Free @nonima
Dokumentacija Oracle

http://stroga.anonima.free.fr

BAZE PODATAKA – OSNOVE PROGRAMIRANJA

Dizajniranje baze podataka

1. Izrada logickog modela baze podataka

Teorijska osnova za izradu baze podataka pociva na logickom modelu baze podataka. Kao sto je pravilo pri modeliranju, i ovaj model predstavlja idealizovanu sliku sistema koji postoji u stvarnosti. Korisnost jednog modela ovisi o tacnosti kojom on predstavlja realni sistem. Model u nasem slucaju najcesce predstavlja preduzece te stoga ima izrazeniju dinamicku nego staticku narav. Neophodno je da model slijedi promjene do kojih dolazi u okviru djelovanja preduzeca i da evoluira zajedno sa preduzecem.

Definicija *Logicki model baze podataka predstavlja elemente podataka koristene u preduzecu i njihove medjusobne relacije.*

Najcesce upotrebljavana metoda pri izradi logickog modela baze podataka je tzv. metoda **J**edinke-**R**elacije [(1)] . Jedinke u logickom modelu predstavljaju osobe, mijesta, objekte, idejne tvorevine. Svaka jedinka je opisana kroz skup atributa koji su joj pridruzeni. U primjeru baze podataka u kojoj pohranjujemo podatke o preduzecu, s tacke gledista sekretarice svaki zaposleni radnik predstavljen je skupom atributa kao sto su njegovo ime i prezime, odjel u kojemu je rasporedjen. Za rukovodioca odjela isti zaposleni radnik je predstavljen skupom koji se ne poklapa obavezno sa sekretaricinim. U tu grupu atributa spadaju na primjer radno mijesto koje zaposleni obnasa, strucna sprema koju zaposleni posjeduje itd. Kao sto vidite iz primjera, rukovodna pravila upravljaju oblikovanjem logickog modela baze podataka.
Nadalje, jedinke ne postoje kao izolovani subjekti, nego su povezane relacijama. Relacije koristimo u cilju primjene rukovodnih pravila. Za primjer mozemo navesti pravilo da je jedan i samo jedan zaposleni ujedno i rukovodilac odjela u kojemu je rasporedjen.
Mozda se pitate, nestrpljivi da pocnete izradu vase baze podataka, koja je svrha izrade ovoga modela ? Prilikom izrade logickog modela baze podataka zadatak vam je da uzmete u obzir sve podatke koristene u okviru djelovanja preduzeca i njihove medjusobne relacije. Pri tome se ne obazirete na detalje primjene, kao na primjer tip podataka koji treba da sadrzi pojedina kolona. Nivo uopstavanja pri izradi logickog modela baze podataka je visi od nivoa upostavanja fizickog modela baze podataka.

1.1. Teoretske osnove relacijske baze podataka

Nemoguce je govoriti o izradi baza podataka a da se barem nakratko ne osvrnemo na teorijske osnove na kojima pociva relacijski sistem baza podataka. Ukratko, te osnove su slijedece :

- Svaka jedinka posjeduje skup atributa koji je opisuju. U bazi koji uzimamo za primjer, jedinka nazvana Klijent opisuje klijente preduzeca.
- Jedna linija u tabli odgovara skupu atributa jedne konkretne jedinke. Konkretna jedinka u primjeru Klijenta bio bi klijent preduzeca kojeg odredjuju slijedeci atributi : 20001256,Dzevad, Sabljakovic, Avenija Branka Copica 25, 71000, Sarajevo, 325879.
- Pojedini atributi jednoznacno oznacavaju svaku konkretnu jedinku. Te atribute ili skupove atributa nazivamo *primarnim kljucem* [(2)]. Za jedinku koja opisuje klijente preduzeca Klijent_ID je primarni kljuc jer nam omogucuje da identifikujemo svakog klijenta (ne postoje dva klijenta koji posjeduju identican Klijent_ID).
- Atribut koji sacinjava primarni kljuc mora posjedovati vrijednost (ne moze biti *NULL*).
- *Strani kljuc* [(3)] jedinke je atribut cija vrijednost mora postojati kao vrijednost u primarnom kljucu druge jedinke.
- Jedinke su medjusobno povezane
- Poredak linija jedinke je nevazan
- Poredak atributa unutar jedinke je nevazan

