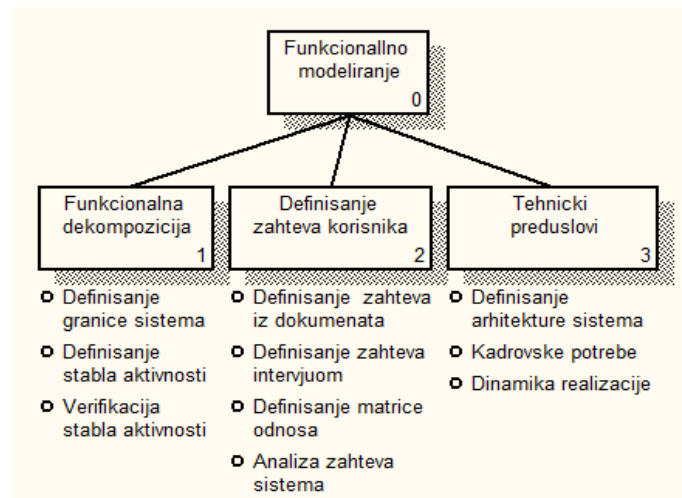
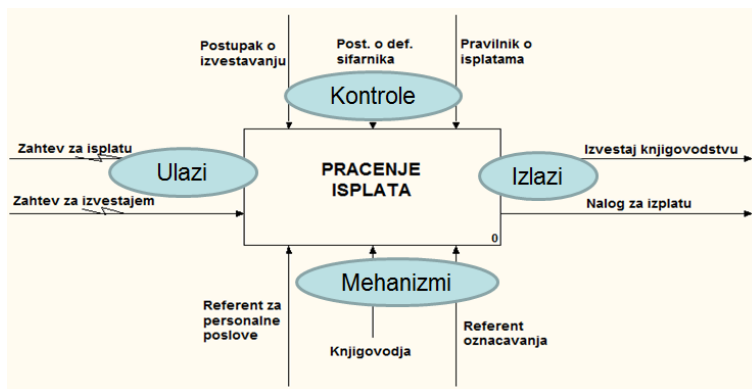


SKRIPTA IZ INFORMACIONIH SISTEMA

--- 01 - Informacioni sistemi u poslovanju organizacije ---

- **Model** = proces sa elementima projekta; pojednostavljeni realni sistemi.
- **Metode za dobijanje informacija o IS** = Top-Down (intervju) - širina, Bottom-up (analiza) - preciznost.
- **Ograničenja koja usporavaju razvoj IS**: neorganizovanost, odbojnost, neznanje.
 1. Poslovni sistem sa većim stepenom organizovanosti pruža manja ograničenja.
 2. Veći stepen uključenosti korisnika u razvoj informacionog sistema smanjuje ograničenje.
 3. Veći stepen znanja i iskustva projektnog tima pruža manja ograničenja.
- **Pretpostavke koje omogućavaju razvoj IS**: ID, CASE alati, DBMS.
 1. Jedinstven sistem označavanja (**ID**) - zasnivaju se na osobini jedinstvenosti i neponovljivosti.
 2. Jedinstvenost modela procesa i podataka (**CASE alati**) : IDEF0 (BPwin) i IDEF1X (ERwin)
 3. Jedinstvenost sistema za upravljanje bazama podataka (SUBP/DBMS) - MS ACCESS
- **Apstrakcija** = kontrolisano isključivanje detalja tj. izvlačenje zajedničkih karakteristika pri opisu sistema.
- **Tehnologija** = razvoj i primena alata, mašina, mat. i procesa koji pomažu u rešavanju ljudskih prob.
- **Tehnika** (umeće, veština) = skup svih znanja i oruđa proizvodnje.
- **Cilj modeliranja** = razvoj tehnologija koje omogućavaju integraciju mreža koje se razlikuju hardverski i softverski.
- **IDEF tehnike** = skupovi grafičkih prikaza, korist da bi ljudi mislili o modeliranju kao o slikovitoj prezentaciji.
- **Reinženjering** = radikalni redizajn poslovnih procesa.
- **Standardi** = IDEF0 - tehnika za funkcionalno modeliranje, IDEF1X za informaciono. *FM postavlja model.
- **Funkc. modeliranje** = dekompozicija, stvaranje dokumentacije, reinž., komunikacija, stvara elemente za IM.
- **Informaciono modeliranje** = pojednostavljeno predstavljanje sistema preko entiteta, atributa i njihovih veza.
- *** 4 faze razvoja IS** = funkcionalno > informaciono > aplikativno modeliranje > implementacija.
- + **Funkcionalno modeliranje** = funkci. dekompozicija > definisanje zahteva korisnika > tehnički preduslovi.
- + **Informaciono modeliranje** = kreiranje ER modela > kreiranje atributa > definisanje poslovnih pravila.
- + **Aplikativno modeliranje** = definisanje fizičkog dizajna > generisanje šeme baze podataka > izrada aplikacije.
- + **Implementacija** = uvođenje > testiranje > održavanje.
- **FM reinženjeringa poslovnog procesa** = postavljanje modela > redizajn > realizacija > implementacija.
- Dijagram konteksta (za sl. lekciju kod dela sa strelicama) i IDEF0 stablo aktivnosti:



--- 02 - Funkcionalna dekompozicija ---

- **Primitivni proces** = najniža aktivnost koju nema smisla više razlagati. Manja od toga je **Atomska aktivnost**.

- **Hijerarhija menadžmenta**:

- * Vrhunski men. | Dugoročni ciljevi | OLAP | Strategijski nivo | Planiranje | Šta treba da se uradi?
- * Men. sred. niv. | Srednjoročni | OLAP/DW | Taktički nivo | Kontrola | Kako treba da se uradi?
- * Men. prve lin. | Kratkoročni | OLTP | Operativni nivo | Nadzor | Ko to treba da uradi?

- **Dijagram konteksta**: ulazne, izlazne, kontrolne i strelice mehanizma definišu granice sistema.

- * **ULAZI** - šta je sve potrebno obezbediti glavnoj aktivnosti da bi se ona odvijala.
- * **IZLAZI** - šta sve očekujemo da će glavna aktivnost dati kao rezultat svog odvijanja.
- * **KONTROLE** - koja su to pravila, standardi i propisi kojih se glavna aktivnost mora pridržavati.
- * **MEHANIZMI** - ko su sve izvršioci koji svojim veštinama učestvuju u odvijanju glavne aktivnosti.

* **Definisanje granice sistema**

- **Sintaksa grafičkog jezika IDEF0** = *pravougaonici, strelice, pravila*. (boxes, arrows, rules)
- **Sintaksa** = bavi se pravopisom ; **Semantika** = izučava šta nešto znači (npr. značenje strelice).
- **Pravougaonik (boxes)** = predstavlja funkcije, definisane kao aktivnosti, procesi i transformacije.
- **Aktivnost unutar pravouganika** ima 3 k-ke = naziv, vremensku dimenziju, rezultat rada.
- * Razlika između aktivnosti u okviru projekta i tekućih operacija = jedna ima p i k, druga nema predviđen kraj.
- **Strelice** = predstavljaju podatke ili objekte vezane za funkcije.
- **Pravila** = definišu kako se komponente koriste, a **dijagrami** omogućuju prikaz modela (graf. i tekstualno).
- * Pr.: Box number se smešta u gornji desni ugao IDEF0 pravouganika, strelice imenovati sa 1-4 reči...

