

1. MODULUA: INFORMAZIOAREN ANTOLAKETA ETA EGITURA

3.gaila: Datu baseen Galdeketa eta Ustiapena

Leire Aldaz, Begoña Eguía eta Leire Urcola



1. MODULUA: Informazioaren Antolaketa eta Egitura
3 Gaia: Datuen Galdeketa eta Ustiapena.

SQL (Structured Query Language) Datu Base erlazionalak galdetzeko lengoia

SQL lengoaiaren sintaxia orokorra:

```
SELECT < atributu zerrenda > | *  
FROM < taula-k >  
[WHERE < baldintza-k > ]  
[GROUP BY < atributua-k > ]  
[HAVING < baldintza-k > ]  
[ORDER BY < atributua-k> ASC | DESC] ;
```

□ **PREDIKATUAK**

Predikatuak WHERE klausularen baldintzarengan ezar daitezke.

BETWEEN ... AND ..., where atributua Between and

IN (balio_zerrenda), where atributua IN(bal1, bal2, bal3,...)

LIKE where atributua LIKE karaktere_segida

* (izartxoak), edozein karaktere kopuru ordezkatzan du

? galdera ikurrak berriz, edozein karaktere bakarra ordezkatzan du.

□ **FUNTZIOAK**

- **Datak maneiatzeko funtzioak:** Date(), Year(data), Month(data), Day(data)

- **Funtzio erantsiak:** Count(exp.), Sum(exp.), Max(exp.), Min(exp.), Avg(exp.)

- **Silnm funtzioa:** Silnm(baldintza; Egia_baldinEkintza; [Faltsu_baldinEkintza])



1. MODULUA: Informazioaren Antolaketa eta Egitura

3 Gaia: Datuen Galdeketa eta Ustiapena.

Sarrera.

SQL “Structured Query Language” lengoaiaren bitartez datu base bat sortu eta manipula dezakegu kontsulta edota eguneraketa eragiketen bitartez. Baina eskuliburu txiki honetan SQL ren erabilpena kontsultak egiteko bakarrik azalduko da. Kontsulta guztiak SELECT aginduarekin inplementatzen dira. Horrela, SELECT agindua datu base baten taula batetako edo anitzeko erregistroak hautatzeko erabiltzen da, hautaketa hauek irizpide batzuren arabera egin daitezkeelarik.

SELECT aginduaren sintaxia orokorra ondorengoa da:

```
SELECT < atributu zerrenda > | *  
    FROM < taula-k >  
    [WHERE < baldintza > ]  
    [GROUP BY < atributua-k > ]  
    [HAVING < baldintza > ]  
    [ORDER BY < atributua-k> ASC | DESC] ;
```

Kortxeteren barruan zehaztutako klausula guztiak aukerazkoak dira. Beraz, nahita nahiezkoa den bakarra FROM klausula izango da, hots, nundik hartuko diren atributuak adierazteko erabiltzen dena.

<atributu zerrendan> pantailan ikusi nahi ditugun atributuen balioak edo expresioak ere adieraziko ditugu.

Adibidez:

Bezeroen taulako datu guztiak pantailaratzeko ondorengo moduan formulatuko genuke galdera:

```
SELECT bezeroid, izena, helbidea, herria, telf  
    FROM bezeroak ;
```

Kontsulta honen emaitzan bezero taulan dauden bezero guztien (erregistro guztien) datuak pantailan izango ditugu.



1. MODULUA: Informazioaren Antolaketa eta Egitura
3 Gaia: Datuen Galdeketa eta Ustiapena.

Taula batetako atributu guztien balioak pantailaratu nahi baldin baditugu, denak banan bana idatzi beharrean, atributu zerrenda zehazterakoan izartxo (*) erabil dezakegu.

