Basics of SQL

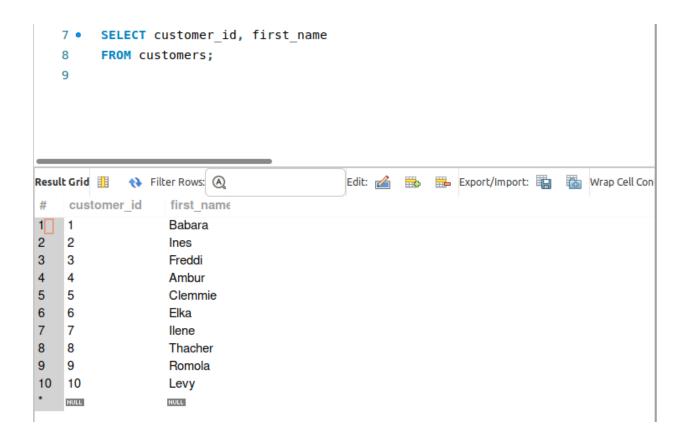
SQL is a standard language for storing, manipulating and retrieving data in databases.

1) SQL - SELECT:

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

```
1 • USE sql_store;
2  -- select
3 • SELECT * FROM customers
4  where customer_id = 1
5  order by first_name
6
7
8
```





For removing duplicates use "DISTINCT"

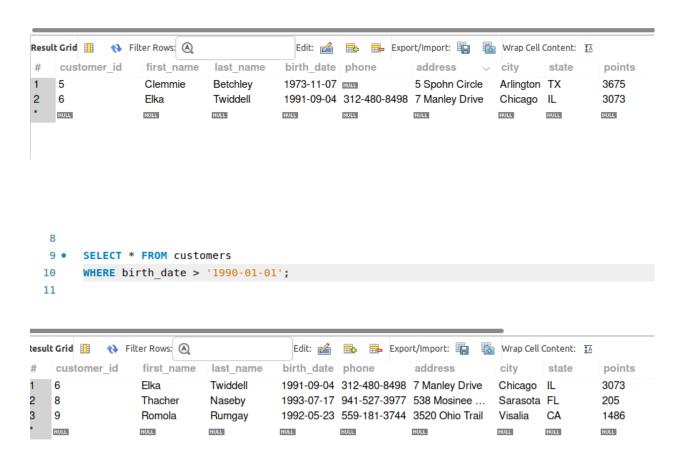
SELECT DISTINCT state

FROM customers

2) SQL - WHERE

The WHERE clause is used to filter records. it extract only those records that fulfill a specified condition.

```
3 • SELECT * FROM customers
4 WHERE points > 3000;
5
```



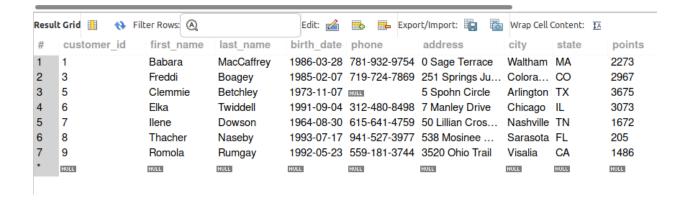
3) SQL - Logical Operators (AND, OR and NOT)

- The WHERE clause can be used in combination with the AND, OR, and NOT operators to filter records based on specific conditions.
- AND operator: a record will be displayed only if all the conditions connected by AND are true.

 OR operator: a record will be displayed if at least one of the conditions connected by OR is true.

```
SELECT * FROM customers
          WHERE birth_date > '1990-01-01' AND points > 1000;
Result Grid 🎚
              Name of the Filter Rows:
                                                  Edit: 🔏
                                                                Export/Import:
                                                                                     Wrap Cell Content: IA
     customer_id
                     first_name
                                  last_name
                                                birth_date phone
                                                                         address
                                                                                                  state
                                                                                                            points
                                                                                         city
                                                1991-09-04 312-480-8498 7 Manley Drive
1
     6
                                                                                                            3073
                     Elka
                                  Twiddell
                                                                                        Chicago IL
2
     9
                     Romola
                                                1992-05-23 559-181-3744 3520 Ohio Trail
                                                                                                            1486
                                  Rumgay
                                                                                        Visalia
                                                                                                  CA
                                  NULL
    NULL
                                                                                                 NULL
                                                                                                            NULL
```

```
5
6 • SELECT * FROM customers
7 WHERE birth_date > '1990-01-01' OR points > 1000;
```

