SQL Commands

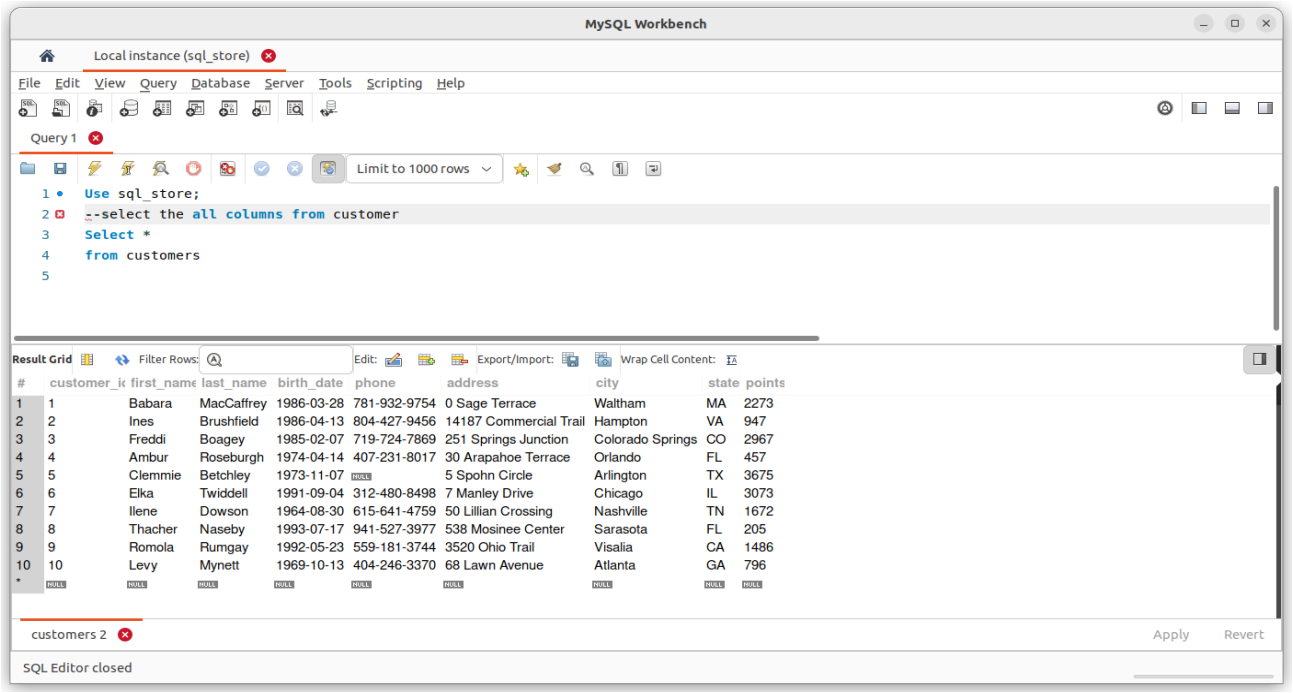
1-SELECT=

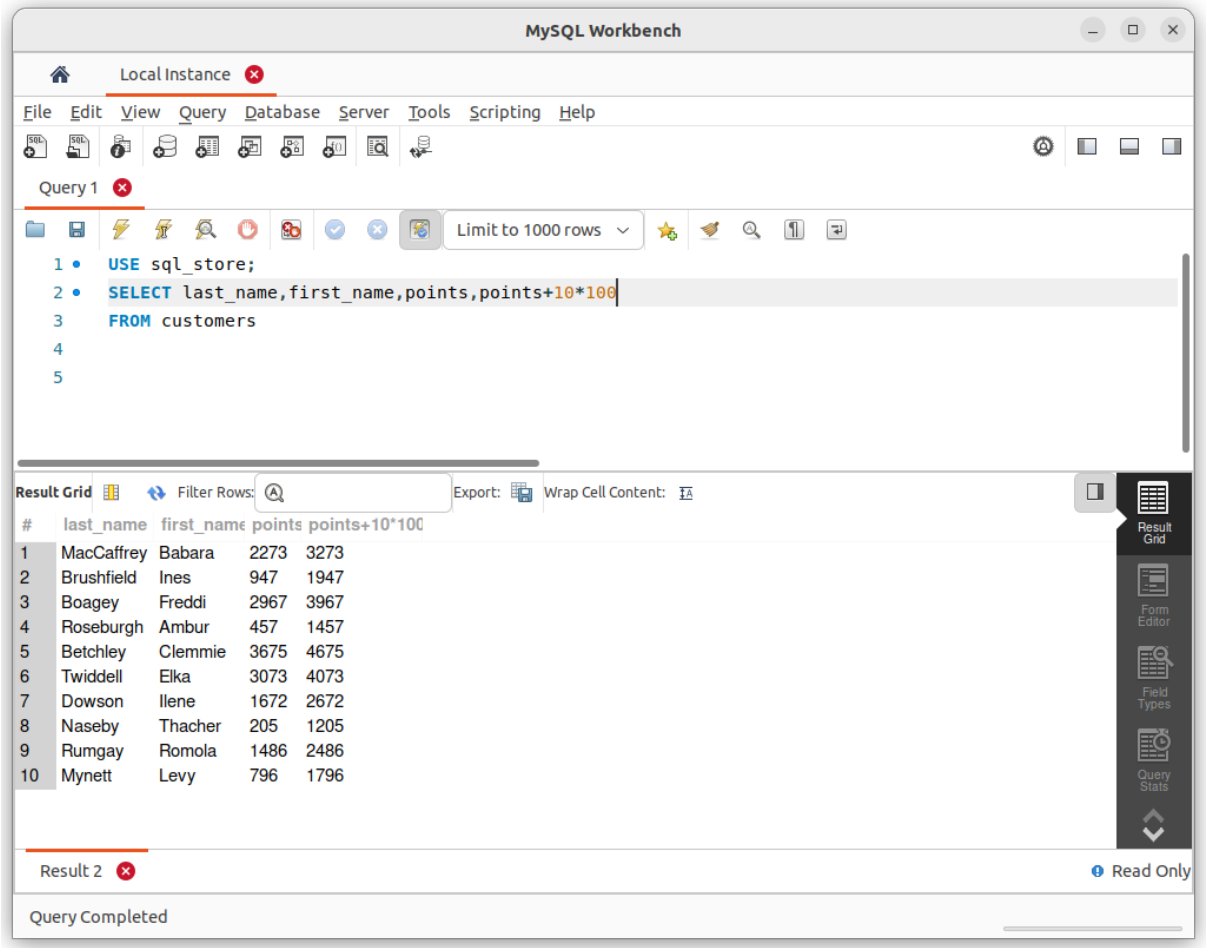
The SELECT statement is used to select data from a database.

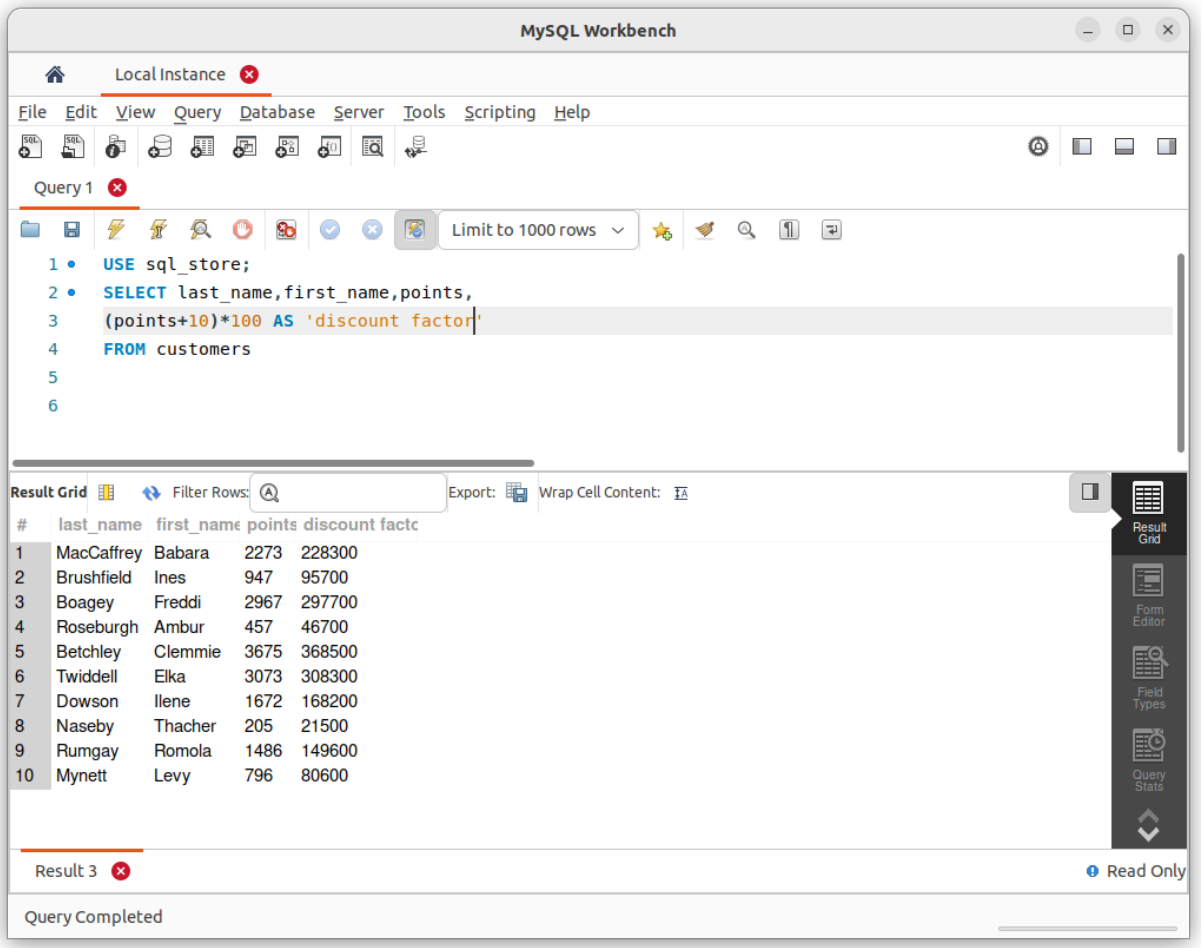
The data returned is stored in a result table, called the result-set.

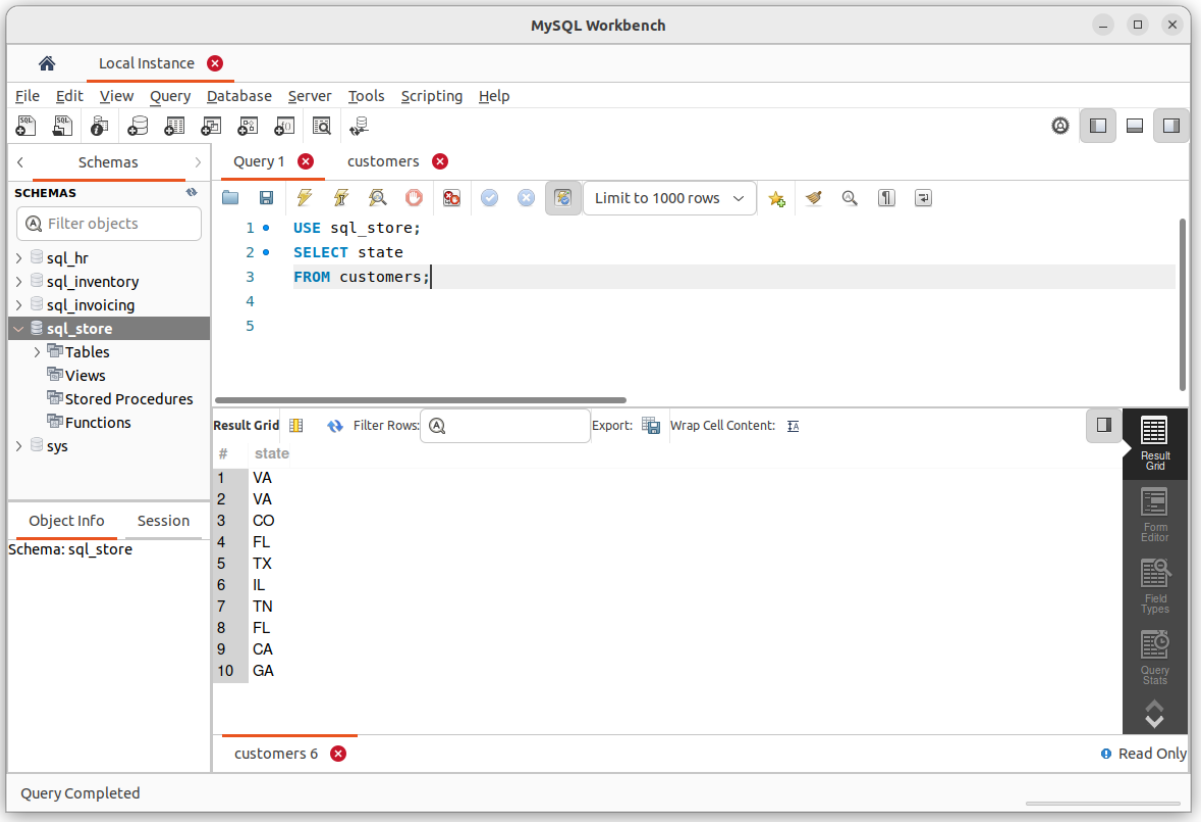
```
-- select all from customer table

SELECT *

FROM customers;

-- select specific columns

SELECT first_name, last_name

FROM customers;

-- select with arthematic expressions

SELECT first_name, last_name, points, points+10

FROM customers;

-- select with an alias

SELECT

first_name,
```

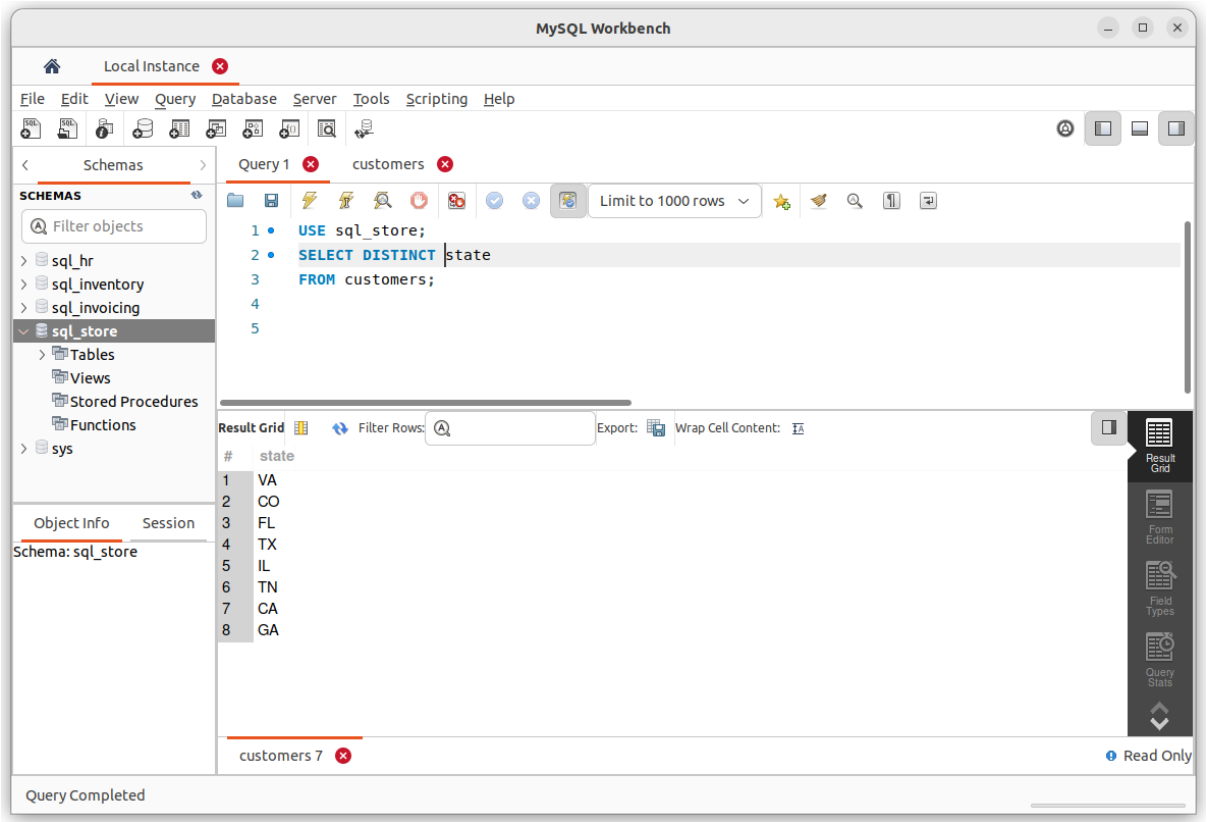
```
last_name,

points,

points+10 AS Points_new

FROM customers;

-- select distinct values

SELECT DISTINCT state

FROM customers;
```




MySQL Workbench
Local Instance
File Edit View Query Database Server Tools Scripting Help
Query 1
Limit to 1000 rows
1 • USE sql_store;
2 • SELECT last_name,first_name,points,points+10*100
3 FROM customers
4
5
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
last_name first_name points points+10*100
1 MacCaffrey Babara 2273 3273
2 Brushfield Ines 947 1947
3 Boagey Freddi 2967 3967
4 Roseburgh Ambur 457 1457
5 Betchley Clemmie 3675 4675
6 Twiddell Elka 3073 4073
7 Dowson Ilene 1672 2672
8 Naseby Thacher 205 1205
9 Rumgay Romola 1486 2486
10 Mynett Levy 796 1796
Result Grid
Form Editor
Field Types
Query Stats
Result 2
Read Only
Query Completed


MySQL Workbench
Local Instance
File Edit View Query Database Server Tools Scripting Help
Query 1
Limit to 1000 rows
1 • USE sql_store;
2 • SELECT last_name,first_name,points,
3 (points+10)*100 AS 'discount factor'
4 FROM customers
5
6
Result Grid Filter Rows: Export: Wrap Cell Content:
last_name first_name points discount factc
1 MacCaffrey Babara 2273 228300
2 Brushfield Ines 947 95700
3 Boagey Freddi 2967 297700
4 Roseburgh Ambur 457 46700
5 Betchley Clemmie 3675 368500
6 Twiddell Elka 3073 308300
7 Dowson Ilene 1672 168200
8 Naseby Thacher 205 21500
9 Rumgay Romola 1486 149600
10 Mynett Levy 796 80600
Result Grid
Form Editor
Field Types
Query Stats
Result 3
Read Only
Query Completed


MySQL Workbench
Local Instance
File Edit View Query Database Server Tools Scripting Help
Schemas
SCHEMAS
Filter objects
sql_hr
sql_inventory
sql_invoicing
sql_store
Tables
Views
Stored Procedures
Functions
sys
Object Info Session
Schema: sql_store
Query 1 customers
Limit to 1000 rows
1 • USE sql_store;
2 • SELECT state
3 FROM customers;
4
5
Result Grid Filter Rows: Export: Wrap Cell Content:
state
1 VA
2 VA
3 CO
4 FL
5 TX
6 IL
7 TN
8 FL
9 CA
10 GA
Result Grid
Form Editor
Field Types
Query Stats
customers 6
Read Only
Query Completed

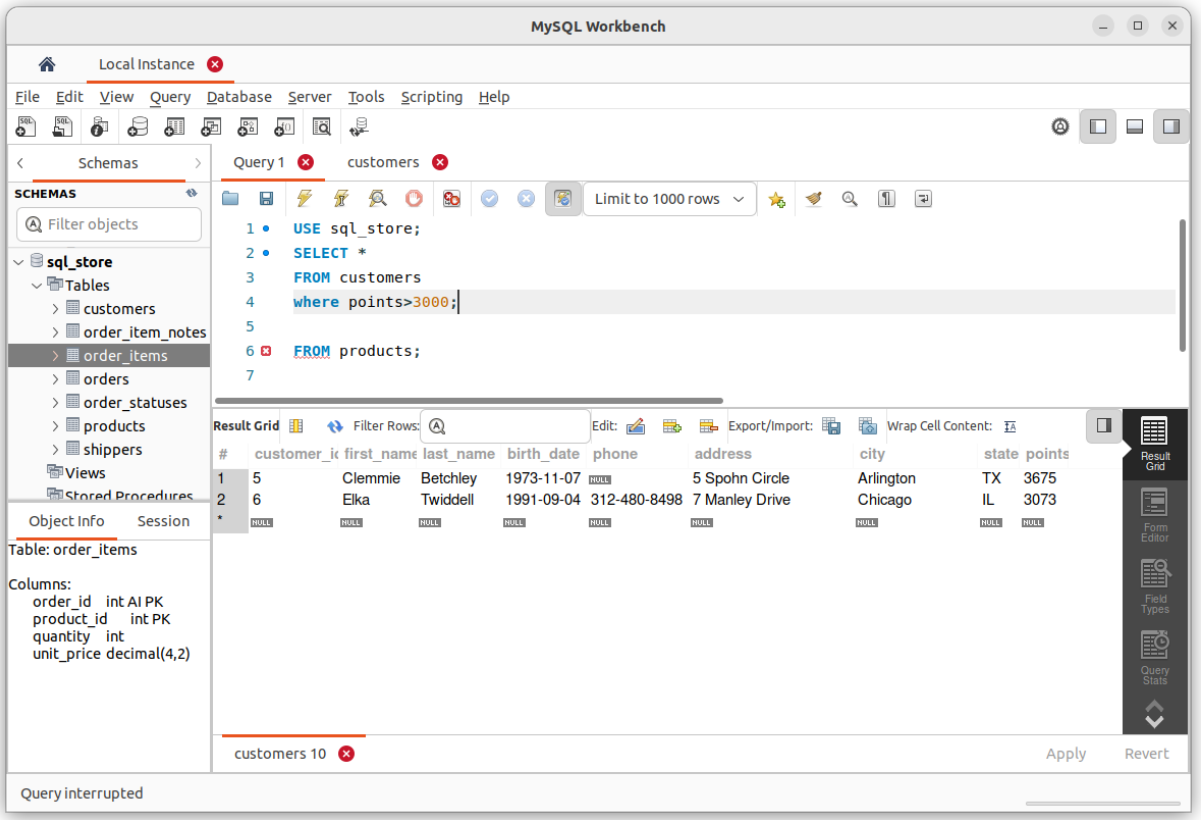


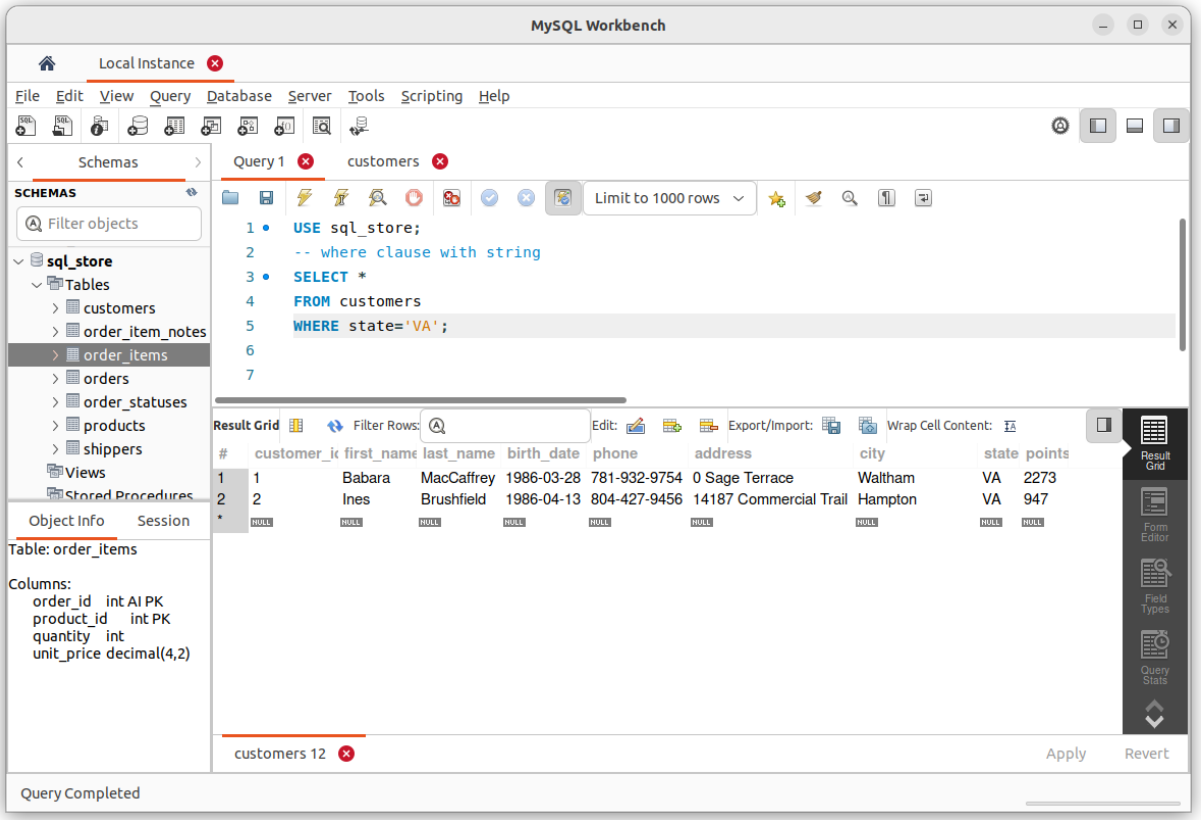
2- where-The WHERE clause is used to filter records.

