Data Type Mapping

Contents

Error rendering macro 'toc'

[com.ctc.wstx.exc.WstxLazyException] com.ctc.wstx.exc.WstxParsingException: String '--' not allowed in comment (missing '>'?) at [row,col {unknown-source}]: [147,4]

HVR's mapping/conversion of data types is complex because each DBMS's data types have a specific range which seldom corresponds the range of another DBMS. For example, data type **varchar(10)** in SQL Server corresponds to **varchar2(10 bytes)** in Oracle, but **varchar(8000)** corresponds to **clob**. Note that the mapping here does not just depend on the 'name' of the data type, but also its 'attributes' like byte length, encoding, scale and precision, etc. If HVR is not accurate in mapping data types, then target tables could be created which are unable to contain the data delivered from the source DBMS.

Data types are not directly mapped/converted from the source DBMS to the target DBMS, instead they are first mapped to HVR's own 'catalog data types' and then these catalog data types are mapped to corresponding data types in target DBMS or file format. Data type mapping/conversion happens in the following manner:

1. When a channel is built using **Table Explore** or using the command **Hvradapt**, the source DBMS data types mapped to HVR's own 'catalog data types'.
   After the tables are added to a channel the mapping of data types in source DBMS (capture location) to HVR's catalog data types can be viewed in **HVRGUI** from the **Table Explore** window.
2. Then during **HVR Refresh** and **Integrate**, HVR's catalog data types are mapped to corresponding data types in the target DBMS (integrate location). This mapping happens at the moment the **HVR Refresh** is used to create target tables.
   After performing **HVR Refresh**/**Integrate** the mapping of HVR's catalog data types to the target DBMS data types can be viewed in **HVRGUI** from the **Table Explore** window.

Customizing Data Type Mapping

If the automatic/default mapping of data type is not appropriate for a specific channel, it can be modified using action **ColumnProperties** **/Datatype**.

For example, by default HVR maps a **number** (without scale or precision) in Oracle to **numeric(38,4)** in SQL Server. By defining the following action, **number** (without scale or precision) in Oracle is mapped to **float** instead:

Group	Table	Action
SRCGRP	*	**ColumnProperties /DatatypeMatch="number[prec=0 && scale=0]" /Datatype="float"**

In the above example, **ColumnProperties** **/DatatypeMatch** is used for mapping all columns with **numb er** (without scale or precision) data type into **float** data type.

Alternatively, **ColumnProperties** **/Name** can be used for mapping the data type of a specific column (e. g. **MYCOLUMN**) into **float** data type.

Group	Table	Action

SRCGRP	*	**ColumnProperties /Name="MYCOLUMN" /Datatype="float"**

Extended Data Type Support

'Extended data types' are the DBMS data types (e.g. **sql_variant** in SQL Server or **xmltype** in Oracle) which are not mapped to native HVR data types. Instead, HVR's **Extended Data Type Support** feature should be used for such data types.

HVR uses action **ColumnProperties /CaptureExpression** for converting an extended data type to a supported data type during **capture**, **compare** or **refresh** (source) and **ColumnProperties /IntegrateExpression** for converting it back to extended data type during **integrate**, **compare** or **refresh** (target) respectively.

Data Type Mapping - Source to Target DBMS

In this section, you can find the mapping of data types in source DBMS to the corresponding data types in target DBMS or file format.

1. Select the required source DBMS in **Source Location Class**; displays the mapping of source DBMS data type to HVR catalog data type.
2. Select the required target DBMS in **Target Location Class**; displays the mapping of HVR catalog data type to target DBMS data type.

For example, to illustrate the data type mapping of **timestamp with time zone** in Oracle to the corresponding data type in Avro:

1. Select Oracle in **Source Location Class**.
   In the table displayed below **Source Location Class**, you can find that **timestamp with time zone** (in column **HVR Catalog Data Type**) is mapped to HVR's catalog data type **timestamp with tz (oracle)** (in column **HVR Catalog Data Type**).
2. Select Avro 1.6 and Avro 1.8 in **Target Location Class**.
   In the table displayed below **Target Location Class**, you can find that **timestamp with tz (oracle)** (in column **HVR Catalog Data Type**) is mapped to **varchar(100)** in Avro 1.6 and **timestamp-millis** in Avro 1.8.