Beraz, aurreko kontsulta era honetan ere formula daiteke:

```
SELECT *  
FROM bezeroak ;
```

Aurreko kontsultaren bidez bezero guztiak aurkezten dira pantailan, euren ezaugarri guzti edo parte batekin. Baina erregistro batzuk bakarrik aurkeztu nahi izango bagenitu orduan WHERE klausula erabili beharko genuke. WHERE klausularen aldamenen ipintzen ditugun baldintzak ximpleak edo konplexuak izan daitezke. Lehenengoak, bi terminoren arteko konparaketak egiteko, eragile erlazioanak erabiliz. Eta besteak, berriz, hainbat baldintza ximple konkatatzeko eragile logikoen bitartez.

```
SELECT *  
FROM bezeroak  
WHERE herria = "Madril" ;
```

Kasu honetan, baldintza ximple hau betetzen duten erregistroak hautatuak izango liratezke bakarrik.

ADIBIDEAK :

1. 1998-ko urtarrilaren lehenengotik aurrera erregistratutako eskaera guztiak pantailaratu.

```
SELECT *  
FROM bezero_eskaerak  
WHERE eskadata > #1/1/98# ;
```

2. 150 pta. baino goragoko unitate prezioa eta 100 baino txikiagoko Stock kantitatea duten artikulu guztiak zerrendatu.

```
SELECT *  
FROM artikuluak  
WHERE up > 150 AND sotck < 100 ;
```



1. MODULUA: Informazioaren Antolaketa eta Egitura
3 Gaia: Datuen Galdeketa eta Ustiapena.

3. Langileen taulatik ondorengo baldintzak betetzen dituzten langileak soilik, pantailan aurkeztu.

- a) Emakume direnak.
- b) Euren soldata 210000 eta 250000 milaren bitartekoa (biak barne).
- c) Eta kategoria aldetik, informatikalari eta ingenerioak direnak.

```
SELECT *  
FROM langileak  
WHERE sexua AND soladata >= 210000 AND soldata <= 2500000 AND  
      ( kategoria = "ingeniero" OR kategoria = "informatiko" );
```

BETWEEN ... AND ... predikatua

Predikatu honek tarte itxi baten barruan dauden balioak aukeratuak izateko balio du. Horrela, aurreko kontsultak ondorengo eran ere formula daitezke:

```
SELECT *  
FROM artikulu  
WHERE up BETWEEN 150 and 200 ;
```

Edo beste hau:

1998-ko lehenengo lau hilabeteren barne erregistratu ez diren eskaerak pantailaratu (hau da, lau hilabete hoietatik kanpo erregistratu egin direnak soilik).

```
SELECT *  
FROM eskaerak  
WHERE eskadata NOT between #1/1/98# AND #30/4/98# ;  
Edo : WHERE eskadata > #30/4/98#
```



1. MODULUA: Informazioaren Antolaketa eta Egitura
3 Gaia: Datuen Galdeketa eta Ustiapena.

Oharra: QBE kuadrrikulan, erizpide lerroan Negado Entre #1/1/98# y #30/4/98# jarriko genuke.

IN(balio zerrenda) predikatua.

In terminoaren ezkerrean ipintzen den expresioa balio zerrendan ipinitako balioekin konparatzen da, eta zerrendako balio baten berdina izatekotan erregistroa hautatua izango da.

Adibidez: Madril, Zaragoza eta Bartzelonako bezeroen izena eta helbideak soilik pantailaratu.

```
SELECT izena, helbidea
FROM bezeroak
WHERE herria In( "Madril", "Zaragoza", "Bartzelona" );
Edo WHERE herria="Madril" OR herria="Zaragoza" OR herria="Bartzelona" ;
```

LIKE <<karaktere-katea>> predikatua

Karaktere jakin batzutat hasten diren datuak pantailaratzeko erabiltzen da. Horretarako * eta ? sinboloak erabil daitezke.

*, izartxoak karaktere segida bat ordezkatzeko du eta galdera ikurrak, berriz, karaktere bakar bat.

Adibideak:

1. 34 zenbakirekin hasten diren bezeroen nortasun agiriak pantailaratu.

```
SELECT *
FROM bezeroak
WHERE Nan LIKE "34 *";
```

2. Bigarren abizena Perez duten bezeroak pantailaratu.

```
SELECT *
```



1. MODULUA: Informazioaren Antolaketa eta Egitura
3 Gaia: Datuen Galdeketa eta Ustiapena.

FROM bezeroak

WHERE izena LIKE "*Perez";

3. A eta M letren artean dauden bezeroen izenak pantailaratu.

SELECT izena

FROM bezeroak

WHERE izena LIKE "[A-M]*" ;

FUNTZIOAK

Funtzioak makro-eragile gisa agertzen diren azpiprograma txikiak dira. Datu base kudea sistemak (DBKS) deitu ditzazkeenak eta behin egikaritu ondoren balio bat itzultzen dutenak. Funtzioak bi elementuz osatzen dira:

- Funtzio izena
- Argumentua-k, hauek parentesi barruan doazte. Baina, badaude funtzio batzuk argumenturik ez daramatzatenak, hala ere, parentesiak ipini behar dira.

