CSE-221

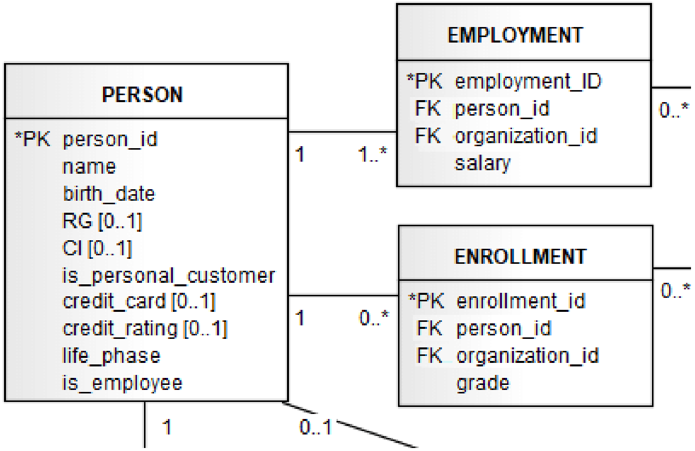
Database Management Systems

Structured Query Language - DDL

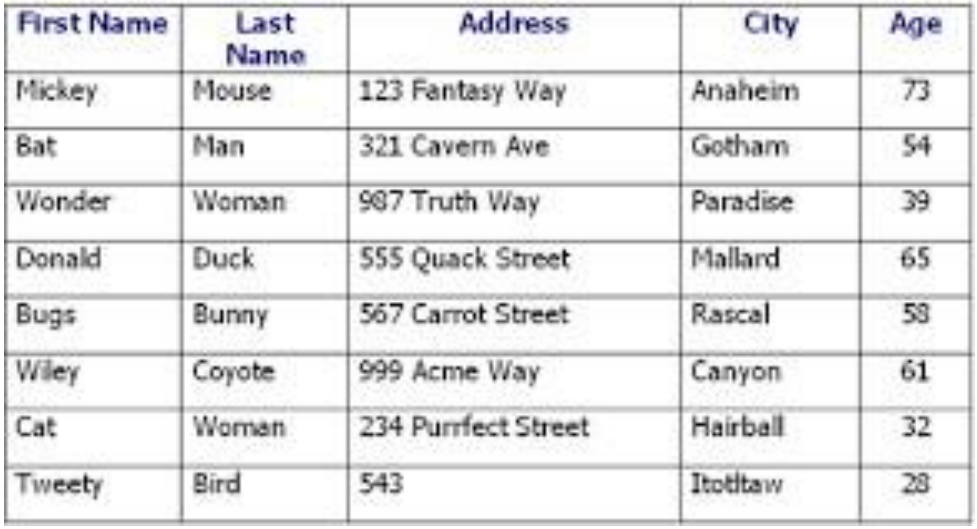
Md. Romizul Islam

romizul@cse.uiu.ac.bd

Schema vs. Relations



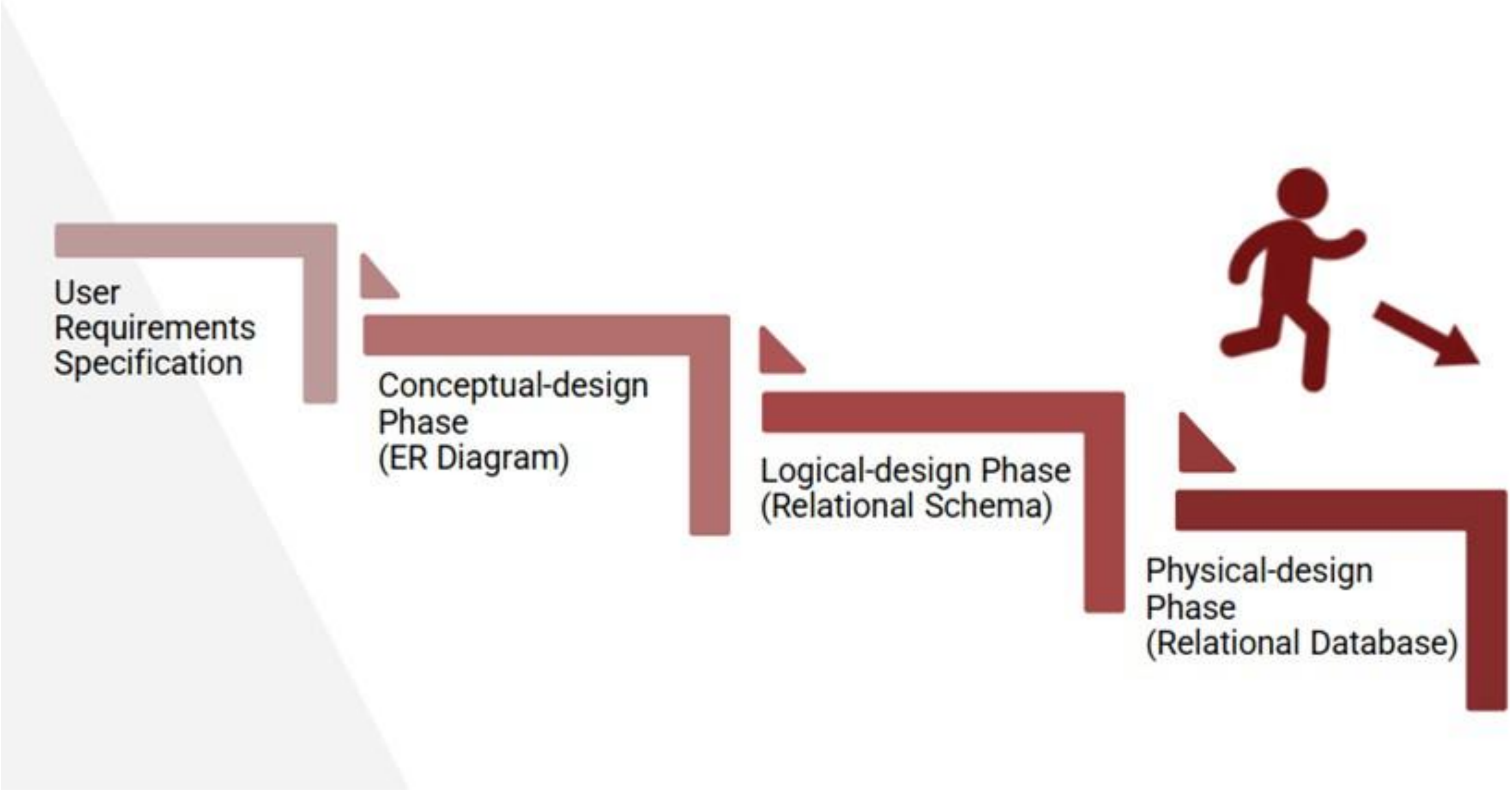
Schema contains design of table/schema



First Name	Last Name	Address	City	Age
Mickey	Mouse	123 Fantasy Way	Anaheim	73
Bat	Man	321 Cavern Ave	Gotham	54
Wonder	Woman	987 Truth Way	Paradise	39
Donald	Duck	555 Quack Street	Mallard	65
Bugs	Bunny	567 Carrot Street	Rascal	58
Wiley	Coyote	999 Acme Way	Canyon	61
Cat	Woman	234 Purrfect Street	Hairball	32
Tweety	Bird	543	Itotltaw	28

Relations contain actual data

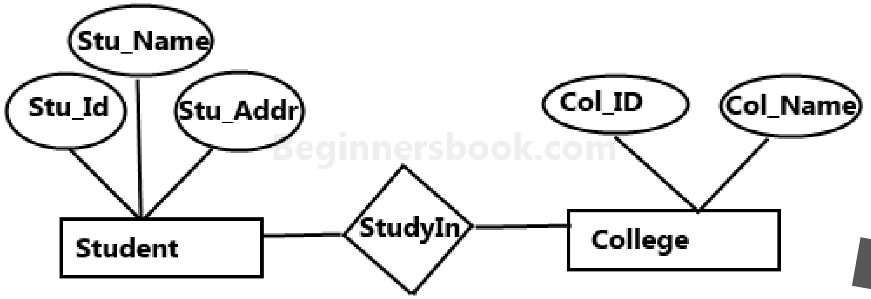
DB Design Steps



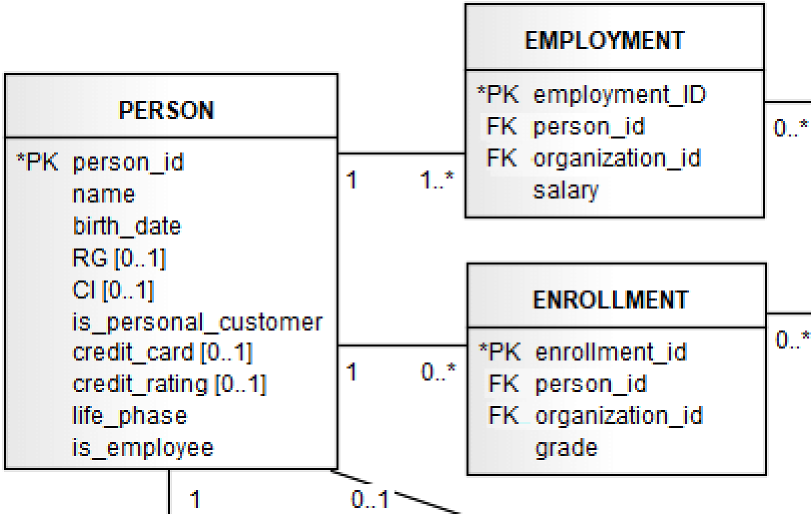
DB Design Steps

1. User Requirement Specification (Planning)

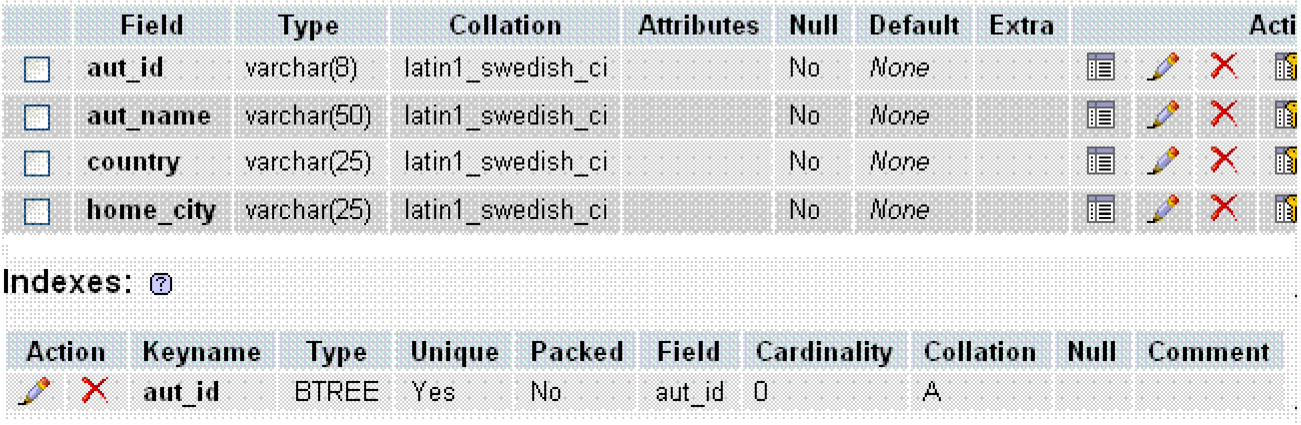
2. ER Diagram (Conceptual Design)



3. Relational Schema (Logical Design)



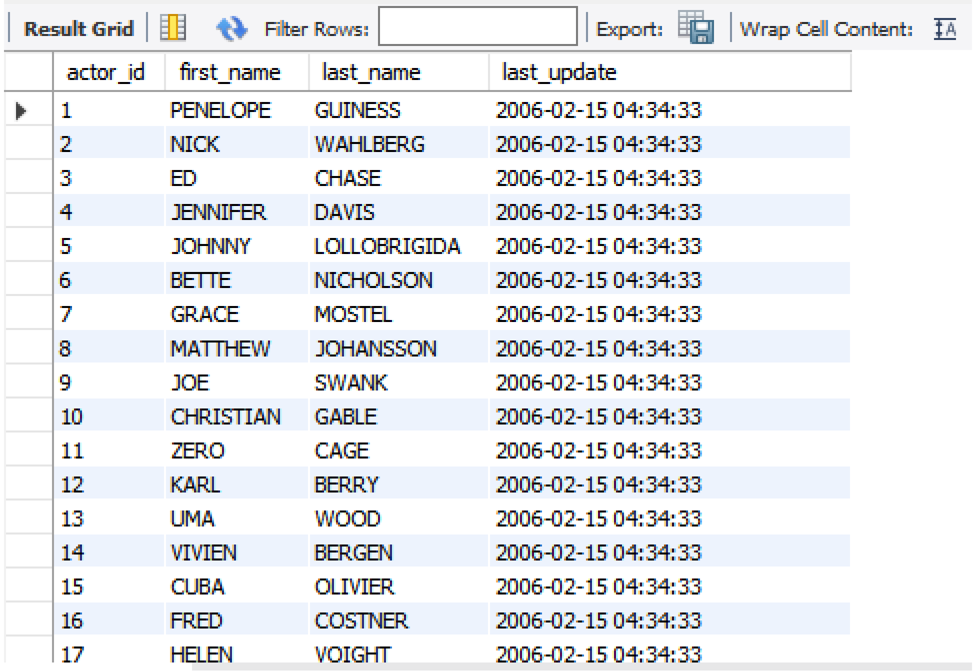
Example Relational Schema and Relations



	Field	Type	Collation	Attributes	Null	Default	Extra				Acti
□	**aut_id**	varchar(8)	latin1_swedish_ci		No	*None*		▤	✎	✗	▨
□	**aut_name**	varchar(50)	latin1_swedish_ci		No	*None*		▤	✎	✗	▨
□	**country**	varchar(25)	latin1_swedish_ci		No	*None*		▤	✎	✗	▨
□	**home_city**	varchar(25)	latin1_swedish_ci		No	*None*		▤	✎	✗	▨

