

Пример за реализация на разпределена база от данни в СУБД, което не поддържа разпределеност

Ирена Вълва

Example for implementation of distributed database in the not distributed DBMS: This paper presents the process of design and implementation of an example distributed database model and application architecture for distributed database in DBMS that does not support distribution. This application can be useful in teaching and demonstrating these subjects for the students in this area.

Key words: Information Systems and Technologies, Databases, Distributed Databases.

1. ВЪВЕДЕНИЕ

Основната причина за разработката на системи за работа с бази от данни (БД) е стремежът към обединяване на цялата обработвана от дадена организация информация в едно единно цяло и осигуряване на контролируем достъп до нея. Макар че това обединение на информацията и контролиране на достъпа да предполагат централизация не това е крайната цел. На практика, създаването на компютърните мрежи води до децентрализирана обработка на данните. Децентрализираният подход е логическо отражение на организационната структура на много компании и организации, които се състоят от отделни подразделения, отдели, проектни групи и прочее, физически разпределени по различни офиси, отдели, предприятия или филиали, като при това всяка отделна производствена единица работи със свое собствено множество от данни [1,3,4]. Появата и развитието на разпределените бази от данни (РБД) е естествен процес, който следва и отразява работата в тези организационни структури. РБД дават възможност за общодостъпност на данните, поддържани във всяко подразделение, като при това се осигурява съхранение на тези данни именно там, където те най-често се използват. Подобен подход разширява възможностите за съвместно ползване на информацията, като едновременно с това повишава ефективността на достъпа до нея [5].

РБД е съвкупност от логически свързани по между си съвкупности от разделими данни (и техните описания), които са физически разпределени в дадена компютърна мрежа или множество от БД в разпределена система, което използващите го приложения виждат като една единна база от данни [1].

Системата за управление на Разпределени бази от данни (СУРБД) се състои от логически единна база от данни, разпределена на определен брой фрагменти. Всеки фрагмент се съхранява на един или няколко свързани помежду си в мрежа компютри, работещи под управлението на тази СУБД. Всеки възел може независимо от останалите да обработва заявки от потребителите към локалните данни, както и да обработва данни, физически записани на другите възли. Потребителите работят с РБД чрез съответни приложения и не е необходимо да познават физическата организация на РБД [1,5].

В таблица 1 са дадени предимствата и недостатъците на СУРБД [2]. Известни са и различните стратегии на разпределение, репликиране и копиране на данните в РБД, като предимствата и недостатъците на тези стратегии могат да се видят в [2].

2. ПРИМЕРНА РЕАЛИЗАЦИЯ НА ЕЛЕМЕНТАРНО ПРИЛОЖЕНИЕ

Примерна реализация на елементарно приложение за работа с разпределена база от данни може да бъде web базирана информационна система за обслужване на библиотека, видеотека, книжарница, електронен магазин и т.н. Тъй като всички тези приложения са практически сходни, за настоящия пример е избрана реализацията на приложение за обслужване на видеотека - отдаване на филми под

наем. За реализацията му се използват три сървъра (необходими за отделните части от разпределената база от данни), на които се записват данните. Визуализирането на данните става през браузър, като връзката между сървърите и браузъра се реализира от елементарно приложение, реализирано примерно на PHP в комбинация с HTML (Hypertext Markup Language) и CSS (Cascading Style Sheets).

