

```
select distinct name, url  
from Users  
left join Views  
on id=user_id  
where url= "Orders"
```

MySQL Workbench

LearnQA x

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas

Filter objects

learnqa  
learnqa\_create\_alter\_drop  
learnqa\_homework  
Tables  
Views  
Stored Procedures  
Functions  
learnqa\_insert\_update\_delete

SQL File 1\* x

1 select distinct name, url  
2 from Users  
3 left join Views  
4 on id=user\_id  
5 where url= "Orders"  
6

Result Grid | Filter Rows: Export: Wrap Cell Content: Result 1 x

name	url
Kim	orders
Stone	orders
Rino	orders
Nicky	orders
Kelly	orders
Sam	orders

Result Grid Form Editor Read Only Context Help Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	13:36:04	use learnqa_homework	0 row(s) affected	0.016 sec
2	13:38:14	select distinct name, url from Users left join Views on id=user_id where url= "Orders" LIMIT 0, 1...	16 row(s) returned	0.031 sec / 0.000 sec

No object selected

Object Info Session

Query Completed

[  
{  
"name" : "Kim",  
"url" : "orders"  
,

```
},
{
  "name" : "Stone",
  "url" : "orders"
},
{
  "name" : "Rino",
  "url" : "orders"
},
{
  "name" : "Nicky",
  "url" : "orders"
},
{
  "name" : "Kelly",
  "url" : "orders"
},
{
  "name" : "Sam",
  "url" : "orders"
},
{
  "name" : "Bill",
  "url" : "orders"
},
{
  "name" : "Marsi",
  "url" : "orders"
},
{
  "name" : "Artur",
  "url" : "orders"
},
{
  "name" : "Eminem",
  "url" : "orders"
},
{
  "name" : "Ann",
  "url" : "orders"
},
{
  "name" : "Lore",
  "url" : "orders"
},
{
  "name" : "Nick",
  "url" : "orders"
},
{
  "name" : "Sit",
  "url" : "orders"
},
{
  "name" : "Stan",
  "url" : "orders"
},
{
  "name" : "Harry",
  "url" : "orders"
}
```

```
    un : orders  
}  
]
```

# The Most Important SQL Commands

## Work with database

SELECT - extracts data from a database

UPDATE - updates data in a database

DELETE - deletes data from a database

INSERT INTO - inserts new data into a database

CREATE DATABASE - creates a new database

ALTER DATABASE - modifies a database

## Work with Index

CREATE INDEX - creates an index (search key)

DROP INDEX - deletes an index

## Work with Table

CREATE TABLE - creates a new table

ALTER TABLE - modifies a table

DROP TABLE - deletes a table

## Select statement

The SELECT statement is used to select data from a database.

The data returned is stored in a result table, called the result-set.

SELECT \*

```
SELECT * FROM Orders
```

The screenshot shows the SSMS interface with the following details:

- Query Window:** The main window displays the query `SELECT * FROM Orders`. The results pane shows a grid of data with columns: id, price, discount, and user\_id. The data includes rows with values such as (1, 3398, 0.05, 42), (2, 4503, 0.5, 39), etc.
- Object Explorer:** On the left, the object explorer shows the schema `learnqa_homework` containing tables like `Orders`, `Users`, and `Views`.
- Status Bar:** The status bar at the bottom indicates "Query Completed".

## SELECT Columns

```
select id,name,age from Cats
```

The following SQL statement selects the columns from the table:

The screenshot shows the SSMS interface with the following details:

- Query Window:** The main window displays the query `select id,name,age from Cats`. The results pane shows a grid of data with columns: id, name, and age. The data includes rows with values such as (1, Vlas, 1), (2, Nemo, 10), etc.
- Object Explorer:** On the left, the object explorer shows the schema `learnqa` containing tables like `Breeds` and `Cats`. The `Cats` table is expanded to show its columns: id, name, birthday, age, gender, price, discount, breed\_id, and shop\_id.

Information :> **Cats 4** X Apply

Table: Breeds

Columns:

	<u><a href="#">id</a></u>	int(11) AI
		PK
	<u><a href="#">name</a></u>	text
	<u><a href="#">character</a></u>	text

Action Output

#	Time	Action	Message
5	16:36:00	select* from Breeds LIMIT 0, 1000	13 row(s) returned
6	17:05:57	select id, name, character from Breeds	Error Code: 1064. You have an error
7	17:06:53	use leamqa	0 row(s) affected
8	17:06:56	use leamqa	0 row(s) affected

Object Info Session

**SELECT DISTINCT** statement is used to return only distinct (different) values

```
select distinct id from Hamsters
```

The screenshot shows the MySQL Workbench interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. Below the menu is a toolbar with various icons for database management tasks. The left sidebar is titled 'Navigator' and contains sections for SCHEMAS, TABLES, VIEWS, and INFORMATION. Under SCHEMAS, the 'learnqa' schema is selected, showing tables like Breeds, Cats, CustomerLog, Hamsters, and others. The 'CustomerLog' table is currently selected. The main workspace has a tab titled 'SQL File 1' containing the query: 'select distinct id from Hamsters'. Below the query is a results grid titled 'Hamsters 6' showing the following data:

id
34
35
36
37
38
39

At the bottom right of the results grid, there are buttons for 'Apply' and 'Revert'. To the right of the results grid is an 'Output' pane titled 'Action Output' which lists three log entries:

#	Time	Action	Message
12	17:26:06	select distinct result, from CustomerLog	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'result, from CustomerLog' at line 1
13	17:26:40	select distinct id, from Hamsters	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'id, from Hamsters' at line 1
14	17:31:19	select distinct id from Hamsters LIMIT 0, 1000	8 row(s) returned

The bottom left of the screen shows tabs for 'Object Info' and 'Session'. The status bar at the bottom indicates 'Query Completed'.

# WHERE Clause

The WHERE clause is used to filter records.

