

JÜRGEN SIEBEN, CONDES GMBH & CO. KG

**EFFIZIENTER UND ELEGANTER
PL/SQL-CODE**

JÜRGEN SIEBEN

CONDES GMBH & CO. KG



Oracle SQL -
das umfassende Handbuch



Oracle PL/SQL -
das umfassende Handbuch



Oracle APEX -
das umfassende Handbuch

RUNDE 1: DIE BASICS

RUNDE 1: DIE BASICS

- ▶ Vermeidung veralteter SQL- und PL/SQL-Funktionen
 - ▶ decode vs. case, elsif vs. case
 - ▶ nvl vs. coalesce
 - ▶ Oracle-proprietäre Joinschreibweise vs. ANSI-Join
 - ▶ Verwendung zeitgemäßer Datumsberechnung
 - ▶ Intervall
 - ▶ DATE und TIMESTAMP-Literale etc.

RUNDE 1: DIE BASICS

- ▶ Adaptierung von Best Practices anderer Sprachen, z.B: Code-Smells
 - ▶ keine Code-Duplizierung, kurze Methoden, kein »Spaghetticode«
 - ▶ Sprechende Variablen- und Methodennamen
 - ▶ Vermeidung von Neid: welche Methode gehört in welches Package?
 - ▶ Auslagerung von Utility-Funktionen und Hilfspackages für generische Aufgaben
 - ▶ Reduzierung von CASE-Anweisungen
 - ▶ Vermeidung (der Notwendigkeit) von Kommentaren
 - ▶ »Sachgerechte« Verwendung der Features (Beispiel)

RUNDE 1: DIE BASICS

- ▶ Beispiel: Application Standards Tracker, S. 53, CREATE_COLLECTIONS
 - ▶ Verwendung von CASE
 - ▶ Keine Cursorschleife für Existenztests
 - ▶ Auslagerung der Validierung in generische Assertionsmethode
 - ▶ Trennung von Validierung und Persistenz

RUNDE 2:

ERWEITERTES PL/SQL

RUNDE 2: ERWEITERTES PL/SQL

- ▶ Records und Cursor als Methodenparameter
 - ▶ Vermeidung von Parameterwüsten
 - ▶ Erhöhung der Flexibilität durch %ROWTYPE
- ▶ Objektorientierung in PL/SQL
 - ▶ Beispiel: Beliebige viele Parameter in Methoden
 - ▶ Beispiel: Objekte als Interface (OO-Pattern)

BEISPIEL: BELIEBIG VIELE PARAMETER IN METHODEN

- ▶ TABLE-Objekte können in SQL und PL/SQL verwendet werden
- ▶ Eignen sich als Parameter für Methoden
- ▶ Lösen ansonsten schwer zu programmierende Probleme
- ▶ Beispiel: Übergabe von Parameterlisten an SQL
- ▶ Beispiel: BULK_REPLACE

BEISPIEL: OBJEKTE ALS INTERFACE

- ▶ PL/SQL kennt keine Interfaces
- ▶ Vererbung von Objekten kann als Interface-Mechanismus genutzt werden
- ▶ Erweiterung durch NULL-Objekte
- ▶ Beispiel: Implementierung von Ausgabemodulen in PIT

RUNDE 3:

EINBINDUNG VON SQL

EINBINDUNG VON SQL

- ▶ Aufwändige CASE-Anweisung können Indiz für falsches Datenmodell sein
- ▶ SQL bietet eine extrem mächtige Skriptsprache mit bestens dokumentierter Syntax
- ▶ Viele logische Probleme können einfach durch SQL ausgedrückt werden
- ▶ Auslagerung von Entscheidungslogik in Daten vereinfacht die Programmierung essenziell
- ▶ Beispiel: Code-Generation

EINBINDUNG VON SQL

[Startseite](#) / [Tabellen administrieren](#) /

Tabelle editieren

Tabelle editieren

Tabellentyp ^{*} ?

Attunity

Schema ^{*} ?

UNIT_TEST

Tabellenname (ohne Kürzel) ^{*} ?

UT_ATTUNITY_01

Kurzname ^{*} ?

XXXRT

Parallelität ?

0

Kommentar ?