(1) Jedinka-Relacija, na engleskom Entity-Relationship
(2) Primarni kljuc, na engleskom Primary key
(3) Strani kljuc, na engleskom Foreign key

1.2. Tabla : skup linija i kolona

S tacke gledista informaticara jedinka je otjelotvorena kao *tabla* u bazi podataka. Atributi jedinke su otjelotvoreni kao *kolone* jedne table. Jedinstven skup atributa odredjene jedinke, odnosno skup vrijednosti kolona jedne table naziva se *linija*. Pojmovi linija i *snimak* (*slog*) [(4)] ravnopravno su koristeni.

Definicija *Kolona u tabli baze podataka predstavlja atribut jedinke.*
Linija je jedinstven skup atributa jedinke, odnosno skup vrijednosti kolona u tabli baze podataka.
Snimak (slog) je predstavljen istim elementima kao i linija.

Napomena *Upravljacki Sistem Relacijske Baze Podataka (RDBMS) Oracle raspolaze internom strukturom fajlova sa podacima tako da nije moguce odrediti koje su table pohranjene u bazi podataka na osnovu strukture fajlova. Pojedini sistemi kao dBASE pohranjuju table kao zasebne fajlove sto nije slucaj kod RDBMS Oracle.*

1.2.1 Nebitnost poretka linija

Kljucni princip teorije relacijske baze podataka jeste da tabla ne posjeduje implicitan poredak. Da bismo dobili linije iz baze podataka u zeljenom poretku neophodno je da taj poredak naznacimo prilikom konstruisanja zahtijeva (upita). [(5)]

1.2.2 Nebitnost poretka kolona

Kao u slucaju linija tako ni kolone jedne table nemaju neki predodredjeni redoslijed. Ukoliko koristite SQL*Plus kako biste dobili opis jedne table (komanda describe) kolone su prikazane u onom poretku u kojem su kreirane. Medjutim, to vas ne sprijecava da definisete drugaciji poredak kolona prilikom kostrukcije vaseg zahtijeva . Mozete takodjer promijeniti definiciju jedne kolone tako da to ne utice na definiciju ostalih kolona iste table. Upravljacki sistem je nadlezan da tom prilikom obavi sve promjene u fizickoj strukturi baze podataka. Stoga kazemo da relacijska baza podataka pruza *logicku neovisnost podataka*.

1.3. Integritet podataka

Prema teoriji relacijskog sistema, svaka jedinka posjeduje skup atributa koji jednoznacno identifikuju svaku od njenih linija. Ona takodjer precizira kako nijedna linija ne treba da je duplicirana unutar table odakle proizilazi da svaka tabla treba da posjeduje primarni kljuc. Ovaj princip je nazvan *integritet podataka*. Primjera radi, maticni broj gradjanina je jedinstven za svakog zaposlenog.

1.4. Primarni kljuc

Svaka jedinka posjeduje skup atributa koji omogucuje de jednoznacno identifikujemo konkretnu jedinku. [(6)]
Taj skup atributa nazivamo primarni kljuc. Primarni kljuc moze da sacinjava jedan atribut - Klijent_ID u slucaju jedinke Klijent, ili vise atributa - u slucaju jedinke Stavke_porudjbine svaka stavka je identificirana atributima Klijent_ID i Artikl_ID. Izbor pojedinih primarnih kljuceva je ocit, sto medjutim nije uvijek slucaj. Da biste ocijenili u kojoj mjeri razumijete ovu problematiku istrazujete postojecu strukturu baze podataka. Pristupite ovom sa odredjenom dozom rezerve i ne vjerujte slijepo u nepogresivost postojece strukture.

