

# Бази даних та SQL

Основи DML та Join

# Бази даних та SQL

## План уроку

- Data Manipulation Language (DML)
  - INSERT
  - UPDATE
  - DELETE
- JOIN's
  - INNER JOIN
  - LEFT OUTER JOIN
  - RIGHT OUTER JOIN
  - CROSS JOIN

## Data Manipulation Language (DML)

# Бази даних та SQL

## Data Manipulation Language

Data Manipulation Language (DML) – це сімейство комп'ютерних мов, що використовуються користувачами баз даних для отримання, вставки, видалення або зміни даних у базах даних.

Функції DML:

**SELECT** – вибірка даних

**INSERT** – додавання даних

**UPDATE** – оновлення даних

**DELETE** – видалення даних

# Бази даних та SQL

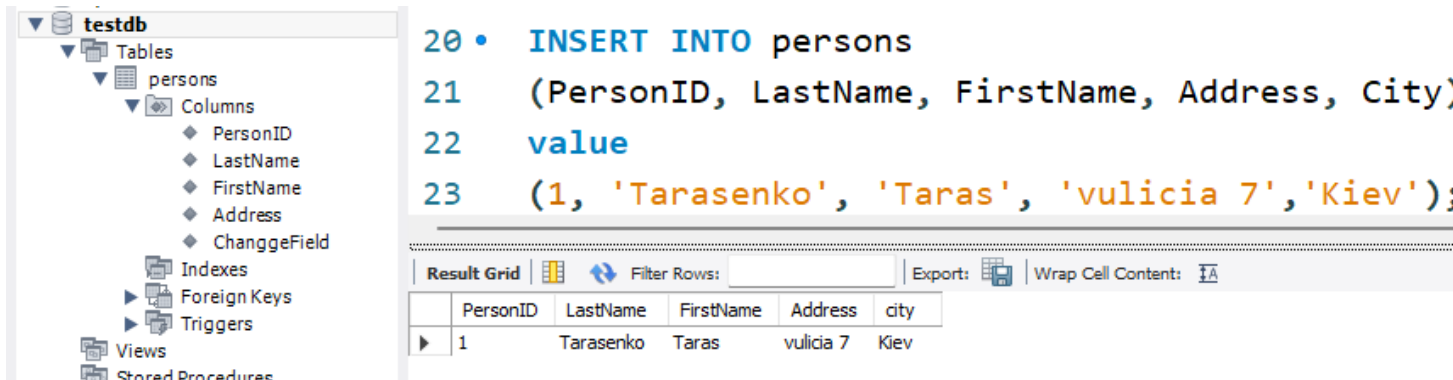
## INSERT INTO

Оператор **INSERT INTO** використовується для вставки нових записів у таблицю

Інструкцію INSERT INTO можна написати двома способами:

**INSERT INTO** *table\_name* (*column1*, *column2*, *column3*, ...)  
**VALUES** (*value1*, *value2*, *value3*, ...);

**INSERT INTO** *table\_name*  
**VALUES** (*value1*, *value2*, *value3*, ...);

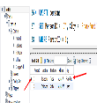


The screenshot shows a database management tool interface. On the left, a tree view displays the database structure: 'testdb' contains 'Tables', which includes 'persons'. The 'persons' table has columns: 'PersonID', 'LastName', 'FirstName', 'Address', and 'ChangeField'. Below the columns are 'Indexes', 'Foreign Keys', 'Triggers', 'Views', and 'Stored Procedures'. On the right, an SQL editor shows the following code:

```
20 • INSERT INTO persons
21 (PersonID, LastName, FirstName, Address, City)
22 value
23 (1, 'Tarasenko', 'Taras', 'vulicia 7', 'Kiev');
```

Below the code, a 'Result Grid' is visible, showing the result of the query:

	PersonID	LastName	FirstName	Address	city
▶	1	Tarasenko	Taras	vulicia 7	Kiev



Якщо ви додаєте значення для всіх стовпців таблиці, вам не потрібно вказувати назви стовпців у SQL-запиті. Однак переконайтеся, що порядок значень відповідає порядку стовпців у таблиці



