ORACLE®

Fundgrube Standard-Edition
Funktionen, die Sie sich schon immer gewünscht haben

Carsten Czarski
Business Unit Database
ORACLE Deutschland B.V. & Co KG

Was Sie sich schon immer gewünscht haben ...?

- **Anwendungen wirklich schnell entwickeln**
  Application Express
- **Benutzerfehler selbst – ohne Backup – korrigieren**
  Flashback Query
- **Relevante Informationen besser finden**
  Ähnlichkeitssuche, Oracle TEXT
- **Gängige Aufgaben schneller erledigen**
  Kreuztabellen, reguläre Ausdrücke, XML erzeugen, ...
- **Daten einfacher auswerten**
  Analytische Funktionen

Die umfangreiche Grundausstattung der Datenbank

ANWENDUNGEN WIRKLICH SCHNELL ENTWICKELN

Erwartungen an die Anwendungsentwicklung

Budget einhalten

Kurze Lernkurve

Intuitive Oberfläche

Skalierbarkeit

Rechtzeitige Fertigstellung

Moderne Anwendung

Themen der Entwickler

Objektrelationales Mapping

MVC Frameworks

POJOs, EJB

Garbage Collection

Deployment-Zyklen

Build-Zeiten

Erwartungen an die Anwendungsentwicklung

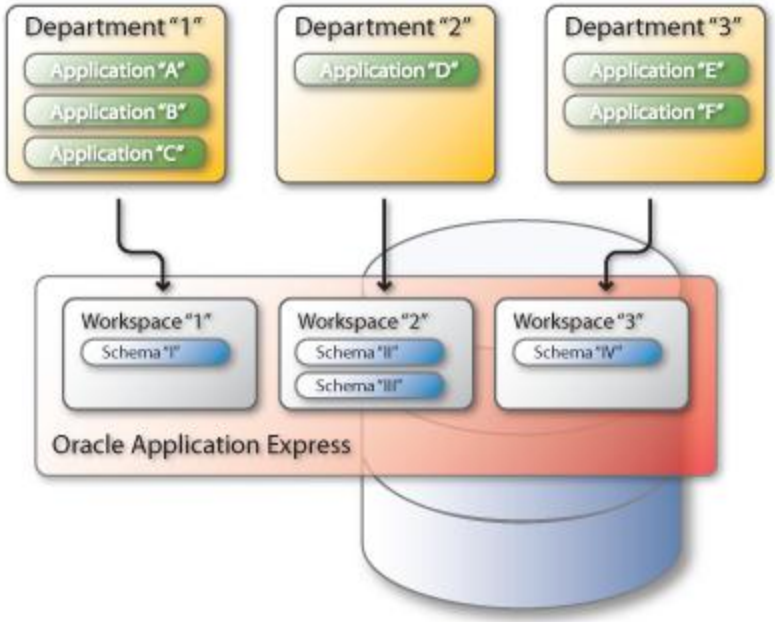
- Bedarf nach unabhängigen Lösungen
- Bedarf nach schnellen Lösungen
- Bedarf nach einfachen Lösungen
- Bedarf nach günstigen Lösungen

Lösung: Oracle Application Express

- Die Vorteile einer Einzelplatzdatenbank ….
  - Einfache Bedienung
  - Schnelle Ergebnisse
  - Flexibilität
- Kombiniert mit einer zentralen Datenbank
  - Sicherheitskonzepte
  - Datenintegrität
  - Skalierbarkeit
  - Verfügbarkeit
- ... durch Web-Technologie überall verfügbar!

APEX Workspaces

- Unabhängige, private Arbeitsbereiche
  - Unabhängige Entwicklung und Administration
  - Eigene Applikationsverwaltung
  - Eigene Userverwaltung
- Zentrale Datenbank
  - Zentrale Wartung
  - Zentrale Datenhaltung



Daten ins Web bringen – schneller geht es nicht!