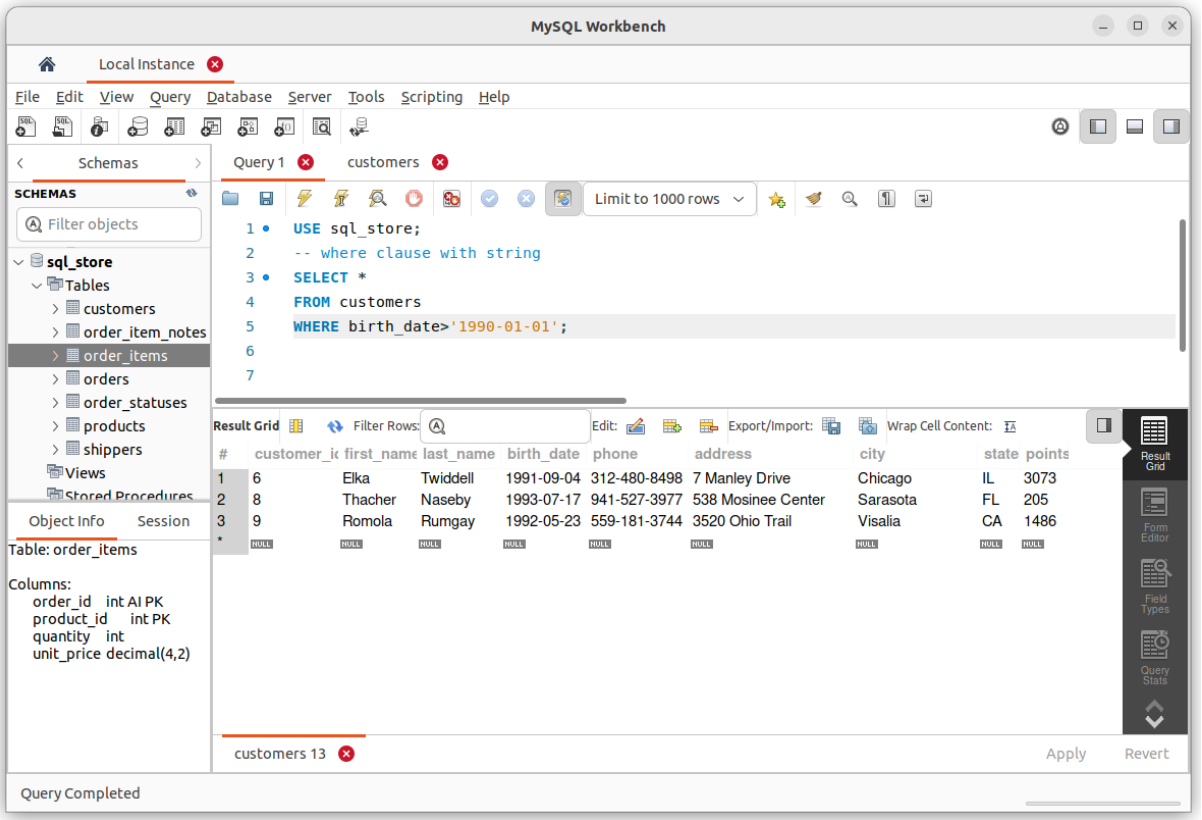
It is used to extract only those records that fulfill a specified condition.

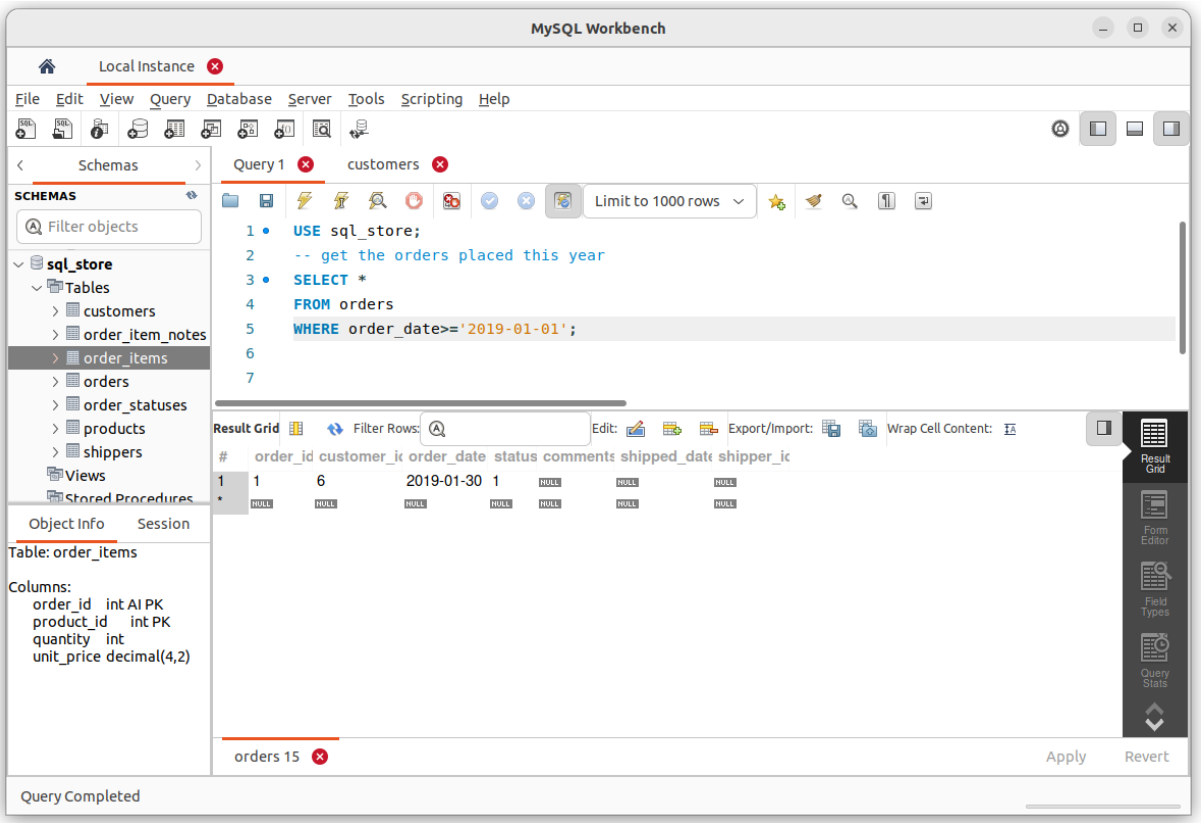
Basic Syntax-

SELECT *column1*, *column2, ...*
FROM *table_name*
WHERE *condition*;


MySQL Workbench
Local Instance
File Edit View Query Database Server Tools Scripting Help
Schemas
Query 1
customers
Limit to 1000 rows
1 USE sql_store;
2 SELECT *
3 FROM customers
4 where points>3000;
5
6 FROM products;
7
SCHEMAS
Filter objects
sql_store
Tables
customers
order_item_notes
order_items
orders
order_statuses
products
shippers
Views
Stored Procedures
Result Grid
Filter Rows:
Edit:
Export/Import:
Wrap Cell Content:
customer_id first_name last_name birth_date phone address city state points
1 5 Clemmie Betchley 1973-11-07 NULL 5 Spohn Circle Arlington TX 3675
2 6 Elka Twiddell 1991-09-04 312-480-8498 7 Manley Drive Chicago IL 3073
Object Info
Session
Table: order_items
Columns:
order_id int AI PK
product_id int PK
quantity int
unit_price decimal(4,2)
customers 10
Apply
Revert
Query interrupted


MySQL Workbench
Local Instance
File Edit View Query Database Server Tools Scripting Help
Schemas
Query 1
customers
Limit to 1000 rows
1 USE sql_store;
2 -- where clause with string
3 SELECT *
4 FROM customers
5 WHERE state='VA';
6
7
SCHEMAS
Filter objects
sql_store
Tables
customers
order_item_notes
order_items
orders
order_statuses
products
shippers
Views
Stored Procedures
Result Grid
Filter Rows:
Edit:
Export/Import:
Wrap Cell Content:
customer_id first_name last_name birth_date phone address city state points
1 1 Babara MacCaffrey 1986-03-28 781-932-9754 0 Sage Terrace Waltham VA 2273
2 2 Ines Brushfield 1986-04-13 804-427-9456 14187 Commercial Trail Hampton VA 947
Object Info
Session
Table: order_items
Columns:
order_id int AI PK
product_id int PK
quantity int
unit_price decimal(4,2)
customers 12
Apply
Revert
Query Completed





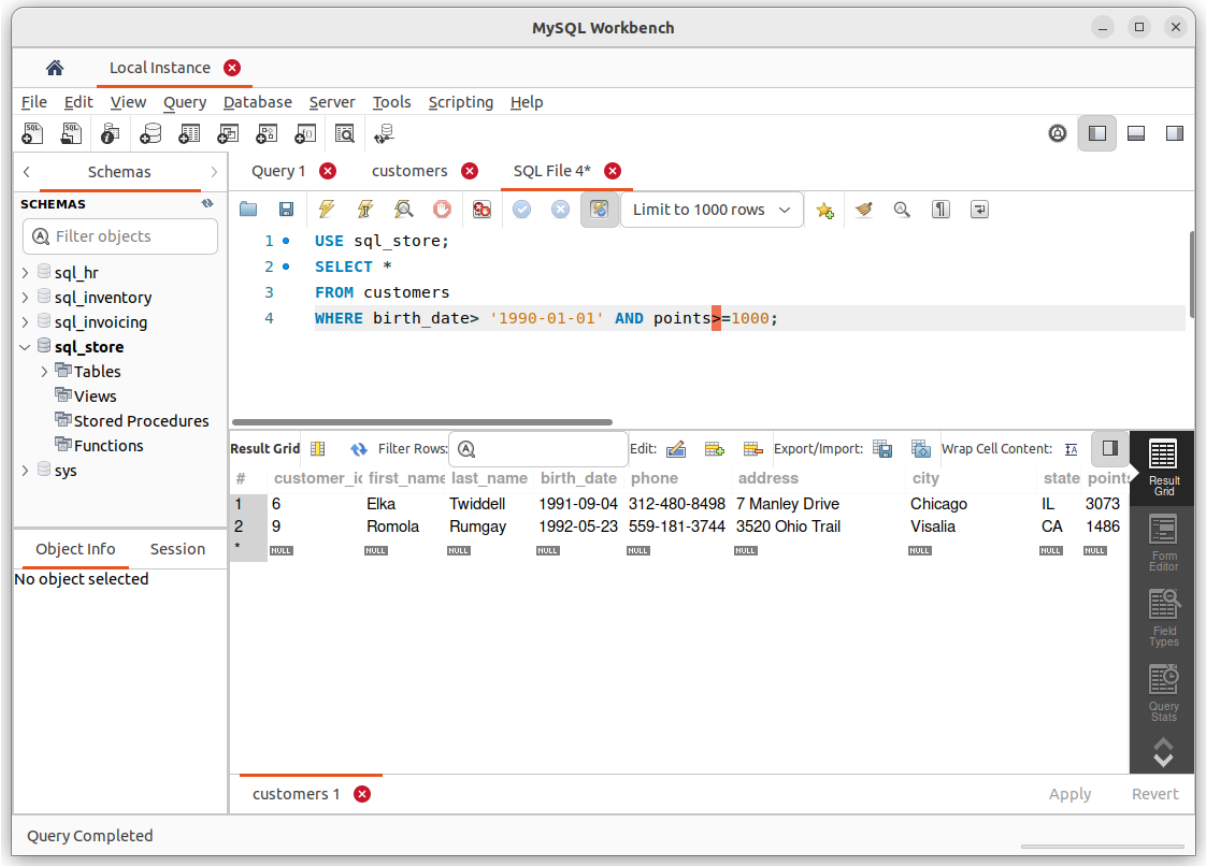
The SQL AND, OR and NOT Operators

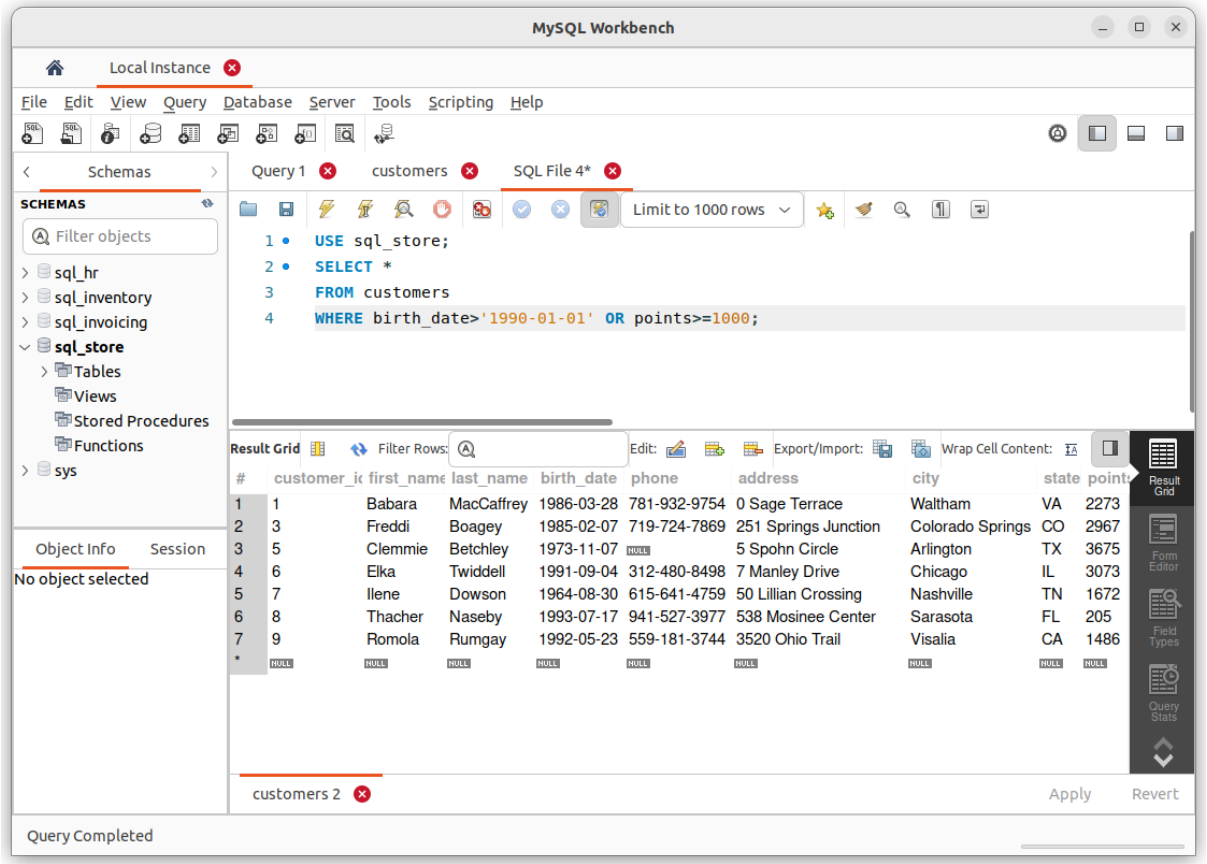
The `WHERE` clause can be combined with `AND`, `OR`, and `NOT` operators.

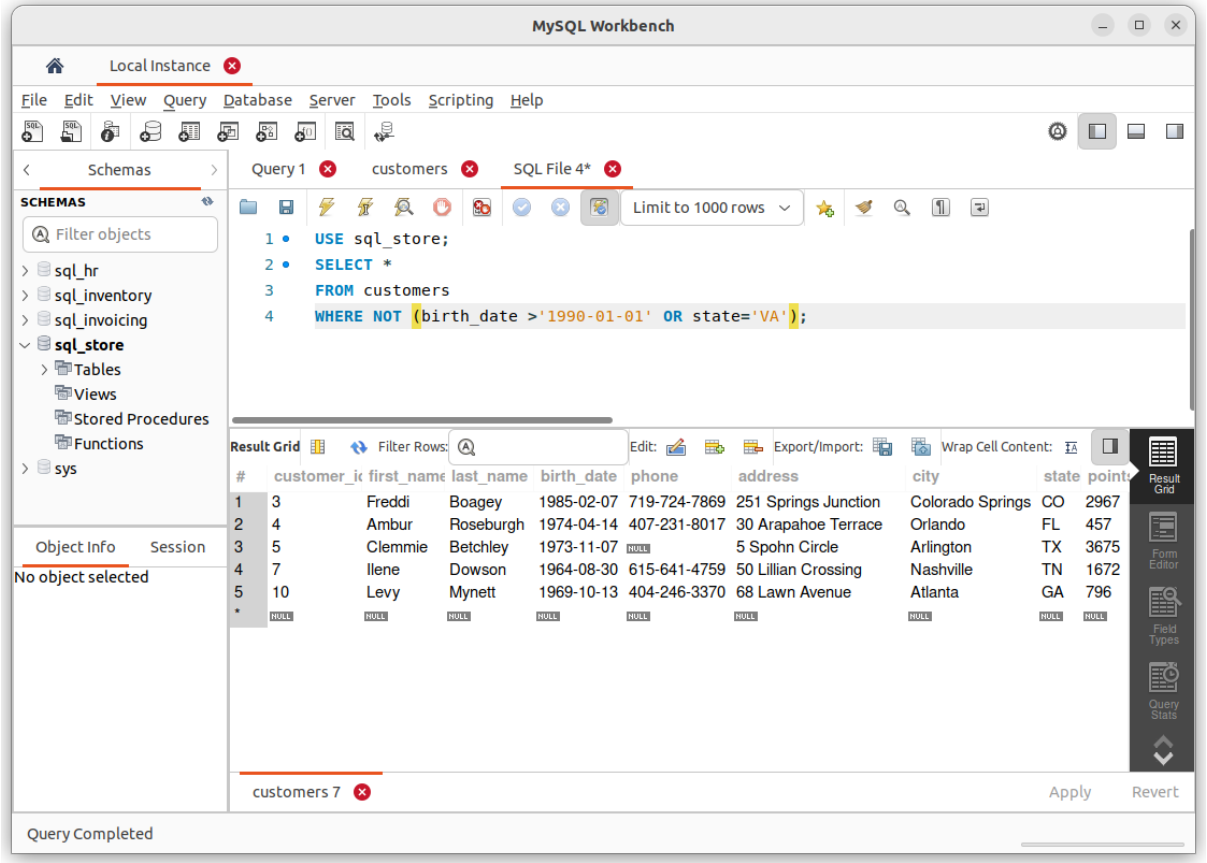
The AND and OR operators are used to filter records based on more than one condition:

- 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.

