

3.6 Групування записів / Group

GROUP BY

дозволяє групувати видобуті рядки

allows to group selected records

- використовується разом з функціями, застосовуваними до груп рядків / it is used with functions applied to groups of records
- агрегатні функції обчислюють одне значення для кожної групи, створеної конструкцією GROUP BY / aggregate functions calculate a single value for each group created by using the GROUP BY statement

3.6 Групування записів / Group

- функції дозволяють дізнатися число рядків в групі, підрахувати середнє значення, отримати суму значень стовпців / functions allow to define a number of records in a group, calculate the average value, get the sum of values for columns
- результуюче значення обчислюється для значень, що не є NULL / a result value is calculated for values do not equal to NULL
- функції можна використовувати в запитах без групування / functions might be used in queries without GROUP BY option

3.6 Групування записів / Group

```
SELECT COUNT(DISTINCT b_cat_ID) FROM books;
```

| Result #1 (1×1) | |
|--------------------------|---|
| COUNT(DISTINCT b_cat_ID) | |
| | 5 |

```
SELECT COUNT(DISTINCT b_cat_ID) AS total FROM books;
```

| Result #1 (1×1) | |
|-----------------|---|
| total | |
| | 5 |

```
SELECT * FROM catalogs WHERE cat_ID = MAX(cat_ID);
```

???

3.6 Групування записів / Group

```
SELECT * FROM catalogs ORDER BY cat_ID DESC LIMIT 1;
```

| catalogs (2×1) | |
|----------------|-------------|
| cat_ID | cat_name |
| 5 | Мультимедиа |


```
SELECT b_cat_ID FROM books  
GROUP BY b_cat_ID ORDER BY b_cat_ID;
```

```
SELECT DISTINCT b_cat_ID FROM books;
```


| books (1×5) | |
|-------------|--|
| b_cat_ID | |
| 1 | |
| 2 | |
| 3 | |
| 4 | |
| 5 | |

3.6 Групування записів / Group

```
SELECT b_cat_ID, COUNT(b_cat_ID) FROM books  
WHERE b_cat_ID > 2  
GROUP BY b_cat_ID  
ORDER BY b_cat_ID;
```


| books (2×3) | |
|--|-----------------|
|  b_cat_ID | COUNT(b_cat_ID) |
| 3 | 4 |
| 4 | 5 |
| 5 | 6 |

```
SELECT b_cat_ID, COUNT(b_cat_ID) AS total FROM books  
GROUP BY b_cat_ID  
HAVING total > 5  
ORDER BY b_cat_ID;
```

| books (2×3) | |
|--|-------|
|  b_cat_ID | total |
| 1 | 9 |
| 2 | 6 |
| 5 | 6 |

3.6 Групування записів / Group

```
SELECT b_cat_ID, COUNT(b_cat_ID) FROM books  
GROUP BY b_cat_ID  
HAVING b_cat_ID > 2  
ORDER BY b_cat_ID;
```

|  b_cat_ID | COUNT(b_cat_ID) |
|--|-----------------|
| 3 | 4 |
| 4 | 5 |
| 5 | 6 |

- **WHERE** – вибірка із застосуванням умови і потім групування результату / selects with a condition and then groups a result
- **HAVING** – групування результату і потім вибірка із застосуванням умови / groups a result and then selects with a condition

3.7 Використання функцій / Using functions

Вбудовані функції MySQL використовуються для вирішення специфічних завдань, що виникають при вибірці даних

