

# Правила Кодда / Codd's rules

Д-р **Едгар Кодд**, після тривалих досліджень щодо реляційної моделі систем баз даних, висунув дванадцять власних правил, які, на його думку, повинна виконувати база даних для того, щоб вважатися справжньою реляційною базою даних.

Dr **Edgar Codd**, after his extensive research on the Relational Model of database systems, came up with twelve rules of his own, which according to him, a database must obey in order to be regarded as a true relational database.

Ці правила можуть бути застосовані до будь-якої системи бази даних, яка управляє збереженими даними, використовуючи лише свої реляційні можливості. Це основне правило, яке є основою для всіх інших правил.

These rules can be applied on any database system that manages stored data using only its relational capabilities. This is a foundation rule, which acts as a base for all the other rules.

# Правило 1: Інформаційне правило / Rule 1: Information

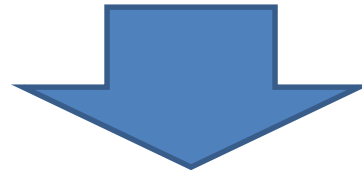
Дані, що зберігаються в базі даних, які можуть бути користувацькими даними або метаданими, повинні бути значенням деякої комірки таблиці. Все в базі даних повинно зберігатися у форматі таблиці.

The data stored in a database, may it be user data or metadata, must be a value of some table cell. Everything in a database must be stored in a table format.

## Console

Press Ctrl+Enter to execute query

```
> SHOW DATABASES
```

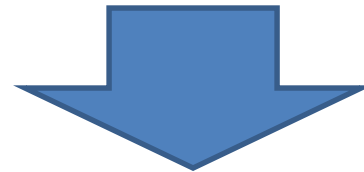


Database
delivery
information_schema
mysql
performance_schema
phpmyadmin
test

```
> SHOW TABLES
```

---

```
>
```



Tables\_in\_delivery

contract

legal\_supplier

private\_supplier

product

supplier

# Правило 2: Гарантований доступ /

## Rule 2: Guaranteed Access

Для кожного елемента даних (значення) гарантований логічний доступ за комбінацією імені таблиці, первинного ключа (значення рядка) та імені атрибута (значення стовпця). Інших засобів, таких як покажчики, не можна використовувати для доступу до даних.

Every single data element (value) is guaranteed to be accessible logically with a combination of table-name, primary-key (row value), and attribute-name (column value). No other means, such as pointers, can be used to access data.

```
USE delivery;  
SELECT supplier.name, supplier.address  
FROM supplier  
WHERE supplier.id = 3
```



**name**

**address**

"Interfrut" LLC Kyiv, Peremohy Av., 154, office 3

## Правило 3: Систематична обробка NULL / Rule 3: Systematic Treatment of NULL Values

Ставлення до значень NULL в базі даних повинне бути систематичним та однорідним. Це дуже важливе правило, оскільки значення NULL може бути інтерпретовано як: дані відсутні, дані невідомі або дані не можуть бути застосовні.

The NULL values in a database must be given a systematic and uniform treatment. This is a very important rule because a NULL can be interpreted as one the following – data is missing, data is not known, or data is not applicable.



# Правило 4: Активний онлайн-каталог

## / Rule 4: Active Online Catalog

Опис структури всієї бази даних має зберігатися в онлайн-каталозі, відомому як словник даних, до якого можуть отримати доступ авторизовані користувачі. Користувачі можуть використовувати ту ж мову запитів для доступу до каталогу, який вони використовують для доступу до самої бази даних.

The structure description of the entire database must be stored in an online catalog, known as data dictionary, which can be accessed by authorized users. Users can use the same query language to access the catalog which they use to access the database itself.

SQL query language statements

Your SQL query has been executed successfully.

SHOW DATABASES

+ Options

**Database**

delivery

information\_schema

mysql

performance\_schema

phpmyadmin

test

Your SQL query has been executed successfully.

SHOW TABLES

+ Options

**Tables\_in\_delivery**

contract

legal\_supplier

private\_supplier

product

supplier

# Правило 5: Повнота підмножини мови / Rule 5: Comprehensive Data Sub-Language

Доступ до БД можна отримати лише за допомогою мови, що має лінійний синтаксис, що підтримує визначення даних, маніпулювання даними та операції з управління транзакціями. Цю мову можна використовувати безпосередньо або за допомогою певної програми. Якщо база даних дозволяє отримати доступ до даних без допомоги цієї мови, то це вважається порушенням.

