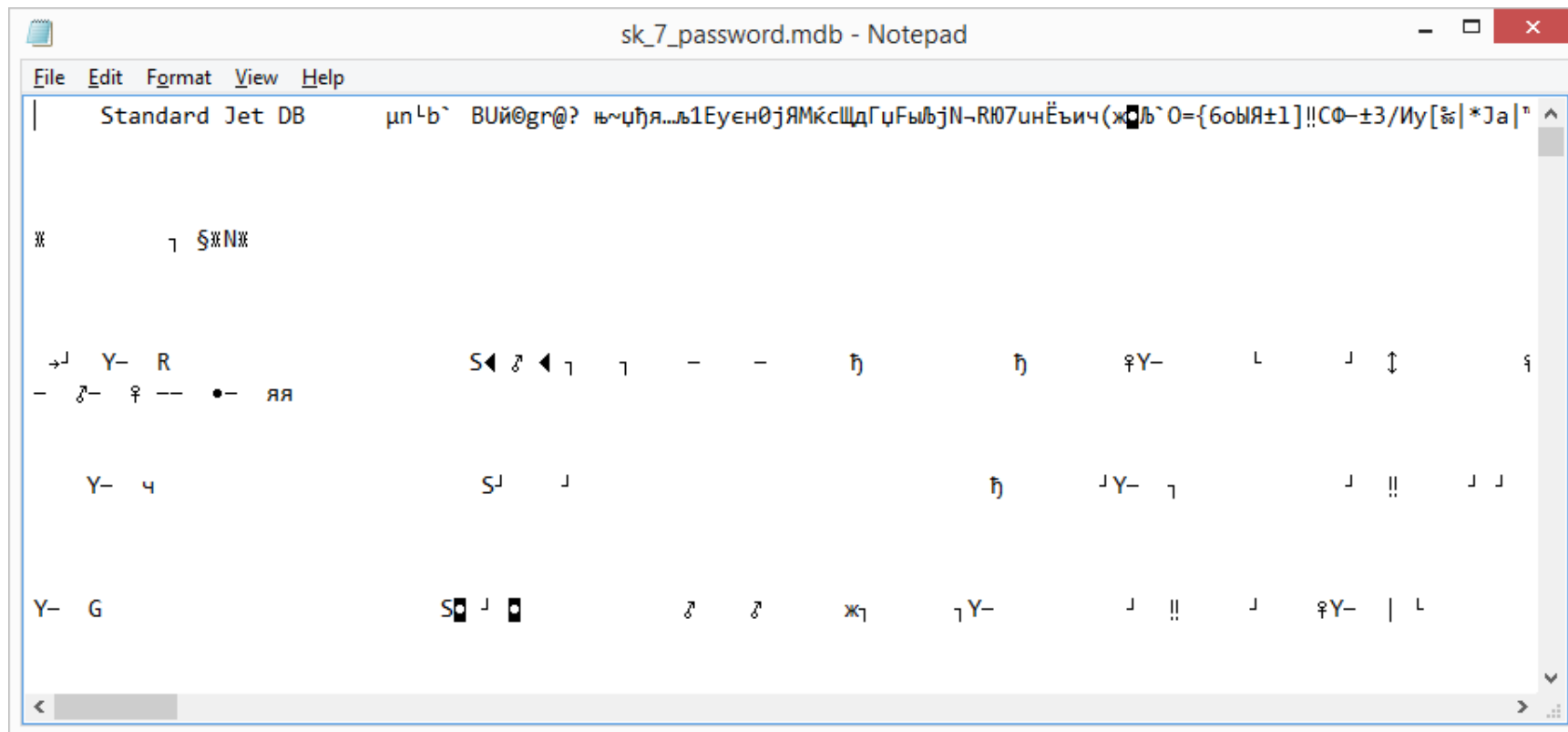


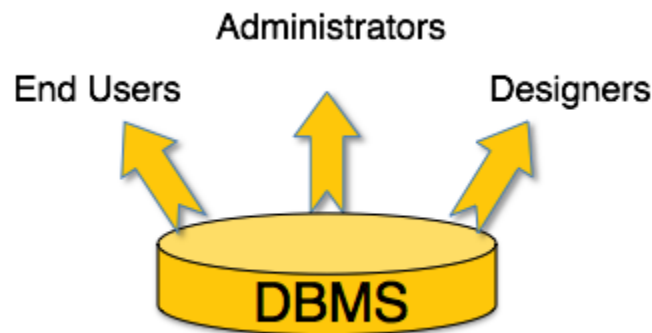
Безпека / Security

- Різні вигляди з різними функціями для різних користувачів / Different views with different features for multiple views
- Обмеження введення та отримання даних / Data entering and retrieving constraints
- СУБД не зберігається на диску як звичайний текстовий файл / DBMS is not saved on the disk as traditional text file



Спроба відкрити файл бази даних як текстовий файл
Trying to open the database file as the text file

Користувачі / Users



- Адміністратори підтримують СУБД і несуть відповідальність за адміністрування бази даних.
- Administrators maintain the DBMS and are responsible for administering the database.

- Дизайнери – це група людей, які безпосередньо працюють над проектуванням бази даних.
- End users are the group of people who actually work on the designing part of the database.
- Кінцеві користувачі – це ті, хто безпосередньо користується перевагами роботи СУБД.
- End users are those who actually reap the benefits of having a DBMS.