Indexes: ⓘ

Action		Keyname	Type	Unique	Packed	Field	Cardinality	Collation	Null	Comment
✎	✗	**aut_id**	BTREE	Yes	No	aut_id	0	A		



	actor_id	first_name	last_name	last_update
▶	1	PENELOPE	GUINESS	2006-02-15 04:34:33
	2	NICK	WAHLBERG	2006-02-15 04:34:33
	3	ED	CHASE	2006-02-15 04:34:33
	4	JENNIFER	DAVIS	2006-02-15 04:34:33
	5	JOHNNY	LOLLOBRIGIDA	2006-02-15 04:34:33
	6	BETTE	NICHOLSON	2006-02-15 04:34:33
	7	GRACE	MOSTEL	2006-02-15 04:34:33
	8	MATTHEW	JOHANSSON	2006-02-15 04:34:33
	9	JOE	SWANK	2006-02-15 04:34:33
	10	CHRISTIAN	GABLE	2006-02-15 04:34:33
	11	ZERO	CAGE	2006-02-15 04:34:33
	12	KARL	BERRY	2006-02-15 04:34:33
	13	UMA	WOOD	2006-02-15 04:34:33
	14	VIVIEN	BERGEN	2006-02-15 04:34:33
	15	CUBA	OLIVIER	2006-02-15 04:34:33
	16	FRED	COSTNER	2006-02-15 04:34:33
	17	HELEN	VOIGHT	2006-02-15 04:34:33

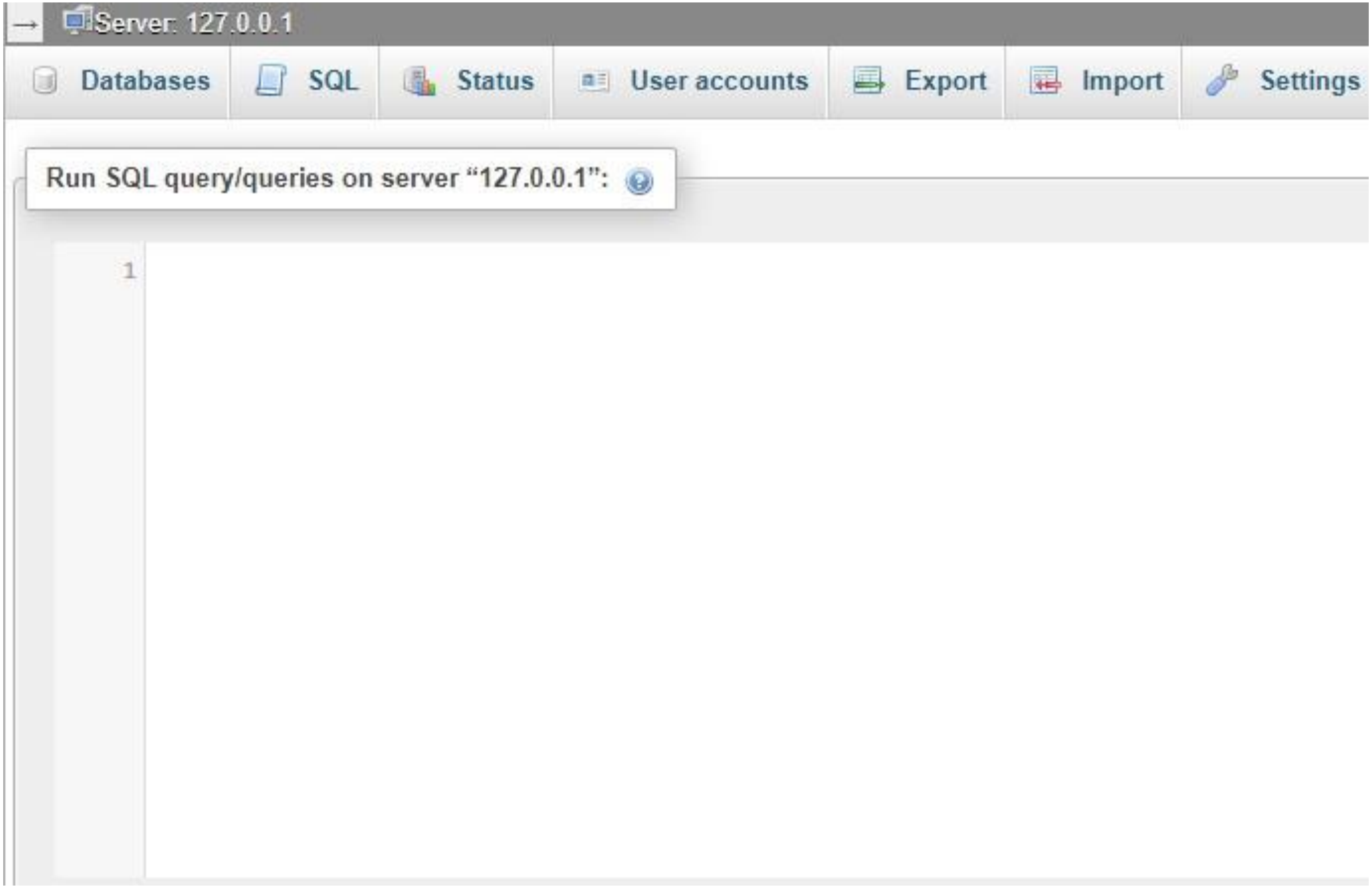
MySQL

- The most popular open source SQL database management system, is developed, distributed, and supported by Oracle Corporation.
- It is written in **C** and **C++**.
- It is named after co-founder Monty Widenius's daughter, **My**.
- The name of the MySQL Dolphin is **Sakila.**

How to Access MySQL?

- Command line
- GUI Tools (PHPMyAdmin, HeidiSQL, etc.)
- Any Programming Language

MySQL Query Execution



Run SQL query/queries on server “127.0.0.1”:

MySQL Comments

- From a # character to the end of the line.
- From a **- - <SPACE>** sequence to the end of the line.
- From a ***/* sequence to the following */*** sequence that is multiline comments.

DDL Statements

- CREATE, ALTER, DROP statements

Create & Delete Database Schema

1. **CREATE** DATABASE **[IF NOT EXISTS]** database_name**;**
   An error occurs if the database exists and you didn’t specify **IF NOT EXISTS.**

2. **DROP** DATABASE **[IF EXISTS]** database_name**;**
   - **IF EXISTS** is used to prevent an error from occurring if the database doesn’t exist.

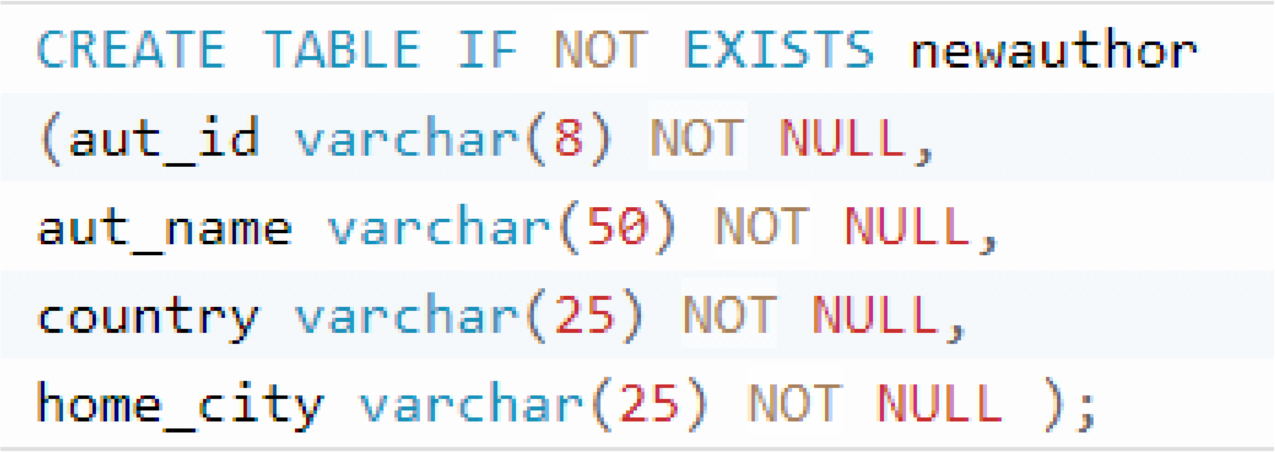
Create & Delete Database Schema

```
CREATE TABLE [IF NOT EXISTS] table_name(

 col1 datatype [NOT NULL] [DEFAULT def_val] [~~UNIQUE~~] [AUTO_INCREMENT] [~~PRIMARY KEY~~] [~~CHECK(expr)~~],
 col2 datatype [NOT NULL] [DEFAULT def_val] [~~UNIQUE~~] [AUTO_INCREMENT] [~~PRIMARY KEY~~] [~~CHECK(expr)~~],
 .
 .
 .
 coln datatype [NOT NULL] [DEFAULT def_val] [~~UNIQUE~~] [AUTO_INCREMENT] [~~PRIMARY KEY~~] [~~CHECK(expr)~~],

 CONSTRAINT constraint_name PRIMARY KEY(col1, col2, ... ),

 CONSTRAINT constraint_name UNIQUE(col3, col4, ... ),

 CONSTRAINT constraint_name CHECK(expr),

 CONSTRAINT constraint_name FOREIGN KEY(col1, col2, ... )
                            REFERENCES ref_tablename(ref_col1, ref_col2, ... )
                            [ON DELETE CASCADE|SET NULL|RESTRICT]
                            [ON UPDATE CASCADE|SET NULL|RESTRICT]
```

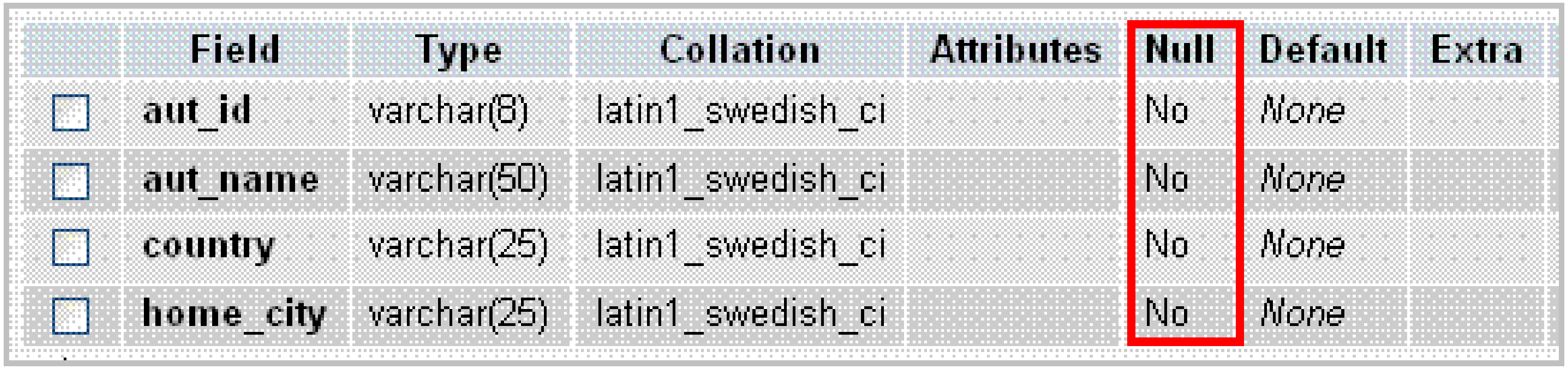
Note

- By default, tables are created in the default database, using the InnoDB storage engine.
- IF NOT EXISTS prevents an error from occurring if the table exists.
- If the constraint names are not defined, then MySQL automatically generates a constraint name.
- CASCADE: delete/update the child table matching rows when delete/update the parent table rows.
- SET NULL: sets the foreign key column to NULL when delete/update the parent table row.
- RESTRICT: rejects the delete/update operation for the parent table

Example (NULL)

