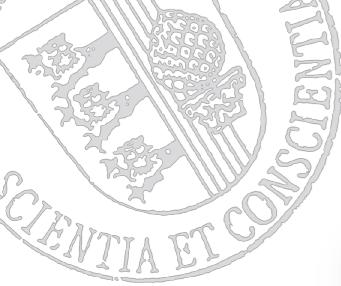


Hinweise

- Dieses Skript beinhaltet evtl. Fehler, die von mir gewollt sind.
- Vermutlich gibt es in diesem Skript auch Fehler, die nicht von mir gewollt waren.
- Manche Folien / Beispiele sind unvollständig. Dies ist Absicht.
- Die Lösungen zu den Beispielen werden in der Vorlesung besprochen.

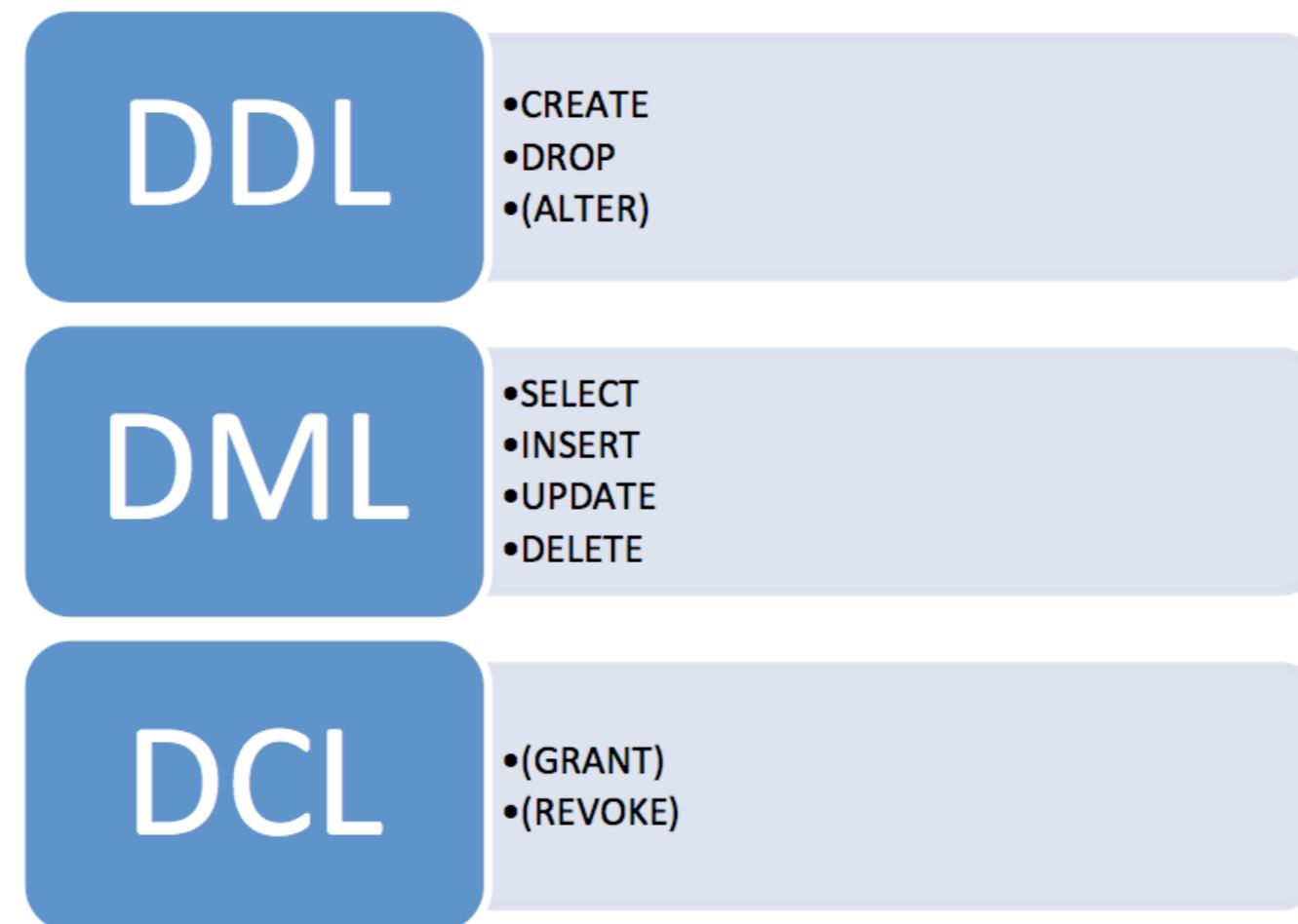


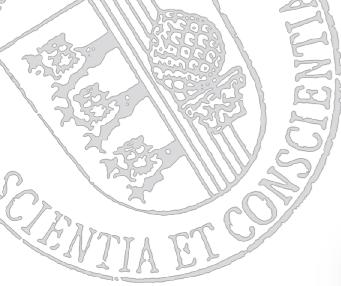
5. Oracle SQL



SQL

- SQL = Structured Query Language (ANSI und ISO standardisiert)
- Vorgänger: SEQUEL (E. Codd, IBM, 1970er)
- leider nicht vollständig von DB-Systemen unterstützt

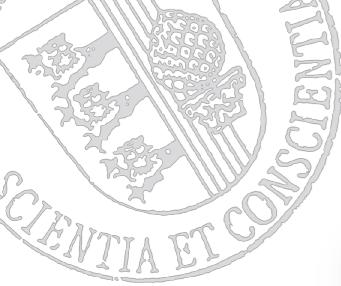




5.1 Oracle SQL DDL

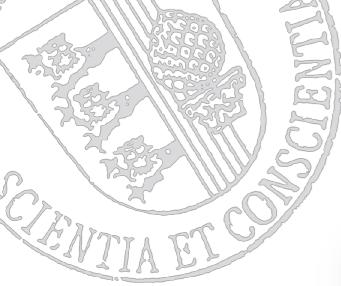
DDL

- CREATE
- DROP
- (ALTER)



Oracle SQL DDL

Object	Description