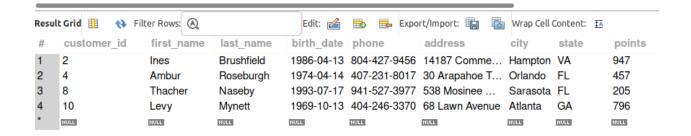


```
11
   12 •
          SELECT * FROM customers
          WHERE NOT birth date > '1990-01-01';
   13
Result Grid 🏭
               Filter Rows: (A)
                                                  Edit: 🕍
                                                                Export/Import:
                                                                                     Wrap Cell Content: 🔣
     customer id
                      first name
                                                birth date phone
                                                                         address
                                                                                         city
                                  last_name
                                                                                                  state
                                                                                                            points
1
     1
                     Babara
                                  MacCaffrey
                                                1986-03-28 781-932-9754 0 Sage Terrace
                                                                                        Waltham MA
                                                                                                            2273
2
     2
                     Ines
                                  Brushfield
                                                1986-04-13 804-427-9456 14187 Comme... Hampton VA
                                                                                                            947
3
     3
                                                                                                            2967
                     Freddi
                                  Boagey
                                                1985-02-07 719-724-7869 251 Springs Ju... Colora... CO
4
                     Ambur
                                  Roseburgh
                                                1974-04-14 407-231-8017 30 Arapahoe T... Orlando
                                                                                                            457
                     Clemmie
5
     5
                                  Betchley
                                                1973-11-07 NULL
                                                                        5 Spohn Circle
                                                                                        Arlington TX
                                                                                                            3675
                                                                                        Nashville
6
                                  Dowson
                                                1964-08-30 615-641-4759 50 Lillian Cros...
                                                                                                            1672
                     llene
                                                                                                            796
     10
                     Levy
                                  Mynett
                                                1969-10-13 404-246-3370 68 Lawn Avenue Atlanta
     NULL
                     NULL
                                  NULL
                                                           NULL
                                                                        NULL
                                                                                        NULL
                                                                                                 NULL
                                                                                                            NULL
```

4) SQL - IN

The IN is a shorthand for multiple OR conditions.

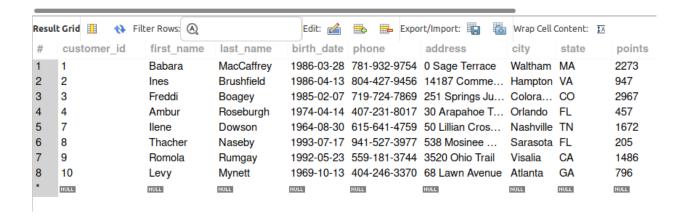
```
2 • SELECT * FROM customers
3 where state IN ('VA','FL','GA');
4
5
6
7
```



5) SQL - BETWEEN

The BETWEEN operator selects values within a given range. This operator is inclusive.

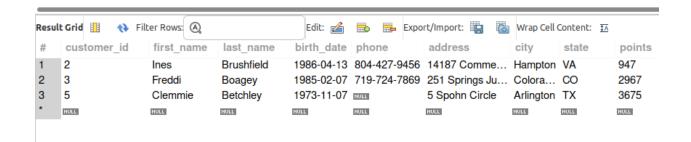
```
2
3 • SELECT * FROM customers
4 WHERE points BETWEEN 100 AND 3000;
5
```



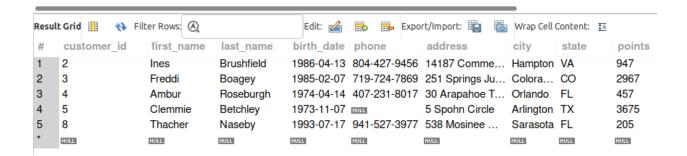
6) SQL - LIKE

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

```
2
3 • SELECT * FROM customers
4 WHERE last_name LIKE 'b%';
5
6
```



```
6 • SELECT * FROM customers
7 WHERE last_name LIKE '%b%';
8
9
```



```
9 •
          SELECT * FROM customers
  10
          WHERE last name LIKE '%Y';
  11
  12
              🙌 Filter Rows: 📵
Result Grid
                                                 Edit: 👍
                                                          Export/Import:
                                                                                        Wrap Cell Content: 🔢
     customer id
                     first name
                                  last name
                                               birth date phone
                                                                                                           points
                     Babara
                                  MacCaffrey
                                               1986-03-28 781-932-9754 0 Sage Terrace
                                                                                       Waltham MA
                                                                                                           2273
2
    3
                     Freddi
                                               1985-02-07 719-724-7869 251 Springs Ju... Colora... CO
                                                                                                           2967
                                  Boagey
3
                     Clemmie
    5
                                  Betchley
                                               1973-11-07 NULL
                                                                        5 Spohn Circle
                                                                                       Arlington TX
                                                                                                           3675
    8
                     Thacher
                                  Naseby
                                               1993-07-17 941-527-3977 538 Mosinee ...
                                                                                        Sarasota FL
                                                                                                           205
5
                                  Rumgay
                                               1992-05-23 559-181-3744 3520 Ohio Trail
                                                                                                           1486
    9
                     Romola
                                                                                       Visalia
                                                                                                CA
                     NULL
                                  NULL
                                                          NULL
                                                                       NULL
                                                                                       NULL
                                                                                                NULL
                                                                                                           NULL
```

