

Univerza  
v Ljubljani

Fakulteta za gradbeništvo  
in geodezijo



# Baze podatkov

doc.dr. Matevž Dolenc  
mdolenc@itc.fgg.uni-lj.si

# Shranjevanje podatkov

Podatki so trajno shranjeni v datotekah v sekundarnem pomnilniku.

Podatki so lahko zapisani v obliki besedila (nizi znakov) ali v binarni obliki (števila v binarnem številskem sistemu).

Trajno shranjeni podatki na datotekah pogosto niso dovolj urejeni (Če je podatkov veliko in/ali če so raznovrstni.)

Potrebujemo učinkovita orodja za shranjevanje, urejanje in uporabo kompleksnih podatkov.

Podatkovna baza, podatkovna zbirka, podatkovno skladišče.



# Podatki v datotekah

Vsebino datoteke določa program, ki jo ustvari in ureja.

Vsak program ima svojo strukturo podatkov v datoteki.

Uporaba podatkov v datotekah zahteva podrobno poznavanje strukture podatkov.

Za upravljanje datotek skrbi operacijski sistem računalnika.

Poznamo več vrst datotek

Datoteke so besedilne (tekstovne) in binarne (oblika zapisa).

Datoteke so lahko zaporedne ali direktne (glede dostopa do podatkov).

# Tekstovne datoteke

Tekstovne datoteke so zapisane v obliki niza znakov. Vsak znak je zakodiran.

ASCII kod: 1 znak je 1 zlog (bajt)

Unicode / UTF16: 1 znak je dolg 2 zloga.

Unicode / UTF8: 1 znak je dolg 1,2 ali 3 zloge.

Posebni kontrolni znaki omogočajo razdelitev na vrstice.

Števila so zapisana v obliki niza znakov.

Tekstovne datoteko lahko urejamo z enostavnim urejevalnikom, na primer s programom Notepad.

Tekstovne datoteke lahko uporabnik urejuje na enoten način.

# Binarne datoteke

V binarnih datotekah so podatki zapisani v binarni obliki.

Cela in realna števila so shranjena binarno v isti obliki, kot jih procesirajo programi.

Procesiranje binarnih datotek je učinkovito.

Ni splošnega orodja za urejanje binarnih datotek.

Urejanje s posebnim programom.

Primer: Word, Excel, AutoCAD, ...

DeliverFromWebDAVActivity.java

```
mdolenc@matevz-142p: ~/work/projects/inteligrid/src/inteligrid-ogsadai/src/com/inteli
//(c) University of Ljubljana, 2005.
//(c) IntelliGrid, 2005.

package com.inteligrid.ogsadai.activity.delivery;

import java.io.IOException;
import java.io.InputStream;

import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NamedNodeMap;

import org.apache.commons.httpclient.HttpException;
import org.apache.commons.httpclient.URIException;
import org.apache.commons.httpclient.HttpURL;
import org.apache.commons.httpclient.HttpsURL;
import org.apache.commons.httpclient.HttpStatus;

import org.apache.webdav.lib.WebdavResource;
import org.apache.webdav.lib.methods.*;

import uk.org.ogsadai.activity.delivery.AbstractDeliverFromAddressActivity;
import uk.org.ogsadai.activity.ActivityCreationException;
import uk.org.ogsadai.activity.ActivityExecutionException;
import uk.org.ogsadai.activity.ActivitySpecificationException;
import uk.org.ogsadai.common.msgs.DAILogger;

/**
 * This activity retrieves data from a specified URL and writes it to an output.
 * The output can then be connected to the input of another activity. See the
 * activity user documentation for more details.
 *
 * @author The IntelliGrid Project Team
 */
public class DeliverFromWebDAVActivity extends AbstractDeliverFromAddressActivit
y {

    /** Copyright statement */
    private static final String COPYRIGHT_NOTICE = "(c) University of Ljubljana 2
005, (c) IntelliGrid 2005.";

}
"DeliverFromWebDAVActivity.java" 257 lines, 9085 characters
```

