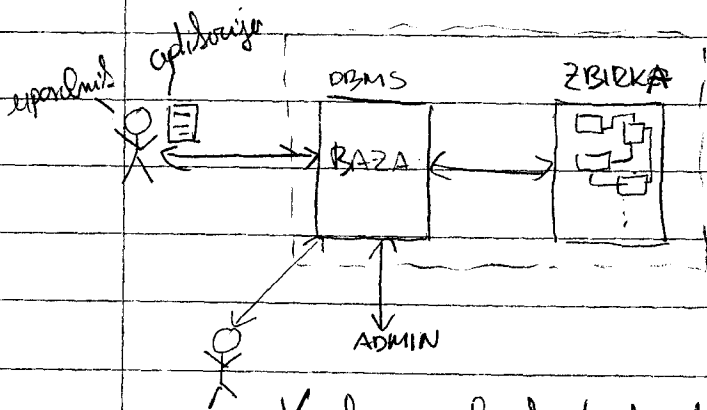


Šifirani (char) - znaki in stringi = ASCII, UTF-8...

Sklicovalni (reference) - samo povežejo kje so te podatki - kazalci

2.12

## BAZA PODATKOV



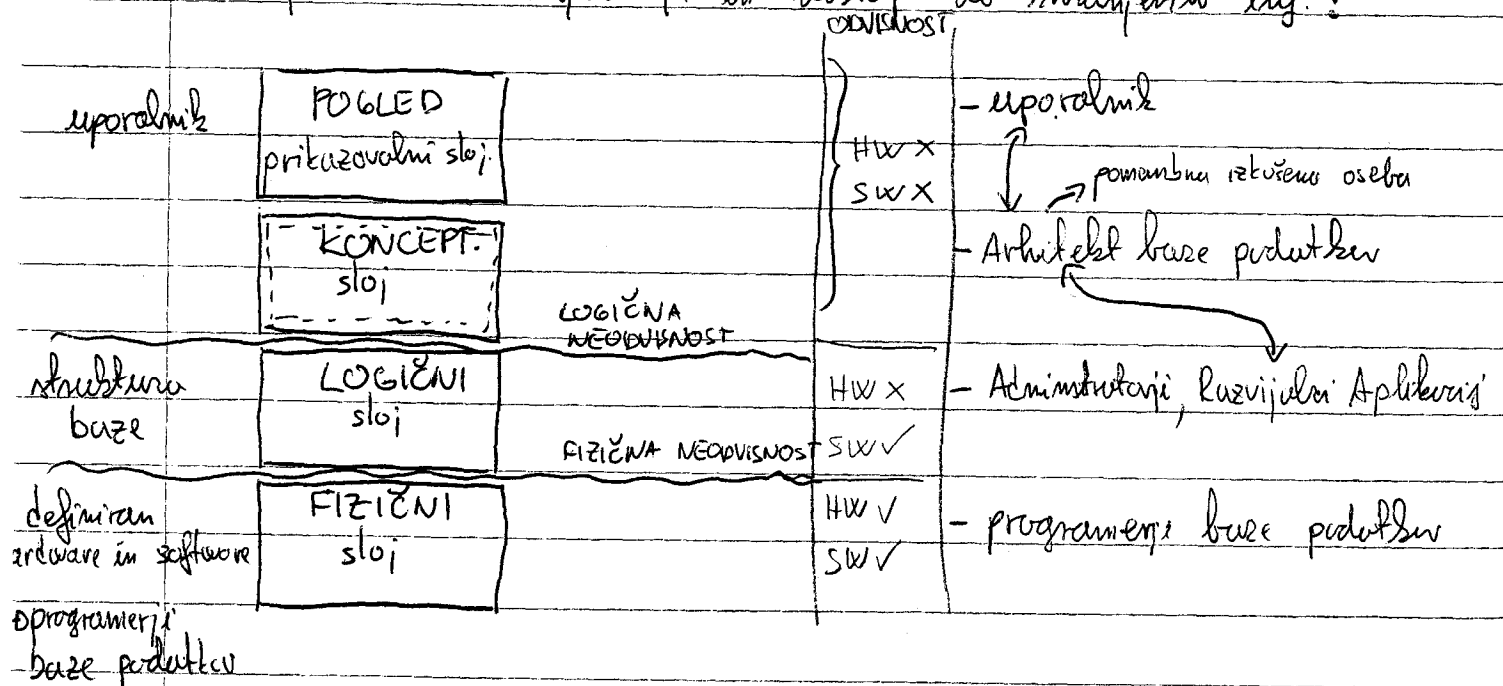
DBMS - database management system

↳ baza

ZBIKA - tabele in njihove referenije

Vsaki uporabnik, tudi Administrator pride do zbirke podatkov prek baze podatkov, brez izjeme.

Baza podatkov: shranjevanje in dostop do shranjenih inf.!