Source Location Class

DB2 for i DB2 for Linux, UNIX and Windows DB2 for z/OS HANA Ingres MariaDB MySQL Oracle PostgreSQL SQL Server / Azure SQL Database

The version of the source location DBMS also affects how data types are mapped in HVR; the table below only corresponds to the latest DBMS versions.

Target Location Class

- The version of the target location DBMS also affects how HVR maps data types; It does not map to data types which are not supported by an older version of DBMS in target location. For simplicity, this is not reflected in the documentation here; the table below only corresponds to the latest supported DBMS versions.
- For data type Attributes (e.g, charlen, bytelen, scale, precision etc.) every feasible combination of values is possible. However, this table contains only enough combinations to illustrate the effect of variation.

Data Type	HVR Capture Support	HVR Catalog Data Type
bigint	Native	bigint
decfloat	Native	decfloat
decimal	Native	decimal
double	Native	double
integer	Native	integer
numeric	Native	numeric (db2i)
real	Native	real
smallint	Native	smallint
date	Native	ansidate
time	Native	time2

HVR Catalog Data Types	Attributes	Aurora MySQL (UTF-8)	Aurora PostgreSQL (UTF-8)	Avro 1.6	Avro 1.8	Az Sy An
ansidate (ingres)		date	date	bigint	date	da

timestamp	Native	timestamp
char	Native	char
clob	Native	clob
varchar	Native	varchar
binary	Native	binary
blob	Native	blob
varbinary	Native	varbinary
datalink	Extended	<<datalink>>
xml	Extended	<<xml>>
rowid	Extended	<<rowid>>
dbclob	Native	clob
graphic	Native	nchar
vargraphic	Native	nvarchar
user defined	Extended	<<user defined>>

Data Type	HVR Capture Support	HVR Catalog Data Type
bigint	Native	bigint
decfloat	Native	decfloat
decimal	Native	decimal
double	Native	double
float	Native	double
integer	Native	integer
numeric	Native	decimal
real	Native	real
smallint	Native	smallint
date	Native	ansidate
time	Native	time
timestamp	Native	timestamp
char	Native	char
char for bit data	Native	binary
clob	Native	clob
long varchar	Native	long varchar (db2)
long varchar for bit data	Native	long varbinary
varchar	Native	varchar
varchar for bit data	Native	varbinary
binary	Native	binary
blob	Native	blob
varbinary	Native	varbinary
dbclob	Native	dbclob
graphic	Native	graphic
long vargraphic	Native	long nvarchar (db2)
nclob	Native	dbclob
vargraphic	Native	vargraphic
xml	Native[1]	db2 xml[1]
rowid	Not supported	
user defined	Not supported	

[1] - Since HVR 5.6.5/2, the 'xml' data type is supported as 'Native' and is mapped to the HVR Catalog Type 'db2 xml'. Action **Capture /AugmentIncomplete=LOB** is required to capture xml type from DB2 for Linux, UNIX and Windows.

Data Type	HVR Capture Support	HVR Catalog Data Type
smallint	Native	smallint
integer	Native	integer
bigint	Native	bigint
decimal	Native	decimal
decfloat	Native	decfloat
real	Native	real
double	Native	double
date	Native	ansidate
time	Native	time2

ansidate		date	date	bigint	date	da
bfile		longblob	bytea	blob	blob	va y(
bigint unsigned	bytelen=8	bigint unsigned	numeric(20)	float	decimal (20)	nu (2
bigint	bytelen=8	bigint	bigint	bigint	bigint	bi
binary	bytelen=10	binary (10)	bytea	blob	blob	bi (1
binary_double	bytelen=8	double	float	float	float	flo
binary_float	bytelen=4	float	real	real	real	re
bit (mysql)	bitlen=32	bit(32)	varchar(32)	varchar (32)	varchar (32)	va (3
bit		tinyint	boolean	boolean	boolean	bit
blob		longblob	bytea	blob	blob	va y(
boolean		tinyint	boolean	boolean	boolean	bit
byte varying	bytelen=10	varbinary(10)	bytea	blob	blob	va y(
byte	bytelen=10	binary (10)	bytea	blob	blob	bi (1
byteint	bytelen=1	tinyint	smallint	smallint	smallint	sn

timstamp without time zone	Native	timestamp
timestamp with time zone	Native	db2 timestamp with time zone
char	Native	char
varchar	Native	varchar
clob	Native	clob
binary	Native	binary
blob	Native	blob
varbinary	Native	varbinary
dbclob	Native	nclob
graphic	Native	nchar
vargraphic	Native	nvarchar
xml	Native	db2 xml[1]
rowid	Not Supported	

[1] - Since HVR 5.6.5/2, the 'xml' data type is mapped to the HVR Catalog Type 'db2 xml'.

