

DATU-BASEAK: Laborategia4 eta Laborategia5

Aplikazio transakzionalak¹

HELBURUA

- SQL sententziak programazio-lengoaia batean (kasu honetan JAVAN) nola barnera daitzekeen ikustea, eta aplikazio transakzionalen oinarritzko kontzeptuak lantzea.

DATU-BASEAK

u017943.g:ehues zerbitzarien Oracle DBKS datu-base sistema ditugu, bat definitu dizkigute: DBGE01, DBGE02..., bigarren gain definitutako erabiltzaileak: DBGE01, DBGE02... berriro ere. Erabiltzaile-kontu horiek taula hauekin nagusitak dira:

ERREALA
BEZEROA (na, izena, helbidea, herria, beherapena, eskaeraKop PRODUKTUA (kodea, izena, prezioa, kopurua)
ESKAERA (na, kodea, data, kantitatea)

ATHLETIC

ESKAERAOSOA (bzemb, pzenb, data, kantitatea, bizena, pizena)
--

Gainera, ERREALAren datu-basean BEZEROA taulak badu integritate-murrizpen bat definituta: *EskeraKop/Mga* deitzen dena eta *eskeraKop* attributuaren balioa 5 baino handiagoa ezin dela izan kontrolatzeko duena.

JDeveloper INGURUNEA

JDeveloper ingurunea erabiliz posible da bai datu-baseko taulen edukia interaktiboki azitzeara (SQLWorkSheet) bai Java (gure kasuan SQLJ) fitxategiak editatzeara, kompliatzea eta exekutatzeara. Bilatu *JDeveloper Studio 11.1.2.3.0* softwarea eta marktan jarri (*Studio Developer All Features*).

SQLWorkSheet

Taulen edukia zuzenean eta interaktiboki aztertu nahi badugu, Oraclek erabiltzeko aukera badago ere, gure laborategia garazeko JDeveloper inguruneak integratuta eskaintzen duenaz balituko gara.

SQLWorkSheet izeneko tresna eskaintzen du. Tresna hau, era independentean erabiltzeko aukera badago ere, gure laborategia garazeko JDeveloper inguruneak hasieratutako (Default Role) eta <Tools>+<Database>/+<SQLWorkSheet>+>+> aukeratutako dugu. Konexioa sortzeko (ikus irudia) kontzinatori eman nahi diogun izena, erabiltzailea (DBGE01, DBGE02...), pasahitza eta datu-basea adieraziko ditugu (errealaren kasuan, *jdbc:oracle:thin:@u017943.gi.ehu.es:1521:erreala*). <Test>

Connection> sakatuz, konexioa gauzatzeko gai den ala ez baiezatuuko dugu. Arrakasta izan badugu (Success!) <OK> sakatu eta konexioa sortuta geratuko da.

Dagoeneko dena prest dago datu-basea SQL lengoaia erabiliz atzizteko. Hemendik aurra, *erreala* datu-bassarekin konektatzeko, sortu berri den konexioa hautatu beharko da: *Tools>+<Database>/+<SQLWorksheet>+Gure_konexio_aukera*. SQL sententziak exekutatzeko WorkSheet-eko triangelu berdea sakatuko dugu (ikus trudia).

ID	NOMBRE	PAÍS	CIUDAD	ESTADO/C.
1	111111111111111111	PERÚ	LA PAZ	LA PAZ
2	222222222222222222	PERÚ	AREQUIPA	AREQUIPA
3	333333333333333333	PERÚ	CALLAO	CALLAO
4	444444444444444444	PERÚ	TRUJILLO	TRUJILLO
5	555555555555555555	PERÚ	PIURA	PIURA
6	666666666666666666	PERÚ	CUZCO	CUZCO
7	777777777777777777	PERÚ	HUANCAYA	HUANCAYA
8	888888888888888888	PERÚ	CHALCO	CHALCO
9	999999999999999999	PERÚ	CALLAO	CALLAO
10	10101010101010101010	PERÚ	TRUJILLO	TRUJILLO

¹ Laborategiaren hasierako discirua eta proposamena: Arantza Iraizorza.

• SQLJ

Datu-baseak atitzeko SQLJ programak erabiliko ditu. Programa honek *egelan* daude. Bezero makinak bakoitzean **DBG** karpetatik edo bestela BIULDuko zuen kontuan).

Fitxategiak editatu, kompilatu eta **ejecutar** konfigurazioa egiteko:

Honako pausoak jarrai itzazu, konfigurazioa egiteko:

- IDEvelopper hasieratu (*default role* aukeratu) – gure kasuan dagoeneko hasieratuta dago.
- Goazen Java programak exekutatzaten hastera. Lehenengo ariketarekin hasiko gara (kontrolatzaleen erabilpena). Lan-erenu berria definitu, menuan <File>+<New>+<Applications>+<GenericApplication>+<Applications> eta laguntzailea jarraitu, arabera, <File>+<New>+<General>+<Applications>+<Applications>) eta honako datuak jarriz:

Aplikazioaren izena: laborategiaSQLJ
Aplikazioaren direktorioa: C:\dbg\laborategiaSQLJ (zuen dbg karpetaren kokaguneaten arabera)

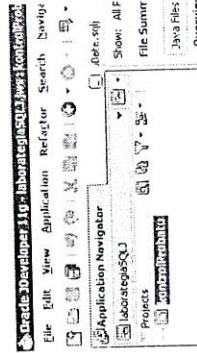
<Next> eta, bertan,

3. Projektu berri bat:

Projektuaren izena: kontrolProbatu

Projektuaren direktorioa: C:\dbg\laborategiaSQLJ\kontrolProbatu

Bukatzenean <Finish> (<Acceptar>) botofia sakatu, eta nabigatzalea horrela ikusiko dugu:



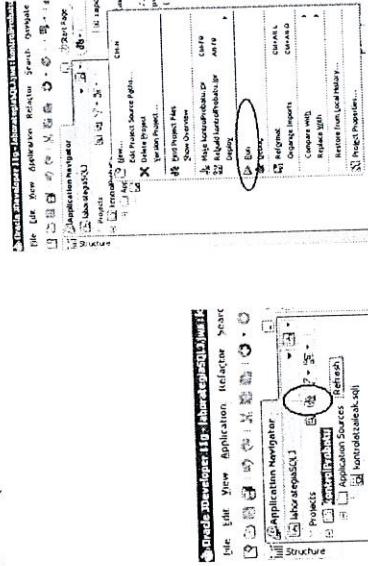
4. Projektuaren propietate lehenetsiak aldatu, horretarako menuan <Application> + <Project Properties> aukera hartu (bertsoaren arabera, <Tools>+<Project Properties>).

Ondorengoko aukeretan honako balioak ezartzi:

- Features aukeran *SQLJ Integration, Java era Extension Development* derrigorrez agertu behar dute, bestela gehitu.
- Libraries and Classpath aukeran *SQLRuntime* eta *Oracle JDBC* paketeak gehitu (<Add Library> botoiarekin).
- Run/Debug/Profile aukeran <Edit> klikatu eta gero <Tool Settings> aukera, hor *Allow Program Input* markatu, eta <Accept>.
- Compiler > SQLJ aukeran *Code generation parametroan oracle* balioa jarririk.

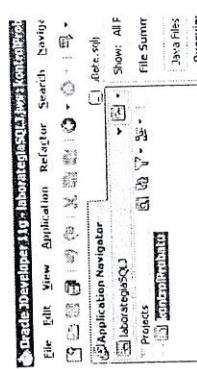
5. C:\dbg\laborategiaSQLJ\kontrol\src karpeta (existitzen ez bada, sortu) lehen ariketarako erabiliko dugun fitxategia kopiatu (kontrolazialeak.sql).

6. Lan-fitxategia kompilatu (*Refresh* egin agertzen ez bada) <Build> eta ejecutar, menuan <Run>+<Run projeto-izena.ipr> aukeratu, projektuaren izenaren gainean saguaren eskuekin botoa sakatu eta azpi-menuan <Run> aukeratu (ikus eskuekin hurrengo irudia).



7. Inoiz inguruna eguneratu nahi bada <View>+<Refresh> (edo bi gezi diuten ikonoa, irudia ikusi) aukeratu eta hor fitxategia klikatu (ikus ezkerreko irudia).

Ute honetan presi gaude lehen ariketari heltzko. Ariketa honekin bukatzen dugunean, era hurrengoei heltzko, honako prozesua jarraituko dugu. Ariketa bakoitzeko projektu berri bat sortuko dugu *laborategiaSQLJ* karpeta (<File>+<New>+<General>+<Projects>) eta 4, 5 eta 6 pausoak jarraituko ditugu.



1. Kontrolatzailen erabilipena. 4. motako JDBC kontrolatzaillea (Fixategia: kontrolatzaila_leak.sql)

Oraceleren *thin* kontrolatzalea erabiliko dugu ("oracle.jdbc.driver.OracleDriver"). Konexioa egiteko datu-basea ere beharko dugu ("jdbc:oracle:thin:@u017943.gi.ehu.es:1521:erreald"). Aplikazioak behar bezala funtzionatzen duela egiaztu ezazu. Proba egiteko, datu-basean jadanik sartuta dauden *na* zenbakiek saritzu (adibidez, 11111111 edo 22222222).

2. Iteratzailaren erabilipena (Fixategia: iteratzaila_leak.sql)

Aplikazioa aldatu, datu-basean erregistratutako bezeroen zerrenda pantailaratu dezan, konkreteki *na eta izena (ini, String)*.

3. Transakzioen definizioa (Fixategia: transakzioak.sql)

Aplikazioa aldatu, "*eskera erregistratu*" erabilen-kasua implementa dezan. Erabiltzen-kasu horren deskribapena: eskaeraren datuak erabiliztaileari eskatu ondoren (produktuaren kodea eta zenbat unitate nahi dituen), eskaera erregistratu behar da eta gainera dagoen erabiliztailearen historikoa (*hots, eskeraKop* attributuan) eguneraketa ere burtu behar da eskaera berri bat egin dela adierazit. Oharra: eskaeraKop attributu eratorri bat da, bezorako egindako eskaera-kopurua adierazten duena, eta ESKAERA taulan dagoen informazioarekin koherentea izan behar da.

Implementazioan transakzio bat definitu behar da. Aplikazioak behar bezala funtzionatzaten duela egiaztu ezazu.

Zein desberdintasun dago *commit* agindua jartzearen eta ez jartzearen artean?

4. Errorean tratamendua (Fixategia: erroreaktratatu.sql)

Aplikazioak "*bezerra erregistratu*" erabilen-kasua implementatzaten du. Konpiliatu eta egikaritu. Bi proba egin:

- (1) na: 87654321; izena: Jon Garmendia → datu-basean ondo erregistratu behar da.
- (2) na: 1111111; izena: Miren Lasa → Errorea itzultzen du.

Bigarren kasuan, zein da itzultitako errorea? Zergatik?