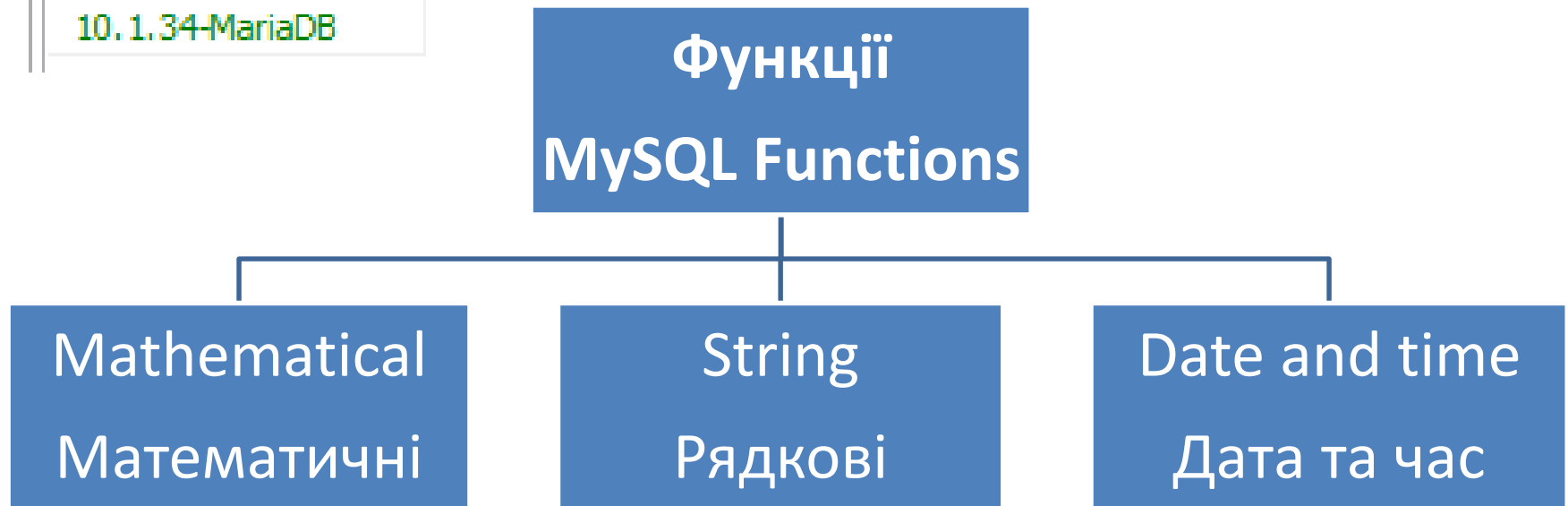
In-built MySQL functions are used to solve specific problems occurred while selecting data

- функція має унікальне ім'я / each function has a unique name
- функція може мати кілька аргументів / each function might have multiple arguments
- круглі дужки вказуються і при відсутності аргументів / parentheses should be set even if arguments are absent

3.7 Використання функцій / Using functions

SELECT VERSION();




| Result #1 (1×1) | |
|-----------------|--|
| VERSION() | |
| 10.1.34-MariaDB | |



<https://www.mysqltutorial.org/>

Query 3

```
SELECT contract.contract_number, contract.contract_date,  
supplied.supplied_product, supplied.supplied_cost, supplier.*  
FROM (supplier INNER JOIN contract ON supplier.supplier_id =  
contract.supplier_id) INNER JOIN supplied ON contract.contract_number =  
supplied.contract_number  
WHERE MONTH(contract.contract_date) = 9 AND  
YEAR(contract.contract_date) = 2018;
```

|  contract_number | contract_date |  supplied_product | supplied_cost |  supplier_id | supplier_address | supplier_phone |
|--|---------------------|--|---------------|---|----------------------------------|-----------------|
| 1 | 2018-09-01 00:00:00 | Audio Player | 700.00 | 1 | Kharkiv, Nauky av., 55, apt. 108 | phone: 32-18-44 |
| 1 | 2018-09-01 00:00:00 | New Product | 100.00 | 1 | Kharkiv, Nauky av., 55, apt. 108 | phone: 32-18-44 |
| 1 | 2018-09-01 00:00:00 | TV | 1,300.00 | 1 | Kharkiv, Nauky av., 55, apt. 108 | phone: 32-18-44 |
| 1 | 2018-09-01 00:00:00 | Video Player | 750.00 | 1 | Kharkiv, Nauky av., 55, apt. 108 | phone: 32-18-44 |
| 4 | 2018-09-24 00:00:00 | Audio Player | 320.00 | 2 | Kyiv, Peremohy av., 154, apt. 3 | |
| 4 | 2018-09-24 00:00:00 | Printer | 332.50 | 2 | Kyiv, Peremohy av., 154, apt. 3 | |
| 4 | 2018-09-24 00:00:00 | TV | 990.00 | 2 | Kyiv, Peremohy av., 154, apt. 3 | |

