



# OSNOVE INFORMACIONIH SISTEMA

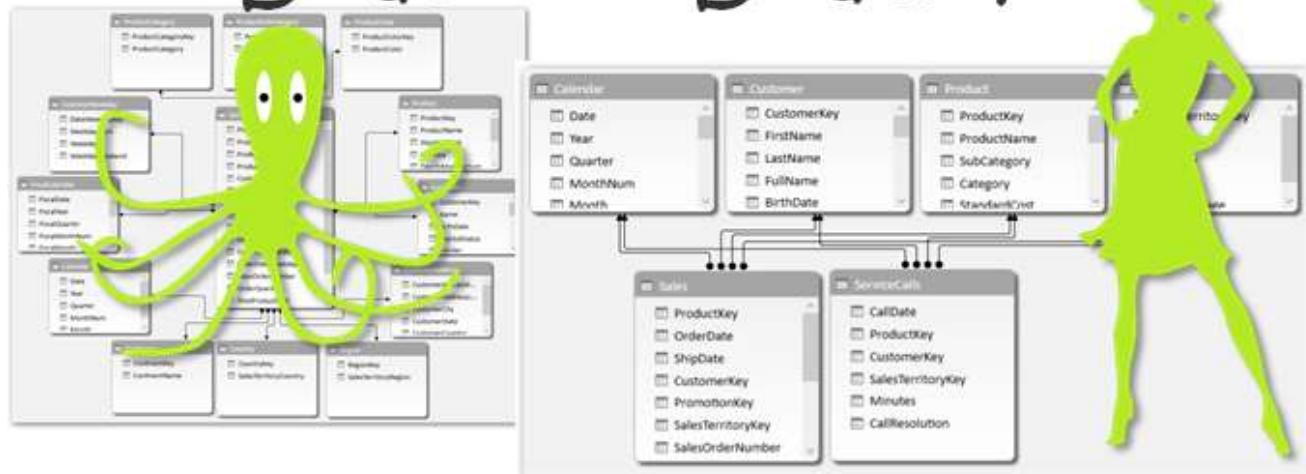
Tutorijal 7

10.12.2024.



# Modeliranje podataka i procesa

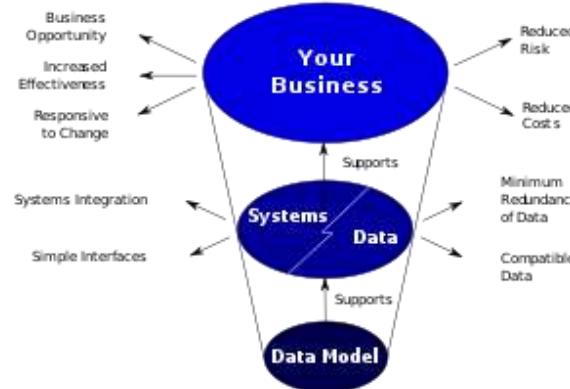
## Data Model: Beast to Beauty





# Modeliranje podataka

- Modeliranje podataka je proces kreiranja modela podataka za podatke koji se pohranjuju u bazi podataka.
- Modeliranje podataka pomaže u vizualnom predstavljanju podataka i provodi poslovna pravila, regulatorne usklađenosti i politike o tim podacima.
- Modeli podataka osiguravaju dosljednost u imenovanju, zadanim vrijednostima, semantici, sigurnosti, istovremeno osiguravajući kvalitetu podataka.





# Modeliranje podataka

Modeliranje podataka odvija se na tri nivoa: fizičkom, logičkom i konceptualnom.

**Konceptualni:** Ovaj model podataka definira šta sistem sadrži. Ovaj model obično kreiraju poslovni stakeholderi i arhitekti podataka. Namjena je organizirati, obuhvatiti i definirati poslovne koncepte i pravila.

**Logički:** definira kako sistem treba implementirati bez obzira na DBMS. Ovaj model obično kreiraju arhitekti podataka i poslovni analitičari. Namjena je razviti tehničku kartu pravila i strukture podataka.

**Fizički:** Ovaj model podataka opisuje kako će se sistem implementirati pomoću određenog DBMS sistema. Ovaj model obično kreiraju DBA i programeri. Svrha je stvarna implementacija baze podataka.



# Modeliranje podataka

## Razlika između nivoa:

Nivo	Fokus	Tehnički detalji
Konceptualni	Poslovni zahtevi, entiteti i veze	Nema tehničkih detalja
Logički	Organizacija podataka, relacije	Tehnički okvir, ali bez detalja o implementaciji
Fizički	Skladištenje i performanse	Konkretni tehnički detalji baze podataka

Podela omogućava postupno prelazak od apstraktnog razmišljanja (poslovni zahtevi) ka tehničkoj implementaciji, što olakšava komunikaciju između poslovnih analitičara, dizajnera baza podataka i inženjera.



# Konceptualni nivo modeliranja

**Šta je?** Ovo je najviši, apstraktni nivo modeliranja. Fokusira se na što bolje razumevanje podataka i njihovih veza iz poslovne perspektive, a ne na tehničke detalje.

**Cilj:** Definisati osnovne entitete i njihove veze, kao i ključne karakteristike koje su relevantne za poslovanje.

## Karakteristike:

- Nema tehničkih detalja, poput vrsta podataka ili tehnologija.
- Koristi dijagrame entiteta i veza (ER dijagram).
- Fokusiran je na potrebe korisnika.



# Logički nivo modeliranja

**Šta je?** Prelazni nivo između apstraktnog konceptualnog modela i tehničkog fizičkog modela. Fokusira se na strukturiranje podataka u obliku koji baza podataka može obraditi, ali i dalje nezavisno od specifične tehnologije.

**Cilj:** Definisati kako će podaci biti organizovani, sa fokusom na relacije, ključeve i tipove podataka.

## Karakteristike:

- Precizno definiše atribute entiteta (npr. ISBN za knjige, ID za člana).
- Uključuje primarne i strane ključeve (identifikatori za relacije).
- Normalizacija se koristi za smanjenje redundancije.



# Fizički nivo modeliranja

**Šta je?** Najdetaljniji i tehnički nivo modeliranja, koji definiše kako će podaci biti stvarno skladišteni u bazi podataka. Fokusira se na implementaciju baze u konkretnom sistemu.

**Cilj:** Optimizovati skladištenje podataka i njihov pristup u skladu s tehnologijom koja se koristi.

## Karakteristike:

- Definiše konkretne tipove podataka (npr. VARCHAR, INT).
- Određuje indekse, particije i pravila o pristupu.
- Fokusiran na performanse i tehničke detalje.



# Modeliranje podataka

## Konceptualni nivo:

- Na konceptualnom nivou modeliranja podataka, fokus je na razumijevanju poslovnih entiteta, njihovih odnosa i pravila koja definiraju organizaciju podataka.
- Ovdje se koriste konceptualni modeli podataka poput entitetsko-relacijskih dijagrama (ERD) kako bi se predstavili entiteti (objekti) i njihovi odnosi.
- Cilj je stvoriti apstraktnu predstavu podataka neovisnu o bilo kojem konkretnom sistemu ili tehnologiji.

## Logički nivo:

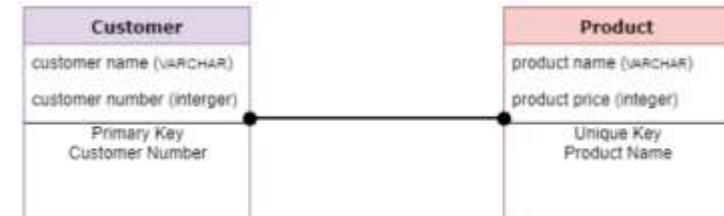
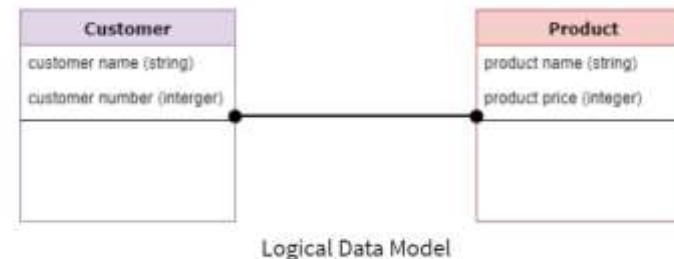
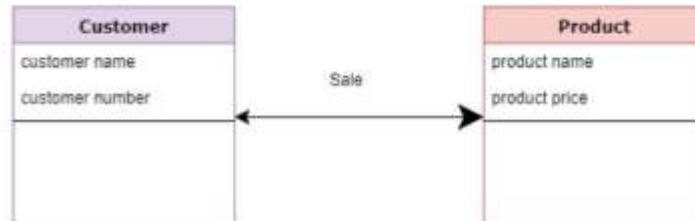
- Na logičkom nivou, konceptualni model se transformira u specifičan oblik prilagođen određenom sistemu upravljanja bazama podataka (DBMS).
- Ovdje se koriste entitetsko-relacijski modeli prilagođeni za odabrani DBMS, kao i dodatne informacije o atributima, ključevima i vezama između entiteta.
- Cilj je stvoriti model koji može poslužiti kao temelj za implementaciju baze podataka, uzimajući u obzir specifičnosti sistema upravljanja bazama podataka.

## Fizički nivo:

- Fizički nivo modeliranja podataka fokusira se na detalje implementacije na razini sklopolja i tehnologije.
- Ovdje se definiraju tehnički detalji, poput tipova podataka, indeksa, particioniranja i performansi.
- Cilj je optimizirati strukturu podataka za konkretnu platformu ili tehnologiju na kojoj će baza podataka biti izvršena.