It is used to extract only those records that fulfill a specified condition.

```
select* from Orders  
where user_id= 42
```

Equal

The screenshot shows the MySQL Workbench interface with the 'LearnQA' database selected. In the 'Navigator' pane, under the 'Tables' section of the 'learnqa\_homework' schema, the 'Orders' table is listed. The 'Result Grid' pane displays the results of the query `select* from Orders where user_id= 42`. The results show 18 rows with columns: id, price, discount, and user\_id. The 'user\_id' column consistently shows the value 42. The 'Output' pane at the bottom shows the execution log with three entries, all indicating successful execution of the query.

id	price	discount	user_id
1	3398	0.05	42
3	1382	0.5	42
6	1774	0.15	42
9	1004	0.5	42
89	3289	0.15	42
100	4492	0.05	42
127	2760	0.15	42

Action Output:

- # Time Action Message  
29 18:33:47 select\* from Orders where discount=0.05 LIMIT 0, 1000 0 row(s) returned
- 30 18:35:02 select\* from Orders where discount=0.05 LIMIT 0, 1000 0 row(s) returned
- 31 18:36:48 select\* from Orders where user\_id= 42 LIMIT 0, 1000 452 row(s) returned

```
select* from Users  
where name = "Stan"
```

The screenshot shows the MySQL Workbench interface with the 'LearnQA' database selected. In the 'Navigator' pane, under the 'Tables' section of the 'learnqa\_homework' schema, the 'Users' table is listed. The 'Result Grid' pane displays the results of the query `select* from Users where name = "Stan"`. The results show 24 rows with columns: id, name, reg\_date, and last\_access. One row has 'name' set to 'Stan'. The 'Output' pane at the bottom shows the execution log with three entries, where the first two failed due to a syntax error ('Error Code: 1146. Table 'learnqa\_homework'.Na') and the third succeeded.

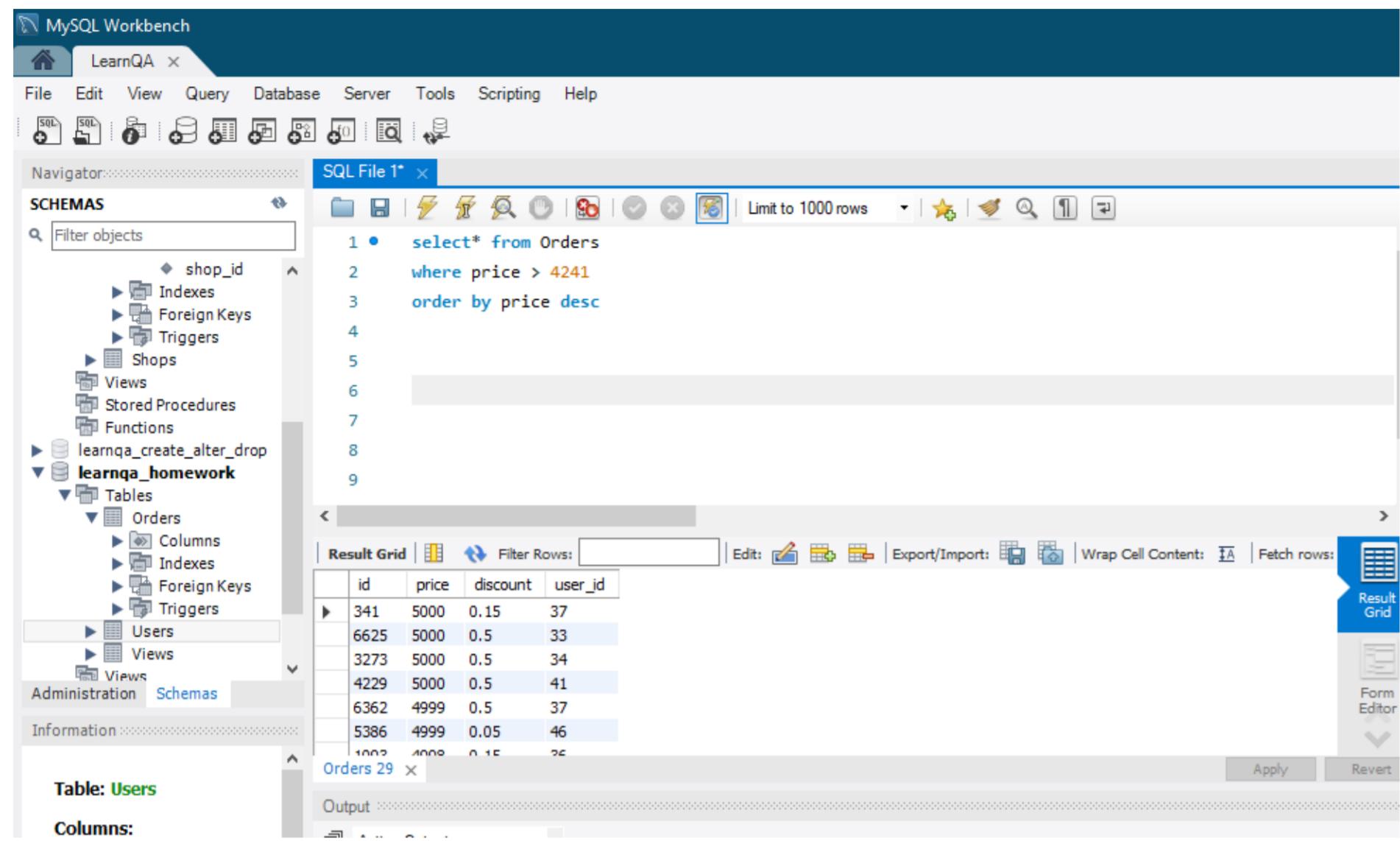
id	name	reg_date	last_access
45	Stan	2017-02-17 11:00:48	2018-01-26 04:12:01
*	NULL	NULL	NULL

Action Output:

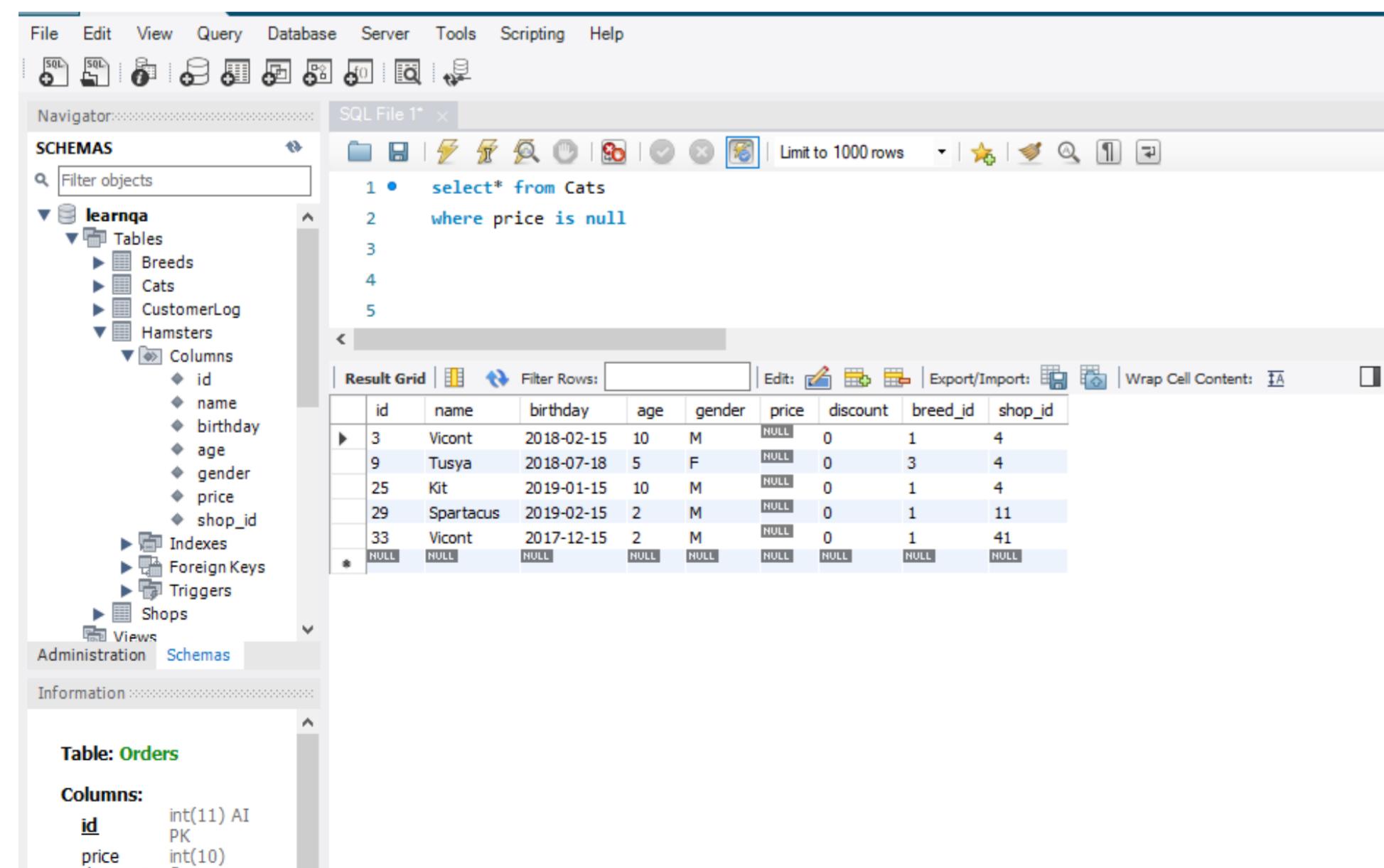
- # Time Action Message  
36 18:42:08 select\* from Name LIMIT 0, 1000 Error Code: 1146. Table 'learnqa\_homework'.Na
- 37 18:42:23 select\* from Users LIMIT 0, 1000 19 row(s) returned
- 38 18:42:43 select\* from Users where name = "Stan" LIMIT 0, 1000 1 row(s) returned

```
select* from Orders  
where price > 4241  
order by price desc
```

# Greater than



```
select* from Cats  
where price is null
```



```
select* from Cats  
where price is not null
```

The screenshot shows the MySQL Workbench interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. Below the menu is a toolbar with various icons for database management. The left sidebar, titled 'Navigator', shows the 'SCHEMAS' section with 'learnqa' selected, displaying tables like Breeds, Cats, CustomerLog, Hamsters, and their columns (id, name, birthday, age, gender, price, shop\_id). It also lists Indexes, Foreign Keys, Triggers, Shops, and Views. The 'Administration' tab is currently active. The main workspace contains a 'SQL File 1\*' tab with the following SQL query:

```
1 • select* from Cats  
2 where price is not null  
3  
4  
5
```

Below the query is a 'Result Grid' showing the results of the query. The grid has columns: id, name, birthday, age, gender, price, discount, breed\_id, and shop\_id. The data consists of 21 rows, each representing a cat. The row for 'Lapa' (id 6) is highlighted with a blue background.

	id	name	birthday	age	gender	price	discount	breed_id	shop_id
1	1	Vlas	2018-11-02	1	M	128	0	2	4
2	2	Nemo	2018-02-12	10	M	108	0.1	1	3
4	4	Zuza	2018-03-04	9	F	100	0.1	1	4
5	5	Mars	2018-05-18	7	M	144	0.15	4	1
6	6	Lapa	2018-06-06	6	F	100	0.05	3	3
7	7	Glaflira	2018-06-17	6	F	150	0.15	4	4
8	8	Tigr	2018-07-09	5	M	100	0.1	1	1
10	10	Musya	2018-08-05	4	F	125	0.15	4	3
11	11	Pusiceta	2018-08-10	4	F	137	0.15	4	3
12	12	Eroshka	2018-08-28	5	M	140	0.1	1	1
13	13	Lisa	2018-08-30	4	F	114	0.1	1	4
14	14	Tiran	2018-09-03	4	M	120	0.05	3	1
15	15	Vasiliy	2018-09-20	3	M	107	0.05	3	1
16	16	Lokki	2018-10-05	2	M	105	0.05	3	1
17	17	Kitty	2017-02-11	22	M	111	0	2	4
18	18	Sammy	2017-12-07	12	M	145	0	1	1
19	19	Showman	2017-12-02	12	M	128	0	3	2
20	20	Funny	2016-12-02	22	M	118	0	5	2
21	21	Liz	2015-08-02	55	M	213	0.5	5	2