Таблица 1. Предимства и недостатъци на СУРБД

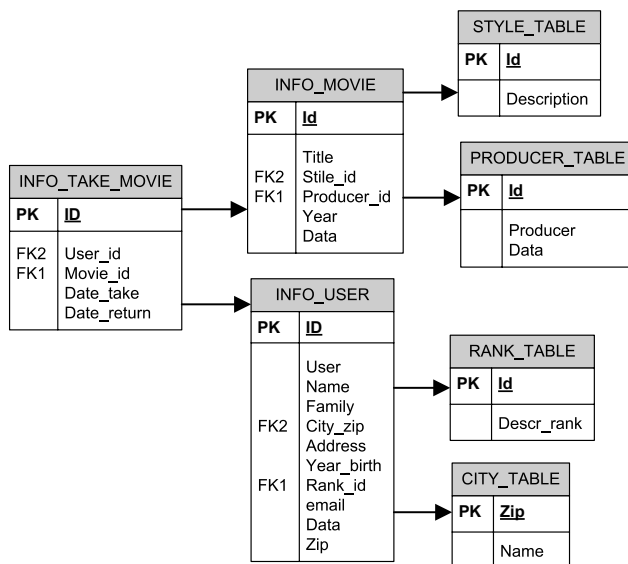
Предимства	Недостатъци
Отразяват точно организационните структури	По-голяма сложност
Разделяемост и локална автономност	Повишаване на цената
Подобрение на достъпността на данните	Повече проблеми за защита на данните
Повишаване на надеждността	Усложнява се управлението на цялостта на данните
Повишаване на производителността	Липса на стандарти
Икономически ползи	Липса на опит
Модулност на системите	Усложняване на процедурите по проектиране на БД

Служителят от видеотека, който работи със системата, има възможност да извършва всички основни операции над БД – да добавя нов филм или нов потребител, да ги изтрива, да прави справка за потребител или филм. Да извежда информация за всички потребители или филми, да извежда информация за всички наети филми и да отбелязва тези, които са върнати. Да прави копие на данните върху избран сървър, както и да изпълнява всякакви сервисни функции над системата

2.1. Модел на базата от данни

При проектиране на РБД са съобразени основните изисквания за нормализация на данните на отделните сървъри и като цяло, както и функционалните изисквания към приложението. В резултат на процеса на проектиране е избран релационният модел, показан на фигура 1. Съобразени са изискванията за цялостност на същностите и връзките и са специфицирани ключовите полета и чуждите ключове за всяка от таблиците [1, 3]. БД се състои от 7 таблици:

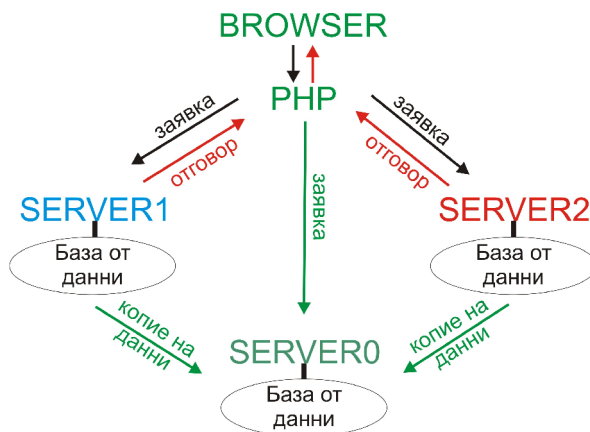
- **Тип на филма** – съдържа жанровете на филмите, много рядко се налага промяна на съдържанието ѝ;
- **Информация за потребител** – съдържа информация за вече регистрираните потребители;
- **Информация за филм** – съдържа данните за наличните филми;
- **Град** – съдържа списък на пощенските кодове и имената на градовете, като веднъж попълнена рядко се налагат корекции над тази таблица;
- **Информация за взети филми** – съдържа данни за взетите филми – кога, от кой потребител, за какъв период;
- **Продуцент** – съдържа списък на продуцентски къщи;
- **Ранк** – тук е записано нивото на потребителя, на базата на което той може да използва или не определени бонуси.



