

Zbirke podatkov 1

Zapiski iz predavanj

Študent: Tadej Gunzl
Profesor: Matej Horvat
Šola: VSŠ L.I.V.E
Semester: 1. semester
Šolsko leto: 2025/26

Kazalo vsebine

1. Info	4
1.1. Predavanja in vaje	4
1.2. Ocena	4
1.3. Projektna naloga	4
1.4. Kolokviji	4
2. Uvod	5
2.1. Gradivo	5
2.2. Orodja	5
2.3. Osnovni pojmi	5
2.3.1. Podatkovna baza (Database)	5
2.3.2. SUPB (Sistem za Upravljanje s Podatkovnimi Bazami)	5
2.3.3. Entiteta	5
2.3.4. Entitetni tip	5
2.3.5. Razmerje/relacija	5
3. Podatek in informacija	6
3.1. Podatek	6
3.2. Informacija	6
3.3. Dimenziije informacije	6
3.3.1. Vrednost	6
3.3.2. Količina	6
3.3.3. Kakovost informacije	6
4. Podatkovna baza	7
4.1. Značilnosti podatkovne baze	7
4.2. Upravljanje podatkovne baze	7
4.2.1. Zagotavljanje razpoložljivosti podatkov	7
4.2.2. Zagotavljanje celovitosti podatkov	7
4.2.3. Zagotavljanje zaupnosti podatkov	7
4.3. Arhitektura podatkovne baze	7
4.4. Podatkovna neodvisnost	8
4.4.1. Fizična neodvisnost	8
4.4.2. Logična podatkovna neodvisnost	8
4.5. PB v poslovнем svetu	8
5. SUPB	10
5.1. Primeri SUPB	10
5.2. Funkcije SUPB	11
5.3. Komponente okolja SUPB	11
5.3.1. Strojna oprema (Hardware)	11
5.3.2. Programska oprema (Software)	11
5.3.3. Podatki	11
5.3.4. Postopki (Procedures ali procedure)	11
5.3.5. Ljudje	11
5.4. Aplikacije v poslovнем okolju	12
5.5. Prednosti SUPB	12
5.6. Slabosti SUPB	12
6. Normalizacija	13
7. Relacijska algebra	14
7.1. Osnove relacijske algebre	14
7.1.1. Projekcija π - pi	14
7.1.2. Selekacija σ - sigma in logični operatorji	14
7.1.3. Preimenovanje ρ	14
7.1.4. Unija U	15
7.1.5. Presek \cap	15
7.1.6. Razlika -	15
8. Jezik SQL	16
8.1. DDL	16
8.1.1. CREATE DATABASE	16
8.1.2. CREATE TABLE	16

8.1.3. ALTER TABLE	16
8.2. Osnovni podatkovni tipi	17
8.2.1. Številski	17
8.2.2. Časovni	17
8.2.3. Veliki objekti	17

1. Info

1.1. Predavanja in vaje

- Število kontaktnih ur: 84 ur (36 ur predavanj, 48 ur vaj).
- Število ur samostojnega dela: 126 ur (36 študij literature, 42 ur vaj, 48 ur seminarska naloga).
- Skupaj 210 ur dela - 7 KT.
- Obvezna je vsaj **80%** prisotnost, izdelava in predstavitev **projektne naloge** ter **pisni izpit**.

1.2. Ocena

Ocena je sestavljena iz vaj + projektne naloge (50%) in pisnega izpita (50%)

1. Opravljene vse vaje (vse vaje morajo biti oddane do roka) - 25%

2. Projektna naloga (Podatkovna baza) - 25%

- načrtovanje,
- ER diagram,
- skripte,
- poizvedbe,
- Zagovor,
- Osnovna dokumentacija

3. Pisni izpit / Tриje kolokviji - 50%

1.3. Projektna naloga

- Izdelava predvidoma med vajami; min. 5 tabel, normalizirana v vsaj 3. normalno obliko.
- Predloge, teme in področja objavljene na Teams; možna individualna izbira teme.
- Min. 10 strani; predstavitev zadnja dva tedna predavanj.
- **Vsebina:**
 - Opis procesa in ciljev naloge.
 - ER ali EER model (vsaj 4 entitetni tipi, vsak z vsaj 3 atributi; povezave).
 - SQL stavki za kreiranje tabel (CREATE).
 - SQL stavki za polnjenje tabel (vsaka tabela vsaj 5 vrstic).
 - SQL poizvedbe: 3x SELECT (vsaka s povezavo vsaj 3 tabel) + agregatne poizvedbe.

