

Arbeiten mit APEX: Konzepte und Begriffe

1. Allgemeines

APEX ist ein Framework für die Entwicklung einfacher datenbankzentrierter Anwendungen im Web auf der Basis von Oracle. Die Entwicklung der Anwendungskomponenten erfolgt deklarativ, d.h. durch Definition von Eigenschaften mittels Editor. Falls erforderlich, können SQL-Funktionen oder Prozeduren benutzt werden, um komplexere Anwendungsfälle abzudecken.

Jeder Benutzer besitzt einen eigenen Workspace (bei uns: Workspace = Benutzername).

APEX besteht aus drei **Komponenten**:

| | |
|----------------------------|---|
| Application Builder | Interaktive Entwicklung von DB-basierten Webanwendungen |
| SQL Workshop | Ausführung von SQL-Befehlen und SQL-Skripten (vergleichbar mit SQLPLUS) |
| Data Workshop | Laden von Daten in die DB, Extrahieren von Daten aus der DB |

Grundbegriffe

| | | |
|--------------------|-----------|---|
| Application | Anwendung | Eine Anwendung ist eine Menge (Collection) von Seiten (Pages), die über Registerkarten, Schaltflächen oder Hypertext-Links miteinander verknüpft sind. Die Seiten einer Anwendung verfügen über eine gemeinsame Definition für den Session-Status und eine gemeinsame Authentifizierungsmethode. |
| Page | Seite | Eine Seite wird definiert danach, wie sie dargestellt wird (Seitenwiedergabe) und wie sie verarbeitet wird (Seitenverarbeitung). Verarbeitung umfasst Events und Logik, die an diese Events geknüpft wird. Damit lassen sich z.B. Berechnungen, Validierungen oder zeitliche Abfolgen definieren. Die Seitenwiedergabe erfolgt dynamisch auf der Grundlage von Metadaten, die vom Entwickler verändert werden können. Das Aussehen einer Seite wird durch Page Templates bestimmt. |
| Region | Region | Eine Seite ist in Regionen aufgeteilt, die von unterschiedlichem Typ sein können und jeweils bestimmte Aufgaben haben. Typen von Regionen können sein: HTML Text, SQL Queries, Charts. Jede Region wird mit einem eigenen Region Template dargestellt. Die Position, wo eine Region auf der Seite dargestellt, wird durch das Page Template bestimmt. |
| Item | Element | Application Items werden genutzt, um dynamisch HTML zu generieren (mithilfe von PL/SQL). Es gibt ca. 50 unterschiedliche Typen von Items: Pop-up-Listen (List of Values), Texteingabefelder (Text Areas), sogenannte Date Pickers (Auswahl eines Datums aus einem Kalenderblatt) usw. Der Wert eines Items wird automatisch im Session State einer Anwendung gespeichert, der von überall her in der Benutzer-Session referenziert werden kann. |

Architektur

Eine Anwendung wird in Echtzeit aus den Daten (Benutzerdaten) bzw. Metadaten (beschreibenden Daten) generiert und in HTML dargestellt. Beim Erzeugen oder Erweitern einer Anwendung, wird keinerlei Code generiert; stattdessen werden die modifizierten **Metadaten** in der Datenbank gespeichert. Bei der Darstellung der Anwendung werden die Metadaten für die Anzeige der Seite ausgelesen

Beim Start einer Anwendung durch einen Benutzer wird eine Anwendungssitzung (**Application Session**) eröffnet. Zu einer Application Session gehören alle Items der Anwendung, deren Werte im Zustand (Session State) gespeichert werden. Jeder Seitenaufruf führt zum Öffnen einer neuen Datenbanksitzung (Database Session). Die Application Session wird durch einen eindeutigen Session Identifier identifiziert und steht nicht in Beziehung zu der Vielzahl der DB Sessions, die bei Aufruf einer Page erzeugt werden.

Struktur einer URL

Jede Applikation hat eine Nummer (genannt Application ID) oder einen alphanumerischen Alias, der diese eindeutig identifiziert. In ähnlicher Weise besitzt jede Seite eine eindeutige Nummer (Page-Nummer) oder einen alphanumerischen Alias. Bei Ausführung einer Applikation, generiert die APEX-Engine eine Session-Nummer, die als Schlüssel für den Session State des Benutzers dient.

