# SQL/PL - Übungen und Lösungen

## Einfache SELECT-Abfragen

Schreiben Sie eine Abfrage, die alle Datensätze der Tabelle products von Lieferanten mit der Nummer 1 liefert.

-- L�sung:

select \* from products where SUPPLIERID=1;

Schreiben Sie eine Abfrage, die alle Datensätze der Tabelle products mit der Lieferanten\_Nummer = 1 und Kategorie\_Nummer = 2 liefert.

-- L�sung:

select \* from products where SUPPLIERID=2 and CATEGORYID=1;

Schreiben Sie eine Abfrage, die alle Datensätze der Tabelle products mit Preisen >= 50 und <= 100 anzeigt.

-- L�sung:

select \* from products where UNITPRICE between 50 and 100;

Schreiben Sie eine Abfrage, die alle Datensätze der Tabelle products mit Netto- und Bruttopreisen anzeigt.

-- L�sung:

select UNITPRICE as Netto, UNITPRICE\*1.16 as Brutto from products;

Schreiben Sie eine Abfrage, die alle Datensätze der Tabelle employees mit Vor- und Nachname, getrennt durch ein Lerrzeichen anzeigt.

-- L�sung:

select FIRSTNAME || ' ' || LASTNAME from employees;

select FIRSTNAME + ' ' + LASTNAME from employees;

select concat(FIRSTNAME , ' ' , LASTNAME) from employees;

Schreiben Sie eine Abfrage, die alle Datensätze der Tabelle employees mit Mitarbeitern aus London, Redmond oder Seattle liefert.

-- L�sung:

select LASTNAME from employees

where CITY in ('London','Redmond','Seattle')

order by 1 ASC

;

--- ORDER BY LASTNAME

Schreiben Sie eine Abfrage, die die Nachnamen aller Mitarbeiter aus den USA anzeigt. Die Nachnamen sollen absteigend sortiert werden.

-- L�sung:

select LASTNAME from employees where COUNTRY = 'USA' order by 1 desc;

Schreiben Sie eine Abfrage, die alle Produktnamen mit deren Kategorie-ID auflistet. Dabei soll jeweils folgender Satz ausgegeben werden:   
Das Produkt X gehört zur Kategorie Y.

-- L�sung:

select

'Das Produkt' || PRODUCTNAME || ' geh�rt zur Kategorie ' || CATEGORYID || '. '

from products;

## LIKE, Platzhalter

Erstellen Sie eine Abfrage, die Ihnen alle Produkte ausgibt, die mit "C" anfangen. Diese sollen absteigen nach dem Produktname sortiert sein.

-- L�sung:

select PRODUCTNAME from products

where PRODUCTNAME

like 'C%'

order by PRODUCTNAME desc

;

Erstellen Sie eine Abfrage, die Ihnen alle Produkte ausgibt, die mit "A", "B", oder "C" anfangen.

-- L�sung:

select PRODUCTNAME from products

where PRODUCTNAME

like 'A%' or

PRODUCTNAME like 'B%' or

PRODUCTNAME like 'C%'

;

Erstellen Sie eine Abfrage, die Ihnen alle Produkte ausgibt, bei denen der 5. Buchstabe ein "c" oder "d" ist. Diese sollen aufsteigend nach dem Produktname sortiert sein.

-- L�sung:

select PRODUCTNAME from products where PRODUCTNAME

like '\_\_\_\_c%' or PRODUCTNAME like '\_\_\_\_d%' order by 1

;

Erstellen Sie eine Abfrage, die Ihnen alle Produkte ausgibt, bei denen der 5. Buchstabe kein "c" ist.

-- L�sung:

select PRODUCTNAME from products where PRODUCTNAME not

like '\_\_\_\_c%'

;

## Gruppierungen

Schreiben Sie eine Abfrage, die Ihnen die Spalte City der Tabelle employees ausgibt. Es soll keine Stadt mehrmals auftauchen.

CITY

----------------

Seattle

Tacoma

Ta

-- L�sung:

select distinct city from employees;

select city from employees group by city order by city;

Schreiben Sie eine Abfrage, die Ihnen die Summe des Warenbestandes (unitsinstock) aus der Tabelle products, gruppiert nach den Lieferantennummern (supplierid) ausgibt.

SUPPLIERID SUMME

----------------

1 69

-- L�sung:

select supplierid,sum(unitsinstock) as Summe from products

group by supplierid

;

Schreiben Sie eine Abfrage, die Ihnen die Summe des Warenbestandes (unitsinstock) aus der Tabelle products, gruppiert nach den Lieferantennummern (supplierid) ausgibt! Es sollen nur die Lieferanten erscheinen, bei denen die Summe des warenbestandes größer als 70 ist.

SUPPLIERID SUMME

----------------

2 133

-- L�sung:

select supplierid,sum(unitsinstock) as Summe from products

group by supplierid having sum(unitsinstock) > 70

;

Schreiben Sie eine Abfrage, die Ihnen die Summe des Warenbestandes (unitsinstock) aus der Tabelle products, gruppiert nach den Lieferantennummern (supplierid) und den Kategorienummern (categoryid) ausgibt!

SUPPLIERID CATEGORYID SUM(UNITSINSTOCK)

----------------------------------------

1 1 56

1 2 13

2 2 133

3 2 126

-- L�sung:

select supplierid,categoryid,sum(unitsinstock) from products

group by supplierid,categoryid

;

Listen Sie alle Lieferanten auf, die mehr als 5 Produkte liefern!

