

# dbarc Lab 3:

## Abfragen auf Data Dictionary, Erstellen von Demo-Schema

Data Dictionary Views stellen Informationen über die Strukturen (Tabellen, Views, Users, Tablespaces, Segmente, Datenfiles, etc.) zur Verfügung. Sie können über SQL-Befehle abgefragt werden. In den folgenden Übungen werden verschiedene Abfragen auf den Data Dictionary der Lab-Datenbank ausgeführt.

In der letzten Übung wird pro Gruppe ein Demo-Schema erstellt, das für die weiteren Labs (Security, Performance Tuning) benötigt wird.

### 1. Abfragen auf Data Dictionary

Data Dictionary Views stellen Informationen über die Strukturen (Tabellen, Views, Users, Tablespaces, Segmente, Datenfiles, etc.) zur Verfügung. Sie können über SQL-Befehle abgefragt werden. In dieser Übung werden verschiedene Abfragen auf den Data Dictionary der Lab-Datenbank ausgeführt.

- Melde Dich mit Deinem persönlichen User auf der Lab-Datenbank an.
- Mit folgender Abfrage kannst Du die USER-Views auf den Data Dictionary ermitteln:

```
SELECT * FROM dictionary
WHERE table_name LIKE 'USER%'
ORDER BY table_name;
```

- Schau Dir mit SQL Developer (oder dem Tool Deiner Wahl) die Struktur der EMP-Tabelle an (Columns, Constraints, Indexes) an:

| Columns     | Data               | Model    | Constraints  | Grants    | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL |
|-------------|--------------------|----------|--------------|-----------|------------|----------|-----------|--------------|---------|------------|---------|-----|
| Actions...  |                    |          |              |           |            |          |           |              |         |            |         |     |
| COLUMN_NAME | DATA_TYPE          | NULLABLE | DATA_DEFAULT | COLUMN_ID | COMMENTS   |          |           |              |         |            |         |     |
| 1 EMPNO     | NUMBER (4, 0)      | No       |              | 1         |            |          |           |              |         |            |         |     |
| 2 ENAME     | VARCHAR2 (10 BYTE) | Yes      |              | 2         |            |          |           |              |         |            |         |     |
| 3 JOB       | VARCHAR2 (9 BYTE)  | Yes      |              | 3         |            |          |           |              |         |            |         |     |
| 4 MGR       | NUMBER (4, 0)      | Yes      |              | 4         |            |          |           |              |         |            |         |     |
| 5 HIREDATE  | DATE               | Yes      |              | 5         |            |          |           |              |         |            |         |     |
| 6 SAL       | NUMBER (7, 2)      | Yes      |              | 6         |            |          |           |              |         |            |         |     |
| 7 COMM      | NUMBER (7, 2)      | Yes      |              | 7         |            |          |           |              |         |            |         |     |
| 8 DEPTNO    | NUMBER (2, 0)      | No       |              | 8         |            |          |           |              |         |            |         |     |

- Versuche die gleichen Informationen durch SQL-Abfragen auf den Data Dictionary zu ermitteln. Verwende dazu die USER-Views des Data Dictionary. Welche Views liefern die notwendigen Daten?  
Die vollständige Dokumentation aller Data Dictionary Views sind in der Dokumentation «Oracle Reference, Part II Static Data Dictionary Views» zu finden:  
<https://docs.oracle.com/en/database/oracle/oracle-database/21/refrn/static-data-dictionary-views.html>
- Kopiere Deine SQL-Abfragen und ersetze die USER-Views durch die entsprechenden ALL-Views des Data Dictionary. Was stellst Du fest, wenn Du die Abfragen ausführst?
- Kopiere Deine SQL-Abfragen und ersetze die USER-Views durch die entsprechenden DBA-Views des Data Dictionary. Was stellst Du fest, wenn Du die Abfragen ausführst?

---

## 2. Abfragen auf Data Dictionary (DBA-Views)

---

Mit dem User PDBADMIN hast Du Zugriff auf die Data Dictionary Views auf Stufe DBA (DBA\_TABLES, DBA\_USERS, etc.). Damit kannst Du auf alle Strukturen der Datenbank zugreifen.