Архітектура / Architecture

- Однорівнева / Single-tier

- Багаторівнева / Multi-tier

Пов'язані але незалежні модулі / Related but independent modules

- Модифікувати / Modify
- Доповнювати / Alter
- Змінювати / Change
- Заміняти / Replace

1-рівнева архітектура / 1-tier architecture

Користувач використовує єдиний об'єкт, яким є безпосередньо СУБД

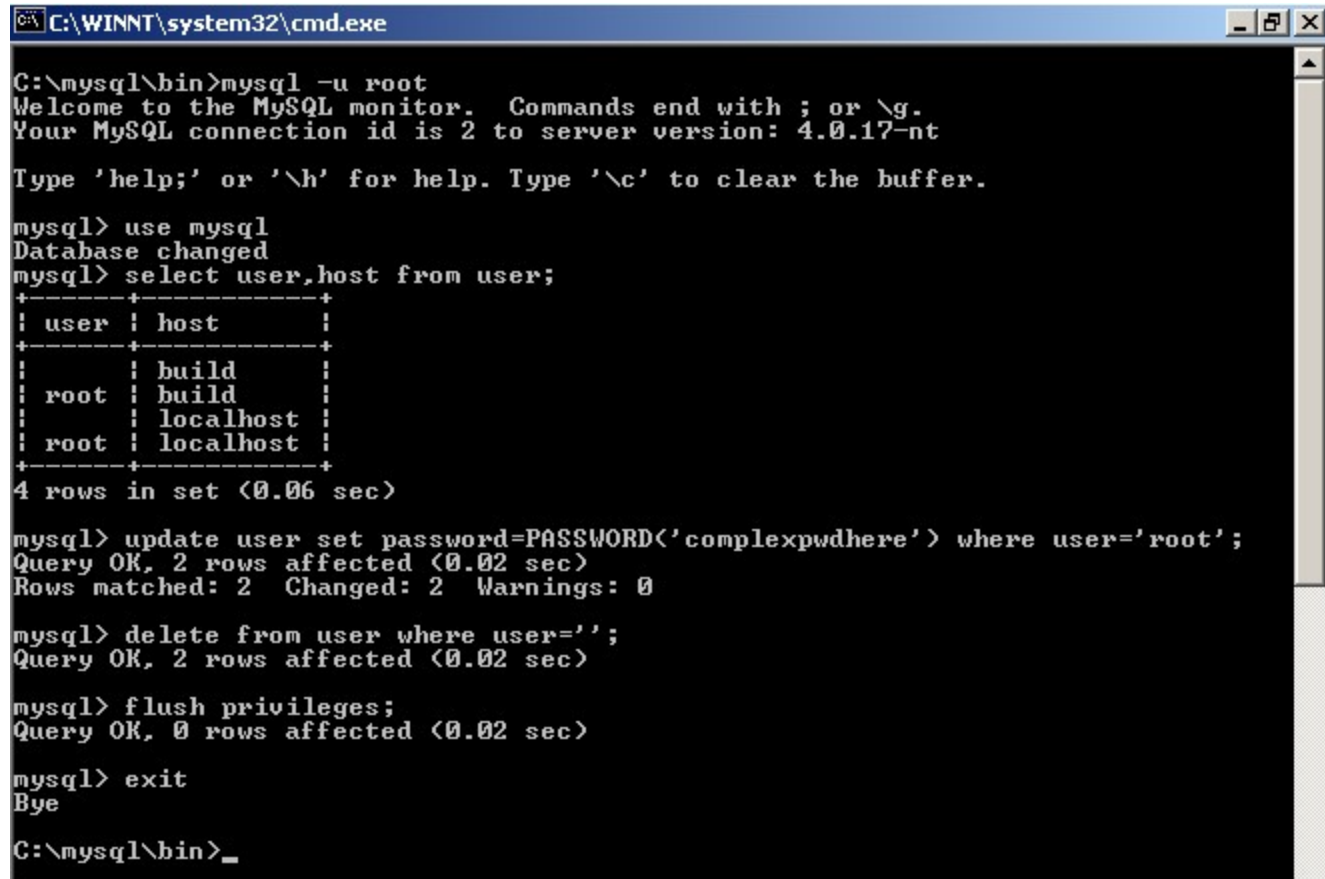
DBMS is the only entity where the user directly sits and uses it

2-рівнева архітектура / 2-tier architecture

СУБД має застосунок, через який можна отримати доступ до СУБД

DBMS must have an application through which the DBMS can be accessed

Tier ?



```
C:\WINNT\system32\cmd.exe

C:\mysql\bin>mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2 to server version: 4.0.17-nt

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> use mysql
Database changed
mysql> select user,host from user;
+-----+-----+
| user | host |
+-----+-----+
|      | build |
| root | build |
|      | localhost |
| root | localhost |
+-----+-----+
4 rows in set (0.06 sec)

mysql> update user set password=PASSWORD('complexpwdhere') where user='root';
Query OK, 2 rows affected (0.02 sec)
Rows matched: 2  Changed: 2  Warnings: 0

mysql> delete from user where user='';
Query OK, 2 rows affected (0.02 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.02 sec)

mysql> exit
Bye
C:\mysql\bin>_
```

<https://www.netikus.net/documents/MySQLServerInstallation/setupsecurity.htm>

Tier ?