The bottom right corner of the result grid has a 'Result Grid' button. The bottom status bar shows tabs for 'Object Info', 'Session', and 'Output'.

```
select* from Cats  
where birthday between '2018-02-12' and '2018-03-04'
```

Between a certain rang

The screenshot shows the MySQL Workbench interface with the following details:

- Top Bar:** MySQL Workbench, LearnQA x, File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Toolbar:** SQL, XML, Text, Diagram, Grid, Results, Log, Database.
- Navigator:** SCHEMAS, Filter objects, learnqa (selected), Tables, Breeds, Cats (selected), Columns (expanded), id, name, birthday, age, gender, price, discount, breed\_id, shop\_id, Indexes, Foreign Keys, Triggers, CustomerLog, Hamsters.
- SQL Editor:** SQL File 1\*, select\* from Cats where birthday between '2018-02-12' and '2018-03-04'.
- Result Grid:** Shows the results of the query, displaying columns: id, name, birthday, age, gender, price, discount, breed\_id, shop\_id. The results are:

	id	name	birthday	age	gender	price	discount	breed_id	shop_id
▶	2	Nemo	2018-02-12	10	M	108	0.1	1	3
▶	3	Vicont	2018-02-15	10	M	NULL	0	1	4
▶	4	Zuza	2018-03-04	9	F	100	0.1	1	4
▶	30	Killa	2018-03-04	9	F	110	0.2	1	10
▶	31	Inga	2018-03-02	1	F	128	0	2	44
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
- Bottom Status:** Column: birthday

## AND, OR and NOT Operators

The WHERE clause can be combined with AND, OR, and NOT operators.

The AND and OR operators are used to filter records based on more than one condition:

The AND operator displays a record if all the conditions separated by AND are TRUE.  
The OR operator displays a record if any of the conditions separated by OR is TRUE.  
The NOT operator displays a record if the condition(s) is NOT TRUE.

```
select* from Cats  
where age =10  
and discount<05
```

The screenshot shows the MySQL Workbench interface. In the SQL editor, the following query is entered:

```
1 • select* from Cats  
2 where age =10  
3 and discount<05  
4  
5  
6  
7  
8
```

The results grid displays the following data:

	id	name	birthday	age	gender	price	discount	breed_id	shop_id
▶	2	Nemo	2018-02-12	10	M	108	0.1	1	3
3	Vicont	2018-02-15	10	M	NULL	0	1	4	
23	Olle	2019-02-02	10	M	128	0	2	4	
24	Lilla	2019-02-12	10	M	108	0.1	1	3	
25	Kit	2019-01-15	10	M	NULL	0	1	4	
*	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL

```
select* from Cats  
where breed_id=3  
or breed_id= 4
```

The screenshot shows the MySQL Workbench interface. In the SQL editor, the following query is entered:

```
1 • select* from Cats  
2 where breed_id=3  
3 or breed_id= 4  
4  
5  
6  
7
```

8

Result Grid | Filter Rows: Edit: Export/Import: Wrap Cell Content:  Result Grid

	id	name	birthday	age	gender	price	discount	breed_id	shop_id
▶	5	Mars	2018-05-18	7	M	144	0.15	4	1
▶	6	Lapa	2018-06-06	6	F	100	0.05	3	3
▶	7	Glaflira	2018-06-17	6	F	150	0.15	4	4
▶	9	Tusya	2018-07-18	5	F	NULL	0	3	4
▶	10	Musya	2018-08-05	4	F	125	0.15	4	3
▶	11	Pusiceta	2018-08-10	4	F	137	0.15	4	3
▶	14	Tiran	2018-09-03	4	M	120	0.05	3	1
▶	15	Vasiliy	2018-09-20	3	M	107	0.05	3	1
▶	16	Lokki	2018-10-05	2	M	105	0.05	3	1
▶	19	Showman	2017-12-02	12	M	128	0	3	2

Cats 46

Output: Action Output # Time Action

Message

```
select* from Shops
where country in ('Russia','Italy')
```

MySQL Workbench LearnQA x

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas

SCHEMAS

learnqa Tables  Breeds  Cats  Columns  id  name  birthday  age  gender  price  discount  breed\_id  shop\_id  Indexes  Foreign Keys  Triggers  CustomerLog  Hamsters Administration Schemas

SQL File 1\* x

1 • select\* from Shops
2 where country in ('Russia','Italy')
3
4
5
6
7

Result Grid | Filter Rows: Edit: Export/Import: Wrap Cell Content:  Result Grid

	id	name	country	city	adds
▶	1	Four Paws	Russia	Moscow	VDNH
▶	2	Mr.Zoo	Russia	St.Petersburg	Vasilyevky Island
▶	5	Cats&Dogs	Italy	Rome	Square of the mouth of truth
▶	10	Cats&Dogs	Italy	Venice	Metropolitan City of Venice
▶	11	Cats&Dogs	Russia	Moscow	Red Square
▶	12	Four Paws	Russia	Moscow	Arbat st
▶	13	Four Paws	Russia	Novosibirsk	Square of Fame
*	HULL	HULL	HULL	HULL	HULL

Cats 48

Output: Action Output # Time Action

Object Info Session 65 20:12:43 select\* from Shops where country in (Russia,Italy) LIMIT 0, 1000 Message Error Code: 1054. Unknown column 'Russia' in 7 row(s) returned 66 20:13:28 select\* from Shops where country in ('Russia','Italy') LIMIT 0, 1000

Query Completed

```
select* from Shops
where not country='USA'
```

MySQL Workbench LearnQA x

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas

SCHEMAS

learnqa Tables  Breeds  Cats  Columns  id  name  birthday  age  gender  price  discount  breed\_id  shop\_id  Indexes  Foreign Keys  Triggers  CustomerLog  Hamsters Administration Schemas

SQL File 1\* x

1 • select\* from Shops
2 where not country='USA'

