

ORACLE und IBM DB2 Datentypen

PHP-User-Group Stuttgart

14.12.2011

- ORACLE Datentypen – ein Überblick
- IBM DB2 Datentypen – ein Überblick

ORACLE und IBM DB2 Datentypen

Wer bin ich ?

Thomas Wiedmann

- ◆ n+1 Jahre Problemlösungen in der Softwareentwicklung
- ◆ Seit „vielen“ Jahren Projekte mit PHP und Oracle PL/SQL bzw. DB2/NT
- ◆ Zend Certified PHP Engineer (ZCE)
- ◆ IBM Certified Solution Expert - DB2 UDB v7.1 Database Administration
- ◆ Autor diverser Fachartikel in der „Toolbox“ und im PHP-Magazin
- ◆ Autor des Buches „DB2 – SQL, Programmierung, Tuning“ © 2001
- ◆ SQL-Tipps, MySQL-EXPLAIN und Performance in der [SQL-Backstube](#)

Datentypen allgemein

1. Standard

- a) Numerische Daten Typen
- b) Alphanumerische Daten Typen
- c) Datum/Zeit Daten Typen

2. Datentypen mit erweiterter Funktionalität

- a) Zahlengeneratoren (Sequence, Identity)

3. Erweiterte „Datentypen“

- a) XML Daten Typen
- b) User-Defined Daten Typen
- c) DB2 Extenders bzw. ORACLE PL/SQL Pakete

ORACLE Datentypen

1. Build-In

a) numerische Datentypen

- **NUMBER(p,s)**
 - Wertebereich $1,0 \times 10^{-130}$ - $1,0 \times 10^{126}$
 - bei NUMBER(n) darf n max. 38 sein.
- **INTEGER, FLOAT**

INT, INTEGER, DEC(x,y), wird gemappt auf NUMBER(p,s)

=> z. B. Umsetzung TINYINT => NUMBER(3)

Der physikalische Speicherplatz von NUMBER(p,s) lässt sich mit der Formel "ROUND((length(p)+s)/2))+1" bestimmen. p ist dabei die Zahl der signifikanten Stellen, s gibt an ob die Zahl positiv ist(0) oder negativ (1).

ORACLE Datentypen

1. Build-In

b) alphanumerische Datentypen (1 Byte Zeichensätze)

- **CHAR(n)**
 - alphanumerische Werte mit fester Länge
 - n max. 2000
- ~~VARCHAR(n)~~ **nicht verwenden, wird gemappt auf
auf VARCHAR2(n)**
- **VARCHAR2(n)**
 - alphanumerische Werte mit variabler Länge
 - n max. 4000

ORACLE Datentypen

1. Build-In

c) alphanumerische Datentypen (UNICODE)

- **NCHAR(n)**
 - alphanumerische Werte mit fester Länge
 - n max. 2000
- **NVARCHAR2(n)**
 - alphanumerische Werte mit variabler Länge
 - n max. 4000

ORACLE Datentypen

1. Build-In

d) LOB Large Object

- **BLOB**
 - Binary large object
 - max. 4 GB
- **CLOB**
 - Character large object
 - max. 4 GB
 - mit PHP 5.2.x und PDO max. 32kB möglich!
- **NCLOB**
 - Character large object für UNICODE
 - max. 4 GB

ORACLE Datentypen

1. Build-In

e) Datum und Zeit

- **DATE**
 - Speichert Datum und Zeit (YYYY-MM-DD HH:MM:SS)
- **TIMESTAMP(n)**
 - Erweiterung von DATE plus Sekundenbruchteile
- **TIMESTAMP(n) WITH TIME ZONE**
- **TIMESTAMP(n) WITH LOCAL TIME ZONE**
- **INTERVAL YEAR(n) TO MONTH**
 - Speichert die Differenz zwischen zwei Datum Werten
- **INTERVAL DAY(n1) TO SECOND(n2)**
 - Speichert die Differenz zwischen zwei Zeit Werten

ORACLE Datentypen

1. Build-In

f) XML (basiert auf speziellen PL/SQL Paketen)

- **XMLTYPE**

- Speichern XML Bäume
- XMLType API auf Basis von PL/SQL
- enthält XPath() Funktionalität

