

Бази даних та SQL

Основи DDL

Бази даних та SQL

План уроку

- Data Definition Language (DDL)
 - CREATE
 - ALTER
 - DROP
- Цілісність даних
- Primary Key
- Foreign Key

Data Definition Language (DDL)

Data Definition Language

Data Definition Language (DDL) – це словник, який використовується для визначення чи опису структури бази даних.

Функції DDL:

CREATE – визначення нових сутностей

ALTER – зміна певних існуючих сутностей

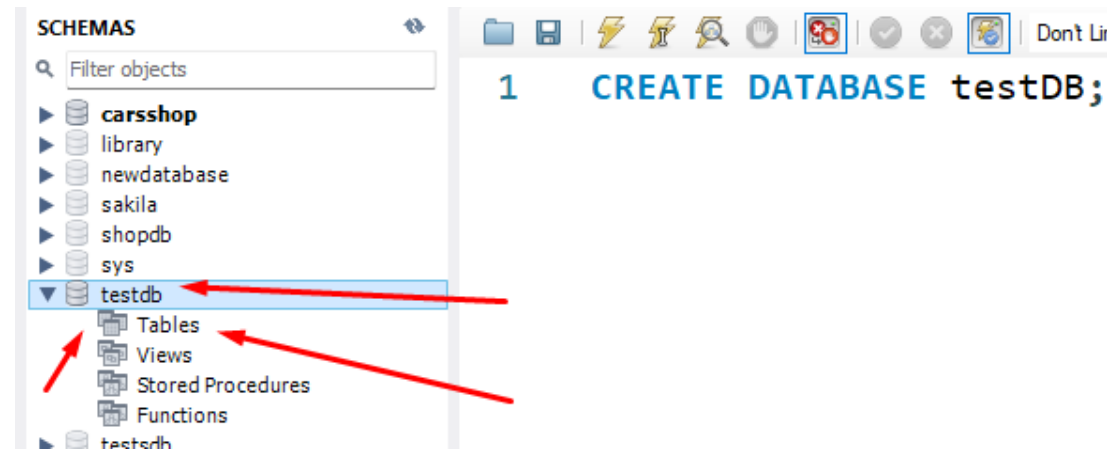
DROP – видалення існуючих сутностей

Бази даних та SQL

CREATE DATABASE

Інструкція **CREATE DATABASE** використовується для створення нової бази даних SQL.

CREATE DATABASE *databasename*



Порада: переконайтеся, що у вас є права адміністратора, перш ніж створювати будь-яку базу даних.



Бази даних та SQL

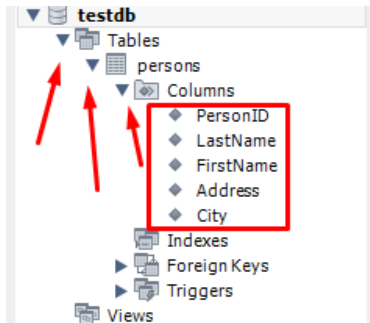
CREATE TABLE

Оператор **CREATE TABLE** використовується для створення нової таблиці в базі даних.

```
CREATE TABLE table_name (  
column1 datatype,  
column2 datatype,  
column3 datatype,  
.... );
```

Параметри стовпців задають імена стовпців таблиці.

Параметр datatype визначає тип даних, які може містити стовпець (наприклад, varchar, integer, date тощо).



```
5 • CREATE TABLE persons(  
6   PersonID int,  
7   LastName varchar(255),  
8   FirstName varchar(255),  
9   Address varchar(255),  
10  City varchar(255)  
11 );
```

Стовпець PersonID має тип int і містить ціле число. Стовпці Прізвище, Ім'я, Адреса та Місто мають тип varchar і містять символи, а максимальна довжина цих полів становить 255 символів.