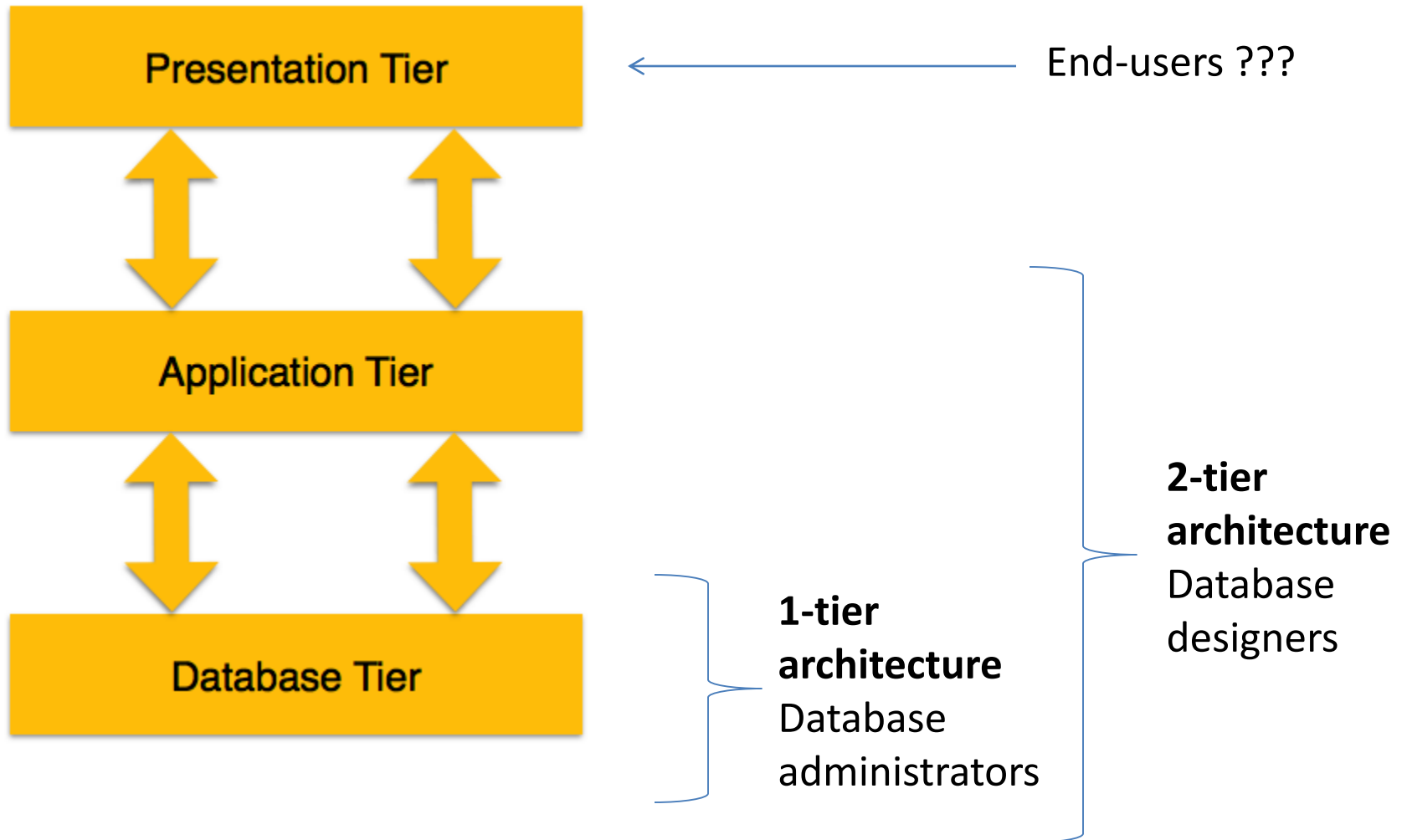
The screenshot displays the phpMyAdmin web interface. On the left, a sidebar shows a tree of databases including 'Usuarios', 'VSet', 'Xss', 'uam', 'uam2', 'ube_db', 'victoria_base', 'vseti', 'world', and 'New'. The 'world' database is selected, and the 'City' table is highlighted. The main panel shows the 'Structure' tab for the 'City' table. It lists five columns: 'ID' (int(11), primary key), 'Name' (char(35), latin1_swedish_ci), 'CountryCode' (char(3), latin1_swedish_ci), 'District' (char(20), latin1_swedish_ci), and 'Population' (int(11)). Below the column list, there are options to 'Check All', 'With selected', and buttons for 'Browse', 'Change', 'Drop', 'Primary', 'Unique', and 'Index'. At the bottom, there is a section for 'Indexes' and a table of 'Space usage' and 'Row Statistics'.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	ID	int(11)			No	None	AUTO_INCREMENT	Change Drop More
2	Name	char(35)	latin1_swedish_ci		No			Change Drop More
3	CountryCode	char(3)	latin1_swedish_ci		No			Change Drop More
4	District	char(20)	latin1_swedish_ci		No			Change Drop More
5	Population	int(11)			No	0		Change Drop More

Space usage	
Data	266.9 KiB
Index	42 KiB
Total	308.9 KiB

Row Statistics	
Format	static
Collation	latin1_swedish_ci
Rows	4,079
Row length	67
Row size	78 B
Next autoindex	4,080
Creation	Apr 03, 2013 at 01:30 PM
Last update	Apr 03, 2013 at 01:30 PM

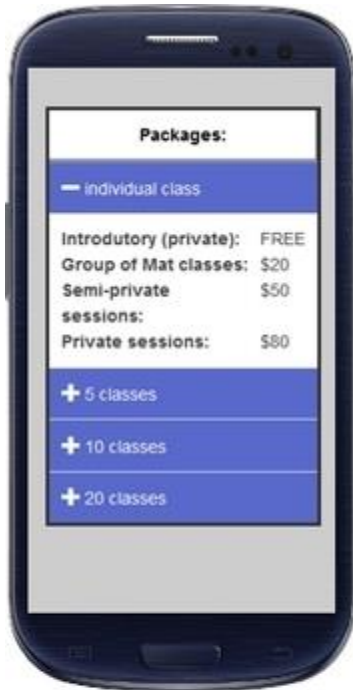
<https://sourceforge.net/projects/phpmyadmin/>



3-рівнева архітектура / 3-tier architecture

- Рівень даних / Database tier
 - Відношення / Relations
 - Мова запитів / Query language
- Рівень застосунків / Application tier
 - Програми, які мають доступ до бази даних / Programs that access the database
 - Посередник між кінцевим користувачем та базою даних / A mediator between the end-user and the database
- Рівень презентації / Presentation tier
 - Програми надають різні перегляди бази даних / Multiple views of the database are provided by applications
 - Кінцеві користувачі не знають про існування бази даних за межами цього рівня / End-users know nothing about any existence of the database beyond this layer

Tier ?