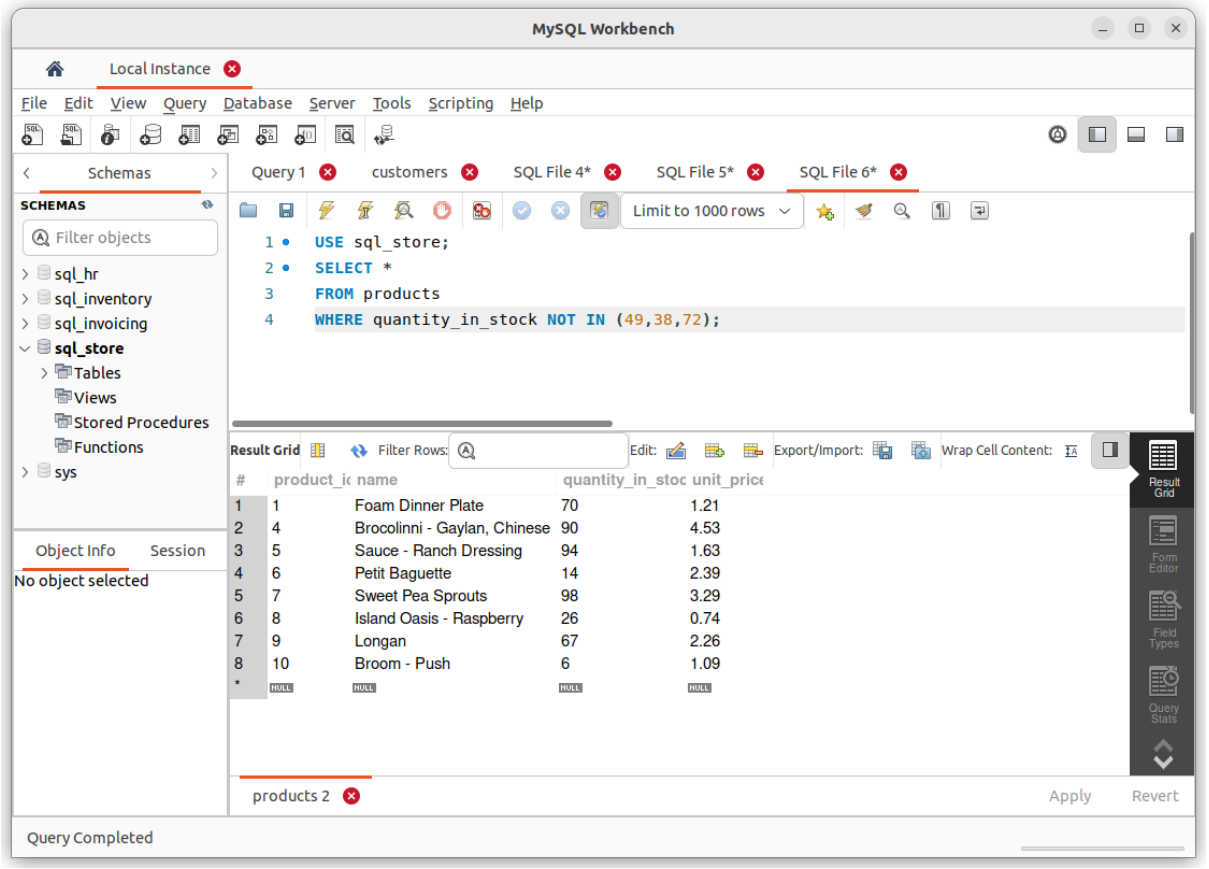



MySQL Workbench
Local Instance
File Edit View Query Database Server Tools Scripting Help
Schemas
Query 1
customers
SQL File 4*
Limit to 1000 rows
1 • USE sql_store;
2 • SELECT *
3 FROM customers
4 WHERE birth_date>'1990-01-01' OR points>=1000;
Result Grid
Filter Rows:
Edit:
Export/Import:
Wrap Cell Content:
customer_id first_name last_name birth_date phone address city state points
1 Babara MacCaffrey 1986-03-28 781-932-9754 0 Sage Terrace Waltham VA 2273
3 Freddi Boagey 1985-02-07 719-724-7869 251 Springs Junction Colorado Springs CO 2967
5 Clemmie Betchley 1973-11-07 5 Spohn Circle Arlington TX 3675
6 Elka Twiddell 1991-09-04 312-480-8498 7 Manley Drive Chicago IL 3073
7 Ilene Dowson 1964-08-30 615-641-4759 50 Lillian Crossing Nashville TN 1672
8 Thacher Naseby 1993-07-17 941-527-3977 538 Mosinee Center Sarasota FL 205
9 Romola Rumgay 1992-05-23 559-181-3744 3520 Ohio Trail Visalia CA 1486
SCHEMAS
Filter objects
sql_hr
sql_inventory
sql_invoicing
sql_store
Tables
Views
Stored Procedures
Functions
sys
Object Info
Session
No object selected
customers 2
Apply
Revert
Query Completed


MySQL Workbench
Local Instance
File Edit View Query Database Server Tools Scripting Help
Schemas
Query 1
customers
SQL File 4*
Limit to 1000 rows
1 • USE sql_store;
2 • SELECT *
3 FROM customers
4 WHERE NOT (birth_date >'1990-01-01' OR state='VA');
Result Grid
Filter Rows:
Edit:
Export/Import:
Wrap Cell Content:
customer_id first_name last_name birth_date phone address city state points
3 Freddi Boagey 1985-02-07 719-724-7869 251 Springs Junction Colorado Springs CO 2967
4 Ambur Roseburgh 1974-04-14 407-231-8017 30 Arapahoe Terrace Orlando FL 457
5 Clemmie Betchley 1973-11-07 5 Spohn Circle Arlington TX 3675
7 Ilene Dowson 1964-08-30 615-641-4759 50 Lillian Crossing Nashville TN 1672
10 Levy Mynett 1969-10-13 404-246-3370 68 Lawn Avenue Atlanta GA 796
SCHEMAS
Filter objects
sql_hr
sql_inventory
sql_invoicing
sql_store
Tables
Views
Stored Procedures
Functions
sys
Object Info
Session
No object selected
customers 7
Apply
Revert
Query Completed

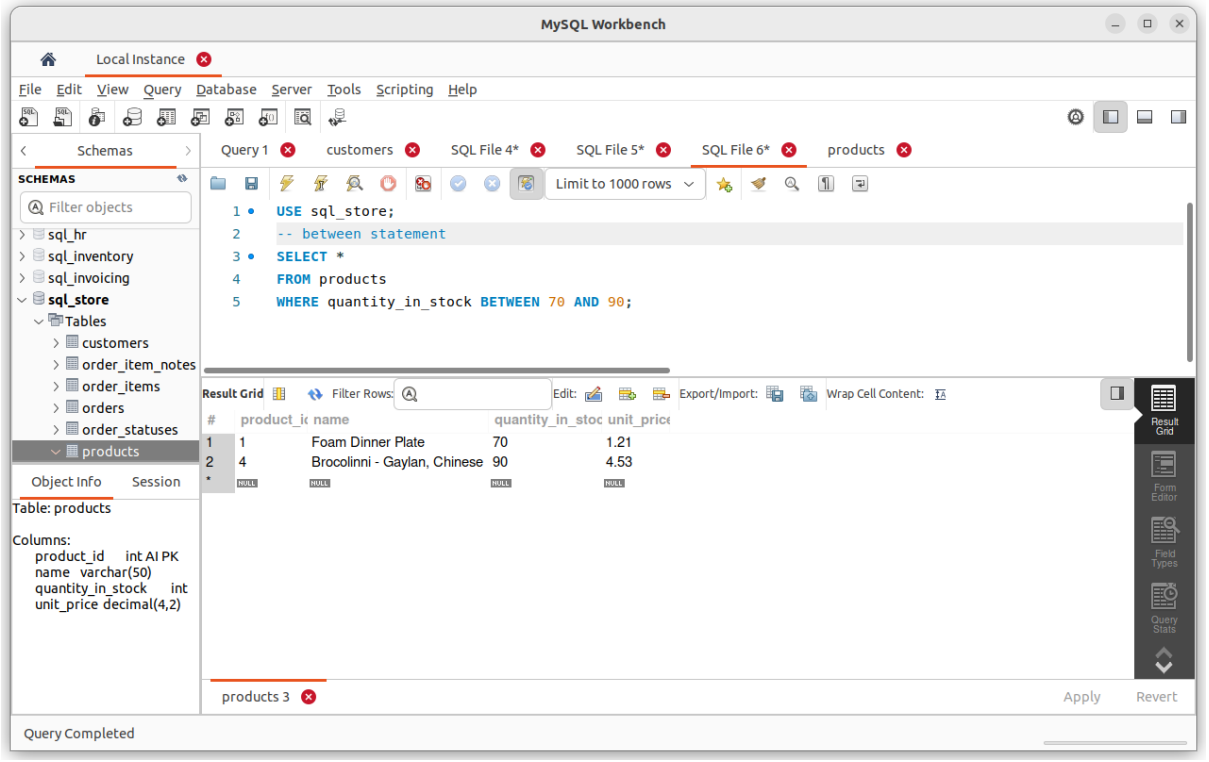
IN operator-

-- in statement
SELECT *
FROM customers
WHERE state IN ('VA' , 'GA' , 'FL



BETWEEN-

SELECT *
FROM table_name
WHERE column_name Between value_1 AND value_2;



LIKE operator-

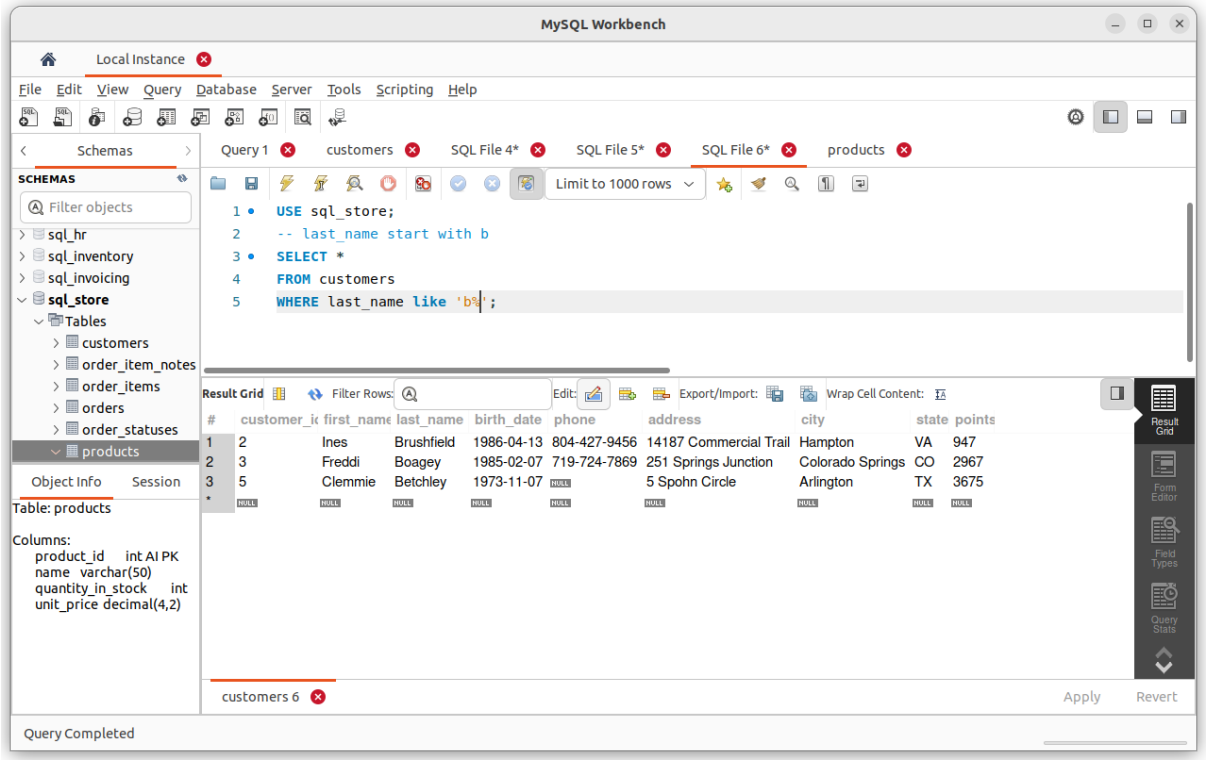
The `LIKE` operator is used in a `WHERE` clause to search for a specified pattern in a column.

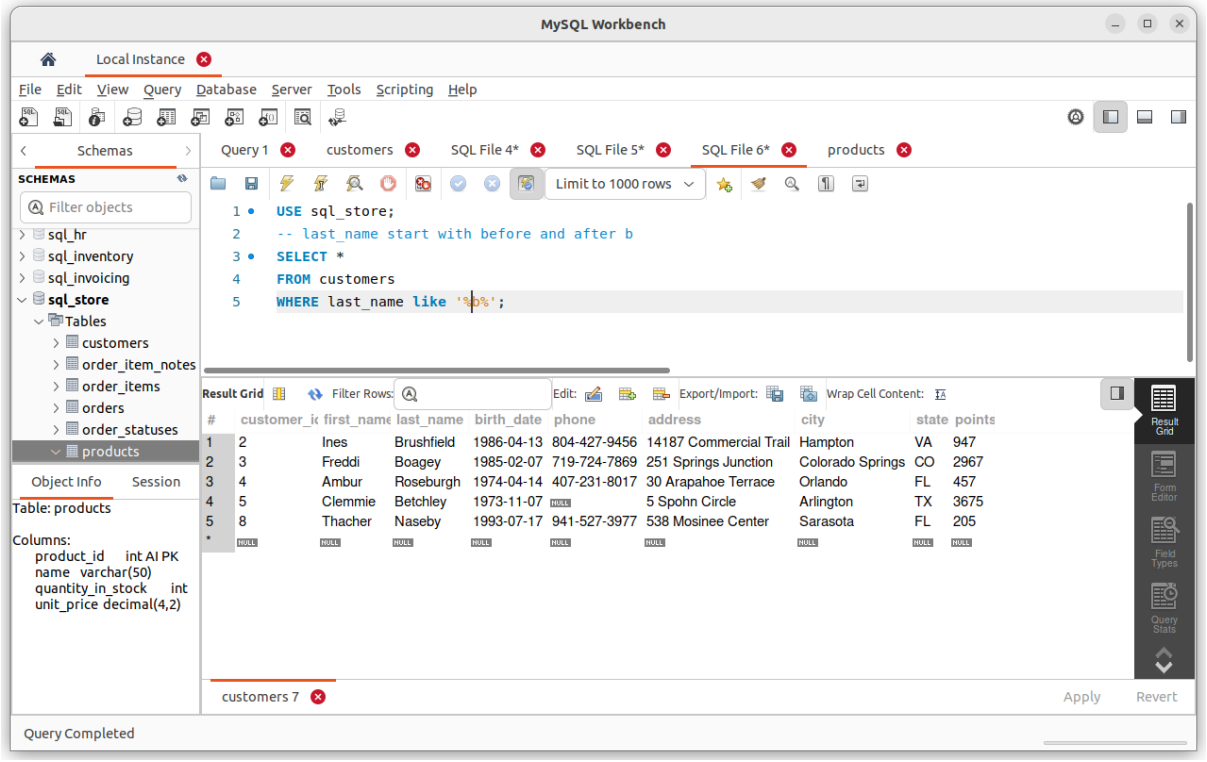
There are two wildcards often used in conjunction with the `LIKE` operator:

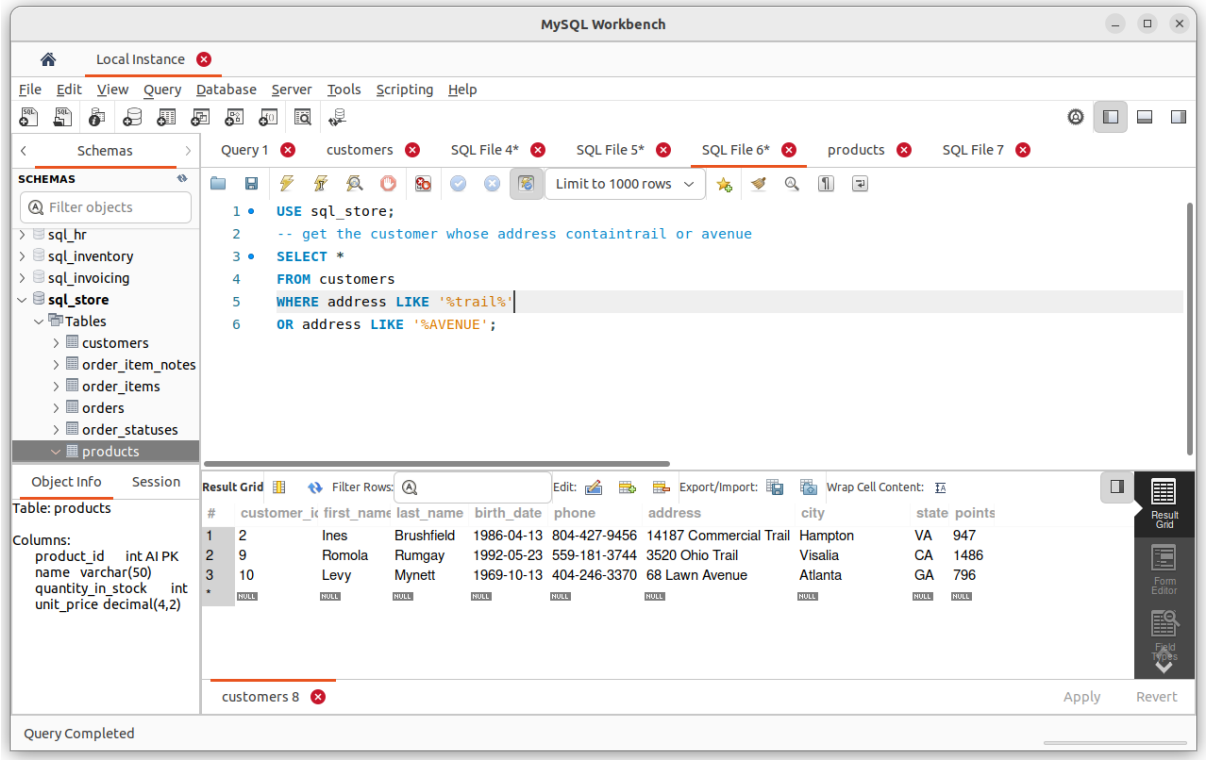
- The percent sign (%) represents zero, one, or multiple characters
- The underscore sign (_) represents one, single character

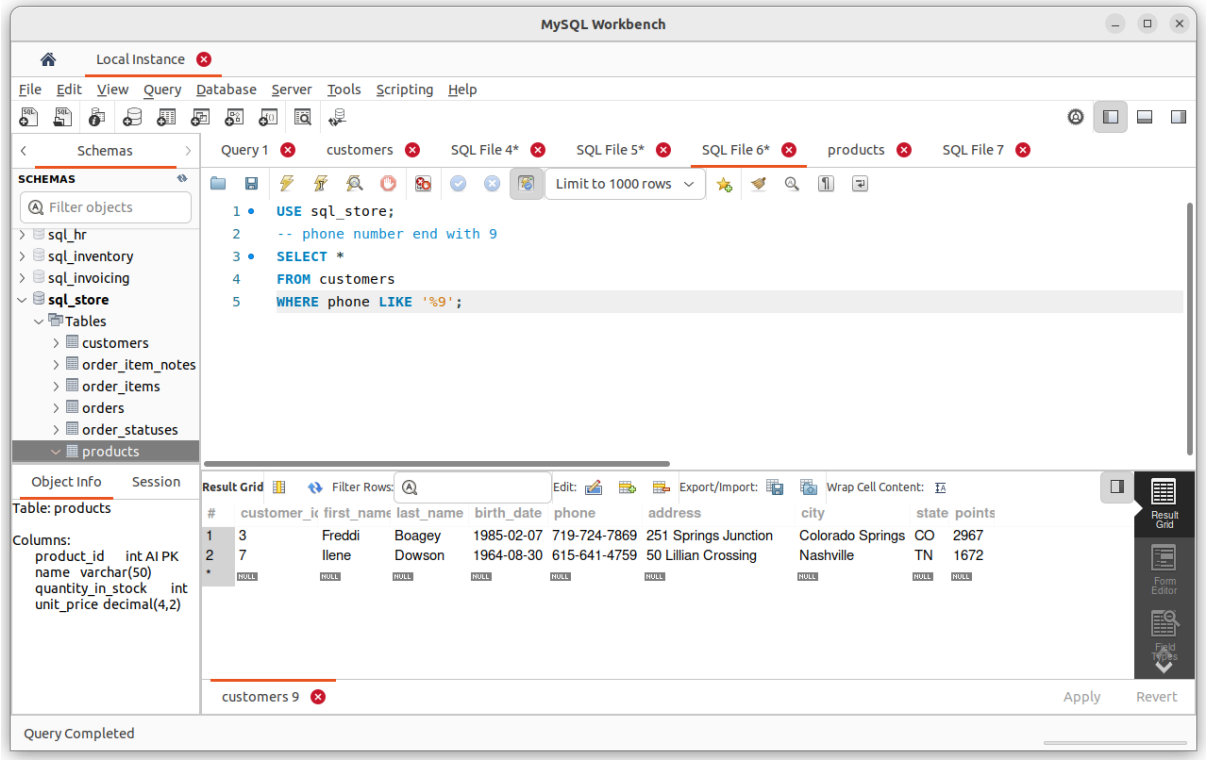
LIKE SYNTAX-

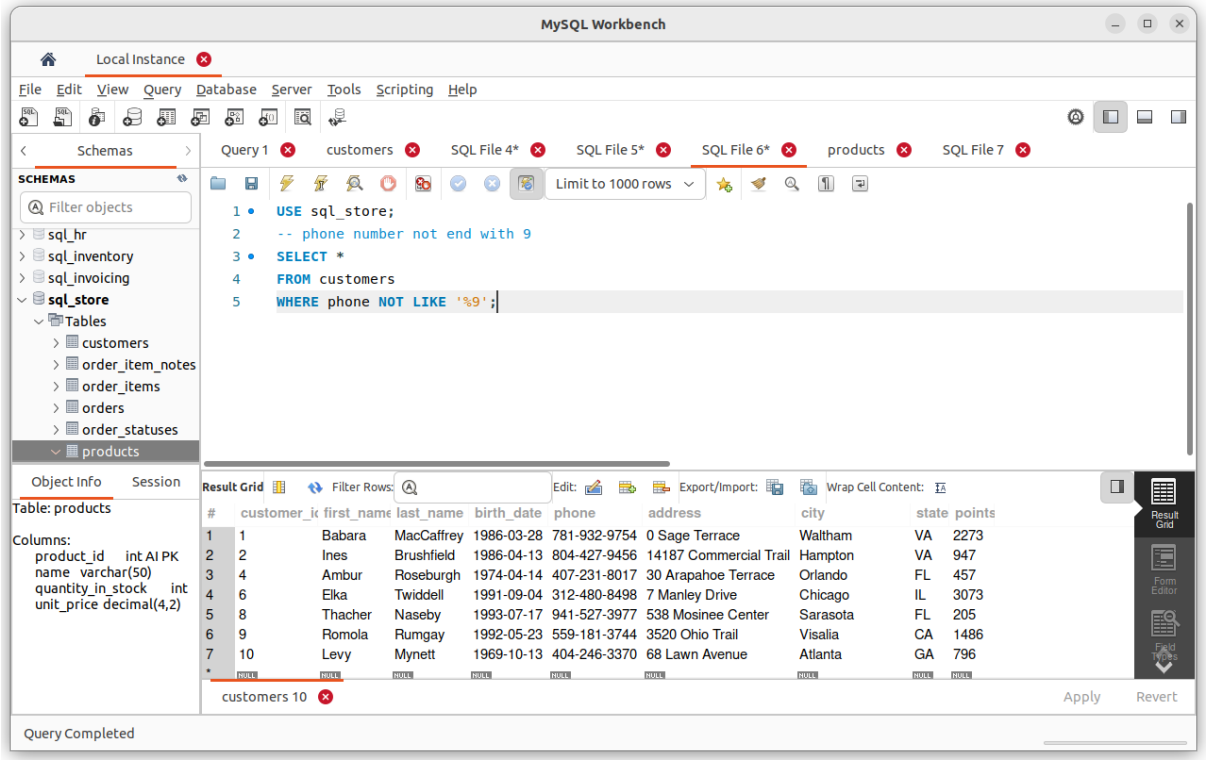
SELECT *column1, column2, ...*
FROM *table_name*
WHERE *columnN* LIKE *pattern*;