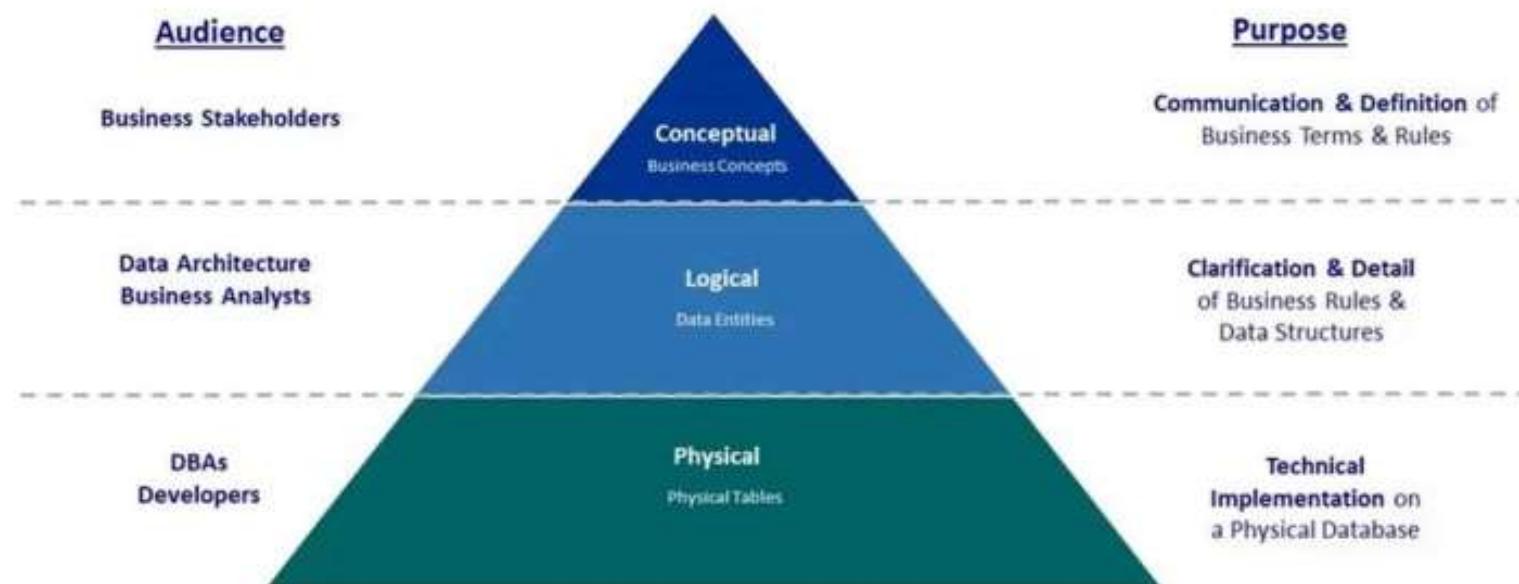
# Modeliranje podataka





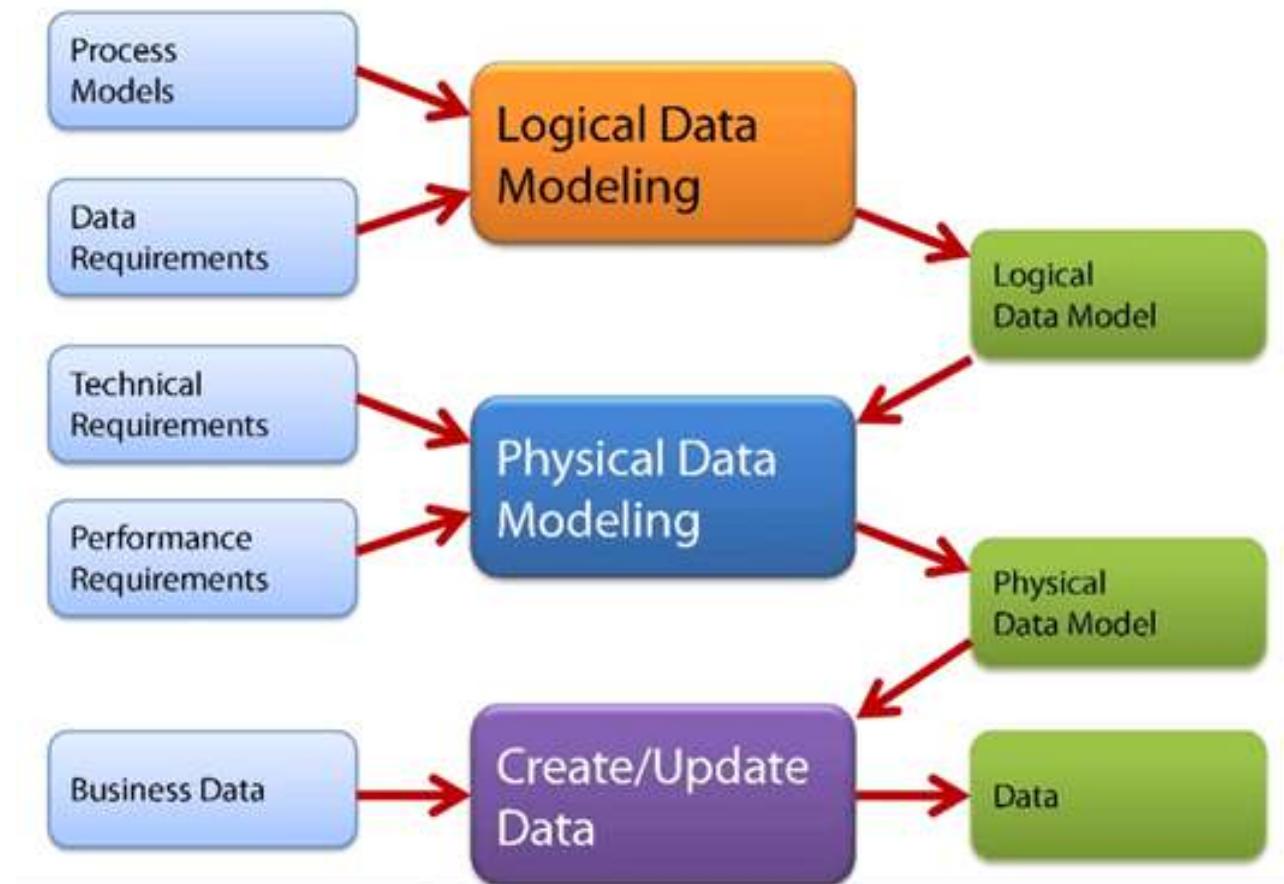
# Modeliranje podataka

## Levels of Data Modeling





# Modeliranje podataka

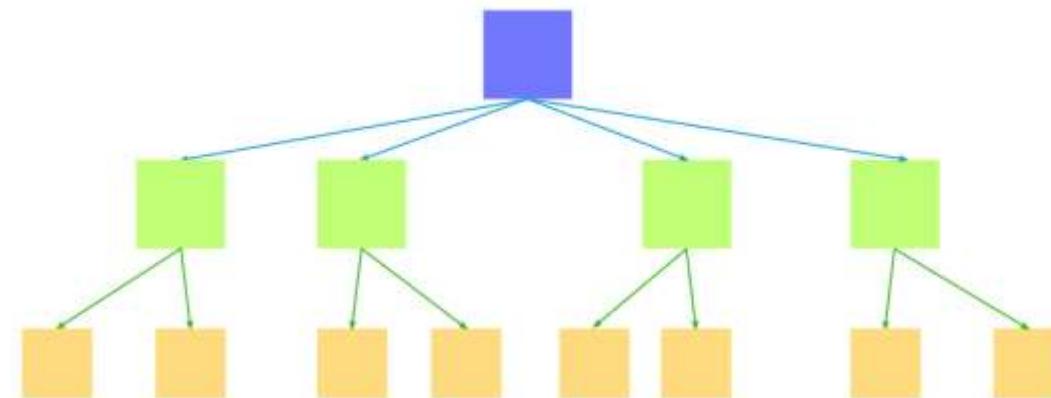




# Vrste modela podataka

## Hijerahiski model

- koristi hijerarhiju za strukturiranje podataka u obliku poput stabla.

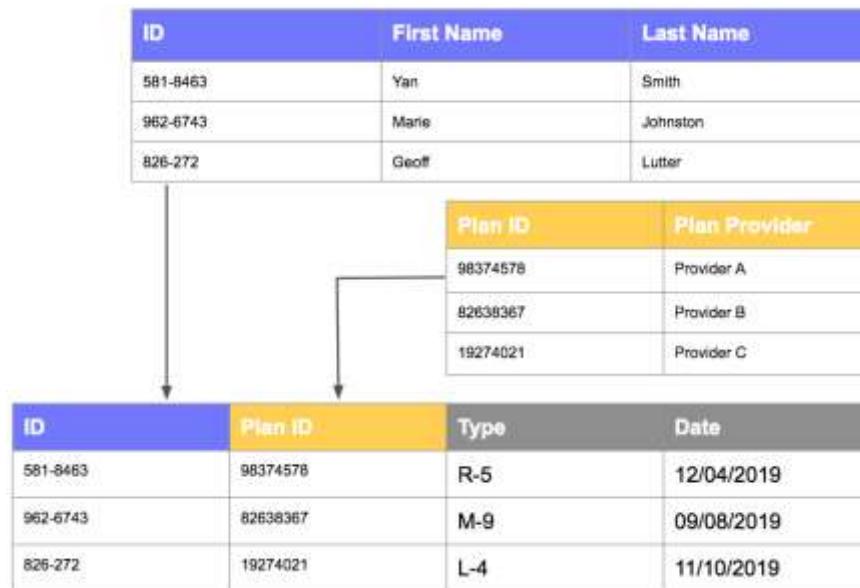




# Vrste modela podataka

## Relacijski model

– podaci predstavljeni u obliku tablica

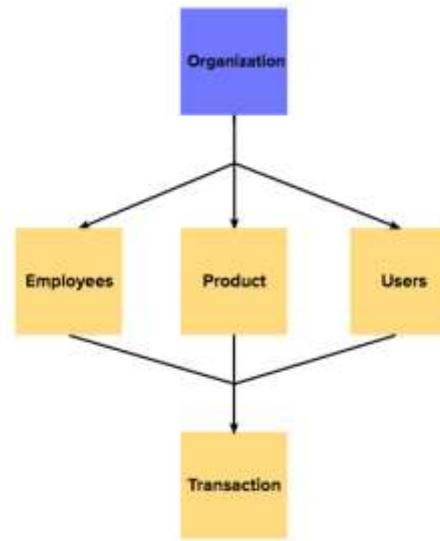




# Vrste modela podataka

## Mrežni model

Mrežni model je inspirisan hijerarhijskim modelom. No, za razliku od hijerarhijskog modela, ovaj model olakšava prenošenje složenih odnosa, jer se svaki zapis može povezati s više matičnih zapisa.