Die URL, die für jede Seite angezeigt wird, verweist auf den Ort der APEX-Engine und beinhaltet Application ID, Page-Nummer und Session-ID. Zum Beispiel:

```
http://ora10glv.imn.htwk-leipzig.de:7777/pls/htmldb/f?p=4000:1:6658881197064008030
```

Diese Beispiel zeigt folgendes an:

- Die Adresse von Oracle APEX ist:
`http://ora10glv.imn.htwk-leipzig.de:7777/pls/htmldb/`
- Die Application ID ist `4000`
- Die Page-Nummer ist `1`
- Die Session ID ist `6658881197064008030`

Die `f?p` Syntax wird genutzt zum Verlinken von Pages einer Anwendung (Details können der Online-Dokumentation entnommen werden).

2. Data Workshop

Im Data Workshop können Daten aus der verwalteten Oracle-Datenbank importiert und exportiert werden.

Datenimport. Daten können aus Textdateien, XML-Dateien und Daten eines Arbeitsblattes (Spreadsheet-Datei) in die Datenbank importiert werden. Voraussetzung für den Import von XML-Daten ist, dass diese in der kanonischen Form vorliegen (d.h. flache Struktur `<ROWSET><ROW>...`). Es können jeweils Daten in eine Tabelle importiert werden. Beim Import von CSV-Dateien besteht die Möglichkeit, ein zu erkennendes Trennzeichen (Delimiter) zwischen den Werten zu definieren.

Datenexport. Daten aus der Datenbank können in Textdateien (.csv) oder XML-Dateien exportiert werden.

3. SQL Workshop

In dieser APEX-Komponente befindet sich der eigentliche SQL Workshop (→ 3.1), ein Browser für die Datenbank (→ 3.2) sowie ein Skriptprozessor (→ 3.3).

3.1 SQL Workshop

Der **SQL Workshop** beinhaltet einen SQL Command Processor (→3.1.1) - vergleichbar mit SQLPLUS - die Definition von Benutzeroberflächen-Standards (→ 3.1.2) sowie einen Assistenten zum Erstellen von Datenbankobjekten (→ 3.1.3).

3.1.1 Command Processor

Bei Aufruf des Command Processors erscheint eine Eingabefläche für die Eingabe von SQL und PL/SQL Befehlen. Es besteht aber auch die Möglichkeit, sich die Struktur von Tabellen anzuschauen, z.B. durch das auch SQLPLUS heraus bekannte Kommando DESC.

3.1.2 Benutzeroberflächen-Standards

Mit Hilfe von **Benutzeroberflächen-Standards** (User Interface, Benutzeroberfläche) können Entwickler die Eigenschaften von Standard-Benutzeroberflächen einer Tabelle, Spalte oder View innerhalb eines angegebenen Schemas zuordnen. Wenn ein Entwickler eine Form oder einen Bericht mit einem Assistenten erstellt, verwendet der Assistent diese Informationen, um Standardwerte für Regions- und Elementeeigenschaften zu erstellen. Auf dieser Seite wird eine Liste der Tabellen angezeigt, die Benutzeroberflächen-Standards verwenden. Um die Standards zu bearbeiten, klickt man neben dem Tabellennamen auf das Symbol zum Bearbeiten.

3.1.3 Assistenten zum Erstellen von Datenbankobjekten

Mit der Funktion **Objekt erstellen** können Datenbankobjekte unterschiedlicher Typen (→ 3.2) erstellt werden. So gibt es z.B. einen Wizard für das Erzeugen einer Tabelle. Dieser Assistent kann auch über den Browser erreicht werden.

3.2 Database Browser

Im **Database Browser** kann sich ein Benutzer die in seinem Schema vorhandenen Datenbankobjekte anzeigen, deren Definition und ggf. deren Inhalt (z.B. im Fall von Tabellen) betrachten.