MySQL Workbench
Local Instance
File Edit View Query Database Server Tools Scripting Help
Schemas
Query 1 customers SQL File 4* SQL File 5* SQL File 6* products
SCHEMAS
Filter objects
sql_hr
sql_inventory
sql_invoicing
sql_store
Tables
customers
order_item_notes
order_items
orders
order_statuses
products
Object Info Session
Table: products
Columns:
product_id int AI PK
name varchar(50)
quantity_in_stock int
unit_price decimal(4,2)
Limit to 1000 rows
1 • USE sql_store;
2 -- last_name start with b
3 • SELECT *
4 FROM customers
5 WHERE last_name like 'b%';
Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:
customer_id first_name last_name birth_date phone address city state points
1 2 Ines Brushfield 1986-04-13 804-427-9456 14187 Commercial Trail Hampton VA 947
2 3 Freddi Boagey 1985-02-07 719-724-7869 251 Springs Junction Colorado Springs CO 2967
3 5 Clemmie Betchley 1973-11-07 NULL 5 Spohn Circle Arlington TX 3675
customers 6
Apply Revert
Query Completed


MySQL Workbench
Local Instance
File Edit View Query Database Server Tools Scripting Help
Schemas
Query 1 customers SQL File 4* SQL File 5* SQL File 6* products
SCHEMAS
Filter objects
sql_hr
sql_inventory
sql_invoicing
sql_store
Tables
customers
order_item_notes
order_items
orders
order_statuses
products
Object Info Session
Table: products
Columns:
product_id int AI PK
name varchar(50)
quantity_in_stock int
unit_price decimal(4,2)
Limit to 1000 rows
1 • USE sql_store;
2 -- last_name start with before and after b
3 • SELECT *
4 FROM customers
5 WHERE last_name like '%b%';
Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:
customer_id first_name last_name birth_date phone address city state points
1 2 Ines Brushfield 1986-04-13 804-427-9456 14187 Commercial Trail Hampton VA 947
2 3 Freddi Boagey 1985-02-07 719-724-7869 251 Springs Junction Colorado Springs CO 2967
3 4 Ambur Roseburgh 1974-04-14 407-231-8017 30 Arapahoe Terrace Orlando FL 457
4 5 Clemmie Betchley 1973-11-07 NULL 5 Spohn Circle Arlington TX 3675
5 8 Thacher Naseby 1993-07-17 941-527-3977 538 Mosinee Center Sarasota FL 205
customers 7
Apply Revert
Query Completed


MySQL Workbench
Local Instance
File Edit View Query Database Server Tools Scripting Help
Schemas
Query 1 customers SQL File 4* SQL File 5* SQL File 6* products SQL File 7
SCHEMAS
Filter objects
sql_hr
sql_inventory
sql_invoicing
sql_store
Tables
customers
order_item_notes
order_items
orders
order_statuses
products
Limit to 1000 rows
1 USE sql_store;
2 -- get the customer whose address containtrail or avenue
3 SELECT *
4 FROM customers
5 WHERE address LIKE '%trail%'
6 OR address LIKE '%AVENUE';
Object Info Session
Table: products
Columns:
product_id int AI PK
name varchar(50)
quantity_in_stock int
unit_price decimal(4,2)
Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:
customer_id first_name last_name birth_date phone address city state points
1 2 Ines Brushfield 1986-04-13 804-427-9456 14187 Commercial Trail Hampton VA 947
2 9 Romola Rumgay 1992-05-23 559-181-3744 3520 Ohio Trail Visalia CA 1486
3 10 Levy Mynett 1969-10-13 404-246-3370 68 Lawn Avenue Atlanta GA 796
customers 8
Apply Revert
Query Completed

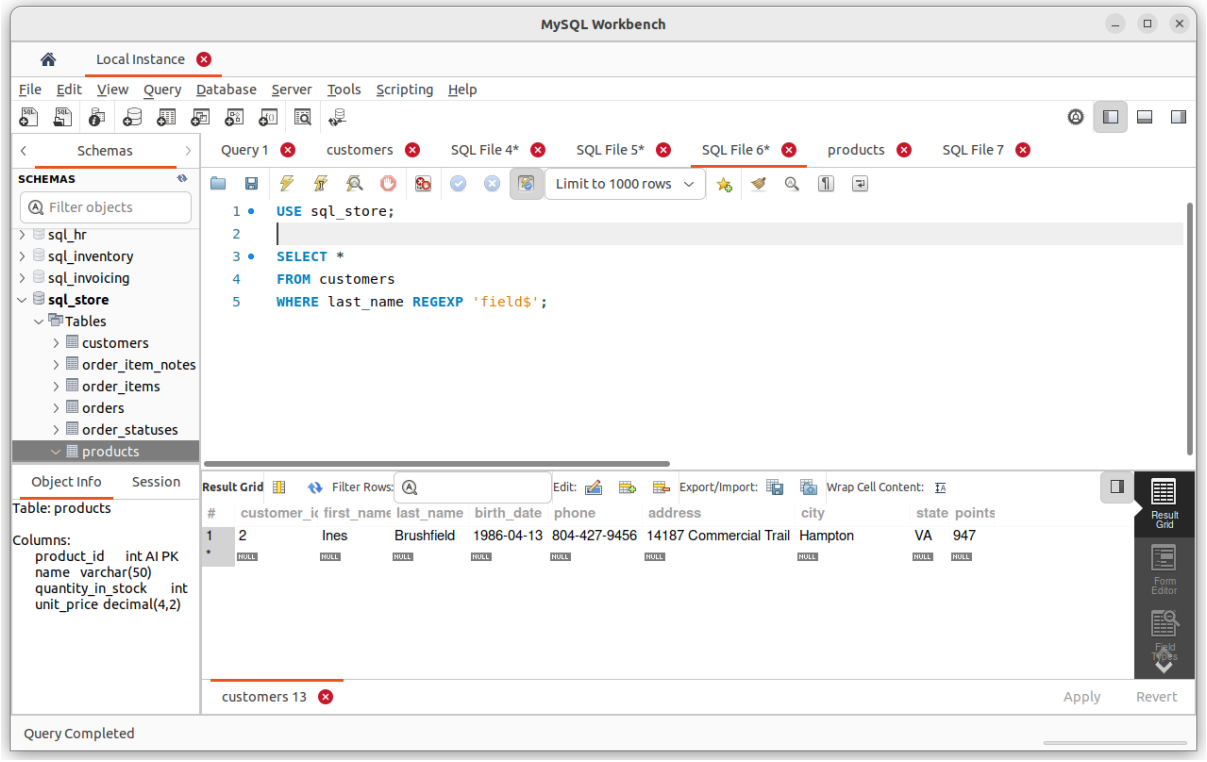

MySQL Workbench
Local Instance
File Edit View Query Database Server Tools Scripting Help
Schemas
Query 1 customers SQL File 4* SQL File 5* SQL File 6* products SQL File 7
SCHEMAS
Filter objects
sql_hr
sql_inventory
sql_invoicing
sql_store
Tables
customers
order_item_notes
order_items
orders
order_statuses
products
Limit to 1000 rows
1 USE sql_store;
2 -- phone number end with 9
3 SELECT *
4 FROM customers
5 WHERE phone LIKE '%9';
Object Info Session
Table: products
Columns:
product_id int AI PK
name varchar(50)
quantity_in_stock int
unit_price decimal(4,2)
Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:
customer_id first_name last_name birth_date phone address city state points
1 3 Freddi Boagey 1985-02-07 719-724-7869 251 Springs Junction Colorado Springs CO 2967
2 7 Ilene Dowson 1964-08-30 615-641-4759 50 Lillian Crossing Nashville TN 1672
customers 9
Apply Revert
Query Completed



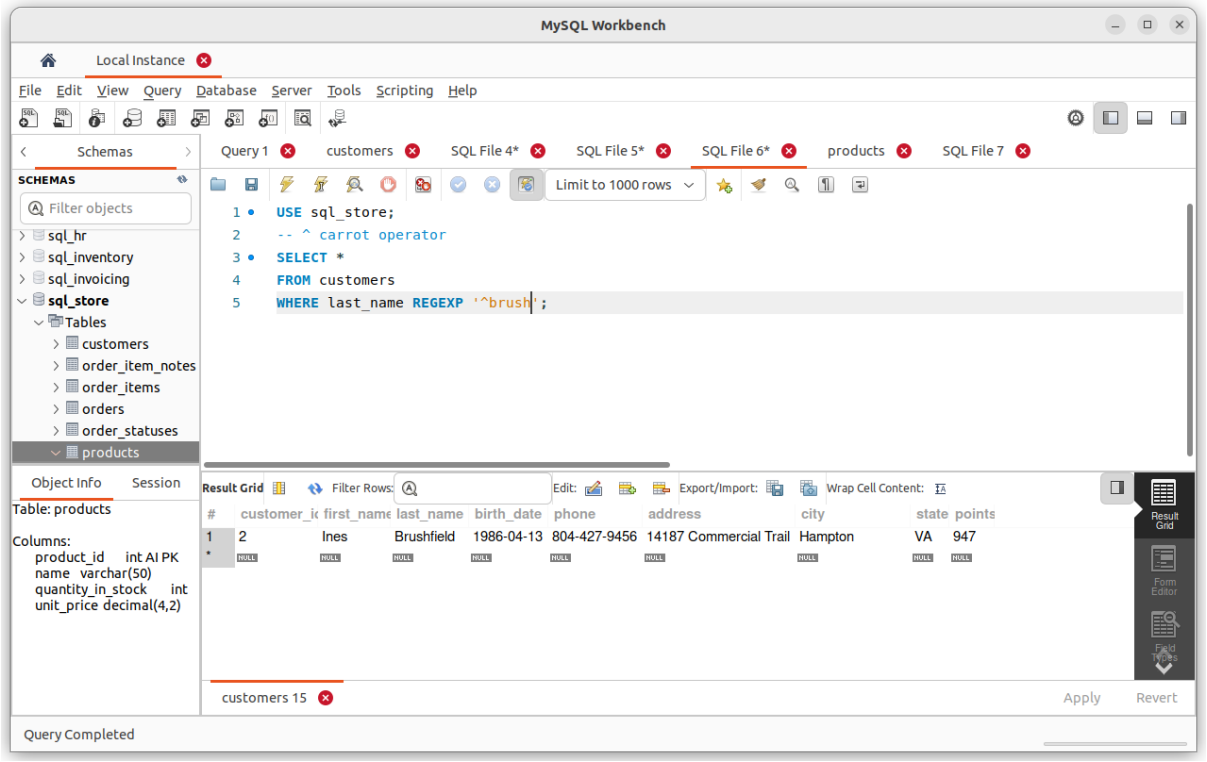
The REGEXP operator:

It is a short for regular expression, and regular expressions are extremely powerful when it comes to searching for strings.

```
-- regexp for string/char anywhere in lastname
SELECT *
FROM customers
WHERE last_name REGEXP 'field';
```



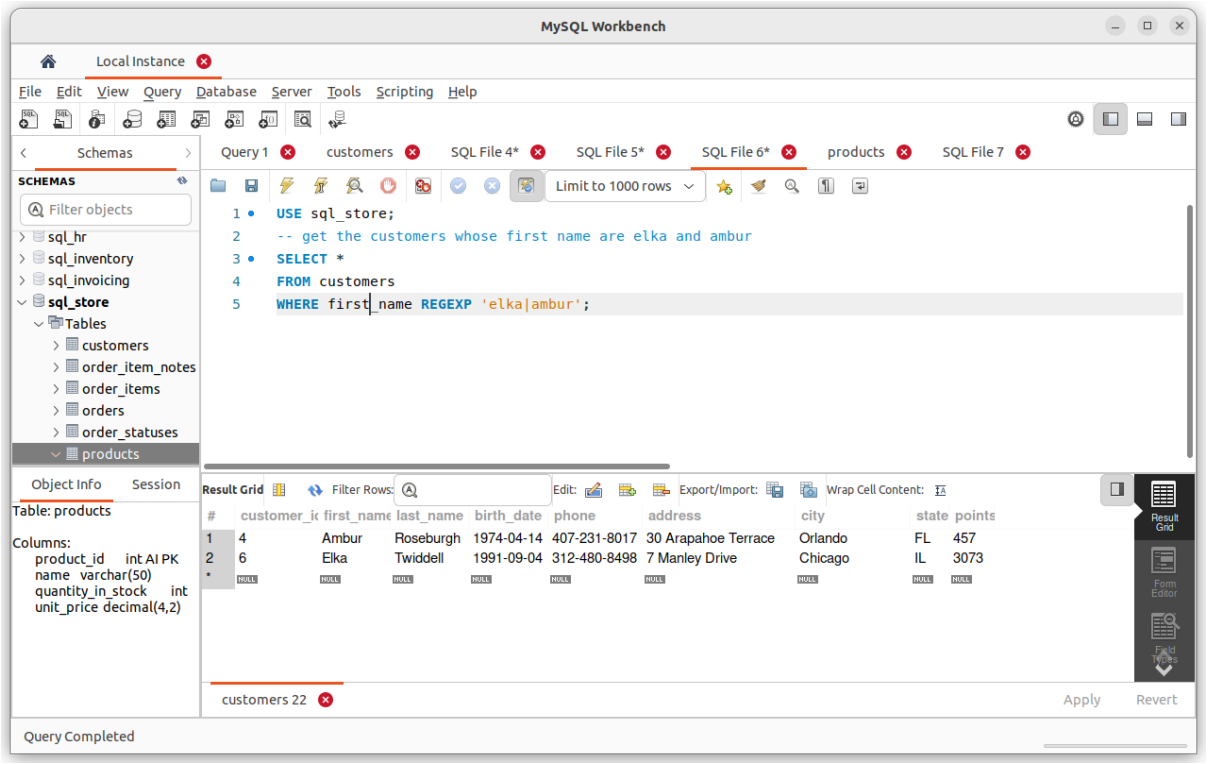
-- regexp for string/char in begining of lastname
SELECT *
FROM customers
WHERE last_name REGEXP '^brush';



-- regexp for multiple string/char anywhere in lastname
SELECT *

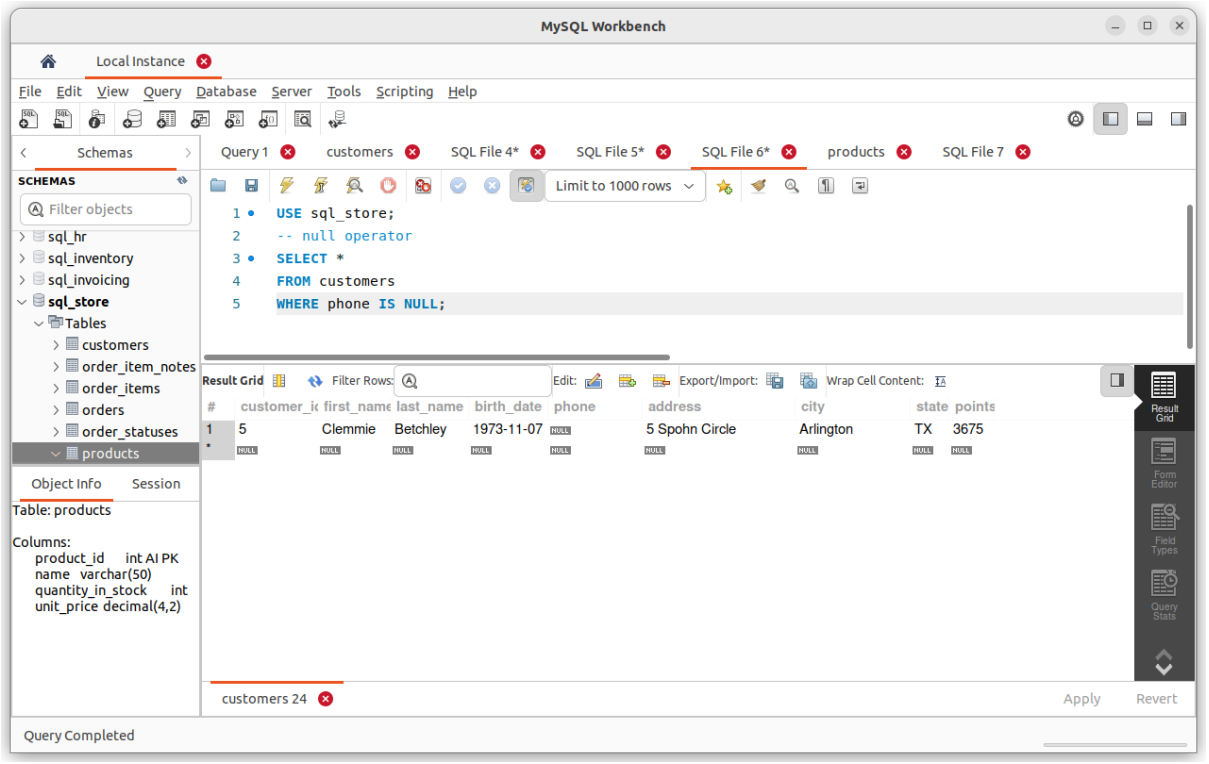
FROM customers

WHERE last_name REGEXP 'field|mac';



IS NULL Syntax

SELECT *column_names*
FROM *table_name*
WHERE *column_name* IS NULL;



