

Архитектура система база података

Ненад Митић

Математички факултет
`nenad.mitic@matf.bg.ac.rs`

Дефиниција

Архитектура система база података је апстрактни опис његових компоненти и њихових интеракција

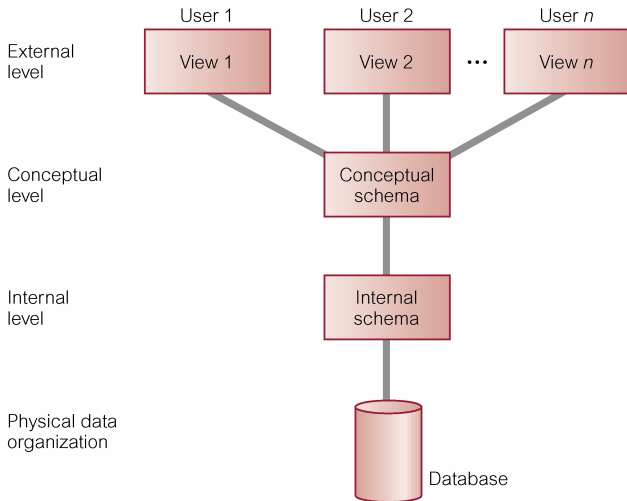
ANSI/SPARC архитектура

- Спољашњи ниво (екстерни ниво, индивидуални кориснички изглед)
- Концептуални ниво (заједнички логички изглед)
- Унутрашњи ниво (интерни ниво, физички изглед)

ANSI - American National Standards Institute

ANSI/SPARC - ANSI/System Planning and Requirements Committee

ANSI/SPARC архитектура



Пример архитектуре са 3 нивоа

- Унутрашњи ниво

```
STORED_EMP    length = 18
  PREFIX      TYPE = BYTE(6), OFFSET = 0, INDEX = EMPX
  EMP#        TYPE = BYTE(6), OFFSET=0,
  DEPT#       TYPE = BYTE(4),  OFFSET = 12
  PAY         TYPE = FULLWORD, OFFSET = 16
```

- Концептуални ниво

```
EMPLOYEE
  EMPLOYEE_NUMBER  CHARACTER(6)
  DEPARTMENT_NUMBER CHARACTER(4)
  SALARY           NUMERIC(5)
```

- Спољашњи ниво

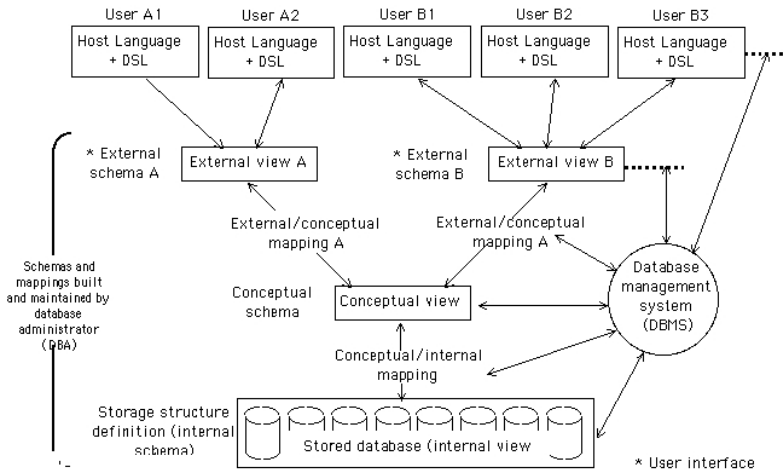
- (PL / I)

```
DCL 1 EMP
    2 EMP#    CHAR(6)
    2 SAL     FIXED BIN(31)
```

- (COBOL)

```
01  EMPC
    02  EMPNO  PIC X(4)
    02  DEPTNO PIC X(4)
```

Детаљна шема системске архитектуре



Спољашњи ниво

Сваки корисник за изражавање захтева има на располагању матични (енг. *host*) језик у који се уграђује језик података (енг. *data sublanguage*, *DSL*)

- матични језици: Java, C, PL/I, COBOL, ...
- DSL: SQL, DB2, QUEL, QBE, DL/I, ...

Повезаност матичног језика и језика података

- Ако матични језик не може јасно да се одвоји од језика података тада се за њих каже да су чврсто везани
- Ако могу јасно и лако да се раздвоје тада се за њих каже да су лабаво везани

Језик података

Језик података је комбинација најмање два подјезика

- 1 Језика за дефиницију података (енг. *Data Definition Language, DDL*) који се користи за дефинисање или декларисање објеката у бази
- 2 Језика за рад са подацима (енг. *Data Manipulation Language, DML*) који се користи при раду и обради објеката из базе
- 3 *Data Control Language, DCL* ?!