Go Actions

Deptno

Hohe Gehälter

Deptno : 10

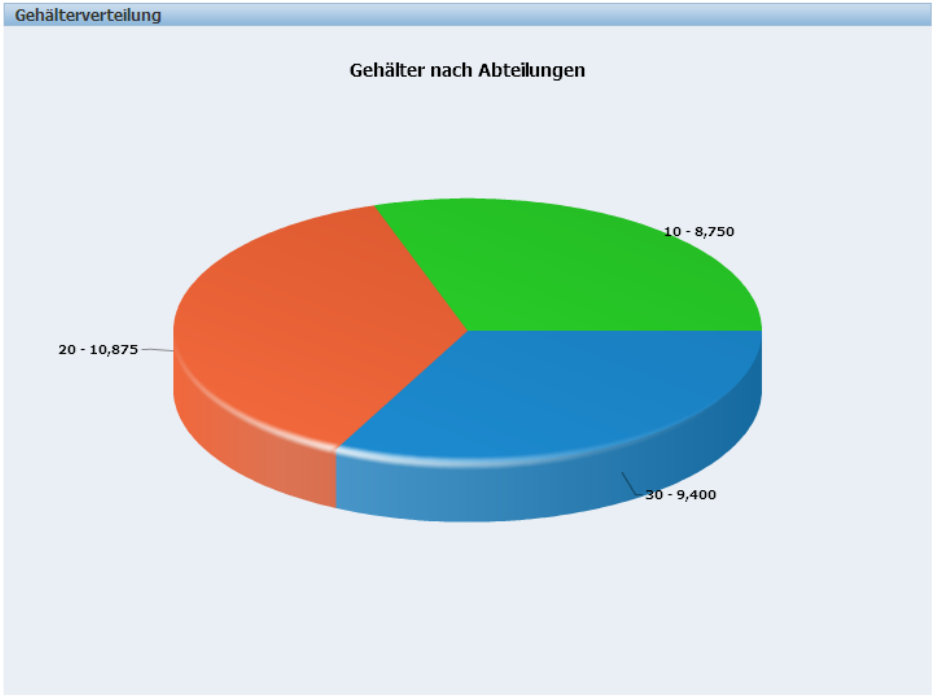
	Empno	Ename	Job	Mgr	Hiredate	Sal	Comm
	7839	KING	PRESIDENT	-	17-NOV-81	5000	-
	7782	CLARK	MANAGER	7839	09-JUN-81	2450	-
	7934	MILLER	CLERK	7782	23-JAN-82	1300	-
						8750	

Deptno : 20

	Empno	Ename	Job	Mgr	Hiredate	Sal	Comm
	7788	SCOTT	ANALYST	7566	09-DEC-82	3000	-
	7902	FORD	ANALYST	7566	03-DEC-81	3000	-
	7566	JONES	MANAGER	7839	02-APR-81	2975	-
	7876	ADAMS	CLERK	7788	12-JAN-83	1100	-
	7369	SMITH	CLERK	7902	17-DEC-80	800	-
						10875	

Deptno : 30

	Empno	Ename	Job	Mgr	Hiredate	Sal	Comm
	7698	BLAKE	MANAGER	7839	01-MAY-81	2850	-
	7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300
	7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0
	7521	WARD	SALESMAN	7698	22-FEB-81	1250	500
	7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400



Die umfangreiche Grundausstattung der Datenbank

BENUTZERFEHLER OHNE BACKUP KORRIGIEREN

Flashback Query

- Abfrage mit AS OF-Klausel oder DBMS_FLASHBACK
  - Alte Zustände wieder sichtbar
  - Über Transaktionsgrenzen hinweg

```
select * from emp
as of timestamp (systimestamp - interval '5' minute)
```

```
select * from emp
as of scn (871876)
```

- Merkmale
  - Informationen werden aus UNDO-Tablespace geholt
  - Steuerung durch Parameter UNDO_RETENTION
  - Tabellendefinition darf sich nicht ändern

Flashback Query: Package DBMS_FLASHBACK

- Flashback "aktivieren"

```
dbms_flashback.enable_at_time(
  query_scn => (systimestamp - interval '5' minute)
);
dbms_flashback.enable_at_system_change_number(
  query_time => 871876
);
```