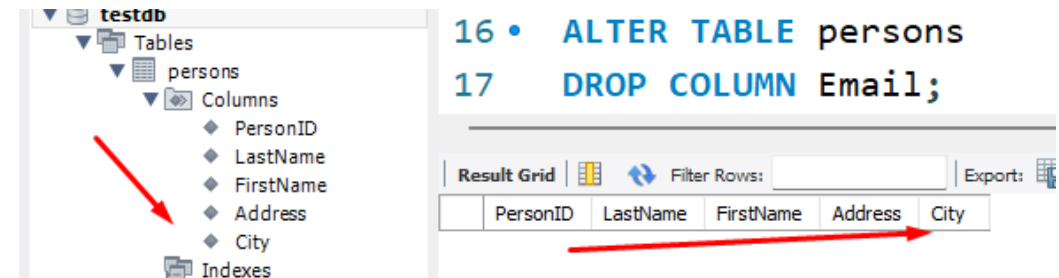
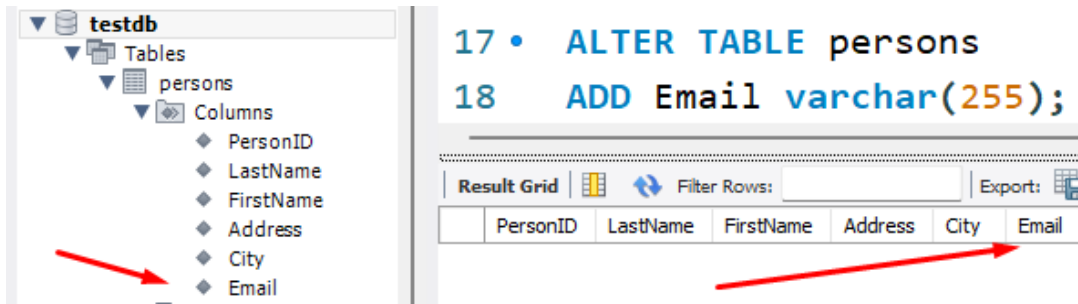
Бази даних та SQL

ALTER TABLE

Оператор **ALTER TABLE** використовується для додавання, видалення або зміни стовпців у існуючій таблиці.

ALTER TABLE *table_name*
ADD *column_name* *datatype*;

ALTER TABLE *table_name*
DROP COLUMN *column_name*;



Бази даних та SQL

ALTER TABLE

Оператор **ALTER TABLE** використовується для додавання, видалення або зміни стовпців у існуючій таблиці.

```
ALTER TABLE table_name  
RENAME COLUMN old_name to new_name;
```

The screenshot shows a database management interface. On the left, a tree view displays the database structure: 'testdb' contains a table 'persons', which has columns 'PersonID', 'LastName', 'FirstName', 'Address', and 'ChanggeField'. A red arrow points to 'ChanggeField'. On the right, the SQL editor shows two lines of code: '17 • ALTER TABLE persons' and '18 RENAME COLUMN City to ChanggeField;'. Below the editor, a 'Result Grid' shows the table structure with columns: 'PersonID', 'LastName', 'FirstName', 'Address', and 'ChanggeField'. A red arrow points to the 'ChanggeField' column in the result grid.

```
17 • ALTER TABLE persons  
18 RENAME COLUMN City to ChanggeField;
```

PersonID	LastName	FirstName	Address	ChanggeField
----------	----------	-----------	---------	--------------



Бази даних та SQL

DROP DATABASE

Інструкція **DROP DATABASE** використовується для видалення існуючої бази даних SQL.

```
DROP DATABASE databasename;
```

Наступний оператор SQL видаляє існуючу базу даних "testDB":

```
DROP DATABASE testDB;
```



Примітка. Будьте обережні, перш ніж видаляти базу даних. Видалення бази даних призведе до втрати повної інформації, що зберігається в базі даних!



Бази даних та SQL

DROP & TRUNCATE TABLE

Інструкція **DROP TABLE** використовується для видалення існуючої таблиці в базі даних.

DROP TABLE table_name

DROP TABLE Persons

Оператор **TRUNCATE TABLE** використовується для видалення даних усередині таблиці, але не самої таблиці.

TRUNCATE TABLE table_name

TRUNCATE TABLE Persons



Примітка. Видалення таблиці призведе до втрати повної інформації, що зберігається в таблиці!