Table	Basic unit of storage; composed of rows
View	Logically represents subsets of data from one or more tables
Sequence	Generates numeric values
Index	Improves the performance of some queries
Synonym	Gives alternative name to an object



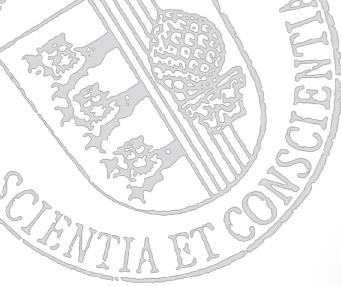
CREATE TABLE

Notwendig

- CREATE TABLE Privileg
- Speicherbereich

Syntax

```
CREATE TABLE [schema.]table
    (column datatype [DEFAULT expr]
     [column_constraint],
     ...
     [table_constraint] [, . . .]);
```



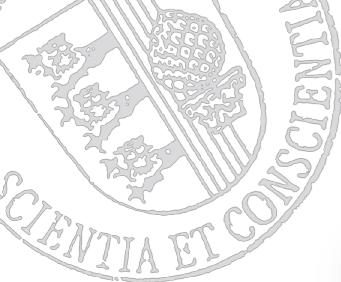
CREATE TABLE

```
CREATE TABLE table
    [(column, column...)]
AS subquery;
```

```
ALTER TABLE employees READ ONLY;

-- perform table maintenance and then
-- return table back to read/write mode

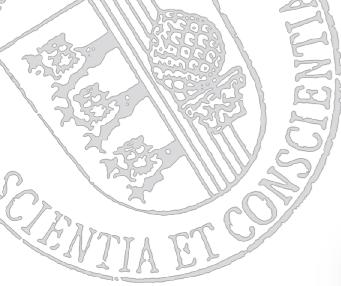
ALTER TABLE employees READ WRITE;
```