1.4. Kolokviji

1. Kolokvij: 5. 11. 2025,
 1. Skupina - 10:30,
 2. Skupina - 09:40,
 3. Skupina - 08:50,
2. Kolokvij - 26. 11. 2025,
 1. Skupina - 10:30,
 2. Skupina - 09:40,
 3. Skupina - 08:50,
3. Kolokvij (vaje) - 8. 12. 2025,
 1. Skupina - Po skupinah,
 2. Skupina - Po skupinah,
 3. Skupina - Po skupinah,

2. Uvod

2.1. Gradivo

- "Načrtovanje in postavitev podatkovnih baz" - Andreja Šet,
- "SQL Tutorial" - W3Schools,
- "SQL Tutorial" - sqltutorial.org

2.2. Orodja

- MySQL Server 8.0 Community Oracle,
- MySQL Workbench 8.0

2.3. Osnovni pojmi

2.3.1. Podatkovna baza (Database)

- Podatkovna baza predstavlja strukturirano shrambo medsebojno povezanih podatkov,
- Omogoča učinkovito shranjevanje, iskanje in vzdrževanje podatkov
- Sama po sebi ne izvaja operacij, vendar deluje kot pasivna shramba podatkov, ki čaka na ukaze sistema za upravljanje. Brez SUPB je baza nedostopna za aplikacije in uporabnike.

2.3.2. SUPB (Sistem za Upravljanje s Podatkovnimi Bazami)

- Aplikacija za upravljanje podatkovne baze,
- Zagotavlja vstavljanje, brisanje, posodabljanje in izvajanje poizvedb
- Primeri so:
 - MySQL
 - PostgreSQL
 - Oracle
 - Microsoft SQL Server
 - IBM DB2
 - Firebird

2.3.3. Entiteta

- je neodvisni podatkovni objekt iz realnega sveta,
- nosilec podatkov,
- lastnosti opisane z atributi: identifikator + opisni atributi → Primer: Janez Novak,
- Primeri:
 - Janez Novak,
 - Remembrance of Earth's Past - Cixin Liu,
 - A8 - Audi,

2.3.4. Entitetni tip

- Abstraktna predstavitev entitet z enakimi atributi,
- Definira šabloni / razred za vse entitete,
- Primeri:
 - Person,
 - Book,
 - Car,

2.3.5. Razmerje/relacija

- je povezava med dvema ali več entitetami,
- ER model → Razmerje,
- Relacijski model (beri: "tabela") → Relacija,
- Vrste relacij:
 - Ena proti mnogo [1:M] → Primer: (1) Work_email : (1) Zaposleni,
 - Mnogo proti ena [M:1] → Primer: (M) Zaposleni : (1) Marketing,
 - Mnogo proti mnogo [N:M] → Primer: (N) Študenti : (M) Predmeti

3. Podatek in informacija

3.1. Podatek

- Je predstavitev dejstva ali koncepta na formalen način,
- Je poljubna predstavitev s pomočjo simbolov ali analognih veličin,
- Pravzaprav le števila ali znaki,
- Ko podatku dodamo pomen, dobimo informacijo.

3.2. Informacija

- Je pomen, ki ga človek pripiše podatkom,
- Lahko so ovrednoteni podatki v specifični situaciji,
- Lahko pa je tudi novo spoznanje, ki ga človek doda svojemu poznovanju sveta.

3.3. Dimenzijs informacije

3.3.1. Vrednost

- Informacijsko vrednost lahko določimo kot vrednost spremembe v obnašanju prejemnika, zmanjšano za stroške pridobitve informacije.
- Vrednost informacije se s časom manjša
- Pravočasnost da se na njeni osnovi dobro odločimo → visoka vrednost
- Prepozna → zelo nizka ali nič.
- Odvisna je tudi od kakovosti.