Accessen funtzio aukera handia dago (Acceseko Laguntza/Funtzioak ikustea komenigarria da), bere artean erabiliko ditugunak ondorengoak dira:

1. Datatzko funtzioak.

Datak eta orduak diren datuengain eragiten dute. Erabilienak ondorengoak dira:

Ahora(), argumenturik ez du eramaten eta gaurko data eta ordua itzultzen ditu.

Fecha(), gaurko data itzultzen du.

Año(arg), datatzko argumentuaren urtea, zenbaki osoan itzultzen du.

Mes(arg), datatzko argumentuaren hilabetea, zenbaki osoan itzultzen du.

Dia(arg), datatzko argumentuaren eguna, zenbaki osoan itzultzen du.

2. Funtzio erantsiak.

Funtzio hauek taula bateko erregistroengain eragiten dute, atributu bat edo gehiagotan, edo eta atributu horietan oinarritutako kalkulu adierazpenengain. Beraien formatua



1. MODULUA: Informazioaren Antolaketa eta Egitura
3 Gaia: Datuen Galdeketa eta Ustiapena.

aurrez adierazitakoaren ezberdina da ez baitute parentesi artean edukirik eramaten, baizik eta QBE kontsulta batean edo SQL sententziaren segmentu batean elementu aktibogisa adierazten dira. Garrantzitsuenak ondorengoak dira:

Suma, atributu baten eremuen edo atributu horrekiko ekspresio baten batuketa.

Promedio, atributu baten eremuen edo atributu horrekiko ekspresio baten batzbesteko aritmetikoaren kalkulua.

Min, atributu baten eremuen edo atributu horrekiko ekspresioaren balio minimoa.

Max, atributu baten eremuen edo atributu horrekiko ekspresioaren balio maximoa.

Cuenta, erregistroak kontatzeko erabiltzen da.

Oharra: batuketa, batzbestekoak, kontaketak, etab. egiteko beharrezkoa da Totalak izeneko lerroa kuadrikulara eranstea. Lerro hau, Acceseko lantresna barran Σ ikonoa pikatuz erantsi daiteke.

Adibideak :

1. Bezeroen_Eskaerak taulatik 30 egun baino gehiagoko antzinakotasuna duten eskaerak pantailaratu.

```
SELECT *  
FROM Bezeroen_Eskaerak  
WHERE eskadata < Fecha( ) - 30 ;
```

Edo: WHERE Fecha() - eskadata > 30 ;

2. Suposa dezagun langileen taulan beste atributuren artean, langile bakoitzaren kontratazio data (dd/mm/aa) formatuan gordetzen dela.

Kontsulta batetan kontratazio dataren ordez, langile bakoitzak duen antzinakotasuna urtetan aurkeztu nahi da.

Antzinakotasunaren balioa ondorengo ekspresiotik ondoriozta daiteke:

$\text{Año}(\text{Fecha}(\)) - \text{Año}(\text{kontradata})$

Beraz,

```
SELECT izena, Year(Date( )) - Year( kontradata) AS antzinakotasuna
```




1. MODULUA: Informazioaren Antolaketa eta Egitura
3 Gaia: Datuen Galdeketa eta Ustiapena.

FROM langileak ;

3. Edo hiru urte baino gehiagoko antzinakotasuna duten langileak aurkeztu nahi izango bagenu, kontsulta ondorengo eran formulatu beharko litzateke:

```
SELECT izena, Year(Date( ))-Year( kontradata) AS antzinakotasuna
FROM langileak
WHERE Year(Date( ))-Year( kontradata) > 3 ;
```

4. 1997-ko martxoa eta irailaren bitartean erregistratutako eskaera guztiak pantailaratu. (kasu honetan, konparaketak egiteko data zehatzak maneia daitezke baina ikusitako funtzioak erabiliko ditugu).

```
SELECT *
FROM bezeroak
WHERE Month(eskadata) >= 3 AND Month(eskadata) <= 9
AND Year(eskadata) = 1997 ;
```

Oharra: aurreko kontsuta Between AND predikatuarekin ere formula daiteke.