# PODATKOVNI MODELI

Osnovni gradniki:

- entiteta (bančna kartica, profesor, študent, predmet...)
- atribut (lastnost) = (ime, priimek, leto rojstva...)
- omejitve (atributov) - samo šole, samo št., dolženo št. predmetu, ...
- relacije (povezave) - kako so podatki razločeni vst med seboj povezani

1:M ena na več - en profesor uči več predmetov

M:N več na več - profesor uči več študentov, študentje hodijo k večim predmetom

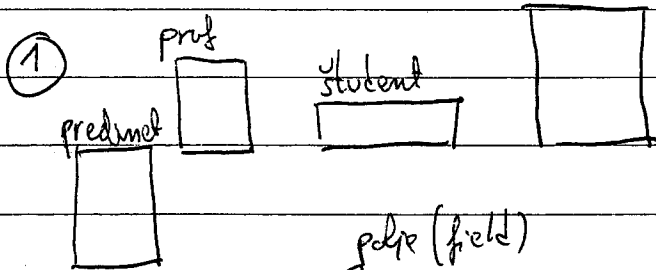
1:1 ena na ena - delam je lahko delam ene šole, in šola ima lahko enega delavca

1. DATOTEČNI SISTEM

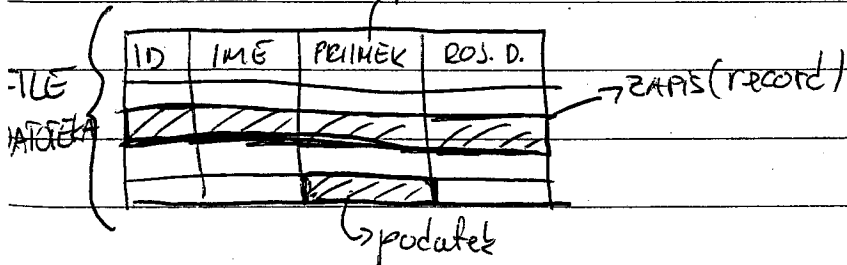
2. HIERARHIČNI MODEL → MREŽNI MODEL

3. RELACIJSKI POD. MODEL

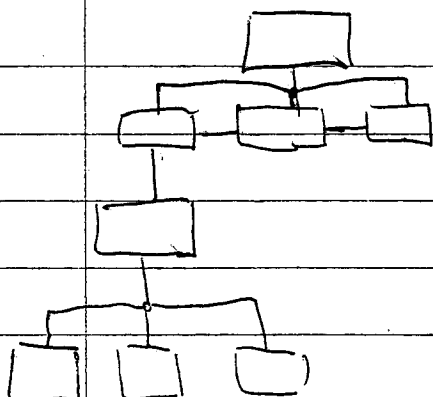
4. OBJEKTIVNO ORIENTIRAN POD. MODEL



DATOTEČNI  
MODEL

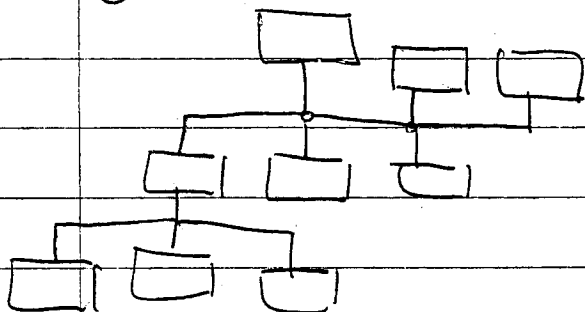


② HIERARHIČNI MODEL - vsak starš ima lahko več otrok, vendar vsak otrok ima lahko samo enega starša.

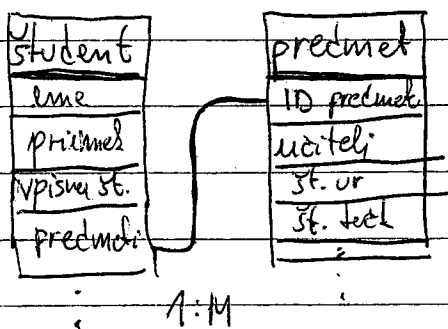


③ MREŽNI MODEL

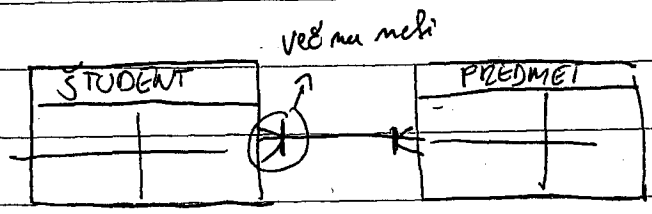
otrok ima lahko več staršev? :)



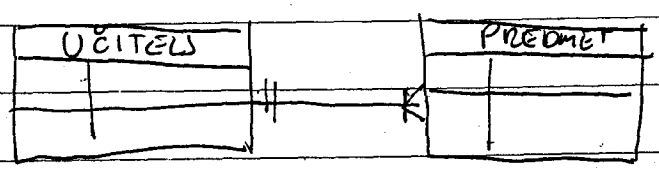
③ RELACIJSKI MODEL (MS SQL, Oracle, MySQL, ...)



