



Ausgabe in Excel leicht gemacht mit dem JDD Spreadsheet Publisher

Dietmar Aust
JDD-Software UG
www.jdd-software.com



▶ Dietmar Aust



▶ 15 Jahre Entwicklung von Webapplikationen auf Basis von Oracle, seit 2006 nur APEX

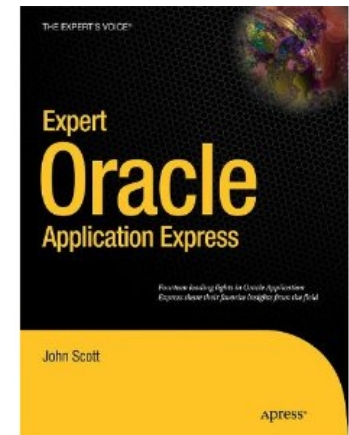
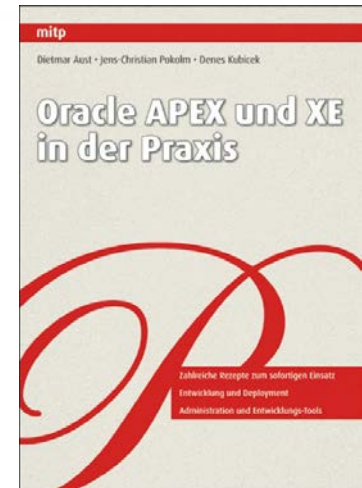
▶ Projektgeschäft, Trainings / Coachings, BLOG, Forum, Bücher

▶ Bisher bekannt als Opal-Consulting



▶ 06/2012 Gründung der JDD Software UG zusammen mit Jens Gauger

▶ Zielsetzung: Produkte für Oracle APEX Entwickler



Problemklassen: Datentyp-konformer Export nach MS Excel

Customer	Period	Registration	Amount	Ident	Street	Street No	City	Zip	Country
Dietmar Aust	01/2012	01.01.12	€10.000,11	0123456789011111	Zum Tilmeshof	11	Köln	50859	Deutschland
Manfred Mustermann	01/2012	01.01.13	€5.000,11	01234567890a	Zum Tilmeshof	11a	Dresden	01067	Deutschland
Manfred Mustermann	01/2012	01.01.13	€5.000,11	0123456789022222222	Am Markt	3500	Leipzig	04178	Deutschland
Manfred Mustermann	01/2012	01.01.13	€5.000,11	012345678901a	Karlsruher Allee	99b	Halle (Saale)	06132	Deutschland
Manfred Mustermann	01/2012	01.01.13	€5.000,11	01234567890	Am Markt	3500	Leipzig	04178	Deutschland

	A	B	C	D	E	F	G	H	I	J
1	Customer	Period	Registration	Amount	Ident	Street	Street No	City	Zip	Country
2	Dietmar Aust	Jan 12	01.01.12	€ 10.000,11	1,23457E+14	Zum Tilmesh	11	Köln	50859	Deutschland
3	Manfred Mu	Jan 12	01.01.13	€ 5.000,11	01234567890a	Zum Tilmesh	11a	Dresden	1067	Deutschland
4	Manfred Mu	Jan 12	01.01.13	€ 5.000,11	1,23457E+18	Am Markt	3500	Leipzig	4178	Deutschland
5	Manfred Mu	Jan 12	01.01.13	€ 5.000,11	012345678901a	Karlsruher A	99b	Halle (Saale)	6132	Deutschland
6	Manfred Mu	Jan 12	01.01.13	€ 5.000,11	1234567890	Am Markt	3500	Leipzig	4178	Deutschland

► Excel interpretiert und verändert die Daten automatisch!