Цілісність даних

Бази даних та SQL

Цілісність даних

Цілісність бази даних – властивість бази даних, що забезпечує коректність і несуперечність даних, що зберігаються в будь-який момент часу.

Цілісність БД не гарантує достовірності інформації, що міститься в ній, але забезпечує принаймні правдоподібність цієї інформації, відкидаючи свідомо неймовірні, неможливі значення.

Цілісність бази даних забезпечується обмеженнями на дані, що вводяться користувачами.

Бази даних та SQL

Типи цілісності

Типи цілісності бази даних:

- 1) доменна цілісність;
- 2) цілісність сутностей;
- 3) посилальна цілісність.

Бази даних та SQL

Цілісність сутностей

Цілісність сутностей гарантує унікальність записів у таблицях (сутностях).

Обмеження, що забезпечують цілісність сутностей:

1. Використання обмежень первинного ключа (**Primary Key**).
2. Використання обмежень на унікальність (**UNIQUE**).

Бази даних та SQL

Доменна цілісність


Доменна цілісність гарантує наявність у деякому стовпці лише допустимих значень.

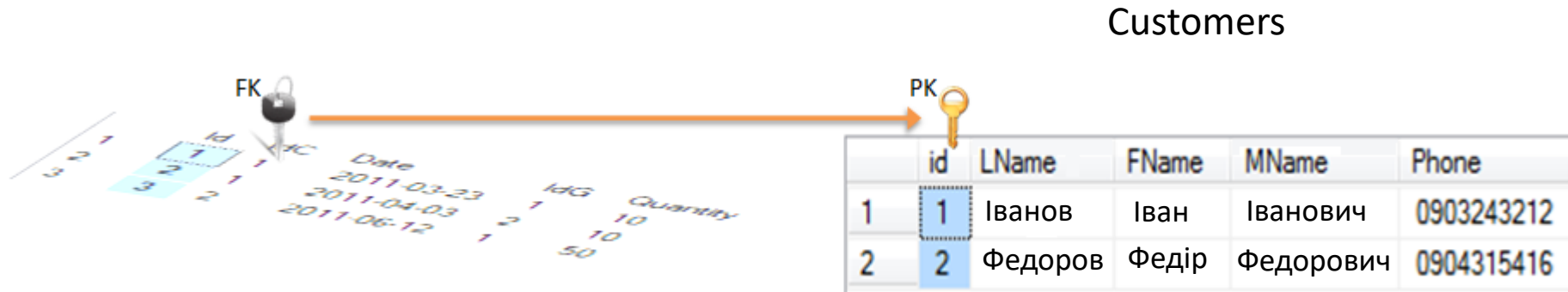
Обмеження, що забезпечують доменну цілісність:

1. Використання обмежень перевірки (**CHECK**).
2. Використання обмежень на замовчування (**DEFAULT**).

Бази даних та SQL

Primary Key

 **Primary Key** (Первинний ключ) – надає посилання для зв'язку з іншими таблицями. Він задає обмеження унікальності для стовпця, на якому задається.