3.3.2. Količina

Informacije je merljiva količina - osnovna enota: bit Informacija odstrani določeno stopnjo neznanja

Osnovna količina informacije - bit:

- 1bit → 2 stanji
- 2bit → 4 stanja
- 3bit → 8 stanj
- 4bit → 16 stanj - nibble

3.3.3. Kakovost informacije

Informacija ni vedno kakovostna. Kakovost informacije se lahko razlikuje od:

1. Dostopnosti - Informacija je dostopna, kadar je enostavno dosegljiva in na voljo uporabnikom takrat, ko jo potrebujejo, brez nepotrebnih ovir ali omejitev.
2. Točnosti - Informacija je točna, kadar je brez napak, pravilna in zanesljiva, ter natančno odraža realnost.
3. Pravočasnosti - Informacija je pravočasna, kadar je na voljo v času, ko je še relevantna in uporabna za odločanje.
4. Popolnosti - Informacija je popolna, kadar vključuje vse potrebne in relevantne podatke brez manjkajočih ali nepotrebnih elementov.
5. Zgoščenosti - Informacija je zgoščena (ali koncizna), kadar je jedrnata, brez odvečne ali ponavljajoče se vsebine, a hkrati ohranja bistvo
6. Ustreznosti - Informacija je ustrezna, kadar je primerna in relevantna za namen uporabe ter kontekst, v katerem se uporablja.
7. Razumljivosti - Informacija je razumljiva, kadar je jasna, enostavna za interpretacijo in prilagojena ravni znanja uporabnika.
8. Objektivnosti - Informacija je objektivna, kadar je nepristranska, brez subjektivnih vplivov ali pristranskosti, ter temelji na dejstvih

4. Podatkovna baza

Podatkovna baza je model okolja, ki služi kot osnova za sprejemanje odločitev in izvajanje akcij.

4.1. Značilnosti podatkovne baze

- je organizirana zbirka podatkov
- je integralni del vsake poslovne aplikacije ali informacijskega sistema
- obsežna shramba podatkov, ki jo lahko hkrati uporablja več uporabnikov
- namesto neurejene množice datotek so vsi podatki shranjeni na enem mestu
- minimizirano podvajanje podatkov
- poleg podatkov vsebuje tudi njihove opise - metapodatki (podatki o podatkih)

4.2. Upravljanje podatkovne baze

4.2.1. Zagotavljanje razpoložljivosti podatkov

- Razpoložljivost (ang.: Availability) pomeni **učinkovit, sočasen dostop vseh uporabnikov do podatkov**, kadarkoli jih pri svojem delu potrebujejo.
- Med uporabnike podatkovne baze poleg **končnih uporabnikov** štejemo tudi **razvijalce aplikacij, skrbnika / administratorja in uporabniške programe**.
- Podatki morajo biti razpoložljivi in dostopni tudi v prihodnje. Podatkovno bazo je treba prilagajati poslovnim zahtevam in posodabljati **infrastrukturo za hranjenje podatkov**.

4.2.2. Zagotavljanje celovitosti podatkov

- Celovitost ali integriteta (ang.: Integrity) podatkov pomeni, da so **konsistentni navznoter in z zunanjim svetom**.
- Širše lahko pri integriteti govorimo tudi o kvaliteti podatkov, ki jih podatkovna baza vsebuje. Ti so pravočasni, popolni in izvirajo iz zanesljivih virov.
- Mehanizmi za zagotavljanje celovitosti podatkov:
 - preverjanje vhodnih podatkov,
 - obnavljanje podatkovne baze,
 - nadzor nad sočasnim dostopom

4.2.3. Zagotavljanje zaupnosti podatkov

- Mehanizmi za upravljanje dostopa (ang.: Access control) uporabnikom omogočajo dostop le do tistih podatkov, ki jih glede na svojo vlogo potrebujejo - potreba po vedenju (ang.: Need to know).
- Pri tem se določi tudi vrsta dostopa (branje, spreminjanje, brisanje, spreminjanje strukture...).
- Upravljanje s pravicami dostopa je ena ključnih nalog administratorja podatkovne baze.

4.3. Arhitektura podatkovne baze

V poslovnih sistemih zaposleni uporabljajo poslovni informacijski sistem le z vidika opravljanja svojih delovnih funkcij. Le v manjših poslovnih okoljih morajo uporabljati celoten informacijski sistem (ki je v teh primerih nekoliko preprostejši).

Arhitektura zbirke podatkov predstavlja različne nivoje abstrakcije podatkov.