SUPPLIERID

----------

7

12

-- L�sung:

select supplierid from products

group by supplierid having count( distinct Productname) > 5

;

Listen Sie die Lieferanten mit Ihren jeweils teuersten Produkt auf. Es sollen nur die Lieferanten angezeigt werden, bei denen das teuerste Produkt billiger als 10 DM ist.

SUPPLIERID MAX(UNITPRICE)

--------------------------

10 4,5

-- L�sung:

select supplierid,max(unitprice) from products

group by supplierid having max(unitprice) < 10

;

Zusatzfrage:   
Listen Sie den Nachnamen des Mitarbeiters (employees) auf, der am längsten angestellt war!

-- L�sung:

SELECT lastname,hiredate FROM employees

WHERE hiredate in

(

SELECT MIN(hiredate) FROM employees

)

;

~~select top (1) lastname,hiredate from employees order by hiredate;~~

;With cte (

SELECT MIN(hiredate) as Min\_Hire\_Date FROM employees

)

~~, cte1 as ()~~

~~, cte2 as ()~~

SELECT lastname,hiredate FROM employees

WHERE hiredate = (select Min\_Hire\_Date from cte)

Was machen folgende Abfragen?

select lastname from employees group by city;

-- L�sung:

-- Fehler -- Falsche Gruppierung.

select lastname from employees where city not in('Berlin','London');

-- L�sung:

-- Anzeige aller Angestellten, die nicht aus Berlin oder London kommen.

select discontinued,sum(unitprice\*unitsinstock) from products group by discontinued;

-- L�sung:

-- Anzeige des Warenbestandes im Warenlager.

select supplierid,sum(unitprice\*unitsinstock) from products group by supplierid;

-- L�sung:

-- Werte der Produkte jedes Lieferanten im Warenlager.

select categoryid,sum(unitprice\*unitsinstock) from products group by supplierid;

-- L�sung:

-- Fehler -- Falsche Gruppierung.

select sum(unitprice\*unitsinstock) from products group by supplierid;

-- L�sung:

-- Syntaktisch richtig aber unlogisch (sum).

select sum(unitprice\*unitsinstock) from products;

-- L�sung:

-- Anzeige des Warenbestandes im Warenlager.

## JOIN

Schreiben Sie eine Abfrage, die Ihnen die Kontaktnamen der Lieferanten ausgibt, die Produkte der Kategorie 1 oder 2 oder 3 liefern!

CONTACTNAME PRODUCTNAME

-------------------------------------------

Charlotte Cooper Chai

Charlotte Cooper Chang

Charlotte Cooper Aniseed Syrup

Shelley Burke Chef Anton's Cajun Seasoning

-- L�sung:

select s.CONTACTNAME,p.PRODUCTNAME from

products p,suppliers s

where p.CATEGORYID in ('1','2','3')  
AND p.SUPPLIERID = s.SupplierId

;

select s.CONTACTNAME,p.PRODUCTNAME from

products p INNER JOIN suppliers s   
ON p.SUPPLIERID = s.SupplierId

where p.CATEGORYID in ('1','2','3')

--- alle supplier ohne producte

select s.CONTACTNAME,p.PRODUCTNAME from

suppliers s LEFT JOIN products p   
ON p.SUPPLIERID = s.SupplierId

where p.PRODUCTNAME is null

;

select s.CONTACTNAME,p.PRODUCTNAME from

products p RIGHT JOIN suppliers s

ON p.SUPPLIERID = s.SupplierId

where p.PRODUCTNAME **is** null

;

Schreiben Sie eine Abfrage, die Ihnen die Kundennamen, die entsprechenden Bestellungnummern (orderid) und den Namen des Angestellten (employees), der die Bestellung bearbeitet hat, ausgibt. Es soll nach dem Contactname geordnet sein!

CONTACTNAME ORDERID LASTNAME

--------------------------------------------------

Alejandra Camino 10281 Peacock

Alejandra Camino 10282 Peacock

Alejandra Camino 10306 Davolio

Alejandra Camino 10917 Peacock

-- L�sung:

select c.CONTACTNAME,o.orderid,e.lastname from customers c, orders o, employees e

where e.employeeID = o.employeeID and o.customerid = c.customerid

;

select c.CONTACTNAME,o.orderid,e.lastname

from customers c INNER JOIN orders o

on o.customerid = c.customerid

INNER JOIN employees e

on e.employeeID = o.employeeID

;

Schreiben Sie eine Abfrage, die Ihnen alle Kundennamen ausgibt, die eine Bestellung aufgegeben haben, die Produkte beinhaltet, die von einem Lieferanten aus Berlin geliefert werden. Es soll absteigend nach den Kundennamen sortiert werden!

-- L�sung:

select c.contactname

from

customers c,orders o,order\_details od,products p,suppliers s

where

o.customerid = c.customerid and

od.orderid = o.orderid and

od.productid = p.productid and

p.supplierid = s.supplierid and

s.city='Berlin'

order by 1

;

Schreiben Sie eine Abfrage, die Ihnen alle Kundennamen ausgibt, die eine Bestellung aufgegeben haben, die von Speedy Express ausgeliefert wurde!