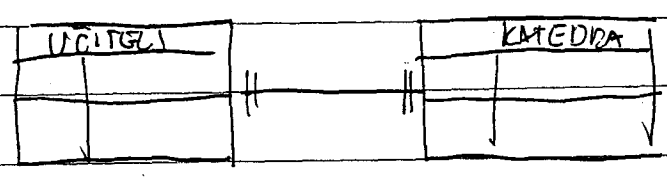
- brez implementacije



Student ima več predmetov  
in vsak predmet posluša več študentov



Učitelj ima več predmetov  
~~in vsak predmet~~

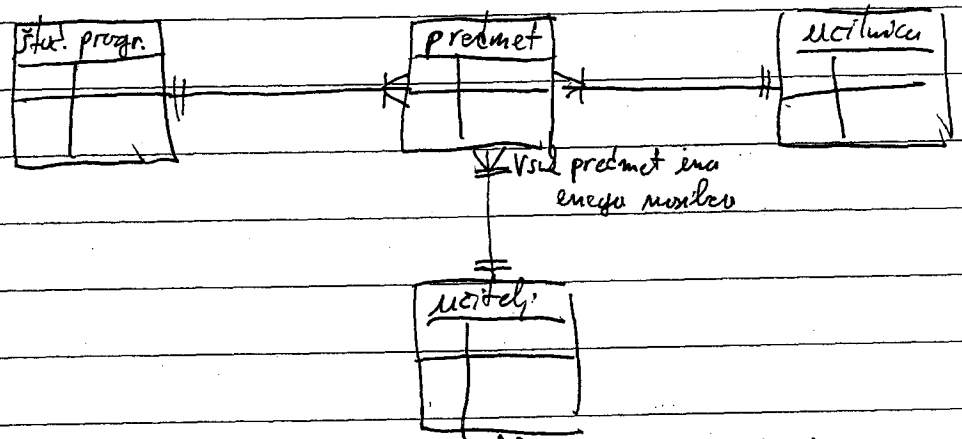


Vsaki učitelj je lahko  
predstojnik samo ene katedre

Študijski program ima več predmetov

predmet pripada enemu štud. progr.

vsaki učitelj je lahko na dan več predmetov



Učitelj lahko uči več predmetov

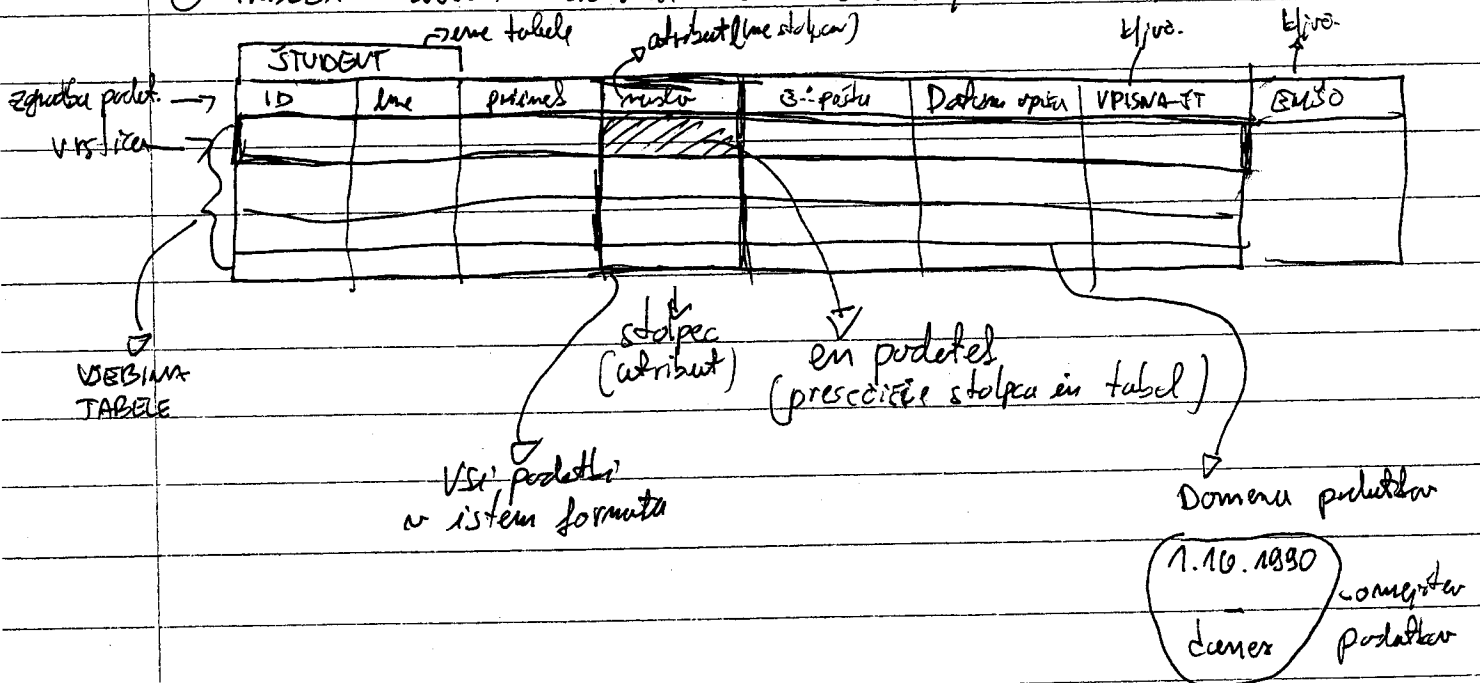
# RELACIJSKI PODATKOVNI MODEL (zbirka tabel + relacije med njimi)

## shema podat. medila (System catalogs)

- povezane med tabelami
  - atributiv. tabel
  - ključni tabel
  - št. stolpcev (atributiv) tabele
  - podat. tipi
  - datum nastanka
  - uporabniške pravice
- } srce baze podatkov

## LOGIČNI SLOS

① TABELA - dvodimenzionalna tabela sestavljena iz stolpcev in vrstic



- zaporedje vrstic in stolpcev NI pomembno.