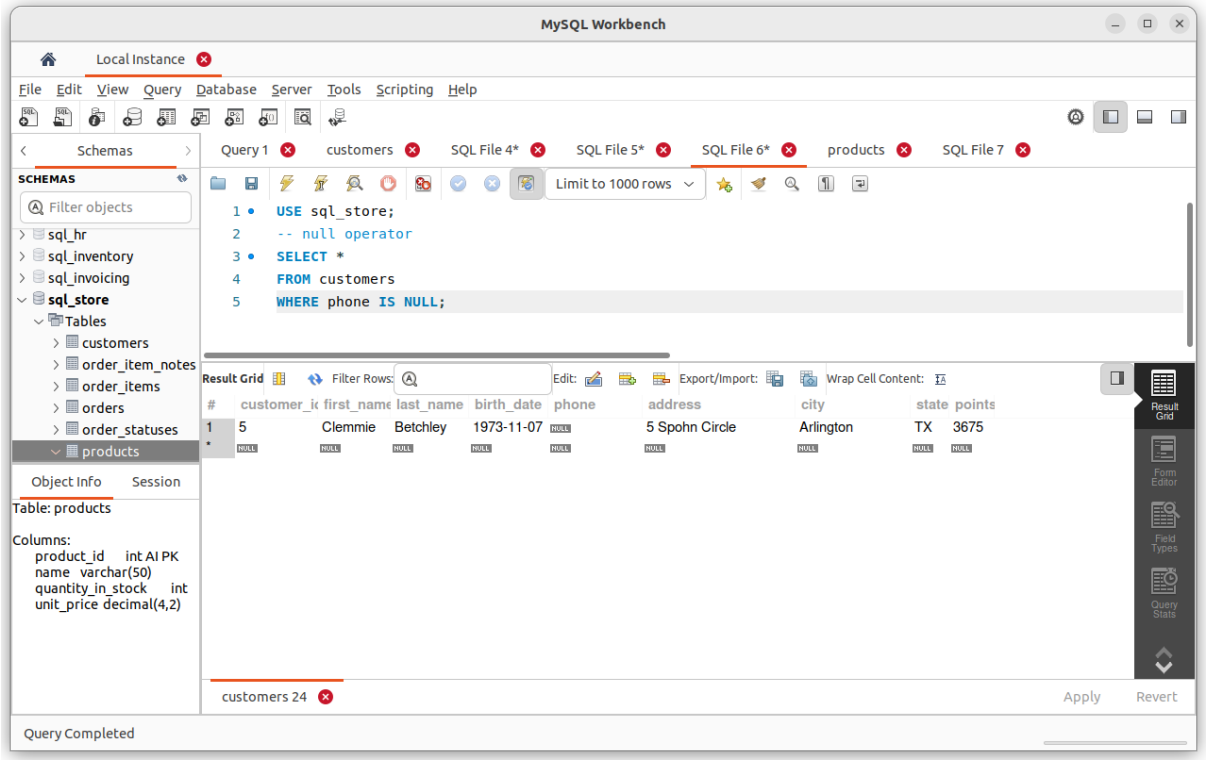
3- ORDER BY

-- order by to sort data in squel queries

SELECT *

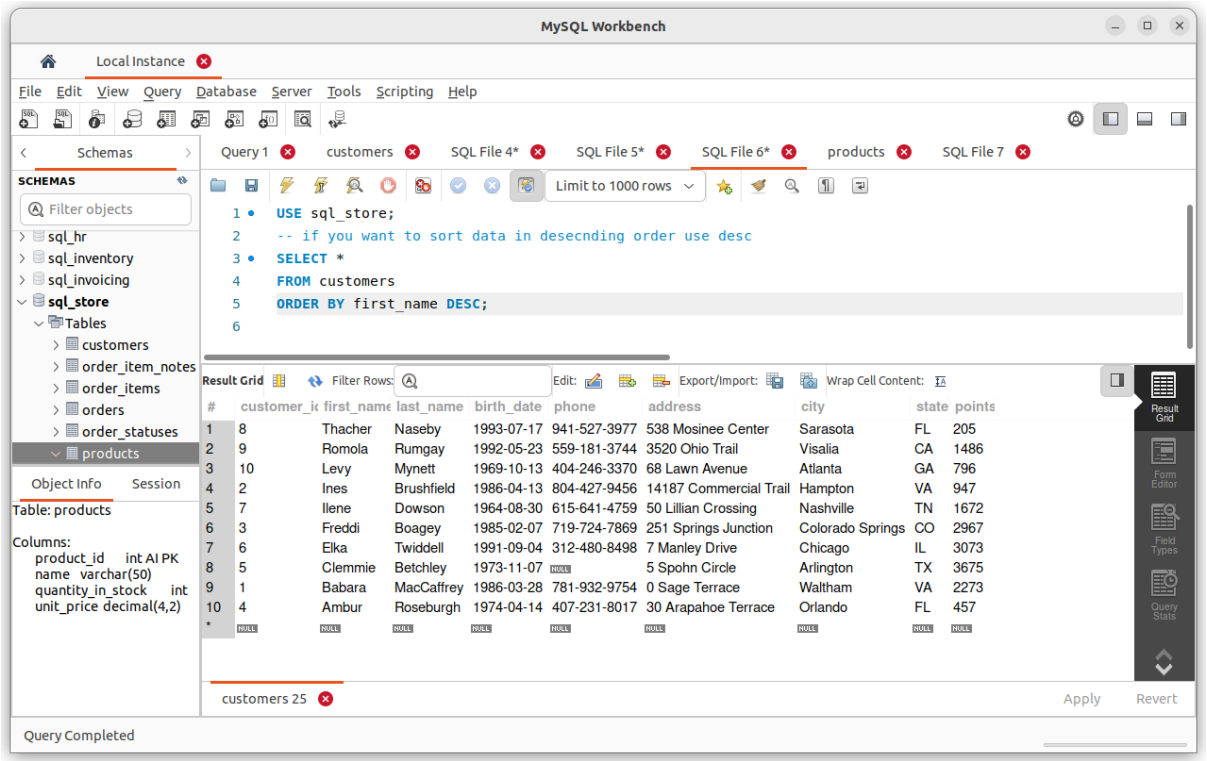
FROM customers

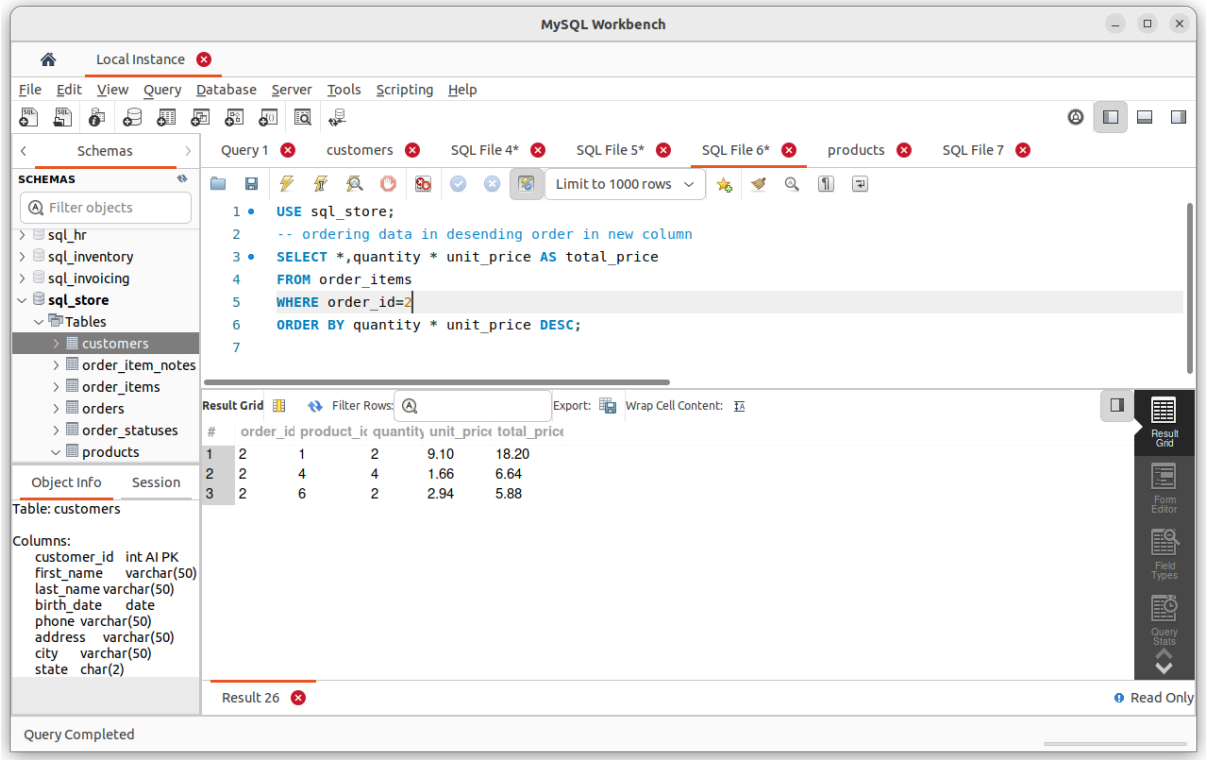
ORDER BY first_name;



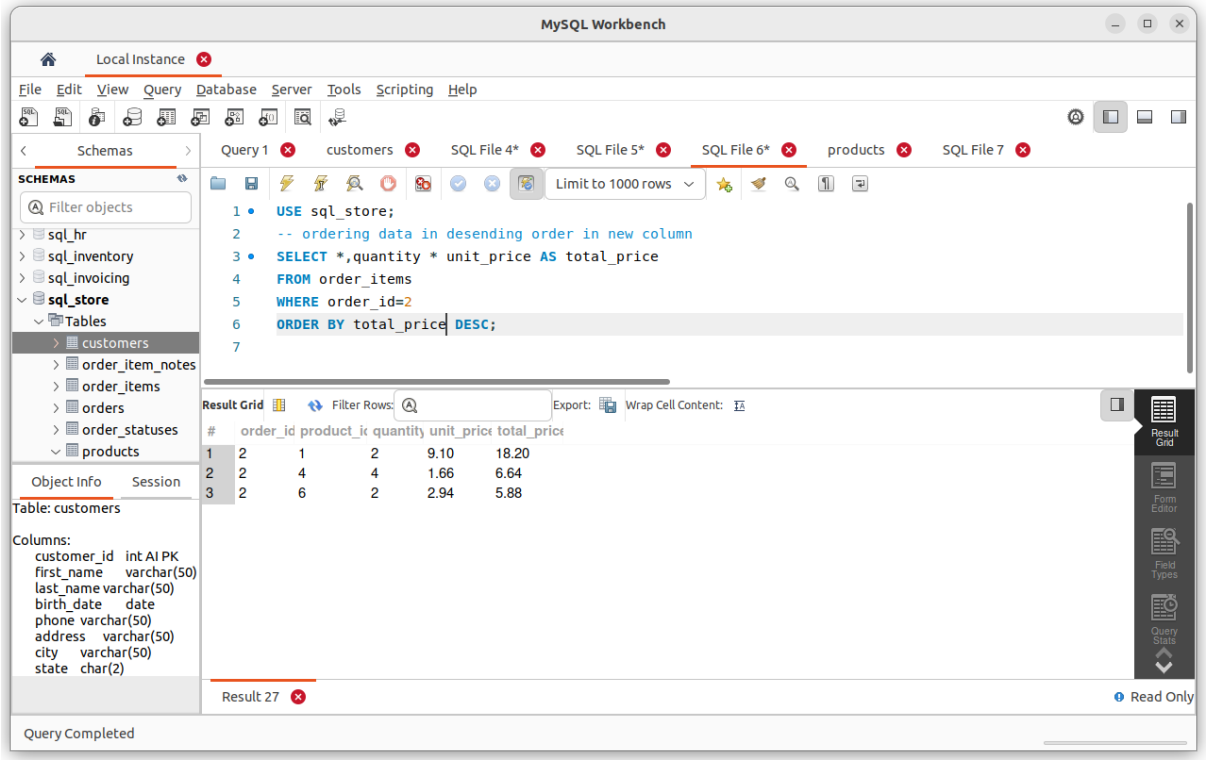
-- if you want to sort data in desecnding order use desc

```
SELECT *
FROM customers
ORDER BY first_name DESC;
```





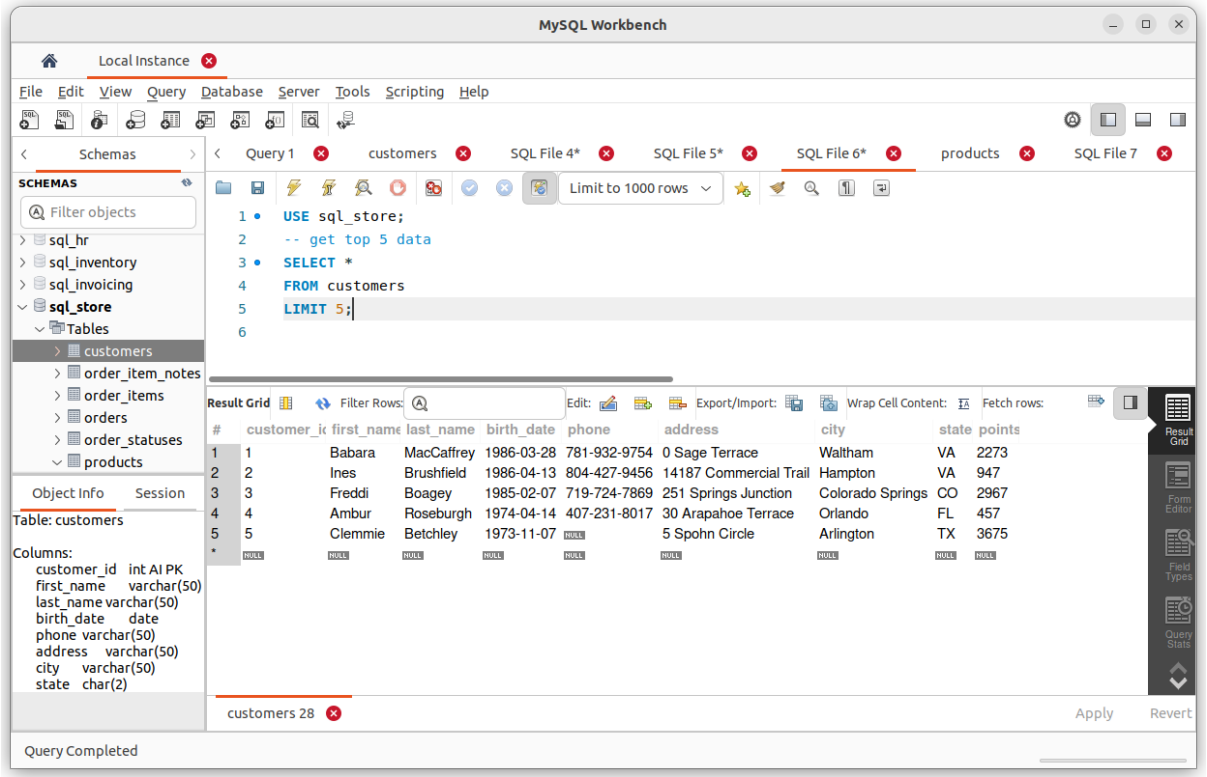
– sort the data by a

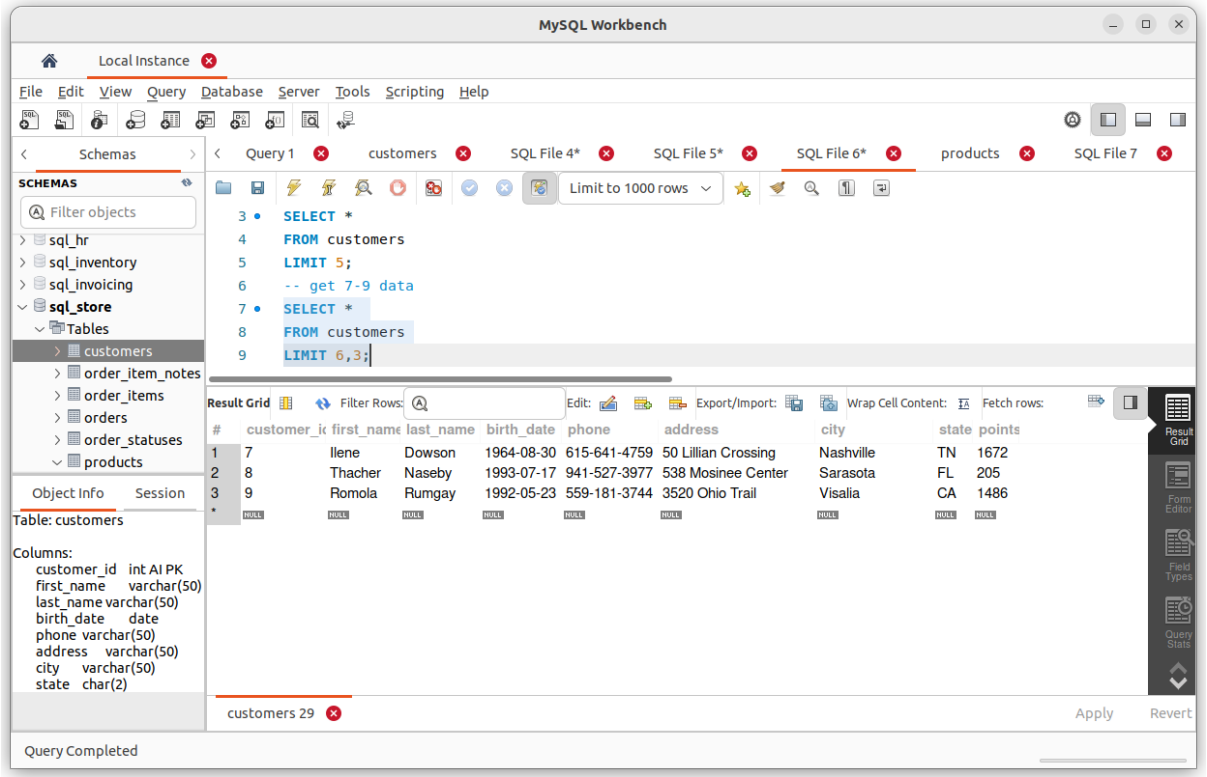


LIMIT CLAUSE-

-- get top 3 data
SELECT *
FROM customers
LIMIT 3;

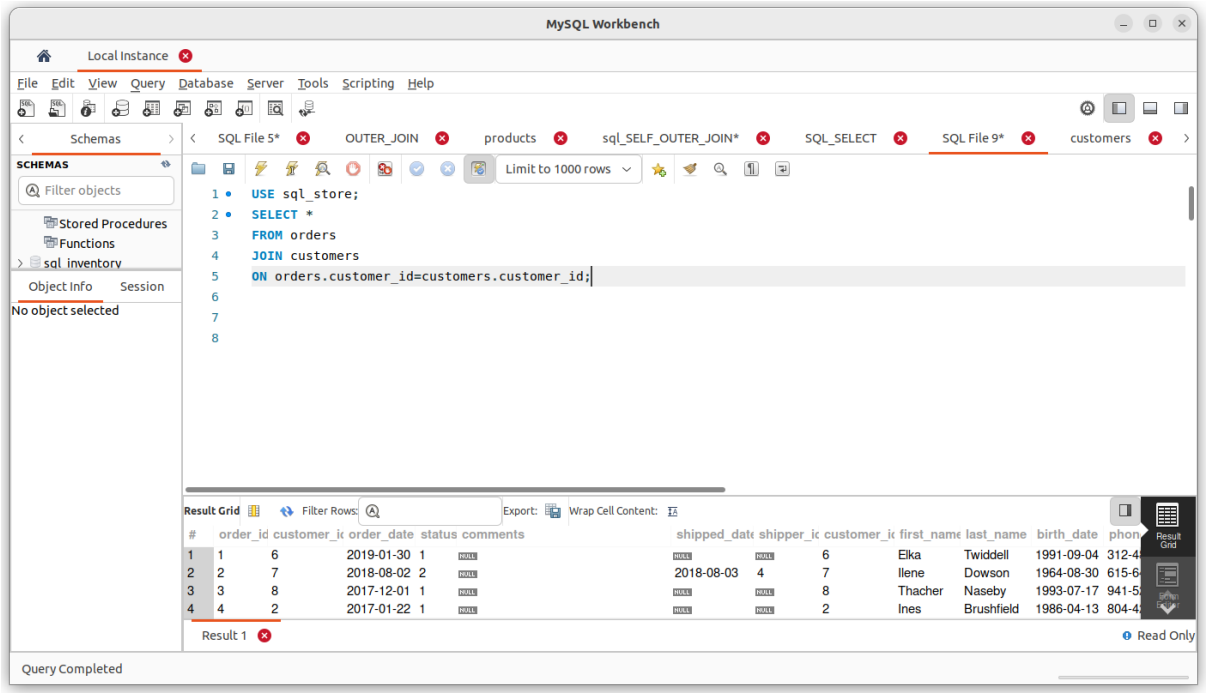
Sql commands 35
-- get 3 data from between the data using offset
SELECT *
FROM customers
LIMIT 6, 3;

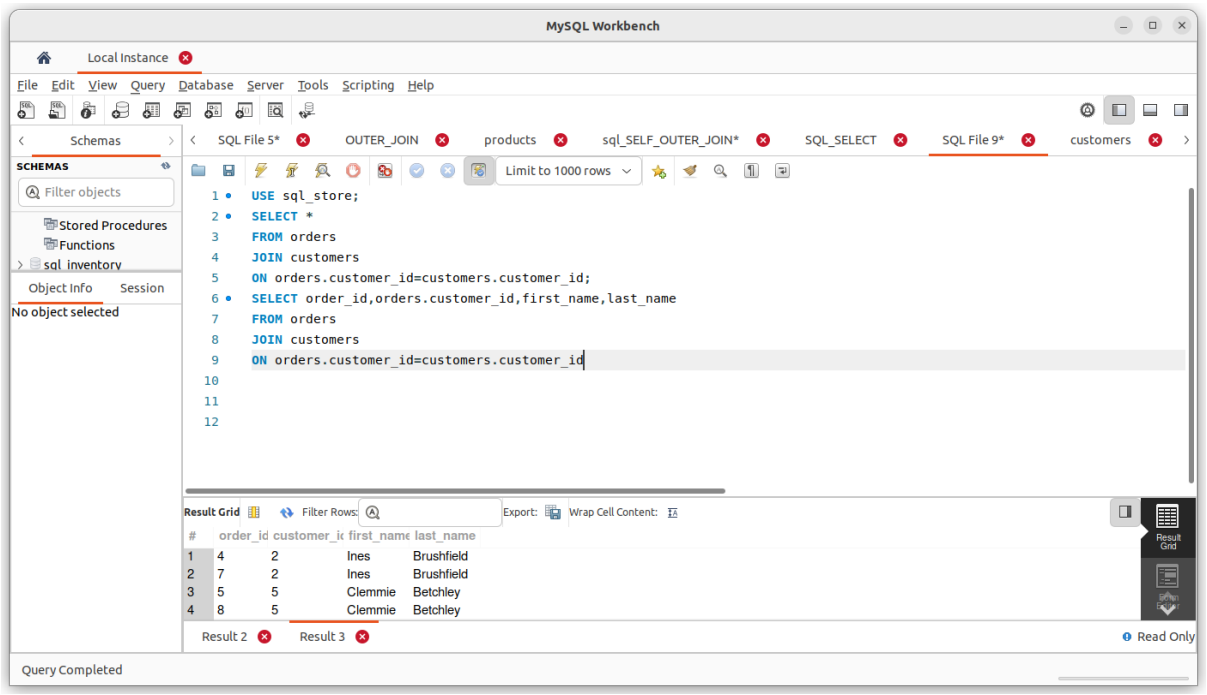

MySQL Workbench
Local Instance
File Edit View Query Database Server Tools Scripting Help
Schemas
SCHEMAS
Filter objects
sql_hr
sql_inventory
sql_invoicing
sql_store
Tables
customers
order_item_notes
order_items
orders
order_statuses
products
Object Info Session
Table: customers
Columns:
customer_id int AI PK
first_name varchar(50)
last_name varchar(50)
birth_date date
phone varchar(50)
address varchar(50)
city varchar(50)
state char(2)
Query 1 customers SQL File 4* SQL File 5* SQL File 6* products SQL File 7
Limit to 1000 rows
1 • USE sql_store;
2 -- get top 5 data
3 • SELECT *
4 FROM customers
5 LIMIT 5;
6
Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content: Fetch rows:
customer_id first_name last_name birth_date phone address city state points
1 1 Babara MacCaffrey 1986-03-28 781-932-9754 0 Sage Terrace Waltham VA 2273
2 2 Ines Brushfield 1986-04-13 804-427-9456 14187 Commercial Trail Hampton VA 947
3 3 Freddi Boagey 1985-02-07 719-724-7869 251 Springs Junction Colorado Springs CO 2967
4 4 Ambur Roseburgh 1974-04-14 407-231-8017 30 Arapahoe Terrace Orlando FL 457
5 5 Clemmie Betchley 1973-11-07 5 Spohn Circle Arlington TX 3675
customers 28
Apply Revert
Query Completed


MySQL Workbench
Local Instance
File Edit View Query Database Server Tools Scripting Help
Schemas
SCHEMAS
Filter objects
sql_hr
sql_inventory
sql_invoicing
sql_store
Tables
customers
order_item_notes
order_items
orders
order_statuses
products
Object Info Session
Table: customers
Columns:
customer_id int AI PK
first_name varchar(50)
last_name varchar(50)
birth_date date
phone varchar(50)
address varchar(50)
city varchar(50)
state char(2)
Query 1 customers SQL File 4* SQL File 5* SQL File 6* products SQL File 7
Limit to 1000 rows
3 • SELECT *
4 FROM customers
5 LIMIT 5;
6 -- get 7-9 data
7 • SELECT *
8 FROM customers
9 LIMIT 6,3;
Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content: Fetch rows:
customer_id first_name last_name birth_date phone address city state points
1 7 Ilene Dowson 1964-08-30 615-641-4759 50 Lillian Crossing Nashville TN 1672
2 8 Thacher Naseby 1993-07-17 941-527-3977 538 Mosinee Center Sarasota FL 205
3 9 Romola Rumgay 1992-05-23 559-181-3744 3520 Ohio Trail Visalia CA 1486
customers 29
Apply Revert
Query Completed

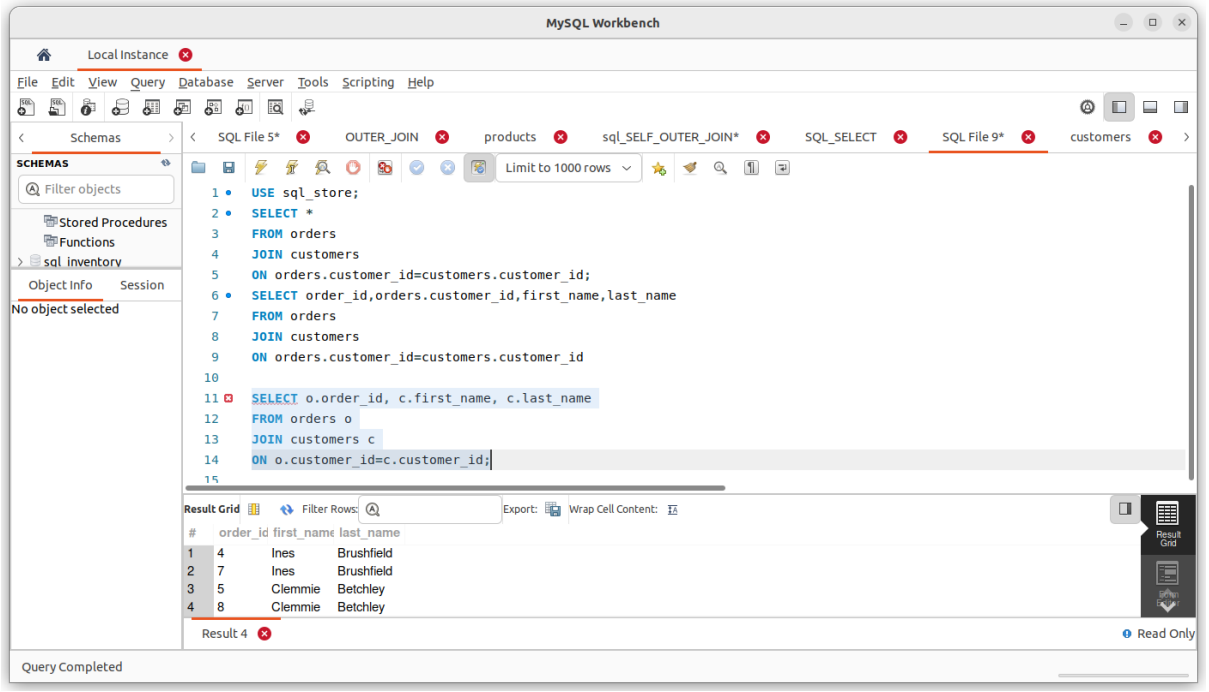
INNER JOIN:

```
-- inner join to join 2 tables
SELECT *
FROM orders o
JOIN customers c
ON o.customer_id=c.customer_id;
```