A database can only be accessed using a language having linear syntax that supports data definition, data manipulation, and transaction management operations. This language can be used directly or by means of some application. If the database allows access to data without any help of this language, then it is considered as a violation.

**WHEN?**

name	address
Ivanov I. I. PE	Kharkiv, Pushkinska Str., 77 (tel. 33-33-44, 12-...
"Interfrut" LLC	Kyiv, Peremohy Av., 154, office 3
Petrov P. P. PE	Kharkiv, Nauky Av., 55, office 108, tel. 32-18-44
"Transservis" LLC	Odesa, Deribasivska Str., 75
Sidorov S. S. PE	Poltava, Svobody Str., 15, apt. 43
John Doe	Kharkiv, Kyrpychova st., 2, 61002

```
START TRANSACTION;
```

```
INSERT INTO supplier (name, address)
VALUES ('John Doe', 'Kharkiv, Kyrpychova st., 2, 61002');
```

```
SELECT name, address FROM supplier;
```

```
ROLLBACK;
```

```
SELECT name, address FROM supplier;
```

**WHEN?**

name	address
Ivanov I. I. PE	Kharkiv, Pushkinska Str., 77 (tel. 33-33-44, 12-...
"Interfrut" LLC	Kyiv, Peremohy Av., 154, office 3
Petrov P. P. PE	Kharkiv, Nauky Av., 55, office 108, tel. 32-18-44
"Transservis" LLC	Odesa, Deribasivska Str., 75
Sidorov S. S. PE	Poltava, Svobody Str., 15, apt. 43

## Правило 6: Оновлення представлення / Rule 6: View Updating

Всі представлення бази даних, які теоретично можуть бути оновлені, також повинні бути придатні до оновлення системою.

All the views of a database, which can theoretically be updated, must also be updatable by the system.

## Правило 7: Вставка, оновлення та видалення високого рівня / Rule 7: High-Level Insert, Update, and Delete

База даних повинна підтримувати вставку, оновлення та видалення високого рівня. Це не повинно обмежуватися одним рядком, тобто вона також повинна підтримувати операції об'єднання, перетину та доповнення для отримання множини записів даних.

A database must support high-level insertion, updation, and deletion. This must not be limited to a single row, that is, it must also support union, intersection and minus operations to yield sets of data records.

```
SELECT * FROM supplier WHERE id < 3  
UNION  
SELECT * FROM supplier WHERE id > 4
```