- brez presledkov, sumnikov

- vsaka tabela potrebuje primarni ključ

- ② KLJUČI - ~~esentni~~ iz enega ali več atributov neke tabele  
 - vsaka tabela naj imi ključ s imenom ID.

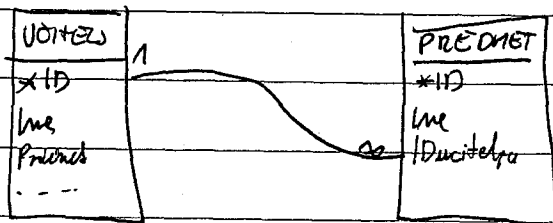
PRSTE KLJUČEV:

določa vrsto v tabeli

- Super ključ - ključ, ki enolično opredeljuje ~~vsake~~ (ene, prived, nasled)
- ↳ ena redundanca (slabost)

- Ključ kandidat (en atribut...)
- Primarni ključ (eden izmed ključev kandidatov)
- Sekundarni ključ ~~ključev~~ -
- tuji/zunanji ključ

- ③ POVEZAVE - podatki med tabelami morajo biti med seboj povezani  
 ↳ določimo jih v fazi načrtovanja



\* primarni ključ

IDucitelja - zunanji/tuji ključ

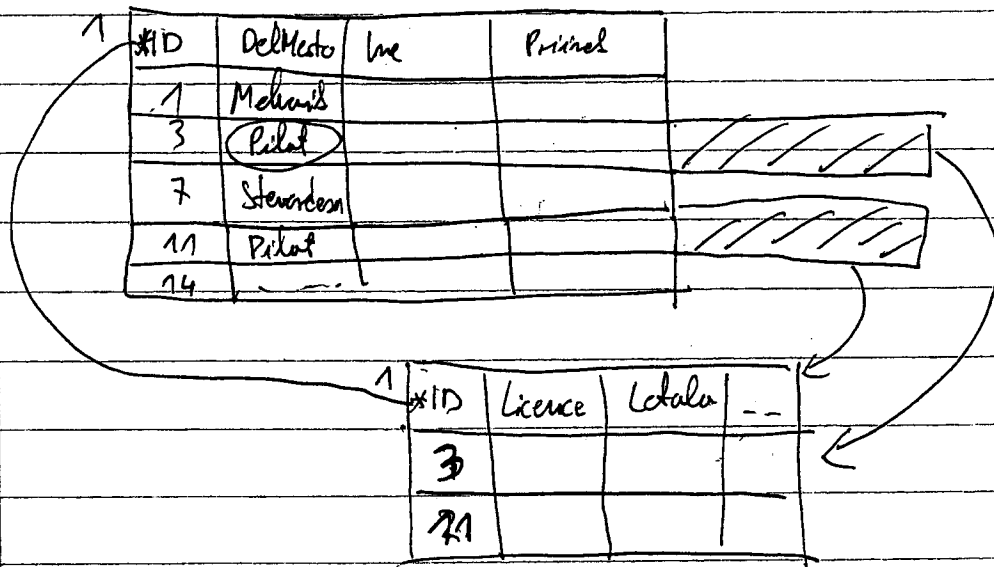
1:M

UCITEZ				
ID	Ime	Prived	...	...
1	...	...	...	...
2	...	...	...	...
3	...	...	...	...
11	...	...	...	...

ID	Ime	IDucitelja
1	...	7
7	...	7
3	...	7
2	...	7

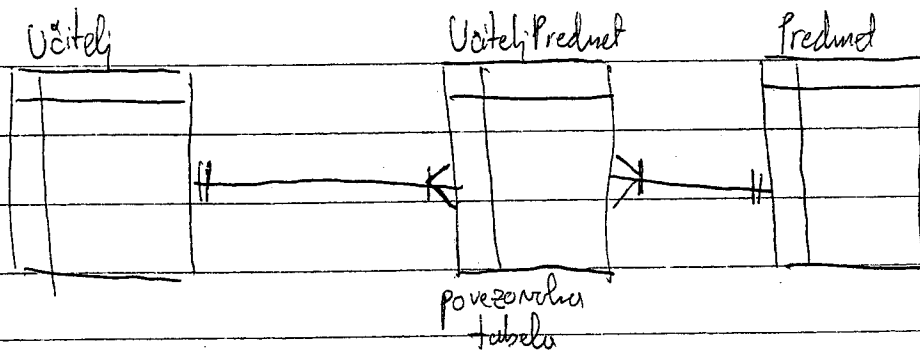
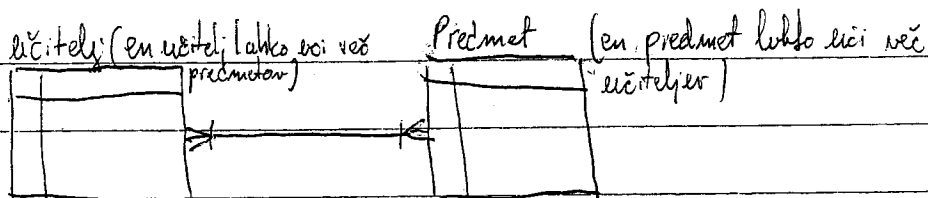
en učitelj - več predmetov