Folgende Typen von **Datenbankobjekten** sind vorhanden und können mittels Browser zugegriffen werden:

- *Tabellen*: Alle vom Benutzer definierten Tabellen (auch wenn sie nicht mittels APEX-Anwendung verarbeitet werden), die in seinem Schema vorhanden sind. Tabellen mit Metadaten gehören also nicht dazu.
- *Indizes*: Diese werden zumeist automatisch für Primärschlüssel zur Zugriffsbeschleunigung angelegt. Die Index-Namenskonvention ermöglicht es, die jeweils indizierten Spalten (und ihre Herkunftstabellen) zu erkennen.
- *Ansichten* (Views): Dabei kann man sich die SQL-Definition anschauen, somit die Basistabellen und verwendeten Spalten
- *Synonyme*: bezeichnen Datenbankobjekte (für unsere Zwecke nicht relevant).
- *Sequenzen*: Zählerobjekte für automatische Nummerierung von Primärschlüsselwerten.
- *Datenbank-Links*: Verbindungen zu anderen Datenbanken (für unsere Zwecke nicht relevant).
- *Packages*: „Behälter“ für PL/SQL Programmeinheiten (Prozeduren, Funktionen, Variablen).
- *Prozeduren*: in der Datenbank gespeicherte Prozeduren, die vom Benutzer definiert sind und in der Sprache PL/SQL implementiert werden.
- *Funktionen*: in der Datenbank gespeicherte Funktionen, die vom Benutzer definiert sind und in PL/SQL implementiert werden, geben im Unterschied zu Prozeduren einen Wert zurück und können somit in Ausdrücken verwendet werden.

- *Trigger*: in der Datenbank gespeicherte Prozeduren, die durch ein Datenbankereignis automatisch ausgelöst werden, in PL/SQL implementiert.
- *Java-Quellen*: auch Java-Code kann in der Datenbank gespeichert werden, weil der Oracle Server über eine Java Virtual Machine (JVM) verfügt – für unsere Zwecke nicht relevant.
- *Typen*: benutzerdefinierte Datentypen, also solche, die über die Oracle-Standardtypen hinausgehen.
- *XML-Schema*: XML-Schemadokumente lassen sich in der Datenbank ablegen, können genutzt werden zum Validieren von Objekten des Typs XML – für unsere Zwecke nicht relevant.

3.3 SQL Skript

Es können unter **SQL Skripts** Skriptdateien und Kontrolldateien verwaltet werden. Ausserdem lassen sich DDL-Anweisungen (Data Definition Language) aus vorhandenen Datenbank-Strukturen generieren. Skript-Kontrolldateien haben eine übergeordnete Aufgabe und steuern die Ausführung mehrerer Skripte.

SQL Skripts werden in einem Repository gehalten. Sie können dorthin hochgeladen und im Repository ausgeführt werden. Es ist aber auch möglich, ein Skript in APEX zu erstellen, abzuspeichern und auszuführen. Dabei ist zu beachten, dass es sich bei den Skripten vorzugsweise um Anweisungen zum Aufbau einer Datenbank handeln sollte, d.h. Ergebnisse von SELECTS werden in der Ausführungsstatistik nicht sichtbar gemacht.

Es besteht die Möglichkeit, invers aus dem Datenbankobjekt-Repository ein **DDL-File** zu generieren, so kann z.B. aus der Menge der Tabellen eine ausgewählt werden. Das System liest die Metadaten und erzeugt daraus eine vollständige CREATE TABLE-Anweisung.

4. Application Builder

Die aus der Datenbank heraus dynamisch generierten Seiten können im Application Builder verändert werden – durch das Setzen von Eigenschaften (von Applikationsobjekten), die das Erscheinungsbild der Seite beeinflussen.

4.1 Seitenwiedergabe (Page Rendering)

Regionen. Eine Region ist ein Bereich einer Seite, der ein spezifisches Template nutzt, um HTML-Content zu generieren. Jede Seite kann eine bestimmte Anzahl von Regionen haben. Mittels Regionen können andere Elemente, z.B. Buttons oder Items, gruppiert werden. Es können einfache Regionen erzeugt werden, die kein zusätzliches HTML generieren, oder komplexe Regionen, die Inhalt innerhalb von HTML-Tabellen einsetzen. Die APEX-Engine zeigt Regionen in einer Sequenz innerhalb von Spalten an. Eine Region kann bedingt angezeigt werden. Hierfür sind dann entsprechende Bedingungen (Conditions) zu definieren.

Buttons (Schaltflächen) werden verwendet, um eine Seite abzuschicken (submit). Wenn eine Seite an den Server gesendet wird, wird diese durch die APEX-Engine verarbeitet, oder es erfolgt die Umleitung zu einer anderen Seite ohne Verarbeitung. Ein Button kann implementiert werden als HTML-Button, als Bild oder durch Verwendung eines Templates.