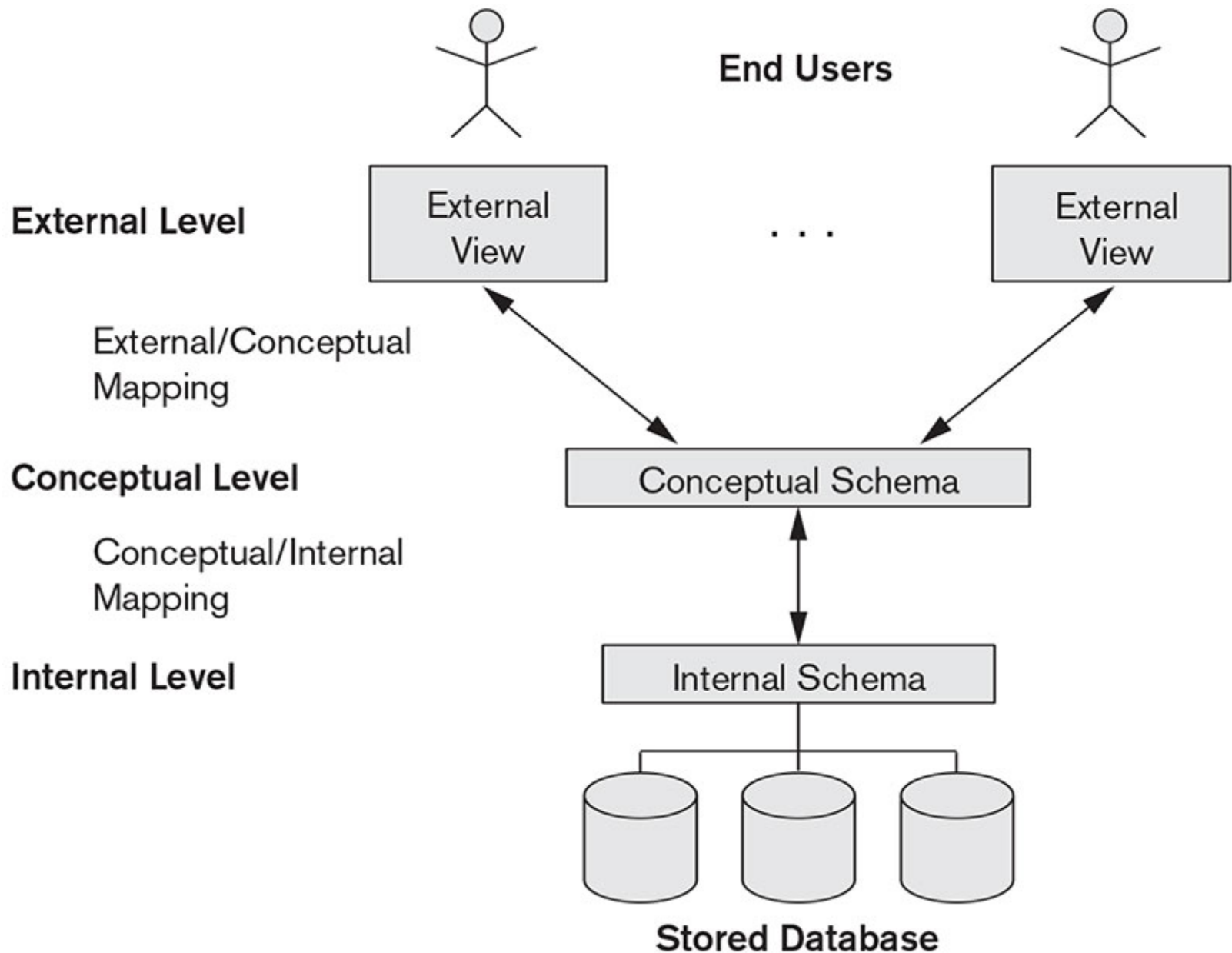
id	name	address
2	Ivanov I. I. PE	Kharkiv, Pushkinska Str., 77 (tel. 33-33-44, 12-...
5	"Transservis" LLC	Odesa, Deribasivska Str., 75
6	Sidorov S. S. PE	Poltava, Svobody Str., 15, apt. 43

## Правило 8: Фізична незалежність даних / Rule 8: Physical Data Independence

Дані, що зберігаються в базі даних, повинні бути незалежними від програм, які мають доступ до бази даних. Будь-які зміни в фізичній структурі бази даних не повинні мати ніякого впливу на те, як зовнішні програми отримують доступ до даних.

The data stored in a database must be independent of the applications that access the database. Any change in the physical structure of a database must not have any impact on how the data is being accessed by external applications.



