# Support for IBM DB2 database

The HotRod DB2 adapter automatically maps known database column types to DAO Java types. In most of the cases this default Java type is well suited to handle the database values. However, when needed the default Java type of a property can be overridden by a custom type if it's provided by the developer.

## Default Java Types

If a custom Java type is not specified HotRod will use the following rules to decide which Java type to use for each Oracle column. In yellow is the DAO property type. In parenthesis the actual object type returned by the IBM DB2 JDBC driver, that on occasions may be different.

Please note that the Java types for the IBM DB2 columns may vary depending on the specific version and variant of the RDBMS, the operating system where the database engine is running, and the JDBC driver version.

|  |  |
| --- | --- |
| **DB2 Column Type** | **Default Java Type** |
| SMALLINT | java.lang.Short |
| INTEGER,  INT | java.lang.Integer |
| BIGINT | java.lang.Long |
| DECIMAL(p,s),  DEC(p,s),  NUMERIC(p,s),  NUM(p,s) | If s is specified and different from zero the Java type is:   * java.math.BigDecimal   if s is not specified or it's zero:   * if p <= 2: java.lang.Byte * if 2 < p <= 4: java.lang.Short * if 4 < p <= 9: java.lang.Integer * if 8 < p <= 18: java.lang.Long * if p > 18: java.math.BigInteger |
| DECFLOAT | java.math.BigDecimal |
| REAL | java.lang.Float |
| FLOAT,  DOUBLE | java.lang.Double |
| CHAR(n),  CHARACTER(n),  VARCHAR(n),  CHARACTER VARYING(n),  CLOB(n),  GRAPHIC(n),  NCHAR(n),  VARGRAPHIC(n),  NVARCHAR(n),  DBCLOB(n),  NCLOB(n),  LONG VARCHAR,  LONG VARGRAPHIC | java.lang.String |
| VARCHAR FOR BIT DATA,  LONG VARCHAR FOR BIT DATA,  CHAR(n) FOR BIT DATA,  BLOB(n) | byte[] |
| DATE | java.sql.Date |
| TIME | java.sql.Time |
| TIMESTAMP | java.sql.Timestamp |
| XML | java.lang.String |
| BOOLEAN (pseudo type) | java.lang.String |
| BINARY | No default HotRod data type |
| VARBINARY | No default HotRod data type |
| ROW | No default HotRod data type |
| ARRAY | No default HotRod data type |

## Custom Java Types

To override the default Java type see the reference section for the tables, views, and selects. The example Custom DAO Property Java Types shows a some cases where a custom type overrides the default type. To do it add a <column> tag in a <table>, <view>, or <select> definition as in:

<table name=*"my\_table"*>

<column name=*"price"* java-type=*"java.lang.Double"* jdbc-type=*"NUMERIC"* />

</table>

This configuration will force the property type to java.lang.Double instead of java.math.BigDecimal (the default type).