Data Type	HVR Capture Support	HVR Catalog Data Type
bigint	Native	bigint
decfloat	Extended	<<decfloat>>
decimal	Native	decimal
double	Native	double
integer	Native	integer
numeric	Native	numeric (db2i)
real	Native	real
smallint	Native	smallint
date	Native	ansidate
time	Native	time2
timestamp	Native	timestamp
char	Native	char
clob	Native	clob
varchar	Native	varchar
binary	Native	binary
blob	Native	blob
varbinary	Native	varbinary
datalink	Extended	<<datalink>>
xml	Extended	<<xml>>
rowid	Extended	<<rowid>>
dbclob	Native	clob
graphic	Native	nchar
vargraphic	Native	nvarchar
user defined	Extended	<<user defined>>

Data Type	HVR Capture Support	HVR Catalog Data Type
bigint	Native	integer8
decimal	Native	decimal
float	Native	float8
float4	Native	float4
float8	Native	float8
integer	Native	integer4
integer1	Native	integer1
integer2	Native	integer2
integer4	Native	integer4
integer8	Native	integer8
real	Native	float4
smallint	Native	integer2
tinyint	Native	integer1
ansidate	Native	ansidate
ingresdate	Native	ingresdate
interval day to second	Native	interval day to second (ingres)
interval year to month	Native	interval year to month (ingres)
time with local time zone	Native	time with local time zone
time with time zone	Native	time with time zone

c	bytelen=10encoding=UTF-8	char(10) charset utf8mb4	char(10)	char(10)	ch
char	bytelen=10encoding=UTF-8	char(10) charset utf8mb4	char(10)	char(10)	ch
char	bytelen=4000encoding=UTF-8	varchar(4000) charset utf8mb4	char(4000)	char(4000)	char(4
char	bytelen=8000encoding=UTF-8	varchar(8000) charset utf8mb4	char(8000)	char(8000)	char(8
char	bytelen=40charlen=10encoding=UTF-8	char(10) charset utf8mb4	char(10)	char(10)	ch
char	bytelen=10encoding=WINDOWS-1252	char(10) charset latin1	char(10)	char(10)	ch
char (oracle)	bytelen=10encoding=UTF-8	char(10) charset utf8mb4	char(10)	char(10)	ch
char (oracle)	bytelen=2000encoding=UTF-8	varchar(2000) charset utf8mb4	char(2000)	char(2000)	char(2

time without time zone	Native	time
timestamp with local timezone	Native	timestamp with local timezone
timestamp with time zone	Native	timestamp with time zone
timestamp without time zone	Native	timestamp (ingres)
c	Native	c
char	Native	char
clob	Native	long varchar
long nvarchar	Native	long nvarchar
long varchar	Native	long varchar
nchar	Native	nchar
nclob	Native	long nvarchar
nvarchar	Native	nvarchar
text	Native	text (ingres)
varchar	Native	varchar
ipv4	Extended	<<ipv4>>
ipv6	Extended	<<ipv6>>
logical_key	Extended	<<logical_key>>
money	Native	money (ingres)
uuid	Extended	<<uuid>>
byte	Native	byte
byte varying / varbyte	Native	byte varying
long byte	Native	long byte
boolean	Native	boolean
circularstring	Extended	<<circularstring>>
compoundcurve	Extended	<<compoundcurve>>
curvepolygon	Extended	<<curvepolygon>>
geometry	Extended	<<geometry>>
geometrycollection	Extended	<<geometrycollection>>
linestring	Extended	<<linestring>>
multicurve	Extended	<<multicurve>>
multilinestring	Extended	<<multilinestring>>
multipoint	Extended	<<multipoint>>
multipolygon	Extended	<<multipolygon>>
multisurface	Extended	<<multisurface>>
point	Extended	<<point>>
polygon	Extended	<<polygon>>
polyhedralsurface	Extended	<<polyhedralsurface>>
tin	Extended	<<tin>>
triangle	Extended	<<triangle>>