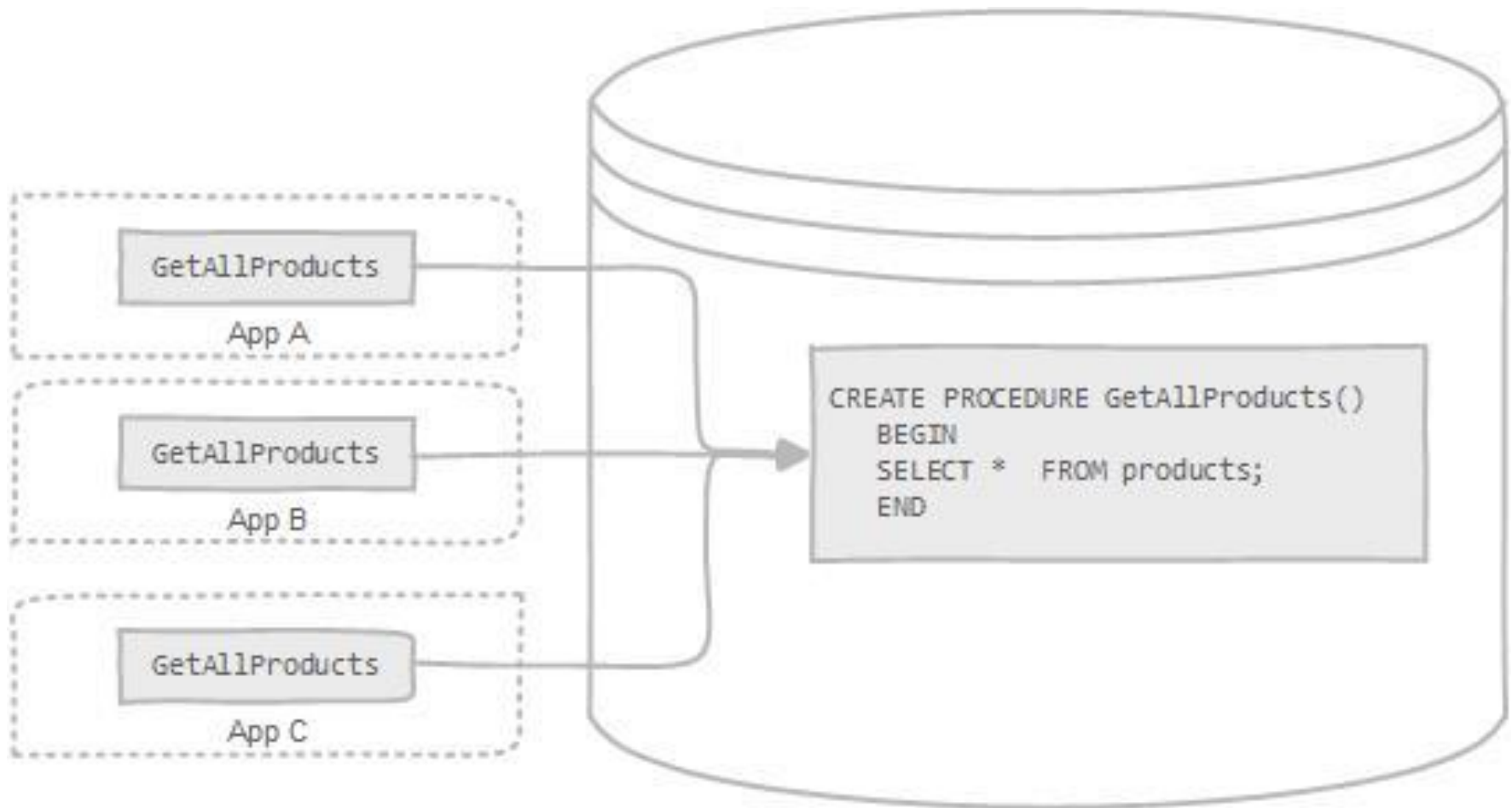


# Правило 9: Логічна незалежність даних /

## Rule 9: Logical Data Independence

Логічні дані в базі даних повинні бути незалежними від перегляду користувача (програми). Будь-яка зміна логічних даних не повинна впливати на програми, що їх використовують. Наприклад, якщо дві таблиці об'єднуються або одна поділяється на дві різні таблиці, користувацькі застосунки не повинні зазнавати впливу або змін. Це одне з найважчих правил щодо виконання.

The logical data in a database must be independent of its user's view (application). Any change in logical data must not affect the applications using it. For example, if two tables are merged or one is split into two different tables, there should be no impact or change on the user application. This is one of the most difficult rule to apply.