-- L�sung:

select distinct c.COMPANYNAME from customers c, orders o, shippers sh

where

c.customerid = o.customerid and

o.SHIPVIA = sh.shipperid and

sh.COMPANYNAME = 'Speedy Express'

;

Schreiben Sie eine Abfrage, die Ihnen die Städte ausgibt, in denen Kunden wohnen und die Städte darunter ausgibt, in denen Lieferanten wohnen. Jede Stadt soll nun einmal auftauchen!

CITY

-------

Aachen

Albuquerque

Anchorage

Ann Arbor

-- L�sung:

select distinct city from customers

union

select distinct city from suppliers;

Schreiben Sie eine Abfrage, die Ihnen die Städte ausgibt, in den sowohl Kunden als auch Lieferanten wohnen. Jede Stadt soll nur einmal auftauchen!

CITY

-------

Berlin

London

-- L�sung:

select distinct city from customers

intersect

select distinct city from suppliers;

Schreiben Sie eine Abfrage, die Ihnen die Städte ausgibt, in den Kunden, nicht aber Lieferanten, wohnen. Jede Stadt soll nun einmal auftauchen!

CITY

------

Aachen

Albuquerque

-- L�sung:

select distinct city from customers

minus --- except

select distinct city from suppliers;

Schreiben Sie eine Abfrage, die Ihnen die Kundennamen ausgibt, die noch nie etwas bestellt haben!

-- L�sung:

select distinct COMPANYNAME from customers

minus --- except

select distinct COMPANYNAME from customers c,orders o

where c.customerid = o.customerid

;

Schreiben Sie eine Abfrage, die Ihnen die Lieferanten ausgibt, die keine Produkte liefern!

-- L�sung:

select distinct s.companyname from suppliers s,products p

minus --except

select distinct s.companyname from suppliers s,products p

where s.supplierid = p.supplierid

;

Schreiben Sie eine Abfrage, die Ihnen die Angestellten und Vorgesetzten ausgibt. Die reportsto-Spalte gibt die employeeid des Vorgesetzten an, der wiederum auch selbst ein Angestellter ist!

-- L�sung:

select e.LASTNAME,e.FIRSTNAME, c.LASTNAME,c.FIRSTNAME from

employees e,employees c

where e.reportsto = c.employeeid(+)

;

Schreiben Sie eine Abfrage, die Ihnen die Kunden und Ihren Gesamtbestellwert (ohne Discount) auflistet.

CUSTO SUM(UNITPRICE

----------------

ALFKI 9192,4

ANATR 2805,9

ANTON 15030,7

AROUT 27613

BERGS 53936,3

BLAUS 4763,6

-- L�sung:

select c.customerid, sum(od.unitprice\*od.quantity) as Umsatz

from customers c

INNER JOIN orders o ON c.customerid = o.customerid

INNER JOIN order\_details od ON o.orderid = od.orderid

group by c.customerid

;

select c.customerid, sum(od.unitprice\*od.quantity) as Umsatz

from customers c, orders o, order\_details od

where c.customerid = o.customerid

and o.orderid = od.orderid

group by c.customerid

;

Wieviel Produkte hat ALFKI bestellt?

-- L�sung:

select count(\*) from orders o,order\_details od

where o.customerid = 'ALFKI' and o.orderid = od.orderid

;

Wie ist der Gesamtumsatz von ALFKI?

-- L�sung:

select to\_char(round(sum(od.UNITPRICE \* od.quantity),2),'9999D00') as Gesamtumsatz

from orders o, order\_details od

where o.customerid = 'ALFKI' and o.orderid = od.orderid

;

Was machen folgende Abfragen?

select \* from products pr,categories ca};

-- L�sung:

-- ??

select pr.productname, ca.categoryname

from products rp,categories ca

where pr.productid=ca.categoryid};

-- L�sung:

-- ??

select pr.productname, ca.categoryname

from products pr,categories ca

where pr.categoryid(+)=ca.categoryid};

-- L�sung:

-- ??

select sum(pr.unitprice\*pr.unitsinstock), ca.categoryname

from products pr,categories ca

where pr.productid=ca.categoryid group by ca.categoryname};

-- L�sung:

-- ??

select sum(pr.unitprice\*pr.unitsinstock), ca.categoryname

from products pr,categories ca group by ca.categoryname

where pr.productid=ca.categoryid};

-- L�sung:

-- ??

## Verschachtelte Abfragen

Wie heißt der älteste Mitarbeiter?

-- L�sung:

select lastname,firstname,birthdate

from employees where birthdate in

(select min(birthdate) from employees)

;

-- Peacock

Wie heißt das teuerste Produkt eines Lieferanten aus "Berlin"?

-- L�sung:

select PRODUCTNAME,unitprice from products p,suppliers s

where s.supplierid = p.supplierid and

unitprice =

(

select max(unitprice) from products p2

where s.supplierid = p2.supplierid and supplierid in

(

select supplierid from suppliers where upper(city) = 'BERLIN'

)

)

;

-- Schoggi Schokolade 43,90

Welche Produkte (productname) sind überdurchschnittlich teuer?

-- L�sung:

select productname,UNITPRICE from products

where UNITPRICE > (select avg(UNITPRICE) from products);

Wie heißt der jüngste Mitarbeiter?

-- L�sung:

select lastname,firstname,birthdate from employees

where birthdate in (select max(birthdate) from employees);

-- Dadworth ??