Query 4



```
SELECT contract.contract_number, contract.contract_date,  
contract.supplier_id, SUM(supplied.supplied_amount *  
supplied.supplied_cost) AS `Sum`  
FROM contract INNER JOIN supplied ON contract.contract_number =  
supplied.contract_number  
GROUP BY contract.contract_number, contract.contract_date,  
contract.supplier_id  
ORDER BY contract.contract_number;
```

| contract_number | contract_date | supplier_id | Sum |
|-----------------|---------------------|-------------|-----------|
| 1 | 2018-09-01 00:00:00 | 1 | 41,000.00 |
| 2 | 2019-03-21 15:18:46 | 1 | 11,350.00 |
| 3 | 2018-09-23 00:00:00 | 3 | 99,600.00 |
| 4 | 2018-09-24 00:00:00 | 2 | 76,112.50 |
| 5 | 2018-10-02 00:00:00 | 2 | 45,630.00 |
| 7 | 2018-12-27 13:30:04 | 1 | 59,985.00 |

Query 6


```
SELECT contract.contract_number, contract.contract_date,  
contract.contract_note, supplier.*, supplied.supplied_amount  
FROM contract, supplied, supplier  
WHERE contract.contract_number = supplied.contract_number AND  
contract.supplier_id = supplier.supplier_id AND  
supplied.supplied_amount = (SELECT  
MAX(supplied.supplied_amount) FROM supplied);
```

Result #1 (7×1)

|  contract_number | contract_date | contract_note |  supplier_id | supplier_address | supplier_phone | supplied_amount |
|---|---------------------|------------------------|---|------------------------------|----------------------|-----------------|
| 3 | 2018-09-23 00:00:00 | Order 56 on 28.08.2018 | 3 | Kharkiv, Pushkinska str., 77 | phone: 33-33-44, fax | 85 |

Query 11

```
SELECT supplier.supplier_id, supplier.supplier_address,  
       IFNULL(supplier_org.supplier_org_name,  
       CONCAT(RTRIM(supplier_person.supplier_last_name), ' ',  
               SUBSTRING(supplier_person.supplier_first_name, 1, 1), '. ',  
               SUBSTRING(supplier_person.supplier_middle_name, 1, 1),  
               '. ')) AS `Supplier`  
FROM (supplier LEFT JOIN supplier_person ON supplier.supplier_id =  
supplier_person.supplier_id) LEFT JOIN supplier_org ON supplier.supplier_id =  
supplier_org.supplier_id;
```

|  supplier_id | supplier_address | Supplier |
|--|------------------------------------|------------------|
| 1 | Kharkiv, Nauky av., 55, apt. 108 | Petrov P. P. |
| 2 | Kyiv, Peremohy av., 154, apt. 3 | Interfruit Ltd. |
| 3 | Kharkiv, Pushkinska str., 77 | Ivanov I. I. |
| 4 | Odesa, Derebasivska str., 75 | Transservice LLC |
| 5 | Poltava, Soborna str., 15, apt. 43 | Sydorov S. S. |

Query 16

```
SELECT supplied_product, SUM(IF(MONTH(contract_date) = 1, supplied_amount, 0)) AS `Jan`,  
      SUM(IF(MONTH(contract_date) = 2, supplied_amount, 0)) AS `Feb`,  
      SUM(IF(MONTH(contract_date) = 3, supplied_amount, 0)) AS `Mar`,  
      SUM(IF(MONTH(contract_date) = 4, supplied_amount, 0)) AS `Apr`,  
      ...  
      SUM(IF(MONTH(contract_date) = 10, supplied_amount, 0)) AS `Oct`,  
      SUM(IF(MONTH(contract_date) = 11, supplied_amount, 0)) AS `Nov`,  
      SUM(IF(MONTH(contract_date) = 12, supplied_amount, 0)) AS `Dec`  
FROM contract, supplied  
WHERE contract.contract_number = supplied.contract_number AND YEAR(contract_date) = 2018  
GROUP BY supplied_product ORDER BY supplied_product;
```

| supplied_product | Jan | Feb | Mar | Apr | May | Jun | Jul | Aug | Sep | Oct | Nov | Dec |
|------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Audio Player | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 58 | 33 | 0 | 0 |
| Monitor | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 85 | 0 | 0 | 0 |
| New Product | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 15 | 0 | 0 | 0 |
| Phone | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 |
| Printer | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 41 | 0 | 0 | 0 |
| TV | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 118 | 14 | 0 | 10 |
| Video Player | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 12 | 17 | 0 | 0 |