Quelltabelle für Raw Vault zum Test der Generierung des Archetypen 'Attunity Tabelle'

▶ Spalten editieren

▶ Indizes editieren

▶ RawVault-Views editieren

EINBINDUNG VON SQL

- ▶ CodeGenerator erzeugt DDL-Anweisungen, basierend auf dem Typ der Tabelle und zugehörigen Templates

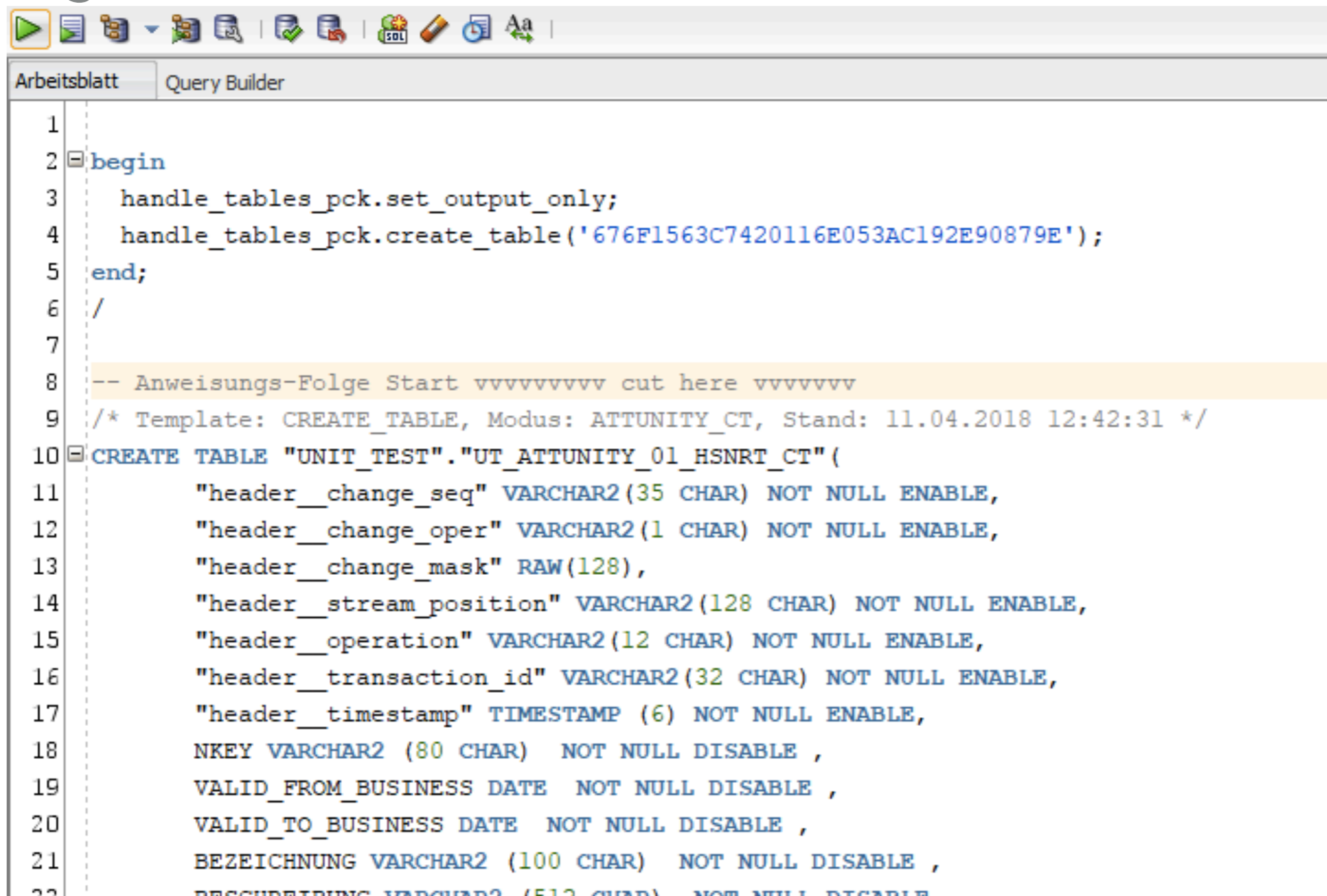
Abfrageergebnis x

Alle Zeilen abgerufen: 152 in 0,147 Sekunden

TMPLT_NAME	TMPLT_TYPE	MODUS	DDL_TEMPLATE	LOG_TEMPLATE
1 RAW_PRE_U...	STMT	DEFAULT	insert into #OWNER#.RAW_#TABLE_NAME# (RV_VALID_FROM,RV_VALID...	(null)
2 RAW	DATA_TYPE	DEFAULT	#DATA_TYPE# (#DATA_LENGTH#)	(null)
3 NCHAR	DATA_TYPE	DEFAULT	#DATA_TYPE# (#DATA_LENGTH# #UNIT#) #DEFAULT# #NOT_NULL# #CHECK#	(null)
4 STATIC_DE...	DELTA_VIEW	STATIC...	create or replace force view #OWNER_ID#.V_#TABLENAME# asWITH...	Erzeuge Delta-View #OWNER_ID#.V...
5 REMOTE_DA...	DELTA_VIEW	REMOTE...	create or replace force view #OWNER_ID#. #VIEWNAME# as select...	Erzeuge View #OWNER_ID#. #VIEWNAME#
6 DELTA_VIE...	DELTA_VIEW	DELTA	CREATE OR REPLACE FORCE VIEW #OWNER_ID#.V_#TABLENAME# ASWITH...	Erzeuge DeltaView #OWNER_ID#.V...
7 QS_QUERY	QS_QUERY	DEFAULT	WITH q AS (#QUERY#)SELECT r.rule_action, r.condit...	(null)
8 CHAR	DATA_TYPE	DEFAULT	#DATA_TYPE# (#DATA_LENGTH# #UNIT#) #DEFAULT# #NOT_NULL# #CHECK#	(null)
9 CLOB	DATA_TYPE	DEFAULT	#DATA_TYPE# #DEFAULT# #NOT_NULL#	(null)
10 DATE	DATA_TYPE	DEFAULT	#DATA_TYPE# #DEFAULT# #NOT_NULL# #CHECK#	(null)
11 NUMBER	DATA_TYPE	DEFAULT	#DATA_TYPE# (#PRECISION#,#SCALE#) #DEFAULT# #NOT_NULL# #CHECK#	(null)
12 NVARCHAR2	DATA_TYPE	DEFAULT	#DATA_TYPE# (#DATA_LENGTH# #UNIT#) #DEFAULT# #NOT_NULL# #CHECK#	(null)
13 TIMESTAMP (0)	DATA_TYPE	DEFAULT	#DATA_TYPE# #DEFAULT# #NOT_NULL# #CHECK#	(null)
