

# MySQL\_Commands

Saturday, November 13, 2021 12:09 AM

Table of contents	
<b>1. Creating databases and table</b> 1. Create database 2. Create table 3. Drop 4. Use 5. Select 6. Data types a. Int b. Varchar	
<b>2. Data insertion and table structure</b> 1. Insert 2. Multiple insert 3. Warnings 4. Not null 5. Default 6. Primary key	
<b>3. CRUD commands</b> 1. Create 2. Select and where clause 3. Read 4. Update 5. Delete	
Commands	Output/Exp
<code>show databases; #shows available databases</code>	<pre>+-----+   Database   +-----+   information_schema     mysql     performance_schema     sakila     sys     world   +-----+ 6 rows in set (0.00 sec)</pre>
<code>create database test_database; # create database</code>	
<code>drop database test_database; #delete database</code>	
<code>use test_database; #use the selected database</code>	
<code>select database(); # tells the currently selected database</code>	<pre>+-----+   database()   +-----+   my_pets   +-----+ 1 row in set (0.00 sec)</pre>
<code>create table cats( age int, address varchar(100) ); #creating table with name cats</code>	
<code>show tables; #shows available tables in the selected database.</code>	<pre>+-----+   Tables_in_my_pets   +-----+   cats   +-----+ 1 row in set (0.01 sec)</pre>
<code>show columns from cats # shows columns from the table cats. OR</code>	<pre>+-----+-----+-----+-----+-----+-----+   Field   Type   Null   Key   Default   Extra   +-----+-----+-----+-----+-----+-----+   name   varchar(100)   YES     NULL       age   int   YES     NULL     +-----+-----+-----+-----+-----+-----+</pre>
<code>desc cats; #performs the same action as above describes the table.</code>	<pre>2 rows in set (0.02 sec)</pre>
<code>drop table cats; # deletes the table cats from the database.</code>	
<b>Inserting data in the tables</b>	
<code>insert into cats (name, age) values ("jetson", 7); #this will insert data into already existing table.</code>	
<code>select * from cats;</code>	<pre>+-----+-----+   name   age   +-----+-----+   jetson   8     victoria   6   +-----+-----+ 2 rows in set (0.01 sec)</pre>
<b>Multiple insert:</b>	
<code>insert into cats (name, age) # don't forget to write table values ("tim", 4), #name. ("john", 5),</code>	

<pre>("katy", 9), ("lens", 20);</pre>	
<pre>show warnings; # shows you warnings.</pre>	
<pre>insert into cats (name) values ("cluadia");  # NULL is yes in the table. It means its ok to have unknown value. # NULL not means its 0.</pre>	<pre>+-----+-----+-----+-----+-----+-----+   Field   Type        Null   Key   Default   Extra   +-----+-----+-----+-----+-----+-----+   name    varchar(50)   YES          NULL               age     int           YES          NULL             +-----+-----+-----+-----+-----+-----+ 2 rows in set (0.00 sec)</pre>
<pre>create table cats2 ( name varchar(50) not null, age int not null ); # this will ensure that name and age columns don't have null values. Default value is speciefies if nothing is provided.</pre>	
<pre>create table cats2 ( name varchar(50) default "name not specified", age int default 20 ); # here if any column entry is null/not provided then default value is replaced.</pre>	
<pre>create table cats2 ( name varchar(50) not null default "name is not speciefies", age int not null default 20 ); # here you can't write null values and if no value provided then replaced by default value.</pre>	<pre>+-----+-----+-----+-----+-----+-----+   Field   Type        Null   Key   Default        Extra   +-----+-----+-----+-----+-----+-----+   name    varchar(50)   NO           name not specified            age     int           NO           20                     +-----+-----+-----+-----+-----+-----+ 2 rows in set (0.01 sec)</pre>
<pre>create table unique_cats ( cat_id int not null, name varchar(50), age int, primary key (cat_id) ); # primary key is unique to each entry.  # auto_increment will increment id as more entries comes #in automatically.  create table employees( id int auto_increment not null, first_name varchar(50), last_name varchar(50), middle_name varchar(50), current_status varchar(50) not null default "employed", primary key(id) );  insert into employees(id, first_name, last_name, current_status) values(1, "dolly", "devil", "internship");</pre>	<pre>+-----+-----+-----+-----+-----+-----+   Field   Type        Null   Key   Default        Extra        +-----+-----+-----+-----+-----+-----+   id      int         NO     PRI   NULL           auto_increment     first_name   varchar(50)   YES          NULL                    last_name    varchar(50)   YES          NULL                    middle_name   varchar(50)   YES          NULL                    current_status   varchar(50)   NO           employed              +-----+-----+-----+-----+-----+-----+ 5 rows in set (0.00 sec)  mysql&gt; select * from employees; +-----+-----+-----+-----+-----+   id   first_name   last_name   middle_name   current_status   +-----+-----+-----+-----+-----+   1   dolly        devil       NULL          internship        +-----+-----+-----+-----+-----+ 1 row in set (0.01 sec)</pre>
<p><b>CRUD Commands(Create, Read, Update, Delete):</b></p> <pre># inserting data in cats table insert into cats(name, breed, age) values('Ringo', 'Tabby', 4), ('Cindy', 'Maine Coon', 10), ('Dumbledore', 'Maine Coon', 11), ('Egg', 'Persian', 4), ('Misty', 'Tabby', 13), ('George Michael', 'Ragdoll', 9), ('Jackson', 'Sphynx', 7);</pre>	
<p><b>Select statement</b></p> <pre>select * from cats; # gives us all the rows in the cats table.</pre>	<pre>+-----+-----+-----+-----+   cat_id   name        breed        age   +-----+-----+-----+-----+   1        Ringo       Tabby        4       2        Cindy       Maine Coon   10      3        Dumbledore   Maine Coon   11      4        Egg         Persian      4       5        Misty       Tabby        13      6        George Michael   Ragdoll     9       7        Jackson     Sphynx       7     +-----+-----+-----+-----+ 7 rows in set (0.01 sec)</pre>
<pre>select name from cats; #Accessing specific columns using #select statement.</pre>	<pre>+-----+   name   +-----+   Ringo     Cindy     Dumbledore     Egg     Misty     George Michael     Jackson   +-----+</pre>

	7 rows in set (0.00 sec)
select name, age from cats;#selecting multiple columns at once. Here order matters as in the next query.	<pre> +-----+-----+   name        age   +-----+-----+   Ringo        4       Cindy        10      Dumbledore   11      Egg          4       Misty        13      George Michael   9       Jackson      7     +-----+-----+ 7 rows in set (0.00 sec) </pre>
select breed, age, name from cats;	<pre> +-----+-----+-----+   breed    age   name        +-----+-----+-----+   Tabby     4     Ringo          Maine Coon   10    Cindy          Maine Coon   11    Dumbledore     Persian   4     Egg            Tabby     13    Misty          Ragdoll   9     George Michael     Sphynx    7     Jackson      +-----+-----+-----+ 7 rows in set (0.00 sec) </pre>
<b>Where clause</b>  select * from cats where age=4;	<pre> +-----+-----+-----+-----+   cat_id   name   breed   age   +-----+-----+-----+-----+   1        Ringo   Tabby   4       4        Egg     Persian   4     +-----+-----+-----+-----+ 2 rows in set (0.00 sec) </pre>
select * from cats where name='Egg'; # you can also write "egg", capital letter #doesn't affect the query.	<pre> +-----+-----+-----+-----+   cat_id   name   breed   age   +-----+-----+-----+-----+   4        Egg     Persian   4     +-----+-----+-----+-----+ 1 row in set (0.00 sec) </pre>
<b>Some practice queries-Select, Where:</b> select cat_id from cats;	<pre> +-----+   cat_id   +-----+   1          2          3          4          5          6          7        +-----+ 7 rows in set (0.00 sec) </pre>
select name, breed from cats;	<pre> +-----+-----+   name        breed   +-----+-----+   Ringo        Tabby     Cindy        Maine Coon     Dumbledore   Maine Coon     Egg          Persian     Misty        Tabby     George Michael   Ragdoll     Jackson      Sphynx   +-----+-----+ 7 rows in set (0.00 sec) </pre>
select name, age from cats where breed='Tabby';	<pre> +-----+-----+   name   age   +-----+-----+   Ringo   4       Misty   13    +-----+-----+ 2 rows in set (0.00 sec) </pre>
select cat_id, age from cats where cat_id=age;	<pre> +-----+-----+   cat_id   age   +-----+-----+   4        4       7        7     +-----+-----+ 2 rows in set (0.00 sec) </pre>
<b>Aliases:</b> select cat_id as id, name as cats_names from cats; #aliases only changes name of the column for showing original column name are not changed.	<pre> +-----+-----+   id   cats_names   +-----+-----+   1    Ringo          2    Cindy          3    Dumbledore     4    Egg            5    Misty          6    George Michael     7    Jackson      +-----+-----+ 7 rows in set (0.00 sec) </pre>
<b>Update statement:</b> Keep in mind! do check before updating that you are updating the right entries, same goes for delete statement.  update cats set breed='Shorthair' where breed='Tabby';#changing breed from 'tabby' to 'shorthair'.	<pre> +-----+-----+-----+-----+   cat_id   name        breed        age   +-----+-----+-----+-----+   1        Ringo        Shorthair    4       2        Cindy        Maine Coon   10      3        Dumbledore   Maine Coon   11      4        Egg          Persian      4     +-----+-----+-----+-----+ </pre>