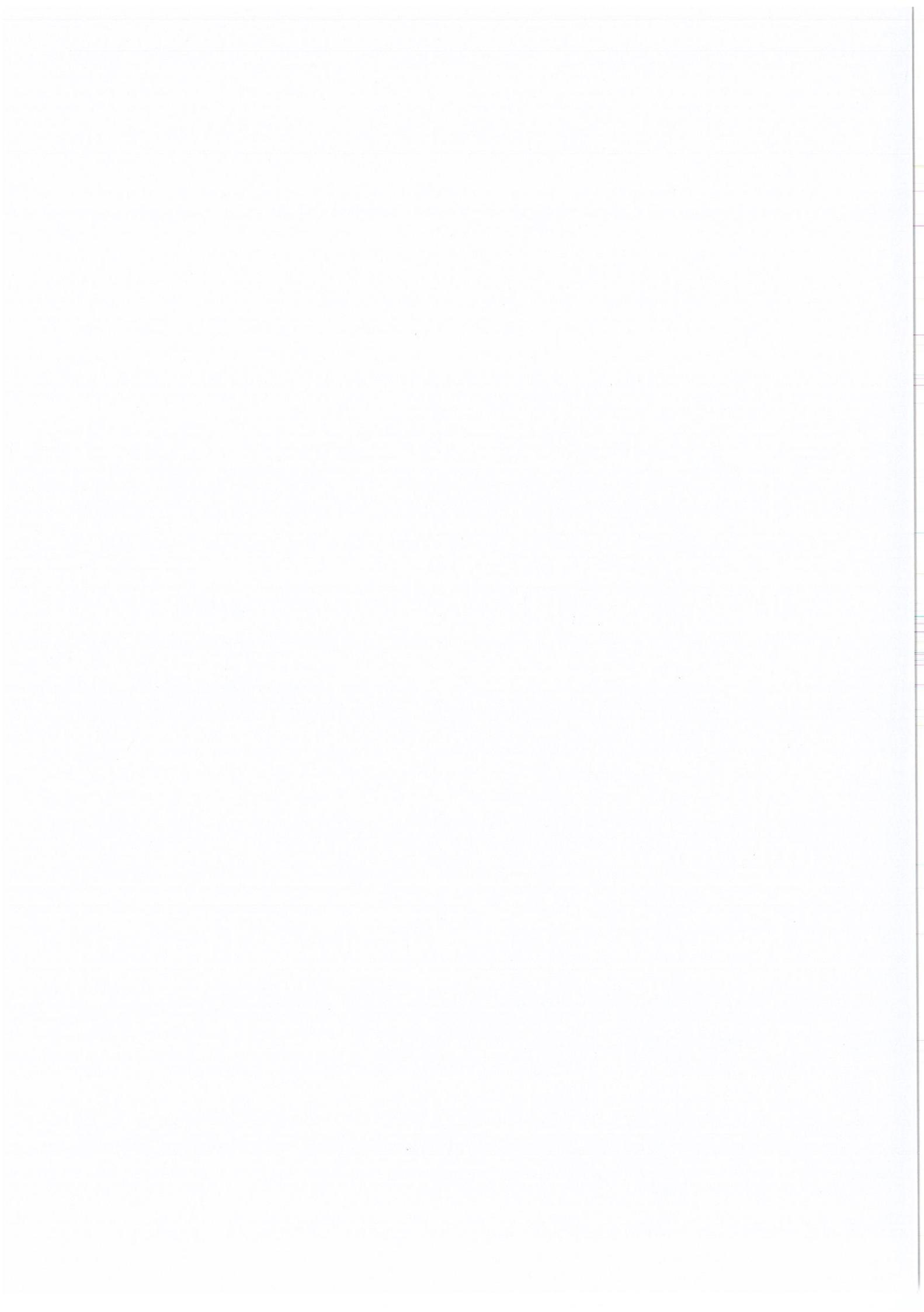
Aplikazioa errore horren aurrean sendoagoa izan dadin, behar diren aldaketak egin itzazu (adibidez, mezua egokia eman ondoren prozesua berriz hasteko aukera ematea).

5. Konexio bat baino gehiagoren erabilipena (Fixategia: datukopia.sql)

Aplikazioa osatu ezazu, ERREALA datu-basean dauden datuak erabiliz ATHLETIC datu-baseko ESKAERAOSOA(bzenb, dzemb, data, kantitate, bizena, pizena) taula bete dadin.

Oharrak:

- Oracleen taulak edo attributuak berriztatzeko ezin da AS erabili; beraz honela egin behar da: `select izena iz from bezeroa A where A.na=1;`
- Iteratziale klasea definitzeko kode-lerroa, klase nagusiaren kanpoan jarri behar da, jaso, "class ..." idatzi baino lehen.
- SQLko kontsulta cdo eragiketa batean iteratziale bat, bere metodoarekin, erabili daitete (bi puntu jarritu, adibidez :iter:izena()), baina batzuetan errorea eman izan du. Kasu horietan iteratzialearen metodoarekin lortutako baiola aldagaia batean jaso era gero SQL eragiketean aildagai hori erabili.
- Fixategiak .sql hedapena izan behar du, bestela java hedapena badu, kompiliatzailak #sql marka guztietan errorea omango du.
- Data bat sartzeko honako formattua erabili: dd-mm-aa, non mm mmm hilabetearren izenezko lehen hizkia diren (gaztelerazi). Data SQL eragiketaren barnean erabilitzten badugu orduan, adibidez, 03-dic-2009' (komatxo simplea) edo String aldagaien batean esleitu eta gero aldagai eragiketaren barnean erabilitzten badugu, orduan String nireData = "03-dic-2009" (komatxo bikotza).
- Klaseen izenak jartzearakoan, kontuz letra larri (maiuskula) eta xehekin (minuskulekin).



transaktion

↳ unitate logica da.

↳ bi buletels modu

Commit → singe buletlu

Acasă

Lo 4 acasă

→ Eguivalență gelde

rollback → deac deschisă

bi multe → anumite ferme

momentan sortabile

→ Înărâtate zilnic → behin-behindare adăucere căză

→ Multe Oameni

→ Înărâtate excepțională

transacție buletels aprobate

1) Isolation Level

- a) uncommitted
- b) committed
- c) repeatable
- d) Serializable

2) Lock table

3) Select For ~~not~~ update

Serial transaction

↳ Ez deoarece nu se acordă.

Ez - Serial transaction

Berdea execută,



- a) Serializabilă → scrierea posibila întâia oară → folosește edo prototip, edibili și urmări din nou în următoarele schimbări.
- b) Non-serializabilă

Prototip

↳ Executată o înserire de prototip

- a) lock_s → peretește → înlocuibile cu inserările
- b) lock_x → obiectivă → folosește
- c) unlock → obiectivă liberă egale

ACID propionate

transkription katalysiertes gomadisell

↳ artells eller-blokkene katalyse
TAA

→ P.S.E. ogenetekle eleidin.

→ Serello triflue unriatka

↳ selekt kompleksaske baino soilik behin.

1.ariketa

```
import java.io.*;
import java.sql.*;
import sql.runtime.ref.DefaultContext;
// iteratzailera SELECT galderarentzat
#sql iterator NireIter (int ina, String izena);

public static void main ( String args[] ) {
    NireIter iter;
    String izena, sizena;
    int kod, kopurua;
    BufferedReader sarrera = new BufferedReader (new InputStreamReader (System.in));
    try {
        /* DBrekin konexioa sortu eta lehenetsitako konexio testuingurua definitu */
        try {
            /* DBrekin konexioa sortu eta lehenetsitako konexio testuingurua definitu */
            Driver kontrolatzailera = (Driver) (Class.forName("oracle.jdbc.driver.OracleDriver")).newInstance ();
            DriverManager.registerDriver(kontrolatzailera);
            DefaultContext test = new DefaultContext("jdbc:oracle:thin:@ijpcl0.si.ehu.es:1512:erreala",
                "erabil", "nerea", false);
            DefaultContext.setDefaultContext(test);
            BufferedReader sarrera = new BufferedReader (new InputStreamReader (System.in));
            try {
                /* Konexioaren salbuespenea tratatu */
                catch(Exception e) {
                    System.err.print ("Errorea konexioan " + e.getMessage ());
                    exit(1);
                }
            }
            while (true) {
                System.out.print ("Irakasgaiaren izena sar ezazu (0 bukatzeko): ");
                izena = sarrera.readLine();
                /* izena.equals("0") */ break;
                /* Irakasgaten izenak ez direla errepikaten suposatzen dugu */
                } // salbuespenea tratatu
            }
            /* iteratzaileraen tratamendua */
        }
        /* salbuespenea tratatu
        */
    }
    /* iteratzaileraen izenak ez direla errepikaten suposatzen dugu */
    /* select kodea, berealdea into :kod, :sizena from irakasgata where izena = :izena;
    #sql kopurua = {select count(*) from ikas-irakas where kodea = :kod};
    iter = {select ina, izena from ikastea natural join ikas-irakas where kodea = :kod};
    #sql kopurua = {select count(*) from ikas-irakas where kodea = :kod};
    System.out.println ("Irakasgai: +" + izena);
    System.out.println ("Saila: +" + sizena);
    while (iter.next())
        System.out.println ("+" + iter.ina() + " " + iter.izena());
    iter.close();
    System.out.println ("Ikastetxe-kopurua: +" + kopurua);
}
} catch(SQLException e) { System.err.println ("Errorea: " + e.getMessage());}
}
```

*Aldegorriko kidea
ez da bide berria
ez da bide berria
ez da bide berria*

1. Anleitung

```
import java.io.*;
import java.sql.*;
import sql;.sunshine.net.DefaultContext;
public class Testberreich {
    Iterator iter
    String
    int
    BufferedReader source = new BufferedReader (new InputStreamReader
    (System.in));
    try {
        try {
            // DB-Abfrageausgabe
            while (true) {
                System.out.print ("Urkundennummer :");

```

```
#sql iter = !Select bere_Soik, from Irdbeslece IJ(Irdbesgaike IJ Projektive CW
Irdbesgaike.bere_soile = Projektive.bere_soile) CW Irdbeslece.bere_soile = Projektive.bere_soile
while (iter.next()) {
    #sql iter1 = !Select Kunde Bereic, from Irdbeslece where bere_soile = :iter
    #sql iter2 = !Select Kode Bereic, from Irdbesgaike where bere_soile = :iter
    #sql iter3 = !Select gloden, from projektive where bere_soik = :iter.bereid;
    while (iter1.next()) {
        System.out.println (iter1.Kunde);
    }
    System.out.println (iter2.Kode);
    while (iter3.next()) {
        System.out.println (iter3.p_kode);
    }
}
```

```
        System.out.println (iter1.Kunde);
    }
    System.out.println (iter2.Kode);
    while (iter3.next()) {
        System.out.println (iter3.p_kode);
    }
}
```

```
import java.io.*;  
import java.sql.*;  
import java.sql.runtime.ref. DefaultContext;
```

class Student:

```
public static void main (String args [ ]) {
```

```
BufferedReader sarrere = new BufferedReader(new InputStreamReader(sistema.getInputStream()));
try {
    try {
        DriverManagerDataSource = (DriverManagerDataSource) forName("oracle.jdbc.driver.OracleDriver").newInstance();
        DefaultContext test = new DefaultContext("jdbc.oracle.thin:@jriplc.5i.ehws.1512.errida", "erridi", "nroce", false);
```

DefaultContext.setDefaultContext(test);

System. cat. prints ("Kontexte abgelaufen").

try catch (Exception e) {

`System.out.println("Error message: " + e.getMessage());`

exit(-1);

while (true) {

iter1 → Index
True → p)

while

~~while (iter.next())~~

~~System.out.println (iter, berechnete);~~

~~System. aut. priimtu (lit. Irklašgaičiai) + Irklašgaičiai (zen.)~~

~~System out.~~

• price = surve.readLine(); // suppose string; car of parallel
• price = surve.readLine(); // suppose double base: //

```
if (filter != null) {  
    System.out.println("Filter: " + filter);  
    System.out.println("System.out.println("Filter: " + filter));  
    System.out.println("System.out.println("Filter: " + filter));  
    System.out.println("System.out.println("Filter: " + filter));  
}
```

while (iter < n || !nxtC) {
if (iter == n) {
 cout << "FileList" << endl;
 break;
}
cout << fileList[iter] << endl;
iter++;
}

#591 ~~lack Sulfide / Sulfate from sulfide~~
#591 ~~lack Sulfide / Sulfate from sulfide~~
#591 ~~lack Sulfide / Sulfate from sulfide~~
#591 ~~lack Sulfide / Sulfate from sulfide~~

```
nan = Starting.readline();  
if (now.equals("0,0")) break;  
if (sql_selecton == 1) select_from_Mailbox();
```

↳ `getAvailableTitles()` `NAU` `seebriefe` `sabtu` (`O bultzelelo`): ";

2.ariketa

```
import java.io.*;
import java.sql.*;
import sqlj.runtime.ref.DefaultContext;

// iteratzileen definizioa
#sql iterator IteraBat(String bere-saila);
#sql iterator IteraBi(String izena);

class SailBakoitzarenTxostena {
    public static void main ( String args[ ] ) {
        IteraBat iter1;
        IteraBi iter2; iter3; iter4; //Irakasleak, Irakasgaiak eta Proiektuak lortzeko
        String izena; //Orokorrean edozein taulatatik izena jasotzeko

        try {
            try {
                /* DBrekin konexioa sortu eta lehenetsitako konexio testuingurua definitu */
                Driver kontrolatzalea = (Driver) (Class.forName("oracle.jdbc.driver.OracleDriver")).newInstance();
                DriverManager.registerDriver(kontrolatzalea);
                DefaultContext test = new DefaultContext("jdbc:oracle:thin:@jiplc0.si.ehu.es:1512:erreala",
                "erabil", "nerea", false);
                DefaultContext.setDefaultContext(test);
                System.out.println ("Konexioa eginda");
            } catch(Exception e) {
                System.err.print ("Errorea konexioan " + e.getMessage());
                exit(-1);
            }
        }

        #sql iter1 = {(select distinct bere-saila from irakaslea) union (select distinct bere-saila from
        irakasgaia) union (select distinct bere-saila from proiektua)}; //sail guztiak lortzeko

        while ( iter1.next( ) ) {
            izena = iter1.bere-saila( );
            System.out.println("SAILA: " + izena);
            #sql iter2 = { select izena from irakaslea where bere-saila = :izena };
            #sql iter3 = { select izena from irakasgaia where bere-saila = :izena };
            #sql iter4 = { select izena from proiektua where bere-saila = :izena };

            System.out.println("Irakasleak: ");
            while (iter2.next( )) System.out.println(" " + iter2.izena( ));
            System.out.println("Irakasgaiak: ");
            while (iter3.next( )) System.out.println(" " + iter3.izena( ));
            System.out.println("Proiektuak: ");
            while (iter4.next( )) System.out.println(" " + iter4.izena( ));

        } iter1.close( ); iter2.close( ); iter3.close( ); iter4.close( ); }
        catch(SQLException e) { System.err.println("Errorea: " + e.getMessage()); } } }
```


Datu-baseak

Ana Arruarte (a.arruarte@ehu.eus) – 240 bulegoa

Edukia: Datu-baseak (DB) izeneko irakasgaiak, graduaren barruan, izen bereko lerroari hasiera ematen dion irakasgai da. Irakasgaiari lotutako oinarrizko kontzeptuak terminologia, datu-baseen inguruo oinarrizko kontzeptuen ezagutza eta eriazionala; datu-base kudeaketako sistema baten arkitektura eta osagaiak; eredu egunairezak SQL lengoaia estandarra erabiliz; datu-base eriazionalak atitzeko aplikazioen garapena; transakzioen definizioa eta ezagutarrak; konkurrentzia-kontrolaren oinarrizko kontzeptuak.

