MS ACCESS DATABASE DATA TYPES

Data Type	Use For	Size
Text	Text or combinations of text and numbers, such as addresses. Also numbers that do not require calculations, such as phone numbers, part numbers, or postal codes.	Up to 255 characters. Microsoft Access only stores the characters entered in a field; it does not store space characters for unused positions in a Text field. To control the maximum number of characters that can be entered, set the FieldSize property.
Memo	Lengthy text and numbers, such as notes or descriptions.	Up to 64,000 characters.
Number	Numeric data to be used for mathematical calculations, except calculations involving money (use Currency type). Set the FieldSize property to define the specific Number type.	1, 2, 4, or 8 bytes. 16 bytes for Replication ID (GUID) only.
Date/Time	Dates and times.	8 bytes.
Currency	Currency values. Use the Currency data type to prevent rounding off during calculations. Accurate to 15 digits to the left of the decimal point and 4 digits to the right.	8 bytes.
AutoNumber	Unique sequential (incrementing by 1) or random numbers automatically inserted when a record is added.	4 bytes. 16 bytes for Replication ID (GUID) only.
Yes/No	Fields that will contain only one of two values, such as Yes/No, True/False, On/Off.	1 bit.
OLE Object	Objects (such as Microsoft Word documents, Microsoft Excel spreadsheets, pictures, sounds, or other binary data), created in other programs using the OLE protocol, that can be linked to or embedded in a Microsoft Access table. You must use a bound object frame in a form or report to display the OLE object.	Up to 1 gigabyte (limited by disk space).
Hyperlink	Field that will store hyperlinks. A hyperlink can be a UNC path or a URL.	Up to 64,000 characters.
Lookup Wizard	Creates a field that allows you to choose a value from another table or from a list of values using a combo box. Choosing this option in the data type list starts a wizard to define this for you.	The same size as the primary key field that is also the Lookup field; typically 4 bytes.

For numeric data types, the field size enables you to further define the type of number, which in turn determines the storage size. The table below shows the seven possible settings in the Numeric Field Size property.
You should make the field size the smallest one possible; Access runs faster with smaller field sizes. Note the first three settings don't use decimal points, but allow increasingly larger positive or negative numbers. Single and Double permit even larger numbers: Single gives you 7 decimal places, and Double allows 15. Use the Double setting when you need many decimal places or very large numbers.

Numeric Field Size Properties

Setting	Description	Decimal Precision	Storage Size
Byte	Stores numbers from 0 to 255 (no fractions).	None	1 byte
Integer	Stores numbers from –32,768 to 32,767 (no fractions).	None	2 bytes
Long Integer	(Default) Stores numbers from –2,147,483,648 to 2,147,483,647 (no fractions).	None	4 bytes
Decimal	Stores numbers from-10^28 -1 through 10^28 -1	28	12 bytes
Single	Stores numbers from –3.402823E38 to –1.401298E–45 for negative values and from 1.401298E–45 to 3.402823E38 for positive values.	7	4 bytes
Double	Stores numbers from –1.79769313486231E308 to –4.94065645841247E–324 for negative values and from 1.79769313486231E308 to 4.94065645841247E–324 for positive values.	15	8 bytes
Replication ID	Globally unique identifier (GUID)	N/A	16 bytes

Caution: If you convert a large FieldSize setting to a smaller one in a field that already contains data, you might lose data. For example, if you change the FieldSize setting for a Text data type field from 255 to 50, data beyond the new 50-character setting will be discarded.
If the data in a Number data type field doesn't fit in a new FieldSize setting, fractional

MYSQL DATA TYPES

TEXT TYPES	
CHAR( )	A fixed section from 0 to 255 characters long.
VARCHAR( )	A variable section from 0 to 255 characters long.
TINYTEXT	A string with a maximum length of 255 characters.
TEXT	A string with a maximum length of 65535 characters.
BLOB	A string with a maximum length of 65535 characters.
MEDIUMTEXT	A string with a maximum length of 16777215 characters.
MEDIUMBLOB	A string with a maximum length of 16777215 characters.
LONGTEXT	A string with a maximum length of 4294967295 characters.
LONGBLOB	A string with a maximum length of 4294967295 characters.