	<pre> 5   Misty            Shorthair   13   6   George Michael   Ragdoll     9    7   Jackson         Sphynx      7    +-----+-----+-----+ 7 rows in set (0.01 sec) </pre>
<pre> update cats set age=14 where name='Misty'; # change age #from 13 to 14. </pre>	<pre> +-----+-----+-----+   cat_id   name            breed        age   +-----+-----+-----+   1        Ringo           Shorthair    4       2        Cindy           Maine Coon   10      3        Dumbledore      Maine Coon   11      4        Egg             Persian      4       5        Misty           Shorthair    14      6        George Michael   Ragdoll      9       7        Jackson         Sphynx       7     +-----+-----+-----+ 7 rows in set (0.00 sec) </pre>
<p><b>Some practice queries-Update:</b></p> <pre> update cats set name='Jack' where name='jackson'; # update #'jackson' to 'jack' </pre>	<pre> +-----+-----+-----+   cat_id   name            breed        age   +-----+-----+-----+   1        Ringo           Shorthair    4       2        Cindy           Maine Coon   10      3        Dumbledore      Maine Coon   11      4        Egg             Persian      4       5        Misty           Shorthair    14      6        George Michael   Ragdoll      9       7        Jack            Sphynx       7     +-----+-----+-----+ 7 rows in set (0.00 sec) </pre>
<pre> update cats set breed='British Shorthair' where name='Ringo'; # update 'Ringo' # breed to 'British #Shorthair'. </pre>	<pre> +-----+-----+-----+   cat_id   name            breed            age   +-----+-----+-----+   1        Ringo           British Shorthair   4       2        Cindy           Maine Coon        10      3        Dumbledore      Maine Coon        11      4        Egg             Persian           4       5        Misty           Shorthair         14      6        George Michael   Ragdoll           9       7        Jack            Sphynx            7     +-----+-----+-----+ 7 rows in set (0.00 sec) </pre>
<pre> update cats set age=12 where breed='Maine Coon'; # update #'Maine Coon' age to 12. </pre>	<pre> +-----+-----+-----+   cat_id   name            breed            age   +-----+-----+-----+   1        Ringo           British Shorthair   4       2        Cindy           Maine Coon        12      3        Dumbledore      Maine Coon        12      4        Egg             Persian           4       5        Misty           Shorthair         14      6        George Michael   Ragdoll           9       7        Jack            Sphynx            7     +-----+-----+-----+ 7 rows in set (0.00 sec) </pre>
<p><b>Delete statement:</b> Before deleting something it is a good practice that you check what are going to delete by using select statement.</p> <pre> delete from cats where name='egg'; # note that the cat_id #4 no longer existing. </pre>	<pre> +-----+-----+-----+   cat_id   name            breed            age   +-----+-----+-----+   1        Ringo           British Shorthair   4       2        Cindy           Maine Coon        12      3        Dumbledore      Maine Coon        12      5        Misty           Shorthair         14      6        George Michael   Ragdoll           9       7        Jack            Sphynx            7     +-----+-----+-----+ 6 rows in set (0.01 sec) </pre>
<pre> delete cats; #this will delete all the data inside the #table but the table structure still exist you can put #data inside it. # drop table will entirely remove your table. </pre>	
<pre> drop database &lt;name of database&gt;; #delete database </pre>	
<p><b>Some practice queries-Delete:</b></p> <pre> delete from cats where age=4; </pre>	<pre> +-----+-----+-----+   cat_id   name            breed        age   +-----+-----+-----+   2        Cindy           Maine Coon   12      3        Dumbledore      Maine Coon   12      5        Misty           Shorthair    14      6        George Michael   Ragdoll      9       7        Jack            Sphynx       7     +-----+-----+-----+ 5 rows in set (0.00 sec) </pre>
<pre> delete from cats where age=cat_id; # deletes data where #age and cat_id are same. </pre>	<pre> +-----+-----+-----+   cat_id   name            breed        age   +-----+-----+-----+   2        Cindy           Maine Coon   12      3        Dumbledore      Maine Coon   12      5        Misty           Shorthair    14      6        George Michael   Ragdoll      9     +-----+-----+-----+ 4 rows in set (0.00 sec) </pre>
<pre> delete from cats; # deletes all data from the table. Table still exists. </pre>	<pre> Empty set (0.00 sec) </pre>
<p><b>Concatenation</b></p> <pre> select </pre>	<pre> +-----+   full_name   +-----+   Thumna Lahiri   </pre>

```
concat(author_fname, ' ', author_lname)
as full_name from books; #combines two columns/strings.
```

```

+-----+
| Jhumpa Lahiri |
| Neil Gaiman   |
| Neil Gaiman   |
| Jhumpa Lahiri |
| Dave Eggers   |
| Dave Eggers   |
| Michael Chabon|
| Patti Smith   |
| Dave Eggers   |
| Neil Gaiman   |
| Raymond Carver|
| Raymond Carver|
| Don DeLillo   |
| John Steinbeck|
| David Foster Wallace|
| David Foster Wallace|
+-----+

```

16 rows in set (0.00 sec)

```
select author_fname as first, author_lname as last,
concat (author_fname, ' ', author_lname) as full
from books; # concatenating two columns and aliasing names as
#full in the 3rd column.
```

```

+-----+-----+-----+
| first | last  | full  |
+-----+-----+-----+
| Jhumpa | Lahiri | Jhumpa Lahiri |
| Neil   | Gaiman | Neil Gaiman   |
| Neil   | Gaiman | Neil Gaiman   |
| Dave   | Eggers | Dave Eggers   |
| Neil   | Gaiman | Neil Gaiman   |
| Raymond | Carver | Raymond Carver |
| Raymond | Carver | Raymond Carver |
| Don     | DeLillo | Don DeLillo   |
| John    | Steinbeck | John Steinbeck |
| David   | Foster Wallace | David Foster Wallace |
| David   | Foster Wallace | David Foster Wallace |
+-----+-----+-----+

```

16 rows in set (0.00 sec)

### Concat with separator

```
select
concat_ws('-', title, author_fname, author_lname) as full
from books; # it will put the '-' after each concatenation #so
you don't need to put separator each time.
```

```

+-----+
| full |
+-----+
| The Namesake-Jhumpa-Lahiri |
| Norse Mythology-Neil-Gaiman |
| American Gods-Neil-Gaiman |
| Interpreter of Maladies-Jhumpa-Lahiri |
| A Hologram for the King: A Novel-Dave-Eggers |
| The Circle-Dave-Eggers |
| The Amazing Adventures of Kavalier & Clay-Michael-Chabon |
| Just Kids-Patti-Smith |
| A Heartbreaking Work of Staggering Genius-Dave-Eggers |
| Coraline-Neil-Gaiman |
| What We Talk About When We Talk About Love: Stories-Raymond-Carver |
| Where I'm Calling From: Selected Stories-Raymond-Carver |
| White Noise-Don-DeLillo |
| Cannery Row-John-Steinbeck |
| Oblivion: Stories-David-Foster Wallace |
| Consider the Lobster-David-Foster Wallace |
+-----+

```

16 rows in set (0.00 sec)

### Substring

```
select substring('Hello World', 1, 4); # unlike python index
start from 1 not 0.
```

```

+-----+
| substring('Hello World', 1, 4) |
+-----+
| Hell |
+-----+

