

1 Tabulky tvorba

1.1.1 Syntaxe CREATE TABLE

```
CREATE [ TEMPORARY ] TABLE [IF NOT EXISTS] table_name
(
  column1 datatype [ NULL | NOT NULL ]
                    [ DEFAULT default_value ]
                    [ AUTO_INCREMENT ]
                    [ UNIQUE KEY | PRIMARY KEY ]
                    [ COMMENT 'string' ],

  column2 datatype [ NULL | NOT NULL ]
                    [ DEFAULT default_value ]
                    [ AUTO_INCREMENT ]
                    [ UNIQUE KEY | PRIMARY KEY ]
                    [ COMMENT 'string' ],

  ...

  | [CONSTRAINT [constraint_name]] PRIMARY KEY [ USING BTREE | HASH ]
(index_col_name, ...)

  | [INDEX | KEY] index_name [ USING BTREE | HASH ] (index_col_name, ...)

  | [CONSTRAINT [constraint_name]] UNIQUE [ INDEX | KEY ]
  [ index_name ] [ USING BTREE | HASH ] (index_col_name, ...)

  | {FULLTEXT | SPATIAL} [ INDEX | KEY] index_name (index_col_name, ...)

  | [CONSTRAINT [constraint_name]]
    FOREIGN KEY index_name (index_col_name, ...)
    REFERENCES another_table_name (index_col_name, ...)
    [ MATCH FULL | MATCH PARTIAL | MATCH SIMPLE ]
    [ ON DELETE { RESTRICT | CASCADE | SET NULL | NO ACTION } ]
    [ ON UPDATE { RESTRICT | CASCADE | SET NULL | NO ACTION } ]

  | CHECK (expression)

  {ENGINE | TYPE} = engine_name
  | AUTO_INCREMENT = value
  | AVG_ROW_LENGTH = value
  | [DEFAULT] CHARACTER SET = charset_name
  | CHECKSUM = {0 | 1}
  | [DEFAULT] COLLATE = collation_name
  | COMMENT = 'string'
  | DATA DIRECTORY = 'absolute path'
  | DELAY_KEY_WRITE = { 0 | 1 }
  | INDEX DIRECTORY = 'absolute path'
  | INSERT_METHOD = { NO | FIRST | LAST }
  | MAX_ROWS = value
  | MIN_ROWS = value
  | PACK_KEYS = {0 | 1 | DEFAULT}
  | PASSWORD = 'string'
  | RAID_TYPE = { 1 | STRIPED | RAID0 }
    RAID_CHUNKS = value
    RAID_CHUNKSIZE = value
  | ROW_FORMAT = {DEFAULT | DYNAMIC | FIXED | COMPRESSED}
  | UNION = (table1, ... )
);
```