1:1 tabele zaposlenih



1.1.2011

M:N - več na več

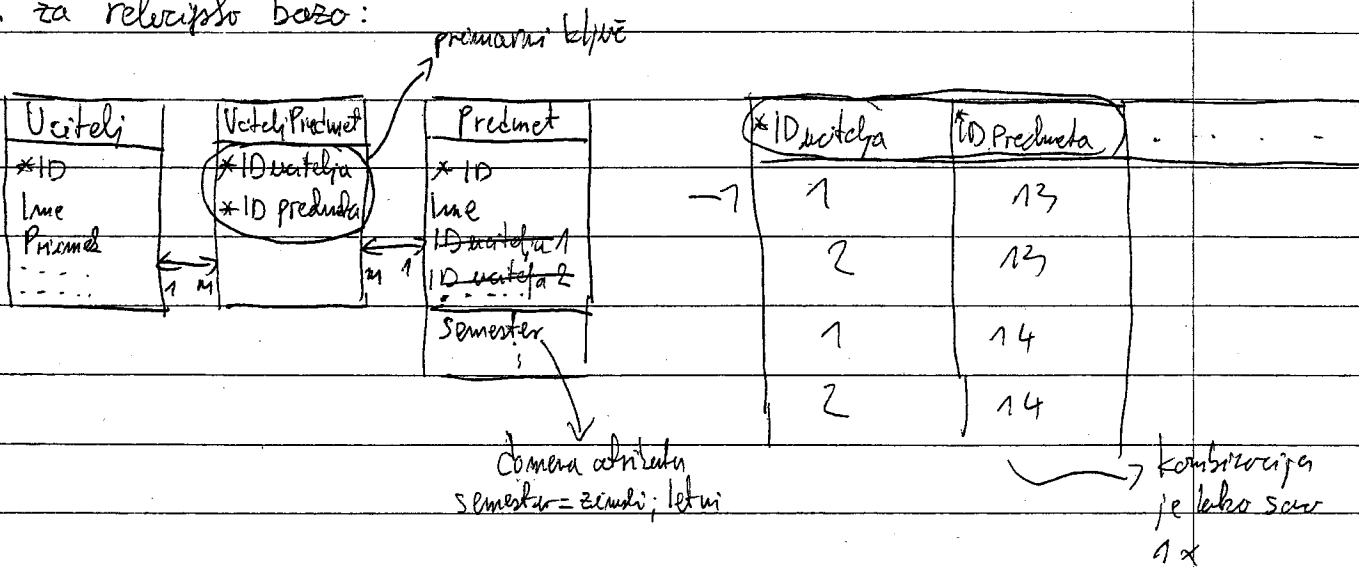


Povezovalne tabele emulirajo šar po tabelah: šar je večeno. v tem primeru UčiteljPredmet.

Rel. Baza M:N → M:1, 1:M (ista zveza)

Primarni ključ se lahko pojavi  
 1x samo. se NE sme podvojati

Diagram za relacijsko bazo:



ŠIFRANJSKA TABELA - lt (list)

lt Semester

ID	oznaka
1	zimski
2	letni
(3)	celoletni

lt Semester
*ID oznaka

Predmet
*ID
Ime
ID Semestra

Predmet je konsistentna, lažje poizvedbe za Semestr.

ID	oznaka	zatenZocelka	colanbanca
1	zimski	1.10	15.2
2	letni	16.2	30.9
(3)	celoletni	1.10	30.9

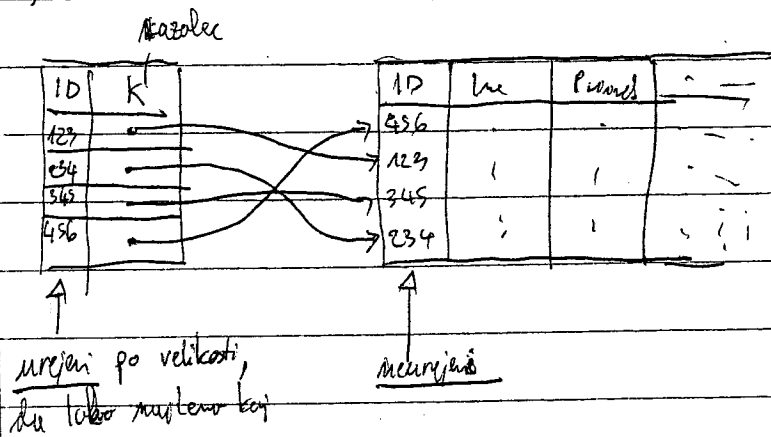
→ tabelo lahko razširimo in dodamo elemente.