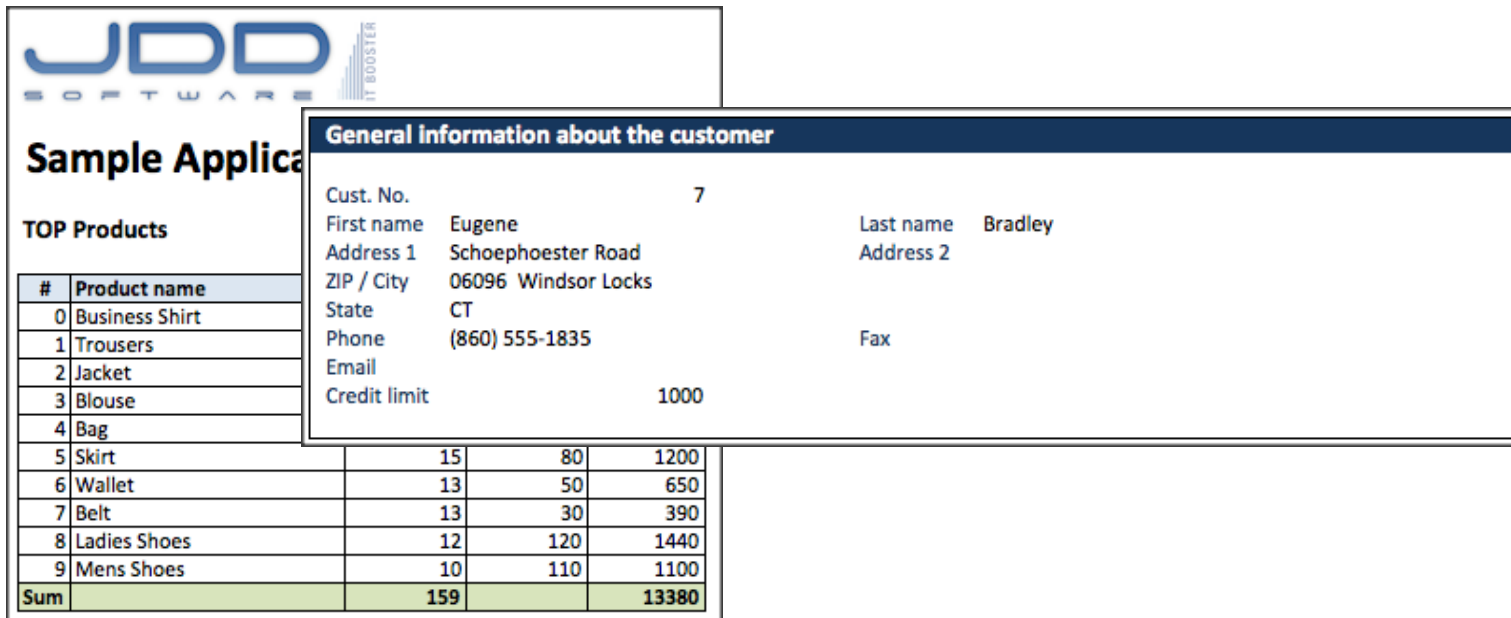
- Hausnummern werden entweder als Zahl oder Text interpretiert (11, 11a)
- PLZ: 01067 wird als Zahl 1067 interpretiert – Führende Nullen sind weg!!!
- Auftragsnummern/Ident-Codes mit mehr als 12 Stellen => E-Schreibweise

Problemklassen

Formatierung der Inhalte

► Formatierung der Inhalte:

- Logos, Informationstext, Schriftarten, Farben, Zahlenformate, Datumsformate, Währung, etc.



The screenshot displays a software application interface. At the top left is the logo for JDD SOFTWARE IT BOOSTER. Below the logo, the text "Sample Application" is visible. A popup window titled "General information about the customer" is open, showing the following details:

Cust. No.	7	Last name	Bradley
First name	Eugene	Address 1	Schoephoester Road
Address 2		Address 2	
ZIP / City	06096 Windsor Locks		
State	CT		
Phone	(860) 555-1835	Fax	
Email			
Credit limit	1000		

Below the popup, a table titled "TOP Products" is displayed:

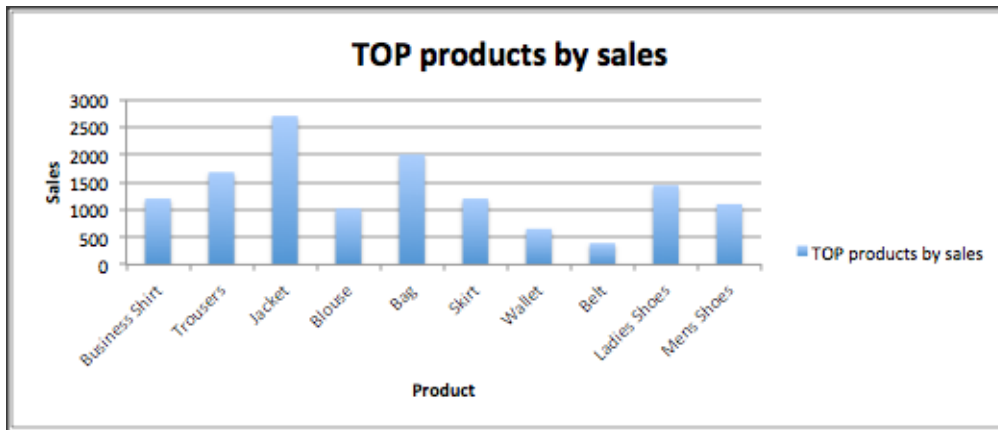
#	Product name			
0	Business Shirt			
1	Trousers			
2	Jacket			
3	Blouse			
4	Bag			
5	Skirt	15	80	1200
6	Wallet	13	50	650
7	Belt	13	30	390
8	Ladies Shoes	12	120	1440
9	Mens Shoes	10	110	1100
Sum		159		13380

Problemklassen

Nutzung nativer Funktionalitäten

► Nutzung nativer Funktionalitäten:

- Autofilter,
- Makros,
- konditionale Formatierung,
- ausgeblendete Spalten,
- Charts,
- Pivot-Tabellen,
- Passwort-Schutz für Blätter und die Arbeitsmappe




Problemklassen

Produktivität der Implementierung

- ▶ Vom Ablauf her bekommen wir oft die MS Excel Vorlage durch Fachseite direkt geliefert
 - Verwenden wir eine Reporting Engine, so müssen wir das Layout kostenintensiv nachbilden
- ▶ Wartbarkeit ?!

Beispiele

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR
2																										Kurzauftrag für EC-Cash																	
4	 Telekom Shop Vertriebsgesellschaft mbH																									Terminal-ID 1. Terminal-ID																	
5	Art des Auftrags \${shdb_standorte.art}																																										
7	Wir bestellen hiermit nachfolgende Lieferungen/Dienstleistungen zu den umseitigen allgemeinen Geschäftsbedingungen der TeleCash sowie den Bedingungen für die Teilnahme am electronic cash System, bzw. den Bedingungen für die Teilnahme am System "Geldkarte" und den Händlerbedingungen für die Teilnahme am POZ-System.																																										
13	Telefon: 089/899643-0, Fax: 089/899643-21																																										
14	Transact Elektronische Zahlungssysteme GmbH																																										
16																																											
18																																											
20																																											
23	1 Kunde												5 eurocheque (ec)-Karten																														
25	Firma:		Telekom Shop Vertriebsgesellschaft mbH																							Das TeleCash Terminal soll wie folgt konfiguriert sein:																	
27	Straße:		\${shdb_standorte.sta_strasse} \${shdb_standorte.sta_ort}																							<input checked="" type="checkbox"/> Wir wollen Zahlungen mit dem electronic-Cash-Verfahren abwickeln. (Garantierte Zahlungen mit online-Abfrage und Eingabe der Geheimzahl PIN)																	
29	PLZ/Ort:		\${shdb_standorte.sta_plz} \${shdb_standorte.sta_ort}																																								
31	Telefon:		\${shdb_standorte.sta_telefon}		Fax:		\${shdb_standorte.sta_fax}																																				
33	Ansprechpartner:		\${shdb_standorte.sta_vp_leiter}																																								
35	Telefon:		\${shdb_standorte.sta_telefon}		Fax:		\${shdb_standorte.sta_fax}																																				
39	8 Anschluß-Art LAN																																										
41																																											
43																																											
45	9 Ausdruck auf dem Kassensstreifen																																										
47	max. 24 Zeichen/Zeile, Breit 12Z/Z, insg. Max. 62 Zeichen																																										
49	Text normal																																										
51	Firma:		Telekom Shop Vertriebsgesellschaft																							<input checked="" type="checkbox"/>																	
53	Straße:		\${shdb_standorte.sta_strasse} \${shdb_standorte.sta_ort}																							<input checked="" type="checkbox"/>																	
55	PLZ/Ort:		\${shdb_standorte.sta_plz} \${shdb_standorte.sta_ort}																							<input checked="" type="checkbox"/>																	