7) SQL - REGEXP

REGEXP is similar to LIKE

```
3 • SELECT * FROM customers
4 WHERE last_name REGEXP 'field';
5
```



```
SELECT * FROM customers
   7
         WHERE last_name REGEXP 'field$';
   8
   9
Result Grid 🎚 🙌 Filter Rows: 📵
                                                        Export/Import:
                                                                                  Wrap Cell Content: 1A
     customer_id
                    first_name
                                              birth_date phone
                                                                     address
                                                                                     city
                                 last_name
                                                                                                        points
                    Ines
                                 Brushfield
                                             1986-04-13 804-427-9456 14187 Comme... Hampton VA
                                                                                                       947
    NULL
                    NULL
                                NULL
                                                        NULL
                                                                     NULL
                                                                                    NULL
                                                                                             NULL
                                                                                                       NULL
   8
   9 •
         SELECT * FROM customers
          WHERE last_name REGEXP '^Brush';
  10
                                               Edit: 🔏
Result Grid 🔢
             🙌 Filter Rows: 📵
                                                             Export/Import:
                                                                                  Wrap Cell Content: IA
     customer id
                    first name
                                              birth date phone
                                 last_name
                                                                      address
                                                                                     city
                                                                                              state
                                                                                                        points
    2
                    Ines
                                 Brushfield
                                              1986-04-13 804-427-9456 14187 Comme... Hampton VA
                                                                                                        947
    NULL
                    NULL
                                NULL
                                                                     NULL
                                                                                                       NULL
```

8) SQL - NULL

The IS NULL command is used in SQL to check if a column is NULL, which represents the absence of a value.

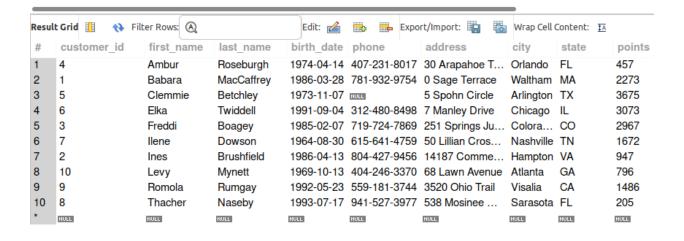
```
2
3 • SELECT * FROM customers
4 WHERE phone IS NULL;
5
```



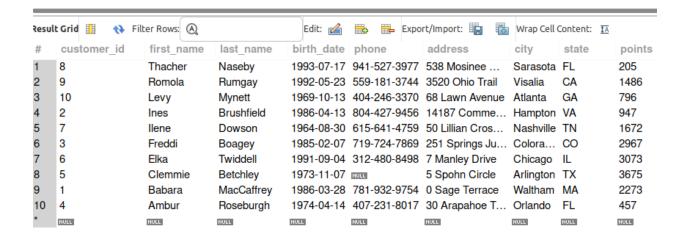
9) SQL - ORDER BY

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

```
2
3 • SELECT * FROM customers
4 ORDER BY first_name;
```



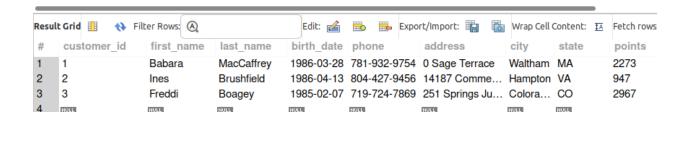
```
6 • SELECT * FROM customers
7 ORDER BY first_name DESC;
```



9) SQL - LIMIT

The LIMIT command is used in SQL to restrict the number of rows returned by a query.

```
3 • SELECT * FROM customers
4 LIMIT 3;
5
```