Definicija *Primarni kljuc je skup atributa koji jednoznacno identifikuje liniju.*

(4) Snimak ili slog, na engleskom Record
(5) Zahtijev ili upit, na engleskom Request,Query
(6) Konkretna jedinka, na engleskom occurrence

1.4.1. Neophodnost i obaveznost vrijednosti atributa unutar primarnog kljuca

Kao sto je vec receno, kljucni princip teorije relacijskih sistema zahtijeva da niti jedan atribut koji ulazi u sastav primarnog kljuca ne moze imati nultu vrijednost (*NULL*). Nakon kraceg razmisljanja doci cete intuitivno do istog zakljucka. Kao sto je vec objasnjeno, primarni kljuc mora da jednoznacno identifikuje svaku liniju jedinke. Prema tome, ukoliko primarni kljuc ili njegov dio ne posjeduje nikakvu vrijednost on nista ne moze da identifikuje. Jedan klijent, ciji primarni kljuc ne posjeduje vrijednost, ne moze ni na koji naci biti identifikovan ili tretiran.

1.5. Referentni integritet [(7)]

Table su medjusobno povezane posredstvom stranih kljuceva. Jedan strani kljuc se sastoji od jedne ili vise kolona cije su vrijednosti u potpunosti sadrzane u primarnom kljucu neke druge table.

Artikl

Art_ID	Naziv	Tip_ID
00120	Blue steel	**0012**
01256	TopTec 2000	**0014**
25125	Blastout	**0008**
02358	Sleepers 2	**0012**
14789	Seven	**0012**
13589	Windows 98	**0001**

AR_TA_FK

Tip artikla

Tip_ID	Naziv
0001	CD software
0008	CD Sony PS
0012	Video kaseta
0014	CD audio

Slika 1.1. *Graficki prikaz stranog kljuca*

Gornji primjer prikazuje table Artikl i Tip artikla, povezane stranim kljucem AR_TA_FK. Ta veza podrazumijeva da sve vrijednost u koloni Tip_ID table Artikl moraju postojati u koloni Tip_ID table Tip artikla. Pokusaj unosa linije koja bi imala Tip_ID = 0025 u tablu Artikl ne bi bio prihvacen s obzirom da nijedna linija table Tip artikla ne posjeduje tu vrijednost.

Referentni integritet je postignut onda kada svaka vrijednost stranog kljuca

- postoji medu vrijednostima primarnog kljuca na koji nas upucuje doticni strani kljuc
- ili je jednaka nultoj vrijednosti (*NULL*)

Od momenta kada smo definisali primarne i strane kljuceve nad bazom podataka garantovanje integriteta podataka i referentnog integriteta je u nadleznosti Upravljackog Sistema Relacijske Baze Podataka (RDBMS) [(8)]

Definicija *Strani kljuc sacinjavaju jedna ili vise kolona cije vrijednosti moraju postojati unutar vrijednosti primarnog kljuca neke druge (ciljne) table.*

1.6. Relacije i jedinke

Relacija definise odnos izmedju dvije jedinke.Unutar relacije jedna jedinka je definisana kao roditelj dok je druga dijete. Relacija posjeduje slijedece karakteristike :

- **Identifikaciona ili neidentifikaciona relacija.** U slucaju identifikacione relacije, primarni kljuc roditelja cini sastavni dio primarnog kljuca djeteta, sto nije slucaj kod neidentifikacionih relacija.
- **Kardinalitet**. Kardinalitet jedne relacije je definisan brojem linija jedinke djeteta koje su u vezi sa jednom linijom jedinke roditelja. Moguce je govoriti o donjem i gornjem kardinalitetu. U slucaju relacije izmedju jedinke zaposleni i jedinke Odjel, gdje svaki odjel okuplja jednog ili vise zaposlenih, mozemo govoriti o kardinalitetu (1,n). Primjeri kardinaliteta sa kojima se susrecemo su (0), (1) i (n) .