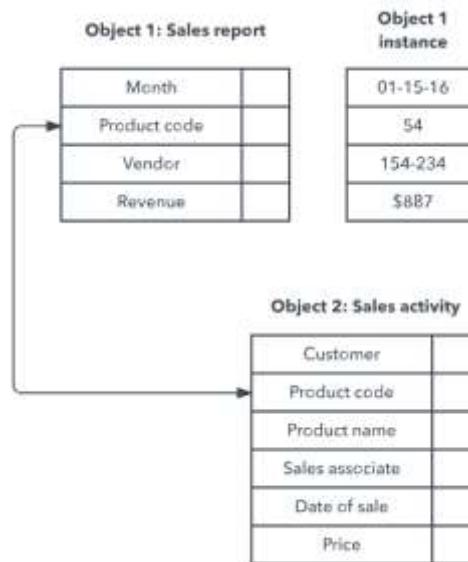




# Vrste modela podataka

## Objektno orijentiran model

- Sastoji se od kolekcije objekata, svaki sa svojim atributima i metodama. Ova vrsta modela baze podataka naziva se i post-relacijski model baze podataka.

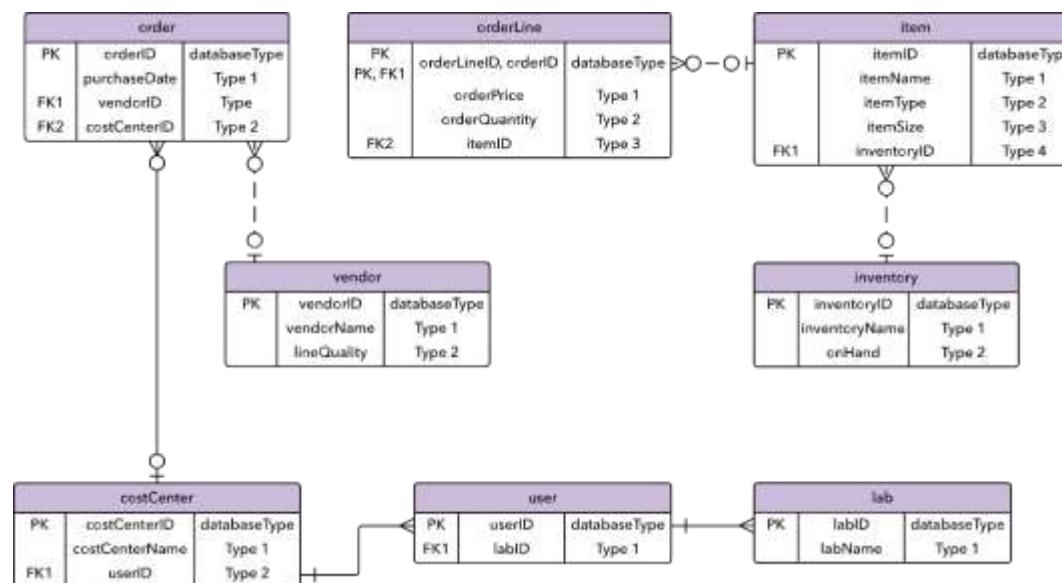




# Vrste modela podataka

## Entity Relationship model

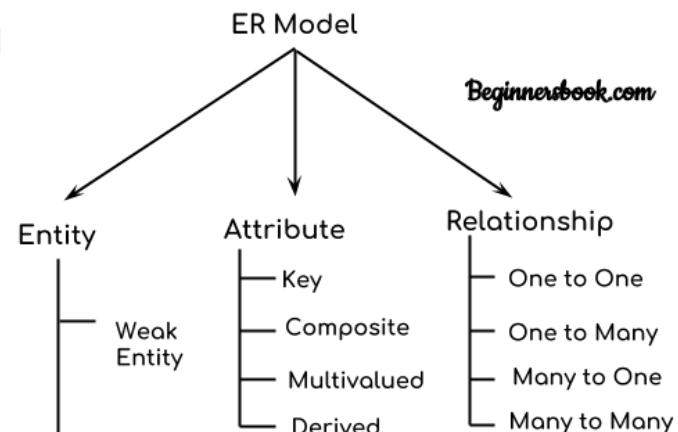
predstavlja entitete i njihove odnose u grafičkom formatu. Entitet može biti bilo što - koncept, podatak ili objekt.





# ER model

- Entitet: objekt ili komponenta podataka; osnovni gradivni blok za model podataka.
  - Može predstavljati osobu, mjesto, događaj ili objekat o kojem se prikupljaju podaci.
  - Naziv entiteta treba biti u obliku <</menica>>.
  - Slabi entitet je onaj koji ne posjeduje primarni ključ.

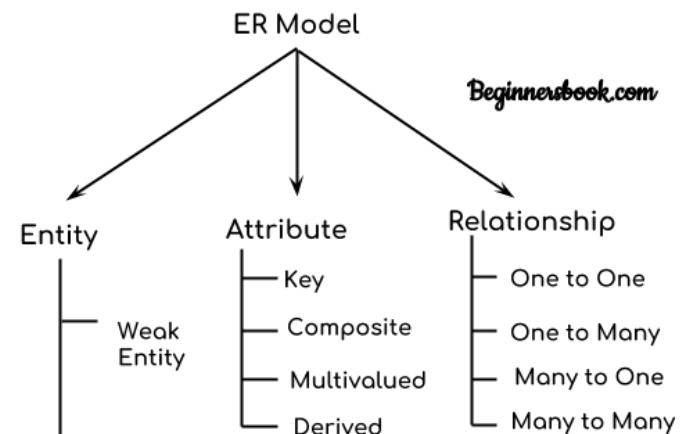


Components of ER Diagram



# ER model

- **Atributi:** Karakteristike ili svojstva entiteta
- Key attribute: jedinstvena karakteristika entiteta ( matični broj, broj lične karte, pasoša,...)
- Composite attribute: kombinacija drugih atributa poznata je kao složeni atribut. ( adresa: ulica, broj, država)
- Multivalued: Atribut koji može sadržavati više vrijednosti (Osoba može imati više telefonskih brojeva)
- Derived: izvedeni atribut je onaj čija je vrijednost dinamička i izведен je iz drugog atributa. (Starost osobe je izvedeni atribut te se s vremenom mijenja i može se izvesti iz drugog atributa (Datum rođenja)).

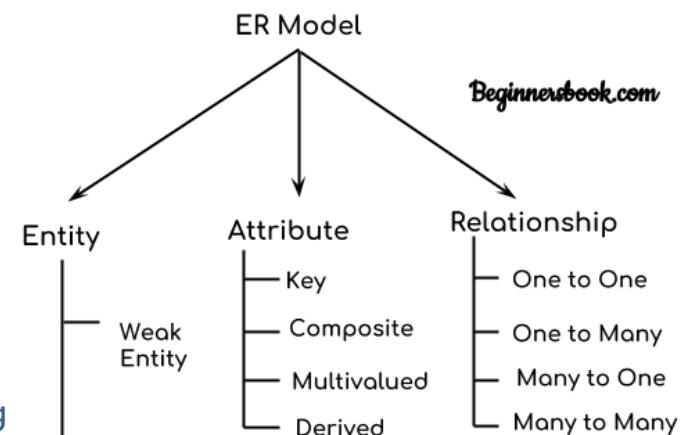


Components of ER Diagram



# ER model

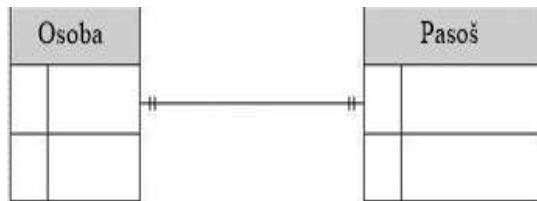
- **Veza:** Ovisnost ili povezanost između dva entiteta
- Veze se označavaju glagolima, koji trebaju pojasniti odnos između dva entiteta.
- **KARDINALNOST:** je odnos broja instanci entiteta i broja instanci drugog entiteta.
  - **One to One** : jedna instance entiteta povezana s jednom instancom drugog entiteta
  - **One to Many** : jedna instance entiteta povezana s više instanci drugog entiteta
  - **Many to One:** više instance entiteta povezano s jednom instancom drugog entiteta
  - **Many to Many:** više od jedne instance entiteta povezano je s više instance drugog entiteta