# INDEKS - TIPI

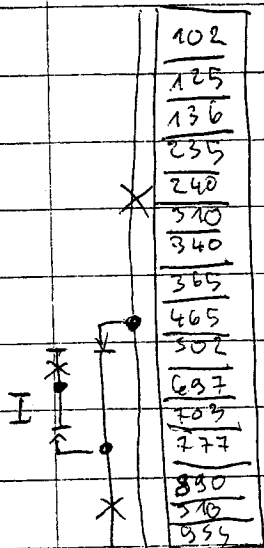
- urejeni indeksi (vrednost? ali text, po abecedi, ali random)
- indeks ma podlegi odtisov (hashi), tabela odtisov

## 1) UREJENI INDEKS



## SISTEM RAZPOVLJANJA

ISČENO 703



1. Postavimo se medje na polovico

$$465 < 703$$

2. Spet se postavimo na sredino

$$703 < 777$$

3. Spet najdemo novo sredino

$$637 < 703$$

4.  $703 = 703$  ✓

4. 3. 2. 1.  
↑  
703

Max št. korakov iskanja

$$n = \lceil \log_2 N \rceil \rightarrow \text{zadovoljeni muzejor!}$$

$N$  = st. vrednosti na seznamu

primer:  $\lceil \log_2 16 \rceil = 4$

primer:  $N = 10^6$

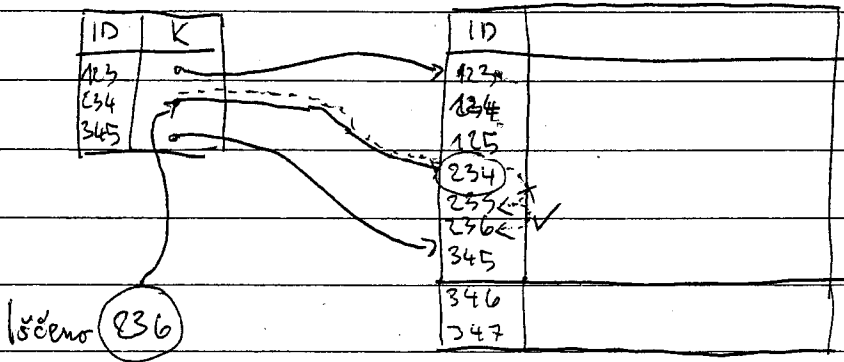
$$\lceil \log_2 10^6 \rceil < 20$$

$m = ?$

$$m = 20$$

INDEXI: - GOSTI → obstaja kazalec do vsakega zapisa v tabeli. Omogoča hitro dostop do podatka v tabeli. Ni potrebe po urejenosti tabeli  
 - REDKI problem je nehitost

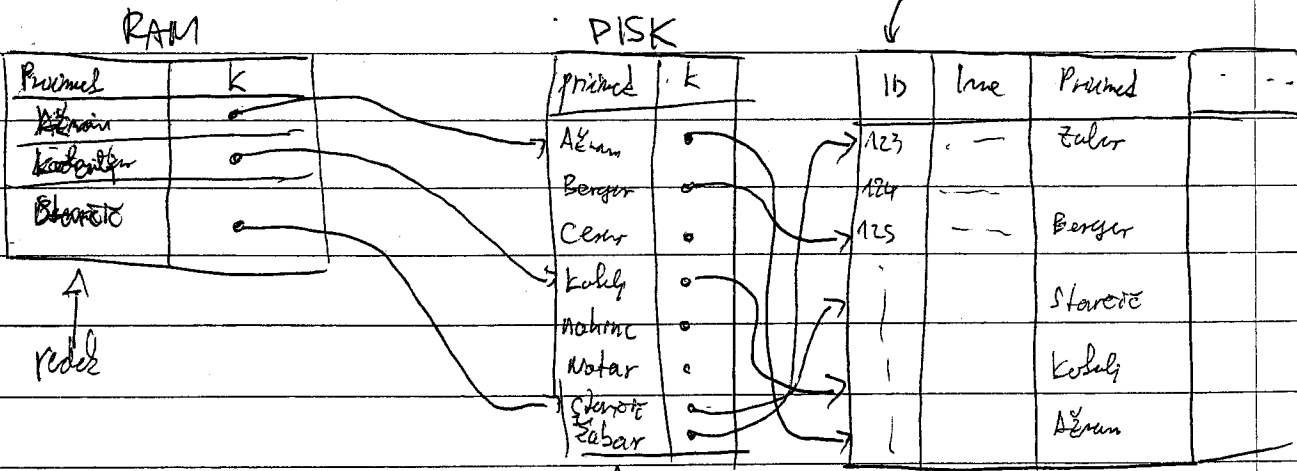
→ obstaja kazalec do samo nedostajajočih zapisa v tabeli  
 manjše hitri urejeni tabeli



Da lažje poiščemo in nujeno nelo št. MOBA  
 tabela hitri urejeni!

VEČNIVOŠKI INDEX

urejeno po ID



TRANSAKCIJA - zaporedje več osumih akcij v tabelah

Transakcija je akcija spremembe, ki mora biti izvedena do KONCA, morajo biti dobro izvedene (pravno denarnja račun =)

Če pride pri transakciji do napake se mora zgoditi ROLLBACK, da pride do prvotnega stanja baze.