(7) Referentni integritet, na engleskom Referential integrity
(8) Upravljacki Sistem Relacijske Baze Podataka, na engleskom Relational Database Management System

Definicija *Kardinalitet definise broj linija jedinke djeteta koje su u vezi sa jednom te istom linijom jedinke roditelja. Kardinalitet jedne relacije moze biti obavezan - primjer u kome jedna linija jedinke roditelja mora obavezno biti vezana sa jednom i samo jednom linijom jedinke djeteta. Kardinalitet moze biti i manje restriktivan - primjer u kome jedna linija jedinke roditelja moze da nema vezu ni sa jednom linijom jedinke djeteta ili naprotiv da posjeduje veze sa vise linija jedinke djeteta.*

- **Obavezna ili neobavezna relacija.** Relacija je obavezna u slucaju kada svaka linija jedinke roditelja mora da je u vezi sa jednom ili vise linija jedinke djeteta. Relacija je neobavezna u slucaju kada linija jedinke roditelja moze da postoji iako joj nijedna linija jedinke djeteta nije pridruzena.
- **Pravila integriteta.** Ova pravila definisu aktivnosti koje je neophodno preduzeti prilikom izmjene linija u jedinki roditelju ili jedinki djetetu. Postoji sest mogucnosti izmjene : dodavanje, izmjena ili brisanje linije u jedinki roditelju i dodavanje, izmjena ili brisanje linije u jedinki djetetu.
  Za svaku od navedenih izmjena moguce je poduzeti jednu od slijedecih aktivnosti :
  ⇒ ogranicavanje odnosno sprijecavanje izmjene
  ⇒ pridruzivanje nulte vrijednosti u kljucu jedinke roditelja ili jedinke djeteta
  ⇒ pridruzivanje za taj slucaj unaprijed definisane vrijednosti u kljucu jedinke roditelja ili jedinke djeteta
  ⇒ primjeniti istovjetnu izmjenu nad odgovarajucim linijama jedinke djeteta

1.7. Pojam nulte vrijednosti (NULL)

Jedna od osnovnih razlika izmedju relacijske baze podataka i starijih baza podataka jeste pojam nulte vrijednosti koji je prisutan u relacijskim bazama podataka. Ova specijalna vrijednost u relacijskim bazama podataka ukazuje na odsustvo bilo kakve vrijednost u polju znamenkastog (9) ili brojcanog (10) tipa.

Definicija *Nulta vrijednost pokazuje da vrijednost kolone ili formule koja ili nije primjenjiva u datom slucaju ili da jos nije dodijeljena.*

U relacijskoj bazi podataka nulta vrijednost u koloni moze izrazavati razlicite koncepte :

- Koloni nije moguce dodijeliti vrijednost za datu liniju.
- Koloni nije jos uvijek dodijeljena vrijednost.

Relacijska baza podataka omogucava dodjelu nulte vrijednost u koloni kao i test kolone koji ukazuje da kolona sadrzi nultu vrijednost.

1.8. Pravila normalizacije

Kreiranje table u Oracle bazi podataka relativno je jednostavan poduhvat za koji nam stoji na raspolaganju vise alata - SQL*Plus, SQL Worksheet, ... O ovim alatima ce biti govora u narednom poglavlju. Medjutim, mnogo vaznije je posvetiti se iznalazenju optimalne sheme koju cete koristiti pri izradi vase baze podataka.
Ovdje cemo se osvrnuti na izetno vazan pojam u teoriji relacijskih baza podataka - normalizaciju. Normalizacija se svodi na izucavanje relacija izmedju tabli, atributa (kolona) i medjusobne ovisnost atributa u cilju :

- minimiziranja izlisnosti podataka
- izbjegavanje anomalija pri azuriranju podataka
- smanjenju broja nelogicnih podataka
- izradi strukture podataka pogodne za odrzavanje

Kako je ovdje rijec o teoriji od velike je vaznosti da razumijete primjenjivanu terminologiju. Primjeticete da razlicite knjige kojima ste se opskrbili u cilju izucavanja ove oblasti, upotrebljavaju razlicite termine za opis identicnih pojmova ovisno o tome da li se radi o akademskim ili komercijalnim djelima. U donjoj tabeli termini koristeni u ovoj oblasti grupisani su prema tipu osoba koje se njima koriste.