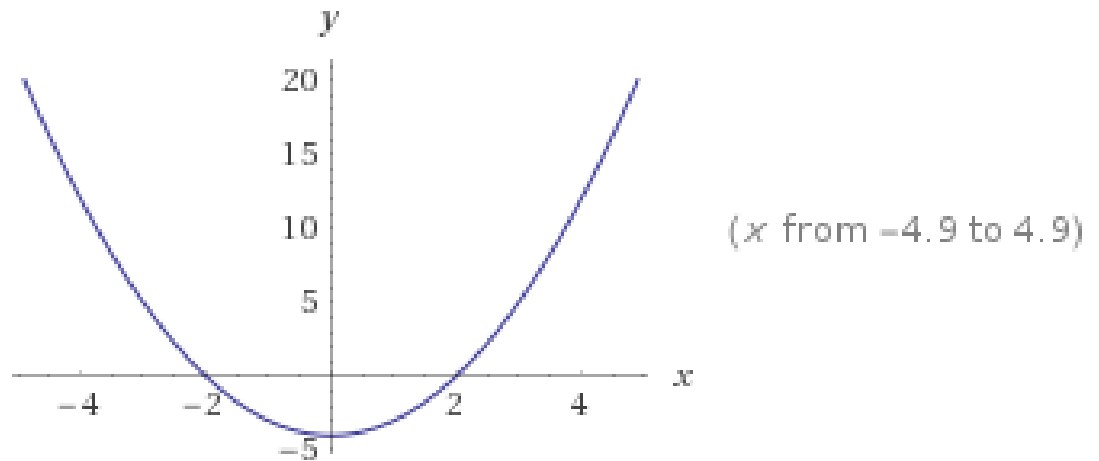
Query 17

```
SELECT supplied.contract_number, supplied.supplied_product,  
supplied.supplied_amount, supplied.supplied_cost,  
contract.contract_date,  
MONTHNAME(contract.contract_date) AS `Month`,  
YEAR(contract.contract_date) AS `Year`  
FROM supplied, contract  
WHERE contract.contract_number = supplied.contract_number;
```

| 🔑 contract_number | 🔑 supplied_product | supplied_amount | supplied_cost | contract_date | Month | Year |
|-------------------|--------------------|-----------------|---------------|---------------------|-----------|-------|
| 1 | Audio Player | 25 | 700.00 | 2018-09-01 00:00:00 | September | 2,018 |
| 1 | New Product | 15 | 100.00 | 2018-09-01 00:00:00 | September | 2,018 |
| 1 | TV | 10 | 1,300.00 | 2018-09-01 00:00:00 | September | 2,018 |
| 1 | Video Player | 12 | 750.00 | 2018-09-01 00:00:00 | September | 2,018 |
| 2 | Audio Player | 5 | 450.00 | 2019-03-21 15:18:46 | March | 2,019 |
| 2 | Stereo System | 11 | 500.00 | 2019-03-21 15:18:46 | March | 2,019 |
| 2 | Video Player | 8 | 450.00 | 2019-03-21 15:18:46 | March | 2,019 |
| 3 | Audio Player | 11 | 550.00 | 2018-09-23 00:00:00 | September | 2,018 |

3.7 Використання функцій / Using functions

$$y = f(x) = x^2 - 4$$



```
CREATE FUNCTION <name> (<args>) RETURNS <type>  
  [NOT] DETERMINISTIC  
BEGIN  
<body>  
END
```

3.7 Використання функцій / Using functions

RETURNS

- визначає тип даних, які повертає функція
- RETURNS defines a data type of returning value

[NOT] DETERMINISTIC

- визначає, чи буде функція детермінованою чи ні
- defines whether the function is deterministic or not

3.7 Використання функцій / Using functions

```
DELIMITER $$
```

```
CREATE FUNCTION my_parabola(x DOUBLE) RETURNS  
    DOUBLE DETERMINISTIC
```

```
BEGIN
```

```
    RETURN x * x - 4;
```

```
END $$
```

```
DELIMITER ;
```

3.7 Використання функцій / Using functions

DELIMITER

використовується для зміни роздільник операцій (закінчення операції) в MySQL

стандартний роздільник – ;

is used to change operations separator (operations end character) in MySQL

Standard operations separator is ;

DELIMITER //

...