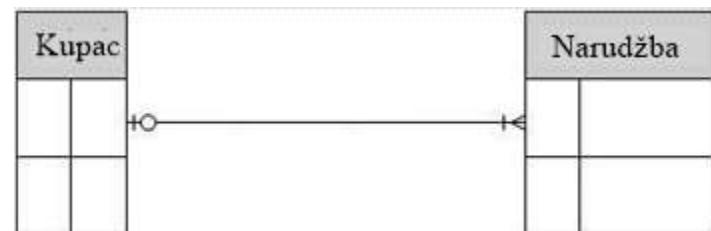
Components of ER Diagram



# Kardinalnost



**1:1 veza**



**1:N veza**



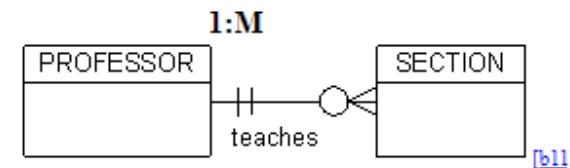
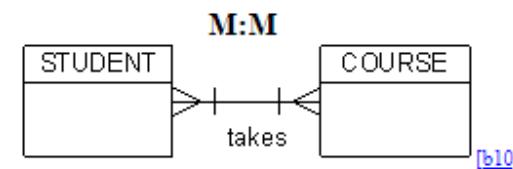
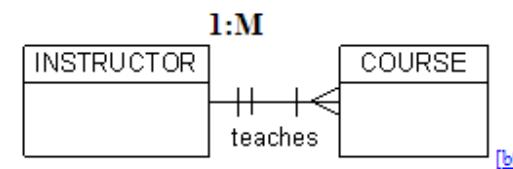
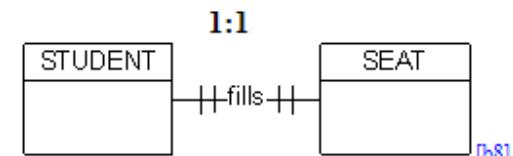
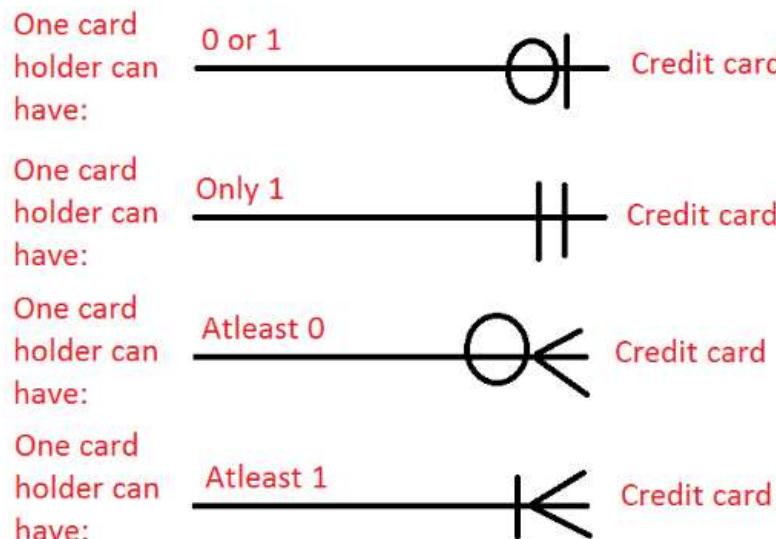
**M:N veza**



**N:1 veza**

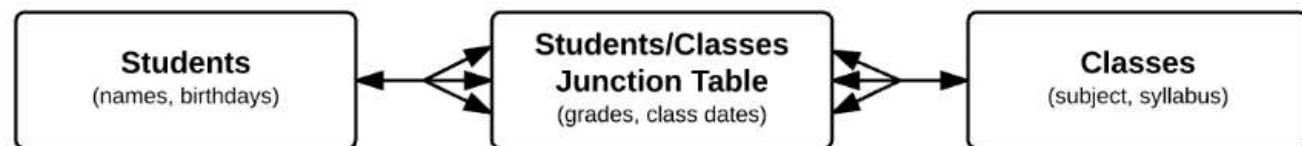
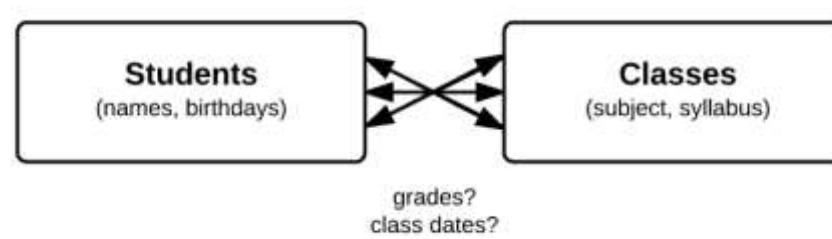


# Primjer





# Many to Many veza





# Many to Many veza



<https://support.airtable.com/hc/en-us/articles/218734758-A-beginner-s-guide-to-many-to-many-relationships>



# Normalizacija

Model podataka : fleksibilan, neredudantan, jednostavan

Normalizacija je tehnika koja organizuje atribute na način da formiraju neredudantne, stabilne, fleksibilne i prilagodljive entitete. Normalizacija uključuje tri koraka:

Prva  
normalna  
forma 1NF

Druga  
normalna  
forma 2NF

Treća  
normalna  
forma 3NF



## 1 NF

- Svaka ćelija tabele treba sadržavati jednu vrijednost .
- Svaki zapis mora biti jedinstven.

Students

FirstName	LastName	Knowledge
Thomas	Mueller	Java, C++, PHP
Ursula	Meier	PHP, Java
Igor	Mueller	C++, Java

Startsituation



Result after Normalisation

Students

FirstName	LastName	Knowledge
Thomas	Mueller	C++
Thomas	Mueller	PHP
Thomas	Mueller	Java
Ursula	Meier	Java
Ursula	Meier	PHP
Igor	Mueller	Java
Igor	Mueller	C++

Example First normal form



# 1 NF – primjer 2

**Cilj:** Postići da tablica sadrži samo atomske vrijednosti u svakoj ćeliji (tj. vrijednosti koje se ne mogu dalje podijeliti) i da svaki stupac ima jedinstven naziv.

## Koraci za postizanje 1NF:

1. Ukloniti višestruke vrijednosti: Pobrinite se da svaki stupac sadrži samo jednu vrijednost po retku. Ako postoje više vrijednosti (npr., lista telefona ili hobija), trebate ih razdvojiti u zasebne redove ili napraviti zasebne stupce.
2. Osigurati jedinstvenost stupaca: Svaki stupac treba imati jedinstven naziv i jasno definiranu domenu (tip i format podataka).



# 1 NF – primjer 2

- Izvorni podaci:

ID	Ime	Telefoni
1	John Doe	12345, 67890

- Transformirani u 1NF:

ID	Ime	Telefon
1	John Doe	12345
1	John Doe	67890



## 2 NF

- 1NF
- ako su svi atributi, koji nisu primarni ključ, ovisni od čitavog primarnog ključa
- Primary keys: IDSt, IDProf

Students

IDSt	LastName	IDProf	Prof	Grade
1	Mueller	3	Schmid	5
2	Meier	2	Borner	4
3	Tobler	1	Bernasconi	6

Startsituation

Students

ID	LastName
1	Mueller
2	Meier
3	Tobler

Result after normalisation

Professors

IDProf	Professor
1	Bernasconi
2	Borner
3	Schmid

Grades

IDSt	IDProf	Grade
1	3	5
2	2	4
3	1	6

Example Second normal form



## 2 NF – primjer 2

**Cilj:** Ukloniti parcijalne zavisnosti, tj. zavisnosti gdje neki atributi zavise samo od dijela primarnog ključa u tablici sastavljenoj od više atributa.

### Koraci za postizanje 2NF:

1. Identificirati i ukloniti parcijalne zavisnosti: Ako tablica ima složeni primarni ključ (sastavljen od više stupaca), svaki stupac koji nije dio ključa mora ovisiti o cijelom ključu, a ne samo o dijelu.
2. Razdvajanje tablica: Ako neki atributi ne ovisi o cijelom ključu, trebali bi biti izdvojeni u zasebnu tablicu gdje će formirati funkcionalnu cjelinu s dijelom ključa o kojem zavise.



## 2 NF – primjer 2

- Izvorni podaci u 1NF:

StudentID	CourseID	Instructor
1	CS101	Dr. Smith
1	MA101	Dr. Jones

- Transformirani u 2NF:

- StudentCourses:

StudentID	CourseID
1	CS101
1	MA101

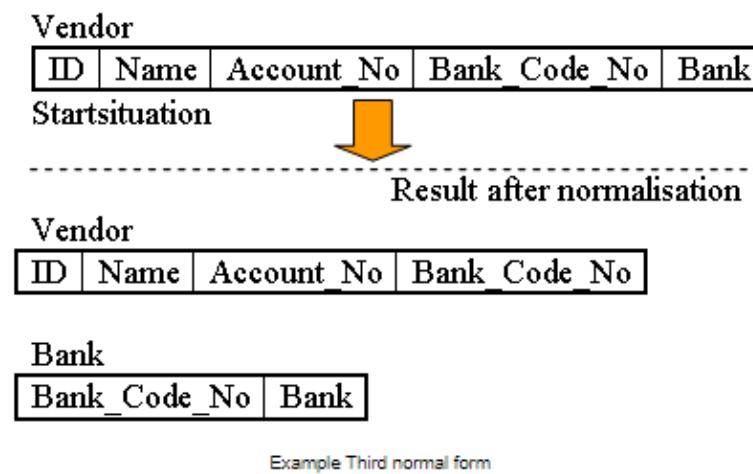
- Courses:

CourseID	Instructor
CS101	Dr. Smith
MA101	Dr. Jones



## 3 NF

- 2NF
- i ako su vrijednosti njegovih atributa, koji nisu primarni ključ, neovisne od drugih atributa, koji nisu primarni ključ.
- Primary Key: ID





## 3 NF – primjer 2

**Cilj:** Ukloniti tranzitivne zavisnosti, gdje atributi zavise od drugih atributa koji nisu ključ.

**Koraci za postizanje 3NF:**

1. Identificirati i ukloniti tranzitivne zavisnosti: Ako neki atribut zavisi od drugih neključnih atributa, te zavisnosti treba ukloniti.
2. Razdvajanje tablica: Atributi koji imaju tranzitivne zavisnosti trebaju biti izdvojeni u zasebne tablice.



