Aurkibidea

ı	INFORMAZIO-SISTEMAN ETA DATO-BASEAN				
	1.1	Informazio-sistemaren kontzeptua	2		
	1.2	Informazio-sistemen osagaiak	5		
	1.3	Fitxategiak egiteko tradiziozko sistemetatik datu-baseetara	7		
	1.4	Datu-baseek fitxategi klasikoen aurrean dituzten abantailak eta desabantailak	8		
2	DATU-BASEAK ETA DATU-BASEAK KUDEATZEKO SISTEMAK				
	2.1	Datu-baseen kontzeptua	11		
	2.2	2 Datu-baseen abstrakzio-mailak			
		✓ Erabiltzailearen egitura logikoa: kanpoko eskema	14		
		✓ Egitura logiko globala: eskema kontzeptuala	14		
		✓ Egitura fisikoa: barne-eskema	15		
	2.3	Datu-baseak eta datu-baseak kudeatzeko sistemak	17		
	2.4	DBKS-en osagaiak	20		
		✓ Datuak definitzeko lengoaia (DDL)	20		
		✓ Datuen biltegiratzea definitzeko lengoaia (DSDL)	20		
		✓ Datuen manipulazio-lengoaia (DML)	21		
		✓ Datuen hiztegia	22		
		✓ Datu-basearen administratzailea	23		
		✓ Datu-basearen erabiltzaileak	24		
	2.5	2.5 Datu-ereduak: kontzeptua eta motak			
		✓ Eredu hierarkikoa	25		
		✓ Sare-modeloa	27		
		✓ Eredu erlazionala	27		
3	DATU-BASE ERLAZIONALAK				
	3.1	3.1 Diseinu-metodologien etapak			
	3.2	Diseinu kontzeptuala: e/e e eredua	32		
		✓ Ereduaren elementuak	33		
	3.3	3 Diseinu logikoa: eredu erlazionala			
	3.4	4 Eskema kontzeptualetik eskema logiko estandarrerako aldakuntza			
	3.5	5 Normalizazioari buruzko teoria			
4	SQL LENGOAIA				
	4.1				
	4.2	.2 SQL-ren komando-motak			



4.3	Datuak definitzeko lengoaia (DDL)		
	✓ Taulak sortzen	55	
	✓ Taula baten egituraren aldarazpena	62	
	✓ Taula bateko mugak eranstea eta ezabatzea	64	
	✓ Taula ezabatzea	64	
	✓ Ikuspegiak sortzea eta erabiltzea	65	
	✓ Sinonimoak sortzea	67	
	✓ Izen-aldaketak	68	
	✓ Aurkibideak sortzea	68	
4.4	Datuak manipulatzeko lengoaia (DML)	69	
	✓ Datuak txertatzen. INSERT agindua	69	
	✓ Datuak aldatzea. UPDATE agindua	71	
	✓ Taula bateko lerroak ezabatzea. DELETE agindua	72	
	✓ Taulen informazioa kontsultatzea. SELECT agindua	72	
4.5	Eranskina: funtzioak	92	
	✓ Funtzio aritmetikoak	92	
	✓ Karaktere Kateen funtzioak	93	
	✓ Zenbaki-balioak itzultzen dituzten funtzioak	95	
	✓ Fitxak erabiltzeko funtzioak	96	
	✓ Bihurketa-funtzioak	97	
	✓ Beste funtzio batzuk	99	
	✓ Multzoen funtzioak	100	
4.6	Transakzioak kontrolatzeko komandoak		
	✓ COMMIT sententzia	101	
	✓ ROLLBACK sententzia	101	
	✓ SAVEPOINT sententzia	102	
	L*Plus		
5.1	SQL*Plus-en ingurua		
5.2	SQL*Plus abian jartzea		
5.3	Komandoak sartzea eta egikaritzea		
5.4	SQL-ren buffer-a		
5.5			
	✓ SPOOL artxiboak		
	✓ Komandoen artxiboa		
5.6	Txostenak sortzea		
	✓ Inguru-parametroak		
	✓ Formatu-komandoak	116	

5



DATU-BASEAK

1 INFORMAZIO-SISTEMAK ETA DATU-BASEAK

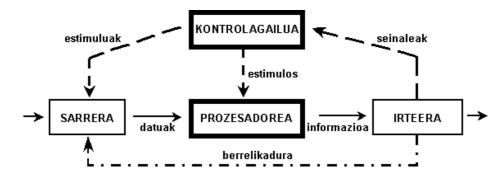
1.1 Informazio-sistemaren kontzeptua

Erakunde guztiek behar dute beren jarduerarako elementuen artean eta, gehienetan, kanpotik sistemara eta sistematik kanpora ere, transmitituko duten informazio-multzoa. Langileen arteko pertsonen arteko harremanen bidez gauzatu ohi da komunikazio horren zati bat, *informazio informalaren sistema* da hori. Horrelako informazio-fluxuak, baina, baldin eta erakundeak konplexuak badira, ez da behar bestekoa izaten, eta garesti suertatzen da. Horregatik, beharrezkoa izaten da *informazio formalaren sistema* edukitzea, *antolaketarako informazio-sistema* ere baderitzona. Sistema horrek, erakundea bera baino goragoko mailako sisteman integraturik, era eraginkorrean eta efikazean emango dio erakundeari berorrek behar duen informazioa.

Informazio formalaren sistema guztiak, zeintzuei hemendik aurrera, besterik gabe, *informazio-sistema (IS)* esango baitiegu, erakunde batek (enpresa batek edo edozein motatako erakunde publikok zein pribatuk) informazioa dela-eta, dituen beharrizanei erantzuteko diseinatzen dira, eta erakundean murgildurik egoten da. Inguruko datuak bildu behar ditu ISk (hala erakundean bertan nola kanpoko iturrietan), eta kudeaketarako eta erabakiak hartzeko erakundeak beharrezko duen informazioa behar dute izan emaitzek; bestalde, IS araupetuko duten helburuak eta jarraibideak markatu beharko ditu erakundearen zuzendaritzak.

Irteerek markaturiko helburuak markatzen duten eraren arabera jarduketa kontrolatzen duena izango da *sistema dinamikoa*; horrela, gehienetan denboran aldagarriak diren inguruaren baldintzetara egokituko da sistema era dinamikoan. Sistemaren kontrola egiteko, barne-mekanismoak erabil daitezke (autorregulaturiko sistemak), inguruan kokaturik dauden mekanismoak, edo bietarikoak.

Sistema dinamikoak inguruarekiko elkarren arteko eraginean daude eta, horrela, sarrerak eta prozesuak etengabe moldatuz doaz, irteera jakin batzuk lortzeko. 1.1. irudian ageri den diagramaren arabera irudika daitezke.



1.1. irudia. Sistema dinamikoaren eskema.

Sistemaren kontrolagailuak, zeinak plangintza- eta gobernu-funtzioak baititu, irteeratik hartzen duen informazioaren arabera dihardu, eta estimuluak bidaliko ditu sarrerako unitatera eta prozesadorera, irteerak sistemaren helburuei erantzun diezaien lortzeko. Horretarako, informazioa bereganatu, interpretatu, aurreikusitako helburuekin erkatu eta sistemaren erregulazioa eskatzen duten kontrol--pultsuak emititzeko gauza izan behar du.

Prozesuan kontsumitu edo aldaratzen diren elementuak dira sistemaren sarrerak. Fabrikazio--prozesuetako lehengaien parekotzat har daitezke; informazio-sistemen kasuan datuak izango dira. Bertan sarrerak kontsumitzen ez direlako bereizten dira IS beste sistema batzuetatik. Suntsitu gabe, aldarazi egiten dira, zeren sistemaren beraren datu-basean gelditzen baitira bildurik.

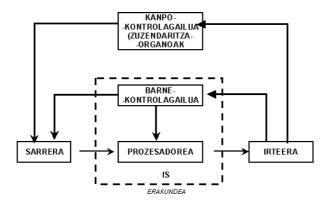
Prozesuan sortzen diren elementuak dira irteerak. Fabrikazio-prozesuen produktu amaitua lirateke; kasu honetan informazioa da irteera.

Tratamendua gauzatzen den tokia da prozesadorea, eta aldarazi gabe eta sortu gabe, bertan parte hartzen duten elementu guztiak bereganatuko ditu, hau da, sarrerak eta irteerak ez direnak.

Sistema askotan berrelikadura ere izango da. Irteeratik sarrerara doa, kontrolagailutik pasa gabe (1.1. irudiko puntu-lerroa).

ISetan (batik bat dinamikoak diren sistemetan), IStik at dagoen kontrola izango da, hain zuzen ere, IS ari deneko eremua ezarriko duten erakundeko zuzendaritza-organoak; aldi berean, baina, garapen--maila handiagoa edo txikiagoa izango duten eta zuzendaritza-organoen aginduak, bai eta goragoko mailetan dauden organoetatik datozen legeak eta arauak ere, interpretatu eta zehaztuko dituzten auto--erregulaziorako mekanismoak eduki beharko ditu ISak barruan, erregulatuak izan behar duten ISaren unitateetara transmititzeko.

ISetan bi mailatan edo gehiagotan emango den kontrola dela esan genezake: batetik, kanpoko kontrola, zuzendaritza-organoek gauzatua, eta barneko autorregulazioa (ikus 1.2. irudia) bestetik. ISaren autonomia handiagoa edo txikiagoa izatea barruko kontrolak kanpokoaren gainetik duen nagusitasunaren arabera dagoen faktorea izango da.



1.2. irudia. Erakunde baten ISaren bi mailako kontrola.

INFORMATIKAKO SISTEMEN ADMINISTRAZIOA

Erakundean barrena ibilaraziz, banatuz eta beharrezkoa den aldeetara eramanez informazioa bultzatzen duen motor batekin erka daiteke IS. Egiteko hori gauzatzeko, beharrezkoa da sistemak aldez aurretik datuak sortzen diren tokian hartu eta informazio erabilgarri bilakarazteko prozesatzea.

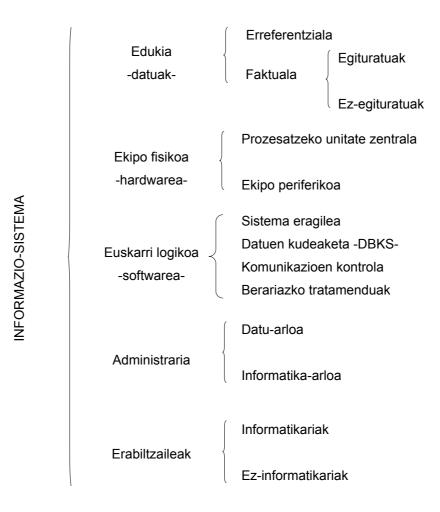
ISaren eta txertatuta dagoen erakundearen artean lotura estua dago; egia esan, IS erakundea osatzen duten sistemetako azpi-sistema besterik ez da. Ezinbestekoa da hori gogoan izatea; izan ere, beharrezkoa den elkarren arteko ekintza izan ez eta bien artean desfasea gertatuz gero, ISk ezin izango ditu diseinatua izatean edukitako helburuak bete.

ISaren eta erakundearen arteko moldaketarik eza dugu eraginkorrak izateko sortuak izaniko sistema askoren akatsen zergatia, eta frogatuta dago porrot horien arrazoiak sarriago egoten direla gizarte- eta giza-karietan diseinu teknologikoan baino.

ISak informatizaturik zertan izan eduki ez eta eskuz erabili daitezkeen arren, gaur egungo ISek informatika-teknikak darabiltzate, eta Datu Baseen Kudeaketa Sistemen bidez gauzatu ohi dira, sarri askotan, informazioaren tratamenduak eta berreskurapenak.

Informazio-sistemen osagaiak

1.3. irudian laburbildurik ageri diren osagaiek osatzen dituzte informazio-sistemak.



1.3. irudia. Informazio-sistemaren osagaiak.

Datu-multzoek (fitxategiek edo datu-baseek) osatzen dute ISaren edukia, dagokion deskribapenarekin, ordenagailuaren euskarrian bildurik. Informazio-sistemaren bidez lortu nahi den helburura moldatu beharko dira datuak eta, horregatik, oro har, helburu horiek anitzak direnez gero, sistemara bilduta dauden datuak ere era ezberdinetakoak izango dira.

Garrantzitsua da bi eratako datuak bereiztea: erreferentzialak eta faktualak. Informazio erreferentzialaren sistemek dokumentuen erreferentzia bibliografikoak izaten dituzte, eta bertan aurki daiteke informazioa, baina ez informazioa bera. Horrela, datua berreskuratzen denean (hau da, erreferentzia), beharrezkoa da iturri den dokumentua lortzea. Sistema faktualek, ordea, bila ibilitako informazioa itzuliko dute, eta zuzenean erabil daiteke informazio hori, bestelako informazio-zirkuituetara jotzen ibili beharrik gabe.

Batez ere, faktualekin erabiltzekoa den beste sailkapen bat formatuarekin legoke lotuta, eta horren arabera, datuak egituratuak edo ez-egituratuak izan daitezke (formateatuak eta ez-formateatuak ere baderitze). Lehenbizikoek nolabaiteko egitura edo formatua daukate, eta eremuek posizio finkoak okupatuko dituzte bertan (horrela, langileen fitxategietan langilearen ENA aurkituko dugu lehenengo, eta fitxategiaren aurreneko zortzi posizioak okupatuko ditu; ondoren izen-deiturak etorriko dira, eta abar). Formatu finkorik ezin izan dezaketen beste datu batzuk daude, ordea. Esate baterako, testuak (dokumentu-sistemek berez izaten dituztenak) edo datu multimediak (ahotsa, irudia, eta abar); datu ez--egituratuak dira. Kudeaketarako bi sistemaren arteko bereizketa egin ohi da, datuak egituratuak ala ez--egituratuak diren.

- ✓ Datu Baseen Kudeaketa Sistemak (DBKS): datu egituratuen tratamenduan (definizioa, eguneratzea eta berreskurapena) dihardute.
- ✓ Informazioa Berreskuratzeko Sistemak (IBS): Datu ez-egituratuen (dokumentuen) tratamenduaz arduratzen dira.

Gaur egun bi motako tratamendu horiek bat egin dezaten da joera argia, horrela, sistema bakar batek eskainiko baititu bi funtzionaltasunak.

Bi azpi-sistemak osaturik dago tratamenduaren edo prozesuaren funtzioari eutsi behar dion ordenagailua: ekipo fisikoa (hardwarea) litzateke bata eta bestea, berriz, euskarri logikoa (softwarea).

Programa multzoak, dokumentazioak, lengoaiek eta abarrek osatzen dute softwarea, eta berorrek kudeatu behar ditu datuak, horretarako Datu Baseen Kudeaketa Sistema (DBKS) erabilirik eta komunikazioak kontrolatu, eta tratamendu berezien beharrizanei (langileriaren kudeaketari esate baterako) erantzuna eman behar die, horretarako guztirako sistema eragilea baliaturik.

Sistemaren beste osagai funtsezko bat administraria dugu edo, hobeto esanda, administrazio--unitatea (pertsona barik, funtzioa baita). Beraren egitekoa kalitatea ziurtatzea eta datuak zuzen eta iraunkorki erabiltzeko bidea ematea da. Administraria ez da datuen jabea, datuen kudeatzailea eta zaintzailea baizik, eta hala sistemaren edukietara nola arlo informatikora hedatzen da beraren erantzukizuna, nahiz eta bi eginkizun horiek (edukiaren administrazioa eta DBKSaren administrazioa) erakundearen unitate ezberdinek bere gain izan ditzaketen.

Azkenik, erabiltzailea, hau da, ISra joko duen pertsona edo pertsona-multzoa ere, sistemaren osagaitzat hartuko dugu. Erabiltzaile horiek informatikariak izan daitezke (analistak eta programatzaileak), edo informatikari dagokionean ezagutza urria duten eta, gehienetan era konbentzionalean eta oso lengoaia errazen bidez, datuak kontsultatu edo eguneratu behar dituzten azken erabiltzaileak.

1.3 Fitxategiak egiteko tradiziozko sistemetatik datu-baseetara

Zenbait informazio-sistemaren egoeraren analisia eginez gero, fitxategien ugalketa duten sistemak aurki ditzakegu, berorietako bakoitza aplikazio jakin baterako berariazkoa izaki (ikus 1.4. irudia). Datuak behin baino gehiagotan biltzen dira, eta hainbat artxibategitan errepikaturik egoten dira. Erredundantzia horrek, baliabideak xahutu ez ezik, emaitzetan dibergentziak sorrarazi ere sorrarazten ditu sarri askotan.

Era horretako informatika-sistemei batzuetan prozesura begira dauden sistemak esan izan zaie, datuei, zein aplikazio jakin baterako diseinaturik dauden fitxategietan biltzen baitira, berorietan ematen zaien trataera azpimarratzen delako. Elkarren artean guztiz independenteak diren eretan analizatzen eta ezartzen dira aplikazioak eta, datuak, beraien artean transferitu ordez, bikoiztu egiten dira lanek behar dituzten bakoitzean.

Planteamendu horrek, bigarren mailako memoriaren alferreko okupazioa sorrarazi ez ezik, prozesu-denborak luzarazi ere egiten ditu, behin eta berriro gertatzen baitira kontrol eta eragiketa berberak fitxategietan. Are larriagoak dira, baina, honelako sistemetan sarritan agertzen diren sendotasunik ezak; izan ere, datu berberak fitxategi batean baino gehiagotan daudenez, eguneratzea ez baita guztietan batera gertatzen.

Bestalde, datuek euskarri fisikoarekiko eta programekiko duten mendekotasunaren ondorioz, malgutasunik eta moldagarritasunik eza izaten da aldaketan, eta horrek oso eragin kaltegarria dauka sistema informatikoaren talde-errendimenduan.

Datu-multzoaren kudeaketa arrazionalagoaren beharrizana argi eta garbi ondorioztatuko da analisi honetatik eta, horrela, datu-basean oinarri hartuko duen ikuspuntu berria azalduko da. Bertan, behin baino ez dira bilduko datuak, tratatzeko era dena delarik ere (ikus 1.6. irudia).

Ohiko sistemetan datuen trataerari dagokionez ageri ziren arazoei irtenbidea emateko, informazio-sistemaren ikuspegian errotiko aldaketa eginarazi zuela ikus dezakegu hortaz. Aplikazio zehatz baterako diseinaturik egon beharrean, erakunde guztiaren informazio-beharrizanei erantzutekoa den multzo egituratu batean antolatu eta mantenduko dira datuak. Kontuan hartu behar dugu beharrizan horiek, denbora aurrera joan ahala, ugarituz joango direla.

Prozesura begira dauden sistemen ordez ari dira ezartzen datuei begira dauden sistema hauek, zeren, aurreko horiek, fidagarritasun urrikoak, errealitatera egokitu gabeak eta konfidentzialtasuna ziurtatzeko tenorean ere eraginkortasun gutxiagokoak izaki, pixkanaka-pixkanaka erabiltzaileen konfiantza galduz joan dira.

1.4 Datu-baseek fitxategi klasikoen aurrean dituzten abantailak eta desabantailak

Datu-baseak datuei begira dauden sistemek sistema informatikoen prestazioen kalitatea hobetzeko eta errendimendua areagotzeko eginiko planteamendu berriarentzako erantzun gisa sortu ziren, eta hainbat abantaila daukate fitxategi-sistema klasikoen ondoan.

Besteak beste, honako hauek dira datu-baseen sistemen abantailak:

Datuen independentzia tratamenduei dagokienez eta alderantziz

Datuek eta tratamenduek elkarren artean duten independentziaren eraginez, azken horietan gertatzen den aldaketa batek ez du derrigortuko datu-basea berriro diseinatzera. Bestalde, eraketa berriak sartu izanak, beste batzuen desagerpenak, egitura fisikoan edota iristeko bideetan gauzaturiko aldaketek ez dituzte programak aldaraziko. Tratamenduek egituraren aurrean daukaten independentzia horrek kontuan hartzeko moduko abantaila dakar, datuetan aldaketak gauzatzean aplikazioak berriro programatzeak sorrarazten duen ahalegin garrantzitsua saihesten delako.

Emaitzen koherentzia

Datu-baseko informazioa behin baino bildu eta gordetzen ez denez, datu berberak erabiliko dira tratamenduetan eta, horregatik, guztien emaitzak koherenteak eta erabat alderagarriak izango dira. Gainera, datuetan erredundantziarik ez denez (edo gutxiagotu egingo da behintzat), sistema klasikoan, datu bat aldatzean, fitxategi zenbait eguneratzera derrigortzeak zekarren arazoa ezabatuta dago. Horrela, fitxategi guztietan eguneratzeak batera gertatu ez izanak sorrarazten duen emaitzen arteko dibergentzien eragozpena ere desagertu egingo da.

Informazio-balio handiagoa

Datu-baseak mundu errealaren isla izan behar duenez gero, datuen arteko loturak bilduko dira bertara eta, horregatik, multzoaren informazio-balioa eratzen duten banakako elementuen informazio-balioaren batuketa baino handiagoa izango da; hau da, sinergia-efektuak dihardu kasu honetan.

Datuak eskuragarriago daude erabiltzaileentzat

Datu-baseen metodologia erabiltzen denean, erabiltzaileak ez dira jadanik datuen jabe izango, aplikazioen multzoaren artean banatuko baitira datuok. Horrela, eskuragarriago daude datuak erabili beharra daukaten guztientzat, baldin eta beraietara iristeko baimena badaukate. Dagoen informazioari dagokionez, gardentasun handiagoa ere izango da, zeren basean dauden datu guztiak leku askotara heda daitekeen eta bitarteko informatikoen bidez hel dakiokeen katalogo edo hiztegi batean egongo baitira.

Informazioaren dokumentazio hobea eta normalizatuagoa, datuek osatuta baitago

Ikuspegi klasikoan eduki semantikotik banandurik daude datuak: lehenbizikoak fitxategietan biltzen dira, eta programetan dagoen programazio-hizkuntzaren bidez deskribatzen dira. Datuen dokumentazioa, analistak edo programatzaileak egina, normalean ez da behar adinakoa izaten, eta batzuetan, izan ere ez da horrelakorik. Gainera, gehienetan ez da inolako estandarizaziorik izaten. Neurri handi batean leunduko da arazo hori datu-baseak erabilita; izan ere, datuak ez ezik, beraien semantika ere gordeko baitira base berean.

Eraginkortasun handiagoa datuak biltzean, balidatzean eta sisteman sartzean

la erredundantziarik ez denez, datuak behin baino ez dira biltzen eta baliozkotzen, eta horrela areagotu egingo da biltzearen aurretiko prozesu guztiaren errendimendua.

Biltze-espazioaren murrizketa

Erredundantzien desagerpenaren (edo gutxiagotzearen), eta trinkotze-tekniken erabilpenaren ondorioz, datu-baseetan gutxiago okupatuko da bigarren mailako memoria -disko magnetikoa-. Kontuan hartu behar da, hala ere, sistemaren elementuek (hiztegiak, erreferentziek, erakusleek eta abarrek) nahikoa espazio okupatzen dutela.

Datu-baseen abantailak laburbiltzen dituen taula ikus dezakegu 1.8 irudian.

		DATU-BASEEN ABANTAILAK
		✓ Datuen independentzia tratamenduekiko eta alderantziz.
	A) DATUAK	✓ Datuak eskuragarriago daude.
ZIA	N DATUAN	✓ Eraginkortasun handiagoa datu-bilketan, kodifikazioan eta sistema- rako sarreran.
EN T	B) EMAITZAK	✓ Koherentzia handiagoa.
FER		✓ Informazio-balio handiagoa.
ERREFERENTZIA		✓ Dokumentazio hobea eta normalizatuagoa.
_	C) ERABILTZAILEAK	✓ Azken erabiltzaileen irispide azkarragoa eta errazagoa.
		✓ Erraztasun gehiago erabiltzaileek datuak konparti ditzaten.
		✓ Malgutasun handiagoa aldatzen diren eskariei erantzuteko.

1.8. irudia. Datu-baseen abantailen laburpena.

Lehenago ere adierazi dugunez, datu-baseek abantailak ez ezik, ISren orientazioa aldatzeari buruzko erabakia hartu baino lehen balioetsi behar diren desabantailak ere eduki ditzake. Honako hauek aipatu behar dira desabantailei dagokienez:

Instalazio garestia

Datu-baseen sistema ezartzeak kostu handia eragin dezake, hala ekipo fisikoari (instalazio berriak eta handiagotzeak) nola logikoari dagokionez (sistema erabiltzeko beharrezkoak diren sistema eragileak, programak, konpiladoreak, eta abar). DBKS erostearen eta mantentzearen kostuak leudeke, gainera, bestalde.

Langile espezializatuak

Ezagutzak ezinbestekoak dira datu-baseak behar bezala eta era eraginkorrean erabiltzeko eta, batez ere, datu-baseak diseinatzeko eta administratzeko, eta horrek langile espezializatuen beharra du berekin.

Ezarpen luzea eta nekeza

Arestian adierazi ditugun zergatiak direla-eta, datu-baseen ezarpena zeregin luze eta nekeza bilaka daiteke. Garapenean zehar agertzen diren zailtasunek, gehienetan, hasiera batean aurreikusitako epeak luzarazi ohi dituzte.

Errentagarritasunik eza epe laburrean

Datu-basearen ezarpenak, bai langiletan eta ekipotan dakartzan kostuengatik, bai eraginkor egon arte behar den denboragatik, ez da errentagarria izaten epe laburrean, baizik eta epe ertainean edo, areago, epe luzean.

Estandarizazio urria

Datu-basea sorrarazteko tenorean agerian azaldu ohi den oso arazo garrantzitsua, erabiltzaileei datu-baseen sistemen erabilpena erraztuko liekeen estandarizazioaren falta izaten da. Datu--baseei dagozkien hainbat estandar definitu diren arren, merkatuan dauden datu-baseen kudeaketarako sistemetako batek ere ez ditu zehaztasun guztiak betetzen eta, gainera, beraien artean bateraezin bilakarazten dituzten ezaugarri erantsiak edukitzen dituzte.

Teoriaren eta praktikaren arteko desfasea

Teoria praktika baino askoz aurreratuago doanez, sarri askotan erabiltzaileek, zuzendaritza-taldeetakoek batik bat, gaur egungo DBKSek eman benetan ematen dituzten baino prestazio hobeak eman ditzaketela pentsatu ohi dute, oraindik teorikoak baino ez diren zenbait alde jadanik errealitate direlako ustean.

Izaten da frustrazio-arriskurik zuzendaritza--taldeetako kideen eta datu-baseen erabiltzaileen aldetik, eta horrek honelako sistemek eskaintzen dituzten benetako aukerak ahaztera eraman gaitzake. Datu-baseak ez dira aurreratuta edo ez hain aurreratuta dagoen teknologia berri hutsa, ISen ikusmolde ezberdin baten fruitu dira eta, horrela izanik, eragin erabakigarria edukiko dute beren inguruaren egituretan eta antolaketan. Hori kontuan hartu ezik, datu--baseek izan ditzaketen abantailetako asko ez dira errealitate bihurtuko eta bai, ordea, areagotuko desabantailak eta arazoak.

DATU-BASEAK ETA DATU-BASEAK KUDEATZEKO SISTEMAK

Datu-baseen kontzeptua

Hainbat eta hainbat dira datu-baseen definizioak eta, arretaz analizatuz gero, elementu jakin batzuei dagokienez ia denetan adostasunak daudela ikusi ahal izango dugu; halatan ere, funtsezkoak edo, behinik behin, garrantzi handikoak diren beste zenbaiten eza ere atzeman daiteke, eta horiek, datu-baseen ezaugarri izateaz gain, kontzeptu honen eta fitxategien kontzeptuaren arteko aldeak markatzen dituzte.

Datu-baseak adierazpena hirurogeiko hamarkadaren hasieran agertu zen lehenengoz. 1963.ean sinposioa egin zen Santa Monikan (AEB) eta izenburuan bazetorren Data Base zioen adierazpena. Bileretako batean, akten arabera denek onartu ez zuten datu-baseen definizio bat proposatu zen. Ondoren, 1967.ean, Codasyl izeneko estandarizazio-taldeak hasiera batean zeukan eta datu-basea adierazpenik ageri ez zen izendapenaren ordez Data Base Task Group izena hartzea erabaki zuen. Apurka-apurka inposatuz zihoazen datu-base zelakoaren kontzeptua eta adierazpena.

Datu-basearen funtsezko ezaugarriak aztertuko ditugu guk lehenbizi. Kasu batzuetan lehenago azaldu ditugun abantailekin bat datoz, eta ezaugarri horiek erabiliko ditugu datu-base kontzeptuaren oinarri gisa.

Lehenik eta behin, eta puntu honetan bat datoz definizio guztiak, esan beharra daukagu aldakorra ez den informatika-euskarri batera bildutako datuen multzoa, bilduma edo gordailua dela datu-basea. Ahalik eta eduki semantiko gehien biltzeko gauza den eredu baten arabera daude elkarrekin loturik eta egituraturik datuak. Mundu errealean datuen arteko loturek duten garrantzia kontuan harturik, ezinbestekoa da datu-baseak lotura horiek biltzeko gaitasuna izatea. Mundu errealean, gainera, badira murrizketa semantikoak, eta gero eta garrantzi handiagoa ematen zaielarik, gaur egungo sistemetan datuekin batera biltzeko joera dago, elkarren arteko loturekin gertatzen den bezala. Datu-eredu batean oinarriturik deskribatu eta manipulatuko dira datu-baseak.