Data Type	**HVR Capture Support**	**HVR Catalog Data Type**
bigint	Native	bigint
bigint unsigned	Native	bigint unsigned
bit	Native	bit (mysql)
bool	Native	tinyint signed
boolean	Native	tinyint signed
dec	Native	decimal
decimal	Native	decimal
decimal unsigned	Native	decimal
double	Native	double
double precision	Native	double
double unsigned	Native	double
fixed	Native	decimal
float	Native	real
float unsigned	Native	real
int	Native	int
int unsigned	Native	int unsigned

char (oracle)	bytelen=40charlen=10encoding=UTF-8	char(10) charset utf8mb4	char(10)	char (10)	char (10)	ch
char (oracle)	bytelen=10encoding=WINDOWS-1252	char(10) charset latin1	char(10)	char (10)	char (10)	ch
clob	encoding=UTF-8	longtext charset utf8mb4	text	text	text	va (m
date		datetime(0)	timestamp (0)	varchar (100)	timestamp-millis	da 2(
date (sybase)		date	date	bigint	date	da
datetime (mysql)	prec_sec=0	datetime(0)	timestamp (0)	varchar (100)	timestamp-millis	da 2(
datetime (sybase)		datetime(3)	timestamp (3)	varchar (100)	timestamp-millis	da 2(
datetime		datetime(3)	timestamp (3)	varchar (100)	timestamp-millis	da
datetime2	prec_sec=0	datetime(0)	timestamp (0)	varchar (100)	timestamp-millis	da 2(

integer	Native	int
mediumint	Native	mediumint
mediumint unsigned	Native	mediumint unsigned
numeric	Native	decimal
real	Native	double
smallint	Native	smallint
smallint unsigned	Native	smallint unsigned
tinyint	Native	tinyint signed
date	Native	ansidate
datetime	Native	datetime (mysql)
time	Native	time (mysql)
timestamp	Native	timestamp (mysql)
year	Native	smallint
binary	Native	binary
blob	Native	varbinary(max)
char	Native	char
char byte	Native	binary
json	Native	varchar(max)
longblob	Native	varbinary(max)
longtext	Native	varchar(max)
mediumblob	Native	varbinary(max)
mediumtext	Native	varchar(max)
text	Native	varchar(max)
tinyblob	Native	varbinary(max)
tinytext	Native	varchar(max)
varbinary	Native	varbinary
varchar	Native	varchar
geometry	Extended	<<geometry>>
geometrycollection	Extended	<<geometrycollection>>
linestring	Extended	<<linestring>>
multilinestring	Extended	<<multilinestring>>
multipoint	Extended	<<multipoint>>
multipolygon	Extended	<<multipolygon>>
point	Extended	<<point>>
polygon	Extended	<<polygon>>

Data Type	HVR Capture Support	HVR Catalog Data Type
bigint	Native	bigint
bigint unsigned	Native	bigint unsigned
bit	Native	bit (mysql)
bool	Native	tinyint signed
boolean	Native	tinyint signed
dec	Native	decimal
dec unsigned	Native	decimal
decimal	Native	decimal
decimal unsigned	Native	decimal
double	Native	double
double precision	Native	double
double precision unsigned	Native	double
double unsigned	Native	double
fixed	Native	decimal
fixed unsigned	Native	decimal
float	Native	real
float unsigned	Native	real
int	Native	int
int unsigned	Native	int unsigned
integer	Native	int
integer unsigned	Native	int unsigned
mediumint	Native	mediumint

datetimeoffset		datetime(0)	timestamp (0) with time zone	varchar (100)	timestamp-millis	da off
db2 timestamp with time zone	prec_sec=0	datetime(0)	timestamp (0) with time zone	varchar (100)	timestamp-millis	da off
db2 xml	encoding=UTF-8	longtext charset utf8mb4	text	text	text	va (m
dbclob		longtext charset utf8mb4	text	text	text	nv (m
decfloat	prec=16	varchar(42) charset utf8mb4	varchar(42)	varchar (42)	varchar (42)	va (4
decfloat	prec=34	varchar(42) charset utf8mb4	varchar(42)	varchar (42)	varchar (42)	va (4
decimal	prec=10scale=3	decimal (10,3)	numeric (10,3)	varchar (12)	decimal (10, 3)	nu (1
decimal	prec=6	decimal(6)	numeric(6)	varchar (8)	decimal (6)	nu (6