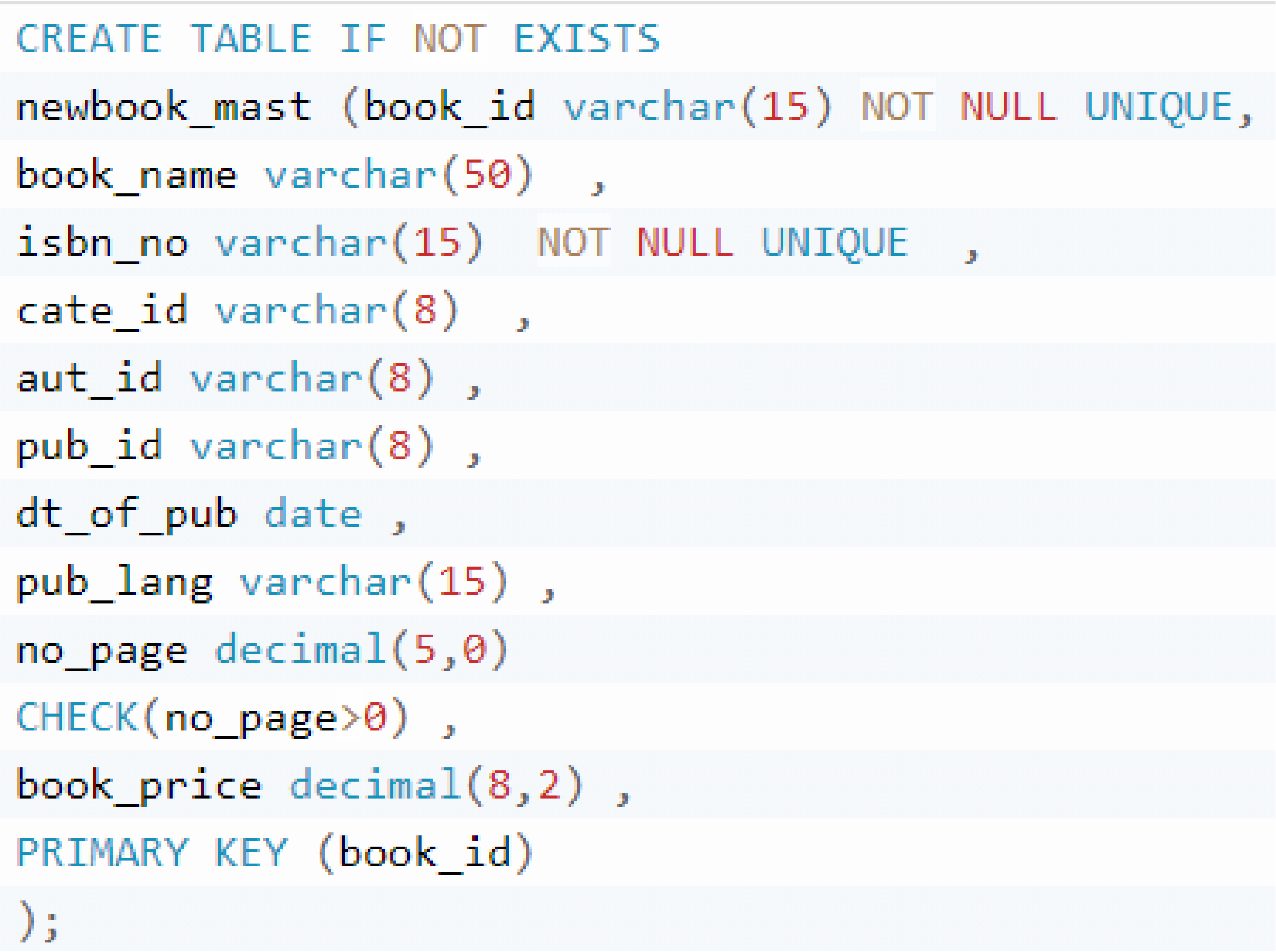


```
CREATE TABLE IF NOT EXISTS newauthor
(aut_id varchar(8) NOT NULL,
aut_name varchar(50) NOT NULL,
country varchar(25) NOT NULL,
home_city varchar(25) NOT NULL );
```



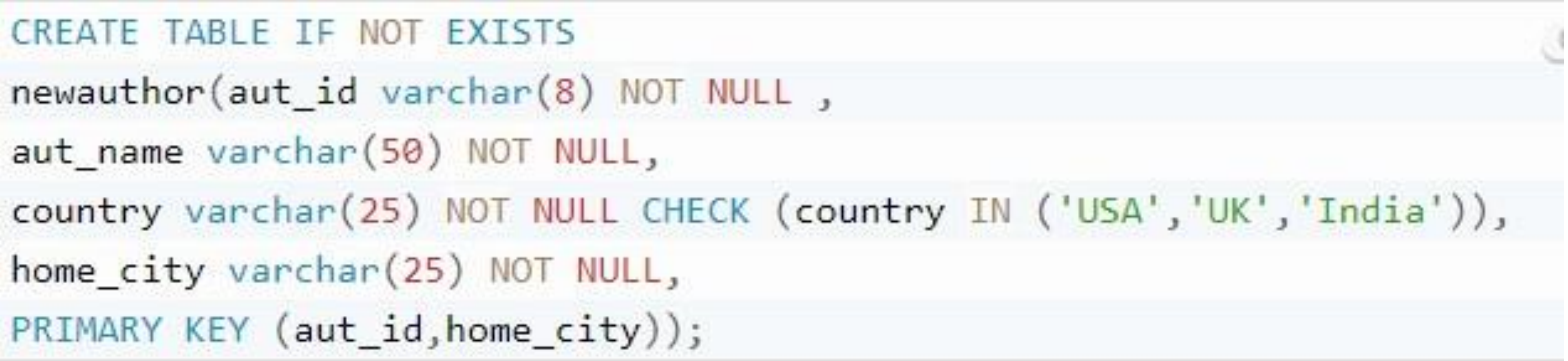
	Field	Type	Collation	Attributes	Null	Default	Extra
□	**aut_id**	varchar(8)	latin1_swedish_ci		No	*None*	
□	**aut_name**	varchar(50)	latin1_swedish_ci		No	*None*	
□	**country**	varchar(25)	latin1_swedish_ci		No	*None*	
□	**home_city**	varchar(25)	latin1_swedish_ci		No	*None*	

Example (CHECK)



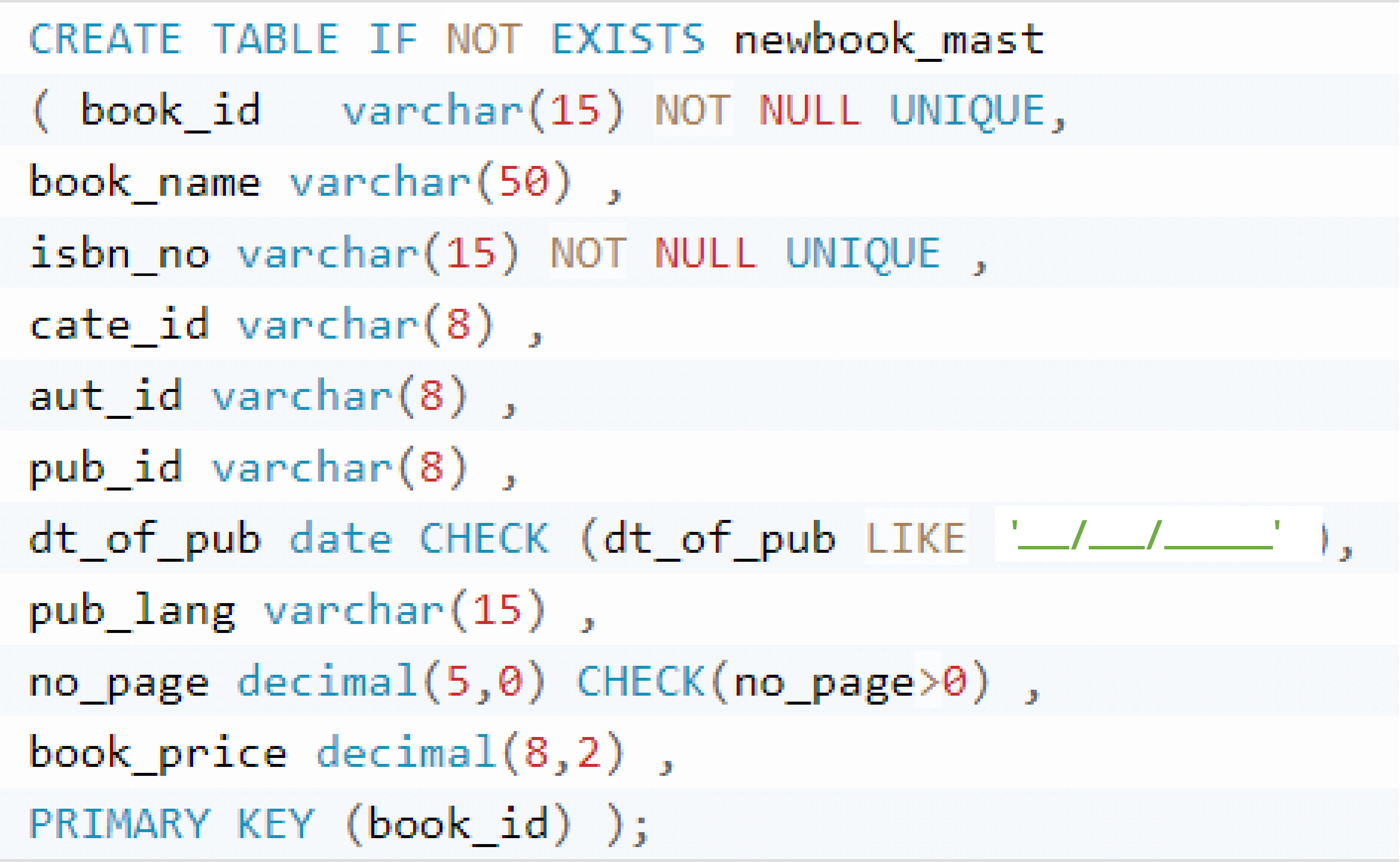
```
CREATE TABLE IF NOT EXISTS
newbook_mast (book_id varchar(15) NOT NULL UNIQUE,
book_name varchar(50)  ,
isbn_no varchar(15)  NOT NULL UNIQUE   ,
cate_id varchar(8)  ,
aut_id varchar(8) ,
pub_id varchar(8) ,
dt_of_pub date ,
pub_lang varchar(15) ,
no_page decimal(5,0)
CHECK(no_page>0) ,
book_price decimal(8,2) ,
PRIMARY KEY (book_id)
);
```

Example (CHECK)



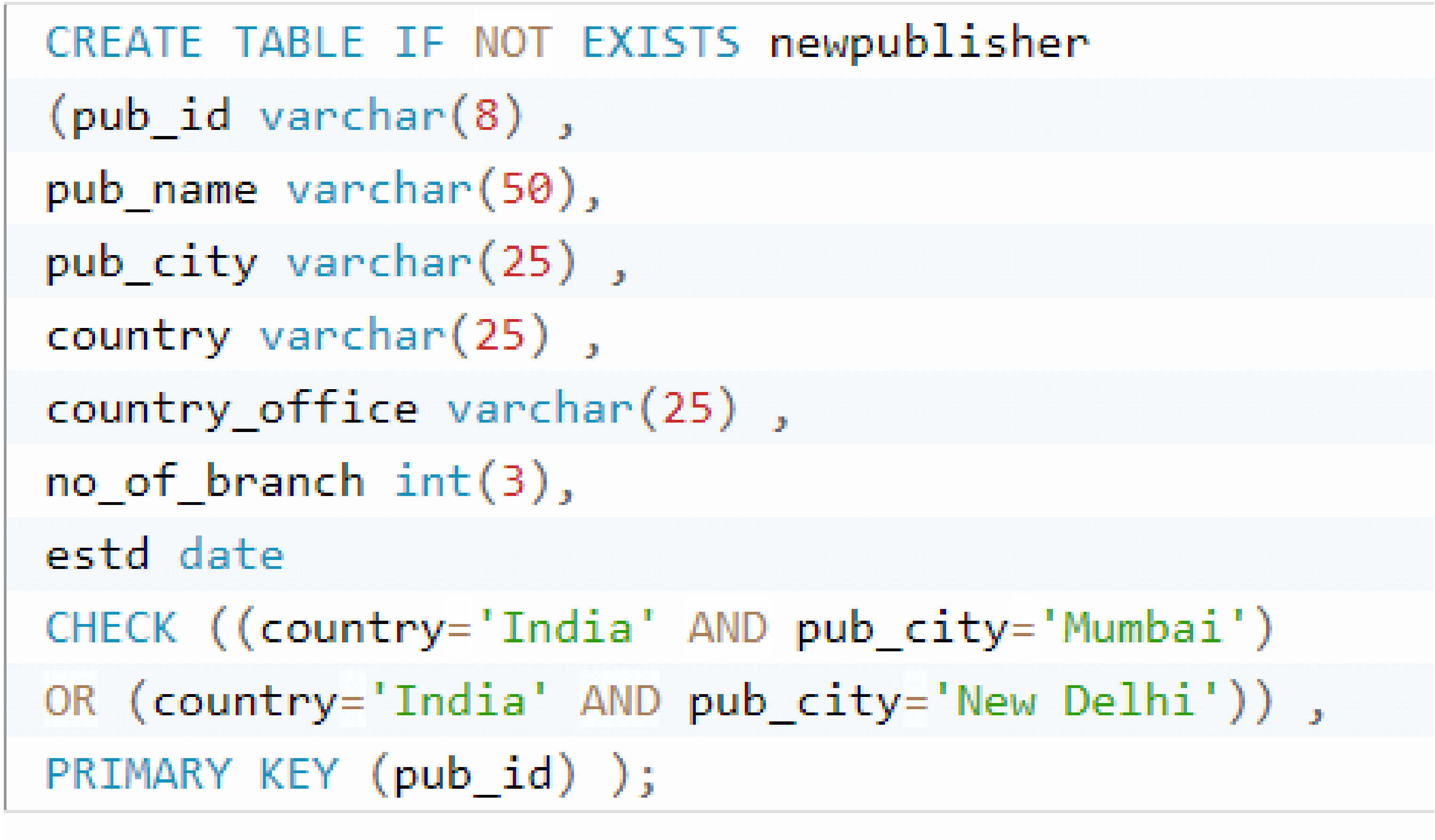
```
CREATE TABLE IF NOT EXISTS
newauthor(aut_id varchar(8) NOT NULL ,
aut_name varchar(50) NOT NULL,
country varchar(25) NOT NULL CHECK (country IN ('USA','UK','India')),
home_city varchar(25) NOT NULL,
PRIMARY KEY (aut_id,home_city));
```