1.1.1 Syntaxe ALTER TABLE

```

ALTER TABLE tbl_name
    [alter_specification [, alter_specification] ...]
    [partition_options]
alter_specification:
    table_options
| ADD [COLUMN] col_name column_definition
    [FIRST | AFTER col_name ]
| ADD [COLUMN] (col_name column_definition,...)
| ADD {INDEX|KEY} [index_name]
    [index_type] (index_col_name,...) [index_option] ...
| ADD [CONSTRAINT [symbol]] PRIMARY KEY
    [index_type] (index_col_name,...) [index_option] ...
| ADD [CONSTRAINT [symbol]]
    UNIQUE [INDEX|KEY] [index_name]
    [index_type] (index_col_name,...) [index_option] ...
| ADD FULLTEXT [INDEX|KEY] [index_name]
    (index_col_name,...) [index_option] ...
| ADD SPATIAL [INDEX|KEY] [index_name]
    (index_col_name,...) [index_option] ...
| ADD [CONSTRAINT [symbol]]
    FOREIGN KEY [index_name] (index_col_name,...)
    reference_definition
| ALGORITHM [=] {DEFAULT|INPLACE|COPY}
| ALTER [COLUMN] col_name {SET DEFAULT literal | DROP DEFAULT}
| CHANGE [COLUMN] old_col_name new_col_name column_definition
    [FIRST|AFTER col_name]
| LOCK [=] {DEFAULT|NONE|SHARED|EXCLUSIVE}
| MODIFY [COLUMN] col_name column_definition
    [FIRST | AFTER col_name]
| DROP [COLUMN] col_name
| DROP PRIMARY KEY
| DROP {INDEX|KEY} index_name
| DROP FOREIGN KEY fk_symbol
| DISABLE KEYS
| ENABLE KEYS
| RENAME [TO|AS] new_tbl_name
| RENAME {INDEX|KEY} old_index_name TO new_index_name
| ORDER BY col_name [, col_name] ...
| CONVERT TO CHARACTER SET charset_name [COLLATE collation_name]
| [DEFAULT] CHARACTER SET [=] charset_name [COLLATE [=] collation_name]
| DISCARD TABLESPACE
| IMPORT TABLESPACE
| FORCE
| {WITHOUT|WITH} VALIDATION
| ADD PARTITION (partition_definition)
| DROP PARTITION partition_names
| DISCARD PARTITION {partition_names | ALL} TABLESPACE
| IMPORT PARTITION {partition_names | ALL} TABLESPACE
| TRUNCATE PARTITION {partition_names | ALL}
| COALESCE PARTITION number
| REORGANIZE PARTITION partition_names INTO (partition_definitions)
| EXCHANGE PARTITION partition_name WITH TABLE tbl_name [{WITH|WITHOUT} VALIDATION]
| ANALYZE PARTITION {partition_names | ALL}
| CHECK PARTITION {partition_names | ALL}
| OPTIMIZE PARTITION {partition_names | ALL}
| REBUILD PARTITION {partition_names | ALL}
| REPAIR PARTITION {partition_names | ALL}
| REMOVE PARTITIONING
| UPGRADE PARTITIONING
index_col_name:
    col_name [(length)] [ASC | DESC]
index_type:
    USING {BTREE | HASH}
index_option:
    KEY_BLOCK_SIZE [=] value
| index_type
| WITH PARSER parser_name
| COMMENT 'string'
table_options:
    table_option [[,] table_option] ... (see CREATE TABLE options)
partition_options:
    (see CREATE TABLE options)

```

1.1.2 Datové typy sloupců

Datový typ	Délka	Popis
TINYINT	1 bajt	Rozsah od -128 do +127 (bez znaménka: od 0 do 255)
SMALLINT	2 bajty	Rozsah od -32 768 do +32 767 (bez znaménka: od 0 do 65 535)
MEDIUMINT	3 bajty	215)
INT, INTEGER	4 bajty	4 294 967 295)
BIGINT	8 bajty	Rozsah od -9 223 372 036 854 775 808 do +9 223 372 036 854 775 807 (bez znaménka: od 0 do 18 446 744 073 709 551 615)
BIT, BOOL	1 bajt	synonymum pro typ TINYINT(1)
FLOAT		Rozsah od -3.402823466E+38 do +3.402823466E+38
DOUBLE		+1.7976931348623157E+308
DOUBLE PRECISION, REAL		synonymum pro typ DOUBLE
DECIMAL(m,d), DEC(m,d), NUMERIC(m,d)		rozsah nastaví parametry "m" a "d", maximální rozsah je stejný jak pro typ DOUBLE
DATE	'0000-00-00'	datum ve formátu "rok-měsíc-den" respektive "RRRR-MM-DD". Rozsah od 1000-01-01 do 9999-12-31
DATETIME	'0000-00-00 00:00:00'	datum a čas. Rozsah od 1000-01-01 00:00:00 do 9999-12-31 23:59:59 (format = "RRRR-MM-DD HH:MM:SS")
TIMESTAMP(n)	'0000-00-00 00:00:00'	- datum a čas v rozsahu od 1970-01-01 00:00:00 do 2037-01-01 00:00:00 (vždy se ukládá všech 14 čísel !)
		- formát zobrazení (a pro dotazy) lze nastavit parametrem "m" s hodnotou 14 (nebo chybějící), 12, 10, 8, 6, 4, nebo 2
		- "RRRRMMDDHHMMSS", "RRMMDDHHMMSS", "RRMMDDHHMM", "RRRRMMDD", "RRMMDD", "YMM", "YY"
		- pokud do buňky tohoto typu nic nezapišeme, pak MySQL sám doplní aktuální čas změny v daném řádku
TIME	'00:00:00'	"HH:MM:SS")
YEAR(n)		0 od 1970 do 2069
CHAR(m)		- délka řetězce "m" může být v rozsahu od 0 do 255
		- pokud je vložený řetězec kratší než nastavíme, pak jsou chybějící znaky automaticky doplněny mezerami (má tedy "pevnou" délku)
		- CHAR (bez "m") je považováno za CHAR(1)
VARCHAR(m)		- délka řetězce "m" může být v rozsahu od 0 do 255
		nedoplňují (má tedy "plovoucí" velikost), ale navíc se ukládá informace o jeho délce
TINYBLOB, TINYTEXT		Délka řetězce je max. 255 znaků
BLOB, TEXT		Délka řetězce je max. 65 535 znaků
MEDIUMBLOB, MEDIUMTEXT		Délka řetězce je max. 16 777 215 znaků
LOB, LONGTEXT		Délka řetězce je max. 4 294 967 295 znaků
ENUM('item1','item2',...)		- pole předem definovaných řetězců (itemů) o max. počtu 65 535
		- v buňce tabulky pak může být pouze jeden z itemů, které jsou předdefinované
		- místo názvů 'item' můžeme používat i jejich pořadí, tedy: 1 (místo 'item1'), 2 (místo 'item2')...
SET('item1','item2',...)		- pole předem definovaných řetězců (itemů) o max. počtu 64
		- v buňce tabulky pak může být i více z itemů, které jsou předdefinované