# 3 NF – primjer 2

- Podaci u 2NF:

- Courses:

CourseID	Instructor	Office
CS101	Dr. Smith	Room 101
MA101	Dr. Jones	Room 102

- Transformirani u 3NF:

- Courses:

CourseID	Instructor
CS101	Dr. Smith
MA101	Dr. Jones

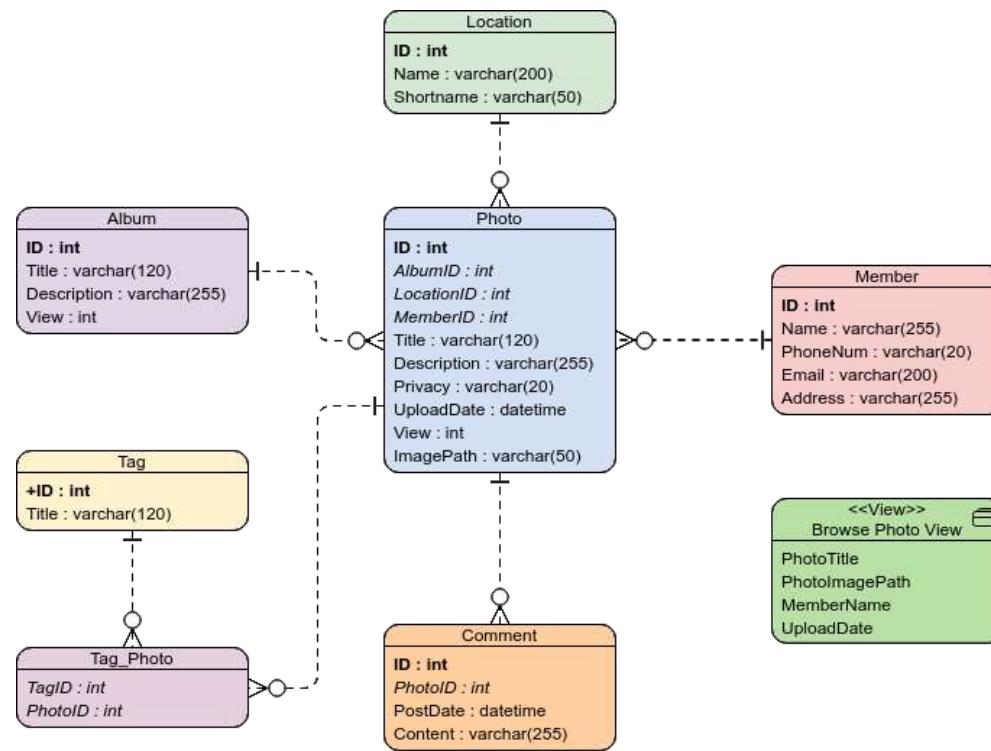
- Instructors:

Instructor	Office
Dr. Smith	Room 101
Dr. Jones	Room 102



# Kako kreirati ERD?

1. Identifikacija entiteta
2. Identifikacija atributa za svaki entitet
3. Kreiranje veza između entiteta



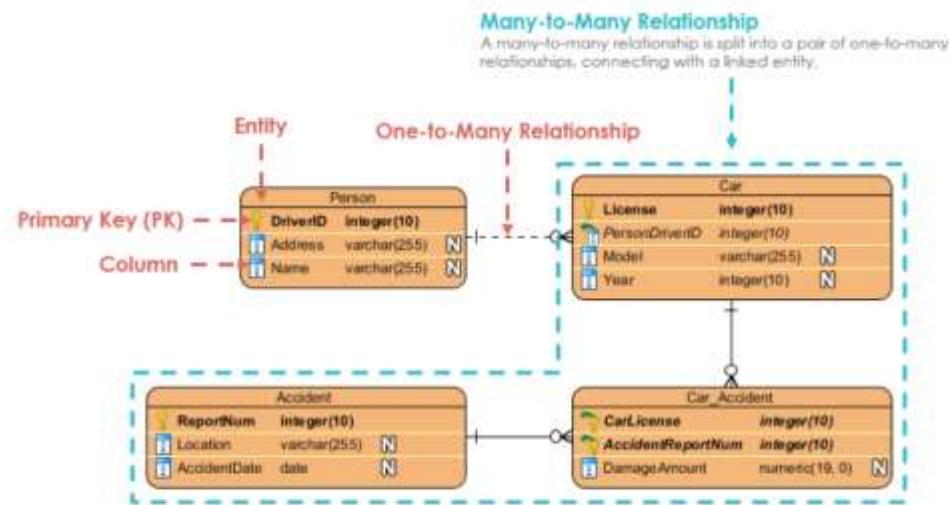


## Primjer

- Kreirati ERD za kompaniju za osiguranje automobila čiji klijent posjeduje po jedan ili više automobila. Svaki automobil može prouzrokovati 0 ili više nezgoda.
- Automobil je opisan modelom, godinom proizvodnje, tablicama.
- Osoba opisana ID vozača, adresom i imenom.
- Nezgoda opisana brojem štete, lokacijom i datumom.
- Također, nezgoda će u odnosu na automobil opisati i iznos štete.



## Primjer



This is an E-R diagram example for a car-insurance company whose customers own one or more cars each. Each car has associated with it zero to any number of recorded accidents.



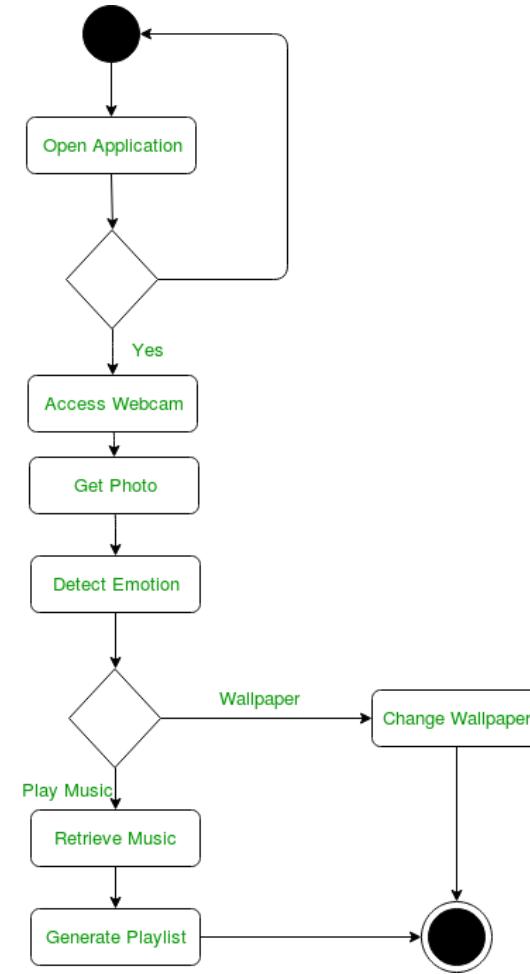
# Modeliranje procesa

Model procesa je način prikazivanja kako bi poslovni sistem trebao raditi. On ilustruje procese ili aktivnosti, te kako se podaci razmjenjuju između njih.



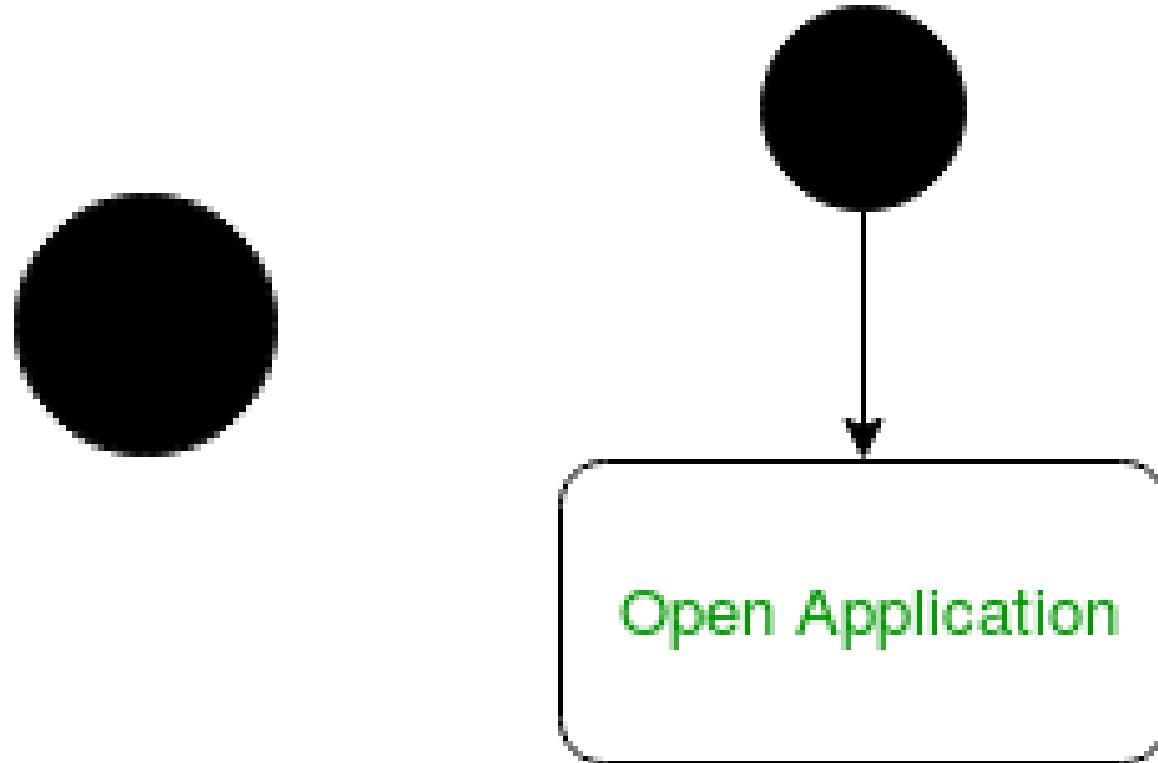
# Dijagrami Aktivnosti

Dijagram aktivnosti je dijagram ponašanja, tj. prikazuje ponašanje sistema.