```

1 row in set (0.00 sec)

```
select substring('Hello World', -3);
```

```

+-----+
| substring('Hello World', -3) |
+-----+
| rld |
+-----+

```

1 row in set (0.01 sec)

```
select substring('Hello World', 4); # if we pass one value
it start from that index to end.
```

```

+-----+
| substring('Hello World', 4) |
+-----+
| lo World |
+-----+

```

OR

```
Select substring ('Hello World', 4, 8); # these both are same
produces same results.
```

1 row in set (0.00 sec)

OR

```
substr('Hello World', 4, 8) # substr() function.
```

```
select substring(title, 1, 5) from books;
```

```

+-----+
| substring(title, 1, 5) |
+-----+
| The N |
| Norse |
| Ameri |
| Inter |
| A Hol |
| The C |
| The A |
| Just |
| A Hea |
| Coral |
| What |
| Where |
+-----+

```

	<pre>   White   Canne   Obliv   Consi +-----+ </pre>
<pre> select substring(title, 1, 5) short_title from books; </pre>	<pre> +-----+   short_title   +-----+   The N   Norse   Ameri   Inter   A Hol   The C   The A   Just   A Hea   Coral   What   Where   White   Canne   Obliv   Consi +-----+ 16 rows in set (0.00 sec) </pre>
<pre> select concat(substr(title, 1, 5), '...') as short_title from books; </pre>	<pre> +-----+   short_title   +-----+   The N...   Norse...   Ameri...   Inter...   A Hol...   The C...   The A...   Just ...   A Hea...   Coral...   What ...   Where...   White...   Canne...   Obliv...   Consi... +-----+ 16 rows in set (0.00 sec) </pre>
<p><b>Replace</b></p> <pre> select replace('Hello World', 'l', '0'); </pre>	<pre> +-----+   replace('Hello World', 'l', '0')   +-----+   He00o Wor0d +-----+ 1 row in set (0.00 sec) </pre>
<pre> select replace('Hello World', 'o', '*'); # replace is case- sensitive only lower case 'o' got replaced. </pre>	<pre> +-----+   replace('Hello World', 'o', '*')   +-----+   Hello W*rld +-----+ 1 row in set (0.00 sec) </pre>
<pre> select replace('cheese bread coffe milk', ' ', ' and '); </pre>	<pre> +-----+   replace('cheese bread coffe milk', ' ', ' and ')   +-----+   cheese and bread and coffe and milk +-----+ 1 row in set (0.00 sec) </pre>
<p><b>Reverse clause</b></p> <pre> select reverse('Hello World'); </pre>	<pre> +-----+   reverse('Hello World')   +-----+   dlrow olleH +-----+ 1 row in set (0.01 sec) </pre>
<p><b>Character length</b></p> <pre> select char_length('Hello World'); </pre>	<pre> +-----+   char_length('Hello World')   +-----+   11 +-----+ 1 row in set (0.00 sec) </pre>
<pre> select author_fname, char_length(author_fname) as length from books; </pre>	<pre> +-----+-----+   author_fname   length   +-----+-----+   Jhumpa        6          Neil          4          Neil          4          Jhumpa        6          Dave          4          Dave          4          Michael       7          Patti         5          Dave          4          Neil          4          Raymond       7          Raymond       7        +-----+-----+ </pre>

	<pre>   Don        3     John       4     David      5     David      5   +-----+ 16 rows in set (0.00 sec) </pre>
select upper('hellow world');	<pre> +-----+   upper('hellow world')   +-----+   HELLOW WORLD            +-----+ 1 row in set (0.00 sec) </pre>
select lower('hellow world');	<pre> +-----+   lower('hellow world')   +-----+   hellow world            +-----+ 1 row in set (0.00 sec) </pre>
select concat('my favorite book is ', upper(title)) as titles from books;	<pre> +-----+   titles +-----+   my favorite book is THE NAMESAKE   my favorite book is NORSE MYTHOLOGY   my favorite book is AMERICAN GODS   my favorite book is INTERPRETER OF MALADIES   my favorite book is A HOLOGRAM FOR THE KING: A NOVEL   my favorite book is THE CIRCLE   my favorite book is THE AMAZING ADVENTURES OF KAVALIER &amp; CLAY   my favorite book is JUST KIDS   my favorite book is A HEARTBREAKING WORK OF STAGGERING GENIUS   my favorite book is CORALINE   my favorite book is WHAT WE TALK ABOUT WHEN WE TALK ABOUT LOVE: STORIES   my favorite book is WHERE I'M CALLING FROM: SELECTED STORIES   my favorite book is WHITE NOISE   my favorite book is CANNERY ROW   my favorite book is OBLIVION: STORIES   my favorite book is CONSIDER THE LOBSTER +-----+ 16 rows in set (0.00 sec) </pre>
<p>Some practice String Functions:</p> <pre> select reverse( upper('hello there im playing with sql ') ) as reverse_upper; # reverse the string alongwith #capitalize it and aliase column name. </pre>	<pre> +-----+   reverse_upper            +-----+   LQS HTIW GNIYALP MI EREHT OLLEH   +-----+ 1 row in set (0.00 sec) </pre>
select replace(concat('I', ' ', 'like', ' ', 'cats'), ' ', '_'); # concatenate the strings and then replace spaces in #that string with '_'.	<pre> +-----+   replace(concat('I', ' ', 'like', ' ', 'cats'), ' ', '_')   +-----+   I_like_cats            +-----+ 1 row in set (0.00 sec) </pre>
select replace(title, ' ', '->') as title from books;	<pre> +-----+   title +-----+   The-&gt;Namesake   Norse-&gt;Mythology   American-&gt;Gods   Interpreter-&gt;of-&gt;Maladies   A-&gt;Hologram-&gt;for-&gt;the-&gt;King-&gt;-&gt;A-&gt;Novel   The-&gt;Circle   The-&gt;Amazing-&gt;Adventures-&gt;of-&gt;Kavalier-&gt;&amp;-&gt;Clay   Just-&gt;Kids   A-&gt;Heartbreaking-&gt;Work-&gt;of-&gt;Staggering-&gt;Genius   Coraline   What-&gt;We-&gt;Talk-&gt;About-&gt;When-&gt;We-&gt;Talk-&gt;About-&gt;Love-&gt;-&gt;Stories   Where-&gt;I'm-&gt;Calling-&gt;From-&gt;-&gt;Selected-&gt;Stories   White-&gt;Noise   Cannery-&gt;Row   Oblivion-&gt;-&gt;Stories   Consider-&gt;the-&gt;Lobster +-----+ 16 rows in set (0.00 sec) </pre>
select author_fname as forwards, reverse(author_fname) as backwards from books;	<pre> +-----+   forwards   backwards   +-----+   Jhumpa     apmuhJ       Neil       lieN         Neil       lieN         Jhumpa     apmuhJ       Dave       evaD         Dave       evaD         Michael    leahciM      Patti      ittaP        Dave       evaD         Neil       lieN         Raymond    dnomyaR      Raymond    dnomyaR      Don        noD          John       nhoJ         David      divaD         David      divaD       +-----+ 16 rows in set (0.00 sec) </pre>

```
select upper(concat(author_fname, ' ', author_lname)) as 'full
name in caps'
from books;
```

```
+-----+
| full name in caps |
+-----+
| JHUMPA LAHIRI     |
| NEIL GAIMAN        |
| NEIL GAIMAN        |
| JHUMPA LAHIRI     |
| DAVE EGGERS        |
| DAVE EGGERS        |
| MICHAEL CHABON     |
| PATTI SMITH        |
| DAVE EGGERS        |
| NEIL GAIMAN        |
| RAYMOND CARVER     |
| RAYMOND CARVER     |
| DON DELILLO        |
| JOHN STEINBECK     |
| DAVID FOSTER WALLACE |
| DAVID FOSTER WALLACE |
+-----+
16 rows in set (0.00 sec)
```

```
select title,
char_length(title) as 'character count'
from books;
```

```
+-----+-----+
| title | character count |
+-----+-----+
| The Namesake | 12 |
| Norse Mythology | 15 |
| American Gods | 13 |
| Interpreter of Maladies | 23 |
| A Hologram for the King: A Novel | 32 |
| The Circle | 10 |
| The Amazing Adventures of Kavalier & Clay | 41 |
| Just Kids | 9 |
| A Heartbreaking Work of Staggering Genius | 41 |
| Coraline | 8 |
| What We Talk About When We Talk About Love: Stories | 51 |
| Where I'm Calling From: Selected Stories | 40 |
| White Noise | 11 |
| Cannery Row | 11 |
| Oblivion: Stories | 17 |
| Consider the Lobster | 20 |
+-----+-----+
16 rows in set (0.00 sec)
```

```
select concat(substring(title, 1, 10), '...') as "short title",
concat(author_lname, ',', author_fname) as author,
concat(stock_quantity, ' in stock') as quantity
from books;
```

```
+-----+-----+-----+
| short title | author | quantity |
+-----+-----+-----+
| The Namesa... | Lahiri,Jhumpa | 32 in stock |
| Norse Myth... | Gaiman,Neil | 43 in stock |
| American G... | Gaiman,Neil | 12 in stock |
| Interprete... | Lahiri,Jhumpa | 97 in stock |
| A Hologram... | Eggers,Dave | 154 in stock |
| The Circle... | Eggers,Dave | 26 in stock |
| The Amazin... | Chabon,Michael | 68 in stock |
| Just Kids... | Smith,Patti | 55 in stock |
| A Heartbre... | Eggers,Dave | 104 in stock |
| Coraline... | Gaiman,Neil | 100 in stock |
| What We Ta... | Carver,Raymond | 23 in stock |
| Where I'm ... | Carver,Raymond | 12 in stock |
| White Nois... | DeLillo,Don | 49 in stock |
| Cannery Ro... | Steinbeck,John | 95 in stock |
| Oblivion: ... | Foster Wallace,David | 172 in stock |
| Consider t... | Foster Wallace,David | 92 in stock |
+-----+-----+-----+
16 rows in set (0.00 sec)
```