Datuen erredundantzia kontrolatu beharra dago, kaltegarriak diren edo beharrezkoak ez diren bikoiztasunak saihesteko, eta eraginkortasunarekin lotuta dauden helburuei erantzuteko xedeaz askotan komenigarri diren erredundantzia fisikoak, sistemaren beraren bidez tratatzeko, trinkotasun faltarik izan ez dadin. Erredundantzia logikorik ez, baina eraginkortasunarekin lotutako zioez nolabaiteko erredundantzia fisikoa onartzen dela esanez laburbil genezake puntu hau. Hortaz, forma bakarrean eguneratuko ditu erabiltzaileak datuak, eta sistema arduratuko da datua errepikatuta egon daitekeen eremu guztiak fisikoki aldatzeaz, baldin eta erredundantzia fisikoa badago. Horri sistemak kontrolaturiko erredundantzia ere baderitzo.

Antolakuntzaren multzoaren zerbitzurako izan ohi dira datu-baseak, eta ohiko baliabideei erantsitako beste bitarteko baten moduan erabiliko dira datuak. Beraz, hainbat erabiltzaileri eta zenbait aplikaziori erantzun beharko diete, fitxategien sistemek ez bezala. Fitxategi-sistemetan, fitxategi bakoitza aplikazio jakin baten beharrei erantzuteko dago diseinaturik.

Datu-baseen beste alderdi garrantzitsu bat, datuen eta tratamenduen artean dagoen independentzia fisiko eta logikoa da. Independentzia hori, datu-baseen funtsezko helburua izanik (aipatua dugu datu--baseen abantailak adieraztean, eta eutsiko diogu hurrengo batzuetan ere), datu-baseak eta fitxategiak bereizten dituen funtsezko ezaugarria da.

Basera bilduta dagoen datu-multzoaren definizioak edo deskribapenak (datu-basearen egitura edo eskema deritzo horri) bakarra izan beharko dute, eta datu berberekin egon beharko dute integraturik. Fitxategietan oinarri hartzen duten sistemetan, fitxategietan bildurik egoten dira datuak, eta berorien deskribapena alde batera egoten da, programen atala izaten baita. Horretarako, lengoaiek datuak deskribatzeko bitartekoak eman behar izaten dituzte. Dokumentazio gehigarria ere izaten da, gainera Euskarria papera izan ohi da, eta gehienetan ez da behar bestekoa izaten, eta ez da eguneratuta egoten. Horrelako antolakuntzak hamaikatxo arazo sorrarazten du; izan ere, batzuetan ez da jakiten fitxategi jakin baten deskribapena zein den, galdu egin delako edo, bestela, dagozkion agiriak behar bezala eguneratu ez direlako, eta tratatzen duen programa ere zehatz-mehatz ezagutzen ez delako. Datu-baseetan, berriz, deskribapena eta kasu batzuetan definizio eta dokumentazio osoak (metadatuak) ere bai, datuekin batera biltzen dira eta, horrela, datuak autodokumentaturik egoten direnez, dokumentazio horretan gertatzen diren aldaketa guztiak sisteman islaturik eta bildurik geratuko dira, horrek dituen abantaila guztiekin.

Datuak eguneratzeko eta berreskuratzeko, DBKSren barne daudelarik, behar bezala zehazturiko prozesuak erabiliko dira. Datuen segurtasuna (konfidentzialtasuna, erabilgarritasuna eta integritatea) mantentzeko diren tresnak ere eman beharko ditu DBKSk.

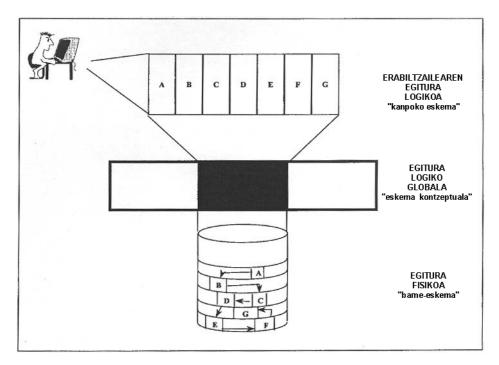
Urteak joan, urteak etorri, datu-baseen kontzeptua aldatuz eta konfiguratuz joan da; gaur egun, eta analizatu berri ditugun ezaugarri horiekin bat etorrita, honakoa izan liteke datu-basearen definizioa:

"Bigarren mailako euskarrian (ez baita aldakorra) eta erredundantzia kontrolatuaz integraturik, bildurik dauden datuen bilduma edo gordailua. Hainbat erabiltzailek eta hainbat aplikaziok konpartituko dituzte datuak eta beraien artean independente iraun behar dute. Beren definizioa (datu-basearen egitura) bakarra izango da eta datuekin batera egongo da bildurik; datuen eredu batean oinarrituko da, datu-eredu horrek emango dio mundu errealean dauden elkarren arteko loturak eta murrizketak atzemateko bidea. Eguneratzeko eta berreskuratzeko prozedurak komunak eta ondo zehaztuak izango dira, eta datuen multzoaren segurtasuna erraztuko dute".

Datu Baseak Kudeatzeko Sistema (DBKS) datu-baseak ezartzeko, bertara iristeko eta mantentzeko bidea ematen duten programen multzoa da. DBKS izango da, datu-basearekin eta erabiltzaileekin batera, Datu Basearen Sistema osatzen duena.

2.2 Datu-baseen abstrakzio-mailak

ISetan bi egitura ezberdin daudela ikus daiteke; logikoa legoke batetik (erabiltzailearen ikuspuntua) eta fisikoa bestetik (datuak nola biltegiratuta dauden). Datu-baseetan hainbat izen hartu izan dituen beste abstrakzio-maila bat ageri da: maila kontzeptuala, logiko globala, eta abar. Tarteko egitura horren bidez, bi mailako arkitekturaren egitura logikoaren eta fisikoaren artean tartejarriko den datuen irudikapen globala egin nahi da, eta independentea izango da, bai ekipoarekiko eta bai erabiltzaile bakoitzarekiko (ikus 2.1. irudia).



2.1. irudia. Datu-baseen sistemen hiru egiturak.

Erabiltzaile bakoitzak era partikularrean datu-baseari dagokionez daukan ikuspegia da erabiltzailearen egitura logikoa edo kanpoko eskema; egitura logiko kontzeptuala (eskema kontzeptuala ere baderitzo), berriz, enpresaren multzoaren ikuspegiari dagokio, eta egitura fisikoa (edo barne-eskema), azkenik, datuak biltegiratze fisikoan antolatzeko era izango da. Datu-basea hiru maila horietan egituratzearen helburu nagusia datuen eta aplikazioen arteko independentzia lortzea izango da.

Gainera, datu-basera jotzen duten pertsonak mailakatu nahi direnean, hiru talde ezberdin izan daitezke. Lehenik eta behin, administrazio-lanak egiten dituztenak daude. Bigarren taldean aplikazioen eta sistemaren programatzaileak daude. Azkenik, erabiltzaileak ditugu. Datu-baseen konplexutasuna ezkutatzeko eta datuen eta programen independentzia erdiesteko definitu dira arkitekturan edo diseinuan datu-baseetara joko duten hiru pertsona-mailei dagozkien hiru ikuspegi edo abstrakzio-maila hauek: fisikoa, kontzeptuala eta kanpokoa.

Erabiltzailearen egitura logikoa: kanpoko eskema

Kanpoko mailak, ikuspegiaren abstrakzioari edo erabiltzaileari dagokionak hain zuzen, erabiltzaileek atzematen dioten bezala deskribatuko du datu-basea. Erabiltzaileontzat, fisikoki daude taulak eta beren erregistroak.

Maila honetan, datu-baseen segurtasunaren eraginez, erabiltzaile bakoitzak erabil dezakeen datu--basearen atala ikusi ahal izango du. Erabiltzaile bakoitzak datu-base bat, kanpoko eskema bat ikusiko du, eta ezberdina izango da eman zaion iriste-mailaren arabera; izan ere, behar dituen datu eta lotura guztietarako irispidea eta definituta dauzkan mugak izango baititu.

Erabiltzaile batek euts diezaiekeen objektuek osatuta dago bere kanpoko maila: taulek, ikuspegiek, formularioek, txostenek eta abarrek. Hau da, kanpoko maila erabiltzaileak datu-baseari dagokionez daukan ikuspegia izango da eta, horrela, datu-baseak dituen adina erabiltzaile izango dira dauden kanpoko mailak. Erabiltzaileak kanpoko maila areagotu ahal izango du datu-basean elementu berriak sorraraziz gero.

Kanpoko maila, erabiltzaileen ikuspuntu funtzionaletik behatuta, kontsulten, formularioen eta txostenen ikus-garapenerako tresneria eta datu-basetik datuetara iristeko bidea emango duten programen sorrerarako balioko duen kontsulta-lengoaia izatean datza. Ikuspegia ere baderitzo kanpoko mailari.

Egitura logiko globala: eskema kontzeptuala

Maila kontzeptualak edo maila logiko globalak datuek datu-basean duten antolaketa eta datuen artean dauden loturak deskribatuko ditu. Maila honen deskribapena eskema kontzeptualaren bidez gauzatuko da, eta honakoak definitzeko bidea emango du. Baliozkotze-arauak, taulen arteko loturak eta eremuak eta berorien ezaugarriak. Datu-baseen programatzaileak izango dira aurreikusita dagoen informazioa, zeina aldez aurretik analistek definitua izan baita, sorrarazteko eta gordetzeko ardura dutenak. Gainera, beraien ardura izango da datu-basearen egitura logikoak -taulak- egokiro sorraraztea. Maila honetan sortuko dira segurtasunaren kontrolerako aginduak eta eremuetarako baimentzen diren murrizketak edo balioak.

Maila kontzeptualean, programatzaileen ikuspegi funtzionaletik, sistemarako deiak egingo dira datuen manipulatze-lengoaiaren edo Data Management Language (DML) delakoaren bidez, eta bertatik iritsiko dira datu-basearen eskemara datuen definizio-lengoaia edo Data Definition Language (DDL) deritzona eta datuen kontrolerako lengoaia edo Data Control Language (DCL) izenekoa erabilita

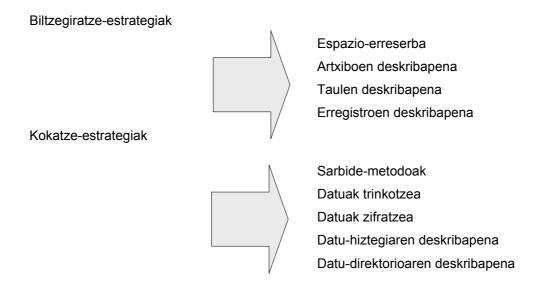
Maila honetan, aurreko lengoaiak ez ezik, datu-baseen kudeatzailean ikus-garapenerako inplementaturik dauden tresnak ere erabil daitezke.

Egitura fisikoa: barne-eskema

Datuen egiturak fisikoki nola biltegiratzen diren deskribatuko du maila fisikoak; taulez gain -biltegiratze logikoa- artxibo fisikoak ere definituko dira; artxibo bati edo gehiagori dagozkion eta taula bat edo gehiago dauzkaten taula-tarteak; artxibo-multzoak eta datu-orrialdeak. Maila fisikoaren deskribapena barne-eskema baten bidez gauzatuko da. Taulak eta taulak ekarrekin nola lotzen diren definitzeko bidea emango duen definizio- eta arau-multzoa da barne-eskema.

Datuak biltegiratzeko gailuetan antolatzeko eraren ikuspuntutik, honako alderdi hauek bereiztuko dira:

- ✓ Biltegiratze-estrategia edo datuen multzorako tarteak ematea; datuak biltzeko beharrezkoa den espazioa gordetzea; datu-basearen artxiboen deskribapena: izena, antolaketa-mota, biltegiratzen diren unitatea edo bolumena eta irispiderako metodoa; erregistroak dauzkaten datuen taulen deskribapena; erregistroen definizioa, luzera eta eremu bakoitzeko datu-mota.
- Kokatze-estrategia erantzunaldiak eta bigarren mailako memoria-tarteak optimizatzeko; jakineko sarbide metodoak: gakoen, indizeen eta erakusleen zehaztapena eta hiztegiaren eta datuen direktorioaren deskribapena, baldin eta sistemak zeregin hori automatikoki gauzatzen ez badu. Gainera, kontuan izango dira datuak konprimitzeko teknikak, kriptografiatzekoak edo datuak zifratzekoak eta barne-eskemaren eta eskema kontzeptualaren artean dagoen lotura, bai eta doitzeko eta fintzeko teknikak ere. Maila honetan deskribatuko dira datuak eusten dien makinaren ikuspuntutik.



2.4. irudia. Datuen deskribapen-maila: maila fisikoa.

Maila honetan esku hartzen duten pertsonek administrazio-lanak egiten dituzte: datu-baseen administratzaileen gainean kokaturik dagoen eta eskema kontzeptualaren arduraduna den enpresa--administratzailea; datu-basearen administratzaileak eta aplikazioen administratzaileak, beroriek baitira beraien ardurapeko aplikazioen kanpoko eskemak prestatzeko ardura dutenak. Hiru era horietako administratzaileak pertsona ezberdinak izan daitezke, edo pertsona bakarra.

Hiru maila horien integrazioa transformazioen bi mailen bidez gauzatuko da:

- ✓ Alde batetik, datuak kanpoko irudikapenetik kontzeptualera aldarazten dituen kanpoko transformazio kontzeptuala dago; kanpoko maila jakina duen erabiltzaileak, kontsulta baten bidez datu jakin batzuetarako sarbidea eskatzen duenean. Datu-basearen kudeaketarako sistemak eskaria interpretatu, erabiltzaile horrentzat definitua dagoen kanpoko eskema egiaztatu eta eskaria kanpoko mailatik maila kontzeptualeraino aldaraziko du.
- ✓ Eskema kontzeptuala egiaztaturik dagoela, eskaria maila kontzeptualetik barne-mailara pasatzen denean gertatuko da bigarren transformazioa. Barne-maila izango da eskaria egitearen zio izan den datu-biltegiratzearen egitura hautatu eta erabiltzailearen eskariari erantzuna emateko beharrezkoak diren eragiketak gauzatuko dituena.

Alderantziz errepikatuko da prozesu hori kontsultaren xede diren erregistroei utzi ahal izateko. Maila bat beste maila bat bihurrarazten duen prozedura-multzoa korrespondentzia-arauak edo maparaketa izenez ezagutzen da.

Laburbilduz, datuak eskema batetik bestera aldarazteko faseak honako hauek dira:

- Erabiltzaileak taula bateko datu jakin batzuk ikusteko eskaria egingo du, kontsulta sorraraziz.
- 2. Erabiltzaile horrentzako kanpoko eskema egiaztatu egingo da.
- 3. Kanpoko eskemaren onarpena.
- 4. Eskariaren transformazioa maila kontzeptualera.
- 5. Eskema kontzeptualaren egiaztapena.
- 6. Eskema kontzeptualaren onarpena. Kontsultaren bidez eskaturiko datuak dauzkaten taulak erabiliko dira.
- 7. Eskariaren transformazioa barne-mailara. Artxibo batek edo gehiago osatutako objektu logikoak diren taula-tarteetan gauzatuko da bilaketa, beroriek baitaukate datuen biltegiratze fisikoari buruzko informazioa, taula bakoitzean hautaturiko erregistroei dagokienez.
- 8. Kontsultaren xede den taula hautatzea.
- 9. Kontsulta burutzea.
- 10. Barne-maila maila kontzeptual bilakaraztea.
- 11. Maila kontzeptuala kanpoko maila bihurraraztea.
- 12. Erabiltzaileari dagozkion erregistroak erakutsiko dira.

Datu-baseak eta datu-baseak kudeatzeko sistemak 2.3

Aurreko ataletan datu-baseak eduki behar dituen eta ohiko biltegiratze-sistemetatik bereizten dituen ezaugarriak deskribatu dira.

Datu-basea zer den eta Datu Baseak Kudeatzeko Sistemak zer diren bereiztea garrantzitsua da; izan ere, bi termino horiek nahastu egiten dira sarri askotan teknologia hau erabilita, informazioaz lan egiten denean.

Orain arte datu-baseen terminoa besterik ez dugu erabili. Datu-baseei buruz dihardugunean, kapitulu honetan azaldutako ezaugarri eta murrizketa guztiak betetzen dituelarik, biltegiratuta dagoen informazioari buruz ari gara.

Informazioa deskribatu den bezala biltegiratu ahal izateko eta informaziorako sarbideak datu-base bati, izen hori eman ahal izateko, exijitzen zaizkion ezaugarriak bete ditzan baina, beharrezkoa da zeregin hori burutu ahal izango duten prozedurak -software sistema- izan daitezen. Software sistema horri deritzo Datu Baseak Kudeatzeko Sistema. Horregatik, datu-basearen erabiltzaileari ondoren zerrendatzen diren egitekoak gauzatzeko beharrezkoak diren baliabideak emango dizkioten aplikazio-programen bilduma da DBKS:

- ✓ Datuen definizioa zenbait abstrakzio-mailatan (fisikoa, logikoa eta kanpokoa).
- ✓ Datuen manipulazioa datu-basean. Hau da, datuak txertatzea, aldatzea, ezabatzea eta berorien sarbide edo kontsulta.
- ✓ Datu-basearen segurtasuna mantentzea. Segurtasuna datuei beraiei dagokienez, bai eta beraien balioei eta beraien arteko loturei dagokienez ere.
- ✓ Datuen pribatutasunaren eta segurtasunaren kontrola datu-basean.
- ✓ Eta, azken batean, datu-base bati exiji dakizkiokeen ezaugarri guztiak ezartzeko beharrezkoak diren bitarteko guztiak.

DEFINIZIOA	KONTROLA	MANIPULAZIOA
CREATE		SELECT
DROP	GRANT	UPDATE
CHECK	REVOKE	INSERT
CONSTRAIN		DELETE

2.7. irudia. Datu Basea Kudeatzeko Sistemaren funtzioak.

Datuak definitzeko funtzioa: Funtzio honen bidez, datu-basea osatzen duten elementuen arteko definizioa eta lotura gauzatu ahal izango dira. Datuak definitzeko lengoaiari (DDL) dagokio. SQL azpimultzo honetan nabarmen daitezkeen aginduak honako hauek dira: datu-basean objektuak sorrarazten edo ezabatzen dituzten CREATE eta DROP -beraien adiera anitzetan-, eta murrizketak eta segurtasun semantikorako arauak definitzen dituzten CHECK eta CONSTRAIN izenekoak.

Datu-baseen kudeatzaile zenbaitek beren lengoaia daukate datuak definitzeko.

Hiru abstrakzio-mailek hartzen dute parte funtzio honetan, zeren maila fisikoan datuen egituren baldintza fisikoak definitzen baitira: diskoan dagoen espazioa, orrialdeen edo taulen tarteen tamaina, disko-unitateak, taulak, eremuak, indizeak eta abar. Maila logikoan, berriz, espazioa gorde zaien eta maila fisikoan ezaugarriak definitu zaizkien objektuen sorrerarako bidea ematen dute, ikus-tresnetatik edo, bestela, datuak definitzeko lengoaiatik. Kanpoko mailan, azkenik, erabiltzaileak, rolak eta baimenak definitzeko edo ezabatzeko ardura daukate.

Hiztegiari / datuen direktorioari eta, horregatik, abstrakzio-maila batzuk beste batzuekin komunikatzen dituzten jarraibideak aldarazteari dagozkienak dira datuen definiziora bilduta dauden zereginetako beste batzuk.

✓ Manipulazio-funtzioa: Datuen manipulazio-funtzioak taula batean edo gehiagotan lan egiteko bide eman behar du eta, horrela, erregistro-multzoak ordenatu, indexatu, berregituratu, trinkotu edo erantsi edo ezabatu egingo dira, edo kontsulta global baten bidez beraiei buruzko informazioa berreskuratuko dira. Erregistro bakoitzaren gainean banaka gauzatzen diren eragiketei ere eragingo die funtzio honek: altei, eguneratzeei eta ezabaketei. Ikus daitekeenez, datu-baseetan datuak ageri diren unetik aurrera egongo da funtzio hau.

Administratzaileak eta programatzaileek erabiltzen SQL-ren datuen kontsultarako eta manipulaziorako lengoaiei (DML) dagokie, nahiz eta kanpoko mailan aldez aurretik definiturik daudelarik, erabiltzaileei kudeaketa erraztuko dieten menuak eta transakzioak erabil daitezkeen. Funtzio honetan gehien bat esku hartzen duten SQL aginduak honakoak dira: SELECT, INSERT, DELETE eta UPDATE.

Kontrol-funtzioa: Ez da ondo definiturik dagoen eta literaturan nortasun propioa duen funtzioa, zeren, zenbait idazleren arabera, aurreko bi funtzioetan diluiturik baitago. Beraren existentzia baliagarritzat hartzen baldin bada, funtzioen bikoiztasunean pentsatu beharra dago; alde batetik erabiltzaileen kudeaketari dagozkionak leudeke eta, bestetik, berriz, sistemaren administrazioarekin lotuta daudenak. Baimenak ematearen edo erabiltzailearen profilen sorreraren ikuspuntutik, kontrolerako lengoaiaren SQL aginduak -GRANT edo REVOKE- datuak definitzeko funtziora bildu beharreko aginduak dira, baina administratzaileek era erabiltzaileak baimen horiek aldaraz eta ezaba ditzaketenez gero, kontrol-funtzioaren atala ere badira.

Erabiltzaileen kudeaketari dagokionez, baimenak kontuan hartuta, datu-basearen kudeaketa-sistemaren elementuetarako sarbidea ematen dieten erabilerek -interfazeekosatuta dago funtzio hau.

Datu-basearen kudeaketari dagokionez, datu-basearen funtzionamendua monitorizatu egin behar dute administratzaileek: estatistikak, gaitasunak, erabilitako espazioak, zatikatzea. Aplikazioetarako edo datu-baseetarako enpresa-mailan exijitzen diren segurtasun-mailak kontrolatu beharra ere badute: sarbideen kontrola, sareekiko konexioen kudeaketa, erabiltzaileena, transakzioena, berreskuratze-kopiak eta abar.

2.4 DBKS-en osagaiak

Aurreko atalean deskribatu diren funtzio guztiak eta beste zenbait gauzatu ahal izateko, beharrezkoa da DKBSk funtzio horiek garatzeko eginkizuna duten osagaiak edukitzea, sistema horietarako ezarrita dauden eskakizunak asetzen dituelarik.

DBKSen "osagaiak" aipatzen direnean, apika arriskutsua izan daitekeen orokortzea egin ohi da, zeren osagai horiek mota askotakoak izan baitaitezke. Adibidez, DBKSk software tresnak izango ditu alde batetik, eta informazioa behar bezala kudeatzeko zereginetan eta ekintzetan espezializatuak diren giza langileak bestetik.

Datuak definitzeko lengoaia (DDL)

Datuen independentzia bermatzeko haiek abstrakzio-maila ezberdinetan definitu beharra baldin badago, beharrezkoa izango da, ondorioz, DBKSk zeregin hori gauzatzen utziko duen osagaia edukitzea. Datuen irudikapen logikoa egiteko bidea emango duen datu-eredu jakin batean oinarri hartzen duen lengoaia artifiziala da datuak definitzeko lengoaia –Data Definition Language (DDL) –.

Gehienetan, DBKSen DDLk morfema-multzo mugatua duen gramatika xumean oinarritzen diren lengoaia sinpleak izaten dira. Horrela datuen definizioa anbiguoa ez izatea bermatuko da. Definizio horrek metatua izan behar du, "makinari" begira dagoen irudikapena sorrarazteko, eta horixe izango da DBKSk prozesu-aldian erabiliko duena.

Prozesu honen bidez lortzen diren datuen irudikapena DBKSk duen eta Datuen Hiztegia deritzon beste osagai batean bilduko da.

Datuen biltegiratzea definitzeko lengoaia (DSDL)

DBKS gehienetan DDL lengoaiari berari esker defini daitezke datuak irudikapen fisikoaren mailan, nahiz eta beste batzuetan datuen biltegiratzea definitzeko lengoaia - Data Storage Definition Language (DSDL)- deritzon azpiosagaia den. Nolanahi ere, DDL edo berorren azpiosagaia den DSDL erabiltzen delarik ere, arazo bat abstrakzioko bi mailetan menderatzeari dagozkion datuak definituko dira, eta datuen definizio horri Datu Basearen Eskema deritzo.

Datuak definitzeko lengoaiaren bidez arazo bat menderatzeari dagozkion datuen irudikapena litzateke datu-basearen eskema. Lengoaia horrek datuen eredu batean hartzen du oinarri. Honakoak definituko dira eskeman:

Arazoaren ezaugarriak deskribapen logikoaren mailan. Arazoa aldatu izan ezik, definizioa hau ez da aldatuko.

- ✓ DBKSren bidez tratatu nahi den arazoaren eremua edo sistema osatzen duten objektu--mota guztiak eta beren ezaugarriak.
- ✓ Objektu-mota hauen artean dauden lotura guztiak eta beren ezaugarriak.
- ✓ Hala objektu-motei nola ezaugarriei eta beraien arteko loturei dagozkien murrizketa guztiak.

Arazoaren ezaugarriak ikuspuntu fisikotik edo eragiketa-ikuspuntutik. Denboraren poderioz alda daiteke deskribapen hau, eragiketa-beharrizanen edo errekerimenduen arabera eta, esate baterako, honako hauek hartuko ditu aintzat:

- ✓ Datuak biltegiratuko diren unitate fisikoak.
- ✓ Erabiltzen diren bolumenak eta artxiboak.
- ✓ Biltegiratze-bitartekoak eta informaziorako sarbide-metodoak: clusterrak, blokeak, indizeak, taulak eta abar.

DSDLaz gain, datuen kontrolaz eta segurtasunaz arduratuko duen azpilengoaia edukiko du DDLk. Azpilengoaia horri datuen kontrolerako lengoaia esaten zaio -Data Control Language (DCL)- eta datuen hiztegira bilduta dagoen informaziorako sarbidearen kontrola (pribilegioen definizioa eta sarbide-motak) eta datuen segurtasunerako kontrola gauzatzeko bidea ematen du.

Datuen manipulazio-lengoaia (DML)

Datuen manipulazio-lengoaia -Data Manipulation Language (DML)- dugu DBKSetako osagai funtsezkoetako beste bat. Datuen kudeaketan oso ezberdin diren bi funtzio gauzatzen dituen lengoaia artifiziala da DML:

- ✓ Datuen kanpoko edo erabiltzailearen mailaren definizioa.
- ✓ Datuen manipulazioa; hau da, datuak txertatzea, ezabatzea, aldaraztea eta datu-basera bilduta dauden datuak berreskuratzea.

DDL bezala, DML ere datu-eredu batean dago oinarriturik eta, ondorioz, datu- ereduetan oinarri hartzen duten DBKSek datu-baseak kudeatzeko sistemaren DML berariazkoa izan dezakete edo, bestela, SQL erabili DML gisa.

Lehenago esan denez, datuen kanpoko ikuspegia deskribatzeko funtzioa ere badu DMLk. Hain justu ere, erabiltzaileek DDLren bidez definituriko datu-basearen eskemari dagokionez erabiltzaileek dituzten "ikuspegiak" edo zati bateko ikuspenak definituko dira DMLren bidez. Azpieskema deritze datuen ikuspegi horiei, eta hainbat eratara gauza daitezke:

- ✓ DML, eta besterik ez, erabilita. Horrela, lengoaia honek bere dituen sententziak erabiliz, datu-basearen eskemaren zati bateko ikuspegiak -azken erabiltzaileari begira daudenak- definituko dira. Zati bateko ikuspegi hauek (ikuspenak) datuen hiztegira bilduko dira gehienetan.
- ✓ C, COBOL, FORTRAN edo bestelako lengoaia ostalaria (host) erabilita. Lengoaia hori izango da erabiltzaileari datu-basearen datuak manipulatzeko bidea emango dioten aplikazio-programak egiteko erabiltzen dena. Programa hauen iturri den kodean DMLren sententziak daude, eta berorien ardura izango dira prozesu hauek, ostalari den lengoaia erabilita gauzatzen diren sententzien helburua informazio-fluxua kontrolatzea eta erabiltzailearen interfazea diren bitartean.

Beste lengoaia batzuetatik: C, COBOL, Ensamblador eta abarretik tauletara iristeko lengoaia ostalari gisara erabil daiteke SQL. Kasu honetan, iturri diren programak aurremetatu beharra dago dagokion metagailuak metatu baino lehen, makina-kodea sorrarazteko eta iturri den kodean murgildurik dauden DML sententziak ostalari den lengoaiaren metagailuak ulertuko duen kodera aldatzeko.