Појединачни корисник

- 1 Појединачног корисника интересује само један део базе /спољашњи изглед)
- 2 Корисник види само спољашње слоге (не одговарају нужно сачуваним слоговима)
- 3 Спољашња схема садржи дефиниције сваког од различитих типова слога у спољашњем изгледу

Концептуални ниво

- Представља информациони контекст целокупне базе података
- Подаци су независни од језика и хардвера
- Концептуални изглед је дефинисан концептуалном схемом
- Концептуална схема укључује дефиниције сваког од типова концептуалних слогова
- Записује се помоћу концептуалног DDL-а

Унутрашњи ниво

Репрезентација базе података на ниском нивоу

- Састоји се од појава различитих типова унутрашњих слогова (ANSI/SPARC термин за сачувани слог)
- Њихове карактеристике су дефинисане унутрашњом схемом и записане помоћу унутрашњег DDL-а

Још увек је изнад физичког нивоа (не ради са адресама, блоковима података или страницама у меморији)

База података и унутрашњи изглед

Termini

- сачувана база података = унутрашњи изглед
- дефиниција сачуваних структура = унутрашња схема

Неки (апликативни) програми могу да раде над унутрашњим изгледом базе

- не препоручује се због сигурности и интегритета
- овај начин обично користе *utility* програми

Пресликавање нивоа

- Пресликавање нивоа је опис повезаности два нивоа
- Једно концептуално/унутрашње пресликавање
 - како су концептуални слогови и поља представљени на унутрашњем нивоу
 - кључно за независност података од промене физичке структуре
- Више спољашње/концептуалних пресликавања

Спољашње/спољашње пресликавање

- У неким системима могуће је дефинисати један спољашњи поглед преко осталих (већ постојећих) - спољашње/спољашње пресликавање
- Чест случај у релационим системима

Промене структуре

- Концептулано/унутрашње пресликавање је кључно за независност података од промене физичке структуре
- Спољашње/концептуално пресликавање је кључно за независност података од промене логичке структуре

Послови ДВА

- Дефинисање концептуалне схеме (логичко пројектовање базе)
- Дефинисање унутрашње схеме (физичко пројектовање базе)
- Комуникација са корисницима
 - да ли су им обезбеђени сви жељени подаци
 - консултација при пројектовању апликација
 - помоћ при решавању проблема, ...

Систем за управљање базом података

(из уводног дела)

- SUBP (енг. Database management system, DBMS)
 - ниво софтвера који се налази између корисника и физичких података у бази
 - штити кориснике базе од детаља на хардверском нивоу
 - управља свим захтевима за директан приступ бази

Кораци у приступу бази

- Корисник испоставља захтев (нпр. SQL упит)
- СУБП прихвата захтев и анализира га
- Да би одредио потребне операције СУБП проверава спољашњу схему корисника, одговарајуће концептуално/унутрашње пресликавање и дефиниције меморијских структура
- СУБП извршава потребне операције (тј. захтев корисника) над базом

Функције СУБП-а

- Дефинисање података (преко DDL процесора)
- Обрада података (преко DML процесора)
 - планска (захтев познат унапред)
 - непланска (захтев непознат унапред, *ad-hoc* захтев)
- Оптимизација извршавања упита
- Обезбеђење заштите и интегритета података

Функције СУБП-а (наставак)

- Обезбеђивање конкурентног приступа подацима и опоравка
- Формирање речника (репозиторијума, каталога) података
 - Садржи информације о дефиницији СВИХ објеката (схема, пресликавања, ограничења, заштите, ...)
 - речник садржи метаподатке (податке о подацима)
- Обезбеђивање што ефикаснијег рада
- СУБП такође представља кориснички интерфејс ка систему база података

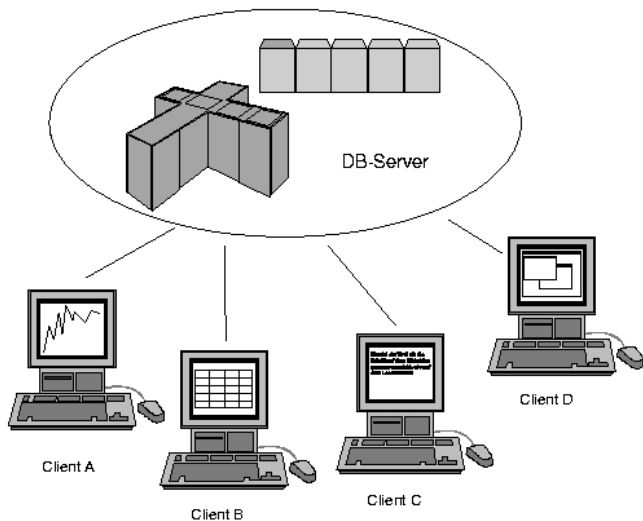
Управљање преносом података

- Захтеви корисника се преносе од места настанка до СУББ-а преко комуникационих порука; резултати се враћају истим путем
- Преносом управља се софтвера који се назива комуникациони управљач података (енг. *data communication manager, DC manager*)
- *DC manager* није део СУБП-а; раде заједно и формирају *DB/DC* систем

Клијент-сервер архитектура

- Систем база података може да се посматра као да има две компоненте
 - сервер (у суштини СУБП)
 - клијент (испоставља захтеве серверу)
- Клијент сервер архитектура је омогућена и у дистрибуираној обради

Клијент-сервер архитектура



Utility програми

- Користе се за различите администраторске послове
 - спољашњи - апликације специјалне намене
 - унутрашњи - део сервера
- Примери
 - LOAD/UNLOAD/RELOAD
 - REORG
 - Програми за статистику, анализу, ...

Дистрибуирана обрада

- Исти посао се (делом) извршава на различитим рачунарима који су спојени у мрежу
- Некада се употребљава као синоним за клијент/сервер
- Подаци из једне базе могу да буду смештени на два или више сервера
- Клијент може да приступи до више сервера

Дистрибуирана обрада