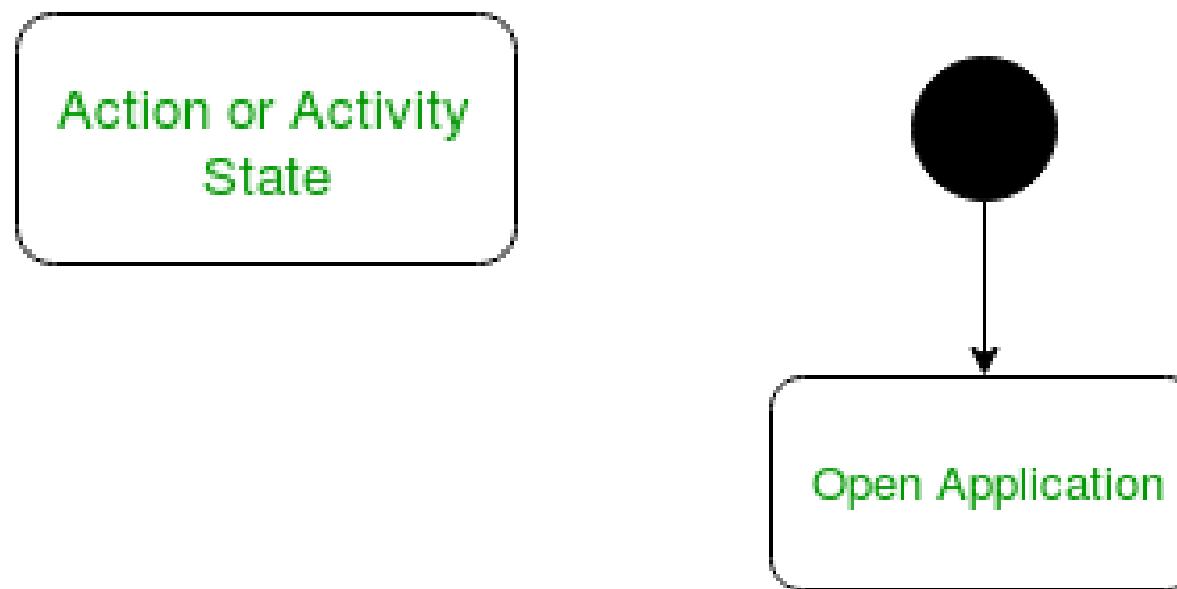
# Dijagrami Aktivnosti - Inicijalno stanje





## Dijagrami Aktivnosti - Akcija ili stanje aktivnosti

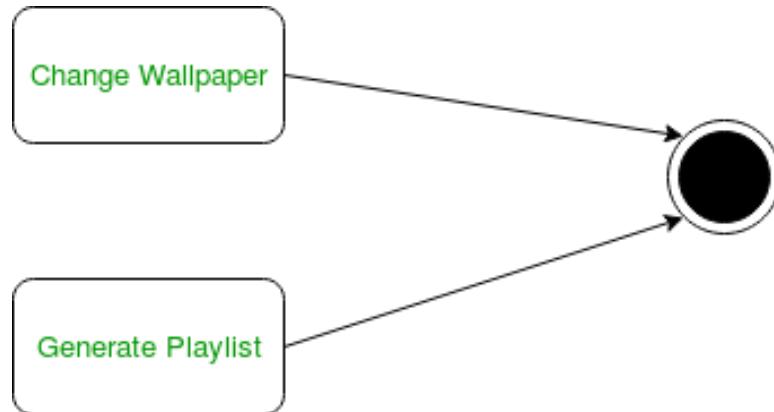
- aktivnost predstavlja izvršavanje akcije na objektima. Predstavljamo aktivnost pomoću pravougaonika sa zaobljenim uglovima. U osnovi svaka radnja ili događaj koji se odvija predstavljen je pomoću aktivnosti.





## Dijagrami Aktivnosti - Akcijski tok ili Kontrolni tokovi

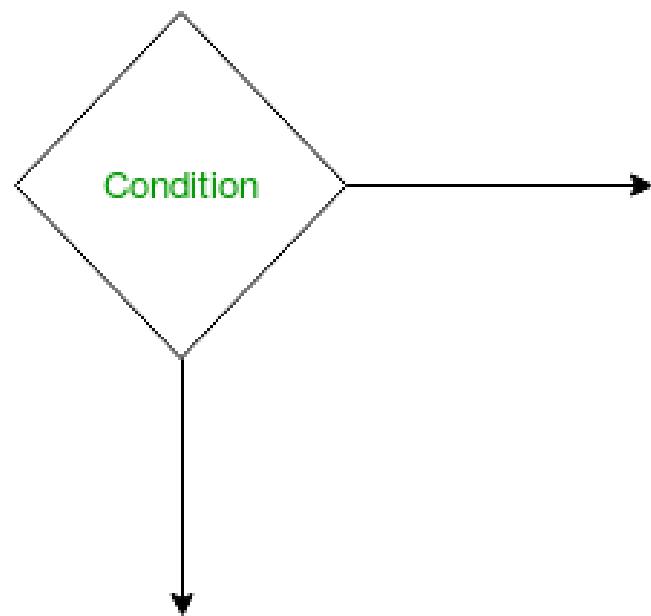
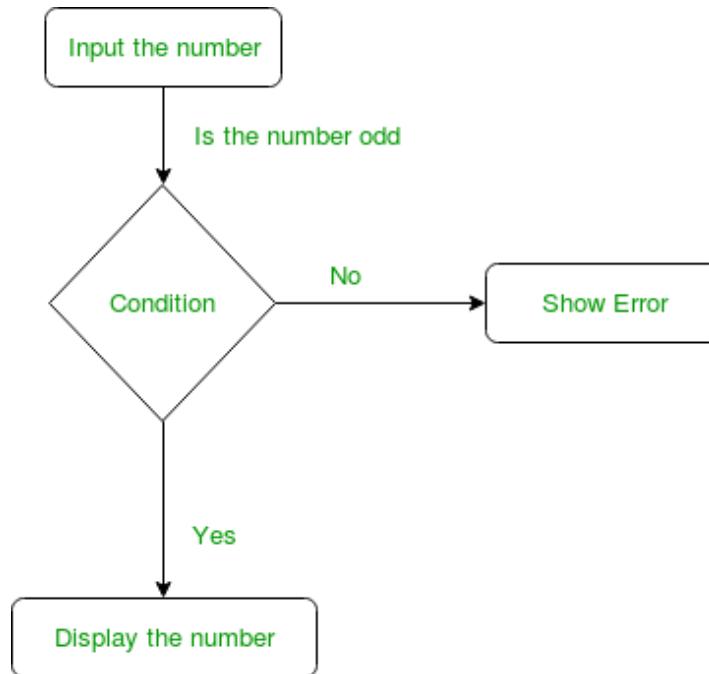
- Akcijski tokovi ili Kontrolni tokovi također se nazivaju putanje i rubovi. Koriste se za prikazivanje prelaska iz jednog stanja u drugo.





## Dijagrami Aktivnosti - Čvor odluke i grananje

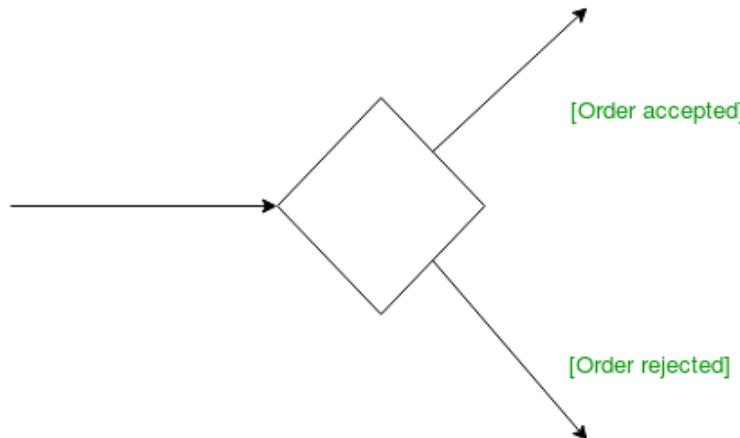
- Kad moramo donijeti odluku prije nego što odlučimo o protoku , koristimo čvor odluke.





## Dijagrami Aktivnosti - Čuvari

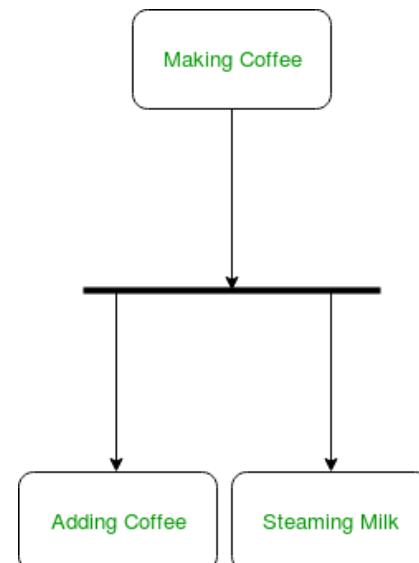
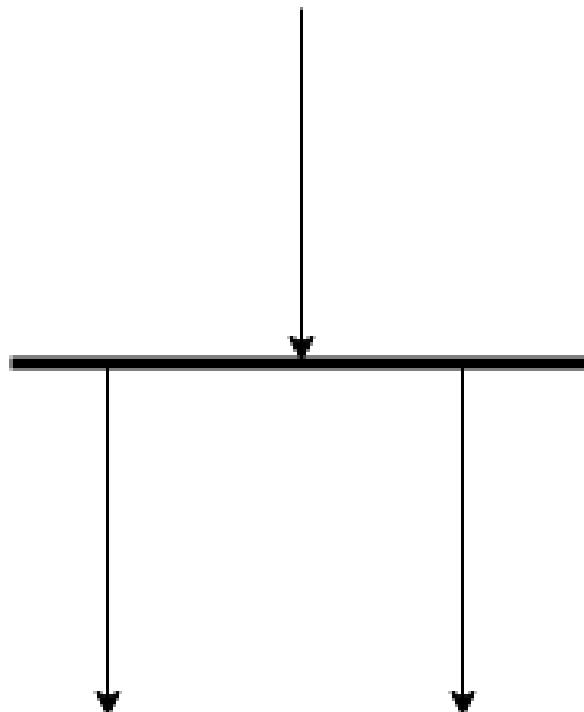
Čuvar se odnosi na izjavu napisanu pored čvora odluke na strelici, unutar uglastih zagrada.