Datuen hiztegia

Datu-baseak biltegira daitezkeen datuei buruzko informazioa daukan artxiboa edo artxibo-multzoa izango da datuen hiztegia. Honakoak dira datuak: taulak, erregistroak, eremuak, loturak eta beren egiturak, edukiak, deskribapenak. Horrela, erabiltzaileek irakur dezaketen informazioa izanik, datu--baseetako bakoitzera bilduta dauden informazio-edukien egituraketa zein den jakin dezakete.

Datuen hiztegira daude bilduta DDLk DBKsren bidez tratatuko den arazoari buruz eginiko definizio guztiak eta DMLk eginikoetako zenbait (nahi direnak). Horiek horrela, honako hauek daude biltegiratuta datuen hiztegian:

- ✓ Datu-basearen eskema logikoa.
- ✓ Datu-basearen eskema fisikoa.
- ✓ Baina, gainera, datuen hiztegian askoz informazio gehiago dago bildurik. Informazio hori honako hauei dagokie:

- 1. Pribatutasun-murrizketei eta datu-basean biltegiratuta dauden datuetarako sarbide--eskubideei.
- 2. Datuen segurtasunarekin lotuta dauden erregelei, arauei edo murrizketei.
- 3. Datu-basean biltegiratuta dauden datuen segurtasuna bermatzeko bidea ematen duen bestelako informazioa.

Datu-basearen administratzailea

Datu-basearen administratzailea - Data Base Administrator (DBA)- dugu DBKSetako osagaietan beste bat. Giza osagaia da, datu-baseen erabilpenak arazo jakin bati irtenbidea ematean izango duen emaitzan berebiziko garrantzia daukana. Baditu erantzukizun batzuk DBAk tratatzen den informazioaren definizioari, administrazioari, segurtasunari, pribatutasunari eta segurtasunari dagokienez, bai eta DBKSk berorren prozesamenduan betetzen duen egitekoari dagokionez ere.

DBAri esleitutako egitekoen artean ondorengo hauek daude:

- ✓ Datu-basearen eskema logikoaren definizioa. Hau da, DBKS erabilita tratatuko den arazoaren ezaugarriak irudikatzen duten definizio-multzoari dagokionez DDLren sententzien bidezko kodetzea. Definizio honetara bilduta daude DBKSk datu-basean biltegiratuta dauden datuen segurtasunari eusteko beharrezkoak diren zehazpenak.
- ✓ Datu-basearen eskema fisikoaren definizioa. Hau da, biltegiratze-egituren zehazpena eta biltegiratze-gailu fisikoetan biltegiratuta dagoen informaziorako sarbide-metodoak. DSDL (edo DDL bera) erabilita gauzatuko da definizio hau, metatuak eta datuen hiztegira bilduko den eta makinak uler dezakeen zehazpen batera itzuliak diren sententzia-multzo baten bidez.
- ✓ Datu-basearen azpieskemen edo kanpoko ikuspenen edo erabiltzaile-ikuspenen definizioa. Hiztegira biltzen diren datu-basearen zati bateko ikuspenak DBAk definituak izango dira, berori baita sarbidea daukan bakarra eta, ondorioz, berorrek baino ez baitauzka osagai horren kudeaketarako beharrezkoak diren pribilegioak.
- ✓ Datuen pribatutasunaren kontrola, erabiltzaileei edo erabiltzaile-multzoei datu-baseetan biltegiratuta dagoen informaziorako sarbiderako pribilejioak emanez. Datu-basearen eskemaren arabera gauzatuko da zeregin hau eta datuak erabilita egin daitezkeen oinarrizko eragiketen arabera (kontsulta, txertaketa, aldarazpena eta ezabaketa), eta ekintza horietako baterako edo zenbaitetarako pribilejioak emango zaizkie datu--basearen eskeman definituta dauden datu-multzoei.

- Eskemak mantentzea. Horrela, hurrengo hauek DBAren ardurapekoak izango dira:
 - 1. Eskema logikoan beharrezkoak diren aldarazpenak sartzea; aldarazpen horiek DBKSk erabilitako arazoan izaniko aldaketaren batek edo berorren handiagotzeak eragindakoak izango dira.
 - 2. Datuen irudikapen fisikoan beharrezkoak diren aldarazpenak sartzea. Horrela, irudikapen hori datu-basearen hedapenarekiko eta errekerimendu funtzional edota jarduera-errekerimendu berriak sartzearekiko paraleloa izango da.
 - 3. Azpi-eskemen edo kanpoko ikuspenen edo erabiltzaile-ikuspenen aldarazpenak eta definizio berriak sartzea, datu-basearen ustiapen eraginkorra ekarririk.
 - 4. Datu-basean biltegiratuta dauden datuen segurtasunari eusteko beharrezkoak diren prozedurak zehaztea. Hau da, datuen aldi baterako galera eragiten duen akats baten ondoren, datuak berreskura daitezkeela bermatuko duten prozesuak noiz, nola eta zein eratan gauzatu eta definitu behar diren.

Datu-basearen erabiltzaileak

DBKSen beste osagaitzat hartu izan dira tradizioz datu-baseen erabiltzaileak, seguruenik sistema horiek erabiltzaileak berorien funtzionamendu zuzenaren, egokiaren eta beharrezkoaren atal garrantzitsutzat hartu zituzten lehenbizikoetakoak izan zirelako.

Jakina, horrenbesteko konplexutasuna duen sisteman sistemara jotzen duten eta sistemarekiko elkarreraginean diharduten mota askotako erabiltzaileak daude eta, horiek horrela, honako hauek har daitezke aintzat:

- ✓ Erabiltzaile terminalak: aplikazio-programen bidez datu-basearekiko elkarreraginean dihardute. Espezializatu gabeko erabiltzaileak izanik, burutzeko pribilegioak dauzkaten aplikazio-programek erabiltzen dituzten kanpoko ikuspenek edo azpieskemek ematen dituzten ikuspegia dute arazoari dagokionez.
- ✓ Erabiltzaile teknikoak: datu-basearen erabiltzaile terminalek erabiliko dituzten aplikazio-programak garatzen dituztenak dira. Hirugarren edo laugarren belaunaldietako lengoaiak erabiliz, arazoaren kudeaketan beharrezkoak diren eragiketak kudeatzeko, erabiltzaileari begira interfazetik deituriko prozedurak prestatzen dituzten profesional informatikoak izaten dira.
- ✓ Erabiltzaile espezializatuak: DBKS, konplexutasun handiagoko edo txikiagoko beste sistema batzuen garapenean tresna gisa darabiltenak dira.

✓ Erabiltzaile kritikoak: talde honen izendapena, "zakarra" izanagatik ere, eman dakiekeen egokiena da. Datu-baseen teknologiari edota elkarreraginean diharduten datu-baseari eusten dion DBKSri buruz, ezagupen tekniko asko izan dezakete, baina baita bat ere ez ere. Hori gorabehera, gehienetan aldez aurretik aurreikusita ez dauden formatuan eta xehetasunetan eta errekerimenduak erabilita (arazoaren analisi-prozesuan eta eskemaren diseinuan), eta ahalik eta denborarik laburrenean eskatzen diote informazioa datu-baseari. Datu-basea ezarri duten enpresetako gerentziak edo staff-eko kideak izaten dira eta, beroriek, kudeaketarako, administraziorako, merkaturako, marketingerako itxaropenen arabera edo, besterik gabe, interes pertsonala dutelako, aurreikusita ez dauden kontsultak egiten dituzte datu-basean biltegiratuta dagoen informazioari buruz.

2.5 Datu-ereduak: kontzeptua eta motak

Datuak, berorien arteko loturak, eragiten dieten segurtasun-mugatzaileak eta erabili beharreko terminologia deskribatzen dituen lanabes kontzeptualen multzoa izaten da datu-eredua.

Datu-base batek enpresarako garrantzizkoa den informazioa era egituratuan biltzen duela kontuan harturik, informazio hori biltegiratzeko erak mota bateko edo besteko datu-baseak eragingo ditu; izan ere, mundu errealetik hartutako informazioa egituraturiko datu bilakarazteko aldez aurretik izan baden abstrakzio- eta arau-multzoa erabiliko baita. Arauen eta abstrakzio-mekanismoen multzo guztiak egokituko zaizkio datuen eredu logiko bati.

Datu-baseen kudeaketarako sistemetan gehienetan erabiltzen diren hiru eredu logikoak honako hauek dira: hierarkikoa, sarekoa eta erlazionala.

Eredu hierarkikoa

Zuhaitz-eredua ere baderitzo eredu honi, alderantziz dagoen zuhaitzaren antzeko egitura erabiltzen delako. Sustraiak diseinuaren goiko aldean egoten dira irudikapenean. Sareko modeloaren modukoa da; izan ere, nodo izeneko erregistroen bidez irudikatzen baitira datuak, eta loturak, berriz, loturen, lotuneen edo arkuen bidez.

IBMk 70eko hamarkadaren hasieran erabilitako IMS (Information Management System) zelakoaren arkitekturaren eta jardunaren abstrakziotik eratorritakoa da eredu hau. IMSetik abiatuta, berorren egitura--ezaugarriak kopiatuko dituzten datu-baseen sistema kudeatzaileak sortuko dira. Gainerako datu-base hierarkikoen erreferentziarako estandartzat har daiteke IMS.

Honakoak dira ezaugarri nagusiak:

- ✓ Zuhaitz-multzo batek datu-basea osatuko du.
- ✓ Nodo edo segmentu deritze erregistroei, eta beraien atributu diren eremuak dauzkate.
- ✓ Mailatan antolatuta daude nodoak. Nodo bakoitzak berekin lotuta dauden nodo-kumeei dagozkien eremuak dauzka. Hierarkian edo zuhaitz-egituran altuen den nodoari sustrai deritzo.
- ✓ Beheragoko mailetako beste nodo batzuekin lotuta dagoen nodoa izango da aita. Nodo bat ezin izan daiteke bere buruaren aita.
- ✓ Aita bereko nodoak dira bikiak.
- ✓ Goragoko mailetako nodo batekin (batzuekin) lotuta daudenak dira kumeak. Aita bereko kume guztiak maila berean daude. Sustrai ez diren nodo guztiek dute aita eta aita diren nodo guztiek izan ditzakete zenbait kume.
- ✓ Nodoak lotzen dituzten lerroak dira bideak. Bi nodoren artean bide bakarra egongo da.
- ✓ Kumerik gabeko nodoak hostoak dira.
- ✓ Bukaera gisa hostoa duten bideak edo lerroak dira adarrak.
- ✓ Zuhaitzaren altuera, zuhaitzak duen maila-kopurua izango da.
- ✓ Hostoen kopurua izango da zuhaitzaren pisua.
- ✓ Zuhaitzak guztira dituen nodoen kopurua izango da momentua.
- ✓ Aita den nodo baten eta berorren kumeen guztirako batuketa eginda lortzen den zenbakia izango da familia.
- ✓ Aurreko kontzeptuetatik ondoriozta daitekeenez, sustrai den nodoaren kume batek, kume dituen nodoek eta kumeen kumeek, osatzen dituzten adar guztiak kontuan hartuta, osatzen duten multzoa izango da erregistroa.
- ✓ Arkuen edo begizten bidez adieraziko dira erregistroen arteko loturak. Ezin adieraz daitezke erregistroen arteko N:M erako loturak, ez eta beraien arteko lotura erreflexiboak ere.

DMLri esker banaka-banaka hauta daitezke erregistroak. Erregistroak hautatzeko, ordena hierarkikoa erabilita ibiliko gara zuhaitzean barrena. Aldez aurretiko ordenan gauzatuko da berori, sustrai den nodotik hasi eta, gero, bertara sartu gabe dirauen kume ezkerrekoenera jota. Bisitatu gabeko ezkerrekoen den aitarengana itzuliko gara ondoren eta berorren kumeetan barrena ibiliko gara, ezkerretik eskuinera, eta horrela geroago ere. Aldez aurretiko ordenan erregistro bat hautatzen den bakoitzean, informaziora sartzeko bidea izango dugu eta erregistroa eguneratu egingo da, ezabatzeko aukera izango dugu edo beste nodo bat sartzekoa. Nodo bat ezabatuz gero, kume dituen nodo guztiak ere ezabatuko dira.

Sare-modeloa

Sistemari aldarazpen gehiegi eskatu ezean, oraindik ere indarrean dirauen modeloa da, nahiz eta 70eko hamarkadako merkataritza-sistemek berez zuten datu-eredua izan.

Sare-ereduetan, nodoen bidez -erregistroak- adieraziko dira datuak, eta datuen arteko loturak, berriz, arkuen edo begizten bidez.

Mota guztietako loturak adierazteko bidea emango du eredu honek, eredu hierarkikoan ezin bete daitezkeen N:M loturak eta lotura erreflexiboak barne. Gainera, bi nodoren artean lotura bat baino gehiago izan daiteke eredu honetan.

Modelo honen ezaugarrietan, ondorengo hauek azpimarra genitzake:

- Elementu gisa definituko da berez independentea eta esangarri den informazio-unitaterik txikiena.
- ✓ Izena ematen zaien elementu-multzoa izango da datu-agregakina.
- ✓ Eremu-multzoa da erregistroa.
- ✓ Eremu guztiek dauzkate elementuak.

Eredu erlazionala

E.F. Coddek –eredu erlazionalaren aita da bera– idatzi zuen jatorrizko "A Relational Model of Data for Large Shared Data Banks" artikulua, Communications of the ACM delakoaren 13. alean argitaratu zen, 1970.eko ekainean, 377-387. orrialdeetan. Bertan, eredu erlazionala definitzean bi kontzeptu bereiztu zituen: relation eta relationship hain zuzen. Lehenengo kontzeptuak, taularen edo entitatearen ideiari dagokion Teoria Erlazionalaren elementu osatzaile gisa aipatzen du erlazioa. Bigarrena, berriz, bi taularen arteko erlazioari (asoziazioari) aplikatzen zaio. Beti ere eremuen arteko erlazioak daudenez, ezin bereiz daiteke eremu bat noiz den taula batekoa edo beste batekoa eta, horregatik, kontzeptu bata zein bestea erabiliko da eta gero erantsiko zaizkie lehen mailako gakoaren eta besteren gakoaren kontzeptuak. Horrela bereiztu ahal izango dira beraietako bakoitzari dagozkion taulak eta eremuak.

Relationship kontzeptua itzultzeko, elkarren arteko erlazioa erabili ohi da zenbait testutan, eta erlazio hitza loturen edukiontzia (taula) adierazteko gorde ohi da. Artikulu horrexetan esaten zaie tuplo erlazioek (taulek) osatzen dituzten elementuei (erregistroei), eta tuplo bakoitza hainbat domeinuk (eremuk) osatuta dago. Kontzeptu horien itzulpenen bikoiztasuna justifikatutakoan, erlazioa, taula, erregistro eta eremu hitzak erabiliko dira testuan zehar.

Codd-ek beste ikertzaile batzuekin batera gauzatu zuen esfortzuak merkataritza-datuetarako hainbat base sorrarazi zituen, eta berorien artean IBMren DB2 eta Oracle izeneko datu-basea nabarmendu ziren, beroriek izan baitziren Codd-ek datu-base erlazionaletarako proposatu zituen ezaugarriak arrakastaz inplementatu zituzten lehenbizikoak.

Erlazioak dauzkan array baterako Codd-ek definitu zituen ezaugarriak ondorengo hauek dira:

- ✓ Lerro bakoitza erlazioaren tuplo baten irudikapena izango da.
- ✓ Lerroen ordenak ez dauka esangurarik.
- ✓ Lerro guztiak dira ezberdinak.
- ✓ Zutabeen ordenak badauka esangura, eta erlazioa definitzen duten domeinuen ordenari dagokio.
- ✓ Zutabe bakoitzaren zati bateko esanahia zein den jakin daiteke domeinu bakoitzaren izena daukaten labelen bidez.

Codd-i jarraiki, esan genezake eredu erlazionalaren abantailak, datuek duten independentzian dautzala, aplikazioekiko eta biltegiratzeko gailuarekiko independentzia duten aldetik (ez dio axola datu-basea asko ala gutxi hazten den) eta datuen euskortasun ezaren ezabaketari dagokionez. Programen arteko independentziaren errendimendua, programok makinaren lengoaian irudikatzea eta datuen antolaketa areagotuko dituzten goi-mailako programak erabiltzeko aukera izango da abantailetan beste bat.

Bada beste abantailarik: eredu erlazionala dugu datuen karakterizazioari, erredundantziari eta sendotasunari dagokienez agertzen diren arazoak tratatzeko funtsa. Azkenik, esan beharra dago eredu honek datuak erabiltzaileei erakusteko era eta datuen irudikapen logikoa sinplifikatu egiten dituela.

Datuen eredu erlazionalari esker, predikatu erlazionalen kalkuluan oinarri hartzen duen datu--lengoaia unibertsala sorraraz daiteke. Lengoaia horren potentzia (maila operatiboan SQLn zehaztuko da, SEQUELen eta SQELen eboluzio gisa), Era Normalean egoteko datuen egiturak aldarazita daudenean geratuko da agerian. Lengoaia hori anfitrioi den lengoaian txerta daiteke, eta komandoen interfazean, prozeduretan edo aplikazioetan erabil daiteke.

Diseinu logikoa zen Codd-entzat garrantzitsuena, eredu abstraktua alegia, eta, horregatik, ez zuen ezer adierazi eredu hau merkataritzaren alorreko datu-baseetan inplementatzeari buruz.

Eredu erlazionalaren inplementazio fisikoari dagokionez, berriz, sistemak espazioa eduki behar du diskoan eta behar adinako memoria; izan ere, baliabide anitz kontsumitzen dituzten datu-baseen kudeaketarako sistemak baitira.

Erakutsi dira lehenago eredu erlazionalean esku hartzen duten elementuak. Era egituratuagoa erabiliz gero, ondoren adierazten den eran defini liteke:

- ✓ Taula, matrizea edo arraya: lerroek eta zutabeek osatzen duten eta entitate baten datuak biltegiratzeko balio duen egitura.
- ✓ **Tuplo** deritzon lerro bakoitzari erregistro bat dagokio. Hortaz, objektu berari dagokion erlazioa osatzen duten eremuen multzoa izango da tuplo.

Honako baldintza hauek bete behar ditu tuploak:

- ✓ Ezin egon daitezke eremu guztiak berdinak, bikoiztuak, dituzten bi erregistro.
- ✓ Mundu errealeko objektuekin daude lotuta.
- ✓ Entitatearen atributuak dira **zutabeak**, eta eremuekin daude lotuta.
- Zutabe baterako edo gehiagorako izan daitezkeen balio guztien multzoa da domeinua.

Bi motako domeinuak daude: jarraituak eta diskretuak. *Domeinu jarraituek*, emaniko bi mugaren artean izan daitezkeen balio guztiak bilduko dituzte. Esate baterako, zenbatekoen eremurako domeinua bi mugaren arteko edozein balio baliagarri izango da, baldin eta muga horiek definituta badaude. *Domeinu diskretuek*, berriz, hautaturiko balio jakin batzuk edukiko dituzte. Esate baterako, sexuen eremurako domeinuak gizona edo emakumea balioak baino ez ditu izango.

✓ Taulen arteko eragiketa erlazionalak, kalkulu erlazionala, erabilirik, beste taula batetik edo beste taula batzuetatik edo ikuspegi batzuetatik abiatuta sorrarazten den taula berria izango da **ikuspegia**.

Taula erlazionalek honako baldintza hauek bete behar dituzte:

- ✓ Mota bateko lerroak baino ez dauzka.
- ✓ Lerro guztiek eduki behar dute zutabe-kopuru berbera.
- ✓ Erlazio baten (taula) maila, dauzkan eremuen kopurua izango da eta, horiek horrela, erlazio bitarra, hirutarra edo n-tarra dela esan daiteke, baldin eta bi eremuk, hiru eremuk edo n eremuk osatuta badago.
- ✓ Erlazio baten (taula) kardinaltasuna dauzkan tuploen kopurua izango da.

Lerroen eta zutabeen bidez irudikatuko dira taulak grafikoki. Lehenengo lerroak zutabe guztien deskribatzailea edukiko du. Gainerako lerroek taularen erregistro guztiak edukiko dituzte. Taularen azpian, taularen gainean edo goiko lerroan, zutabe batez diseinatutakoa bera, idatziko da taularen izena.

INFORMATIKAKO SISTEMEN ADMINISTRAZIOA

Honako hauek dira zutabeek dauzkaten ezaugarriak:

- ✓ Taula bakoitzak duen zutabe-kopurua finkoa da eta aldez aurretik egongo da zehazturik, nahiz eta aldaraz daitekeen.
- ✓ Zutabe bakoitza bakarra izango da, eta izen bakarra izango du kasuan kasuko taulan.
- ✓ Ezin egin daiteke zutabe bikoizturik.
- ✓ Zutabe bakoitzak balio bakarra edukiko du.
- ✓ Adierazten duen domeinura bilduta egon beharko du zutabe bakoitzaren balioak, baina balio NULUA izan daiteke. Horrelakoetan ezezaguna edo aplikaezina izango da.

3 DATU-BASE ERLAZIONALAK

Datu-baseen oinarriak azalduta dauden honetan, datu-baseen diseinuen arazo praktikoari egin diezaiokegu aurre. Mota guztietako ekipoetarako eman eta merkatuan nagusitu den DBKSEen hedapen handiak garamatza datu-base erlazionalen diseinua sakonago aztertzera.

Kapitulu honetan proposatuko dugun diseinu-metodologiak, hasteko, mundu erreala deskribatuko du lehenengo fasean entitatea/interrelazio (E/I E) ereduan, ondoren, gertatzen den E/E eskema eredu erlazionalera aldatzeko. Normalizazioari buruzko teoriak, eta berorri ere aurre egiten zaio (lehenengo hiru era normaletan baino ez, ordea) kapitulu honetan, eskema erlazional batek betekizun batzuk betetzen dituenentz egiaztatzen utziko digu, eta, horrela izan ezean, egokien irizten zaion era normalera eraman ahal izango dugu.

Diseinu-metodologien etapak

Azken urteotan, teknologiaren alorreko aurrerapenei esker, DBKSEen hedapen handia ari gara ikusten, eta edozein plataformatan euts dakieke bere produktuei, hasi superordenagailuetatik eta ordenagailu pertsonaletaraino. Halatan ere, eta hainbat ikerlarik eta gaiari buruzko azterketak egin dituzten adituek gauzaturiko esfortzua gorabehera, DB baten sorrera zeregin luzea eta nekeza da gaur egun ere.

Datu-baseen diseinuak berekin dituen zailtasun horiei aurre egiteko tenorean, prozedura ordenatuak eta metodikoak erabili behar dira datu-baseen diseinuaren metodologiaren esparruan.

Datu-baseak diseinatzeko prozesuan hiru fase nagusi bereiztuko ditugu:

- Diseinu kontzeptuala: enpresaren informazio-baliabideen irudikapen egokia du helburu, erabiltzaile edo aplikazio partikularrak direnak direlarik ere, eta ordenagailuaren eraginkortasunari buruzko ideiak alde batera lagata. Entitatea / Erlazioa Eredua (E/E E) da erabili ohi den eredua.
- Diseinu logikoa: aurreko etapan lorturiko eskema kontzeptuala aldaraztea du helburu, erabiliko den DBKSk eusten dion datu-eredura egokiturik. Guk Erlazio-Ereduari buruz jardungo dugu, baina era analogoan molda liteke diseinu logikoaren etapa hau datuen beste eredu batzuetara, Hierarkikora edo Sare-Eredura esaterako.
- ✓ Diseinu fisikoa: eskema logikoaren ahalik eta instrumentaziorik eraginkorrena lortzea du helburu.

Datuen ereduek datu-baseen diseinuan duten egitekoa aztertu genuenean, honakoa adierazi genuen: diseinuaren lehenengo etapan abantailak dituela eredu konbentzionalekiko independentea den eta mundu errealaren semantika bilduko duen eskema kontzeptuala izateak.

Datu-baseak diseinatzean agertzen den arazoa izaten da diseinugilearen eta erabiltzailearen arteko komunikazioa; azken horrentzat ezagun askoa da aplikazioaren domeinua eta gehienetan hori ez da horrela izaten diseinugilearentzat, baina askotan ez du jakiten era zuzenean adierazten eta are gutxiago era zehatzean. Diseinuaren lehenengo fasean E/E E erabiliz gero, errazagoa izango da, datuak egituratzeko teknikak ezagutzen dituen arren, aplikazioaren domeinua arrotz zaion informatikariaren eta erabiltzailearen arteko elkarrizketa. Azken horrek ondo ezagutzen du mundu erreala, baina ez da eredu konbentzional batek ezartzen dituen baldintzen pean deskribatzeko gauza izaten.

E/E E xumea, baina aldi berean behar besteko potentzia duena izanez gero, zalantzak argitzen eta modelatu beharreko diskurtsoaren unibertsoaren aldeak argi utziko dituen elkarrizketa ezarri ahal izango da erabiltzailearen eta diseinugilearen artean. Horrela, errazagoa izango da espezialistek erabiltzaileekiko lankidetzan jardun dezaten eta azken horiek partaidetza eraginkorra izango dute, eta protagonista ere izan daitezke, diseinugintzan.

Metodologiaren lehenengo bi faseak era eskematikoan irudika ditzakegu (diseinu kontzeptuala eta diseinu logikoa). Liburutegi baten diseinu-prozesua ageri da bertan; diseinugileak mundu errealari so egiten dio helburu batzuk aintzat harturik eta, hasierako etapan E/E eskeman oinarri hartuta, eskema kontzeptualera iritsiko da, eta berorri arau batzuk aplikatuko zaizkio, egitura erlazionala (taula-multzoa) bilakarazteko xedearekin.

Egitura erlazionala produktu zehatzak inposatu dituen mugetara eta eraginkortasunari dagozkion exijentzietara moldatuta dagoenean, izango dugu suertatzen den egituran liburutegia kudeatzeko beharrezkoak diren datuak biltegiratzeko aukera: liburuak mailegatzea, gai jakin bati buruzko artikuluen kontsultak, egileei buruzkoak eta abar.

Diseinu kontzeptuala: e/e e eredua

Chen-ek 70eko hamarkadaren erdialdean definitu zuen entitatea-erlazioa ereduaren oinarria, entitate izena ematen zaien objektuak eta beraien arteko erlazio izeneko loturak izatean dago. Enpresarentzat interesa duten izaki errealen abstrakzioak dira entitateak, eta deskribatzeko balio dituzten ezaugarrien bidez, atributuen bidez hain zuzen, definitzen dira.

Eredu honek ez du kontuan hartzen datuen inplementazio fisikoa. Datuen maila kontzeptuala baino ez da interesgarria, beren egiturek —entitateek— nahi den enpresa-errealitatea irudikatzeko aukera ematen dutelako. Aplikazio praktikoari dagokionez, berriz, eredu hau erabilita egin ohi dute analistek datu-baseen diseinua, eta eredu erlazionalera itzultzen dute merkataritzaren alorreko datu-baseen sistema kudeatzailean inplementatzeko; izan ere, datu-baseen egitura logikoa grafikoki irudikatzeko, entitatea-erlazioa eredua erabil baitaiteke.

Honako ezaugarri hauek ditu entitatea-erlazioa ereduak:

- ✓ Goi-mailako abstrakzioa mundu errealari dagokionez. Entitate deritzelarik, errealitatearen irudi diren elementuak sorrarazten ditu, bere esentzia eta ezaugarriak aldarazi gabe.
- ✓ Erregistroak, entitate bakoitzera bilduta dauden datuen multzoak alegia, bere abstrakzio-maila eta xehetasunekoa ondorengo erabilerarekiko independente dira, prozesu-abiaduraren biltegiratzeko mugekiko eta datu-basea inplementatuko zaion sistemarekiko.
- ✓ Biltegiratzeko euskarriaren independentzia fisiko horri esker, hala entitate-kopurua nola erregistroena hazi eta aldarazi egin daiteke.

Eredu honetan dauden arauak edo murrizketak, berriz, ondoren adierazitakoak dira:

- ✓ Lehen mailako klabea dago.
- ✓ Entitateek erlazio baten bidez egon behar dute lotuta.
- ✓ Ezin lot daitezke erlazioak beren artean.

Ereduaren elementuak

Honakoak dira eredu honetan dauden funtsezko elementuak: *entitateak* eta beren *atributuak, erlazioak* eta *domeinuak* edo balio-multzoak.

Entitateak

Enpresarentzat informazio interesgarria duelako, informazio hori gorde nahi zaion edozein pertsona, toki, gauza, kontzeptu edo gertaera da *entitatea*. Substantiboa edo izen arrunta erabilita izendatuko dira entitateak. Barruan entitatearen izena idatzita duten laukien bidez irudikatuko dira.



3.1. irudia. Entitate-moten irudikapena.

Egitura generikoari *entitate-mota* esango diogu eta entitate-mota horren errealizazio zehatzei, berriz, *agerraldi* edo *entitate-ale*.

Horrela, esate baterako, IRAKASLEA deritzon entitate-motak berekin lotuta duen predikatua "Gai edo arte batean diharduen edo irakasten duen pertsona" izanik, berari dagokion "Sánchez" alea dauka; izan ere, predikatu hori betetzen baitu.

