

BAZE PODATAKA – 1. MI

Organizacijski sustav

* ORGANIZACIJSKI SUSTAV je složeni sustav koji sadrži tehnicke i humane podustave

- poduzeće, ustanova, djelatnost, društvena organizacija, tehnički sustav kao npr.

telekomunikacijska mreža i sl.

- često se organizacijski sustav naziva i POSLOVNIM SUSTAVOM, iako pojam

organizacijski sustav ima nešto šire značenje.

Informacija, podatak (Information, Data)

* INFORMACIJA je sadržaj koji primatelju opisuje nove činjenice.

* Taj sadržaj se materijalizira u obliku PODATAKA koji služe za prikaz informacija u svrhu spremanja, prijenosa i obrade.

* Podatak je skup simbola (znakova).

* Informacija je i obradeni podatak koji za primatelja ima karakter novosti, otklanja neizvjesnost i služi kao podloga za odlučivanje.

Informacijski sustav (Information System)

* Ukupna infrastruktura, organizacija, osoblje i komponente koje služe za prikupljanje, obradu, pohranu, prijenos, prikaz, širenje i raspolaganje informacijama

- Središnji dio informacijskog sustava je BAZA PODATAKA

Baza podataka (Database)

* BAZA PODATAKA je skup podataka koji su pohranjeni i organizirani tako da mogu zadovoljiti zahtjeve korisnika.

* BAZA PODATAKA je skup međusobno povezanih podataka, pohranjenih zajedno, uz isključenje bespotrebne zalihosti (redundancije), koji mogu zadovoljiti različite primjene.

Podaci su pohranjeni na način neovisan o programima koji ih koriste. Prilikom dodavanja novih podataka, mijenjanja i pretraživanja postojećih podataka

primjenjuje se zajednički i kontrolirani pristup. Podaci su strukturirani tako da služe kao osnova za razvoj budućih primjena.

Entitet (Entity)

* bilo što, što ima suštinu ili bit i posjeduje značajke s pomoću kojih se može razlučiti od svoje okoline

Atribut (Attribute)

* Entitet posjeduje neka SVOJSTVA ili ATTRIBUTE koji ga karakteriziraju.

* Izbor svojstava (atributa) koje ćemo pratiti ovisi o namjeni informacijskog sustava

Skup entiteta (Entity Set)

* Slični entiteti se svrstavaju u skupove entiteta

* Slični su oni entiteti kojima se promatraju ista svojstva

* Svi entiteti koji su članovi istog skupa entiteta imaju iste atribute

Domena i vrijednost atributa (Domain, Attribute Value)

* Za svaki entitet, atribut poprima vrijednosti iz određenog skupa vrijednosti koji predstavlja domenu tog atributa

Identifikatori entiteta, ključevi

* Skupove atributa čije vrijednosti jednoznačno određuju entitet u promatranom skupu entiteta (dakle ne postoje dva entiteta s posve istim vrijednostima tih atributa) nazivamo IDENTIFIKATORIMA ili KLJUČEVIMA SKUPA ENTITETA.

Modeliranje stvarnog svijeta

* Modeliranje stvarnog svijeta predstavlja preslikavanje stvarnog svijeta u oblik pogodan za računalnu obradu;

* Model stvarnog svijeta predstavlja se uz pomoć nekog formalnog sustava;

* Model podataka je formalni sustav koji koristimo kod modeliranja baza podataka

Model podataka (Data Model)

* Model podataka je formalni sustav koji se sastoji od:

- skupa objekata - osnovnih elemenata (koncepta) baze podataka

- skupa operacija koje se provode nad objektima

- skupa integritetskih ograničenja (integrity constraints)

* implicitno ili eksplicitno definiraju skup konzistentnih stanja podataka, promjena stanja, ili oboje

Relacijski model podataka – objekti

* elementi skupa objekata u relacijskom modelu podataka su relacije

* relacija je skup n-torki

* shema relacije obuhvaća naziv relacijske sheme i skup atributa

Arhitektura baze podataka

* Shema (struktura) baze podataka se opisuje na tri razine apstrakcije:

- Na konceptualnoj razini opisuje se KONCEPTUALNA SCHEMA

- Na unutarnjoj razini opisuje se INTERNA SCHEMA

- Na vanjskoj razini opisuju se EKSTERNE SCHEME

* Jedna baza podataka ima jednu konceptualnu, jednu internu i (najčešće) više eksternih shema