Gaiak:

1. Sarerra.
2. Eredu erlazioan eta aljebra erlazionala.
3. SQL lengoia: DBak sortu, eguneratu eta kontsultatu.
4. SQL aurreratu.
5. Transakzioak. Definizioa eta erabilpena.
6. DBak atitzeko aplikazioen garapena.

Eskumenak: -Datu-baseen inguruko kontzeptuak ezagutzea eta hainbat lotutako terminologia egoki erabilizte; DB, DBKS, DBS, katalogoa, erabitzaleak eta datu-ereduak, bestek beste.
-DBsak fixategiekin konparatzen baditugu eskaintzen dituzten abantailak identifikatzea.

-Datu-baseen eskeema logikoa kudeatzea eredu eriazionalak baliatuz.
-DBaren gaineko kontsultak zehazteko aljebra eriazionaleko eragiketak erabiltzen ikastea.

-SQL lengoaia estandarrean trebezia lortzea DBKS eriazional desberdinetan, bestek beste, DB eskemak sortu eta eguneratzek; baita kontsulta simpleak bai aurreratuak zehazteko ere.
-DB eriazionalak aplikazioetarakit atitzten ikastea.

-Transakzio bat zer den ulertzea eta bere propietate nagusia kudeatzzen ikastea.
-DBS-en konkurrentzia-kontrolen oinarrizko kontzeptuak ulertzeara.

Titulazioaren baitan jasotzen diren oinarrizko eskumenen dagokienez, honakoak landuko dira:
-Oinarrizko edukien eta teknologiarren ezagutza eskuratzeara, batetik, etorriko diren metodo eta teknologia berrien ikasketa eta garapena eratzeko, bestetik, egoera berrierrara egokitzeko analmenea eskuratzeko.

-Arazoi ekimen propioarekin aurrez egitea.

-Zeharkako eskumenen dagokienez:

-Antolaketa eta sintesi ahalmena landuko dira, batez ere kontzeptu teorikoei lotuta.
-Talde-lana.
-Egoera berrierrara egokitzea.

Metodologia: Sei kredituko (4 teoriko eta 2 praktiko) irakasgai homek, 150 ordu eskatzen ditu guztira, 60 aurrez aurrekoak eta 90 aurrez aurtekoak. Hiru dira aurrez aurreko jarduera motak: eskola teorikoak ego magistrak, ariketa-eskolak eta laborategi-saiok. Irakasgaiari lotutako eskumenak eskuratzeko orduan, bi azkeneko jarduera mojetan ikasleen parte-hartzea aktiboa handiagoa izango den arren, eskola teorikoetan ere parte-hartze hori bultzatuko da galdera-erantzun laburen bidez.

Laborategi-saiok 5 izango dira. Aurrez aurrekoak ez diren orduak praktikak prestatzeko eta bukaerako txostenak egiteko erabiliko dira. Ariketa-eskola batzuetaen ikasleen lanak jasoko dira, eskola-orduetan garatutakoak, erablatu ahal izateko. Gainera, ikasleei banakako arreta emateko asmoarekin, tutoretza-orduak egongo dira.

Ebaluazioa: Irakasgai gainditzeko bi bide disjuntu daude: bildumazko ebaluazioa (edo globala) edo prestakuntzazko ebaluazioa (jarraitua).

Prestakuntzazko ebaluazioan jarduera desberdinak honako pisua izango du:
-Irakasgaiaren jarraipenari lotutako probak (data zehatzetan egindako hiru hurren): % 70. Bi azken azterketetan 3,5eko nota minima (3,5/10) eskatuko da
ebaluazio jarraituarekin irakasgai gainditu ahal izateko.

- 1. azterketa (1 eta 2 gaiak): 2016ko otsailak 25, osteguna, 1. aste trinkoa (9:00) - (% 15)
- 2. azterketa (3 eta 4 gaiak): 2016ko apirilak 21, osteguna, 2. aste trinkoa (9:00) - (% 35)
- 3. azterketa (5 eta 6 gaiak): 2016ko maiatzak 27, ohiko deialdia, ostirala (15:00) - (% 20)

-Lanen edo ariketen ebaluazioa (aldez, aurretik ezarri gabeko datetan jasotako lanen edo ariketen ebaluazioa): % 20
-Laborategietako lan praktikoen ebaluazioa: % 10
Ikasleek erabaki beharko dute zein ebaluazio mota nahiago duten, hori bai, prestakuntzazko ebaluazioa hautatu ahal izateko derrigorezkoa da irakasgaiari lotuta dauden jardueretan parte-hartze aktiboa izatea. Prestakuntzazko hautua egiteko, ikasleek irakasgaiaren iraupenaren % 60 eta % 80 anera iristean artean egingo dute hautua, eta irakasleak erabakiko du, ikaslearen ordura arteko errendimendua kontuan hartuta, ikaslea onartu ala ez.

Ebaluazio globalen notaren % 100 azterketa batzen bidez lortuko da (ezinbestekoak da azterketan bai SQLri dagokion atala ganditzea bai SQL eta transakzioei dagokion atala) beti ere laborategiak zure kabuz egin eta entregatu badituzu azterketa eginenerako.
Ohiko deialdia: 2016ko maiatzak 27, ostirala (15:00).
Ezohiko deialdia: 2016ko ekainak 27, astelena (9:00).

Tutoretza-orduak: Eskola-egunak irauten duten bitartean:
Astearteak - 10:30-12:30 era 15:00-17:00
Ostegunak - 10:30-12:30
Ordutegia desberdina da aste berezietaan eta eskolak bukatu ostean (GAU/Ren begiratutu).

Bibliografía

Oinarriakoa

Datu-base sistemak. Oinarriak. R.A. Elmasri eta S. B. Navathe. UPV/EHU, 2009.

Fundamentals of Database Systems. R.A. Elmasri eta S.B. Navathe. 7th Edition. Pearson Addison Wesley, 2015.

Bestelakoak

Datu-baseen aljipena SQLaren bitartez. Sintaxia eta transakzioen diseinua. A. Irazorza Gorri, Barne-Ixostenak UPV/EHU/LSI/TR 9-2002, 2002.

Fundamentos de Bases de Datos. H. Korth, A. Silberschatz eta S. Sudarshan. Mc. Graw-Hill, 2002.

Sistemas de Bases de Datos. Un enfoque práctico para Diseño, Implementación y Gestión. T. Connolly y C. Begg. Addison Wesley, 2005.

Introducción a los Sistemas de Bases de Datos. C.J. Date. Prentice-Hall, 2001.

GUIA LAN Times de SQL. J. Groff y P. Weinberg. Osborne McGraw-Hill, 1998.

Introducción a los Sistemas de Bases de Datos. J. Ullman eta J. Widom. 1999, Prentice Hall Data Base Principles Programming Performance. Second Edition. P. O'Neil. 2001, Morgan Kaufmann.

SQL y Java: guía para SQLJ, JDBC y tecnologías relacionadas. J. Melton eta A. Eisenberg. 2002, Ra-Ma.

Java programming with Oracle SQLJ. J. Price. 2001, O'Reilly.

2. Arzite

- a) Gdlo - nuristopen, existitzen der Oinnerstele Gdele setzen
a leglich
~~Integritate erreferen~~
- b) Arcorile ex
- c) Entite - integratice, gde Oinnerstele gdele cai de null i2ca
- d) Integratice - erreferentiale, ex de existitzen subverbali
baile koreferencie.
- e) Integratice - erreferentiale, basic tuple betraeten erfullbar de
- f) ~~Ex-Arc~~ Arcorile ex
- g) Arcorile ex
- h) Integratice - erreferentiale, existitzen O koreverbali ~~existitzen~~
existitzen litzalee, baile ex za idago iweren. ~~Re~~
- i) Arcorile ex
- j) ~~Integratice erreferentiale / Baile koreverbali existitzen litzalee
baile 1 koreverbali ex Arcorile ex~~
- k) Gdlo - nuristopen
Integratice - erreferentiale, existitzen ex der subverbali zerbolini
eslitiles litzalee.
- l) Arcorile ex
- m) Integratice - erreferentiale, ex de existitzen subverbali ex
Subverbali = 3 der subverbali.

6. Ariketa

Ibai Baglietto, Xabier Garmendia

Aitzol Etxeberria

Ingelesa Bai ↔ Hiszpania = ingelesa (Hiszpania)

Ingelesa Eta ↔ Hiszpania - Ingelesa Bai

NA Ingelesa Bai ↔ ΠNA (Ingelesa Bai)

NA Ingelesa Eta ↔ ΠNA (Ingelesa Eta)

NA Ingelesa + ↔ NA Ingelesa Bai - NA Ingelesa Eta

NA Ingelesa ↔ NA Ingelesa Bai - NA Ingelesa +

E6 ↔ ΠNA, Izen (Gidori * NA Ingelesa)

12. Ariketa

HS - Hotelbo-Txango ↔ Η Helmuga, Irteeradatu (Hotelbo-Txango)

HS Bidai ↔ Η Helmuga, Irteeradatu (Bidai)

HS Eta Txango ↔ HS Bidai - HS Hotelbo-Txango

E12 ↔ Η Helmuga HS Eta Txango

13. Ariketa

HErabilgari ↔ Helmuga, Irteeradatu, Izen Hotel (Hotel-Bidaia)

E13 ↔ Σ MAX cant. Hotel (HErabilgari)

14. Ariketa

Gidori Bidai ↔ (ΠNA, Izen (Gidori) * Η Helmuga, Irteeradatu, NA (Bidai))

Bezero Bidai ↔ (ΠNA, Izen (Bezero) * Η Helmuga, Irteeradatu, NA (Hotel_Bidai... Bezero))

E14 ↔ Σ (Η Helmuga, Irteeradatu, Izen (Gidori Bidai) * Η Helmuga, Irteeradatu, Izen (Gidori Bidai))

Konforme.

E1 Konsiderakoen ere posible de. Soluzioa.

Ardo batea
era zego
soluzioa eiter.

5. Arikete

Txango Bezero \leftarrow Helmuge, Interesatx, NA \rightarrow cont. Zeballos (Hautzulo, Txango-Bezero)

Ideia ✓

Bezeroak \leftarrow cont. Zeballos \rightarrow Txango Bezero

Berizendakete fette de

Hotel-Bezeroak \leftarrow Hautzulo Hotel-Bidaia-Bezero \rightarrow Bezeroak

E5 \leftarrow $\Pi_{\text{Helmuge, Interesatx, Deux}}$ Hotel-Bezeroak \times Hotelak

8. Arikete

PMax \leftarrow MAX prezio (Hautzulo-Txango)

HTMax \leftarrow $\Pi_{\text{Prezio = pMax}} (\text{Hautzulo-Txango})$

Ideia / Berizendakete
Kontuz!

Bidai Max \leftarrow $\Pi_{\text{Helmuge, Interesatx (HTMax) \times Bidaia}}$

Gideri Max \leftarrow Gideri \times Bidai Max

E8 \leftarrow $\Pi_{\text{Helmuge, Interesatx, Deux, NA}} (\text{GideriMax})$

15. Arikete

B6a Kapurra \leftarrow Helmuge, Interesatx, NA \rightarrow sum. Gallopurra (Hold-Bidaia-Bezero)

30Gcu \leftarrow $\Pi_{\text{sum. Gallopurra > 30 fGallopurra}}$

NA galduke,
ole latu?

