Agjencia Kombëtare e Arsimit, Formimit Profesional dhe Kualifikimeve

Sektori i Hartimit të Kurrikulave dhe Materialeve Mbështetëse

MATERIAL MËSIMOR

Në mbështetje të mësuesve të drejtimit mësimor

BAZA TË DHËNASH

Niveli IV në KSHK

Ky material mësimor i referohet:

- **Lëndës profesionale:" Baza të dhënash",**

  kl.12 (L-26-561-21).

Përgatiti:
Jetmir Shtjefni
Anisa Melishte
Ornela Çerpja
Besmir Kanushi

Tiranë, 2022

Tema 1. Bazat e të dhënave dhe përdoruesit e tyre

Bazat e të dhënave dhe sistemet e bazave të të dhënave janë bërë një komponent themelor në jetën e përditshme në shoqërinë moderne. Gjatë një dite, shumë prej nesh ndeshen me disa aktivitete të cilat shkaktojnë bashkëveprim me një bazë të dhënash.
Ndërveprimi me bazat e të dhënave është bërë tashmë i zakonshëm: gjetja e një numri në listën e kontakteve telefonike, postimi në Facebook, etj. Në këtë pjesë do të paraqiten elementet përbërës të bazave të të dhënave dhe funksioni i saj për te ofruar informacione të rëndësishme me shpejtësi dhe lehtësi.

Blloqet përbërës të një baze të dhënash
Një bazë të dhënash është një koleksion i të dhënave të lidhura që mund të rezervohen, renditen, organizohen dhe pyeten. Shumë nga veprimet si përdorimi i një ATM-je, blerjet online, dhe regjistri i klasës, krijojnë të dhëna që duhet të rezervohen, menaxhohen dhe të përdoren nga të tjerët. Me shumë gjasa, për secilin nga këta probleme është krijuar një bazë të dhënash që merr dhe rezervon të dhënat e krijuara dhe lejon që këto të dhëna të trajtohen dhe të përdoren nga të tjerët. Duke krijuar një strukturë të organizuar për të dhënat, baza e të dhënave i bën më të kuptimta të dhënat dhe si rrjedhim më të dobishme. Bazat e të dhënave i kthejnë në mënyrë efektive të dhënat në informacion.

Pse duhet të merren njohuri mbi bazat e të dhënave? Përderisa njerëzit ndërveprojnë me bazat e të dhënave çdo ditë, të kuptuarit sesi bazat e të dhënave funksionojnë dhe se çfarë mund të bëhet me bazat e të dhënave do të ndihmojë njerëzit të ndërveprojnë me to me më shumë efektivitet. Për shembull, kategorizimi, renditja dhe filtrimi janë atribute kyç për shumicën e bazave të të dhënave. Nëse një bazë të dhënash nuk është ngritur në mënyrë korrekte ose nëse përdoruesi nuk di si ta përdorë bazën e të dhënave, ai nuk mund të marrë informacionin që kërkon.

Hyrje në bazat e të dhënave (shtresat)
Një bazë të dhënash (Databazë) është një grumbullim i të dhënave të lidhura dhe të organizuara në një mënyrë të tillë që të dhënat të jetë e mundur të gjenden, administrohen dhe përditësohen lehtësisht. Bazat e të dhënave mund të përfytyrohet si një magazinë ku të dhënat e jetës reale ruhen dhe nga përpunimi i tyre nxirret një informacion.

- **Të dhënat** janë fakte të jetës reale, si për shembull emri, mosha, nota mesatere e një studenti.
- **Informacionet** janë të dhëna të organizuara ose të përgatitura në një formë e cila përdoret në vendim-marrje, si për shembull lista e pesë studentëve me mesataren më të lartë.

Një sistem menaxhimi i bazave të të dhënave (DBMS) është një software që mundëson krijimin, ruajtjen dhe manipulimin e të dhënave. Një DBMS është një mjet i përdorur për të kryer çdo lloj veprimi me të dhënat në një databazë, për të siguruar mbrojtje dhe siguri për databazat, dhe për të mbikqyrur konsistencën e të dhënave në rastin e përdoruesve të shumtë. Disa prej sistemeve DBMS më të njohura që janë sot në shfrytëzim janë MySql, Oracle, Sybase, Microsoft SQL Server, PostgreSQL, IBM DB2, etj.

Pjesët përbërëse të një sistemi bazë të dhënash jepet në figurën 1-1. Përdoruesi nënkupton administratorin e bazës së të dhënave, zhvilluesin e sistemit dhe përdoruesit fundorë.

Figura.1-1 Pjesët përbërëse të një sistemi bazë të dhënash (DBMS)

Administratori i bazës së të dhënave
Çdo kompani që përdor një sistem të menaxhimit të bazës së të dhënave (DBMS) për të menaxhuar të dhënat, kërkon minimumin një administrator i cili të sigurojë përdorimin efektiv të bazës së të dhënave të kompanisë.
Administratori i bazës së të dhënave ose DBA (DataBase Administrator), është një specialist përgjegjës për funksionimin e rregullt të bazave të të dhënave dhe është personi i parë që kontaktohet në raste problemesh të aplikacioneve që i aksesojnë këto të dhëna.

Utilitetet e sistemit të bazës së të dhënave
Shumica e DBMS-ve kanë utilitete që e ndihmojnë DBA të menaxhojë sistemin e DB.
- *Ngarkimi* - përdoret për të ngarkuar skedarët ekzistues të të dhënave, si skedarët tekst apo sekuencialë në DB
- *Backup* - krijon një kopje backup-i të bazës së të dhënave
- *Riorganizimi i skedarëve* - përdoret për të riorganizuar skedarë të DB në një organizim të ndryshëm të tij për të rritur performancën
- *Monitorimi i performancës* - monitoron përdorimin e DB dhe siguron statistika për DBA.

Tema 2. Karakteristikat e përafrimit të bazës së të dhënave

DBMS është një softuer sistemi me qëllim të përgjithshëm i cili lehtëson proceset e *përcaktimit, ndërtimit, manipulimit* dhe të përdorurit bashkë (*sharing*) të DB midis përdoruesve dhe aplikacioneve të ndryshëm.
Përcaktimi i DB përfshin specifikimin e tipeve të të dhënave, strukturave dhe konstreineve (shtrëngesë, detyrim) për të dhënat që do ruhen në DB. Përcaktimi i bazës së të dhënave ruhet gjithashtu nga DBMS në formën e katalogut ose fjalorit të DB; ai quhet **meta-data**. DBMS

ka gjuhën e përcaktimit të të dhënave (DDL - Data Definition Language) për ndërtimin ose ngarkimin e strukturës fillestare dhe të dhënave nga një medium tjetër.
Ndërtimi i DB është procesi i ruajtjes së vetë të dhënave në ndonjë medium (depozitë) e cila kontrollohet nga DBMS.
Manipulimi i DB përfshin funksione të tilla si bërja e kërkesave (query) ndaj DB për të marrë të dhëna specifike dhe për të gjeneruar raporte nga të dhënat, modifikimi i DB për të reflektuar ndryshimet në minibotë duke shtuar, fshirë apo modifikuar përmbajtjen e saj, aksesimi i bazës së të dhënave nëpërmjet aplikacioneve Web. DBMS ka gjuhën e manipulimit të të dhënave (DML - Data Manipulation Language) për të realizuar këto veprime.

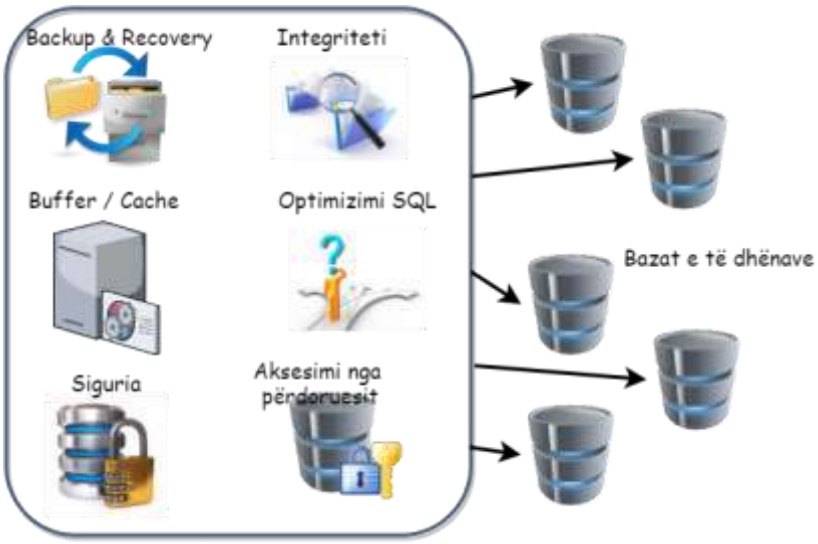


Figura 2-1 Funksionet dhe shërbimet e DBMS

Funksionet dhe shërbimet kryesore të një sistemi DBMS janë:

i. **Menaxhimi i fjalorit të të dhënave**. DBMS ruan përkufizimet e elementëve të të dhënave dhe marrëdhëniet midis tyre (metadata) në një fjalor të të dhënave.
ii. **Menaxhimi i ruajtjes së të dhënave**. DBMS krijon dhe menaxhon strukturat komplekse që nevojiten për ruajtjen e të dhënave, kështu ju lehtëson punën e vështirë të përcaktimit dhe programimit të karakteristikave fizike të të dhënave. *Rregullimi i performancës* lidhet me aktivitetet që e bëjnë bazën e të dhënave të kryejnë në mënyrë më efikase dhe të shpejtë ruajtjen dhe aksesimin e të dhënave.
iii. **Përpunimi dhe prezantimi i të dhënave**. DBMS transformon të dhënat në hyrje në përputhje me strukturën e të dhënave. (data)
iv. **Menaxhimi i sigurisë**. DBMS krijon një sistem sigurie që forcon sigurinë e përdoruesit dhe privatësinë e të dhënave.
v. **Menaxhimi i shumë përdoruesve**. Për të siguruar integritetin dhe qëndrueshmërinë e të dhënave, DBMS përdor algoritme të sofistikuara për të siguruar që përdorues të shumtë mund të përdorin bazën e të dhënave njëkohësisht pa kompromentuar integritetin e databazës.
vi. **Menaxhimi i backup dhe recover**. DBMS ofron mundësinë për duplikimin (backup) dhe rekuperimin (recover) e të dhënave për të siguruar sigurinë dhe integritetin e tyre.

vii. **Menaxhimi i integritetit të të dhënave**. DBMS promovon dhe zbaton rregullat e integritetit, duke minimizuar kështu tepricën e të dhënave dhe maksimizuar qëndrueshmërinë e tyre.

viii. **Gjuhët e aksesimit të bazave të të dhënave dhe ndërfaqja e programimit të aplikacioneve**. DBMS ofron aksesimin e të dhënave nëpërmjet një gjuhe të bazuar në pyetje. Kjo *gjuhë me pyetje (query language)* është joproceduriale - që lejon përdoruesin të specifikojë se çfarë duhet të bëhet pa pasur nevojë të specifikojë se si duhet bërë.

ix. **Ndërfaqja e komunikimit e bazës së të dhënave**. Gjeneratat aktuale të DBMS pranojnë kërkesat e përdoruesve nëpërmjet rrjeteve të shumtë dhe të ndryshëm:
   - Përdoruesit mund të marrin përgjigje për pyetjet e tyre duke plotësuar një format nga ekrani nëpërmjet shfletuesit.
   - DBMS mund të publikojë automatikisht raporte të paracaktuara në një faqe interneti.
   - DBMS mund të lidhet me sisteme të tjerë për të shpërndarë informacionin nëpërmjet e-mail ose aplikacioneve të tjera.

Tema 3. Aktorët në skenë dhe avantazhet e përdorimit të DBMS

Një **bazë të dhënash** është një strukturë e integruar kompjuterike që ruan një koleksion:
- Të dhëna përfundimtare (end-user data), që janë fakte të papërpunuara në interes të përdoruesve
- **Metadata**, ose të dhëna rreth të dhënave, që shërbejnë për të integruar dhe menaxhuar të dhënat përfundimtare

Një **sistem i menaxhimit të të dhënave** (DBMS) është një koleksion i programeve që menaxhojnë strukturën e bazës së të dhënave dhe kontrollojnë aksesimin e të dhënave të ruajtura në të

Roli dhe avantazhet e DBMS

DBMS Sistemi i menaxhimit të të dhënave:

- shërben si ndërmjetës midis përdoruesit dhe bazës së të dhënave
- i fsheh përdoruesve dhe programeve kompleksitetin e brendshëm të databazës
- Mundëson që të dhënat të ndahen midis shumë aplikacioneve apo përdoruesish
- Integron pikëpamjet e ndryshme të përdoruesve për të dhënat në një depo të vetme gjithë-përfshirëse

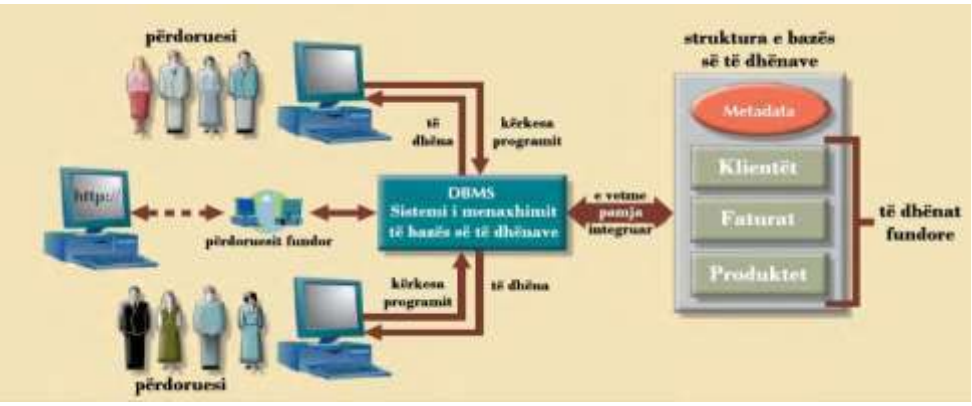


Figura 2-2 DBMS menaxhon ndërveprimin midis përdoruesit dhe bazës së të dhënave

Avantazhet e DBMS

I. **Përmirësim në shpërndarjen e të dhënave.** Ndihmon në krijimin e një mjedisi në të cilin përdoruesit e kanë më të lehtë aksesin mbi një sasi më të madhe të dhënash.

II. **Përmirësimi i sigurisë së të dhënave.** Rritja e numrit të përdoruesve sjell më tepër rreziqe për sigurinë e tyre. DBMS ofron një strukturë që ndihmon në rritjen e privatësisë dhe sigurisë së të dhënave. (përdorues me akses të limituar)

III. **Integrim më i mirë i të dhënave.** Një akses më i gjerë në të dhënat e mirë-menaxhuara ndihmon që të kemi një pamje të integruar dhe më të qartë të operacioneve të organizatës.

IV. **Minimizim të mospërputhjeve midis të dhënave.** *Mospërputhjet e të dhënave* ekzistojnë kur versione të ndryshme të të dhënave të njëjta shfaqen në vende të ndryshme. (Departamentet hedhin: "Emër Mbiemër" dhe "E. Mbiemër“).

V. **Përmirësim në aksesimin e të dhënave.** DBMS bën të mundur kthimin e përgjigjeve të shpejta ndaj pyetjeve të rastit. Nga pikëpamja e bazës së të dhënave, një pyetje (query) është një kërkesë specifike që i drejtohet DBMS për përpunimin e të dhënave.

VI. **Vendim marrje e përmirësuar.** Të dhënat e mirë-menaxhuara dhe përmirësimi i mënyrës së aksesimit të tyre bën të mundur gjenerimin e një informacioni me cilësi më të lartë, që ndikon në marrjen e vendimeve më të mira. Që *të dhënat* të konsiderohen *cilësore* duhet ti kushtohet vëmendje saktësisë, vlefshmërisë dhe afateve kohore të tyre.

VII. **Rritja e produktivitetit të përdoruesve.** Disponueshmëria e të dhënave, e kombinuar me mjetet që i transformojnë të dhënat në informacion të dobishëm, i jep mundësinë përdoruesve të marin vendime të shpejta të bazuara mbi informacion.

Tema 4. Tipet e organizimit të bazës së të dhënave

Llojet e bazave të të dhënave

Bazat e të dhënave mund të klasifikohen duke u bazuar mbi: Numrin e përdoruesve, Vendndodhjen/t e të dhënave, Llojin e përdorimit dhe Shkallën e strukturimit të të dhënave.