* Shema baze podataka se relativno rijetko mijenja

* Sadržaj ili instanca baze podataka (skup svih podataka baze podataka u određenom trenutku) se ČESTO mijenja

Konceptualna shema

* često se koristi i naziv LOGIČKA SCHEMA

* sadrži opis svih entiteta i veza, atributa, domena i integritetska ograničenja

* konceptualna shema se može opisati korištenjem modela podataka, npr. relacijskog ili ER modela

Interna shema

* opisuje detalje fizičke strukture pohrane i metode pristupa podacima: kako su podaci pohranjeni i koje se metode koriste za pristup podacima

Fizička nezavisnost podataka

* izmjena interne sheme ne utječe na konceptualnu shemu

Eksterna shema

* eksterna shema opisuje "pogled" na dio baze podataka koji je namijenjen specifičnoj grupi korisnika

* osnova za opis eksternih shema je konceptualna shema

Logička nezavisnost podataka

* izmjena konceptualne sheme ne mora izazvati izmjenu eksternih shema -> izmjena konceptualne sheme ne utječe

na korisnike i aplikacijske programe koji ih koriste

Sustav za upravljanje bazom podataka - SUBP

* Sustav za upravljanje bazom podataka - SUBP (Database Management System - DBMS) je programski sustav koji omogućava upravljanje podacima u bazi podataka.

* SUBP se temelji na odabranom modelu podataka.

* Prema modelu podataka na kojem se temelje, SUBP-ove dijelimo na: hijerarhijske, mrežne, relacijske, objektno-relacijske, objektno-orientirane.

* sakriva od korisnika detalje fizičke pohrane podataka

* osigurava logičku i fizičku nezavisnost podataka

* omogućuje definiciju i rukovanje podacima

- DDL - Data Definition Language

- DML - Data Manipulation Language

* obavlja funkciju zaštite podataka

- integritet podataka

- pristup podacima - autorizacija, sigurnost

- kontrola paralelnog pristupa

- obnova u slučaju razrušenja

* obavlja optimiranje upita

Jezici baze podataka

* DDL (Data Definition Language)

- omogućava imenovanje i opis entiteta, atributa, veza i pripadnih ograničenja integriteta i

pravila sigurnosti

- koristi se za definiranje nove sheme baze podataka ili modificiranje postojeće

- obavljanje DDL operacija rezultira izmjenom sadržaja rječnika podataka (metapodataka)

* DML (Data Manipulation Language)

- omogućava korištenje skupa operacija za rukovanje podacima u bazi podataka

- upitni jezik (query language)

* pojam se koristi ne samo za operacije dohvata podataka, već i za operacije izmjene,

brisanja i unosa podataka

* proceduralni jezici, neproceduralni jezici

Zašto SUBP, a ne samo datoteke?

1. Jednostavan pristup podacima

- SUBP omogućuje fleksibilniji dohvat i izmjenu podataka.

2. Upravljanje zalihostima i nekonzistentnošću

- SUBP omogućuje bolju kontrolu zalihosti od datotečnog sustava.

3. Transakcijska obrada

- SUBP obično ima ugrađenu podršku za transakcijsku obradu, što je vrlo teško ostvariti u datotečnom sustavu.

4. Složeni odnosi među podacima

- SUBP omogućuje predstavljanje različitih odnosa među podacima, definiranje novih odnosa kad se oni pojave te jednostavan dohvat i izmjenu međusobno povezanih podataka

5. Istovremeni pristup više korisnika
- U datotecnom sustavu je kontrolu istovremenog pristupa daleko teže provesti.
6. Autorizirani pristup
- SUBP omogućava određivanje nacina na koji će različiti korisnici pristupati podacima.
- Uloge osoba u životnom ciklusu baze podataka
1. Projektanti baze podataka
- oblikuju bazu podataka prema zahtjevima korisnika definirajući strukturu za pohranu podataka
2. Analiticiari sustava i programeri aplikacija
- Analiticiari sustava prikupljaju zahtjeve korisnika i pišu specifikacije za razvoj aplikacija za pristup bazi podataka
3. Administratori baze podataka
- instaliraju i nadograđuju SUBP
4. Korisnici
- Pristupaju bazi tako da postavljaju upite, mijenjaju podatke i izrađuju izvještaje
- *****