### LASTNOSTI TRANSAKCIJE

- atomičnost (ali vse v celoti, ali nič)
- konsistentnost (da ne misla podoben, da je denarja denarj. -)
- izoliranost (zagotovilo, da ena transakcija ne moti druge)
- trajnost (stanje po transakciji mora biti vedno zapisano, da obstaja)

Primer: račun A in B

$A \rightarrow P \rightarrow B$

1. prebrano stanje A
  2. odštevanje P od A
  3. zapisano novo stanje A
  4. prebrano stanje B
  5. prištevanje P k B
  6. zapisano novo stanje B
- TRANSAKCIJA

# SQL - Structured Query Language (struktuiran poveljni jezik)

14.1.2011

- poleg povelj omogoča funkcionalnost:
  - = Povelja
  - Definicija podatkov
  - Manipulacija s podatki
  - Zagotavljanje integritete (do podatkov ne izpuščam, da ne pise št. ali besed. -)
  - Podpora transakcijam
  - Definicija upravljalov (od odnosa do mrežnih uporabnikov), shranjenih procedur
  - Uvodniške funkcije (muj, ker program delu za nas)
  - Avtentikacija in Avtorizacija

## SINTAKSA SQL (stavki, ključne besede)

1) STAVKI → se začnejo s ključno besedo, ki je edinstvena

2) KLJUČNE BESEDE

↳ ukaz na prnem mestu

↳ drugače, na ustreznem mestu v stavku glede na sintakso

select

SELECT ✓

Select

3) PARAMETRI

Št. vrednost atributa

starost=30

ime

ime = 'Janez' <sup>enojni navedenosti</sup>

PO

4) POSEBNI ZNAKI: \* → pomeni se na vse attribute, tabele

% ↙

? → za en sam znak

<sup>ukaz</sup>  
 Primes: SELECT Ime, Priimec, VpisnaStevilka  
 FROM Student  
 WHERE Priimec (LIKE 'A%') OR (VpisnaStevilka > 1000)  
 ↓ parameter    ↓ operator    ↓ parameter (vrednost)  
 parameter

Komentarji: - SELECT \*  
 FROM Studij.dbo.Student  
 ime baze    lastnik baze    tabela n bazi Studij  
 databaseowner

- SELECT \*  
 FROM Studij.Student    - če ne vemo lastnika baze

- SELECT \*  
 FROM [193.164.133.25].isp.dbo.Student    → to je za oddaljen strežnik

Komentarji: dva zaporedna minusa

-- izberemo vse študente iz trenutne baze

SELECT \*

FROM Student -- Tabela študentov

WHERE spol = 'M' -- študenti moškega spola

# SQL - ukazi, ~~se~~ ni grafičnega ormesila

CREATE DATABASE }  
CREATE TABLE } Kreiranje  
CREATE INDEX }

ALTER DATABASE } spreminjanje  
ALTER TABLE } lastnosti

DROP TABLE } izbris  
DROP INDEX }

Primer: CREATE TABLE student

zreda => (

VpisnaStevilka INT IS NOT NULL IDENTITY(1000, 1),

Ime nvarchar IS NOT NULL,

Priimec nvarchar IS NOT NULL,

Spol char(1),

PRIMARY KEY (VpisnaStevilka)

↳ obličina

nvarchar -> uporablja Unicode

↳ primer za Vstopno  
auto number začne s 1000, poveča se za 1  
↳ 1000, 1001, 1002, ...

ASCENDING A-Z  
DESCENDING Z-A

Manipulacije podatkov: - SELECT ukaz

- INSERT INTO

- UPDATE

- DELETE

Primer: SELECT Ime, Priimek, Email, spol

o in 100000 veljaba v pogoj

FROM Student

WHERE (spol = 'M') AND (VpisnaStenka BETWEEN 0 AND 100000)

ORDER BY Priimek DESC

INSERT INTO Student (Ime, Priimek)

VALUES 'Jamez', 'Kranjski'

} isti vrstni red kot  
v obliki

INSERT INTO Student (Ime, Priimek)

SELECT Ime, Priimek

FROM SeznamKandidatov

WHERE VpisniPogoj = 1

ORDER BY Priimek, Ime

UPDATE Student

SET Email = 'mezzan'

Vsem emailom, ki so prazni dodamo

WHERE Email IS NULL

'mezzan'

UPDATE Ceni

SET Ceni = Ceni \* 1,03

~ DELETE

FROM student

WHERE Diploma = 1

Upamaba zbirnih funkcij: - COUNT, MIN, MAX, SUM, AVG

SELECT COUNT(\*) AS <sup>Koliko bes.</sup> StanovStudenten <sup>moza izabrati atribut</sup>  
FROM Student

StanovStudenten
150

SELECT SUM (KreditneTocke) AS UscataTocke

FROM Predmet

WHERE (Letnik = 1) AND (Program = 3)

SELECT Letnik, count (\*) AS Predmetov

FROM Predmet

GROUP BY Letnik

Letnik	Predmetov	Prestepo letnika
1	10	predmetov je v
2	10	vsaki letniku
3	8	
4	12	
:		