Schemas

- Tables
  - Breeds
  - Cats
    - Columns
      - id
      - name
      - birthday
      - age
      - gender
      - price
      - discount
      - breed\_id
      - shop\_id
    - Indexes
    - Foreign Keys
    - Triggers
    - CustomerLog
    - Hamsters
- Administration
- Schemas

Information

Column: birthday

Definition: birthday date

Object Info Session

Query Completed

Result Grid

	id	name	country	city	adds
1	Four Paws	Russia	Moscow	VDNH	
2	Mr.Zoo	Russia	St.Petersburg	Vasilyevky Island	
4	Cats&Dogs	UK	London	Soho	
5	Cats&Dogs	Italy	Rome	Square of the mouth of truth	
6	猫之犬	Japanese	Tokio	Rome	場所
8	Mr.Zoo	UK	Manchester	John Dalton St	
9	Murzila	Japanese	Kyoto	Shimogyo Ward	
10	Cats&Dogs	Italy	Venice	Metropolitan City of Venice	
11	Cats&Dogs	Russia	Moscow	Red Square	
12	Four Paws	Russia	Moscow	Arbat st	

Shops 50 x

Output

Action Output

#	Time	Action	Message
67	20:16:13	select* from Shops LIMIT 0, 1000	14 row(s) returned
68	20:16:58	select* from Shops where not country='USA' LIMIT 0, 1000	12 row(s) returned

```
select* from Shops
where country ='USA'
and city='New Yourk'
or country ='Japanese'
and city='Tokio'
```

MySQL Workbench

LearnQA x

File Edit View Query Database Server Tools Scripting Help

Navigator: SQL File 1\*

SCHEMAS

- learnqa
  - Tables
    - Breeds
    - Cats
      - Columns
        - id
        - name
        - birthday
        - age
        - gender
        - price
        - discount
        - breed\_id
        - shop\_id
      - Indexes
      - Foreign Keys
      - Triggers
      - CustomerLog
      - Hamsters

Information

Column: birthday

Definition: birthday date

Object Info Session

Query Completed

Result Grid

	id	name	country	city	adds
6	猫之犬	Japanese	Tokio	場所	
*	NULL	NULL	HULL	NULL	NULL

Shops 52 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
69	20:21:24	select* from Shops LIMIT 0, 1000	14 row(s) returned	0.031 sec / 0.000 sec
70	20:22:47	select* from Shops where country ='USA' and city='New Yourk' or country ='Japanese' and ci...	1 row(s) returned	0.031 sec / 0.000 sec

SQLAdditions:

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Context Help Snippets

Windows Taskbar: 22, S, ENG, 20:24, 25.11.2021, 2

# SQL ORDER BY Keyword

The ORDER BY keyword is used to sort the result-set in ascending or descending order.

The ORDER BY keyword sorts the records in ascending order by default. To sort the records in descending order, use the DESC keyword.

## ORDER BY Syntax

```
SELECT column1, column2, ...
FROM table_name
ORDER BY column1, column2, ...
ASC | DESC;
```

```
select* from Breeds
order by id
```

The screenshot shows the MySQL Workbench interface. In the top navigation bar, the 'Query' tab is selected. Below it, the 'SQL File 1\*' tab contains the following SQL code:

```
1 • select* from Breeds
2   order by id
```

In the center of the window, the 'Result Grid' displays the data from the 'Breeds' table, ordered by 'id'. The columns are 'id', 'name', and 'character'. The data rows are:

id	name	character
1	maine coon	friendly
2	sphynx	inquisitive
3	bengal	playful
4	british	loyalty
5	abyssinian	playful
6	american wirehair	loyalty
7	american curl	bad boy
8	balinese-javanese	funny
9	birman	brave
10	american shorthair	lazy
11	ocicat	inquisitive
12	persian	lazy

```
select* from Cats
```

The screenshot shows the MySQL Workbench interface. In the top navigation bar, the 'Query' tab is selected. Below it, the 'SQL File 1\*' tab contains the following SQL code:

```
1 • select* from Cats
2
```

In the center of the window, the 'Result Grid' displays the data from the 'Cats' table. The columns are 'id', 'name', and 'character'. The data rows are:

id	name	character
1	maine coon	friendly
2	sphynx	inquisitive
3	bengal	playful
4	british	loyalty
5	abyssinian	playful
6	american wirehair	loyalty
7	american curl	bad boy
8	balinese-javanese	funny
9	birman	brave
10	american shorthair	lazy
11	ocicat	inquisitive
12	persian	lazy

Tables Views Stored Procedures Functions learnqa\_insert\_update\_delete Administration Schemas Information Schema: learnqa

**Result Grid** Filter Rows: Edit: Export/Import: Wrap Cell Content: Result Grid Form Editor Field Types

	id	name	birthday	age	gender	price	discount	breed_id	shop_id
▶	1	Vlas	2018-11-02	1	M	128	0	2	4
	2	Nemo	2018-02-12	10	M	108	0.1	1	3
	3	Vicont	2018-02-15	10	M	NULL	0	1	4
	4	Zuza	2018-03-04	9	F	100	0.1	1	4
	5	Mars	2018-05-18	7	M	144	0.15	4	1
	6	Lapa	2018-06-06	6	F	100	0.05	3	3
	7	Glaflra	2018-06-17	6	F	150	0.15	4	4
	8	Tigr	2018-07-09	5	M	100	0.1	1	1
	9	Tusya	2018-07-18	5	F	NULL	0	3	4
	10	Musya	2018-08-05	4	F	125	0.15	4	3
	11	Pusiceta	2018-08-10	4	F	137	0.15	4	3
	12	Frndka	2018-08-28	5	M	140	0.1	1	1

```
select distinct age from Cats
order by age asc
```

MySQL Workbench LearnQA x

File Edit View Query Database Server Tools Scripting Help

Navigator SCHEMAS

learnqa Tables Breeds Cats CustomerLog Hamsters Shops Views Stored Procedures Functions

learnqa\_create\_alter\_drop Tables Views Stored Procedures Functions

learnqa\_insert\_update\_delete Administration Schemas Information Schema: learnqa

