ALTER TABLE

The ALTER TABLE command is used  to change the structure of an existing table. It helps to add or delete columns, create or destroy indexes, change the type of existing columns, rename columns or the table itself. It  can also be used to change the comment for the table and type of the table.

**Basic Examples**

Create a table testtable as shown below:

CREATE TABLE testtable (col1 INT(11), col2 VARCHAR(15));

* To rename the table from testtable to testset, use the following statement.

ALTER TABLE testtable RENAME testset;

* To change column col1 from INTEGER to TINYINT NOT NULL (leaving the name the same), and to change column b from VARCHAR(15) to CHAR(25) as well as renaming it from col2 to col3, the following statement can be used.

ALTER TABLE testset

MODIFY col1 TINYINT NOT NULL,

CHANGE col2 col3 VARCHAR(25);

* To add a new TIMESTAMP column named col4, the following statement can be used.

ALTER TABLE testset ADD col4 TIMESTAMP;

* To add an index on column col4 and a UNIQUE index on column col1, the following statement can be used.

ALTER TABLE testset ADD INDEX (col4), ADD UNIQUE (col1);

* To remove column col3 from the table w3r1, the following statement can be used.

ALTER TABLE testset DROP COLUMN col3;

* To add a new AUTO\_INCREMENT integer column named col3, the following statement can be used.

ALTER TABLE testset

ADD col3 INT UNSIGNED NOT NULL AUTO\_INCREMENT,

ADD PRIMARY KEY (col3);

Here in the above example, we indexed col3 (as a PRIMARY KEY) because AUTO\_INCREMENT columns must be indexed, and we declare col3 as NOT NULL because primary key columns cannot be NULL.

* To change the data type of col1 into BIGINT, the following statement can be used.

ALTER TABLE testset MODIFY col1 BIGINT;

* If you want to include the attributes UNSIGNED DEFAULT 1 and COMMENT 'test column', show the below statement -

ALTER TABLE testset

MODIFY col1 BIGINT UNSIGNED DEFAULT 1

COMMENT 'test column';

* To change the table default character set and all character columns (CHAR, VARCHAR, TEXT) to a new character set, use a statement like this:

ALTER TABLE testset CONVERT TO CHARACTER SET latin1;

**MySQL ALTER TABLE insert column FIRST**

Here is the structure of *newbook\_mast* table.

Sample Output:

+------------+--------------+------+-----+------------+-------+

| Field | Type | Null | Key | Default | Extra |

+------------+--------------+------+-----+------------+-------+

| book\_id | varchar(15) | NO | PRI | | |

| book\_name | varchar(50) | NO | | | |

| isbn\_no | varchar(15) | NO | | | |

| cate\_id | varchar(8) | NO | | | |

| aut\_id | varchar(8) | NO | | | |

| pub\_id | varchar(8) | NO | | | |

| dt\_of\_pub | date | NO | | 0000-00-00 | |

| pub\_lang | varchar(15) | YES | | NULL | |

| no\_page | decimal(5,0) | NO | | 0 | |

| book\_price | decimal(8,2) | NO | | 0.00 | |

+------------+--------------+------+-----+------------+-------+

10 rows in set (0.00 sec)

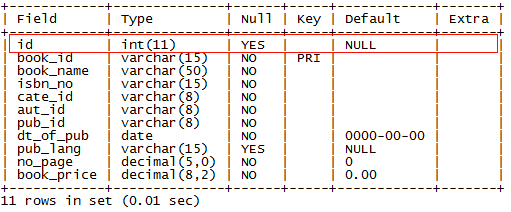
If you want to insert a column *id* of integer type, as first column of the table *newbook\_mast*, the following statement can be used.

ALTER TABLE newbook\_mast

ADD id INT FIRST ;

Copy

Here is the structure of the *newbook\_mast* after adding a column *id* at first.



**ALTER TABLE to insert column AFTER a column**

Here is the structure of *newbook\_mast* table.

+------------+--------------+------+-----+------------+-------+

| Field | Type | Null | Key | Default | Extra |

+------------+--------------+------+-----+------------+-------+

| book\_id | varchar(15) | NO | PRI | | |

| book\_name | varchar(50) | NO | | | |

| isbn\_no | varchar(15) | NO | | | |

| cate\_id | varchar(8) | NO | | | |

| aut\_id | varchar(8) | NO | | | |

| pub\_id | varchar(8) | NO | | | |

| dt\_of\_pub | date | NO | | 0000-00-00 | |

| pub\_lang | varchar(15) | YES | | NULL | |

| no\_page | decimal(5,0) | NO | | 0 | |

| book\_price | decimal(8,2) | NO | | 0.00 | |

+------------+--------------+------+-----+------------+-------+

10 rows in set (0.00 sec)

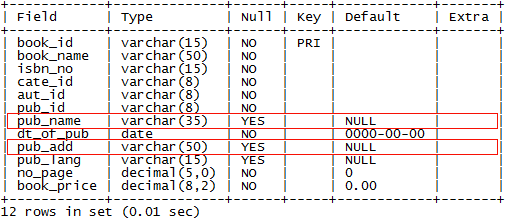
If you want to add two specific columns *pub\_name* and *pub\_add* after *pub\_id* and *dt\_of\_pub* columns respectively, the following statement can be used.

ALTER TABLE newbook\_mast

ADD pub\_name VARCHAR(35) AFTER pub\_id,

ADD pub\_add VARCHAR(50) AFTER dt\_of\_pub;

Here is the structure of the *newbook\_mast* after add two columns in specific position said above.



**ALTER TABLE ADD PRIMARY KEY**

Here is the structure of *tstpurch* table. The below figure shows that it has no primary key.