Items (Elemente). Innerhalb einer Region werden Items zusammengefasst. Items sind Elemente von HTML-Formularen wie z.B. Texteingabefelder, Select-Boxen oder Check-Boxen. Ihr Inhalt ist mit einem Session State verbunden.

Computations (Berechnungen) werden durchgeführt bei der Darstellung der Seite. Eine Berechnung ist eine Logik-Einheit, die Items Werte aus einem Session State zuweist.

Processes (Prozesse) werden ausgeführt bei der Darstellung der Seite. Ein Prozess ist ein Steuerungselement zur Ausführung von DML-Anweisungen (Data Manipulation Language) oder

PL/SQL. Damit können zum Beispiel einem Session State zum Zeitpunkt der Seitendarstellung Werte zugewiesen werden.

4.2 Seitenverarbeitung (Page Processing)

Folgende Steuerungselemente werden angeboten:

Computations (Berechnungen) werden bei der Verarbeitung der Seite ausgeführt. Es sind Logik-Bausteine zur Zuweisung von Session States an Items.

Validations erlauben die Überprüfung der Benutzereingabe auf Gültigkeit. Ein Beispiel ist das Ausfüllen eines Pflichtfeldes.

Processes (Prozesse) werden nach Absenden der Seite ausgeführt. Es handelt sich um Steuerungselemente zur Ausführung von DML- oder PL/SQL-Befehlen.

Branching. Damit lassen sich logische Verzweigungen innerhalb einer Anwendung definieren. Hierbei gibt es unterschiedliche Verzweigungspunkte (branching points), z.B. On Submit / Before Validation.

4.3 Gemeinsame Komponenten (Shared Components)

Hierbei handelt es sich um gemeinsame Elemente, die auf jeder Seite innerhalb einer Anwendung verwendet werden können. Nachfolgend einige Beispiele:

Tabs (Registerkarten). Tabs erlauben es, effektiv zwischen den Seiten einer Anwendung hin und her zu navigieren. Die Anwendung kann auf die Weise in unterschiedliche Bereiche eingeteilt werden, z.B. je nach zu bearbeitenden Tabellen. Dabei werden zwei unterschiedliche Typen unterschieden: Standard-Tabs (Standard-Registerkarten) und Parent Tabs (übergeordneten Registerkarten). Eine Anwendung, die nur eine Ebene von Tabs besitzt, nutzt den Standardsatz. Ein Standard Tab ist mit einer speziellen Seite assoziiert. Ein Parent Tab funktioniert wie ein Container, der eine Menge von Standard Tabs aufnimmt. Zugleich stellt ein Parent Tab noch ein anderes Navigationsniveau dar.

List of Values (Werteliste). Eine List of Values (LOV) ist eine statische oder dynamische Definition von Werten, die bei der Anzeige bestimmter Seitenelemente, z.B. Checkboxes, Select-Listen, auf der Seite erscheinen. LOVs können *statisch* sein, d.h. sie basieren auf einer festen Menge vordefinierter Anzeige- und Return-Values. In der *dynamischen* Variante basieren sie auf SQL-Anfragen, die die Werte aus DB-Tabellen auslesen. LOVs können mittels Wizard definiert werden und unter einem Namen im Repository einer Anwendung abgelegt werden.

Menüs. Ein Menü ist eine hierarchische Liste von Links, die mithilfe von Templates angezeigt werden. Ein Menü kann durch eine Liste von Links dargestellt werden oder durch einen sogenannten Breadcrumb Path. Die Breadcrumb (engl.: Brotkrümel) oder auch Krümelnavigation zeigt in der Hierarchie den aktuellen Standort an und ermöglicht das vertikale Navigieren nach oben. So findet der Nutzer jederzeit auch aus sehr tiefen Ebenen wieder nach oben, bis hin zur Homepage.

Listen (Lists). Eine Liste ist eine Sammlung von Links, die mit Hilfe eines Templates dargestellt wird. Für jeden Listeneintrag ist ein Anzeigetext, eine Ziel-URL und andere Attribute zu definieren, die die Anzeige des Listeneintrages steuern. Die Darstellung der Liste und das Erscheinungsbild aller Listeneinträge lässt sich durch Verbindung mit einem Template steuern.