Oracle Datentypen

Datentyp	Beschreibung	Bereich
VARCHAR2(s)	String variabler Länge	$1 \leq s \leq 4000$ Zeichen
NVARCHAR2(s)	String variabler Länge, spez. char. sets	$1 \leq s \leq 4000$ Bytes
CHAR(s)	String fester Länge	$1 \leq s \leq 2000$ Zeichen
NCHAR(s)	String fester Länge, spez. char. sets	$1 \leq s \leq 2000$ Bytes
LONG	String variable Länge	bis 2 GByte
NUMBER(p,s)	Zahl mit p Stellen und Skalierung s	$1 \leq p \leq 38, -84 \leq s \leq 127$
DATE	Datum- und Zeitangaben	1.1.4712 BC bis 31.12.9999
RAW(s)	binäre Daten	bis 2000 Byte
LONG RAW	binäre Daten	bis 2 GByte
CLOB	Character Large Object	bis 4 GByte
NCLOB	Character Large Object, spez. char. sets	bis 4 GByte
BLOB	Binary Large Object	bis 4 GByte
BFILE	Verweis auf Binärdatei im Filesystem	bis 4 GByte
ROWID	hexadez. String, für ROWID-Pseudo-Spalte	
UROWID[(s)]	hexadez. String, log. Adresse einer Zeile	

http://download.oracle.com/docs/cd/B19306_01/server.102/b14200/sql_elements001.htm



NUMBER

Eine Zahl vom Typ **NUMBER(p, s)** hat p relevante Stellen und wird auf s Stellen hinter dem Komma gerundet.

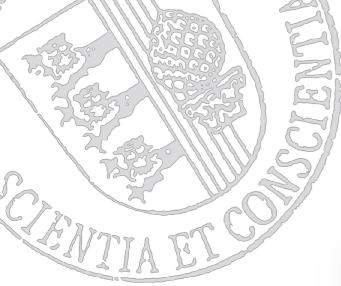
Beispiel:

Die Zahl

123,89

soll in verschiedenen Formaten in einer Tabelle abgelegt werden:

Format (NUMBER (p [, s]))					
Zahl					



DEFAULT Werte

Es können Standardwerte angegeben werden, mit denen ein Attribut belegt wird, falls dieses nicht explizit beim `INSERT` gesetzt wird.

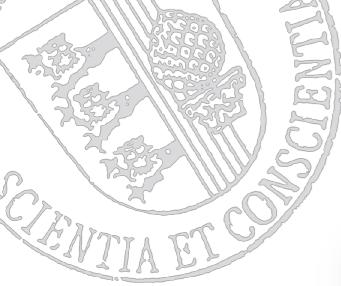
Standardmäßig ist `NULL` der Default-Wert.

```
CREATE TABLE [schema.]table  
(column datatype [DEFAULT expr] [, ...]);
```

-- Beispiele:

...

```
year NUMBER DEFAULT EXTRACT (YEAR FROM SYSDATE),  
tag DATE DEFAULT TRUNC(SYSDATE),  
jahr VARCHAR2(4) DEFAULT TO_CHAR(SYSDATE, 'YYYY'),  
...
```

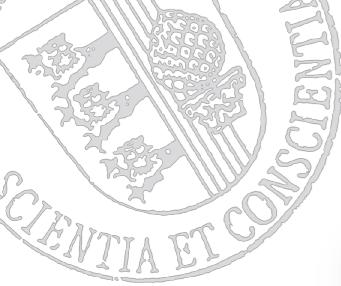


Sequenzen

- **Aufgabe:** Erzeuge z.B. automatisch **fortlaufenden Primärschlüssel**
- MySQL (AUTO_INCREMENT)
- SQL Server (IDENTITY)
- **Oracle: Sequenzen**
- Ab **Oracle 12c** gibt es sog. Identitätsspalten (IDENTITY) => später