(9) Znamenke, na engleskom chars
(10) Brojke, na engleskom numbers

Tabela 1.1. *Orijentacije u terminologiji baze podataka*

Teoreticar	***Analiticar***	***Informaticar programer***
Relacija	Jedinka	Tabla
Atribut	Atribut	Kolona
Torka (n-torka)	Linija, konkretna jedinka	Linija, snimak, slog

Pod uslovom da razumijete sta ovi termini predstavljaju, mozete ih ravnopravno upotrebljavati i eventualno povoditi se prema tome koji vam se najvise dopada.

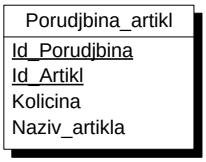
Teorija normalizovanja omogucuje nam da opisemo zeljenu strukturu tabli i kolona unutar baze podataka u skladu sa *pravilima normalizacije*. U ovom poglavlju opisani su prvo, drugo i trece pravilo normalizacije koja jos nazivamo i *normalnim formama*, te se stoga cesto susrecemo sa kraticama 1NF, 2NF i 3NF. Ako vam ovi pojmovi izgledaju suvise uopsteni i teoretski vidjecete da su oni u stvari izuzetno konkretni i od velike vaznosti. Preostala pravila normalizacije, Boyce-Codd-ova, cetvrta i peta normalna forma, ticu se znatno slozenijih problema i prevazilaze okvire knjige koja se bavi osnovama baza podataka.

1.8.1. Prvo pravilo normalizacije (Prva normalna forma 1NF)

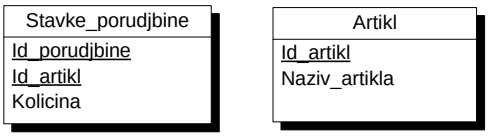
Jedinka (tabla) zadovoljava prvo pravilo normalizacije ukoliko su svi atributi koje posjeduje *elementarni*. Atribut je elementaran ukoliko sadrzi jedinstvenu informaciju o jedinki. Primjera radi navedimo da u jedinki koja je namjenjena pohranjivanju informacija o zaposlenima necete koristiti atribut Izdrzavane_osobe u koji biste unosili ime osoba koje zaposleni izdrzava. Obzirom da jedan zaposleni moze imati vise osoba koje izdrzava (ili nijednu) ova informacija nije elementarna.

1.8.2. Drugo pravilo normalizacije (Druga normalna forma 2NF)

Ovo pravilo podrazumijeva da tabla zadovoljava prvo pravilo normalizacije i tice se samo tabli sa komponovanim primarnim kljucem (komponovanim od vise atributa). Mozemo zakljuciti da se drugo pravilo normalizacije svodi na *normalizaciju relacija* koristenih u dijagramima Jedinka-Relacija.
Drugo pravilo normalizacije zahtijeva da svi atributi (kolone), koji ne ulaze u sastav primarnog kljuca, ovise iskljucivo i u cjelosti o primarnom kljucu jedinke. Drugim rijecima ovo pravilo zahtijeva da tabla ne sadrzi izlisne informacije.
Uzmimo za primjer tablu Porudjba_Artikl koja identifikuje artikle jedne porudjbine i ciji je primarni kljuc komponovan od Id_Porudjbina i Id_Artikl. Tabla koja pored atributa Kolicina sadrzava i atribut Naziv_artikla ne zadovoljava drugo pravilo normalizacije buduci da atribut Naziv_artikla zavisi samo od dijela primarnog kljuca - Id_Artikla.