- **Simple XML Table**

- CREATE TABLE <table_name> OF SYS.XMLTYPE
XMLSCHEMA "<.xsd_doc>"
ELEMENT "<element_name>" ID <integer>
TABLESPACE <tablespace_name>;

http://www.morganslibrary.org/reference/xml_tables.html

ORACLE Datentypen

1. Build-In

g) sonstiges

- **BFILE**

- Pointer auf eine Datei im Filesystem
- Speicher liegt außerhalb der Datenbank
- max. 4 GB (abhängig vom OS)

- **Geo und Spatial Data**

- Spezielle PL/SQL Pakete
- z. B. CREATE INDEX territory_idx
ON territories (territory_geom)
INDEXTYPE IS MDSYS.SPATIAL_INDEX;

http://download.oracle.com/docs/cd/B10501_01/appdev.920/a96630/sdo_index_query.htm

ORACLE Datentypen

2. User-Defined Type (Objektorientierung)

a) create type

- Object types
- Varray type
- Nested type

Beispiel:

```
create or replace type person_t as object (  
    first_name varchar2(30) ,  
    last_name  varchar2(5)  
);
```

ORACLE Datentypen

3. Neues im PL/SQL Umfeld

- BOOLEAN
- ANYDATA
- ARRAY
- ROW
- CURSOR

- derzeit nicht mit PDO möglich

```
SELECT a.id, a.name ,
       CURSOR(SELECT k.id, k.art, k.nummer
              FROM kontakte k
              WHERE k.id = a.id
              ) AS kontakte
FROM adresse a;
```

```
+-----+-----+-----+
! ID   ! NAME           ! KONTAKTE      !
+-----+-----+-----+
!  1   ! Muster1       ! <cursor>      !
!  2   ! Muster2       ! <cursor>      !
+-----+-----+-----+
```

ORACLE Datentypen - Zusammenfassung

- a) Numerische Datentypen basieren auf NUMBER(x,y)
- b) VARCHAR2(n) anstatt VARCHAR(n)
- c) Im Zweifel immer die Original Referenz nutzen
 - Versionsunterschiede , 32/64 Bit, OS, etc.
- d) PL/SQL hat teilweise weitere Datentypen und andere Wertebereiche (VARRAY, PLS_INTEGER...)
- e) Autoincrement ? => SEQUENCE
- f) <http://www.morganslibrary.org/reference/datatypes.html>