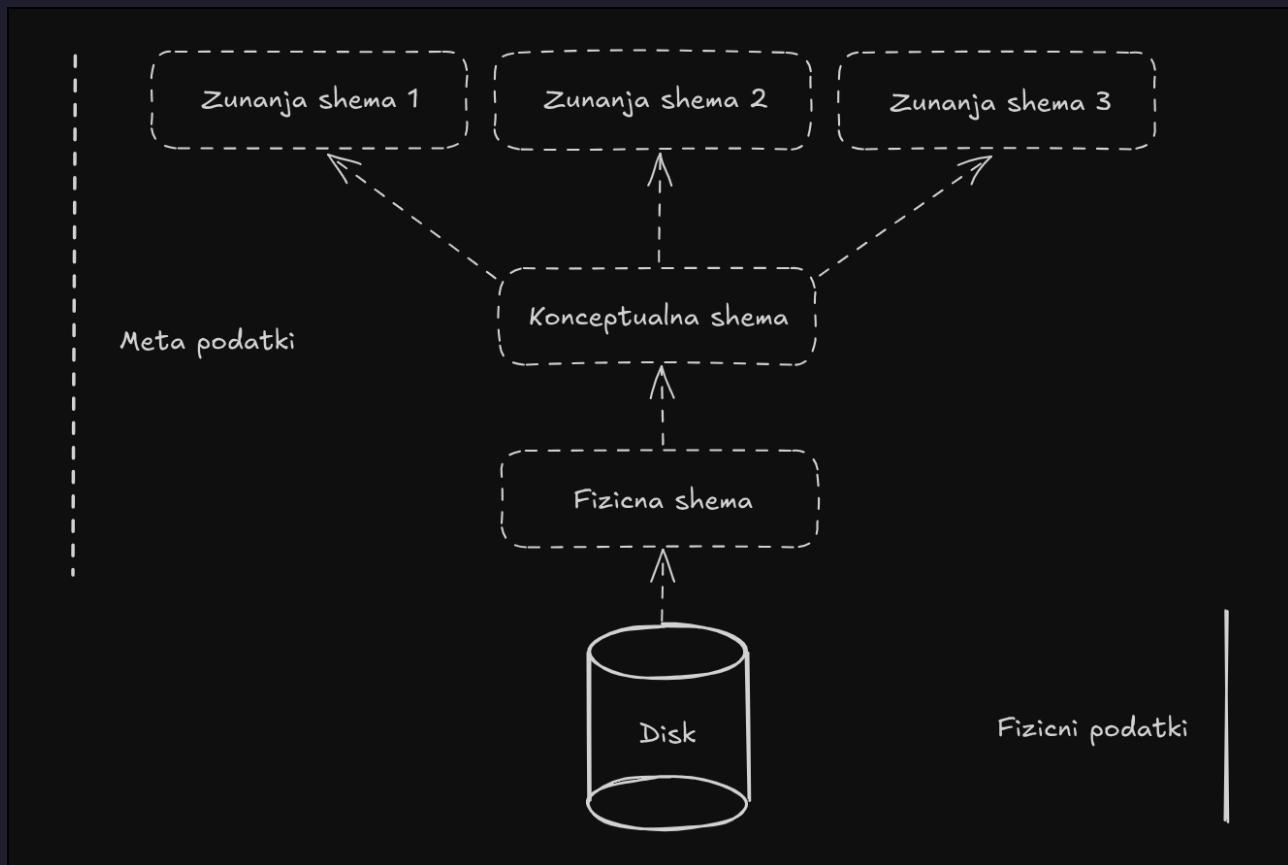
Tri-nivojska arhitektura:

- Zunanji nivo
- Konceptualni nivo
- Notranji nivo

PB omogoča podatkovno neodvisnost

Programi so neodvisni od fizičnega načina shranjevanja in strukturiranja podatkov v PB. Da bi dosegli podatkovno neodvisnost podatke v PB opišemo na treh ravneh:

1. Zunanja shema - se uporablja za dostop do podatkov, ki je prilagojen določenemu uporabniku ali skupini uporabnikov. Vsaka zunanja shema se sestoji iz enega ali več pogledov (views). Pogled je logična tabela, ki ne obstaja v fizični podatkovni bazi.
2. Konceptualna ali logična shema - opisuje podatke z vidika podatkovnega modela, ki ga PB uporablja. Npr. podatki o entitetnih tipih (profesor, študent, predmet, predavalnica, ...) in povezavah (predava, posluša...).
3. Fizična (notranja) shema - fizična shema podaja podrobnosti o shranjevanju podatkov. Predstavi, kako so podatki iz konceptualne sheme dejansko shranjeni (npr. na disku), kako so shranjeni indeksi...



