# How to Create New Schema in Oracle 19c?

Here is the complete set of commands to create a schema named busycoder with password bca in Oracle 19c, including all necessary privileges to connect via JDBC:

## 🔐 Login as SYSDBA

sqlplus / as sysdba

### ➤ Explanation:

* sqlplus → Launches the SQL\*Plus utility (Oracle's command-line client).
* / → Tells SQL\*Plus to use the current OS user for authentication.
* as sysdba → Grants SYSDBA privileges, required for administrative tasks like creating users, starting/stopping the database, etc.

### ✅ This login method works only when:

* You're on the same machine where Oracle is installed.
* Your OS user (e.g., Windows user) is part of the ORA\_DBA group (on Windows) or dba group (on Linux).

## 🧱 Step-by-Step SQL Commands (Run as SYS or SYSTEM):

### 1️⃣ Create the user (which also creates the schema)

CREATE USER busycoder IDENTIFIED BY bca;

### 2️⃣ Grant basic privileges to allow login and object creation

GRANT CONNECT, RESOURCE TO busycoder;

### 3️⃣ (Optional but recommended) Allow unlimited tablespace usage

Required if the user is creating large objects or many tables:

ALTER USER busycoder QUOTA UNLIMITED ON USERS;

### 4️⃣ (Optional but JDBC-friendly) Unlock the user if required

Note: Newly created users are usually unlocked by default.

ALTER USER busycoder ACCOUNT UNLOCK;

## 📝 Additional Notes:

* Ensure the USERS tablespace exists and is the **default tablespace** for new users.

### You can also specify default and temporary tablespaces:

CREATE USER busycoder IDENTIFIED BY bca

DEFAULT TABLESPACE users

TEMPORARY TABLESPACE temp

QUOTA UNLIMITED ON users;

## 💻 JDBC Connection (Using Thin Driver)

Use the following format to connect from Java:

### 🔹 SID-based URL

java

CopyEdit

String url = "jdbc:oracle:thin:@localhost:1521:orcl"; // orcl = SID

String user = "busycoder";

String password = "bca";

Connection conn = DriverManager.getConnection(url, user, password);

### 🔹 Service Name-based URL (Recommended for 19c PDBs like XEPDB1 or ORCLPDB1)

java

CopyEdit

String url = "jdbc:oracle:thin:@//localhost:1521/XEPDB1"; // for pluggable DB

# ✅ Lesson Objectives: User Privileges & Roles in Oracle 19c

After completing this lesson, students will be able to:

* ✅ Differentiate **system privileges** from **object privileges**
* ✅ Grant **privileges on tables**
* ✅ Grant and manage **roles**
* ✅ Distinguish between **privileges and roles**

## 🔹 1. Privileges in Oracle

### ✅ What is a Privilege?

A **privilege** is the right to perform a specific action on the database.

### 🔸 A. System Privileges

#### 📘 Definition:

System privileges are permissions to perform actions on the **database as a whole**, such as creating users, tables, or managing system settings.

#### ✅ Examples:

* CREATE TABLE
* CREATE USER
* DROP ANY TABLE
* ALTER SYSTEM

### 🔸 Granting System Privileges:

GRANT CREATE TABLE TO busycoder;

GRANT CREATE SESSION TO trainee\_user;

### 🔸 Revoking System Privileges:

REVOKE CREATE TABLE FROM busycoder;

### 🔸 B. Object Privileges

#### 📘 Definition:

Object privileges allow specific actions on **individual database objects** like tables, views, or procedures.

#### ✅ Examples:

* SELECT
* INSERT
* UPDATE
* DELETE
* EXECUTE (for stored procedures)

### 🔸 Granting Object Privileges:

-- Grant SELECT and INSERT on emp table to trainee\_user

GRANT SELECT, INSERT ON emp TO trainee\_user;

### 🔸 With GRANT OPTION (Allow user to further grant privileges to others):

GRANT SELECT ON emp TO trainee\_user WITH GRANT OPTION;

### 🔸 Revoking Object Privileges:

REVOKE INSERT ON emp FROM trainee\_user;

## 🔹 2. Roles in Oracle

### 📘 What is a Role?

A **role** is a named collection of privileges that can be granted to users or to other roles. It simplifies the process of managing multiple permissions.

### ✅ Benefits of Using Roles:

* Easier to manage privileges for multiple users.
* Centralized, reusable permission sets.

### 🔸 Creating a Role:

CREATE ROLE trainer\_role;

### 🔸 Granting Privileges to a Role:

GRANT CREATE TABLE, CREATE VIEW TO trainer\_role;

### 🔸 Granting Role to a User (busycoder):

GRANT trainer\_role TO busycoder;

### 🔸 Revoking Role from a User:

REVOKE trainer\_role FROM busycoder;

## 🔹 3. Summary: Privileges vs Roles

| Feature | Privileges | Roles |
| --- | --- | --- |
| Type | Individual rights | Group of privileges |
| Scope | Specific system or object | Abstracted collection |
| Management | Harder to manage many users | Easier (assign role once) |
| Example | SELECT on emp | HR\_ROLE (includes SELECT, UPDATE) |

## 🔹 4. Practice Examples

### ➤ Scenario:

You are the DBA. You want to:

1. Allow user **ekta** to SELECT and UPDATE data from the emp table.
2. Create a role **data\_entry** that allows INSERT, UPDATE on the emp table.
3. Assign the **data\_entry** role to user **ajay**.

### ✅ Queries:

-- Step 1: Grant object privilege directly to ekta

GRANT SELECT, UPDATE ON emp TO ekta;

-- Step 2: Create the role

CREATE ROLE data\_entry;

-- Step 3: Grant privileges to the role

GRANT INSERT, UPDATE ON emp TO data\_entry;

-- Step 4: Assign the role to user ajay

GRANT data\_entry TO ajay;