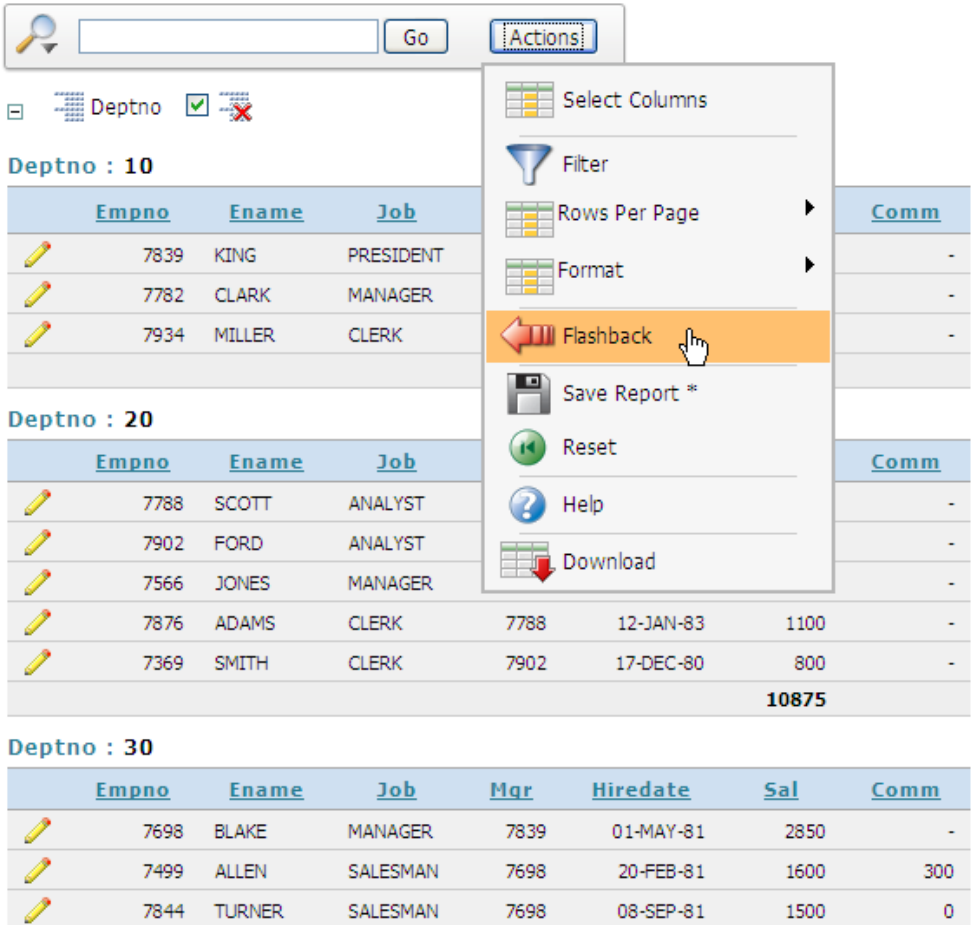
- Abfragen durchführen

```
select * from emp
as of timestamp (systimestamp - interval '5' minute)
```

- Flashback deaktivieren

```
dbms_flashback.disable;
```

Fashback Query für den Endanwender



Flashback Query

- Wie weit kommt man "in die Vergangenheit" ...?
- Abhängig von
  - Der Größe des UNDO-Tablespace
  - Der Transaktionslast auf der Datenbank

- Parameter UNDO_RETENTION
  - Angabe in Sekunden: Default ist **900 = 15 Minuten**
  - "Ziel" der Datenbank für Flashback Query
  - UNDO-Tablespace wächst entsprechend, wenn möglich

Die umfangreiche Grundausstattung der Datenbank

RELEVANTE INFORMATIONEN BESSER FINDEN

Unscharfe Namenssuche

Oracle 11.2.0.2

Oracle Text Name Search Feature

- Namenssuche
  - mit Rücksicht auf verschiedene Schreibweisen
  - in unterschiedlichen Kulturen
  - Anwendung eines Regelwerks
- Basis ist Oracle TEXT
  - Volltextindex nötig – User Data Store für relationale Daten
  - SQL-Funktion CONTAINS mit NDATA-Operator

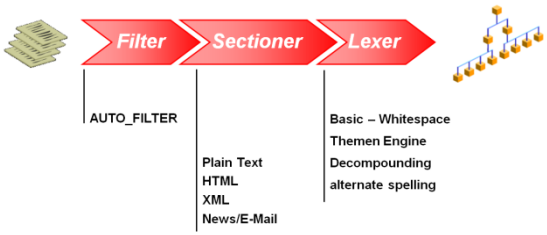
```
select * from names
where contains(name, 'NDATA(name,Tsaarski)') > 0

ID          NAME
---------- ----------------------------------------
          4 Carsten Czarski
```

Exkurs: Was ist Oracle TEXT?