mediumint unsigned	Native	mediumint unsigned
numeric	Native	decimal
numeric unsigned	Native	decimal
real	Native	double
real unsigned	Native	double
smallint	Native	smallint
smallint unsigned	Native	smallint unsigned
tinyint	Native	tinyint signed
tinyint unsigned	Native	tinyint unsigned
date	Native	ansidate
datetime	Native	datetime (mysql)
time	Native	time (mysql)
timestamp	Native	timestamp (mysql)
year	Native	smallint
binary	Native	binary
blob	Native	varbinary (max)
char	Native	char
long varchar	Native	varchar(max)
longblob	Native	varbinary (max)
longtext	Native	varchar(max)
mediumblob	Native	varbinary (max)
mediumtext	Native	varchar(max)
long	Native	varchar(max)
text	Native	varchar(max)
tinyblob	Native	varbinary (max)
tinytext	Native	varchar(max)
varbinary	Native	varbinary
varchar	Native	varchar
enum	Extended	<<enum>>
set	Extended	<<set>>
geometry	Extended	<<geometry>>
geometrycollection	Extended	<<geometrycollection>>
linestring	Extended	<<linestring>>
multilinestring	Extended	<<multilinestring>>
multipoint	Extended	<<multipoint>>
multipolygon	Extended	<<multipolygon>>
point	Extended	<<point>>
polygon	Extended	<<polygon>>
json	Extended	<<json>>

Data Type	HVR Capture Support	HVR Catalog Data Type
binary_double	Native	binary_double
binary_float	Native	binary_float
float	Native	number
long	Native	long
number	Native	number
date	Native	date
interval day to second	Native	interval day to second
interval year to month	Native	interval year to month
timestamp	Native	timestamp (oracle)
timestamp with local time zone	Native	timestamp with local tz (oracle)
timestamp with time zone	Native	timestamp with tz (oracle)
char	Native	char (oracle)
nchar	Native	nchar (oracle)
nvarchar2	Native	nvarchar2
varchar	Native	varchar2

double	bytelen=8	double	float	float	float	flo
epoch		timestamp(0) null	timestamp (0)	bigint	timestamp-millis	da 2(
float	bytelen=8	double	float	float	float	flo
float4	bytelen=4	float	real	real	real	rea
float8	bytelen=8	double	float	float	float	flo
graphic	charlen=10	char(10) charset utf8mb4	char(10)	char (10)	char (10)	nc (1
image		longblob	bytea	blob	blob	va y(
image (sybase)		longblob	bytea	blob	blob	va y(
ingresdate		datetime(0)	timestamp (0)	varchar (100)	timestamp-millis	da 2(
int unsigned	bytelen=4	int unsigned	bigint	bigint	bigint	big
int	bytelen=4	int	integer	integer	integer	int
integer	bytelen=4	int	integer	integer	integer	int
integer1	bytelen=1	tinyint	smallint	smallint	smallint	sm
integer2	bytelen=2	smallint	smallint	smallint	smallint	sm

varchar2	Native	varchar2
long raw	Native	long raw
raw	Native	raw
bfile	Native	bfile
blob	Native	blob
clob	Native	clob
nclob	Native	nclob
rowid	Native	rowid
urowid	Native	urowid
extended data types	Extended	<<data type>>
ref	Extended	<<data type>>
varrays	Extended	<<data type>>
nested tables	Extended	<<data type>>
anytype	Extended	<<data type>>
anydata	Extended	<<data type>>
anydataset	Extended	<<data type>>
uri	Extended	<<data type>>
xmltype	Extended	<<xmltype>>
sdo_geometry	Extended	<<data type>>
sdo_georaster	Extended	<<data type>>
sdo_topo_geometry	Extended	<<data type>>
media	Extended	<<data type>>