10) SQL - INNER JOIN

The INNER JOIN command in SQL is used to combine rows from two or more tables based on a related column between them.

```
SELECT first_name, last_name, orders.customer_id FROM orders
          JOIN customers ON orders.customer id = customers.customer id
    4
Result Grid 🎚
                                                 Export: Wrap Cell Content: 1A
              Note: Filter Rows:
     first_name
                  last_name
                               customer_ic
1
     Ines
                 Brushfield
                              2
2
    Ines
                 Brushfield
                              2
3
                              5
    Clemmie
                 Betchley
    Clemmie
                              5
4
                 Betchley
5
    Elka
                 Twiddell
                              6
6
    Elka
                 Twiddell
                              6
7
     llene
                 Dowson
                              7
8
    Thacher
                 Naseby
9
                               10
     Levy
                 Mynett
10
                 Mynett
                               10
    Levy
```

11) SQL - Joining across databases

Joining tables of different databases.

```
3 • SELECT * FROM order_items oi
4    JOIN sql_inventory.products p ON oi.product_id = p.product_id;
5
```

esul	t Grid	🔢 🙌 Fi	lter Rows:	A.		Export: Wrap Cell Content: 12	ī	
#	orde	r_id produ	ct_ic quan	tity unit_p	rice produ	ict_ic name	quantity_	in_stoc unit_pric
1	2	1	2	9.10	1	Foam Dinner Plate	70	1.21
2	6	1	4	8.65	1	Foam Dinner Plate	70	1.21
3	10	1	10	6.01	1	Foam Dinner Plate	70	1.21
4	5	2	3	9.89	2	Pork - Bacon,back Peameal	49	4.65
5	6	2	4	3.28	2	Pork - Bacon,back Peameal	49	4.65
6	3	3	10	9.12	3	Lettuce - Romaine, Heart	38	3.35
7	4	3	7	6.99	3	Lettuce - Romaine, Heart	38	3.35
3	6	3	4	7.46	3	Lettuce - Romaine, Heart	38	3.35
9	7	3	7	9.17	3	Lettuce - Romaine, Heart	38	3.35
10	1	4	4	3.74	4	Brocolinni - Gaylan, Chinese	90	4.53
11	2	4	4	1.66	4	Brocolinni - Gaylan, Chinese	90	4.53
12	6	5	1	3.45	5	Sauce - Ranch Dressing	94	1.63
13	8	5	2	6.94	5	Sauce - Ranch Dressing	94	1.63
14	2	6	2	2.94	6	Petit Baguette	14	2.39
15	9	6	5	7.28	6	Petit Baguette	14	2.39
16	8	8	2	8.59	8	Island Oasis - Raspberry	26	0.74
17	10	9	9	4.28	9	Longan	67	2.26
18	4	10	7	6.40	10	Broom - Push	6	1.09

12) SQL - SELF JOIN

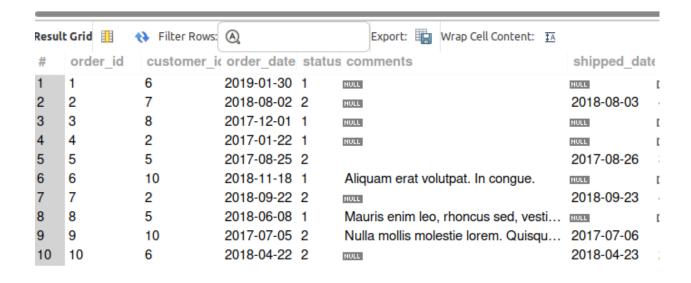
```
USE sql_hr;
    4
    5 •
          SELECT
    6
          e.employee id,
    7
          e.first_name,
    8
          m.first name AS Manager
    9
          FROM employees e
   10
          JOIN employees m ON e.reports_to = m.employee_id;
Result Grid
               Filter Rows: (A)
                                                 Export: 📺 Wrap Cell Content: 🏗
     employee_ic first_name Manager
     33391
1
                 D'arcy
                           Yovonnda
2
     37851
                 Sayer
                           Yovonnda
3
                 Mindy
     40448
                           Yovonnda
4
     56274
                 Keriann
                           Yovonnda
5
                           Yovonnda
     63196
                 Alaster
6
     67009
                 North
                           Yovonnda
7
                 Elladine
     67370
                           Yovonnda
8
     68249
                 Nisse
                           Yovonnda
9
     72540
                 Guthrey
                           Yovonnda
     72913
10
                 Kass
                           Yovonnda
     75900
                 Virge
                           Yovonnda
11
12
     76196
                 Mirilla
                           Yovonnda
13
     80529
                 Lynde
                           Yovonnda
14
     80679
                 Mildrid
                           Yovonnda
15
     84791
                 Hazel
                           Yovonnda
16
     95213
                 Cole
                           Yovonnda
17
     96513
                 Theresa
                           Yovonnda
18
    98374
                 Estrellita
                           Yovonnda
19
    115357
                 lvy
                           Yovonnda
```