14 TIMESTAMP (6)	DATA_TYPE	DEFAULT	#DATA_TYPE# #DEFAULT# #NOT_NULL# #CHECK#	(null)
15 VARCHAR2	DATA_TYPE	DEFAULT	#DATA_TYPE# (#DATA_LENGTH# #UNIT#) #DEFAULT# q'[]' #NOT_NUL...	(null)
16 VARCHAR2	DATA_TYPE	TARGET	#COLUMN_NAME# #DATA_TYPE# (#DATA_LENGTH# #UNIT#) #DEFAULT_VA...	(null)
17 VARCHAR2	DATA_TYPE	EXTERN...	"#COLUMN_NAME#" #FORMAT_MASK_SOURCE#	(null)
18 NUMBER	DATA_TYPE	TARGET	#COLUMN_NAME# #DATA_TYPE# #PRECISION ((38##SCALE ,))# #...	(null)
19 CHAR	DATA_TYPE	TARGET	#COLUMN_NAME# #DATA_TYPE# (#DATA_LENGTH# #UNIT#) #DEFAULT_VA...	(null)
20 DATE	DATA_TYPE	TARGET	#COLUMN_NAME# #DATA_TYPE# #DEFAULT_VALUE DEFAULT ^DEFAULT ON	(null)

EINBINDUNG VON SQL

► Ergebnis des Aufrufs



```
1
2 begin
3   handle_tables_pck.set_output_only;
4   handle_tables_pck.create_table('676F1563C7420116E053AC192E90879E');
5 end;
6 /
7
8 -- Anweisungs-Folge Start vvvvvvvvvv cut here vvvvvvvv
9 /* Template: CREATE_TABLE, Modus: ATTUNITY_CT, Stand: 11.04.2018 12:42:31 */
10 CREATE TABLE "UNIT_TEST"."UT_ATTUNITY_01_HSNRT_CT" (
11     "header_change_seq" VARCHAR2(35 CHAR) NOT NULL ENABLE,
12     "header_change_oper" VARCHAR2(1 CHAR) NOT NULL ENABLE,
13     "header_change_mask" RAW(128),
14     "header_stream_position" VARCHAR2(128 CHAR) NOT NULL ENABLE,
15     "header_operation" VARCHAR2(12 CHAR) NOT NULL ENABLE,
16     "header_transaction_id" VARCHAR2(32 CHAR) NOT NULL ENABLE,
17     "header_timestamp" TIMESTAMP (6) NOT NULL ENABLE,
18     NKEY VARCHAR2 (80 CHAR) NOT NULL DISABLE ,
19     VALID_FROM_BUSINESS DATE NOT NULL DISABLE ,
20     VALID_TO_BUSINESS DATE NOT NULL DISABLE ,
21     BEZEICHNUNG VARCHAR2 (100 CHAR) NOT NULL DISABLE ,
22     BESCHREIBUNG VARCHAR2 (512 CHAR) NOT NULL DISABLE
```