Bi motako entitateak daude:

- ✓ Entitate sendoa edo erregularra: izan ahal izateko, ez dago beste entitate baten mende. Esate baterako: klub entitatea, izango bada, ez dago beste entitate batzuen mende. Jokalari entitateak, ordea, kluba izatea behar du bera izango bada, zeren ez baitago klubik gabeko jokalaririk. Beste adibide bat: pertsona entitatea sendoa da bezero, hornitzaile eta langile entitateekiko.
- ✓ Entitate ahula: Beste entitate baten mende dago, baldin eta izango bada. Aurreko adibideetan, jokalaria edota bezeroa entitate ahulak dira. Hautaturiko beste entitate batzuekiko izango da entitate bat sendoa edo ahula. Bezeroa pertsonarekiko izango da ahula, baina fakturarekin lotuz gero, ordea, sendoa izango da; izan ere, ezin baita fakturarik luzatu, baldin eta hartuko dituen bezerorik ez bada. Entitate hauek ertz bikoitza duen errektangeluaren bidez irudikatuko dira.



3.2. irudia. Entitate ahul baten irudikapena.

Atributuak eta klabeak

Entitate batek dituen eta informazioa gorde nahi zaien ezaugarri hautatuei deritze atributu. Izen esanguratsua eta entitatearen barruan bakarra eduki behar dute atributuek.

Entitatea-atributua diagramei esker, era sinbolikoan aurkez daitezke entitate baten atributuak. Irudikapenak egiteko era hori ezberdina izango da eredu batzuetan eta besteetan, eta guk erabiliko duguna *Designer/2000-*k eutsitakoa da.

PERTSONA
Izena
* Sexua
O Jaiotze data

3.3. irudia. Entitate-atributuen irudikapena.

- φ Derrigorrezkoa esan nahi du
- # Derrigorrezkoa esan nahi du eta identifikatzaile bakarra edo klabea (edo berorien atala) da.
- # Aukerakoa esan nahi du.

Atributuak eduki ditzakeen balioen multzoa da domeinua, eta baliozkotze komunei lotuta dago.

Esate baterako, **Pertsonen Soldata** edota **Proiektuaren Aurrekontua** atributuak DIRUA domeinuan dauden atributuak dira, eta bi hamartardun balio positiboak baino ez dituzte.

Entitate-adibide bakoitza era bakarrean identifikatzen duten atributuen multzoari esaten zaio **klabe** edo identifikatzaile. Erregistroan klabe bat dago gutxienez, atributu guztiek osatutakoa hain zuzen. Hona hemen zenbait adibide: jokalari entitatearen klabeak dira honako hauek osatzen duten eremuen multzoak: izenak, deiturek eta klubak. Dortsal eta klub eremuek osatuta dago beste klabe bat. Hona hemen beste bat ere: jokalariaren IFZ.

Klabe hautagaia gutxieneko klabea da, hau da, ahalik eta eremu-kopururik txikienak osatuta dagoena. Aurreko adibidean jokalariaren kirol-izena edota bere IFZ izango lirateke klabe hautagai.

Lehen mailako klabea, datu-basearen diseinugileak entitate-adibide bakoitza identifikatzeko hautatzen duen klabe hautagaia izango litzateke. Ondorengo hauek dira lehen mailako klabea definitzeko tenorean dauden murrizketak: ezin eduki dezake balio nulurik, ezaguna eta sortzen erraza izan behar du eta, denbora igarota ere, ez da aldarazi behar. Erregistroaren identifikatzaile bakartzat IFZ eremua aukeratu baldin bada, berori izango da lehen mailako klabea.

Hautatzen ez diren klabe hautagaiei aukerako klabea izena emango zaie. Adibide honetan jokalariaren kirol-izena atributua litzateke aukerako klabea.

Besteren klabea edo arrotza, berriz, entitatearekiko harremanak dituen beste entitate batean lehen mailako klabea osatzen duen atributua edo atributu-multzoa litzateke.

Erlazioak

Bi entitateren artean dagoen eta elkarren arteko ekintza sorrarazten duen lotura da. Lotura hauek izan daitezen, entitateek berek izan behar dute eta, gauzak horrela, ezin izan daiteke entitateekiko independente den erlaziorik. Modu infinitiboan dagoen aditza erabiliko da izena emateko, eta bi entitateen arteko lerro baten bidez irudikatuko dira.

Hiru ezaugarri ditu erlazio orok: era bakarrean identifikatzeko erabiltzen den izena; izan ere, datu-base berean ezin izan baitaitezke izen berbera duten bi erlazio. Horrek irudimena erabiltzea eskatuko die diseinugileei izenak errepika ez daitezen, eta erlazioen artean diskriminatzeko bidea ematen duten kalifikatzaileak bilatu beharko dira, beti ere komodin diren izenak saihestuz: izan, eduki, erlazionatu, lotu eta abar; aurrerago garatuko diren beste ezaugarriak hauexek dira: erlazioaren maila eta aukeragarritasuna.

a) Erlazio-maila

Entitate bakoitzak erlazioan daukan partaidetza adieraziko du erlazio-mailak.

Hiru motakoak daude:

- ✓ 1:1 (banan-banakoa). Lehenengo entitateko entitate-adibide bakoitzari bigarren entitateko entitate-adibide bakarra dagokio, eta alderantziz. Esate baterako, pertsona bakoitzari historia kliniko bakarra dagokio, eta historia kliniko bakoitza pertsona bakar batena izango da.
- ✓ 1:N (bana eneri). Lehenengo entitateko entitate-adibide bakoitzari bigarren entitateko adibide bat edo gehiago egoki dakioke, eta bigarren mailako entitateko entitate-adibide bakoitzari, ordea, lehenengo entitateko entitate-adibide bakarra egokituko zaio. Adibidez: talde batek zenbait txapelketa ditu irabazirik.

LANBIDE EKIMENA (36



✓ **N:M (enena emeri)** Lehenengo entitateko entitate-adibide bakoitzari bigarren entitateko zenbait entitate-adibide egoki dakioke, eta alderantziz. Adibidez: enpresa batean pertsona batek hainbat proiektutan dihardu lanean, eta proiektu bakoitza hainbat laguni agindu zaio.

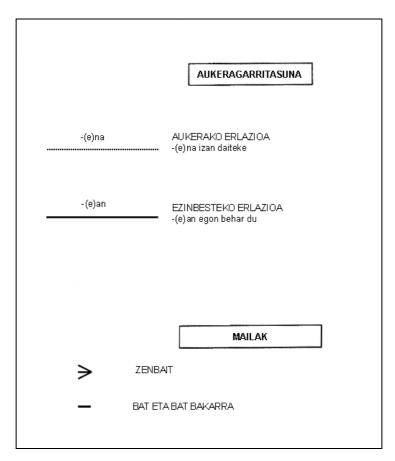
b) Aukeragarritasuna

Erlazioak honelakoak izan daitezke:

- ✓ Nahitaezkoak
- ✓ Aukerakoak

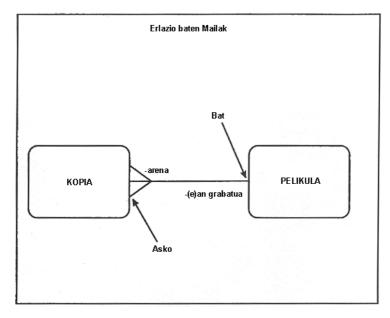
Izan ere, batzuetan, badaude entitate baten agerraldiak, erlazio batean parte hartzen ez dutenak, beste agerraldi batzuek parte hartuta ere.

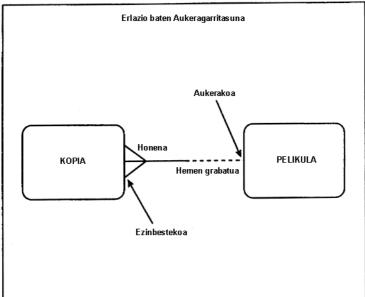
3.4. irudian ikus daiteke erlazio baten aukeragarritasuna eta maila nola irudikatzen diren:



3.4. irudia. Erlazio baten Mailak eta Aukeragarritasuna.

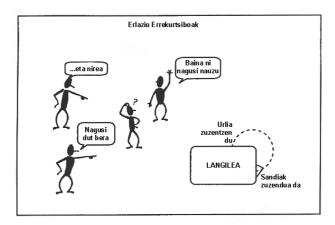
Adibideak:





Erlazio berezien mota bat dago, eta Erlazio Errekurtsibo deritze. Entitate bereko agerraldien artean izaten diren erlazioak dira.

Entitate-mota batek bere buruarekiko erlazioak dituenean, erlazio errekurtsibo esaten zaio horri. Esate baterako, langileen eta zuzendarien arteko lotura 3.5. irudian adierazten den bezala molda daiteke:



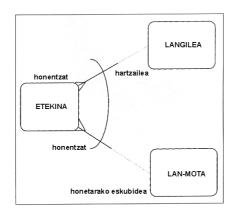
3.5. irudia. Erlazio Errekurtsibo baten adibidea.

Horrela, irtenbidea emango zaio entitate batek "aita" izan behar duenean sortzen den arazoari; izan ere, ezinezkoa izango bailitzateke hierarkia-mailarik altuenean sorraraztea. Berdin-berdina gertatuko litzateke entitateak "kumea" izanez gero.

Esklusibotasun-murrizketa duten bestelako erlazioak daude gainera. Honelako erlazioei *Erlazio Esklusiboa* edo *Arkua* esaten zaie.

Bi erlazio-motak bi erlazioetan parte hartzen duen entitate-mota batekiko esklusibotasun-murrizketa dutela esango dugu, baldin eta entitate-mota bakoitzeko ale bakoitza erlazio-motetako bati baino ezin egoki dakiokeelarik, bati dagokionean bestearena izaterik ez badauka.

3.8. irudian ageri den adibidean ikus dezakegunez, etekin bakoitzak, edo LANGILE bakar batentzat izan behar du edo, bestela, LAN-MOTA bakar baterako.



3.6. irudia. Erlazio esklusiboak.

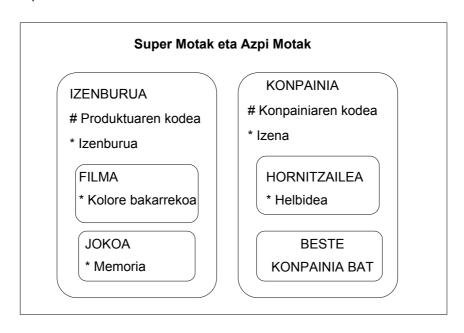
Arku baten bidez irudikatuko da esklusibotasuna, eta honako hauek bete behar dira:

- ✓ Arku bateko erlazioen alderdiei dagokienez, edo denak dira ezinbestekoak, edo denak dira aukerakoak.
- ✓ Arku bateko alderdi guztiek entitate berekoak izan behar dute.
- ✓ Arku bakarrera baino ezin bil daiteke erlazioa.

Azkenik, kontuan hartu behar dugu badela entitate bateko agerraldien multzoak taldekatzerik.

Entitate zenbaitek komunak zaizkien atributuak edo erlazioak dituztenean, elkarrekin taldeka daitezke *Supermota* izenaz ezagutuko dugun entitatearen barruan. Super-mota baten barneko entitateei *Azpimota* esango diegu.

Entitate bateko azpi-serie baterako arauak ezberdinak izan daitezkeela onartuz gero, azpi-motak definitu ahal izango ditugu. Esate baterako, interesa eduki dezakegu IZENBURUA kolore bakarrekoa denentz jakiteko. Horrela izatera, FILMA izango da. Baliteke zenbaterainoko memoria behar duen ere jakin nahi izatea. Horrela izatera, JOKOA izango da. 3.9. irudian ikus dezakegu nola irudikatzen diren supermotak eta azpimotak:



3.7. irudia. Super Moten eta Azpi Moten irudikapena.

Herentzia dugu hemen oso ezaugarri garrantzitsua; izan ere, supermoten atributu guztiak azpimoten atributu izatera pasako baitira. Aurreko adibidean, hala FILMEK nola JOKOEK duten Produktuaren kodea Izenburua da.

Era berean, azpimota guztiei eragiten dieten erlazioak supertipoekin lotuko dira, eta dagokion azpimotak baino partzen hartzen ez duen erlazio bereziak azpimotetarako lagako dira.



3.3 Diseinu logikoa: eredu erlazionala

Datu-baseak garatzeko metodologiek, diseinu-fasean, Codd-ek 1970.ean diseinatu zuen eredu erlazionala hartzen dute ardaztzat. Eredu erlazionalaren (EE) helburu nagusia erabiltzailea eta datuen egitura fisikoak isolatzea izaten da, aplikazioak datuekiko independentzia izatea lortzen baita horrela, horixe delarik, izan, datu-baseen hastapenetatik erdietsi nahi izan den jomuga.

Honako hauek dira Eredu Erlazionalaren osagaiak:

Taula

- ✓ Edozein Datu Base Erlazionali eusten dion datu-egitura da.
- ✓ Informazio-sisteman aintzat hartu nahi den entitatea irudikatuko du. Honako hauen irudikapena izan daiteke:
 - a) Entitate bakarrarena
 - b) Entitateen elkarte batena

Zutabea

- ✓ Taulak irudikaturiko entitatearen Atributu bakar baten irudikapena izango da.
- ✓ Honako hauek osatuta dago:
 - a) Izenak
 - b) Balio-multzoak
- ✓ Izenaren bidez identifikatuko ditugu beti zutabeak, eta inoiz ere ez posizioaren bidez.
- ✓ Taula batean zutabeek duten ordenak ez dauka inolako garrantzirik.

Lerroa

✓ Taularen bidez irudikatzen den entitatearen edo elkartearen Agerraldia irudikatzen du.

Balioa

✓ Lerro baten eta zutabe baten ebaketa-hutsune bakoitzean balio bakarra agertuko da, eta ez balioen multzo errepikakorra.

NULL balioa

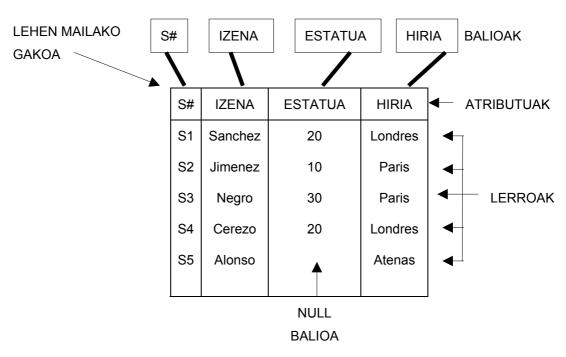
✓ Baliorik ezaren edo aplikaezina den propietatearen irudia dugu hau.

Lehen mailako klabea (Primary Key)

- ✓ Taulako agerraldi guztiak identifikatzeko bidea ematen duen zutabea edo zutabe--multzoa.
- ✓ Lehen mailako klabeak bakarrak, gutxienekoak eta ezinbestekoak (NOT NULL) izango dira nahitaez.

Besteren klabea (Foreign Key)

- ✓ Taulen arteko erlazioak irudikatzen ditu (1:1 edo 1:M).
- ✓ Taula bateko zutabe bat edo batzuk izango dira, eta beren balioek beste taularen bateko lehen mailako balioei dagozkienak izan beharko dute.



3.8. irudia. Eredu Erlazionalaren Osagaiak.

Eskema kontzeptualetik eskema logiko estandarrerako aldakuntza

E/E E eskema erlazional bilakarazteko oinarrizko arauak ondorengo hauek dira:

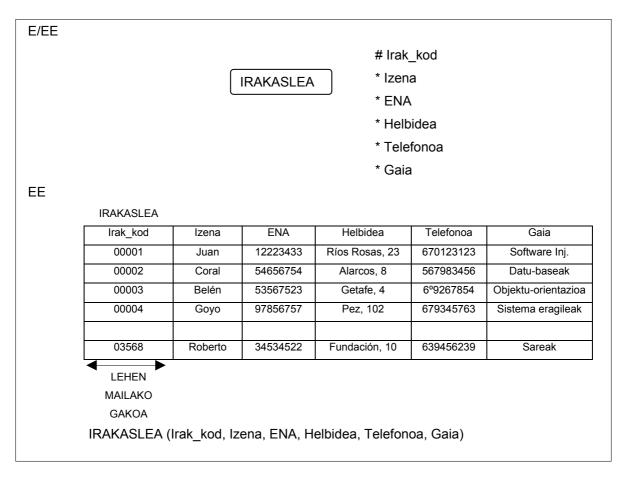
Entitateen aldakuntza

Entitate guztiak taula bihurtuko dira. Entitatetik eratorritako taulak, entitatearen izena, pluralean ezarria, hartuko du izentzat.

Atributuen aldarazpena

Entitate baten atributu guztiak entitateak sorrarazitako taulako zutabe bihurtuko dira.

Identifikatzaile nagusi den atributua edo atributuak taularen lehen mailako klabe (PRIMARY KEY) izatera pasako dira.



3.9. irudia. Entitate baten aldakuntza.

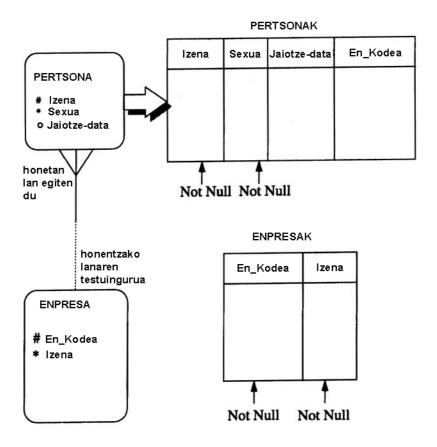
Elkarren arteko loturen aldakuntza

Erlazioaren korrespondentzia-motaren arabera, eskema erlazionalerako aldakuntza egiteko era ere aldatu egingo da eta, horregatik, ondorengo azpiarauetan xehatuko dugu arau hau:

√ 1:N erlazioak

Honelako erlazioetarako ohiko araua, besteren gako (FOREIGN KLEY) bilakaraztearena izaten da, hau da, 1 maila duten entitatearen atributu identifikatzaile nagusiak N maila duen entitatera hedatzearena.

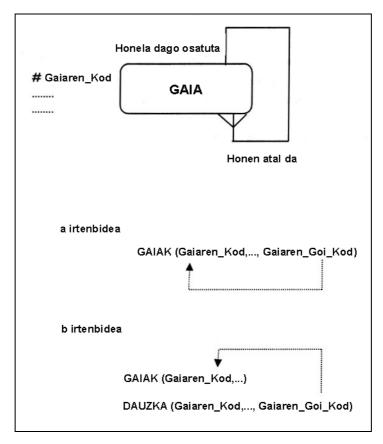
3.10 irudian ikus dezakegu aldakuntza hori:



3.10. irudia. 1:N erlazioaren aldakuntza.

Hori bai, hau ez da arau bakarra, zeren baliteke irtenbide gisa beste taula bat sorraraztea ere. Kasu honetan, sortzen den erlazioaren lehen mailako gakoa, N maila duen taularen lehen mailako gakoa litzateke.

Ondoren datorren 3.11. irudian ikus dezakegu nola eman dakiokeen irtenbidea, bi arauak erabilita, erlazio errekurtsiboaren kasuari:



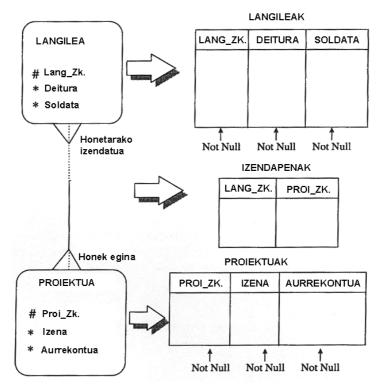
3.11. irudia. Erlazio errekurtsibo baten irtenbidea.

✓ Elkarren arteko N:M loturak

Elkargune den eta lehen mailako gako gisa, biltzen dituen entitateen lehen mailako klabeen kateaketa izango duen taula bilakatuko da N:M erlazioa.

Gainera, erlazio honen lehen mailako klabea osatzen duten atributuetako bakoitza, bilduta dauden entitateetako bakoitzaren erreferentzia diren besteren gakoak izango dira.

3.12 irudian ikus dezakegu aldakuntza:



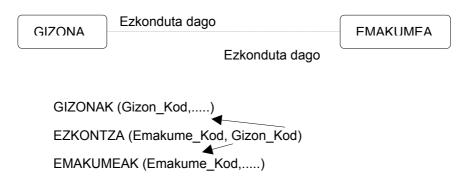
3.12. irudia. N:M erlazio baten aldakuntza.

√ 1:1 erlazioak

N:M erlazio baten edo, baita 1:N erlazioaren kasu partikularra dugu 1:1 erlazioa eta, horiek horrela, eta dago honelako erlazioak erlazio-eredura aldarazteko arau finkorik. Horregatik, taula berria sor genezake edo, bestela, besteren gakoa sortu.

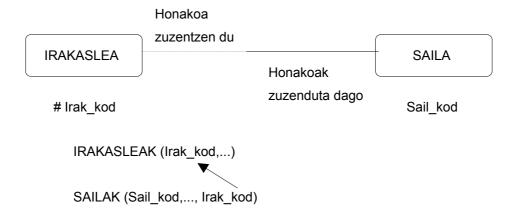
Aurrean dugun kasuaren arabera helduko diogu irizpide batari edo besteari.

Beraz, elkartzen diren entitateak 0:1 motakoak izanez gero, baliteke beste taula bat sorraraztea komenigarria izatea:



Kasu honetan, EMAKUMEA gakoa GIZONAK taulara zabalduz gero edo alderantziz eginez gero agertuko liratekeen balio nulu guztiak saihestuko ditugu, zeren ez gizon guztiak, ezta emakume guztiak ere, ez baitaude ezkonduta.

Gainerako kasuetan, berriz, komenigarria izango da 1:1 erlazioa duen entitatearen klabea beste entitatera zabaltzea eta besteren klabea sortzea:



Kasu honetan, IRAKASLEA gakoa SAILAK taulara zabaldu dugu eta horrela balio nuluak saihestu dira (irakasle bat sail baten arduradun izan daiteke gehien jota, eta baliteke ezein sail ez izatea bere ardurapean, baina sail guztiek eduki behar dute beti arduraduna).

Moten eta azpimoten aldakuntza

Entitate-mota baten eta bere azpimoten aurrean, eredu erlazionalera aldarazteko irtenbide bat baino gehiago izango da. Hiru azpimarratuko ditugu guk:

> Supermotaren eta azpimoten atributu guztiak taula bakar batera biltzea. Oro har, azpi--motek atributu ezberdin gutxi izan eta gainerako entitateekin lotzen dituzten erlazioak denentzat (edo ia denentzat) berberak direnean hartuko dugu irtenbide hau. Kasu honetan, azpi-mota baten agerraldiak eta beste batenak bereizten utziko duen MOTA izeneko beste zutabe bat erantsiko dugu sortzen den taula bakarrean.



IRASKALEAK (Irak kod, Izena,..., Mota, Dok urtea, Dok gaia)

✓ Supermotarentzako taula eta dauden azpimota adina taula sorraraztea, dagozkien atributuekin. Horixe izango da irtenbide egokia, baldin eta azpimoten artean atributu ezberdin asko badaude eta dena dela ere, denek komunean dituzten atributuak erlazio batean mantendu nahi izanez gero.



✓ Azpimotentzako taula ezberdinak hartzea aintzat. Taula horiek, bereak diren atributuak ez ezik, komunak diren atributuak ere bilduko dituzte.

3.5 Normalizazioari buruzko teoria

Modelatu kontzeptualaren etapan egindako E/E eskema eredu erlazional bilakarazteak eskemaren diseinuan izaniko akatsetatik edo eredu erlazionalera iragaitetik eratorrita dauden zenbait arazo aurkez ditzake.

Arazo horietan, ondoko hauek azpimarra genitzake:

- ✓ Zenbait egintza biltegiratzen gaitasunik eza
- ✓ Erredundantziak eta, ondorioz, trinkotasunik eza
- ✓ Anbiguotasuna
- ✓ Informazio-galera
- ✓ Balio nuluak izatea
- ✓ Irregulartasunak txertaketan, ezabaketan eta aldaketetan

Bekak eskatu dituzten ikasleei eta bekei berei buruzko informazioa ez ezik, ikasleek beka eskatu duten datari buruzko datuak biltzen dituen taula dugu 3.15. irudian. Taulari arretaz begiratuz gero, arestian aipatu ditugun arazoetako batzuk dauzkala ikus dezakegu:

IKASLEEK_BEKA_ESKATU_DUTE								
lkasle_kod	lkasle_izena	Deitura	ENA	Helbidea	Beka_kod	Izena	Betekizuna	Data
012323	Roberto	Hens	456367	Antonio Lopez 43	A22321	METRIKA	Inj, Tek.	98/10/10
763476	Luis	García	345347	Ciudades Etorb. 29	B56784	ERASMU	Inj, Tek.	98/11/12
763476	Luis	García	345347	Ciudades Etorb. 29	A22321	METRIKA	Inj, Tek.	98/10/14
763476	Luis	García	345347	Ciudades Etorb. 29	G65434	HIMMPA	Injenie.	98/09/15
012323	Roberto	Hens	456367	Antonio Lopez 43	G65434	HIMMPA	Injenie.	98/09/17
987765	Gregorio	Celada	885764	Países Pl. 67	G65434	HIMMPA	Injenie.	98/09/21
012323	Roberto	Hens	456367	Antonio Lopez 43	B56784	ERASMU	Inj, Tek.	98/11/11
987765	Gregorio	Celada	885764	Países Pl. 67	B56784	ERASMU	Inj, Tek.	98/10/10
012323	roberto	Hens	456367	Antonio Lopez 43	A22321	METRIKA	Inj, Tek.	98/11/12
232457	Mercedes	García	345347	Río Miño 2	A22321	METRIKA	Inj, Tek.	98/09/17

3.13. irudia. Diseinu ezegokiaren adibidea.

Eskema erlazional jakin bat zuzena den ala ez erabakitzean sor daitezkeen zalantzak direla-eta, hobe izango da eskeman oker egon daitekeena zehaztuko duen eta betekizun horiek beteko direla bermatuko digun beste eskema batera iristeko bidea emango digun analisi formalerako metodoa aplikatzea; metodo formal hori normalizazioari buruzko teoria dugu.

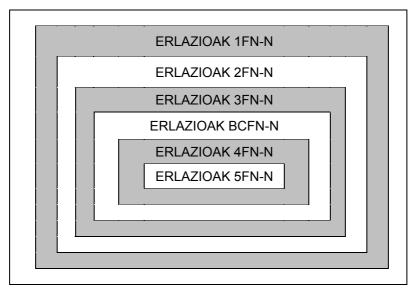
Forma normalak izenaz ezagutzen duguna hartzen du ardatz gisa normalizazioari buruzko teoriak. Taula forma normal jakin batean dagoela esango dugu, baldin eta murrizketa-multzo berezi bat asebetetzen badu.

Taulek duten forma normala zenbat eta altuagoa izan, hainbat eta txikiagoak izango dira datu-basea mantentzean agertuko diren arazoak. Mendekotasun funtzionaletan oinarri hartzen zuten hiru forma normal proposatu zituen Codd-ek hasieran: lehenbizi (1FN), bigarren (2FN) eta hirugarren (3FN) forma normalak. Datu-baseak eguneratzean ahalik eta erredundantziarik txikiena eta ahalik eta irregulartasun gutxien izatea zen Codd-ek lortu nahi zuen helburua.

3FN-n artean ere arazoek irauten zutenez, 3 FN-ren definizio murriztaileagoa sartu zuten Codd-ek Boyce-rekin batera. Boyce-Codd-en Forma Normala izena eman zitzaion forma hari.

Ondoren, laugarren (4FN) eta bosgarren (5FN) forma normalak sartu zituen Fagin-ek. Funtzionalak ez ziren mendekotasunak hartu zituen oinarritzat forma horietarako.

Taula forma normalean dagoenean, forma normal horren azpiko forma normaletan ere badago inplizituki.



3.14. irudia. Erlazioen unibertsoa elkarren segidako forma normaletan.

Lehenbiziko forma normala (1FN)

Eredu erlazionalak berez duen murrizketa da lehenbiziko forma normala (1FN). Hortaz, ezinbestean bete behar da eta atributuek taulan har dezaketen balio-kopuruari eragingo dio.

Taula bat 1FN-n egon dadin, ezin onar dezake Multzo Errepikakorrik. Atributu bakoitzak balio bakarra eduki beharko du bere entitatearen edozein agerralditarako, edozein denbora-unetan. 3.17. irudian ikus dezakegun adibidean, egile bat baino gehiago duten liburuetan aurki daiteke multzo errepikakorra.

LIBURUA (Kodea, Izenburua, Idazlea)

a)

KODEA	IZENBURUA	EGILEA	
02154989	Data models	Tsichiritzis	
02104000	Data modelo	Lochovsky	
87654353	A guide to DB2	Date	
65465465	Bases de datos	Gardarin	
00400400	Dases de datos	Valduriez	

EZ DAGO 1FN-n

Multzo errepikakorrak daude



b)

LIBURUA (Kodea, Izenburua, Idazlea)

KODEA	IZENBURUA	EGILEA
02154989	Data models	Tsichiritzis
02154989	Data models	Lochovsky
87654353	A guide to DB2	Date
65465465	Bases de datos	Gardarin
65465465	Bases de datos	Valduriez

1FN-N DAGO

3.15. irudia. Multzo errepikakorrak dituen taula.