Relacijski model podataka

- Ciljevi relacijskog modela podataka:
- * osigurati visoki stupanj nezavisnosti podataka
 - * postaviti temelje za rješavanje problema semantike, konzistentnosti i redundancije podataka (normalizacija)
 - * omogućiti razvoj DML jezika temeljenih na operacijama nad skupovima
 - * objekti u relacijskom modelu podataka su RELACIJE
 - * relacija je imenovana dvodimenzionalna tablica
 - atribut je imenovani stupac relacije
 - domena je skup dopuštenih vrijednosti atributa
 - * nad istom domenom može biti definiran jedan ili više atributa
 - n-torka (tuple) je redak relacije

Svojstva relacija

- * relacija posjeduje ime koje je jedinstveno unutar sheme baze podataka
- * atributi unutar relacije imaju jedinstvena imena
- * jedan atribut može poprimiti vrijednost iz samo jedne domene
- * u jednoj relaciji ne postoje dvije jednake n-torke
- * redoslijed atributa unutar relacije je nebitan
- * redoslijed n-torki unutar relacije je nebitan

Stupanj i kardinalnost relacije

- * stupanj relacije: broj atributa (stupaca) - degree
- * kardinalnost relacije: broj n-torki (redaka) - cardinality

Schema i instanca baze podataka

- * Schema baze podataka je skup relacijskih schema
- ocito, relacijske sheme u jednoj shemi baze podataka moraju imati različita imena
- * Instanca baze podataka definirana na shemi baze podataka
- * shema baze podataka se relativno rijetko mijenja
- * instanca baze podataka se često mijenja

Operacije u relacijskom modelu podataka

- ∪ unija (*union*)
- ∩ presjek (*intersection*)
- \ razlika (*set difference*)
- + dijeljenje (*division*)
- π projekcija (*projection*)
- σ selekcija (*selection*)
- × Kartezijev produkt (*Cartesian product*)
- ρ preimenovanje (*renaming*)
- ▷◁ spajanje (*join*)
- agregacija, grupiranje

Predikatni račun

- * Predikatni račun je neproceduralan
- ne navodi se redoslijed operacija
- navode se predikati koje n-torke (domene) moraju zadovoljavati

SQL - Kratki pogled

- * SQL (Structured Query Language) je temeljen na relacijskom modelu podataka.
- * temelji se na predikatnom računu i relacijskoj algebri
- * objekti u SQL-u su tablice, a ne (formalno definirane) relacije
- * jedan SUBP može istovremeno upravljati s više baza podataka
- * opisivanje relacijske sheme (kreiranje relacije)
- kreira praznu relaciju
- ujedno je moguće definirati i integritetska ograničenja

Relacijska algebra

- * Unarne operacije
- projekcija, selekcija, preimenovanje
- agregacija, grupiranje
- * Binarne operacije
- skupovske operacije (set operations)
- * temelje se na relacijama kao skupovima n-torki
- * unija, presjek, razlika
- * operandi su relacije, a rezultat obavljanja operacije je uvijek relacija. To znači:
- skup relacija je zatvoren s obzirom na operacije relacijske algebre

Unijska kompatibilnost

- * Dvije relacije su unijski kompatibilne ukoliko vrijedi: relacije su istog stupnja i korespondentni atributi su definirani nad istim domenama
- * kod ocjene jesu li relacije unijski kompatibilne poredak atributa nije bitan i imena atributa nisu bitna

Skupovske operacije: unija, presjek, razlika

- * Skupovske operacije (unija, presjek, razlika) mogu se obavljati isključivo nad UNIJSKI KOMPATIBILNIM relacijama

SQL - Prirodno spajanje

- * prirodno spajanje se razlikuje od spajanja s izjednačavanjem po tome što se istoimeni atributi iz dviju relacija izbacuju (tako da od svakog ostane samo po jedan)

SQL - Interna pohrana NULL vrijednosti

- * NULL vrijednost se interno pohranjuje drugačije od bilo koje druge dopuštene vrijednosti

NULL vrijednost u izrazima

- * Neka je binarni operator $\alpha \in \{ +, -, *, / \}$, a X i Y su izrazi
- ako jedan ili oba operanda X , Y poprimaju NULL vrijednost, tada je rezultat izraza $X \alpha Y$ također NULL vrijednost
- * Neka je unarni operator $\beta \in \{ +, - \}$, a X je izraz
- ako operand X poprima NULL vrijednost, tada je rezultat izraza βX također NULL vrijednost