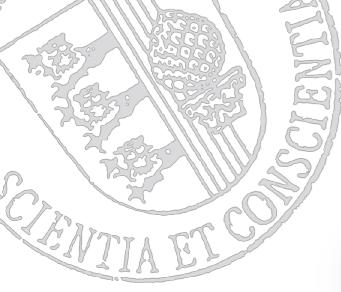
Fortlaufende Zahlenwerte, die umfangreich konfiguriert werden können.

```
CREATE SEQUENCE <Sequenzname>
    [ INCREMENT BY <Wert> ]
    [ START WITH <Startwert> ]
    [ MINVALUE <Minimum> | NOMINVALUE ]
    [ MAXVALUE <Maximum> | NOMAXVALUE ]
    [ CYCLE | NOCYCLE ]
    [ CACHE <Cachegröße> | NOCACHE ]
    [ ORDER | NOORDER ];
```



Sequenzen

- **INCREMENT BY**: (ganzzahlige) Schrittweite bei der Inkrementierung der Sequenz. Standard ist 1.
- **START WITH**: Der Startwert, der zwischen Minimum und Maximum liegen muss.
Bei positiver Schrittweite ist der Standardwert **MINVALUE**, bei negativer ist es **MAXVALUE**.
- **MINVALUE**: Der minimale Sequenzwert. Standard ist 1.
- **NOMINVALUE** setzt den minimalen Sequenzwert auf den Standardwert.
- **MAXVALUE**: Der maximale Sequenzwert. Standard ist $10^{27}-1$
- **NOMAXVALUE** setzt den maximalen Sequenzwert auf den Standardwert.
- **CYCLE**: Ist die Wertgrenze (Minimum oder Maximum) der Sequenz erreicht, beginnt die Zahlenfolge bei der anderen Wertgrenze erneut. Um diese Option zu verwenden, muss jeweils die Wertgrenze angegeben werden.
- **NOCYCLE**: Ist der maximale Wert der Sequenz erreicht, können keine weiteren Werte mehr gelesen werden.
- **CACHE**: Es wird (zur Performance-Steigerung) nicht jeder Wert der Sequenz von der Festplatte gelesen. Bei Festplattenzugriff werden die nächsten <Cachegröße> Sequenzwerte errechnet, die dann nach und nach zurückgegeben werden. Bei der Verwendung eines Caches kann nicht garantiert werden, dass alle Werte der Sequenz vergeben werden – es können Lücken auftreten. Standard ist ein Cache mit 20 Elementen.
- **NOCACHE**: Jeder Wert der Sequenz wird direkt von der Festplatte gelesen.
- **ORDER**: Garantiert, dass die Werte genau in der Reihenfolge der Anforderungen vergeben werden. **NOORDER** ist der Standard.



Sequenzen

Die folgenden beiden Statements sind gleichbedeutend:

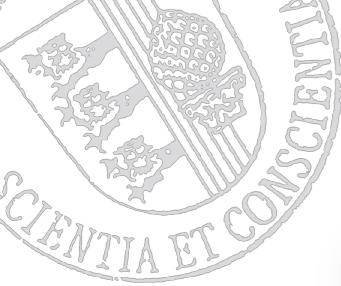
CREATE SEQUENCE <Sequenzname>

und

Sequenzen werden mittels

DROP SEQUENCE <Sequenzname>

gelöscht.