- Spezieller Volltextindex (CTXSYS.CONTEXT)
- SQL-Funktionen zur Recherche (CONTAINS)
  - Linguistische Abfragefunktionen
  - Relevanz-Ranking
  - Ergebnis-Aufbereitung: *Highlight, Markup, Snippet*
  - Volltextindex zwingend nötig

```
select score(1), dokument
from dokument_tab
where CONTAINS(dokument, 'Software AND Oracle') > 0
```

Oracle TEXT: Indizierung



- Unterstützung aller gängigen Datentypen
  - XMLTYPE, VARCHAR2, CLOB, BLOB
  - Filter bei Binärdaten
- Achtung: Textindex ist *asynchron*
- Mehrere Indizes pro Tabelle möglich

```
create index idx_textindex
on dokument_tab (dokument)
indextype is CTXSYS.CONTEXT
parameters ('{index-parameters}')
```

Oracle TEXT: Abfragemöglichkeiten I

- Exakte Wort/Phrasensuche
  ... where contains(text, 'Hund')>0
- Logische Kombinationen
  ... where contains(text, 'Hund AND Katze') >0
- Wildcard-Suche
  ... where contains(text, 'Hu%d AND Kat_e') >0
- Einfaches Fuzzy matching
  ... where contains(text, '?Hunt') >0
- Namenssuche
  ... where contains(text, 'NDATA(name, Hunt)') >0
- Multilinguale Stammsuche
  ... where contains(text, '$läuft') >0

Oracle TEXT: Abfragemöglichkeiten II

- NEAR-Operator
  ... where contains(text, 'near(Hund, Katze), 4') >0
- Suche in Sektionen, Sätzen und Paragraphen (XML)
  ... where contains(text, 'Hund WITHIN TITEL') >0
- ISO 2788 konformer Thesaurus
  ... where contains(text, 'SYN(Hund)' >0)

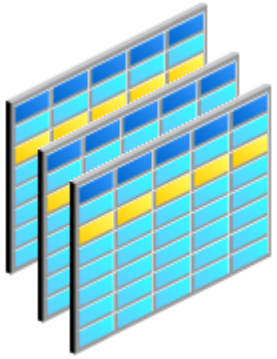
Abfragetechnik: Progressive Relaxation

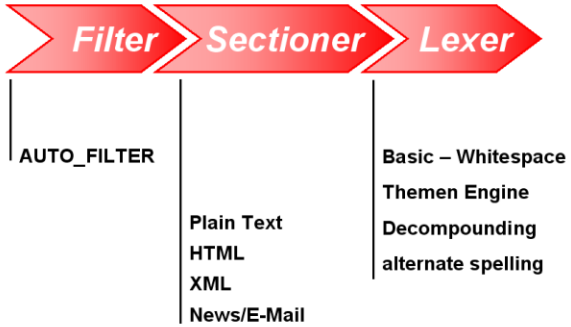
- Abfrage beginnt exakt und wird erweitert ...
  - ... immer unschärfer ...
  - ... bis die gewünschte Trefferzahl erreicht ist.

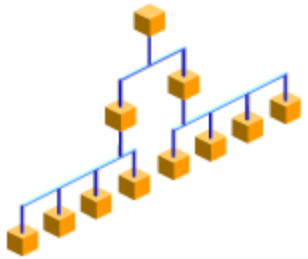
```
select score(1), title from test_table
where contains (
  dokument,
  '<query>
     <textquery>
       <progression>
         <seq>"Czarski Carsten"</seq>
         <seq>Czarski Carsten</seq>
         <seq>?Czarski ?Carsten</seq>
         <seq>NDATA(name, Czarski Carsten)</seq>
         :

and rownum <= 5
```

Wie nutzt man das für "normale" Tabellendaten?