NULL vrijednost u uvjetima usporedbe

- * Neka su X i Y izrazi, a y je operator usporedbe
- * ako niti jedan od operandi X , Y nije NULL vrijednost, tada je rezultat izraza $X y Y$ logička vrijednost istina (true) ili logička vrijednost laž (false)
- * ako jedan ili oba operanda X , Y jesu NULL vrijednosti, tada je rezultat izraza $X y Y$ logička vrijednost nepoznato (unknown)

Trovalentna logika

AND	true	unknown	false
true	true	unknown	false
unknown	unknown	unknown	false
false	false	false	false

OR	true	unknown	false
true	true	true	true
unknown	true	unknown	unknown
false	true	unknown	false

Prirodno spajanje - NULL vrijednosti

* pri obavljanju operacije prirodnog spajanja potrebno je voditi računa o tome da se spajaju samo one n-torke za koje uvjet spajanja ima logičku vrijednost istina (true)

Prirodno vanjsko spajanje

* kod vanjskog spajanja uz uvjet i vanjskog spajanja s izjednačavanjem u shemi rezultata se pojavljuju svi atributi obje relacije

* kod prirodnog lijevog vanjskog spajanja iz sheme rezultata se izbacuju istoimeni atributi desnog operanda (jer iinako mogu poprimiti ili vrijednosti jednake vrijednostima korespondentnih atributa lijevog operanda ili NULL vrijednosti)

* kod prirodnog desnog vanjskog spajanja iz rezultata se izbacuju istoimeni atributi lijevog operanda (jer iinako mogu poprimiti ili vrijednosti jednake vrijednostima korespondentnih atributa desnog operanda ili NULL vrijednosti)

* kod prirodnog punog vanjskog spajanja potrebno je u shemi rezultata zadržati sve attribute obje relacije, te primijeniti operator preimenovanja atributa

SQL - Uvod

* objedinjuje funkcije jezika za definiciju podataka (DDL) i jezika za rukovanje podacima (DML)

* neproceduralnost - naredbom je dovoljno opisati što se želi dobiti kao rezultat - nije potrebno definirati kako do tog rezultata doći

WHERE Clause

* Vrijednosti svake n-torke iz relacije table se uvrštavaju u Condition (a to je u stvari predikat). Ako je dobiveni sud istinit (true), n-torka se pojavljuje u rezultatu

FROM Clause

* klasična sintaksa (classical, comma-delimited) može se koristiti za obavljanje operacija:

- Kartezijev produkt
- spajanje uz uvjet i spajanje s izjednačavanjem
- prirodno spajanje

* uvjeti spajanja se navode u WHERE dijelu SELECT naredbe, zajedno s eventualnim uvjetima selekcije (uvjeti spajanja i uvjeti selekcije se u tom slučaju povezuju logičkim operatorom AND)

* ANSI sintaksa za spajanje relacija. Može se koristiti za obavljanje operacija:

- Kartezijev produkt
- spajanje uz uvjet i spajanje s izjednačavanjem
- prirodno spajanje
- vanjsko spajanje

- * vanjsko spajanje uz uvjet
- * vanjsko spajanje s izjednačavanjem
- * prirodno vanjsko spajanje

* Ukoliko se obavlja operacija spajanja i selekcija, uvjete spajanja treba navesti u ON dijelu, a uvjete selekcije treba navesti u WHERE dijelu SELECT naredbe

Preimenovanje relacija unutar upita

* alias ime je vidljivo samo unutar upita (ne utječe na stvarno ime relacije u bazi podataka)

* na relaciju koja je u upitu dobila alias ime, moguće je referencirati se isključivo preko tog istog alias imena => to je drugačije nego kod preimenovanja atributa - kad preimenujemo atribut, možemo ga referencirati samo preko originalnog imena.

* preimenovanje relacija unutar upita treba se koristiti onda kada se ista relacija pojavljuje u više uloga unutar istog upita

* u GROUP BY nije dopušteno koristiti izraze ili zamjenska imena atributa (display_label)

HAVING Clause

* U Condition koji se navodi u HAVING dijelu naredbe dopušteno je u izrazima izvan agregatnih funkcija koristiti samo one attribute koji su navedeni u GROUP BY dijelu naredbe

ORDER BY Clause

* Ako se smjer sortiranja ne navede, podrazumijeva se uzlazni (ASC) smjer sortiranja

* U ORDER BY dijelu naredbe mogu se koristiti i izrazi koji nisu navedeni u listi za selekciju