Strelice

- * **Ulazne** = predstavljaju materijal/informaciju koja se koristi/transformiše radi definisanja izlaza.
- * **Kontrolne** = regulišu kako će se aktivnost izvesti tj. kakvi će izlazi biti.
- * **Izlazne** = materijali ili informacije stvorene aktivnošću.
- * **Mehanizmi** = izvori koji izvide aktivnosti, a sami se ne „troše“. Mogu biti ljudi, mašine, oprema...
- * **Poziv** (mehanizam u sup. smeru) = označava da pozivajući pravougaonik nema sopstveni dijagram.
- **ICOM dijagrami** (drugi naziv za IDEF0 dijagrame) : Input, Control, Output, Mechanism.
- **Input** = nešto što se koristi u aktivnosti. ; **Control** = kontrole ili uslovi izvođenja aktivnosti.
- **Output** = rezultat izvođenja aktivnosti. ; **Mechanism** = nešto što se koristi u aktivnosti ali se ne menja.
- **Kontekstni dijagram** = A0 ; **Prvi nivo dekompozicije** = A1.
- **Širina modela** = vezana za definisanje kontekstnog dijagrama i prvog nivoa dekompozicije.
- * Koristi se za posmatranje problematike sa manje detalja.
- **Dubina modela** = definisana nivoima dekomponovanja.

* **Definisanje stabla aktivnosti** (slika iz prethodne lekcije)

- **Postupak definisanja** = do stabla se dolazi dekomponovanjem/razlaganjem složenijih na detaljnije aktivnosti.
- * **Prvi nivo** dekompozicije = A1, A2, A3. ; **Drugi nivo** dekompozicije (prve aktivnosti) = A1.1, A1.2, A1.3.
- Aktivnost na vrhu (root) uvek je označena sa **0**. Nadređena aktivnost se zove **parent**, a podređena **child**.

* **Verifikacija stabla aktivnosti**

- Momenat gde rukovodstvo odlučuje da li je to ono što se želi postići reinženjeringom poslovnih procesa.

--- 03 - Definisane zahteva korisnika ---

- Druga podaktivnost glavne aktivnosti **funkcionalno modeliranje**.

1. Definisane zahteva iz dokumenta - Bottom Up
2. Definisane zahteva intervjuom - Top Down
3. Definisane matrice odnosa - CRUD
4. Analiza zahteva korisnika - verifikacija

1. Definisane zahteva iz dokumenta = od jednostavnog ka složenom, od manjeg ka većem, uz piramidu.

* Pr.: Izrada mašine (deo, podsklop, sklop, mašina) : (tranzistor, grafička kartica, matična ploča, računar).

- Ovaj pristup omogućava **prikupljanje izvornih zahteva** (dokumenata), **preciznost i detaljnost**.

- Pitanja u vezi dokumenata = kako se koriste, koliko često se koriste, odakle potiču podaci...

- **Organizacioni propisi** = proučavanje postojećih propisa, naknadno pisanje nepostojećih propisa.

* Za postupak rada sa dokumentima definisane su odgovarajuće procedure i organizacioni propisi.

- **Analiza dokumenata** pomaže analitičaru da formira mišljenje o čitkosti tih dokumenata.

2. Definisane zahteva intervjuom = od složenijeg ka jednostavnijem, od većeg ka manjem, niz piramidu.

* Pr.: Izrada pravnog sistema (ustav, zakoni, podzakonska akta, pravilnici...)

- Ovaj pristup omogućava definisanje **potrebe za informacijama, ciljeva, korisničkih zahteva**.

- Opšte pripreme za izvođenje intervjua = vreme, teme za razgovor, potvrde termina...

- **Cilj intervjua** = identifikovanje redundantnosti podataka, razjašnjenje odgovornosti, razumevanje procesa poslovanja.

- Razumevanja procesa poslovanja = koje su nadležnosti i odgovornosti, šta su osnovni ciljevi...

- **Nakon intervjua** = razmatranje odgovora, pisanje rezimea, izdvajanje problema, definisanje zaključaka.

* Pr. definisanja zahteva za karton isplata na kome nisu svi podaci = nivoi znanja jezika, stručna sprema, radno mesto...

3. Definisane matrice odnosa = CRUD (Create, Retrieve, Update, Delete)

- Dokumenti su definisani odgovarajućim rubrikama, čijom se analizom određuju odgovarajući **entiteti**.

- **Entitet** = prezentacija realnih i apstraktnih stvari koje se prepoznaju pod istim tipom podataka jer dele k-ke.

- Entiteti na ovom nivou predstavljaju objekat koji se može opisati nekim osobinama.

- Entitet se u okviru neke aktivnosti kreira (**Create**), pretražuje (**Retrieve**), ažurira (**Update**), i briše (**Delete**).

- **Dokument** (levo) i **CRUD** matrica za primer aktivnosti „*Praćenje isplata*“ (desno).

KARTON ISPLATA			
SIFRA RADNIKA	IME I PREZIME	STRANI JEZIK	ODELJENJE
7359	Zoran Starcevic	Engleski, Ruski	Razvoj
RADNO MESTO: Tehnolog		Francuski	
R.BR. ISPLATE	IZNOS ISPLATE	DATUM ISPLATE	PRIMEDBE
01	150,00	08.12.1996	
02	450,00	20.02.1997	
03	800,00	30.09.1997	

Naziv aktivnosti	Naziv entiteta	CRUD
ODRZAVANJE PODATAKA O PREVODIJCIMA	ISPLATA	C R U D
	OSOBA	C R U D
	JEZIK	R
	CERTIFIKAT	C R U D
	ODELJENJE	R
ODRZAVANJE SIFARNIKA	R.MESTO	R
	JEZIK	C R U D
	ODELJENJE	C R U D
IZRADA IZVESTAJA	R.MESTO	C R U D
	ISPLATA	R
	JEZIK	R
	ODELJENJE	R
	OSOBA	R
	CERTIFIKAT	R

4. Analiza zahteva korisnika

- Postavke definisane u prethodnim koracima treba da verifikuje odgovarajući verifikacioni organ preduzeća.

- Zadatak verifikacionog tela = usvaja rezultate rada stručnog tima u kontrolnim tačka - krajevima pojedinih faza projekta.

* Pr. zahteva vlasnika sistema = da se izradi softver koji će omogućiti knjigovodstvo, da se hardver obnovi...

--- 04 - Tehnički preduslovi ---

- **Tehnički preduslovi** = sistem hardver (opipljivo), sistem softver (neopipljivo), sistem dokumentacije (opis prva 2).

1. Sistem hardvera :

- **Radna memorija** = sadrži operativnu (RAM) i postojanu (ROM, PROM, EPROM) memoriju.

- **Masovna memorija** = magnetni diskovi, diskete, optički diskovi, magnetne trake, kasete, CD.

- **Ulazna jedinica** = omogućuje da se u računar unose instrukcije, programi, podaci (tastatura).

- **Izlazna jedinica** = koristi se da se preko nje saopštavaju rezultati rada računara (monitor).

- **Centralna procesorska jedinica** = izvršava instrukcije uskladištene u operativnoj memoriji.

2. Sistem softver = operativni sistemi, jezički procesori (interpreteri i kompajleri), aplikativni softver.

3. Sistem dokumentacije = uputstva za korišćenje kao i razna korisnička dokumentacija.

*** Definisane arhitekture sistema**

- **Cilj** = ukazivanje na osnovne uslove koji se moraju ispuniti da bi došlo do reinž. poslovnih procesa.

- **Tehnika** = skup svih oruđa i znanja proiz. koja omogućuju čoveku delovanje na prirodu u svoju korist.

- **Tehnologija** = razvoj i primena alata, mašina, mat. i procesa koji pomažu u rešavanju ljudskih problema.

- **Tehničko-tehnološki preduslovi:**

* Kvalitet komunikacionog sistema = nezavistan od rač. opreme, obezbeđ. neogr. kom. među svim kor.

* Kompatibilnost računarske opreme = radi njenog povezivanja i međusobnog rada.

* Fizički i logički nivo povezivanja = sistem zahteva kompatibilnost računarske opreme radi povezivanja.

* Aplikativni nivo povezivanja = kompatibilnost aplikativnog nivoa i nivoa predstavljanja pdtk prema OSI.

- **OSI** (otvoreni sistem međupovezivanja) = najkorišćeniji apstraktni opis arhitekture mreže.

* Opisuje interakciju uređaja (hardware-a), programa, servisa (software-a) i protokola pri mrežnim komunikacijama.

- **7 nivoa povezivanja i komuniciranja u arhitekturi (7 slojeva OSI sistema):**

1. Fizički sloj = transmisija signala, prevodi zahteve od sloja veze u specifične operacije u tehničkim uređ.

* Ovaj sloj definiše električna i fizička svojstva mrežnih uređaja. Definišu se naponski nivoi, broj parica...