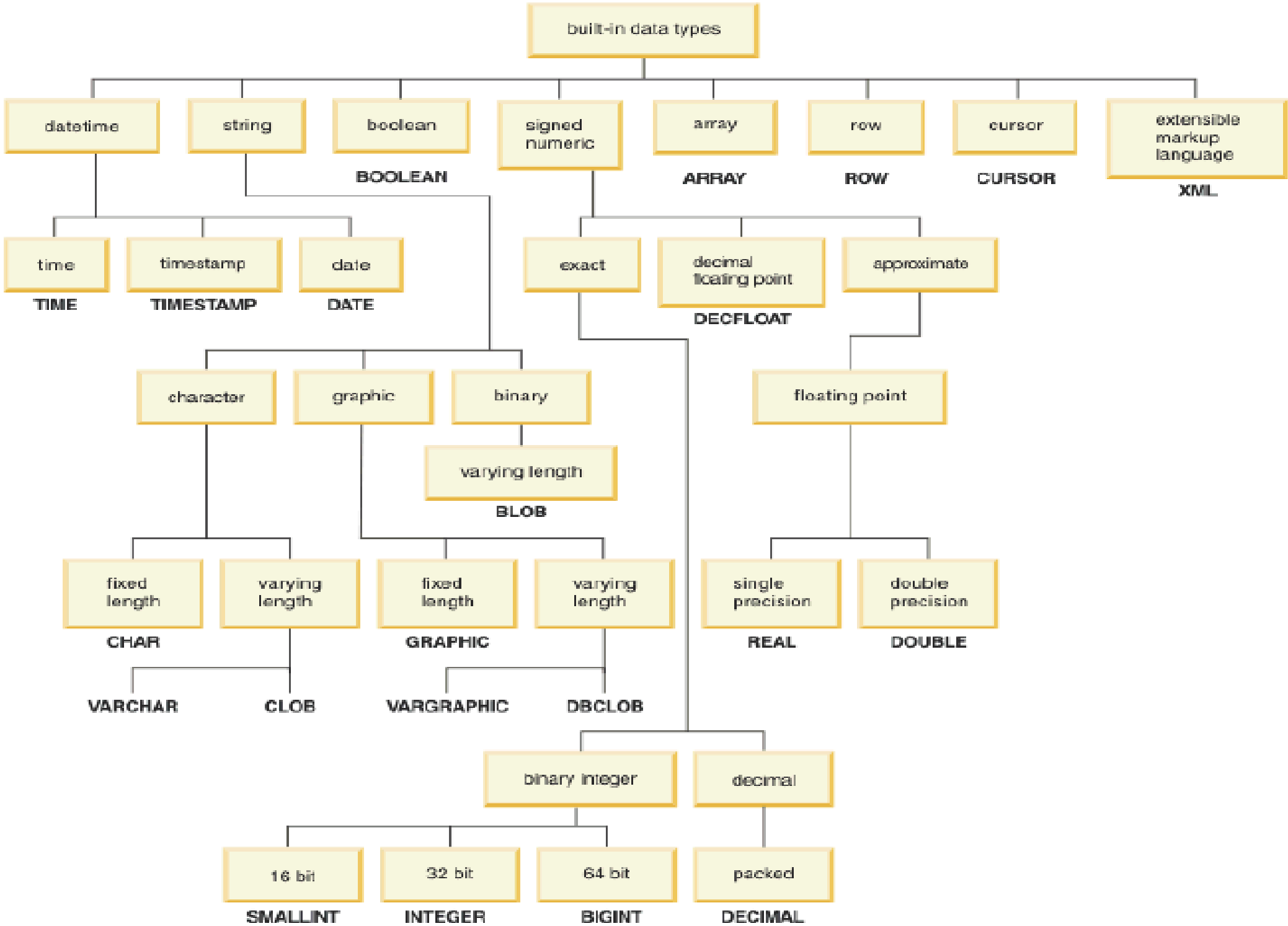


Abbildung 8.1 – In DB2 eingebaute Datentypen

IBM DB2 Datentypen

1. Build-In

a) numerische Datentypen

- **DEC(p,s), DECIMAL(p,s) , NUMERIC(p,s)**
- **INTEGER / INT**
- Wertebereich -2.147.483.648 - +2.147.483.647
- **Big Integer / BIGINT (64 Bit Integer)**
- **REAL / FLOAT (n) – single-precision floating-point**
- **DOUBLE (n)**
- **SMALLINT (Wertebereich -32.768 bis +32.768)**

IBM DB2 Datentypen

1. Build-In

b) alphanumerische Datentypen (1 Byte Zeichensätze)

- **CHAR(n)**
 - alphanumerische Werte mit fester Länge
 - n max. 255 (neue Versionen max. 32kB)
- **VARCHAR(n)**
 - alphanumerische Werte mit variabler Länge
 - n max. 32.672
- **LONG VARCHAR(n)**
 - alphanumerische Werte mit variabler Länge
 - n max. 32.700 (ähnlich CLOB)

IBM DB2 Datentypen

1. Build-In

c) alphanumerische Datentypen (UNICODE)

- **GRAPHIC**
- **VARGRAPHIC**
- **LONG VARGRAPHIC**

IBM DB2 Datentypen

1. Build-In

d) LOB Large Object

- **BLOB**
 - Binary large object
 - max. 1 bzw. 2 GB
- **CLOB**
 - Character large object
 - max. 1 bzw. 2 GB
- **DBCLOB**
 - Double byte character large object (Unicode)

IBM DB2 Datentypen

1. Build-In

e) Datum und Zeit

- **DATE (4 Byte packed, external 10 Byte)**
- Speichert Datum (YYYY-MM-DD)
- **TIME (8 Byte)**
- Speichert Datum (HH:MM:SS)
- **TIMESTAMP (26 Byte)**
- Speichert Datum (YYYY-MM-DD-HH:MM:SS:NNNNNN)

IBM DB2 Datentypen

1. Build-In

f) XML

- **pureXML**
 - native XML Speicherung
 - enthält XPath und XQuery Funktionalität

```
create table customer(info XML);
```

```
select xmlquery('$i/customerinfo[phone = "905-555-4789"]/name'  
passing info as "i") from customer;
```