- Möglich mit dem USER_DATASTORE
  - PL/SQL-Prozedur liefert die zu indizierenden "Dokumente"
  - Generierung "temporärer Dokumente" aus Tabellen







Tabelle(n)

PL/SQL-Prozedur

Index-Engine

Volltextindex

Die umfangreiche Grundausstattung der Datenbank

GÄNGIGE AUFGABEN SCHNELLER ERLEDIGEN

VARCHAR2 zusammenfassen: LISTAGG

- Neue Aggregatsfunktion für Zeichenketten
- Beispiel:

```
select
 deptno,
 listagg(ename, ':') within group (order by ename) ename_list
from emp
group by deptno

    DEPTNO ENAME_LIST
---------- ----------------------------------------
        10 CLARK:KING:MILLER
        20 ADAMS:FORD:JONES:SCOTT:SMITH
        30 ALLEN:BLAKE:JAMES:MARTIN:TURNER:WARD
```

Reguläre Ausdrücke in SQL und PL/SQL

- REGEXP_LIKE — Existiert das Muster?
- REGEXP_INSTR — Position des Musters?
- REGEXP_SUBSTR — Muster extrahieren
- REGEXP_REPLACE — Muster ersetzen
- REGEXP_COUNT — Wie oft existiert das Muster?

```
select * from message_table
where REGEXP_LIKE(message, 'ORA-.[0-9]{5}')

select REGEXP_SUBSTR(
  'Max Mustermann, 80992 München',
  '[[:digit:]]{5}')
from dual
```

Reguläre Ausdrücke: Weitere Beispiele

- Reduziert komplexe Logik auf eine Zeichenkette
- Email-Adresse:
  ^[a-z0-9\-_]?[a-z0-9.\-_]+[a-z0-9\-_]?@[a-z.-]+\.[a-z]{2,}$
- URL:
  ^http[s]?://[-a-zA-Z0-9_.:]+[-a-zA-Z0-9_:@&?=+,.!/~*'%$]*$
- KFZ-Kennzeichen
  ^[A-Z]{1,3}-[A-Z]{1,2} [0-9]{1,4}$
- Uhrzeitformat (24h)
  ^([01]?[0-9]|2[0123])[:]?([0-5][0-9])$

XML erzeugen: SQL/XML-Funktionen

- XMLElement()
- XMLForest()
- XMLAgg()
- XMLComment()
- XMLCDATA()
- XMLPI()
- XMLRoot()
- XMLSerialize()

XML erzeugen: Ein Beispiel

- SQL-Abfrage (kann auch als View erstellt werden)

```
select
  XMLElement("employee",
    XMLElement("ename",
      XMLAttributes(empno as "id", hiredate as "hire-date"
    ),
    XMLForest(
      ename as "name",
      job as "job",
      sal as "salary",
      comm as "commission"
    )
  ) as xml from emp
```