Zu welcher Kategorie gehört das teuerste Produkt (categoryname)

-- L�sung:

select CATEGORYID,unitprice from products

where unitprice =

(

select max(unitprice) from products

)

;

Schreiben Sie eine Abfrage, die Ihnen die Lieferanten (Companyname) und daneben das teuerste Produkt dieses Lieferanten (Productname) und daneben den Preis ausgibt.

-- L�sung:

SELECT s.companyname, p.productname, p.unitprice

FROM suppliers s, products p

WHERE p.supplierid = s.supplierid AND p.unitprice =

(

SELECT MAX(p2.unitprice)

FROM products p2, suppliers s2

WHERE p2.supplierid = s2.supplierid

AND s2.supplierid=s.supplierid

)

;

Welcher Kunde hat bisher am meisten bestellt?

-- L�sung:

select companyname, UMSATZ from

(

select c.companyname, sum(od.unitprice\*od.quantity)

as Umsatz from orders o,order\_details od, customers c

where o.orderid = od.orderid and

c.customerid = o.customerid group by c.companyname

)

where UMSATZ =

(

select max(UMSATZ) from

(

select c.companyname, sum(od.unitprice\*od.quantity) as Umsatz

from orders o,order\_details od, customers c

where o.orderid = od.orderid and

c.customerid = o.customerid

group by c.companyname

)

)

;

Erstellen Sie eine Abfrage, die Ihnen die Kunden (contactname) und die bisherigen Umsätze auflistet.

-- L�sung:

select c.contactname, sum(od.unitprice\*od.quantity) as Umsatz

from orders o,order\_details od, customers c

where o.orderid = od.orderid and

c.customerid = o.customerid

group by c.contactname

;

Welche Kunden haben bisher Chai bestellt?

-- L�sung Variante 1:

SELECT DISTINCT o.customerid

FROM orders o, order\_details od, products p

WHERE o.orderid = od.orderid

AND od.productid = p.productid

AND UPPER(p.productname) = 'CHAI'

;

-- L�sung Variante 2:

select distinct companyname from customers

where customerid in

(

select customerid from orders where orderid in

(

select orderid from order\_details where PRODUCTID =

(

select PRODUCTID from products where PRODUCTNAME = 'Chai'

)

)

)

;

Wie heißt der zweitälteste Mitarbeiter?

-- L�sung Variante 1:

select lastname,firstname,birthdate from employees

where birthdate in

(

select min(birthdate) from employees where birthdate >

(

select min(birthdate) from employees

)

)

;

-- L�sung Variante 2:

select lastname from employees where birthdate=

(

select min(birthdate)

from

(

select lastname, birthdate from employees

minus -- except

select lastname, birthdate from employees

where birthdate = (select min(birthdate) from employees)

)

);

Wie heißt der drittjüngste Mitarbeiter?

-- L�sung:

select lastname,firstname,birthdate from employees

where birthdate =

(

select max(birthdate) from employees where birthdate <

(

select max(birthdate) from employees where birthdate <

(

select max(birthdate) from employees

)

)

)

;

## Funktionen

Schreiben Sie eine Abfrage, die Ihnen alle lastname der Employees in Großbuchstaben ausgibt.

-- L�sung:

select upper(lastname) from employees;

Schreiben Sie eine Abfrage, die Ihnen alle lastnames der Employees in Kleinbuchstaben ausgibt.

-- L�sung:

select lower(lastname) from employees;

Schreiben Sie eine Abfrage, die Ihnen den ersten Buchstaben von lastnames der Employees in Großbuchstaben und den Rest in Kleinbuchstaben ausgibt.

-- L�sung:

select initcap(lastname) from employees;

Schreiben Sie eine Abfrage, die Ihnen den ersten und den letzten Buchstaben von lastnames der Employees in Großbuchstaben und den Rest in Kleinbuchstaben ausgibt.

-- L�sung:

select initcap(substr(lastname,1,length(lastname)-1)) ||

upper(substr(lastname,-1,1))

from employees;

Schreiben Sie eine Abfrage, die Ihnen alle Employees ausgibt, in deren lastname zwei l (also ll) vorkommen. Es soll dabei egal sein, ob es jeweils ein großes oder kleines l sind.

-- L�sung:

select lastname from employees where instr(lower(LASTNAME),'ll') > 0;

Schreiben Sie eine Abfrage, die Ihnen eine Spalte als Ergebnismenge zurückgibt, in welcher der erste Buchstaben des Vornamens, dahinter ein Punkt und dahinter, getrennt durch ein Leerzeichen, der lastname ausgegeben wird. Bei dem lastname sollte lediglich der erste Buchstabe groß sein. (z.B. R. King)

-- L�sung:

select upper(substr(firstname,1,1)) || '. ' || initcap(lastname)

from employees;

Beantworten Sie folgende Prüfungsfrage:   
Which SELECT statement will get the result 'elloworld' fromt the string 'Hello World'?

SELECT SUBSTR ('HelloWorld',1) FROM dual;

SELECT INITCAP(TRIM('HellowWorld', 1,1) FROM dual;

SELECT LOWER(SUBSTR ('HellowWorld', 2,1) FROM dual;

SELECT LOWER(SUBSTR('HellowWorld', 2,1) FROM dual;

SELECT LOWER(TRIM ('H' FROM 'HelloWorld')) FROM dual;

-- L�sung:

Die letzte Abfrage ist richtig.