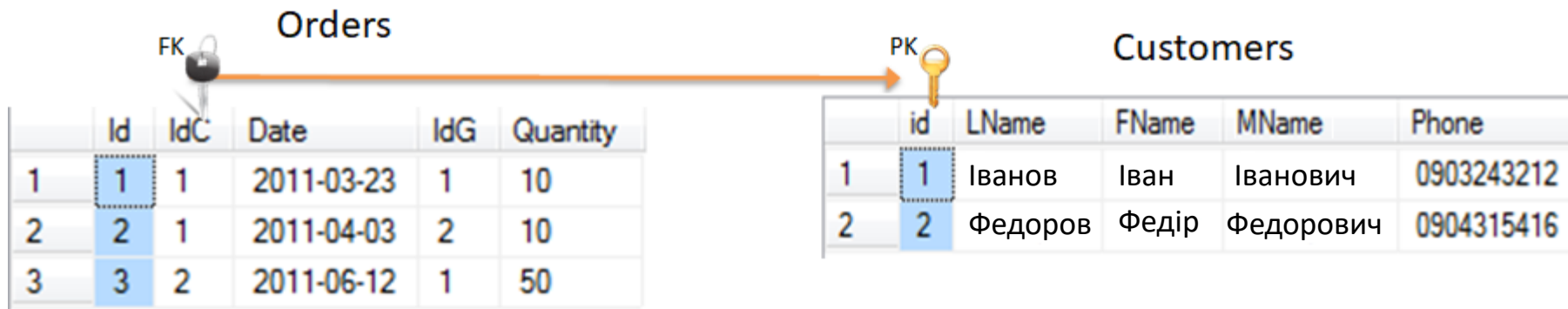
Первинні ключі повинні містити значення UNIQUE і не можуть містити значення NULL. Таблиця може мати лише ОДИН первинний ключ; а в таблиці цей первинний ключ може складатися з одного або кількох стовпців (полів).

Бази даних та SQL

Foreign Key



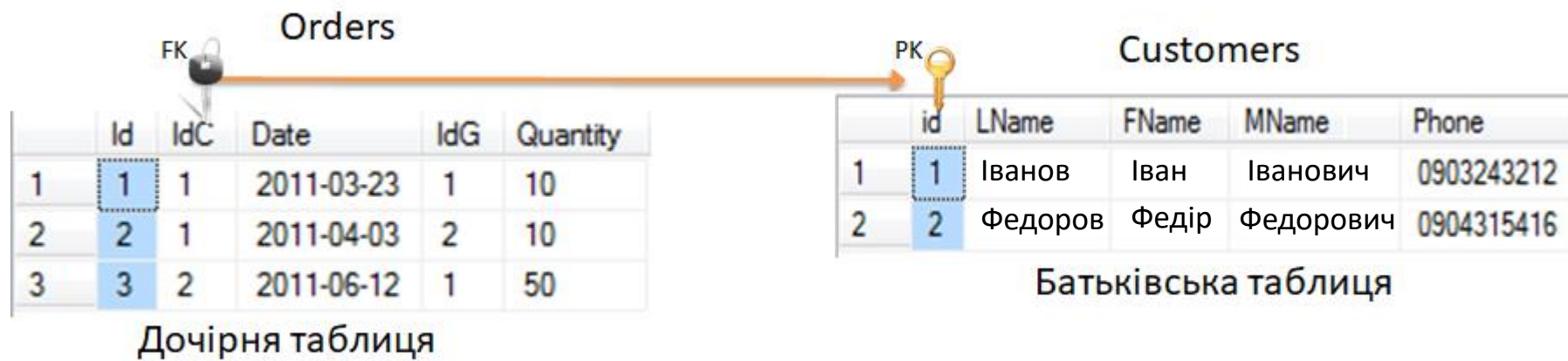
Foreign Key (Зовнішній ключ) – задає стовпець, який посилається на обмеження Primary Key або UNIQUE. Обмеження Foreign Key забороняє вводити дані, які не існують у стовпцях посилання (стовпці з обмеженням Primary Key або UNIQUE).



Бази даних та SQL

Child and Parent tables

Таблиця, яка містить FK, називається дочірньою таблицею за відношенням до таблиці, яка посилається (батьківської таблиці) і містить первинний ключ.



Батьківська таблиця – таблиця, яку посилаються.