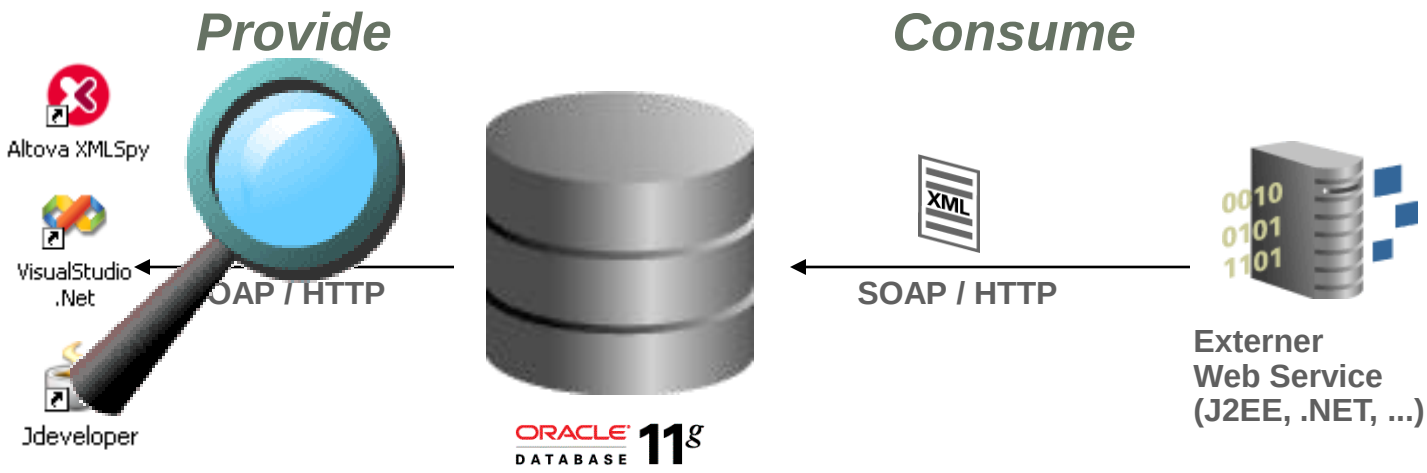
XML erzeugen: Ein Beispiel

- Ergebnis der Abfrage:

```
<employee>
  <ename id="7369" hire-date="1980-12-17">
    <name>SMITH</name>
    <job>CLERK</job>
    <salary>800</salary>
  </ename>
</employee>

<employee>
  <ename id="7499" hire-date="1981-02-20">
    <name>ALLEN</name>
    <job>SALESMAN</job>
    <salary>1600</salary>
    <commission>300</commission>
  </ename>
</employee>
```

Oracle 11g und Web Services

- Web Service Consumer
  - Ab Oracle10g: Database Web Services
- Web Service Provider
  - Ab Oracle11g: XML DB Native Web Services



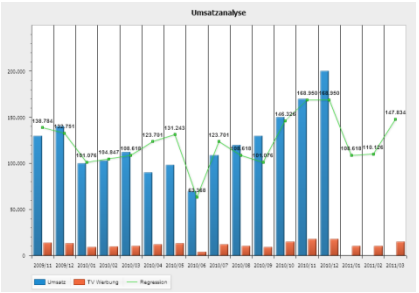
Oracle11g Native Web Services

- XML DB Protokollserver als HTTP-Endpoint
- Alle Datenbankobjekte
  - SQL-Abfragen, XQuery-Abfragen
  - PL/SQL-Prozeduren, -Funktionen und -Packages
- **Ohne** zusätzliche Entwicklung

Die umfangreiche Grundausstattung der Datenbank

DATEN EINFACHER AUSWERTEN

Analytische Funktionen: Das Konzept



- Bessere Abfrage-Performance
  - Berichte
  - OLAP
  - BI-Applikationen
- Enge Integration in den Datenbankkern
  - Verarbeitung ist effizient und skalierbar (Optimizer)
  - Code ist einfacher und flexibler
  - Reduzierte Ausführungen auf der Client-Side

Statistische SQL- und PL/SQL-Funktionen

- Ranking-Funktionen
  - **rank, dense_rank, cume_dist, percent_rank, ntile**
- Analytische Funktionen (Query Window)
  - **Avg, sum, min, max, count, variance, stddev, first_value, last_value**
- Analytische Funktionen (Aggregate)
  - **Sum, avg, min, max, variance, stddev, count, ratio_to_report**
- Interrow-Berechnungen
  - **Lag, lead**
- Statistische Aggregate
  - **Korrelation, Lineare Regression, Kovarianz**