<http://www-01.ibm.com/software/data/db2/xml/>

IBM DB2 Datentypen

1. Build-In

g) sonstiges

- **DATALINK**
 - Logische Reference auf eine Datei die im „Data Link Manager Server“ gespeichert ist
- **Identity Column**
 - Entspricht in etwa dem MySQL Autoincrement, kann aber auch Spalten (z.B. Nettopreis + MWST) zusammen rechnen.
 - z. B. `..kunde INT GENERATED ALWAYS AS IDENTITY,`

IBM DB2 Datentypen

2. User-Defined Data Type

User-defined distinct

Datentypen mit zusätzlichen Eigenschaften

- .. **DISTINCT TYPE euro AS DECIMAL(10,2) ..**
- .. **DISTINCT TYPE renminbi AS DECIMAL(10,2) ..**
- .. **DISTINCT TYPE usadollar AS DECIMAL(10,2) ..**

=> Ein fehlerhaftes Addieren von z. B. **euro** + **usadollar** ist so nicht möglich, es muss immer eine korrekte Typumwandlung erfolgen.

IBM DB2 Datentypen

2. User-Defined Data Type (Objektorientierung)

User-defined structured

Eigene Datentypen aus mehreren build-in Datentypen zusammenbauen.

```
CREATE TYPE angestellter AS (  
    NAME          VARCHAR(32),  
    ID            INT,  
    ABT          VARCHAR(32),  
    GEHALT       DECIMAL(10,2)  
) MODE DB2SQL;
```

```
CREATE TYPE bigboss UNDER angestellter AS (  
    BONUS        DECIMAL(10,2)  
) MODE DB2SQL;
```

IBM DB2 Datentypen

Zusammenfassung

- a) pureXML – nativer XML Support
- b) Im Zweifel immer die Original Referenz nutzen
 - Versionsunterschiede , 32/64 Bit, OS, etc
- c) IBM DB2 9.7 unterstützt ORACLE PL/SQL
- d) Autoincrement ? => SEQUENCE oder IDENTITY

http://publib.boulder.ibm.com/infocenter/iwedhelp/v6r0/index.jsp?topic=%2Fcom.ibm.db2e.doc%2Fdbsap_b2.html

Download: <http://www-01.ibm.com/software/data/db2/linux-unix-windows/edition-express-c.html>

ORACLE und IBM DB2 Datentypen

Zusammenfassung

- ◆ Im Zweifel immer die Original Referenz nutzen (Versionsunterschiede , 32/64 Bit, OS, API etc.)
- ◆ Den „richtigen“ Datentyp auswählen.
 - keinen Platten-/Speicherplatz verschwenden
 - richtige Datentypen = bessere Performance
- ◆ IBM DB2 v9.7 „nähert“ sich ORACLE an
 - http://docs.oracle.com/cd/E18464_01/doc.30/e18460/oracle_db2_compared.htm

ORACLE und IBM DB2 Datentypen

Blick in die Zukunft

Gibt es einen neuen Datenbanktrend (-Hype?), dann wird der über kurz oder lang auch in diesen Datenbanken unterstützt.

- ◆ Objektorientierte Datenbank
- ◆ XML Datenbank
- ◆ NoSQL Techniken

Oracle NoSQL

- ◆ <http://www.oracle.com/us/products/database/nosql/overview/index.html>
- ◆ http://www.silicon.de/technologie/software/0,39044013,41556280,00/auch_oracle_macht_in_nosql.htm

IBM DB2 NoSQL

- ◆ <http://nosql.mypopescu.com/post/12156340083/ibm-db2-to-include-nosql-features>

ORACLE und IBM DB2 Datentypen

Blick in die Zukunft

Gibt es einen neuen Datenbanktrend (-Hype?), dann wird der über kurz oder lang auch in diesen Datenbanken unterstützt.

The End!

- ◆ Objektorientierte Datenbank
- ◆ XML Datenbank
- ◆ NoSQL Techniken

Oracle NoSQL

- ◆ <http://www.oracle.com/us/products/database/nosql/overview/index.html>
- ◆ http://www.silicon.de/technologie/software/0,39044013,41556280,00/auch_oracle_macht_in_nosql.htm

IBM DB2 NoSQL

- ◆ <http://nosql.mypopescu.com/post/12156340083/ibm-db2-to-include-nosql-features>