Example (CHECK)



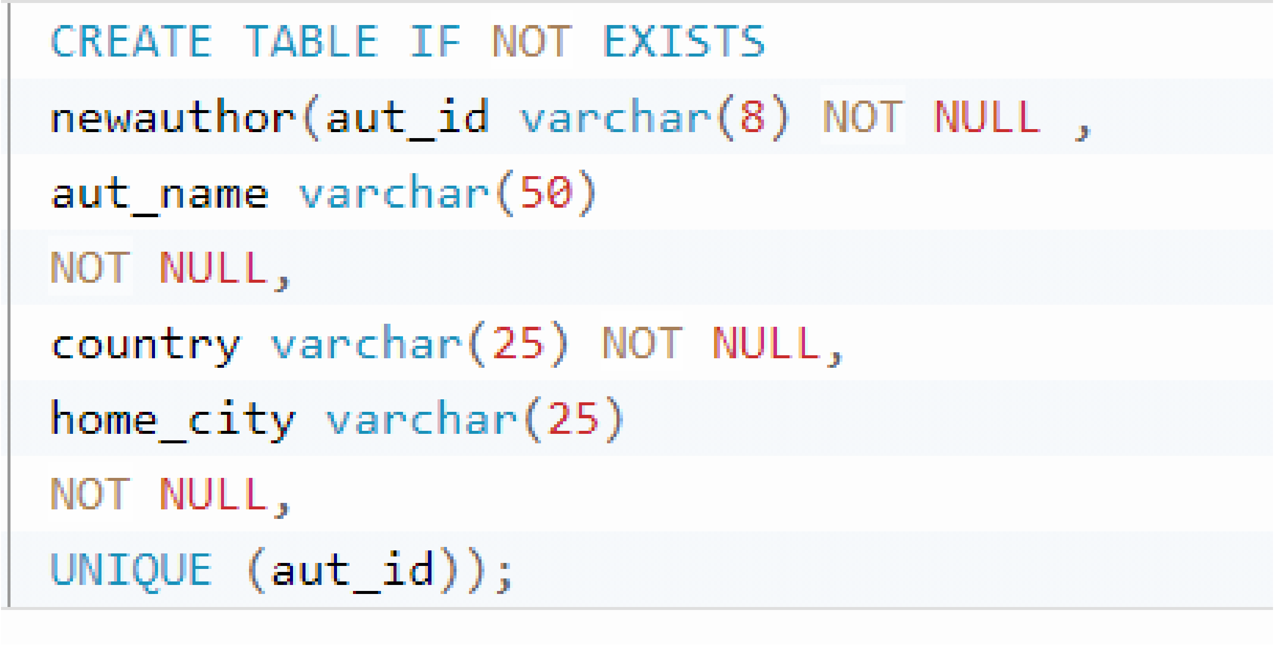
```
CREATE TABLE IF NOT EXISTS newbook_mast
( book_id   varchar(15) NOT NULL UNIQUE,
book_name varchar(50) ,
isbn_no varchar(15) NOT NULL UNIQUE ,
cate_id varchar(8) ,
aut_id varchar(8) ,
pub_id varchar(8) ,
dt_of_pub date CHECK (dt_of_pub LIKE '__/__/____' ),
pub_lang varchar(15) ,
no_page decimal(5,0) CHECK(no_page>0) ,
book_price decimal(8,2) ,
PRIMARY KEY (book_id) );
```

Example (CHECK)

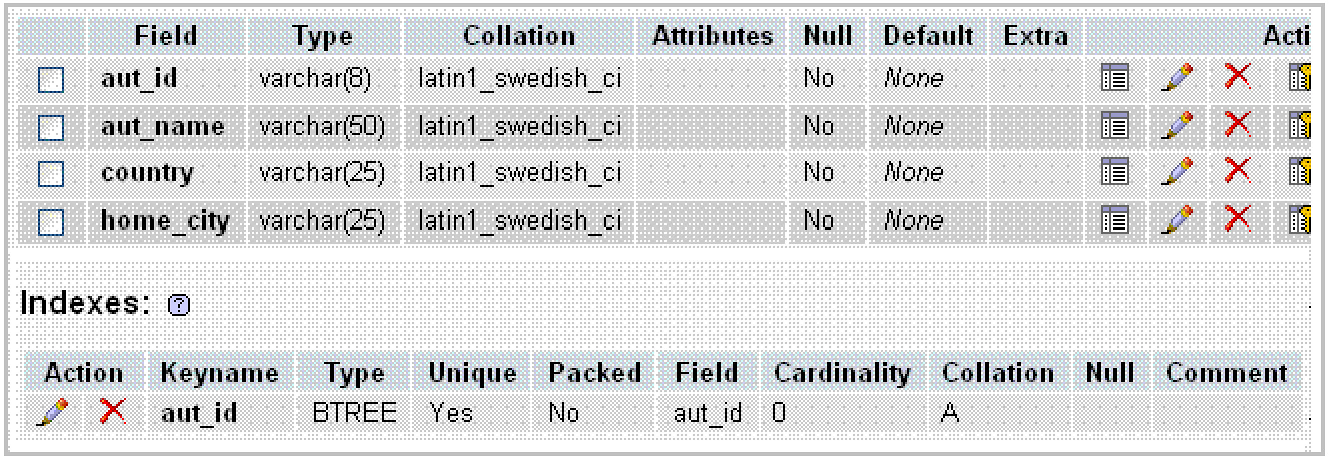


```
CREATE TABLE IF NOT EXISTS newpublisher
(pub_id varchar(8) ,
pub_name varchar(50),
pub_city varchar(25) ,
country varchar(25) ,
country_office varchar(25) ,
no_of_branch int(3),
estd date
CHECK ((country='India' AND pub_city='Mumbai')
OR (country='India' AND pub_city='New Delhi')) ,
PRIMARY KEY (pub_id) );
```

Example (UNIQUE)



```
CREATE TABLE IF NOT EXISTS
newauthor(aut_id varchar(8) NOT NULL ,
aut_name varchar(50)
NOT NULL,
country varchar(25) NOT NULL,
home_city varchar(25)
NOT NULL,
UNIQUE (aut_id));
```

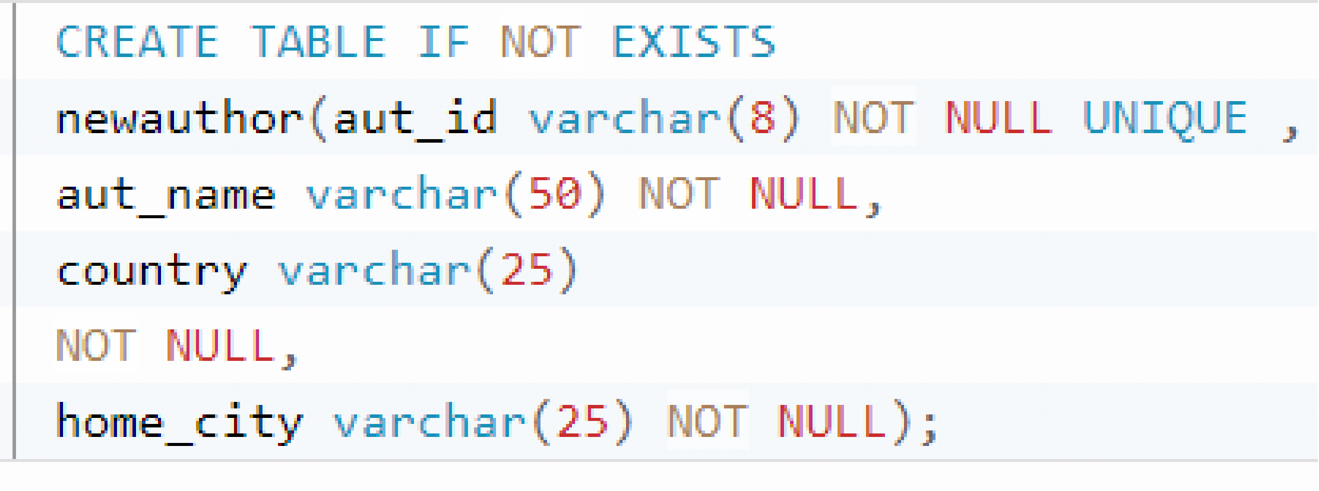


	Field	Type	Collation	Attributes	Null	Default	Extra				Acti
□	**aut_id**	varchar(8)	latin1_swedish_ci		No	*None*		▤	✎	✕	▥
□	**aut_name**	varchar(50)	latin1_swedish_ci		No	*None*		▤	✎	✕	▥
□	**country**	varchar(25)	latin1_swedish_ci		No	*None*		▤	✎	✕	▥
□	**home_city**	varchar(25)	latin1_swedish_ci		No	*None*		▤	✎	✕	▥

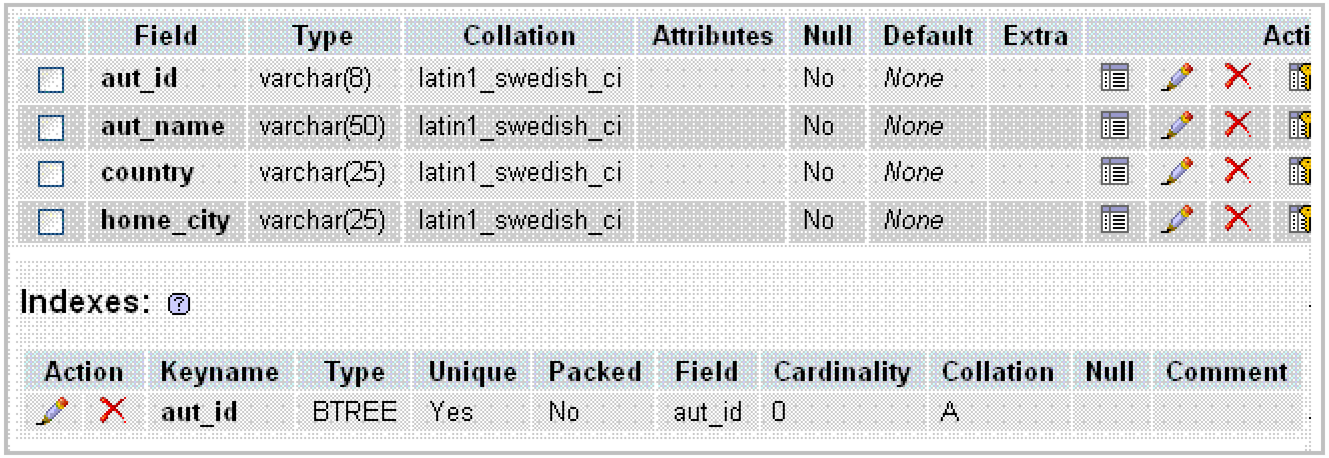
Indexes: ⓘ

Action		Keyname	Type	Unique	Packed	Field	Cardinality	Collation	Null	Comment
✎	✕	**aut_id**	BTREE	Yes	No	aut_id	0	A		

Example (UNIQUE)



```
CREATE TABLE IF NOT EXISTS
newauthor(aut_id varchar(8) NOT NULL UNIQUE ,
aut_name varchar(50) NOT NULL,
country varchar(25)
NOT NULL,
home_city varchar(25) NOT NULL);
```