E5 \leftarrow $\Pi_{\text{NA, Deux - (30Gcu * Bezero)}}$

9. Arbeit

JBidic & THelmuge = Indirektes Interatext = 2013-06-01 (Bidic)

PBidic & TRegundoReBido (JBidic)

Garestick < TRegundoReBido > PBidic (Bidic)

HGrestick < THelmuge, SteercData (Garestick)

BedorGarestick < Hotel-Bidic, Bearer * HGrestick)

RE9 & NA, helbider (Bedor * BedorGarestick)

6. Arbeit

IngleseBei & THelmuge = inglesc (Aislauntz)

IngleseEZ & THelmuge - IngleseBei

NAIngleseBei & NA (Inglese. Bei)

NAIngleseEZ & NA (Inglese. EZ)

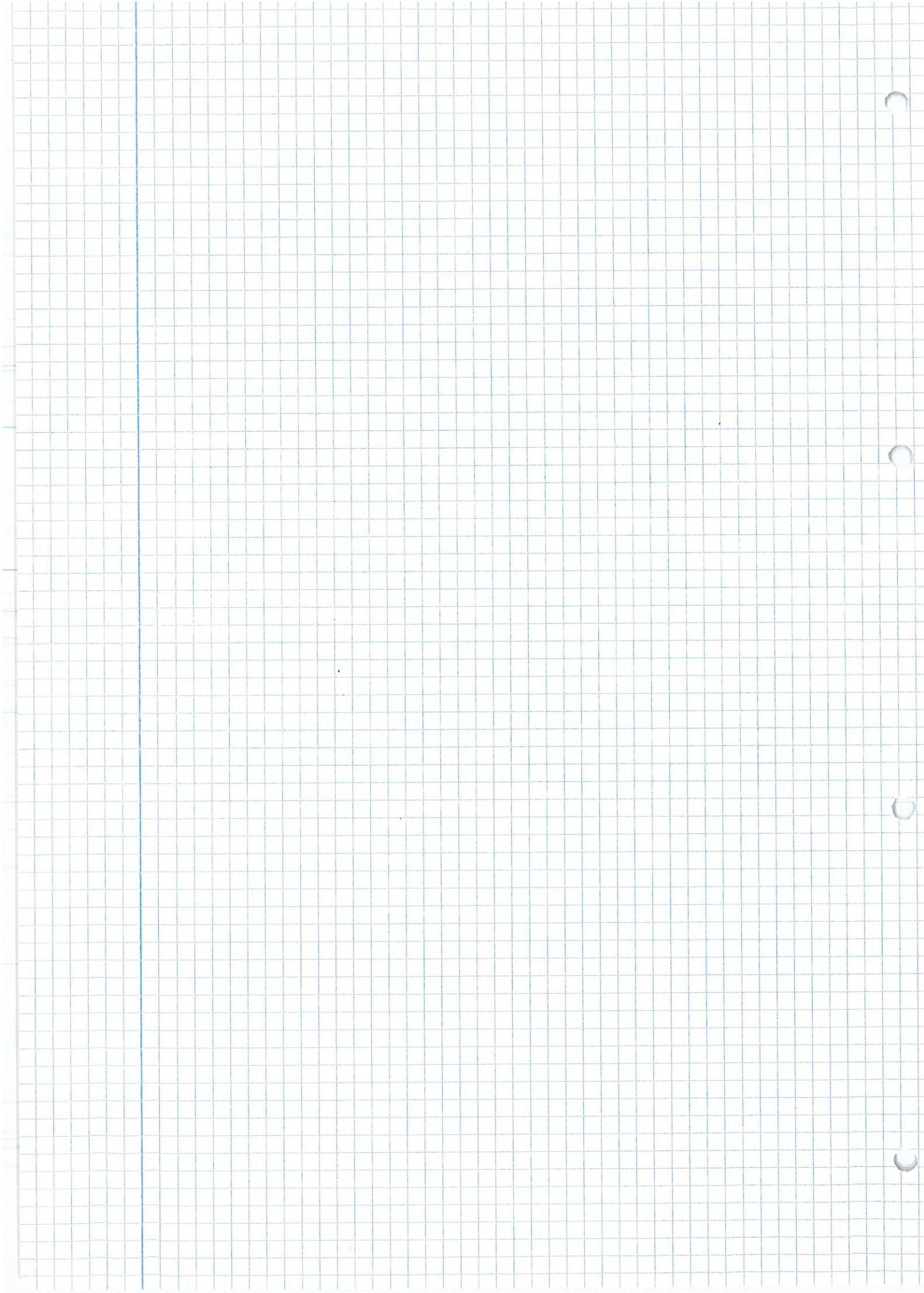
NAInglese + NAIngleseBei & NAIngleseEZ

NAIngleseEZ & NAIngleseBei = NAInglese +

E6 & NA, Izu (Gidari * NAIngleseEZ)

12. Arbeit

E12 & TH_{helmuge} Entrance



1. Arillete

1.

HM (NA, Hiskuntza) ← NA \approx Kontua Hiskuntza (Aialkuntza)

EHM

← ♂ Hiskuntza > 2 (HM)

E1

← ♂ ~~HM~~ (Gidori * EHM)

4.

Txango Bezero

← Helmuga, intercDate, NA \approx Kontua Zentzuli (Heldak - Txango - Bezero)

Baldintzakungo

← ♂ Kontua Zentzuli > 3 (Txango Bezero)

Bezerroku

← ♂ NA (Baldintzakungo)

E4

← ♂ Den, Helbidea (Bezero * Bezerroku)

2.

Bidic Donastik

← ♂ Interchiric = Donastik (Bidic)

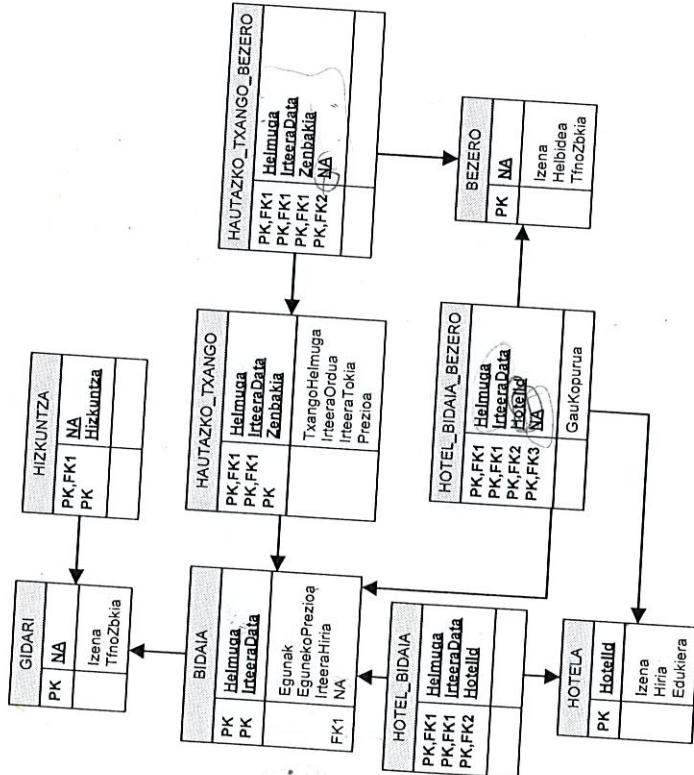
Donastik Gidori

← ♂ NA (Bidic Donastik)

BIDAIAK: aljebra erlazioaleko ariketak

Lanetik erretiratuta dauden pertsonentzat etsun bat baino gehiago irauten duten bidaiaik amolatzentzutu dituen ONDO-helmuga eta irteera-data erabilizten dira. Gainera, bidaiaaren iraupena egunetan, eguneko prezioa, irteera-hankuntzak ezagutzen dira.

- Hotel bakoitzeko bere identifikazio zenbaki bakarra, izena, hiria eta edukiera gordetzen dira.
- Bezeroen, NArekin identifikatuak, izena, helbidea eta telefono-zenbakia ezagutzen dira.
- Bidaiar batetan parte hartzen duen bezero bakoitzeko zenbat gau pasa dituen bezeroak bidaia horri esleitutako hoteletan gordetzen nahi da. Bidaia egun bat baino gehiago irauten duenez gerta daiteke bidaia beren hotel bat baino gehiago esleitu izana.
- Bidai bakoitzak hautazko txango bat edo gehiago eduki dituzte. Hautazko txango bakoitzaren ordena-zenbakia (1., 2., 3., ...), helmuga, prezioa, irteera-ordua eta irteera-tokia gordetzen dira. Zein bezeroek egun duen hautazko txango bakoitzera ere gordetzen da.
- Bezeroen, NArekin identifikatuak, izena, hiria eta edukiera gordetzen dira.
- Bidaiar batean parte hartzen duen bezero bakoitzeko zenbat gau pasa dituen bezeroak bidaia horri esleitutako hoteletan gordetzen nahi da. Bidaia egun bat baino gehiago irauten duenez gerta daiteke bidaia beren hotel bat baino gehiago esleitu izana.
- Bidai bakoitzak hautazko txango bat edo gehiago eduki dituzte. Hautazko txango bakoitzaren ordena-zenbakia (1., 2., 3., ...), helmuga, prezioa, irteera-ordua eta irteera-tokia gordetzen dira. Zein bezeroek egun duen hautazko txango bakoitzera ere gordetzen da.



Ondorenko kontsultak **aljebra erlazioalean** adierazez:

1. Bi hizkuntza baino gehiago menperatzen dituzten gidarien izenak.
2. Donostiatik ateratzen diren bidaia guzu-guztietan gidari lanak egin dituzten gidarien izenak (gildari bat emaitzan azalduko bada Donostiatik ateratzen diren bidaia guztietan parte haru behar du).
3. Bidiarik gabeko gidarien izena lor ezazu.
4. Lor itzazu bidaia berean hautazko hiru txango baino gehiago egin dituzten bezeroen izena eta helbidea.
5. Bidaia berean hautazko hiru txango baino gehiago egin dituzten bezeroentzat, lortu bidaia horietan erabiliz dituzten hotelen izenak. Emaitzan bidaien identifikazioa ere agertuko da.
6. Lor itzazu ingelesez baino hitz egiten ez duten gidarien NA-a eta izena, hau da, ingelesez soilik hitz egiten dutenak.
7. Lor ezazu 2011-10-02an Granada helmuga izanik, bidaia horri lotuta eskaini ziren hautazko txango guztietan baino egon **ez ziren** bezeroen NA-a, hau da, bidaia horri lotuta eskaini ziren hautazko txango guztietan soilik egon zirenen.
8. Jakin nahi da zein gidarkiak gidatu duen hautazko txango garestienari dagokion bidaia. Emaitzan, bidaiaren identifikazioaz gain, gidarien NA-a eta izena ere agertuko dira.
9. Lor itzazu 2013-06-01 irteera-data eta India helmuga duen bidaia baino bidaiar garestiagotan parte hartu duten bezeroen NA-a eta helbidea.
10. Lor itzazu hautazko txangorik gabeko bidaia guztiek gidatu dinuen gidariaren NA-a eta izena.
11. Helmuga bakoitzeko jakin nahi da zein bidaian (Helmuga + IrteeraData) lortu zen diru kopuru handiena hautazko txangoekin. Emaitzan tupla bat agertuko da helmuga bakoitzeko era bertan, helmugaz gain, bidaia horretan lortutako diru-sarrera azalduko dira.
12. Lor itzazu hautazko txangorik gabeko bidaien helmuga-hiria.
13. Lor ezazu hotel erabilgarri gehien duen bidaiaik duen hotel kopurua.
14. Lor ezazu honako baldintza betetzen duten bidaien identifikazioa: bidaia berean izen berdineko gidariak eta bezeroak daude (Mikel-Mikel, Maite-Maite...).
15. Lor itzazu bidaia berean guztira 30 gau baino gehiago pasatu duten bezeroen NA-a eta izena.
16. Lor itzazu honako baldintza betetzen duten gidarien NA-a eta izena: hautazko txangoen batean bezerrik badago.
17. Lor ezazu Paris **ez beste** helmuga duen (Paris helmuga ez duten) bidaia **guzziak solik** gidatu dituen gidariaren izena.