2. Sloj veze = fizičko adresiranje i pristup medijumu, zadužen za postavljanje, prijem i prenos paketa putem lokalne veze dvaju čvora. Brine se za razmenu podataka između mrežnih uređaja, i za korekciju mogućih grešaka u fizičkom sloju.

3. Mrežni sloj = opisuje protokole i servise koji obezbeđuju identifikaciju korisnika mreže, kao i rute među njima.

4. Transportni sloj = zadužen za segmentaciju podataka sa viših nivoa, tj. za njihovo ponovno spajanje.

5. Sloj sesije = uspostavljanje veze između krajnjih korisnika i sinhronizacija iste. Pr. ne želimo video bez tona.

6. Sloj prezentacije = odgovara na zahteve sloja aplikacije i dalje ih prosleđuje sloju sesije.

* Njegove primarne funkcije su kodiranje i konverzija, kompresija i enkripcija podataka.

7. Sloj aplikacije = opisuje rad aplikacija u mreži i njihovu interakciju sa servisima/protokol. nižih slojeva.

- **Dva osnovna oblika/modula opreme u k/s arhitekturi za potrebe distribuiranog sistema:**

1. računarska oprema zajedničkih resursa (strana servera).

2. računarska oprema krajnjeg korisnika (strana klijenta).

- **Distribuirani sistem** = sastoji se od više samostalnih rač. koji komuniciraju putem računarske mreže.

*** Kadrovske potrebe**

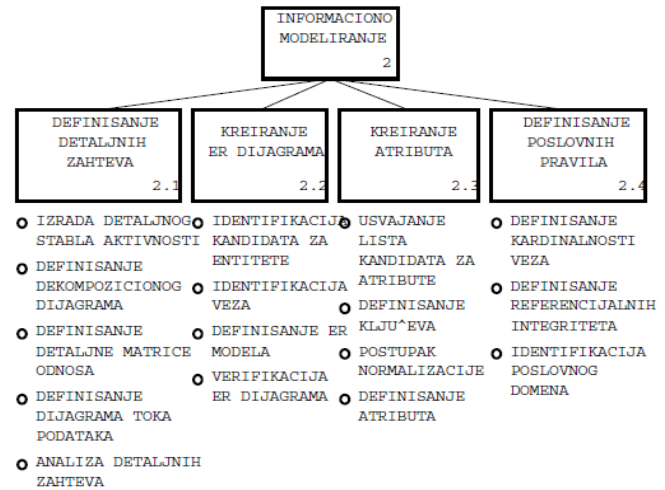
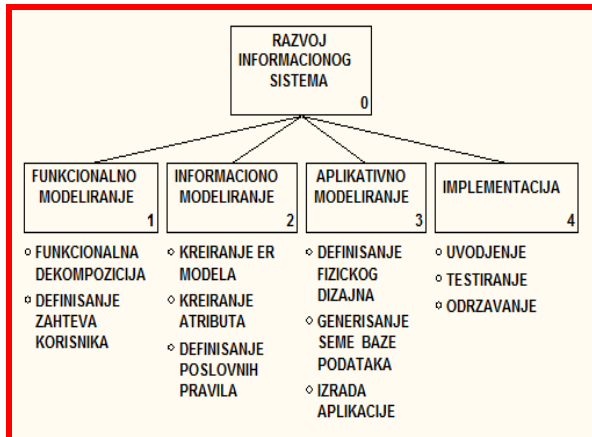
- Minimum potreban kadar = rukovodilac, projektant za modeliranje procesa/softverskih rešenja/baze...

- Obrazovanje kadrova = kompjutersko opismenjavanje; modeliranje procesa **BPwin**; mod. pdtk. **ERwin**.

*** Dinamika realizacije**

- Troškovi real. se posmatraju u okviru grupa kao : t. razvoja aplikacija; t. teh-teh. res; t. eksploatacije.

--- 05 - Kreiranje ER modela ---



- **Informacioni model** = prikazuje u kakvom su međusobnom odnosu podaci u nekom realnom sistemu.

** **Definisanje detaljnih zahteva** = revizija elemenata iz studija za funkciju za koju se sprovodi reinž.

- **Izrada detaljnog stabla aktivnosti** = idejni projekat na osnovu dobijene informacije od korisnika.

- **Def. decomp. dijagrama** = uspostavljanje veza između podaktivnosti koje su povezane strelicama.

* **^ Ulaz** = detaljno stablo aktivnosti i informacije od korisnika ; **Izlaz** = detaljni decomp. dijagram.

- **Elementi dekompozicionog dijagrama** = pravougaonici i strelice (sa granjanima/udruž./povratne/skrivene).

- **Granične strelice** = strelice nastale izvan trenutnog dijagrama ; **Interne strelice** = povezuju aktivnosti (izlaz 1 je ulaz 2).

- **Povratne strelice** = kada izlaz iz jedne aktivnosti predstavlja ulaz, kontrolu ili mehanizam prethodne.

* **Rekrucija** = kada strelica ide u prethodni ulaz ili kontrolu ; **Podrška** = kada ide u mehanizam.

- **Skrivene strelice** = za kada je dijagram natrpan; () na kraju strel. = nema prikaza u detetu, obr. nema u prik. u rod.

Naziv aktivnosti	Naziv entiteta	CRUD	Naziv atributa	IRUN
ODRZAVANJE PODATAKA O PREVODIOCIIMA	OSOBA	CRUD	DATUMZ	I UN
			IME	I U
			PLATA	I U
			PREZIME	I U
			SIFRA ODELJENJA	N
			SIFRA RM	N
			SIFRA OSOBE	IR
			STIMULACIJA	I UN
			JMBG	I U
			VRSTA	I R
	CERTIFIKAT	CRUD	SIFRA JEZIKA	R
			SIFRA OSOBE	R
			STEPEN ZNANJA	R U
	ISPLATA	CRU	BROJ ISPLATE	IR
			SIFRA OSOBE	R
			DATUM ISPLATE	I U
			IZNOS	I U
			NAZIV JEZIKA	I U
ODRZAVANJE SIFARNIKA	JEZIK	CRUD	SIFRA JEZIKA	IRU
			MESTO	I U
	ODELJENJE	CRUD	NAZIV ODELJENJA	I U
			SIFRA ODELJENJA	IRU
			SIFRA RM	IRU
	RAD.MESTO	CRUD	NAZIV RM	I U

- **CRUD matrica** = sa jedne strane su entiteti, a sa druge strane aktivnosti koje koriste te entitete.

- **IRUN matrica** = definicije korišćenja svakog atributa u određenom entitetu i određenoj aktivnosti.

- **CRUD** = kreirati entitete (**C**), pretraživati entitete (**R**), ažurirati entitete (**U**), brisati entitete (**D**).

- **IRUN** = ubacivanje atributa (Insert), vađenje (Retrieve), ažuriranje (Update), dodela nula vrednosti (Null).

CRUD i IRUN matrica za dokument „Karton isplata“.

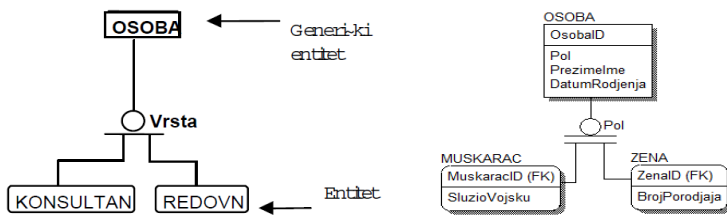
- **Definisanje dijagrama toka podataka**: zove se DFD (Data Flow Diagram) u BPwin-u.

* **DTP** = dijagram toka podataka, grafički opis koji povezuje procese, dokumenta, skladišta podataka.

- **Analiza detaljnih zahteva** = provera informacija, modela, validacija (ocena) i verifikacija (potvrda) modela.