## Dijagrami Aktivnosti - Fork

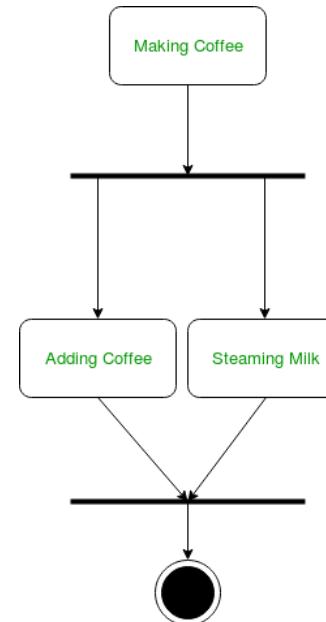
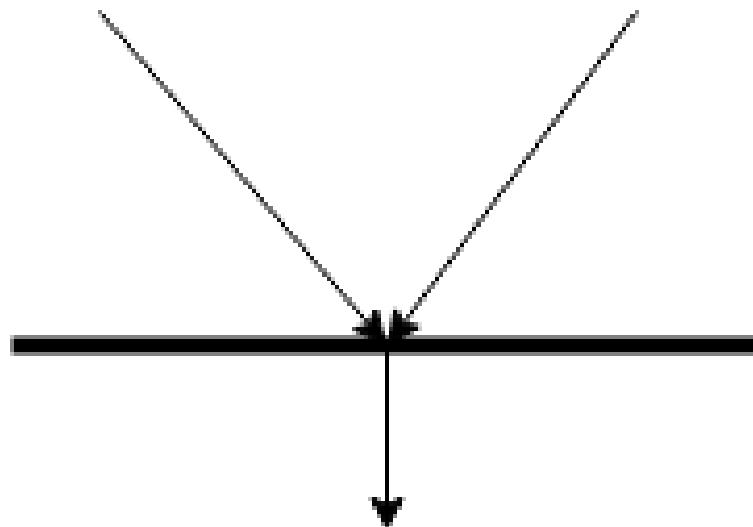
- Čvorovi Fork koriste se za podršku paralelnim aktivnostima.
- Odluka se ne dijeli prije podjele izvršavanja paralelnih aktivnosti. Oba dijela se moraju izvršiti u slučaju fork-a.





## Dijagrami Aktivnosti - Join

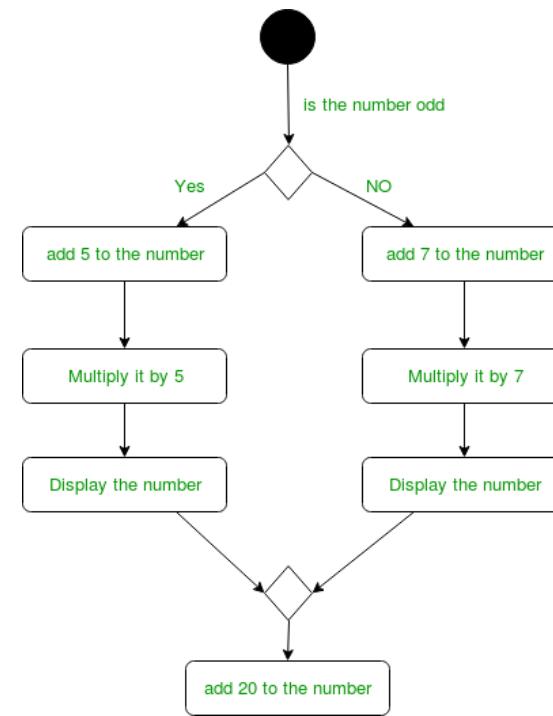
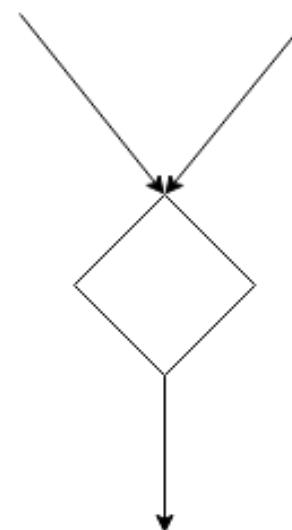
- Čvorovi pridruživanja koriste se za podršku istodobnim aktivnostima konvergiranjem u jednu.





## Dijagrami Aktivnosti - Spajanje ili spajanje događaja

- scenariji nastaju kada aktivnosti koje se ne izvršavaju istovremeno moraju biti spojene. Za takve scenarije koristimo oznaku spajanja. Dvije ili više aktivnosti možemo spojiti u jednu ako kontrola pređe na sljedeću aktivnost bez obzira na odabrani put.

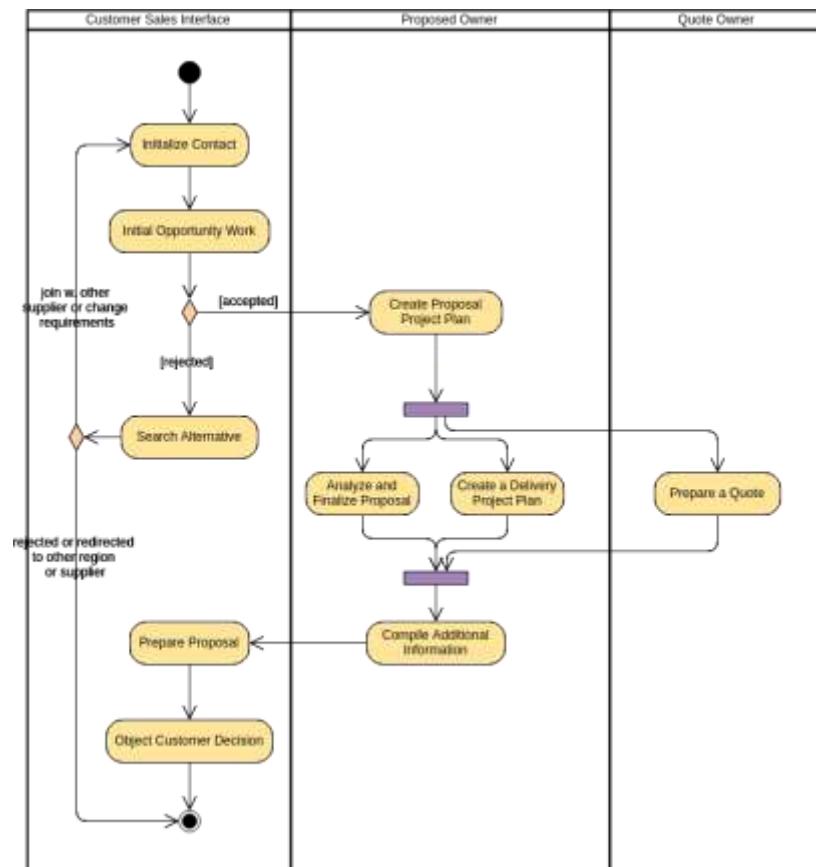
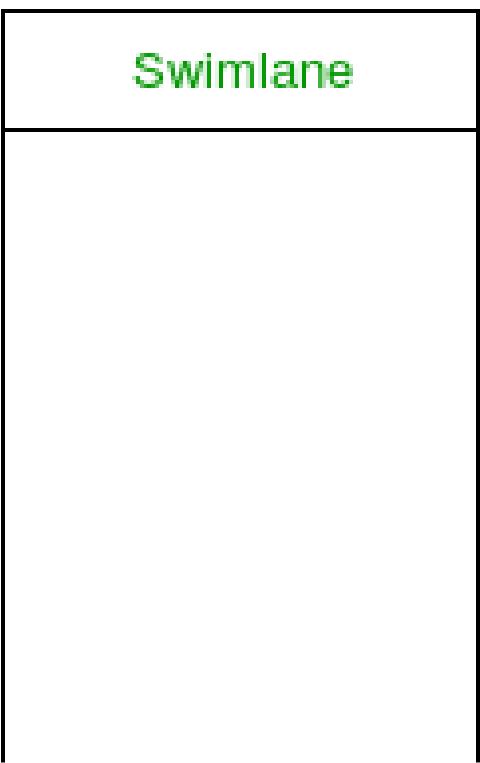




## Dijagrami Aktivnosti - Swimlanes

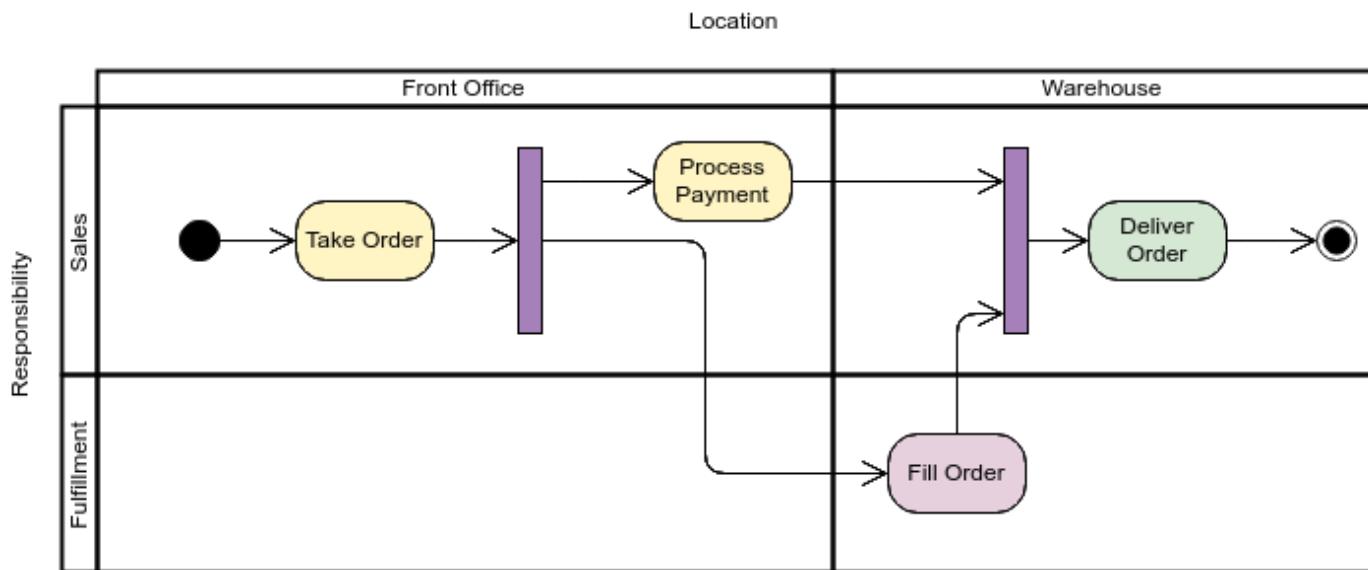
- Koristimo swimlanes za grupiranje povezanih aktivnosti u jednoj koloni. Swimlanes mogu biti vertikalni i horizontalni. Nije obavezna upotreba swimlanes-a. Obično daju više jasnoće dijagramu aktivnosti.