Themes und Templates. Templates steuern das "Look and Feel" der Seiten einer Anwendung. Templates können für Seiten, Regionen, Reports, Listen, Labels, Menüs, Buttons und Wertelisten (LOV) definiert werden. Gruppen von Templates sind in benannten Kollektionen zusammengefasst, die als Themes bezeichnet werden.

Navigation Bar. Eine Navigation Bar ermöglicht eine einfache Navigation zwischen Seiten einer Anwendung. Die Platzierung der Navigation Bar wird durch das zugehörige Seiten-Template bestimmt. Einträge von Navigation Bars sind häufig Links. Im Anwendungsbeispiel gehören Print und Logout dazu.

Sicherheit. Ein *Authentifizierungsschema* ist eine gespeicherte Konfiguration für eine Anwendung. Es kann ein vorhandenes Schema bearbeitet oder ein neues definiert werden. Der Gebrauch für eine Anwendung erfolgt durch Festlegung als „aktuelles Element“. Die Authentifizierung geschieht mittels Benutzername und Passwort, durch die der Zugang zur Anwendung ermöglicht wird, wobei die benötigten Daten in einer Tabelle der Anwendung verwaltet werden (da es sich nicht um ORACLE-User handelt).

Die Berechtigung für Anwendungen, Seiten und die meisten anderen Komponenten werden über *Berechtigungsschemas* verwaltet. Die Zuweisung eines Berechtigungsschemas erfolgt auf der Attributsseite der jeweiligen Komponente. Ein Berechtigungsschema überprüft, ob vorgegebene Berechtigungskriterien erfüllt werden (Autorisierung). Gängige Typen für Berechtigungsschemas enthalten SQL-Abfragen auf der Basis "Exists" und "Not Exists" sowie die PL/SQL-Funktion zur Rückgabe von booleschen Werten.

5. Häufige Aufgaben

5.1 Lookup-Tabellen

Zwei Tabellen, die über einen Fremdschlüssel miteinander verbunden sind, stehen in einer 1:n-Beziehung, z.B. eine Abteilung mit n Angestellten. Der Fremdschlüssel, der für die Referenzierung des Masterdatensatzes verantwortlich ist, sollte dem Endanwender nicht bekannt sein. Hierfür bietet sich die Möglichkeit, die referenzierte Tabelle als Lookup-Tabelle zu definieren, so dass eine Ersetzung des Fremdschlüsselwerts durch den referenzierten Datensatz geschieht.

5.2 Anlegen einer neuen Anwendung

Eine neue Anwendung wird auf der Homepage von APEX angelegt. Dabei gibt es mehrere Möglichkeiten:

- Völlig neu (from scratch): Dabei können Registerkarten angegeben werden und ein Theme für die Oberfläche.
- Auf vorhandener Anwendung: Dabei kann die Kopie einer anderen Anwendung mit demselben Theme (Template-Satz) für die Benutzeroberfläche ohne die Seiten erstellt werden.
- Auf vorhandener Tabelle: Dabei kann eine Anwendung erstellt werden, die auf vorhandenen Tabellen basiert. Diese Option enthält eine Berichtssseite (Report Page) für jede Seite, eine Form-Seite zum Einfügen und Aktualisieren von Daten sowie eine Seite mit einem Diagramm.
- Demo-Anwendung: Man kann eine Demo-Anwendung erstellen.
- Auf Arbeitsblatt basierend: Auf Basis einer Spreadsheet-Datei kann eine schnell einsetzbare Anwendung erstellt werden (analog zum Data Workshop).

Nach Auswahl eines Anwendungs-Namens wird ein Wizard durchlaufen, um verschiedene Festlegungen zu treffen, z.B.: Name der übergeordneten Registerkarte (Parent Tab), Namen der Registerkarten und Seiten (Pages), Theme. Am Ende kann eine Anwendung ausgeführt werden (Ampel-Symbol). Mit entsprechendem Authentifizierungsschema ist zur Ausführung ein Login erforderlich, was eine Benutzertabelle voraussetzt (z.B. DEMO_USERS).