5. Langileen taulatik, soldata maximoa, minimoa, soldaten batura, soldaten batzbestekoa eta taula horretan gordeta dauden langile kopurua pantailaratu.

```
SELECT Max(soldata) AS maximoa, Min(soldata) AS minimoa,
Avg(soldata) AS batzbestekoa, Count(nan) AS langile_kopurua
FROM langileak ;
```

QBE :

Atributuak:	Maximoa: Soldata	Minimoa: Soldata	Batura:Soldat a	Media: Soldata	Langile kopurua: NAN
Taulak:	Langileak	Langileak	Langileak	Langileak	Langileak
Totalak:	Max	Min	Suma	Promedio	Cuenta
Orden:					
Erakutsi:	V	V		V	V
Erizpideak:					
Edo:					



1. MODULUA: Informazioaren Antolaketa eta Egitura
3 Gaia: Datuen Galdeketa eta Ustiapena.

6. Artikuluaren taulatik, stockaren zenbateko totala pantailan aurkeztu.

```
SELECT Sum([up] *[stock]) AS Stock_balioa
FROM artikuluak;
```

7. Langileen taulatik zenbatek 220000 pta. baino gehiago irabazten dute?

```
SELECT Count(nan) AS langile_kopurua
FROM langileak
WHERE soldata > 220000;
```

QBEn:

Atributuak:	Langile_kopurua:Nan	Soladata
Taulak:	Langileak	Langileak
Totalak:	Cuenta	
Ordena:		
Erakutsi:	V	
Erizpideak:		>220000
Edo:		

ARIKETAK:

1. Langileen taulatik, enpresak 60 urtetik gorako langileak jubilatzea nahi ditu. Langile hauei enpresan egin duten urte bakoitzagatik hileko soldata oso bat emango zaie. Langile hauen izenak eta dagozkien ordain apartak pantailaratu.



1. MODULUA: Informazioaren Antolaketa eta Egitura
3 Gaia: Datuen Galdeketa eta Ustiapena.

2. Enpresak aurtan zenbat langile kontratatu izan ditu?
3. Langileen soldata maximoa eta minimoaren arteko diferentzia pantailan aurkeztu.

Silnm(baldintza, Egiazko_partea, Gezurrezko_partea) funtzioa

Silnm funtzioan adierazitako baldintza betetzen den kasuan, egiazko partean zehaztutako ekintza egikarituko litzateke, baldintza betetzen ez bada berriz, gezurrezko partean zehaztutakoa exekutuko litzateke.

Adibidez:

Langileen taulatik, bi semealabatik gora dituzten langileen soldatak, semealaba bakoitzeko, 5000 ptako. gehikuntzarekin aurkeztu , eta gainontzekoak, dagozkien soldatak gehikuntzerik gabe.

Atributos:	Izena	PAGA: Silnm(semealabak >2, soldata + 5000 * (semealabak - 2), soldata)
Tablas:	Langileak	Langileak
Orden:		
Mostrar	V	V
Criteria:		
O:		

```
SELECT Izena, IIF( semealabak>2, soldata+5000*(semealabak - 2), soldata)  
FROM Langileak ;
```

GROUP BY klausula

Group by klausula, datuak taldekatzeko erabiltzen da. Klausula honen aldamenean, datuak zein atributuren arabera taldekatuko diren adierazi behar da. (Logikoki taldekatetako, beti, errepikatzen diren balioekin egiten dira).