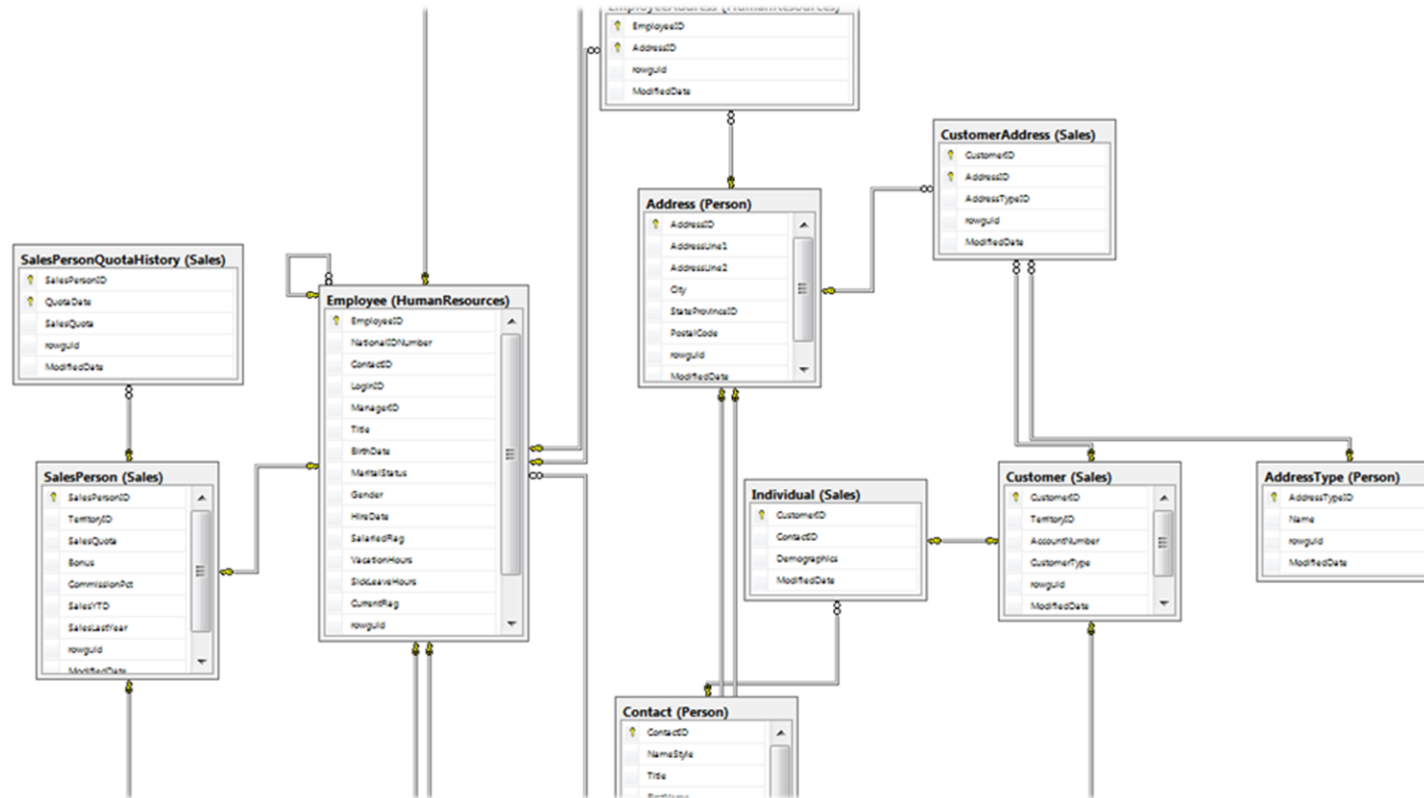
Дочірня таблиця – таблиця, яка посилається.

Бази даних та SQL

Relations

Зв'язок – це певна асоціація між двома таблицями, яка реалізована за допомогою пари FK -> PK або FK -> UNIQUE.