### Osnove informacionih sistema





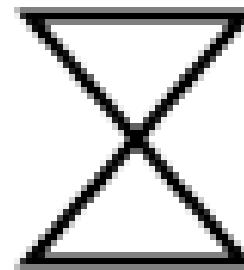
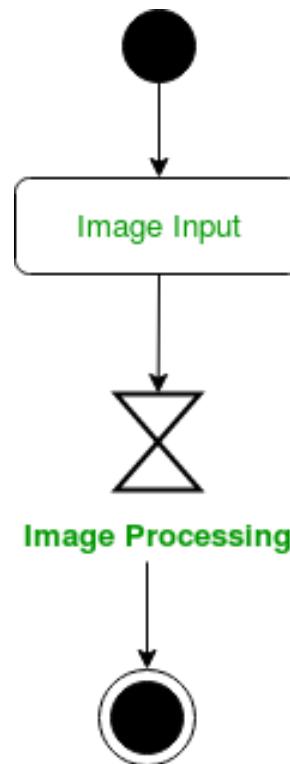
## Dijagrami Aktivnosti - Swimlanes





## Dijagrami Aktivnosti - Vremenski signal

Možemo imati scenarij u kojem je nekom događaju potrebno određeno vrijeme.



Time Event



## Dijagrami Aktivnosti - Signali

Signal ukazuje na to da aktivnost prima događaj iz spoljnog procesa.

- Vremenski signal:



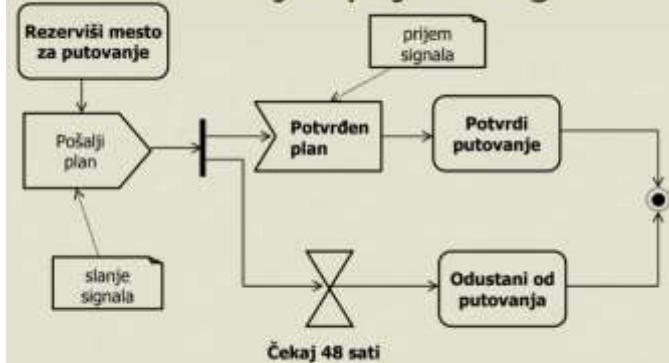
- Prijem signala:



- Slanje signala:



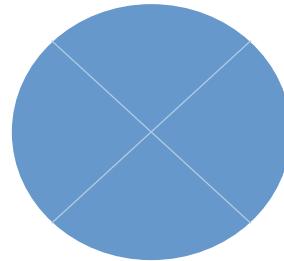
### Primer slanja i prijema signala





## Dijagrami Aktivnosti - Kraj toka

Označava kraj određenog toka, ali ne i kraj kompletne aktivnosti.



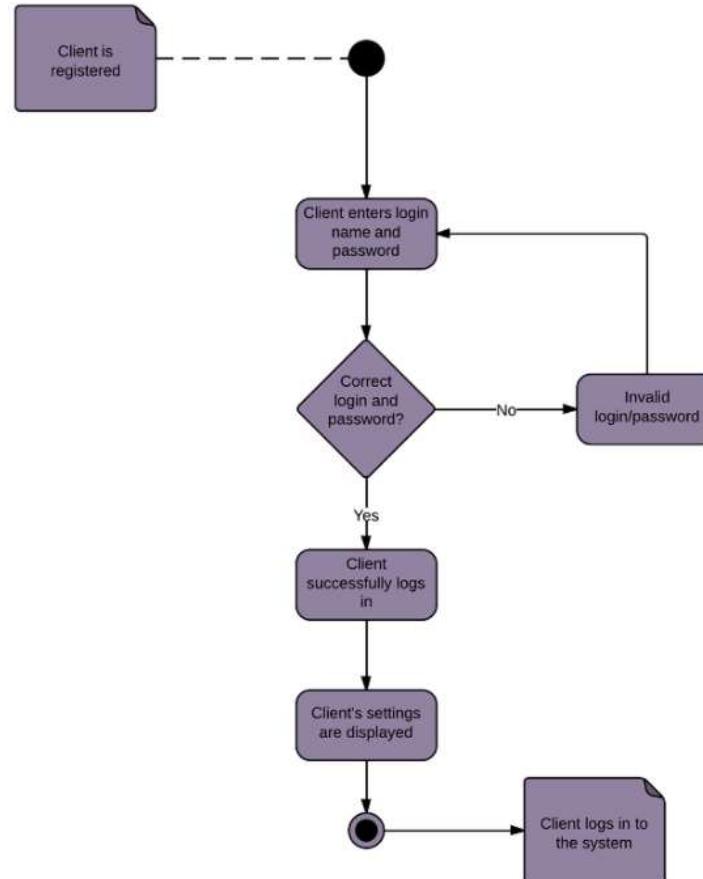


# Dijagrami Aktivnosti - Primjer

- Nacrtati dijagram aktivnosti za login akciju Sistema. Korisnik unosi username i password i kada su oni ispravni se uspješno loguje na sistem.
- U suprotnom mu se otvara prozor sa porukom o nevalidnosti kredencijala te ga se ponovo vraća na login stranicu.
- Kada se korisnik uspješno loguje prikazane su mu njegove postavke.



# Dijagrami Aktivnosti - Primjer



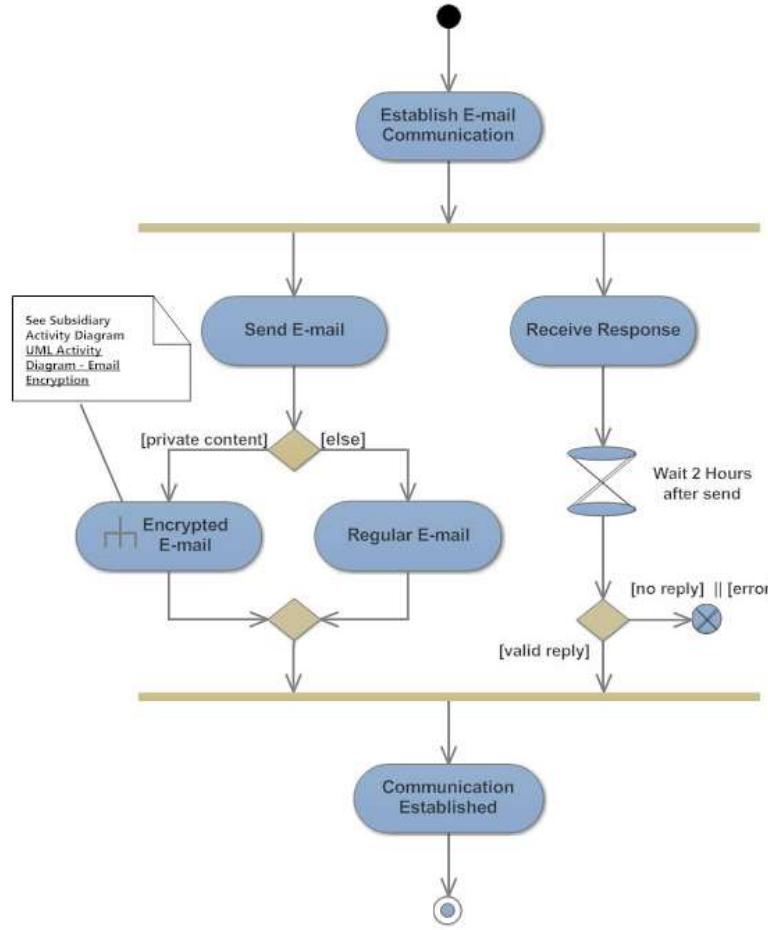


# Dijagrami Aktivnosti - Primjer

- Kreirati dijagram aktivnosti za email komunikaciju. Prvobitno se uspostavlja email komunikacija te se nakon toga paralelno izvrše radnje slanja emaila i primanja odgovora.
- Ukoliko je sadržaj privatan, šalje se enkriptovan email, u suprotnom sistem šalje standardan email. Nakon slanja bilo koje vrste emaila, sistem jednako nastavlja sa radom.
- Kada je riječ o primanju odgovora, sistem čeka 2 sata na odgovor te provjerava da li je odgovor validan ili ne. Ukoliko je odgovor nevalidan ili ne postoji prekida se izvršenje, u suprotnom izvršenje se nastavlja.
- Nakon završetka svih pomenutih paralelnih aktivnosti, sistem potvrđuje uspostavu email komunikacije.



# Dijagrami Aktivnosti - Primjer





# Pitanja i odgovori

Osnove informacionih sistema

**Pitanja i odgovori ???**



# Projektni zadatak

- Modeliranje procesa za vaš IS:
  - Activity dijagrami
- Modeliranje podataka za vaš IS :
  - ER dijagram
- Izvještaj
- Projektni zadatak poslati do 16.12.2024. 23:59 na email:  
[amer.koric@etf.unsa.ba](mailto:amer.koric@etf.unsa.ba)