

# Vertiefung

Abfragesprache SQL in ORACLE

# Literatur

## Oracle Online-Dokumentation – Database Administration

[http://docs.oracle.com/database/121/nav/portal\\_4.htm](http://docs.oracle.com/database/121/nav/portal_4.htm)

- [Database SQL Language Reference](#)
- [Database SQL Language Quick Reference](#)

# AGENDA

- 1 Data Dictionary**
- 2 DML – Vertiefung**
  - 2.1 Views
  - 2.2 Constraints

# 4. Data Dictionary

Primary-Key, Foreign-Key

# Data Dictionary

- Speichert **META-Daten** über Tabellen, Views etc. in der Datenbank. Darunter:
  - Beschreibungen angelegter Tabellen (Name, Spalten, Datentypen) und Views
  - Informationen über angelegte Constraints
  - Daten über Nutzer und ihnen zugewiesene Rechte
- Meta-Daten werden in Tabellen verwaltet und können so direkt oder über vorhandene Views abgefragt werden.

# Beispiele für Data Dictionary Abfragen

- Abfrage aller Tabellen, die vom aktuellen Nutzer angelegt wurden

```
SELECT  table_name  
FROM    user_tables;
```

- Abfrage aller Views, die vom aktuellen Nutzer angelegt wurden

```
SELECT  view_name, text  
FROM    user_views;
```

- Abfrage aller Constraints, die vom aktuellen Nutzer angelegt wurden

```
SELECT  constraint_name  
FROM    user_constraints;
```

# 3.1 Views

# SQL Views – Anlegen und Löschen

- View kann als virtuelle Tabelle angesehen werden
- Enthält keine Daten, sondern basiert auf einer gespeicherten SQL-Abfrage, die Daten aus einer oder mehreren Tabellen selektiert
- Ähnlich einer Tabelle kann eine Abfrage auf Views gemacht werden

```
CREATE (OR REPLACE) VIEW viewname AS  
SELECT-STATEMENT;
```

- SELECT-Statement wird bei jedem Zugriff auf die View ausgeführt

```
CREATE OR REPLACE VIEW v_emp_dept AS  
SELECT empno, ename, hiredate, d.deptno, dname  
FROM emp e  
INNER JOIN dept d ON (e.deptno = d.deptno);
```



# Views - Verwendung

```
SELECT ename, dname  
FROM v_emp_dept;
```

## V\_EMP\_DEPT

```
SELECT empno, ename, hiredate,  
       d.deptno, dname  
FROM emp e  
INNER JOIN dept d ON (e.deptno = d.deptno);
```

## Ergebnis

ENAME	DNAME
CLARK	ACCOUNTING
KING	ACCOUNTING
...	...

## Basistabellen der View EMP und DEPT

EMPNO	ENAME	...	DEPTNO
7782	CLARK	...	10
7839	KING	...	10
7369	SMITH	...	20

DEPTNO	DNAME	LOC
20	RESEARCH	DALLAS
30	SALES	CHICAGO

1. Erzeuge eine View, die folgendes Ergebnis repräsentiert, das nur Verkäufe vom 27.06.2015 umfasst:

VNR	VNAME	UNR	ANZAHL	ANR	ANAME	APREIS
4321	Jahred	1010	7	12	Stiefel	89,99
3401	Schmitz	1000	10	12	Stiefel	89,99
4321	Jahred	1011	15	11	Pullover	35,2
1010	Mueller	1006	40	11	Pullover	35,2
5337	Ritsch	1003	70	11	Pullover	35,2
4321	Jahred	1009	35	13	Wintermantel	123,8
1010	Mueller	1007	35	13	Wintermantel	123,8

2. Wie viele Verkäufe hat der Vertreter Mueller am 27.06.15 durchgeführt?
3. Wie viele Artikel hat der Vertreter Mueller am 27.06.15 verkauft?
4. Wie viele Artikel wurden durchschnittlich am 27.06.15 verkauft?
5. Welcher Artikel (anr und aname) wurde am 27.06.15 nicht verkauft?