Beispiele

	A	B	(D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF
1	Empfänger																	Absender:														
2	DEUTSCHE TELEKOM AG CC CON CON A23 SSS 00000 Town																	Deutsche Telekom														
3																		Telekom Shop Vertriebsgesellschaft GmbH														
4																		Kontaktangaben für Debitorenrückmeldung														
5																		E-Mail: Email-Address@customer.de														
6																		Telefon: 000/000					Telefax: 000/000									
7	Deutsche Telekom																	Debitoren - Stammdaten														
8																		Konto Gruppe : INTN / INTR														
9	<input type="text" value="{shdb_standorte.art}"/>																															
10	wirksam ab <input type="text" value="{shdb_standorte.wirks"/>																															
11	Debitoren-Nummer <input type="text" value="ndorte.sta_vp"/> Kto-Grp. <input type="text" value="INTR"/>																															
12	GSB <input type="text" value="OA"/> OZT <input type="text" value="9949"/> Region <input type="text" value="ndorte.sta_"/>																															
13	Firma <input type="text" value="Telekom Shop Vertriebsgesellschaft mbH"/>																															
14	OrgE <input type="text"/>																															

Beispiele

	AB	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI		
1																																				
2			Filialanmeldung:																																	310
3			Terminal:																																	
4																																				
5																																				
6			xxx																																	
7			Lyoner Sr. 36																																	
8																																				
9			111111 Frankfurt am Main																																	
10																																				
11																																				
12																																				
13																																				
15																																				
16																																				
18			Rechnungsanschrift													Filialanschrift																				
19																																				
20			Name:																																	
21			c/o:																																	
22			Straße:																																	
23			PLZ/Ort:																																	
24			Telefon:																																	
25			Fax:																																	
27			Terminal-Informationen													Bankverbindung oder Zentrale Abrechnung																				
28																																				
29			Netzbetreiber:																																	
30			Terminalbezeichnung:																																	
31			Referenznummer:																																	
32																																				
33			Terminal-Nr./ID:																																	

Export nach MS Excel - Ansätze

► Normaler APEX Mechanismus, Export nach CSV

- Workaround, um führende Nullen korrekt darzustellen
- => Textformat erzwingen (z.B. für 04178 Formatierung als Formel: ="04178")

```
select  country
        , ident
        , customer
        , customer_registered_on
        , customer_period
        , sum_of_purchases
        , street
        , street_no
        , city
        , decode(:request, 'CSV', '='||zip||'', zip) zip
from    jdd_csv_demo
```

Export nach MS Excel - Ansätze

► Workaround funktioniert bei interaktiven Berichten nicht! => CSV Export LEER!