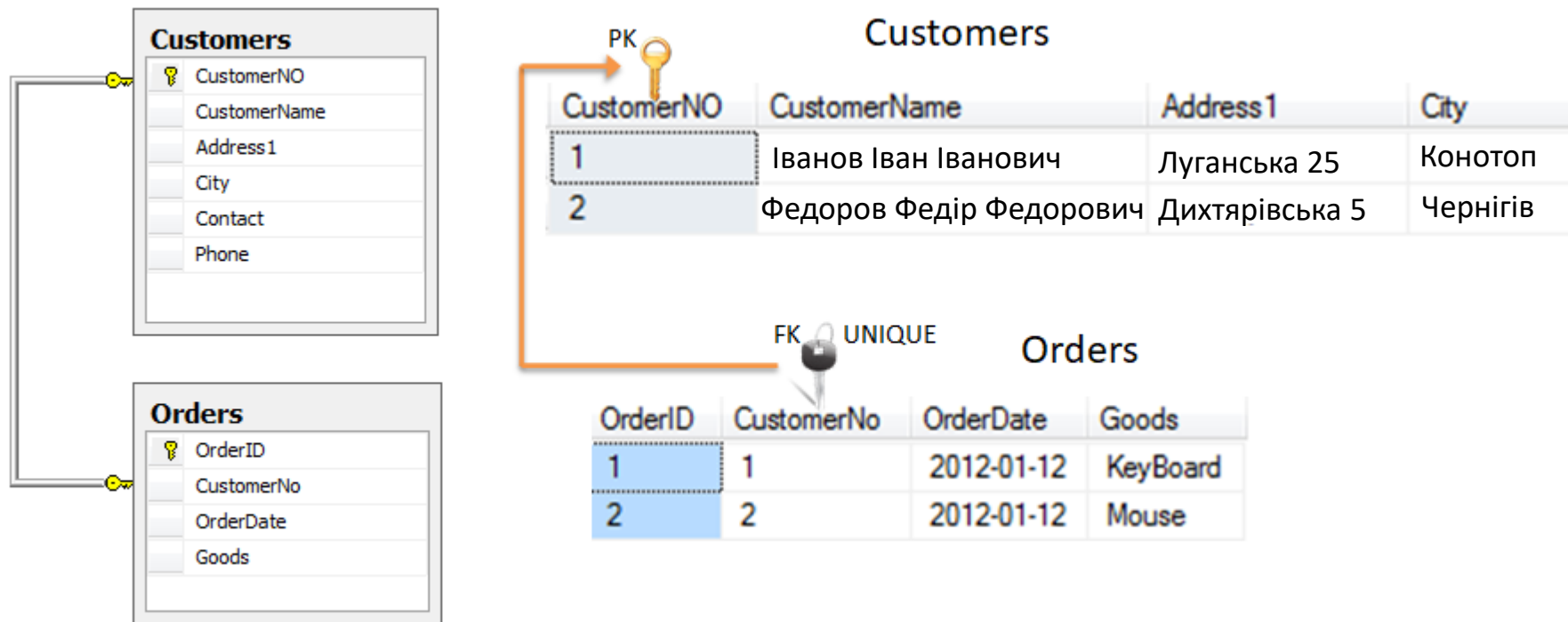
Одна таблиця може бути зв'язана з іншою таблицею або сама із собою.