## Distinct

```
select distinct author_fname from books; # selects unique
values from the #author_fname column.
```

```
+-----+
| author_fname |
+-----+
| Jhumpa       |
| Neil         |
| Dave         |
| Michael      |
| Patti        |
| Raymond      |
| Don          |
| John         |
| David        |
| Dan          |
| Freida       |
| George       |
+-----+
12 rows in set (0.00 sec)
```

```
select distinct author_fname, author_lname from books;
#distinct is applied to #both author_fname, author_lname
#columns.
```

```
+-----+-----+
| author_fname | author_lname |
+-----+-----+
| Jhumpa       | Lahiri       |
| Neil         | Gaiman       |
| Dave         | Eggers       |
| Michael      | Chabon       |
| Patti        | Smith        |
| Raymond      | Carver       |
| Don          | DeLillo      |
| John         | Steinbeck    |
| David        | Foster Wallace |
| Dan          | Harris       |
| Freida       | Harris       |
| George       | Saunders     |
+-----+-----+
12 rows in set (0.00 sec)
```



Order by

select author\_lname from books order by author\_lname; # by default the order is in ascending order alphabetical.

```
+-----+
| author_lname |
+-----+
| Carver       |
| Carver       |
| Chabon       |
| DeLillo      |
| Eggers       |
| Eggers       |
| Eggers       |
| Foster Wallace |
| Foster Wallace |
| Gaiman       |
| Gaiman       |
| Gaiman       |
| Harris       |
| Harris       |
| Lahiri       |
| Lahiri       |
| Saunders     |
| Smith        |
| Steinbeck    |
+-----+
19 rows in set (0.01 sec)
```

select author\_lname from books order by author\_lname desc;#desc will order by author\_lname in descending order.

```
+-----+
| author_lname |
+-----+
| Steinbeck    |
| Smith        |
| Saunders     |
| Lahiri       |
| Lahiri       |
| Harris       |
| Harris       |
| Gaiman       |
| Gaiman       |
| Gaiman       |
| Foster Wallace |
| Foster Wallace |
| Eggers       |
| Eggers       |
| Eggers       |
| DeLillo      |
| Chabon       |
| Carver       |
| Carver       |
+-----+
19 rows in set (0.00 sec)
```

select title, released\_year, pages from books order by pages;

```
+-----+-----+-----+
| title                                     | released_year | pages |
+-----+-----+-----+
| What We Talk About When We Talk About Love: Stories | 1981          | 176   |
| Cannery Row                                       | 1945          | 181   |
| Interpreter of Maladies                         | 1996          | 198   |
| Coraline                                         | 2003          | 208   |
| 10% Happier                                     | 2014          | 256   |
| The Namesake                                    | 2003          | 291   |
| Norse Mythology                                 | 2016          | 304   |
| Just Kids                                       | 2010          | 304   |
| White Noise                                     | 1985          | 320   |
| Oblivion: Stories                             | 2004          | 329   |
| Consider the Lobster                           | 2005          | 343   |
| A Hologram for the King: A Novel                | 2012          | 352   |
| Lincoln In The Bardo                           | 2017          | 367   |
| fake_book                                       | 2001          | 428   |
| A Heartbreaking Work of Staggering Genius       | 2001          | 437   |
| American Gods                                  | 2001          | 465   |
| The Circle                                     | 2013          | 504   |
| Where I'm Calling From: Selected Stories        | 1989          | 526   |
| The Amazing Adventures of Kavalier & Clay       | 2000          | 634   |
+-----+-----+-----+
19 rows in set (0.00 sec)
```

select title, released\_year, pages from books order by 2;  
# here 2 means the 2nd position element which is #released\_year in this query.

```
+-----+-----+-----+
| title                                     | released_year | pages |
+-----+-----+-----+
| Cannery Row                                       | 1945          | 181   |
| What We Talk About When We Talk About Love: Stories | 1981          | 176   |
| White Noise                                     | 1985          | 320   |
| Where I'm Calling From: Selected Stories        | 1989          | 526   |
| Interpreter of Maladies                         | 1996          | 198   |
| The Amazing Adventures of Kavalier & Clay       | 2000          | 634   |
| American Gods                                  | 2001          | 465   |
| A Heartbreaking Work of Staggering Genius       | 2001          | 437   |
| fake_book                                       | 2001          | 428   |
| The Namesake                                    | 2003          | 291   |
| Coraline                                         | 2003          | 208   |
| Oblivion: Stories                             | 2004          | 329   |
| Consider the Lobster                           | 2005          | 343   |
| Just Kids                                       | 2010          | 304   |
| A Hologram for the King: A Novel                | 2012          | 352   |
| The Circle                                     | 2013          | 504   |
| 10% Happier                                     | 2014          | 256   |
| Norse Mythology                                 | 2016          | 304   |
| Lincoln In The Bardo                           | 2017          | 367   |
+-----+-----+-----+
19 rows in set (0.00 sec)
```

select author\_fname, author\_lname from books order by

```
+-----+-----+
| author_fname | author_lname |
+-----+-----+
```

author\_lname, author\_fname;

author_fname	author_lname
Raymond	Carver
Raymond	Carver
Michael	Chabon
Don	DeLillo
Dave	Eggers
Dave	Eggers
Dave	Eggers
David	Foster Wallace
David	Foster Wallace
Neil	Gaiman
Neil	Gaiman
Neil	Gaiman
Dan	Harris
Freida	Harris
Jhumpa	Lahiri
Jhumpa	Lahiri
George	Saunders
Patti	Smith
John	Steinbeck

19 rows in set (0.00 sec)

### Limit

```
select title, released_year from books order by released_year
limit 3;
```

title	released_year
Cannery Row	1945
What We Talk About When We Talk About Love: Stories	1981
White Noise	1985

3 rows in set (0.00 sec)

```
select title,
released_year
from books
order by released_year desc
limit 0, 5;
#index start from 0 irrespective of start index of string
methods from 1.
```

title	released_year
Lincoln In The Bardo	2017
Norse Mythology	2016
10% Happier	2014
The Circle	2013
A Hologram for the King: A Novel	2012

5 rows in set (0.00 sec)

```
select title,
released_year
from books
order by released_year desc
limit 1, 5; # start from index 1 and give next 5 values
starting from it.
```

title	released_year
Norse Mythology	2016
10% Happier	2014
The Circle	2013
A Hologram for the King: A Novel	2012
Just Kids	2010

5 rows in set (0.00 sec)

```
select title,
released_year
from books
order by released_year desc
limit 5, 3251525; #if want to get all the entries after some index
then pass a large number on the second parameter.
```

title	released_year
Just Kids	2010
Consider the Lobster	2005
Oblivion: Stories	2004
The Namesake	2003
Coraline	2003
American Gods	2001
A Heartbreaking Work of Staggering Genius	2001
fake_book	2001
The Amazing Adventures of Kavalier & Clay	2000
Interpreter of Maladies	1996
Where I'm Calling From: Selected Stories	1989
White Noise	1985
What We Talk About When We Talk About Love: Stories	1981
Cannery Row	1945