☐ Zip = '04178'

Customer	Period	Registration	Amount	Ident	Street	Street No	City	Zip	Country
Manfred Mustermann	01/2012	01.01.13	€5.000,11	0123456789022222222	Am Markt	3500	Leipzig	04178	Deutschland
Manfred Mustermann	01/2012	01.01.13	€5.000,11	01234567890	Am Markt	3500	Leipzig	04178	Deutschland

1 - 2

```
select "CUSTOMER", "CUSTOMER_PERIOD", "CUSTOMER_REGISTERED_ON" , "SUM_OF_PURCHASES", "IDENT",  
"STREET", "STREET_NO" , "CITY", "ZIP", "COUNTRY",  
count(*) over () as apxws_row_cnt  
from  
(select *  
from  
(select country , ident , customer , customer_registered_on , customer_period ,  
sum_of_purchases , street , street_no , city , decode(:request, 'CSV', '='||zip||'', zip)  
zip  
from jdd_csv_demo  
) r  
where  
(  
"ZIP" = :APXWS_EXPR_1  
)  
) r  
where rownum <= to_number(:APXWS_MAX_ROW_CNT);
```

Export nach MS Excel - Ansätze

► Manueller Export nach CSV mit Stored procedure

- <http://spendolini.blogspot.de/2006/04/custom-export-to-csv.html>
- <http://www.astral-consultancy.co.uk/cgi-bin/hunbug/doco.cgi?11270>

► Manueller Export nach HTML mit Stored Procedure, Mime-Type=MS Excel setzen

- <http://roelhartman.blogspot.de/2006/07/how-to-create-neatly-formatted-excel.html>

	A	B	C	D	E
1	EMPLOYEE INFO				
2					
3	Test				
4	Customer	Period	Ident	Street No	ZIP
5	Dietmar Aust	01.01.12	1,23457E+14	11	50859
6	Manfred Mustermann	01.01.12	01234567890a	11a	1067
7	Manfred Mustermann	01.01.12	1,23457E+18	3500	4178
8	Manfred Mustermann	01.01.12	012345678901a	99b	6132
9	Manfred Mustermann	01.01.12	1234567890	3500	4178

Bei CSV / HTML:
Datentypen sind
fehlerhaft!

Export nach MS Excel - Ansätze

- ▶ Manueller Export nach xlsx, ZIP File Manipulation
- ▶ Plugins, z.B. von Anton Scheffer
 - http://apex-plugin.com/oracle-apex-plugins/process-type-plugin/ir-report-to-excel-xlsx_74.html
 - Nur max. 26 Spalten exportierbar, Zahlen nicht erkannt, Summe nicht bildbar

Customer	Customer Register	Customer Period	Sum Of Pu	Street	Street No	City	Zip	Country
Dietmar Au	1.Jan.2012	1.Jan.2012	10000,11	Zum Tilmes	11	Köln	50859	Deutschland
Manfred M	1.Jan.2013	1.Jan.2012	5000,11	Zum Tilmes	11a	Dresden	01067	Deutschland
Manfred M	1.Jan.2013	1.Jan.2012	5000,11	Am Markt	3500	Leipzig	04178	Deutschland
Manfred M	1.Jan.2013	1.Jan.2012	5000,11	Karlsruher	99b	Halle (Saale)	06132	Deutschland
Manfred M	1.Jan.2013	1.Jan.2012	5000,11	Am Markt	3500	Leipzig	04178	Deutschland

- ▶ BI-Publisher, teuer und Integration nicht trivial
- ▶ FOP Implementierung mit Cocoon:
<http://ubuntuforums.org/showthread.php?t=1004742>