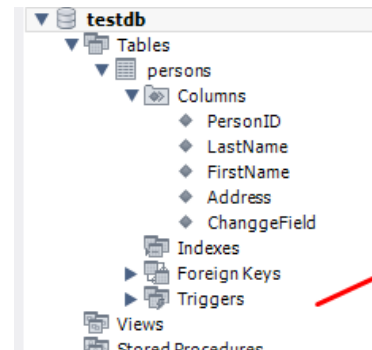
# Бази даних та SQL

## UPDATE

Оператор **UPDATE** використовується для зміни існуючих записів у таблиці

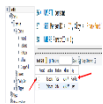
```
UPDATE table_name  
SET column1 = value1, column2 = value2, ...  
WHERE condition;
```

	PersonID	LastName	FirstName	Address	city
▶	1	Tarasenko	Taras	vulicia 7	Kiev
	2	Dmirovich	Dmitro	vulicia 77	Uman



```
26 • UPDATE persons  
27   SET PersonID = '7', City = 'Frankfurt'  
28   WHERE PersonID = 1;
```

	PersonID	LastName	FirstName	Address	city
	7	Tarasenko	Taras	vulicia 7	Frankfurt
	2	Dmirovich	Dmitro	vulicia 77	Uman



Будьте обережні, оновлюючи записи в таблиці! Зверніть увагу на оператор WHERE в операторі UPDATE. Оператор WHERE визначає, які записи потрібно оновити. Без WHERE, усі записи в таблиці будуть оновлені!

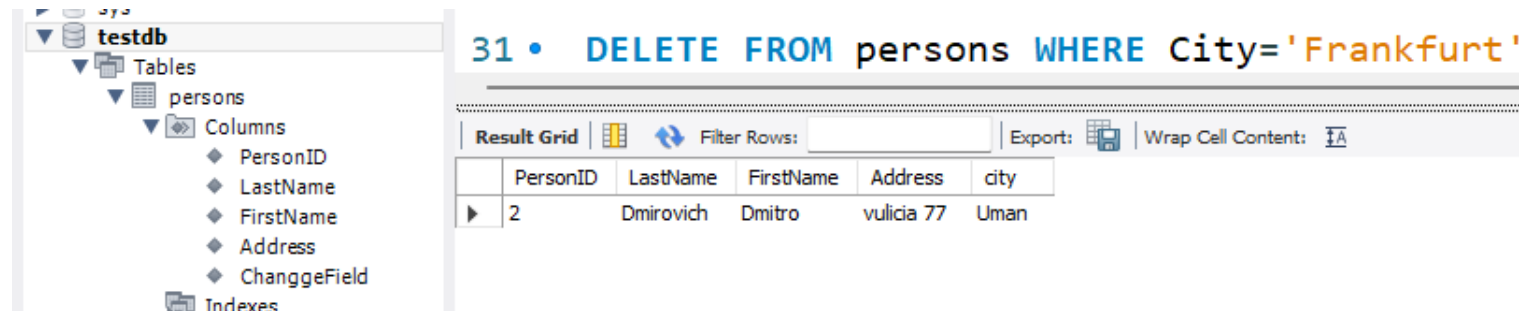


# Бази даних та SQL

## DELETE

Оператор **DELETE** використовується для видалення існуючих записів у таблиці

```
DELETE FROM table_name  
WHERE condition
```



The screenshot shows a database management interface. On the left, a tree view displays a database named 'testdb' containing a table 'persons' with columns: PersonID, LastName, FirstName, Address, and ChangeField. On the right, a SQL query editor shows the command: **31 • DELETE FROM persons WHERE City='Frankfurt'**. Below the query, a 'Result Grid' is visible, showing a single row of data:

	PersonID	LastName	FirstName	Address	city
▶	2	Dmirovich	Dmitro	vulicia 77	Uman

Примітка. Будьте обережні, видаляючи записи в таблиці! Зверніть увагу на умову WHERE в операторі DELETE. Саме WHERE вказує, які записи потрібно видалити. Якщо пропустити WHERE, усі записи в таблиці будуть видалені!