Bidezile

1. Arikete

Hizkuntza (NA, Hizkuntza-kop)

← NA

↓ Count Hizkuntza (Hizkuntza)

T2

←

↑ Hizkuntza-kop > 2 (Hizkuntza)

T3

←

Gideri * ↑ NA (T2)

E1

←

Hizkuntza (T3)

2. Arikete

Donostia-Bidezia

↓ D1

↑ Helmuga, Interceta, Tercera Distric = "Donostia" (Bidezia)

E2

← P ↑ Helmuga, Interceta, (Bidezia) → Donostia-Bidezia

↑ Gideri * D1

3. Arikete

BEGidetzka ↑ NA (Gideri) = ↑ NA (Bidezia)

E3 a ↑ Hizkuntza (Gideri * BEGidetzka)

4. Arikete

T1 (Helmuga, Interceta, NA, Zerbitzu-kop) → Helmuga, Interceta, NA ↓ Count Zerbitzu (Hurtado-Traugo-Berriozar)

T2

← ↑ NA

(↑ Zerbitzu-kop > 3 (T1))

E4

←

* ↑ NA, Hebihizte (T2 * Bezero)

5. Arikete

T1 (Helmuga, Interceta, NA, Zerbitzu-kop) → Helmuga, Interceta, NA ↓ Count Zerbitzu (H. + B.)

T2

←

↑ Zerbitzu-kop > 3 (T1) → T3

t4 ← HotelBidezia Bezero * T3

t_5 or $\neg \text{Hotel}(t_4)$
 $\neg \text{Hotel}_{\text{Hermes}, \text{IntereDate}}(t_4)$

E_5 or $\neg \text{Hotel}_{\text{Feme}}(t_5)$
 $\neg \text{Hotel}_{\text{Hermes}, \text{IntereDate}}(t_5)$

6. Arithmetic

$$T_1(\text{NA}, \text{Zemb}) = \text{NA} \text{ Front-Hiskuntz. (Hiskuntz)}$$

$$T_2 \leftarrow T_{\text{current}} = 1 + t_1$$

t_3 a Hintertantz

t₄ $\leftarrow t_{NA} \oplus (\text{Titikuntza} = \text{tugelso } (t_3))$

56 ♂ AN, +_{zona} (Gidari * t4)

Z. Arilkutz

+1 ~~at~~ Helmweg = Gracida, InteractionDate = 2011-10-02 (Habichtshagen) ~~1000~~

$T_2 \leftarrow HVA$ (Hutubessy - Bezeau)

$$t_3 \leftarrow t_{NAC}(t_1)$$

$$t_4 = T_2 + t_3$$

+5 → #NA (F Helmige > Grandes or IntercoData > 2011-10-02 (Total...))

$$E2 \leftarrow T_3 - T_5$$

S. Arilleté

~~Hilversum, theater~~ Hilversum, theater
~~Persiske~~ Persiske } MAX Prefixe (Hilversum-theater)

T_2 $\leftarrow H_{NA, \text{Hebung}, \text{Intervall}} \text{ (Bildung } \star T_1\text{)}$

t_3 at $T_{\text{c}} \approx 10^{\circ}\text{K}$

68 a) Helmig, InterData, NA, Scen(t₃)

9. Arilote

- $t_1 \leftarrow \text{Helmuge} = \text{India}$ and $\text{InterDate} = 2013-06-01$ (Bidaic)
 $t_2 \leftarrow \text{Helmuge, InterDate, Eguchi} * \text{Precision}(t_1)$
 $t_3 \leftarrow t_2 \times \text{Precision} \rightarrow \text{Eguchi} * \text{Precision Bidaic}$
 ~~$t_4 \leftarrow \text{Helmuge, InterDate } (t_5)$~~
 $E9 \leftarrow \text{HNA(Helbidaic } (t_4 * \text{Hotel Bidaic-Bedec}) * \text{Bedec}$

10. Arilote

- $t_1 \leftarrow \text{Helmuge, InterDate (Bidaic)}$
 $t_2 \leftarrow \text{Helmuge, InterDate, Embodiment (Autodesk-Hxang)}$
 $t_3 \leftarrow t_1 - t_2$
 $t_4 \leftarrow \text{Bidaic} * t_3$
 ~~$t_5 \leftarrow \text{HNA}(t_4)$~~
 ~~$E10 \leftarrow \text{HNA, Dene } (t_5 * \text{Gidori})$~~

12. Arilote

- $t_1 \leftarrow \text{Bidaic - Autodesk-Hxang}$
 $E12 \leftarrow \text{H Helmuge } (t_1)$

anhang 4

a)

I-Saile \leftarrow ♂ SIBER = "Überhunte" (Saile)

EA

\leftarrow ♂ Abizem, IZEN Dein Abizem (Langile + Saile)

b)

5 Projekte \leftarrow ♂ S2BKDA = 5 (Projekte)

5 Lan

\leftarrow ♂ LGS2 (From 5 Projekte * Langile - Du)

\leftarrow ♂ IZEN (Langile * 5 LAN)

c)

LGs2 \leftarrow Count Familiertelko - Izen (Familiertelko)

F(Gs2, Familiertelko - Kop)

EC \leftarrow ♂ Abizem, IZEN (Langile * ♂(Familiertelko - Kop (≥ 1 (F))))

d)

FAMGS2/Gs2 \leftarrow ♂ Gs2 (Familiertelko)

Estdambs2 \leftarrow

ED \leftarrow FAMGS2 = (♂ Gs2 (Langile))

e) \leftarrow ♂ Abizem, IZEN (Estdambs2 * Langile)

Buru (LGs2) \leftarrow ♂ Buru GS2 (Saile)

FanBuru(Gs2) \leftarrow ♂ (Familiertelko * Buru)

EE

\leftarrow ♂ Abizem, IZEN (Langile * FanBuru)

Aufgabe 1

a) Arealen e2

b) Integrität erreichbar \Rightarrow 2 e2 existieren Seil & Blatt

c) Integrität - erreichbar, Gelenk - unerreichbar
(6S2 \Rightarrow existiert α : $\alpha_0 = \alpha_{null}$)

d)
e) Arealen e2 Integrität - erreichbar
f) Arealen e2 Entferne Integrität

g) Integrität - erreichbar \Rightarrow Längigkeit der Kette bestimmt
die Größe.

h) Integrität - erreichbar

i) Arealen e2

j) Integrität - erreichbar

k) Arealen e2

Ariketa 2 - BANKETXEA (2)

BEZEROA		KONTUA-BEZEROA		MAI	
IZENA GSZ SZ		IZENA GSZ SZ		ANG	
Iker	11	Bilbo	Tirso	IZENA GSZ SZ	3. Ondorengo aljebra erlazionaleko eragiketak eginda zein erlazio
Ara	22	Juan	Legazpi, 1	Kontua NAN	lortuko dugun/ditugun marratzu:
Karmele	44	Pedro	Guridi, 5	Zokta NAN	a) 1 proiektuan ordubete baino gehiago lana egiten dutenen GSZ:
		Pedro	Guridi, 5	0	PR1_1GEHIAGO → $\sigma_{Pz=1}$ AND $ORD>1$ (LANA-NON)
		Jon	33	0	E1 ← π_{GSZ} (PR1_1GEHIAGO)
		Ana	22	1	
		Karmele	44	2	
					b) Langileak dauzkaten proiektutako PZ: E2 ← π_{Pz} (LANA-NON)
					c) Langileik gabeko proiektutako PZ: PRO_GUZTIAK ← π_{Pz} (PRO) E3 ← PRO_GUZTIAK - E2
					d) Persona bakoitzaren datuak bere saileko datuekin: E4 ← LANG $\bowtie_{Sz=ZBKA} SAIL$
					e) Persona bakoitzaren izena bere sailaren izenarekin lotuta: E5 ← π_{IZENA} , SIZE (E4)

engoaia et Sistema Informatikoak saila

2. gata

engoaia et Sistema Informatikoak saila

2. gata

4.

Ariketa 3: aljebra eragiketak

f) Langile batek lana egiten duen proiektu bakoitzeko

lortu langilearen izena, bere GSZ eta proiektuaren

zenbakia (PZ):

PERT ← $\pi_{IZENA, GSZ}(LANG)$
LANA(GSZ, PZ) ← $\pi_{GSZ, PZ}(LANA)$
E6 ← PERT * LANANA

ROI

PZ

IZENA

GSZ

SZ

1

2

3

4

5

6

7

8

9

ANG

IZENA

GSZ

SZ

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

KONTUA

Kontu
Zokta

Inte.

Saldoa

Bankua

Suk
Zokta

1

2

3

4

5

6

7

8

9

10

11

ROI

PZ

IZENA

Heb

1

2

3

4

5

6

7

8

9

10

11

KONTUA

Kontu
Zokta

Inte.

Saldoa

Bankua

Suk
Zokta

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

ROI

PZ

IZENA

Heb

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

KONTUA

Kontu
Zokta

Inte.

Saldoa

Bankua

Suk
Zokta

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

KONTUA

Kontu
Zokta

Inte.

Saldoa

Bankua

Suk
Zokta

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

KONTUA

Kontu
Zokta

Inte.

Saldoa

Bankua

Suk
Zokta

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

KONTUA

Kontu
Zokta

Inte.

Saldoa

Bankua

Suk
Zokta

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

KONTUA

Kontu
Zokta

Inte.

Saldoa

Bankua

Suk
Zokta

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

KONTUA

Kontu
Zokta

Inte.

Saldoa

Bankua

Suk
Zokta

1



4. Arbeit

a)

~~Merkenz- π_{S2} & Merkenz-Seile~~

~~Merkenz-Seile~~

Merkenz-Seile \leftarrow $\pi_{S2} \text{Merkenz}(\text{Seile})$

$\pi_{Merkenz(S2)} \leftarrow \pi_{SailKettensatz(MerkenzSeile)}$

Merkenz \leftarrow Langile $* \pi_{Merkenz}$

Langile-Merkenz $\leftarrow \pi_{LangileHeld}(Merkenz)$

b)

Bott-Sailkettensatz \leftarrow $\pi_{Bott} = 5$ (Projektive)
 $\pi_{BSP(P8)}$

$\pi_{Sailkettensatz}$ \leftarrow $\pi_{BottSailkettensatz}$

$\pi_{SailkettensatzLGS2}$ $\leftarrow \pi_{LGS2}(\pi_{Sailkettensatz})$

$\pi_{Sailkettensatz} \leftarrow$ Langile $\times_{LGS2=LGS2} \pi_{SailkettensatzLGS2}$

Langile $\leftarrow \pi_{Langile}(Sailkettensatz)$

c)

π_{Fam}

$\leftarrow LGS2 \cap \pi_{Fam}(LGS2) \text{ (Familienstamm)}$

~~Geobefestigung LGS2~~

$\leftarrow \pi_{MontageLGS2}(Befestigung)$

Ec

$\leftarrow \pi_{Tabizer, Fam.} (\text{Langile} * \pi_{Geobefestigung})$

d)

Familietello^(GS2) \leftarrow π_{LGS2} (Familietello)

Langile GS2 \leftarrow π_{GS2} (Langile)

Familiegebello^(GS2) LangileGS2 - Familietello GS2

Familiegebello \leftarrow Langile \rightarrow Familiegebello GS2

Familiegebello \leftarrow $\pi_{ABidez, Izen}$ (Familiegebello Lur)

e)

Sai-Buru^(LGS2) \leftarrow $\pi_{BuruGS2}$ (Saila)

Buru Familie \leftarrow Familietello * Sai-Buru^(LGS2)

Buru Familie GS2 \leftarrow π_{LGS2} (Buru Familie)

BuruZegia \leftarrow Langile \star Buru Familie GS2

BuruZegia \leftarrow $\pi_{ABidez, Izen}$ (BuruZegia D)

Aljere erlazendello arrizaleku

3. Arikete

NA Gidari \leftarrow π_{NA} (Gidari)

NA Bidaciori \leftarrow π_{NA} (Bidaciori)

~~NA Bidaciori~~ \leftarrow ~~NA Gidari - NA Bidaciori~~

E3 \leftarrow π_{Izen} (Gidari * Bidaciori)

10. Arikete

H1 Hezkero-Argo \leftarrow $\pi_{Hezkero, Izen}$ (Hezkero-Argo)

NA Gidari \leftarrow Gidari * NA Bidaciori

H1 Bidaciori \leftarrow $\pi_{Hezkero, Izen}$ (Bidaciori)

E10 \leftarrow $\pi_{NA, Izen}$ (NA Gidari)

H1 Etxango \leftarrow H1 Bidaciori - H1 Hezkero-Argo

Bidaciori Etxango \leftarrow Bidaciori \neq H1 Etxango

NA Bidaciori ET \leftarrow π_{H1DA} (Bidaciori-Etxango)

Datu-baseak

HAVING

43. Bezeroek egin dituzten bidaia guztik kontuan haututa, lortu hotel hautatuaren izena eta esan zenbat aldi izan den hautatua.

```
SELECT Izena, COUNT(*)  
FROM HOTEL_BIDIA_BEZERO NATURAL JOIN HOTELA  
GROUP BY HotelId, Izena  
HAVING COUNT(*) >= ALL (SELECT COUNT(*)  
                      FROM HOTEL_BIDIA_BEZERO  
                      GROUP BY HotelId)
```

Lehenengo GROUP BY-ean Izena sartu behar select-ean erabiltzen baita funtzi agregatuarekin daima lotutako hautazko txangoen prezioaren batua: bidaia garestiena izatea, hautazko txangoen prezioen batuari erreparratzu.

```
SELECT NA, Izena, SUM (Prezios) AS Guztira  
      FROM (GIDARI AS G NATURAL JOIN BIDATA) NATURAL JOIN HAUTAZKO_TXANGO)  
      GROUP BY Helimgua, Irteeradata, GIDARI.NA, Izena  
      HAVING SUM (Prezios) >= ALL (SELECT SUM (Prezios)  
                                    FROM HAUTAZKO_TXANGO  
                                    GROUP BY Helimgua, Irteeradata)
```

OUTER JOIN

47. Lor itzazu hautazko txango guztien identifikazioa eta zenbat bezeroak hautatu duen txango bakoitza.

```
SELECT HT.Helimgua, HT.Irteeradata, HT.Zenbakia, Count (HT.NA)  
      FROM HAUTAZKO_TXANGO_BEZERO AS HTB RIGHT JOIN HAUTAZKO_TXANGO AS HT  
      ON HTB.Helimgua=HT.Helimgua AND HTB.Irteeradata=HT.Irteeradata AND HTB.Zenbakia=HT.Zenbakia  
      GROUP BY HT.Irteeradata, HT.Helimgua, HT.Zenbakia  
      Count (NA)=0 null baldin bada NA-ren balioa  
      berrian Count (*) jartzten badugu gaizki ergongo da Count (null)=1 baita
```

Datu-baseak

ZATIKETARANTZ...