Packages:	Introductory (private)	Group of Mat classes	Semi-private sessions	Private sessions
individual class	FREE	\$20	\$55	\$80
5 classes	\$148	\$95	\$250	\$375
10 classes	-	\$180	\$450	\$700
20 classes	-	\$340	\$860	\$1300

Моделі даних / Data models

- Визначають логічну структуру бази даних, що моделюється
- Define how the logical structure of a database is modeled
- Яким чином дані зв'язуються між собою / How data is connected to each other
- Яким чином дані обробляються та зберігаються у системі / How data is processed and stored inside the system

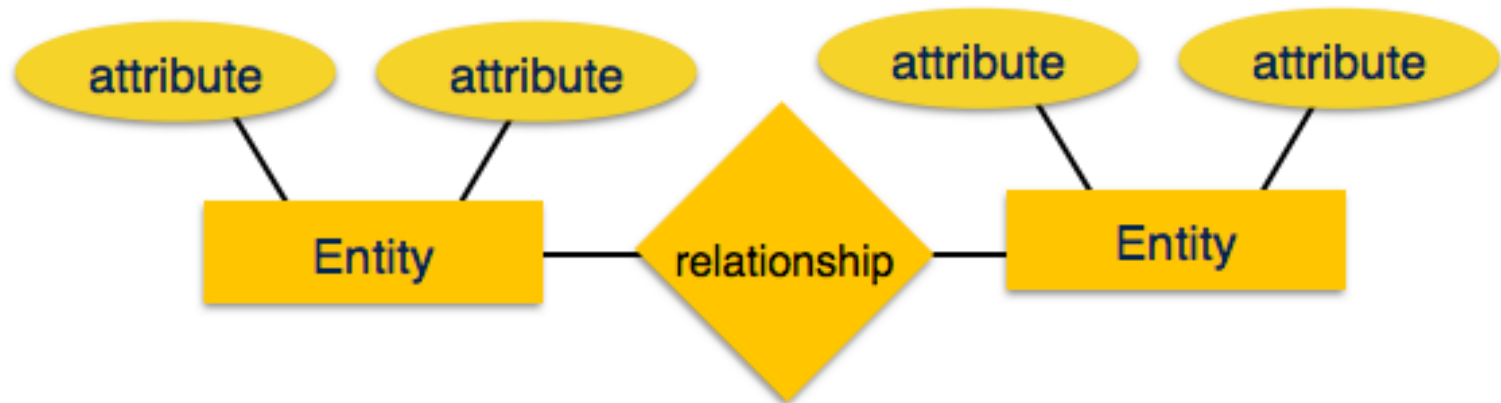
Модель сутність-зв'язок / Entity-relationship model

- Заснована на відомостях про реальні сутності та відношення між ними
- Is based on the notion of real-world entities and relationships among them

Використовується для концептуального проектування бази даних / is used for the conceptual design of a database

ER модель включає / ER model is based on:

- **Сутності** та їх **атрибути** / **Entities** and their **attributes**
- **Відношення** між сутностями / **Relationships** among entities



Кожний атрибут визначається набором значень, який називається **доменом**

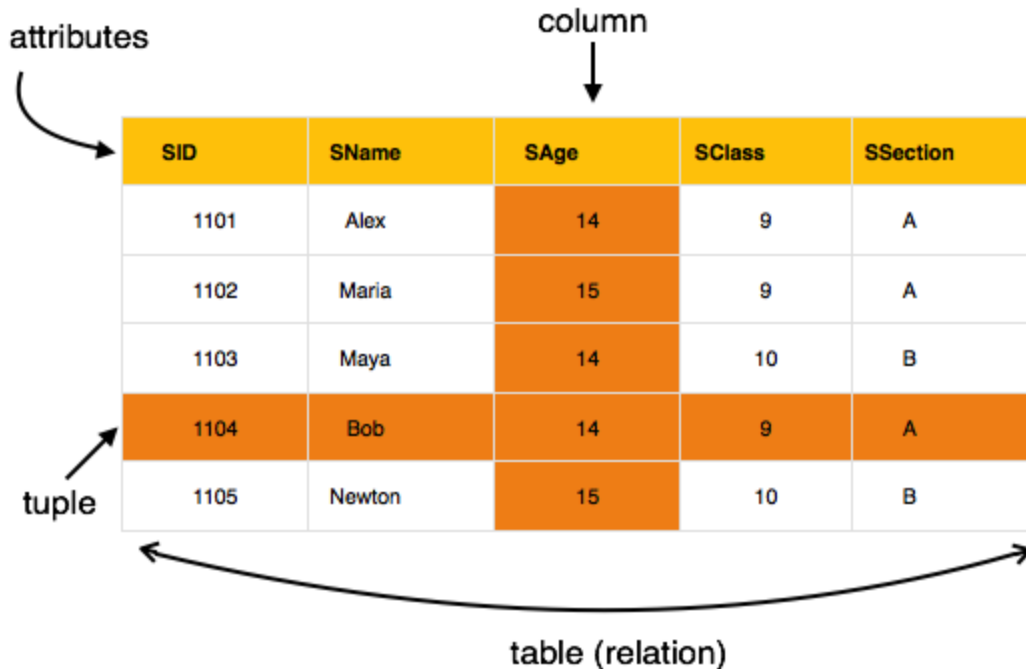
Every attribute is defined by its set of values called **domain**