Svodjenje na drugu normalnu formu zahtijevalo bi da ovu tablu nadomjestimo tablama Stavke_porudjbine i Artikl, koje bi obje zadovoljavale drugo pravilo normalizacije.

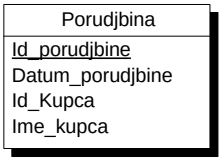


1.8.3. Trece pravilo normalizacije (Treca normalna forma 3NF)

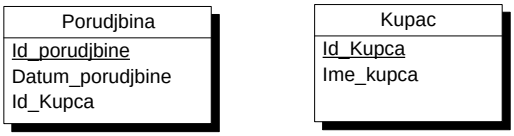
Ovo pravilo normalizacije podrazumijeva da se tabla vec nalazi u drugoj normalnoj formi. Cilj primjene treceg normalnog pravila jeste da se izbjegnu tranzitivne (posredne) ovisnosti u okviru jedne table. Pri privodjenju table u trecu normalnu formu slijedite navedene etape :

- Zadrzite u tabli samo atribute ovisne direktno o primarnom kljucu u njegovoj cjelosti.
- Grupisite u jednu tablu sve atribute koji su tranzitivno ovisni, atribut o kojem su ovisni navedeni atributi duplicirajte u pocetnoj tabli i odaberite ga za primarni kljuc nove table.

Kao primjer navodimo tablu Porudjbina u kojoj atribut Ime_Kupca ovisi tranzitivno o primarnom kljucu (posredstvom atributa Id_Kupca koji ovisi direktno o primarnom kljucu).



Primjena treceg pravila normalizacije navodi nas da nadomjestimo tablu Porudjbina sa dvije slijedece table.



1.9. Normalizacija baze podataka

Kao sto smo napomenuli ranije, teorija normalizacije se ne zaustavlja na trecem pravilu normalizacije. Po pitanju primjene teorije vodi se debata izmedju « teorijskih cistunaca » i profesionalaca gdje prvi smatraju da sve table moraju biti barem u trecoj normalnoj formi dok profesionalci zagovaraju *denormalizaciju* u cilju poboljsanja performansi. Pod denormalizacijom oni podrazumijevaju reduciranje stepena normalizacije iz trece u drugu normalnu formu.

Preporuke

Dobro
Ucinite sve kako bi vasa shema bila u trecoj normalnoj formi (3NF)

Pogresno
Predpostaviti da su performance lose bez obavljanja kompletnih i preciznih ispitivanja. Zasnujte testove performansi na uzorku koji odgovara realnom stanju baze u eksploataciji. Tek nakon ovakvih testova mozete razmatrati mogucnost denormalizacije strukture baze podataka.

Dobro
Nastojte rjesavati probleme performanci dajuci prednost materijalnim poboljsanjima nad kompromisima u koncepciji baze podataka. Planirajuci na duzi vremenski period, materijal vise kvalitete je jeftiniji od odrzavanja denormalizovane baze podataka.

Pogresno
Denormalizovati table bez potpunog razumijevanja nedostataka primjenjenog tehnickog kompromisa. Primjer : poboljsanje performanci uz povecanje broja izlisnih podataka, kompleksnu logiku azuriranja i dodatne poteskoce u evoluiranju baze tokom njenog zivotnog ciklusa.

Definicija

Shema baze podataka je skup objekata koji otjelotvoruju logicki model baze podataka : table, kolone, primarni kljucevi, strani kljucevi itd.