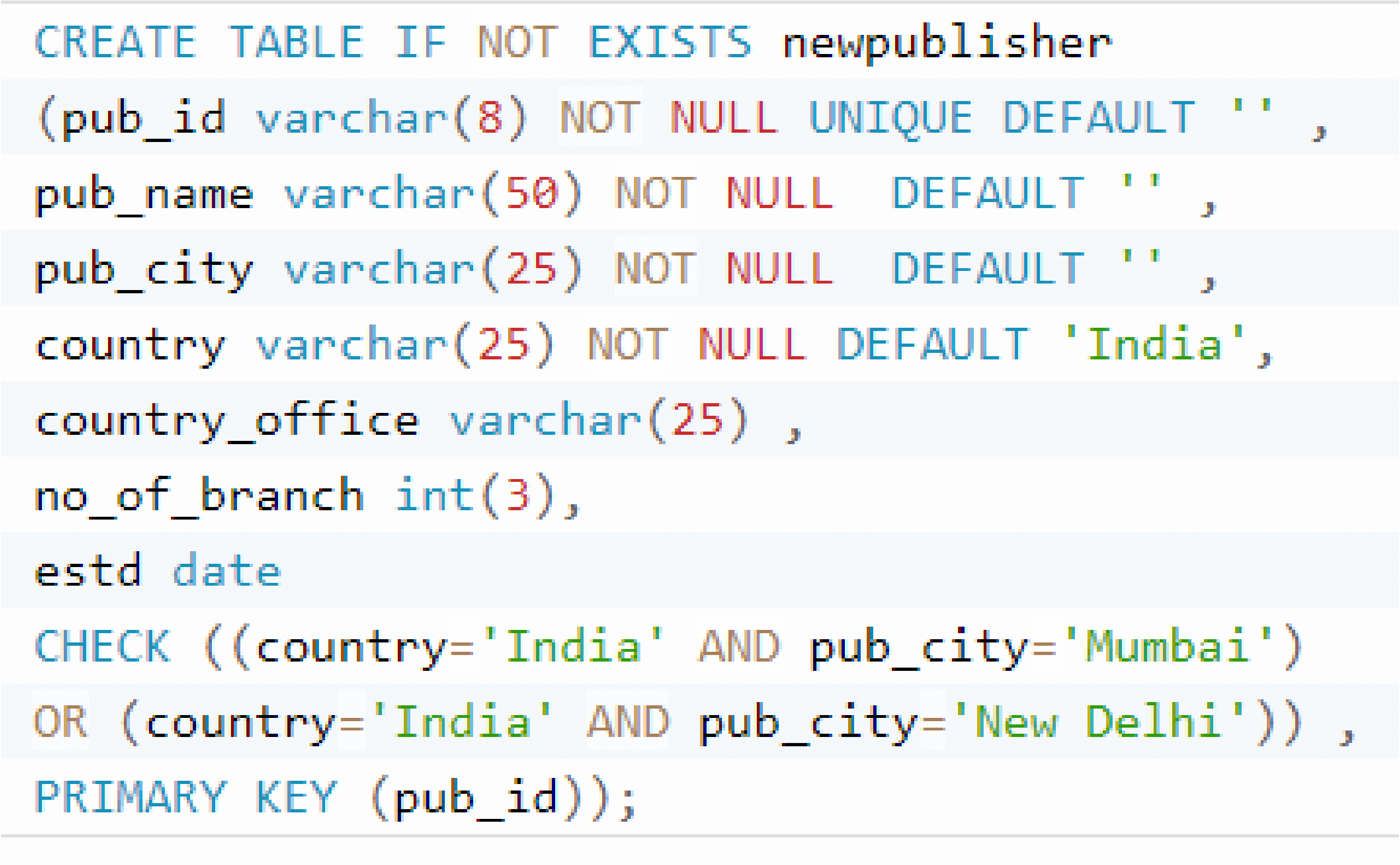


	Field	Type	Collation	Attributes	Null	Default	Extra				Acti
□	**aut_id**	varchar(8)	latin1_swedish_ci		No	*None*		▤	✎	✕	▦
□	**aut_name**	varchar(50)	latin1_swedish_ci		No	*None*		▤	✎	✕	▦
□	**country**	varchar(25)	latin1_swedish_ci		No	*None*		▤	✎	✕	▦
□	**home_city**	varchar(25)	latin1_swedish_ci		No	*None*		▤	✎	✕	▦

Indexes: ?

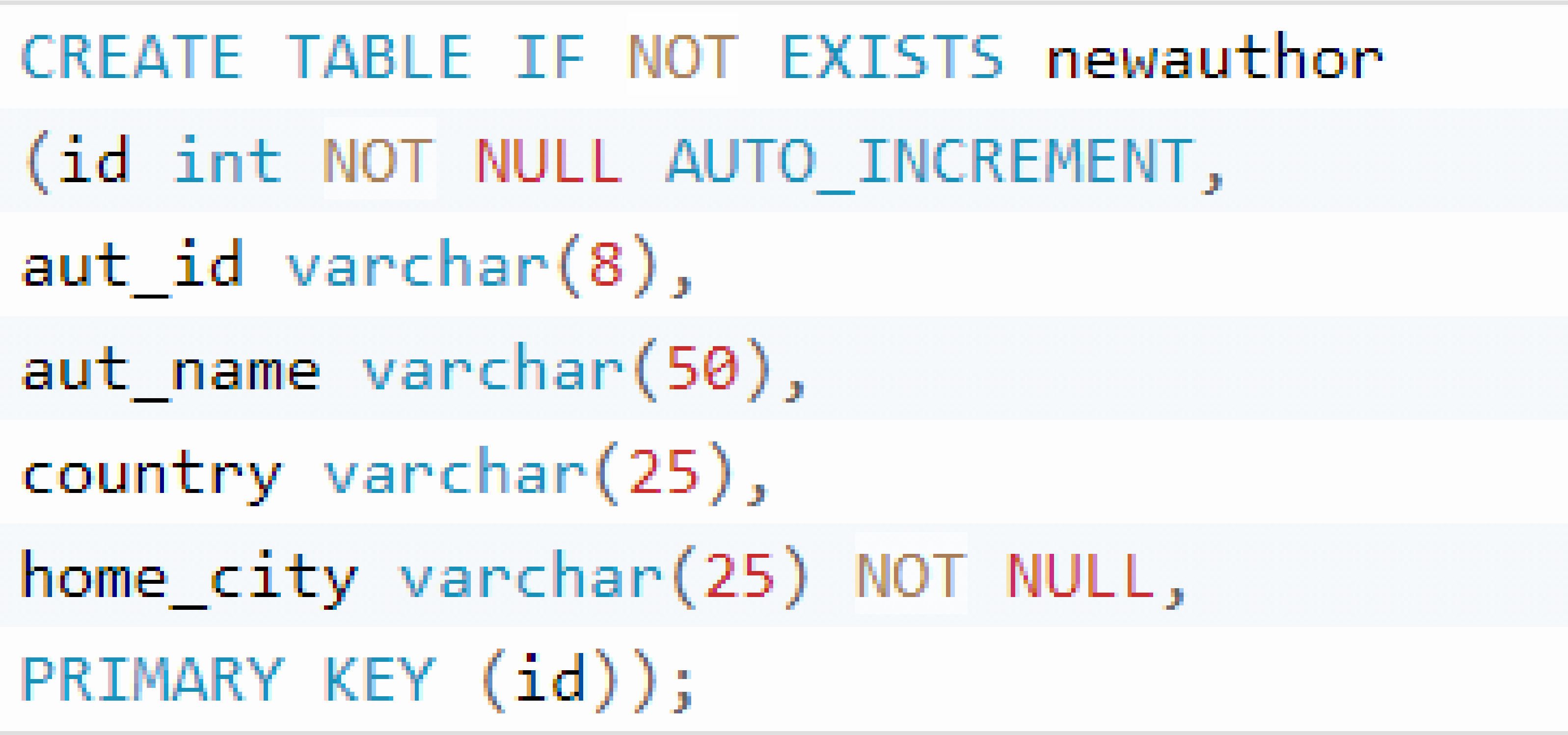
Action		Keyname	Type	Unique	Packed	Field	Cardinality	Collation	Null	Comment
✎	✕	**aut_id**	BTREE	Yes	No	aut_id	0	A		

Example (DEFAULT)



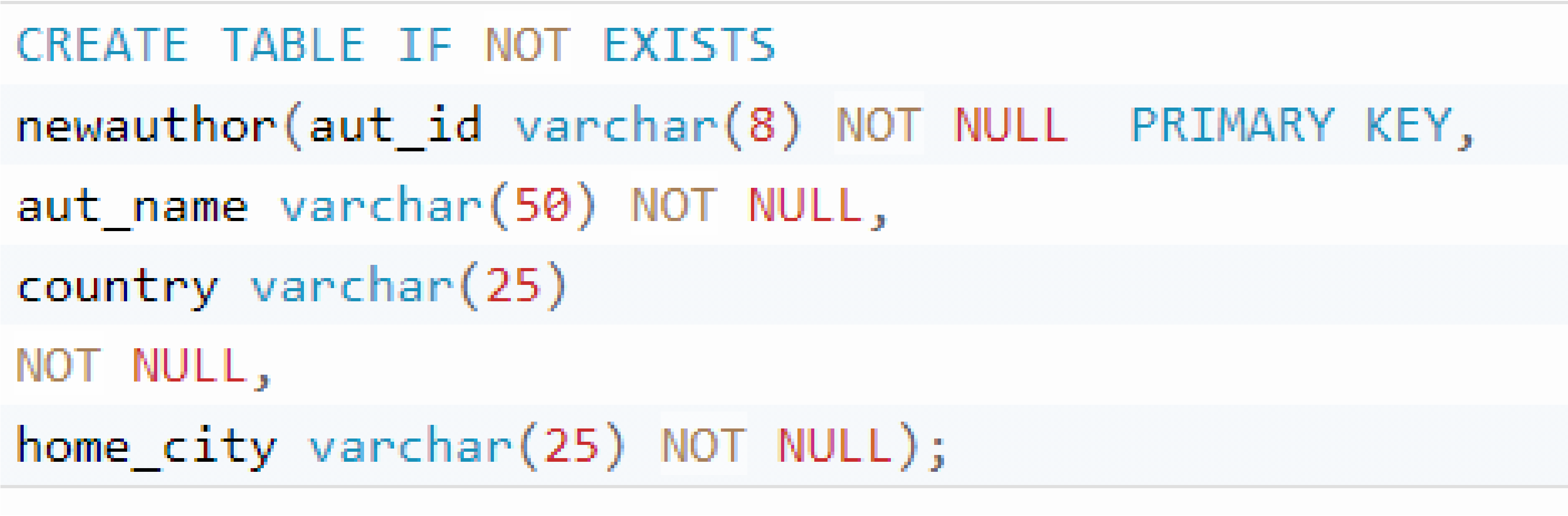
```
CREATE TABLE IF NOT EXISTS newpublisher
(pub_id varchar(8) NOT NULL UNIQUE DEFAULT '' ,
pub_name varchar(50) NOT NULL  DEFAULT '' ,
pub_city varchar(25) NOT NULL  DEFAULT '' ,
country varchar(25) NOT NULL DEFAULT 'India',
country_office varchar(25) ,
no_of_branch int(3),
estd date
CHECK ((country='India' AND pub_city='Mumbai')
OR (country='India' AND pub_city='New Delhi')) ,
PRIMARY KEY (pub_id));
```

Example (AUTO_INCREMENT)



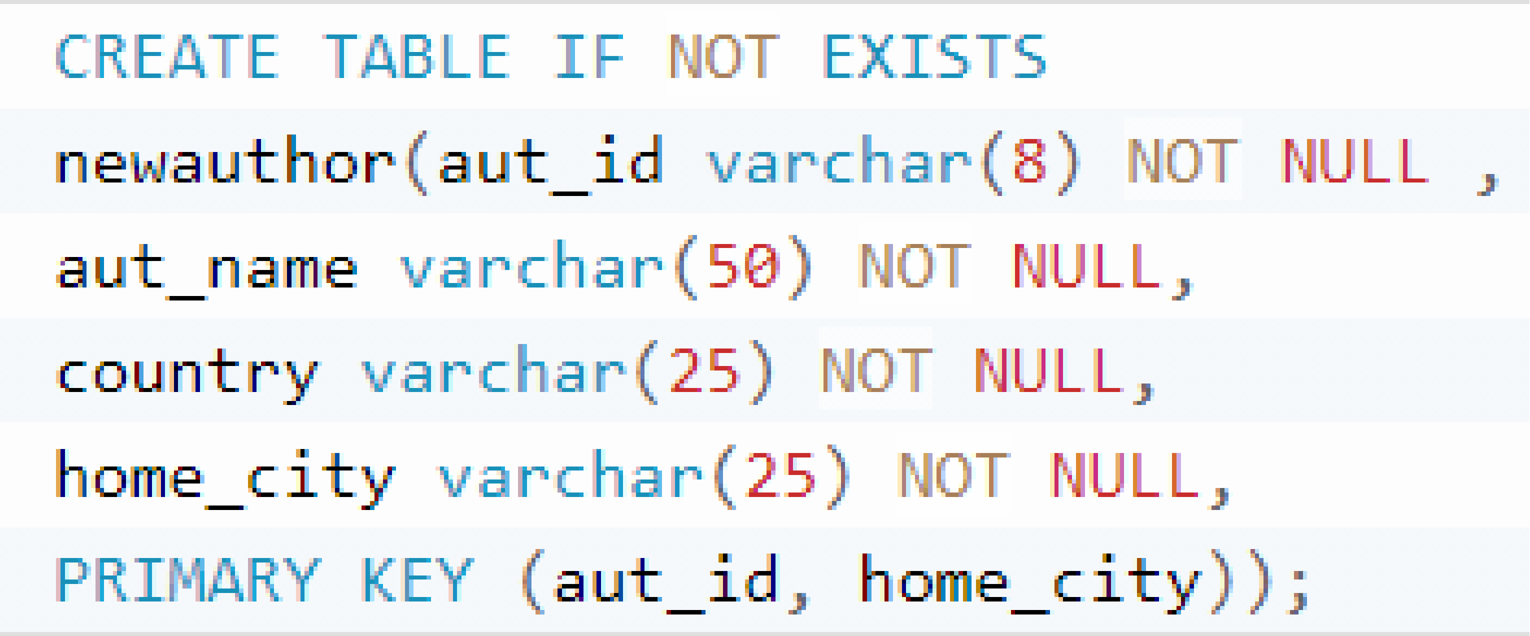
```
CREATE TABLE IF NOT EXISTS newauthor
(id int NOT NULL AUTO_INCREMENT,
aut_id varchar(8),
aut_name varchar(50),
country varchar(25),
home_city varchar(25) NOT NULL,
PRIMARY KEY (id));
```

Example (PK)



```
CREATE TABLE IF NOT EXISTS
newauthor(aut_id varchar(8) NOT NULL  PRIMARY KEY,
aut_name varchar(50) NOT NULL,
country varchar(25)
NOT NULL,
home_city varchar(25) NOT NULL);
```