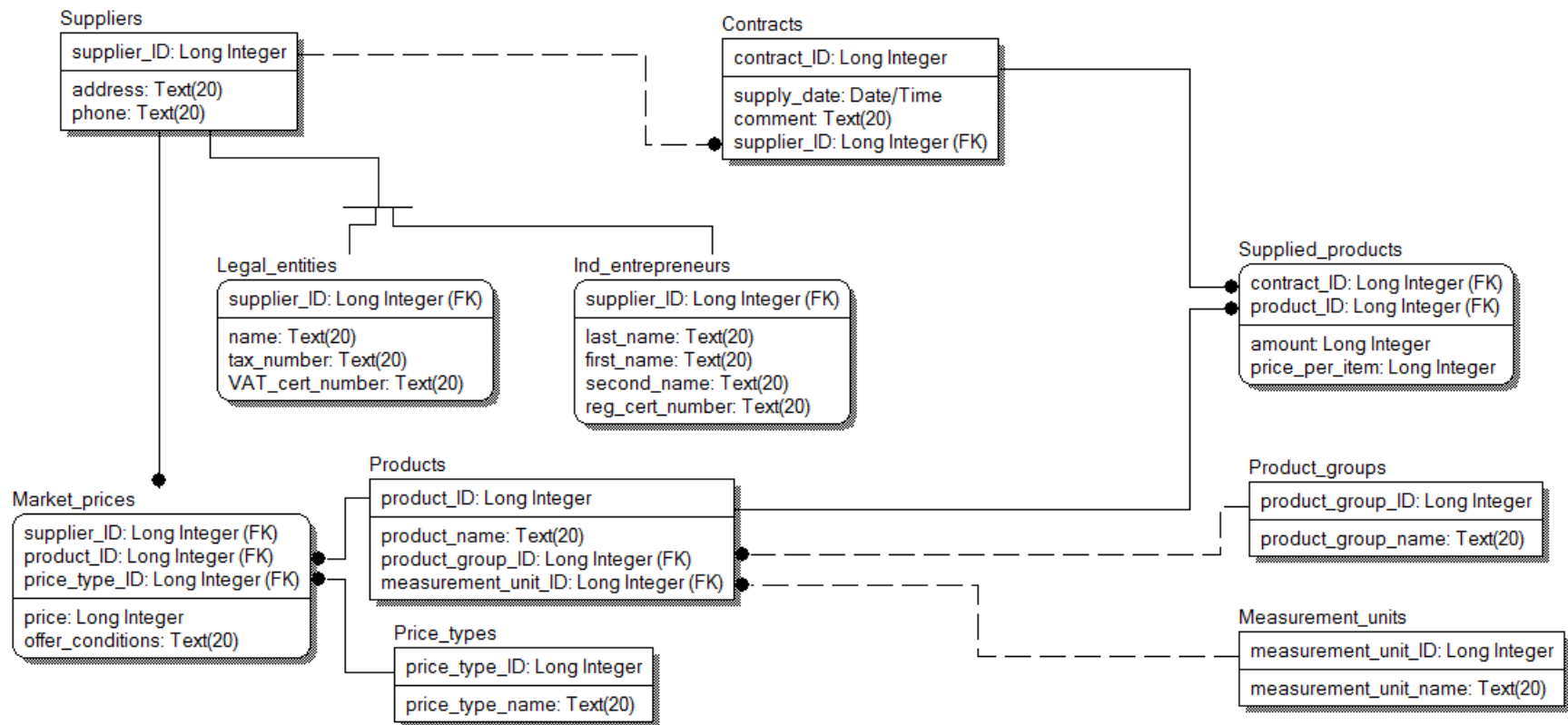
Потужність відношення визначає число асоціацій між двома сутностями
Mapping **cardinalities** define the number of association between two entities

1-to-1, 1-to-many, many-to-1, many-to-many

Реляційна модель / Relational model

- Визначає таблиці у вигляді n-арного відношення
- Defines table as an n-ary relation





Приклад моделі даних з лабораторного практикуму
The example of the data model from lab classes

Основні особливості реляційної моделі / The main highlights of the relational model:

- Дані зберігаються в таблицях – **відношеннях** / Data is stored in tables called **relations**
- Відношення можуть бути **нормалізовані** / Relations can be **normalized**
- У нормалізованих відношеннях збережені значення є **атомарними** / In normalized relations, values saved are **atomic** values
- Кожен рядок у відношенні містить **унікальне** значення / Each row in a relation contains a **unique** value
- Кожен стовпець у відношенні містить значення одного і того ж **домену** / Each column in a relation contains values from a same **domain**

Нотація IDEF1X / IDEF1X notation

Сутність / Entity

Представлення класу реальних або абстрактних речей

The representation of a class of real or abstract things

Домен / Domain

Іменована множина значень даних одного й того ж типу

A named set of data values all of the same data type

Атрибут / Attribute

Властивість або характеристика, яка є загальною для деяких або усіх екземплярів сутності

A property or characteristic that is common to some or all of the instances of an entity

Ключ / Key

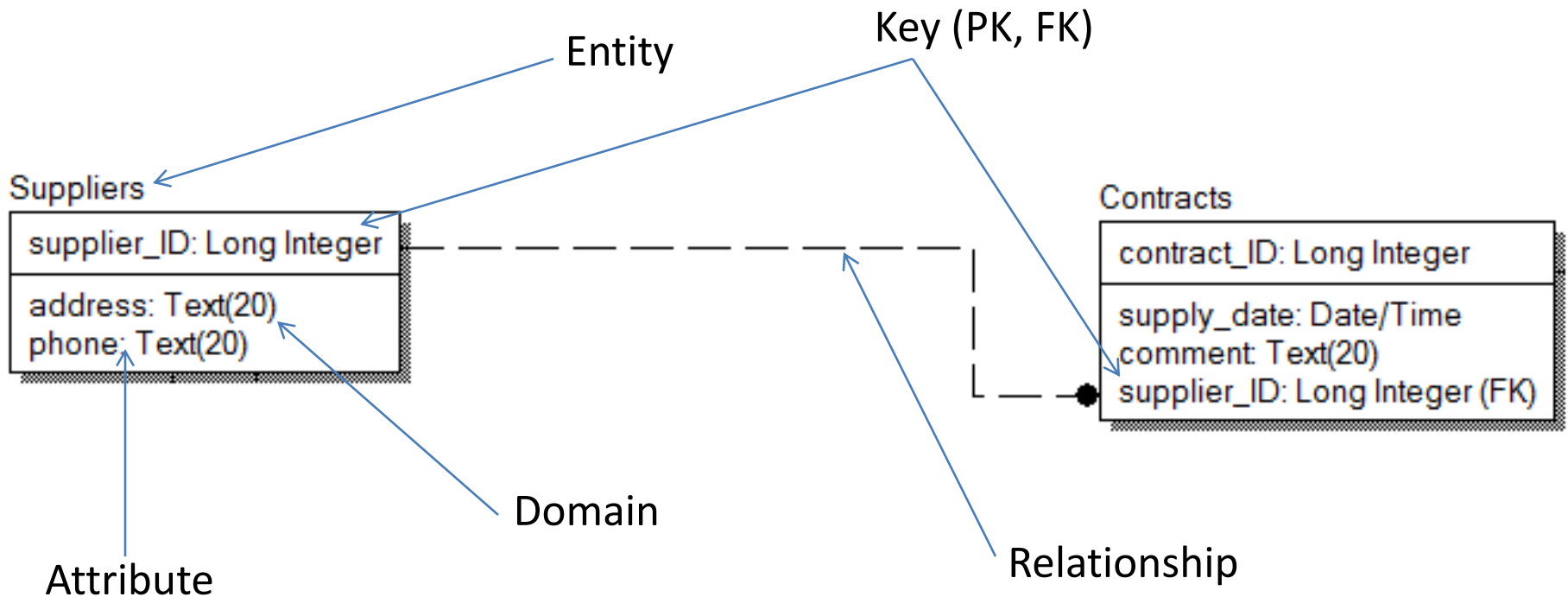
Атрибут або комбінація атрибутів сутності, значення якого однозначно ідентифікує кожен екземпляр сутності. Кожен такий набір являє собою потенційний ключ.