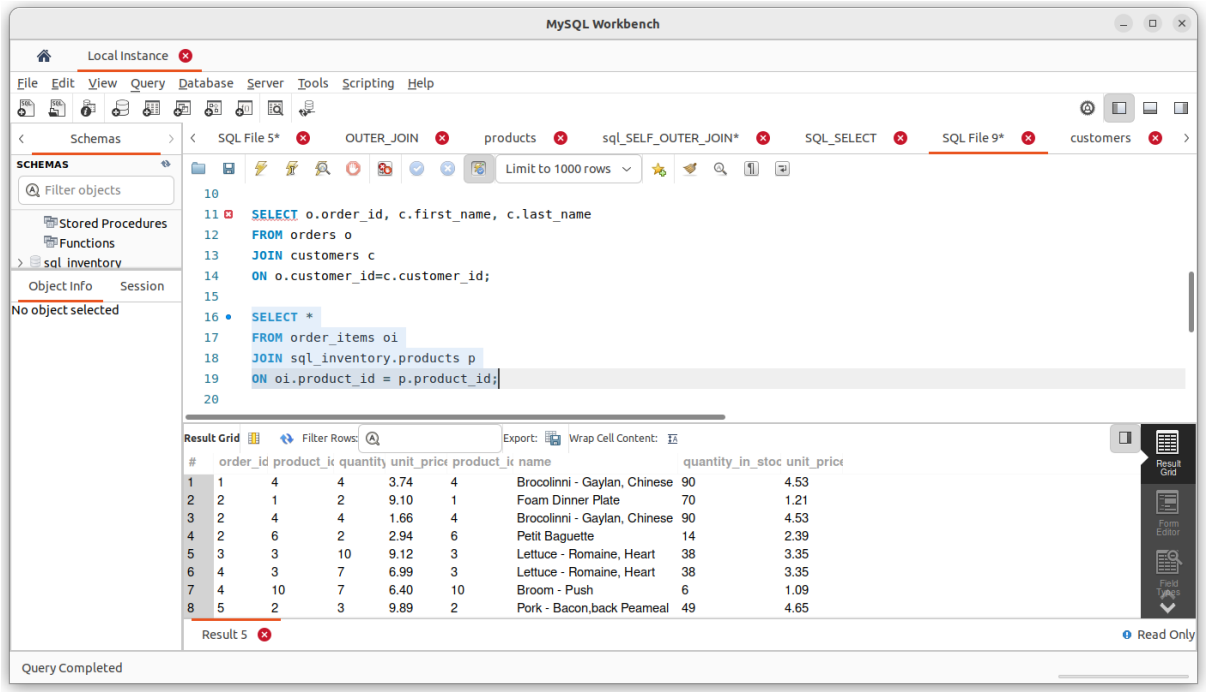




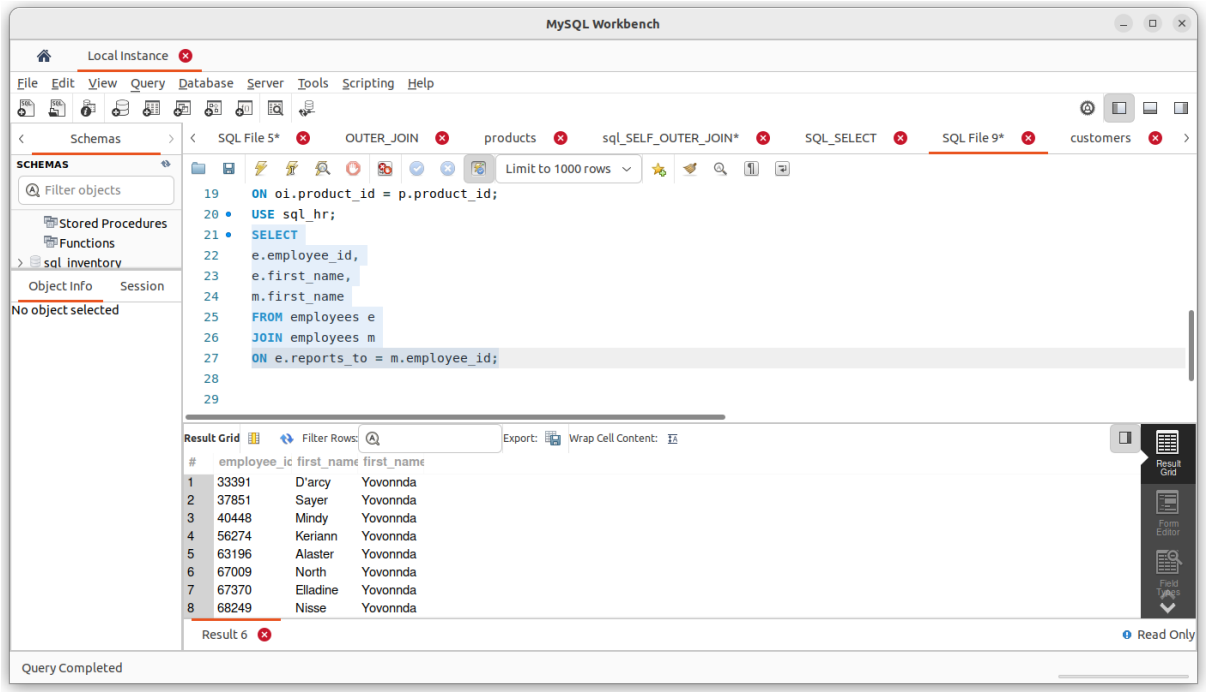
```
SELECT o.order_id, c.first_name, c.last_name
FROM orders o
JOIN customers c
ON o.customer_id=c.customer_id;
```



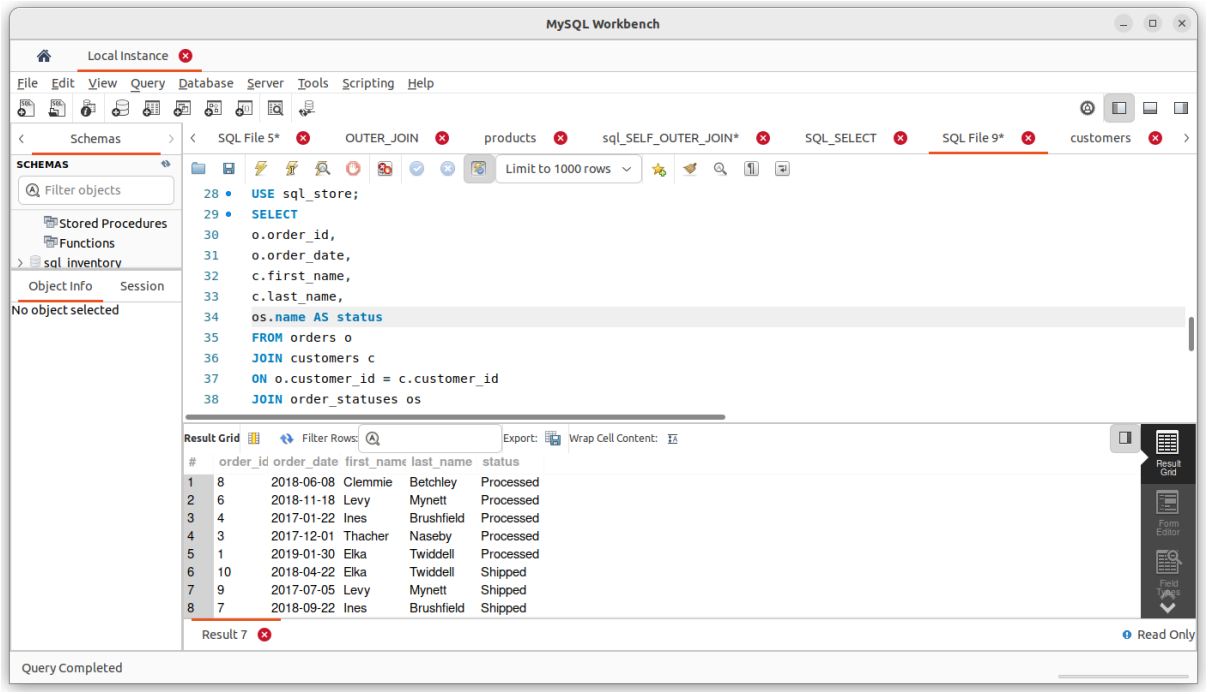

```
SELECT *
FROM orders o, customers c
WHERE o.customer_id = c.customer_id;
-- joining across databases
SELECT *
FROM order_items oi
JOIN sql_inventory.products p
ON oi.product_id = p.product_id;
```




```
-- self join
USE sql_hr;
SELECT
e.employee_id,
e.first_name,
m.first_name
FROM employees e
JOIN employees m
ON e.reports_to = m.employee_id;
```




```
-- joining multiple tables
USE sql_store;
SELECT
o.order_id,
o.order_date,
c.first_name,
c.last_name,
FROM orders o
JOIN customers c
ON o.customer_id = c.customer_id
JOIN order_statuses os
ON o.status = os.order_status_id;
```

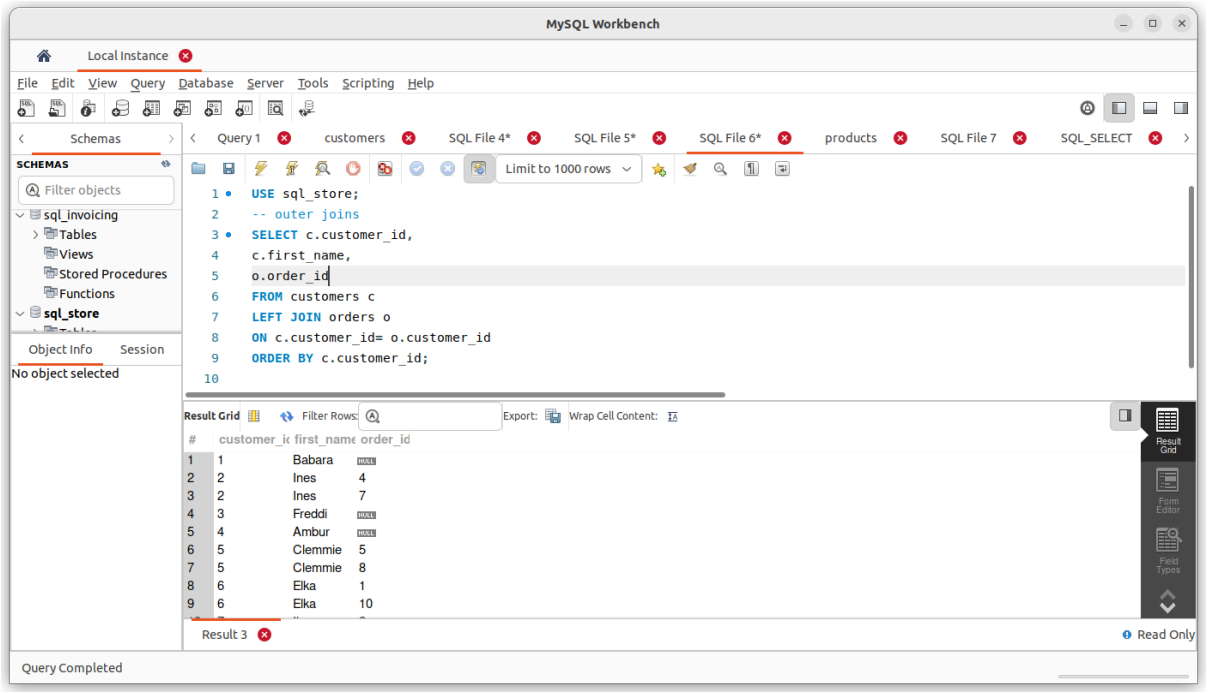


OUTER JOIN=

There are two types of outer joun

1- LEFT JOIN-

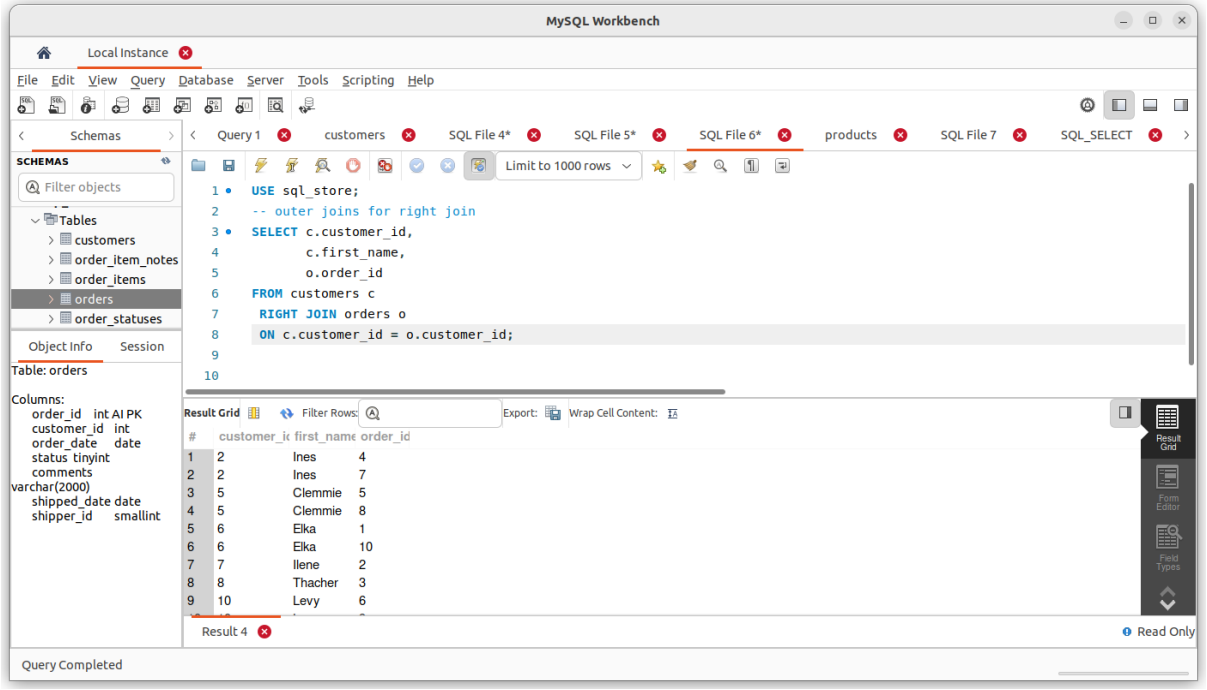
```
-- outer joins for left join
SELECT c.customer_id,
c.first_name,
o.order_id
FROM customers c
LEFT JOIN orders o
ON c.customer_id= o.customer_id
ORDER BY c.customer_id;
```



2-RIGHT JOIN-

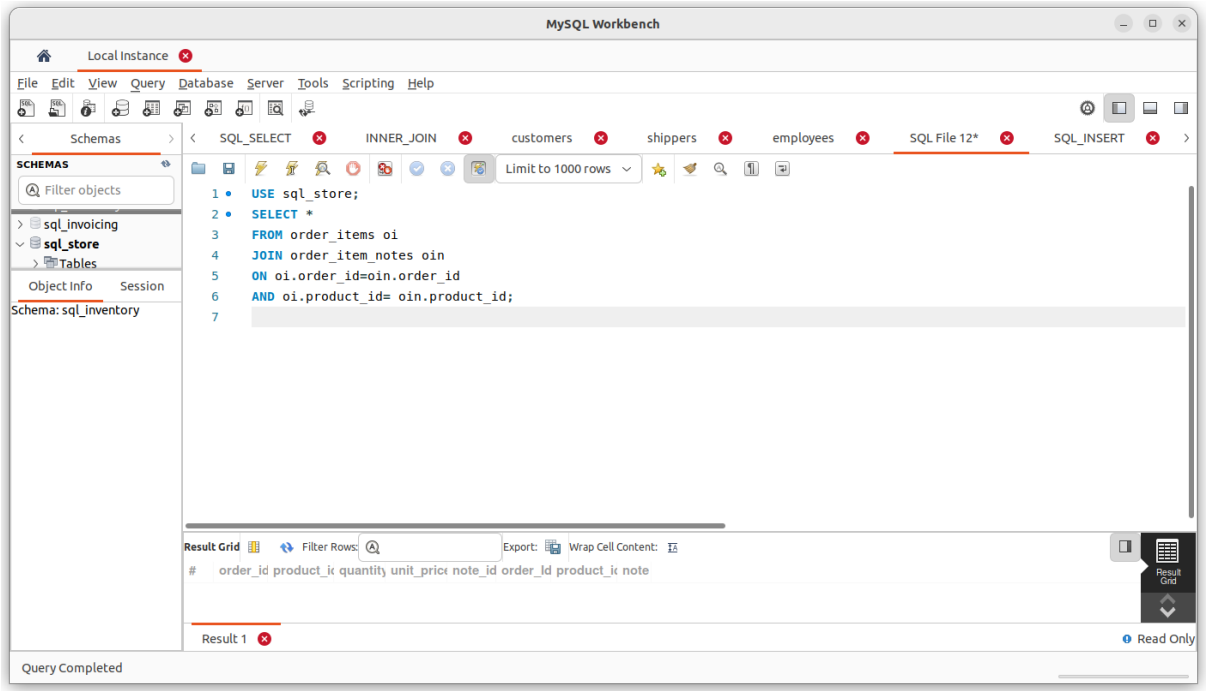
```
-- outer joins for right join
SELECT c.customer_id,
        c.first_name,
        o.order_id
FROM customers c
 RIGHT JOIN orders o
```

ON c.customer_id = o.customer_id;



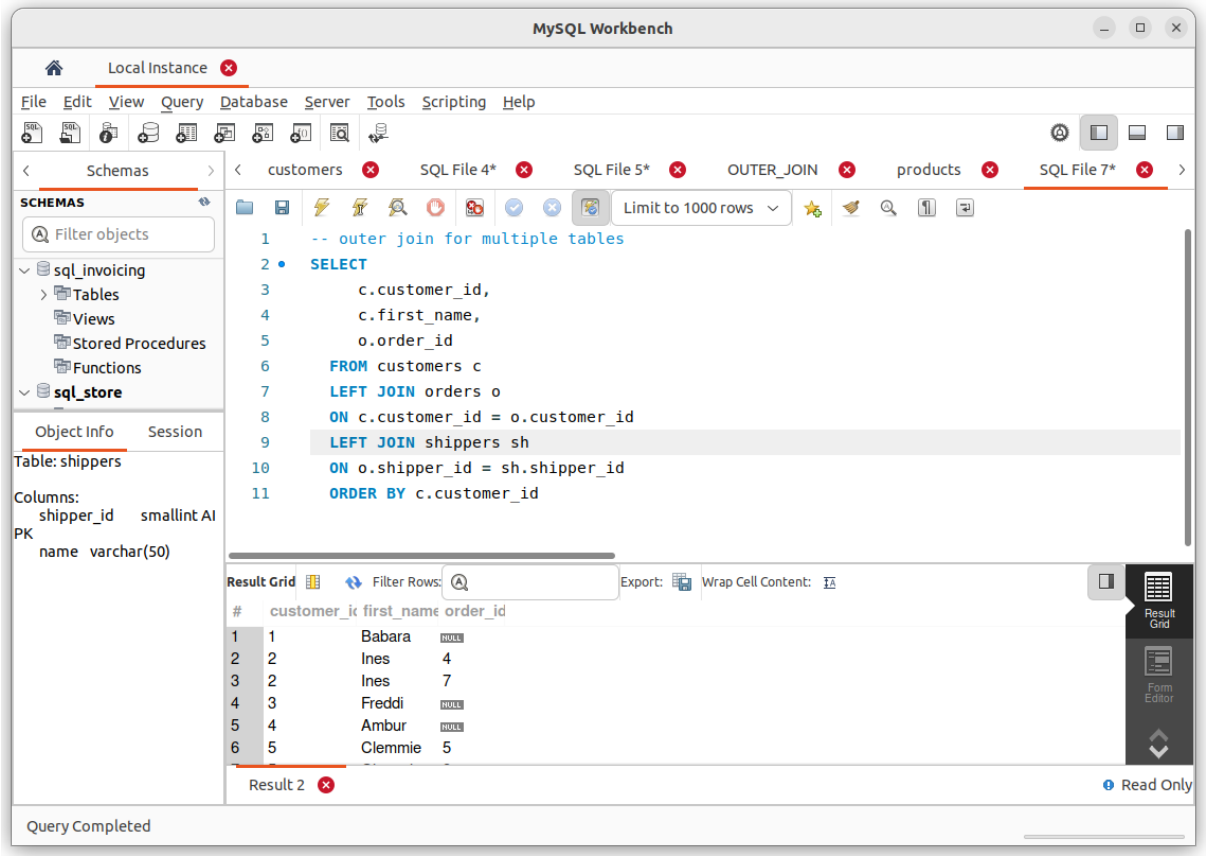
COMPOUND JOIN CONDITIONS-

```
SELECT *
FROM order_items oi
JOIN order_items_notes oin
ON oi.order_id=oin.order_id
AND oi.product_id= oin.product_id;
```