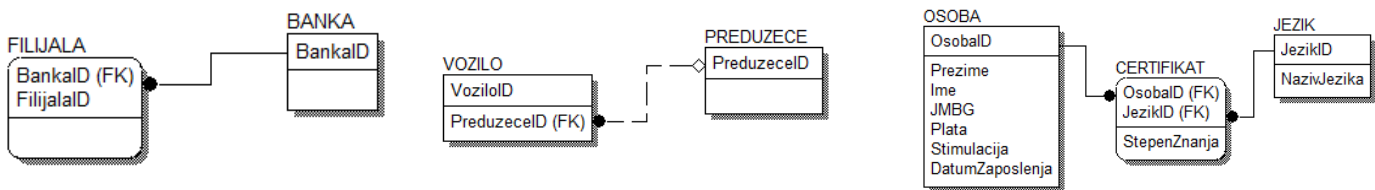
**** Kreiranje ER modela**

- **Entitet** = sve što se može jednoznačno identifikovati, pa samim tim i izolovati iz okoline i opisati.
- **Interne strelice** = kandidati za entitete (buduće tabele) jer su one nosioci informacija.
- **Identifikacija kandidata za entitete** = pr. fizički objekti (vozila, zgrade...), osobe (studenti, profesori...)...
- * Nakon toga određuju se nazivi entiteta i formiraju grupe entiteta, potom dodaju novi ako je potrebno.
- * Atribut može biti entitet a i atribut nekog entiteta (npr. entitet ulica i atribut ulica entiteta kuća).
- **Identifikacija veza** = povezivanje ent. na osnovu odgovarajućih interesa, definisanje zavisnosti ent....
- **Referencijalni integritet** = čuva bazu od pogrešnog unosa i obezbeđuje korektno povezivanje objekata.
- * Entiteti su imenice, veze su glagoli tipa sadrži, definiše, šalje, prima (npr. OSOBA prima ISPLATU).
- **Definisanje ER modela** = def. nezavisnih (roditelj) entiteta > definisanje zavisnih (child) ent. > def. veza.
- **Nezavisni entitet** = ne zavisi od drugih entiteta. Ident. i postojanje zavisnih ent. zavisi od nezavisnih.
- **Zavisni entiteti** = dele se na karakteristične, asocijativne, projektne i entitete kategorije.
- **Karakteristični** = su atributi koji se ponavljaju više puta za jedan ent. i identifikuju se preko nezavisnog.
- **Asocijativni** = nastaju od više veza između 2+ entiteta. Npr. sertifikatu treba OsobaID i JezikID iz razl. ent.
- **Projektni** = sličan asocijativnom samo što nema sopstvene attribute (npr. asoc. ima atr. stepen znanja).



- **Entitet kategorija** = zavisni entitet koji ima vezu tipa potkategorija. Npr. OSOBA ima konsultant i redovan (nepotpuna kategorija jer se može dop.).
- * **Potpuna kategorija** = nema dopunjavanja kao kod nepotpune (npr. pol može biti samo M ili Ž).

- **Identifikujuća veza** = ona koja entitet „dete“ identifikuje kroz njegovu vezu sa ent. „roditelj“.
- **Neidentifikujuća veza** = ne identifikuje dete kroz roditelja već dete ima svoju identifikaciju.
- **Veza kategorije** = veza prema podtipovima ; **Neodređujuća veza** = više prema više.
- **Identifikujuća veza (L)** = roditelj predstavlja deo identiteta deteta. Dete nasleđuje PK roditelja.
- **Neidentifikujuća veza (M)** = neobavezna ima romb ispred, obavezna znači ne sme prazno polje.
- **Neodređujuća veza (D)** = rezultat uvođenja asocijativnog ili projektnog entiteta.
- **Veza kategorije (iznad)** = hijerarhijska veza generičkog ent. koji sadrži zajed. osobine. podređ. ent.



- **Verifikacija ER dijagrama** = opšta provera entiteta (nazivi, ponavljanje, zavisnosti itd...).

TEKST	ER KONCEPT
IMENICA	ENTITET
GLAGOL	VEZA
PRIDEV	ATRIBUT ENTITETA
PRILOG	ATRIBUT VEZE
GLAGOL.IMENICA	MESOVITI TIP ENTITETA-VEZA
RECENICA	OBJEKTI POVEZANI VEZAMA
POGLAVLJE	LOKALNI PODMODEL

--- 06 - Kreiranje atributa ---

- **Atribut** = svojstvo ili k-ka uobičajena za neke ili sve instance entiteta.

- **Definisanje liste kandidata za attribute** = definiše se interes posmatranja (da li će obj. biti atr. ili ent.).

1. Svaki entitet ima proizvoljan broj atributa, nema ograničenja u njihovom broju.
2. Određeni atribut pripada jednom i samo jednom entitetu (osim prenesenih ključeva (FK)).
3. Svako pojavljivanje/instanca entiteta ima vrednosti za sve attribute tog entiteta.
4. Pravilo jednoznačnosti = atribut određene instance entiteta može imati samo jednu vrednost.
5. Pravilo doslednosti = svaki atribut predstavlja neku činjenicu, tako da svaka vrednost mora imati neko značenje.

OSOBA

Sifra osobe
Prezime (IE1.1)
Ime (IE1.2)
Sifram (FK)
JMBG (AK2.1)
rukov.Sifra osobe (FK)
Datum zaposlenja
Plata
Stimulacija
Sifra odeljenja (FK)
Pol
Vrsta

- **Definisanje ključeva** = ključ je atribut kojim se može pronaći određeni primerak entiteta npr. OsobaID.

- **Primarni ključ** (PK) = uvek jedinstven, not null, nikada se ne menja. (šifra osobe)

- **Alternative Key** (AK) = jedinstven, može null npr. JMBG jer je on jedinstven a nije PK.

- **IE n Key** = grupa ključeva koji nisu jedinstveni (unique) npr. više ljudi se zove Milan.

- **Foreign Key** (FK) = primaran u drugoj tabeli. Povezuje dete sa roditeljom.

- **Tipovi ključeva po postojanju** = prirodni (JMBG) i veštački (OsobaID).

- **Tipovi ključeva po složenosti** = prosti i složeni.

* Ako je entitet jak (nezav.) onda spoljni ključ ide u neključne attribute (ispod crte), ako je slab (zavi.) obr.

- **Postupak normalizacije** = proces zamene relacija skupom relacija koje su u pogodnijem obliku. od 1 do 5NF.

* Svrha normalizacije je **izbegavanje redundantnosti** tj. postizanje **integriteta** (celovitost) baze podataka.

- **1NF** = entitet je u prvoj norm. formi ako nema atributa sa >1 vrednosti za 1 instancu entiteta.

- **Redudansa** = suvišnost ili višak informacija, ovde to znači višestruko praćenje istih podataka.

* Umesto višeznačajnog atributa Adresa stavljamo Ulica, Broj, Mesto, Poštanski broj i tako dovodimo 1NF tj. nedeljivost podataka.

* Adresa = Vukasovićeva 50, 11000 Beograd => Ulica = Vukasovićeva, Broj = 50, Poštanski broj = 11000, Mesto = Beograd.

- **Kompozitni ključ** = kada 2 atributa čine primarni ključ (npr. ime i prezime)

- **2NF** = 1NF + svi neglavni atributi zavise od svih ključnih (primarnih) ključeva/ kompozitnog ključa.

* Imamo tabelu sa imenima (ime, prez.) i tabelu sa hobijima (ime, prez. hobi) => hobi zavisi od ime i prez.

- **3NF** = 2NF + svi neprimarni atributi isključivo zavise od ključeva, a ne od drugih neprim. atributa.