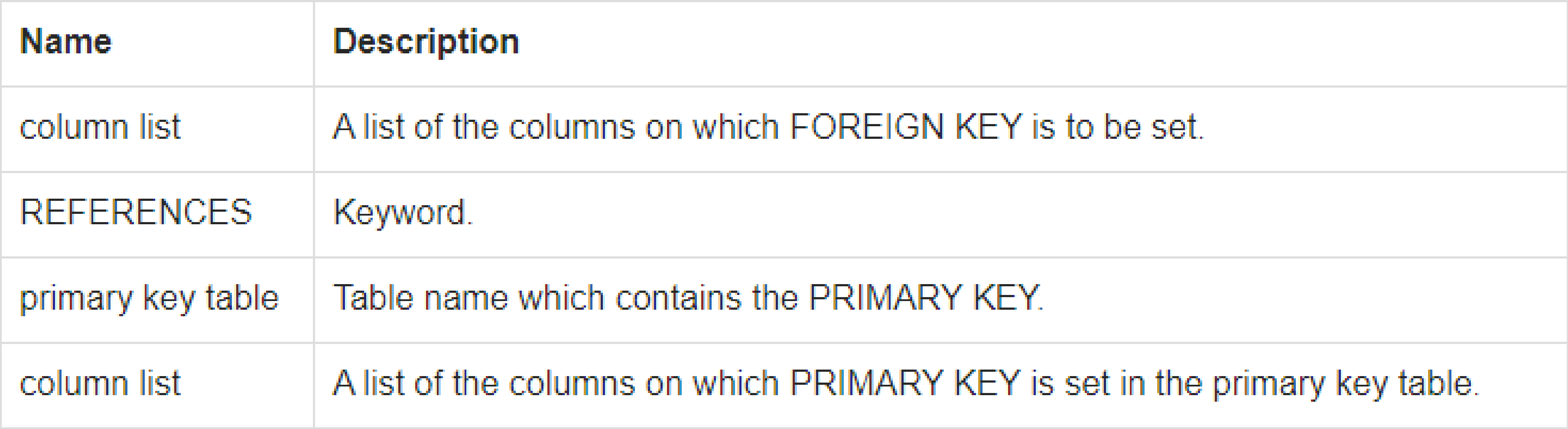
Example (PK)



```
CREATE TABLE IF NOT EXISTS
newauthor(aut_id varchar(8) NOT NULL ,
aut_name varchar(50) NOT NULL,
country varchar(25) NOT NULL,
home_city varchar(25) NOT NULL,
PRIMARY KEY (aut_id, home_city));
```

Foreign Key Constraints

```
FOREIGN KEY [column list] REFERENCES [primary key table]
([column list]);
```



Name	Description
column list	A list of the columns on which FOREIGN KEY is to be set.
REFERENCES	Keyword.
primary key table	Table name which contains the PRIMARY KEY.
column list	A list of the columns on which PRIMARY KEY is set in the primary key table.

Example

A new table 'newbook_mast' will be created.

The PRIMARY KEY for that table 'newbook_mast' is 'book_id'.

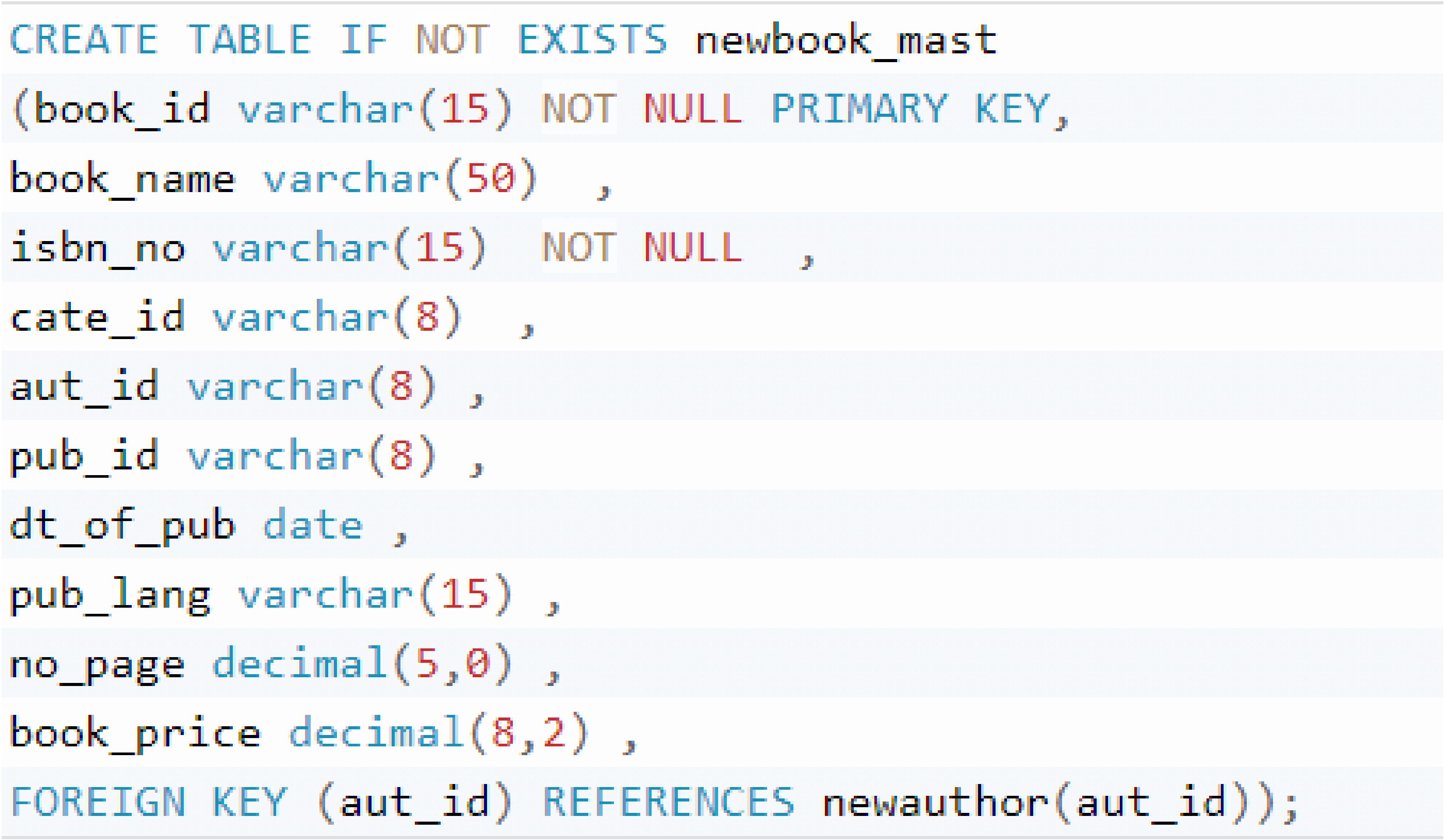
The FOREIGN KEY for the table 'newbook_mast' is 'aut_id'.

The 'aut_id' is the PRIMARY KEY for the table 'newauthor'.

The FOREIGN KEY 'aut_id' for the table 'newbook_mast' points to the PRIMARY KEY 'aut_id' of the table 'newauthor'.

That means the 'aut_id's which are present in the 'newauthor' table, only those authors will come to the 'newbook_mast' table.

Example

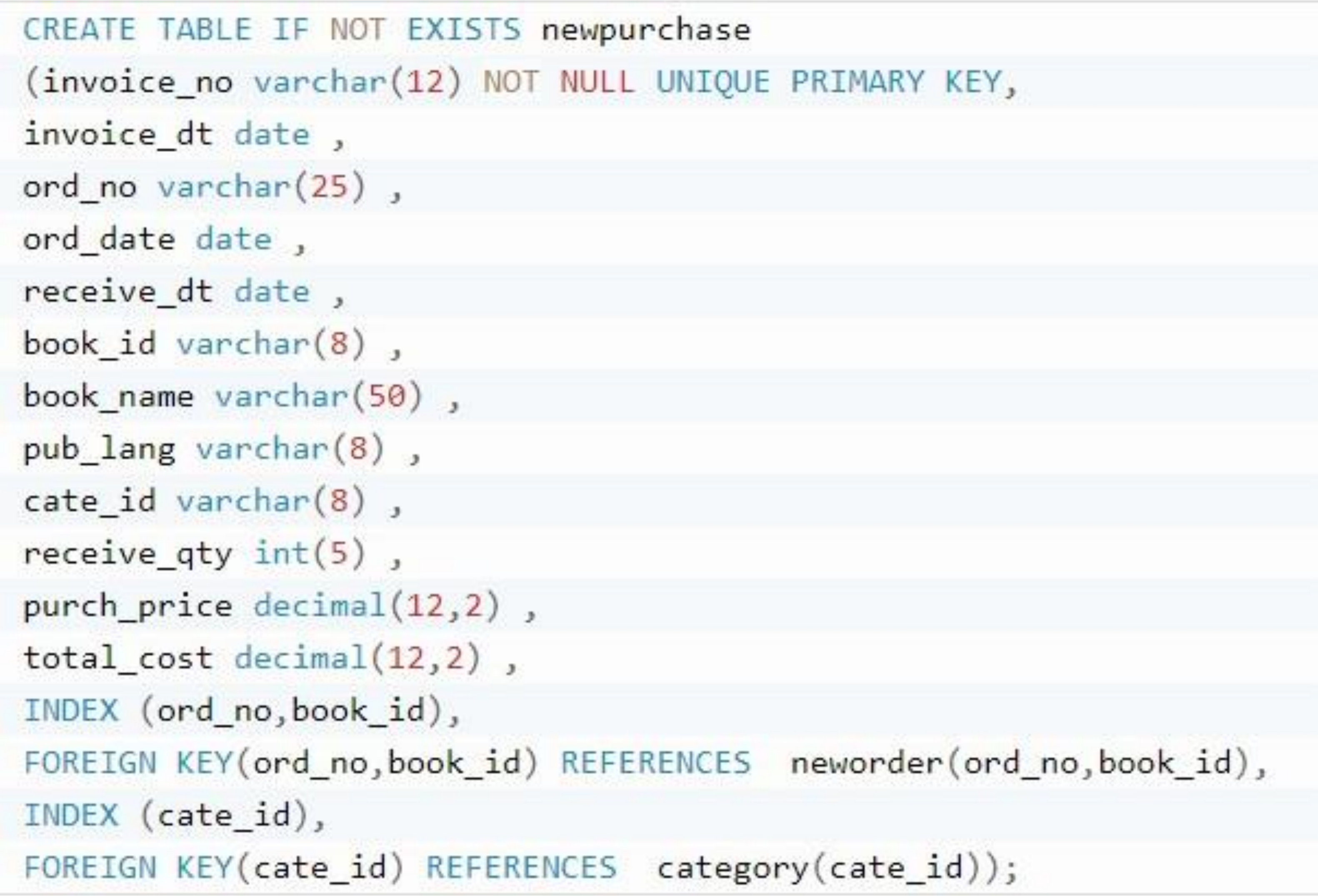


```
CREATE TABLE IF NOT EXISTS newbook_mast
(book_id varchar(15) NOT NULL PRIMARY KEY,
book_name varchar(50)  ,
isbn_no varchar(15)  NOT NULL  ,
cate_id varchar(8)  ,
aut_id varchar(8) ,
pub_id varchar(8) ,
dt_of_pub date ,
pub_lang varchar(15) ,
no_page decimal(5,0) ,
book_price decimal(8,2) ,
FOREIGN KEY (aut_id) REFERENCES newauthor(aut_id));
```

Multiple FK Example

A new table 'newpurchase' will be created.
The PRIMARY KEY for that table 'newpurchase' is 'invoice_no'.
The one FOREIGN KEY for the table 'newpurchase' is a combination of 'ord_no' and 'book_id'.
The another FOREIGN KEY for the table 'newpurchase' is 'cate_id'.
The 'ord_no' and 'book_id' combination is the PRIMARY KEY for the table 'neworder'.
The 'cate_id' is the PRIMARY KEY for the table 'category'.
The FOREIGN KEY 'ord_no' and 'book_id' combination for the table 'newpurchase', which points to the PRIMARY KEY 'ord_no' and 'book_id' combination of the table 'neworder'.
That means the distinct ('ord_no' and 'book_id') combination which are present in the in the 'neworder' table only those unique 'order number' and 'book id' combination will come in the 'newpurchase' table.
The another FOREIGN KEY 'cate_id' for the table 'newpurchase', which points to the PRIMARY KEY 'cate_id' of the table 'category'. That means the 'cate_id' which are present in the 'category' table only those 'category' will come in the 'newpurchase' table.