DELIMITER ;

3.7 Використання функцій / Using functions

```
SELECT b_name, b_author, b_year, b_price,  
       my_parabola(b_price)  
FROM books;
```

| books (5×30) | | | | |
|--|---------------|--------|---------|----------------------|
| b_name | b_author | b_year | b_price | my_parabola(b_price) |
| JavaScript в кармане | Рева О.Н. | 2008 | 42.00 | 1,760 |
| Visual FoxPro 9.0 | Клепинин В.Б. | 2007 | 660.00 | 435,596 |
| C++ Как он есть | Тимофеев В.В. | 2009 | 218.00 | 47,520 |
| Создание приложений с помощью C# | Фаронов В.В. | 2008 | 169.00 | 28,557 |
| Delphi. Народные советы | Шкрыль А.А. | 2007 | 243.00 | 59,045 |
| Delphi. Полное руководство | Сухарев М. | 2008 | 500.00 | 249,996 |
| Профессиональное программирование на PHP | Шлоссейгл Дж. | 2006 | 309.00 | 95,477 |
| Совершенный код | Макконнелл С. | 2007 | 771.00 | 594,437 |
| Практика программирования | Керниган Б. | 2004 | 214.00 | 45,792 |

3.7 Використання функцій / Using functions

DELIMITER \$\$

CREATE FUNCTION LAST_ORDERED_BOOK() **RETURNS** INT **NOT DETERMINISTIC**

BEGIN

DECLARE last_order INT **DEFAULT** -1;

SELECT o_book_ID **INTO** last_order **FROM** orders **ORDER BY** order_ID **DESC** **LIMIT** 1;



RETURN last_order;

END \$\$

DELIMITER ;

SELECT * **FROM** books

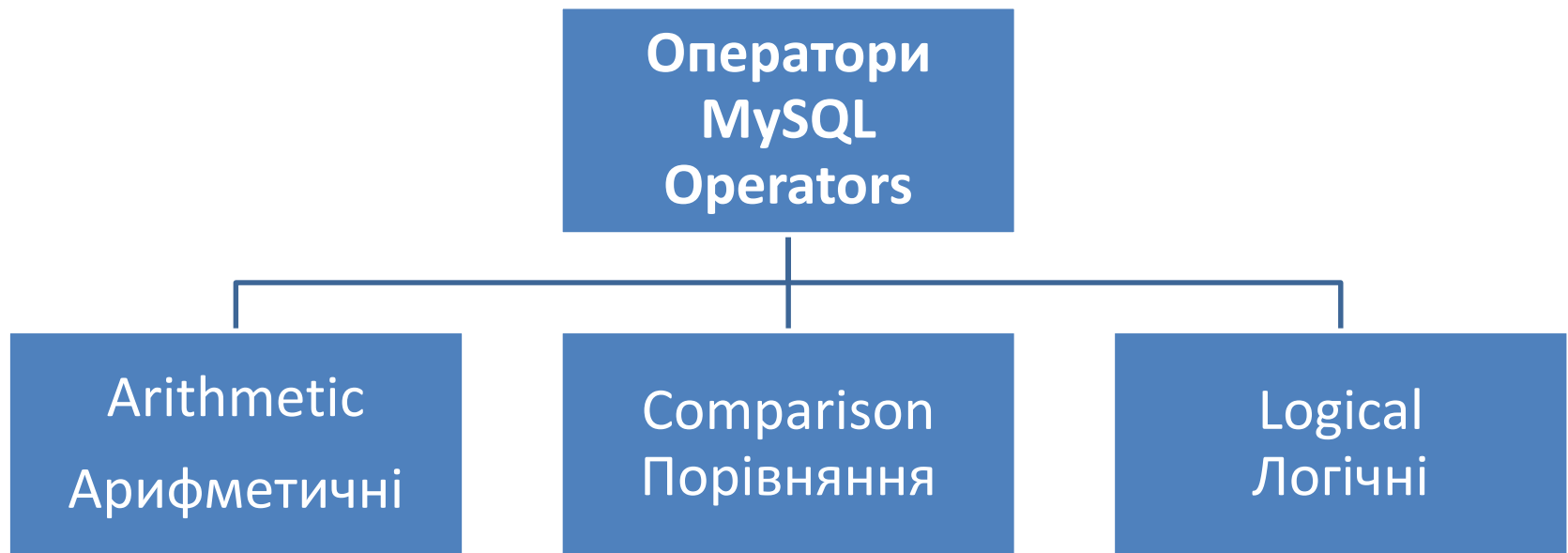
WHERE book_ID = ***LAST_ORDERED_BOOK()***;