14 rows in set (0.00 sec)

### Like clause

```
select title,
author_fname
from books
where author_fname like '%da%'; #gives author_fname with
#specific 'da' having in it. % is called wildcard meaning
#something before 'da' and something after 'da'.
```

title	author_fname
A Hologram for the King: A Novel	Dave
The Circle	Dave
A Heartbreaking Work of Staggering Genius	Dave
Oblivion: Stories	David
Consider the Lobster	David
10% Happier	Dan
fake_book	Freida

7 rows in set (0.00 sec)

```
select title,
author_fname
from books
where author_fname like 'da%'; #find words start with 'da' and
#something after it.
```

title	author_fname
A Hologram for the King: A Novel	Dave
The Circle	Dave
A Heartbreaking Work of Staggering Genius	Dave
Oblivion: Stories	David
Consider the Lobster	David
10% Happier	Dan

6 rows in set (0.00 sec)

```
select title,
author_fname
from books
```

title	author_fname
-------	--------------

where title like '%the%';	<pre>   The Namesake   Jhumpa     A Hologram for the King: A Novel   Dave     The Circle   Dave     The Amazing Adventures of Kavalier &amp; Clay   Michael     Consider the Lobster   David     Lincoln In The Bardo   George   +-----+-----+ 6 rows in set (0.00 sec) </pre>
<pre> select title, stock_quantity from books where stock_quantity like '____'; #this format has 4 underscores and in like will search for 4 character long values. </pre>	<pre> +-----+-----+   title   stock_quantity   +-----+-----+   Lincoln In The Bardo   1000   +-----+-----+ 1 row in set (0.00 sec) </pre>
<pre> select title from books where title like '%\%%'; #If we have specific '%' character in the book title name. </pre>	<pre> +-----+   title   +-----+   10% Happier   +-----+ 1 row in set (0.00 sec) </pre>
<pre> select title from books where title like '%\_%'; #If we have specific '_' character in the book title name. </pre>	<pre> +-----+   title   +-----+   fake_book   +-----+ 1 row in set (0.00 sec) </pre>
<p><b>Some practice refining selections problems:</b></p> <pre> select title from books where title like '%stories%'; #find title #with 'stories' in it. </pre>	<pre> +-----+   title   +-----+   What We Talk About When We Talk About Love: Stories     Where I'm Calling From: Selected Stories     Oblivion: Stories   +-----+ 3 rows in set (0.00 sec) </pre>
<pre> select title, pages from books order by pages desc limit 1; </pre>	<pre> +-----+-----+   title   pages   +-----+-----+   The Amazing Adventures of Kavalier &amp; Clay   634   +-----+-----+ 1 row in set (0.00 sec) </pre>
<pre> select concat(title, ' - ', released_year) as 'summary' from books order by released_year desc; </pre>	<pre> +-----+   summary   +-----+   Lincoln In The Bardo - 2017     Norse Mythology - 2016     10% Happier - 2014     The Circle - 2013     A Hologram for the King: A Novel - 2012     Just Kids - 2010     Consider the Lobster - 2005     Oblivion: Stories - 2004     The Namesake - 2003     Coraline - 2003     American Gods - 2001     A Heartbreaking Work of Staggering Genius - 2001     fake_book - 2001     The Amazing Adventures of Kavalier &amp; Clay - 2000     Interpreter of Maladies - 1996     Where I'm Calling From: Selected Stories - 1989     White Noise - 1985     What We Talk About When We Talk About Love: Stories - 1981     Cannery Row - 1945   +-----+ 19 rows in set (0.00 sec) </pre>
<pre> select title, author_lname from books where author_lname like '% %'; # author_lname having space. </pre>	<pre> +-----+-----+   title   author_lname   +-----+-----+   Oblivion: Stories   Foster Wallace     Consider the Lobster   Foster Wallace   +-----+-----+ 2 rows in set (0.00 sec) </pre>
<pre> select title, released_year, stock_quantity from books order by stock_quantity limit 3; </pre>	<pre> +-----+-----+-----+   title   released_year   stock_quantity   +-----+-----+-----+   Where I'm Calling From: Selected Stories   1989   12     American Gods   2001   12     What We Talk About When We Talk About Love: Stories   1981   23   +-----+-----+-----+ 3 rows in set (0.00 sec) </pre>
<pre> select title, author_lname from books order by author_lname, title; </pre>	<pre> +-----+-----+   title   author_lname   +-----+-----+   What We Talk About When We Talk About Love: Stories   Carver     Where I'm Calling From: Selected Stories   Carver     The Amazing Adventures of Kavalier &amp; Clay   Chabon     White Noise   DeLillo     A Heartbreaking Work of Staggering Genius   Eggers     A Hologram for the King: A Novel   Eggers     The Circle   Eggers     Consider the Lobster   Foster Wallace     Oblivion: Stories   Foster Wallace     American Gods   Gaiman     Coraline   Gaiman     Norse Mythology   Gaiman   +-----+-----+ </pre>

	<pre>   10% Happier   fake_book   Interpreter of Maladies   The Namesake   Lincoln In The Bardo   Just Kids   Cannery Row +-----+   Harris   Harris   Lahiri   Lahiri   Saunders   Smith   Steinbeck +-----+ </pre>
<pre> select concat(   'MY FAVORITE AUTHOR IS ',   author_fname, ' ', author_lname, '!' ) as 'yell' from books; </pre>	<pre> +-----+   yell +-----+   MY FAVORITE AUTHOR IS Jhumpa Lahiri!   MY FAVORITE AUTHOR IS Neil Gaiman!   MY FAVORITE AUTHOR IS Neil Gaiman!   MY FAVORITE AUTHOR IS Jhumpa Lahiri!   MY FAVORITE AUTHOR IS Dave Eggers!   MY FAVORITE AUTHOR IS Dave Eggers!   MY FAVORITE AUTHOR IS Michael Chabon!   MY FAVORITE AUTHOR IS Patti Smith!   MY FAVORITE AUTHOR IS Dave Eggers!   MY FAVORITE AUTHOR IS Neil Gaiman!   MY FAVORITE AUTHOR IS Raymond Carver!   MY FAVORITE AUTHOR IS Raymond Carver!   MY FAVORITE AUTHOR IS Don DeLillo!   MY FAVORITE AUTHOR IS John Steinbeck!   MY FAVORITE AUTHOR IS David Foster Wallace!   MY FAVORITE AUTHOR IS David Foster Wallace!   MY FAVORITE AUTHOR IS Dan Harris!   MY FAVORITE AUTHOR IS Freida Harris!   MY FAVORITE AUTHOR IS George Saunders! +-----+ </pre>
<pre> select concat(   'MY FAVORITE AUTHOR IS ',   upper(author_fname), ' ', upper(author_lname), '!' ) as 'yell' from books order by author_lname; #get all sorted author full #Names with the string 'MY FAVORITE AUTHOR IS'. </pre>	<pre> +-----+   yell +-----+   MY FAVORITE AUTHOR IS RAYMOND CARVER!   MY FAVORITE AUTHOR IS RAYMOND CARVER!   MY FAVORITE AUTHOR IS MICHAEL CHABON!   MY FAVORITE AUTHOR IS DON DELILLO!   MY FAVORITE AUTHOR IS DAVE EGGERS!   MY FAVORITE AUTHOR IS DAVE EGGERS!   MY FAVORITE AUTHOR IS DAVE EGGERS!   MY FAVORITE AUTHOR IS DAVID FOSTER WALLACE!   MY FAVORITE AUTHOR IS DAVID FOSTER WALLACE!   MY FAVORITE AUTHOR IS NEIL GAIMAN!   MY FAVORITE AUTHOR IS NEIL GAIMAN!   MY FAVORITE AUTHOR IS NEIL GAIMAN!   MY FAVORITE AUTHOR IS DAN HARRIS!   MY FAVORITE AUTHOR IS FREIDA HARRIS!   MY FAVORITE AUTHOR IS JHUMPA LAHIRI!   MY FAVORITE AUTHOR IS JHUMPA LAHIRI!   MY FAVORITE AUTHOR IS GEORGE SAUNDERS!   MY FAVORITE AUTHOR IS PATTI SMITH!   MY FAVORITE AUTHOR IS JOHN STEINBECK! +-----+ </pre>
<b>Aggregate functions:</b>  <pre> select count(*) from books; #count tells the number of rows from the #selected data </pre>	<pre> +-----+   count(*)   +-----+        19   +-----+ 1 row in set (0.00 sec) </pre>
<pre> select count(author_fname) from books; </pre>	<pre> +-----+   count(author_fname)   +-----+        19   +-----+ 1 row in set (0.00 sec) </pre>
<pre> select count(distinct author_fname) from books; #gives count of unique rows. </pre>	<pre> +-----+   count(distinct(author_fname))   +-----+        12   +-----+ 1 row in set (0.01 sec) </pre>
<pre> select count(distinct author_fname, author_lname) from books; # gives unique count of author first and last name combined. </pre>	<pre> +-----+   count(distinct author_fname, author_lname)   +-----+        12   +-----+ 1 row in set (0.00 sec) </pre>
<pre> select count(*) from books where title like '%the%'; #Gives count of title having the in it. </pre>	<pre> +-----+   count(*)   +-----+        6   +-----+ 1 row in set (0.00 sec) </pre>
<b>Group by clause</b>  Group by applies with some aggregate function. Aggregate/summarize identical data into single rows.	
<pre> select author_lname, count(*) from books group by author_lname; </pre>	<pre> +-----+   author_lname   count(*)   +-----+ </pre>