1. Numri i përdoruesve

i. **DB me një përdorues (single-user database)** i shërben vetëm një përdorues në një kohë të caktuar. Kur ndodhet në një PC quhet desktop DB.

ii. **DB me shumë përdorues (multiuser database)** përballon shumë përdorues në të njëjtën kohë. Kur përballon deri në 50 persona quhet **bazë të dhënash për një grup pune (workgroup database)**. Për një numër më të madh se 50 quhet **bazë të dhënash ndërmarrje (enterprise database).**

2. Vendndodhja

i. Nëse të dhënat janë të vendosura në një vend të vetëm kemi të bëjmë me një **bazë të dhënash të centralizuar (centralized database).**

ii. Kur të dhënat janë të shpërndara nëpër vende të ndryshme atëhere quhet një **bazë të dhënash e shpërndarë (distributed database).**

3. Mënyra e përdorimit

i. Baza e të dhënave e projektuar për të mbështetur operacionet e përditshme të kompanisë klasifikohet si një **bazë të dhënash operative (operational/ transactional/production db).** Përqëndrohen kryesisht në saktësinë dhe shpejtësinë e ruajtjes së të dhënave. (psh: shitje produkti apo sherbimesh, pagesa dhe blerje, furnizimi).

ii. Një **depo të dhënash (data warehouse)** fokusohet kryesisht në magazinimin e të dhënave që përdoren për të gjeneruar informacionin e nevojshëm për të marrë vendime strategjike *(vendosja e cmimeve, parashikimi shitjeve, produkt i ri*). Përqëndrohet në sasi të mëdha të dhënash dhe përpunimin e tyre.

4. Shkalla e strukturimit të të dhënave

i. **Të dhënat e pastrukturuara** janë të dhënat që ekzistojnë në formën e tyre origjinale. Ato ekzistojnë në një format që nuk lejon përpunimin për të nxjerrë informacionin që përmbajnë.

ii. **Të dhënat e strukturuara** janë rezultat i përpunimit të të dhënave të pastrukturuar në mënyrë të tillë për të lehtësuar ruajtjen, përdorimin, dhe gjenerimin e informacionit.

iii. **Të dhënat gjysëm të strukturuara** janë të dhëna të cilat janë përpunuar në një farë mase. **XML** është një gjuhë e veçantë që përdoret për të përfaqësuar dhe përpunuar elementet e të dhënave në një format tekst. (524.809 text) Tabela

Tabela 4-1 Llojet e bazave të të dhënave

Llojet e bazave të të dhënave								
Produkti	numri i përdoruesve			vendndodhje e të dhënave		Përdorimi i të dhënave		XML
	Me një përdorues	Me shumë përdorues		Centralizuar	Shpërndarë	Operative	Data Warehouse	
		Grup pune	Ndërmarrje					
MS Access	X	X		X		X		
MS SQL Server	X**	X	X	X	X	X	X	X
IBM DB2	X**	X	X	X	X	X	X	X
MySQL	X	X	X	X	X	X	X	X*
Oracle DBMS	X**	X	X	X	X	X	X	X

* Mbështet vetëm funksionet XML. Të dhënat XML ruhen në objekte të mëdha tekst.
** Shitësi ofron version të veçantë personal të DBMS me një përdorues.

Tema 5. Modelet e të dhënave, skemat dhe instancat

Modelet e të dhënave (modeli konceptual, llogjik, fizik)

Modelimi i të dhënave është hapi i parë që ndjekin zhvilluesit në dizenjimin e një databaze. Një model baze të dhënash përkufizohet si një bashkësi konceptesh dhe rregullash të të dhënave që përcaktojnë strukturën e databazës dhe se si ato mund të ruhen, organizohen dhe manipulohen.

Së pari, paraqesim pjesët përbërëse kryesore të një modeli të dhënash dhe terminologjinë përkatëse.

i. **Entiteti**. Përkufizohet si diçka që ekziston dhe është e mundur të përshkruhet, si për shembull studenti, pedagogu, kursi i studimit në një universitet. Një entitet përmban

njësitë që njihen si raste apo skeda. Për shembull, entiteti "autor" përmban raste apo skeda me informacione të detajuara për autorë individualë.

ii. **Atributi**. Është karakteristikë e një entiteti që identifikon, lidh dhe përshkruan. Një atribut që identifikon rastet e entitetit është një çelës kanditat. Një atribut që lidh entitetet është një çelës i jashtëm. Një atribut është përshkrues nëse shpreh një karakteristikë të një skede entiteti, por nuk identifikon apo lidh. Çdo atributi i caktohet një bashkësi vlerash të vlefshme për elementët e të dhënave që njihet me termin domain. Lloji i të dhënave është një komponent i domain-it.

iii. **Çelësat**. Një çelës përbëhet nga një ose më shumë atribute, vlera e të cilave në mënyrë të vetme identifikon një rast apo skede entiteti dhe përcakton lidhjet midis entiteteve. Një çelës kandidat është një atribut ose bashkësi atributesh që përdoren për të identifikuar në mënyrë të vetme një skede entiteti. Çdo entitet mund të ketë disa çelësa kandidatë, por duhet të ketë të paktën një të tillë. Nga bashkësia e çelësave kandidatë zgjidhet një i quajtur çelës primar i cili është i vetëm dhe do të përdoret për të përcaktuar çelësat e jashtëm në entitetet e varura. Çelësi primar garanton unicitetin e çdo skede entiteti, ku vlera e çdo komponenti të tij nuk mund të jetë e munguar apo e panjohur dhe e pamundur të jetë e ndryshueshme. Çelësat e jashtëm identifikojnë lidhjet midis skedave të entiteteve.

iv. **Lidhjet**. Një lidhje përcakton se si entitete të ndryshme bashkëshoqërohen me njëra tjetrën. Dy tiparet që karakterizojnë një lidhje janë: kardinaliteti dhe detyrueshmëria. Kardinaliteti shpreh numrin e rasteve që ekzistojnë midis një çifti entitetesh të lidhura. Kemi tre lloje kardinalitetesh. Kardinaliteti i tipit "një me një" (1:1) identifikon një lidhje të formës "një pedagog jep e shumta një lëndë mësimore dhe një lëndë mësimore jepet e shumta nga një pedagog". Kardinaliteti i tipit "një më shumë" (1:N) identifikon lidhjen e formës "një pedagog jep shumë lëndë dhe një lëndë jepet e shumta nga një pedagog". Kardinaliteti i tipit "shumë me shumë" (M:M) identifikon lidhjet e formës "një pedagog jep shumë lëndë dhe një lëndë jepet nga shumë pedagogë". Detyrueshmëria shpreh karakteristikën fundore të lidhjes. Për shembull, lidhja 1:1 "lënda jepet nga vetëm një mësues" është e detyrueshme, ndërsa lidhja "lënda mund të jepet nga 1 ose asnjë mësues" është jo e detyrueshme.

Modelimi i të dhënave kryhet në tre faza duke zhvilluar sipas rradhës tre modelet që vijojnë.

Modeli konceptual. Qëllimi i këtij modeli është dizenjimi i databazës në trajtë skematike të pavarur nga software –i i databazës dhe pjesëve fizike. Rezultati përshkruan entitetet kryesore të të dhënave, lidhjet dhe kufizimet e fushave, por jo specifikimin e atributeve dhe çelësave. Përmbajtja është vetëm përshkruese dhe treguese, pra e përbërë nga elementë grafikë dhe përshkrimet me tekst të tyre. Ajo paraqet një strukturë abstrakte të databazës që përfaqëson objektet e botës reale në mënyrën me realiste të mundshme. Në kuptimin praktik ajo paraqet një përshkrim të qartë të biznesit dhe pjesëve funksionale të saj. Hapat që ndiqen në zhvillimin e këtij modeli janë: a) analiza e të dhënave dhe kërkesat; b) modelimi i lidhjeve të entiteteve; c) verifikimi i modelit të të dhënave; d) dizenjimi i modelit përfundimtar të databazës.

Modeli llogjik. Dizenjimi llogjik përbën fazën e dytë të punës në zhvillimin e databazës. Qëllimi i saj është të dizenjojë një databazë bazuar në një model specifik të të dhënave, por të pavarur nga komponentet fizikë. Konkretisht, në këtë fazë kërkohet që të gjithë objektet në modelin konceptual të skicohen në konstrukte të veçanta të përdorura prej modelit të zgjedhur të databazës. Për shembull, modeli llogjik i një databaze relacionale përfshin specifikimet për tabelat, lidhjet dhe kufizimet, atributet dhe çelësat. Hapat që ndiqen në zhvillimin e një

modeli llogjik janë: a) skicimi i modelit konceptual në komponentë të modelit llogjik; b) vleftësimi (validate) i modelit nëpërmjet normalizimit; c) vleftësimi i kufizimeve të integritetit; d) vleftësimi i modelit kundrejt kërkesave të përdoruesit.

Modeli fizik. Dizenjimi fizik përbën proçesin e përcaktimit të organizimit të ruajtjes së të dhënave dhe karakteristikat e aksesit të të dhënave me qëllim garantimin e integritetit, sigurisë dhe performancës. Për shembull, në një model të tillë realizohet konvertimi i entiteteve në tabela dhe atributeve në kollona, përcaktohen lidhjet nëpërmjet çelësave të jashtëm dhe modifikohet modeli i të dhënave bazuar në kufizimet fizike. Hapat e realizimit të këtij modeli janë: a) përcaktimi i organizimit të ruajtjes së të dhënave; b) përcaktimi i komponentëve për të siguruar integritet dhe mbrojtje; c) përcaktimi i komponentëve për të garantuar performancën.

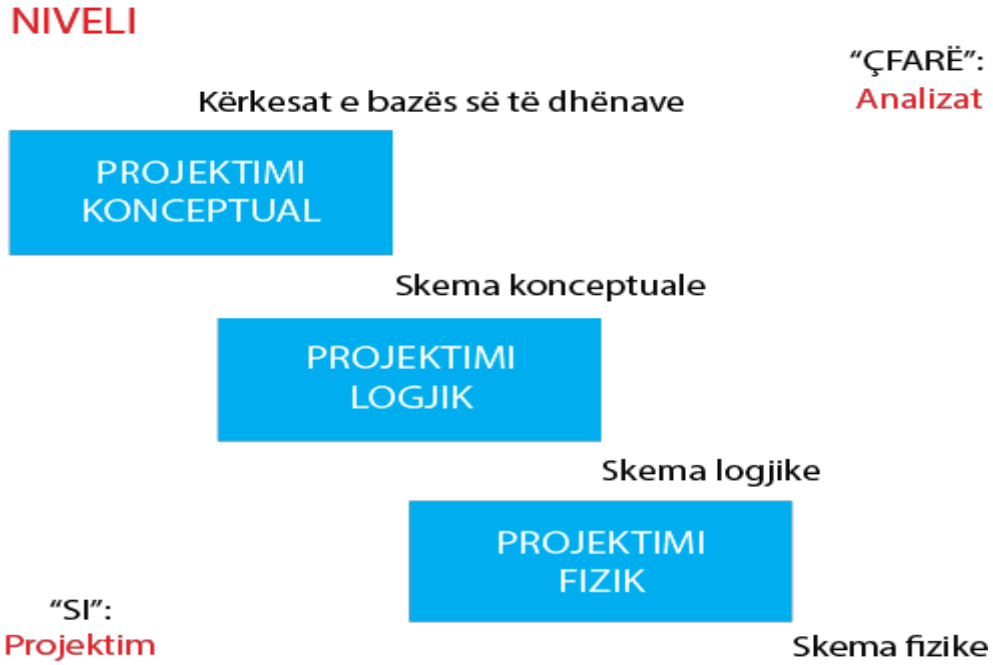


Figura 5- 1. Modeli konceptual, llogjik, fizik

Skemat, instancat dhe gjendja e DB

Në çdo model të dhënash është e rëndësishme të dallojmë përshkrimin e Databazes nga vetë DB.

- **Përshkrimi i bazës** së të dhënave quhet **skema e DB** e cila specifikohet gjatë dizenjimit të saj dhe nuk pritet të ndryshojë shpesh.
- Shumë modele të dhënash kanë disa konvencione për të paraqitur skemat si diagrama, e cila quhet **diagrami i skemes.** (shiko figurwn 5.2)
- Çdo objekt në skemë si STUDENT ose LËNDA do t'a quajmë **konstrukt i skemës**.

Të dhënat në bazën e të dhënave në një moment të caktuar në kohë quhet **gjëndja e saj** ose **snapshot**. Kur përcaktojmë një DB të re, ia specifikojmë skemën e saj vetëm DBMS. Gjendja e saj wshtw **bosh.**

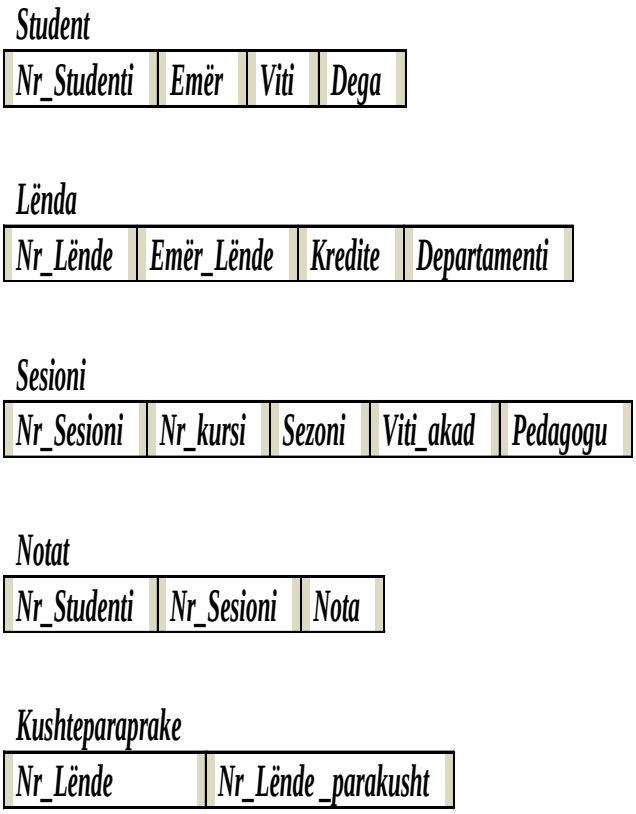


Figura. 5-2 Diagrama relacionale

Tema 6. Arkitekturë me tre skema dhe pavarësia e të dhënave

Dizenjimi i një DBMS varet nga arkitektura e tij, e cila mund të jetë 1-shtresore apo n-shtresore. Arkitektura n-shtresore e ndan sistemin në n module të pavarura që mund të modifikohen, ndryshohen apo zëvendësohen në mënyrë të pavarur.

Në arkitekturën 1-shtresore, përdoruesi në mënyrë të drejtpërdrejtë është pjesë e sistemit dhe e përdor atë. Në arkitekturën 2-shtresore, ekziston një aplikacion përmes së cilës mund të aksesohet sistemi DBMS. Në këtë rast, aplikacioni është tërësisht i pavarur prej databazës në kuptimin e operimit, dizenjimit dhe programimit. Arkitektura 3-shtresore i ndan komponentët e saj nga njëra tjetra bazuar në kompleksitetin e përdoruesve dhe se si ata i përdorin të dhënat në databazë. Shtresat në arkitekturën 3-shtresore janë:

i. **Databaza**. Në këtë nivel, baza e të dhënave qëndron së bashku me gjuhët e saj të përpunimit "query". Ne gjithashtu kemi marrëdhënie që përcaktojnë të dhënat dhe kufizimet e tyre në këtë nivel.
ii. **Aplikacioni**. Në këtë nivel qëndrojnë serverat e aplikacionit dhe programet që aksesojnë bazën e të dhënave. Për një përdorues, kjo paraqet një pamje abstrakte të të dhënave. Shtresa e aplikacionit qëndron në mes dhe vepron si një ndërmjetës midis përdoruesve fundorë dhe bazës së të dhënave.
iii. **Përdoruesi**. Përdoruesit fundorë operojnë në këtë nivel dhe nuk dinë asgjë rreth ekzistencës së databazës përtej kësaj shtrese. Në këtë shtrese, pamje shumëfishe të bazës së të dhënave mund të nxirren prej aplikacionit, të cilat janë të gjeneruara prej aplikacioneve që qëndrojnë në shtresën e mësipërme të aplikacionit.

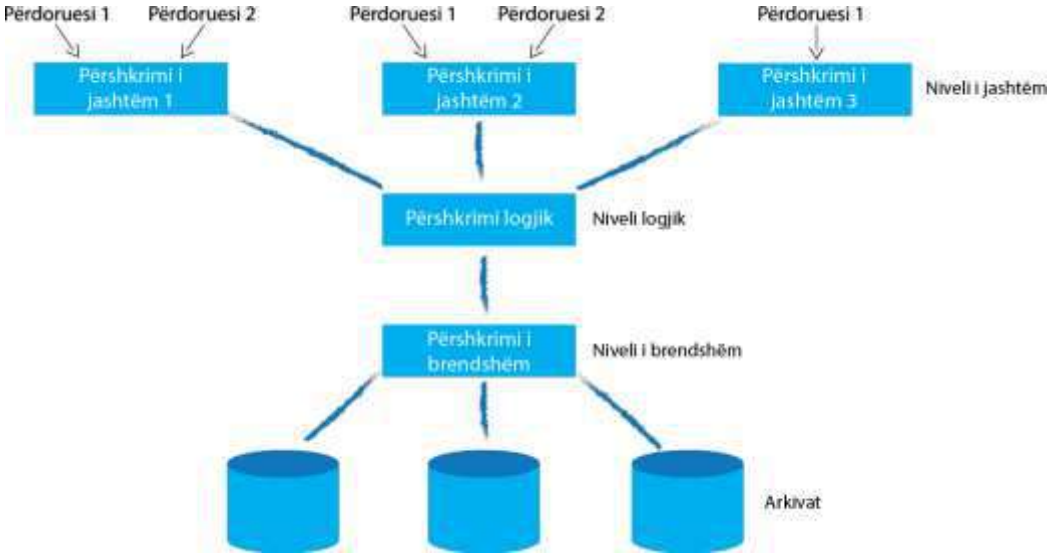


Figura 6-1 Arkitektura me tre skema

Arkitektura me tre nivele

Qëllimi i arkitekturës me tre nivele është që të ndajë aplikacionet e përdoruesit nga baza e të dhënave fizike.