Sequenzen

- Der Zugriff auf die Werte einer Sequenz erfolgt mit:

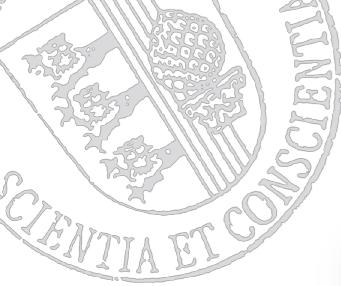
`<Sequenzname>.NEXTVAL` -- der nächste Wert
`<Sequenzname>.CURRVAL` -- der aktuelle Wert
- Dabei ist CURRVAL allerdings nur dann belegt, wenn in der gleichen Session bereits mind. ein Mal NEXTVAL aufgerufen wurde

Beispiel:

```
CREATE SEQUENCE gerade_zahlen
INCREMENT BY 2
START WITH 2;
```

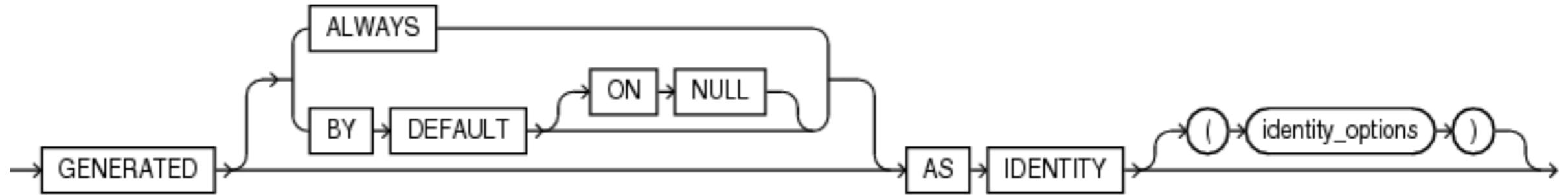
```
SELECT gerade_zahlen.NEXTVAL FROM dual; -- Ergebnis: 2

-- Einfügen in Tabelle
INSERT INTO <tablename> VALUES( . . . , <Sequenzname>.NEXTVAL, . . . );
```



- Sequenzen wurden ab **Oracle 12** durch **IDENTITY Columns** abgelöst
- **Auto-Increment** des Primärschlüssels
- Benötigt **CREATE SEQUENCE** Privileg

```
CREATE TABLE t (
    ID NUMBER GENERATED ALWAYS AS IDENTITY
    START WITH 1000 INCREMENT BY 1,
    text VARCHAR2(50)
);
```