Ermitteln Sie alle Sonntagskinder aus der Tabelle employees. Zurückgegeben werden soll der lastname und das Geburtsjahr.

-- L�sung:

select lastname, birthdate, to\_char(birthdate,'fmDAY')

from employees where to\_char(birthdate,'fmDAY') = 'SONNTAG'

;

Ermitteln Sie das Datum, an dem die employees ihr 25-jähriges Firmenzugehörigkeitsjubiläum feiern können. Folgendermaßen soll das Eregbnis dargestellt werden:

LASTNAME Jubilee

-------------------- -----------------

Davolio 01.Mai 2017

Fuller 14.August 2017

Leverling 01.April 2017

Peacock 03.Mai 2018

Buchanan 17.Oktober 2018

Suyama 17.Oktober 2018

-- L�sung:

select LASTNAME,

to\_char(add\_months(hiredate,25\*12),'dd.Month YYYY')

as Jubilee

from employees

;

Ermitteln Sie den Mitarbeiter, der am längsten angestellt war.

-- L�sung:

select lastname, hiredate from employees

where hiredate in

(select min(hiredate) from employees)

;

Ermitteln Sie den Mitarbeiter, der am ältesten war, als er eingestellt wurde.

-- L�sung:

select lastname, birthdate,hiredate, months\_between(hiredate,birthdate)

from employees

where months\_between(hiredate,birthdate) =

(select max(months\_between(hiredate,birthdate)) from employees)

;

Ermitteln Sie die Mitarbeiter, die gegenwärtig länger als 10 Jahre in der Firma arbeiten

-- L�sung:

select lastname, hiredate, months\_between(sysdate,hiredate)/12 as years

from employees

where months\_between(sysdate,hiredate)/12 > 10

;

Schreiben Sie ein Abfrage, die den Produktpreis mit jeweils zwei Nachkommastellen ausgibt.

-- L�sung:

select productname,to\_char(unitprice,'99999D00') from products;

## CREATE, INSERT, MODIFY

Erstellen Sie folgende Tabellen (Legen Sie alle Anweisungen als Skript in einer Datei ab).

-- Tabelle Lehrer

Anrede Name Vorname

------------------

Frau Janker Hanni

Herr Hubert Jan

Frau Hinze Jutta

Herr Hub Frank

Herr Kreis Klaus

-- L�sung:

create table lehrer (

anrede varchar(40) not null,

name varchar(40) not null,

vorname varchar(40) not null

);

insert into lehrer values ('Frau','Janker','Hanni');

insert into lehrer values ('Herr','Hubert','Jan');

insert into lehrer values ('Frau','Hinze','Jutta');

insert into lehrer values ('Herr','Hub','Frank');

insert into lehrer values ('Herr','Kreis','Klaus');

select \* from lehrer;

Fügen Sie zur Tabelle eine weitere Spalte namens Gehalt (Typ: int) hinzu.

-- L�sung:

alter table lehrer add (gehalt int);

Vergrößern Sie die Länge des Datentypes für die Spalte Name um 5.

-- L�sung:

alter table lehrer modify (name varchar(45));

Setzen Sie das Gehalt für alle Herren auf 3000.

-- L�sung:

UPDATE lehrer SET gehalt = 3000 where ANREDE = 'Herr';

commit;

Setzen Sie das Gehalt für alle Frauen auf 3100.

-- L�sung:

UPDATE lehrer SET gehalt = 3100 where ANREDE = 'Frau';

commit;

Erstellen und füllen Sie eine Tabelle namens Herren (Name, Vorname, Gehalt).

-- L�sung:

create table herren as select name,vorname,gehalt from lehrer where ANREDE = 'Herr';

Löschen Sie aus der Tabelle Lehrer alle Herren.

-- L�sung:

delete from lehrer where ANREDE = 'Herr';

commit;

Benennen Sie die Tabelle Lehrer in Frauen um.

-- L�sung:

rename lehrer to frauen;

## Contraints

Erstellen Sie alle Beziehung der Northwind-Datenbank. Speichern Sie die Anweisungen in einer Datei!

-- L�sung:

---Primary Keys:

ALTER TABLE employees

ADD CONSTRAINT empl\_pk PRIMARY KEY (employeeid);

ALTER TABLE employeeterritories

ADD CONSTRAINT emplterr\_pk PRIMARY KEY (employeeid,territoryid);

ALTER TABLE territories

ADD CONSTRAINT terr\_pk PRIMARY KEY (territoryid);

ALTER TABLE region

ADD CONSTRAINT region\_pk PRIMARY KEY (regionid);

ALTER TABLE customers

ADD CONSTRAINT customers\_pk PRIMARY KEY (customerid);

ALTER TABLE orders

ADD CONSTRAINT orders\_pk PRIMARY KEY (orderid);

ALTER TABLE order\_details

ADD CONSTRAINT order\_details\_pk PRIMARY KEY (orderid,productid);

ALTER TABLE products

ADD CONSTRAINT products\_pk PRIMARY KEY (productid);

ALTER TABLE categories

ADD CONSTRAINT categories\_pk PRIMARY KEY (categoryid);

ALTER TABLE suppliers

ADD CONSTRAINT suppliers\_pk PRIMARY KEY (supplierid);

ALTER TABLE shippers

ADD CONSTRAINT shippers\_pk PRIMARY KEY (shipperid);

-- Foreign Keys:

ALTER TABLE products

ADD CONSTRAINT prod\_fk\_cat

FOREIGN KEY (categoryid)