Beispiel: Gleitender Durchschnitt

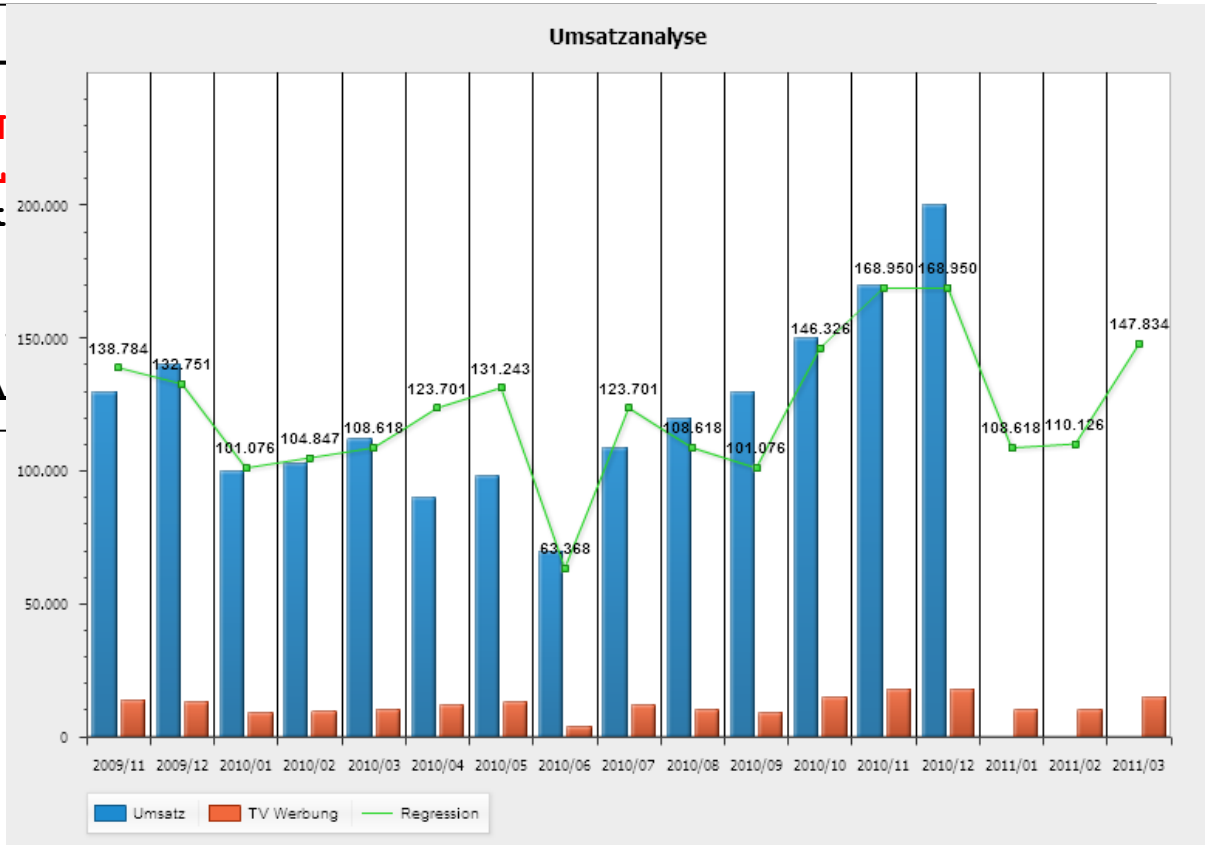
```
SELECT manager_id id, last_name, hire_date, salary, AVG(salary)
  OVER (
    PARTITION BY manager_id ORDER BY hire_date
    ROWS BETWEEN 1 PRECEDING AND 1 FOLLOWING
  ) s_avg
FROM employees;
```

ID	LAST_NAME	HIRE_DATE	SALARY	S_AVG
100	Kochhar	21-SEP-89	17000	17000
100	De Haan	13-JAN-93	17000	15000
100	Raphaely	07-DEC-94	11000	11966.6667
100	Kaufling	01-MAY-95	7900	
100	Hartstein	17-FEB-96	13000	
100	Weiss	18-JUL-96	8000	
100	Russell	01-OCT-96	14000	
100	Partners	05-JAN-97	13500	
100	Errazuriz	10-MAR-97	12000	
100	Fripp	10-APR-97	8200	
200				

Beispiel: Einfache lineare Regression

- Prognose "UMSATZ" anhand "TV-Werbung"

```
with regr_
 select
    REGR_IN
    REGR_SL
 from verk
)
select
   a + b *
from VERKA
```



Weitere Informationen

- APEX Community
  http://tinyurl.com/apexcommunity
- DBA Community
  http://tinyurl.com/dbacommunity

- Blog: SQL und PL/SQL auf Deutsch
  http://sql-plsql-de.blogspot.com
- Blog: Oracle TEXT auf Deutsch
  http://oracle-text-de.blogspot.com

Fragen & Antworten