Adibidez,

1. Bezeroen taulatik, herri bakoitzeko bezeroen kopurua pantailaratu.



1. MODULUA: Informazioaren Antolaketa eta Egitura
3 Gaia: Datuen Galdeketa eta Ustiapena.

```
SELECT herria, Count(bezeroid) AS bezero_kopurua  
FROM bezeroak  
GROUP BY herria ;
```

QBEn:

Atributuak:	Herria	bezero_kop: bezeroid
Taulak:	Bezeroak	Bezeroak
Totalak	Agrupar por	Cuenta
Ordena:		
Erakutsi:	V	V
Erizpideak:		
Edo:		

2. Eskaera bakoitzean eskatutako artikulu kopurua pantailaratu (hau da, eskaera bakoitzak dituen artikulu lerroak, ez eskatutako kantitateak).

Oharra: eskatutako_artikuluen taulan, eskazbkia eskaera jakin batetan eskatzen diren artikuluaren arabera adina bider errepikatuta azalduko da.

```
SELECT eskazbkia, Count(artikuluid) AS artikulu_kopurua  
FROM eskatutako_artikuluak  
GROUP BY eskazbkia ;
```

QBEn:

Atributuak:	Eskazbkia	Artikulu_kop: artikuluid
Taulak:	Eskatutako_artikuluak	Eskatutako_artikuluak
Totalak	Agrupar por	Cuenta
Ordena:		
Erakutsi:	V	V
Erizpideak:		
Edo:		



1. MODULUA: Informazioaren Antolaketa eta Egitura
3 Gaia: Datuen Galdeketa eta Ustiapena.

2. Zenbat eskaera egiten ditu bezero bakoitzak ?

```
SELECT bezeroid, Count(eskazbkia) AS eskaera_kopurua
FROM bezeroen_eskaerak
GROUP BY bezeroid ;
```

Oharra: bezeroen_eskaerak taulan, bezeroaren identifikadorea, bezeroak egiten dituen eskaera desberdinen arabera adina bider errepikatuta azalduko da.

QBEn:

Atributuak:	bezeroid	eskaera_kop: eskazbkia
Taulak:	Bezeroak	Bezeroak
Totalak	Agrupar por	Cuenta
Ordena:		
Erakutsi:	V	V
Erizpideak:		
Edo:		

3. Langileen taulatik, sail bakoitzaren barruan lan egiten duten langile kopurua eta euren soldataren batzbestekoa pantailaratu.

```
SELECT saila, Count(Nan) AS langile_kopurua, Avg(soldata) AS batzbestekoa
FROM langileak
GROUP BY saila ;
```

Oharra: enpresako sail bakoitzaren barruan, hainbat langilek lan egiten dute.

QBEn:

Atributuak:	Saila	Langile_kop: Nan	Batzbestekoa: Soldata
Taulak:	Langileak	Langileak	Langileak
Totalak	Agrupar Por	Cuenta	Promedio
Ordena:			
Erakutsi:	V	V	V
Erizpideak:			
Edo:			



1. MODULUA: Informazioaren Antolaketa eta Egitura
3 Gaia: Datuen Galdeketa eta Ustiapena.

HAVING klausula.

Klausula honek bi erabilpen ditu:

- a) Aldez aurretik egindako taldekaketarekin baldintzaren bat osatu behar denean.
- b) Baldintza bat zehazterakoan funtzio erantsiren bat erabili behar baldin bada, nahiz eta taldekaketarekin zerikusirik ez izan. (WHERE klausulak ez ditu funtzio erantsirik onartzen).

Adibidez:

Hiru eskaera baino gehiago dituzten bezeroak pantailaratu.

```
SELECT bezeroid, Count(eskazbkia) AS eskaera_kopurua
FROM bezeroen_eskaerak
GROUP BY bezeroid
HAVING Count(eskazbkia) > 3 ;
```

ORDER BY klausula

Informazio guztia, atributu baten edo hainbat atributuren arabera sailkatuta (ordenatuta) aurkezteko erabiltzen da. Sailkapena gorakorra edo beherakorra izan daitekeelarik. (Gorakorra: balio txikienetik handienara. Letrak: A-tik – Z-ra. Eta Numeroak: 0-tik – 9-ra.

Beherakorra: balio handienetik txikienera. Letrak: Z-tik – A-ra. Numeroak: 9- tik – 0-ra).

```
ORDER BY atributua [ASC | DESC] ;
```

Atributuaren atzean sailkapen mota adierazten ez bada gorakorra dela esan nahiko luke, eta DESC ipintzen baldin bada, berriz, orden beherakorra adieraziko luke.