13) SQL - joining multiple tables

```
USE sql store;
          SELECT o.order_id,
          o.order date,
    5
          c.first_name,
    7
          c.last_name
          FROM orders o
    8
    9
          JOIN customers c ON o.customer_id = c.customer_id
          JOIN order_statuses os ON o.status = os.order_status_id;
  10
Result Grid 🔢
              Name of the Filter Rows:
                                                Export: Wrap Cell Content: 1A
               order_date first_name
                                       last name
1
    1
               2019-01-30 Elka
                                       Twiddell
2
                                       Dowson
               2018-08-02 Ilene
    3
3
               2017-12-01 Thacher
                                       Naseby
4
    4
               2017-01-22 Ines
                                       Brushfield
5
               2017-08-25 Clemmie
                                       Betchley
6
    6
               2018-11-18 Levy
                                       Mynett
7
    7
               2018-09-22 Ines
                                       Brushfield
8
               2018-06-08 Clemmie
                                       Betchley
               2017-07-05 Levy
                                       Mynett
10 10
               2018-04-22 Elka
                                       Twiddell
```

14) SQL - Implicit Join Syntax

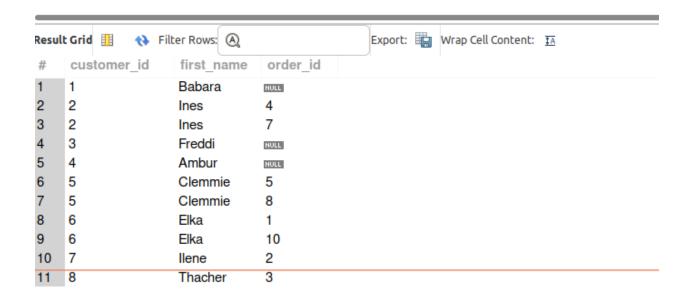
```
3 • SELECT * FROM orders o, customers c
4 WHERE o.customer_id = c.customer_id;
```



15) SQL - OUTER JOIN

LEFT JOIN:

```
3 • SELECT c.customer_id,
4     c.first_name,
5          o.order_id
6     FROM customers c
7     LEFT JOIN orders o
8     ON c.customer_id = o.customer_id
9     ORDER BY c.customer_id;
10
```

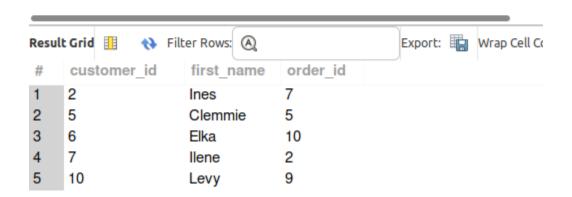


RIGHT JOIN:

```
12
        -- right join
  13 •
         SELECT c.customer id,
  14
         c.first_name,
  15
              o.order_id
  16
         FROM customers c
         RIGHT JOIN orders o
  17
  18
         ON c.customer_id = o.customer_id
  19
         ORDER BY c.customer_id;
  20
  21
                                              Export: Wrap Cell Content: 🚁
Result Grid 🔢
             🙌 Filter Rows: 🔌
    customer id
                               order id
                    first name
1
    2
                               4
                   Ines
    2
                               7
2
                   Ines
3
    5
                               5
                   Clemmie
4
    5
                   Clemmie
                               8
5
    6
                   Elka
                               1
    6
                   Elka
6
                               10
7
    7
                   llene
                               2
8
    8
                   Thacher
                               3
    10
                   Levy
                               6
    10
10
                   Levy
                               9
```

17) SQL - Outer join between multiple tables

```
4
 3 •
      SELECT
      c.customer id,
 4
      c.first name,
 5
 6
      o.order id
 7
      FROM customers c
      LEFT JOIN orders o
 8
      ON c.customer id = o.customer id
 9
      JOIN shippers sh
10
      ON o.shipper id = sh.shipper id
11
      ORDER BY c.customer id;
12
13
```



18) SQL - SELF OUTER JOIN

```
3 • USE sql_hr;
 4
 5 •
     SELECT
 6
     e.employee_id,
     e.first name,
7
     m.first_name AS Manager
8
    FROM employees e
9
    Left JOIN employees m
10
11
     ON e.reports_to = m.employee_id;
```

Resul	t Grid 🏢 🐧	Filter Rows:	(A)	Export:	Wrap Cell Co
#	employee_i	ic first_name	Manager		
1	33391	D'arcy	Yovonnda		
2	37270	Yovonnda	NULL		
3	37851	Sayer	Yovonnda		
4	40448	Mindy	Yovonnda		
5	56274	Keriann	Yovonnda		
6	63196	Alaster	Yovonnda		
7	67009	North	Yovonnda		
8	67370	Elladine	Yovonnda		
9	68249	Nisse	Yovonnda		
10	72540	Guthrey	Yovonnda		
11	72913	Kass	Yovonnda		