4.4. Podatkovna neodvisnost

Poznamo dve vrsti podatkovne neodvisnosti:

- **Fizično neodvisnost**
- **Logično neodvisnost**

4.4.1. Fizična neodvisnost

Konceptualna shema zagotavlja fizično podatkovno neodvisnost, saj skrije podrobnosti o tem, kako so podatki dejansko shranjeni na disku, o strukturi datotek in o indeksih. Dokler ostaja konceptualna shema nespremenjena, spremembe na fizičnem nivoju ne vplivajo na programe, ki podatke uporabljajo. Lahko pa spremembe vplivajo na učinkovitost.

4.4.2. Logična podatkovna neodvisnost

Logična neodvisnost zagotavlja, da spremembe v konceptualni shemi (npr. dodajanje novih tabel ali atributov) ne vplivajo na zunanje poglede ali aplikacijske programe. Zunanja shema skrije podrobnosti konceptualne sheme, zato dokler ostaja zunanja shema nespremenjena, spremembe na logičnem nivoju ne zahtevajo prilaganja programov, ki podatke uporabljajo. To olajša razširjanje baze brez motenj obstoječih aplikacij.

4.5. PB v poslovnem svetu

V poslovnem svetu se PB uporabljajo predvsem z dvema namenoma:

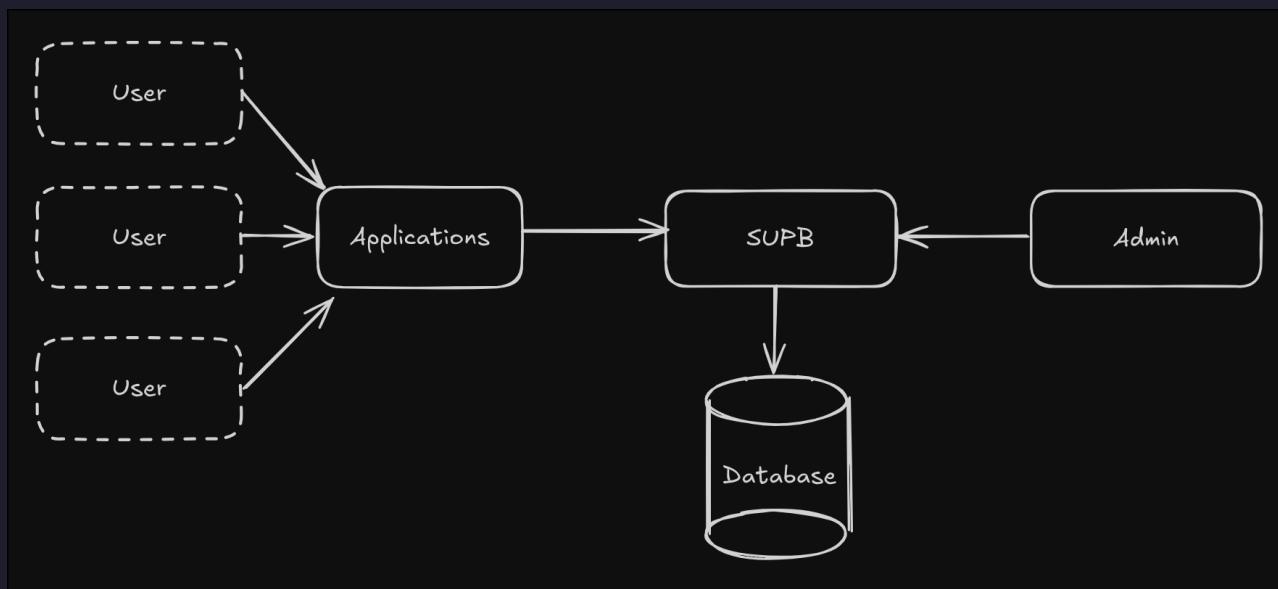
1. za hranjenje transakcijskih podatkov, ki se uporabljajo pri izvajanjju različnih poslovnih procesov
2. za upravljanje poslovnega sistema

5. SUPB

- Sistem za upravljanje podatkovnih baz (SUPB, ang.: Database management system) je programska oprema, ki omogoča obvladovanje velikih količin podatkov.
- Predstavlja vmesni člen med podatkovno bazo in aplikacijo
- Navadno do SUPBja dostopa le skrbnik podatkovne baze