REFERENCES categories(categoryid);

ALTER TABLE products

ADD CONSTRAINT prod\_fk\_supp

FOREIGN KEY (supplierid)

REFERENCES suppliers(supplierid);

ALTER TABLE order\_details

ADD CONSTRAINT orderd\_fk\_prod

FOREIGN KEY (productid)

REFERENCES products(productid);

ALTER TABLE order\_details

ADD CONSTRAINT orderd\_fk\_orders

FOREIGN KEY (orderid)

REFERENCES orders(orderid);

ALTER TABLE orders

ADD CONSTRAINT orders\_fk\_shippers

FOREIGN KEY (shipvia)

REFERENCES shippers(shipperid);

ALTER TABLE orders

ADD CONSTRAINT orders\_fk\_customers

FOREIGN KEY (customerid)

REFERENCES customers(customerid);

ALTER TABLE employeeterritories

ADD CONSTRAINT emplterr\_fk\_employ

FOREIGN KEY (employeeid)

REFERENCES employees(employeeid);

ALTER TABLE employeeterritories

ADD CONSTRAINT emplterr\_fk\_terr

FOREIGN KEY (territoryid)

REFERENCES territories(territoryid);

ALTER TABLE territories

ADD CONSTRAINT terr\_fk\_region

FOREIGN KEY (regionid)

REFERENCES region(regionid);

ALTER TABLE employees

ADD CONSTRAINT employ\_fk\_employ

FOREIGN KEY (reportsto)

REFERENCES employees(employeeid);

Stellen Sie folgendes sicher (Speichern Sie auch diese Anweisungen in einer Datei): Der unitprice muss größer oder gleich 0 sein (products und order\_details).

-- L�sung:

ALTER TABLE products ADD CONSTRAINT prod\_1\_CK1 CHECK (unitprice >= 0);

ALTER TABLE order\_details ADD CONSTRAINT orderd\_1\_CK1 CHECK (unitprice >= 0);

Das Land der Kunden darf nicht Irak oder Nordkorea oder Kuba sein.

-- L�sung:

ALTER TABLE customers

ADD CONSTRAINT custmom\_CK1

CHECK (lower(city) NOT IN ('iraq','north korea','north corea','cuba'))

;

Das Shippeddate darf nicht vor dem Orderdate liegen (Tabelle orders).

-- L�sung:

ALTER TABLE orders

ADD CONSTRAINT orders2ship\_CK1

CHECK (shippeddate >= orderdate)

;

Das Requireddate darf nicht vor dem Orderdate liegen (Tabelle orders).

-- L�sung:

ALTER TABLE orders

ADD CONSTRAINT req2order\_CK1

CHECK (requireddate >= orderdate)

;

Kontrolle:

select

TABLE\_NAME,

CONSTRAINT\_NAME,

SEARCH\_CONDITION,

R\_CONSTRAINT\_NAME,

INDEX\_NAME

from user\_constraints

where user = 'SYS'

and last\_change > sysdate-1

;

Abfrage: Liefert mehrfach vorkommende Werte in employees:

-- L�sung:

select EMPLOYEEID from employees minus select distinct EMPLOYEEID from employees;

select customerid from orders minus select distinct customerid from orders;

## PL/SQL

Erstellen Sie eine Prozedur New\_Empl, mit der Sie in die Tabelle employees neue Mitarbeiter hinzufügen können. Die Mitarbeiternummer des neuen Mitarbeiters muss um 1 größer sein als die größte Mitarbeiternummer. Aufgerufen werden soll die Prozedur mit exec new\_empl('Schroeder','Gerd').

-- L�sung:

/\*

set serveroutput on;

create or replace procedure New\_Empl(nachname varchar2,vorname varchar2) is

current\_table varchar2(30) := 'sys.employees';

max\_employeeid int := 0;

current\_employeeid int := 0;

begin

select max(employeeid) into max\_employeeid from employees;

current\_employeeid := max\_employeeid+1;

insert into employees (employeeid,lastname,firstname)

values (current\_employeeid,nachname,vorname);

commit;

Exception

when others then

dbms\_output.put\_line('Error');

end;

/

show errors;

exec new\_empl('Schroeder','Gerd')

Erstellen Sie eine Prozedur Prim\_Int, welche die Primzahlen zwischen in einem bestimmten Intervall ausgibt. Aufgerufen werden soll die Prozedur mit   
exec Prim\_Int(10,20)   
Hilfe: select mod(11,4) from dual;

-- L�sung Variante 1:

create or replace procedure Prim\_INT(unten int, oben int) is

temp int:=1;

begin

If (oben < unten) OR (oben < 1) OR (unten < 1) then

dbms\_output.put\_line('witzig');

else

for x in unten .. oben loop

temp:=1;

for y in 2 .. (x-1) loop

If mod(x,y) = 0 then

temp:=0;

End If;

end loop;

If temp=1 then

dbms\_output.put\_line(x);

End If;

end loop;

end if;

Exception

when others then

dbms\_output.put\_line('unbekannte Ausnahmeverletzung');

end;

/

show errors;

exec Prim\_Int(5,10);