1. **Niveli i brendshëm** ka një skemë të brendshme, e cila përshkruan strukturën e depozitimit fizik të bazës së të dhënave.
2. **Niveli konceptual** ka skemën konceptuale, e cila përshkruan strukturën e të gjithë DB për komunitetin e përdoruesve.
3. **Niveli i jashtëm** përfshin një numër skemash të jashtme ose pamjet e përdoruesve. Çdo skemë e jashtme përshkruan pjesën e bazës së të dhënave për të cilën është i interesuar në një grup përdoruesish dhe fsheh pjesën tjetër të saj nga ky grup përdoruesish.

Proceset e transformimit të kërkesave dhe rezultateve midis niveleve quhen **përshtatje** (mappings).

Figura 6-2 Arkitektura me tre nivele

Pavarësia e të dhënave

Pavarësia e të dhënave, e cila përcaktohet si aftësi për të ndryshuar skemën në një nivel të sistemit të DB pa ndryshuar skemën në nivelin tjetër më lartë.

- **Pavarësia llogjike e të dhënave** është aftësia për të ndryshuar skemën konceptuale pa ndryshuar skemën e jashtme ose aplikacionet.
- **Pavarësia fizike e të dhënave** është aftësia për të ndryshuar skemën e brendshme pa ndryshuar skemën konceptuale.

Pavarësia e të dhënave është përmbushur sepse kur skema ndryshohet në ndonjë nivel, skema e nivelit tjetër më të lartë mbetet e pandryshuar.

Tema 7. Mjedisi i sistemit të bazës së të dhënave, gjuhët e DB dhe ndërfaqet

Termi **sistemi i bazës së të dhënave (database system)** i referohet një organizimi të komponentëve që përcaktojnë dhe rregullojnë mbledhjen, ruajtjen, menaxhimin dhe përdorimin e të dhënave brenda mjedisit të bazës së të dhënave. Nga një pikëpamje e përgjithshme e menaxhimit, sistemi i bazës së të dhënave është i përbërë nga pesë pjesë kryesore: pajisjet fizike (hardware), programet (software), njerëzit, procedurat dhe të dhënat.

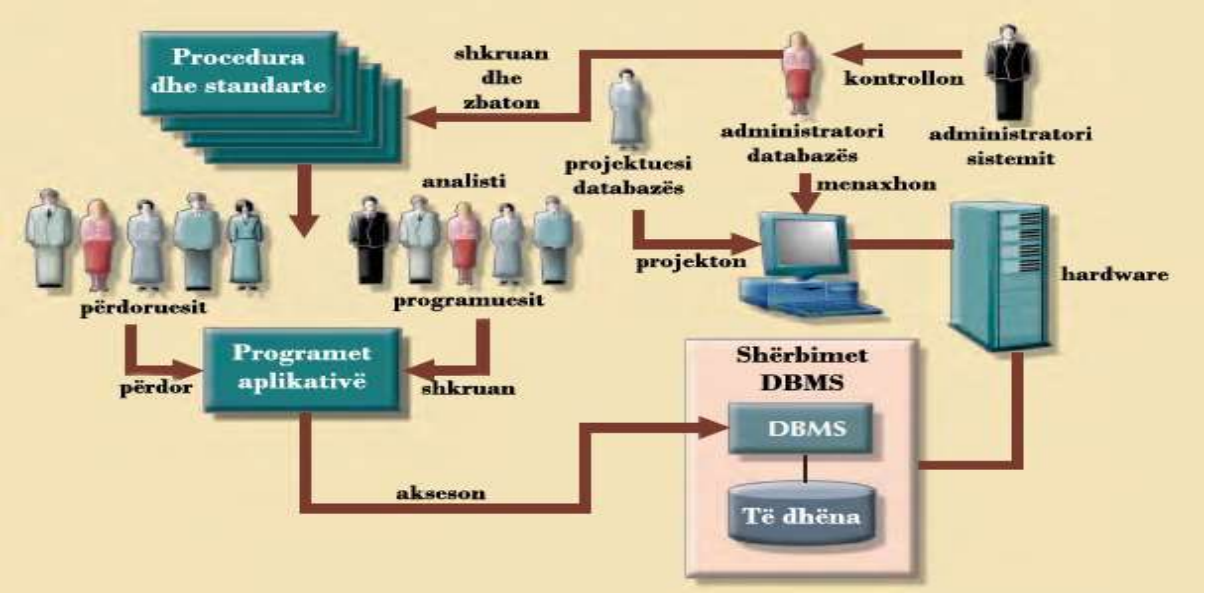


Figura. 7-1 Mjedisi i sistemit te bazës se të dhënave

Hardware i referohet të gjitha pajisjeve fizike të sistemit, për shembull: kompjuterave (PC, workstation, server dhe superkompjuter), pajisjeve të magazinimit, printerave, pajisjeve të rrjetit (hub, switch, routers, fibra optikë), pajisje të tjera (Automated Teller Machines, lexues ID).

Software. Edhe pse programi kryesor mund të konsiderohet vetë DBMS, që sistemi i bazës së të dhënave të funksionojë plotësisht, nevojiten tre lloje programesh:

1. *Programi i sistemit operativ* menaxhon të gjithë komponentet hardware dhe bën të mundur që gjithë programet e tjera të funksionojnë në kompjuter.
2. *Programet e DBMS* menaxhojnë bazën e të dhënave brenda sistemit.
3. *Programet aplikativë dhe të shërbimeve* përdoren për të aksesuar dhe përpunuar të dhënat në DBMS dhe për të menaxhuar mjedisin kompjuterik ku kryen këto veprime.

Njerëzit . Ky komponent përfshin të gjithë përdoruesit e sistemit me bazë të dhënash:

1. *Administratorët e sistemit* mbikëqyrin operacionet e përgjithshme të sistemit.

2. *Administratorët e bazës së të dhënave* (DBA), menaxhojnë DBMS-në dhe sigurojnë që baza e të dhënave është duke funksionuar siç duhet.
3. *Projektuesit e bazës së të dhënave* projektojnë strukturën e bazës së të dhënave.
4. *Analistët e sistemit dhe programuesit* hartojnë dhe zhvillojnë programet aplikativë.
5. *Përdoruesit fundorë* janë njerëzit që përdorin programet aplikativë për të kryer operacionet e përditshme të organizatës.

Procedurat janë udhëzimet dhe rregullat që qeverisin projektimin dhe përdorimin e sistemit me bazë të dhënash. Ato forcojnë zbatimin e standarteve si në komunikimin brenda organizatës ashtu edhe me klientët.
Të dhënat. Fjala "të dhëna" nënkupton koleksionin e fakteve të ruajtura në bazën e të dhënave.

Gjuhët e DB dhe ndërfaqet
Në shumë DBMS ku nuk ka ndarje strikte të niveleve, përdoret një gjuhë e quajtur gjuha e përcaktimit të të dhënave (**data definition language - DDL)** e cila përdoret nga DBA dhe dizenjuesit e DB për të përcaktuar të dyja skemat.

- Një gjuhë tjetër, gjuha e përcaktimit të depozitimit (**storage definition language - SDL)** përdoret për të specifikuar skemën e brendshme.
- Gjuha e përcaktimit të pamjeve (**view definition language - VDL)** për të përcaktuar pamjet e përdoruesve dhe lidhjet e tyre me skemën konceptuale.

Pasi janë kompiluar skemat e bazës së të dhënave dhe ajo është populluar me të dhëna, përdoruesit duhet të kenë disa mënyra për ta manipuluar atë. Disa manipulime tipike janë marjet, shtimi, fshirja dhe modifikimi i të dhënave. DBMS ofron një gjuhë që është gjuha e manipulimit të të dhënave (**data manipulation language - DML)** për këtë qëllim. DML e nivelit të lartë e përdorur në mënyrë interaktive e vetme quhet **gjuha query**.
Gjuha e kontrollit të të dhënave **(data control language - DCL)** është një gjuhë që përdoret nga DBA për kontrollin edhe aksesin e përdoruesve në DB dhe mbi të dhënat e saj.

Hyrje në SQL
SQL është një gjuhë për bazat e të dhënave e dizenjuar për të menaxhuar të dhënat në sistemet e menaxhimit të bazave të të dhënave relacionale (RDBMS). Ajo është një gjuhë e standardizuar ANSI (American National Standards Institute) e cila u zhvillua fillimisht nga IBM për të përcaktuar, modifikuar dhe kërkuar bazat e të dhënave relacionale duke përdorur shprehje deklarative.

Figura 7-2. Bashkëveprimi i SQL me bazën e të dhënave

SQL është e përbërë nga komanda që i mundësojnë përdoruesve të realizojnë:

- Krijimin e një baze të dhënash.

- Krijimin e tabelave në bazën e të dhënave.
- Shtimin, modifikimin dhe fshirjen e rekordeve.
- Kërkimin e të dhënave ndërmjet relacioneve duke kënaqur kritere të ndryshme.
- Trajtimin duke kryer veprime aritmetike dhe llogjike.
- Nxjerrja e rezultateve qoftë për shtyp, qoftë për të shërbyer si vlera në një relacion të ri.

Gjuha është konceptuar që të përdoret dhe si gjuhë e pyetjeve më vete edhe si zgjerim në gjuhët e programimit të nivelit të lartë. SQL është një gjuhë e bazuar tek bashkësitë. Kjo nënkupton që SQL mund të kërkojë të dhëna në shumë rreshta nga një ose disa tabela nëpërmjet një komande të vetme. Gjithashtu SQL është një gjuhë jo proçeduriale, ajo përshkruan çfarë kërkon përdoruesi dhe është sistemi i menaxhimit të bazës së të dhënave (DBMS - Database Management System) që gjen mënyrën më të mirë për t'iu përgjigjur kërkesës së përdoruesit.
Pavarësisht se SQL është standard, disa sisteme bazash të dhënash kanë implementuar versionin e tyre të gjuhës SQL duke e zgjeruar atë. Në mësimet tona do të përdorim Microsoft SQL Server 2017 si shembull për të demostruar query-it e ndryshme SQL.
Disa nga DBMS më të njohura janë: Microsoft SQL Server me disa versione si: enterprise, developer, express i cili është pa pagesë, Oracle, MySQL, Microsoft Access, IBM DB2, Sybase, etj.

SQL është formalisht dhe faktikisht gjuha standarde për përcaktimin dhe manipulimin e bazave të të dhënave.

Funksionet e SQL përshtaten në tre kategori:

1. *Gjuha për përcaktimin e të dhënave* (DDL – Data Definition Language) e cila përfshin komanda për të krijuar objekte të bazës së të dhënave të tilla si Tabelat, Pamjet si dhe komanda për të përcaktuar integritetin e të dhënave. Ajo përmban tre komanda kryesore:
   - **CREATE** – krijon një objekt të ri në DB.
   - **DROP** – fshin një objekt ekzistues në DB.
   - **ALTER** – modifikon strukturën e një objekti ekzistues në DB.
2. *Gjuha për manipulimin e të dhënave* (DML – Data Manipulation Language) e cila përfshin komanda për të fshirë, modifikuar dhe shtuar të dhëna në tabelë si edhe komanda për marrjen e të dhënavenga tabelat. Komandat kryesore janë:
   - **INSERT INTO** – shton të dhëna (një rekord) në tabelë.
   - **UPDATE** – modifikon të dhëna në tabelë.
   - **DELETE** – fshin të dhëna (rreshta) nga tabela.
   - **SELECT** – merr të dhëna nga tabela.
3. *Gjuha për administrimin e përdoruesve të bazës së të dhënave* (DCL – Data Control Language) e cila përfshin komanda për të manipuluar dhe administruar të gjithë përdoruesit dhe rolet e tyre mbi objektet e bazës së të dhënave.

Në zemër të SQL janë pyetjet (Query). Një query është një pyetje bazuar në kërkesat e përdoruesit. Në fakt, në mjedisin e SQL, fjala query mbulon si pyetjet dhe veprimet. Pra, për një DBMS, një query është**thjesht** një deklaratë SQL e cila duhet të ekzekutohet.

Para se të shqyrtojmë sintaksën e SQL për krijimin dhe përcaktimin e tabelave dhe të elementëve të tjerë të saj, do të shqyrtojmë modelin e bazës së të dhënave mbi të cilat do të

krijohen tabelat.

Modeli i bazës së të dhënave Furnitor_Pjesë

Demostrimin e përdorimit të SQL do ta realizojmë me anë të shembujve të ndryshme në Sql Server. Do të marrin në shqyrtim një bazë të dhënash e cila në modelin relacional përbëhet nga tre tabela: Furnitor, Pjesë dhe Transporton. Tabela Furnitor ruan kodin e furnitorit, emrin e tij, statusin dhe një vendndodhje. Tabela Pjesë paraqet lloje të ndryshme pjesësh, kodin e saj, emrin, ngjyrën, peshën dhe qytetin ku ruhen ato. Tabela transporton tregon se cilat pjesë transportohen nga cili furnitor dhe me çfarë sasie. Një furnitor mund të transportojë shumë pjesë ndërsa një pjesë mund të transportohet nga disa furnitorë. Skema relacionale së bashku me çelësat kryesorë të nënvijëzuar është:

Furnitor(F_ID, F_Emri, Statusi, Qyteti).
Pjesë(P_ID, P_Emer, Ngjyra, Pesha, Qyteti).
Transporton(F_ID, P_ID, Sasia)

Krijimi i bazës së të dhënave

Për ndërtimin e këtij modeli relacional duhet të realizohen në fillim dy detyra: së pari, të krijohet struktura e bazës së të dhënave dhe së dyti, krijimi i tabelave që do të mbajnë të dhënat përfundimtare.

Kur krijohet një bazë të dhënash e re, RDBMS krijon skedarët fizikë që do të mbajnë atë. Kur krijohet një bazë të dhënash e re, RDBMS krijon automatikisht tabelat e fjalorit të të dhënave në të cilat do të ruhen metadatat dhe krijon një administrator për bazën e të dhënave. Krijimi i këtyre skedarëve do të thotë bashkëveprim me sistemin operativ dhe me sistemin e skedarëve të mbështetur në sistemin operativ.

Për të përfunduar detyrën e parë, në SQL kemi komandën ***CREATE DATABASE*** e cila pasohet nga emri unik i bazës së të dhënave, për shembull:

CREATE DATABASE *Furnitor_Pjese;*

Një bazë të dhënash mund të ndërtohet dhe nëpërmjet ndërfaqes grafike.

Ndërfaqja e SQL Server

Microsoft është krijuesi i SQL Server. Ai ka disa edicione ku SQL Server Express mund të shkarkohet dhe përdoret pa pagesë. SQL Server përbëhet nga Database Engine i cili nuk ka ndërfaqe grafike por është thjesht një shërbim i cili ekzekutohet në background dhe nga Management Studio i cili është një mjet grafik për të konfiguruar dhe parë informacionet në bazën e të dhënave.

SQL Server Management Studio është një mjet GUI i përfshirë në SQL Server për konfigurimin, menaxhimin dhe administrimin e të gjithë komponentëve të tij. Ky mjet përfshin si editorët e skriptimin ashtu edhe ato grafikë që punojnë me objekte dhe tipare të server-it.

Figura në vijim tregon pjesët kryesore përbërëse të Management Studio. Object Explorer lejon të shfletoni, selektoni dhe të veproni me çdo objekt në server. Gjithashtu mund të krijojmë baza të dhënash të reja, të modifikojmë ato ekzistuese duke shtuar apo modifikuar tabela. Ai ka edhe dritaren e query-it në të cilën i shkruajmë dhe i ekzekutojmë ato.