OUTER JOINS BETWEEN MULTIPLE TABLES-

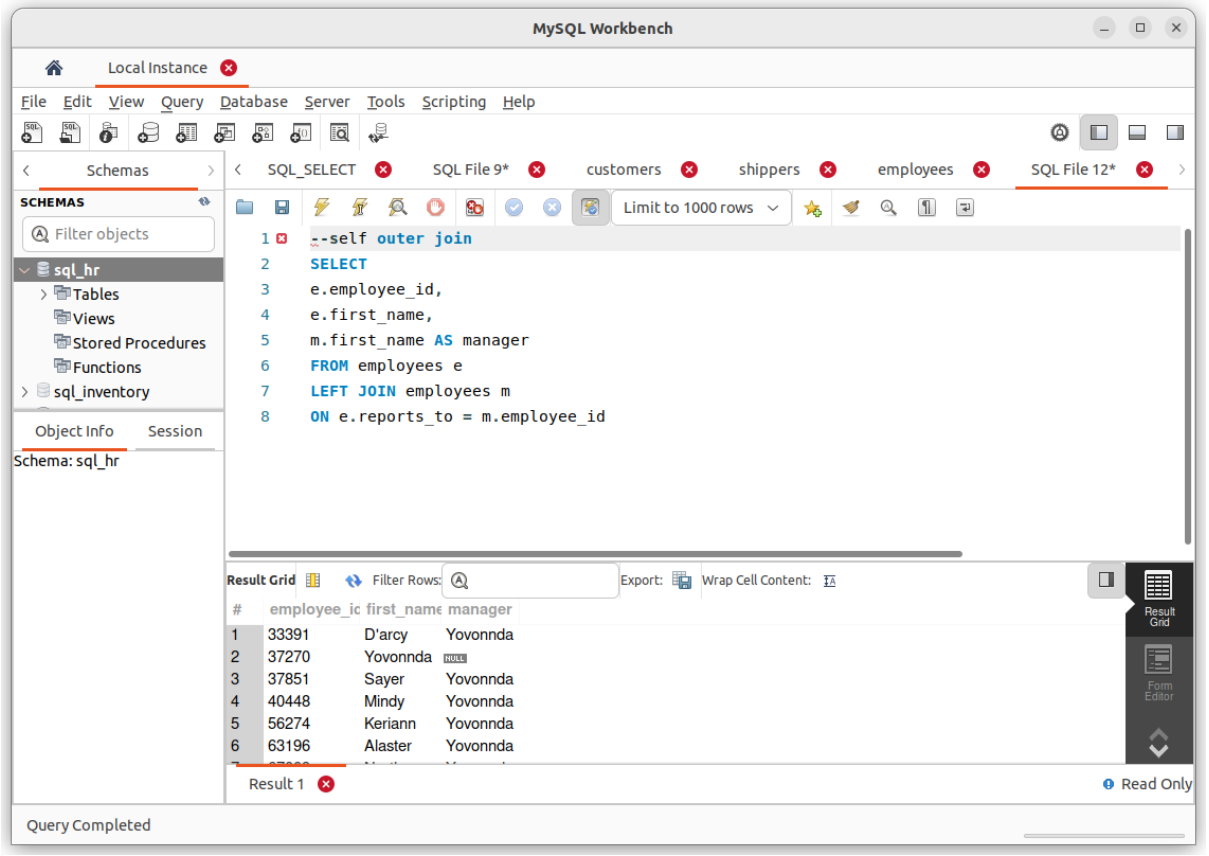
```
-- outer join for multiple tables
SELECT
        c.customer_id,
        c.first_name,
        o.order_id
  FROM customers c
  LEFT JOIN orders o
  ON c.customer_id = o.customer_id
  JOIN shippers sh
  ON o.shipper_id = sh.shipper_id
  ORDER BY c.customer_id
```



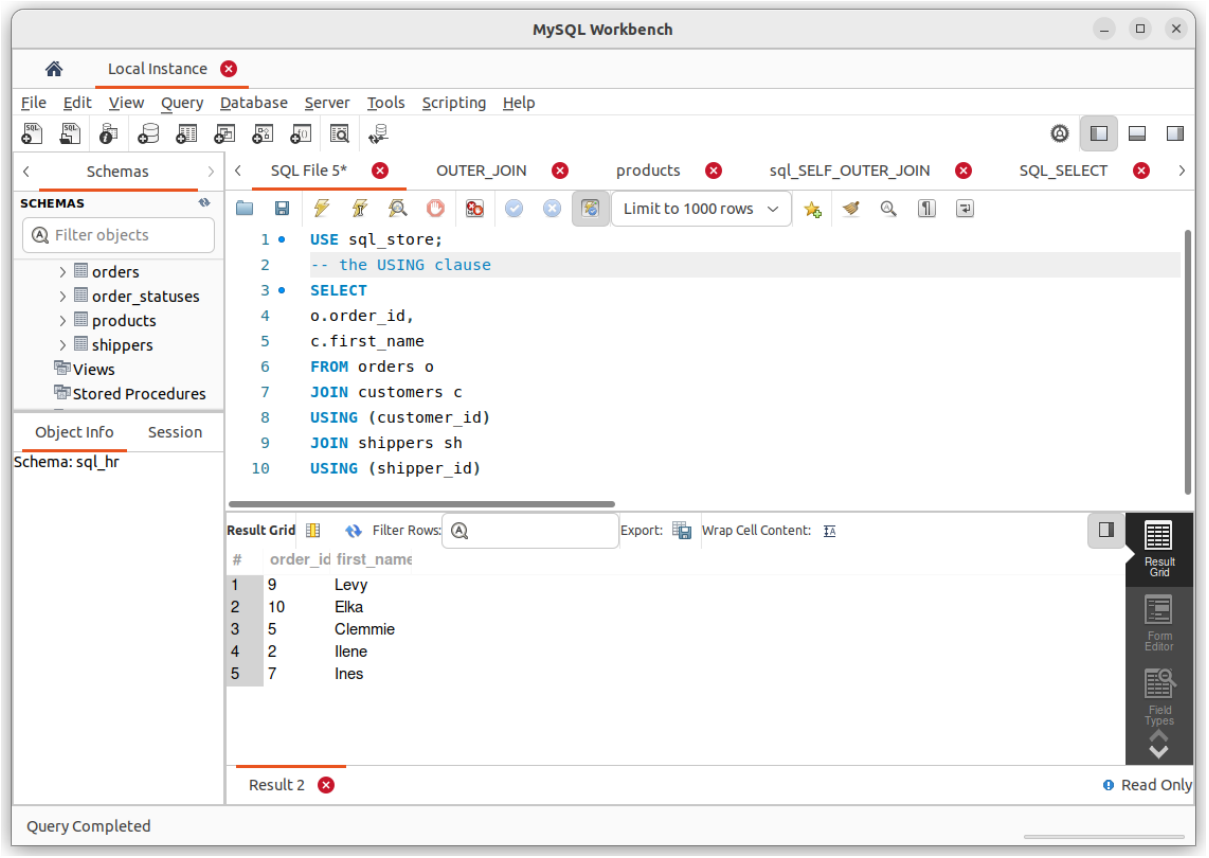
SELF OUTER JOIN-

```
--self outer join
SELECT
e.employee_id,
e.first_name,
```

m.first_name AS manager
FROM employees e
LEFT JOIN employees m
ON e.reports_to = m.employee_id

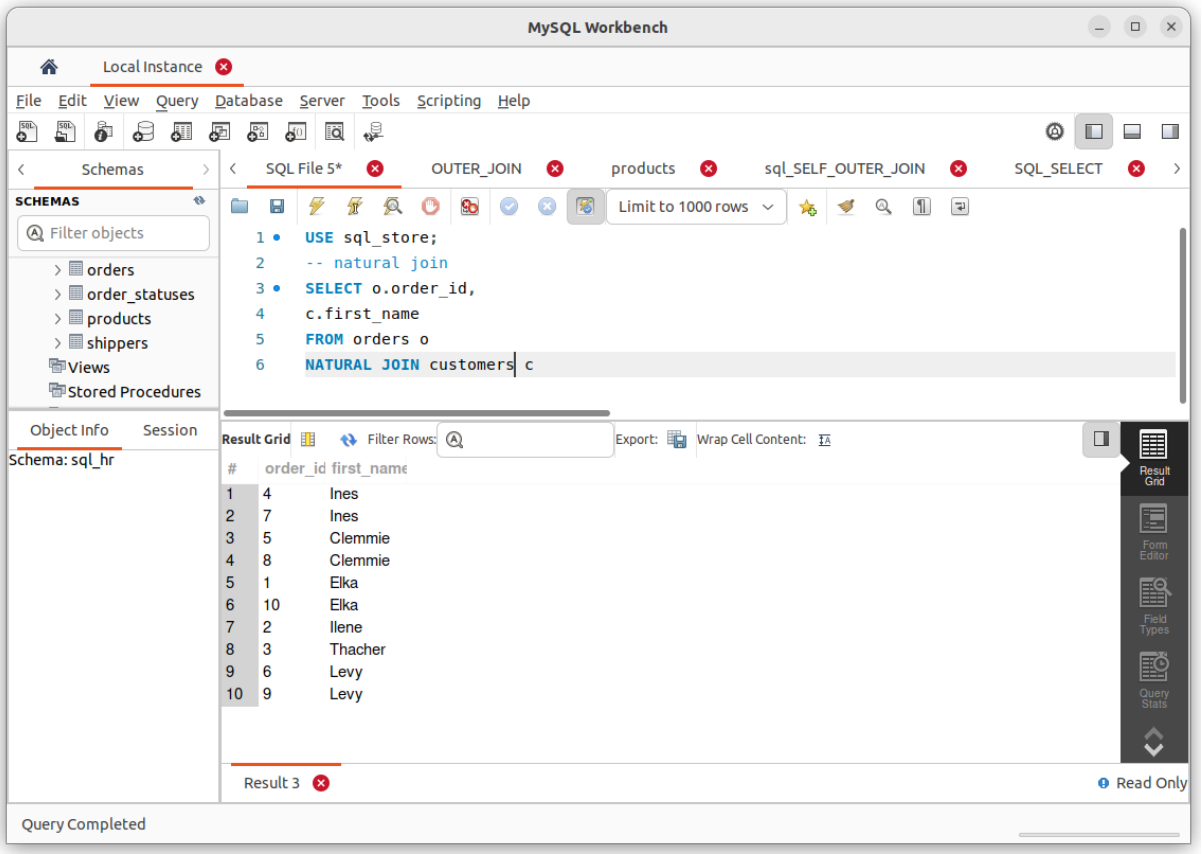


THE USING CLAUSE-
the USING clause
SELECT
o.order_id,
c.first_name
FROM orders o
JOIN customers c
USING (customer_id)
JOIN shippers sh
USING (shipper_id)



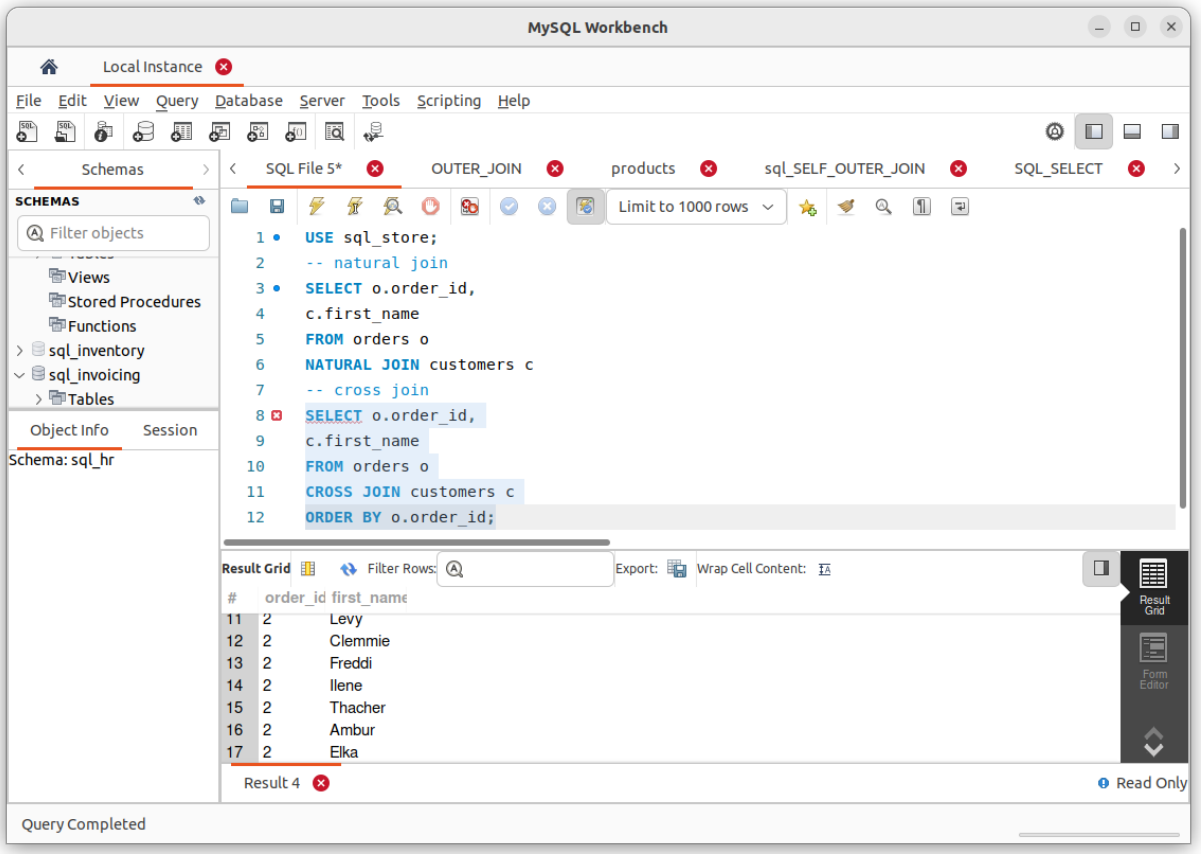
NATURAL JOIN-

```
-- natural join
SELECT o.order_id,
c.first_name
FROM orders o
NATURAL JOIN customers c
```



CROSS JOIN-

```
SELECT o.order_id,
c.first_name
FROM orders o
CROSS JOIN customers c
ORDER BY o.order_id;
```



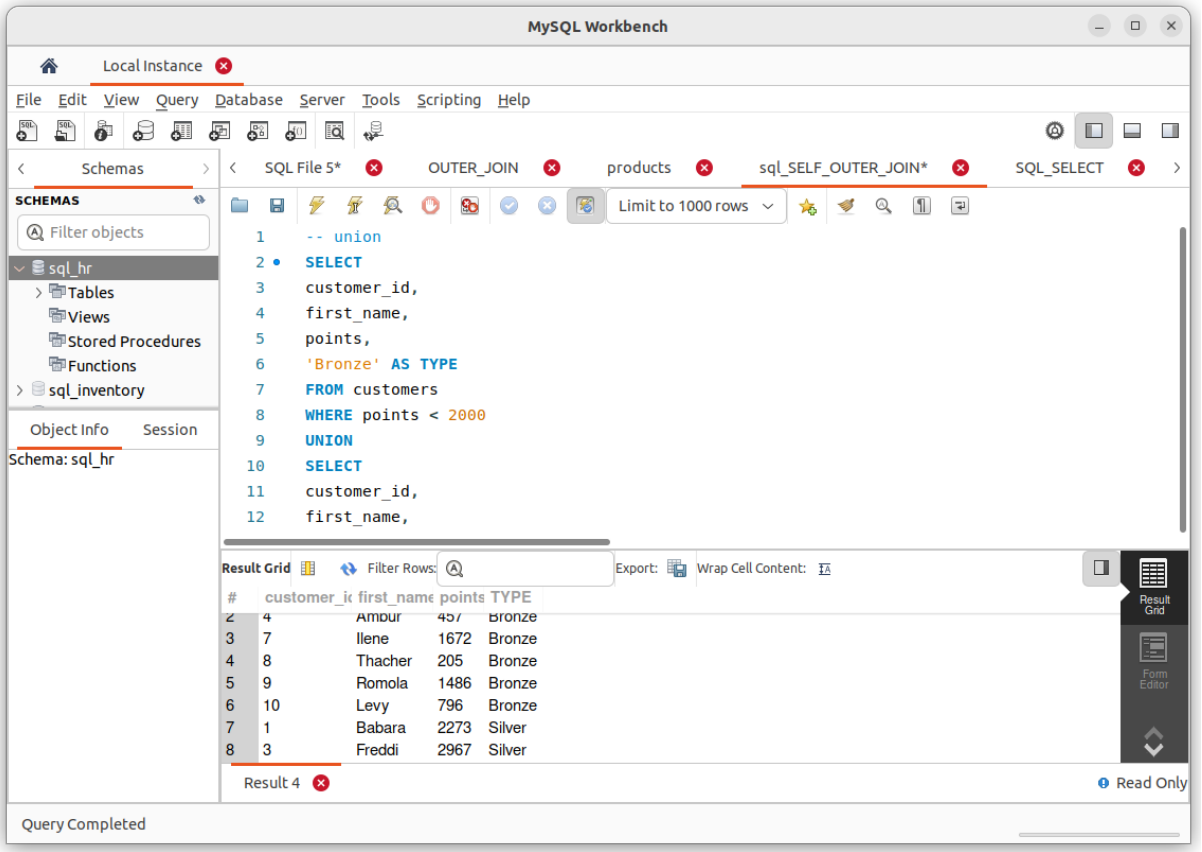
UNIONS-

The `UNION` operator is used to combine the result-set of two or more `SELECT` statements.

- Every `SELECT` statement within `UNION` must have the same number of columns
- The columns must also have similar data types
- The columns in every `SELECT` statement must also be in the same order

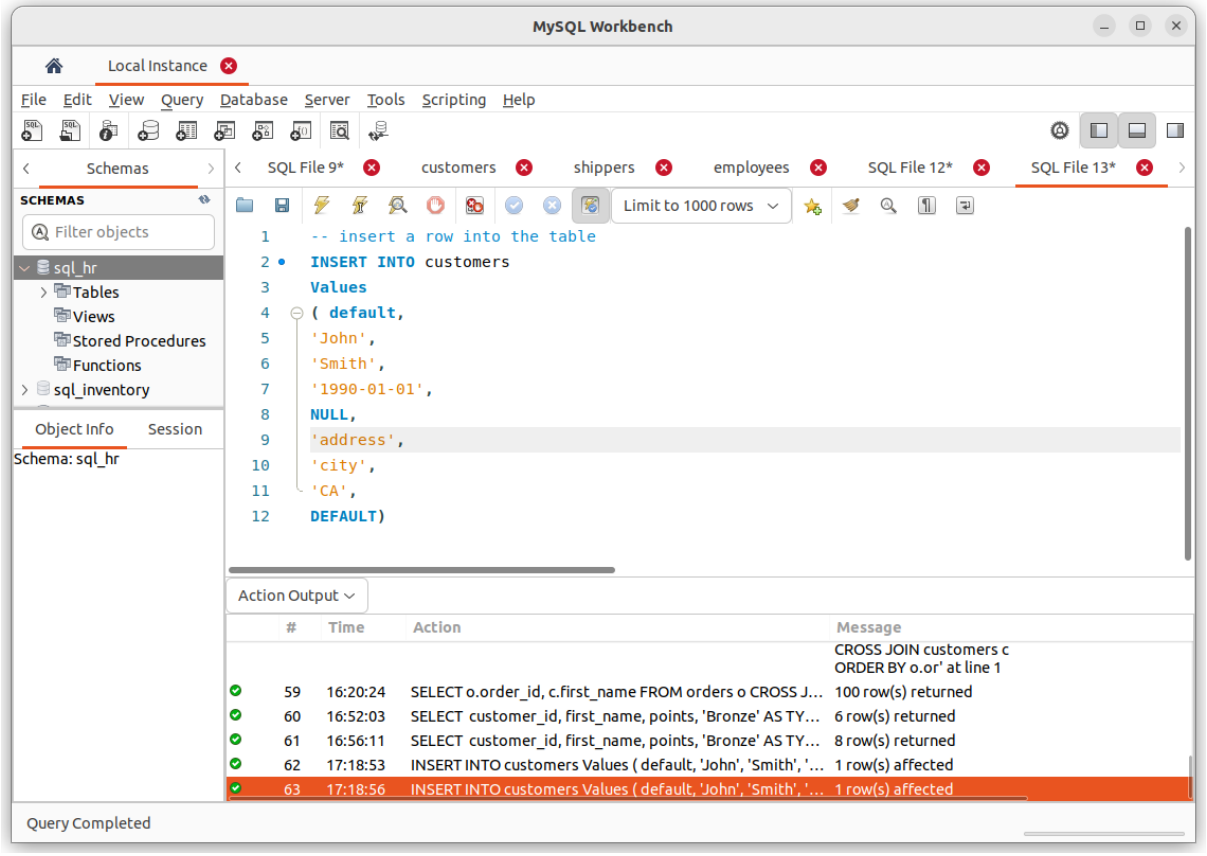
```
-- union

SELECT

customer_id,

first_name,

points,

'Bronze' AS TYPE
```

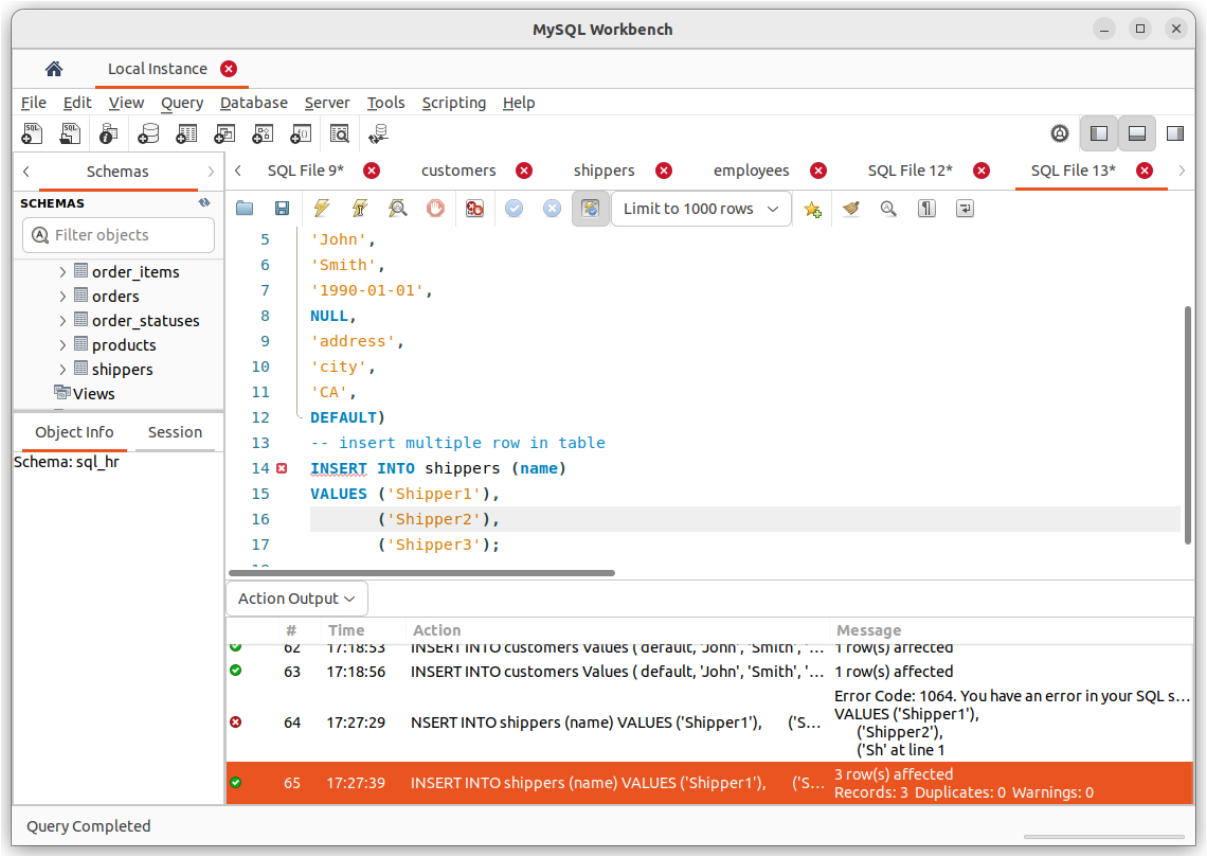
```
FROM customers
WHERE points < 2000
UNION
SELECT
customer_id,
first_name,
points,
'Silver' AS TYPE
FROM customers
WHERE points BETWEEN 2000 AND 3000
```