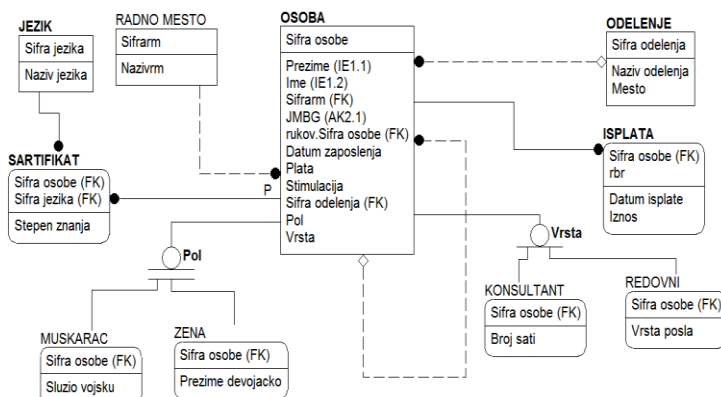
* Imamo OsobaID, DatumIsplate, IznosIsplate, SumaIsplate => Izbacujemo SumaIsplate jer zavisi od IznosIsplate.

* Nastavnik će tražiti poznavanje 3NF. Četvrta i peta nisu obavezne već samo informativno prikazane.

- **4NF** = kada u relaciji nisu date dve ili više nezavisne višeznačajne činjenice.

- **5NF** = relacija je u 5NF ako i samo ako se svaka zavisnost spajanja može pripisati kandidatu za ključ.

- **Definisanje atributa** u 3 koraka = identifikacija atr. > alociranje (premeštanje) atr. > revizija atributa.



- **Identifikacija** = definiše se na osnovu zahteva korisnika i poslovne dokumentacije.

- **Alociranje** = izvodi se u zavisnosti od toga da li atribut zavisi od ključa ili je opisni.

- **Revizija** = eliminiše ponavljanje vrednosti atributa pojedinog entiteta i proverava se „ispravnost“.

ER dijagram za dokument „karton isplata“.

--- 07 - Definisanje poslovnih pravila ---

- **Integritet entiteta** = onemogućuje pojavu dveju n-torki sa istim primarnim ključem ili da PK bude null.

- **Pravila integriteta** = definisanje poslovnih pravila vezano je za:

* **Ograničenja** kojima se definišu dozvoljena stanja baze podataka,

* **Operacije** koje mogu potencijalno ugroziti ograničenja i

* **Akcije** koje treba preduzeti ukoliko dođe do narušavanja ograničenja.

- **Domen** = skup vrednosti podataka istih tipova, preko kojih se formiraju vrednosti atrib. objekta.

- **Vrste ograničenja**:

* **Strukturno ograničenje** = integritet entiteta i referencijalni integritet (pogrešan unos i korektno povez.).

* **Ograničenje nad stand. domenom** = tip podataka (char, number...) i dužina podatka (CHAR(30) i dr.).

* **Ograničenja nad vrednošću domena** = operatori poređenja (<,>,<=,>=), IN, BETWEEN, NOT NULL.

* **Ogr. nad kardinal. veza** = kardinalnost je osobina koja pokazuje koliko nečega 1og ent. može biti u drugom.

* Kod kard., **totalno** učešće znači da svi entiteti deteta učestvuju u barem jednoj vezi sa roditeljem, dok **parcijalno** ne.

- **Operacije** = ubacivanje novog sloga/n-torke (**insert**), izmena sloga (**update**) i brisanje sloga (**delete**).

- **Akcije** = **restrict** (odbija op.), **cascade** (prosleđ. op. na vezni ent.), **default**, **set null**, **none**.

- **Definisanje kardinalnosti veza** = između entiteta mogu da postoje 3 vrste odnosa:

1. **1-prema-1** = ako jedan i samo jedan element entiteta A odgovara j i samo j el. ent. B. Država-gl. grad;

2. **1-prema-više** = ako 1 el. ent. A može da se odnosi na više el. ent. B => Žanr-film ;

3. **više-prema-više** = ako više elemenata ent. A može da se odnosi na više el. ent. B. => Predmet-Student;

- **Definisanje kardinalnosti veza** = imperativno (mora, zahteva) i opciono (može, dozvoljava).

- **Identifikujuća veza** = dete zavisi od roditelja, bez roditelja ne može da se identifikuje niti da postoji.

* Iz entiteta Banka uzimamo BankaID i ubacujemo u entitet Filijala (BankaID (FK) i FilijalaID su preko crte).

- **Neid. veza** = dete entitet može da postoji samostalno i identifikacija mu je nezavisna od roditelja.

* Iz preduzece uzimamo PredID i stavljamo ispod crte u entitet vozilo (iznad crte VoziloID, ispod PredID (FK)).

* **Identifikujuća veza se crta punom linijom, neidentifikujuća isprekidanom.**

* Kod neid. veza, **romb** označava da je null dozvoljeno (parcijalno), **crni kružić** no null (totalno učešće).

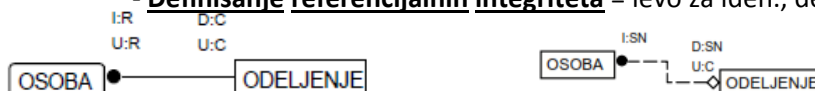
- **Rekruzivna veza** = uzmemo ProizvodID i provedemo ga i u ProizvodID (FK) i KompProd(FK) u drugom ent.

- **Neodređujuća veza** = nastanak trećeg entiteta kao rezultat 2 prenesena ključa razl. identiteta.

- **N-arna veza** = u suštini proširena neod. veza, rezultat 3 entiteta i još ima neke svoje osim tih FK-ova.

- **Veza kategorije** = nepotpuna se može dopuniti novim potkategorijama, potpuna ne može (npr. pol).

- **Definisanje referencijalnih integriteta** = levo za iden., desno za neiden. itd., na svim ostalima je isto.



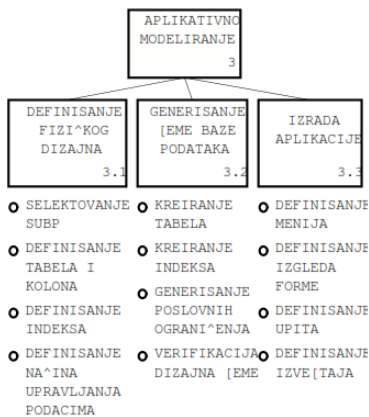
Ubacivanje (Insert) omogućuje izvođenje sledećih akcija:			Unos (Insert) omogućuje izvođenje sledećih akcija:		
Entitet ODELJENJE		Bez ograničenja	Entitet ODELJENJE		Bez ograničenja
Entitet OSOBA	Restrict	Ne može se uneti Osoba (I:R) bez 'roditelj' Odeljenje	Entitet OSOBA	Set Null	Ne može se uneti Osoba bez 'roditelj' Odeljenje, sem ako se preneseni ključ Osobe ne setuje na NULL (I:SN)
Brisanje (Delete) omogućuje izvođenje sledećih akcija:			Brisanje (Delete) omogućuje izvođenje sledećih akcija:		
Entitet ODELJENJE	Cascade	Brišu se sve Osobe za izbrisanog 'roditelj' Odeljenje (D:C)	Entitet ODELJENJE	Cascade	Brišu se sve Osobe za izbrisanog Odeljenje kao 'roditelj'
Entitet OSOBA	Restrict	Ne dozvoljava brisanje Odeljenja ako je 'roditelj' Osobi	Entitet ODELJENJE	Restrict	Ne dozvoljava se brisanje Odeljenja ako postoji Osoba za koje je Odeljenje 'roditelj'
Entitet OSOBA		Bez ograničenja	Entitet OSOBA	Set Null	Sve Osobe za koje je Odeljenje 'roditelj' setuju preneseni ključ Osobe na NULL (D:SN)
Izmena (Update) omogućuje izvođenje sledećih akcija:			Izmena (Update) omogućuje izvođenje sledećih akcija:		
Entitet ODELJENJE	Cascade	Automatska izmena Osoba za izmenu nad Odeljenjem (U:C)	Entitet ODELJENJE	Cascade	Izmeni se sve Osobe za izmenu Odeljenja (U:C)
Entitet OSOBA	Restrict	Ne može se izmeniti Osoba (U:R) ako ne postoji Odeljenje	Entitet OSOBA		Bez ograničenja

- **Identifikacija poslovnog domena**

- **Domen** = skup mogućih vrednosti iz kojih atributi uzimaju neku vrednost.