SQL File 1\*

1 • select distinct age from Cats
2 order by age asc

Result Grid Filter Rows: Export: Wrap Cell Content: Result Grid Form Editor Field Types

age
1
2
3
4
5
6
7
9
10
12
21
22

```
select distinct age from Cats
order by age desc
```

**ORDER BY DESC**

MySQL Workbench LearnQA x

File Edit View Query Database Server Tools Scripting Help

Navigator SCHEMAS

learnqa Tables Breeds Cats CustomerLog Hamsters Shops Views Stored Procedures Functions

learnqa\_create\_alter\_drop Tables Views Stored Procedures Functions

learnqa\_insert\_update\_delete Administration Schemas Information Schema: learnqa

SQL File 1\*

1 • select distinct age from Cats
2 order by age desc

Result Grid Filter Rows: Export: Wrap Cell Content: Result Grid Form Editor Field Types

age
60
55
22
21
12
10
9
7
6
5
4
3

```
select* from Cats  
where price is not NULL  
order by price asc, discount desc
```

## ORDER BY Several Columns

The screenshot shows the MySQL Workbench interface with a query editor containing the following SQL code:

```
1 • select* from Cats  
2 where price is not NULL  
3 order by price asc, discount desc
```

The results are displayed in a grid:

ID	Name	Birthday	Age	Gender	Price	Discount	Breed ID	Shop ID
4	Zuza	2018-03-04	9	F	100	0.1	1	4
8	Tigr	2018-07-09	5	M	100	0.1	1	1
26	Sam	2019-03-04	9	F	100	0.1	1	4
6	Lapa	2018-06-06	6	F	100	0.05	3	3
16	Lokki	2018-10-05	2	M	105	0.05	3	1
15	Vasiliy	2018-09-20	3	M	107	0.05	3	1
2	Nemo	2018-02-12	10	M	108	0.1	1	3
24	Lilla	2019-02-12	10	M	108	0.1	1	3
32	Nemo	2011-02-12	12	M	108	0.1	1	333
30	Killa	2018-03-04	9	F	110	0.2	1	10
17	Kitty	2017-02-11	22	M	111	0	2	4
13	Iisa	2018-08-30	4	F	114	0.1	1	4

```
select user_id,  
round(  
(price* (1-discount)),  
2  
)  
as real_price from Orders  
order by real_price desc  
limit 1
```

The screenshot shows the MySQL Workbench interface with a query editor containing the following SQL code:

```
1 • select user_id,  
2 round(  
3 (price* (1-discount)),  
4 2  
5 )  
6 as real_price from Orders  
7 order by real_price desc  
8 limit 1  
9
```

The results are displayed in a grid:

ID	Price	Discount	User ID	Real Price
1	100	0.1	1	90.00
2	100	0.1	1	90.00
3	100	0.1	1	90.00
4	105	0.05	3	100.75
5	107	0.05	3	101.35
6	108	0.1	1	97.20
7	108	0.1	1	97.20
8	110	0.2	1	88.00
9	111	0	2	111.00
10	114	0.1	1	102.60

Indexes  
Foreign Keys  
Triggers  
Users  
Views

Administration Schemas

Information

Column: price

Definition: price int(10)

Result Grid

Result 13 x Read Only

Action Output

#	Time	Action	Message
22	18:02:54	select user_id from Orders where price=38 LIMIT 0, 1000	0 row(s) returned
23	18:04:09	select*from Orders where price=38 LIMIT 0, 1000	0 row(s) returned

# SQL Logical Operators

## Operator Description Example

ALL TRUE if all of the subquery values meet the condition  
AND TRUE if all the conditions separated by AND is TRUE  
ANY TRUE if any of the subquery values meet the condition  
BETWEEN TRUE if the operand is within the range of comparisons  
EXISTS TRUE if the subquery returns one or more records  
IN TRUE if the operand is equal to one of a list of expressions  
LIKE TRUE if the operand matches a pattern  
NOT Displays a record if the condition(s) is NOT TRUE  
OR TRUE if any of the conditions separated by OR is TRUE  
SOME TRUE if any of the subquery values meet the condition

The WHERE clause can be combined with AND, OR, and NOT operators.

The AND and OR operators are used to filter records based on more than one condition:

The AND operator displays a record if all the conditions separated by AND are TRUE.  
The OR operator displays a record if any of the conditions separated by OR is TRUE.  
The NOT operator displays a record if the condition(s) is NOT TRUE.

```
select* from Cats
where age =10
and discount<05
```

The screenshot shows the MySQL Workbench interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The left sidebar has sections for Schemas, Tables, Columns, Indexes, Foreign Keys, Triggers, CustomerLog, and Hamsters. The main area has tabs for SQL, Results, and Data. A query window titled "SQL File 1" contains the following code:

```
1 • select* from Cats
2 where age =10
3 and discount<05
```

Below the code is a results grid with the following data:

	id	name	birthday	age	gender	price	discount	breed_id	shop_id
▶	2	Nemo	2018-02-12	10	M	108	0.1	1	3
▶	3	Vicont	2018-02-15	10	M	NULL	0	1	4
▶	23	Olle	2019-02-02	10	M	128	0	2	4
▶	24	Lilla	2019-02-12	10	M	108	0.1	1	3
▶	25	Kit	2019-01-15	10	M	NULL	0	1	4

```
select* from Cats  
where breed_id=3  
or breed_id= 4
```

The screenshot shows the MySQL Workbench interface with the 'LearnQA' database selected. In the 'Navigator' pane, the 'Tables' section is expanded, showing 'Breed' and 'Cats'. The 'Cats' table is selected, and its columns (id, name, birthday, age, gender, price, discount, breed\_id, shop\_id) are listed. A query editor window titled 'SQL File 1\*' contains the following SQL code:

```
1 • select* from Cats  
2 where breed_id=3  
3 or breed_id= 4  
4  
5  
6  
7  
8
```

The results grid displays 19 rows of data from the 'Cats' table, showing columns: id, name, birthday, age, gender, price, discount, breed\_id, and shop\_id. The data includes entries like Mars (id 5), Lapa (id 6), Glaflira (id 7), Tusya (id 9), Musya (id 10), Pusiceta (id 11), Tiran (id 14), Vasilii (id 15), Lokki (id 16), and Showman (id 19). The 'Result Grid' tab is active.

```
select* from Shops  
where country in ('Russia','Italy')
```