## 3.2 Constraints

Primary-Key, Foreign-Key, Check

- Constraints (Bedingungen) können für Spalten einer Tabelle definiert werden
- Sie stellen Regeln dar, die beim Einfügen von Werten in diese Spalten eingehalten werden müssen
- Folgende Constraints können definiert werden
  - **NOT NULL – Constraint**
  - **Primary Key - Constraint**
  - **Foreign Key – Constraint**
  - UNIQUE – Constraint
  - Check - Constraint

# NOT NULL Constraint

- NOT NULL Constraint kann für eine Spalte einer Tabelle angelegt werden
- Verhindert, dass NULL-Werte in diese Spalte eingefügt werden
- Kann beim Anlegen der Tabelle erzeugt oder nachträglich hinzugefügt werden

## NOT NULL Constraint bei Anlegen der Tabelle

```
CREATE TABLE person
(
    nachname VARCHAR2(30) NOT NULL,
    geburstag DATE,
    gewicht NUMBER(4,1)
);
```

## NOT NULL Constraint nachträglich hinzufügen

```
ALTER TABLE person
MODIFY (nachname NOT NULL);
```

# Primary Key Constraint

Eine/mehrere Spalten können über den Primary Key Constraint als Primärschlüssel festgelegt werden

PNR	Vorname	Nachname	Geburtsdatum	Geschlecht
1	Peter	Kiel	01.01.1980	maennlich
2	Lisa	Lebe	02.02.1970	weiblich

## ► **Primärschlüssel besteht aus einer Spalte**

jeder Datensatz muss über den Wert in dieser Spalte eindeutig identifiziert werden können.

### **Beispiel – Primärschlüssel auf PNR:**

Jede PNR darf nur einmal vorhanden sein

## ► **Primärschlüssel besteht aus mehreren Spalten**

jeder Datensatz muss über die Wertkombination in diesen Spalten eindeutig identifiziert werden können

### **Beispiel – Primärschlüssel auf Vorname, Nachname, Geburtstag:**

Es dürfen keine zwei Personen mit identischem Namen und Geburtsdatum eingefügt werden

# Primary Key Constraint

PNR	Vorname	Nachname	Geburtsdatum	Geschlecht
1	Peter	Kiel	01.01.1980	maennlich
2	Lisa	Lebe	02.02.1970	weiblich

## ► Primärschlüssel bei Tabellenerzeugung anlegen

```
CREATE TABLE person
(
    pnr NUMBER(8,0),
    nachname VARCHAR2(30),
    geburtstag DATE,
    CONSTRAINT pkper PRIMARY KEY (pnr)
);
```

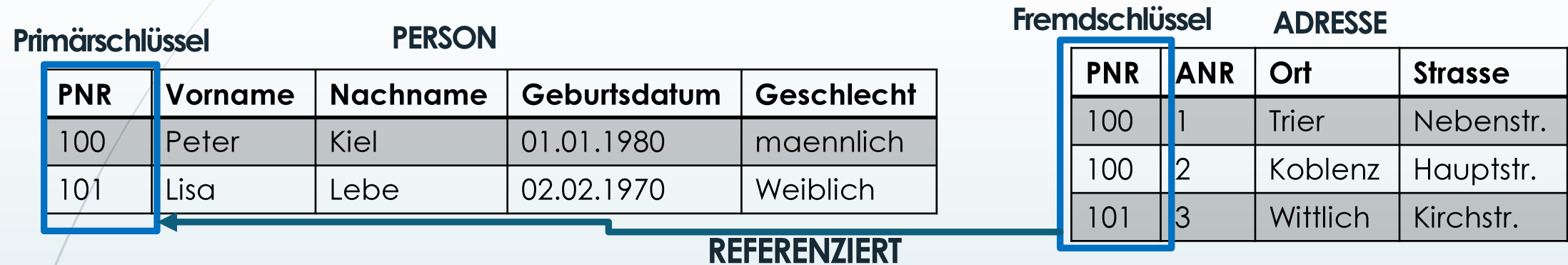
```
CREATE TABLE person
(
    pnr NUMBER(8,0) CONSTRAINT pkper PRIMARY KEY,
    nachname VARCHAR2(30),
    gewicht NUMBER(4,1)
);
```