* ORDER BY dio naredbe je jedino mjesto u SELECT naredbi u kojem je dopušteno referencirati se na zamjensko ime atributa (display_label)

* U jednoj SELECT naredbi može se pojaviti samo jedan ORDER BY dio naredbe

- ukoliko se u SELECT naredbi koristi UNION, ORDER BY se nalazi iza posljednjeg SELECT dijela naredbe

* SQL standard zahtijeva da se NULL vrijednosti pri sortiranju smatraju ili uvijek manjim ili uvijek većim od svih drugih vrijednosti

* IBM Informix NULL vrijednosti pri sortiranju uvijek tretira kao da su manje od svih ostalih vrijednosti

Podupiti

* podupit je upit koji je ugrađen u neki drugi upit

* podupit se u vanjski upit može ugraditi

- u uvjet (Condition) u WHERE dijelu vanjskog upita

- u uvjet (Condition) u HAVING dijelu vanjskog upita

- u listu za selekciju (SELECT List) vanjskog upita

* podupit može sadržavati sve do sada spomenute dijelove SELECT naredbe osim ORDER BY dijela naredbe

Podupiti u WHERE dijelu naredbe

* Ukoliko podupit čiji bi rezultat trebao biti skalar vratiti više od jedne n-torke ili više nego jedan atribut, sustav će dojaviti pogrešku

* Ukoliko podupit čiji bi rezultat trebao biti skalar ne vratiti niti jednu n-torku, dobivena skalarna vrijednost će biti NULL

Korelirani podupit (Correlated subquery)

* Ako se u podupitu koriste atributi iz vanjskog upita, za podupit i vanjski upit se kaže da su korelirani (correlated)

* Najčešće se korelirani podupit mora (fizički) izvršiti po jedanput za svaku n-torku iz vanjskog upita

Korištenje atributa vanjskog upita u podupitu

* u podupitu se mogu koristiti atributi iz vanjskog upita (obratno ne vrijedi)

* ukoliko je potrebno razriješiti dvosmislenost (npr. ista relacija se koristi u FROM dijelu vanjskog upita i FROM dijelu podupita, a u podupitu se koriste atributi relacije iz vanjskog upita), dovoljno je preimenovati relaciju u vanjskom upitu ili u podupitu

Jednostupčani podupit (Single-column subquery)

* Rezultat jednostupčanog podupita je relacija stupnja jedan, s (moguće) više n-torki

* jednostupčani podupiti se koriste u WHERE dijelu ili HAVING dijelu vanjskog upita

* jednostupčani podupiti se ne koriste u listi za selekciju

* izraz { < | <= | = | <> | > | >= } ALL (podupit)

- true ako je izraz { < | <= | = | <> | > | >= } od svih vrijednosti dobivenih podupitom

* izraz { < | <= | = | <> | > | >= } SOME (podupit)

- true ako je izraz { < | <= | = | <> | > | >= } od barem jedne vrijednosti dobivene podupitom

* ANY je sinonim za SOME

* izraz IN (podupit)

- true ako se u (multi)skupu vrijednosti dobivenih podupitom nalazi barem jedan element jednak vrijednosti izraza

- ekvivalentno sa: izraz = SOME (podupit)

* izraz NOT IN (podupit)

- true ako se u (multi)skupu vrijednosti dobivenih podupitom nenalazi niti jedan element jednak vrijednosti izraza

- ekvivalentno sa: izraz <> ALL (podupit)

* EXISTS (podupit)

- true ako rezultat podupita sadrži barem jednu n-torku (bilo kakvu). Pri tome nije važno koliko u dobivenoj n-torci ili n-torkama ima atributa (podupit ne mora biti jednostupčan) niti koje su vrijednosti njihovih atributa

* NOT EXISTS (podupit)

- true ako rezultat podupita ne sadrži niti jednu n-torku

* na rezultat vanjskog upita ne utječe eventualna pojava NULL vrijednosti u rezultatu podupita

shema SELECT naredbe

SELECT DISTINCT

nazivi atributa AS aliasatributa

FROM

nazivi tablica i/ili aliasa

JOIN

naziv tablice AS alias ON uvjet

WHERE

zadani uvjeti

GROUP BY

nazivi atributa po kojima se grupira

HAVING

naziv uvjeta po kojem se grupiranje izvodi

ORDER BY

nazivi atributa i metode po kojoj se sortira