DeliverFromWebDAVActivity.class

```
mdolenc@matevz-142p: ~/work/projects/inteligrid/src/inteligrid-ogsadai/build/classes/
@P@<8d> ^@N@<8e> ^@N@<8f> ^@N@<90> ^@N@<91>
^@N@<92>^H@<93>^K@<94>^K@<95>^K@<96>^K@<97>^K@<98>^H@<99>^K@<9a>^<9b>^
K@<97>^<9c>
^<9d>^<9e>^H@<9f>^H@ ^H@i^H@<88>
^e^E^H@e^G@#
^N@; ^N@S
^e^e^G@e
^e^Y@e^H@e
^e^Y@e^H@e^H@e^H@e ^N@e:
^e^Y@e
^e ^?^H@e ^H@e#
^e<9d>^e^H@e.
^N@e.
^e?^e^G@e
^e^e
^e^e
^e^e^e ^N@e^G@eA
^N@eA
^N@eA
^eA^eA
^N@eA ^N@eE^K@eC@eE^G@eE^G@eE
^N@eE
^e^eI
^N@eI^H@eI
^e<9d>^eI^G@e#
^e^eN@G@e#
^e?^eN@G@e#
^eA^e# ^e#e#^H@e#
^e#eU^H@eU^H@eU
^e/^eU^H@eY^H@eP
^e/^e#
^eA^e#
^e#e#^G@e#
^e ^e#^G@e#A^e#PCOPYRIGHT_NOTICE^A^e#RLjava/lang/String;^A^e#MConstantValue^H@e
#A^e#CLOG^A^e#Luk/org/ogsadai/common/msgs/DAILogger;^A^e#EmHost^A^e#EmFile^A^e
mUsername^A^e mPassword^A^e
^
^
```

# Naloge povezane s podatki

Določitev strukture podatkov.

Urejanje strukture podatkov (shema podatkov).

Urejanje podatkov.

Dodajanje (vnos). Brisanje (odstranjevanje). Spreminjanje.

Poizvedbe - iskanje podatkov.

Poizvedovanje po podatkih, ki ustrezajo določenim pogojem.

Izdelava poročil.

Pregledna predstavitev podatkov.

# Celovitost - integriteta podatkov

Celovitost pomeni, da so podatki taki, kot morajo biti - pravilni.

V podatkih ni napak. Podatki ne manjkajo. Podatki ustrezajo vsem zahtevam.

Struktura (zgradba) podatkov je določena.

Sistem odnosov, ki morajo veljati med podatki.

Omejitve vrednosti določenih podatkov.

Tipi podatkov: celo število, realno število, niz znakov, datum, ...

Za celovitost podatkov lahko skrbi:

Uporabnik (možne napake) ali računalniški program (hitrost, zanesljivost, ... )

# Varovanje podatkov

Vsi uporabniki nimajo enakih pravic glede uporabe podatkov.

Administrator, navaden uporabnik, skrbnik rezervnih kopij, ...

Določenim uporabnikom lahko dovolimo (avtorizacija):

pregledovanje podatkov (dostop do podatkov),

urejanje podatkov (spreminjanje, dodajanje, brisanje),

urejanje podatkovnih struktur,

določanje pravic drugim uporabnikom.

## Avtorizacija

Za avtorizacijo se lahko uporabljajo gesla.

Za avtorizacijo se lahko uporabi vloga, ki jo igra uporabnik, ko se prijavi na računalnik:  
(administrator, navaden uporabnik, ... )

## Izdelava rezervnih kopij podatkov

Obstojnost medijev

Format zapisov



# Sistem za upravljanje podatkovnih baz

DBMS - Database management system

Programski sistem, ki omogoča celovito uporabo podatkovnih zbirk. Pokriva vse naloge s podatki.

Neposredno urejanje podatkov z uporabniškim vmesnikom.

Urejanje podatkov s programskimi komponentami v drugih programih.

Lahko zagotavlja celovitost podatkov.

Vsebuje sistem za varovanje podatkov.

Sistem je učinkovit.

Omogoča uporabo podatkov na visokem nivoju.

Sistem je lahko standardiziran

Poizvedovalni jezik: SQL - Standard Query Language, XQuery - XML Query Language

Programski vmesnik: ODBC - Open Database Connectivity

# Sistemi za upravljanje podatkovnih baz

## Relacijske podatkovne baze (SQL podatkovne baze).

Osnovna podatkovna struktura je tabela.

Tabela je niz podatkov urejen v vrstice in stolpce.

Tabele med seboj niso močno povezane.

Podpora XML: osnova sta podatkovni standard XML in poizvedovalni jezik XQuery.