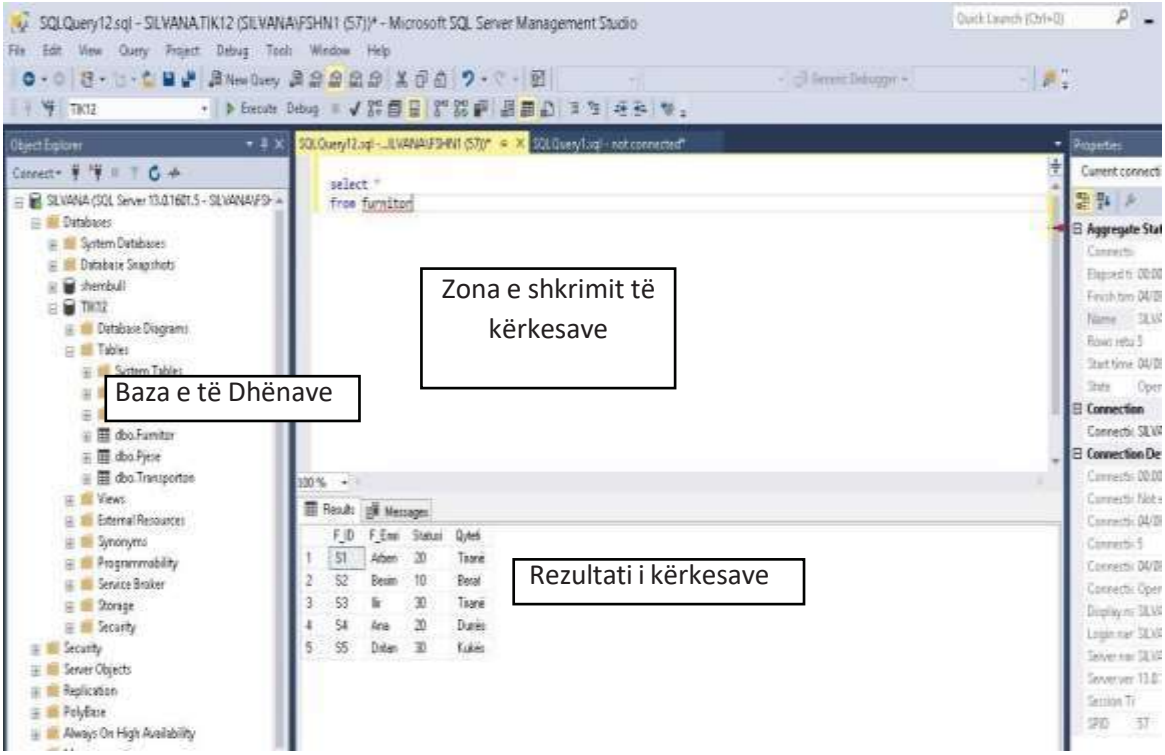


Figura 7-3 Ndërfaqja e SQL Server

Për të shkruajtur komanda dhe query SQL do të përdorim "Query Editor" i cili merret duke klikuar "New Query" në shufrën e mjeteve (Toolbar). Me SQL dhe "Query Editor" mund të bëjmë pothuajse gjithçka me kod por ndonjëherë mund të përdorim mjetet e ndryshme dizenjuese për të bërë punën pa shumë kod. Për të ekzekutuar kodin klikojmë butonin Execute dhe shohim dhe rezultatin e ekzekutimit të query-it.

Krijimi i një baze të dhënash të re

Klikojmë me të djathtën në nyjen "Databases" dhe zgjedhim "New Database...".

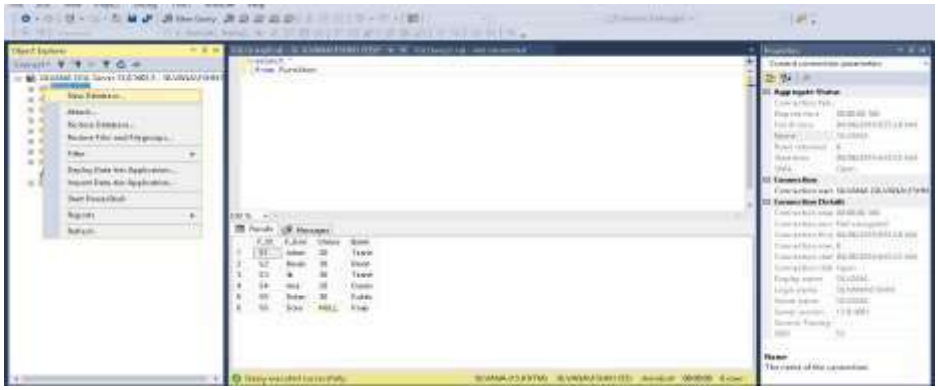


Figura 7-4 Krijimi i një baze të dhënash

I vetmi informacion i nevojshëm është emri i bazës së të dhënave, për parametrat e tjerë ka vlera default (standarde).

Krijimi i tabelave

Për të krijuar një tabelë do të përdorim fjalët kyçe CREATE TABLE ku duhet të

përcaktojmë: emrin e tabelës, emrin e fushave dhe tipin e të dhënave të tyre, shtrëngimet (kufizimet) që duam.
Emrat e tabelave dhe fushave duhet të jenë sa më përshkruese, emrat e fushave unike brenda një tabele, ndërsa emrat e tabelave unike brenda një baze të dhënash. Tabela kur krijohet është pa të dhëna. Sintaksa e krijimit të saj është:

```
CREATE TABLE emër_tabelë
(Fusha1 tip_të_dhënash [shtrëngim1],
Fusha2 tip_të_dhënash [shtrengim2],
...
FushaN tip_të_dhënash [shtrëngimN],
[, shtrëngim_tabele1]
...
[, shtrëngim_tabeleM]
);
```

Tip_të_dhënash përcakton se çfarë të dhënash do të ruhen në kolonën përkatëse. Tipet e të dhënavenë SQL Server janë të organizuar në kategori ku më kryesorët janë:
numrat ekzakt: bigint, int, smallint, tinyint, money,
numeric, bit, decimal, etj,numrat e përafërt: float, real,
data dhe koha: date, datetime, time, etj,stringjet: char,
varchar, text, stringjet unikode: nchar, nvarchar, ntext ,etj.
Disa nga tipet e të dhënave më të përdorura janë:
int - Ruajnë numra të plotë,
Float - Ruajnë numra me presje,
Char(n) - Ruan karaktere me gjatësi fikse. Kujtesa zihet sa është numri n, pavarësisht se sa është madhësia e fjalës.
VARCHAR(N) - Ruan karaktere me gjatësi të ndryshueshme, pra kujtesa zihet për aq karaktere sa është fjala që ruhet. Madhësi maksimale 4000 bytes.
Date - Ruan të dhënat për datën: ditë, muaj, vit.

Krijimi i tabelës furnitor:

```
CREATE TABLE Furnitor
( F_ID    varchar(5),
  F_Emri  varchar(15),
  Statusi int,
  Qyteti  varchar(10)
)
```

Tabela mund të ndërtohet edhe grafikisht duke klikuar me të djathtën mbi **Tables** dhe zgjedhim **New** pastaj **Table**. Në dritaren që del plotësojmë emrat e fushave tek Column Name, tipet e të dhënave tek Data Type dhe klikojmë nëse duam të lejojë vlera Null fusha përkatëseose jo, për shembull tabela furnitor.

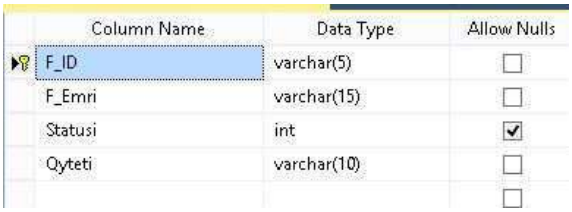


Figura 7-5 Krijimi i tabelës grafikisht në Sql Server

Popullimi i tabelave

Mbushja me të dhëna e tabelave mund të bëhet me anë të komandës Insert të SQL ose grafikisht si në figurë:

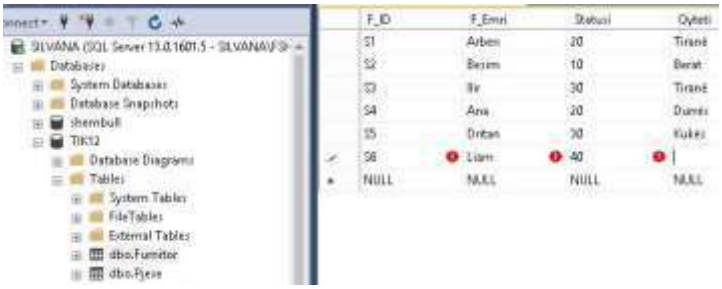


Figura 7-6 Shtimi i të dhënave grafikisht

Tema 8. Ndërtimi i modelit ER për konceptimin e bazës së të dhënave

Ndërtimi i skemës llogjike të bazës së të dhënave

Modeli "entitet-lidhje" është një teknikë grafike e përdorur gjerësisht për kuptimin dhe organizimin e të dhënave duke specifikuar strukturën llogjike të përgjithshme të një databaze. Le të trajtojmë zhvillimin e një modeli të tillë nëpërmjet një shembulli praktik. Supozojmë se kërkohet dizenjimi i një modeli të tillë për zhvillimin e një databaze të një universiteti. Universiteti ka organizimin e mëposhtëm:

- Universiteti përmban disa departamente;
- Çdo departament ofron disa kurse studimi;
- Në një department punojnë disa pedagogë;
- Një pedagog punon vetëm në një department;
- Çdo department ka një përgjegjës;
- Çdo pedagog mund të drejtojë vetëm një department;
- Çdo pedagog jep disa kurse;
- Një kurs jepet vetëm prej një pedagogu;
- Një student mund të regjistrohet në disa kurse;
- Çdo kurs studimi merret nga disa studentë

Hapi i parë: Identifikimi i entiteteve. Entitetet që nevojiten në rastin e shembullit janë Departamenti, Kursi, Pedagogu dhe Studenti.

Hapi i dytë. Identifikimi i lidhjeve. Meqë një department ofron shumë kurse, ndërsa një kurs mund të ofrohet vetëm prej një departamenti, atëherë kardinaliteti midis departamentit

dhe kursit të studimit është i tipit "një me shumë" (1:N). Ngjashmërisht, meqë një department ka shumë pedagogë dhe një pedagog bën pjesë vetëm në një department, kardinaliteti department-instruktor është i tipit "një me shumë". Meqë një department mund të ketë vetëm një përgjegjës dhe një përgjegjës mund të jetë përgjegjës vetëm në një department, atëherë kardinaliteti përgjegjës-departament është "një me një" (1:1). Meqë në një kurs mund të regjistrohen shumë student dhe një student mund të ndjeki disa kurse studimi, kardinaliteti student-kurs është i tipit "shumë me shumë" (M:N). Së fundi, meqë një pedagog mëson disa kurse dhe një kurs jepet vetëm prej një pedagogu, kardinaliteti kurs-pedagog është "shumë me një" (N:1).

Hapi i tretë: Identifikimi i atributeve çelës. "Emër_Departamenti" mund të shërbejë si atribut çelës për të identifikuar në mënyrë unike departamentin. Ngjashmërisht, "Kursi_ID", "Pedagogu_ID" dhe "Studenti_ID" mund të shërbejnë si atribute çelës përkatësisht për entitetet Kursi, Pedagogu dhe Studenti.

Hapi i katërt: Identifikimi i atributeve të tjera. Në lidhje me shembullin tonë supozojmë se janë identifikuar atribute të tjera si vijon:

Entiteti Departamenti – Adresa;
Entiteti Kursi – Emri_Kursi, Kohëzgjatja;
Entiteti Pedagogu – Emri, Mbiemri, Telefoni;
Entiteti Studenti – Emri, Mbiemri, Telefoni;

Hapi i pestë: Skicimi i diagramës [1]së plotë "lidhje-entitet". Diagrama e shembullit në fjalë jepet nëpërmjet skicës së figurës 8-1, ku një figure ovale përdoret për atributet, një figure drejtkëndëshe përdoret për një entitet, dhe një figure në formë rombi përdoret për të identifikuar lidhjen llogjike midis dy entiteteve. Në të dyja anët e rombit përcaktohet tipi i kardinalitetit me të cilën lidhen entitetet përkatëse.

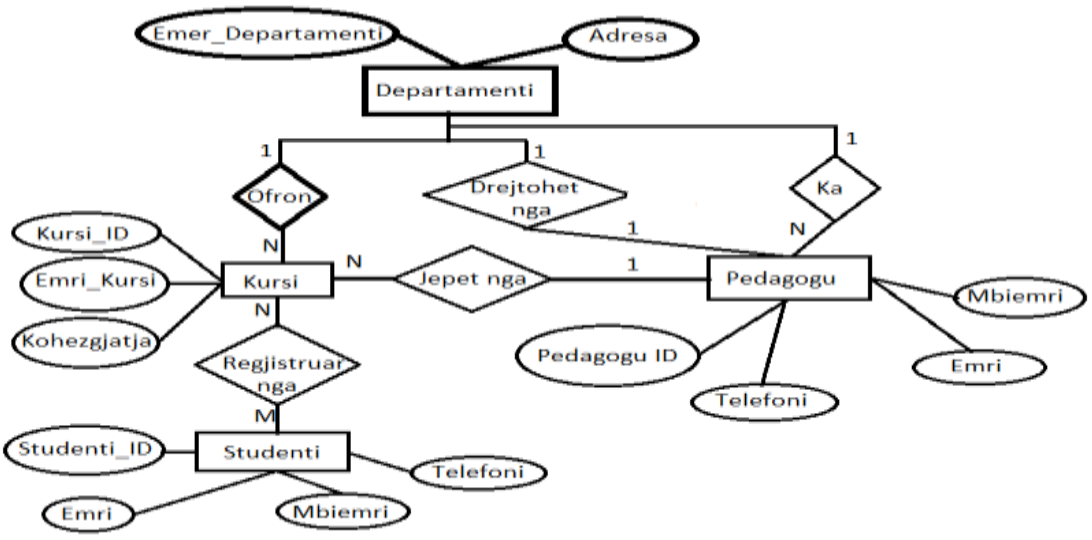


Figura 8-1 Diagrama ER "Universiteti"

Hapi i gjashtë: Meqë databazat relacionale nuk i suportojnë lidhjet "shumë me shumë", është e nevojshme që të bëhet zbërthimi i tyre në lidhje të tipit "një me shumë" duke futur një entitet të ri shoqërues ndërmjet dy entiteteve ekzistues. Në shembullin e mësipërm, për

[1] Ndërtimi dhe skicimi i digramës ER realizohet në MS Visio ose www.draw.io

lidhjen M:N kursi-studenti, përcaktojmë një entitet shoqërues të quajtur Regjistrimi me lidhjet 1:N student-regjistrimi dhe kursi –regjistrimi. Çelësat primarë të dy entiteteve ekzistues duhet të jenë atribute të entitetit shoqërues.

Së fundi, theksojmë se hapave të mësipërm i shtohet dhe përcaktimi i llojeve të të dhënave dhe normalizimi të cilat do të shqyrtohen në tema të veçanta. Një model llogjik [2]i plotë i shembullit të mësipërm jepet në figurën 8-9.

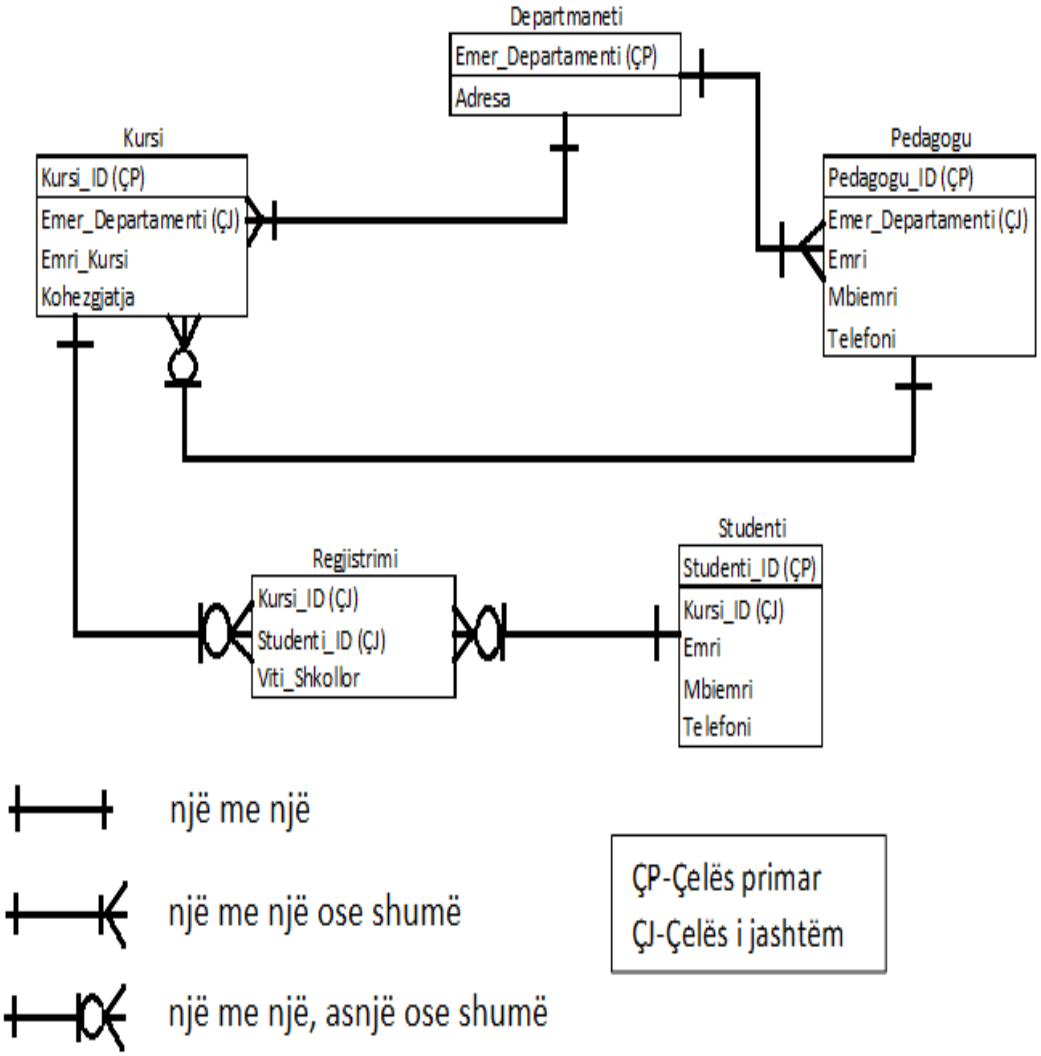
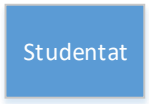


Figura 8-2 Modeli logjik

Para se të filloni me krijimin e diagramit shikoni tabelën e simboleve të cilat përdoren për Diagramën ER

[2] Ndërtimi dhe skicimi i digramës Relacionale dhe logjike realizohet në MS Visio ose www.draw.io

Tabela 8.1 Simbolet për dizenjimin e Diagramës ER

Simboli	Përshkrimi i funksionalitetit
	Paraqet Entitetin në ER Diagram, kurse në SQL paraqet tabelën
	Paraqet Atributin, kurse në SQL kolonën apo fushën
	Paraqet lidhjen në mes të Entiteteve. Lidhja mund të jetë: 1:1 (Një me Një) 1:M apo M:1 (Një me Shumë apo Shumë me Një) M:M (Shumë me Shumë)
	Paraqet mënyrën Opsionale për krijimin/regjistrimin e të dhënave në bazën e të dhënave
 ose	Paraqet mënyrën Obligative për krijimin/regjistrimin e të dhënave në bazën e të dhënavë
	Paraqet lidhjen në mes të entitetit dhe atributit *Primary Key* si dhe atributeve të tjera
	Paraqet lidhjen në mes të entitetit dhe atributit *Foreign Key*.

