

# BAZE PODATAKA 2 PRVA PARCIJALA TEORIJA

## P1 Uvod u DB

Baza podataka je kolekcija povezanih podataka.

Baza podataka – djeljiva kolekcija logički povezanih podataka i njihovog opisa, dizajnirana s ciljem da udovolji potrebama za informacijama u organizaciji.

Baza podataka se dizajnira, razvija i puni podacima radi specifične svrhe.

3 razine definiranja baze podataka :

- Konceptualna razina predstavlja logički pogled na pitavu bazu podataka
- Vanjska razina predstavlja korisnički pogled na bazu podataka
- Unutarnja (fizička) razina opisuje implementiranje baze podataka na konkretnom hardveru.

Sistem za upravljanje bazom podataka (engl. Database management system – DBMS) - softver opće namjene koji olakšava procese definiranja, konstruiranja, manipuliranja i dijeljenje baza podataka između različitih korisnika i aplikacija.

Relacija u relacijskim bazama podataka je TABLICA.

Ključevi :

- Super ključ (engl. superkey) – atribut ili skup atributa koji jedinstveno određuje n-torku unutar relacije
- Kandidat ključ (engl. candidate key) – super ključ takav da nema nijedan odgovarajući podskup koji bi bio super ključ unutar relacije

Kandidat za PK mora zadovoljiti 2 uvjeta : jedinstvenost, minimalnost

- Primarni ključ (engl. primary key) – kandidat ključ koji je odabran da jedinstveno odredi n-torku unutar relacije.
- Jedinstveni/Alternativni ključ (engl. unique/alternate key) – kandidat ključ koji nije odabran da bude primarni ključ

Relacija može imati SAMO JEDAN primarni ključ, dok jedinstvenih ključeva može imati više.

Povezivanje relacija omogućeno je putem vanjskog ili stranog ključa.

Vanjski ili strani ključ (engl. foreign key) – atribut ili skup atributa unutar jedne relacije koji odgovara kandidatu ključu neke (moguće i iste) relacije.

Surratni primarni ključ: Novi atribut dodan entitetu / relaciji a koristi se umjesto izvornog primarnog ključa (u pravilu, složenog).

Ograničenja u relacijskom modelu

- Integritet ograničenja (engl. integrity constraints) općenito su ograničenja vrijednosti neke varijable ili kombinacija varijabli koja je dozvoljena

AKO se određena n-torka pojavljuje u određenoj relaciji, ONDA ta n-torka zadovoljava određeni uvjet. => To je logička implikacija (AKO p ONDA q) gdje su p i q boolean izrazi. Implikacija je neistinita ako je p istinito i q neistinito, a inače je istinita.

Izraz je PREDIKAT, a varijabla PARAMETAR toga predikata.

ZLATNO PRAVILO (engl. Golden Rule): Niti jedna operacija ažuriranja ne smije se sprovesti nad vrijednostima relacije koja bi uzrokovala da njen predikat postane NEISTINA.

Kako svaki atribut ima pridruženu domenu, postoje domenska ograničenja (engl. domain constraints) koja čine restrikcije nad skupom dozvoljenih vrijednosti atributa relacije.

NUL (engl. null) – predstavlja vrijednost atributa koja je trenutno nepoznata ili nije primjenjiva za konkretnu n-torku.

Nul znači ODSUSTVO (nepostojanje) vrijednosti.

Dva opća ograničenja:

1. entitetni integritet - Primarni ključ ne može imati nul vrijednosti.
2. referencijalni integritet - Svaka vrijednost vanjskog ključa mora biti uparena

Operacije u relacijskom modelu

Grupe operatora :

Tradicionalni :

- Unija (Sve n-torke i prve i druge tablice)
- Presjek (zajednicke n-torke za obe relacije)
- Razlika (nalaze se u prvoj, a ne u drugoj)
- Kartezijev proizvod (se sastoji od svih mogućih kombinacija parova n-torki, pri čemu je prva n-torka iz prve, a druga iz druge tablice.)

Izvedeni :

- Selekcija (ograničenje, restrikcija, izbor.. , samo n-torke koje zadovoljavaju postavljene uvjete)
- Projekcija (izbor određenih atributa polazne relacije)
- Spajanje (stvora se nova relacija od svih kombinacija parova n-torki koji zadovoljavaju uvjete)

Spajanje JOIN

## P2 SQL Structured Query Language

Osobine SQL-a :

- Ne-proceduralni jezik
- Mogu ga koristiti korisnici različitog profila : DB Administratori, programeri i sl.
- Osigurava naredbe za različite zadatke : upite nad podacima, CRUD u tablicama, kreiranje mijenjanje i brisanje objekata sheme, kontrolu pristupa bazi podataka i objektima sheme, osigurava konzistentnost baze podataka

3 Osnovna dijela SQL-a

- DDL (Data Definition Language) – definisanje podataka
- DML (Data Manipulation Language) – upravljanje podacima
- DCL (Data Control Language) – kontrola i sigurnost bp

DDL – Create, Drop, Alter, Truncate

DML – Insert, Update, Delete

DCL – Grant, Revoke

TCL – Commit, Rollback, Save point

DQL – Select

RENAME: It is possible to change name of table with or without data in it using simple RENAME command

sy : RENAME TABLE <name> TO <newName>;

TRUNCATE : This command is used to remove all rows from the table, but the structure of the table still exists.

sy : TRUNCATE TABLE <name>;

INSERT : This command is used to enter the information or values into a row

sy : INSERT INTO <name> ('columnName <dataType>,...) VALUES ('value1','value2');

UPDATE: This command is used to alter existing table records

sy : UPDATE <name> SET <columnName=value> WHERE <condition>;