Taula bat lehenbiziko forma normalera pasatu nahi izanez gero, klabearen balio jakin baterako hainbat balio dituzten zutabeak identifikatu behar ditugu (multzo errepikakorrak). Hori eginda daukagunean, beste taula bat osatuko dugu zutabe horiekin, jatorrizkotik ezabaturik. Taula berri horren edo erlazioaren gako nagusia osatzeko, bere zutabeetako bat edo batzuk eta taula zaharreko gako nagusia kateatuko ditugu.

E/E eredua zuzena baldin bada, taula guztiak egongo dira lehenbiziko forma normalean.

Adibidea:

ESKARIAK (Eskari-Zk., Bezero_kod, Data, Art_Kod, Prezioa, Kopurua)

Eta demagun eskari batean artikulu bat baino gehiago eskatu direla



Honakoa izango da multzo errepikakorra:

- ✓ Art Kod
- Prezioa
- Kopurua

Irtenbidea → Bi taula:

ESKARIAK (Eskari-Zk., Bezero_Kod, Data)

ESKARI_LERROAK (Eskari-Zk., Art Kod, Prezioa, Kopurua)

Bigarren forma normala (2FN)

Erabateko mendekotasunaren eta taula bateko atributu nagusien (klabeetakoren batean daude) eta nagusi ez direnen (ez daude ezein klabetan) artean dauden erlazioen kontzeptuetan hartzen du oinarri bigarren forma normalak (2FN).

Honakoak beteko dira erlazio bat 2FN-n dagoenean:

- ✓ 1FN-n egongo da.
- ✓ Nagusi ez diren eta gako hautagaietakoak ez diren atributu guztiek erabateko mendekotasun funtzionala eduki behar dute biltzen dituen taulako lehen mailako gakoarekiko.

Gako nagusia soila duen edozein taula, hau da, atributu bakarreko klabea duen edozein taula, jada 2FN-n dagoela baiezta daiteke.

Taula bat bigarren forma normalera igaroarazteko, honako hauek egin beharra dago:

- ✓ Lehen mailako gakoa osatzen duten zutabe guztien mende ez eta zati baten mende dauden zutabeak identifikatzea.
- ✓ Aurreko pausoan identifikatu den lehen mailako gakoaren zutabe bakoitzarekin eta zutabe horren mendeko diren zutabe guztiekin, beste taula bat osatuko dugu. Taula horrek gako moduan identifikatu dugun zutabea eta bere mendeko ziren gainerako zutabeak izango ditu. Taula horiek, gainera, ezabatu egingo ditugu jatorrizko taulatik.

Adibidea:

Aurreko adibideari eutsiko diogu orain ere:

ESKARI_LERROAK (<u>Eskari-Zk., Art_Kod, Prezioa, Kopurua</u>)

Prezioa Art Kod-earen araberakoa izango da funtzionalki (artikuluek prezio bakarra izaten dute beti)

Irtenbidea → Bi taula:

ESKARI LERROAK (Eskari-Zk., Art Kod, Prezioa, Kopurua) ARTIKULUAK (Art Kod, Prezioa)

Hirugarren forma normala (3FN)

Erlazio bat 3FN-n dagoela esateko, honako hauek bete behar dira nahitaez:

- ✓ 2FN-n dago
- ✓ Nagusi ez diren eta klabe hautagaiei ez dagozkien atributu guztiek, balio baterako entitatean dagoen identifikatzaile bakarraren mendekoak baino ez dute izan behar.

Honakoak egin beharko ditugu erlazioa hirugarren forma normalera pasaraziko badugu:

- ✓ Beste zutabe baten mende dauden eta lehen mailako gakoaren atal ez diren zutabeak identifikatu behar ditugu.
- ✓ Beste taula bat osatuko dugu zutabe horiek erabilita, eta gainerako zutabeak bere mende dituen zutabea finkatuko dugu lehen mailako gako gisa. Honelakoetan, ezabatu egingo ditugu zutabe hauek jatorrizko taulatik.

Adibidea:

Demagun honako taula hau dugula:

AUTOAK (Matrikula, Marka, Modeloa, Kolorea)

✓ Marka modeloaren araberakoa izango da funtzionalki:

Modeloa — → Marka (modeloa beti izango da marka bakar batekoa)

✓ Modeloa ez da funtzionalki Markaren araberakoa izango

Marka — → Modeloa (marka batean hainbat modelo daude)

Irtenbidea → bi taula:

AUTOAK (Matrikula, modeloa, Kolorea) MARKAK (Marka, Modeloa)

4 SQL LENGOAIA

4.1 SQL-ren ezaugarriak

Datu-baseak kudeatzeko sistema batez lan egiteko, ORACLErekin lan egiteko esaterako, Eskari Lengoaia egituratuaren komandoak erabili behar dituzte (SQL, Structured Query Language) aplikazioek.

Honako ezaugarri hauek azpimarra ditzakegu SQL-k dituenetan:

- ✓ Komando-lengoaia sinple honi esker, datu-baseen administratzaileek, programatzaileek eta aplikazioen erabiltzaileek hurrengo hauek gauza ditzakete:
 - a) Datuak berreskuratu, sartu, eguneratu eta ezabatu datu-basean.
 - b) Datu-baseko objektuak, taulak esaterako, sortu, aldarazi eta desagerrarazi.

Hau da, datuak definitzeko bidea ematen duen lengoaia (DDL) eta datuok kudeatzeko bidea ematen duen lengoaia (DML) dugu, datu-basean sorrarazitako elementu guztiak biltzen baititu.

- ✓ Ez da prozedura-lengoaia. Erabiltzaileak ZER lortu nahi duen zehaztu behar du eta ez, ordea, NON edo NOLA.
- ✓ Datu-baseak programatzeko lengoaia gisa erabil daiteke, beste lengoaia batzuetan txertaturik (C, COBOL, Fortran, Ada, Pascal...) edo Oracle-ren PL/SQL programazio--lengoaia gisa zuzenean erabilirik.
- ✓ Xumea eta sendoa denez gero, modu orokorrean erabiltzen da datuak era masiboan biltzeko eta berreskuratzeko.

4.2 SQL-ren komando-motak

SQL izena Structured Query Language-tik badator ere, horrek ez du esan nahi SQL queries (kontsultak) direlakoak egiteko baino erabiltzen ez denik. SQL ondoren adierazten diren eragiketak erabil daitekeen lengoaia dugu une honetan:

✓ Datuak Definitzeko DDL Lengoaia (Data Definition Language)

Objektuak sorrarazi, aldarazi eta ezabatu egiten dituzte datu-basean. Datu-baseko objektu gehienek CREATE, ALTER, DROP eta antzeko komandoak dituzte.

✓ Datuak Manipulatzeko DML Lengoaia (Data Modification Language)

Oracle-ko datu-baseetan taula bateko lerroak berreskuratu, txertatu, eguneratu eta desagerrarazi egiten dituzte. Honako hauek dira oinarrizko DML lau komandoak: SELECT, INSERT, UPDATE eta DELETE.

✓ Datuak kontrolatzeko DCL Lengoaia (Data Control Language)

Oracle-ko datu-base batera sartzen diren erabiltzaileak kontrolatzeko. Gehien erabiltzen diren hiru DCL komandoak ondorengoak dira: GRANT, REVOKE eta SET ROLE.

✓ Transakzioak Kontrolatzeko Komandoak

Beraien arteko lotura duten SQL sententziak gauzatzen dituen lan-unitatea da datu-baseen transakzioa. Datu-base bateko informazioaren segurtasuna zaintzeko bi instrukzio dira beharrezko: COMMIT eta ROLLBACK. Bada interesgarria den beste instrukzio bat, SAVEPOINT deritzona, hain zuzen ere.

4.3 Datuak definitzeko lengoaia (DDL)

Taulak sortzen

Taula sortzeari ekin aurretik, oso komenigarria da honako hauek planifikatzea:

- ✓ Taularen izena
- ✓ Zutabe bakoitzaren izena
- ✓ Zutabe bakoitzean bilduko den datu-mota
- ✓ Zutabeen tamaina
- ✓ Informazio gehigarria (ezinbesteko zutabeak, lehen mailako gakoak eta kanpoko gakoak)

Taularen izenari buruzko mugak

- ✓ Gehienezko luzera: 30 karaktere.
- ✓ Alfabetoko karakterea izan beharko du hasieran. Gainerakoak, berriz, letrak, zenbakiak nahiz azpimarratzeko karakterea izan daiteke.
- ✓ Berdin dio letrak larriak zein xeheak izan.
- ✓ Ezin errepika daitezke, ez taulak eta ez ikuspegiak.
- ✓ Zutabeak zabal definitu behar dira. Horrek ez du areagotuko diskoan tauletarako behar den espazio-beharra, taula bakoitzak balioari eusteko behar den adinako espazioa agortuko baitu.

Datu-motak

VARCHAR2 (Size)

Letra larriak nahiz xeheak edo zenbaki bidezko edozein karaktere, gehi + . - , \$ eragiketa-karaktereak edo beste karaktere berezi batzuk eduki ditzaketen zutabeak. Luzera aldakorrekoak izango dira. Gehien jota, 2.000 karaktere izango diztute. Definituta dagoena baino luzera txikiagoko katea sartuz gero, luzera horrekin biltegiratuko da eta ez da definitutako luzera osatu arte, zuriunez edo eskuinerako karakterez beteko. Zehaztutakoa baino luzera handiagoko katea sartuz gero, berriz, akatsa itzuliko du Oracle-k.

CHAR (Size)

Letra larriak nahiz xeheak edo zenbaki bidezko edozein karaktere, gehi + . - , \$ eragiketa-karaktereak edo beste karaktere berezi batzuk eduki ditzaketen zutabeak. Luzera finkokoak izango dira. Gehienezko luzera 255 karakterekoa dute. Datuek definizioak baino gutxiago okupatuz gero, zuriuneak erantsiko dira eskuinaldean luzera osatu arte. Zehaztutakoa baino luzera handiagoko katea sartuz gero, berriz, akatsa itzuliko du Oracle-k.

LONG

Letra larriak nahiz xeheak edo zenbaki bidezko edozein karaktere, gehi + . - , \$ eragiketa-karaktereak edo beste karaktere berezi batzuk eduki ditzaketen zutabeak. Luzera aldakorrekoak dira. Gehienezko luzera 2 gigabytekoa izango da. Memo edo iruzkin-erako eremuak aurkezteko erabiltzen dira gehienetan.

NUMBER (p.s.)

Zenbaki bidezko edozein karaktere, zenbaki osoak zein hamarrenekoak, zeinudunak nahiz zeinurik gabeak alegia, eduki ditzaketen zutabeak. Digituen kopurua guztira izango da "p" eta "s", berriz, hamarren-kopurua. 1 eta 38 bitartekoa izango da "p"-k izan dezakeen balioa. "s"-ren maila, berriz, -84tik 127ra bitartekoa izango da.

DATE

K.a. 4712. urteko urtarrilaren 1etik K.o. 4712. urteko abenduaren 31ra arteko datak eta orduak eduki ditzaketen zutabeak. Honako informazio hau biltegiratuko da Date--mota bakoitzari dagokionez:

Mendea / Urtea / Hilabetea / Eguna / Ordua / Minutuak / Segundoak

RAW (Size)

Gehien jota 255 byteko tamaina edo luzera duten datu bitarrak onartzen dituzten zutabeak.

LONG RAW

RAW bezala, baina 2 Gb-ko luzeraz. RAW eta LONG RAW datu-motak VARCHAR eta LONG bezalakoak dira, honako hauei dagokienean izan ezik:

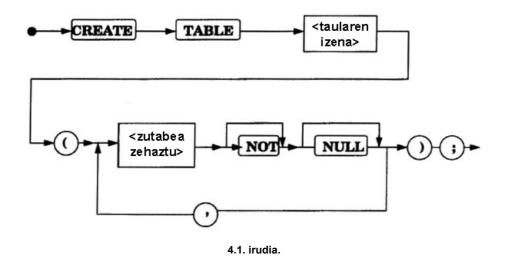
- ✓ Ez da komeni bertan testu-informaziorik biltzea.
- √ Hexadezimaleko datuak edo karaktere-kateak sartuz gero, bitar bilakatuko dira automatikoki.
- ✓ Irudiak eta soinuak biltegiratzeko ere balio dute.
- ✓ BLOB ere baderitze (Binary Large Objects).

Balio Nulua

Balio nuluak (NULL) honako ezaugarri hauek ditu:

- ✓ Baliorik ez dela esan nahi du.
- ✓ Ez da zero edo tartea.
- ✓ Lerroan ez du espaziorik okupatuko.
- ✓ Zutabe batek balio nulurik edukitzerik ez badu, NOT NULL kalifikatzailearen bidez zehaztu beharko da taula sorrarazten denean.

Taula baten sorreraren sintaxia



CREATE TABLE gaixoa

(inskripzioa	NUMBER(5) NOT NULL,
deitura	VARCHAR2(25),
helbidea	VARCHAR2(30),
jaiotze-data	DATE,
sexua	VARCHAR2(1)
gs_zk	NUMBER(9));

Sententzia horrek GAIXOA izeneko taula sorraraziko du. Taulak sei zutabe izango ditu, eta zutabeetan lehenengoak balioren bat eduki beharko du ilara berria txertatzeko.

Taula bat sorrarazi mugak izanik

Taula bat eraikitzeko tenorean, datu-basearen datuak aldez aurretik definiturik dauden muga batzuetara egokituta daudela ziurtatu beharra daukagu. Segurtasuneko muga, hortaz, taulako zutabe baterako edo gehiagorako balioak mugatzen dituen araua izango da.

CREATE TABLE izeneko aginduari esker zenbait muga-mota bereiz daitezke taula baten gainean: lehen mailako gakoak, besteren gakoak, derrigortasuna, lehenespenak eta baldintzen egiaztapena.

Horiek horrela, Entitate Segurtasunarekin lotutako muga batzuk ondorengo hauek izango lirateke:

Lehen mailako gakoa (PRIMARY KEY)

- ✓ Zutabea edo zutabe-multzoa ezarriko du taulako lehen mailako gako gisa.
- ✓ Bakarra izan beharko du, ez-nulua eta ezinbestekoa.
- ✓ Gehien jota, bat izango da taulako.
- ✓ Besteren gakoak erreferentziatua izan daiteke.
- ✓ Gaitu edo sorrarazten denean, taularako sarbidea erraztuko duen aurkibidea sorraraziko du era automatikoan.
- ✓ Era esplizituan adierazi beharrik izan gabe, bakar eta ezinbesteko gisa arituko da.

Derrigortasuna (NOT NULL)

- ✓ Zutabeek balioak eduki behar dituzte.
- ✓ Salbuespena eta edozein sententziaren Rollback-a eragingo dira edozein zutabe NULL uzten saiatuz gero.

UNIQUE

- ✓ Zutabe berean balio errepikatuak egon daitezen eragotziko du.
- ✓ Zutabe bat edo batzuk eduki ditzake.
- ✓ NULL balioak onartuko ditu.
- ✓ Gaitzen denean automatikoki sorraraziko du aurkibidea eta gaitasuna galtzean, berriz, ezabatu egingo du.
- ✓ Ez dago ikuspegi edo sinonimo baten gainean sorrarazterik.

CHECK

- ✓ WHERE klausulan ezarritako iragazkia lerro guztietarako bete dadin adierazteko.
- ✓ Tauletako zutabe askok maila baten

Lehenespenak (DEFAULT)

- ✓ Lehenespena emango du txertaketan zutabea zehazten ez denean.
- Honako hauek izango dira adierazpen baliagarriak: konstanteak, SQL funtzioak eta UID eta SYSDATE aldagarriak.
- ✓ Ezin da zutabeei edo PL/SQL funtzioei buruzko aipamenik egin.

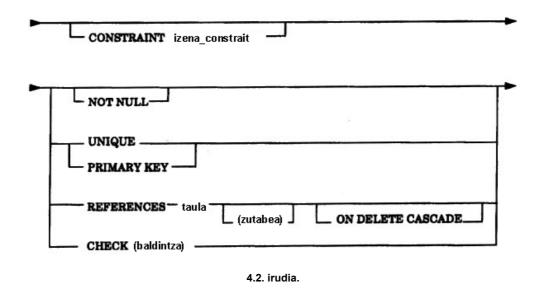
Besteren gakoa (FOREIGN KEY)

- ✓ Beste taula bateko edo taula bereko lehen mailako gakoari lotutako zutabe batek edo batzuek osatuta dago.
- ✓ Lehen mailako gakoaren eta besteren gakoaren arteko trinkotasuna bermatuko du.
- ✓ Ez dago mugarik besteren gakoen kopuruari dagokionean.
- ✓ Lehen mailako gakoa dagoen taula berean egon daiteke.
- ✓ Honako hauek izan daitezke zutabeon balioak: NULL izan daiteke edo, bestela, erreferentziatzat erabiltzen den gakoaren balio baten modukoa (erreferentzia- segurtasunaren araua).

Baliorik gabeko txertaketen, ezabaketen edo eguneratzeen aurkako prebentziorako balio duten bestelako mugak ere badaude. Hona hemen Erreferentzia Segurtasunari eusteko ekintzetako zenbait:

- ✓ Ez dago lerroak ezabatzerik, baldin eta besteren gakoren baten bidezko erreferentzia baldin bada (ezabaketa-muga).
- ✓ Lehen mailako gakoaren eguneratzea ere eragotziko da (eguneratze-muga).
- ✓ Gidari den taulako lerroetako bat ezabatuz gero, besteren gakoa erreferentziatua duen xehetasun-taulako lerro guztiak ere automatikoki ezabatuko dira (ezabaketa turrustan).
- ✓ Xehetasun-taulan adieraziko da muga.

CREATE TABLE sententzia erabiliko dugu muga horiek (CONSTRAINS) zehazteko.



Segurtasun-muga edo taula bateko zutabe baten edo gehiagoren balioa mugatuko duen araua dugu CONSTRAIT delakoa.

Izena eta Datuen Hiztegiaren barruan definizioa dituen objektu gisa gordeko da CONSTRAIT-a, eta hiztegiko tauletan kontsulta eginez gero lor daiteke berorien gaineko informazioa.

- ✓ USER_CONSTRAITS: erabiltzailearenak diren taulen mugen definizioak.
- ✓ ALL_CONSTRAINS: erabiltzailea irits daitekeen taulei dagozkien mugen definizioak.
- ✓ DBA_CONSTRAINS: taula guztiei dagozkien mugen definizio guztiak.
- ✓ USER_CONS_COLUMNS: erabiltzailearen tauletako zutabeen mugei buruzko informazioa.
- ✓ ALL_CONS-COLUMNS: erabiltzailea irits daitekeen tauletako zutabeen mugei buruzko informazioa.
- ✓ DBA_CONS_COLUMNS: zutabeen muga guztiei buruzko informazioa.

Constrait-a definitzean, *CONSTRAIT izena_constrait* formula erabilita izena adierazi ezik, Oracle-k automatikoki izendatuko du SYS_COOn izena emanda. Izen horretan, Oracle-k automatikoki emaniko zenbakia izango da "n".

Adibidea

LANG taularen sorrera, SAIL-en eta LANG-en artean 1:N lotura eta LANG-en eta LANG-en artean (bestearen nagusi izatea) beste 1:N lotura bat izanik:

CREATE TABLE Lang

NUMBER(4) CONSTRAIT pk lang PRIMARY KEY, (lang-zk deitura VARCHAR2(16) CONSTRAIT nn_dei NOT NULL, lanbidea VARCHAR2(10), nagusia NUMBER(4) CONSTRAIT fk_nagusi REFERENCES lang(lang-zk), alta-data DATE DEFAULT SYSDATE, soldata NUMBER(10, 2) CONSTRAIT ch_soldata CHECK(soldata>1000), komisioa NUMBER(9, 2) DEFAULT NULL, sail-zk NUMBER(2) CONSTRAIT zk_saila NOT NULL CONSTRAIT fk_saila REFERENCES saila(sail-zk));

Adibide horretan in line dira muga guztiak, hau da, zutabe-mugak. Ez in line mugen adibide bat, hau da, taula-muga baten adibidea, LANG taula gako konposatuaz eratzea izango litzateke.

CREATE TABLE Lang

(lang-zk NUMBER(4), deitura VARCHAR2(16) CONSTRAIT nn_dei NOT NULL, lanbidea VARCHAR2(10), nagusia NUMBER(4) CONSTRAIT fk_nagusi REFERENCES lang(lang-zk), DATE DEFAULT SYSDATE, alta-data soldata NUMBER(10, 2) CONSTRAIT ch_soldata CHECK(soldata>1000), komisioa NUMBER(9, 2) DEFAULT NULL, sail-zk NUMBER(2) CONSTRAIT zk_saila NOT NULL CONSTRAIT fk saila REFERENCES saila(sail-zk), CONSTRAIT pk_lang PRIMARY KEY (lang_zk, sail_zk));

Taula baten sorrera kontsulta batetik berreskuratutako datuak erabilita

Hala taula sorrarazteko, nola azpi-kontsultak itzultzen dizkigun erregistroen bidez betetzeko, AS klausula azpi-kontsulta gisa aplikatzea litzateke taula sorrarazteko beste metodo bat.

Ondorengo hau dugu sintaxia:

CREATE TABLE izena_taula (izena zutabea,...) AS SELECT atributua(k) FROM taula WHERE baldintza(k)

- ✓ Zehaztutako zutabeen izenez sortuko da taula eta SELECT sententziaren bidez berreskuratutako erregistroak taulan txertatuak izango dira.
- ✓ Baliteke zutabe baten definizioak zutabearen izena eta lehenespena besterik ez edukitzea.
- ✓ Zutabeen izenak emanez gero, zutabe-kopurua SELECT delakoan zehaztutako zutabe--kopuru berbera izango da.
- ✓ Taula berriko zutabeetarako izenik zehaztu ezean, azpi-kontsultako izen berberak izango dira beroriek.
- ✓ Ez da zutabeen tamaina edo zutabe-motarik zehaztu behar; izan ere, azpi-kontsultan berreskuratzen diren zutabeen tamainak eta zutabe-motak zehazturik etorriko baitira.

Taula baten egituraren aldarazpena

Taula bat sortuta dagoenean, egitura aldaraz dakioke ALTER TABLE sententzia erabilita. Honetarako erabiliko dugu sententzia hori:

- ✓ Beste zutabe bat eransteko
- ✓ Lehendik dagoen zutaberen bat aldarazteko
- ✓ Beste zutabe bati lehenespena emateko

Honako hau dugu taula baten egitura aldarazteko sintaxia:

ALTER TAULA izena_taula

ADD (izena_zutabea mota_zutabea (NOT NULL),,);

ALTER TAULA izena_taula

MODIFY (izena_zutabea mota_zutabea (NOT NULL),,);

Adibidea:

ALTER TABLE gaixorik
ADD (altuera NUMBER(3,2));

Adibide honen bidez, "altuera" zutabea erantsiko zaio "gaixoa" taulari. Zutabe hori 3 digitu izango dituen zenbaki-erako zutabea izango da, eta zenbaki horietako bi hamarrenen izango dira.

Adibidea:

ALTER TABLE gaixoa

MODIFY (altuera NUMBER(4,2) CONSTRAIT nn_altuera NOT NULL);

Arestian gehitutako zutabea aldaraziko dugu adibide honen bidez. Digitu batean areagotuko dugu zutabearen guztizko tamaina eta NOT NULL muga ezarriko diogu.

Honako hauek hartu beharko dira kontuan:

- ✓ Taula bati zutabe bat gehitu nahi izanez gero, zutabea NOT NULL gisa definituta egon ezean, edozein unetan gehi diezaiokegu zutabea.
- ✓ Zutabea NOT NULL gisa definituta egonez gero, berriz, taula hutsik dagoenean baino ezin izango diogu zutabea gehitu.
- ✓ MODIFY klausularen bidez ez dago zutabe baten tamaina txikiagotzerik, baldin eta hutsik ez badago (NULL balioa badauka).
- ✓ NUMBER erako zutabe batean baliteke posizio-kopurua areagotzea edo txikiagotzea.
- ✓ Taulak, aldatu nahi den zutabean balio nuluko lerrorik ez duenean baino ezin izango da erabili MODIFY ...NOT NULL aukera.
- ✓ Muga gehitu nahi delarik, taulan lerroak izan eta lehendik dauden lerroek CONSTRAIT berria bete ezean, errorea sortuko da eta taulari ez zaio mugarik erantsiko.

Taula bateko mugak eranstea eta ezabatzea

ALTER TABLE sententzia berbera erabiliko dugu taula batean beste muga batzuk eransteko edo lehendik dauden mugak ezabatzeko. Honako hau da sintaxia:

ALTER TABLE izena_taula

ADD CONSTRAINT (definizioa_constrait)

ALTER TABLE izena_taula

DROP UNIQUE (izena_zutabea)

PRIMARY KEY

CONSTRAINT definizioa_constraint;

Adibideak:

ALTER TABLE saila

ADD CONSTRAINT up_deiturak UNIQUE (deiturak);

ALTER TABLE saila

DROP CONSTRAINT fk_saila;

ALTER TABLE saila DROP PRIMARY KEY;

Taula ezabatzea

Taula bat ezabatzeko formatua ondorengo hau da:

DROP TABLE izena_taula

(CASCADE CONSTRAINTS);

Sententzia hori erabilita, honako hauek ezabatuko dira:

- Datu-basearen taula. Erabiltzaile bakoitzak bere taulak ezaba ditzake; basearen administratzaileak edo DROP ANY TABLE pribilegioa duen erabiltzaileren batek baino ezin ditzakete beste erabiltzaile baten taulak ezaba.
- ✓ Taulara bilduta dagoen informazioa, hau da, taularen lerro guztiak.
- ✓ Taula bat ezabatzean berarekin lotuta dauden aurkibideak eta pribilegioak ere ezabatuko dira. Taula horretatik abiatuta sorrarazitako ikuspegiek eta sinonimoek ez dute jadanik funtzionatuko, baina datu-basean jarraituko dute. Beraz, ezabatu beharra egongo litzateke.

Taularen gainean beste taula batzuei eragiten dieten constrait direlakoak definitu direnean erabiliko da CASCADE CONSTRAINTS izeneko klausula. Taula nagusiaren lehen mailako gakoari erreferentzia egiten dion besteren gakodun taula mendeko batekin lotuta dagoen taula nagusi baten ezabaketa izango da kasurik arruntena; klausula hori adierazi ezean, errorea emango luke; izan ere, besteren gakoak jadanik izan ez den lehen mailako gako bati egingo bailioke erreferentzia.

Ikuspegiak sortzea eta erabiltzea

Taula bateko edo batzuetako informazioa ikusteko edo aldatzeko balioa duen leihoa da ikuspegia. Taulen egitura logiko berbera du, hau da, zutabeak eta lerroak, baina ez dauka, ordea, daturik, hau da, ez dauka taulen egitura fisikoa. Datuen hiztegian gordetzen da bere definizioa.

Taulak bezala tratatuko ditugu ikuspegiak. Aplikazioek, taulen gainean lan egin beharrean, ikuspegien gainean lan egin dezaten erabili ohi dira, horrela independenteak izango baitira taularen aplikazioa eta taularen definizioa.

Taula bat desagerraraziz gero, berarekin lotuta dagoen ikuspegiak balioa galduko du.

Ikuspegien gaineko eragiketak:

Ikuspegia sortzeko

CREATE (OR REPLACE) VIEW izena ikuspegia (zutabea1, zutabea2...) AS

SELECT gorputza select;

Sortuta dauden ikuspegien kontsulta egiteko, USER VIEWS izeneko ikuspegia edukiko dugu.

LANBIDE EKIMENA 65



Ikuspegia ezabatzeko

DROP VIEW izena_ikuspegia

Ikuspegi baten kontsulta

SELECT izena_zutabea1, izena_zutabea2,...

FROM izena_ikuspegia

WHERE baldintza

Ikuspegia eguneratzeko

Ikuspegien gainean gauza daitezkeen eragiketak taulen gainean burutzen diren eragiketa berberak dira: SELECT, INSERT, UPDATE eta DELETE egin dezakegu.

Ikuspegietan zuzenean zein definitzeko balio dieten taulen bidez egunera daiteke informazioa. Ikuspegiak mugatzeko tenorean, bada mugarik:

- ✓ Ikuspegi baten bidez taula bateko lerroak ezabatzeko, taula bakarreko lerroen bidez sortutakoa izan beharko du ikuspegiak, eta ezin izango dira GROUP BY eta DISTINCT klausulak erabili.
- ✓ Lerroak ikuspegi baten bidez eguneratzeko, aurreko mugen gainean definiturik egongo da ikuspegia eta, gainera, eguneratu beharreko zutabeetatik bat ere ez da adierazpen gisa definituta egongo.
- ✓ Ikuspegi baten bidez taula batean lerroak txertatzeko, aurreko muga guztiak eduki behar dira kontuan eta, gainera, lotutako taularen ezinbesteko zutabe guztiek agertu behar dute ikuspegian.