1) Ze bueltatuko luke honako sententzia honek?

SELECT Izena

FROM IRASLE AS I

WHERE NOT EXISTS

(SELECT Kod BEZERO).

FROM IRAKSAGAI

WHERE Mota='Hautazko'

AND

NOT EXISTS (SELECT *

FROM MATRIKULA

WHERE I.NA=MATRIKULA.NA AND IRAKSAGAI.Kod=MATRIKULA.Kod.)

False

Datu hauetan?

NA	Izena	Jatorria
HA	Hautazko	IRAKSAGAI
HB	Hautazko	MATRIKULA

NA	Izena	Jatorria
HA1	Hautazko	IRAKSAGAI
HA2	Hautazko	MATRIKULA

NA	Izena	Jatorria
HA1	Hautazko	IRAKSAGAI
HA2	Hautazko	MATRIKULA

NA	Izena	Jatorria
HA1	Hautazko	IRAKSAGAI
HA2	Hautazko	MATRIKULA

Buruzko hautazko txangoen erantzuna litzateke:

"EZ dago hautazko irakasgairik non ikaslea EZ den mariatuatu". Beraz, elementu guztiek ezagarriz betetzearen parekoa da honakoa: ezagarrria betezten ez

Nahi izanez gero, probatu goiko sententzia bestetik hauetan?

IRAKSAGAI
IRASLE
MATRIKULA

IK1 Hau...
IK1 EH1

IK2 Hau...
IK2 EH1

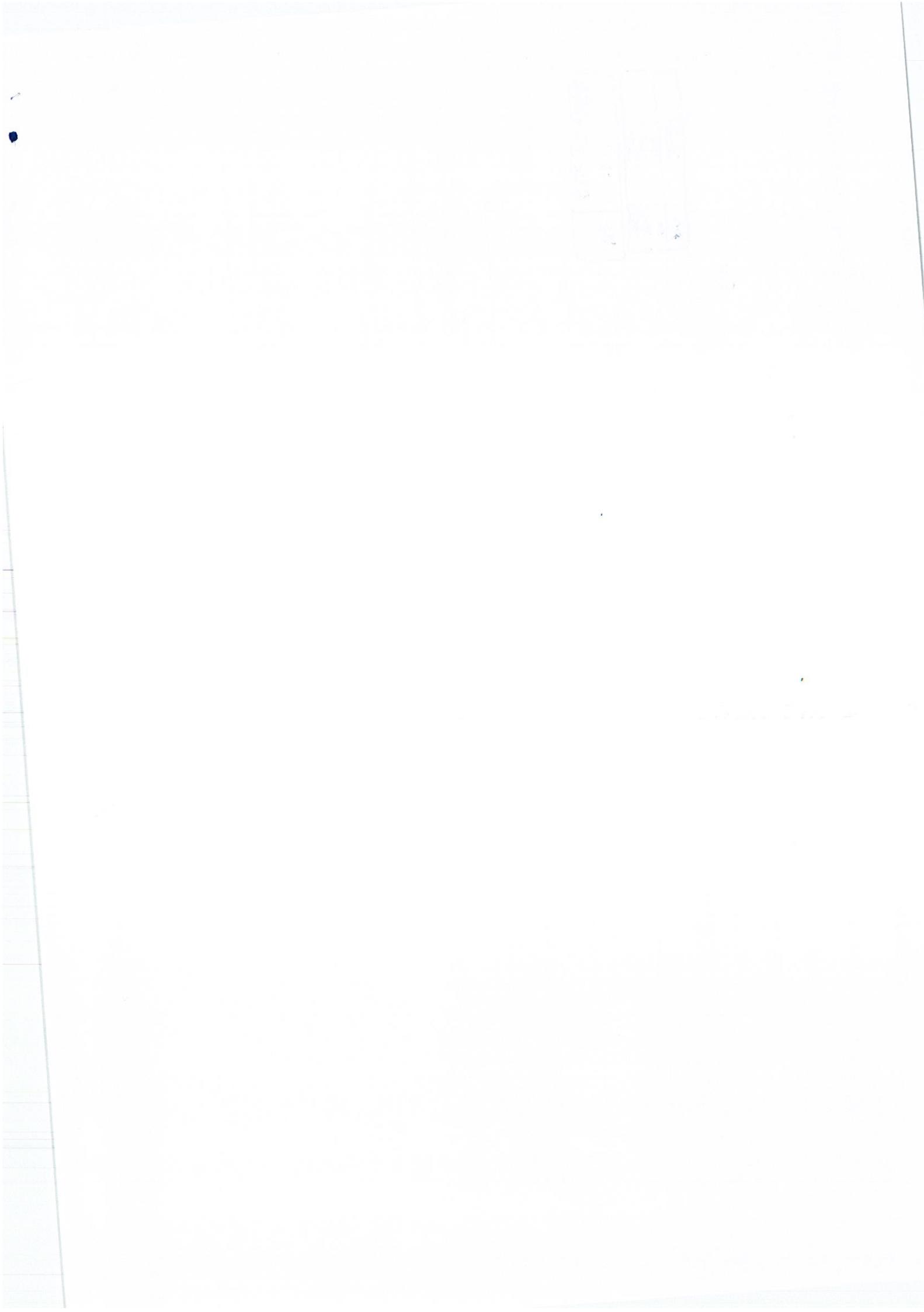
IK3 Hau...
IK3 EH1

IK2 H2
IK2 EH2

IK2 EH1

ARTETAK

- 2) Ior ezazu honako baldintza betetzen duten irakasgaien koea: irakasgaien Cadizeko ikasle guztiek matrikulatu dira.
- 3) Ior ezazu informatika graduoko lehen mailako irakasgai guztiek gainditu dituzten ikasleen Na-a.
- 4) BIDAIK orrialdeko 38 eta 39.



Datu-baseak

EBAZPEN BATZUK

25.Lor itzazu egunean 100 euro baino gehiago balio duen bidaiaren batean gidari direnen NA-a eta izena.

```
SELECT G.NA, G.Izena
FROM GIDARI AS G
WHERE G.NA IN (SELECT B.NA FROM BIDAIA WHERE EgunekoPrezioa > 100)
```

JOIN erabiliz egiten bada DISTINCT jarri beharko dugu errepikatuak kentekeo

```
SELECT DISTINCT G.NA, G.Izena
FROM GIDARI NATURAL JOIN BIDAIA
WHERE EgunekoPrezioa > 100
```

26.Lor itzazu honako baldintza hau betetzen duten bezeroen NA-a eta izena: bezeroak hotelaren egoitza Durangon duen bidaiaren batean parte hartu du.

```
SELECT DISTINCT B.NA, B.Izena
FROM BEZERO AS B INNER JOIN HOTEL_BIDAIA_BEZERO AS HBB ON B.NA=HBB.NA
INNER JOIN HOTELA AS H ON HBB. HotelId = H. HotelId
WHERE H.Hiria = 'Durango',
SELECT NA, Izena
FROM BEZERO
WHERE NA IN (SELECT NA
FROM HOTEL_BIDAIA_BEZERO NATURAL JOIN HOTELA
WHERE Hiria = 'Durango') ; edo INNER JOIN HotelId attributa NA, Izena kantoko taulan daudenez badakigu BEZERO tauilarri dagozkiela - berizendatzea ez da beharrezkoa
```

27.Lor ezazu hautazko txangorik hartu ez dituzten bezeroen NA-a.

```
SELECT NA
FROM BEZERO
WHERE NA NOT IN (SELECT NA FROM HAUTAZKO_TXANGO_BEZERO)
```

29.Lor ezazu honako baldintza hau betetzen duten bidaien kodea: 2011ko uztailaren 7an irufetik ateratako bidaiaren helmuga bera eta egun kopuru berdina izatea.

```
SELECT Helmuga, IrteeraData
FROM BIDAIA
WHERE Helmuga, Egunak) IN
(SELECT Helmuga, Egunak FROM BIDAIA WHERE IrteeraHiria = 'Iruña'
AND IrteeraData = '2011-7-7')
```

Datu-baseak

31.Lor ezazu Erronkara egindako bidaia guztien eguneko prezioa gainditzen duten bidaien kodeak.

```
SELECT Helmuga, IrteeraData
FROM BIDAIA
WHERE EgunekoPrezioa > ALL (SELECT EgunekoPrezioa FROM BIDAIA WHERE Helmuga = 'Erronka')
```

32.Lor ezazu Erronkara egindako bidaiaren baten eguneko prezioa gainditzen duten bidaien kodesa.

```
SELECT Helmuga, IrteeraData
FROM BIDAIA
WHERE EgunekoPrezioa > ANY (SELECT EgunekoPrezioa FROM BIDAIA WHERE Helmuga = 'Erronka')
```

33.Lor ezazu bidaia garestienaren kodea.

```
SELECT Helmuga, IrteeraData
FROM BIDAIA
WHERE EgunekoPrezioa*Egunak > ALL (SELECT EgunekoPrezioa*Egunak FROM BIDAIA)
SELECT Helmuga, IrteeraData
WHERE Egunak*EgunekoPrezioa = (SELECT MAX(Egunak*EgunekoPrezioa) FROM BIDAIA)
```

35.Lor ezazu honako baldintza hau betetzen duten bezeroen NA-a: bezeroak ez zuen parte hartu 2011ko urtarrilaren 1ean Asilah helmuga gisa zeukan bidaian.

```
SELECT NA
FROM BEZEROA
WHERE NA <> ALL (SELECT NA FROM HOTEL_BIDAIA_BEZERO WHERE Helmuga = 'Asilah'
<> ALL IrteeraData = '2011-1-1')
<> ALL jarri ordez NOT IN jarizeko aukera dago
```



2. Andar

Select Node

From Target node As If

where Not Exists (Select Node)

From IKeSle As If

where IsEmpty Like ('%', Cdi, '%')

AM

Not Exists (Select *

From Matriule AS H

where ILU.Node = H.Node AND

Es decir ~~que el resultado de la consulta es diferente~~ (Cdi, Node) ~~que el resultado de la consulta es diferente~~

ILU.Node = (I.Node)

where ILU.Node = H.Node AND

ILU.Node ~~que el resultado de la consulta es diferente~~

Select NA

From IKeSle As If

where Not Exists (Select *

From Target AS If

where ~~que el resultado de la consulta es diferente~~ AND AM = F

Not Exists (Select Node)

From Matriule AS H

where H.Node = LN.Node AND NM.Node

Note 25

Introducción a la consulta SQL es doblemente igualmente igualmente es doblemente

3. Idatzi ebazpen zuenear dagoekien SQL sententzia.					
Hardware	Izeneak	Izenak	Izenak	Izenak	Izenak
H1	D1	81	Izenak1	Izenak1	NA1
H2	D2	82	Izenak2	Izenak2	NA2
H3	D3	83	Izenak3	Izenak3	NA3
H4	D4	84	Izenak4	Izenak4	NA4
WA3					
WA4					
WA5					
WA6					