	<pre>   Lahiri        2     Gaiman        3     Eggers        3     Chabon        1     Smith         1     Carver        2     DeLillo       1     Steinbeck     1     Foster Wallace   2     Harris        2     Saunders      1   +-----+ 11 rows in set (0.00 sec) </pre>
<pre> select author_fname, author_lname, count(*) from books group by author_lname; #here count refers to the grouped columns after the group by operation. </pre>	<pre> +-----+-----+-----+   author_fname   author_lname   count(*)   +-----+-----+-----+   Jhumpa        Lahiri        2            Neil          Gaiman        3            Dave          Eggers        3            Michael       Chabon        1            Patti         Smith         1            Raymond       Carver        2            Don           DeLillo       1            John          Steinbeck     1            David         Foster Wallace   2            Dan           Harris        2            George        Saunders      1          +-----+-----+-----+ 11 rows in set (0.00 sec) </pre>
<pre> select released_year, count(*) from books group by released_year; </pre>	<pre> +-----+-----+   released_year   count(*)   +-----+-----+   2003            2            2016            1            2001            3            1996            1            2012            1            2013            1            2000            1            2010            1            1981            1            1989            1            1985            1            1945            1            2004            1            2005            1            2014            1            2017            1          +-----+-----+ 16 rows in set (0.00 sec) </pre>
<pre> select concat('In ', released_year, ' ', count(*), ' Books released.') as year from books group by released_year; </pre>	<pre> +-----+   year   +-----+   In 2003 2 Books released.     In 2016 1 Books released.     In 2001 3 Books released.     In 1996 1 Books released.     In 2012 1 Books released.     In 2013 1 Books released.     In 2000 1 Books released.     In 2010 1 Books released.     In 1981 1 Books released.     In 1989 1 Books released.     In 1985 1 Books released.     In 1945 1 Books released.     In 2004 1 Books released.     In 2005 1 Books released.     In 2014 1 Books released.     In 2017 1 Books released.   +-----+ 16 rows in set (0.00 sec) </pre>
<p><b>Min and Max functions</b></p> <pre> select title, max(pages) from books; </pre>	<pre> +-----+-----+   title   max(pages)   +-----+-----+   The Namesake   634   +-----+-----+ 1 row in set (0.00 sec) </pre>
<pre> select min(pages) from books; </pre>	<pre> +-----+   min(pages)   +-----+   176   +-----+ 1 row in set (0.00 sec) </pre>
<p><b>Sub queries</b></p> <p>Sometime the when performing min, max or some other operations two queries are independent and the result of the resultant is not what we are expecting. In this case we use sub queries.</p> <pre> select title, max(pages) from books; # the problem is like this the name should be The 'Amazing Adventures of Kavalier &amp; Clay' because they have the most pages but we are getting 'The Namesake'. </pre>	<pre> +-----+-----+   title   max(pages)   +-----+-----+   The Namesake   634   +-----+-----+ 1 row in set (0.00 sec) </pre>
<pre> select title, </pre>	<pre> +-----+-----+   title   +-----+ </pre>

<pre> pages from books where pages =(     select max(pages)     from books ); </pre>	<pre>   title   pages   +-----+-----+   The Amazing Adventures of Kavalier &amp; Clay   634   +-----+-----+ 1 row in set (0.00 sec) </pre>
<pre> select title, pages from books where pages =(     select min(pages)     from books ); </pre> <p><b>OR</b></p> <p>The above query takes too much time below is efficient.</p> <pre> select title, pages from books order by pages asc limit 1; </pre>	<pre> +-----+-----+   title   pages   +-----+-----+   What We Talk About When We Talk About Love: Stories   176   +-----+-----+ 1 row in set (0.00 sec) </pre> <p><b>OR</b></p> <pre> +-----+-----+   title   pages   +-----+-----+   What We Talk About When We Talk About Love: Stories   176   +-----+-----+ 1 row in set (0.01 sec) </pre>
<p><b>Min/Max with group by</b></p> <p>Find the year each author published their first book.</p> <pre> select author_fname, author_lname, min(released_year) from books group by author_fname, author_lname; </pre>	<pre> +-----+-----+-----+   Jhumpa   Lahiri   1996     Neil   Gaiman   2001     Dave   Eggers   2001     Michael   Chabon   2000     Patti   Smith   2010     Raymond   Carver   1981     Don   DeLillo   1985     John   Steinbeck   1945     David   Foster Wallace   2004     Dan   Harris   2014     Freida   Harris   2001     George   Saunders   2017   +-----+-----+-----+ 12 rows in set (0.00 sec) </pre>
<p>Find the longest page count for each author.</p> <pre> select author_fname, author_lname, max(pages) from books group by author_fname, author_lname; </pre> <p><b>OR sorted pages</b></p> <pre> select author_fname, author_lname, max(pages) from books group by author_fname, author_lname order by max(pages) desc; </pre>	<pre> +-----+-----+-----+   Jhumpa   Lahiri   291     Neil   Gaiman   465     Dave   Eggers   504     Michael   Chabon   634     Patti   Smith   304     Raymond   Carver   526     Don   DeLillo   320     John   Steinbeck   181     David   Foster Wallace   343     Dan   Harris   256     Freida   Harris   428     George   Saunders   367   +-----+-----+-----+ 12 rows in set (0.00 sec) </pre> <pre> +-----+-----+-----+   author_fname   author_lname   max(pages)   +-----+-----+-----+   Michael   Chabon   634     Raymond   Carver   526     Dave   Eggers   504     Neil   Gaiman   465     Freida   Harris   428     George   Saunders   367     David   Foster Wallace   343     Don   DeLillo   320     Patti   Smith   304     Jhumpa   Lahiri   291     Dan   Harris   256     John   Steinbeck   181   +-----+-----+-----+ 12 rows in set (0.00 sec) </pre>
<pre> select concat(author_fname,' ', author_lname) as 'author', max(pages) as 'longest book' from books group by author_fname, author_lname; </pre>	<pre> +-----+-----+-----+   Jhumpa Lahiri   291     Neil Gaiman   465     Dave Eggers   504     Michael Chabon   634     Patti Smith   304     Raymond Carver   526     Don DeLillo   320     John Steinbeck   181     David Foster Wallace   343     Dan Harris   256     Freida Harris   428     George Saunders   367   +-----+-----+-----+ 12 rows in set (0.00 sec) </pre>
<p><b>Sum function</b></p> <pre> select concat(author_fname, ' ', author_lname), sum(pages) </pre>	<pre> +-----+-----+-----+   Jhumpa Lahiri   489     Neil Gaiman   977     Dave Eggers   1293     Michael Chabon   634   +-----+-----+-----+ </pre>

```

from books
group by author_fname,
        author_lname;
# each author total book page count.