Për të krijuar një Diagram ER duhet të ndjekim hapat dhe instruksionet e mëposhtme:

1. Pasi hapet Visio duhet të klikoni në “Blank Drawing” sic është treguar në figurën 8-3:

Figura 8-3

2. Pasi hapet ambjenti punës, duhet të zgjidhet “US Units” dhe të klikohet në butonin “Create”. Dhe do t`iu paraqitet figura në vijim:

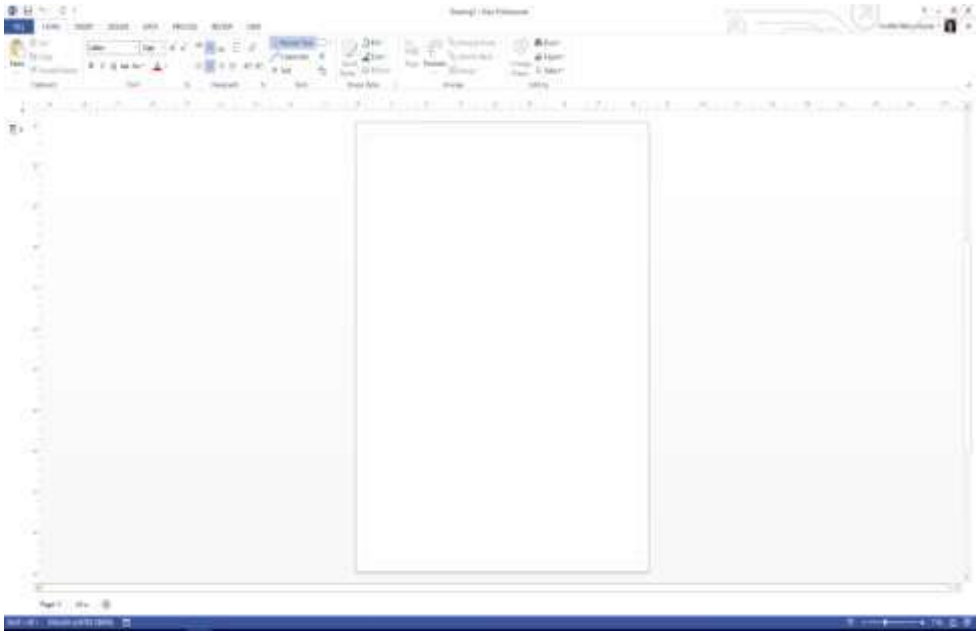


Figura 8 -4

3. Për të paraqitur format bazë për krijimin e Diagramit ER duhet të klikoni në ikonën përkatëse në anën e majtë-lartë të dritares, sic tregohet në figurën 8-5:

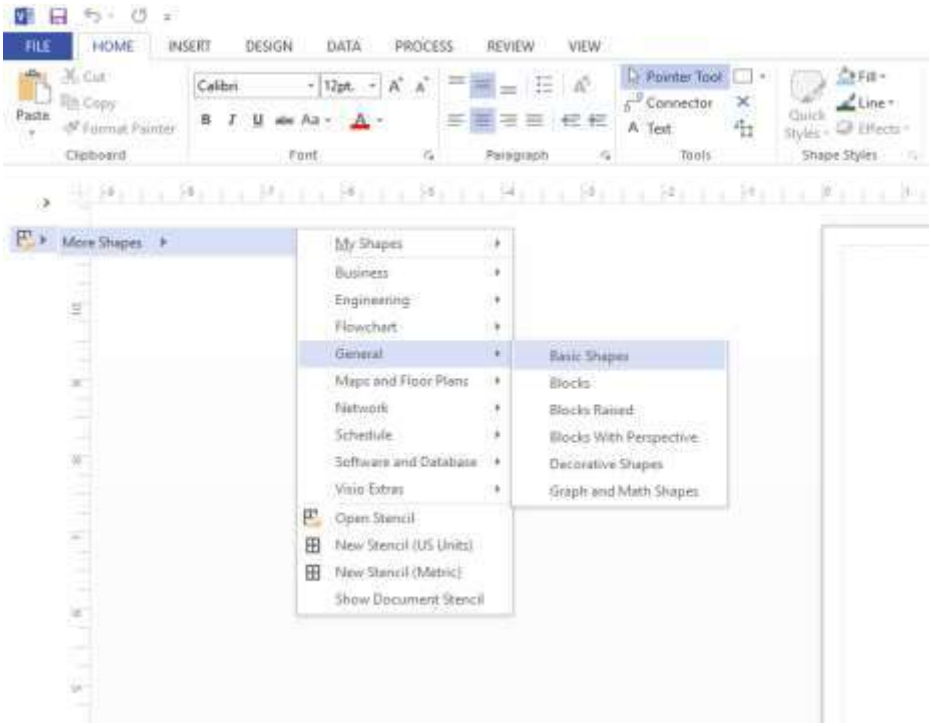


Figura 8-5

4. Pasi të klikoni në Basic Shapes do t`iu paraqiten format për krijimin e ER Diagramit në anën e majtë të dritares. Zgjidhni format/figurat gjeometrike për ER Diagramin tuaj duke klikuar mbi formën e përshtatshme (P.sh. Katror apo diamont) me ndihmën e tastit të majtë të mausit dhe duke e mbajtur tastin e majtë të shtypur lëvizni në mes

të formës për ta vendosur formën e zgjedhur duke e lëshuar mausin. Për të vendosur tekst brenda figurës duhet të klikoni dy herë mbi figurë dhe të filloni të shkruani tekstin përkatës psh:student për entitetin e caktuar, pasi në këtë rast në figurë kemi zgjedhur formën Katrore.

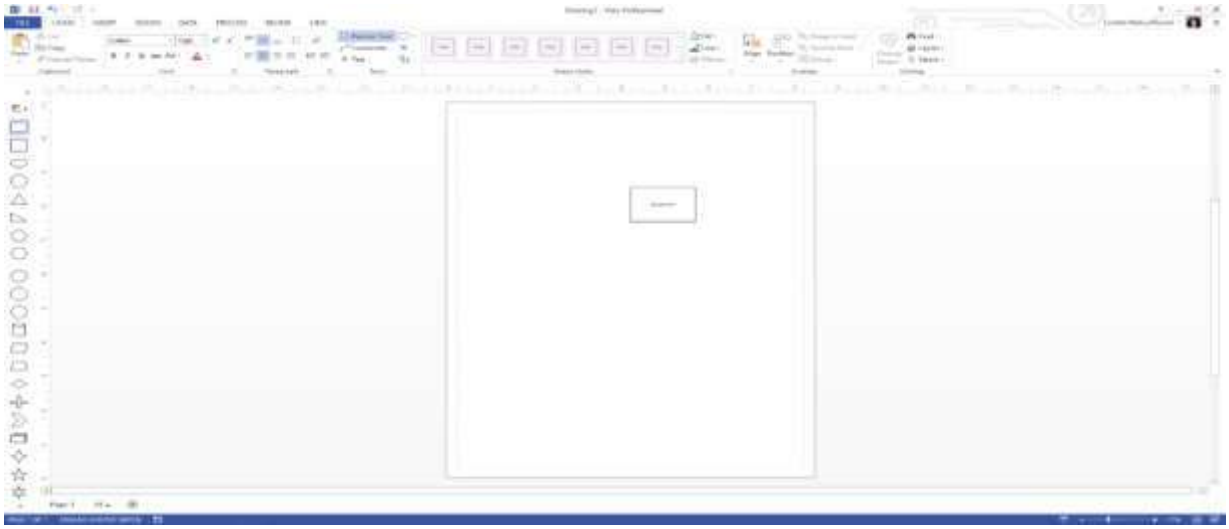


Figura 8-6

Për të lidhur dy objekte në ER Diagram përdoret konektori, i cili gjendet në toolbar në mes të figurës me emrin “Connector”. Klikohet në këtë konektor dhe zgjidhet objekti i parë i cili do të lidhet me objektin e dytë duke e mbajtur tastin e djathtë të mausit të shtypur derisa të përfundoj lidhja mes dy objekteve, si në figurën 8-7:

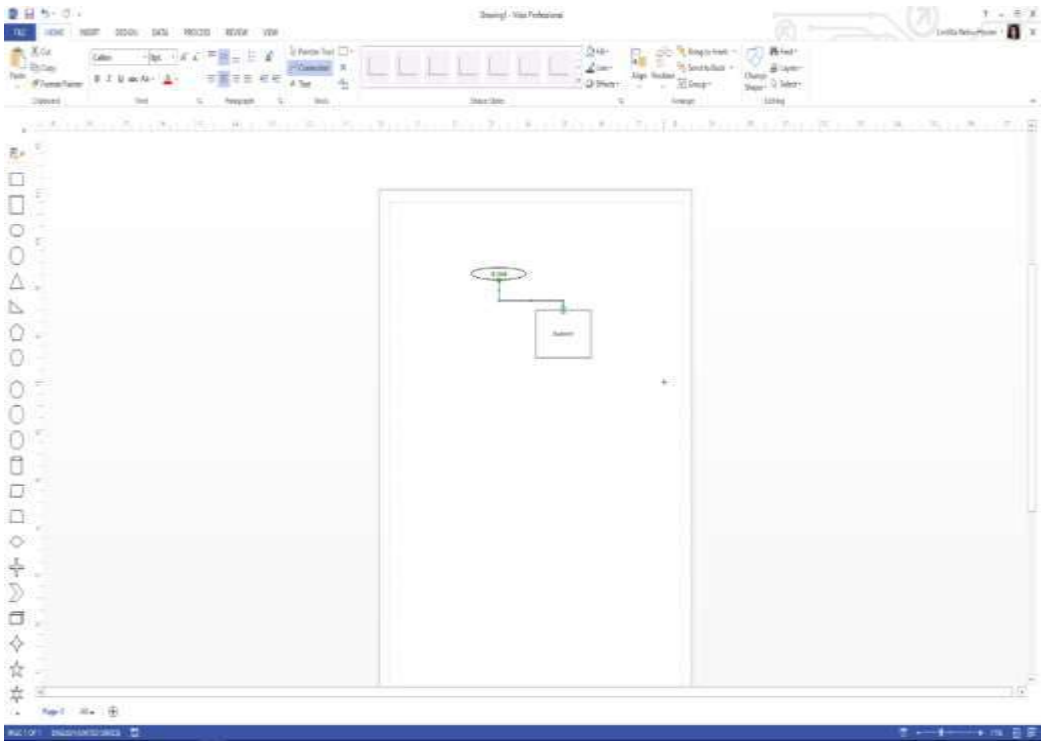


Figura 8-7

VINI RE:

Në rast se konektori paraqitet me shigjetë ju mund ta ndërroni/ndryshoni atë duke klikuar në Line->Arrows më pas zgjidhni vijën e nevojshme/përshtatshme e cila është e para në listë. Shih figurën ne vijim.

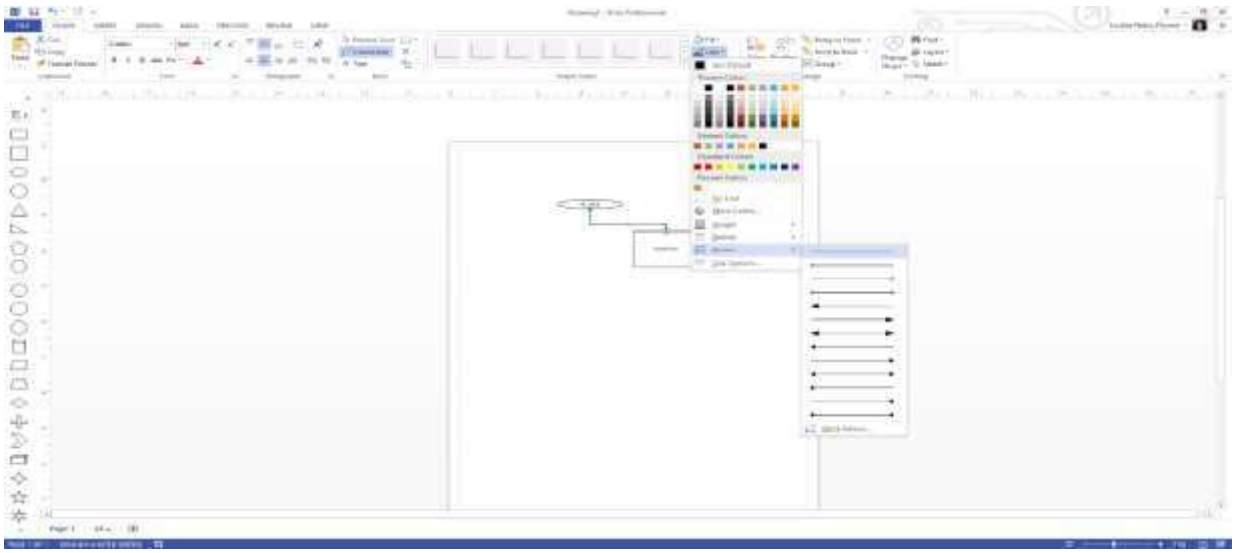


Figura 8-8

Për të Paraqitur Primary Key në atribut (fushë) duhet të klikohet “Underline” sikurse në MS word, ndërsa për ta paraqitur Foreign Key në atribut (fushë) atëherë duhet të zgjidhet Line->Dashes, vija e tretë në listë.