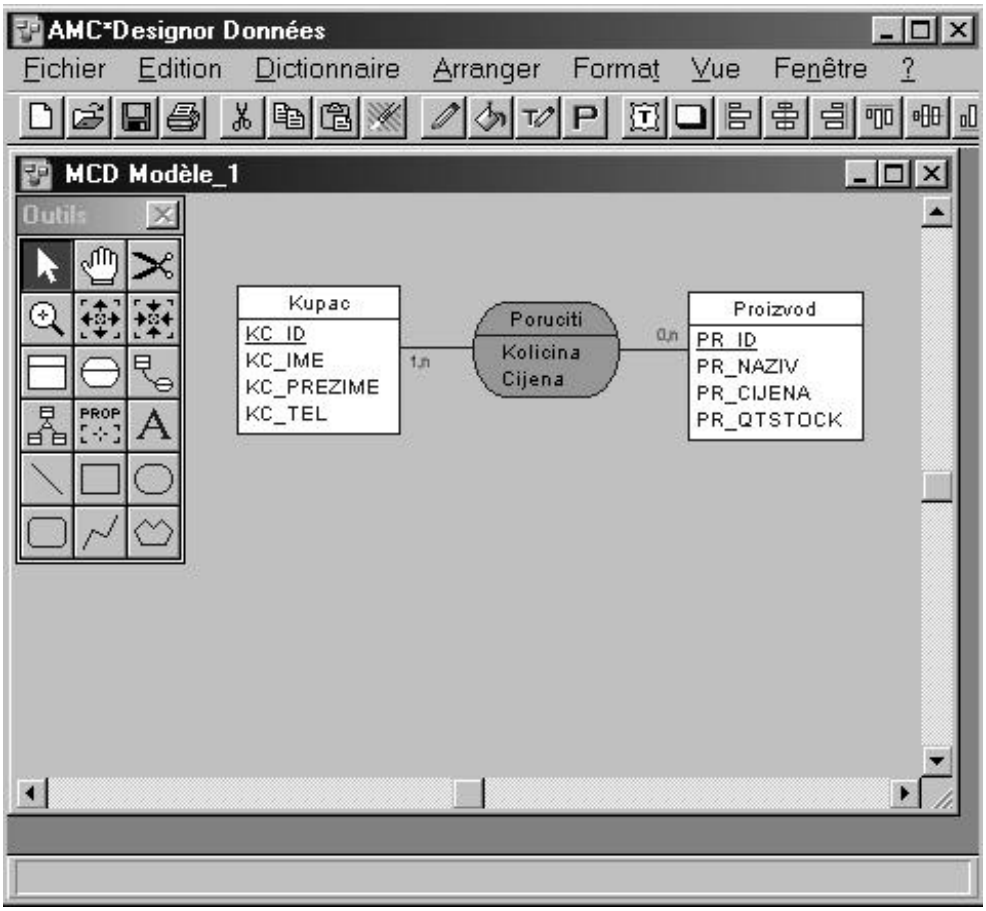
1.10. Alati za izradu dijagrama Jedinka-Relacija

Da biste si olaksali posao izrade logickog modela baze podataka mozete u tu svrhu koristiti jedan od postojecih alata za modeliziranje : Designer/2000, Oracle Data Designer, Sybase S-Designer ...
Svi ovi alati su raspolozivi u vezijama Windows 9x ili Windows NT. Ovi alati omogucuju da na lak nacin, u maniru alata za crtanje, dodajemo jedinke u model, stvaramo relacije izmedju njih, mijenjamo osobine jedinki i relacija itd. Prednosti upotrebe ovih alata su slijedece :

- Pojednostavljuju postupak izrade dijagrama.
- Ukljucuju alate za automatski prelaz iz logickog u fizicki model baze podataka.
- Proizvode instrukcije SQL za kreiranje tabli, indeksa, kljuceva i drugih elemenata baze podataka.
- Ukljucuju elemente postojece standardizacije i omogucuju izradu dokumentacije.

Pojedini od ovih alata kao Oracle Designer/2000 spadaju u grupu tzv. CASE proizvoda i domen njihove upotrebe je znatno siri.

Na donjoj slici prikazan je AMC*Designor preduzeca Sybase. Rijec je o starijoj verziji za koju je autor sentimentalno vezan i cija je glavna vrijednost modul posvecen dizajniranju baze podataka. Jedna od prednosti ovog alata je i ta da je uskladjen sa metodom MERISE, jednom od osnovnih metoda u oblasti izrade informacijskih sistema koristenih u Francuskoj.