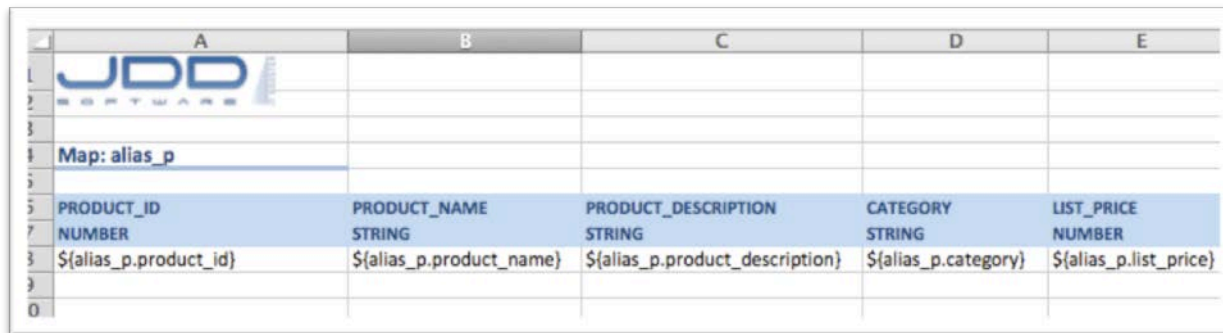
JDD-Spreadsheet Publisher - Konzept

► Konzept:

1. Daten bereitstellen (<n> SQL Abfragen, die jeweils über einen Alias unterschieden werden)
 - Bsp.: `alias_p` für Produktinformationen:

```
SELECT PRODUCT_ID,  
       PRODUCT_NAME,  
       PRODUCT_DESCRIPTION,  
       CATEGORY,  
       LIST_PRICE  
FROM demo_product_info
```

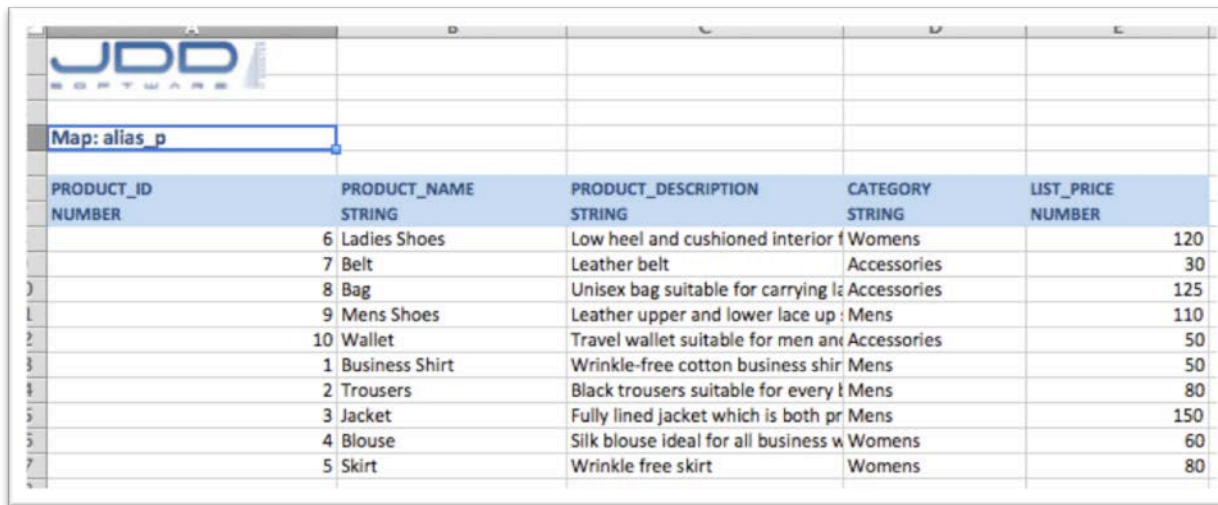
2. Excel "Vorlage" enthält Referenzen zu den Abfragen und den jeweiligen Spalten über `${<Alias der Abfrage>.<Spaltenname>}`:



	A	B	C	D	E
1	JDD SOFTWARE				
2					
3					
4	Map: alias_p				
5	PRODUCT_ID	PRODUCT_NAME	PRODUCT_DESCRIPTION	CATEGORY	LIST_PRICE
6	NUMBER	STRING	STRING	STRING	NUMBER
7	\$(alias_p.product_id)	\$(alias_p.product_name)	\$(alias_p.product_description)	\$(alias_p.category)	\$(alias_p.list_price)
8					
9					
10					

► Konzept:

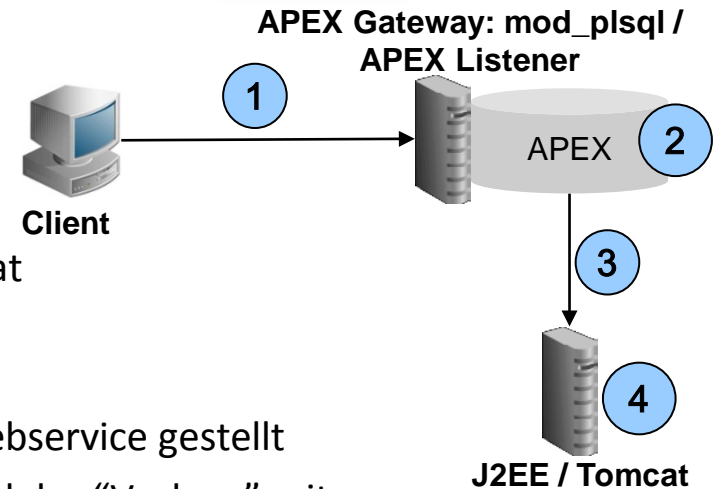
3. Zur Laufzeit werden die Platzhalter in der Excel "Vorlage" durch die Daten der SQL-Abfragen ersetzt, Formate bleiben erhalten.



PRODUCT_ID NUMBER	PRODUCT_NAME STRING	PRODUCT_DESCRIPTION STRING	CATEGORY STRING	LIST_PRICE NUMBER
6	Ladies Shoes	Low heel and cushioned interior f	Womens	120
7	Belt	Leather belt	Accessories	30
8	Bag	Unisex bag suitable for carrying l	Accessories	125
9	Mens Shoes	Leather upper and lower lace up	Mens	110
10	Wallet	Travel wallet suitable for men and	Accessories	50
1	Business Shirt	Wrinkle-free cotton business shir	Mens	50
2	Trousers	Black trousers suitable for every t	Mens	80
3	Jacket	Fully lined jacket which is both pr	Mens	150
4	Blouse	Silk blouse ideal for all business w	Womens	60
5	Skirt	Wrinkle free skirt	Womens	80

► Konzept:

1. Client fordert Excel Ausgabe an
2. PLSQL API generiert Anfrage an J2EE Server:
 - Daten der SQL Abfragen im XML Format
 - Excel "Vorlage" (Uuencoded)
3. Über utl_http wird der Request an einen Webservice gestellt
4. Der Webservice generiert aus den Daten und der "Vorlage" mit Apache POI dann ein befülltes Excel Sheet.



▶ API Aufruf für Entwickler

```
CREATE OR REPLACE PROCEDURE generate_excel(p_blob IN BLOB)
IS
  l_result  JDD_XLSP_PUBLISHER.result_t;
  l_queries JDD_XLSP_PUBLISHER.queries_t;
BEGIN
  l_queries.EXTEND ();
  l_queries(1).alias := 'alias_p';
  l_queries(1).sql_text :=
    'SELECT PRODUCT_ID, PRODUCT_NAME, PRODUCT_DESCRIPTION,
      CATEGORY, LIST_PRICE
    FROM demo_product_info';

  l_result_t :=
    JDD_XLSP_PUBLISHER.render_template (
      p_queries => l_queries,
      p_template => p_blob);

  -- Once you have the generated excel sheet as a BLOB,
  -- you can either store it in a database table, send it attached to an email
  -- or output it directly in the user's browser:
  OWA_UTIL.mime_header ('application/vnd.ms-excel', FALSE);
  OWA_UTIL.http_header_close;
  WPG_DOCLOAD.download_file (p_blob => l_result.document);
END;
/
```

- ▶ APEX Sample Application erweitert um Excel Ausgaben
- ▶ Demo erstellen mit Template Manager
 - Neues Template mit Wizard (alias_users und alias_sales)
 - Alias_users, alias_sales
 - Select * from jdd_csv_demo
 - Geschütztes Arbeitsblatt, ausgeblendete Spalten
 - Table_as_is => original Demo
- ▶ Interactive Report Demo für 4.2