The screenshot shows the MySQL Workbench interface with the 'LearnQA' database selected. In the 'Navigator' pane, the 'Tables' section is expanded, showing 'Breed' and 'Shops'. The 'Shops' table is selected, and its columns (id, name, country, city, adds) are listed. A query editor window titled 'SQL File 1\*' contains the following SQL code:

```
1 • select* from Shops  
2 where country in ('Russia','Italy')  
3  
4  
5  
6  
7
```

The results grid displays 13 rows of data from the 'Shops' table, showing columns: id, name, country, city, and adds. The data includes entries like Four Paws (id 1), Mr.Zoo (id 2), Cats&Dogs (id 5), Cats&Dogs (id 10), Cats&Dogs (id 11), Four Paws (id 12), and Four Paws (id 13). The 'Result Grid' tab is active.

Object Info Session  
65 20:12:43 select\* from Shops where country in (Russia,Italy) LIMIT 0, 1000  
66 20:13:28 select\* from Shops where country in ('Russia','Italy') LIMIT 0, 1000  
Error Code: 1054. Unknown column 'Russia' in  
7 row(s) returned  
Query Completed

```
select* from Shops  
where not country='USA'
```

MySQL Workbench LearnQA x

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas

SCHEMAS: learnqa

Tables: Breeds, Cats

Columns: id, name, birthday, age, gender, price, discount, breed\_id, shop\_id

Indexes, Foreign Keys, Triggers, CustomerLog, Hamsters

Administration, Schemas

Information: Column: birthday

Definition: birthday date

Object Info Session

Query Completed

SQL File 1\* x

1 • select\* from Shops  
2 where not country='USA'  
3  
4  
5  
6  
7

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Result Grid | Form Editor | Field Types |

	id	name	country	city	adds
▶	1	Four Paws	Russia	Moscow	VDNH
2	Mr.Zoo	Russia	St.Petersburg	Vasilyevky Island	
4	Cats&Dogs	UK	London	Soho	
5	Cats&Dogs	Italy	Rome	Square of the mouth of truth	
6	猫と犬	Japanese	Tokio	Rome 場所	
8	Mr.Zoo	UK	Manchester	John Dalton St	
9	Murzila	Japanese	Kyoto	Shimogyo Ward	
10	Cats&Dogs	Italy	Venice	Metropolitan City of Venice	
11	Cats&Dogs	Russia	Moscow	Red Square	
12	Four Paws	Russia	Moscow	Arbat st	

Shops 50 x Apply Revert

Action Output

#	Time	Action	Message
67	20:16:13	select* from Shops LIMIT 0, 1000	14 row(s) returned
68	20:16:58	select* from Shops where not country='USA' LIMIT 0, 1000	12 row(s) returned

Output

Object Info Session

Query Completed

```
select* from Shops  
where country = 'USA'  
and city='New Yourk'  
or country = 'Japanese'  
and city='Tokio'
```

MySQL Workbench LearnQA x

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas

SCHEMAS: learnqa

Tables: Breeds, Cats

Columns: id, name, birthday, age, gender, price, discount, breed\_id, shop\_id

Indexes, Foreign Keys, Triggers

SQL File 1\* x

1 • select\* from Shops  
2 where country = 'USA'  
3 and city='New Yourk'  
4 or country = 'Japanese'  
5 and city='Tokio'  
6  
7  
8

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Result Grid | Form Editor | Field Types |

	id	name	country	city	adds
▶	6	猫と犬	Japanese	Tokio	場所
*	NULL	NULL	NULL	NULL	NULL

SQLAdditions: Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Object Info Session

Column: birthday

Definition: birthday date

Object Info Session

Query Completed

Shops 52 x

Action Output

#	Time	Action	Message	Duration / Fetch
69	20:21:24	select* from Shops LIMIT 0, 1000	14 row(s) returned	0.031 sec / 0.000 sec
70	20:22:47	select* from Shops where country ='USA' and city='New Yourk' or country ='Japanese' and ci...	1 row(s) returned	0.031 sec / 0.000 sec

## The SQL LIKE Operator

The `LIKE` operator is used in a `WHERE` clause to search for a specified pattern in a column.

There are two wildcards often used in conjunction with the `LIKE` operator:

- The percent sign (%) represents zero, one, or multiple characters
- The underscore sign (\_) represents one, single character

```
select* from Breeds
order by name
```

MySQL Workbench

LearnQA x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

learnqa

Tables

- Breeds
- Cats
- CustomerLog
- Hamsters
- Shops

Views

Stored Procedures

Functions

learnqa\_create\_alter\_drop

learnqa\_homework

Tables

Views

Stored Procedures

Functions

learnqa\_insert\_update\_delete

Administration Schemas

Information

Schema: learnqa

Result Grid

	id	name	character
5	abyssinian	playful	
7	american curl	bad boy	
10	american shorthair	lazy	
6	american wirehair	loyalty	
8	balinese-javanese	funny	
3	bengal	playful	
9	birman	brave	
4	british	loyalty	
1	maine coon	friendly	
11	ocicat	inquisitive	
12	persian	lazy	
13	ragamuffin	active	

Breeds 34 x

```
select* from Breeds
where name like "american %"
```

MySQL Workbench

LearnQA x

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas

SCHEMAS Filter objects

learnqa Tables Breeds Cats CustomerLog Hamsters Shops Views Stored Procedures Functions

learnqa\_create\_alter\_drop Tables Views Stored Procedures Functions

learnqa\_homework Tables Views Stored Procedures Functions

learnqa\_insert\_update\_delete

Administration Schemas

Information Schema: learnqa

Breeds 35 x

Result Grid | Filter Rows: Edit: Export/Import: Wrap Cell Content: Result Grid Form Editor Field Types

	id	name	character
▶	6	american wirehair	loyalty
▶	7	american curl	bad boy
▶	10	american shorthair	lazy
*	NULL	NULL	NULL