1.11. Oracle terminologija : prikljucak, korisnik i seansa.

Prije nego sto pocnemo govoriti o nacinu implementacije sheme baze podataka u konkretnu bazu podataka osvrnucemo se na pojmove i Oracle alate cije poznavanje je neophodan preduslov. Dva pojma zasluzuju posebnu paznju : *prikljucak* na bazu podataka i *korisnik* baze podataka.

Definicija *Prikljucak na bazu podataka predstavlja korisnika koji je proslijedio ime i lozinku i koji je u prilici da proslijedjuje zahtijeve (upite) doticnoj bazi podataka.*
Korisnik baze podataka je proces koji raspolaze jedinstvenim imenom i lozinkom i kojeg kao takvog prepoznaje baza podataka.

1.11.1 Prikljucak na bazu podataka

Prikljuciti se na bazu podataka znaci proslijediti bazi podataka vazece *ime* korisnika i *lozinku* koji ce kao takvi biti prihvaceni od strane baze podataka. Korisnik se moze prikljuciti na Oracle bazu podataka koristeci ma koji od Oracle alata ili neki od « trecih » alata (sto ce reci alata koji nisu proizvod marke Oracle). Pojmovi prikljucka, seanse i prikljucka na bazu podataka ravnopravno se koriste i sve se odnose na jednu te istu stvar : aktivnosti koje korisnik baze podataka obavi od trenutka prikljucenja do trenutka iskljucenja.
Prikljucak na bazu podataka moze biti *lokalni* ili *daljinski*. *Lokalni* prikljucak ukazuje na to da se program koji korisnik upotrebljava i baza podataka nalaze na istoj masini. *Daljinski* prikljucak, naprotiv, ukazuje na cinjenicu da se koristeni program i baza podataka nalaze na razlicitim masinama.

1.11.2 Korisnik baze podataka

Svaki prikljucak na bazu podataka je obavljen za racun jednog korisnika. Pojmovi koji se ravnopravno koriste i koji ukazuju na korisnika baze podataka su : vlasnik table, korisnik i konto/racun (nacinite paralelu sa vasim bankovnim kontom/racunom).
Vlasnik table je istovremeno korisnik baze podataka koji, istina, posjeduje tablu ali ostaje iznad svega korisnik. Korisnik baze podataka moze ne posjedovati nijednu tablu u bazi podataka i istovremeno imati pravo izvrsavanja upita i eventualno azuriranja nad tablama u vlasnistvu drugih korisnika.
Za svaku tablu u bazi podataka mora postojati korisnik koji je posjeduje. Jedan od pristupa se sastoji u tome da kreiramo vlasnika tabli koji odgovara organizaciji a ne pojedinacnom clanu organizacije. U stvarnosti , podaci jedne organizacije ne pripadaju nikome osobno. Imajuci u vidu dinamicki karakter organizacije mozemo reci da ljudi, clanovi organizacije, dolaze i odlaze dok organizacija traje.
Za potrebe realizacije pokazne baze podataka kreiracemo jednoga korisnika kojeg cemo nazvati SAFUSR i koji ce biti vlasnik svih tabli koje cemo potom kreirati.

1.12. Prikljucak na Oracle bazu podataka

Prije nego sto pristupite kreiranju novog korisnika morate znati kako da se prikljucite na bazu podataka. Vecina Oracle alata vam prikazuje na startu ekran u koji trebate unijeti informacije neophodne za prikljucenje na bazu. Podaci koji su zahtijevani u ovom ekranu su :

- ime korisnika koje koristite za prikljucak
- lozinka
- ime baze podataka

Ukoliko koristite Personal Oracle unosenje imena baze podataka nije neophodno. Ukoliko se radi o serveru Oracle rasporedjenom na jednoj od masina u mrezi moracete unijeti ime baze podataka.