## ► Primärschlüssel nach Tabellenerzeugung hinzufügen

```
ALTER TABLE person
ADD CONSTRAINT pkper PRIMARY KEY (pnr) ;
```

# Foreign Key Constraint

Fremdschlüssel-Spalte referenziert Primärschlüssel (bzw. Unique Key-Spalte) einer Tabelle



## Foreign Key beim Erzeugen der Tabelle anlegen

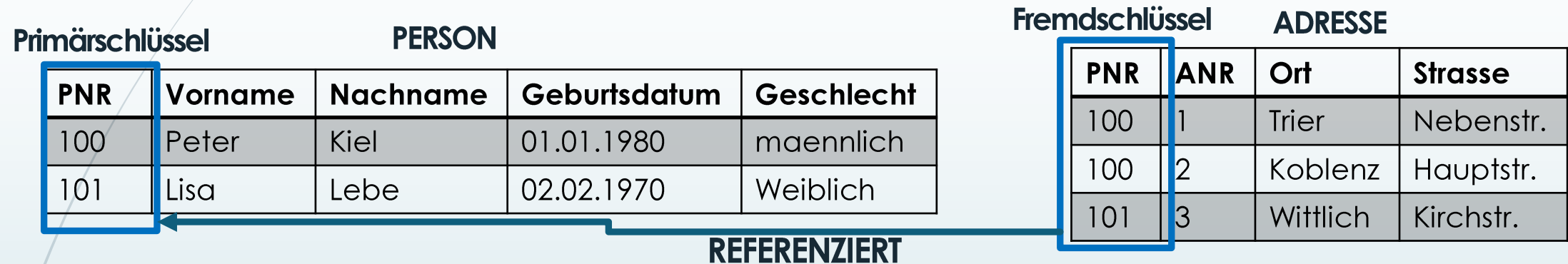
```
CREATE TABLE adresse
(
    pnr NUMBER(4,0) CONSTRAINT fkper REFERENCES PERSON(pnr) ,
    ... ,
);
```

```
CREATE TABLE adresse
(
    pnr NUMBER(4,0) ,
    ... ,
    CONSTRAINT fkper FOREIGN KEY(pnr) REFERENCES PERSON(pnr)
);
```



# Foreign Key Constraint

Fremdschlüssel-Spalte referenziert Primärschlüssel (bzw. Unique Key-Spalte) einer Tabelle



➔ Foreign Key nach Tabellenerzeugung hinzufügen

```
ALTER TABLE adresse  
ADD CONSTRAINT fkper FOREIGN KEY(pnr) REFERENCES person(pnr) ;
```

# Constraints löschen

```
ALTER TABLE tabellenname  
DROP CONSTRAINT constraintname;
```

- Mit folgendem Befehl wird der Primary Key Constraint **PKVERTRETER** gelöscht, der für die Tabelle **Vertreter** angelegt wurde.

```
ALTER TABLE vertreter  
DROP CONSTRAINT PKVERTRETER;
```

Führe das Skript dbVertreter\_ohneCons.sql aus.

1. Lege für die Tabellen Vertreter, Verkauf und Artikel Primary Key Constraints an.
2. Lege für die Tabelle Verkauf alle notwendigen Foreign Key Constraints an.
3. Lösche den Foreign Key Constraint FK\_DEPTNO. Dieser ist für die Spalte deptno in der Tabelle emp definiert.
4. Lege den benötigten Foreign Key Constraint für die Tabelle emp neu an.