## ZDUŽEVANJE TABEL

```
SELECT PredmetID, Ime, ProgramID  
FROM Predmet, Program  
WHERE Predmet.IDPrograma = Program.ProgramID  
ORDER BY Predmet.PredmetID
```

## INNER JOIN

///

```
SELECT PredmetID, Ime, ProgramID  
FROM Predmet INNER JOIN Program  
ON Predmet.IDPrograma = Program.ProgramID
```

```
SELECT PredmetID, Ime, ProgramID  
FROM Predmet LEFT OUTER JOIN Program  
ON Predmet.IDPrograma = Program.ProgramID
```

LEFT se nanaša na Predmet  
RIGHT bo se na Program

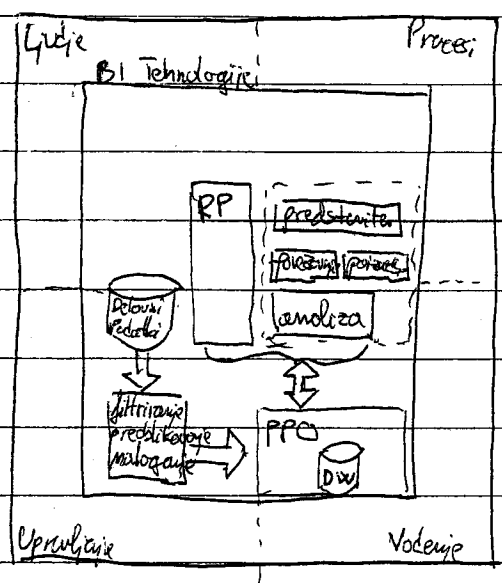
Podatki in informacije potrebni za odločitve. Povezovanje podatkov je znanje.

Intelligence

- inteligenca
- informacija?
- spoznanje, preverjanje
- podatki

BI - Business Intelligence: Termin, ki opisuje

oblastno in med seboj povezane tehnike orodij in procesov za zajem, zbiranje, shranjevanje in analizo podatkov z namenom pridobivanja in predstavitve informacij za podporo odločanja.



- zbiranje in shranjevanje delovnih podatkov (DP)
- združevanje DP in PPD
- analiza PPD → pridobimo informacije
- predstavitev rezultatov analize upravljenim
- redno poslovanje odločitev → novi podatki
- spremljanje rezultatov in ocena izidov odločitev → novi pod.

PRO - Podatki za podporo odločanja

DW - Data Warehouse - skladišče podat.

RP - Ročanje podatkov

## DELOVNI PODATKI

- kratko časno obdobje
- atomski podatki (ne moremo razdeliti na dele)
- posamezna transakcija

↓  
Relacijska DB (baza)

## FPO (Podatki za Podpno Odlučanje)

- daljše časno obdobje
- različne stopnje združevanja (prodaja izdelka X s več izdelki)
- več dimenzij podatkov

↓  
Skladišče podatkov

## SKLADIŠČE PODATKOV

- Integrirani (podatki iz več virov)
  - časovno ohranjeni
  - nezbrisljivi (zgodovinska zbirka)
- } zbirka podatkov

## ANALIZA PODATKOV - OLAP (Online Analytical Processing)

- 1 - večdimenzionalne tehnike analize
- 2 - preprost uporabniški vmesnik
- 3 - odjemalec/strežnik
- 4 - napredna podpora PB

Podatki o prodaji

- čas

- lupe

- prodajalci

- izdelki

⋮


Podrobne informacije

dimenzija proizvoda      dimenzija časa

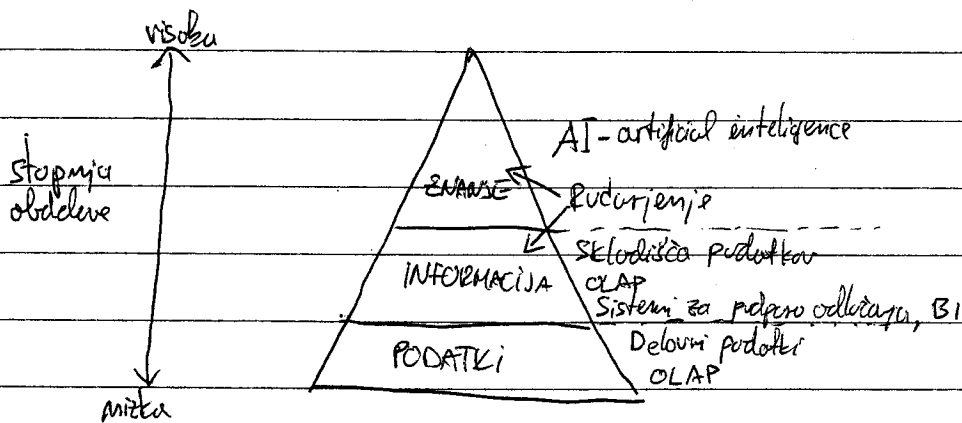
Proizvod	Januar	Februar	Skupaj
X	10	20	30
Y	30	45	55
Z	100	120	220
	Σ	Σ	Σ

Večdimenzionalna tehnika

analize

# RUDARLENJE PODATKOV

- Proces izločanja podatkov??



KONEC PREDAVANJ

izpiti: 4.2.2011

18.2.2011