Бази даних та SQL

Один до одного

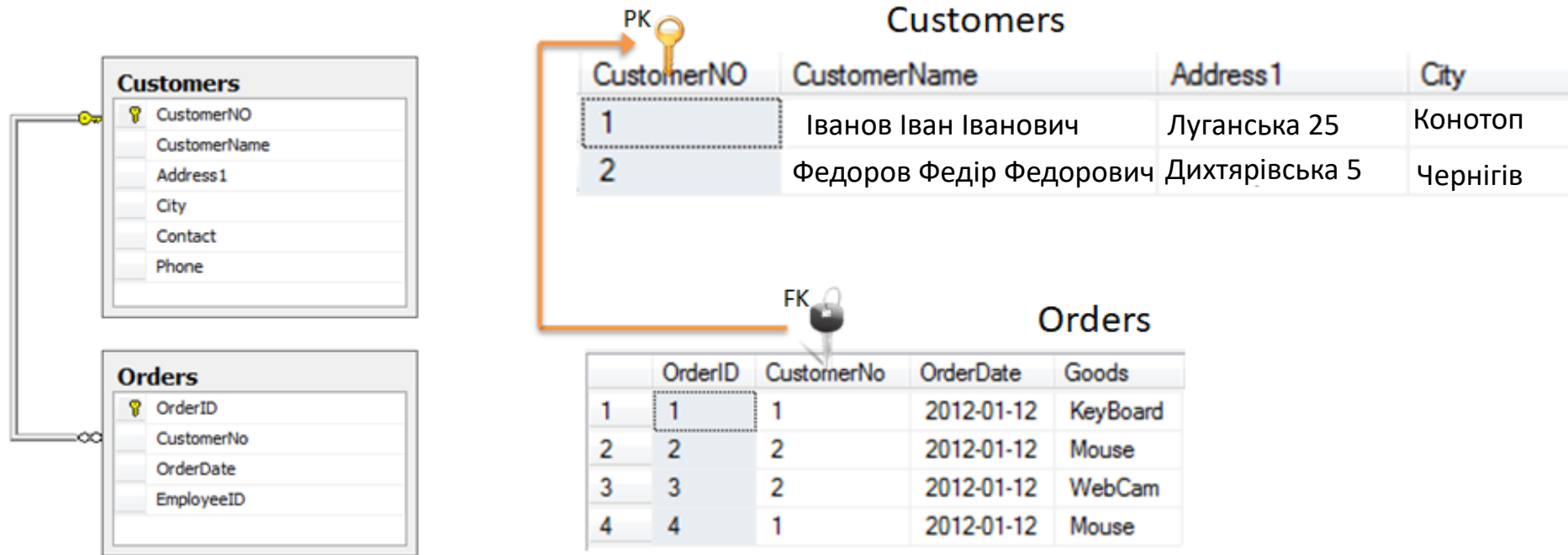
Один-до-одного. Коли кожен елемент у кожній таблиці відображається лише один раз. Наприклад, у кожного працівника може бути лише один автомобіль компанії.



Бази даних та SQL

Один до багатьох

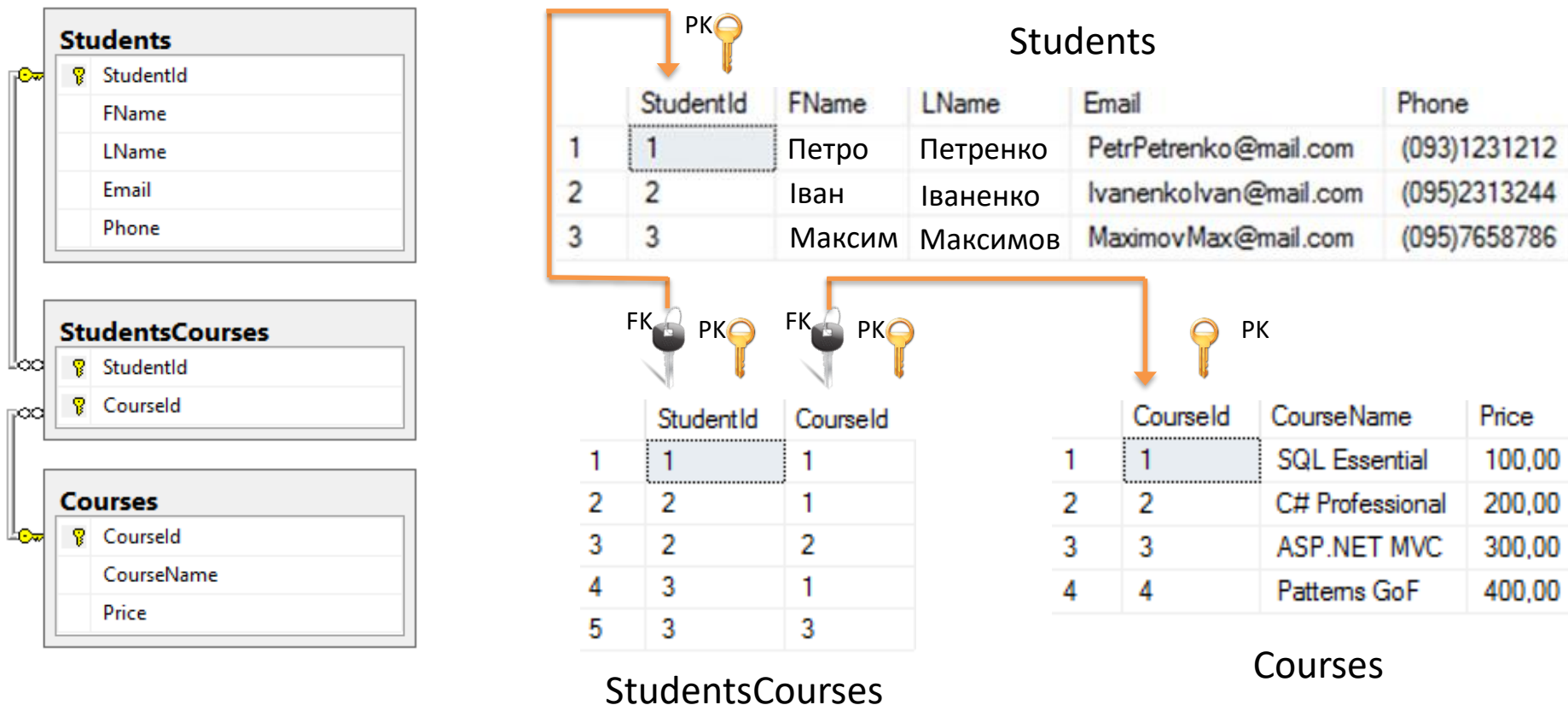
Один-до-багатьох. Коли один елемент в одній таблиці може мати зв'язок із кількома елементами в іншій таблиці. Наприклад, кожне замовлення на придбання може включати кілька продуктів.