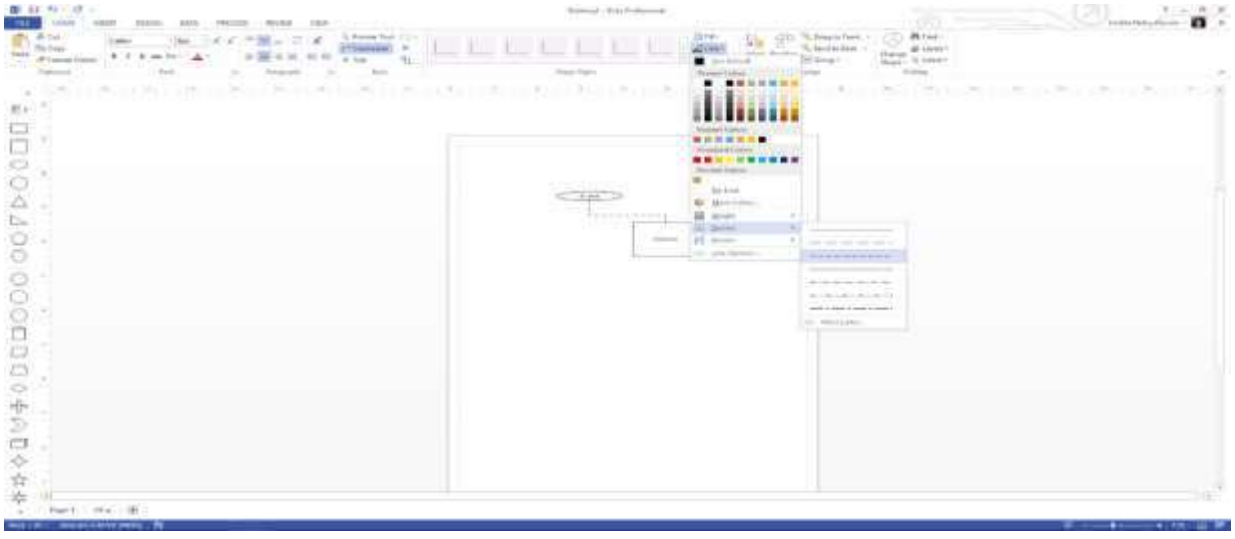
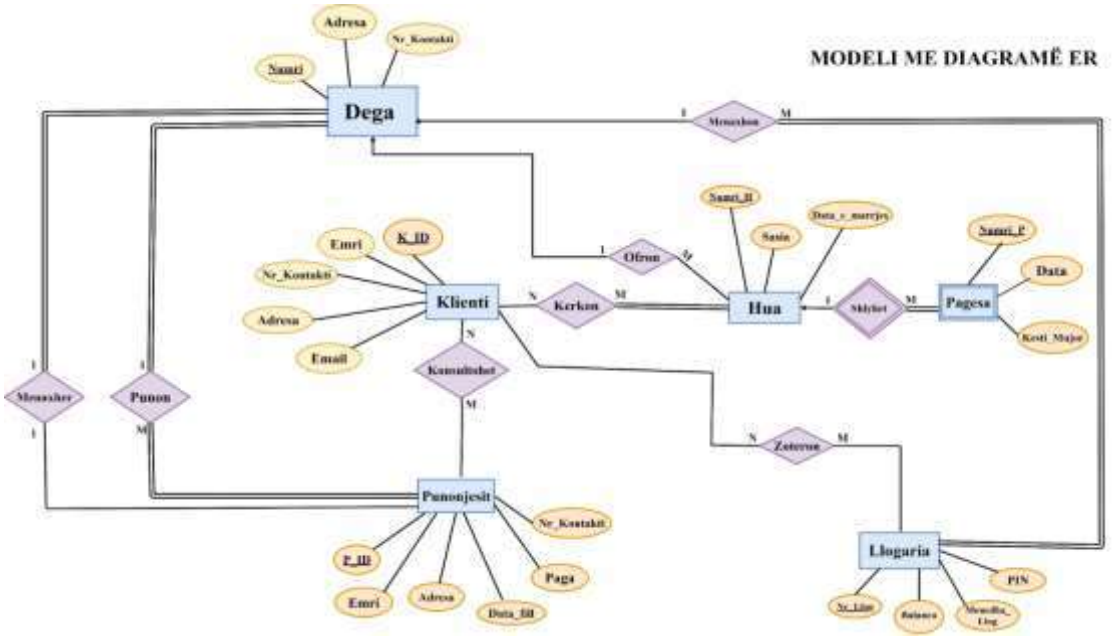
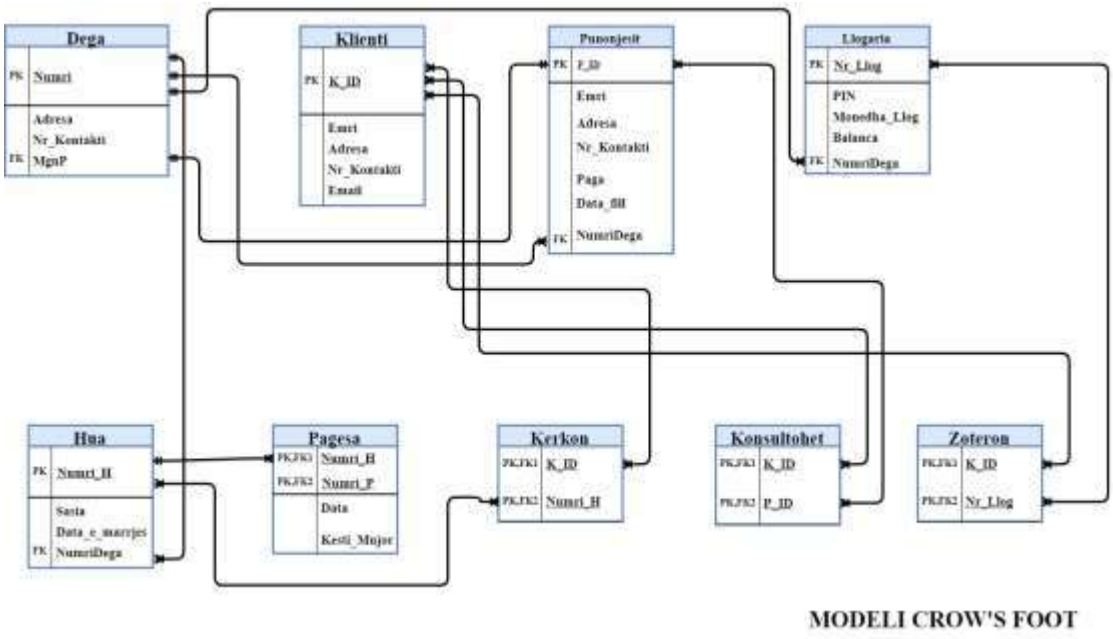


Figura 8-9

Shembull: Një bankë kërkon të ndërtojë një Databazë ku të ruajë të dhëna për degët, klientët, punonjësit, huat që ata marrin, pagesat e tyre, si dhe llogaritë e hapura.
1 - Cdo degë ka *një numër unik, adresë dhe numër kontakti.*
2 - Për klientin ruhet: *Id e tij, emri, numër kontakti, adresa dhe email-i.*

3 - Gjithashtu banka i ofron klientit hapjen e një llogarie. Për llogarinë ruhet: *një numër unik, PIN-I, balanca dhe monedha e llogarisë.*
4 - Cdo degë ka disa punonjës. Një punonjës mund të jetë menaxher i degës ose pjesë e stafit. Për punonjësit kërkohet: *Id, emri, adresa, numri i kontaktit, data e fillimit si punonjës dhe paga.*
5 - Klienti ka të drejtë të marrë hua një sasi të caktuar lekësh të cilën duhet ta shlyejë. Për huan ruhet: *një numër unik, sasia dhe data e marrjes.*
6 - Banka jep te drejtën që klienti të ketë mundësi ta paguajë huan dhe me këste në muaj. Përvec *kësitit mujor* te pagesa ruhet edhe *një numër unik që e identifikon si dhe data.*





Tema 9. Dizenjimi i bazës së të dhënave relacionale

Çfarë është një bazë e të dhënave relacionale? Në një bazë të të dhënave relacionale të dhënat janë grupuar në tabela të ndryshme mbështetur në lidhje logjike. Për shembull: të gjitha adresat e studentëve dhe informacioni i kontakteve është grupuar në një tabelë ndërsa informacioni mbi banesën dhe ushqimin në një tabelë tjetër. Në bazën e të dhënave relacionale, një lidhje ndërmjet dy tabelave vendoset nëpërmjet fushave të përbashkëta. Në Figurën 9-1, "StudentID" është fusha e përbashkët ndërmjet tabelës "KonviktiCaktuar" dhe "InformacioniStudent" dhe "KonviktID" është fusha e përbashkët ndermjet tabelave "Konviktet" dhe "KonviktiCaktuar". Fusha e përbashkët në një tabelë është çelësi primar i kësaj tabele dhe fusha e përbashkët në tabelën e lidhur me të quhet çelës i jashtëm (foreign key)

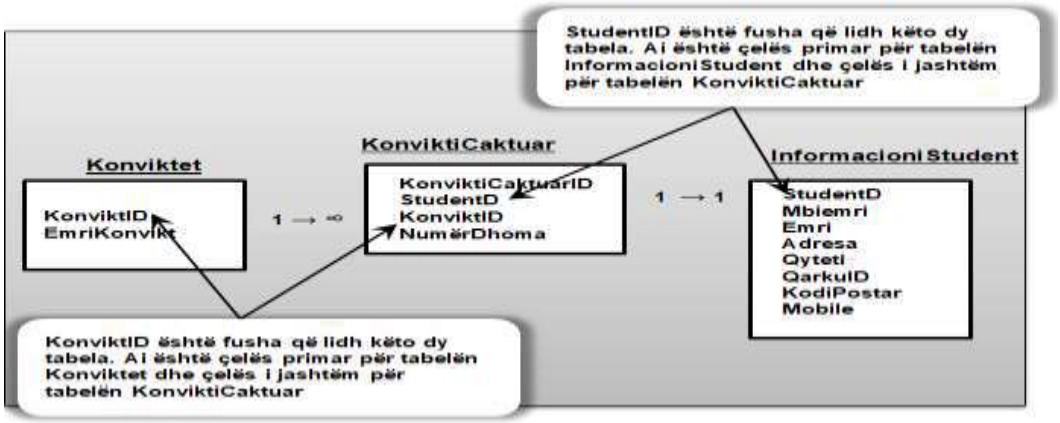


Figura 9-1 Çelësi i jashtëm është i njëjtë me çelësin primar në tabelat e lidhuara

Duke qenë se të dhënat në një bazë të dhënash janë në ndryshim të vazhdueshëm, është e rëndësishme që të sigurohet që të dhënat në tabelat e lidhura të jenë të sinkronizuara dhe që një regjistrim nuk mund të shtohet apo të modifikohet në një tabelë dhe pa qenë në tabelën tjetër. Për të shmangur regjistrime jetime dhe për të mbajtur të dhënat e lidhura ekziston mundësia e përdorimit të integritetit referencial për çdo lidhje. Me *integritet referencial* kuptohet që për çdo vlerë në tabelën me çelës të jashtëm ekziston një vlerë korresponduese në tabelën me çelës primar. Për shembull, nuk mund të futet një regjistrim studenti tek tabela "KonviktiCaktuar" me një StudentID që nuk ekziston në tabelën InformacioniStudent" (tabelë me çelës primar) nëse është zbatuar integriteti referencial. Do të duhej së pari të shtohej studenti në tabelën "InformacioniStudent" dhe pastaj të futet studenti në tabelën "KonviktiCaktuar".

Në bazat e të dhënave relacionale a ka tipe të ndryshme lidhjeje? Një lidhje në një bazë të dhënash relacionale mund të jetë në një nga tre format e mëposhtme:

1. Lidhja **një-me-shumë** karakterizohet nga fakti që një regjistrim që shfaqet vetëm një herë në një tabele ka mundësinë të shfaqet shumë herë në një tabelë të lidhur. Për shembull, siç tregohet në Figurën 9-1, numri 1 pas fushës KonviktID në tabelën "Konviktet" tregon që ka vetëm një rast për çdo KonviktID në tabelën "Konviket", dhe simboli ∞ (infinit) pas fushës KonviktID në tabelën "KonviktiCaktuar" tregon që shumë regjistrime mund të kenë të njëjtin KonviktID në tabelën "KonviktiCaktuar". Pra për secilin konvikt ka vetëm një regjistrim, por ai konvikt mund të shfaqet shumë

herë për çdo student që banon në të. Lidhja një-me-shumë është shumë e përhapur në praktikën e bazave të të dhënave relacionale.

2. Lidhja **një-me-një** (one-to-one) tregon që për çdo regjistrim në një tabele ka një dhe vetëm një regjistrim korrespondues në një tabelë të lidhur. Për shembull, çdo studenti i është caktuar vetëm një dhomë. Duke qenë se një student mund të banojë në vetëm një konvikt në të njëjtën kohë, atëherë caktohet një lidhje një-me-një, siç tregohet në figurën 9-1 me shifrën 1 të vendosur pas fushës StudentID në tabelën "KonviktiCaktuar" dhe në tabelën "InformacioniStudent".
3. Lidhja **shumë-me-shumë** karakterizohet nga fakti që regjistrimet në një tabelë lidhen me shumë regjistrime në një tabelë tjetër dhe anasjelltas. Për shembull një tabelë e studentëve mund të lidhet me një tabelë të punëdhënësve. Punëdhënësit mund të punësojnë shumë studentë dhe studentët mund të punojnë për më shumë se me një punëdhënës.

Bazat e të dhënave relacionale janë mjaft të mira për të dhëna që mund të vendosen në tabela dhe që mund të organizohen në fusha dhe regjistrime. Por të dhëna të tilla si imazhet, audiot dhe videot mund të menaxhohen më mirë ne bazë të dhënash të orientuara objekt.

Tema 10. Varësitë funksionale

Relacionet e mirë - strukturuara

Një relacion që përmban tepri minimale të të dhënave dhe lejon përdoruesit të shtojnë, të fshijnë dhetë përditësojnë rreshtat pa shkaktuar paqëndrueshmëri të të dhënave.

Qëllimi është të shmangim (minimizojmë) anomalitë.

- Anomalia e shtimit: shtimi i rreshtave të rinj e detyron përdoruesin të krijojë dublikim të të dhënave.
- Anomalia e fshirjes: fshirja e një rreshti mund të shkaktojë humbjen e të dhënave të tjera qëpërfaqësojnë fakte krejtësisht të ndryshme.
- Anomalia e modifikimit: ndryshimi i të dhënave në një rresht detyron ndryshime edhe në rreshtatë tjerë për shkak të dublikimit.

Si rregull i përgjithshëm: një tabelë nuk duhet të mbajë, ruajë të dhëna më shumë se të një lloj entiteti.

Koncepti i varësisë funksionale

*Përkufizim: **Varësi funksionale***:

Le të jetë dhënë një relacion R(X,Y,Z) (ku X,Y dhe Z janë bashkësi përbërësish. Z mund të jetë boshe), do të themi që ekziston një varësi funksionale ndërmjet X dhe Y e shënuar X → Y, atëherë dhe vetëm atëherë kur cilado qoftë vlera e (X,Y,Z) dhe (X,Y',Z') kemi:

//R(X,Y,Z)// dhe //R(X,Y',Z')//⇒Y=Y'

Më thjesht themi se:

Varësia Funksionale: vlera e një atributi përcakton vlerën e një atributi tjetër. A→B lexohet "Atributi B varet në mënyrë funksionale nga A".

A→B do të thotë se nëse dy rreshta kanë të njëjtën vlerë për A, ata patjetër kanë të njëjtën vlerë edhe për B.

A→B do të thotë se nëse dy rreshta kanë të njëjtën vlerë për A, ata patjetër kanë të njëjtën vlerë edhe për B.

Varësitë funksionale (VF) përcaktohen nga semantika. Nuk mund të themi që një VF ekziston vetëm duke shikuar të dhënat por mund të themi nëse ajo nuk ekziston duke shikuar të dhënat.

Veti të varësisë funksionale

Varësitë funksionale i përgjigjen disa rregullave:

1. **Refleksiviteti**: $Y \subseteq X \Rightarrow X \rightarrow Y$,rregulli që një bashkësi përcakton vetveten ose një pjesë të saj.
2. **Shtimi**. Në qoftë se $X \rightarrow Y$ atëherë $XZ \rightarrow YZ$; d.m.th. në qoftë se X përcakton Y të dy përbërësit mund të pasurohen me një të tretë.
3. **Vetia e kalimit (tranzitiviteti)** : në qoftë se $X \rightarrow Y$ dhe $Y \rightarrow Z$ atëherë $X \rightarrow Z$.

Tre rregullat e mësipërme formojnë aksiomat e varësisë funksionale dhe njihen me emrin "Aksiomat e Armstrongut".

Tabela 10.1 Konceptet e varësisë funksionale

Konceptet e varësisë funksionale	
Koncepti	**Përkufizimi**
Varësi funksionale	Atributi B ka varësi të plotë funksionale nga atributi A në qoftë se çdo vlerë e A përcakton një dhe vetëm një vlerë të B. Shembull: PRO_ID → PRO_EM ("PRO_ID përcakton funksionalisht PRO_EM"). Në këtë rast, PRO_ID njihet si atributi "përcaktues", dhe PRO_EM njihet si atributi "i varur".
Varësi funksionale (përkufizim i përgjithësuar)	Atributi A përcakton atributin B (B funksionalisht i varur nga A) nëse të gjitha rreshtat në tabelë që kanë vlerë të njëjtë të atributit A do kenë vlerë të njëjtë edhe për atributin B.
Varësi e plotë funksionale (çelës i përbërë)	Nëse atributi B është funksionalisht i varur nga një çelës i përbërë (A), por jo nga ndonjë prej nëngrupeve të tij, atëhere atributi B ka varësi të plotë funksionale nga (A).

Është e rëndësishme për të kuptuar këto koncepte për shkak se ato përdoren për të nxjerrë varësitë funksionale për një lidhje të caktuar. Procesi i normalizimit punon një lidhje në një kohë, duke identifikuar varësitë në atë lidhje dhe duke e normalizuar atë. Normalizimi fillon me identifikimin e varësive në një lidhje të caktuar dhe në mënyrë progresive duke e ndarë këtë lidhje (tabelë) në një grup lidhjesh të reja (tabela) bazuar në varësitë e identifikuara.

Tema 11. Format normale

Normalizim është proçesi i identifikimit të pozicionit më të mirë ku çdo e dhënë bën pjesë. Ajo përbëhet nga një bashkësi rregullash dhe teknikash zbatimi i të cilave minimizon anomalitë dhe tepritë e të dhënave dhe optimizon strukturën e të dhënave prej vendosjes në mënyrë sistematike dhe siç duhet të elementëve të të dhënave në grupime të përshtatshme.