* Pr. domen broj definiše sve brojeve, tip podataka je integer, kod domena naziv tip podataka je char(30) itd.

--- 08 - Definisanje fizičkog dizajna ---



- **Sistem za upravljanje bazama podataka (SUBP)** = softver za čuvanje i pretraživanje podataka; skup programa čija je namena da na zahtev aplikativnih programa vrši manipulaciju podacima.

- **Baza podataka (BP)** = skup uzajamno povezanih podataka koji služe raznim aplikacijama. Podaci su memorisani nezavisno od programa kojima služe.

- Definisanjem fizičkog dizajna logički nivo modela se prevodi u **fizički nivo**.

- Selektovanje SUBP

- **Tipovi podataka** = numerički, znakovni, struktuirani (entiteti, atributi, ključevi).

- Nakon prelaska sa fizičkog modela u **fizičku bazu podataka**:

* Entiteti postaju tabele, atributi kolone, instance redovi, u preseku redova i kolona nastaju polja.

- **Relaciona baza** = više međusobno povezanih tabela. Prikazuje odnos među podacima u obliku tabela.

- **Funkcije SUBP** = memorisanje i održavanje podataka i kontrolisan pristup do njih i njihovo prikazivanje.

- **Postupak menjanja polja**:

* red se traži u mem. preko adr. ili ključa > prebacuje se iz ext. u OP mem. > menja se > zapisuje se ponovo u ext. mem.

* **Brisanje** redova zahteva pažnju - da se brišu pravi redovi, tj. poštovanje referencijalnog integriteta.

* **Dodavanje** novih redova zahteva obezbeđenje potrebnog memorijskog prostora i poš. ref. int.

* **Traženje** je uspelo ako je vrednost PK reda identična vrednosti ključa datog u argumentu pretrage.

* **Pretraživanje** se vrši argumentima datim kao LOGIČKI izraz. Sastavlja se lista zahtevanih svojstava i kriterijuma po kojima se vrši pretraživanje tabela i izdvajaju svi oni redovi koji u potpunosti udovoljavaju zahtevima.



Razlika između SUBP i datoteka => eliminisanje redundantne sličnih datoteka u razl. aplikacijama.

- **SQL upitni jezik** (neproceduralan) = CREATE TABLE/VIEW/SYNONIM/INDEX ; ALTER TABLE ; DROP T/V/S/I.

* Pretraživanje podataka = **SELECT** (FROM, WHERE, TOP, GROUP BY...) (=,>,<, IN, LIKE, BETWEEN itd.).

* Manipulacija podacima (**DML**) = INSERT, UPDATE, DELETE (ubacivanje redova u tabelu / izmena / brisanje).

* Upravljanje podacima (**DCL**) = GRANT CONNECT/REVOKE, COMMIT/ROLLBACK, LOCK TABLE, AUDIT.

- **Definisanje tabela i kolona** = levo je logički nivo, desno fizički.

Osoba	Osoba
OsobaID	OsobaID: Long Integer
Prezime	Prezime: Text(50)
Ime	Ime: Text(50)
JMBG	JMBG: Text(13)
Plata	Plata: Currency
Stimulacija	Stimulacija: Currency
DatumZaposlenja	DatumZaposlenja: Date/Time
OdeljenjeID (FK)	OdeljenjeID: Integer
RadnoMestoID (FK)	RadnoMestoID: Integer
RukovodilacID (FK)	RukovodilacID: Long Integer

- Ovde se definišu karakteristike atributa (njihov tip podatka, dozvoljena dužina, da li smeju biti NULL, da li smeju biti veći ili manji od x ...)

- **Trigeri** = okidači, skup SQL izraza koji se izvršavaju kada se desi određen događaj.

- **Procedure** = skup SQL izraza koji se izvršavaju po pozivu.

- **Funkcija** = izdvojena programska celina (potprogram) koja transformiše unesene podatke u novi podatak. (proc. ne vraća ništa, funk. vraća)

- Definisanje indeksa

- Postupak pretraživanja zahtevanog podatka u tabeli može da potraje dugo, pa se za rešavanje problema pretraživanja koriste specijalni tipovi tabela pod imenom **indeksi**, u kojima se nalaze adrese redova.

- Definisanje načina upravljanja podacima

- Obuhvata skladištenje, ponovno pristupanje i kontrolu podataka. Ostvaruje se u okviru k/s arhitekture.

1. Integritet baze podataka = omogućava tačnost podataka i štiti baze od pogrešnog ažuriranja.

- **Rečnik podataka** = baza podataka o bazi (meta-baza), u njoj su opisi tabela, relacija među njima, objekti...

- **Integritet domena** = dozvoljen skup vrednosti (npr. maloletnik ne može biti stariji od 18 godina).

- **Integritet tabele** = ispunjen ako je svaki red u tabeli jedinstven (tj. svaki red ima svoj ispravan PK).

- **Referencijalni integritet** = ili integritet relacija omogućuje vezu između raznih kolona i tabela (PK i FK).

- **Autoreferencijalni integritet** = spoljni i PK su u istoj tabeli, ta kolona ima iste vrednosti kao i PK kolona.

2. Transakciona obrada podataka = definišu se dva tipa transakcija:

* **DML transakcija** koja povezuje veći broj DML naredbi, naziva se i eksplicitna (npr. SELECT, UPDATE...).

* **DDL transakcija** kojom se izvodi samo jedna naredba, implicitna (npr. BEGIN TRANSACTION, COMMIT...).

- **Transakcija** = operacija kojom se izvodi serija izmena nad jednom ili više tabela.

3. Sigurnost podataka = mehanizam zaštite podataka od neovlašćenog korišćenja (samo vlasnik daje dozvole).

* Za dodeljivanje dozvola = **GRANT** <privilegija> **ON** <tabela ili pogled> **TO** <korisnik ili grupa korisnika>.

4. Zaključavanje podataka = mehanizam za upravljanje istovremenim pristupom podacima, vrši se kada korisnik pokuša da izvrši izmenu nad podacima u bazi => Zaključavanje zapisa, tabele, baze podataka.

* **Shared** zaključavanje ne da izmenu, mogući upiti nad podacima, **exclusive** ne da nikom drugom da zaključa trenutno.

5. Oporavak (recovery) baze podataka = povratak u stanje pre hard. ili soft. otkaza sistema.

* **Kompletno arhi.** znači da se prenose sve transakcije, **inkrementno** znači da se na kopiju prenose izmenjeni slogovi.

--- 09 - Generisanje šeme baze podataka ---

- **BPwin** = alat za funkcionalno modeliranje ; - **ERwin** = alat za informaciono modeliranje, on generiše ŠBP.

- **Direktni inženjering** = proces generisanja (fizičke) baze podataka iz logičkog modela podataka.

1. U ACCESS-u se pravi baza : File > New > Blank Database

2. U ERwin-u se vrši konekcija : Server > Access Connection ; kod Database kliknemo browse do baze.

3. U ERwin-u se generiše šema : Tasks > Forward Engineer > Schema Generation > Generate.

4. Nakon rada izvršiti diskonekciju : Server > Access Connection > Disconnect.

* Nakon diskonekcije, otvaramo .mbd fajl u Access-u koji je sada napunjen tabelama i vezama.

* Kada se generiše šema baze podataka, entiteti prelaze u tabele, atributi u kolone, a veze u relacije i definišu se referencijalni integritet, trigeri, procedure, indeksi i druge osobine koje podržava izabrani SUBP.

- **Kreiranje tabela** = CREATE TABLE => <naziv table> <kolone> <tip> <velicina> <index>.

- **Kreiranje indexa** = -II- INDEX => omogućava direktan pristup redovima i tako smanjuje vreme pristupa.

- **Traženje podataka** = uspelo ako je vrednost ključa n-torke ista kao vrednost ključa datog u upitu pretrage.

* **FIND 2 LOOK IN** OsobaID => Naći će n-torku kojoj je OsobaID = 2.

- **Pretraživanje -II-** = upit dat logičkim iskazima, sastavlja se lista uslova i izdvajaju n-torke koje ispunjavaju sve uslove.