- Melde Dich mit dem User PDBADMIN auf der Lab-Datenbank an.
- Beantworte folgende Fragen durch SQL-Abfragen auf DBA-Views des Data Dictionary:
  - Welches sind die vier grössten Tabellen im Tablespace USERS. Wieviel Platz (in Kilo-byte) belegen sie?
  - Welches sind die vier grössten Segmente im Tablespace USERS. Wieviel Platz (in Kilo-byte) belegen sie?
  - Welcher Datenbank-User besitzt die meisten Tabellen?
  - Was für Tablespaces sind vorhanden, und wie gross (in Megabytes) sind sie?
- Gibt es in SQL Developer (bzw. anderen Tools) eine einfachere Möglichkeit, diese Informationen anzuzeigen?

---

## 3. Deployment von Demo-Schema (Gruppenarbeit)

---

Ein Schema ist eine logische Gruppierung von zusammengehörenden Datenbankobjekten (Tabellen, Views, Indexes, etc.). In Oracle ist ein Schema ein Datenbank-User mit eigenen Objekten. Üblicherweise wird pro Applikation (oder bei komplexeren Systemen pro Applikationsschicht) ein Schema erstellt.

Für die weiteren Labs verwenden wir pro Gruppe ein Demo-Schema. Ziel dieser Übung ist es, dieses Schema bereitzustellen, damit es für weitere Übungen zur Verfügung steht.

### Gruppenzuteilung

| Team                     | Schema | Tablespace |
|--------------------------|--------|------------|
| Team 1: Shana + Ramanan  | DBARC1 | DBARC1_TS  |
| Team 2: Arman + Jonathan | DBARC2 | DBARC2_TS  |
| Team 3: Bianca + Danijel | DBARC3 | DBARC3_TS  |
| Team 4: Marc + Tobias    | DBARC4 | DBARC4_TS  |

### Erstellen von Tablespace und Schema-User

Die folgenden Schritte müssen mit dem User PDBADMIN ausgeführt werden:

- Erstellt für Euer Team einen eigenen Tablespace, in dem alle Tabellen und Indizes des Demo-Schemas erstellt werden. Dazu könnt Ihr die einfachste Variante (Oracle Managed Files) verwenden, sodass weder Filename noch Grösse angegeben werden muss.
- Überprüft mit einer Abfrage auf DBA\_DATA\_FILES, wo das zugehörige Database File angelegt wurde und wie gross es ist.
- Nun legt Ihr einen Schema-User für Euer Team an. Damit alle Datenbankobjekte im richtigen Tablespace gespeichert werden, kann beim Erstellen des Users ein Default-Tablespace angegeben werden. Damit der User den Tablespace verwenden kann, müssen ausserdem Quotas freigegeben werden (QUOTA UNLIMITED). Dies kann alles direkt beim Erstellen des Users angegeben werden. Syntax siehe **SQL Language Reference: [CREATE USER](#)**.

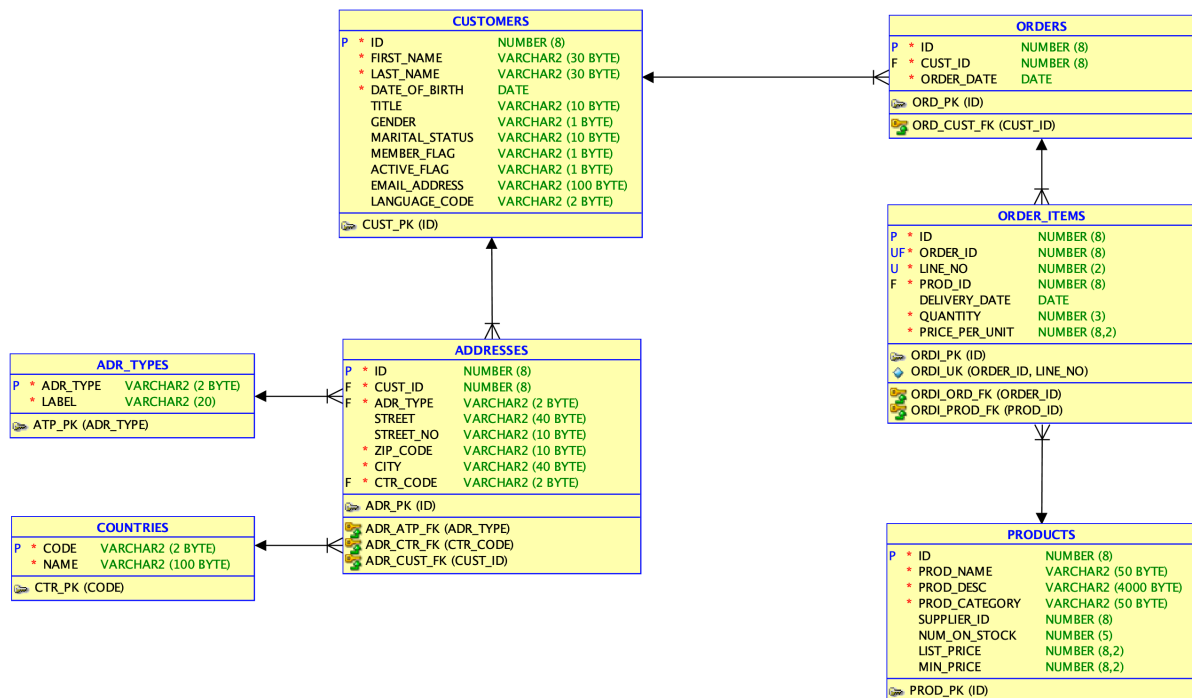
- Um mit dem User einloggen und Tabellen anlegen zu können, sind verschiedene Systemprivilegien notwendig (CREATE SESSION, CREATE TABLE, etc.). Die wichtigsten Privilegien sind in einer Rolle DBARC\_SCHEMA\_ROLE zusammengefasst, die Ihr mit folgendem Befehl Eurem Schema-User zuweisen könnt:

```
GRANT dbarc_schema_role TO <Schema-User>;
```