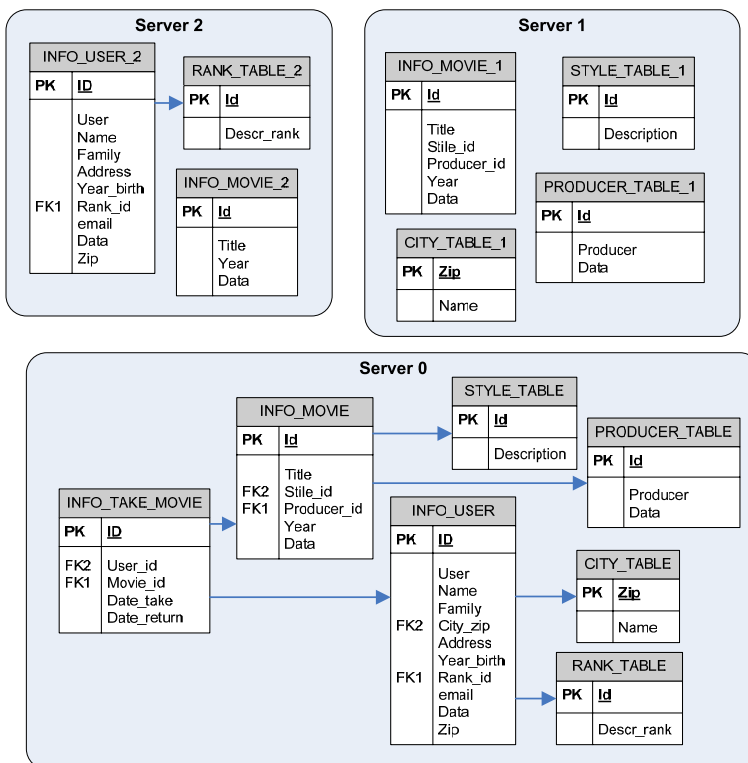
Фиг.1. Реляционен модел

За да бъде РБД, в случая хоризонтално РБД с дублиране на информацията, тези седем таблици са разделени физически на три отделни сървъра (фигура 2 и 3). На фигура 2 са показани информационните и управляващи потоци между отделните сървъри

На сървъри 1 и 2 (условно наречени с тези номера) се намират тези части от съдържанието на показаните на фигура 3 таблици, които се използват най-често от потребителите на съответния сървър. Третата БД е на трети сървър (сървър 0), тя съдържа и седемте таблици, като на определен интервал от време се осъществява синхронизация на информацията на сървър 1 и сървър 2, като данните от тях се дублират на сървър 0.



Фиг.2. Схема на разпределение



Фиг.3. Модел на разпределената база от данни

2.2. Основни функции на примерното приложение :

- Регистриране на нов потребител;
- Въвеждане на нов филм;
- Справка за потребител;
- Справка за филм;
- Списък на всички филми;
- Списък на всички потребители;
- Изтриване на потребител;
- Изтриване на филм;
- Отразяване на нает филм от потребител;
- Потвърждаване на върнат филм от потребител;
- Копиране на данните.