* **SELECT** Iznos,OsobaID **FROM** Isplate **WHERE** Mesec = 'mart'

* **CREATE INDEX** <index_ime, tabela, kolona>

- **Generisanje poslovnih ograničenja**

- **Svrha ograničenja** = definisanje ranga validnih vrednosti. (domeni kolona, pravila RI, integriteta...).

* CREATE TABLE Kupci (IDKupca INTEGER PRIMARY KEY, PrezimeKupca TEXT (50)) - Ogr. na 50 znakova.

1. **Ograničenja nad primarnim ključem** = kolona ne sme da primi NULL vrednost ni u jednom redu.

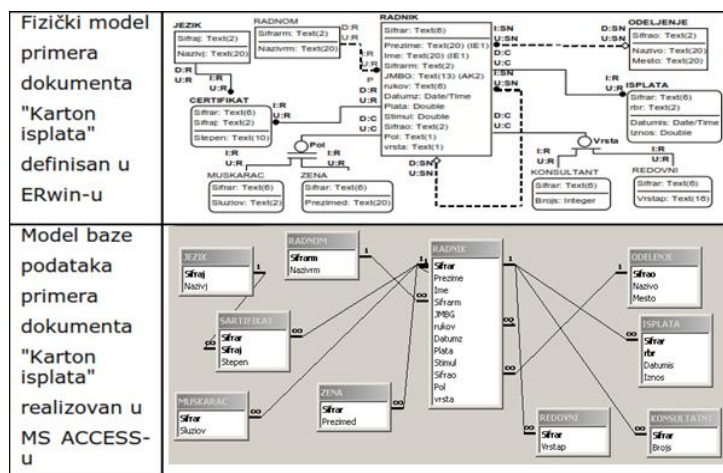
2. **Ograničenja UNIQUE** = definiše jedinstven ključ tabele tj. svaki red mora ima različitu vrednost.

3. **Ograničenje NOT NULL** = zabrana NULL vrednosti (praznog polja) u polju.

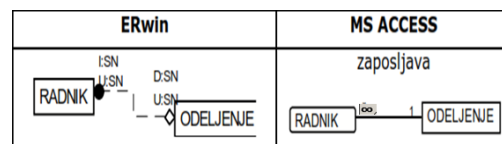
4. **Ograničenje nad prenesenim FK ključem** = ako je FK ključan (iznad crte) ne sme biti NULL, suprotno može.

5. **CHECK ograničenje** = uslovi se mogu odnositi na više kolona, ono je narušeno samo ako je izraz FALSE.

* U suštini IF naredba => "Plata" FLOAT NOT NULL; **CHECK (>0)**.



- Generisanjem fizičkog modela iz ERwin-a u konkretni model baze podataka - MS ACCESS, automatski se uspostavljaju relacije između polja u selektovanim tabelama prema indeksima definisanim nad tim tabelama.



U Odeljenju je zaposleno više Radnika, a Radnik radi u nula ili jednom Odeljenju.

- **Verifikacija dizajna** = vrši se unošenjem test podataka u kreirane tabele.

- **Inverzni inženjering** = proces dobijanja fizičkog i logičkog dizajna iz postojeće fizičke baze podataka.

--- 10 - Izrada aplikacije ---

- **Izrada aplikacije** = zavisi od specifikacija SUBP-a i klijent server arhitekture.

- **Dvoslojna klijent-server arhitektura** : između slojeva se nalaze komm. sklopovi, primena <15 PC-eva.

* **1. Sloj** = server-strana u kojem se nalazi baze podataka i SUBP/DBMS.

* **2. Sloj** = klijent-strana gde su definisane klijentove (korisničke) aplikacije.

- **Prednosti 2-sloj** = centralizovano upravljanje resursima, jednostavnije podešavanje sigurnosti podataka.

* **Mana** = nedostatak **skalabilnosti** - osobina sistema za efikasan rad pri velikom br. kor. i ne smeta dodavanje novih.

- **Troslojna k-s arhitektura**: između slojeva kom. sklopovi (ruteri, kablovi), primena kod sistema >15 PC.

* **1. sloj** = server-strana gde su baze podataka i SUBP.

* **2. sloj** = aplikativni server koji sadrži zajedničke aplikacije koje „napada“ klijent.

* **3. sloj** = klijent-strana gde se definišu delovi aplikacija koje su specifične za konkretnog korisnika.

- **Prednosti 3-sloj**= skalabilnost, pouzdanost, fleksibilnost, apl. se može menjati i u toku rada sistema...

- **Klijentska strana** = radi sa redovima iz tabele, koristi GUI, vrši logiku apl., proverava ispravnost unesenih podataka...

- **Aplikativna strana**= 2. sloj, posrednik između kl. i serv. strane, to su alati za razvoj apl., (jezici III i IV gen).

- **Serverska strana** = smešten SUBP, nalazi se BP sa tabelama i podacima, isporučuje podatke klij. str.

- **Definisanje menija** = definiše se meni (treba biti koncizan, na vrhu bla bla...).

- **Definisanje izgleda formi** = forme su objekti SUBP-a, omogućavaju korisniku prikaz podataka iz baze i unos u nju.

* Forme u sebi sadrže druge objekte. U suštini setiti se formi iz poslovnog softvera, unose se podaci a mogu i da se čitaju.

- **Definisanje upita** = upiti se dele na upite nad jednom tabelom i na upite nad više tabela.

- **SQL** = strukturni jezik za postavljanje upita i manipulaciju podacima, njim se obavlja komm. između kl. i serv.

* Klijent pošalje upit serveru (SELECT X FROM <TABELA>), a server pošalje klijentu odgovor tj. željene podatke iz baze.

* SELECT * FROM RADNIK => Izlista sve kolone (SIFRA, IME, PREZIME...) iz tabele RADNIK.

* SELECT TOP 4 prezime, plata FROM radnici ORDER BY prezime => Prva 4 radnika sa njihovim platama, po abecedi.

* SELECT TOP <broj> PERCENT <x> FROM <tabela> [ORDER BY <y>] => Prvih n% radnika (ako ima 10, 20% je 2 radnika).

* SELECT DISTINCT <x> FROM <tabela> => Eliminacija duplih slogova (ako postoji 3 žanra koji su u 14 polja, da ne piše 14 puta već samo 3)

* SELECT ime,prezime FROM radnici WHERE RadnikID='02' => Prikazuje nam ime i prezime radnika čiji je RadnikID dva.

* WHERE <kolona> <operator poređenja (<, >, !=, <>, NOT LIKE, LIKE, BETWEEN...)> <kolona, vrednost, konstanta>.

* SELECT * FROM radnici WHERE sifrap='30'; SELECT ime, plata FROM radnici WHERE plata BETWEEN 800 AND 900

* SELECT * FROM radnici WHERE sifrap IN ("10", "30") => lista one kojima je sifrap 10 ili 30.

* SELECT * FROM radnici WHERE sifrap="10" OR sifrap="30" => lista one kojima je sifrap 10 ili 30.

* SELECT ime+prezime FROM radnici => pravi novu kolonu koja u stvarnosti ne postoji ali sadrži i ime i prezime.

* SELECT prezime FROM radnici WHERE prezime LIKE '?U*' => Kao AIOS, npr. VUKIC, SUSIC...

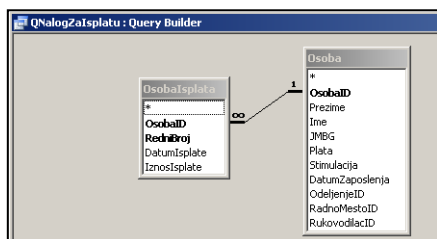
- **Definisanje izveštaja** = za svaki izveštaj neophodno je da postoji izvor podataka.

SELECT DISTINCTROW Osobalsplata.OsobaID, Osoba.Prezime, Osoba.Ime, Osobalsplata.Datumsplate,

Osobalsplata.Iznosplate, [Prezime] & " " & [Ime] AS Osoba

FROM Osoba INNER JOIN Osobalsplata ON Osoba.OsobaID = Osobalsplata.OsobaID.