5.1. Primeri SUPB

- Oracle,
- Microsoft SQL Server,
- MySQL,
- PostgreSQL,
- Firebird,
- IBM DB2,
- Informix,
- SAP DB,
- Microsoft Access,



- shranjevanje podatkov v datoteke direktno na disk - slabosti:
 - vsaka aplikacija je morala vsebovati tudi funkcije, ki jih danes opravlja SUPB,
 - vzdrževanje takšnih aplikacij je bilo zahtevno, vsebovale so veliko vrstic kode,
 - spremenjanje podatkovnih struktur je bilo zahtevno in zamudno,
 - pogosto aplikacije niso imele implementirane varnosti,
- **Prednosti SUPB:**
 - Neodvisnost podatkov: Spremembe v strukturi podatkov (npr. dodajanje stolpcev) ne zahtevajo sprememb v aplikacijski kodi, kar olajša razvoj in vzdrževanje.,
 - Centralizirano upravljanje: Podatki so shranjeni na enem mestu, kar zmanjša ponavljanje (redundanco), izboljša konsistenco in olajša upravljanje celotne baze.,
 - Sočasni dostop in nadzor konkurenčnosti: Omogoča varen dostop več uporabnikom hkrati z mehanizmi zaklepanja, ki preprečujejo konflikte in zagotavljajo pravilnost podatkov.,
 - Vgrajena varnost: Vključuje avtentifikacijo, avtorizacijo in nadzor dostopa, kar ščiti podatke pred nepooblaščenim dostopom ali spremembami.,
 - Podpora za integriteto in validacijo: Zagotavlja pravila (omejitve) za preverjanje pravilnosti podatkov, kot so unikatnost ali omejene vrednosti, kar preprečuje napake.,
 - Avtomatska obnova in rezervne kopije: Omogoča hitro obnovitev po izpadu ali napaki z uporabo dnevnikov in kopij, kar zmanjša izgubo podatkov.,
 - Učinkovito pridobivanje podatkov: Uporablja poizvedbe (npr. SQL) in optimizacijo za hitro iskanje in analizo velikih količin podatkov brez ročnega brskanja po datotekah.

5.2. Funkcije SUPB

1. shranjevanje, pridobivanje in spremjanje podatkov - SUPB omogoča varno shranjevanje podatkov v strukturirani obliki, njihovo hitro pridobivanje po zahtevah in enostavno spremjanje za posodobitve ali popravke.
2. dostopnost kataloga PB - SUPB zagotavlja enostaven dostop do kataloga podatkovne baze, ki vsebuje metapodatke o tabelah, indeksih in drugih strukturah za učinkovito upravljanje.
3. Podpora transakcijam - SUPB podpira transakcije, ki zagotavljajo zanesljivo obdelavo operacij z lastnostmi ACID (atomarnost, skladnost, izolacija, trajnost), da se izognejo delnim spremembam.
4. Mehanizem za sočasni dostop do PB - SUPB upravlja sočasni dostop več uporabnikov z mehanizmi, kot so zaklepanje in nadzor konkurence, da prepreči konflikte in zagotovi pravilnost podatkov.
5. Obnavljanje PB po nesrečah - SUPB omogoča samodejno obnovitev podatkovne baze po izpadu ali napaki z uporabo dnevnikov in rezervnih kopij, da se vrne v skladno stanje.
6. Auth - SUPB izvaja avtentikacijo uporabnikov in avtorizacijo dostopa, da omeji pravice do branja, pisanja ali izvajanja operacij samo na pooblaščene osebe.

5.3. Komponente okolja SUPB

Okolje SUPB predstavljajo strojna in programska oprema, podatki, postopki in ljudje.

5.3.1. Strojna oprema (Hardware)

Je odvisna od potreb poslovnega sistema in zmogljivosti, ki jih mora zagotovljati:

- količina podatkov,
- število uporabnikov,
- število transakcij,
- stopnja varnosti

Včasih za namestitev SUPB zadostuje že osebni računalnik, ponavadi pa ga namestimo na strežniški sistem.

5.3.2. Programska oprema (Software)

- Upravlja SUPB in ostalo programsko opremo, ki omogoča poizvedovanje po bazi,
- Ponavadi uporablja QBE ali SQL,
- Pomembna je tudi omrežna programska oprema, če bo SUPB deloval v omrežju.