5.3 Hinzufügen eines Berichts

Das Hinzufügen eines Berichts (Reports) zu einer Seite erfolgt durch Erzeugen einer neuen Region. Im Application Builder wird im Bereich Region das Plus-Symbol angeklickt, um eine neue Region zu erzeugen. In den weiteren Menüs kann dann Bericht bzw. „einfacher Bericht ausgewählt werden“. Mit

Berichtsattributen kann die Darstellung eines Berichts genau gesteuert werden. Zu den Attributen gehören: Berichts-Header, Berichts-Templates, Seitennummerierungen, Sortierungen und Formatierungen. Dementsprechend können Spaltenattribute verändert werden (z.B. Ausrichtung, Überschrift).

5.4 Hinzufügen eines Master-Detail-Formulars

Ein Master-Detail-Formular ist anwendbar auf 1:n-Beziehungen, wenn die Detail-Datensätze zusammen mit dem Master-Datensatz angezeigt und bearbeitet werden sollen. Um ein solches Formular zu erzeugen, sind folgende Schritte notwendig:

Im Application Builder „Seite erstellen“ aufrufen. Anschließend ist „Seite mit Komponente“ auszuwählen, die nächsten Auswahlsschritte sind *Form* bzw. *Master-Detail-Form*. Danach erfolgt dann die Auswahl der *Master-Tabelle* und der zu verwendenden Spalten (entweder alle oder Auswahl einzelner Spalten). In gleicher Weise erfolgt die Definition der *Detail-Tabelle*. Beim Durchlaufen des Wizard erfolgt anschließend die Abfrage nach der *Quelle für die Primärschlüsselwerte*. So ist es z.B. möglich eine Sequenz auszuwählen, die bei Erzeugen eines neuen Satzes den jeweiligen Primärschlüsselwert generiert. Die Sequenz muss allerdings vorher erzeugt worden sein. Mögliche Alternativen sind die Verwendung eines Triggers oder eine selbstgeschriebene PL/SQL-Funktion.

Danach besteht die Möglichkeit, weitere *Master-Optionen* festzulegen: So kann eine Master-Zeilennavigation gewählt werden, d.h. es wird eine Spalte festgelegt für die Navigationsreihenfolge (im Normalfall wird in der Reihenfolge der Primärschlüsselwerte navigiert). Standardmäßig erstellt der Wizard eine Master-Berichtsseite, dieses kann jedoch ausgeschaltet werden.

Als nächste kann das *Layout* der Master-Detail-Form verändert werden. Es gibt die Möglichkeit, die Detail-Datensätze in tabellarischer Form auf derselben Seite darzustellen oder in einer separaten Seite. Es können Seitenattribute wie z.B. Überschriften verändert werden. Zuletzt können *Registerkarten-Optionen* eingestellt werden. So kann eine optionale Registerkartengruppe angegeben werden, die angezeigt wird, wenn Master- und Detailseiten aufgerufen werden. Die angegebene Standardregisterkarte wird auf allen Master-Detailseiten angezeigt. Alle Registerkarten müssen Bestandteil einer Registerkartengruppe sein. Abschließend wird die Erstellung der Master-Detail-Form beendet und kann auch gleich ausgeführt werden (Ampel-Symbol). Hierzu ist u.U. ein Einloggen erforderlich.

5.5 Verändern von Applikations-Objekten

Bei Ausführung einer Applikation kann am unteren Rand die Funktion *Show Edit Links* ausgewählt werden. Daraufhin erscheinen hinter jedem Element 4 Pünktchen, die angeklickt werden können. Je nach Elementtyp können zur Laufzeit die Eigenschaften eines Elements noch verändert werden, so z.B. die Anzeigebreite oder das Datumsformat von einzelnen Feldern.

5.6 Hinzufügen von Wertelisten

Eine Werteliste (List of Values, LOV) kann als gemeinsame Komponente im Application Builder erstellt werden. Dies kann völlig neu geschehen oder durch Kopieren einer Werteliste aus einer anderen Anwendung. Nach dem Kopieren einer Werteliste kann man deren Attribute verändern. Bei der Erstellung einer neuen Liste muss ein Name dafür angegeben werden sowie die Auswahl, ob die Werteliste statisch oder dynamisch sein soll. Statische Listen basieren auf vordefinierten Paaren aus Anzeige- und Rückgabewerten. Dynamische Listen basieren auf einer selbst geschriebenen SQL-Anfragen, die Werte aus einer Tabelle auswählen. Dabei gibt die SQL-Anfrage zwei Spalten zurück. Die erste Spalte ist der Anzeigewert, die zweite Spalte der Rückgabewerte. Wenn eine Anfrage nicht mit SELECT beginnt, wird von einer Funktion ausgegangen, die eine Zeichenfolge zurückgibt, bei der es sich um eine SQL-Anfrage handelt.

