a user with limited privileges, often used for development, testing, or sandbox environments. The user may have permissions to query certain objects, but not to access or modify the entire database schema.
- System (or SYS/SYSDBA in Oracle): A highly privileged account, often used for administrative tasks, managing database schemas, and performing critical maintenance activities. Under this context, users can access all objects within the database.

Connecting as these users enables different scopes of object discovery. As Labuser, the focus is on objects within the user's schema or those explicitly accessible, while as System, the scope widens to include all objects in the database.

2. Privilege Requirements for Object Discovery

To find objects based on naming patterns, specific privileges are necessary:

- SELECT privilege on data dictionary views (e.g., ALL_OBJECTS, DBA_OBJECTS, USER_OBJECTS).
- Access to dictionary views: Depending on user privileges, access may be restricted to certain views (e.g., USER_OBJECTS shows objects owned by the current user; ALL_OBJECTS shows objects accessible to the user; DBA_OBJECTS shows all objects in the database).
- Role grants: For comprehensive searches as System, users typically need DBA privileges or SELECT_CATALOG_ROLE.

Understanding these privilege requirements is critical before attempting object queries, especially when transitioning from a limited user to a privileged user.

Connecting as Labuser: Finding All Objects Starting With Db*_

1. The Objective and Use Cases

The primary goal when connected as Labuser is to identify all objects within the user's accessible schemas whose names start with Db*_*. This task might be part of:

- Auditing for naming conventions and security compliance.
- Identifying objects related to specific modules or functionalities.
- Preparing for migration or cleanup activities.

2. Step-by-Step Procedure

Step 1: Connect to the Database as Labuser

Use your preferred client (SQLPlus, SQL Developer, etc.):

```
```sql
CONNECT labuser/password@database
```
```

Step 2: Query Data Dictionary Views

Since Labuser may have access to USER_OBJECTS and ALL_OBJECTS, the query should utilize these views accordingly.

- To find objects owned by Labuser starting with Db*_*:

```
```sql
SELECT OBJECT_NAME, OBJECT_TYPE, OWNER
FROM USER_OBJECTS
WHERE OBJECT_NAME LIKE 'Db*_*' ESCAPE '\';
```

---

- To include objects that Labuser has access to (via privileges):

```
```sql
SELECT OWNER, OBJECT_NAME, OBJECT_TYPE
FROM ALL_OBJECTS
WHERE OBJECT_NAME LIKE 'Db\_%' ESCAPE '\';
```
```

### Step 3: Understand the Output

The result will list all matching objects' names, types, and owners. This data helps in understanding what objects are present, their purpose, and ownership.

Note: The pattern matching is case-sensitive in some systems; ensure the case matches the naming conventions used.

## 3. Limitations and Considerations

- Access restrictions: If Labuser does not have privileges to see certain schemas or objects, the results will be incomplete.
- Naming conventions: Confirm the exact case and pattern, especially if the database is case-sensitive.
- Object types: Be aware that Db\\_ could prefix different object types, such as tables, views, procedures, or indexes.

---

# Connecting as System: Finding All Objects Starting With Db\*\_

## 1. The Broader Scope and Objectives

Connecting as System (or a similarly privileged account) enables a comprehensive search across the database. The goal is to identify all objects owned by any user whose names start with Db\*\_ , regardless of schema ownership.

This task is vital for:

- Full inventory of database objects.
- Security auditing and compliance checks.
- Identifying unauthorized or legacy objects.
- Preparing for cleanup, refactoring, or migration.

## 2. The Procedure for Global Object Discovery

Step 1: Connect as System

```
```sql
CONNECT sys/password@database AS SYSDBA
```
```

Step 2: Query the DBA\_OBJECTS View

The DBA\_OBJECTS view contains information about all objects in the database.

```
```sql
SELECT OWNER, OBJECT_NAME, OBJECT_TYPE
FROM DBA_OBJECTS
WHERE OBJECT_NAME LIKE 'Dba\_%' ESCAPE '\';
```
```

### Step 3: Analyze the Results

The output will include:

- OWNER: who owns the object.
- OBJECT\_NAME: the object's name, starting with Dba\_.
- OBJECT\_TYPE: such as TABLE, VIEW, PROCEDURE, PACKAGE, etc.

This comprehensive list provides a complete view of all relevant objects across schemas.

## 3. Additional Filtering and Analysis

- To focus on specific object types:

```
```sql
WHERE OBJECT_NAME LIKE 'Dba\_%' ESCAPE '\'
AND OBJECT_TYPE IN ('TABLE', 'VIEW', 'PROCEDURE');
```
```

- To identify objects across a particular schema:

```
```sql
```

```
WHERE OWNER = 'SCHEMA_NAME' AND OBJECT_NAME LIKE 'Dba\_%' ESCAPE '\';
```

```
```
```

- To generate reports or export data, use tools such as SQL Developer, SQLPlus, or scripting languages.

## 4. Privilege Considerations and Best Practices

- Ensure the user has SELECT privileges on DBA\_OBJECTS. Usually, SYS or DBA roles are required.
- Be cautious with sensitive data; avoid unnecessary exposure.
- Use scripts to automate periodic scans for ongoing monitoring.

```

```

## Analytical Insights and Best Practices

### 1. The Significance of Naming Conventions

Consistent naming conventions like Dba\_ prefixes facilitate object management, security, and clarity in complex environments. Recognizing and locating such objects quickly aids in:

- Security audits: Identifying objects that may have elevated privileges.
- Maintenance: Pinpointing specific modules or functionalities.

- Compliance: Ensuring naming standards are followed.

## **2. Privilege Management and Security Implications**

- Over-privileged accounts like System should be used judiciously. Excessive privileges can lead to unintended modifications.
- Regular audits of objects with specific prefixes help detect anomalous or unauthorized creations.
- Use of roles and minimal privileges enhances security posture.

## **3. Automation and Tooling**

- Automate object discovery with scripts that run at regular intervals.
- Utilize database management tools for visualization and reporting.
- Implement alerts for new objects matching prefix patterns to detect unauthorized changes.

## **4. Impacts on Database Performance and Maintenance**

- Large schemas with numerous objects require optimized queries.
- Use appropriate indexing on dictionary views if querying large datasets.
- Be cautious with scripts that scan entire database schemas, as they can impact performance.

## Conclusion

The process of finding objects with specific naming patterns like Db\*\_ across a database hinges on understanding user privileges, schema organization, and effective querying of data dictionary views. Connecting as Labuser allows targeted searches within limited scopes, suitable for development and testing environments. Conversely, connecting as System provides the comprehensive visibility necessary for administrative oversight, security audits, and maintenance.

By mastering these techniques, database administrators and developers can streamline their workflows, enforce naming conventions, enhance security, and maintain high standards of database integrity. Whether operating within a confined user context or at the highest privilege levels, understanding how to locate and analyze objects based on naming patterns remains an essential skill in modern database management.

## **When You Connected As Labuser Find All Objects Which Names Starts With Db\*\_ Connect As System Find All**

When You Connected As Labuser Find All Objects Which Names Starts With Db\*\_ Connect As System Find All

## Related Articles

- [A Few Risks That Do NOT Arise When We Connect To The Internet AreO Stolen PII Or Credit Card DataO SpamO](#)
- [A Monochromatic Light Source Illuminates A Double Slit And Theresulting Interference Pattern Is Observed](#)
- [Which Accurately Describes The Supremacy Clause?States Can Violate Federal Law With A Judges Consent.No](#)

[Back to Home](#)