## Об'єднання таблиць – JOIN-и



# Бази даних та SQL

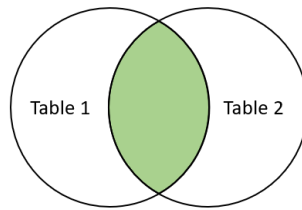
## Основні типи JOIN-ів

**INNER JOIN**: повертає записи, які мають відповідні значення в обох таблицях

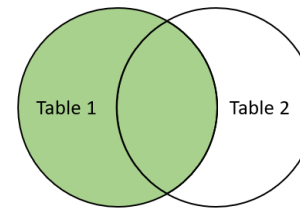
**LEFT JOIN**: повертає всі записи з лівої таблиці та відповідні записи з правої таблиці

**RIGHT JOIN**: повертає всі записи з правої таблиці та відповідні записи з лівої таблиці

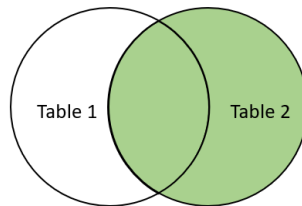
**CROSS JOIN**: повертає всі записи з обох таблиць



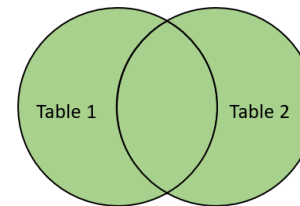
**INNER JOIN**



**LEFT JOIN**



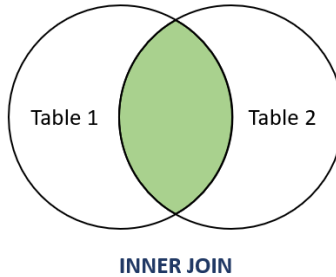
**RIGHT JOIN**



**CROSS JOIN**

# Бази даних та SQL

## INNER JOIN



Ключове слово **INNER JOIN** вибирає всі рядки з обох таблиць, якщо є збіг між стовпцями. Якщо в таблиці «Замовлення» є записи, які не збігаються в «Клієнти», ці замовлення не відобразяться!

Table: Customers

customer_id	first_name
1	John
2	Robert
<u>3</u>	David
4	John
<u>5</u>	Betty

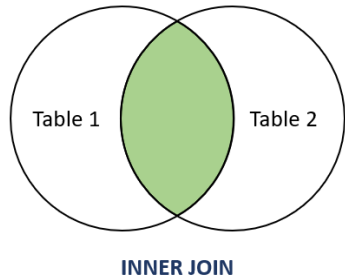
Table: Orders

order_id	amount	customer
1	200	10
2	500	<u>3</u>
3	300	6
4	800	<u>5</u>
5	150	8

customer_id	first_name	amount
3	David	500
5	Betty	800

# Бази даних та SQL

## INNER JOIN



Ключове слово **INNER JOIN** вибирає записи, які мають відповідні значення в обох таблицях

```
SELECT column_name(s)
FROM table1
INNER JOIN table2
ON table1.column_name = table2.column_name;
```

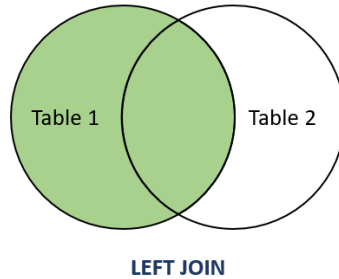
```
38 • SELECT * FROM
39                               JoinTest2
40                               INNER JOIN
41                               JoinTest1
42 ON id_jt1 = id_jt2;
```

Result Grid				
	id_jt2	name	id_jt1	name
▶	1	one	1	one
	2	two	2	two
	3	three	3	three
	4	four	4	four
	5	five	5	five



# Бази даних та SQL

## LEFT JOIN



Ключове слово **LEFT JOIN** повертає всі записи з лівої таблиці і відповідні записи (якщо є) з правої таблиці.

