Felix Pojtinger

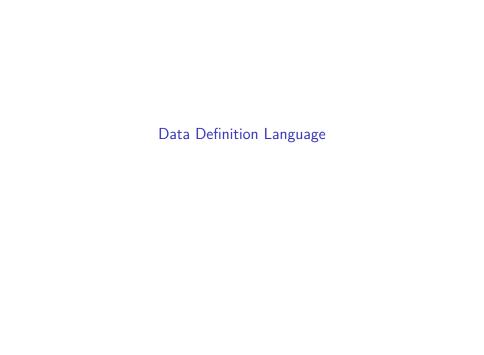
June 26, 2021

Data Definition Language

Data Manipulation Language

"Come, let us go down and confuse their language so they will not understand each other" - Genesis 11:7, Die Bibel

Mehr Details unter https://github.com/pojntfx/uni-db1-notes. Dieses Dokument ist nur als Schnell-Übersicht gedacht.



# **Tabellen**

```
Tabelle erstellen
create table persons (
    person_id number primary key not null ,
    first name varchar2(50),
    last name varchar2(50) default 'Duck' not null
);
Tabelle löschen
drop table persons;
Tabelle umbenennen
alter table persons rename to people;
```

# Spalten

```
Spalten hinzufügen
alter table persons add (phone varchar2(20), email varchar

Spalten bearbeiten
alter table persons modify (birthdate date null, email var

Spalten löschen
alter table persons drop column birthdate;
```

### Constraints

# Constraints hinzufügen

alter table purchase\_orders add constraint purchase\_orders

#### Constraints löschen

alter table purchase\_orders drop constraint purchase\_orders

#### Views

Views löschen

```
Views erstellen
create view employees_years_of_service
as select
   employee_id, first_name || ' ' || last_name as full_name
   floor(months_between(current_date, hire_date) / 12) as
from employees;
```

drop view employees\_years\_of\_service;

# **Indizes**

#### Indizes erstellen

create index members\_full\_name on members(first\_name, last\_

#### Indizes löschen

drop index members\_full\_name;

```
Trigger
   Trigger erstellen
   create trigger customers_credit_trigger
       before update of credit_limit
       on customers
   declare
       current day number;
   begin
       current day := extract(day from sysdate);
       if current day between 28 and 31 then
            raise_application_error(-20100, 'Locked at the end
       end if;
   end:
   Trigger löschen
   drop trigger customers_credit_trigger;
   Exceptions handlen
```

create trigger users ensure trigger

### **Functions**

drop function get\_my\_sum;

```
Function erstellen
create or replace function get_my_sum( a integer, b integer
is
        multiplier number := 2;
begin
        return a + b * multiplier;
end;
Function callen
select get my sum(1, 2) from dual;
Function löschen
```

### Procedure

```
Procedure erstellen
create or replace procedure get_sum ( a integer, b integer
is
        multiplier number := 2;
        result number := 0;
begin
        result := a + b * multiplier;
        insert into results ( result ) values ( result );
end:
Procedure callen
exec get_sum(1, 2);
Procedure löschen
drop procedure get_sum;
```



# Datentypen

- ► CHAR | CHARACTER (size)
- ► VARCHAR2 (size)
- DATE
- INTERVAL YEAR TO MONTH
- INTERVAL DAY TO SECOND
- ► INTEGER | INT
- ▶ NUMBER (precision [, scale ])
- ► FLOAT (precision)

```
Zeilenoperationen
   Insert
   insert into discounts(
        discount_name,
        amount,
        start date,
        expired_date
   ) values (
        'Summer Promotion',
        9.5.
        date '2017-05-01',
        date '2017-08-31'
   Update
   update products
   set list_price = 420
   where list_price < 69;</pre>
    Delete
```

### Unions

Gleiche Anzahl von Spalten, mehr Zeilen.

```
select
    first_name,
    last_name,
    email,
    'contact' as role
from contacts
union select
    first name,
    last_name,
    email,
    'employee' as role
from employees order by role
```

# Joins Mehr Spalten & mehr Zeilen Inner Join select a.id as id\_a, a.color as color a, b.id as id b, b.color as color\_b from palette a a inner join palette b b using(color); Left Outer Join select a.id as id a, a.color as color\_a, b.id as id b, b.color as color\_b from palette\_a a left outer join palette\_b b using(color);

```
Trigger
   Insert-Trigger
   :old ist nicht vorhanden.
   create or replace trigger customers_credit_trigger
       before insert of credit limit
       on customers
   declare
       current day number;
   begin
       current day := extract(day from sysdate);
       if current_day between 28 and 31 then
           raise_application_error(-20100, 'Locked at the end
       end if;
   end:
   Update-Trigger
   create or replace trigger customers_credit_limit_trigger
       before update of credit_limit
```

### Ort der Verdammnis

#### menhir

Wenn einem der Syntax schon nicht kompliziert genug ist, dann darf man vor das declare-Statement eines Triggers auch noch folgendes sinnloses Konstrukt packen und statt :new :neu schreiben:

referencing new as neu old as alt

Danach hat man auch fünf Zeilen. Und fünf Hirnzellen weniger.

Wo wir schon dabei sind: Ist der sonst universelle Negations-Operator != zu einfach? Zu simpel und zu verständlich? Wie wäre es mit <>; macht das genau selbe, ist aber komplizierter<sup>TM</sup>!