- Richtet in SQL Developer eine zusätzliche Connection ein, um mit dem neu erstellen User einloggen zu können. Details dazu sind in Lab 1 beschrieben.

## Erstellen und Laden des Demo-Schemas

Unser Demo-Schema besteht aus 7 Tabellen:



Um diese Tabellen zu erstellen und mit Daten zu laden, loggt Ihr mit dem neu erstellen Schema-User ein und führt danach folgende Schritte aus:

- Die Tabellen können mit dem Script **dbarc\_create\_tables.sql** erstellt werden. Nach Ausführung des Scripts sind die Tabellendefinitionen im Data Dictionary vorhanden, aber es wurden noch keine Segmente erstellt. Dies kann durch Abfragen auf USER\_TABLES und USER\_SEGMENTS überprüft werden.
- Mit dem Script **dbarc\_load\_data.sql** werden Demodaten in die Tabellen geladen. Diese werden aus dem Schema DBARC\_BASE kopiert, damit alle Teams die gleichen Daten zur Verfügung haben.
- Überprüft nach Abschluss des Ladescripts mit dem User PDBADMIN, wie gross das Database File Eures Tablespace nun ist.
- Mit dem Script **dbarc\_finish\_setup.sql** werden verschiedene Abschlussarbeiten im Schema durchgeführt. Unter anderem werden Primary und Foreign Key Constraints angelegt, um die Beziehungen zwischen den Tabellen zu definieren.
- Überprüft nochmals mit dem User PDBADMIN die Grösse des Database Files. Warum ist es nun grösser als nach dem Laden der Tabellen?
- Nach Abschluss aller Deploymentschritte könnt Ihr mittels SELECT COUNT(\*) auf den einzelnen Tabellen überprüfen, wieviele Rows in jeder Tabelle vorhanden sind.

---

#### 4. Abfragen auf Demo-Schema (Gruppenarbeit, optional)

---

Da wir das Demo-Schema für die weiteren Labs verwenden werden, ist es sinnvoll, dass Ihr Euch einen ersten Überblick über die vorhandenen Daten verschafft. Gleichzeitig ist dies auch eine gute Übung, um Eure SQL-Kenntnisse aufzufrischen. Versucht mittels SQL-Abfragen in Eurem Schema folgende Fragen zu beantworten:

1. Wie viele Länder sind in der Tabelle COUNTRIES vorhanden?
2. Wie viele unterschiedliche Länder kommen in der Tabelle ADDRESSES vor? Welches sind die drei häufigsten verwendeten Länder?
3. Wo wohnt Harry Potter?
4. Was ist die Adresse von James Bond?
5. Warum hat Miss Money Penny zwei verschiedene Adressen?
6. Welcher Kunde oder welche Kundin hat am meisten Bestellungen (ORDERS) ausgeführt?
7. An welchem Datum sind die letzten Bestellungen eingetroffen, und wie viele waren es?
8. Wie oft wurde das Produkt „Y Box“ im Dezember 2022 bestellt?