Тут команда SQL вибирає стовпці *customer\_id* і *first\_name* (з *Customers* таблиці) і *amount* стовпець (з *Orders* таблиці).

Набір результатів міститиме ті рядки, де є збіг між *customer\_id* (таблиці *Customers*) і *customer* (таблиці *Orders*), а також усі інші рядки таблиці *Customers*.

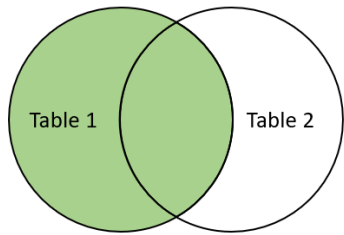
customer_id	first_name
1	John
2	Robert
<u>3</u>	David
4	John
<u>5</u>	Betty

order_id	amount	customer
1	200	10
2	500	<u>3</u>
3	300	6
4	800	<u>5</u>
5	150	8

customer_id	first_name	amount
1	John	
2	Robert	
3	David	500
4	John	
5	Betty	800

# Бази даних та SQL

## LEFT JOIN



LEFT JOIN

Ключове слово **LEFT JOIN** повертає всі записи з лівої таблиці і відповідні записи (якщо є) з правої таблиці.

```
SELECT column_name(s)
FROM table1
LEFT JOIN table2
ON table1.column_name = table2.column_name;
```

```
47 • SELECT * FROM JoinTest2
48     LEFT OUTER JOIN JoinTest1
49     ON id_jt1=id_jt2;
```

Result Grid				
Filter Rows: <input type="text"/>				
Export:  Wrap Cell C				
	id_jt2	name	id_jt1	name
▶	1	one	1	one
	2	two	2	two
	3	three	3	three
	4	four	4	four
	5	five	5	five
	6	six	NULL	NULL
	7	seven	NULL	NULL
	8	eight	NULL	NULL

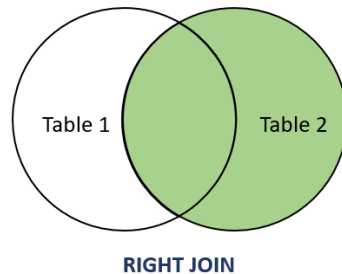


Примітка. Ключове слово **LEFT JOIN** повертає всі записи з лівої таблиці (Клієнти), навіть якщо в правій таблиці (Замовлення) немає збігів.



# Бази даних та SQL

## RIGHT JOIN



Ключове слово **RIGHT JOIN** повертає всі записи з правої таблиці і відповідні записи (якщо є) з лівої таблиці.

Тут команда SQL вибирає стовпці *customer\_id* і *first\_name* (з *Customers* таблиці) і *amount* стовпець (з *Orders* таблиці).

І набір результатів міститиме ті рядки, де є збіг між *customer\_id* (таблиці *Customers*) і *customer* (таблиці *Orders*), а також усі інші рядки з *Orders* стіл.

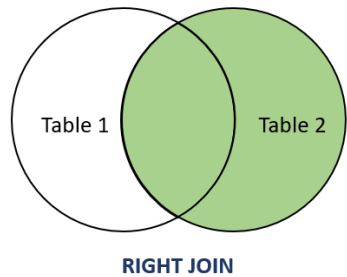
Table: Customers		Table: Orders		
customer_id	first_name	order_id	amount	customer
1	John	1	200	10
2	Robert	2	500	<u>3</u>
<u>3</u>	David	3	300	6
4	John	4	800	<u>5</u>
<u>5</u>	Betty	5	150	8

customer_id	first_name	amount
3	David	500
5	Betty	800
		200
		300
		150

# Бази даних та SQL

## RIGHT JOIN



Ключове слово **RIGHT JOIN** повертає всі записи з правої таблиці і відповідні записи (якщо є) з лівої таблиці.

```
SELECT column_name(s)
FROM table1
RIGHT JOIN table2
ON table1.column_name = table2.column_name;
```

```
54 • SELECT * FROM JoinTest2
55     RIGHT OUTER JOIN JoinTest1
56     ON id_jt1 = id_jt2;
```