-- L�sung Variante 2:

set serveroutput on;

create or replace procedure Prim\_Int(first\_num int,sec\_num int) is

zahl number(5):=0;

n number(5):=0;

prim number(5):=1;

begin

dbms\_output.put\_line('.');

dbms\_output.put\_line('Zwischen '||first\_num||' und '||sec\_num||' gibt es folgende Primzahlen:');

for x in first\_num .. sec\_num loop

for n in 2 .. (trunc(x/2)) loop

select mod(x,n) into zahl from dual;

if zahl > 0

then

prim:=1;

zahl:=0;

else

prim:=0;

end if;

end loop;

if prim = 1

then dbms\_output.put\_line(x);

end if;

end loop;

Exception

when too\_many\_rows then

dbms\_output.put\_line('Sie haben zu viele Werte');

when others then

dbms\_output.put\_line('unbekannte Ausnahmeverletzung');

end;

/

show errors;

Zusatzaufgabe 1:   
Wandeln Sie die Prozedur Prim\_Int um, indem Sie andere Schleifenkonstrukte verwenden.

-- L�sung:

??

Zusatzaufgabe 2:   
Verbessern Sie die Gültigkeitsprüfung bei der Parameterübergabe.

-- L�sung:

??

## Neues von Oracle 9i

Welche Kunden (contactname) haben Produkte der Kategorie beverages bestellt. Benutzen Sie NATURAL JOINs, soweit es möglich ist.

-- L�sung:

select distinct contactname from customers

NATURAL JOIN orders

NATURAL JOIN order\_details

NATURAL JOIN products

NATURAL JOIN categories

where lower(categoryname) = 'beverages'

;

Welcher Kunde hat noch nie etwas bestellt? Benutzen Sie einen OUTER JOIN.

-- L�sung:

select distinct contactname, orderid from customers c

left join orders o on c.customerid = o.customerid

where orderid is null

;

Welche Kunden wurden bisher von Speedy Express beliefert? Benutzen Sie INNER JOIN und NATURAL JOIN, soweit es möglich ist.

-- L�sung:

create view shippers\_view as

select shipperid as shipvia, companyname as sh\_companyname from shippers

;

select distinct companyname, sh\_companyname from customers

NATURAL JOIN orders

NATURAL JOIN shippers\_view

where lower(sh\_companyname) = 'speedy express'

;

drop view shippers\_view

;

Erstellen Sie eine Abfrage, die Ihnen rechts den Angestellten und links den Vorgesetzten ausgibt. Der Chef sollte in der Spalte Vorgesetzter BOSS zu stehen haben (Tipp: NVL).

-- L�sung:

select e.lastname Mitarbeiter,nvl(boss.lastname,'Boss') as Chef

from employees e

left join employees boss on e.reportsto = boss.employeeid

;

Erstellen Sie eine Abfrage, welche die Durchschnittskommission von scott.emp ermittelt. Ein Null-Wert sollte hierbei als 0 interpretiert werden und demnach in der Durchschnittsberechnung mit einbezogen werden

-- L�sung:

select avg(nvl(COMM,0)) from scott.emp;

Schreiben Sie eine Abfrage, welche die Mehrwertsteuer der Produkte berechnet. Produkte der Kategorie 1 haben die Mehrwertsteuer 0%, der Kategorie 2 und 3 Mehrwertsteuer 7% und der Rest 16%.

-- L�sung Variante 1:

SELECT productname,

CASE

WHEN p.categoryid = 1 THEN 0

WHEN p.categoryid IN (2,3) THEN unitprice\*0.07

ELSE unitprice\* 0.16

END AS MwSt

FROM products p

;

-- L�sung Variante 2:

select productname, to\_char(unitprice, '999g999d99'),

to\_char(unitprice\*

(

case categoryid

when 1 then 0

when 2 then 0.07

when 3 then 0.07

else 0.16

end

), '999g999d99') as MWST

from products natural join categories

;

## Abschlussübung

Sie haben den Auftrag erhalten, für eine Schule eine Datenbank zu erstellen. Mit Hilfe dieser Datenbank soll es möglich sein, die Lehrer, Schüler, Räume, Fächer und Klassen zu verwalten. Es soll auch ein Stundenplan möglich sein. Weiterhin soll es auch möglich sein, einen Raumbelegungsplan einschließlich eines Stundenplanes für die Lehrer zu erstellen.

1. Überlegen Sie sich eine Struktur für diese Datenbank (2 Stunden) - mit anschließender Präsentation und Diskussion.
2. Setzen Sie diese Struktur in die Praxis um.
3. Falls Sie noch Zeit haben, schreiben Sie ein kleines Front-End mit einen Programm Ihrer Wahl.

-- L�sung:

create user schule identified by schule;

grant dba to schule;

connect schule/schule;

-- TABLEs

Create table Klasse (

Klassen\_id number(5) constraint kl\_pk primary Key,

Klassenname varchar2(10) not null);

Create table Fach (

Fach\_id number(5) constraint fa\_pk primary Key,

Fachname varchar2(10) not null,

Fach\_kurz varchar2(5));

Create table Schueler (

Schueler\_id number(5) constraint sch\_pk primary Key,

Nachname varchar2(20) not null,

Vorname varchar2(20) not null,

Geb\_Datum date,

Geschlecht varchar2(10) not null,

Bemerkung varchar2(80),

Klassen\_id number(5) not null

constraint kl\_fk references klasse(klassen\_id));

Create table Lehrer (

Lehrer\_id number(5)

constraint le\_pk primary Key,

Nachname varchar2(20) not null,

Vorname varchar2(20) not null,

Geb\_Datum date);