SELECT HT.* , NA , Izena
 FROM HAUTAZKO_TXANGO AS HT LEFT JOIN HAUTAZKO_TXANGO_BEZERD
 ON HT.Helmuaga=HTB.Zenbakia NATURAL JOIN BEZERO
 AND HT.Irteeradat=HTB.Irteeradat
 AND HT.Zenbakia=HTB.Zenbakia NATURAL JOIN BEZERO

2. Ze buletzuen du honako proposamen honenk? Irudikatu graffiko!

2. Ze buletzuen du honako proposamen honenk? Irudikatu graffiko!					
Hardware	Izeneak	Izenak	Izenak	Izenak	Izenak
H1	D1	81	Izenak1	Izenak1	NA1
H2	D2	82	Izenak2	Izenak2	NA2
H3	D3	83	Izenak3	Izenak3	NA3
H4	D4	84	Izenak4	Izenak4	NA4
WA3					
WA4					
WA5					
WA6					

1. Ze emaitza buletatu beharko luke 46. arriketak? Irudikatu graffiko!

HAUTAZKO_TXANGO_BEZERD			
Helmuaga	Irteeradat	Zenbakia	NA
H1	D1	Z1	NA1
H2	D2	Z2	NA2
H3	D3	Z3	NA3
H4	D4	Z4	NA4

HAUTAZKO_TXANGO_BEZERD			
Helmuaga	Irteeradat	Zenbakia	NA
H1	D1	Z1	NA1
H2	D2	Z2	NA2
H3	D3	Z3	NA3
H4	D4	Z4	NA4

HAUTAZKO_TXANGO_BEZERD		
Helmuaga	Irteeradat	Zenbakia
H2	D4	Z2
H2	D3	Z1
H1	D1	Z1
H1	D1	Z1

Loritzazu hautazko txangoen datu guztia, eta, txangoko horiek bezeroek aukeratutak izan badira, loritzazu, halaber, bezeroen NA-a eta izena (ez badira aukeratutak izan, bezeroen bi zutabe horietan NULL aterra beharko litzateke).

46. arriketa

38. Lortitzu honako baldintza betetzen duten bezeroen NA-a eta izena: bezeroak hautazko txangoak hartu ditu hotelaren egoitza Parisen duten bere bidaia guzilean. Baldintza beste era batera esanda: bezeroak hautazko bangoak haritu ditu bere bidaia guzilean, zeinetan hotelaren egoitza Parisen batitago.

Horela eginda gizki ateratzen da ez baitugu ziurratzen bezero hori Pariseko hotel batean egon, baina bideziaz eta ez badu bidaiarik egin ere) ateratzen da.

```

SELECT B.NA, B.IRENA
From BEZEROA AS B
WHERE NOT EXIST (SELECT *

```

FROM HOTEL_BIDATA_BEER0 AS HB NATURAL JOIN HOTELA
WHERE Hira = "Paris" AND B.NA=HB.NA
AND NOT EXISTS (SELECT *
FROM HAUTAZKO TYANCO WHERE ...)

WHERE B.NA=HTB.NA AND HBB.Helmuga=HTB.Helmuga
AND HBB.IrteeraData=HTB.IrteeraData))

Goikoa konpontzeko baldintza berri bat

```
SELECT B.NR, B.Izena  
      FROM BEZEROA AS B  
 WHERE NOT EXIST (SELECT *  
                   FROM HOTEL_BIDATA_BEZERO AS HBB NAT  
                   WHERE HBB.Hiria = "Paris" AND B.NA=HBB.  
                   AND NOT HBB.NA=
```

```

AND NO. EXIST (SELECT * FROM HAUTAZKO_TXANGO_BEZERO AS HTB WHERE B.NA=B.NA AND HBB.Helmutga=HTB.Helm AND HBB.IrteeraData=HTB.IrteeraData))
AND EXIST (SELECT * FROM HOTEL_BIDATA_BEZERO AS HBD_NATURAL JOIN HOTELA WHERE Hilir_ = "Paris" AND B.NF=HBD.NA)

```

Beste aukera batzuk:

```

FROM BEZERO AS B
WHERE NOT EXISTS (SELECT *
                   FROM HOTEL_BIDAI_BEZERO AS HBB_NATURAL JOIN HOTELA
                   WHERE Hiria = "Paris" AND B.NA=HBB.NA
                   AND B.NA NOT IN (SELECT HTB.NA
                                     FROM HAUZAIZO_TXANGO_BEZERO AS HTB
                                     WHERE HTB.Haimuga=HTB.Haimuga
                                           AND HBB.Irteeradata=HTB.Irteeradata))
AND EXISTS (SELECT *
            FROM HOTEL_BIDAI_BEZERO AS HBB_NATURAL JOIN HOTELA
            WHERE Hiria = "Paris" AND B.NA=HBB.NA

```

```

ELECT B.NA, B.Izena
  FROM BEZERO AS B
 WHERE B.NA NOT IN (SELECT HOTEL_BIDALIA_BEZERO
                      FROM HOTEL_VIAJE_CLIENTE AS HBB_NATURAL JOIN HOTELA
                      WHERE Hirlia = "Paris"
                            AND B.NA NOT IN (SELECT HTB.NA
                                              FROM HAUTAZKO_ITXANGO_BEZERO AS HTB
                                              WHERE HBB.HelMutiga=HTB.HelMutuga
                                    AND HBB.IrteeraData=HTB.IrteeraData))
AND EXIST (SELECT *
  FROM HOTEL_BIDALIA_BEZERO AS HBB_NATURAL JOIN HOTELA

```

WHERE Hiria = "Paris" AND B.NA=HBB.NA)

Aukera guztiak probatuta daude.

39. Lor itzazu honako baldintza hau betetzen duten bezeroen NA-a eta izena: bezeroak ez du inolako hautazko txangonik hartu hotelaren egoitza Parisen duen bere bidaiaren batean (geria lileke, noski, bezero horrek hautazko txangoren bat hartuera izatea hotelaren egoitza lako hautazko bestetik bidaia batean. Baldintza beste era batera esanda: bezeroak ez du ino- batitago.

```

SELECT B.NA, B.Izena
  From BEZERO AS B
 WHERE EXITS ((SELECT *
                  FROM HOTEL_BIDAIAREN_BEZERO AS HBB_NATURAL JOIN HOTELA
                WHERE Hiria = "Paris" AND B.NA=HBB.NA
          AND NOT EXISTS (SELECT *
                            FROM HAUTAZKO_TXANCO_BEZERO AS HTB
                           WHERE B.NA=HTB.NA AND HBB.Helmuiga=HTB.Helmuiga AND
HBB.Irteeradata=HTB.Irteeradata))

```

THE JOURNAL OF CLIMATE

Beste Batzule

49. Arikete

Select Helmuge, IrteeraDate

From Bidaiak

Where Precia * Egundea.Precio < ALL (Select Precia * Egundea.Precio
From Bidaiak)

50. Arikete

Select HT.Helmuge, HT.IrteeraDate, HT.Zurbelakia

From Hautealde-Txango-Berero

Where Count(NA) > ALL (Select Count(NA)

From Hautealde-Txango-Berero As HT2

Where HT1.Helmuge = HT2.Helmuge AND

HT1.IrteeraDate = HT2.IrteeraDate AND

HT1.Zurbelakia = HT2.Zurbelakia)

51. Arikete

Select B.NA, Idena

From Berero As B INNER JOIN Hautealde-Txango-Berero As HTB ON
B.NA = HTB.NA

Where 3000 < ALL (Select Precia)

From Hautealde-Txango As HT

Where HT.Helmuge = HTB.Helmuge AND

HT.IrteeraDate = HTB.IrteeraDate AND

HT.Zurbelakia = HTB.Zurbelakia)

52. Arikete

Select NA, Idena

54. Ariketa

Select NA, Idem

From Gideri

Except

Select NA

From Bideria

where Helmuge like 'Errone'

55. Ariketa

Select Helmuge, Idem, Gideria

From Haukalla-Tango ^{Natural JOIN} ~~INNER JOIN~~ Haukalla-Tango-Bideria

ON where NA IS NULL

56. Ariketa

Select NA, Idem

From Gideri INNER JOIN Haukalla ON Gideri.NA = Haukalla.NA

where Haukalla like 'Bideria', NA NOT IN (Select NA

From Bideria

where Helmuge like

EBAZPEN BATZUK

3. Lor ezazu honako baldintza betetzen duten bidaien kodea: Donostiatik atera eta Madril helmuga gisa eduki, eta Toledo hirira hautazko txangoren bat eskaintzea.

```
SELECT B.Helmuga, B.IrteeraData  
FROM BIDAIA AS B NATURAL JOIN HAUTAZKO_TXANGO AS HT  
WHERE B.IrteeraHiria = 'Donostia' AND B.Helmuga = 'Madril' AND HT.TxangoHelmuga= 'Tol'
```

5. Lor itzazu honako baldintza hau betetzen duten bidaien identifikazioa, eta bezeroen NA-a eta izena: bezeroaren izenak eta egindako bidaien gidariaren izenak berdinak izan behar dute (adibidez, Mikel-Mikel).

```
SELECT DISTINCT B.Helmuga, B.IrteeraData, BEZ.NA, BEZ.Izena  
FROM (GIDARI AS G NATURAL JOIN BIDAIA AS B) INNER JOIN (BEZERO AS BEZ NATURAL JOIN  
HOTEL_BIDAIA_BEZERO AS HBB)  
ON B.Helmuga=HBB.Helmuga AND B.IrteeraData=HBB.IrteeraData WHERE BEZ.Izena= G.Izena  
*DISTINTC: bezero bat bidaia berdinean hotel desberdinietan egoteko aukera dagoenez,  
bezero bat gidari berarekin behin baino gehiagotan agertu baitaiteke
```

6. Lortu 2011-05-03 datan Errromara egindako bidaian NA=15238432 daukan bezeroak Gabriella izeneko hotelean igaro zituen gauak baino gau gehiago hotel berean (bidaia berean edo besteren batean, eta ez nahitaez Gabriella hotelean) egondako bezeroen izena eta helbidea.

```
SELECT DISTINCT B.Izena, B.Helbidea  
FROM (HOTEL_BIDAIA_BEZERO AS HBB1 NATURAL JOIN BEZERO AS B)  
INNER JOIN  
(HOTEL_BIDAIA_BEZERO AS HBB2 NATURAL JOIN HOTEL AS H)  
ON HBB1.GauKopurua > HBB2.GauKopurua  
WHERE HBB2.NA="15238432" AND HBB2.IrteeraData=11/05/03 AND HBB2.Helmuga="Errroma" AND  
H.Izena="Gabriella"
```

7. Lor itzazu gidaririk gabeko bidaien kodeak.

```
SELECT Helmuga, IrteeraData  
FROM BIDAIA  
WHERE NA IS NULL
```

11. Lortu gidarien eta bezeroen izenak.

```
SELECT Izena  
FROM GIDARI  
UNION  
SELECT Izena  
FROM BEZERO
```

13. Lor ezazu gidarien izenak, bezero baten izenarekin bat ez datozenak.

```
SELECT Izena  
FROM GIDARI  
EXCEPT  
SELECT Izena  
FROM BEZERO
```

14. Lor ezazu Donostiatik ateratzen diren bidaien batez besteko prezioa.

```
SELECT AVG (Egunak*EgunekoPrezioa)  
FROM BIDAIA  
WHERE IrteeraHiria = 'Donostia'
```

16. Lor ezazu Pariseraren bidaia bat egin duten gidari desberdinien kopurua.

```
SELECT COUNT (DISTINCT NA)  
FROM BIDIA  
WHERE Helmuga = 'Paris'
```

17. Gidari bakitzeko kalkula ezazu zenbat hizkuntza menperatzen dituen.

```
SELECT NA, COUNT(*) (edo COUNT(Hizkuntza)- baina beti ez da berdina * erabili edo ez  
FROM HIZKUNTZA  
GROUP BY NA
```

18. Helmuga bakitzeko kalkulu zenbat gidari dagoen, italieraz hitz egiteko gaitasuna izanik, helmuga horretara bidaia bat gidatuko duena, baldin eta bidaia horretan Erromako hotelen batean ostatu hartzeko aukera badago.

```
SELECT B.Helmuga,COUNT(DISTINCT B.NA)  
FROM (BIDAIA AS B NATURAL JOIN HIZKUNTZA AS HIZ) NATURAL JOIN  
(HOTEL_BIDAIA AS HB NATURAL JOIN HOTELA AS H)  
WHERE HIZ.Hizk = "Italiera" AND H.Hiria = "Errroma"  
GROUP BY B.Helmuga
```