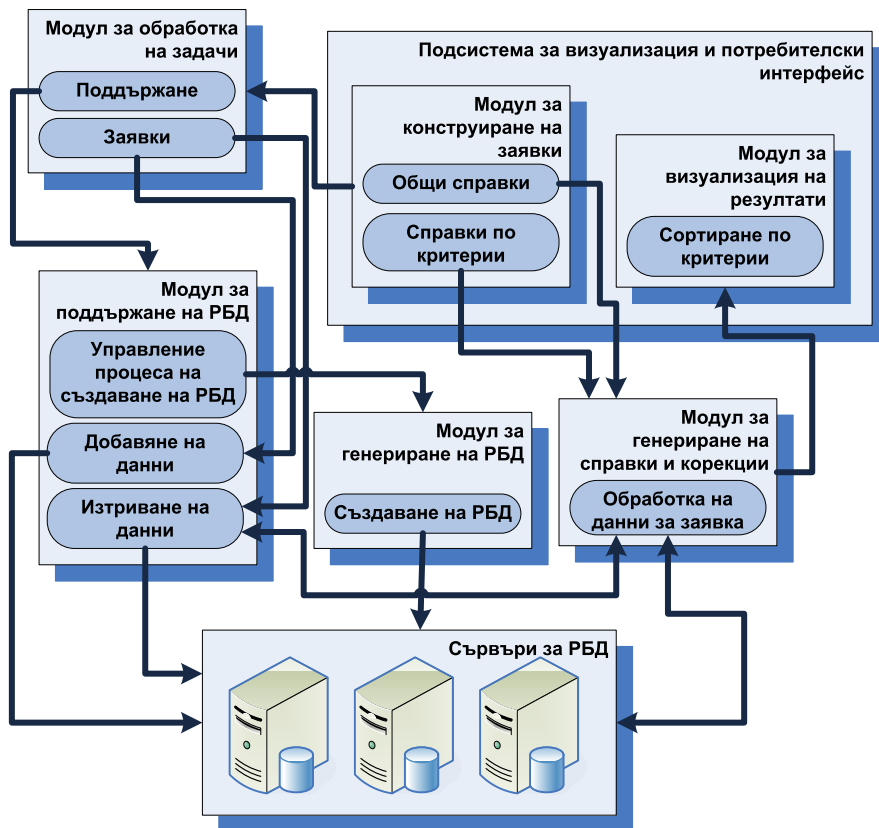
2.3. Архитектура на приложението

Архитектурната схема на примерното приложение е показана на фигура 4. За създаването на РБД и нейното поддържане се грижи основно *Модулът за обработка на задачи*. От него се инициират всички действия, касаещи:

- създаване на РБД - при започване на работа с приложението, като тази операция се извършва само от потребители със съответни права за изпълнението ѝ.

Самото създаване на РБД се осъществява от *Модула за генериране на РБД*, който реализира самото разпределение на данните по сървъри;

- добавяне на нова информация – при регистриране на нов потребител или при доставка на нови филми;
- корекция на съществуващи данни - при заемане или връщане на филми, смяна на данните за клиент;
- изтриване на информация при бракуване на филми или отказ на клиенти.



фиг. 4. Архитектурен модел на приложението

Операциите, свързани с добавяне, корекции и изтриване се управляват и реализират от *Модул за поддържане на РБД*. При програмната реализация на приложението именно в този модул е заложена основната идея за разпределеност и логиката на този модул се грижи за поддържането на сигурността и цялостта на данните по отделните сървъри и като цяло в РБД. Този модул се свързва с модула за *генериране на справки*, с цел осигуряване на точните данни и установяване на точното им местоположение в РБД.

За организиране на потребителския интерфейс и формиране на критериите за заявките към РБД се грижи *Подсистема за визуализация и потребителски интерфейс*. Тя се състои от два основни елемента – *модул за конструиране на*

заявки – който подпомага задаването на критерии за търсените данни и *модул за визуализация на резултати*, който оформя върнатите данни от модула за генериране на справки и корекции по разбираем за крайния потребител начин.

ЗАКЛЮЧЕНИЕ

Това примерно приложение показва, че реализацията и поддържането на РБД в СУБД, които не поддържат разпределеност не е невъзможно. Важно е обаче цялостното управление и поддръжка на РБД да се предвиди и реализира в самото приложение на език от високо ниво. Този вариант на поддържане на разпределеност на БД е приложим за малки реализации, по-скоро с учебна и демонстрационна цели.

ЛИТЕРАТУРА

[1] Ceri, Stefano, Distributed Databases: Principles and Systems, Tata McGraw-Hill, 1st Edition , ISBN: 0070265119

[2] Connolly, Thomas M., Carolyn E. Begg Database Systems: A Practical Approach to Design, Implementation and Management, Addison Wesley; 5 edition, 24 Feb 2009, ISBN: 0321523067.

[3] Date, C.J. An Introduction to Database Systems, 7th edn. Reading, MA: Addison-Wesley, 2000.

[4] Dye, Charles, Oracle Distributed Systems, O'Reilly Media, April 1999 , ISBN:978-1-56592-432-1

[5] Ozsu, M. Tamer, Patrick Valduriez, Principles of Distributed Database Systems, Prentice Hall, 2nd Edition, 1999, ISBN: 9780136597070

За контакти:

Гл.ас. д-р Ирена Вълова, Катедра “Компютърни системи и технологии”, Русенски университет “Ангел Кънчев”, тел.: 082-888 685, e-mail: irena@ecs.uni-ruse.bg

Докладът е рецензиран.