1.1.3 Omezení

1.1.3.1 Integritiní

- **UNIQUE** -
- **DEFAULT vychozi_hodnota** -

1.1.3.2 Ostatní

- **AUTO_INCREMENT**-
- **BINARY** -
- **FULLTEXT INDEX** -
- **INDEX** –
- **NOT NULL** -
- **NULL**
- **PRIMARY KEY** -
- **UNSIGNED**-
- **ZEROFILL** – podpora končí

1.1.4 Typy databázi

Pro pokročilejší správu tabulek (například restrikce) musíte změnit typ tabulky

- **INNODB** - uzamykání tabulky je vykonáváno na úrovni řádků; před použitím je nutná kompilace MySQL s podporou INNODB
- **MYISAM** - standard MySQL od verze 3.23.0
- **ISAM** - standardní typ tabulky ve starších databázích; dnes nahrazen typem MYISAM
- **ERGE** - formát vhodný pro spojení MYISAM tabulek se stejně nadefinovanými poli
- **HEAP** - tabulka tohoto typu je uložena pouze v paměti (může být velmi rychlá), má ale řadu omezení
- **BDB** - typ tabulky podobný INNODB; zatím ve fázi testování

1.1.5 Restrikce

- **UPDATE** – změna záznamu
- **DELETE** – smazání záznamu
- **UPDATE**
 - **ON UPDATE RESTRICT** - cizí klíč se nemůže změnit na požadavek změny odkazu, tudíž se změna neprovede. Neboli nemůžeme změnit Id u zákazníka.
 - **ON UPDATE CASCADE** – pokud se změní tak se změní i odkazy. Tj. pokud změníme Id u zákazníka tak se změní i u kontaktu
- **DELETE**
 - **ON DELETE CASCADE** - pokud cizí klíč ztratí referenci, tak se dany záznam se smaže
 - **ON DELETE RESTRICT** - záznam s cizím klíčem se nemůže smazat na požadavek odkazu, tudíž se odkaz ani záznam s cizím klíčem nesmaže.
 - **ON DELETE SET NULL** - pokud je odkaz smazán, tak se cizí klíč nastaví na NULL (pouze, pokud podle definice může být sloupec NULL) , Např. při smazání „*mítnosti*“ dojde k „*uvolnění nábytku*“. Ten zůstane zachován, ale již nebude přiřazen k žádné místnosti.