NUMBER TYPES	
TINYINT( )	-128 to 127 normal 0 to 255 UNSIGNED.
SMALLINT( )	-32768 to 32767 normal 0 to 65535 UNSIGNED.
MEDIUMINT( )	-8388608 to 8388607 normal 0 to 16777215 UNSIGNED.
INT( )	-2147483648 to 2147483647 normal 0 to 4294967295 UNSIGNED.
BIGINT( )	-9223372036854775808 to 9223372036854775807 normal 0 to 18446744073709551615 UNSIGNED.
FLOAT	A small number with a floating decimal point.
DOUBLE(,)	A large number with a floating decimal point.
DECIMAL(,)	A DOUBLE stored as a string , allowing for a fixed decimal point.

DATE TYPES	
DATE	YYYY-MM-DD.
DATETIME	YYYY-MM-DD HH:MM:SS.
TIMESTAMP	YYYYMMDDHHMMSS.
TIME	HH:MM:SS.

MISC TYPES	
ENUM ( )	Short for ENUMERATION which means that each column may have one of a specified possible values.
SET	Similar to ENUM except each column may have more than one of the specified possible values.

ENUM is short for ENUMERATED list. This column can only store one of the values that are declared in the specified list contained in the ( ) brackets.

ENUM('y','n')

You can list up to 65535 values in an ENUM list. If a value is inserted that is not in the list, a blank value will be inserted.

SET is similar to ENUM except SET may contain up to 64 list items and can store more than one choice.

Access Data Type	MySQL Equivalent Type	Limits
Auto-Number - Long Integer	Int with auto-increment enabled	-2147483648 to 2147483647 normal 0 to 4294967295 UNSIGNED
Auto-Number - ReplicationID	CHAR(36)[1]	N/A
Text	VARCHAR or TINYTEXT	255 characters
Memo	Text	65535 characters
Number - Byte	TinyInt	-128 to 127 normal 0 to 255 UNSIGNED.
Number - Integer	SMALLINT	-32768 to 32767 normal 0 to 65535 UNSIGNED.
Number - Long Integer	INT	-2147483648 to 2147483647 normal 0 to 4294967295 UNSIGNED
Number - Single	FLOAT?	7-digits
Number - Double	DOUBLE	15 digits
Number - Replication ID	CHAR(36)[1]	N/A
Number - Decimal	DECIMAL	?
Date/Time	DATETIME	N/A
Currency	N/A	?
Yes/No	N/A[2]	0 or 1
OLE Object	BLOB	?[3]
Hyperlink	N/A[4]	?

Some notes about the table above:

1. VARCHAR(36) is not exactly equivalent to the proprietary Replication ID (or UUID or GUID) format. A Replication ID is actually a 128-bit, 36-character unique identifier. CHAR(36) is the closest you can currently get with MySQL.
2. There is no MySQL equivalent for the Access “Yes/No” data type, unfortunately. The closest you can get is to set a TINYINT(1) data type and use your backend scripting language to restrict it to strictly 0 and 1.
3. The BLOB, MEDIUMBLOB and LONGBLOB data types in MySQL are much more efficient and versatile than the OLE Object data type in Access.
4. There is no MySQL equivalent to the Access “Hyperlink” data type. You don’t really need one, either. The closest you can get is the VARCHAR data type.
5. **Very Important** - As you can see in the table above, the “Text” data type has a limit of 255 characters. It is not equivalent to the “TEXT” data type in MySQL.
   The “Memo” field, is equivalent to the “TEXT” data type in MySQL. There is no Access equivalent to the MySQL “LONGTEXT” data type.

Some other notes

Please be aware, if you are brand new to working with Access, that there are some major differences in the use of SQL statements, as well.

- In MySQL, you should always wrap your database entity names (table name, column name, etc.) in accent marks (`). In Access, as long as there are no spaces in the entity name, no wrappers are required. However, if there is a space in the name of your entity (yes, that’s allowed in Access), then it needs to be wrapped by square brackets ([Your Name]).

- In MySQL, all data can be wrapped with single quotes, no matter its type. In Access, only strings can be wrapped in single quotes. Numbers need to be sent through with no wrappers, and dates **must** be wrapped by pound/number symbols (#).

OPERATORS AND MORE

Operator	Meaning
=	Equals
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to
!=	Not equal to
IS NOT NULL	Has a value
IS NULL	Does not have a value
BETWEEN	Within a range
NOT BETWEEN	Outside of a range
OR	Where one of two conditionals is true
AND	Where both conditionals are true
NOT	Where the condition is not true
LIKE	For character matching (i.e., LIKE 'Stanfo%')
NOT LIKE	To rule out characters