DELETE : It deletes all archives from a table. This command is used to erase some or all of the previous table's records.

sy : DELETE FROM <name> WHERE <condition>;

SELECT: This command is used to get data out of the database

sy : SELECT \* FROM <name>;

GRANT: This command gives users access privileges to the database.

sy : GRANT SELECT,UPDATE ON <name> TO <user1>,<user2>;

REVOKE: This command withdraws the user's access privileges given by using the GRANT command

sy : REVOKE SELECT ON <name> FROM <user1>;

Transformiranje apstraktnog modela baze podataka u dizajn baze podataka i potom u fizički model baze

Transformiranje u dizajn

1. Definiranje kandidata za primarni ključ
2. Odabir primarnog ključa
3. Odabir jedinstvenog ključa
4. Imenovanje i dodavanje atributa za strani ključ
5. Definiranje i imenovanje domena

Fizička razina – implementiranje dizajna

1. Kreiranje zaglavlja relacije u DBMS-u
2. Kreiranje integriteta relacija u DBMS-u
3. Kreiranje tijela relacije u DBMS-u
4. Pretraživanje podataka pohranjenih u DBMS-u (upiti)
5. Uspostavljanje kontrole pristupa podacima u DBMS-u

## MS SQL Server – Tipovi podataka – kategorije

- Exact numeric
- Approximate numeric
- Unicode character strings
- Character strings
- Date and time
- Binary strings
- Other data type

## P3 SQL Ključevi

### DDL izrazi

Dodavanje ključeva nakon kreiranja zaglavlja relacija :

sy : ALTER TABLE <name> ADD CONSTRAINT pk\_<name> PRIMARY KEY(<sifra>)

sy : ALTER TABLE <name> ADD CONSTRAINT fk\_<name2> FOREIGN KEY(<sifra>)  
REFERENCES <name3>(<sifra>)

### Foreign key –NO ACTION

- defaultna postavka
- ako promjena dovodi do toga da FK u child tablici nema odgovarajuću vrijednost ključa (PK/UQ) u parent tablici, baza odbija update/delete

### Foreign key – CASCADE

- ako ažuriranje vrijednosti u parent tablici utiče na FK, baza automatski ažurira sve retke u child tablici kako bi referencirali na novu vrijednost u parent tablici
- brisanje vrijednosti u parent povlači za sobom brisanje svih redaka u child

### Foreign key – SET NULL

- Vrijednost FK se postavlja na NULL ako u parent tablici dođe do brisanja ili ažuriranja vrijednosti na koje referencira FK
- Preduvjet: FK mora biti definiran kao null

## Foreign key – SET DEFAULT

-FK se postavlja na default vrijednost ako dođe do brisanja/promjene odgovarajuće vrijednosti (PK/UQ) u parent tablici na koju FK referencira.

-Preduvjet: svi stupci koji pine FK moraju imati definirane default vrijednosti.

WHERE uvjet podrazumijeva 3 elementa:

1. ime stupca
2. operator usporedbe
3. ime stupca, konstantu ili listu vrijednosti.

OPERATOR USPOREDBE može biti:

- logički ( =,>,>=,<,<=)
- SQL ( BETWEEN, AND, IN, LIKE, IS NULL)

SELECT – projekcija

WHERE – selekcija

Agregatne funkcije :

- COUNT
- SUM
- AVG
- MIN
- MAX

Izvršavaju se nad jednim stupcem i vraćaju jednu vrijednost.

Specifična je COUNT(\*) jer ne eliminira null vrijednosti.

DISTINCT – eliminira duple vrijednosti. Nema efekta na MIN I MAX, ali ima na SUM I AVG.

Može se navesti samo jednom u upitu.

Agregatne funkcije se mogu koristiti SAMO u SELECT listi i HAVING iskazu.

GROUP BY – koristi se za grupisanje redaka u manje skupine.

**PRAVILO** : Bilo koji stupac ili izraz u SELECT listi koji nije agregatna funkcija mora biti i u GROUP BY iskazu.

**HAVING** – koristi se za restrikciju grupa koje se prikazuju tj. za ispitivanje vrijednosti agregatnih funkcija.

**WHERE** – za restrikciju pojedinih redaka.

## P4 Rad sa više tablica

Kartezijev proizvod/produkt

Spajanje 2 i više tablica bez uvjeta spajanja.

ex . SELECT ime,naziv FROM djelatnici,odjeli;

Osnovni načini spajanja :

- equi join (spajanje po jednakosti)
- non equi join (po nejednakosti)
- outer joins
- self joins

### Equi

Koristi se FROM ili WHERE iskaz.

ex : SELECT ime,radno\_mjesto,naziv\_odjela FROM djelatnici,odjeli WHERE djelatnici.odjel=odjeli.sifra; (ili INNER JOIN)

### Non-equi

Koristi se kada se vrijednosti nalaze u okviru nekog raspona i ne postoji izravna veza između stupaca u tablicama.

### Outer joins

Koristi se za prikaz redaka koji inače ne zadovoljavaju uvjet spajanja.(npr null)

## RIGHT (OUTER) JOIN

Svi redovi iz desne tablice i updareni redovi iz lijeve, ako ne postoji red u lijevoj bice vrijednost NULL.

## LEFT (OUTER) JOIN

Isto ko za desni sam obrnuto.

## FULL OUTER JOIN

Svi redovi iz obe tablice sa updarenim redovima sa obe strane ako postoje, a ako ne tu ce biti vrijednost NULL.

## SELF JOIN

Rekurzivna veza, tablica se veže na samu sebe.

## CROSS JOIN

Vraća Kartezijev proizvod.