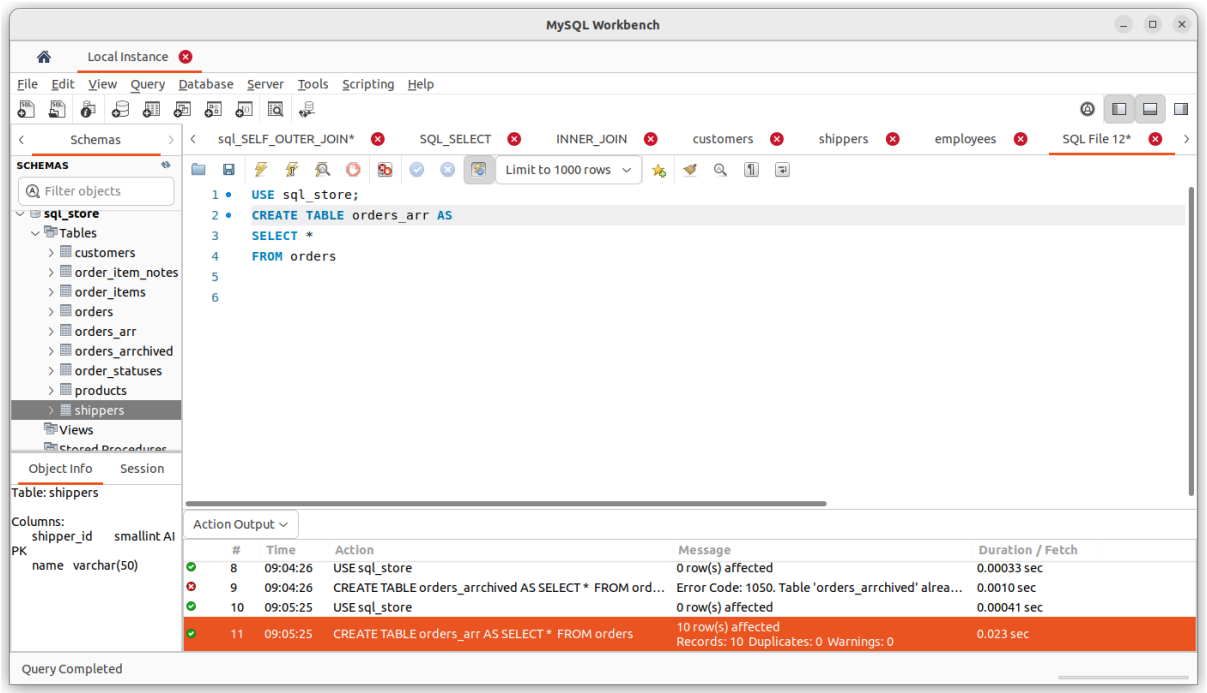
INSERT INTO-The `INSERT INTO` statement is used to insert new records in a table.

```
-- insert a row into the table
INSERT INTO customers
Values
( default,
'John',
'Smith',
'1990-01-01',
NULL,
'address',
'city',
'CA',
DEFAULT)
-- insert multiple row in table
INSERT INTO shippers (name)
VALUES ('Shipper1'),
        ('Shipper2'),
        ('Shipper3');
```



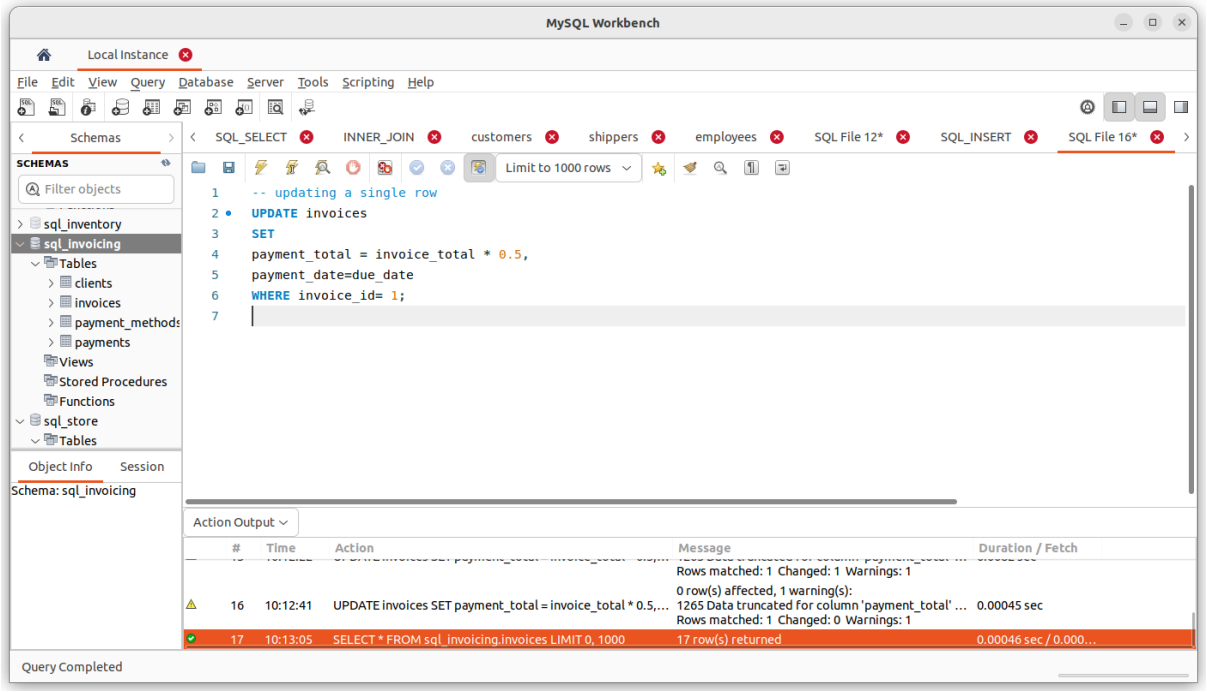


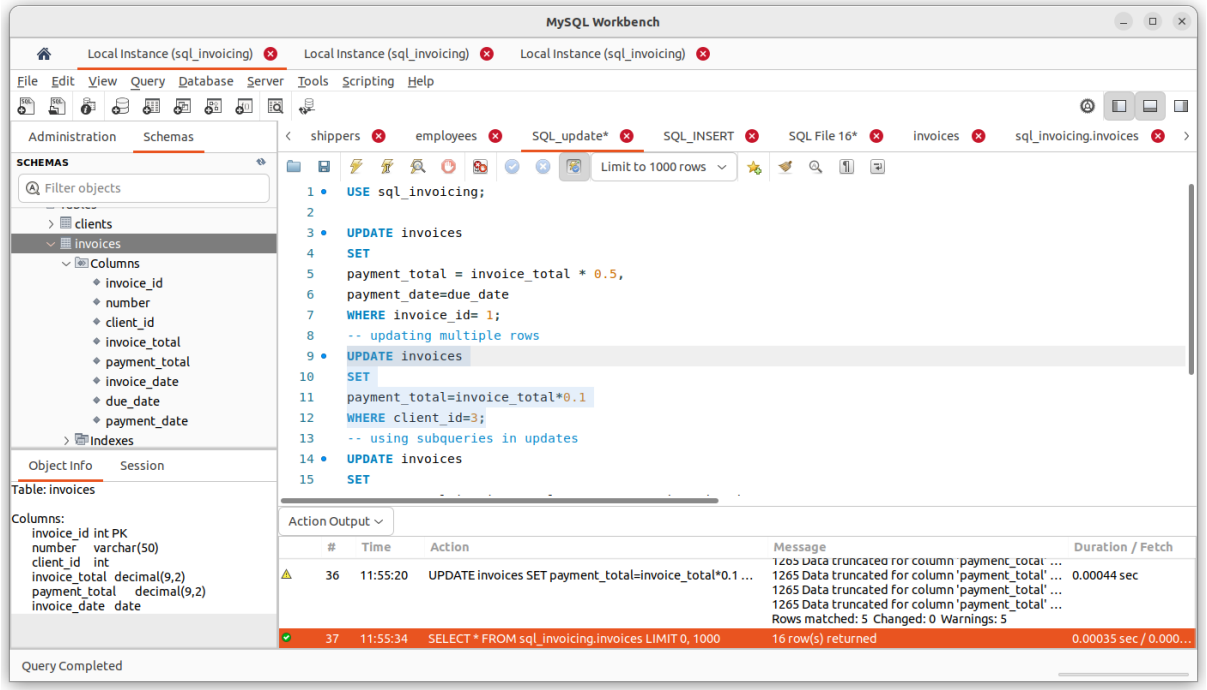
COPY TABLE-
CREATE TABLE orders_arrchived AS
SELECT *
FROM orders

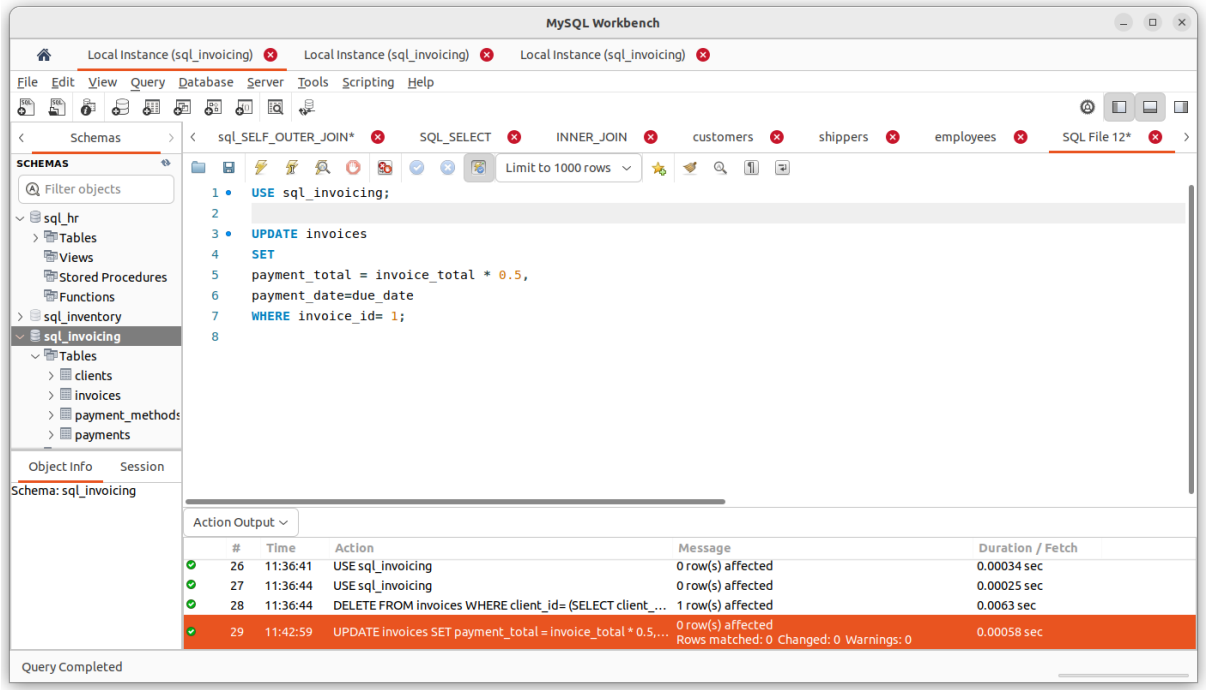


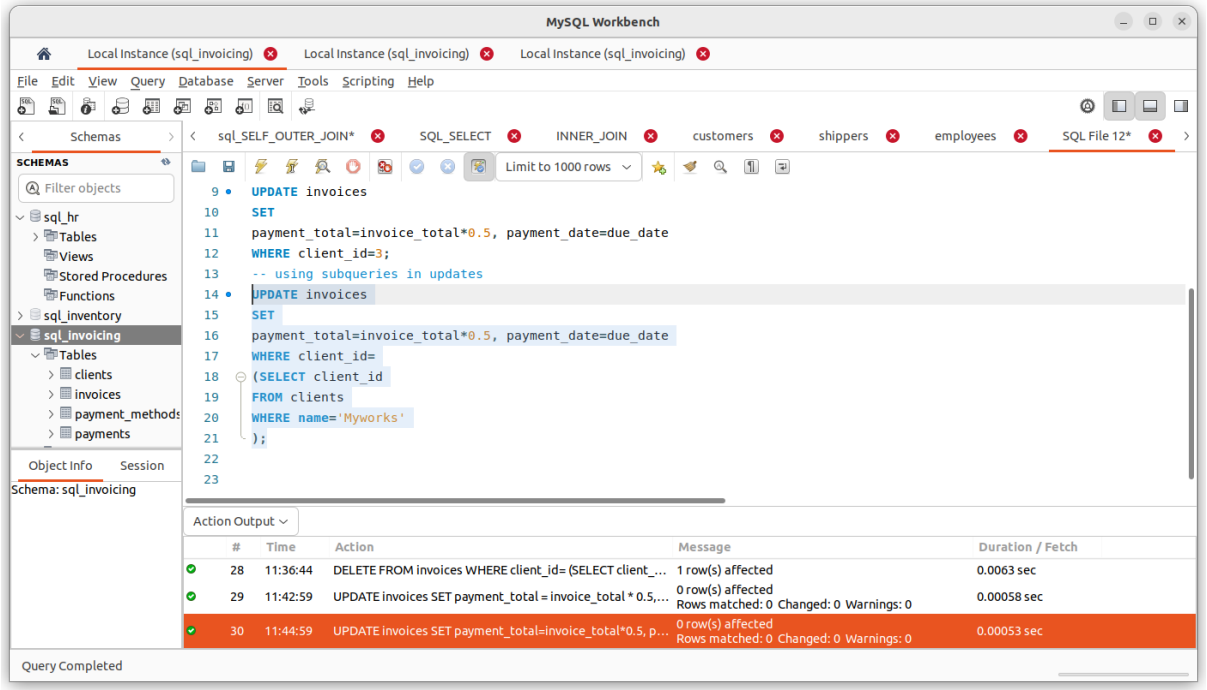
UPDATE-

```
-- updating a single row
UPDATE invoices
SET
payment_total = invoice_total * 0.5,
payment_date=due_date
WHERE invoice_id= 1;
-- updating multiple rows
UPDATE invoices
SET
payment_total=invoice_total*0.5, payment_date=due_date
WHERE client_id=3;
-- using subqueries in updates
UPDATE invoices
SET
payment_total=invoice_total*0.5, payment_date=due_date
WHERE client_id=
(SELECT client_id
FROM clients
WHERE name='Myworks'
);
```



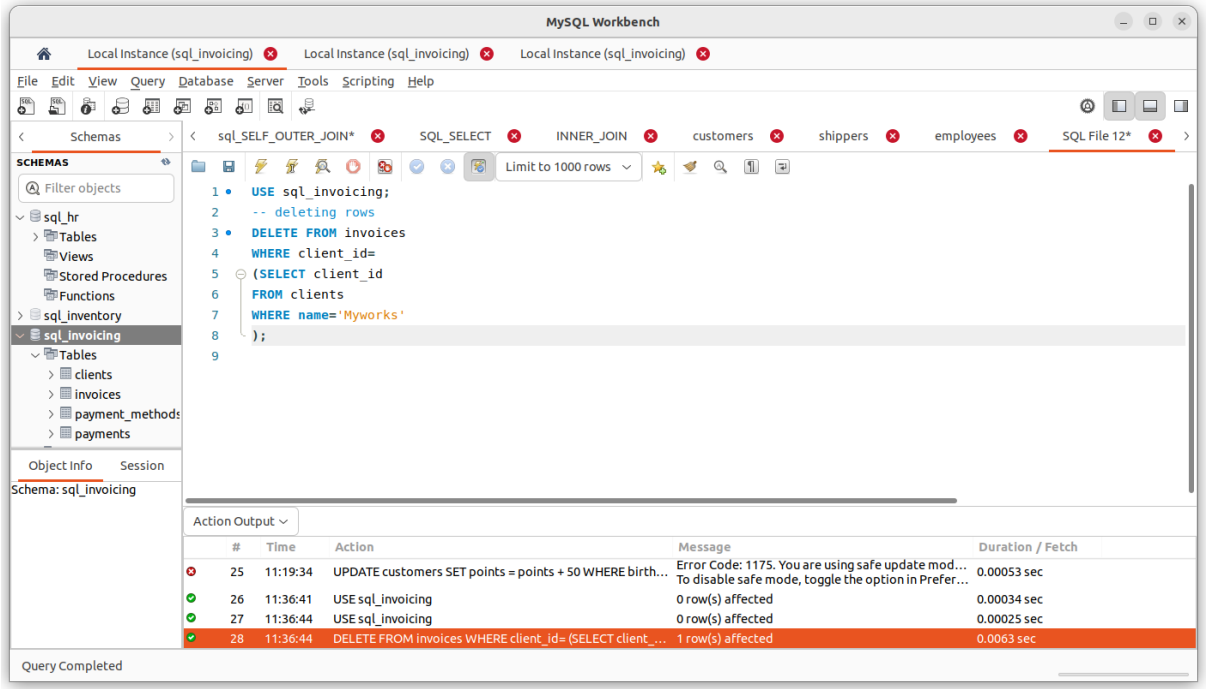

MySQL Workbench
Local Instance (sql_invoicing)
File Edit View Query Database Server Tools Scripting Help
Administration Schemas
SCHEMAS
Filter objects
clients
invoices
Columns
invoice_id
number
client_id
invoice_total
payment_total
invoice_date
due_date
payment_date
Indexes
Object Info Session
Table: invoices
Columns:
invoice_id int PK
number varchar(50)
client_id int
invoice_total decimal(9,2)
payment_total decimal(9,2)
invoice_date date
shippers employees SQL_update* SQL_INSERT SQL File 16* invoices sql_invoicing.invoices
Limit to 1000 rows
1 • USE sql_invoicing;
3 • UPDATE invoices
4 SET
5 payment_total = invoice_total * 0.5,
6 payment_date=due_date
7 WHERE invoice_id= 1;
8 -- updating multiple rows
9 • UPDATE invoices
10 SET
11 payment_total=invoice_total*0.1
12 WHERE client_id=3;
13 -- using subqueries in updates
14 • UPDATE invoices
15 SET
Action Output
Time Action Message Duration / Fetch
36 11:55:20 UPDATE invoices SET payment_total=invoice_total*0.1 ... 1265 Data truncated for column 'payment_total' ... 0.00044 sec
1265 Data truncated for column 'payment_total' ...
1265 Data truncated for column 'payment_total' ...
Rows matched: 5 Changed: 0 Warnings: 5
37 11:55:34 SELECT * FROM sql_invoicing.invoices LIMIT 0, 1000 16 row(s) returned 0.00035 sec / 0.000...
Query Completed


MySQL Workbench
Local Instance (sql_invoicing)
File Edit View Query Database Server Tools Scripting Help
Schemas
SCHEMAS
Filter objects
sql_hr
Tables
Views
Stored Procedures
Functions
sql_inventory
sql_invoicing
Tables
clients
invoices
payment_methods
payments
Object Info Session
Schema: sql_invoicing
sql_SELF_OUTER_JOIN* SQL_SELECT INNER_JOIN customers shippers employees SQL File 12*
Limit to 1000 rows
1 • USE sql_invoicing;
3 • UPDATE invoices
4 SET
5 payment_total = invoice_total * 0.5,
6 payment_date=due_date
7 WHERE invoice_id= 1;
Action Output
Time Action Message Duration / Fetch
26 11:36:41 USE sql_invoicing 0 row(s) affected 0.00034 sec
27 11:36:44 USE sql_invoicing 0 row(s) affected 0.00025 sec
28 11:36:44 DELETE FROM invoices WHERE client_id= (SELECT client_... 1 row(s) affected 0.0063 sec
29 11:42:59 UPDATE invoices SET payment_total = invoice_total * 0.5,... 0 row(s) affected
Rows matched: 0 Changed: 0 Warnings: 0 0.00058 sec
Query Completed



DELETE-

```
USE sql_invoicing;
-- deleting rows
DELETE FROM invoices
WHERE client_id=
(SELECT client_id
FROM clients
WHERE name='Myworks'
);
```


MySQL Workbench
Local Instance (sql_invoicing)
Local Instance (sql_invoicing)
Local Instance (sql_invoicing)
File Edit View Query Database Server Tools Scripting Help
Schemas
SCHEMAS
Filter objects
sql_hr
Tables
Views
Stored Procedures
Functions
sql_inventory
sql_invoicing
Tables
clients
invoices
payment_methods
payments
Object Info
Session
Schema: sql_invoicing
sql_SELF_OUTER_JOIN*
SQL_SELECT
INNER_JOIN
customers
shippers
employees
SQL File 12*
Limit to 1000 rows
1 USE sql_invoicing;
2 -- deleting rows
3 DELETE FROM invoices
4 WHERE client_id=
5 (SELECT client_id
6 FROM clients
7 WHERE name='Myworks'
8);
9
Action Output
Time Action Message Duration / Fetch
25 11:19:34 UPDATE customers SET points = points + 50 WHERE birth... Error Code: 1175. You are using safe update mod... To disable safe mode, toggle the option in Prefer... 0.00053 sec
26 11:36:41 USE sql_invoicing 0 row(s) affected 0.00034 sec
27 11:36:44 USE sql_invoicing 0 row(s) affected 0.00025 sec
28 11:36:44 DELETE FROM invoices WHERE client_id= (SELECT client_... 1 row(s) affected 0.0063 sec
Query Completed