5.3.3. Podatki

- S stališča uporabnika najpomembnejši del SUPB,
- Predstavljajo most med tehnološkimi in človeškimi komponentami,
- Sestavljajo jih operativni podatki in metapodatki.

5.3.4. Postopki (Procedures ali procedure)

- Obsegajo navodila za načrtovanje in uporabo podatkovne baze,
- načini prijave v SUPB,
- načini uporabe posameznih orodij,
- postopki za zagon in zaustavitev
- postopki za izdelave varnostnih kopij,
- postopki v primeru okvar,
- postopki za restavriranje podatkovne baze,
- postopki za spremjanje strukture podatkovne baze.

5.3.5. Ljudje

- Uporabniki in skrbniki,
- analitiki in načrtovalci,
- razvijalci aplikacij, ki PB uporabljajo,
- končni uporabniki.

5.4. Aplikacije v poslovnem okolju

Dandanes ni industrije, ki ne bi bila povezana z zbirkami podatkov.

Zbirke podatkov predstavljajo jedro IS (??) vsakega poslovnega sistema.

Uporabniki izpolnjujejo obrazce in poročila, za dodajanje, spremjanje, poizvedovanje in brisanje.

Sodobne aplikacije, so razvite kot spletnne aplikacije (spletne strani), pred tem pa so se uporabljale klasične aplikacije sistema odjemalec-strežnik.

Ker so dostopne preko spletja, niso potrebne posodobitve klientov, kar zmanjša stroške vzdrževanja.

5.5. Prednosti SUPB

- Nadzor redundance (Nadzoruje, vendar ne odpravi v celoti),
- Celovitost (consistency)
- Ista količina podatkov nudi več informacij in delitev podatkov,
- Večja integriteta in varnost,
- Uveljavljanje standardov,
- Enostavnejše vzdrževanje,
- Prihranek denarja,
- Večja sočasnost,
- Večja dostopnost in odzivnost,
- Varnostne kopije,
- Večja produktivnost

5.6. Slabosti SUPB

- Kompleksnost,
- Velikost,
- Visoka cena,
- Dodatni stroški za strojno opremo,
- Cena prehoda,
- Izvedba.

6. Normalizacija

7. Relacijska algebra

Tabele so množice, na katerih lahko izvajamo matematične operacije.

7.1. Osnove relacijske algebре

1. Projekcija π
2. Selekcija σ
3. Preimenovanje ρ
4. Unija U
5. Presek \cap
6. Razlika -
7. Kartezični produkt \times
8. Stiki

Primer izpeljane operacije - Deljenje

7.1.1. Projekcija π - pi

Ohrani samo izbrane attribute (stolpc) iz tabele (relacije).

Primer - Tabela hlace:

$\pi_{znamka, vrsta} (\text{hlace})$

```
SELECT znamka, vrsta FROM hlace;
```

7.1.2. Selekcija σ - sigma in logični operatorji

Ohrani samo vrstice iz tabele, ki izpolnjujejo pogoj.

$\sigma_{vrsta='športna'} (\text{hlace})$

```
SELECT * FROM hlace WHERE vrsta = 'športna';
```

Logični operatorji:

1. \wedge - in
2. \vee - ali
3. \neg - ne

$\sigma_{vrsta='športna'} \wedge \neg(znamka='S.Oliver') (\text{hlace})$

Primer z uporabo logičnih operatorjev (\Leftrightarrow bi lahko zamenjali tudi z $\neg=$):

```
SELECT * FROM hlace WHERE vrsta = 'športna' AND znamka <> 'S.Oliver';
```

7.1.3. Preimenovanje ρ

Preimenuje atribut ali tabelo.