Adibidez:

Bezeroen eskaerak taulatik, eskaera guztiak eskaera dataren arabera orden beherakorrean eta bezeroaren identifikazioaren arabera orden gorakorrean sailkatuta aurkeztu. (Kontsultaren emaitzan, lehenengo lerroetan data berrienak, hots, handienak



1. MODULUA: Informazioaren Antolaketa eta Egitura
3 Gaia: Datuen Galdeketa eta Ustiapena.

azalduko liratezke eta data berdinaren barruan sailkapena bezeroaren identifikazioaren arabera egingo litzateke).

```
SELECT eskadata, bezeroid
      FROM bezeroen_eskaerak
      ORDER BY eskadata DESC, bezeroid ;
```

Langileen taulatik, enpresak urte bakoitzean kontratatutako langile kopurua pantailaratu, baina bakarrik 1992. urtetik ondoren kontratatu izan direnak aurkeztu, eta informazio hau kontratatutako urtearen arabera orden beharokorrean sailkatu.

```
SELECT Year(kontradata) AS Urtea, Count(Nan) AS kontratu_kopurua
      FROM langileak
      GROUP BY Year(kontradata)
      HAVING Year(kontradata) >1992
      ORDER BY Year(kontradata) DESC ;
```

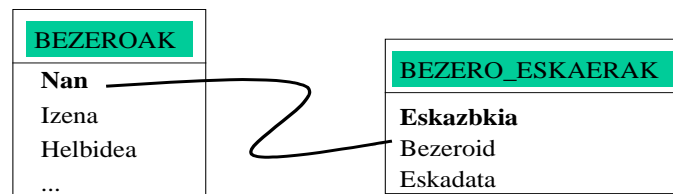
Atributuak:	Urtea: Año(kontradata)	Kontratu Kopurua: NAN
Taulak:	Langileak	Langileak
Totalak:	AgruparPor	Cuenta
Ordena:	Descendente	
Erakutsi:	V	V
Erizpideak:	>= 1992	
Edo:		



1. MODULUA: Informazioaren Antolaketa eta Egitura
3 Gaia: Datuen Galdeketa eta Ustiapena.

HAINBAT TAULEI KONTSULTAK

Datu base kudea sistema batean (DBKS), Accessen kasua den bezala, taula edo kontsulta bat baino gehiagotik datozen datuak batera erabil daitezke. Integritate erlazionala ezartzen bada, elkarloturiko tauletan nahi diren erregistro aldaketa edo ezabaketak integritate arauari lotuta egongo dira. Adibidez, taula nagusiko erregistrorik ezingo da ezabatu taula subordinatuan berari erlazionaturiko erregistroak existitzen badira; aurrez taula subordinatuan daudenak ezabatu baharko dira edo bestela ezabaketa kateatua egingo degu, hau da, taula nagusitik nahi diren erregistro ezabatu eta automatikoki binkulatutako taulatik elkarloturiko erregistro guztiak ezabatuko dira.



Accessen, taulen arteko loturak bi motatakoak izan daitezke:

1. Batetik baterakoa (1:1), taula nagusiko erregistro bakoitzeko binkulatutako taulan erregistro bakarra existitzen denean (hau da, taula nagusiko erregistro bakoitzari binkulatutako taulan erregistro bakar bat dagokionean).
2. Batetik askotarakoa (1:N), erlazio mota arruntena dugu. Taula nagusiko erregistro bakoitzari binkulatutako taulan hainbat erregistro dagozkionean.