## noSQL podatkovne baze

Shranjevanje nestrukturiranih podatkov

Različni podatkovni tipi in načini shranjevanja

## Predmetne (objektne) podatkovne baze.

Osnovna podatkovna struktura je predmet.

Predmet ima podatke in operacije.

Predmeti so urejeni v razrede predmetov.

# Relacijske podatkovne baze

Danes se uporabljajo največ.

Podatki so urejeni v tabelah.

Relacija je matematično ime za tabelo.

Je matematični pojem.

Definirane teoretične osnove. Relacijska algebra - operacije z relacijami.

Vseh podatkov ne moremo predstaviti v obliki tabel.

Drevesno strukturo podatkov ni enostavno zapisati v obliki tabel.

V zadnjem času se uveljavljajo relacijske baze s podporo standardu XML.

# Relacijske podatkovne baze

Tabela je zbirka podatkov o določeni vrsti informacije (tema, vrsta predmetov)

Tabela oseb, stroškov, naročil, ocen, ...

Tabela je urejena:

v stolpce - polja (field),

v vrstice - zapise (record).

Stolpci imajo imena - attribute.

Podatki v stolpcu so vedno istega tipa.

Zapis določa podatke o določenem predmetu (osebku, entiteti, abstrakciji).

<b>Ind</b>	<b>Ime</b>	<b>Ocena</b>
1	Petra	9
2	Matjaž	10
3	Jaka	8

# Tipi podatkov

Podatek v stolpcu mora biti enostaven (nesestavljen, atomaren).

Ne more biti sestavljen iz drugih podatkov.

Celo število. Realno število.

Datum.

Niz znakov predpisane dolžine.

Poljubno dolg niz znakov.

Niz bajtov (zlogov) spremenljive dolžine (binarni podatki).

Pri novejših bazah je dovoljen tip XML dokument ali XML fragment.

# Lastnosti relacij

Podatki v zapisih smejo manjkati.

Podatek NULL.

V tabeli ne sme biti dveh popolnoma enakih zapisov.

Vrstni red zapisov ni pomemben.

V tabeli moramo določiti polje (stolpec), ki enolično določa zapis - ključ.

Imenujemo ga glavni ključ (Primary Key).

Vpisna številka študenta.

Glavni ključ lahko določa tudi več polj.

Ime in priimek osebe.

# Odnosi med tabelami - relacije

Tabele so lahko med seboj povezane

Odnosi med tabelami

Isti podatek lahko nastopa v različnih tabelah.

Vpisna številka nastopa v tabeli Študent in Izpit: tuji ključ (foreign key) ustreza glavnemu ključu v drugi tabeli.

tabela: Študenti

Vpisna št.	Ime
26100001	Petra
26100002	Matjaž
26100003	Jaka

Primarni ključ

tabela: Izpiti

Ind	Izpit. rok	Vpisna št.	Ocena
1	1	26100001	NP
2	1	26100002	10
3	2	26100001	NP
4	3	26100001	9
5	3	26100003	8

Primarni ključ

Tuji ključ

# Vrste odnosov med tabelami - vrste relacij

Odnose določimo s ključi in tujimi ključi.

Odnos 1:mного med tabelama A in B.

Enemu zapisu v tabeli A ustreza mnogo zapisov v tabeli B.

Enemu zapisu v tabeli B ustreza en sam zapis v tabeli A.

Odnos 1:1 (redko uporabljamo).

V tem primeru lahko tabeli združimo v eno samo tabelo.

Odnos mnogo:mного med tabelama A in B.

izvedemo z odnosi 1:mного s pomočjo dodatne tabele (AB).

A:AB (odnos 1:mного), B:AB (odnos 1:mного).