Primeri:

$\rho_{znamka / firma}(\pi_{znamka} (\text{hlace}))$

```
SELECT DISTINCT znamka AS firma FROM hlace;
```

$\rho_{kategorija, znamka / firma, vrsta} (\text{hlace})$

```
SELECT kategorija, znamka AS firma, vrsta FROM hlace;
```

$\rho_{Kavbojke(kategorija, znamka, vrsta)} (\text{hlace})$

```
SELECT kategorija, znamka AS firma, vrsta FROM hlace AS kavbojke
```

7.1.4. Unija U

Združitev dveh tabel

majice U hlace

```
SELECT * FROM majice
UNION
SELECT * FROM hlace;
```

7.1.5. Presek \cap

Presek dveh tabel (npr. znamke, ki so v tabeli hlač in majic).

hlace \cap majice

Oracle - intersect, v MySQL pa tega (neposredno) ni.

```
SELECT DISTINCT znamka FROM hlace WHERE IN (
    SELECT znamka FROM majice
);
```

7.1.6. Razlika -

Razlika med vrednostmi v dveh tabelah. Oracle podpira minus, MySQL pa neposredno ne.

```
SELECT id FROM t1 LEFT JOIN t2 USING (id) WHERE t2.id IS NULL;
```

^ Vzamemo vse vrednosti v tabeli t1 (left join), ki niso v tabeli t2 (oz vse tiste kjer je vrednost t2 NULL)

8. Jezik SQL

SQL (izgovorjeno sequel) je programski jezik, ki ga uporabljamo za kreiranje in spreminjanje podatkovnih baz, ter dodajanje in spreminjanje podatkov v bazah. Delimo ga na:

- DDL (Data Definition Language)
 - create
 - alter
 - drop
- DML (Data Manipulation Language)
 - select
 - insert
 - update
 - delete
- DCL (Data Control Language)
 - grant
 - revoke
 - set
- TCL (Transaction Control Language)
 - begin
 - commit
 - rollback

8.1. DDL

8.1.1. CREATE DATABASE

```
CREATE DATABASE IF NOT EXISTS college
CHARACTER SET utf8mb4
COLLATE utf8mb4_slovenian_ci;
```

8.1.2. CREATE TABLE

Z ukazom CREATE TABLE definiramo tabelo in njene stolpce.

```
CREATE TABLE IF NOT EXISTS college.students (
    ID INT NOT NULL AUTO_INCREMENT,
    first_name VARCHAR(255) NOT NULL,
    last_name VARCHAR(255) NOT NULL,
    PRIMARY KEY (ID)
);
```

Dodatne vrednosti, ki jih lahko dodamo v CREATE TABLE:

- NOT NULL,
- UNIQUE,
- DEFAULT,
- PRIMARY KEY,
- CHECK,
- GENERATED, VIRTUAL, COMMENT...

8.1.3. ALTER TABLE

Z ukazom ALTER TABLE spreminjamо stolpce v tabeli. Lahko dodajamo, brišemo ali spreminjamо stolpce.

```
ALTER TABLE college.students
ADD COLUMN gender VARCHAR(255) NOT NULL DEFAULT 'male';
```

8.2. Osnovni podatkovni tipi

8.2.1. Številski

Podatkovni tip	Opis vrednosti
BIT	0, 1
TINYINT	0 do 255 ali -128 do 127
SMALLINT	0 do 65535 ali -32768 do 32767
MEDIUMINT	0 do 16777215 ali -8388608 do 8388607
INT	0 do 4294967295 ali -2147483648 do 2147483647
BIGINT	0 do 18446744073709551615 ali -9223372036854775808 do 9223372036854775807
DECIMAL(m,d)	-99999999999999999999999999999999 do 99999999999999999999999999999999
FLOAT	-3.402823466E+38 do 3.402823466E+38
SERIAL	BIGINT UNSIGNED NOT NULL AUTO_INCREMENT UNIQUE

Dodatni tipi, ki jih lahko definiramo:

- Unsigned,
- Auto Increment

8.2.2. Časovni

Podatkovni tip	Opis vrednosti
DATE	YYYY-MM-DD
DATETIME	YYYY-MM-DD HH:MM:SS
TIMESTAMP	YYYY-MM-DD HH:MM:SS.mmm
TIME	HH:MM:SS
YEAR	YYYY

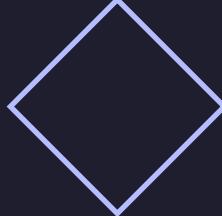
8.2.3. Veliki objekti

Binarni

Podatkovni tip	Opis vrednosti
BLOB	binary data
BINARY	binary data
VARBINARY(n)	binary data

Besedilni

Podatkovni tip	Opis vrednosti
CHAR(n)	string
VARCHAR(n)	string
TINYTEXT	string
TEXT	string
MEDIUMTEXT	string
LONGTEXT	string



Konec

Zbirke podatkov 1
Tadej Gungl, 2025