Normalizimi u krijua prej Codd-it në fillimet e viteve 1970, i njohur si zhvilluesi i modelit relacional. Ashtu si modeli relacional i të dhënave, normalizimi mbështetet tek principet matematike të teorisë së bashkësive. Ai është një proçes llogjik i organizimit të kollonave (atributeve) dhe tabelave (lidhjeve) në një databazë relacionale me qëllim reduktimin e paqartësive dhe përmirësimin e integritetit të të dhënave. Një model i normalizuar i të dhënave garanton që çdo entitet është i mirë formuar dhe çdo atribut i është caktuar siç duhet entiteteve përkatëse. Normalizimi përbëhet prej disa formave normale. Codd ishte i pari që në

vitet 1971 dhe 1972 përshkroi tre format e para normale, të cilat u pasuan më vonë nga forma normale shtesë të përkufizuara nga të tjerë.

Hapat e normalizimit

Në figurë janë treguar hapat bazë në procesin e normalizimit. Së pari janë përcaktuar pamjet e përdoruesve. Pas kësaj çdo pamje e përdoruesit është konvertuar në një relacion të panormalizuar. Pasi hiqen grupet përsëritës nga relacioni i panormalizuar, atëhere relacioni është në formën e parë normale (1NF). Pas kësaj hiqen varësitë e pjesshme dhe relacionet kalojnë në 2NF. Në fund hiqen varësitë tranzitive dhe relacionet kalojnë në 3NF. Procesi i normalizimit është i lidhur ngushtë me përkufizimet e varësive funksionale dhe realizohet nëpërmjet procesit të dekompozimit të vazhdueshëm. Më poshtë do të shpjegojmë më në detaje hapat e normalizimit te relacioneve deri në 3NF.

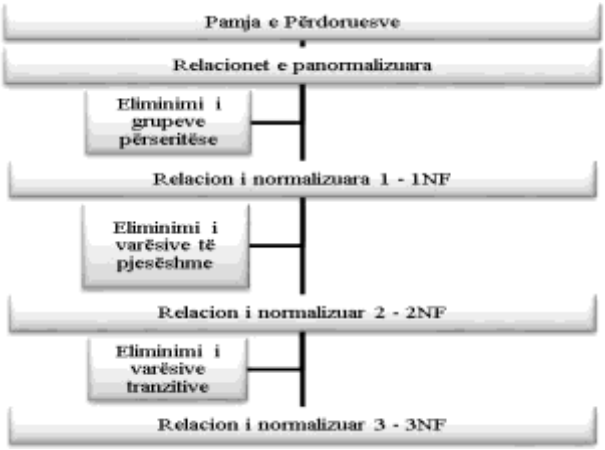


Figura 11-1 Hapat e normalizimit

Tema 12. Normalizimi relacioneve

Forma normale e parë (FN1). Objektivi i FN1 është eleminimi i të dhënave joatomike dhe grupeve të përsëritura nga një entitet. Kur arrihet një FN1, çdo atribut i një entiteti është një fakt diskret i vetëm, pra atomik. Në një databazë relacionale, një rresht është në FN1 atëherë dhe vetëm atëherë kur të gjithë fushat e saj përmbajnë vetëm vlera atomike (Nuk përmban komponentë më të vegjël të kuptimtë). Për të normalizuar një model të dhënash në FN1, eliminohen grupet me përsëritje vlerash në entitetet e veçanta. Pra, nuk përdoren atribute shumëfishe në një entitet të vetëm për të ruajtur të dhëna të ngjashme.

Shqyrtojmë shembullin e tabelës 12-1 që paraqet të dhënat e entitetit STUDENT nga një databazë e një universiteti. Siç shihet, të dhënat përmbajnë disa shkelje të FN1. Së pari, identifikohen kurse që janë në grupe me përsëritje për studentët. Informacioni i kursit duhet të zhvendoset në entitete të ndara. Për më tepër, nevojitet të specifikohen identifikuesit për të dy entitetet. Identifikuesi është çelësi primar për entitetin. Duhet patur kujdes në zgjedhjen e çelësit primar të përshtatshëm të çdo entiteti. Kështu, fillimisht mund të mendohet të zgjidhet Kursi_Num për entitetin KURSI. Por në këtë rast, nevojitet më shumë informacion për të identifikuar një kurs. Data e përfundimit të kursit aplikohet në një kombinim të STUDENT-it dhe KURS-it – një kurs nuk mund të përfundohet po të mos jetë regjistruar studenti dhe ta ketë ndjekur atë.

Një shkelje e dytë e FN1 janë ekzistenca e të dhënave joatomike në atributin STUDENT_EMER. Emri i studentit mund të ndahet në pjesë si: emri, babai, mbiemri. Pra,

nuk është e pandashme dhe prandaj thyen rregullën e të qenurit FN1. Rezultati fundor i të qenurit FN1 përshkruhet në tabelën 12-2 dhe tabelën 12-3.

Tabela 12-1 Të dhënat e panormalizuara të studentit

Student ID	Student Emër	Dega ID	Student Dega	Kursi NUM	Kursi Emër	Datë Përfundim Kursi
2907	Arben R Tiku	MAT	Matematike	MAT0011 MAT0027 EGL0010	Matematike 1 Matematike 2 Anglisht	01-08-2016 30-04-2016 30-12-2015
4019	Gerald S Leka	FIZ	Filozofi	FIZ0010 CS00100	Filozofi Programim	30-04-2016 30-04-2016
5145	Meri N Lona	EGL	Letersi Shqipe	SOC0102	Sociologji	01-08-2016
6132	Blerta B Bara	MUZ	Muzike	MUS0002 SOC0102	Histori muzike Sociologji	30-04-2016 01-08-2016
7810	Nora M Beka	CS	Informatike			
8966	Marta L Mara	EGL	Letersi Shqipe	EGL0010 EGL0101	Anglisht Letersi 1	30-12-2015 01-08-2016

Tabela 12-2: Entiteti STUDENT në FN1

Student ID	Emri	Mbiemri	Atësia	Dega ID	Student Dega
2907	Arben	Tiku	R	MAT	Matematike
4019	Gerald	Leka	S	FIZ	Filozofi
5145	Meri	Lona	N	EGL	Letersi Shqipe
6132	Blerta	Bara	B	MUZ	Muzike
7810	Nora	Beka	M	CS	Informatike
8966	Marta	Mara	L	EGL	Letersi Shqipe

Tabela 12-3: Entiteti KURS në FN1

Student ID	Kursi NUM	Kursi Emër	Datë Përfundim Kursi
2907	MAT0011	Matematike 1	01-08-2016
2907	MAT0027	Matematike 2	30-04-2016
2907	EGL0010	Anglisht	30-12-2015
4019	FIZ0010	Filozofi	30-04-2016
4019	CS00100	Programim	30-04-2016
5145	SOC0102	Sociologji	01-08-2016
6132	MUS0002	Histori muzike	30-04-2016
6132	SOC0102	Sociologji	01-08-2016
8966	EGL0010	Anglisht	30-12-2015
8966	EGL0101	Letersi 1	01-08-2016

Forma e parë normale (first normal form) (1NF) përshkruan formatin tabelë në të cilin:

- Të gjitha atributet çelës janë të përcaktuar.
- Nuk ka grupe të përsëritur në tabelë. Me fjalë të tjera, çdo kryqëzim rresht/kolonë përmban njëdhe vetëm një vlerë, jo një grup vlerash.
- Të gjitha atributet janë të varur nga çelësi primar.

Forma normale e dytë (FN2). Forma e dytë normale, shkurt FN2, siguron që të gjitha atributet e çdo entiteti të jenë të varura nga çelësi primar. Për të transformuar të dhënat FN1 në FN2, krijohen entitete të ndara për bashkësitë e atributeve që aplikohen në rekordet me vlera shumëfishe dhe caktohet një çelës i jashtëm në entitetin e ri për të lidhur atë me entitetin e saj të mëparshëm. Pra, vlerat në një entitet nuk duhet të varen në asgjë tjetër përveç se në çelësin primar të entitetit. Një rresht është në FN2 atëherë dhe vetëm atëherë kur është në FN1 dhe çdo atribut jo çelës është plotësisht i varur nga çelësi.

Po të analizojmë të dhënat e tabelave në FN1 vihet re se ka përsëritje në entitetin KURSI, siç janë Anglisht dhe Sociologji. Kjo situatë tregon një thyerje të standardit FN2. Për ta mundësuar këtë nevojitet identifikimi i atributeve që nuk varen nga çelësi primar dhe të hiqen ato. Atributet e hequra së bashku me çelësin primar nga i cili varen, vendosen në një entitet të ri, të quajtur REGJISTRIMI. Çelësi primar i plotë i entitetit origjinal mbetet me entitetin origjinal. Një përfitim tjetër i proçesit të normalizimit është që ju do të ndeshni atribute të reja që nevojitet të specifikohen për entitetet e reja që janë krijuar. Për shembull, ndoshta entiteti i ri KURSI na sjell ndër mend që çdo kursi i caktohet një numër kreditesh që llogaritet për diplomim. Kështu, ne krijojmë një atribut të ri për të ruajtur numrin e krediteve për çdo kurs të veçantë. Rezultatet fundore të normalizimit në FN2 janë përshkruar në tabelat 12-4 dhe 12-5. Theksojmë se për entitetin STUDENT nuk ka ndryshime në FN2.

Tabela 12-4: Entiteti REGJISTRIMI në FN2

Student ID	Kursi NUM	Datë Përfundim Kursi
2907	MAT0011	01-08-2016
2907	MAT0027	30-04-2016
2907	EGL0010	30-12-2015
4019	FIZ0010	30-04-2016
4019	CS00100	30-04-2016
5145	SOC0102	01-08-2016
6132	MUS0002	30-04-2016
6132	SOC0102	01-08-2016
8966	EGL0010	30-12-2015
8966	EGL0101	01-08-2016

Tabela 12-5: Entiteti kursi në FN2

Kursi NUM	Kursi Emër	Kredite
MAT0011	Matematike 1	3
MAT0027	Matematike 2	4
EGL0010	Anglisht	3
FIZ0010	Filozofi	3
CS00100	Programim	3
SOC0102	Sociologji	3
MUS0002	Histori muzike	3
EGL0101	Letersi 1	4

Një tabelë është në **formën e dytë normale (second normal form) (2NF)** kur:

1. Është në 1NF dhe
2. Nuk përmban varësi të pjesshme, asnjë atribut nuk varet nga vetëm një pjesë e çelësit primar. Një tabelë në 2NF mund të shfaqë varësi kalimtare, çelësi primar mund të mbështetet në një apo më shumë atribute joprimare për të përcaktuar funksionalisht atributet e tjera joprimare, sic tregohetnga një varësi funksionale midis atributeve joprimar.

Forma normale e tretë (FN3). Forma normale e tretë siguron që asnjë lidhje midis atributeve nuk ekziston brenda një entiteti. Çdo atribut në një entitet duhet të varet vetëm nga

çelësi primar. Përkufizohet se një rresht është në FN3 atëherë dhe vetëm atëherë kur është në FN2 dhe çdo atribut jo çelës është në mënyrë jokalimtare i varur nga çelësi primar. Rregulla për të identifikuar të mos qenurit FN3 është të shikohet për grupe atributesh vlerat e të cilave mund të aplikohen në më shumë se në një rast i vetëm entiteti. Kur zbulohen atribute të tilla, ato lëvizen në një entitet të veçantë.

Le të shqyrtojmë si shembull të dhënat e STUDENT-it në tabelën e entitetit STUDENT në FN1. Vihet re që ka studentë në të njëjtën degë dhe prandaj ka përsëritje për degën. Për të rregulluar problemin, hiqen atributet e degës që në mënyrë kalimtare varen nga çelësi dhe krijohet një entitet i ri për ta. Tabelat 12-6 dhe 12-7 tregojnë të dhënat e rregulluara në FN3.

Tabela 12-6: Entiteti STUDENT në FN3

Student ID	Emri	Mbiemri	Atësia	Dega ID
2907	Arben	Tiku	R	MAT
4019	Gerald	Leka	S	FIZ
5145	Meri	Lona	N	EGL
6132	Blerta	Bara	B	MUZ
7810	Nora	Beka	M	CS
8966	Marta	Mara	L	EGL

Tabela 12-7: Entiteti DEGA në FN3

Dega ID	Student Dega
MAT	Matematike
FIZ	Filozofi
EGL	Letersi Shqipe
MUZ	Muzike
CS	Informatike

Për të realizuar modelin e plotë të studentit në FN3, diagrama e paraqitur në figurën 12-1 jep modelin e zhvilluar për këtë qëllim.

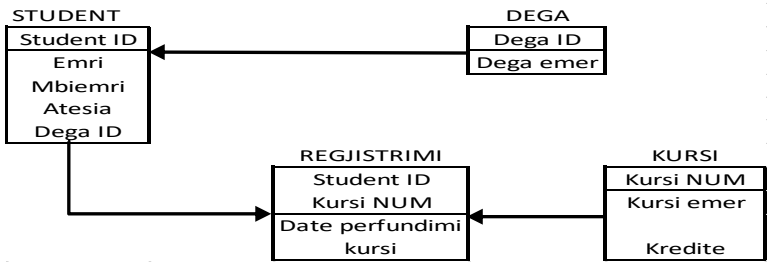


Figura 12-1 Diagrama e normalizuar

Një tabelë është në **formën e tretë normale (third normal form) (3NF)** kur:

1. Është në 2NF dhe
2. Nuk përmban varësi kalimtare.

Forma normale të tjera. Ekzistojnë forma të tjera normalizimi të përkufizuara dhe dokumentuara, por në praktikë ato ndeshen dhe nevojiten më rrallë se tre format e mësipërme. Të tilla janë:

- Forma normale Boyce Codd (FNBC). Kjo formë është rregullim i mëtejshëm i FN3. Një rresht është në FNBC atëherë dhe vetëm atëherë nëse çdo determinant është një çelës kandidat. Determinant është çdo atribut vlera e të cilit përcakton vlerat e tjera brenda një rekordi entiteti. Në përgjithësi, pjesa më e madhe e FN3 janë tashmë në FNBC.
- Forma normale e katërt (FN4). Kjo formë kërkon që asnjë entitet nuk ka më shumë se një lidhje të vetme "një me shumë" nëse atributet "një me shumë" janë të pavarura nga njëra tjetra.
- Forma normale e pestë (FN5). Kjo formë specifikon që çdo varësi bashkimi (join dependency) për entitetet duhet të jetë rrjedhojë e çelësave kandidatë të tyrë.

Tema 13. Koncepte mbi modelin relacional

Modeli relacional u krijua fillimisht nga Ted Codd i kompanisë IBM research në vitin 1970. Modeli përdor konceptet e *relacioneve matematikore* e cila ngjan si një tabelë vlerash. Implementimi i parë tregëtar i modelit relacional u bë në fillimet e vitit 1980 në sistemet e operimit IBM dhe Oracle DBMS. Modelet relacionale komerciale më popullore aktualisht janë DB2, Oracle, SQL Server dhe Access, por ka edhe sisteme open source si Mysql dhe PostgreSQL.

Përkufizime informale

Modeli relacional e paraqet bazën e të dhënave si një bashkësi relacionesh. Çdo relacion është një tabelë vlerash ku çdo rresht në tabelë paraqet një koleksion të dhënash që kanë lidhje me njëra-tjetrën (atribute) të një entiteti real (p.sh. nxënës). Emri i tabelës dhe i kolonave ndihmojnë në interpretimin e kuptimit të të dhënave në çdo rresht. Çdo rresht paraqet fakte rreth një nxënësi të caktuar.

Figura e mëposhtme jep një shembull të entitetit NXËNËS.

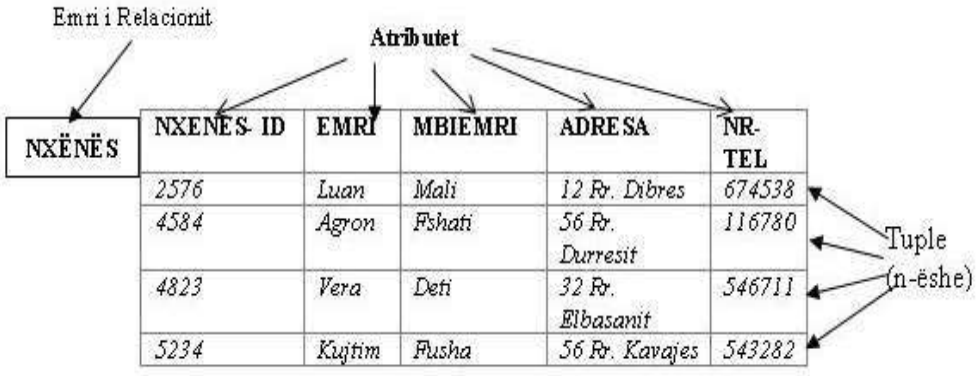


Figura 13-1 Tributet dhe tuple-at e relacionit nxënës

Emrat e kolonave (Nxënës-ID, *Emri, Mbiemri, Adresa, Nr-Tel*) janë vetitë individuale të çdo entiteti.

Të gjithë vlerat në një kolonë janë të të njëjtit tip. Në terminologjinë formale të modelit relacional, rreshti njihet me emrin '**tuple**' ose **n-eshe**, koka e kolonës njihet me emrin '**atribut**' dhe tabela njihet me emrin '**relacion**'. Tipi i të dhënave të çdo atributi ka një fushë

(domain) të vlerave të mundshme. Për shembull, fusha e emrit është bashkësia e karaktereve, fusha e moshës është një vlerë integer (numër i plotë).