Data Type	HVR Capture Support	HVR Catalog Data Type
smallint	Native	smallint
integer	Native	integer
int	Native	integer
bigint	Native	bigint
int2	Native	smallint
int4	Native	integer
int8	Native	bigint
decimal	Native	number
numeric	Native	number
double precision	Native	float
real	Native	real
smallserial	Native	smallint
serial2	Native	smallint
serial	Native	integer
bigserial	Native	bigint
serial8	Native	bigint
time	Native	time
time with time zone	Native	time with time zone
timestamp	Native	postgres timestamp
timestamp with time zone	Native	postgres timestamp with time zone
date	Native	postgres date
interval	Extended	<<interval>>
char / character	Native	char
varchar	Native	varchar or clob (if char length is large)
text	Native	clob
bytea	Native	blob
json	Native	json
jsonb	Native	jsonb
boolean	Native	boolean
money	Extended	<<money>>
point	Extended	<<point>>
line	Extended	<<line>>
lseg	Extended	<<_lseg>>
box	Extended	<<_box>>
path	Extended	<<path>>
polygon	Extended	<<polygon>>
circle	Extended	<<circle>>
cidr	Extended	<<cidr>>
inet	Extended	<<inet>>

integer4	bytelen=4	int	integer	integer	integer	int
integer8	bytelen=8	bigint	bigint	bigint	bigint	big
interval day to second (ingres)		datetime(0)	timestamp(0)	varchar(100)	interval month to second	da 2(
interval day to second		datetime(0)	timestamp(0)	varchar(100)	interval month to second	da 2(
interval month to second		datetime(0)	timestamp(0)	varchar(100)	interval month to second	da 2(
interval year to month (ingres)		datetime(0)	timestamp(6)	varchar(100)	interval month to second	da 2(
interval year to month		datetime(0)	timestamp(6)	varchar(100)	interval month to second	da 2(
json		longtext charset utf8mb4	json	text	text	va (m
jsonb		longtext charset utf8mb4	jsonb	text	text	va (m
long byte		longblob	bytea	blob	blob	va y(l

macaddr	Extended	<<macaddr>>
macaddr8	Extended	<<macaddr8>>
bit	Extended	<<bit>>
bit varying / varbit	Extended	<<varbit>>
tsvector	Extended	<<tsvector>>
tsquery	Extended	<<tsquery>>
uuid	Native[1]	uniqueidentifier[1]
xml	Extended	<<xml>>
int4range	Extended	<<int4range>>
int8range	Extended	<<int8range>>
numrange	Extended	<<numrange>>
tsrange	Extended	<<tsrange>>
tstzrange	Extended	<<tstzrange>>
daterange	Extended	<<daterange>>
oid	Extended	<<oid>>
pg_lsn	Extended	<<pg_lsn
domain	Extended	<<domain>>
user defined	Extended	<<user defined>>

[1] - Since HVR 5.7.5/2 and 5.7.0/12, the 'uuid' data type is supported as 'Native' and is mapped to the HVR Catalog Type 'uniqueidentifier'.

Data Type	HVR Capture Support	HVR Catalog Data Type
int	Native	int
bigint	Native	bigint
smallint	Native	smallint
tinyint	Native	tinyint
numeric	Native	numeric
decimal	Native	numeric
bit	Native	bit
money	Native	money
smallmoney	Native	smallmoney
float	Native	float
real	Native	real
date	Native	ansidate
datetime	Native	datetime
datetime2	Native	datetime(2)
datetimeoffset	Native	datetimeoffset
smalldatetime	Native	smalldatetime
time	Native	time
char	Native	char
varchar	Native	varchar
text	Native	text (sqlserver)
nchar	Native	nchar
nvarchar	Native	nvarchar
ntext	Native	ntext
binary	Native	binary
varbinary	Native	varbinary
image	Native	image
uniqueidentifier	Native	uniqueidentifier
xml	Native	xml
cursor	Extended	<<cursor>>
hierarchyid	Extended	<<hierarchyid>>
rowversion	Extended	<<rowversion>>
sql_variant	Extended	<<sql_variant>>
spatial geometry types	Extended	<<geometry>>
spatial geography types	Extended	<<geography>>
table	Extended	<<table>>

long char	encoding=UTF-8	longtext charset utf8mb4	char	text	text	va(m
long nvarchar (db2)		longtext charset utf8mb4	text	text	text	nv(m
long nvarchar		longtext charset utf8mb4	text	text	text	nv(m
long raw		longblob	bytea	blob	blob	va y(l
long varbinary		longblob	bytea	blob	blob	va y(l
long varchar (db2)	encoding=UTF-8	longtext charset utf8mb4	text	text	text	va(m
long varchar	encoding=UTF-8	longtext charset utf8mb4	text	text	text	va(m
long	encoding=UTF-8	longtext charset utf8mb4	text	text	text	va(m