An attribute, or combination of attributes, of an entity whose values uniquely identify each entity instance. Each such set constitutes a candidate key.

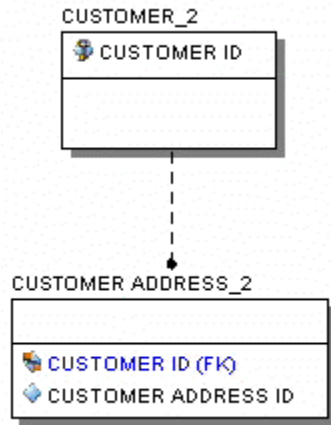
Відношення / Relationship

Асоціація між екземплярами двох сутностей або між екземплярами одної й тієї ж сутності.

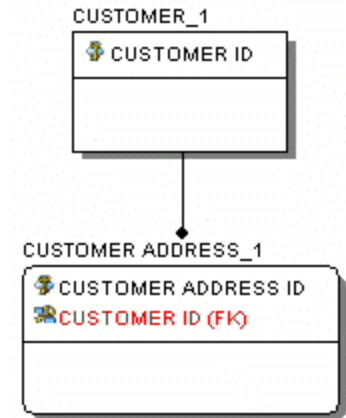
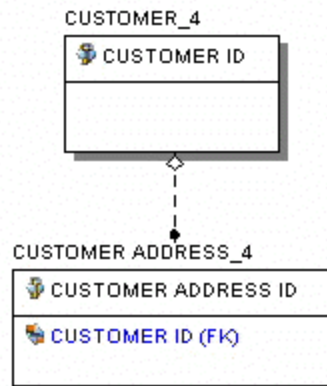
An association between the instances of two entities or between instances of the same entity.



Ідентифікуюче відношення / Identifying relationship



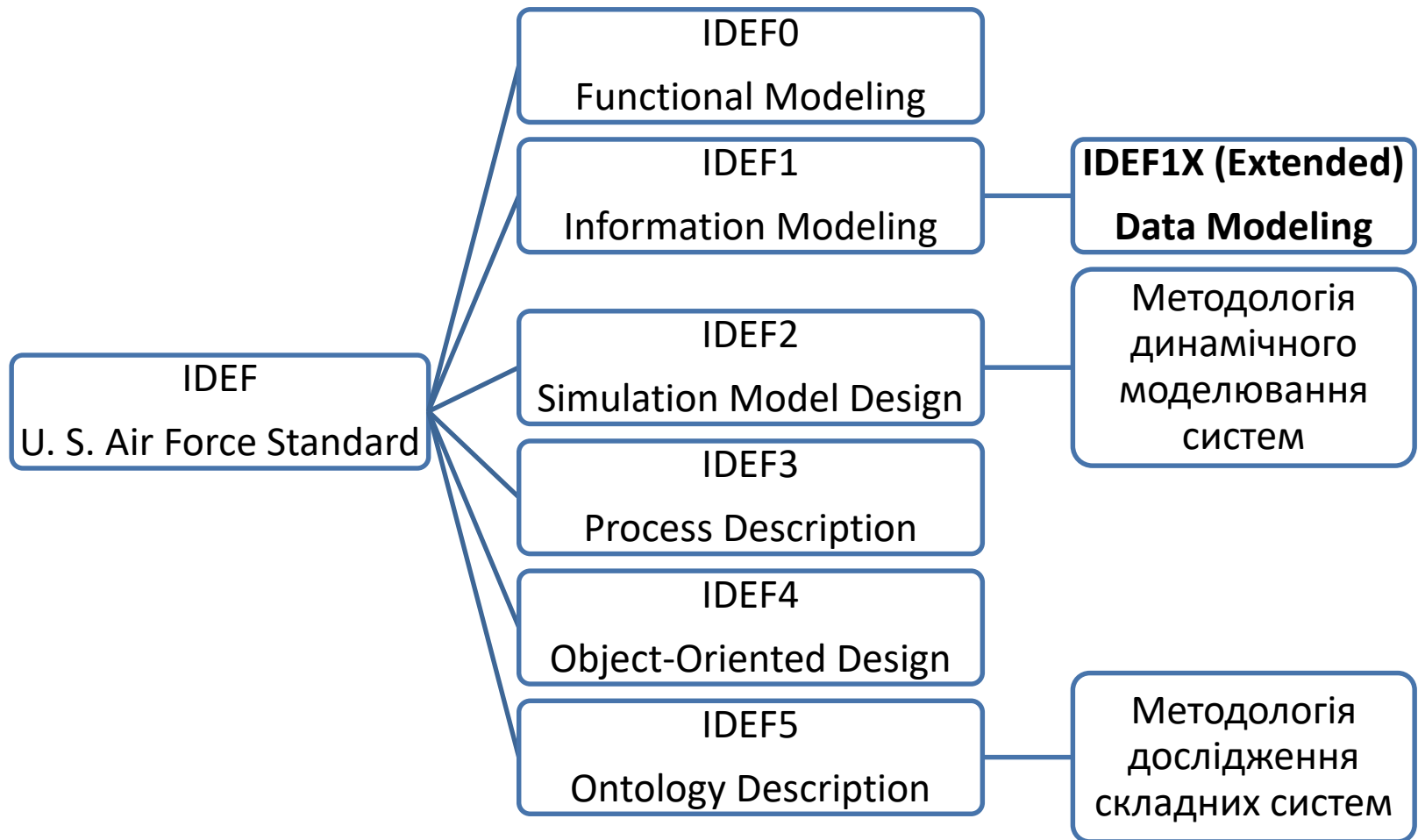
**Non-Identifying, Optional
Relationship**