1. MODULUA: Informazioaren Antolaketa eta Egitura
3 Gaia: Datuen Galdeketa eta Ustiapena.

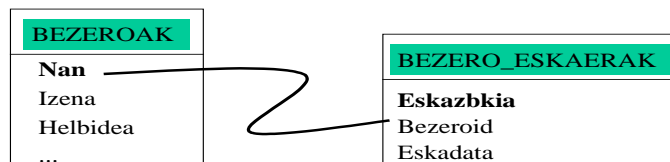
Adibideak :

1. Gutxienez eskaera bat egin duten bezeroen datuak ezagutzeko WHERE klausulan ondorengo hau ipiniko genuke: Bezeroak.bezeroid = Bezero_eskaerak.bezeroid

Bi tauletan bezero identifikazio berdina azaltzen duten erregistroak soilik, hautatuak izango dira.

```
SELECT izena, helbidea, tfnoa, eskadata
FROM bezeroak, bezero_eskaerak
WHERE bezeroak.bezeroid = bezero_eskaerak.bezeroid
```

QBE:



Atributuak:	Izena	Helbidea	Tlfnoa	eskadata
Taulak:	Bezeroak	Bezeroak	Bezeroak	Bezero_eskaerak
Ordena:				
Erakutsi:	V	V	V	V
Erizpideak:				
Edo:				

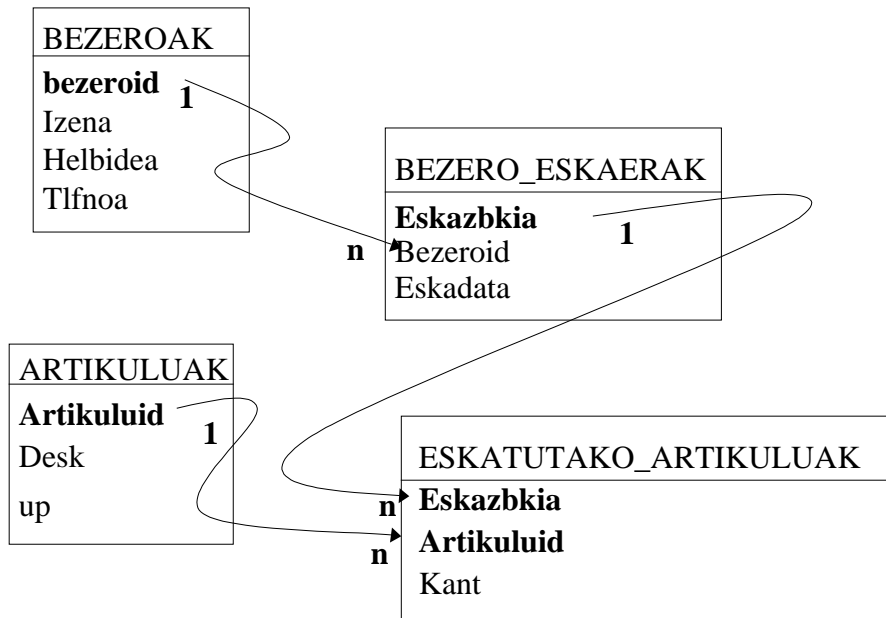


1. MODULUA: Informazioaren Antolaketa eta Egitura
3 Gaia: Datuen Galdeketa eta Ustiapena.

3. Eskara bakoitzaren zenbateko totala zerrendatu.

```
SELECT Eskazbkia, Sum([up]*[kant]) AS zebateko_totala
FROM Eskatutako_artikuluak, Artikuluak
WHERE Eskatutako_artikuluak.artikuluid = Artikuluak.artikuluid
GROUP BY Eskazbkia ;
```

QBE :



Atributuak:	Eskazbkia	Expr: [up]*[kant]	
Taulak:	Bezeroak	Bezeroak	
Totalak	Agrupar por	Suma	
Ordena:			
Erakutsi:	V	V	
Erizpideak:			
Edo:			



*1. MODULUA: Informazioaren Antolaketa eta Egitura
3 Gaia: Datuen Galdeketa eta Ustiapena.*

4. Bezero bakoitzaren datu orokorraren aldamenean, bakoitzak egindako eskaeren kopurua pantailaratu.

```
SELECT Bezero_eskaerak.bezeroid, izena, helbidea, tfnoa,  
       Count(Eskazbkia) AS eskaera_kopurua  
FROM Bezeroak, Bezero_eskaerak  
WHERE Bezeroak.bezeroid = Bezero_eskaerak.bezeroid  
GROUP BY Bezero_eskaerak.bezeroid ;
```

Oharra: bezeroid bi tauletan azaltzen danez, atributua hau zehazterakoan zein taulatik hartuko den adierazi behar izango da.