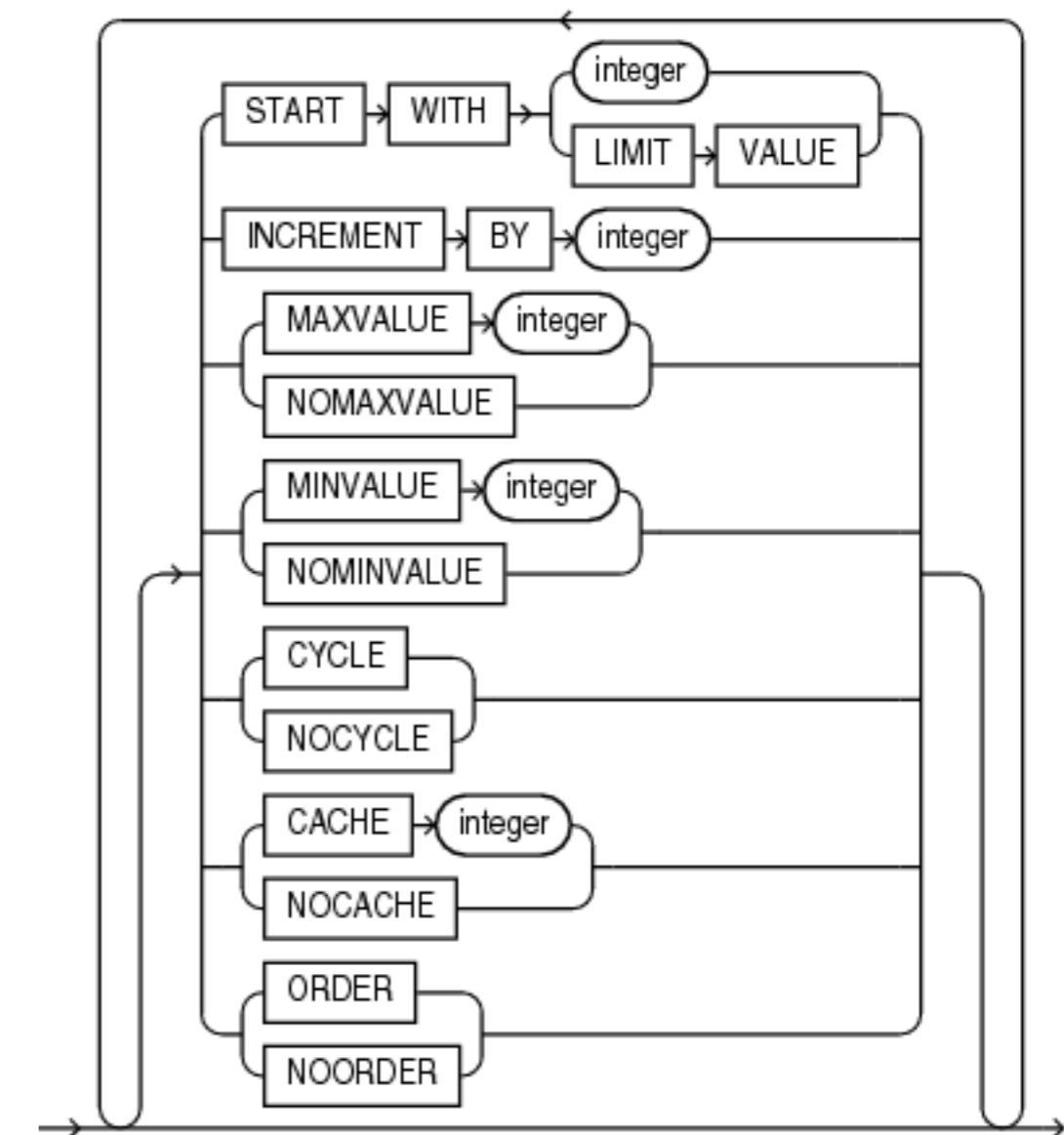
... **GENERATED**

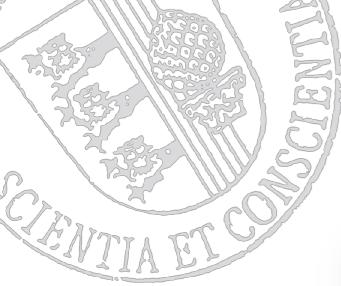
```
[ ALWAYS | BY DEFAULT [ ON NULL ] ]  
AS IDENTITY [ ( identity_options ) ]
```

- **GENERATED ALWAYS:** Es ist nicht möglich eigene Primärschlüssel einzufügen
- **BY DEFAULT:** Eigene Primärschlüssel können vergeben werden. Die Verwendung von **NULL** führt zum Fehler
- **BY DEFAULT ON NULL:** Eigene Primärschlüssel können vergeben werden. Die Verwendung von **NULL** führt zum Einfügen eines Primärschlüssels

- **identity_options**: Analog zu Sequenzen

```
{ START WITH ( integer | LIMIT VALUE )
  | INCREMENT BY integer
  | ( MAXVALUE integer | NOMAXVALUE )
  | ( MINVALUE integer | NOMINVALUE )
  | ( CYCLE | NOCYCLE )
  | ( CACHE integer | NOCACHE )
  | ( ORDER | NOORDER ) }...
```





Synonyme

Synonyme sind **alternative Namen** für beliebige Datenbankobjekte wie z. B. Tabellen oder Views.

```
CREATE [ PUBLIC ] SYNONYM <Name>  
FOR <Objekt>
```

```
DROP [ PUBLIC] SYNONYM [ schema.] <synonym_name> [ FORCE] ;
```

PUBLIC: Synonym lässt sich global benutzen, d.h. von allen Benutzern

Beispiele:

- dual: Synonym für sys.dual
- Tabelle used_cars_in_stock von Übungsblatt 4