Tabele koje se koriste za kreiranje izveštaja

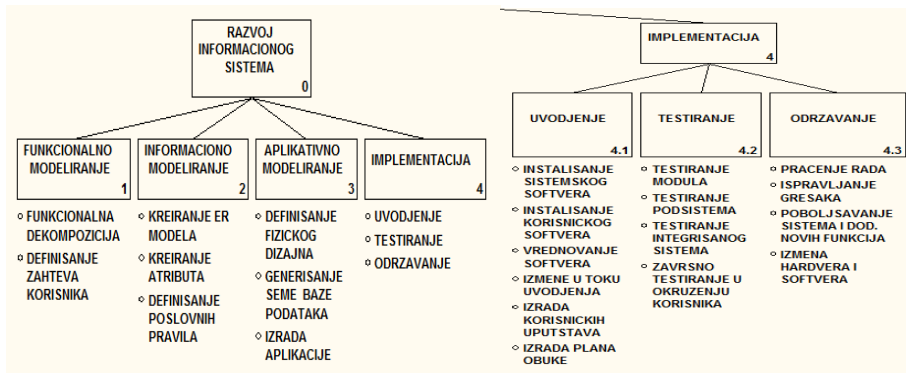


Design mode izveštaja

Print preview mode izveštaja.

OsobaID	PREZIME I IME	Datum isplate	IZNOS
1	EREMIJA ZORAN	4.8.2011	300,00 Din.
1	EREMIJA ZORAN	4.8.2011	200,00 Din.
2	ALAGIC MILAN	4.8.2011	15.000,00 Din.
2	ALAGIC MILAN	4.8.2010	500,00 Din.
5	MARTIC ZORA	5.8.2011	500.000,00 Din.
UKUPNO			516000

--- 11 - Implementacija - uvođenje ---



- **Sistem inženjering** = oblast koja se fokusira kako da dizajnirate/upravljate projektima životnog ciklusa softvera.
- **Sistemske softver** = operativni sistem, drajveri, BIOS... Deli se na prevodiocce, linkere i usluž. programe.
- **Operativni sistem** = organizuje resurse računara kako bi omogućio njihovo najefikasnije korišćenje.
- **Program** = niz instrukcija koji ostvaruje neki algoritam.
- **Proces** = program koji se trenutno izvršava (učitan je u radnu memoriju).
- * Nakon učitavanja računara, boot program učitava OS u glavnu memoriju PC-a i tada OS dalje upravlja...
- **Instalacija korisničkog/apl. softvera** = programi koje korisnici računara koriste za razne poslove.
- **Vrednovanje softvera** = podskup aktivnosti softverskog inženjerstva, sistem vrednovanja (evaluacija).
- * *Specificiranje zahteva > Zahtevi za kvalitetom (pr. efikasnost) > Formulisanje faktora kvaliteta > Metrika i kvalitet > Ocenjivanje.*
- **Efektivnost** = mera realizacije planiranih aktivnosti i planiranih rezultata.
- **Efikasnost** = odnos ostvarenih rezultata i upotrebljenih resursa.
- **Funkcionalnost** = služenje određenoj svrsi ; - **Upotrebljivost** = mera služenja određenoj svrsi.
- **Izmene u toku izvođenja** = na osnovu definisanih primedbi izvode se izmene.
- **Izrada korisničkih uputstava** = opšta uputstva za rad sa aplikacijom.
- **Izrada plana obuke** = pravi se plan obuke za buduće korisnike.

--- 12 - Testiranje ---

- Završni **test sistem** vrše korisnici koji koriste stvarne podatke tokom dužeg vremenskog perioda. 3 nivoa:
- 1. Verifikacija** – podrazumeva postojanje pisanog zahteva koje rezultati razvoja moraju da ispune. Rezultati ispitivanja pokazuju da li je rezultat razvoja zadovoljio tražene karakteristike.
- 2. Validacija** – ispitivanje funkcionalnosti rezultata razvoja u stvarnim uslovima upotrebe.
- 3. Revizija** – preispitivanje ili obnavljanje procesa testiranja.
- * Verifikacija - testiranje se vodi u simuliranom okruženju korišćenjem simuliranih podataka.
- * Validacija - testiranje se vodi u živom okruženju koristeći stvarne podatke.
- * Revizija - testiranje potvrđuje da je sistem bez greške i da je spreman da se stavi u funkciju.
- * Verifikacija se radi u **određenoj fazi** razvoja, validacija **na kraju** procesa razvoja.
- **Cilj testiranja** = otkrivanje nedostataka i grešaka.
- **Greška** = nastala od strane ljudi ; - **Nedostatak** = rezultat greške. ; - **Otkaz** = nastaje pri radu softvera sa nedostacima.
- **Incident** = ponašanje u vezi nedostataka, upozorava korisnika na mogućnost otkaza.
- **Provera modula** = da li sve forme rade, da li su svi upiti ispravni, makroi...
- **Testiranje podsistema** = neke od aktivnosti su spajanje modula i testiranje rada grupe modula, test. interfejsa...
- **Testiranje integrisanog sistema** = koristi realne podatke; funkcionalnost celine, performanse, restart i oporavak sistema.
- **Završno testiranje u okruženju korisnika** = ako samo jedan korisnik testira alfa-testiranje, ako mnogo više onda je beta.

--- 13 - Održavanje ---

- **Održavanje softvera** = proces modifikacije softvera nakon isporuke radi ispravljanja grešaka, poboljšanja performansi.
- **Reaktivno održavanje** = unazad, kao posledica nečeg :
- * **Adaptivno održavanje** = modifikacija koja se izvodi sa ciljem da se softveru sačuva upotrebna vrednost.
- * **Korektivno održavanje** = reaktivna modifikacija softvera koja se vrši radi ispravke otkrivenih grešaka.
- **Proaktivno održavanje** = unapred, pre eventualnog incidenta ili greške :
- * **Perfektivno održavanje** = modifikacija softvera radi unapređenja performansi ili održivosti.
- * **Preventivno održavanje** = modifikacija sa ciljem da se detektuju i isprave skrivene greške pre nego što deluju.
- * Ako je razlog unapređenje onda je to **Perfektivno** održavanje, a ako je razlog samo manja ispravka to je **Preventivno** održavanje.
- **Praćenje rada** = aktivnost se izvodi kontinualno, sve dok ne bude potrebno da se izvede zamena na osnovu dobijenih info.
- **Ispravljanje grešaka** = ispravljaju se greške i vrši prilagođavanje korisniku.
- **Poboljšavanje sistema i dodavanje novih funkcija** = ovo spada u unapređenje i perfektivno održavanje.
- **Izmena hardvera i softvera** = nastaju usled novih zahteva, predmet izmene mogu biti topologija mreže...
- * **Zvezdasta mreža** = istovremena komunikacija zavisi od centralnog čvora.
- * **Linijaska mreža** = komunikacija se izvodi jedan po jedan između čvorova.
- * **Prstenasta mreža** = obavlja se istovremena komunikacija bez zavisnosti od centra čvora.
- Potrebni elementi za klijent-server sistem:
- * **Mrežni hardver** = mrežni adapter (i bežični), modemi i telefonske linije.
- * **Mrežni softver** = omogućuje kontrolu pristupa mreži, upravljanje resursima na njoj.
- * **Komunikacioni softver** = omogućava da razni računari u mreži primaju i šalju pakete preko mreže.
- **Protokol** = jezik sporazumevanja računara. TCP/IP i SPX/IPX.
- **Softverski sloj (Middleware)** = posrednik za rad sa bazom (između baze i klijenta), aktivan i na klijentu i na serveru.
- **Aktivnost klijent** = izvršava apl. koje se povezuju sa serverom baze i pri tom se definiše user i pass k-ka.
- **Aktivnost server** = obrađuje zahteve aktivnosti klijenta i vodi tabelu dnevnika transakcija (log file).
- **Starenje softvera** = dugotrajno održavanje, nefleksibilnost... moguće rešenje : **reinženjering**.