Multiple FK Example



```
CREATE TABLE IF NOT EXISTS newpurchase
(invoice_no varchar(12) NOT NULL UNIQUE PRIMARY KEY,
invoice_dt date ,
ord_no varchar(25) ,
ord_date date ,
receive_dt date ,
book_id varchar(8) ,
book_name varchar(50) ,
pub_lang varchar(8) ,
cate_id varchar(8) ,
receive_qty int(5) ,
purch_price decimal(12,2) ,
total_cost decimal(12,2) ,
INDEX (ord_no,book_id),
FOREIGN KEY(ord_no,book_id) REFERENCES  neworder(ord_no,book_id),
INDEX (cate_id),
FOREIGN KEY(cate_id) REFERENCES  category(cate_id));
```

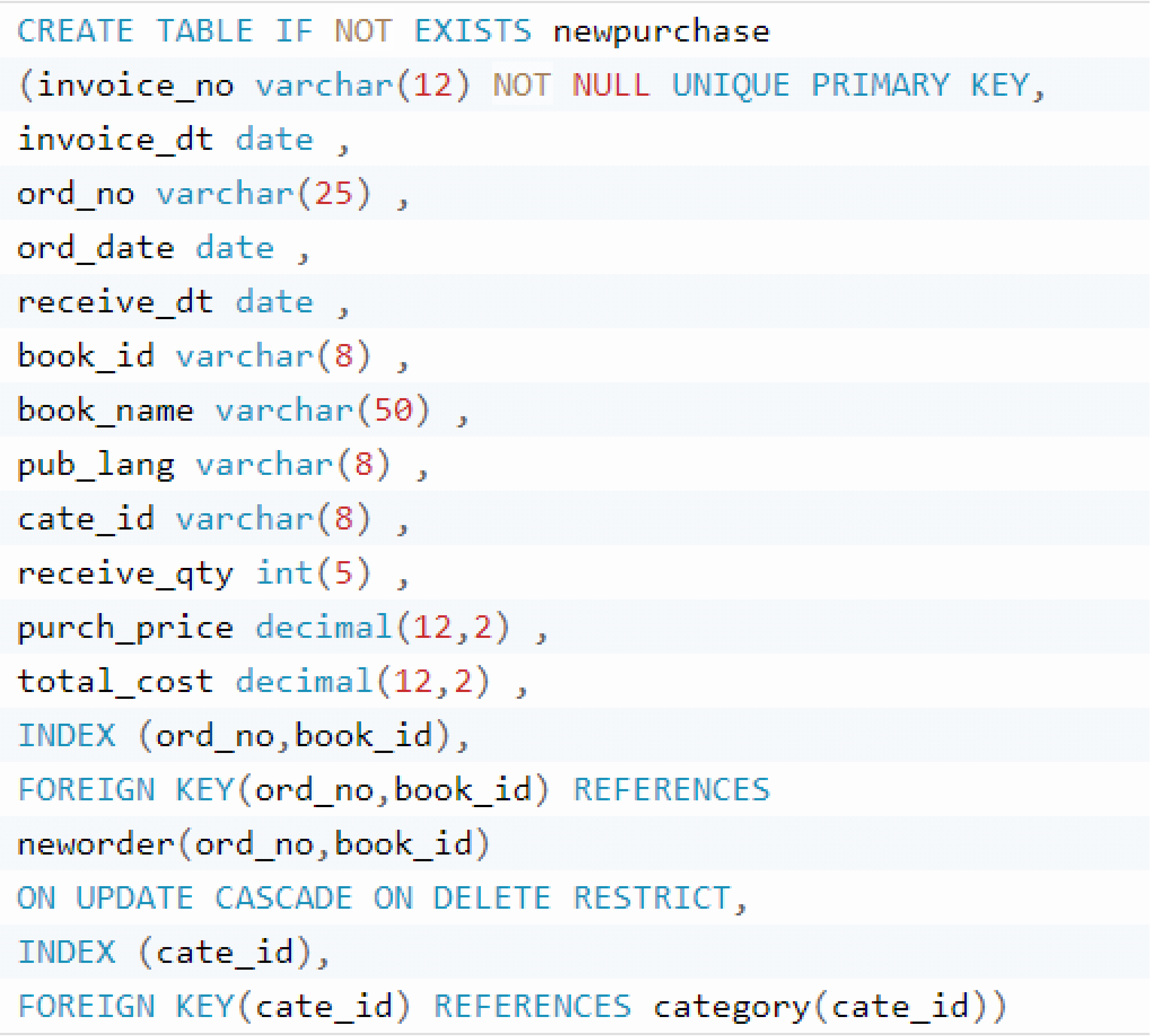
Multiple FK Multiple Table Example

A new table 'newbook_mast' will be created.
The PRIMARY KEY for that table 'newbook_mast' is 'book_id'.
The one FOREIGN KEY for the table 'newbook_mast' is 'aut_id'.
The another FOREIGN KEY for the table 'newbook_mast' is 'pub_id'.
The 'aut_id' is the PRIMARY KEY for the table 'newauthor'.
The 'pub_id' is the PRIMARY KEY for the table 'newpublisher'.
The FOREIGN KEY 'aut_id' for the table 'newbook_mast', which points to the PRIMARY KEY 'aut_id' of the table 'newauthor'.
That means the 'aut_id' which are present in the in the 'nuwauthor' table only those authors will come to the 'newbook_mast' table.
The another FOREIGN KEY 'pub_id' for the table 'newbook_mast' , which points to the PRIMARY KEY 'pub_id' of the table 'newpublisher'.
That means the 'pub_id' which are present in the in the 'newpublisher' table only those publishers will come to the 'newbook_mast' table.

CASCADE & RESTRICT Example

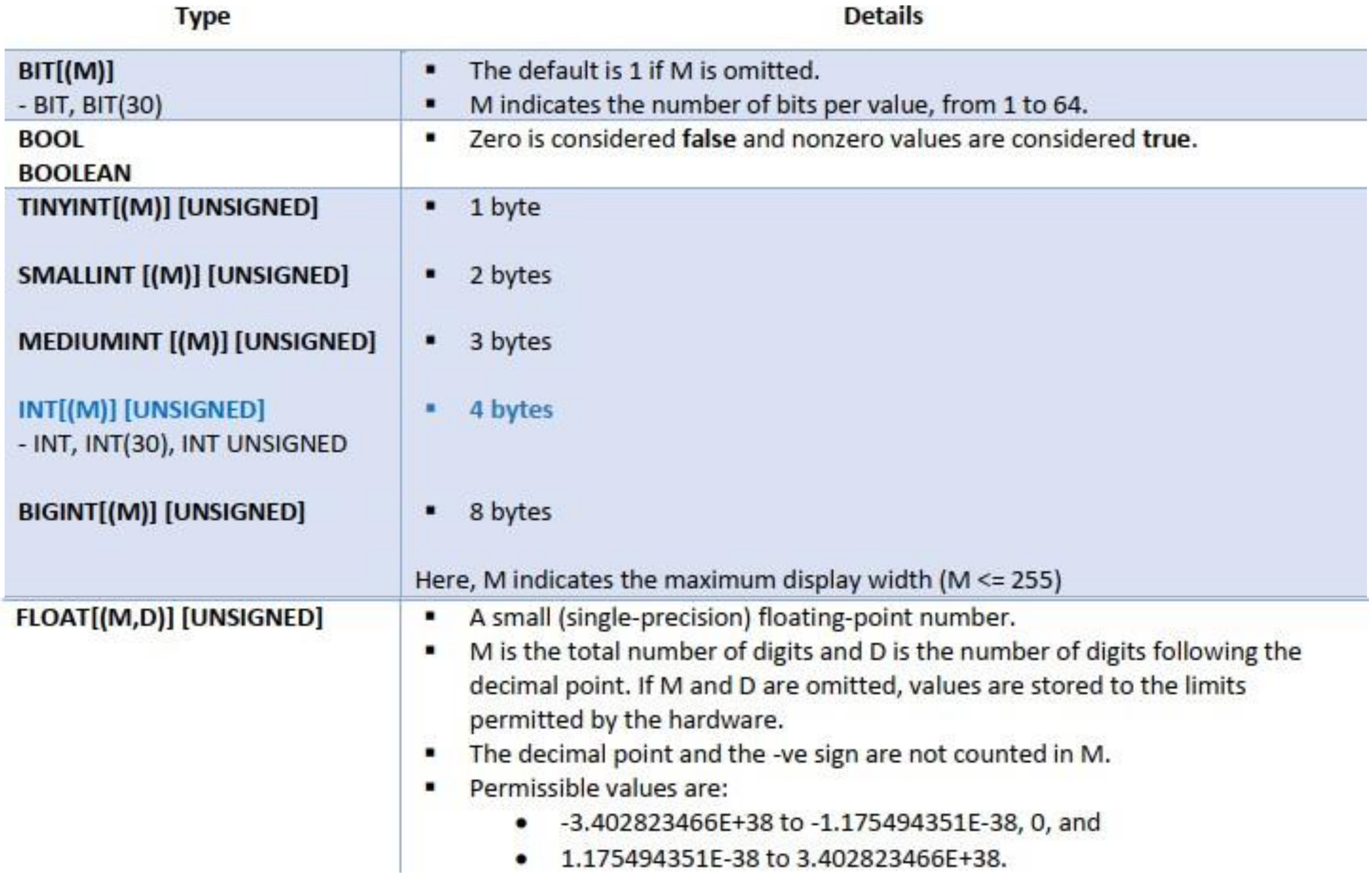
A new table 'newpurchase' will be created.
The PRIMARY KEY for that table 'newpurchase' is 'invoice_no'.
The one FOREIGN KEY for the table 'newpurchase' is a combination of 'ord_no' and 'book_id'.
The another FOREIGN KEY for the table 'newpurchase' is 'cate_id'.
The 'ord_no' and 'book_id' combination is the PRIMARY KEY for the table 'neworder'.
The 'cate_id' is the PRIMARY KEY for the table 'category'.
The FOREIGN KEY 'ord_no' and 'book_id' combination for the table 'newpurchase', which points to the PRIMARY KEY 'ord_no' and 'book_id' combination of the table 'neworder'. That means the distinct ('ord_no' and 'book_id') combination which are present in the 'neworder' table only those unique 'order number' and 'book id' combination will come in the 'newpurchase' table.
The another FOREIGN KEY 'cate_id' for the table 'newpurchase' , which points to the PRIMARY KEY 'cate_id' of the table 'category'. That means the 'cate_id' which are present in the 'category' table only those 'category' will come in the 'newpurchase' table.
The ON UPDATE CASCADE ensures that the records inside the child table 'newpurchase' always points to the PRIMARY KEY inside the parent table 'neworder'.
If any record gets deleted/updated from the 'neworder' table MySQL handles the deletion/updating of the records from 'newpurchase' table.
ON DELETE RESTRICT prevents a record in a parent table 'neworder' being deleted or altered when it is still referenced from a child table 'newpurchase'.

CASCADE & RESTRICT Example



```
CREATE TABLE IF NOT EXISTS newpurchase
(invoice_no varchar(12) NOT NULL UNIQUE PRIMARY KEY,
invoice_dt date ,
ord_no varchar(25) ,
ord_date date ,
receive_dt date ,
book_id varchar(8) ,
book_name varchar(50) ,
pub_lang varchar(8) ,
cate_id varchar(8) ,
receive_qty int(5) ,
purch_price decimal(12,2) ,
total_cost decimal(12,2) ,
INDEX (ord_no,book_id),
FOREIGN KEY(ord_no,book_id) REFERENCES
neworder(ord_no,book_id)
ON UPDATE CASCADE ON DELETE RESTRICT,
INDEX (cate_id),
FOREIGN KEY(cate_id) REFERENCES category(cate_id))
```

Data Types

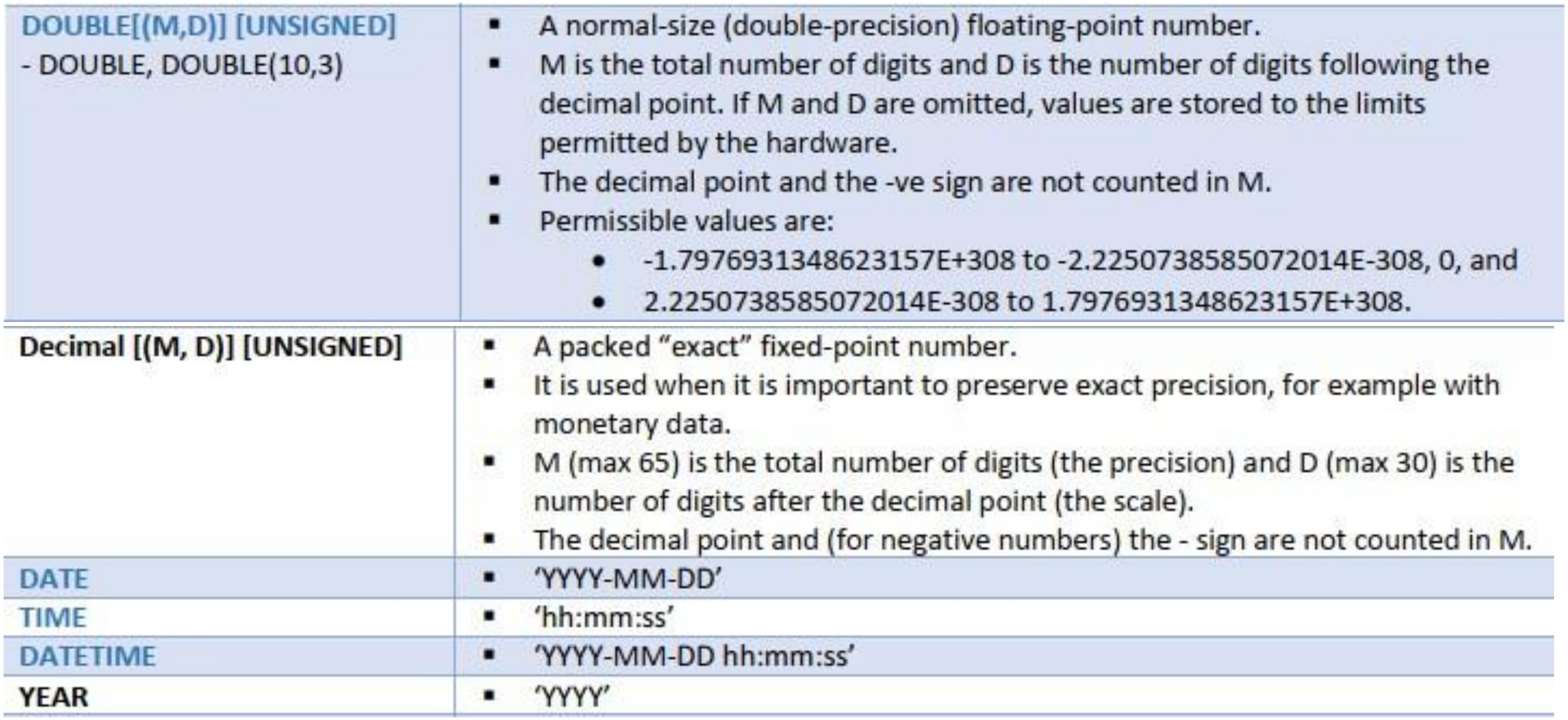


Type	Details
BIT[(M)] - BIT, BIT(30)	▪ The default is 1 if M is omitted. ▪ M indicates the number of bits per value, from 1 to 64.
BOOL **BOOLEAN**	▪ Zero is considered **false** and nonzero values are considered **true**.
TINYINT[(M)] [UNSIGNED]	▪ 1 byte
SMALLINT [(M)] [UNSIGNED]	▪ 2 bytes
MEDIUMINT [(M)] [UNSIGNED]	▪ 3 bytes
INT[(M)] [UNSIGNED] - INT, INT(30), INT UNSIGNED	▪ **4 bytes**
BIGINT[(M)] [UNSIGNED]	▪ 8 bytes
	Here, M indicates the maximum display width (M <= 255)
FLOAT[(M,D)] [UNSIGNED]	▪ A small (single-precision) floating-point number. ▪ M is the total number of digits and D is the number of digits following the decimal point. If M and D are omitted, values are stored to the limits permitted by the hardware. ▪ The decimal point and the -ve sign are not counted in M. ▪ Permissible values are: • -3.402823466E+38 to -1.175494351E-38, 0, and • 1.175494351E-38 to 3.402823466E+38.