Неідентифікуюче відношення / Non-Identifying relationship

- обов'язкове / mandatory
- необов'язкове / optional

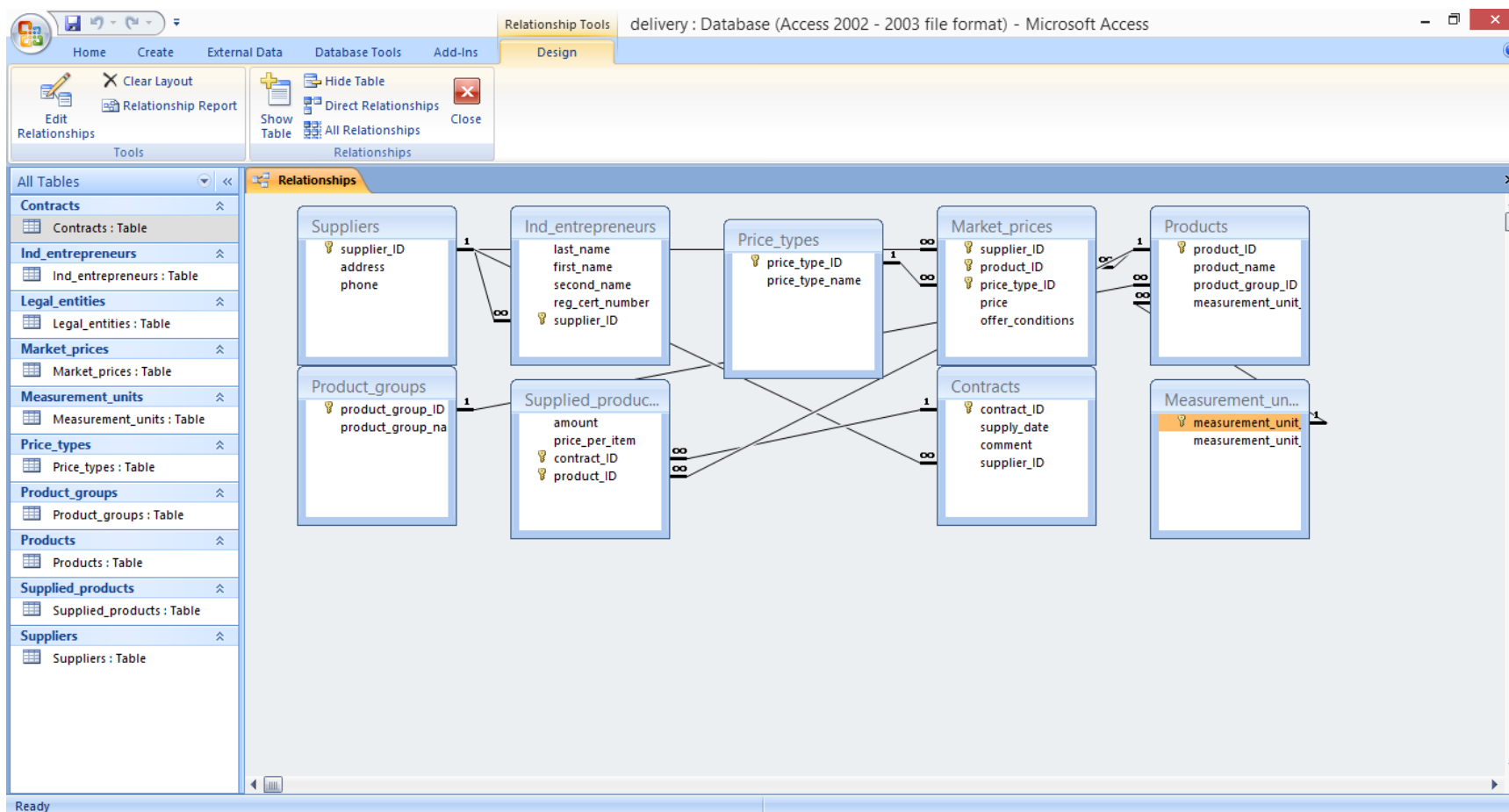
IDEF (Integrated DEFinition)



