

A JDBC típusok

Mivel az SQL típusok és a Java típusok különböznek egymástól, ezért a paraméter átadás megoldásához szükséges, hogy új úgynevezett JDBC típusokat vezessünk be, melyek reprezentálják az SQL típusokat. Ezeket a típusokat a [java.sql.Types](#) osztály definiálja. A már felépített kapcsolaton keresztül lehetőségünk nyílik arra, hogy mindenféle hasznos információkat kérdezzünk le az adatbázisunkról, illetve a lekérdezés eredménytáblájáról. Elbbit a DatabaseMetaData interfészen, utóbbit a ResultSetMetaData interfészen keresztül kérhetjük le.

A DatabaseMetaData interfészt megvalósító objektumhoz a Connection objektum getMetaData metódusával juthatunk mely az alábbi fontosabb metódusokat tartalmazza:

- **getTypeInfo** – egy ResultSet típusú objektumot ad vissza, mely tartalmazza az adatbázis illetve a JDBC meghajtó által támogatott JDBC típusokat.
- **getUDTs** – egy ResultSet típusú objektumot ad vissza, mely tartalmazza a felhasználói típusok leírását egy meghatározott keresési mintán belül. Felhasználói típusnak megadhatunk Types.DISTINCT, Types.STRUCT, és Types.JAVA_OBJECT típusokat, melyeket a későbbiekben részletezni fogok. Ezt a funkciót hiába próbáljuk MySQL esetén, nem fog visszaadni semmit csak egy üres ResultSet objektumot mivel nem támogatja a felhasználói adattípusokat, ellenben például a PostgreSQL adatbáziskezével szépen működik.
- **getColumnns** – egy ResultSet típusú objektumot ad vissza, mely minden információt tartalmaz a táblákban lévő mezőkről egy meghatározott keresési mintán belül. Például: COLUMN_NAME, DATA_TYPE, TYPE_NAME, COLUMN_SIZE

A ResultSetMetaData interfészt megvalósító objektumhoz a ResultSet objektum getMetaData metódusával juthatunk mely az alábbi fontosabb metódusokat tartalmazza:

- **getColumnType** – egy integer értéket ad vissza, mely reprezentálja az SQL-beli típust.
- **getColumnTypeName** – egy String értéket ad vissza, mely megnevezi az SQL-beli típust.

Típusmegfeleltetés

Az alábbi táblázat megmutatja, hogy adott JDBC típusnak milyen Java típus felel meg:

JDBC Type	Java Type
CHAR	String
VARCHAR	String
LONGVARCHAR	String
NUMERIC	java.math.BigDecimal
DECIMAL	java.math.BigDecimal
BIT	boolean
BOOLEAN	boolean
TINYINT	byte
SMALLINT	short
INTEGER	int
BIGINT	long
REAL	float
FLOAT	double
DOUBLE	double
BINARY	byte[]
VARBINARY	byte[]
LONGVARBINARY	byte[]
DATE	java.sql.Date
TIME	java.sql.Time
TIMESTAMP	java.sql.Timestamp
CLOB	Clob
BLOB	Blob
ARRAY	Array

DISTINCT	mapping of underlying type
STRUCT	Struct
REF	Ref
DATALINK	java.net.URL
JAVA_OBJECT	underlying Java class

	T Y P E	S M A L L I N T	I N T E G E R	B I G I N T	R E A L	F L O A T	D O U B L E	D E C I M A L	N U M E R I C	B I T	C H A R	V A R C H A R	L O N G V A R C H A R	B I N A R Y	V A R B I N A R Y	L O N G V A R B I N A R Y	D A T E	T I M E	T I M E S T A M P
String	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
java.math.BigDecimal	x	x	x	x	x	x	x	x	x	x	x	x	x						
Boolean	x	x	x	x	x	x	x	x	x	x	x	x	x						
Integer	x	x	x	x	x	x	x	x	x	x	x	x	x						
Long	x	x	x	x	x	x	x	x	x	x	x	x	x						
Float	x	x	x	x	x	x	x	x	x	x	x	x	x						
Double	x	x	x	x	x	x	x	x	x	x	x	x	x						
byte[]														x	x	x			
java.sql.Date											x	x	x				x		x
java.sql.Time											x	x	x					x	
java.sql.Timestamp											x	x	x				x	x	x

Az értékeket lekérdezni, beállítani illetve frissíteni a get<típus> , set<típus> és update<típus> kezdet metódusokkal lehet.