► Einfache Notation:

- `${alias.column}`

► Advanced Syntax:

- Operatoren mit JEXL anwenden `${p.quantity * p.list_price}`
- Aggregationen: sum, min, max, avg, count
- Explizite Schleifen mit `<jx:forEach ..> </jx:forEach>`
- Bedingungen mit `<jx:if ..> </jx:if>`
- Daten filtern `<jx:forEach select="${products.list_price} >= 100}" ..> </jx:forEach>`
- Daten gruppieren `<jx:forEach groupBy="product_name" ..> </jx:forEach>`

- ▶ Formeln in Excel
- ▶ Für die Verwendung von Formeln im Publisher wird die Formel noch mit `$[< Formel >]` umgeben.
- ▶ Bezüge werden dynamisch aufgelöst:
 - Bsp.: Es werden 8 Datensätze dynamisch eingefügt

`$[SUM(C2) * 2]`

=>

`SUM(C2:C10) * 2`

Vorteile

- ▶ Korrekte Datentypen!
- ▶ Saubere Entkopplung der Daten vom Layout!
- ▶ Unterstützung multipler Abfragen, diese können unterschiedlich formatiert werden
- ▶ Kann über DBMS_SCHEDULER / DBMS_JOB serverseitig ohne APEX die Berichte generieren und per Mail verschicken
- ▶ Corporate Design einfach umzusetzen => Die Fachseite liefert schon die Vorlage, muss nicht mehr in Reporting Layout übersetzt werden!
- ▶ Einfach zu warten, Anpassungen sind sehr schnell umzusetzen

- ▶ “Nur” ein Webservice, ist also grundsätzlich technologie-neutral
 - ▶ PL/SQL API übernimmt Kommunikation mit Webservice
 - ▶ APEX Template Manager stellt Tabellen und Management GUI bereit
-
- ▶ Kunden
 - Telekom Shop Vertriebsgesellschaft mbH
 - T-Systems International GmbH
 - Robert Bosch GmbH
 - Siemens AG
 - Weber Data Service IT GmbH

Q&A

Dietmar Aust
Opal-Consulting, Köln

www.opal-consulting.de
daust.blogspot.com
dietmar.aust@opal-consulting.de

JDD Spreadsheet Publisher - Notation

► Einfache Notation:

- `${alias.column}`

► Berechnung mit JEXL:

- The overall stock value of product `${alias_p.product_name}` is `${alias_p.quantity * alias_p.list_price}`

► Aggregationen:

- `${Function_Name(field):collection}`, sum, min, max, avg, count

Total count of products: <code>\${count(product_id):alias_products}</code>		
---	--	--

Total count of products: 10		
------------------------------------	--	--

► (explizite) Schleifen mit Foreach

```
<jx:forEach items="${alias_products}" var="products">
  ${products.product_name}
</jx:forEach>
```

<jx:forEach items="\${alias_products}" var="products">		
\${products.product_name}	\${products.quantity}	\${}
</jx:forEach>		

► Gruppierungen

0	<jx:forEach items="\${alias_products}" groupBy="product_name">			
1	Product: \${group.item.product_name}			
2	<jx:forEach items="\${group.items}" var="products">			
3	\${products.quantity}	\${products.list_price}	\${products.sales}	\${products.product_id}
4	</jx:forEach>			
5	</jx:forEach>			

► Gruppierungen

QUANTITY NUMBER	LIST_PRICE NUMBER	SALES NUMBER	PRODUCT_ID NUMBER
<i>Product: Belt</i>			
2	30	60	7
<i>Product: Wallet</i>			
2	50	100	10
<i>Product: Business Shirt</i>			
3	50	150	1

► Daten filtern

```
<jx:forEach items="{alias_products}" var="products"  
  select="{products.list_price >= 100}"  
    {products.product_name}  
</jx:forEach>
```