

# SQL Week 6

---

# Topics List

---

- Create

- Alter

- Drop

# SQL Week 6

## Create Database

---

- To create a database (or schema), use:

```
CREATE DATABASE IF NOT EXISTS database_name;
```

Or

```
CREATE SCHEMA IF NOT EXISTS schema_name;
```

# SQL Week 6

## Create Table

---

- To create a table use:

```
CREATE TABLE [IF NOT EXISTS] table_name(  
    column_list  
    ) engine=table_type;
```

# SQL Week 6

## Create Table

---

- Let's examine the syntax in greater detail:
  - First, you specify the name of table that you want to create after the *CREATE TABLE* clause. The table name must be unique within the database.
  - Second, you specify a list of columns for the table in the *column\_list* section. Columns are separated by a comma (,).
  - Third, you may specify the storage engine for the table in the *engine* clause. You can use any storage engine such as InnoDB, MyISAM, etc... If you don't explicitly declare the storage engine, MySQL will use InnoDB by default.

# SQL Week 6

## Create Table

---

```
create table Book (  
ISBN varchar(15) not null,  
title varchar(50) not null,  
publisher varchar(30),  
publishedDate date,  
category varchar(30),  
price decimal(5,2),  
primary key (ISBN)  
);
```

# SQL Week 6

## Defining columns

---

- To define a column for the table in the *CREATE TABLE* statement, you use the following syntax:

```
column_name data_type[size] [NOT NULL|NULL] [DEFAULT value]  
[AUTO_INCREMENT]
```

- The most important components of the syntax above are:
  - *column\_name* specifies the name of the column. Each column must be always associated with a specific data type and the size.
  - *NOT NULL* or *NULL* indicates that the column accepts a *NULL* value or not.
  - *DEFAULT value* is used to specify the default value of the column.
  - *AUTO\_INCREMENT* indicates that the value of column is increased by one whenever a new record is inserted into the table.

# SQL Week 6

## Text data types

---

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



# SQL Week 6

## Numeric data types

---

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

# SQL Week 6

## Date data types

---

<b>DATE</b>	YYYY-MM-DD.
<b>DATETIME</b>	YYYY-MM-DD HH:MM:SS.
<b>TIMESTAMP</b>	YYYYMMDDHHMMSS.
<b>TIME</b>	HH:MM:SS.

# SQL Week 6

## Keys

---

- **Primary key**

- If you want to set particular columns of the table as the primary key, you use the following syntax:

```
PRIMARY KEY (col1,[col2,...])
```

- **Foreign key**

- To set a foreign key, we explicitly link an attribute in one table with the primary key value of another table.

```
constraint fk_book foreign key(ISBN) references Book(ISBN)  
on update cascade  
on delete no action
```

# SQL Week 6

## Keys

---

```
create table BookCopy(  
  copyId int auto_increment not null,  
  ISBN varchar(15) not null,  
  dateAcquired date not null,  
  dateDestroyed date,  
  primary key (copyId),  
  constraint fk_book foreign key(ISBN) references Book(ISBN)  
  on update cascade  
  on delete no action  
);
```

# Topics List

---

- Create

- Alter

- Drop

# SQL Week 6

## Alter

---

- To alter a table, you can use the any of the following commands:

Add a column:

```
ALTER TABLE table_name  
ADD column_name datatype
```

Delete a column:

```
ALTER TABLE table_name  
DROP COLUMN column_name
```

Modify a column:

```
ALTER TABLE table_name  
CHANGE COLUMN old_column_name new_column_name datatype
```

or

```
ALTER TABLE table_name  
MODIFY COLUMN column_name datatype
```

# Topics List

---

- Create
- Alter
- Drop

# SQL Week 6

## Drop

---

- To drop a database, execute either of the following commands:

```
DROP DATABASE database_name;
```

```
DROP SCHEMA schema_name;
```

- To drop a table, you can use the following command:

```
DROP TABLE table_name;
```