mediumint unsigned	bytelen=3	mediumint unsigned	integer	integer	integer	int
mediumint	bytelen=3	mediumint	integer	integer	integer	int
money (ingres)		decimal (14,2)	numeric (14,2)	varchar (16)	decimal (14,2)	m
money		decimal (19,4)	numeric (19,4)	varchar (21)	decimal (19,4)	m
nchar	charlen=10	char (10) charset utf8mb4	char(10)	char (10)	char (10)	nc (1
nchar (oracle)	charlen=10	char (10) charset utf8mb4	char(10)	char (10)	char (10)	nc (1
nclob		longtext charset utf8mb4	text	text	text	nv (m
ntext		longtext charset utf8mb4	text	text	text	nv (m
number		decimal (65,4)	numeric	float	decimal (1000,4)	nu (3
number	prec=10scale=-127	double	float	float	float	flo

number	prec=10scale=3	decimal(10,3)	numeric(10,3)	float	decimal(10,3)	nu(1
number	prec=26	decimal(26)	numeric(26)	float	decimal(26)	nu(2
number	prec=6	int	numeric(6)	integer	integer	int
numeric (db2i)	prec=10scale=3	decimal(10,3)	numeric(10,3)	varchar(12)	decimal(10,3)	nu(1
numeric (db2i)	prec=6	decimal(6)	numeric(6)	varchar(8)	decimal(6)	nu(6)
numeric	prec=10scale=3	decimal(10,3)	numeric(10,3)	varchar(12)	decimal(10,3)	nu(1
numeric	prec=26	decimal(26)	numeric(26)	varchar(28)	decimal(26)	nu(2
numeric	prec=6	decimal(6)	numeric(6)	varchar(8)	decimal(6)	nu(6)
nvarchar	charlen=10	varchar(10) charset utf8mb4	varchar(10)	varchar(10)	varchar(10)	nv(1
nvarchar(max)		longtext charset utf8mb4	text	text	text	nv(m

nvarchar2	charlen=10	varchar(10) charset utf8mb4	varchar(10)	varchar(10)	varchar(10)	nv(1
postgres date		date	date	bigint	date	da
postgres timestamp with time zone		datetime(0)	timestamp(0) with time zone	varchar(100)	timestamp-millis	da off
postgres timestamp		datetime(0)	timestamp(0)	varchar(100)	timestamp-millis	da 2(
raw	bytelen=10	varbinary(10)	bytea	blob	blob	va y(1
real	bytelen=4	float	real	real	real	re
rowid		char(18) charset ascii	char(18)	char(18)	char(18)	ch
rowversion	bytelen=10	binary(10)	bytea	blob	blob	bi (1
smalldatetime		datetime(0)	timestamp(0)	varchar(100)	timestamp-millis	sm eti
smallint unsigned	bytelen=2	smallint unsigned	integer	integer	integer	int
smallint	bytelen=2	smallint	smallint	smallint	smallint	sm

smallmoney		decim a l (10,4)	numeric (10,4)	varc har (12)	deci mal (10, 4)	sn ne
text (ingres)	bytelen=10enc oding=UTF-8	varcha r(10) charse t utf8mb4	varchar(10)	varc har (10)	varc har (10)	va (1
text (sqlserver)	encoding=WIN DOWS-1252	longte x t charse t latin1	text	text	text	va (m
text(sybase)	encoding=UTF -8	longte x t charse t utf8mb4	text	text	text	va (m
time (mysql)	prec_sec=0	time(0)	time(0)	inte ger	time (3)	tin
time (sybase)	prec_sec=0	time(3)	time(3)	inte ger	time (3)	tin
time with local time zone		time(0)	time(0)	inte ger	time (3)	tin
time with time zone		time(0)	time(0) with time zone	inte ger	time (3)	tin
time	prec_sec=0	time(0)	time(0)	inte ger	time (3)	tin