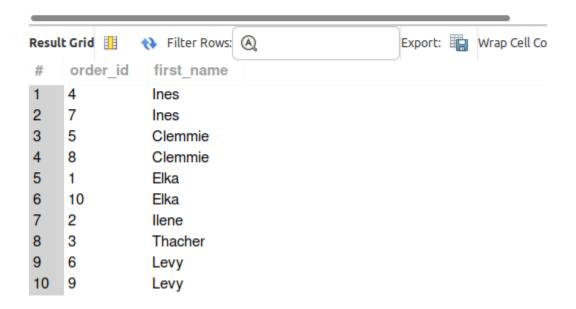
19) SQL - USING

```
4
    3 •
          USE sql store;
          SELECT
    4 •
          o.order id,
    5
    6
          c.first name,
    7
          sh.name AS shipper
    8
          FROM orders o
    9
          JOIN customers c USING (customer id)
   10
          JOIN shippers sh USING (shipper id);
   11
              Name of the Filter Rows:
                                                 Export: Wrap Cell C
Result Grid
#
     order id
               first name
                             shipper
1
     9
                Levy
                            Hettinger LLC
2
    10
                Elka
                            Schinner-Predovic
3
    5
                Clemmie
                            Satterfield LLC
     2
                llene
                            Mraz, Renner and Nolan
    7
                            Mraz, Renner and Nolan
                Ines
```

20) SQL - NATURAL JOIN

used to join two or more tables based on columns that have the same name and data type in both tables.

```
3 • SELECT
4 o.order_id,
5 c.first_name
6 FROM orders o
7 NATURAL JOIN customers c;
8
```



21) SQL - CROSS JOINS

Cartesian product of tables.

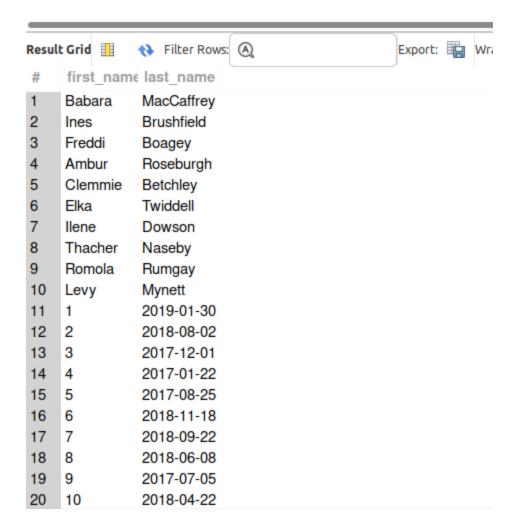
```
3 • SELECT
4 c.first_name,
5 p.name AS product
6 FROM customers c
7 CROSS JOIN products p;
```

Resul	t Grid 🕕 🕠	Filter Rows: (A) Export: Wri
#	first_name	product
1	Levy	Foam Dinner Plate
2	Romola	Foam Dinner Plate
3	Thacher	Foam Dinner Plate
4	llene	Foam Dinner Plate
5	Elka	Foam Dinner Plate
6	Clemmie	Foam Dinner Plate
7	Ambur	Foam Dinner Plate
8	Freddi	Foam Dinner Plate
9	Ines	Foam Dinner Plate
10	Babara	Foam Dinner Plate
11	Levy	Pork - Bacon,ba
12	Romola	Pork - Bacon,ba
13	Thacher	Pork - Bacon,ba
14	llene	Pork - Bacon,ba
15	Elka	Pork - Bacon,ba
16	Clemmie	Pork - Bacon,ba
17	Ambur	Pork - Bacon,ba
18	Freddi	Pork - Bacon,ba
19	Ines	Pork - Bacon,ba
20	Babara	Pork - Bacon,ba

22) SQL - UNION

To add multiple commands output.

```
SELECT first_name, last_name
FROM customers
UNION
SELECT order_id, order_date
FROM orders
```



23) SQL - INSERT

```
INSERT INTO customers

VALUES(DEFAULT, 'JOHN', 'SMITH', NULL, NULL, 'ADDRESS', 'CITY', 'CA', DEFAULT);

6

7
8
9
```