```

Patti Smith	304
Raymond Carver	702
Don DeLillo	320
John Steinbeck	181
David Foster Wallace	672
Dan Harris	256
Freida Harris	428
George Saunders	367

12 rows in set (0.00 sec)

#### Average function

```

Calculate the average released_year across all books
select released_year, avg(stock_quantity)
from books
group by released_year;

```

released_year	avg(stock_quantity)
2003	66.0000
2016	43.0000
2001	134.3333
1996	97.0000
2012	154.0000
2013	26.0000
2000	68.0000
2010	55.0000
1981	23.0000
1989	12.0000
1985	49.0000
1945	95.0000
2004	172.0000
2005	92.0000
2014	29.0000
2017	1000.0000

16 rows in set (0.00 sec)

```

select author_fname,
        author_lname,
        avg(pages) as 'Average pages'
from books
group by author_fname,
        author_lname;

```

Jhumpa	Lahiri	244.5000
Neil	Gaiman	325.6667
Dave	Eggers	431.0000
Michael	Chabon	634.0000
Patti	Smith	304.0000
Raymond	Carver	351.0000
Don	DeLillo	320.0000
John	Steinbeck	181.0000
David	Foster Wallace	336.0000
Dan	Harris	256.0000
Freida	Harris	428.0000
George	Saunders	367.0000

12 rows in set (0.00 sec)

#### Some practice Aggregate functions problems:

```

select count(*) from books; #print number of books in database.

```

count(*)
19

1 row in set (0.01 sec)

```

select released_year, count(*) from books group by
released_year;

```

released_year	count(*)
2003	2
2016	1
2001	3
1996	1
2012	1
2013	1
2000	1
2010	1
1981	1
1989	1
1985	1
1945	1
2004	1
2005	1
2014	1
2017	1

16 rows in set (0.00 sec)

```

select sum(stock_quantity) from books;

```

sum(stock_quantity)
2450

1 row in set (0.00 sec)

```

select concat(author_fname, ' ', author_lname) as 'authors',
        avg(released_year)
from books
group by author_fname,
        author_lname;

```

authors	avg(released_year)
Jhumpa Lahiri	1999.5000
Neil Gaiman	2006.6667
Dave Eggers	2008.6667
Michael Chabon	2000.0000
Patti Smith	2010.0000
Raymond Carver	1985.0000
Don DeLillo	1985.0000
John Steinbeck	1945.0000
David Foster Wallace	2004.5000
Dan Harris	2014.0000
Freida Harris	2001.0000
George Saunders	2017.0000

12 rows in set (0.00 sec)

```
select concat(author_fname, ' ', author_lname) as 'authors',
       max(pages)
from books
group by author_fname,
       author_lname;
```

```
+-----+-----+
| authors | max(pages) |
+-----+-----+
| Jhumpa Lahiri | 291 |
| Neil Gaiman | 465 |
| Dave Eggers | 504 |
| Michael Chabon | 634 |
| Patti Smith | 304 |
| Raymond Carver | 526 |
| Don DeLillo | 320 |
| John Steinbeck | 181 |
| David Foster Wallace | 343 |
| Dan Harris | 256 |
| Freida Harris | 428 |
| George Saunders | 367 |
+-----+-----+
12 rows in set (0.00 sec)
```

```
select concat(author_fname, ' ', author_lname) as 'authors',
       max(pages)
from books
group by author_fname,
       author_lname
order by max(pages) desc;
```

```
+-----+-----+
| authors | max(pages) |
+-----+-----+
| Michael Chabon | 634 |
| Raymond Carver | 526 |
| Dave Eggers | 504 |
| Neil Gaiman | 465 |
| Freida Harris | 428 |
| George Saunders | 367 |
| David Foster Wallace | 343 |
| Don DeLillo | 320 |
| Patti Smith | 304 |
| Jhumpa Lahiri | 291 |
| Dan Harris | 256 |
| John Steinbeck | 181 |
+-----+-----+
12 rows in set (0.00 sec)
```

```
select released_year as 'year',
       count(*) as '# books',
       avg(pages) as 'avg pages'
from books
group by released_year;
```

```
+-----+-----+-----+
| year | # books | avg pages |
+-----+-----+-----+
| 2003 | 2 | 249.5000 |
| 2016 | 1 | 304.0000 |
| 2001 | 3 | 443.3333 |
| 1996 | 1 | 198.0000 |
| 2012 | 1 | 352.0000 |
| 2013 | 1 | 504.0000 |
| 2000 | 1 | 634.0000 |
| 2010 | 1 | 304.0000 |
| 1981 | 1 | 176.0000 |
| 1989 | 1 | 526.0000 |
| 1985 | 1 | 320.0000 |
| 1945 | 1 | 181.0000 |
| 2004 | 1 | 329.0000 |
| 2005 | 1 | 343.0000 |
| 2014 | 1 | 256.0000 |
| 2017 | 1 | 367.0000 |
+-----+-----+-----+
16 rows in set (0.00 sec)
```

## Data Types

- Char(4)
- Varchar(variable bytes)
- Decimal(total number of digits, digits after decimal) - fixed point. Calculations are exact.
- Float(4 bytes) - floating point calculations are approximate: Takes more storage.
- Double(8 bytes): For more storage but it's not accurate.

## Dates and time

- Date: 'YYYY-MM-DD'
- Time: 'HH:MM:SS'
- Datetime: 'YYYY-MM-DD HH:MM:SS'
- timestamp

```
curdate()-gives current data.
curtime()-gives current time.
now()-gives current datetime.
```

```
select curdate();
```

```
+-----+
| curdate() |
+-----+
| 2021-11-21 |
+-----+
1 row in set (0.00 sec)
```

```
select now();
```

```
+-----+
| now() |
+-----+
| 2021-11-21 22:58:45 |
+-----+
1 row in set (0.00 sec)
```



select curtime();	<pre> +-----+   curtime()   +-----+   22:59:10   +-----+ 1 row in set (0.00 sec) </pre>
SELECT name, day(birthdate) FROM people;	<pre> +-----+-----+   name   day(birthdate)   +-----+-----+   Padma   11     Larry   25     Toaster   21   +-----+-----+ 3 rows in set (0.00 sec) </pre>
<b>Formatting Dates</b> SELECT name, birthdate, day(birthdate) FROM people; #gives the day of that date.	<pre> +-----+-----+-----+   name   birthdate   day(birthdate)   +-----+-----+-----+   Padma   1983-11-11   11     Larry   1943-12-25   25     Toaster   2021-11-21   21   +-----+-----+-----+ 3 rows in set (0.00 sec) </pre>
SELECT name, birthdate, dayname(birthdate) FROM people; # gives the dayname of the date.	<pre> +-----+-----+-----+   name   birthdate   dayname(birthdate)   +-----+-----+-----+   Padma   1983-11-11   Friday     Larry   1943-12-25   Saturday     Toaster   2021-11-21   Sunday   +-----+-----+-----+ 3 rows in set (0.01 sec) </pre>
SELECT name, birthdate, dayofweek(birthdate) FROM people;	<pre> +-----+-----+-----+   name   birthdate   dayofweek(birthdate)   +-----+-----+-----+   Padma   1983-11-11   6     Larry   1943-12-25   7     Toaster   2021-11-21   1   +-----+-----+-----+ 3 rows in set (0.00 sec) </pre>
SELECT name, birthdate, dayofyear(birthdate) FROM people;	<pre> +-----+-----+-----+   name   birthdate   dayofyear(birthdate)   +-----+-----+-----+   Padma   1983-11-11   315     Larry   1943-12-25   359     Toaster   2021-11-21   325   +-----+-----+-----+ 3 rows in set (0.01 sec) </pre>
SELECT name, birthtime, dayofyear(birthtime) FROM people;	<pre> +-----+-----+-----+   name   birthtime   dayofyear(birthtime)   +-----+-----+-----+   Padma   10:07:35   325     Larry   04:10:42   325     Toaster   23:01:34   325   +-----+-----+-----+ 3 rows in set (0.00 sec) </pre>
SELECT name, birthdt, dayofyear(birthdt) FROM people;	<pre> +-----+-----+-----+   name   birthdt   dayofyear(birthdt)   +-----+-----+-----+   Padma   1983-11-11 10:07:35   315     Larry   1943-12-25 04:10:42   359     Toaster   2021-11-21 23:01:34   325   +-----+-----+-----+ 3 rows in set (0.00 sec) </pre>
SELECT name, birthdt, month(birthdt) FROM people;	<pre> +-----+-----+-----+   name   birthdt   month(birthdt)   +-----+-----+-----+   Padma   1983-11-11 10:07:35   11     Larry   1943-12-25 04:10:42   12     Toaster   2021-11-21 23:01:34   11   +-----+-----+-----+ 3 rows in set (0.01 sec) </pre>
SELECT name, birthdt, monthname(birthdt) FROM people;	<pre> +-----+-----+-----+   name   birthdt   monthname(birthdt)   +-----+-----+-----+   Padma   1983-11-11 10:07:35   November     Larry   1943-12-25 04:10:42   December     Toaster   2021-11-21 23:01:34   November   +-----+-----+-----+ 3 rows in set (0.01 sec) </pre>
SELECT name, birthdt, hour(birthdt) FROM people;	<pre> +-----+-----+-----+   name   birthdt   hour(birthdt)   +-----+-----+-----+   Padma   1983-11-11 10:07:35   10     Larry   1943-12-25 04:10:42   4     Toaster   2021-11-21 23:01:34   23   +-----+-----+-----+ 3 rows in set (0.00 sec) </pre>
SELECT name, birthdt, minute(birthdt) FROM people;	<pre> +-----+-----+-----+   name   birthdt   minute(birthdt)   +-----+-----+-----+   Padma   1983-11-11 10:07:35   7     Larry   1943-12-25 04:10:42   10     Toaster   2021-11-21 23:01:34   1   +-----+-----+-----+ 3 rows in set (0.00 sec) </pre>

```
SELECT DATE_FORMAT(birthdt, 'Was born on a %W') FROM people;
#format dates specified in the function. See documentation
table for #more info.
```

```
+-----+
| DATE_FORMAT(birthdt, 'Was born on a %W') |
+-----+
| Was born on a Friday                      |
| Was born on a Saturday                   |
| Was born on a Sunday                     |
+-----+
3 rows in set (0.01 sec)
```

```
SELECT DATE_FORMAT(birthdt, '%m/%d/%Y') FROM people;
```

```
+-----+
| DATE_FORMAT(birthdt, '%m/%d/%Y') |
+-----+
| 11/11/1983                         |
| 12/25/1943                         |
| 11/21/2021                         |
+-----+
3 rows in set (0.00 sec)
```

```
SELECT DATE_FORMAT(birthdt, '%m/%d/%Y at %h:%i') FROM people;
```

```
+-----+
| DATE_FORMAT(birthdt, '%m/%d/%Y at %h:%i') |
+-----+
| 11/11/1983 at 10:07                   |
| 12/25/1943 at 04:10                   |
| 11/21/2021 at 11:01                   |
+-----+
3 rows in set (0.00 sec)
```