BIT example

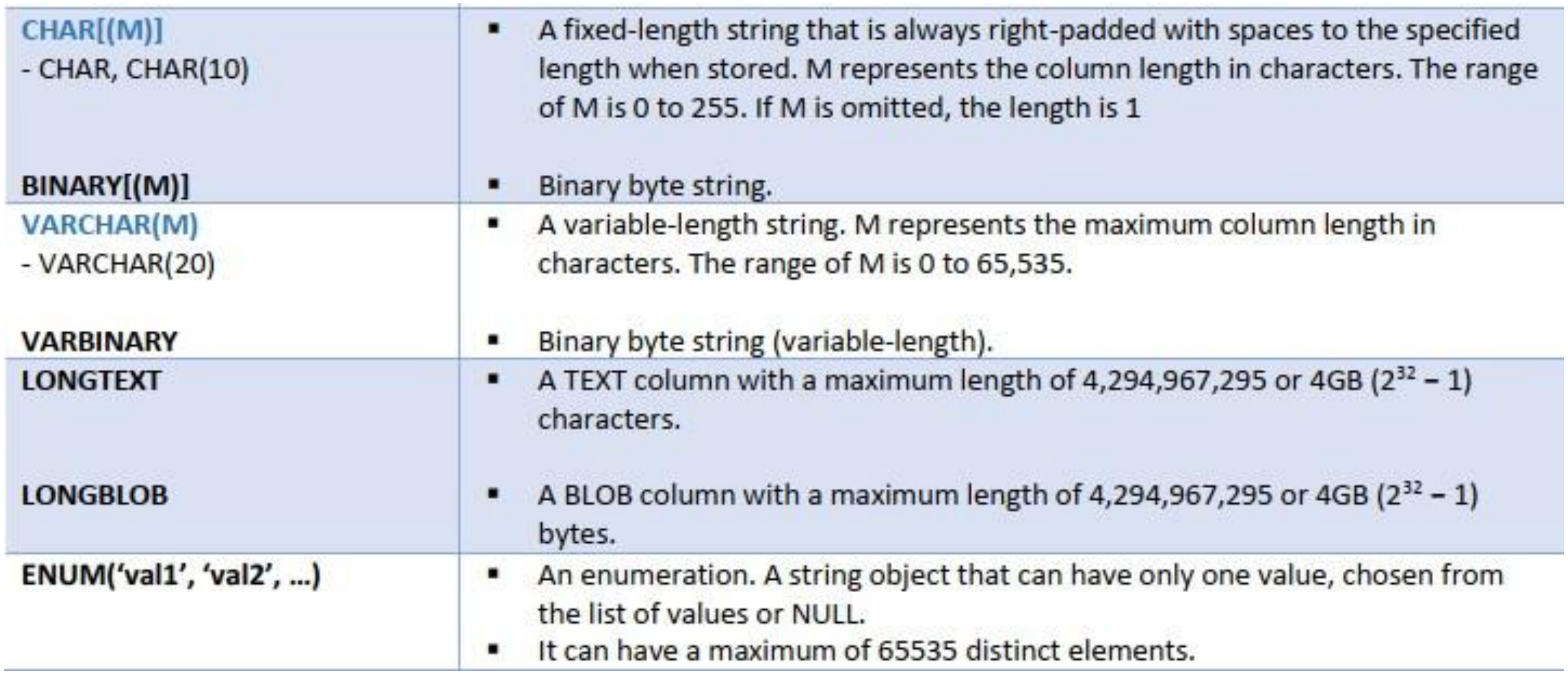
```
CREATE TABLE my_calendars(
    years INT,
    weeks INT,
    days BIT(7),
    PRIMARY KEY(years, weeks)
);

INSERT INTO my_calendars (years, weeks, days) VALUES(2020, 2, B
'1111100');
```

Data Types



DOUBLE[(M,D)] [UNSIGNED] - DOUBLE, DOUBLE(10,3)	▪ A normal-size (double-precision) floating-point number. ▪ M is the total number of digits and D is the number of digits following the decimal point. If M and D are omitted, values are stored to the limits permitted by the hardware. ▪ The decimal point and the -ve sign are not counted in M. ▪ Permissible values are: • -1.7976931348623157E+308 to -2.2250738585072014E-308, 0, and • 2.2250738585072014E-308 to 1.7976931348623157E+308.
Decimal [(M, D)] [UNSIGNED]	▪ A packed "exact" fixed-point number. ▪ It is used when it is important to preserve exact precision, for example with monetary data. ▪ M (max 65) is the total number of digits (the precision) and D (max 30) is the number of digits after the decimal point (the scale). ▪ The decimal point and (for negative numbers) the - sign are not counted in M.
DATE	▪ 'YYYY-MM-DD'
TIME	▪ 'hh:mm:ss'
DATETIME	▪ 'YYYY-MM-DD hh:mm:ss'
YEAR	▪ 'YYYY'

Data Types



CHAR[(M)] - CHAR, CHAR(10)	▪ A fixed-length string that is always right-padded with spaces to the specified length when stored. M represents the column length in characters. The range of M is 0 to 255. If M is omitted, the length is 1
BINARY[(M)]	▪ Binary byte string.
VARCHAR(M) - VARCHAR(20)	▪ A variable-length string. M represents the maximum column length in characters. The range of M is 0 to 65,535.
VARBINARY	▪ Binary byte string (variable-length).
LONGTEXT	▪ A TEXT column with a maximum length of 4,294,967,295 or 4GB (2^{32} – 1) characters.
LONGBLOB	▪ A BLOB column with a maximum length of 4,294,967,295 or 4GB (2^{32} – 1) bytes.
ENUM('val1', 'val2', …)	▪ An enumeration. A string object that can have only one value, chosen from the list of values or NULL. ▪ It can have a maximum of 65535 distinct elements.

Operators

Bitwise

&, ~, |, ^, <<, >>

Arithmetic

DIV (integer div), / (floating point div)
- (minus), - (negative sign)
%, MOD (modulus)
+ (plus)
* (multiplication)

Logical

AND, &&
OR, ||
NOT, !
XOR

Assignment

= (to assign value)

Comparison

>, >=, <, <=, !=, <> (not equal), = (equality check), <=>

BETWEEN … AND …
NOT BETWEEN … AND …

IN(val1, val2, …)
NOT IN(val1, val2, …)

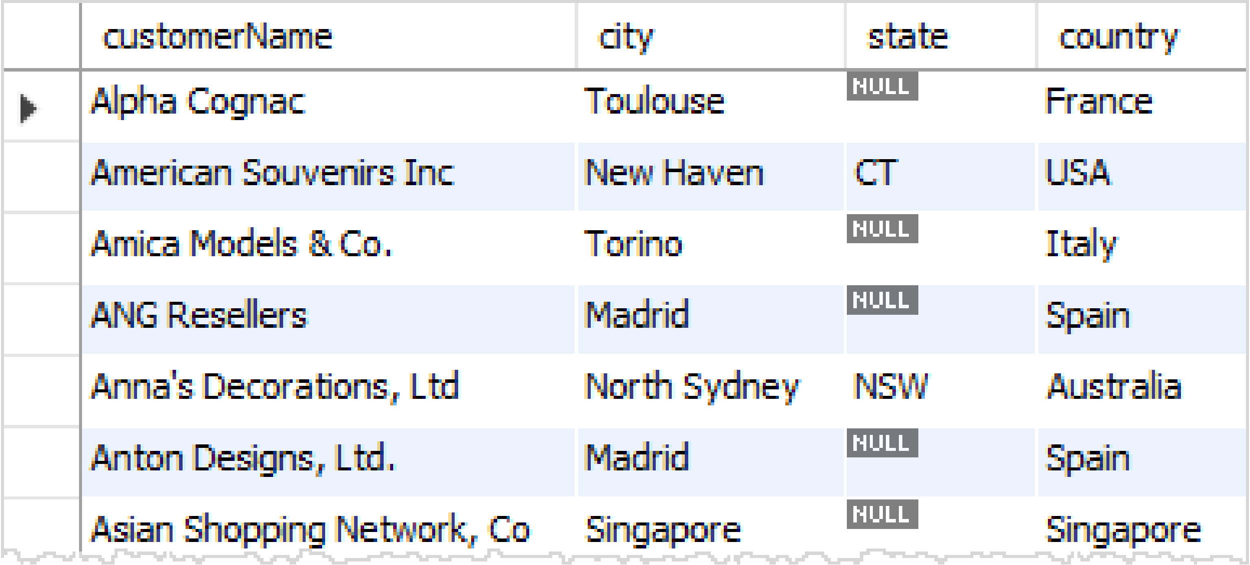
LIKE pattern
NOT LIKE pattern
here, % = 0 to many chars and _ = exactly 1 char

IS boolean
IS NOT boolean

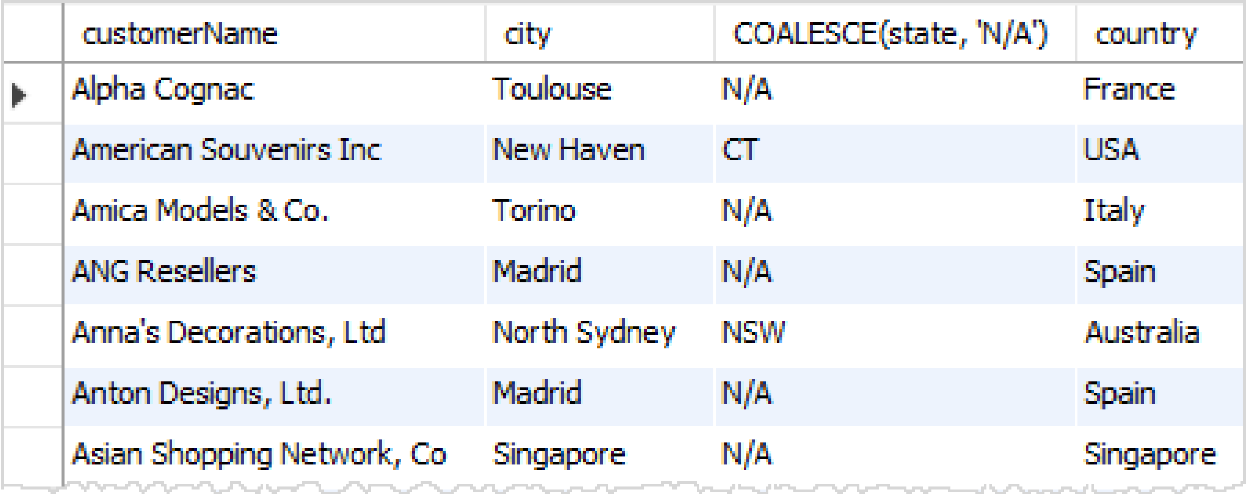
IS NULL
IS NOT NULL

COALESCE(val1, val2, … …)

COALESCE example



customerName	city	state	country
Alpha Cognac	Toulouse	NULL	France
American Souvenirs Inc	New Haven	CT	USA
Amica Models & Co.	Torino	NULL	Italy
ANG Resellers	Madrid	NULL	Spain
Anna's Decorations, Ltd	North Sydney	NSW	Australia
Anton Designs, Ltd.	Madrid	NULL	Spain
Asian Shopping Network, Co	Singapore	NULL	Singapore



customerName	city	COALESCE(state, 'N/A')	country
Alpha Cognac	Toulouse	N/A	France
American Souvenirs Inc	New Haven	CT	USA
Amica Models & Co.	Torino	N/A	Italy
ANG Resellers	Madrid	N/A	Spain
Anna's Decorations, Ltd	North Sydney	NSW	Australia
Anton Designs, Ltd.	Madrid	N/A	Spain
Asian Shopping Network, Co	Singapore	N/A	Singapore

Reference

1. https://www.slideshare.net/imamhossain75054/presentations
2. https://www.w3resource.com/mysql/creating-table-advance/constraint.php