# Primer

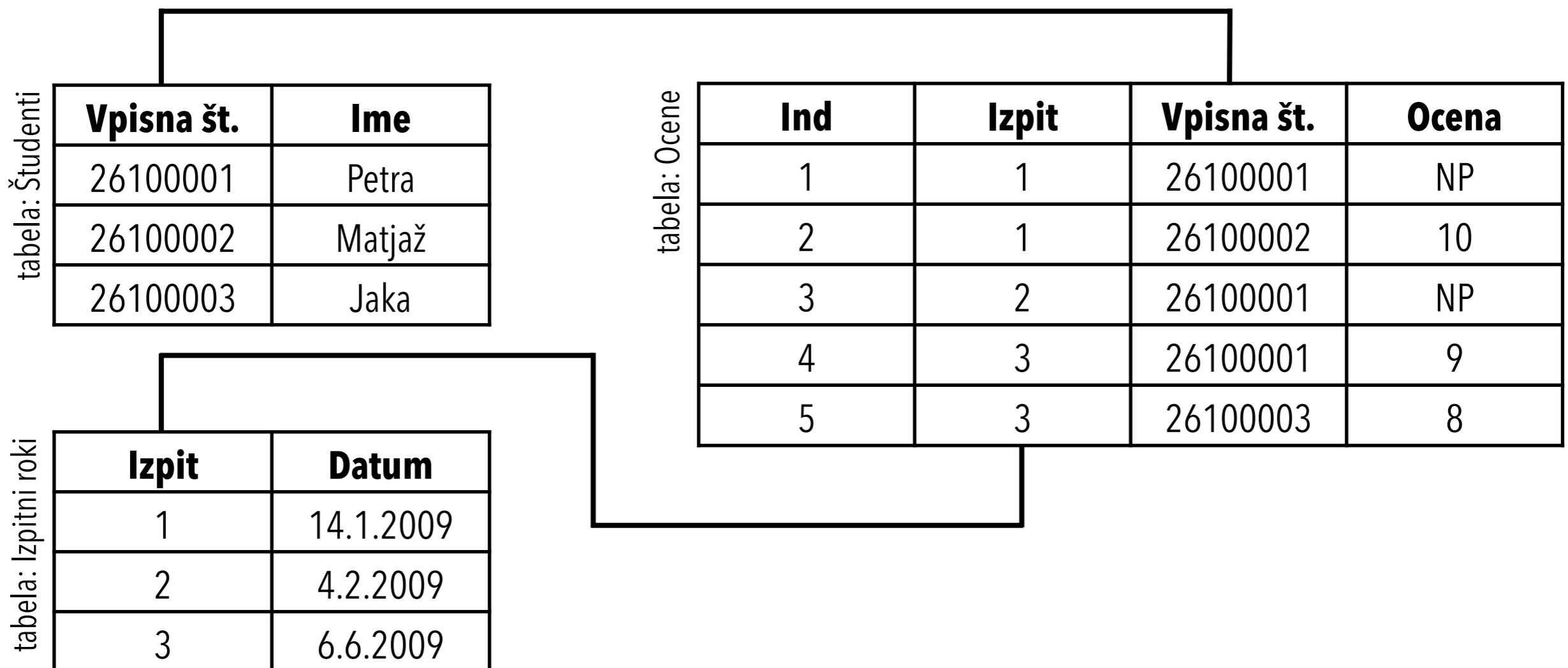
## Naloga

Zanima nas kako študenti opravljajo izpite pri enem konkretnem predmetu. Sestavi tebele in določi relacije med njimi

# Primer

## Naloga

Zanima nas kako študenti opravljajo izpite pri enem konkretnem predmetu. Sestavi tebele in določi relacije med njimi



# SQL - Structured Query Language

Standardni jezik za uporabo relacijskih podatkovnih baz.

ANSI/ISO standard

Sistem stavkov

Slovnična pravila: `SELECT * FROM vozli;`

Kategorije stavkov

DDL (Data Definition Language) - Jezik za definicijo podatkov - shema (definicija tabel)

DML (Data Manipulation Language) - Jezik za uporabo podatkov

Razni kontrolni stavki - Transakcijski kontrolni stavki



# NoSQL podatkovne baze

NoSQL podatkovne baze nimajo vnaprej definirane sheme tabel

Primerne za nestrukturirane podatke

Različni podatkovni tipov

Key Value Store

Document Store

Column Store

Graph Base

Feature	NoSQL Databases	Relational Databases
Performance	High	Low
Reliability	Poor	Good
Availability	Good	Good
Consistency	Poor	Good
Data Storage	Optimized for huge data	Medium sized to large
Scalability	High	High (but more expensive)

EROSPIKE



cassandra



Clustrix



Couchbase



amazon  
DynamoDB

APACHE  
HBASE



MarkLogic



memsql  
Speed. Scale. Simplicity.



mongoDB



NUODB

ORACLE  
NOSQL  
DATABASE

riak

splice  
MACHINE

TRANS/ATTICE



VOLTDDB