Apply Revert

The screenshot shows the MySQL Workbench interface with a query window titled "SQL File 1". The query is:

```
1 • select* from Breeds
2 where name like "american %"
```

The results grid displays three rows of data from the "Breeds" table:

	id	name	character
▶	6	american wirehair	loyalty
▶	7	american curl	bad boy
▶	10	american shorthair	lazy
*	NULL	NULL	NULL

select\* from Breeds  
where name like "a%"

The following SQL statement selects all breeds with a Name starting with "a":

MySQL Workbench

LearnQA x

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas

SCHEMAS Filter objects

learnqa Tables Breeds Cats CustomerLog Hamsters Shops Views Stored Procedures Functions

learnqa\_create\_alter\_drop Tables Views Stored Procedures Functions

learnqa\_homework Tables Views Stored Procedures Functions

learnqa\_insert\_update\_delete

Administration Schemas

Information Schema: learnqa

Breeds 36 x

Result Grid | Filter Rows: Edit: Export/Import: Wrap Cell Content: Result Grid Form Editor Field Types

	id	name	character
▶	5	abyssinian	playful
▶	6	american wirehair	loyalty
▶	7	american curl	bad boy
▶	10	american shorthair	lazy
*	NULL	NULL	NULL

Apply Revert

The screenshot shows the MySQL Workbench interface with a query window titled "SQL File 1". The query is:

```
1 • select* from Breeds
2 where name like "a%"
```

The results grid displays four rows of data from the "Breeds" table:

	id	name	character
▶	5	abyssinian	playful
▶	6	american wirehair	loyalty
▶	7	american curl	bad boy
▶	10	american shorthair	lazy
*	NULL	NULL	NULL

## SQL Aggregate Functions

Useful aggregate functions:

AVG() - Returns the average value  
COUNT() - Returns the number of rows  
FIRST() - Returns the first value  
LAST() - Returns the last value  
MAX() - Returns the largest value  
MIN() - Returns the smallest value  
SUM() - Returns the sum

### The AVG() Function

```
select avg(price)  
from Cats
```

The screenshot shows the MySQL Workbench interface. In the top navigation bar, the database is set to 'LearnQA'. The main area displays a SQL query in the 'SQL File 1' tab: 'select avg(price) from Cats'. The results are shown in a 'Result Grid' table, which contains one row with the value '123.8571'. The left sidebar shows the database schema, including the 'Cats' table with columns: id, name, birthday, age, gender, price, discount, breed\_id, and shop\_id.

```
SELECT breed_id, price FROM Cats  
WHERE Price>(SELECT AVG (Price) FROM Cats)
```

The screenshot shows the MySQL Workbench interface with the same database setup as the previous screenshot. In the 'SQL File 1' tab, a more complex query is entered: 'SELECT breed\_id, price FROM Cats WHERE Price>(SELECT AVG (Price) FROM Cats)'. This query filters the 'Cats' table to return only those entries where the price is greater than the average price of all cats.

The screenshot shows the MySQL Workbench interface with the 'Result Grid' pane open. The grid displays two columns: 'breed\_id' and 'price'. The data consists of 17 rows, with the first few rows being: 2, 128; 4, 144; 4, 150; 4, 125; 4, 137; 1, 140; 1, 145; 3, 128; 5, 213; 2, 154; 2, 128; 2, 128.

breed_id	price
2	128
4	144
4	150
4	125
4	137
1	140
1	145
3	128
5	213
2	154
2	128
2	128

```
SELECT name, age, price* (1-discount)
from Cats
where price is not null
```

The screenshot shows the MySQL Workbench interface with the 'Result Grid' pane open. The grid displays three columns: 'name', 'age', and 'price\* (1-discount)'. The data consists of 49 rows, with the first few rows being: Vlas, 1, 128; Nemo, 10, 97.1999983906746; Zuza, 9, 89.9999985098839; Mars, 7, 122.3999914169312; Lapa, 6, 94.99999254942; Glafira, 6, 127.4999910593033; Tigr, 5, 89.9999985098839; Musya, 4, 106.2499925494194; Pusiceta, 4, 116.4499918341637; Eroshka, 5, 125.9999979138374; Lisa, 4, 102.5999983012676; Tiran, 4, 113.9999991059303.

name	age	price* (1-discount)
Vlas	1	128
Nemo	10	97.1999983906746
Zuza	9	89.9999985098839
Mars	7	122.3999914169312
Lapa	6	94.99999254942
Glafira	6	127.4999910593033
Tigr	5	89.9999985098839
Musya	4	106.2499925494194
Pusiceta	4	116.4499918341637
Eroshka	5	125.9999979138374
Lisa	4	102.5999983012676
Tiran	4	113.9999991059303

```
select max(id) from Users
```

The screenshot shows the MySQL Workbench interface with the 'Result Grid' pane open. The grid displays one column: 'max(id)'. The data consists of a single row with the value 1.

max(id)
1

Schemas

- learnqa\_create\_alter\_drop
- learnqa\_homework
- Tables
- Views
- Stored Procedures
- Functions

Administration Schemas

Information

Schema: learnqa

Result Grid | Filter Rows: Export: Wrap Cell Content: Result 50 x Read Only

	max(id)
	49

Result Grid | Filter Rows: Export: Wrap Cell Content: Result 51 x Read Only

url	count(*)
store	591
robots	570
orders	568

```
select url, count(*) from Views group by url
order by count(*) desc
limit 3
```

MySQL Workbench

LearnQA x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- breed\_id
- shop\_id
- Indexes
- Foreign Keys
- Triggers
- CustomerLog
- Hamsters
- Shops
- Views
- Stored Procedures
- Functions

learnqa\_create\_alter\_drop

learnqa\_homework

- Tables
- Views
- Stored Procedures
- Functions

Administration Schemas

Information

Schema: learnqa

SQL File 1\*

```
1
2 •   select url, count(*) from Views group by url
3     order by count(*) desc
4     limit 3
5
```

Result Grid | Filter Rows: Export: Wrap Cell Content: Fetch rows: Result 51 x Read Only

url	count(*)
store	591
robots	570
orders	568