time	prec_sec=3	time(3)	time(3)	integer	time (3)	tim
time2		time(6)	time(0)	integer	time (3)	tim
timestamp (db2)	prec_sec=0	datetime(0)	timestamp (0)	varchar (100)	timestamp-millis	da 2(
timestamp (ingres)		datetime(0)	timestamp (0)	varchar (100)	timestamp-millis	da 2(
timestamp (mysql)		timestamp(0) null	timestamp (0)	bigint	timestamp-millis	da 2(
timestamp (oracle)		datetime(0)	timestamp (0)	varchar (100)	timestamp-millis	da 2(
timestamp (sqlserver)	bytelen=10	binary (10)	bytea	blob	blob	bi (1
timestamp (sybase)	bytelen=10	varbinary(10)	bytea	blob	blob	va y(
timestamp with local time zone		datetime(0)	timestamp (0)	varchar (100)	timestamp-millis	da 2(

timestamp with local tz (oracle)		datetime(0)	timestamp(0)	varchar(100)	timestamp-millis	da 2(
timestamp with time zone		datetime(0)	timestamp(0) with time zone	varchar(100)	timestamp-millis	da off
timestamp with tz (oracle)		datetime(0)	timestamp(0) with time zone	varchar(100)	timestamp-millis	da off
timestamp	prec_sec=0	datetime(0)	timestamp(0)	varchar(100)	timestamp-millis	da 2(
tinyint signed	bytelen=1	tinyint	smallint	smallint	smallint	sm
tinyint unsigned	bytelen=1	tinyint unsigned	smallint	smallint	smallint	tin
tinyint	bytelen=1	tinyint unsigned	smallint	smallint	smallint	tin
uniqueidentifier		binary(16)	uuid	char(36)	char(36)	un en
unitext		longtext charset utf8mb4	text	text	text	nv (m

univarchar	charlen=10	varchar(10) charset utf8mb4	varchar(10)	varchar (10)	varchar (10)	nv (1
unsigned bigint	bytelen=8	bigint unsigned	numeric(20)	float	decimal (20)	nu (2
unsigned int	bytelen=4	int unsigned	bigint	bigint	bigint	big
unsigned smallint	bytelen=2	smallint unsigned	integer	integer	integer	int
urowid	bytelen=100	varchar(100) charset ascii	varchar (100)	varchar (100)	varchar (100)	va (1
varbinary	bytelen=10	varbinary(10)	bytea	blob	blob	va y(
varbinary (sybase)	bytelen=10	varbinary(10)	bytea	blob	blob	va y(
varbinary(max)		longblob	bytea	blob	blob	va y(
varbyte	bytelen=10	varbinary(10)	bytea	blob	blob	va y(
varchar	bytelen=10encoding=UTF-8	varchar(10) charset utf8mb4	varchar(10)	varchar (10)	varchar (10)	va (1
varchar	bytelen=8000encoding=UTF-8	varchar (8000) charset utf8mb4	varchar (8000)	varchar (8000)	varchar (8000)	va (8

varchar	bytelen=40charlen=10encoding=UTF-8	varchar(10) charset utf8mb4	varchar(10)	varchar (10)	varchar (10)	va (1
varchar	bytelen=10encoding=WINDOWS-1252	varchar(10) charset latin1	varchar(10)	varchar (10)	varchar (10)	va (1
varchar (sybase)	bytelen=10encoding=UTF-8	varchar(10) charset utf8mb4	varchar(10)	varchar (10)	varchar (10)	va (1
varchar (sybase)	bytelen=8000encoding=UTF-8	varchar (8000) charset utf8mb4	varchar (8000)	varchar (8000)	varchar (8000)	va (8
varchar (sybase)	bytelen=40charlen=10encoding=UTF-8	varchar(10) charset utf8mb4	varchar(10)	varchar (10)	varchar (10)	va (1
varchar (sybase)	bytelen=10encoding=WINDOWS-1252	varchar(10) charset latin1	varchar(10)	varchar (10)	varchar (10)	va (1
varchar(max)	encoding=WINDOWS-1252	longtex t charset latin1	text	text	text	va (m
varchar2	bytelen=10encoding=UTF-8	varchar(10) charset utf8mb4	varchar(10)	varchar (10)	varchar (10)	va (1

varchar2	bytelen=40charlen=10encoding=UTF-8	varchar(10) charset utf8mb4	varchar(10)	varchar(10)	varchar(10)	va(1
varchar2	bytelen=10encoding=WINDOWS-1252	varchar(10) charset latin1	varchar(10)	varchar(10)	varchar(10)	va(1
vargraphic	charlen=10	varchar(10) charset utf8mb4	varchar(10)	varchar(10)	varchar(10)	nv(1
xml		longtext charset utf8mb4	text	text	text	nv(m