Result Grid				
Filter Rows: <input type="text"/>				
Export:  Wrap Cell <input type="checkbox"/>				
	id_jt2	name	id_jt1	name
▶	1	one	1	one
	2	two	2	two
	3	three	3	three
	4	four	4	four
	5	five	5	five
	NULL	NULL	9	nine
	NULL	NULL	10	ten

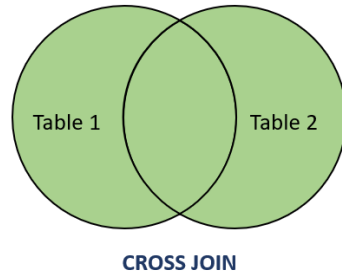


Примітка. Ключове слово **RIGHT JOIN** повертає всі записи з лівої таблиці (Клієнти), навіть якщо в правій таблиці (Замовлення) немає збігів.

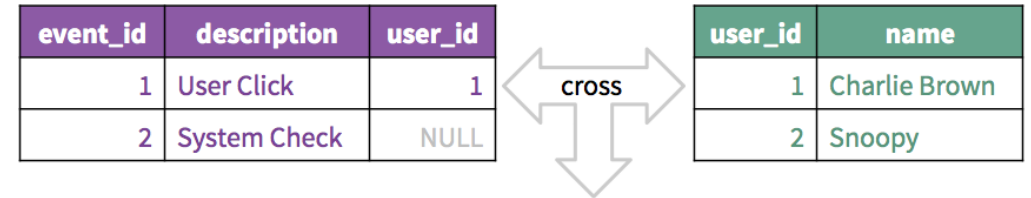


# Бази даних та SQL

## CROSS JOIN



Ключове слово **CROSS JOIN** повертає всі записи з обох таблиць. Запит створює рядок для кожної комбінації рядків та об'єднаного перегляду.

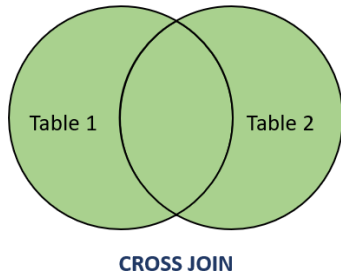


event_id	description	user_id	user_id	name
1	User Click	1	1	Charlie Brown
1	User Click	1	2	Snoopy
2	System Check	NULL	1	Charlie Brown
2	System Check	NULL	2	Snoopy



# Бази даних та SQL

## CROSS JOIN



Ключове слово **CROSS JOIN** повертає всі записи з обох таблиць

```
SELECT column_name(s)
FROM table1
CROSS JOIN table2;
```

```
61 • SELECT * FROM JoinTest1
62 CROSS JOIN JoinTest2;
```

Result Grid | Filter Rows: | Export:

	id_jt1	name	id_jt2	name
▶	10	ten	1	one
	9	nine	1	one
	5	five	1	one
	4	four	1	one
	3	three	1	one
	2	two	1	one
	1	one	1	one
	10	ten	2	two
	9	nine	2	two
	5	five	2	two
	4	four	2	two
	3	three	2	two
	2	two	2	two
	1	one	2	two
	10	ten	3	three
	9	nine	3	three
	5	five	3	three



# Manual QA

Що ми сьогодні вивчили

# Бази даних та SQL

## План уроку

- Data Manipulation Language (DML)
  - SELECT
  - INSERT
  - UPDATE
  - DELETE
- JOIN's
  - INNER JOIN
  - LEFT OUTER JOIN
  - RIGHT OUTER JOIN
  - CROSS JOIN

# Manual QA

## Підсумки

Що одне, найголовніше, ви дізнались сьогодні?

# Manual QA

Дякую за увагу! До нових зустрічей!

# Інформаційний відеосервіс для розробників програмного забезпечення