Ac	Action Output ~							
‡	Time	Action	Message					
56	16:22:50	SELECT o.order_id, c.first_name, sh.name AS shipper FR	Error Code: 1146. Table 'sql_hr.or					
57	16:22:55	SELECT o.order_id, c.first_name, sh.name AS shipper FR	Error Code: 1146. Table 'sql_hr.or					
58	16:22:55	SELECT o.order_id, c.first_name, sh.name AS shipper FR	Error Code: 1146. Table 'sql_hr.or					
59	16:23:19	USE sql_store	0 row(s) affected					
60	16:23:19	SELECT o.order_id, c.first_name, sh.name AS shipper FR	5 row(s) returned					
61	16:26:05	SELECT o.order_id, c.first_name FROM orders o NATUR	10 row(s) returned					
62	16:27:51	SELECT c.first_name, p.name AS product FROM custome	100 row(s) returned					
63	16:31:07	SELECT first_name, last_name FROM customers UNION \dots	20 row(s) returned					
64	16:35:24	INSERT INTO customers VALUES(DEFAULT,'JOHN','SMITH	1 row(s) affected					

insert into multiple rows

```
INSERT INTO shippers (name)

VALUES ('shipper1'),

('shipper2'),

('shipper3');
```

Ad	Action Output ~							
#	Time	Action	Message					
58	16:22:55	SELECT o.order_id, c.first_name, sh.name AS shipper FR	Error Code: 1146. Table 'sql_hr.or					
59	16:23:19	USE sql_store	0 row(s) affected					
60	16:23:19	SELECT o.order_id, c.first_name, sh.name AS shipper FR	5 row(s) returned					
61	16:26:05	SELECT o.order_id, c.first_name FROM orders o NATUR	10 row(s) returned					
62	16:27:51	SELECT c.first_name, p.name AS product FROM custome	100 row(s) returned					
63	16:31:07	SELECT first_name, last_name FROM customers UNION	20 row(s) returned					
64	16:35:24	INSERT INTO customers VALUES (DEFAULT, 'JOHN', 'SMITH	1 row(s) affected					
65	16:39:32	INSERT INTO shippers (name) VALUES ('shipper1'), ('ship	3 row(s) affected Records: 3 Duplicates: 0 Warning					

Insert Hierarchical rows

```
4
   3 •
         INSERT INTO orders (customer_id, order_date,status)
   4
         VALUES (1,'2019-01-01',1);
   5
         SELECT last insert id()
   6 •
   7
         -- LAST INSERTED ID() if new entry is there we can use last inserted id
   8
   9
         -- it will automaticly increment the id and can be assigned to new entry
  10
  11
            🙌 Filter Rows: 🙉
                                           Export: Wrap Cell Content: IA
Result Grid
    last insert id(
  11
```

24) SQL - CREATING COPY OF TABLE

```
✓ 
□ sql_hr

                         3 • CREATE TABLE orders_archived AS
  > 🛅 Tables
                               SELECT * FROM orders

■ Views

   Stored Procedure
   Functions
> 🛢 sql_inventory
> 🛢 sql_invoicing

√ 
□ sql_store

√ Tables

    > 🗏 customers
    > order_item_nd
    > order_items
    > orders
    > orders_archive
    > order_statuses
    > III products
```

25) SQL - UPDATE

Update single row

```
1  -- Updating rows
2  use sql_invoicing;
3  UPDATE invoices
4  SET payment_total = 10,
5  payment_date = '2019-01-01';
6
7
```

Ac	tion Output	~]	
‡ 	Time	Action	Message
:60	16:23:19	SELECT o.order_id, c.first_name, sh.name AS shipper FR	5 row(s) returned
:61	16:26:05	SELECT o.order_id, c.first_name FROM orders o NATUR	10 row(s) returned
:62	16:27:51	${\tt SELECTc.first_name,p.nameASproductFROMcustome}$	100 row(s) returned
:63	16:31:07	SELECT first_name, last_name FROM customers UNION	20 row(s) returned
:64	16:35:24	${\tt INSERTINTOcustomersVALUES(DEFAULT,'JOHN','SMITH}$	1 row(s) affected
:65	16:39:32	INSERT INTO shippers (name) VALUES ('shipper1'), ('ship	3 row(s) affected Records: 3 Duplicates: 0 Warnings
66	16:45:19	INSERT INTO orders (customer_id, order_date, status) V	1 row(s) affected
:67	16:45:19	SELECT last_insert_id() - LAST_INSERTED_ID() if new en	1 row(s) returned
:68	16:49:04	CREATE TABLE orders_archived AS SELECT * FROM orders	11 row(s) affected Records: 11 Duplicates: 0 Warning
:69	16:52:44	use sql_invoicing	0 row(s) affected
70	16:52:44	UPDATE invoices SET payment_total = 10, payment_date	Error Code: 1175. You are using sa To disable safe mode, toggle the c