Бази даних та SQL

Багато до багатьох

Багато-до-багатьох. Якщо один або кілька елементів в одній таблиці можуть мати зв'язок з одним або більше елементами в іншій таблиці. Наприклад, кожне замовлення може мати кілька товарів, і кожен товар може відображатися в багатьох замовленнях.



Приклади

Бази даних та SQL

Приклад створення PRIMARY KEY

PRIMARY KEY на CREATE TABLE

Наступний SQL створює ПЕРВИННИЙ КЛЮЧ у стовпці "ID" під час створення таблиці "Особи":

```
CREATE TABLE Persons (  
  ID int NOT NULL,  
  LastName varchar(255) NOT NULL,  
  FirstName varchar(255),  
  Age int,  
  PRIMARY KEY (ID)  
);
```



Бази даних та SQL

Приклад створення складного PRIMARY KEY

PRIMARY KEY на CREATE TABLE

Наступний SQL створює ПЕРВИННИЙ КЛЮЧ у стовпці "ID" під час створення таблиці "Особи":

```
CREATE TABLE Persons (  
  ID int NOT NULL,  
  LastName varchar(255) NOT NULL,  
  FirstName varchar(255),  
  Age int,  
  CONSTRAINT PK_Person PRIMARY KEY (ID,LastName)  
);
```



Примітка. У прикладі вище є лише ОДИН ПЕРВИННИЙ КЛЮЧ (PK_Person). Однак ЗНАЧЕННЯ первинного ключа складається з ДВОХ СТОВПЦІВ (ID + Прізвище).



Бази даних та SQL

Приклад створення FOREIGN KEY

Наступний SQL створює ЗОВНІШНИЙ КЛЮЧ у стовпці "PersonID" під час створення таблиці "Orders":

```
CREATE TABLE Orders (  
  OrderID int NOT NULL,  
  OrderNumber int NOT NULL,  
  PersonID int,  
    PRIMARY KEY (OrderID),  
    FOREIGN KEY (PersonID) REFERENCES Persons(PersonID)  
);
```



Manual QA

Що ми сьогодні вивчили

Бази даних та SQL

План уроку

- Основи Data Definition Language (DDL)
 - CREATE
 - ALTER
 - DROP
- Цілісність даних
- Primary Key
- Foreign Key

Manual QA

Підсумки

Що одне, найголовніше, ви дізнались сьогодні?

Manual QA

Дякую за увагу! До нових зустрічей!

Інформаційний відеосервіс для розробників програмного забезпечення