Sinonimoak sortzea

Taula edo ikuspegi baterako sinonimoak sortzeko bidea ematen du SQL-k. Horiek horrela, bi izen ezberdin erabil daitezke objektu berbererako.

Beste erabiltzaile batzuen tauletarako sarbidea izanez gero izango dira interesgarriak sinonimoak; taula horiei erreferentzia egiteko erabil daitezke sinonimoak eta, horrela, ez dago sarbidea dugun taularen aurrean kontsulta egin nahi dugun bakoitzean taularen jabe den erabiltzailearen izena zertan idatzirik.

Ondorengoa izango da sententzia honen sintaxia:

CREATE SYNONYM izena_sinonimoa FOR (id_erabiltzailea).izena_taula;

Adibidea:

CREATE SYNONYM zerrenda_oheak For okupazioa

Erabiltzaile bati beste baten tauletara sartzeko pribilegioa eman zaionean ere interesgarria izango da aukera hau.

Adibidea:

CREATE SYNONYM langileria FOR erabiltzailea1.langileria

Sinonimoa ezabatzeko

Sinonimoak sortzen diren bezala, honako formatu hau erabilita ezabatu ere egin daitezke:

DROP SYNONYM izena_sinonimoa

Izen-aldaketak

Taula, ikuspegi edo sinonimo baten izena aldarazteko erabiliko dugu RENAME komandoa. Honako hau da sententziaren sintaxia:

RENAME izena_zaharra TO izena_berria;

Objektuaren izena aldaraziko dugu sententzia honen bidez eta, horrela, errore-mezua itzuliko digu objektura lehengo izenaz sartu nahi izanez gero.

Objektuari emaniko segurtasun-mugak, aurkibideak eta baimenak automatikoki transferituko dira objektu berrira.

Berriz izendaturiko objektuaren mendeko diren objektu guztiak baliorik gabe utziko ditu Oracle-k: berriz izendaturiko taulari erreferentzia egiten dioten ikuspegiak, sinonimoak eta biltegiratuko prozedurak.

Aurkibideak sortzea

Datuetarako sarbidea optimizatzeko sortuko dira aurkibideak, baina erabiltzaileek ezin egin diezaiekete erreferentziarik, sistemak hartuko baitu aurkibidea erabiltzeko edo ez erabiltzeko erabakia. Lehenespenez, lehen mailako gakoen gaineko aurkibideak sorraraziko dira automatikoki, baina erabiltzaileak beste atributu batzuen gaineko aurkibide gehigarriak sorraraz ditzake.

Aurkibidea sorrarazteko:

CREATE (UNIQUE) INDEX izena_aurkibidea ON izena_taula (izena-zutabea (ASC/DEC), ...);

Zutabe bakoitzerako ordena zehaztu ezean, lehenespenez ASC bereganatuko da.

Eremuen ezkerretik eskuinerako sekuentzia handienetik txikienerako antolamenduari dagokio. Balio nuluak beste edozein balio baino handiagotzat hartuko dira.

Taulan, aurkibide-balio berbera duten bi erregistro ezin egon daitezkeela zehaztuko du UNIQUE aukerak. Lehen mailako gakoetarako erabiliko da.

Aurkibidea sortuta dagoenean, automatikoki iraungo du harik eta ezabatzen denera arte.

Aurkibidea ezabatzeko:

DROP INDEX izena_aurkibidea;

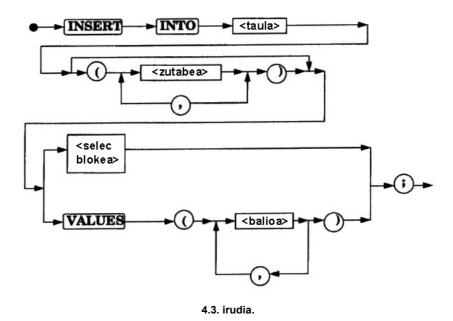
4.4 Datuak manipulatzeko lengoaia (DML)

Taulak sortuta daudenean, honako hauek gertatuz gero exekutatuko dira DML sententziak:

- ✓ Taulari erregistro berriak erantsi
- ✓ Lehendik dauden erregistroak aldarazi
- ✓ Lehendik dauden erregistroak desagerrarazi
- ✓ Taula bateko erregistroak errekuperatu

Datuak txertatzen. INSERT agindua

Taula batean lerroak txertatzeko sintaxia ondorengoa da:



- ✓ Zutabeak izenarekin identifikatuko dira eta lotura, berriz, kokapen-balioarekin.
- ✓ Definituta dauden datu-motetara egokiturik egon behar dute balioek.
- ✓ Zutabeak balioak zehazteko erabilitako ordena berberean dituen taula definitu eta zutabe--kopurua balio-kopuruarekin bat etorriz gero, ez dago zutabeen izenak zertan jarririk.
- ✓ Zutabeen zerrendan ageri ez diren zutabe guztiek NULL balioa hartuko dute, baldin eta NOT NULL gisa definituta ez badaude. Horrela izanez gero, INSERTek huts egingo du.
- ✓ Zutabeen zerrenda eman ezik, zutabe guztietan sartu beharko dira balioak.
- ✓ Karaktere edo data motako balio konstanteek komatxo soilen artean (") sartuta joan beharko dute.
- ✓ Txertaketa gauzatzeko, beste taula batzuetako balioak atera ahal izango dira.

Adibideak:

INSERT INTO langileria

(kod_ospitale, kod_aretoa, lang_zk, deitura, betebeharra, txanda, soldata)

VALUES (22,2,12345,'García J', 'erizaina', 'M', 3000);

INSERT INTO langileria

VALUES (22,2,12345, 'García J', 'erizaina', 'M', 3000);

INSERT INTO langileria (lang_zk, deitura)

VALUES (12345,'García J');

INSERT INTO gizonezkoa

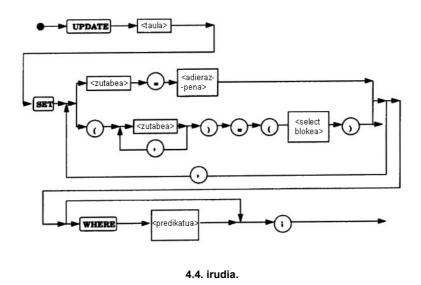
SELECT inskripzioa, deitura, helbidea, jaiotze-data, gsz

FROM gaixoa

WHERE sexua = 'G'

Datuak aldatzea. UPDATE agindua

Taula bateko lerroak aldarazteko sintaxia honako hau da:



- Zer zutabe eta zein baliorekin eguneratuko den adieraziko du SET klausulak.
- ✓ WHERE klausulak lan egiteko erabiliko diren lerroek bete beharreko baldintza adieraziko du. Adierazi ezean, taularen agerraldi guztiei eragingo die eguneratzeak.
- Beste taula batzuetatik ateratako datuen bidez ere egin daiteke aldarazpena.

Adibideak:

UPDATE gaixoa

SET helbidea = 'Castellana, 75'

inskripzioa = 74835; WHERE

UPDATE gaixoa

SET helbidea = NULL;

UPDATE gaixoa

SET (helbidea, jaiotze-data) =

(SELECT helbidea, jaiotze-data

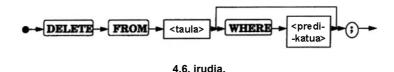
FROM gaixoa

WHERE inskripzioa = 67328)

WHERE inskripzioa = 12345;

Taula bateko lerroak ezabatzea. DELETE agindua.

Taula batean lerroak ezabatzeko sintaxia honakoa da:



✓ WHERE klausula funtsezkoa da ezabatu nahi ditugun lerroak (eta besterik ez) ezabatzeko. WHERE klausularik izan ezik, DELETE-k taulako lerro guztiak ezabatuko ditu.

Adibideak:

DELETE

FROM gaixoa

WHERE inskripzioa = 74385;

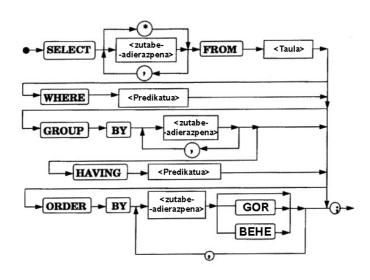
DELETE FROM gaixoa;

Agindu horrek GAIXOA taulako lerro guztiak ezabatuko ditu, baina ez, ordea, hiztegiko definizioa.

Taulen informazioa kontsultatzea. SELECT agindua

SELECT sententziari esker, Datu-Baseko informazioa berreskuratuko dugu. Mota guztietako kontsultak egiteko bidea ematen du, hasi xumeenetatik eta kontsultarik konplexuenetaraino.

SELECT sententziaren formatu osoa ondorengoa da:



4.7. irudia.

SELECT klausula

- ✓ Zutabe jakin batzuk edo zutabeotatik eratorritako balioak aukeratzeko balio du.
- ✓ Zenbaki-, alfabeto- eta bestelako eragileak dauzkaten adierazpenak zehazteko bidea emango du.
- ✓ Zenbait zutabe bereizteko, komen bidez bananduko dira.
- ✓ Taularen atributu guztiak adieraziko ditu izartxo batek.
- ✓ Kontsulta zenbait taularen gainean gauzatzen denean, taularen izena edo goitizena ezarri beharra dago zutabearen aurrean.

Adibideak:

SELECT lang-zk, deitura, lanbidea, helb, alta-data, soldata, komisioa, sail-zk

FROM lang;

SELECT

FROM lang;

SELECT lang-zk, sail-zk

FROM lang;

Kontsulta Datu Basean egiten denez, zutabeen izenak aurkezpen-buru gisa erabiliko dira. Izena luzeegia, laburregia edo kriptikoegia izanez gero, kontsultarako SQL sententzia bera erabil daiteke beraren ordez, GOITIZENA sorrarazita:

SELECT sail-zk "Sail-Zenbakia", sail-izena "Saila"

FROM saila;

FROM klausula

- ✓ Zutabeak hautatzeko erabiliko ditugun taulak definitzen ditu.
- ✓ Erabiltzailea taularen jabea izan ezean, izena_erabiltzailea.izena_taula erabiliko du:

SELECT deitura "Joanen izenak"

FROM Joan.gaixoa; ✓ Izen berria taula bati lot dakioke, aldi baterako eta kontsulten ondorioetarako, baldin eta goitizen batekin lotzen bada:

> SELECT gaixoa.inskripzioa, gaixoa.deitura

FROM gaixoa

WHERE inskripzioa =63827;

SELECT e.inskripzioa, e.deitura

FROM gaixoa e

WHERE inskripzioa =63827;

✓ FROM klausularen SELECT sententzia erantsi beharko da beti ere. Taularik erabili nahi ez denean, DUALi egingo zaio erreferentzia. Taula batenak ere ez diren datuekin ari garenean erabiliko dugu.

SELECT sysdate FROM DUAL;

WHERE klausula

- ✓ Entitatearen agerraldiek, lerroek alegia, hautatuak izateko bete beharreko baldintza edo baldintzak ezarriko ditu.
- ✓ Bilaketa-irizpidearen konplexutasuna ia mugarik gabea izan daiteke.
- ✓ Alderaketa-eragileak, aritmetikoak, karaktere-kateetakoak eta logikoak erabili ahal izango dira.
- ✓ Bilaketa-irizpide anitzak erabil daitezke, baldin eta eragile boolearrak erabiltzen badira: AND, OR eta NOT. Parentesiak ere erabil daitezke, baldin eta baldintzen ebaluaketa--ordena bortxatu nahi bada.

Eragile aritmetikoak

Konstanteekin, zutabe-balioekin eta zutabe-balioen funtzioekin konbinaturik dauden adierazpenak eratzeko erabiliko dira.

- Batuketa
- Kenketa
- Biderketa
- Zatiketa

DATU-BASEAK KUDEATZEKO SISTEMAK

SELECT zut1 * zut2, zut1 – zut2

FROM taula1

WHERE zut1 + zut2 = 34;

Datetarako eragileak

Data + n Data bati n egun gehituko dizkio.

Data – n Data bati n egun kenduko dizkio.

Data1 – Data2 Bi dataren kenketa egingo du. Emaitza, egun-kopurua izango da.

Adibideak:

SELECT sysdate + 5 FROM dual;

SELECT sysdate – jaiotze-data, deitura

FROM gaixoa;

Datak letra eta zenbakien formatura eta alderantziz aldatzeko, honako funtzioak erabil daitezke:

TO_DATE eta TO_CHAR

Adibidea:

SELECT sysdate - TO_DATE('1-JAN-02')

FROM dual;

Eragile logikoak

AND TRUE itzuliko du baldin eta baldintza guztiak egiazkoak badira.

OR TRUE itzuliko du baldin eta baldintzaren bat egiazkoa bada.

NOT TRUE itzuliko du baldin eta hurrengo baldintza FALSE bada.

Karaktere-kateetarako eragileak

katea_1 | katea_2 1. katea 2.arekin uztartuko du.

Adibidea:

SELECT deitura | helbidea FROM gaixoa;

Alderaketa-eragileak

Alderatzaile orokorrak ditugu honako hauek:

Berdin

<> Ez berdin

> Handiago (baino)

< Txikiago (baino)

Handiago edo berdin >=

Txikiago edo berdin <=

Adibidea:

SELECT soldata, komisioa

FROM saltzailea

WHERE komisioa > soldata

Karaktere-kateen alderaketa-eragileak dira.

[NOT] LIKE Horri esker, ondoren adierazten diren karaktere bereziak erabil daitezke alderaketa-kateetan:

'%' 0 karaktere edo gehiago dituen edozein kate.

edozein karaktere

Adibideak:

LIKE 'Zuzendaria'

LIKE 'M%' LIKE '%X&' LIKE '__M' LIKE 'N_' LIKE '_R%'

Letra larriak edo xeheak izateak badauka zerikusia.

Alderaketa logikoetarako eragileak ditugu honako hauek:

[NOT] BETWEEN balioa_1 AND balioa_2 Bi balioren artean (biak barne)

[NOT] IN (koma_arteko_balioen_zerrenda) Balio-zerrenda IS [NOT] NULL Balio nulua da

Adibideak:

SELECT deitura, soldata

FROM langileria

soldata BETWEEN 1000 AND 2000; WHERE

SELECT deitura, jaiotze-data

FROM gaixoa

WHERE UPPER (deitura) BETWEEN 'A' AND 'B'

AND jaiotze-data BETWEEN '01-Jan-35' AND '21 Dec-45';

SELECT deitura

FROM lang

WHERE sail_zk IN (30, 40)

SELECT deitura FROM saltzailea

komisioa is null WHERE

ORDER BY klausula

- ✓ Kontsultaren emaitzaren sailkapenerako irizpidea zehaztuko du, kontsultan zerikusia duten tauletako zutabeetako edozeinen balioak oinarri hartuta.
- ✓ Balio nuluak emaitzaren hasieran egongo dira.
- ✓ Lehenespenez sekuentzia goranzkoa izanagatik ere, beheranzko sekuentzia ere zehatz daiteke.
- ✓ Ordenatzeko zenbait irizpide adieraz daiteke, eta jatorri-ordena ezkerretik eskuinerakoa izango da.
- ✓ Zutabeen ordez, zutabeen balioak dituzten adierazpenak ere adieraz daitezke.

Adibideak:

ORDER BYkod aretoa, kod ospitalea BEHER ORDER BYmod (lang_zk, 6)

DISTINCT klausula

- ✓ DISTINCT aukeraren bidez, kontsulta baten emaitza den multzoan dauden errepikaturiko lerroak ezabatuko ditugu.
- ✓ Hautatzekoak diren zutabeen gainean gauzatuko da eragiketa.

Adibideak:

SELECT DISTINCT funtzioa FROM langileria;

Multzoen hautaketa: GROUP BY eta HAVING klausulak

- ✓ Multzoka, informazio totalizatzailea lortzeko, kontsulta baten emaitza diren lerroak taldekatu nahi ditugunean erabiliko dugu.
- ✓ GROUP BY klausula ez da ordenazioak egiteko erabiltzen. Jatorrizko taulako lerroak, adierazten den adierazpenaren balioarekin bat etorrita, aldi baterako taula batean taldekatuko direla kontrolatzeko balio du.
- ✓ HAVING klausula, berriz, multzo horietatik zein ikusiko diren kontrolatzeko erabili ohi da. WHERE klausularen pareko betebeharra dauka. GROUP BY-rekin baino ezin erabil daiteke.
- ✓ Klausulen ebaluazioa exekuzio-aldian honako ordena honetan gauzatuko da: WHERE, GROUP BY, HAVING, ORDER BY.

Adibideak

Hiri bakoitzean zenbat argitaletxe dauden jakin nahi dugu:

SELECT hiria, COUNT(*) FROM argitaletxea **GROUP BY hiria:**

Gutxienez bi argitaletxe dituzten hirien gaineko kontsulta egin nahi izanez gero:

SELECT hiria FROM argitaletxea **GROUP BYhiria** HAVING COUNT (*) >= 2;

▶ SELECT sententziak dituzten adierazpenak

SELECT sententziak dituzten adierazpenak sorraraz ditzakegu, baldin eta ondorengo eragileak erabiltzen baditugu:

UNION, INTERSECT, MINUS

- ✓ Edozein SELECT sententzia baliozkoren emaitza diren lerroak har ditzakegu multzotzat, multzo hutsa barne ("no rows selected").
- ✓ Multzo-eragile guztiek beraietako bi batera konbinatuz lan egiten dute, hau da, mota honetako eragile bakoitzak, multzo ere izango diren bi eragigai behar ditu.

UNION

Lehen multzoko lerro guztiak konbinatuko ditu bigarreneko lerro guztiekin. Emaitza den multzoan gertatzen den lerro-errepikapen guztiak lerro bakarrera murriztuko da.

Adibidea:

Hautatu 22 ospitalean lanean diharduten barneko guztiak eta bertako sendagileak:

SELECT deitura

FROM langileria

WHERE funtzioa = 'Barnekoa' AND kod_ospitalea = 22

UNION

deitura SELECT FROM doktorea

WHERE kod_ospitalea = 22

INTERSECT

Sarrera-multzoko lerroak aztertu eta bietan ageri direnak itzuliko ditu. Errepikaturik dauden lerro guztiak ezabatu egingo dira emaitza den multzoa sorrarazi aurretik.

Adibidea:

Hautatu edozein ospitaletan lan egiten duten eta barneko sendagile diren langile guztiak:

SELECT deitura

FROM langileria

WHERE funtzioa= 'Barnekoa'

INTERSECT

SELECT deitura

FROM doktorea;

MINUS

Lehenengo multzoan bai, baina bigarrenean ez dauden lerroak itzuliko ditu. Lehenengo multzoko lerro errepikatuak lerro bakarrera murriztuko dira beste multzoarekiko alderaketari ekin baino lehen.

Adibidea:

Hautatu edozein ospitaletan lan egiten dutelarik, barneko doktoreak ez diren edo bestelakoak diren langileak.

SELECT deitura

FROM doktorea

MINUS

SELECT deitura FROM langileria

WHERE funtzioa = 'Barnekoa'

Multzo-eragileak edozein konbinaziotan katea daitezke, eta multzoak ezkerretik eskuinera ebaluatuko dira. Parentesiak erabilita, aurrekotasuna derrigor daiteke.

Adibidea:

```
(SELECT ....UNION SELETC) INTERSECT ...
eta beste gauza bat litzateke honakoa:
SELECT ... UNION (SELECT ... INTERSECT...)
```

Hainbat taularen gaineko kontsultak

Taula bakarra hartu izan dute oinarritzat orain arte erabilitako SQL agindu guztiek. Hala ere, hainbat taulatan egin daiteke kontsulta aldi berean, eta horretarako hauxe erabiliko dugu:

- ✓ Konbinazioak (JOINS)
- ✓ Azpi-kontsultak (SUB-QUERIES)

KONBINAZIOAK (JOINS)

Beraiei esker, taula biren edo gehiagoren informazio konbinatua lor daiteke. Hobeto ulertzeko, ikus dezagun adibide bat:

Demagun mediku-proba jakin batzuen emaitza zein den jakin nahi dugula, eta behar dugun informazioa daukaten TESTA eta LABORATEGIA izeneko bi taula ditugula:

LAB-ZK	IZENA	HELBIDEA	TELEFONOA
16	Atcon	Gran Via, 14	532-4433
42	Clini	Castellana, 55	447-6484
56	Alpha	Juan Bravo, 18	929-9611
84	Nucro	Alcala, 62	368-9793

INSKRIPZIOA	LAB-ZK	TCO	EMA
10995	16	ABC	POS
14024	42	PTY	NEG
10995	16	BCD	NEG

Horixe izango da SELECT bi sententzia hauek erabiliz gero:

SELECT* SELECT* FROM laborategia FROM testa;

Demagun laborategiaren zenbakia jakiteko interesa izan beharrean, laborategiaren izena jakitea interesatzen zaigula. Bi taulak elkartu beharko ditugu behar dugun informazio hori lortuko badugu. Ondorengo sententzia honekin egingo dugu proba:

SELECT I.lab_zk, I.izena, I.telefonoa, t.lab_zk, t.tkodea

t.inskripzioa, t.emaitza

FROM laborategia I, test t;

Ondorengoa izango da emaitza:

LAB-ZK	IZENA	TELEFONOA	LAB-ZK	TCO	INSKRIP.	EMAITZA
16	Atcon	532-4453	16	ABC	10995	POS
42	Clini	447-6448	16	ABC	10995	POS
56	Alpha	929-9611	16	ABC	10995	POS
84	Nucro	368-9793	16	ABC	10995	POS
16	Atcon	532-4453	42	PTY	14024	NEG
42	Clini	447-6448	42	PTY	14024	NEG
56	Alpha	929-9611	42	PTY	14024	NEG
84	Nucro	368-9793	42	PTY	14024	NEG
16	Atcon	532-4453	16	BCD	10995	NEG
42	Clini	447-6448	16	BCD	10995	NEG
56	Alpha	929-9611	16	BCD	10995	NEG
84	Nucro	368-9793	16	BCD	10995	NEG

12 erregistro daude hautaturik

Ikus daitekeenez, ez da zehatz-mehatz gu bila gabiltzana. Aurreko bi taulen "biderkadura cartesiarra" dugu taula hau. Horrela, ORACLEk LABORATEGIA taularen lehenengo lerroa hartu eta TESTA taularen lerro guztietan erantsi du. Eragiketa berbera errepikatu du LABORATEGIA izeneko taularen lerro guztietarako.

Atera zaigun taulari begiratu eta 10995 zenbakidun bezeroa, 16 zenbakidun laborategia (Atcon) duena hartuz gero, zuzena den lerro bat dagoela ikus dezakegu: gure adibidearen lehenengo lerroa da. Lerro honek, gainerakoek ez bezala, eta azken horiek ez daukate zentzurik, laborategi-zenbakiak elkartu ditu.

Biderkadura cartesiarraren multzo bat hautatu, horixe da egin behar duguna eta, hain zuzen ere, zuzenak diren lerroak. Eragiketa hori gauzatu ahal izateko, WHERE klausula erabili beharko dugu:

SELECT I.lab-zk, I.izena, I.telefonoa, t.lab-zk, t.tkodea

t.inskripzioa, t.emaitza

FROM laborategia I, testa t WHERE I.lab-zk = t.lab-zk;

Honakoa da emaitza:

LAB-ZK		TELEFONOA		тсо	INSKRIP.	EMAITZA
16	Atcon	532-4453	16	ABC	10995	POS
42	Clini	447-6448	42	PTY	14024	NEG
16	Atcon	532-4453	16	BCD	10995	NEG

Demagun inskripzio-zenbaki hutsaz gain, bezeroaren izena zein den ere jakin nahi dugula. Jakina, GAIXOAK izeneko taulan egongo da bezeroaren izena. Beste taula hori lotu beharra daukagu eta horretarako GAIXOA eta TEST taulen arteko zutabe komuna aurkitzea zaigu beharrezkoa.

SELECT I.izena, I.telefonoa, t.kodea

g.deitura, t.emaitza

FROM laborategia I, testa t, gaixoa g

WHERE I.lab-zk = t.lab-zk AND

t.inskripzioa = e.inskripzioa

IZENA	TELEFONOA	тсо	EMAITZA	DEITURA
Atcon	532-4453	ABC	POS	Laguía, M.
Clini	447-6448	PTY	NEG	Laguía, M.
Atcon	532-4453	BCD	NEG	Fernández, M.

Adibide hori ikusita, konbinaziorako sintaxi orokorra ondorengo hau dela esan dezakegu.

SELECT "from2 klausulan aipatzen diren tauletako zutabeak

FROM taula 1, taula2...

WHERE taula1.zutabea = taula2.zutabea

- ✓ WHERE klausulari, zeinek konbinazioren baldintza zehazten duen, ezikusi eginez gero, biderkadura kartesiarra izango da emaitza.
- ✓ Biderkadura kartesiarrak taula bateko lerro guztiak beste taula bateko ilarekin uztartuko ditu.
- ✓ Ikusgai ageri zaizkigun lerro bakarrak, WHERE klausulan zehazturiko irizpide guztiak uztartzen dituztenak izango dira.

- ✓ Hona hemen konbinaziorako zenbait arau:
 - a) Nahi adina taula elkar daitezke.
 - b) Tauletarako uztartze-irizpideari konbinazio-predikatua edo konbinazio-irizpidea deritzo.
 - c) Zutabe-bikote bat baino gehiago erabil daiteke bi taularen arteko konbinaziorako baldintza zehazteko, taulak direnak direlarik ere.
 - d) SELECT sententzian taula guztietako zutabeak aipa daitezke.
 - e) FROM klausulan ageri diren tauletan izen berbera duten zutabeak izanez gero, izena_taula.izena_zutabea zehaztu beharko da. Hala ere, zutabearen izena taula bakar batean agertzen bada, ez da horrelakorik egin behar izango, nahiz eta eginez gero irakurterrazagoa izango litzatekeen.

✓ Konbinazio-motak:

PAREKO KONBINAZIO (EQUAL JOIN)

- 1. Konbinaziorako irizpidea berdin (=) eragilearen bidez ezartzen zaiona da.
- 2. JOIN komandoan aipatzen diren zutabeetako bakoitzean balio parekatuak dituzten lerroak berreskuratzeko erabili ohi da.

KANPOKO KONBINAZIO (OUTER JOIN)

- 1. Bi taulatan erregistraturiko informazio guztia errekuperatuko du, tauletako batean beste taulan pareko lerrorik ez duten lerroak daudelako, konbinaziorako baldintza bete ez arren.
- 2. Horretarako "+" gehituko da dagokion lerroa agertzerik espero ez dugun WHERE klausularen zutabean.
- 3. Ez dago bi tauletako zutabe berberean "+" aldi berean jartzerik. Horretarako bi kontsulta egin beharko da, bi SELECT erabilita, eta ELKARKETAren bidez elkartu.
- 4. Lerroa ageri ez den taulari, konbinaziorako baldintza bete zedin, erabat nulua litzatekeen lerroa gehituta lortuko genukeen emaitza berbera lortuko genuke honela ere.

Adibidea: Esperimentu bakoitzari dagokion informazioa errekuperatu bi taulatan erregistraturik, bata ESPERIMENTUAK eta beste EMAITZAK izenekoetan, hain zuzen. Aintzat hartu 30. esperimentuan oraindik emaitzarik ez delako kasua.

SELECT *

FROM esperimentuak;

KODE-ZK **ESPERIMENTUA**

esp1

esp2

esp3

SELECT *

FROM emaitzak;

KODE-ZK	EMA1	EMA2
10	2	100
20	2	1000

EQUAL JOIN delakoa erabiliz gero:

SELECT e.kode-zk, e.esperimentuak, r.ema1, r.ema2

FROM esperimentuak e, emaitzak r

WHERE e.kode-zk = r.kode-zk;

Honako hau izango litzateke emaitza:

KODE-ZK	<u>ESPERIMENTUA</u>	EMA1	EMA2
10	esp1	2	100
20	esp2	2	1000

OUTER JOIN erabiliz gero, berriz:

SELECT e.kode-zk, e.esperimentuak, r.ema1, r.ema2

FROM esperimentuak e, emaitzak r WHERE e.kode-zk = r.kode-zk (+);

KODE-ZK	ESPERIMENTUA	EMA1	EMA2
10	esp1	2	100
20	esp2	2	1000
esp3			

SELF JOIN

- 1. Taula bat bere buruarekin konbinatzean datza.
- 2. Kasu honetan, bi label ezberdin eslei diezaiokegu taula berari, bi taula ezberdin izango balira bezala erabili ahal izateko.