Çdo rresht ka një vlerë ose bashkësi vlerash që e identifikon këtë rresht në mënyrë unike në tabelë dhe që quhet **çelës**. Në tabelën nxënës, Nxënës-ID është çelës.

Ndonjëherë Id e rreshtave ose numrat sekuencialë përcaktohen si çelësa për të identifikuar rreshtat në një tabelë dhe këto quhet çelësa artificialë ose zëvendësues.

Përkufizime formale: Domain-et, atributet, tuple-at dhe relacionet
Një fushë ose domain 'D' është një bashkësi vlerash atomike ku lëviz një atribut, ku me atomike kuptojmë vlera të pandashme. Një metodë e zakonshme për specifikimin e domain-it është ajo e përcaktimit të një tipi të dhënash, gjithashtu mund të ketë edhe një format për ta përcaktuar atë.
Skema e një relacioni R shënohet R(A1, A2, …, An) ku R është emri i relacionit dhe atributet e tij janë A1, A2, …, An. Çdo atribut Ai ka një domain ose bashkësi vlerash të vlefshme që ai merr. Shkallë e një relacioni është numri 'n' i atributeve të relacionit.

Shembull:
NXËNËS (Nxënës-ID, *Emri, Mbiemri, Adresa, Nr-Tel*).
Duke përdorur edhe tipin e të dhënave për çdo atribut:
NXËNËS (Nxënës-ID: *integer, Emri: string, Mbiemri: string, Adresa: string, Nr-Tel: string*)
Ku **NXËNËS** është emri i relacionit, i cili ka pesë atribute: Nxënës-ID, *Emri, Mbiemri, Adresa, Nr-Tel*.
Çdo atribut ka një domain ose një bashkësi vlerash të vlefshme. Tuple është një listë vlerash e renditur e vendosur midis '<...>'. Çdo vlerë derivohet nga domein-i përkatës. Një relacion R(A_1, A_2, …, A_n) është bashkësia e n-tuple.
Tabela e mëposhtme jep një përmbledhje të termave të përdorur.

Tabela 13-1 përmbledhje të termave të përdorur

Terma informalë	Terma Formalë
tabelë	relacion
kolonë	atribut
të gjithë vlerat e mundshme të kolonave	domain
rresht	tuple
përcaktimi i tabelës	skemë e relacionit
tabelë e populluar	gjendja e relacionit

Karakteristikat e relacioneve

Disa nga karakteristikat e relacioneve janë:

- Radha e tuple-ave në relacion: Relacioni është një bashkësi tuple-sh. Matematikisht, elementët e bashkësisë nuk kanë një radhë mes tyre. Tuple-at nuk konsiderohen si të renditur edhe pse ato shfaqen në formë tabelare.
- Radha e atributeve në skemën relacionale R: do t'i konsiderojmë atributet në R(A_1, A_2, …, A_n) dhe vlerat në t=<v_1, v_2, …, v_n> si të renditura sipas radhës që ato shfaqen. Një përcaktim më i gjerë i relacionit nuk e kërkon këtë renditje.

- Vlerat në një tuple: Çdo vlerë në tuple është atomike. cdo vlerë në tuple duhet të jetë nga domain-i i atributit për atë kolonë. Ato mund të kenë edhe një vlerë speciale NULL e cila përdoret për të paraqitur vlerat që nuk njihen, ose vlera që ekzistojnë por nuk janë të disponueshme ose nuk aplikohen në atë tuple t.

Do t'i referohemi komponentëve të vlerave të një tuple t si: t[Ai] ose t.A1. Kjo është vlera vi e atributit A1 për tuple-in t.

Konstreinet e modelit relacional dhe skemat

Në një bazë të dhënash ka shumë relacione dhe tuple-t në këto relacione janë zakonisht të lidhura në mënyra të ndryshme. Shtrëngimet përcaktojnë se cilat vlera lejohen dhe cilat jo në një bazë të dhënash. Konstreinet në një bazë të dhënash i ndajmë në tre tipe kryesore:

a. Konstreine të trashëgueshme ose implicite që bazohen në modelin e të dhënave.
b. Konstreine që mund të shprehen direkt në skemën e modelit të të dhënave duke i përcaktuar ato me DDL. Këto quhen konstreine të bazuara në skemë ose eksplicite.
c. Konstreine që nuk mund t'i shprehim direkt në skemë por duhet të realizohen me anë të aplikacioneve. Këto quhen konstreine të bazuara në aplikacione ose semantike.

Konstreinet e domain

Konstreinet e domain-it specifikojnë që për çdo tuple, vlera e çdo atributi A mund të jetë një vlerë atomike nga domain dom(A).

Shtrëngimet e integritetit relacional

Shtrëngimet janë kushte të cilat duhet të jenë të vërteta gjatë gjithë kohës për relacionin. Ato janë të tre tipeve që mund të shprehen në modelin relacional:

Shtrëngimet çelës

Relacioni është një bashkësi tuple-sh dhe nga përcaktimi të gjithë elementët në këtë bashkësi janë të ndryshëm. Nuk ka dy tuple që të kenë të njëjtin kombinim vlerash për të gjithë atributet e tyre. Ka një nënbashkësi të atributeve SK për të cilat vetia e mësipërme është e vërtetë. Për dy tuple të ndryshëm t1 dhe t2 kemi t [SK]≠t [SK] Ky kusht duhet të jetë i vërtetë për çdo gjendjet e vlefshme të R.

Bashkësia e atributeve SK quhet superçelës. Çelësi i R është një superçelës minimal. Pra një superçelës K i tillë që heqja e ndonjë atributi nga K çon në një bashkësi atributesh që nuk është më superçelës. Superçelësi mund të ketë atribute të tepërta, ndryshe nga çelësi. Çelësi është superçelës por jo e kundërta. Çdo bashkësi atributesh që përmban çelësin është superçelës.

Çelësi kënaq dy konstreinet:

a. Dy tuple të ndryshëm në çdo gjëndje të relacionit nuk mund të kenë vlera identike për të gjithë atributet e çelësit.
b. Ai është superçelës minimal, një superçelës nga i cili nuk mund të heqim asnjë atribut pa cënuar unicitetin.

Kushti i parë zbatohet edhe tek çelësat dhe superçelësat, kurse i dyti kërkohet vetëm tek çelësat. Një relacion mund të ketë më shumë se një çelës dhe në këtë rast secili prej tyre quhet "çelës kandidat". Njëri prej tyre zgjidhet si çelës primar për të identifikuar tuple në atë relacion.

Skema e një baze të dhënash relacionale
Skema e një baze të dhënash S është bashkësia e skemave të relacioneve S={R_1, R_2, …, R_m}, dhe një bashkësi konstreinesh integriteti IC. Pra S është emri i gjithë skemës së bazës së të dhënave. Një gjendje database që kënaq të gjithë konstreinet e integritetit quhet gjendje e vlefshme.
Integriteti i entitetit. Atributet e çelësit primar PK të çdo skeme relacion R në S nuk mund të kenë vlera null në asnjë tuple të R. Kjo sepse vlerat e çelësit primar përdoren për të identifikuar tuple-at individualë. Nëse PK ka disa atribute atëherë vlera null nuk lejohet në asnjë prej tyre.

Integriteti referencial është një konstrein i cili përfshin dy relacione. Përdoret për të përcaktuar lidhjet midis tuple-ave në dy relacione: relacioni referues dhe ai i referuar. Tuple-at në relacionin referues R1 kanë atributet FK (të quajtura atribute foreign key) që i referohen atributeve të çelësit primar PK të relacionit të referuar R_2. Thuhet se tuple t_1 i R_1 i referohet tuple-it t_2 në R_2 në qoftë se t1 [FK]= t_2 [PK]. Konstreini i integritetit referencial shfaqet në skemë si një hark nga R_1.FK tek R_2.PK.

Vlera në kolonën e çelësit të huaj (ose kolonat) FK të relacionit referues R_1 mund të jetë:

1. një vlerë e vlerës ekzistuese të çelësit primar PK të R_2; atributi FK i referohet relacionit R_2.
2. ose është null.

Në rastin e mëparshëm, ne kemi t_1[FK]=t_2[PK] dhe themi që tuple t_1 i referohet tuple t_2. Gjithashtu një çelës i huaj mund ti referohet dhe relacionit të vet. Ka edhe tipe të tjera konstreinesh që njihen si konstreine të integritetit semantik të cilat nuk mund të shprehen nga modeli.

Paraqitja e skemës së një baze të dhënash relacionale dhe shtrëngimet e saj

Çdo relacion shfaqet si një rresht me emrat e atributeve dhe emri i tij shkruhet lart tyre. Atributet e çelësit primar paraqiten të nënvijëzuar. Konstreinet foreign key (integriteti referencial) paraqitet me një hark të orientuar nga atributet e foreign key tek tabela e referuar, ose për qartësi tek çelësi primar i tij.

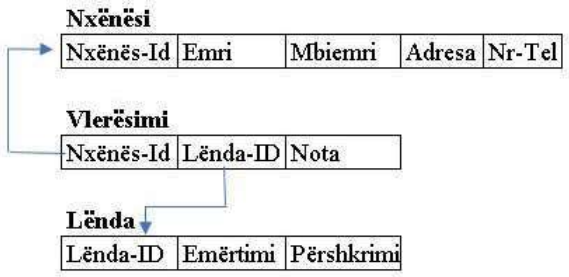


Figura 13-2 Paraqitja e shtrëngimeve të integritetit referencial

Tema 14. Algjebra relacionale

Veprimet e algjebrës relacionale
Sistemet e databazave relacionale janë të pajisura me një gjuhë kërkese për informacion (query). Algjebra relacionale është një gjuhë kërkese proceduriale e cila merr si hyrje një bashkësi të dhënash në formën e rekordeve në tabela dhe nxjerr në dalje një bashkësi tjetër të dhënash po në formën e rekordeve në tabela. Ajo përdor operatorët për të ekzekutuar kërkesat. Veprimet themelore të algjebrës relacionale janë si vijon:

Zgjedhja. Zgjedh rekordet nga një tabelë që kënaqin kushtet e dhëna. Përdoret shënimi $\sigma_p r$. Simboli σ i referohet llojit të operatorit në fjalë, r i referohet tabelës, ndërsa p është një formulë që përbëhet nga një kombinim i operatorëve llogjike "and" (dhe), "or" (ose) dhe "not" (nuk), dhe operatorëve të zakonshëm matematikë të krahasimit =, ≠, ≥, < , >, ≤. Për shembull,

$$\sigma_{subjekti\ =\text{"databaze"}}(libra)$$

$$\sigma_{subjekti\ =\text{"databaze"}\ AND\ çmimi\ =\text{"450"}}\ (libra)$$

$$\sigma_{(subjekti\ =\text{databaze" AND çmimi="450")}\ OR\ viti=\text{2016"}}(libra)$$

Rezultati i shembullit të parë është bashkësia e rekordeve të tabelës "libra" të cilat e kanë vlerën e atributit "subjekti" me etiketën "databaze". Rezultati i shembullit të dytë është bashkësia e rekordeve të tabelës "libra" të cilat i përkasin subjektit "databaze" dhe janë me çmim 450. Rezultati i shembullit të tretë është bashkësia e rekordeve të tabelës "libra" të cilat i përkasin subjektit "databaze" dhe janë me çmim 450, ose rekordeve të lidhjes libra që janë të vitit "2016". Një shembull i skematizuar jepet në figurën 14-1.

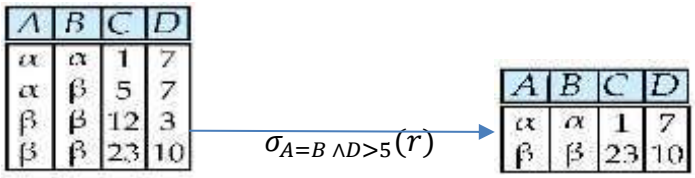


Figura 14-1

Projektimi. Projekton kollonat që kënaqin një kusht të dhënë. Përdoret shënimi $\prod_{A1,\ A2,An}(r)$, ku A1, A2, … , An janë emrat e atributeve të tabelës r. Ndërkohë, rreshtat e dublikuara janë automatisht të eliminuara. Për shembull, $\prod_{subjekti,\ autori}(libra)$, zgjedh dhe projekton kollonat e atributeve "subjekti" dhe "autori" nga tabela "libra". Një shembull i skematizuar jepet në figurën 14-2.

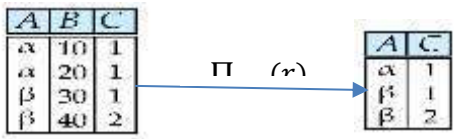


Figura 14-2

Bashkimi. Përdoret shënimi $r \cup s = \{t | t \in r\ ose\ t \in s\}$, ku r dhe s janë tabela të ndryshme të databazës. Për këtë veprim nevojitet plotësimi i tre kushteve: tabelat duhet të kenë numër të njëjtë atributesh, bashkësitë e vlerave të atributeve duhet të jenë të përputhshme, dhe rekordet e dublikuara janë automatikisht të shmangura. Për shembull, $\prod_{autori}(libra) \cup \prod_{autori}(artikuj)$ projekton emrat e autorëve të cilët kanë shkruar një libër ose një artikull, ose të dyja. Një shembull i skematizuar jepet në figurën 14-3.

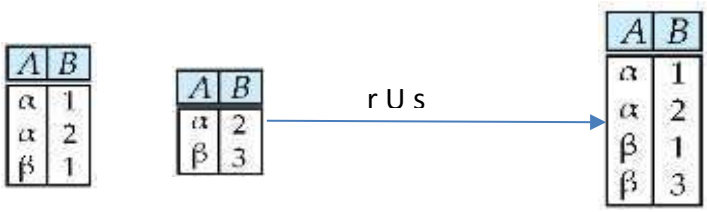


Figura 14-3

Ndryshesa. Rezultat i këtij veprimi janë rekordet që janë në një tabelë, por nuk janë në tabelën e dytë. Shënimi i përdorur është $r - s = \{t | t \in r \; dhe \; t \notin s\}$, ku r dhe s përfaqësojnë dy tabela të ndryshme. Për shembull, $\prod_{autori}(libra) - \prod_{autori}(artikuj)$ nxjerr emrat e autorëve që kanë shkruar libra por nuk kanë shkruar artikuj. Një shembull i skematizuar jepet në figurën figurën 14-4.

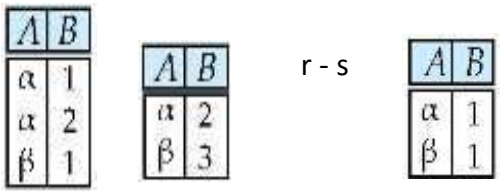


Figura 14-4

Prodhimi kartezian. Kombinon informacionin e dy tabelave të ndryshme në një të vetme. Tabela e re ka të gjithë atributet e dy tabelave dhe përmban si rekorde të gjithë kombinimet e mundshme të rekordeve nga të dyja tabelat. Shënimi i përdorur për dy tabela r dhe s është $r \; X \; s = \{qt | q \in r \; dhe \; t \in s\}$. Shembulli $\sigma_{autori="Kadare"}(libra \; X \; artikuj)$ prodhon një tabelë që paraqet gjithë librat dhe artikujt e shkruar nga autori Kadare. Një shembull i skematizuar jepet në Figura 14-5.

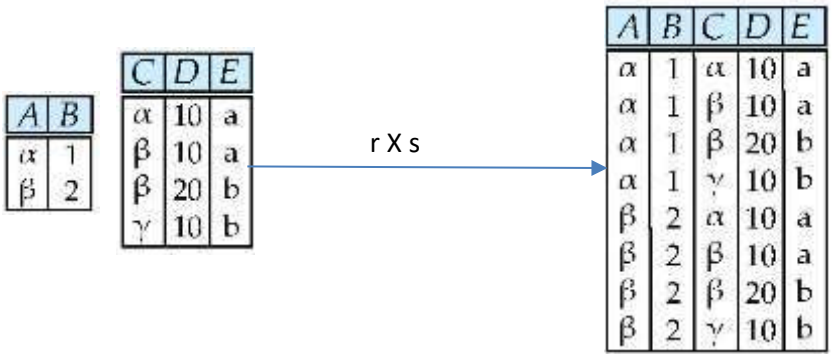


Figura 14-5

Riemërtimi. Përfundimet e algjebrës relacionale janë tabela pa ndonjë emërtim. Ky veprim na mundëson të emërtojmë tabelën rezultat. Shënimi i përdorur është $\rho_X E$, i cili nënkupton që rezultati i shprehjes E ruhet me emrin X.

Përfundimisht, theksojmë se veprimet e bashkësisë {σ, π, ∪, ρ, –,X} përbëjnë bashkësinë e plotë të veprimeve të pavarura, që nënkupton se çdo veprim tjetër i algjebrës relacionale shprehet nëpërmjet një serie veprimesh nga kjo bashkësi. Për shembull, veprimi i prerjes shprehet R ∩ S = (R ∪ S) – ((R – S) ∪ (S – R)).