## Правило 10: Незалежність контролю цілісності / Rule 10: Integrity Independence

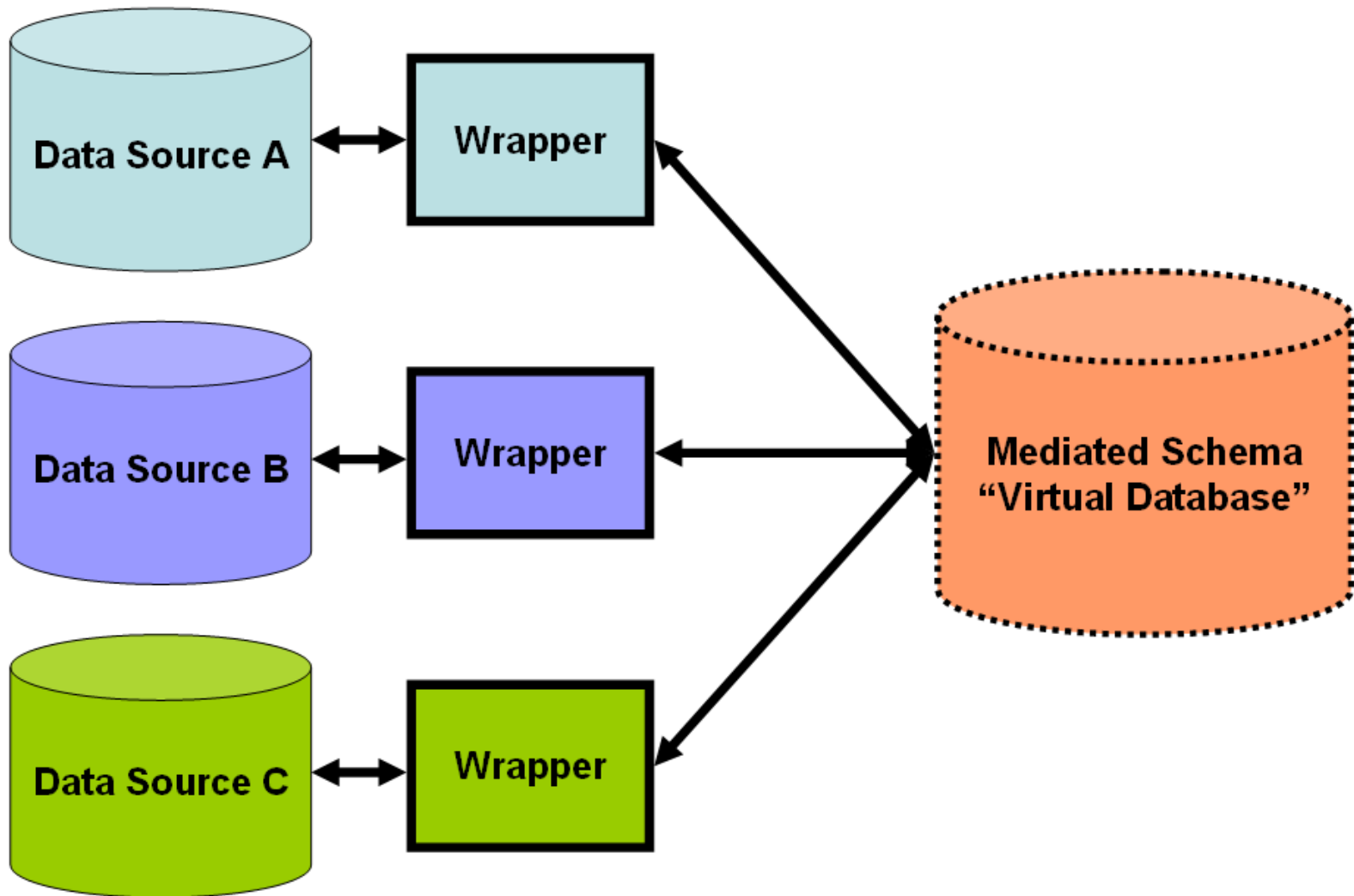
База даних повинна бути незалежною від програми, яка її використовує. Усі її обмеження цілісності можуть бути незалежно модифіковані без необхідності будь-яких змін у додатку. Це правило створює базу даних, незалежну від зовнішньої програми та її інтерфейсу.

A database must be independent of the application that uses it. All its integrity constraints can be independently modified without the need of any change in the application. This rule makes a database independent of the front-end application and its interface.

# Правило 11: Незалежність від розміщення / Rule 11: Distribution Independence

Кінцевий користувач не повинен мати змоги бачити, що дані розподіляються по різних місцях. Користувачам завжди потрібно створити враження, що дані розміщуються лише в одному місці. Це правило розглядається як основа розподілених систем баз даних.

The end-user must not be able to see that the data is distributed over various locations. Users should always get the impression that the data is located at one site only. This rule has been regarded as the foundation of distributed database systems.



## Правило 12: Узгодженість мовних рівнів / Rule 12: Non-Subversion

Якщо в системі є інтерфейс, що забезпечує доступ до записів низького рівня, то інтерфейс не повинен мати змоги зламати систему та обходити обмеження безпеки та цілісності.

If a system has an interface that provides access to low-level records, then the interface must not be able to subvert the system and bypass security and integrity constraints.