Die LOV kann ausgewählt werden bei der Festlegung von Anzeigeeigenschaften für Attribute mit aufzählbaren Wertemengen.

5.7 Anlegen neuer Benutzer

Zugrunde liegt die Authentifizierung von APEX. Die Einrichtung von Benutzern mit entsprechenden Rechten erfolgt bei der Administration des Workspace. Durch Auswahl von *Benutzer verwalten* gelangt man zu einem Hauptmenü, das verschiedene Funktionen anbietet. So können neue Benutzer erstellt werden, Benutzer bearbeitet oder auch entfernt werden. Gleiches gilt für die Verwaltung von Benutzergruppen. Beim Erstellen eines neuen Benutzer-Account kann anschließend ein Berichtsprüfungsschema erstellt werden, das die Berechtigungsprüfung über die im Account gespeicherten Daten (Benutzername, Kennwort) ausführt. Einem Benutzer kann die Berechtigung als Workspace-Administrator erteilt werden. Ebenso kann ein Benutzer einer Benutzergruppe zugeordnet werden. Zu beachten ist, dass es sich bei den Benutzer nicht um Oracle-Benutzer handelt, sondern um Benutzer, die von der Applikation selbst in einer eigenen Tabelle verwaltet werden.

5.8 Hinzufügen von Content zu einer Seite

Wenn beispielsweise eine Übersichtsseite erstellt werden soll mit einem Report und einem Balkendiagramm (Bar Chart), so kann dies mit dem Application Builder erzeugt werden. Entsprechend ist die Seite auszuwählen, die erweitert werden soll. Danach ist eine Region hinzuzufügen (Plus-Symbol). Als Regionstyp ist Bericht auszuwählen, als Implementation SQL-Bericht. Die entsprechenden Attribute (wie Name, Template usw.) werden durch einen Wizard abgefragt. Bei einem SQL-Bericht kann individuell eine SQL-Anfrage formuliert werden. Dabei handelt es sich typischerweise um Datenverdichtungen (d.h. Anwendung von Gruppierung und Aggregationsfunktionen). Der erstellte Bericht kann unmittelbar mittels Run-Symbol ausgeführt werden. Danach ist eine Modifikation der Berichtsattribute möglich.

Analog funktioniert die Erzeugung eines Diagramms (Chart). Verschiedene Typen lassen sich auswählen: z.B. Balkendiagramme oder sogenannte Dials. Anschließend lassen sich die Eigenschaften des Diagramms editieren und natürlich die SQL-Anfrage, die die zu visualisierenden Daten liefert.

Die Positionierung der Elemente auf der Seite geschieht über sogenannte Sequences, die die Darstellungsreihenfolge bestimmen, sowie die Angabe einer Spalte. So wird in der Beispiel-Anwendung (APEX Sample Application) auf der linken Seite das Diagramm dargestellt (Spalte 1), rechts der Bericht (Spalte 2).

Weiterführende Referenzen:

Carsten Czarski: Web-Entwicklung mit der Oracle HTML DB:

<http://aktuell.de.selfhtml.org/artikel/projekt/html-db/>

Creating a Database-Centric Web Application Using Oracle HTML DB (Tutorial):

<http://www.oracle.com/technology/obe/obe10gdb/develop/htmldb/htmldb.htm>

Michael Wunsch, Eike Petersen: [APEX Tutorial](#)

Bernhard Fischer-Wasels: Rund um Oracle APEX, erstes BLOG in DEUTSCH mit Hauptfokus Oracle Application Express (vormals HTML DB):

<http://htmldb-de.blogspot.com/2005/12/ajax-tutorial.html>

Oracle APEX Community (D): <http://www.oracle.com/global/de/community/index.html>

Diskussionforum (E): <http://forums.oracle.com/forums/forum.jspx?forumID=137>