LANG (BEHARGINA)

EMPNO ENAME MGR 7839 KING 7698 **BLAKE** 7839 7782 CLARK 7839 7839 7566 **JONES** 7654 MARTIN 7698 7499 ALLEN 7698

LANG (NAGUSIA)

EMPNO	ENAME
7839	KING
7839	KING
7839	KING
7698	BLAKE
7698	BLAKE

SELECT behargina.ename, nagusia.ename

FROM lang behargina, lang nagusia WHERE behargina.mgr = nagusia.empno

Azpikontsultak (SUB-QUERIES)

Nahi ditugun datuen azpimultzoak definitzeko erabili izan dugu WHERE klausula orain arte. Zutabe bateko balio bat konstante batekin edo konstante-multzo batekin parekatzeko SQL erabiliz gauzatu ditugu. Zer gertatzen da, baina, konstanteen balioak zein diren ez dakigunean? Ikus dezagun adibide bat:

Batezbestekoaren gainetik dauden soldatak dituzten langileen izenak jakin nahi ditugu.

Lehenengo eta behin, batezbesteko soldata zein den jakin beharko dugu, instrukzio hau erabilirik:

SELECT AVG (soldata)

FROM langileria;

Soldaten batez bestekoa zein den dakigunean, ondorengo instrukzio hau burutu beharko dugu:

SELECT deitura

FROM langileria

WHERE soldata > ...

Bada hori guztia hobetzerik; izan ere, SQL-ri esker SELECT-en adierazpen osoa sar daiteke WHERE komandoaren atal gisa: HABIARATURIKO AZPIKONTSULTA dugu.

Goiko bi adierazpenak erabili beharrean, erraz kondentsatuko ditugu, ondoren adierazten den bezala:

SELECT deitura **FROM** langileria

WHERE soldata > (SELECT AVG (SOLDATA)

FROM langileria)

- ✓ Lehenbizi kontsulta exekutatuko dugu eta, ondoren, kontsulta nagusian sartuko dugu ateratako balioa.ententzia baten bidez errekuperaturiko balio-multzoa.
- ✓ Hona sintaxi orokorra:

SELECT... FROM... WHERE balio_zehaztua alderaketa_eragilea (SELECT... FROM... WHERE...)

AZPIKONTSULTAREN EMAITZA	ALDERAKETA-ERAGILEA
Balio bakuna	Aritmetika-karakterea
Balio bat baino gehiago	Logikoa

Azpikontsulten motak

• Balio bakarra itzultzen duten azpikontsultak

JIMENEZek duen lan berbera duten langileen izenak eta lanbideak aurkeztu.

SELECT deitura FROM langilea

lanbidea = (SELECT WHERE lanbidea

> FROM lang

WHERE UPPER (deitura) = 'JIMENEZ');

• Balio bat baino gehiago itzultzen dituzten azpikontsultak, balioetako batekin alderatu behar delarik

30. saileko edozein langilek baino diru gehiago irabazten duten langileei buruzko informazioa:

SELECT

FROM lang

WHERE soldata > ALL (SELECT soldata

FROM lang

WHERE sail-zk = 30);

Beste era batera:

SELECT

FROM lang

WHERE soldata > (SELECT MAX(soldata)

> FROM lang

WHERE sail-zk = 30);

• Balio-zerrendak itzultzen dituzten azpikontsultak

SALMENTAK saileko langileetako edozeinek duen lanaren modu-modukoa duten 20 saileko langile guztien izenak eta lanbideak lortu.

```
SELECT deitura, lanbidea
FROM
         lang
WHERE
         sail-zk = 20
         AND lanbidea IN (SELECT lanbidea
                          FROM
                                  lang
                          WHERE sail-zk IN
                                    (SELECT sail-zk
                                     FROM saila
                                     WHERE UPPER (sizena = 'SALMENTAK'));
```

Adibide hori erabilita, honakoak ikus ditzakegu:

- ✓ Habiaratuta dagoen SELECT batek beste SELECT habiaratu bat eduki dezake. Mugagabea da habiaratzeak darabiltzan mailen kopurua.
- ✓ Kasu honetan, erabili dugun alderaketa-eragilea IN da, zeren barruko kontsultak lerro bat baino gehiago itzultzen baitu.
- ✓ Barruko kontsultak lerroak itzultzeko darabilen ordenak ez du garrantzirik eta, ondorioz, ezin zehatz daiteke.

Lortu MADRILen edo BARTZELONAn lan egiten duten langile guztiei buruzko datuak.

```
SELECT
          deitura, lanbidea, alt-data, sail-zk, soldata
FROM
          lang
WHERE
          sail-zk IN
                 (SELECT sail-zk
                  FROM saila
                  WHERE loc IN ('MADRIL', 'BARTZELONA'))
ORDER BYdeitura;
```

Kanpoko kontsultak edo kontsulta nagusiak balioak berreskuratzeko erabiltzen duten taula ez den beste taula batetiko balioak errekupera ditzake barruko kontsultak.

• Zutabe bat baino gehiago itzultzen dituzten azpikontsultak

JIMENEZek dituen lanbidea eta soldata berbera dituzten langileen zerrenda lortu.

SELECT deitura FROM lang

WHERE (lanbidea, soldata) =

> (SELECT lanbidea, soldata

FROM lang

WHERE UPPER (deitura) = 'JIMENEZ');

- ✓ Hainbat kontsulta zehaztuz gero, WHERE klausula zutabe bakarra zehazten badugu baino askozaz mugatuagoa izango da.
- WHERE klausulan zenbait eremu xehatuz gero, parentesi artean sartu beharko dira.
- ✓ Parentesi arteko zutabeek edo adierazpenek bat etorri beharko dute, zenbakiari eta datu-motari dagokienez, aurreko kontsultako SELECT zerrendak bere dituenekin.
- ✓ WHERE klausularen zutabe bakoitzak, azpikontsultan egokitzen zaionaren gainekoa izan beharko du.

• Hainbat azpikontsulta

Erakutsi JIMENEZek egiten duen lan berbera edo FENANDEZek besteko soldata edo berak baino soldata handiagoa duten langileak (izena, lanbidea, soldata eta kontratazio-data).

SELECT deitura, soldata, lanbidea, alt-data

FROM lang

WHERE lanbidea =

> (SELECT lanbidea **FROM** lang

WHERE UPPER (deitura) = 'JIMENEZ')

OR

Soldata > =

(SELECT soldata

FROM lang

WHERE UPPER (deitura) = 'FERNANDEZ')

4.5 Eranskina: funtzioak

Funtzio aritmetikoak

ABS (n) Xedea: "n"-ren balio absolutua itzultzen du.

Adibidea:

SELECT ABS(-15) "ABSOLUTUA" FROM DUAL;

Emaitza: ABSOLUTUA

15

CEIL (n) Xedea: "n"-ren balioa edo goragoko hurrengo balio osoa itzultzen du.

Adibidea:

SELECT CEIL (15.7) "GOIKOA" FROM DUAL;

Emaitza: **GOIKOA**

16

FLOOR (n) Xedea: "n"-ren balioa edo beheragoko hurrengo balio osoa itzultzen du.

Adibidea:

SELECT FLOOR (15.7) "BEHEKOA" FROM DUAL;

Emaitza: **BEHEKOA**

17

MOD (m,n) Xedea: "m" zati "n" egitean gertatzen den hondarra itzultzen du.

Adibidea:

SELECT MOD (11,4) "HONDARRA" FROM DUAL;

Emaitza: **HONDARRA**

3

POWER (m,n) Xedea: "m"-ren "n" berretuaren balioa itzultzen du.

Adibidea:

SELECT POWER (3,2) "POTENTZIA" FROM DUAL;

Emaitza: **POTENTZIA**

Xedea: "m" hamarrenetan biribildutako "n" balioa itzultzen du. ROUND (n [,m])

Adibidea:

SELECT ROUND (1.5634,1) "BIRIBILTZEA" FROM DUAL;

Emaitza: **BIRIBILTZEA**

1.6

SQRT (n) Xedea: "n"-ren erro karratua itzultzen du. "n"-ren balioa ezin izan daiteke negatiboa.

Adibidea:

SELECT SQRT (4) FROM DUAL;

Emaitza: SQRT(4)

2

TRUNC (n[,m]) Xedea: "n" itzultzen du, "m" hamarren bilakaturik. "m" negatiboa izan daiteke.

Adibidea:

SELECT TRUNC (1.5634, 1) FROM DUAL;

Emaitza: TRUNC (1.5634,1)

1.5

(Zenbakia digitu dezimal bilakaraziko du)

Karaktere Kateen funtzioak

CHR (n) Xedea: balio bitarra "n"-ren pareko duen karakterea itzultzen du.

Adibidea:

SELECT CHR (75) "CARACTER" FROM DUAL;

Emaitza: <u>CHR</u>

Κ

CONCAT (char1, char2) Xedea: "CHAR1" itzultzen du, "CHAR2"-rekin kateatuta.

(||) eragilearen pareko da funtzio hau.

Adibidea:

SELECT CONCAT (KATEAT (deitura, 'es'), lanbidea)

"KATEATZEA" FROM EMP;

Emaitza: **KATEATZEA**

SANCHEZ LANGILEA DA

LOWER (char) Xedea: "char" katea itzultzen du, letra guztiak xehe bilakaturik.

Adibidea:

SELECT LOWER ('MARCOS JAUNA') LETRA XEHEA;

Emaitza: LETRA XEHEA

marcos jauna

LPAD (char1,n [,char2])

Xedea: "char1" itzultzen du "n" luzeraz eta eskuinera doiturik. "char2", berriz, ezkerretik betetzeko erabiltzen den katea izango da.

Adibidea:

SELECT LPAD ('P', 5, '*') "ADIBIDEA" FROM DUAL;

Emaitza: **ADIBIDEA**

****P

LTRIM (char [,set])

Xedea: "char" itzultzen du "set" karaktere-multzoa katearen ezkerretik alde batera utzita.

Adibidea:

SELECT LTRIM ('xyxXxyLAST WORD', 'xy') "ADIBIDEA" FROM DUAL;

Emaitza: **ADIBIDEA**

XxyLAST WORD

REPLACE (char, search_string [,replacement_string])

Xedea: "char" itzultzen du "replacement string"-en bidez ordezkaturiko "search string" delakoa ren gertaldi bakoitzean.

Adibidea:

SELECT REPLACE ('JACK AND JUE','J', 'BL') "AALDAKETAK" FROM DUAL;

Emaitza: **ALDAKETAK**

BLACK AND BLUF

RPAD (char1,n [,char2])

Xedea: LPAD funtzioaren modukoa, baina eskuinetik beteaz.

Adibidea:

SELECT RPAD (deitura, 11, '*') "ADIBIDEA" FROM DUAL;

Emaitza: **ADIBIDEA**

Diaz B.****

Hernandez J.

RTRIM (char[,set])

Xedea: LTRIM funtzioak duenaren antzekoa, baina katearen eskuinean alde batera utzita

SUBSTR (char, m[,n])

Xedea: "m" zenbakidun posiziotik "n" zenbakiduneraino bere baitara biltzen duen azpikatea itzultzen du. "n"-ren balioa ezin izan daiteke 1etik beherakoa. "m"-ren balioa, berriz, negatiboa izan daiteke.

Adibidea:

SELECT SUBSTR ('ABCDEFG', 3, 2) "azpikatea" FROM DUAL;

Emaitza: AZPIKATEA

CD

TRANSLATE (char, from, to)

Xedea: "char" itzultzen du "from2-en dauden karaktereak "to" karaktere bilakaturik. "from" argudioak "to" delakoak baino karaktere gehiago eduki dezake.

SELECT TRANSLATE ('2KRW229', '0123456789ABCDEFGHIUKLMNOPQRSTUVWXYZ',

"ADIBIDEA" FROM DUAL

Emaitza: ADIBIDEA 9XXX999

UPPER (char)

Xedea: "char" itzultzen du, eduki guztia letra larrietan emanda.

Adibidea:

SELECT UPPER ('Zabala') "LETRA LARRIAK" FROM DUAL;

Emaitza: LETRA LARRIAK ZABALA

Zenbaki-balioak itzultzen dituzten funtzioak

ASCII (char)

Xedea: "char" katearen ASCII balioa itzultzen du.

Adibidea:

SELECT ASCII ('Q') FROM DUAL;

Emaitza: ASCII ('Q')

81

INSTR (char1, char2 [,n [,m]])

Xedea: "char1"-en emegarren gertaldia itzultzen du "char2"n, bilaketari "n" posizioan ekinda. Adibidea:

SELECT INSTR ('ORACLE CORPORATION', 'OR', 3, 2) "ADIBIDEA" FROM DUAL;

Emaitza: **ADIBIDEA**

12

LENGHT (char)

Xedea: "char" posizioen kopurua itzultzen du.

Adibidea:

SELECT LENGHT ('IKASTURTEA') "LUZERA" FROM DUAL;

Emaitza: LUZERA

5

Fitxak erabiltzeko funtzioak

ADD_MONTHS (d,n)

Xedea: "d" data itzultzen du, "n" hilabetetan gehiturik.

Adibidea:

SELECT ADD_MONTHS(DATA_ALT, 2) FROM EMP;

LAST_DAY (d)

Xedea: "d" daukan azken hilabetearen azken egunari dagokion data itzultzen du.

Adibidea:

SELECT LAST DAY ('12-OTS-95') FROM DUAL;

MONTHS_BETWEEN (d1, d2)

Xedea: "d1"-i eta "d2"-ri dagozkien daten artean hilabetetan dagoen aldea itzultzen du.

Adibidea:

SELECT MONTHS BETWEEN('01-URT-96', '20-MAR-95') FROM DUAL;

NEXT_DAY (d, char)

Xedea: "d" dataren ondorengo "char" astearen lehenbiziko egunari dagokion data itzultzen du.

Adibidea:

SELECT NEXT_DAY (sysdate, 'asteartea') FROM DUAL;

Bihurketa-funtzioak

TO-NUMBER (char[, fmt [, nlsparams']])

Xedea: "char" katea (zenbaki bat daukan katea) "NUMER" erako bilakaraziko du adierazten den "fmt" formatupean.

Adibidea:

SELECT TO NUMBER ('123456) "ZENBAKIA" FROM DUAL;

Emaitza **ZENBAKIA**

123456

TO_CHAR (d, fmt [, nlsparams'])

Xedea: "d" data "VARCHAR2" erako bilakarazten du zehaztutako "fnt" formatuan. "nlsparams" parametroak zehaztuko da hilabetea eta eguna berreskuratzeko lengoaia.

Adibidea:

SELECT TO CHAR (ALT-DATA, 'Month DD, YYYY') "DATA BERRIA" FROM ENT;

Emaitza: **DATA BERRIA**

> December 17, 1980 February 20, 1980

TO_DATE (char [,fmt[,'nlsparama']])

Xedea: 'VARCHAR2'motako "char" "DATE" motako balio bilakarazten du. "FMT", berriz, datarako hautatzen den formatua izango da.

Adibidea:

SELECT TO-CHAR (TO_DATE ('940101', 'yymmdd), 'Month', 'NLS_DATE_LANGUAGE = Basque') "HILABETEA" FROM DUAL.

Emaitza: **HILABETEA**

Urtarrila

TO_CHAR-ek eta TO_DATE-k formatu-maskarak darabiltzate. Datuen bilakaerarako funtzioen kontrolerako bidea emango duten azken datuaren "pictures" direlakoak dira.

Zenbakien formatu-maskarak

cc edo scc Mendearen balioa

y,yyy edo sy,yyy Urtea komaz, zeinuarekin edo zeinurik gabe

yyyy Urtea zeinurik gabe

Urtearen azken hiru digituak ууу Urtearen azken bi digituak уу Urtearen azken digitua У Zenbatgarren hiruhilabetea q ww Urteko zenbatgarren astea Hileko zenbatgarren astea W Urteko zenbatgarren hilabetea mm ddd Urteko zenbatgarren eguna Hileko zenbatgarren eguna dd Asteko zenbatgarren eguna

hh edo hh12 Ordua (1-12)
hh24 Ordua (1-24)
mi Minutuak
ss Segundoak

ssss Gauerditik igarotako segundoak

j Juliotarra

▶ Karaktereen Formatu-maskarak

syear edo year Urtea ingelesez (ad.:nineteen-eighty-two)

month Hilabetearen izena

mon Hilabetearen izenaren laburdura hiru letratan

day Asteko egunaren izena

dy Egunaren izenaren laburdura hiru letratan

a.m. edo p.m. Ante-meridiam edo post-meridiam (goiza edo arratsaldea)b.c. edo a.d. Urtearentzako adierazlea (Kristo aurretik edo Kristo ondoren)

Formatu-maskaren atzizkiak

th Zenbakiaren ordinala (st, nd, rd, th) (ad.: 4th)

sp Zenbaki baten balioari dagokion testua idatziko du (ad.: twenty-five)

spth edo thsp Aurreko bi ekintzen konbinazioa) (ad: twenty-fifth)

Beste funtzio batzuk

DECODE (var, val1, val2, cod2, ... lehenespenezko_balioa)

Baldin eta var berdin zerrendaren edozein balio, dagokion kodea itzuliko du; bestela bada, ordea, lehenespenezko baliotzat adierazitako balioa itzuliko du. Val eta var mota bereko datuak izango dira ezinbestean.

SELECT DECODE (izena

'Susperraldia' 'Hobetzen' 'Larria' 'Zainketa Intentsiboak' 'Psikiatrikoa' 'Burukoa' 'Bihotzekoa' 'Kardiologia'

'Azterketa-faseko gaixoa')

FROM aretoa

WHERE...

SELECT sententzia horrek aretoaren izenari lotuta dagoen balioa itzuliko du, DECODE funtzioaren argudio den zerrendaren arabera. Aretoaren jatorrizko izena ere itzuliko da, erreferentzia gisa.

GREATEST (adier, adier, ...)

Zerrendako baliorik handiena itzuliko du.

LEASTEST (adier, adier, ...)

Zerrendako baliorik txikiena itzuliko du (GREATEST-ek egiten duenaren kontrakoa).

NUL (x, adier)

Baldin eta x NULL bada, expr itzuliko du; NULL ez bada, ostera, x itzuliko du. Edozein motatakoa izan daiteke x-en eta adier-en balioa. Emaitza den balioa x-en mota berekoa izango da.

- ✓ Adierazpen aritmetikoetan eta funtzioetan balio nuluak saihesteko erabiltzen da.
- ✓ Balio nuluek ez dute esku hartzen taldeei begira eginiko funtzioetan.
- ✓ Adierazpenetako balio nuluek beti izango dute balio nulua emaitza gisa.
- ✓ Hortaz, aurreko bi puntuetan deskribatu diren ekintzak saihestu nahi ditugunean. erabiliko dugu funtzio hau.

INFORMATIKAKO SISTEMEN ADMINISTRAZIOA

SELECT soldata + NVL (komisioa, 0) FROM lang WHERE lanbidea = 'SALTZAILEA'

Kontuan hartu behar da NVL funtzioa erabili ezean, komisioan balio nulua izanez gero nulua izango zela emaitza, soldatari dagokion zutabearen balioa dena delarik ere.

SOUNDEX (adier)

"adier"-en irudikapen fonetikoa daukan karaktere-katea itzuliko du.

Multzoen funtzioak

AVG(n)

n-ren batez besteko balioa (balio nuluak alde batera lagata).

COUNT ({* | adier})

Adier-ek balio nulua ez duen daturen bat zenbat aldiz ebaluatzen duen. '*' aukerak hautaturiko lerro guztiak zenbatzen ditu.

MAX (adier)

Adier-en gehienezko balioa.

MIN (adier)

Adier-en gutxieneko balioa.

STDDEV (adier)

Adier-ek izan ohi duen desbideraketa balio nuluak kontuan izan gabe.

SUM (adier)

Adier-en balioen batuketa.

VARIANCE (adier)

Adier-en aldakuntza balio nuluak kontuan hartu gabe.

DISTINCT eta ALL klausulekin erabil daitezke funtzio hauek.

Transakzioak kontrolatzeko komandoak 4 6

Datu-multzo baten gainean exekutatzen diren eta lanerako unitate logikoa (LUL) osatzen duten eragiketak dira transakzioak. Hau da, unitate gisa tratatzen diren SQL sententzien sekuentzia dugu LUL. Sententzia horiek gauzatzen dituzten aldaketa guztiak iraunkor bihurtuko dira edo aldi berean desegingo. Bi sententzia erabil daitezke.

COMMIT sententzia

Azken COMMIT-etik edo ROLLBACK-etik exekutaturiko INSERT, UPDATE edo DELETE sententzien bidez gauzaturiko aldaketak baieztatuko ditu COMMIT sententziak. Erabiltzaileak egin izan dituen aldakuntzei onirizpena eman diela eta BD- an iraun dezaten nahi duela esan nahi du horrek.

SQLPLUS-i esker, automatikoki baliozkotu ditzakegu geure transakzioak, era esplizituan adierazten ibili gabe. Horretarako balio du AUTOCOMMIT komandoak. Parametro horren balioa SHOW aginduaren bidez erakuts daiteke, era honetara:

> SQL> SHOW ATOCOMMIT; autocommit off

OFF dugu ez egitezko balioa eta, horrela, transakzioak ez dira behin-betikoak izango COMMIT egiten ez dugun bitartean. Behin-betikoak izan daitezen nahi badugu, AUTOCOMMIT parametroari eragin beharko diogu, SET agindua emanda:

SQL> SET AUTOCOMMIT ON;

ROLLBACK sententzia

ROLLBACK sententziak, INSERT, UPDATE edo DELETE sententzien bidez, azken COMMIT-etik edo ROLLBACK-etik egindako aldaketak desegingo ditu. Erabiltzaileak, egin izan dituen aldaketak azkenean BD-n gordetzea ez duela nahi esan nahi du horrek.

COMMIT edo ROLLBACK baten eta hurrengoaren artean, bi irudi edo balio gordeko dira aldatutako datu bakoitzeko: lehenagoko irudia edo "lehen" irudia eta irudi berria edo "gero" irudia. Lehenagoko balioa aldi baterakoa den eta ROLLBACK-en segmentua deritzon alderdian gordeko da. Balio berria, bere aldetik, BD-n gordeko da lehenagoko balioa biltegiratu den ROLLBACK-en segmentuaren tokiari erreferentzia eginez.

Horrela, erabiltzaileak COMMIT egiten duenean, "lehen" irudiak gordeta zeuzkan aldi baterako alderdia hustu egiten da (eta horrela beste erabiltzaile batzuei eman dakieke tarte hori) eta baieztatu gisa markatuko dira balioak, lehenagoko balioari eginiko erreferentzia ezabatuz.

Erabiltzaileak ROLLBACK eginez gero, ukitutako erregistroen lehenagoko balioak berreskuratuko ditu ROLLBACK-en segmentutik, eta BD-n idatziko dira gordetzen joaniko "gero" irudiak zehaturik. Horrela, BD aldakuntzak egin aurretik zegoen bezala geratuko da. Gainera, dagozkion sarrerak libre utziko dira ROLLBACK-en segmentuan, beste aldakuntza batzuetarako tarte hutsak egon daitezen.

Sisteman izaten diren arazo jakin batzuen aurrean (esate baterako, argia doanean), ROLLBACK automatikoa gertatuko da eta, horrela, gauzaturiko aldaketak baieztatu gabe dituen erabiltzaileak aldakuntza guztiak gal ditzake.

Lehenespenez, irteera normal batek COMMIT delakoa eragingo du. Adierazi gabe ere, COMMIT inplizitua exekutatzera behartzen duten zenbait SQL agindu daude.

Transakzioak COMMIT, ROLLBACK edo BD-rekiko konexio baten ondoren ematen den lehenengo sententzia exekutagarriarekin batera hasten direla esan dezakegu. COMMIT, ROLLBACK edo BD-tiko deskonexioa izango dira, berriz, transakzioen amaiera.

Atal honetara bilduta dagoen beste sententzia bat honako hau litzateke:

SAVEPOINT sententzia

Sententzia honen bidez puntu bat identifikatuko dugu transakzioan eta, geroago, ROLLBACK eginez, berriro itzuli ahal izango gara puntu horretaraino. Honakoa da sintaxia:

SAVEPOINT <izena_puntua_haustura>

Eta segurtasun-kopiaren puntu baterainoko ROLLBACK-a egiteko sintaxia, berriz, beste hau litzateke:

ROLLBACK TO <izena_puntua_haustura>

Gauzatzen dihardugun transakzioaren atalak desegiteko erabili ohi dira SAVEPOINT direlakoak, ROLLBACK komandoarekin batera. Beste era batera esanda, aurreko COMMIT/ROLLBACK-era bitarteko ROLLBACK osoa egiten ibili beharrean, SAVEPOINT-aren ondorengo aldaketak besterik ez dugu desegingo.

5 SQL*Plus

SQL*Plus-en ingurua

Oracle-ren produktua da SQL*Plus. Oracle Datu Basean kontsulta egin eta berori eguneratu ahal da bertatik, BD erlazionaletara sartzeko SQL lengoaiaren bidez.

SQL datu-basearen lengoaia eta prozedura-luzapenerako PL/SQL lengoaia konbinatuta erabili ahal izango dugu SQL*Plus.

Oso kontuz ibili beharra dago SQL lengoaia (Oracle tresna guztietatik erabilia izan daitekeena) SQL*Plus-en komandoekin ez nahasteko.

Ondorengo funtzio hauek dauzka SQL*Plus-ek:

- ✓ Elkarreraginezko lana Oraclerekin, SQL-ren bidez.
- ✓ Erabiltzailearen izena eta kontrazeinua behar den bezala sartutakoan, SQL*Pluserako sarbidea izango dugu.
- ✓ Berehala exekuta daitezke SQL komandoak. Buffer batean biltegiratu eta Oraclera pasako dira.
- ✓ Laguntza elkarreragilea dauka.

5.2 SQL*Plus abian jartzea

SQL*Plus-en sartzeko, dagokion ikonoan kliskatuko dugu. Datu Baserako baliagarria den erabiltzaile--izena eta bere pasahitza tekleatu beharko ditugu.



5.1. irudia.

Datuak sartuta daudenean, ondorengo honen antzekoa den mezua agertuko da:

SQL*Plus: Release 8.0.4.0.0 - Production on Mar jul 30 10:0:34 2002

(c) Copyright 1997 Oracle Corporation. All rights reserved.

Conectado a:

Oracle8 Personal Edition Release 8.0.4.0.0 - Production PL/SQL Release 8.0.4.0.0 - Production SQL>

5.2 irudia.

Zenbakiak eta erabilitako idazkera bat edo beste izango dira erabiltzen dugun Oracleren bertsioaren arabera.

SQL*Plus-en saio aktiboa utzi eta sistema eragilera itzuli nahi izanez gero, EXIT idatzi behar da.

5.3 Komandoak sartzea eta egikaritzea

SQL*Plus-en prompt delakotik (gehienetan SQL> izaten da), bi motako komando ezberdinak sar ditzakegu.

- ✓ SQL komandoak datu-baseko informazioa erabilita lan egiteko.
- ✓ SQL*Plus komandoak, formatu zehatza duten txostenak lortzeko.

SQL komandoa sartzen dugunean, SQL*Plus-ek exekutatu eta prompt-a itzuliko du beste komando bat (edo komando berbera) sartzeko. SQL*Plus komando bat lerro batean sartzerik ez badago, "-" jarraipen-karakterea dago, hurrengo lerroan jarraitzeko.

Komandoak sartzeko erak ez dauka inolako konplikaziorik: lerro bat sartu ondoren, begiratuta egiaztatuko dugu ondo idatzita dagoen. Horrela ez bada, Backspace izeneko tekla erabiliko dugu eginiko akatsak zuzentzeko eta, lerroa ondo idatzita dagoela egiaztatutakoan, Enter tekla sakatuko dugu. Horrela "2" agertuko da hurrengo lerroan, sententziaren edizioak aurrera jarraitzen duela adieraziaz.

Hiru era dauzkagu komandoaren amaiera adierazteko:

- ✓ Nahikoa da idatzita dugun azken lerroak ";" jartzea.
- ✓ Lerro bat zuri utzita, komandoa bukatuta dagoela baina une horretan egikaritzeko asmorik ez dugula adieraziko dugu.
- ✓ Idatzita dagoen azken lerroaren ondorengo lerroan "/" idaztea. Komandoa amaituta dagoela eta berehala egikaritzekoa dela adieraziko dugu horrela.

LANBIDE EKIMENA (104



5.4 SQL-ren buffer-a

Erabiltzaileak sartutako SQL sententzia SQL komandoen buferrean biltegiratuko da. Honakoak egin daitezke SQL sententziarekin.

- ✓ Berehala egikaritu (komandoari ";" erantsita).
- ✓ Egikaritu aurretik, editatu eta zerrendatu:
 - a) RUN erabilita, buferraren edukia ikusi eta egikaritu egingo dugu.
 - b) / erabiliz gero, berriz, komandoa ez da exekutatu aurretik agerian azalduko.

Bakoitzean SQL sententzia bati eutsiko dio bufferrak, eta beste SQL sententzia bat sartzen dugunean, gainean idatziko du.

SQL*Plus-en bidez unean uneko SQL sententzia edita dezakegu (bufferrean biltegiratuta), lerroko editore xume baten bidez.