#### Date math

Adding days, months, minutes, etc to dates.

```
select name, birthdate, datediff(now(), birthdate) from people;
```

```
+-----+-----+-----+
| name   | birthdate | datediff(now(), birthdate) |
+-----+-----+-----+
| Padma  | 1983-11-11 | 13890                      |
| Larry  | 1943-12-25 | 28456                      |
| Toaster | 2021-11-21 | 0                          |
+-----+-----+-----+
3 rows in set (0.01 sec)
```

```
select birthdt, date_add(birthdt, interval 1 month) from
people;
```

```
+-----+-----+
| birthdt | date_add(birthdt, interval 1 month) |
+-----+-----+
| 1983-11-11 10:07:35 | 1983-12-11 10:07:35 |
| 1943-12-25 04:10:42 | 1944-01-25 04:10:42 |
| 2021-11-21 23:01:34 | 2021-12-21 23:01:34 |
+-----+-----+
3 rows in set (0.00 sec)
```

```
select birthdt, birthdt + interval 1 month from people;
```

```
+-----+-----+
| birthdt | birthdt + interval 1 month |
+-----+-----+
| 1983-11-11 10:07:35 | 1983-12-11 10:07:35 |
| 1943-12-25 04:10:42 | 1944-01-25 04:10:42 |
| 2021-11-21 23:01:34 | 2021-12-21 23:01:34 |
+-----+-----+
3 rows in set (0.00 sec)
```

```
select birthdt, birthdt + interval 1 day from people;
```

```
+-----+-----+
| birthdt | birthdt + interval 1 day |
+-----+-----+
| 1983-11-11 10:07:35 | 1983-11-12 10:07:35 |
| 1943-12-25 04:10:42 | 1943-12-26 04:10:42 |
| 2021-11-21 23:01:34 | 2021-11-22 23:01:34 |
+-----+-----+
3 rows in set (0.00 sec)
```

```
select birthdt, birthdt - interval 1 day from people;
```

```
+-----+-----+
| birthdt | birthdt - interval 1 day |
+-----+-----+
| 1983-11-11 10:07:35 | 1983-11-10 10:07:35 |
| 1943-12-25 04:10:42 | 1943-12-24 04:10:42 |
| 2021-11-21 23:01:34 | 2021-11-20 23:01:34 |
+-----+-----+
3 rows in set (0.00 sec)
```

```
select birthdt, birthdt + interval 1 month + interval 1 day
from people;
```

```
+-----+-----+
| birthdt | birthdt + interval 1 month + interval 1 day |
+-----+-----+
| 1983-11-11 10:07:35 | 1983-12-12 10:07:35 |
| 1943-12-25 04:10:42 | 1944-01-26 04:10:42 |
| 2021-11-21 23:01:34 | 2021-12-22 23:01:34 |
+-----+-----+
3 rows in set (0.00 sec)
```

#### Creating a table that stores timestamp for updated content

```
create table comments2 (
  content varchar(100),
  created_at timestamp default now() on update now()
);
# here on update will change created_at value to now every
time created_at is changed.
#To update date whenever the content field is updated we used
this logic.
```

```
insert into comments2 (content)
values('asdassrv');
```

```
insert into comments2 (content)
values('jrtsbvf');
```

```
select * from comments2;
```

```
+-----+-----+
| content | created_at |
+-----+-----+
```

	<pre>   jrtsbvf   2021-11-22 00:33:13     asdassrv   2021-11-22 00:33:23   +-----+-----+ 2 rows in set (0.00 sec) </pre>
<pre> update comments2 set content = 'This is totally great !!!' where content = 'jrtsbvf';  update comments2 set content = 'Enjoy this sweet moment!' where content = 'asdassrv'; </pre>	
<pre>select * from comments2;</pre>	<pre> +-----+-----+   content   created_at   +-----+-----+   This is totally great !!!   2021-11-22 00:36:02     Enjoy this sweet moment!   2021-11-22 00:37:19     I like bannana pie!   2021-11-22 00:34:28   +-----+-----+ 3 rows in set (0.00 sec) </pre>
<pre>select * from comments2 order by created_at desc;</pre>	<pre> +-----+-----+   content   created_at   +-----+-----+   Enjoy this sweet moment!   2021-11-22 00:37:19     This is totally great !!!   2021-11-22 00:36:02     I like bannana pie!   2021-11-22 00:34:28   +-----+-----+ 3 rows in set (0.00 sec) </pre>
<b>Logical Operators</b> <b>(and, or,!=,not like, greater than, less than, between, in and not in)</b>	
<pre>select title from books where released_year = 2017;</pre>	<pre> +-----+   title   +-----+   Lincoln In The Bardo   +-----+ 1 row in set (0.00 sec) </pre>
<pre>select title from books where released_year != 2017;</pre>	<pre> +-----+   title   +-----+   The Namesake     Norse Mythology     American Gods     Interpreter of Maladies     A Hologram for the King: A Novel     The Circle     The Amazing Adventures of Kavalier &amp; Clay     Just Kids     A Heartbreaking Work of Staggering Genius     Coraline     What We Talk About When We Talk About Love: Stories     Where I'm Calling From: Selected Stories     White Noise     Cannery Row     Oblivion: Stories     Consider the Lobster     10% Happier     fake_book   +-----+ 18 rows in set (0.00 sec) </pre>
<pre>select title from books where title like '%w%';</pre>	<pre> +-----+   title   +-----+   A Heartbreaking Work of Staggering Genius     What We Talk About When We Talk About Love: Stories     Where I'm Calling From: Selected Stories     White Noise     Cannery Row   +-----+ 5 rows in set (0.00 sec) </pre>
<pre>select title from books where title not like '%w%'; # not like opposite of 'like' clause.</pre>	<pre> +-----+   title   +-----+   The Namesake     Norse Mythology     American Gods     Interpreter of Maladies     A Hologram for the King: A Novel     The Circle     The Amazing Adventures of Kavalier &amp; Clay     Just Kids     Coraline     Oblivion: Stories     Consider the Lobster     10% Happier     fake_book     Lincoln In The Bardo   +-----+ </pre>