+-------------+---------------+------+-----+------------+-------+

| Field | Type | Null | Key | Default | Extra |

+-------------+---------------+------+-----+------------+-------+

| invoice\_no | varchar(12) | NO | | | |

| invoice\_dt | date | NO | | 0000-00-00 | |

| ord\_no | varchar(25) | NO | | NULL | |

| ord\_date | date | NO | | 0000-00-00 | |

| receive\_dt | date | NO | | 0000-00-00 | |

| book\_id | varchar(8) | NO | | | |

| book\_name | varchar(50) | NO | | | |

| pub\_lang | varchar(8) | YES | | NULL | |

| cate\_id | varchar(8) | YES | | NULL | |

| receive\_qty | int(5) | NO | | 0 | |

| purch\_price | decimal(12,2) | NO | | 0.00 | |

| total\_cost | decimal(12,2) | NO | | 0.00 | |

+-------------+---------------+------+-----+------------+-------+

12 rows in set (0.01 sec)

If you want to l create a PRIMARY KEY on *invoice\_no* column for the table *tstpurch*, the following statement can be used.

ALTER TABLE tstpurch

ADD PRIMARY KEY invoice\_no (invoice\_no);

Here is the primary key after adding a primary key named *invoice\_no* on *invoice\_no* column.

+-------------+---------------+------+-----+------------+-------+

| Field | Type | Null | Key | Default | Extra |

+-------------+---------------+------+-----+------------+-------+

| invoice\_no | varchar(12) | NO | **PRI** | | |

| invoice\_dt | date | NO | | 0000-00-00 | |

| ord\_no | varchar(25) | NO | | NULL | |

| ord\_date | date | NO | | 0000-00-00 | |

| receive\_dt | date | NO | | 0000-00-00 | |

| book\_id | varchar(8) | NO | | | |

| book\_name | varchar(50) | NO | | | |

| pub\_lang | varchar(8) | YES | | NULL | |

| cate\_id | varchar(8) | YES | | NULL | |

| receive\_qty | int(5) | NO | | 0 | |

| purch\_price | decimal(12,2) | NO | | 0.00 | |

| total\_cost | decimal(12,2) | NO | | 0.00 | |

+-------------+---------------+------+-----+------------+-------+

12 rows in set (0.01 sec)

Here is the details of the index.

**ALTER TABLE ADD FOREIGN KEY**

Here are the structure of*torder*table and *newbook\_mast* table.

Table: torder

+-----------+-------------+------+-----+------------+-------+

| Field | Type | Null | Key | Default | Extra |

+-----------+-------------+------+-----+------------+-------+

| ord\_no | varchar(15) | NO | | | |

| ord\_date | date | NO | | 0000-00-00 | |

| book\_id | varchar(15) | NO | PRI | | |

| book\_name | varchar(50) | NO | | | |

| cate\_id | varchar(8) | NO | PRI | | |

| pub\_lang | varchar(15) | NO | | | |

| ord\_qty | int(5) | NO | | 0 | |

+-----------+-------------+------+-----+------------+-------+

Table: newbook\_mast

+------------+--------------+------+-----+------------+-------+

| Field | Type | Null | Key | Default | Extra |

+------------+--------------+------+-----+------------+-------+

| book\_id | varchar(15) | NO | PRI | | |

| book\_name | varchar(50) | NO | | | |

| isbn\_no | varchar(15) | NO | | | |

| cate\_id | varchar(8) | NO | | | |

| aut\_id | varchar(8) | NO | | | |

| pub\_id | varchar(8) | NO | | | |

| dt\_of\_pub | date | NO | | 0000-00-00 | |

| pub\_lang | varchar(15) | YES | | NULL | |

| no\_page | decimal(5,0) | NO | | 0 | |

| book\_price | decimal(8,2) | NO | | 0.00 | |

+------------+--------------+------+-----+------------+-------+

10 rows in set (0.00 sec)

If you want to create a FOREIGN KEY with the combination of *book\_id* and *cate\_id* columns of *newbook\_mast* table with a reference from *torder* table, the following statement can be used.

ALTER TABLE newbook\_mast

ADD FOREIGN KEY(book\_id,cate\_id)

REFERENCES torder(book\_id,cateid);

**ALTER TABLE ADD and DROP column, INDEX, PRIMARY KEY and FOREIGN KEY**

Here is the structure and index of *tstpurch* and *torder* table.

Table: tstpurch

+-------------+---------------+------+-----+------------+-------+

| Field | Type | Null | Key | Default | Extra |

+-------------+---------------+------+-----+------------+-------+

| id | int(5) | YES | | NULL | |

| invoice\_no | varchar(12) | NO | PRI | | |

| invoice\_dt | date | NO | | 0000-00-00 | |

| ord\_no | varchar(25) | NO | MUL | NULL | |

| ord\_date | date | NO | | 0000-00-00 | |

| receive\_dt | date | NO | | 0000-00-00 | |

| book\_id | varchar(8) | NO | | | |

| book\_name | varchar(50) | NO | | | |

| pub\_lang | varchar(8) | YES | | NULL | |

| cate\_id | varchar(8) | YES | MUL | NULL | |

| receive\_qty | int(5) | NO | | 0 | |

| purch\_price | decimal(12,2) | NO | | 0.00 | |

| total\_cost | decimal(12,2) | NO | | 0.00 | |

+-------------+---------------+------+-----+------------+-------+

If you want to modifies the structure of 'tstpurch' table in the following manner -

1. Drop the 'id' column
2. existing primary key

following statement can be used:

ALTER TABLE tstpurch

DROP id,

DROP PRIMARY KEY,