KOMANDOA	<u>LABURDURA</u>	FUNTZIONALTASUNA
APPEND text	A text	Testua eransten du lerroaren amaieran
CHANGE old/new	C/old/new	"new" edukia "old" edukiaren ordez
CHANGE / text	C/text	Lerro bateko testua ezabatzen du
CLEAR BUFFER	CL BUFF	Lerro guztiak ezabatzen ditu
DEL	(bat ere ez)	Lerro bat ezabatzen du
INPUT	1	Lerro bat edo gehiago eransten du
INPUT text	I text	Testu-lerro bat eransten du
LIST	L	SQL bufferraren lerro guztien zerrenda
LIST n	L n edo n	Lerro baten zerrenda
LIST *	L*	Unean uneko lerroaren zerrenda
LIST LAST	L LAST	Azken lerroaren zerrenda
LIST m n	Lmn	Lerro-maila baten zerrenda ("m"-ik "n"-ra)

Jarduneko SQL instrukzioa erabilita, lan egiteko darabilgun makinaren sistema eragilearen editoretik sar gaitezke bitarteko memoriaren edukira (buffer). Ondoren, eduki hori aldarazi ahalko dugu, editoreak ematen dizkigun erraztasun guztiak erabilita.

SQL*Plus-ek AFIEDT.BUF izeneko artxiboa sorraraziko du lehenespenez, eta tarteko memoriaren eduki guztia biltegiratuko du horrek.

Artxiboaren aldakuntzarekin batera, bufferraren edukia ere aldaraziz doa eta, editorea uztean, SQL*Plus-en esku lagako da kontrola atzera ere.

5.5 Artxiboak erabiltzea

Hona hemen atal honetan aztertuko ditugun aukerak: spool-artxiboak eta komandoen artxiboak.

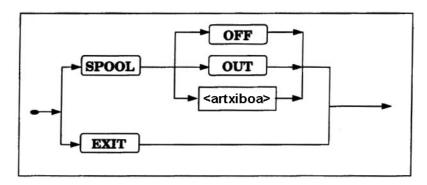
SPOOL artxiboak

SQL*Plus-ek sorrarazten dituen emaitzak etengabe atzeman ditzakegu SPOOL komandoa erabiliz gero. Horri esker, pantailan ageri den guztia kopia dezakegu sistema eragilearen artxibo batera eta, nahi izatera, inprimagailura bidaliko dugu.

SPOOL-aren ekintzak bere horretan jarraituko du SPOOL-komando berria erabili arte.

- ✓ SPOOL OFF SPOOL-ek sorrarazitako artxiboa itxiko du.
- ✓ SPOOL OUT SPOOL artxiboaren edukia sistemaren inprimagailura bidaliko du zuzenean.

SQL*Plus-en saio bati amaiera ematen zaionean, SPOOL jarraibidean sorturiko artxibo guztiak automatikoki itxiko dira



5.3. irudia.

Komandoen artxiboa

Orain arte, komandoak sistema eragilearen fitxategi batean biltegiratzea honako hauek biltegiratzera murriztu da:

- ✓ Unean uneko SQL komandoa.
- ✓ SQL komando bakarra.

Fitxategi batean hainbat komando biltegiratzea ere, beraietako batzuk SQL komandoak eta besteak, berriz, SQL*Plus komandoak izaki, baliagarri izango litzaiguke.

Komandoen artxiboak, hortaz, sistema eragilearen artxibo batean biltegiraturik dauden SQL*Plus komandoekin batera joan daitezkeen SQL-sententzien multzoa dira. Bata bestearen ondoren egikarituko dira, sekuentzia-eran, artxiboaren exekuzioa botatzean.

SQL*Plus-en bidez komando bat edo gehiago biltegiratu ahalko dugu fitxategi batean. Komandoen fitxategia sortu ondoren, berreskuratu, editatu eta egikaritu ahal izango da.

Bitarteko memoriaren edukia (zeina SQL jarraibidea den) salba daiteke SQL*Plus-i esker, baldin eta SAVE komandoa erabiltzen bada.

SAVE izena_artxiboa

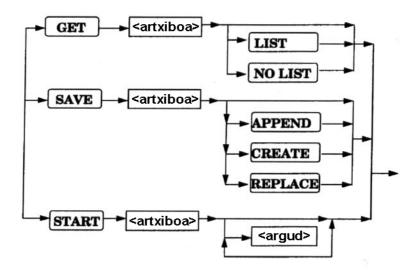
.SQL luzapena izango du lehenespenez sorrarazitako fitxategiak.

SQL*Plus-en bidez aurrerago salbaturiko fitxategiaren edukia berreskura daiteke baldin eta honako komandoa erabiltzen bada:

GET izena_artxiboa

Salbaturiko artxiboa berreskuratua eta berehala egikaritua izan daiteke, honako komando hau erabilita:

START izena_artxiboa



5.4. irudia.

Komandoen artxibo batera bil ditzakegun lerro-motak hauexek dira:

- ✓ SQL komandoen lerroak
- ✓ SQL*Plus komandoen lerroak
- ✓ Iruzkin-lerroak: SQL*Plus-en REM komandoa erabilita edo SQL-ren /*....*/ mugatzaileak erabilita.
- ✓ Egikaritze-lerroak: / barra makurra besterik ez daukate eta SQL sententzia bakoitzaren ondoren sartuko dira, exekuzioa adieraziz.

Adibidea:

```
REM
REM Ospitaleari dagokion Datu Basearen Taulak Sortzea
REM
CREATE TABLE
                   diagnostikoak
inskripzioa
             NUMBER(4)
                                NOT NULL PRIMARY KEY,
kodea
             NUMBER(2)
                                NOT NULL,
mota
             VARCHAR2(20)
                                NOT NULL,
konplikazioak VARCHAR2(20)
                                NOT NULL,
prek_info
             VARCHAR2(20)
                                NOT NULL
)
CREATE TABLE
                   laborategia
(
lab-zk
             NUMBER(4)
                                NOT NULL PRIMARY KEY,
izena
             VARCHAR2(20),
helbidea
             VARCHAR2(20),
telefonoa
             VARCHAR(8)
)
```

Komandoen beste prozedura bat abian jartzeko bide emango digun SQL*Plus jarraibidea ager daiteke komandoen fitxategi baten eduki gisa; *habiaratutako komandoen fitxategiez* ari gara horrelakoetan.

Komandoen fitxategia abian jartzeko, START sententzia erabiliko dugu (@-ren eragin berbera dauka). Sententzia horrek .SQL luzapena bereganatuko du lehenespenez.

Hona hemen komandoen artxiboaren beste osagai batzuk:

- Konstanteak
- Aldagaiak, honakoak izan daitezkeenak:
 - ✓ **Ordezkapenekoak**: aurrean "&" duten erabiltzaile-aldagaiak.
 - ✓ Erabiltzailearenak, komandoen fitxategietan behin eta berriro erabiltzeko definituak.
 - ✓ Sistemarenak, SQL*Plus-en lan-inguruari dagozkionak.

Ordezkapeneko aldagaiak

SQL*Plus-ek komando batean ordezkapeneko aldagaiari antzematen dionean, SQL*Plus-ek komandoa exekutatuko du, aldagaiaren balioa hartuta.

Aurrean "&" duten aldagai-izen baten bidez definituko ditugu ordezkapeneko aldagaiak.

Aldagaioi ezein baliorik eman ez, eta erreferentzia komandoen prozeduran egiten zaienean, mezua azalduko da pantailan beren balioa eskatuz:

> SQL> select lang_zk, deitura, soldata, komisioa 2 from lang 3 where lang_zk=&zenb Sartu balio bat zenb.rako = 7369 lehengoa 3: where lang_zk=7369 berria 3: where lang zk=7369

LANG_ZK	DEITURA	SOLDATA	KOMISIOA
7369	SANCHEZ	104000	
SQL>			

5.5. irudia.

Aurreko adibidean, alderatuko den datua zenbakia denez, &izena_ald kateaz egingo zaio erreferentzia aldagaiaren balioari. Datua karakterea edo alfanumerikoa denean, ordea, '&izena ald' erabilita egingo diogu erreferentzia aldagaiaren balioari.

SQL> select lang_zk, deitura, soldata, komisioa

2 from lang

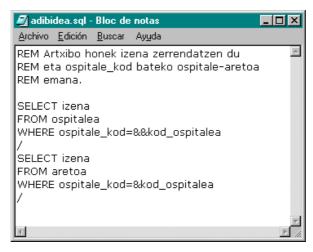
3 lang zk= '&DEIT'

Sartu balio bat deit-erako: SANCHEZ lehengoa 3: WHERE deitura = '&DEIT' berria 3: WHERE deitura = 'SANCHEZ'

LANG_ZK	DEITURA	SOLDATA	KOMISIOA
7369	SANCHEZ	104000	

5.6. irudia.

SQL*Plus-ek balioaren gaineko galdera aldagaia aurkitzen duen lehenbiziko aldian baino egin ez dezan nahi izanez gero,"&&" karakterea erabiliko dugu:



5.7. irudia.

Honako hau izango da emaitza:

SQL > start adibidea

Sartu balioa kod ospitalea: 13-rako

Lehengoa 3: WHERE ospitale_kod = &&kod_ospitalea

Berria 3: WHERE ospitale_kod 0 13

IZENA PROBINTZIAKOA

Karakter bati eteniko sarrera

DATU-BASEAK KUDEATZEKO SISTEMAK

Lehengoa 3: WHERE ospitale_kod = &kod_ospitalea Berria 3: WHERE ospitale kod = 13

IZENA ZAINKETA INTENTSIBOAK

PSIKIATRIKOA

5.8. irudia.

Erabiltzaile-aldagaiak

DEFINE komandoaren bidez definituko ditugu erabiltzaile-aldagaiak. Erabiltzaile-aldagaia sorrarazi eta CHAR balioa emango dio. Erabiltzaile-aldagai guztiak eta bere motak kontsultatzeko, sartu DEFINE.

> SQL> DEFINE langilea = SMITH SQL> **SQL> DEFINE** DEFINE_SQLPLUS_RELEASE= "800040000" (CHAR) DEFINE_EDITOR = "Notepad" (CHAR) DEFINE_O_VERSION = Oracle8 "Personal Edition Release 8.0.4.0.0. Production PL/SQL Release 8.0.4.0.0 – Production" (CHAR) DEFINE_O_RELEASE = "800040000" (CHAR) DEFINE LANGILEA= "SMITH" (CHAR) SQL>

Aldagai batek balio jakina baldin badauka esleiturik, esleipen horrek bere horretan iraungo du harik eta:

- ✓ Berriro definitu arte
- ✓ UNDEFINE izena_ald erabilita desaktibatu arte
- √ Komandoen artxiboen egikaritzea amaitu arte

Aldagaiak definitzeko baliagarria den beste metodo bat ACCEPT komandoa erabiltzearena da. Honakoa da bere sintaxia:

ACCEPT izena_ald [char | number] [Prompt 'mezua'] [Hide]

Adibidea:

SQL> ACCEPT wsoldata number PROMPT 'Sartu gutxieneko soldata:'

Sartu gutxieneko soldata: 200000

SQL> SELECT lang_zk, deitura, soldata

3: WHERE soldata < 200000 berria

LANG_ZK	DEITURA	SOLDATA
7369	SANCHEZ	104000
7521	SALA	162500
7654	MARTIN	182000
7844	TOVAR	195000
7876	ALONSO	143000
7900	JIMENO	123500
7934	MUÑOZ	169000

7 lerro hautaturik

Kasu batzuetan, HIDE hitza erabilita eman dakioke amaiera ACCEPT aginduari. Horrela, erabiltzaileak tekleaturiko sarrera ez da pantailan ikusgai agertuko. Kontrazeinuak eta datu gordeak sartzeko bidea ematen du.

Hona hemen aldagaien kasu berezia:

Parametroak

Komandoen artxibo bat abian jartzean, bederatzi parametroraino adieraz daitezke:

Kokalekuari dagokionez haiei dagozkien aldagaien bidez gagozkie parametroei komandoen artxiboan:

Ez da bederatzi parametro besterik onartuko; gutxiago badira, dagozkien balioek ez dute esleituriko baliorik. Nahi beste aldiz eta nahi den ordenan jardun daiteke parametroez komandoen artxibotik.

Adibidea:

SQL> INSERT INTO lang VALUES (&1, &2, &3, &4, &5, &6, &7, &8);

Sartu 1erako balioa: 7999

Sartu 2rako balioa: **UGARTE** Sartu 3rako balioa: LANGILEA

Sartu 4rako balioa: 7698

Sartu 5erako balioa: 81-UZT-21 Sartu 6rako balioa: 225000 Sartu 7rako balioa: **NULL** Sartu 8rako balioa: 10

Lehengo 1: INSERT INTO lang VALUES (&1, &2, &3, &4, &5, &6, &7, &8)

Berria 1: INSERT INTO lang VALUES (7999, 'UGARTE', 'LANGILEA', 7698, '81-UZT-21', 225000,

NULL, 10)

Lerro 1 sortu da

Ondoko lerroa daukan EJ1.SQL fitxategia sorraraz dezaket:

INSERT INTO lang VALUES (&1, &2, &3, &4, &5, &6, &7, &8);

eta gero komandoen fitxategi hau egikaritzea bi eratara agindu.

SQL> START EJ1

Sarrera etena karaktere 1era

Sartu 1erako balioa: 9000 Sartu 2rako balioa: **PEPE**

Sartu 3rako balioa: SALTZAILEA

Sartu 4rako balioa: 7902

Sartu 5erako balioa: 90-MAR-19

Sartu 6rako balioa: 25000 Sartu 7rako balioa: 15000 Sartu 8rako balioa: 20

Lehengo 1: INSERT INTO lang VALUES (&1, &2, &3, &4, &5, &6, &7, &8)

Berria 1: INSERT INTO lang VALUES (9000, 'PEPE', 'SALTZAILEA', 7902, '90-MAR-19', 25000, 15000, 20)

Lerro 1 sortu da

SQL> START EJ1 8888 ZABALETA SALTZAILEA 7902 87-URT-30 150000 0 30;

Sarrera etenda karaktere batera

Lehengoa 1: INSERT INTO lang VALUES (&1, &2, &3, &4, &5, &6, &7, &8)

1: INSERT INTO lang VALUES (8888, 'ZABALETA', 'SALTZAILEA', 7902, '87-URT-30', 150000, 0, 30)

Lerro 1 sortu da

Txostenak sortzea 56

SQL*Plus-i esker, kontsulta baten emaitzak formateatu ahal ditugu, amaituta dagoen txostena ekoizteko.

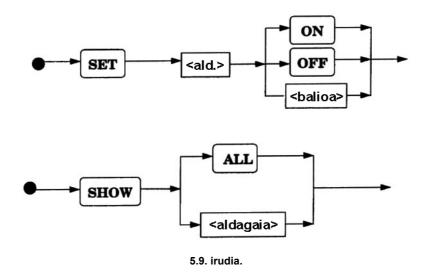
Honako hauek egin ditzakegu SQL*Plus erabilita:

- ✓ Zutabeen buruak aldatu.
- ✓ NUMBER, VARCHAR2, LONG eta DATE erako zutabeak formateatu.
- ✓ Zutabeak aurkezteko atributuak kopiatu eta zerrendatu.
- ✓ Orrialdeen neurriak definitu.
- ✓ Kalkuluak egin eta inprimatu (guztirakoak, batez bestekoak, gutxienekoak, gehienezkoak, eta abar).
- ✓ Orrialdeen buruetan eta behe-aldeetan tituluak kokatu.
- ✓ Izenburuetan zutabeen balioak eta egunari dagokion data edo orrialde-zenbakia aurkeztu.
- ✓ Txostena formateatzeko aukera ematen didaten SQL*Plus komandoak:
- ✓ Ez dira SQL-ren bufferrean gordetzen.
- ✓ Egikaritzeak ez du jarduneko SQL aldaraziko.
- ✓ Baliogabetu, aldatu edo SQL*Plus erabilita aritutako lan-jarduna utzi arte izango du eragina.

Inguru-parametroak

Aldagai global gisa biltegiratuko dira inguru-parametroak. Balio bat izango du lehenespenez parametro bakoitzak: SET komandoa erabilita, SQL*Plus-en zenbait aldagairen balioak defini daitezke. SQL*Plus-en aldagaien balioa ikusi nahi badugu, SHOW komandoa erabiliko dugu.

Honakoa da bi komandoetarako sintaxia:



Irteerako zenbait parametro:

SET LINESIZE	Lerroko gehienez	izan daitezkeen	karaktereen	kopurua.	Lehenespenez,
--------------	------------------	-----------------	-------------	----------	---------------

80koa da bere balioa. Zutabe bat inoiz ere ez da bi lerroren gainean

luzatuko. Gehienez, 999.

SET PAGESIZE Irteerako lerro-kopurua beste orri bati ekin aurretik. Lehenespenez 25ekoa

da bere balioa. Titulua eta geldialdi-lerroak barne daude.

SET HEADING [ON/OFF]

Zutabe-kateen erabilpena aktibatu eta desaktibatu egiten du. Lehenes-

penez, "ON" du balioa.

SET NULL "Testua"

Balio nuluak (NULL) ordezkatuko dituen karaktere-katea adierazten du.

"" da lehenespenez duen balioa.

SET ECHO [ON/OFF]

"ON" egoerari esker, SQL*Plus-ek komandoak ikusaraz ditzake komandoen

artxibo batetik exekutatuak diren heinean.

"OFF" egoerak, bere aldetik, beharrezkoa du ikustaraztea. "OFF" da

lehenespenez duen balioa.

SET FEEDBACK (nn/ON/OF)

Kontsulta bat egitean berreskuratzen den erregistro-kopurua erakusten

du, baldin edo "nn" erregistro edo gehiago berreskuratzen badira.

"ON" egoerak 1 balioa hartuko du "nn"-tzat.

"OFF" egoerak 0 balioa hartuko du "nn"-tzat.

SET VERIFY [ON/OFF]

Baieztapen-irteera kontrolatzen du ordezkapen-aldagaien balioei

dagokienez. "ON" da lehenespenez duen balioa.

SET HEADSEP SQL*Plus-i izenburu bat lerro bitan edo gehiagotan zati dezala esaten

dion karakterea identifikatzen du.

REMARK Iruzkinak eransteko erabiltzen da.

Formatu-komandoak

Toki-aldagai gisa biltegiratzen dira formatu-zehazpenak eta LOGIN.SQL izeneko artxiboan sar daitezke. Oinarrizko bost komando daude:

- ✓ TTITLE
- ✓ BTITLE
- ✓ COLUMN
- ✓ BREAK
- ✓ COMPUTE

Izenburuen definizioa eta orrialde-neurriak

Badago izenburuak orrialde guztien burualdean eta oinean kokatzeko, orrialdeko lerro-kopurua sartzeko eta lerro bakoitzaren zabalera zehazteko aukera.

Honako hauek dira tituluak definitzen dituzten komandoak:

- ✓ TTITLE
- ✓ BTITLE

Burualdearen eta orrialde-bukaeraren tituluak definitzen dituzte. Zehaztutako lerroaren zabaleraren arabera zentratuko dira tituluak. Ez dute komatxoen beharrik baldin eta karaktere-kateek tarte zuririk ez badute.

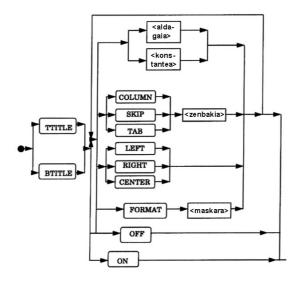
Besteak beste honako klausula hauek erabili ohi dituzte:

COL n hurrengo balioa n posizioan ager dadin adierazten du.

SKIP_n n lerroko jauzia egingo du.

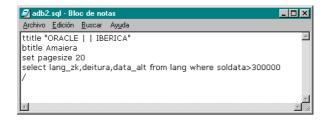
LEFT/CENTER/TIGHT lerrotzeko balio du.

Hona hemen sintaxi osoa:



5.10. irudia.

Adibidea:



SQL> start ej2

Sarrera etena karaktere 1era

Asteazk uzt 31 1. orrialdea

ORACLE

IBERICA

LANG_ZK	DEITURA	DATA_AL	
7566	JIMENEZ	81/04/02	
7698	NEGRO	81/05/01	
7782	CEREZO	81/06/09	
7788	GIL	87/03/30	
7839	REY	81/11/17	
7902	FERNANDEZ	81/12/03	

Amaiera

DEFINE komandoa erabilita, izenburu luzeak (>500 karaktere) sartzeko aukera dago.

Adibidea:

SQL> lerroa1 definitzen du = "Hau lehenengo lerroa da...."

SQL> lerroa2 definitzen du = "Hau bigarren lerroa da...."

SQL> lerroa3 definitzen du = "Hau hirugarren lerroa da...."

SQL> TTITLE CENTER lerroa1 SKIP 1 LEFT lerroa2 SKIP 1 RIGHT lerroa3;

Zutabeen komandoak

Honakoak ikusiko ditugu atal honetan:

- ✓ NUMBER, VARCHAR2, LONG eta DATE zutabeak nola formateatu
- ✓ Zutabeen burualdeak nola definitu
- ✓ Izenburuetan zutabeen balioak nola ikusarazi
- ✓ Sistemaren data izenburuan nola aurkeztu

NUMBER, VARCHAR2, LONG eta DATE zutabeak formateatzeko, COLUMN komandoa erabiliko dugu zutabeak banan-banan formateatzeko erabiltzen den FORMAT klausularekin konbinaturik.

Hauxe da sintaxia:

COLUMN izena zutabea FORMAT formatu maskara

Honako unitate hauek erabilita eraikiko dugu formatu-maskara:

ALFANUMERIKOA

An "n" digituko zabalera.

ZENBAKIZKOA

- Digitu baten posizioa zehaztuko du, betetze-zeroak ezabatuz
- 0 Digitu baten posizioa zehaztuko du, betetze-zeroak erabilirik
- \$ "\$"-ren posizioa
- "."-ren posizioa
- ","-ren posizioa
- V Zenbaki errealen hamarrenak lerrotuko ditu hamarrenen puntua erakutsi gabe
- B Zuriuneak zero bilakaraziko ditu zutabeen hasieran
- MI Minus zeinua ezarriko du zifra baten eskuinaldean, baldin eta negatiboa bada
- PR Txikiago eta handiago zeinuak (<>) ezarriko ditu, zifra bat, negatiboa izanez gero, sarturik

COLUMN komandoaren HEADING klausula erabiltzen badugu, badaukagu zutabeen burualdeak definitzeko aukera, honako formatu honekin:

COLUMN izena_zutabea HEADING burualdea_zutabea

Definitzen diren burualdeek bere horretan iraungo dute beste burualde batzuk sartu arte edo SQL*Plus-etik irteten garen arte.

Burualdeak lerro bat baino gehiago erabilita aurkeztu nahi baditugu, "|" barra erabili behar dugu.

Komatxo soilak erabili beharko ditugu hitz batek baino gehiagok osatzen duten burualdeetarako.

Zutabe bati baino gehiagori aurkezpen-atributu berberak esleitu behar zaizkionean, komandoen luzera sinplifika daiteke COLUMN komandoaren LIKE klausula erabilita:

Adibidea:

SQL> COLUMN komisioa LIKE soldata HEADING bonus

Zutabe baten balioak izenburuan ikusarazteko, nahi dugun balioa aldagai batean biltegiratu eta aldagai hori TTITLE komandoan erreferentziatu beharko dugu. Hori egiteko bidea ematen digun klausula NEW_VALUE dugu:

COLUMN izena_zutabea NEW_VALUE izena_aldagaia

LANBIDE EKIMENA (119



Izenburuetan sistemaren data aurkeztu nahi izanez gero, berriz, izenburuan data-balioa tekleatzea nahikoa da edo, bestela, data txostena exekutatzean automatikoki ager dadin nahi badugu, sistemaren data iraultzeko balio digun aldagaia sorraraziko dugu.

Adibidea:

SQL> BREAK ON gaur SQL> COLUMN gaur NEW_VALUE data SQL> SELECT TO_CHAR (SYSDATE, 'ddMonthYYYY') GAUR FROM DUAL:

GAUR _ _ _ _ _ _ _ _ _ _ _ _ _ _ 31 uztaila 2002

COLUMN-ekin batera joan daitezkeen beste klausula batzuk:

[WRAPPED/WORD-WRAPPED/TRUNCATED]

Zutabearen balioa berarentzat zehaztu den zabalera baino handiagoa bada irteeran zer egin behar den zehaztuko digu.

NEWLINE

Lerro berria azalaraziko du zehaztutako zutabea agertu aurretik.

NULL 'testua'

Zutabean null balio baten ordez jarri beharreko testua zehaztuko du.

[PRINT/NO PRINT]

Zutabea inprimatu behar denentz kontrolatuko du. Lehenespenezkoa da inprimaketa.

[ON/OFF]

Zutabearen formatu-etengailuak konektatu edo deskonektatuko ditu. OFF-ek lehenespenez eraberrituko ditu balioak. ON-ek, bere aldetik, zehazturiko parametroak eraberrituko ditu.

Kontrol-hausturen definizioa

Hala lerro bat berreskuratu bitartean, nola orrialde baten amaieran edo zutabe bateko balioak aldatzean jazotzen den gertaera dugu kontrolaren haustura (break). BREAK komandoak bikoizturik dauden balioak ezabatu egingo ditu adierazitako zutabean/zutabeetan edo adierazpenean/adierazpenetan.

Ez dugu ahaztu behar ORDER BY izeneko klausularekin batera erabiliko dela.

Honako hau da sintaxia:

BREAK ON zutabea_haustura1 ON zutabea_haustura2 ...

Honako kasu hauetan defini daiteke kontrol-haustura:

✓ Expr Expr-ren balioa aldatzen denean.

ROW Lerro-aldaketa guztietan.

PAGE Orrialde guztien amaieran.

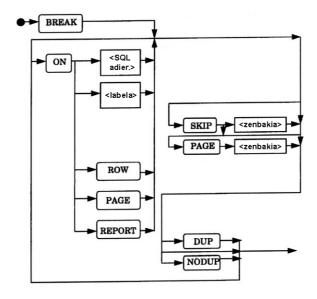
REPORT Txosten edo kontsulta baten amaieran.

Kontrol-haustura gertatzen denean, honako ekintza hauei ekin diezaiekegu:

SKIP n n lerroren jauzia egingo du.

SKIP PAGE Hurrengo orrialdearen burualdera egingo du jauzi.

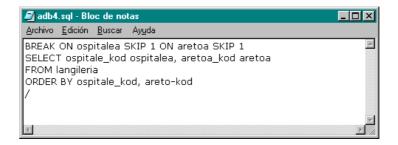
✓ DUP/NODUP Kontrol-hausturako zutabe gisa zehaztutako zutabeetan bikoizten diren balioak ez dira ikusgai azalduko (lehenespenez, NODUP)



5.11. irudia.

Une jakin batean BREAK komando bakarrak jardun dezake eragile eta, horregatik, hainbat zutabe zehaztu beharko dira BREAK komando bakarrean batera eragin dezan nahi izanez gero.

Adibidea:



SQL> start ej4 Sarrera etenda karaktere 1era

ARETOA
6
4
1
2
6
1
4

10 lerro hautatu dira.

Zerrendetan haustura-puntuak gauzatzen direnean, interesgarria izan daiteke COMPUTE komandoa erabilita erregistroen gainean kalkuluak egitea. Honako hau izango da orduan sintaxia:

BREAK ON zut_haustura COMPUTE funtzioa OF zutabea ON zut_haustura

Honako funtzio hauek buru ditzakegu:

AVG Zutabeko balioen batezbestekoa kalkulatuko du.

COUNT Zutabean dauden balio ez-nuluak kontabilizatuko ditu.

MAX Zutabean dagoen gehienezko balioa kalkulatuko du.

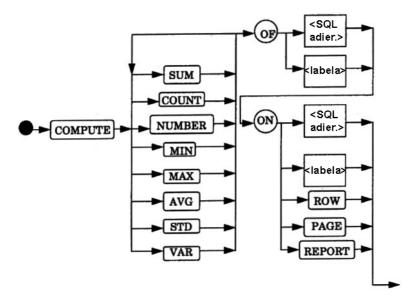
MIN Zutabean dagoen gutxienezko balioa kalkulatuko du.

NUM Zutabean dagoen erregistro-kopurua kalkulatuko du.

STD Zutabearen balioetan ematen den desbideraketa estandarra kalkulatuko du.

SUM Zutabeko balioen batuketa kalkulatuko du.

VAR Zutabeko balioen aldagarritasuna kalkulatuko du.



5.12. irudia.

Adibidea:



SQL> start ej5 Sarrera etenda karaktere 1era

OSPITALE-KOD	DEITURA	SOLDATA
40		
13	DIAZ B.	226200
	HERNANDEZ J.	275000

avg		250600
sum		501200
18	KARPLUS, W.	337900

avg		337900
sum		337900
uy		

CLEAR komandoa

CLEAR komandoaren bidez, honako aukera hauek garbi ditzakegu:

✓	BREACKS	BREAK komandoaren bidez markaturiko haustura-puntuak garbituko ditu.
✓	BUFFER	Buffer aktiboaren testua garbituko du.
✓	COLUMNS	COLUMN komandoaren aukerak garbituko ditu.
✓	COMPUTES	COMPUTE komandoaren aukerak garbituko ditu.
✓	SCREEN	Pantaila garbituko du.
✓	SQL	SQL-ren bufferra garbituko du.