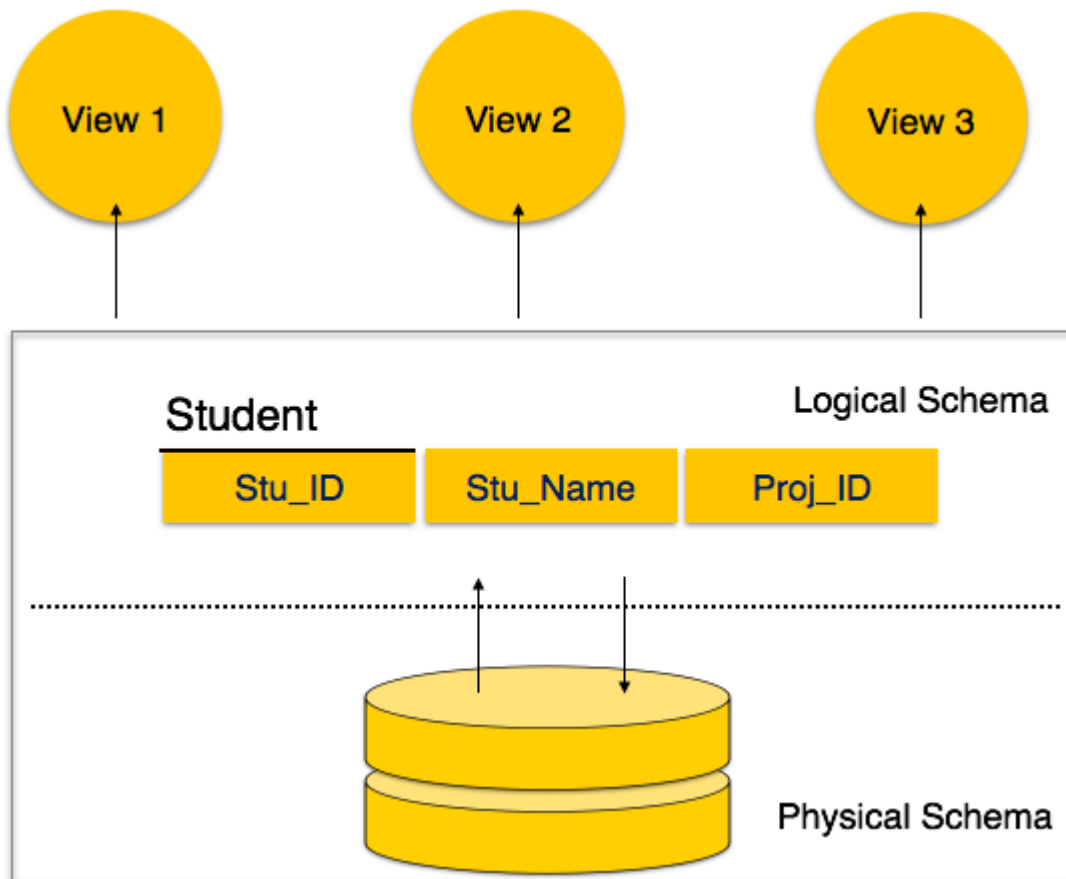
... IDEF6 – IDEF14

Схема бази даних / Database schema

- Схема бази даних – це “скелет” структури, що представляє логічний вигляд усієї бази даних, визначає яким чином організовані дані та якими відношеннями вони зв’язані, визначає усі обмеження, які накладаються на дані
- A database schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized and how the relations among them are associated. It formulates all the constraints that are to be applied on the data.



Приклад схеми даних у СУБД Microsoft Access
The example of the database schema in DBMS Microsoft Access



Таблиці, уявлення та обмеження
цілісності
Tables, views, and integrity
constraints

Фактичне зберігання даних
у формі файлів, індексів тощо
Actual storage of data in the
form of files, indices, etc.

Екземпляр бази даних != Схема бази даних

Database instance != Database scheme

Екземпляр бази даних – стан операційної бази даних, що містить дані, у будь-який час.

Database instance is a state of operational database with data at any given time.

Екземпляри бази даних можуть змінюватись з часом.

Database instances tend to change with time.

Незалежність даних / Data independence

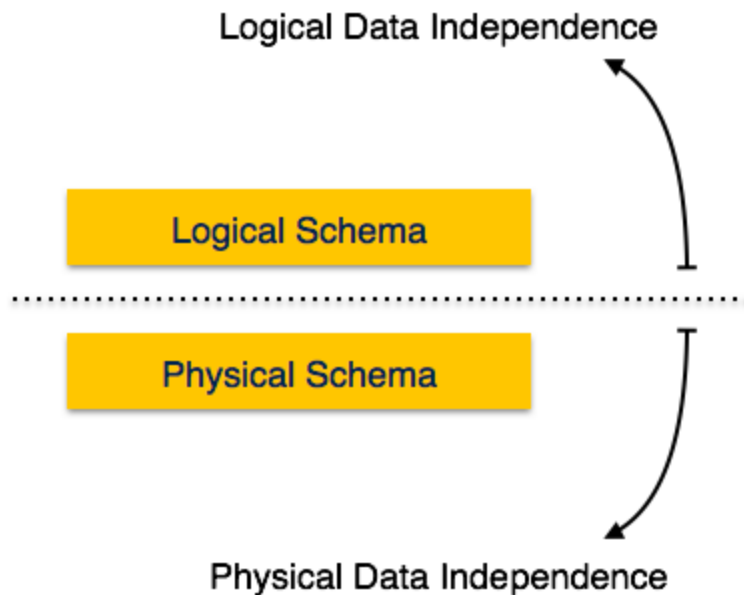
СУБД містить / DBMS contains:
дані користувачів / users' data
метадані / metadata

Логічна незалежність даних – механізм, що відокремлює метадані від фактичних даних, які зберігаються на диску.

Logical data independence is a kind of mechanism, which liberalizes data about database from actual data stored on the disk.

Фізична незалежність даних – механізм, який забезпечує можливість зміни фізичних даних без впливу на схему бази даних або логічні дані.

Physical data independence is the power to change the physical data without impacting the schema or logical data.



Внесення змін у формат таблиці не повинно вплинути на дані, які знаходяться на диску
If we do some changes on table format, it should not change the data residing on the disk

Заміна HDD на SSD не повинна вплинути на логічні дані або схеми
Suppose we want to replace hard-disks with SSD – it should not have any impact on the logical data or schemas