| books (7×1) | | | | | | |
|---|-------------------|---------------|--------|---------|---------|--|
|  book_ID | b_name | b_author | b_year | b_price | b_count |  b_cat_ID |
| 20 | Компьютерные сети | Танненбаум Э. | 2007 | 630.00 | 6 | 4 |

3.8 Оператори / Operators

Оператор – конструкція мови, яка виконує перетворення даних (операндів).

Operator is a language construction used to transform data (operands).



3.8 Оператори / Operators

Арифметичні оператори / Arithmetic operators:

- (+) додавання / addition
- (-) віднімання / subtraction
- (*) множення / multiplication
- (/) поділ / division
- **DIV** цілочисельне ділення / integer division

Поділ на 0 дає безпечний результат NULL

Division by zero produces a safe result NULL

3.8 Оператори / Operators

SELECT NULL = NULL;

| Result #1 (1×1) | |
|-----------------|--|
| NULL = NULL | |
| (NULL) | |

SELECT NULL <> NULL;

| Result #1 (1×1) | |
|-----------------|--|
| NULL <> NULL | |
| (NULL) | |

SELECT NULL IS NULL;

| Result #1 (1×1) | |
|-----------------|--|
| NULL IS NULL | |
| 1 | |

3.8 Оператори / Operators

Оператори порівняння / Comparison operators:

| Оператор Operator | Опис Description |
|----------------------|--|
| = | Повертає 1, якщо операнди рівні, і 0, якщо не рівні Returns 1, if operands are equal, and 0, if they are not equal |
| <=> | Те ж саме, але не повертає NULL It is similar to the previous operator, but it does not return NULL |
| <> | Повертає 1, якщо операнди не рівні, і 0, якщо рівні Returns 1, if operands are not equal, and 0, if they are equal |
| < | Повертає 1, якщо лівий операнд менше правого Returns 1, if the left operand is left than right operand |
| <= | Повертає 1, якщо лівий операнд менше правого або вони рівні Returns 1, if the left operand is left than right operand or they are equal |

3.8 Оператори / Operators

| Оператор Operator | Опис Description |
|--------------------------------|--|
| > | Повертає 1, якщо лівий операнд більше правого Returns 1, if the left operand is greater than right operand |
| >= | Повертає 1, якщо лівий операнд більше правого або вони рівні Returns 1, if the left operand is greater than right operand or they are equal |
| n BETWEEN min AND max | Повертає 1, якщо перевіряється значення n знаходиться між min і max Returns 1, if the value n is between min and max |
| IS NULL IS NOT NULL | Перевіряє, чи є значення значенням NULL чи ні Checks, whether a value is NULL or not |
| n IN (set) | Повертає 1, якщо перевіряється значення n входить в множину Returns 1, if the value n is included into a set |

3.8 Оператори / Operators

Логічні оператори / Logical operators:

| Оператор Operator | Опис Description |
|----------------------|---|
| n AND m | true AND true = true, false AND any = false |
| n OR m | true OR any = true, false OR false = false |
| NOT n | NOT true = false, NOT false = true |
| n XOR m | true XOR true = false, true XOR false = true, false XOR true = true, false XOR false = false |

- ненульове значення, відмінне від NULL, інтерпретується як «істина»
- non-zero and non-NULL value is interpreted as “true”