Create table Lehrer\_Fach (

Fach\_Lehrer\_id number(5) constraint le\_fa\_pk primary Key,

Fach\_id number(5) not null

constraint fa\_fk references Fach(Fach\_id),

Lehrer\_id number(5) not null

constraint le\_fk references Lehrer(Lehrer\_id));

Create table Klassenraum (

Raum\_id number(5) constraint ra\_pk primary Key,

Raumbezeichnung varchar2(20) not null,

Besonderes varchar2(80));

Create table Wochentag (

Tag\_id number(5) constraint wo\_pk primary Key,

Tag varchar2(10) not null);

Create table Stunden (

Stunden\_id number(5) constraint std\_pk primary Key,

Stundennummer varchar2(10) not null);

Create table Stundenplan (

Plan\_id number(5) constraint srdpl\_pk primary Key,

Stunden\_id number(5) not null

constraint std\_fk references Stunden(Stunden\_id),

Tag\_id number(5) not null

constraint ta\_fk references Wochentag(Tag\_id),

Raum\_id number(5) not null

constraint ra\_fk references Klassenraum(Raum\_id),

Klassen\_id number(5) not null

constraint kl2\_fk references Klasse(Klassen\_id),

Fach\_Lehrer\_id number(5) not null

constraint fa\_le\_fk references Lehrer\_Fach(Fach\_Lehrer\_id)

);

commit;

-- CONSTRAINTs

ALTER TABLE stundenplan ADD

CONSTRAINT uk\_T\_S\_L\_sp UNIQUE (Tag\_id,Stunden\_id,Fach\_lehrer\_id);

ALTER TABLE stundenplan ADD

CONSTRAINT uk\_T\_S\_R\_sp UNIQUE (Tag\_id,Stunden\_id,Raum\_id);

ALTER TABLE stundenplan ADD

CONSTRAINT uk\_T\_S\_K\_sp UNIQUE (Tag\_id,Stunden\_id,Klassen\_id);

-- alter table stundenplan drop constraint uk\_R\_K\_L\_sp ;

-- views

-- Lehrer

create or replace view V\_Lehrer as

select l.nachname Nachname,

w.tag Tag,

s.stundennummer Stunde,

f.Fachname Fach,

r.raumbezeichnung Raum,

k.klassenname

from Stundenplan sp,

Lehrer l,

wochentag w,

stunden s,

fach f,

klassenraum r,

klasse k,

Lehrer\_fach lf

where sp.fach\_lehrer\_id = lf.fach\_lehrer\_id and

lf.lehrer\_id = l.lehrer\_id and

sp.tag\_id=w.tag\_id and

sp.stunden\_id=s.stunden\_id and

lf.fach\_id=f.fach\_id and

sp.raum\_id=r.raum\_id and

sp.klassen\_id=k.klassen\_id

order by l.nachname,w.tag\_id,s.stunden\_id;

-- Klassen

create or replace view V\_Klasse as

select k.klassenname,

w.tag Tag,

s.stundennummer Stunde,

r.raumbezeichnung Raum,

f.Fach\_Kurz Fach,

l.nachname lehrer

from Stundenplan sp,

Lehrer l,

wochentag w,

stunden s,

fach f,

klassenraum r,

klasse k,

Lehrer\_fach lf

where sp.fach\_lehrer\_id = lf.fach\_lehrer\_id and

lf.lehrer\_id=l.lehrer\_id and

sp.tag\_id=w.tag\_id and

sp.stunden\_id=s.stunden\_id and

lf.fach\_id=f.fach\_id and

sp.raum\_id=r.raum\_id and

sp.klassen\_id=k.klassen\_id

order by k.klassenname,w.tag\_id,s.stunden\_id;

-- Raumbelegung

create or replace view V\_Raum as

select r.raumbezeichnung Raum,

w.tag Tag,

s.stundennummer Stunde,

k.klassenname,

f.Fach\_Kurz Fach,

l.nachname lehrer

from Stundenplan sp,Lehrer l,wochentag w,stunden s,fach f,klassenraum r,klasse k,

Lehrer\_fach lf

where sp.fach\_lehrer\_id=lf.fach\_lehrer\_id and lf.lehrer\_id=l.lehrer\_id and

sp.tag\_id=w.tag\_id and

sp.stunden\_id=s.stunden\_id and

lf.fach\_id=f.fach\_id and

sp.raum\_id=r.raum\_id and

sp.klassen\_id=k.klassen\_id

order by r.raumbezeichnung,w.tag\_id,s.stunden\_id;

#### Oracle-Datenbank-Client mit M$-Access

Eine Bastellösung für Verspielte.

Ein Einloggen ist als 'sys as sysdba' ist nicht möglich. Deshalb ist ein entsprechender User anzulegen.   
Access starten, Leere DB, OK, Dateiname vergeben (\*.mdb) und speichern.   
Menü Datei | Externe Daten | Tabellen verknüpfen, Dateityp: ODBC   
Neu, Oracle in OraHome92, Weiter, Name der Datenquelle: testdb29, Fertigstellen   
Username: schule   
Password: schule   
Servicename: testdb29   
Tabellen verknüpfen   
Tabelle SCHULE.STUNDENPLAN auswählen. OK   
SCHULE.STUNDENPLAN in der Liste anklicken zum editieren.   
Menü Datein | Speichern. (commit wird automatisch ausgeführt.)