update multiple rows

Ac	tion Output	~]	
‡ (61	Time 10:20:05	Action SELECT 0.0rder Id, C.FIFSC Name FROM Orders 0 NATUR	Message Tu row(s) recurned
62	16:27:51	SELECT c.first_name, p.name AS product FROM custome	100 row(s) returned
63	16:31:07	${\tt SELECTfirst_name, last_nameFROMcustomersUNION\dots}$	20 row(s) returned
64	16:35:24	${\tt INSERT\ INTO\ customers\ VALUES(DEFAULT,'JOHN','SMITH}$	1 row(s) affected
65	16:39:32	INSERT INTO shippers (name) VALUES ('shipper1'), ('ship	3 row(s) affected Records: 3 Duplicates: 0 Warnings
66	16:45:19	INSERT INTO orders (customer_id, order_date, status) V	1 row(s) affected
67	16:45:19	${\tt SELECT\ last_insert_id()\ -LAST_INSERTED_ID()\ if\ new\ en}$	1 row(s) returned
68	16:49:04	CREATE TABLE orders_archived AS SELECT * FROM orders	11 row(s) affected Records: 11 Duplicates: 0 Warning
69	16:52:44	use sql_invoicing	0 row(s) affected
70	16:52:44	UPDATE invoices SET payment_total = 10, payment_date	Error Code: 1175. You are using sa To disable safe mode, toggle the c
71	16:55:26	UPDATE invoices SET payment_total = 20 WHERE client_i	11 row(s) affected Rows matched: 11 Changed: 11 W

using sub queries in update

Ac	tion Output	~]	
¥	Time	Action	Message
64	16:35:24	${\tt INSERTINTOcustomersVALUES(DEFAULT,'JOHN','SMITH}$	1 row(s) affected
65	16:39:32	INSERT INTO shippers (name) VALUES ('shipper1'), ('ship	3 row(s) affected Records: 3 Duplicates: 0 Warnings
66	16:45:19	INSERT INTO orders (customer_id, order_date, status) V	1 row(s) affected
67	16:45:19	${\tt SELECTlast_insert_id()-LAST_INSERTED_ID()} \ if \ new\ en$	1 row(s) returned
68	16:49:04	CREATE TABLE orders_archived AS SELECT * FROM orders	11 row(s) affected Records: 11 Duplicates: 0 Warning
69	16:52:44	use sql_invoicing	0 row(s) affected
70	16:52:44	UPDATE invoices SET payment_total = 10, payment_date	Error Code: 1175. You are using sa To disable safe mode, toggle the o
71	16:55:26	UPDATE invoices SET payment_total = 20 WHERE client_i	11 row(s) affected Rows matched: 11 Changed: 11 W
72	16:58:40	UPDATE invoices SET payment_total = 50 WHERE client_i	Error Code: 1054. Unknown colum
73	17:00:05	UPDATE invoices SET payment_total = 50 WHERE client_i	1 row(s) affected Rows matched: 1 Changed: 1 War

25) SQL - DELETE

```
1 -- deleting rows
2
3 • DELETE FROM invoices
4 WHERE invoice_id = 1;
5
```

Ac	tion Output	~	
¥	Time	Action	Message
65	16:39:32	INSERT INTO shippers (name) VALUES ('shipper1'), ('ship	3 row(s) affected Records: 3 Duplicates: 0 Warning:
:66	16:45:19	INSERT INTO orders (customer_id, order_date, status) V	1 row(s) affected
:67	16:45:19	${\tt SELECTlast_insert_id()-LAST_INSERTED_ID()} \ if \ new\ en$	1 row(s) returned
:68	16:49:04	CREATE TABLE orders_archived AS SELECT * FROM orders	11 row(s) affected Records: 11 Duplicates: 0 Warning
:69	16:52:44	use sql_invoicing	0 row(s) affected
70	16:52:44	UPDATE invoices SET payment_total = 10, payment_date	Error Code: 1175. You are using sa To disable safe mode, toggle the c
71	16:55:26	UPDATE invoices SET payment_total = 20 WHERE client_i	11 row(s) affected Rows matched: 11 Changed: 11 W
:72	16:58:40	UPDATE invoices SET payment_total = 50 WHERE client_i	Error Code: 1054. Unknown colum
73	17:00:05	UPDATE invoices SET payment_total = 50 WHERE client_i	1 row(s) affected Rows matched: 1 Changed: 1 War
74	17:02:00	DELETE FROM invoices WHERE invoice_id = 1	1 row(s) affected