Emprasse SQL Konsultant

a)

Select, Izena

From Langilee inner join FamiliaTelea on GSZ = LGZ and Izence Familiaz

b)

Select L.Izena, L.Abizen

From Langilee Linner join Langile G

Where L.Gainbegsz = G.GSZ and G.Izene = "Federico" and G.ABIZEN = "Vizcarra"

c)

Select Izene, Abizen

From Langilee Linner join Proiektua Langiteen_du, PZ = Proiektua_Pentzia ~~Eduker~~ Join
Langilee ON LGZ = GSZ
Where

PZene = X Proiektua_Ald Adule > 10 AND Szintza = 5

e) Select GSZ
From Langilee

* Except

Select LGZ AS GSZ

From ~~Langilee~~ Langiteen_du

~~Where GSZ != LGZ~~

d) Select Projektive_Pflichten_Sum (Ordnung)
From Projektive INNER JOIN LangListe ON Pflichten = Pf
Where
Group BY Pflichten

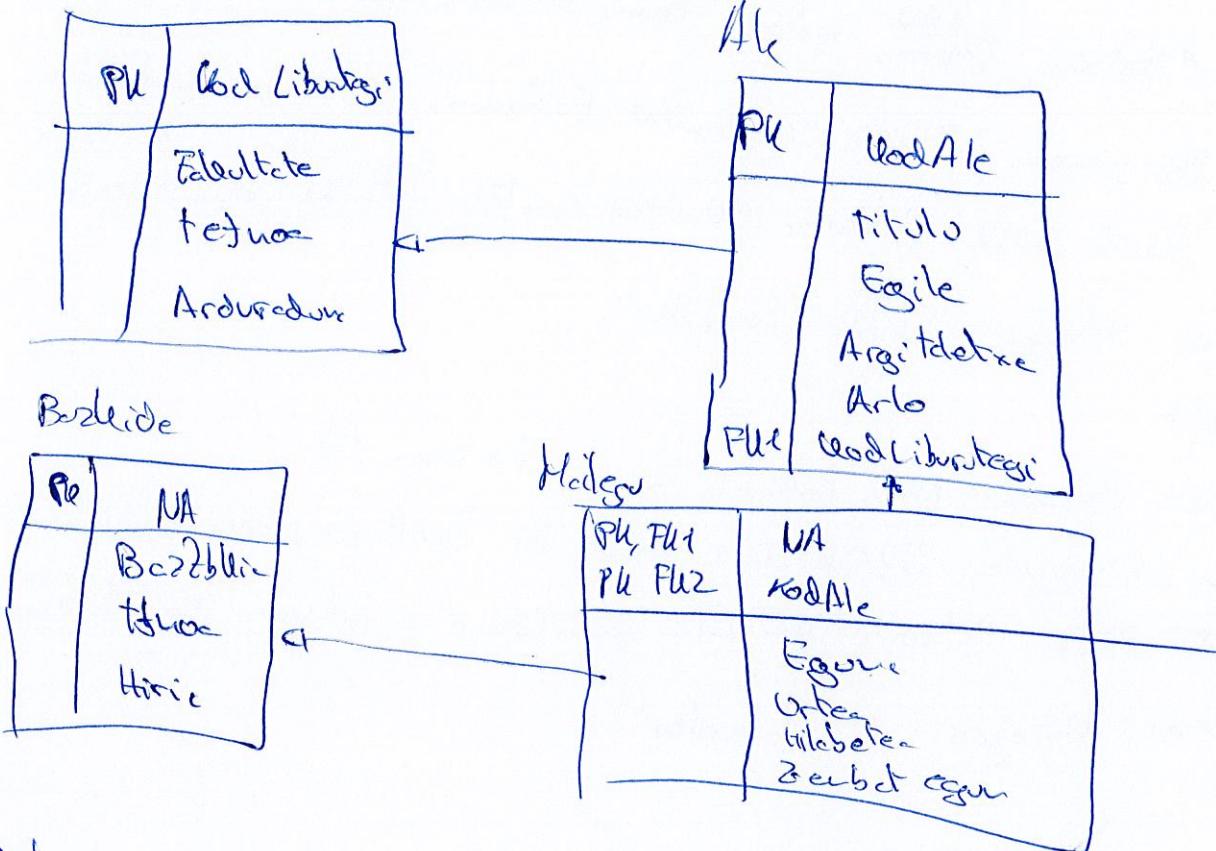
e) Select STZene AVG(Soldate)
From STZene INNER JOIN LangListe ON STZ = STZene
Group BY STZene

f) Select AVG(Soldate)
From LangListe
Where Sexu = "Enderner"

g) Select Person_Abisen
From LangListe INNER JOIN Seile ON Buru GSZ = GSZ
Except
Select Person_Abisen
From Familienliste

h) Select STZene, Count(*)
From LangListe INNER JOIN Seile ON Seibkunz = STZ
Where Sexu = "Gitarre/Bass"
Group BY STZene

Liburtegi



Bestelde berberale
araloak liburu bilatu tituluak
maileguen leku dituen informazioa

Select A1.titulo, A2.titulo

From Mailegu
~~(Bestelde INNER JOIN Mailegu ON Bestelde.NA = Mailegu.NA)~~
 INNER JOIN Ale AS A1 ~~INNER JOIN Ale AS A2 ON Ale.Kod.Ale = A1.Kod.Ale~~
 ON A1.titulo <> A2.titulo AND ~~Ale AS A2 INNER JOIN Mailegu AS M2 ON M2.NA = A2.titulo~~
 Where Bestelde.NA = "X" AND A1.Arlo Like "Info" AND A2.Arlo Like "Info"

Select A1.titulo, A2.titulo

From Mailegu AS M1
 INNER JOIN Ale AS A1 ON Mailegu.Kod.Ale = A1.Kod.Ale
 INNER JOIN (Mailegu AS M2) INNER JOIN Ale AS A2 ON Mailegu.Kod.Ale = A2.Kod.Ale
 ON A1.titulo <> A2.titulo AND M1.NA = M2.NA
 Where Mailegu.NA = "X" AND A1.Arlo Like "Info" AND A2.Arlo Like "Info"

a)

Select * FROM Person, NAN, Person_Izen, Person_Adina
FROM Hirigunea, Etxe_Per, Etxe_Zonaketa =
~~Etxe... Zonaketa~~
~~INNER JOIN Person ON Etxe_Zonaketa = Person_Zonaketa~~
~~INNER JOIN Person ON Etxe_Zonaketa = Person_Zonaketa~~
Where Utegoria Like "Luxuzko"

Except

Select Person_NAN, Person_Izen, Person_Adina
From Hirigunea, INNER JOIN Etxe_Per ON H.Zonaketa = G.Zonaketa,
INNER JOIN Person ON Etxe_Zonaketa = Person_Zonaketa
Where Utegoria != "Luxuzko"

c)

Select Pisu_Idaria, Pisu_Zerbitzua

Case M2

When ≤ 50 then 'Pisu txikia'

When $50 < 75$ then 'Pisu ordua'

When $75 >$ then 'Pisu handia'

(Pisu INNER JOIN Etzematxo ON P.Idaria = E.Idaria AND E.Zerbitzua = Zerbitzua)
From Pisu INNER JOIN Person ON Zerbitzua = Zerbitzua_P
Where Zerbitzua < 10 AND Pe_Adina > 60

18. Arikete

Select count(HotelId) AS count(HotelId, NA)

From Hotel INNER JOIN Hotel-Bidic ON HotelId = HotelId AND HotelId = HotelId

Where Hotel.HotelId = Hotel-Bidic.HotelId AND Hotel.HotelId = Hotel-Bidic.HotelId

Group By Hotel.HotelId

Taximundulerien studio

a)

Select Taximundulerien.Dortsde, Taximundulerien.tzne, Taximundulerien.tzne, Taximundulerien.tzne, Sum(Denbore)

From Taximundulerien Natural JOIN Portekartzen.d

Where Taximundulerien.Lesterhete.Pego = true

Group by Taximundulerien.Dortsde, Taximundulerien.tzne, Taximundulerien.tzne

Ordered by Sum(Denbore) ASC

b)

T2.Dortsde

Select Denbore, tzne, T2.tzne, T2.Dortsde

From Etepe INNER JOIN Taximundulerien ON Etepe.Drebdole = Taximundulerien.tzne

From Etepe INNER JOIN Taximundulerien AS T1 ON Etepe.Drebdole = T1.tzne

Etepe INNER JOIN Taximundulerien AS T2 ON Etepe.Drebdole = T2.tzne

Taximundulerien AS T2 INNER JOIN Portekartzen.d ON T2.Dortsde = Portekartzen.d

ON T2.tzne = M.tzne

Where Etepe.Zblli = 10 AND Portekartzen.d.Etepe = 10



13. Arillete

Select Tzen
From Bidze

Except

Select Tzen

From Bezero

14. Arillete

Select Avg (Price + Egndlo Price * Egndlo)

From Ga_Bidze

Where Intercity Like "Domestic"

16. Arillete

Select Count(Distinct NA)

From Bidze

Where Helmuge Like "Paris"

6. Arillete

Select Bezero.Tzen - Bezero.Helbide

From Hotel INNER JOIN H_B_B Hotel, HotelId = H_B_B.HotelId AS H_B_B1

INNER JOIN H_B_B2A_H_B_B2 ON H_B_B2.Gruppe > H_B_B1.Gruppe

INNER JOIN Bezero ON H_B_B2.NA = Bezero.NA

Where Intercity = 2011-05-03 AND H_B_B2.Helmuge Like "Paris",

AND Hotelc.Tzen Like "Gabriel" H_B_B1.NA Like "15238432"

17. Arillete

Select NA Count(Hizlunza)

From Hizlunza

Group By NA

SQL arrilera

3. Arrilera

Select Bidaiac.Helbuge Bidaiac.InteretData
 From Bidaiac INNER JOIN Hotel-Balio-Txango ON Bidaiac.Helbuge = Hotel-Balio-Txango.Helbuge
 AND Bidaiac.HelbugeInteretData = Hotel-Balio-Txango.InteretData
 where InteretHiria = Domestic AND Helbuge = Madrid AND
 Hotel-Balio-Txango.Helbuge = Toledo

5. Arrilera

Select Bidaiac.Helbuge, Bidaiac.InteretData, Bezero.NA, Bezero.Izene
 From (Bidaia INNER JOIN Gidari ON Gidari.NA = Bidaia.NA) INNER JOIN
 Hotel_Bidaia_Bezero ON Bidaia.Helbuge = H-B-B.Helbuge AND Bidaia.InteretData =
 H-B-B.InteretData
 INNER JOIN Bezero ON Gidari.Izene = Bezero.Izene
 where Gidari.Izene = Bezero.Izene.

7. Arrilera

Select Helbuge, InteretData
 From Bidaia
 Where NA IS NULL

11. Arrilera

Select Gidari.Izene
 From Gidari
 UNION
 Select Izene
 From Bezero

3. laborategia. MySQL

Dokumentu honetan laborategia, honen helburuak eta hainbat ataza deskribatzen dira. Dokumentua laborategira sartu aurretik irakurtea gomendatzzen dizuegu.

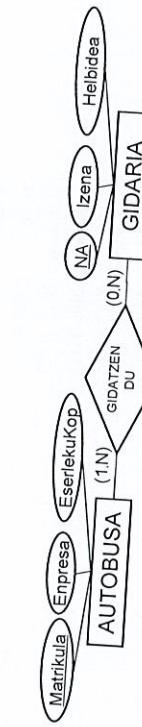
Bestela, zurek egindako instalazioan datu-base bat sortzeko, izen bat idatziko dugu "Sortu datu-base bat" gelaskan.

2. ataza: Taulak sortzea eta oinarriko gakoak definitzea.

Helburuak

- Laborategia bukatutakoan ondorengo eragiketak aurerra eraman ahal izateko MySQL zerbitzari batetik nola erabilizten den ezagutuko dugu:
- Taulen eta hauen eremuuen definizioa.
 - Taula bateko oinarriko gakoaren zehaztapena eta aldaketa.
 - Integritate murrizapenen (erreferentziala) definizioa. Murriztapenak hausten direnean (terrenkadak ezabatzeko edo aldatzen direlako), DBK'Saren porteraeraen definitzioa.
 - DBan datuak sartzea eta integritate murriztapenak betezten direla egiaztagatza.
 - SQL kontsultak egitea.
 - Datu-base batzen importazioa egitea.

E/E eskema



Eredua erlazionatua

AUTOBUS	Matrikula	Empresa	Eserlekukop
GIDARIA	NA	Izena	Helbidea
GIDATZEN_DU	Matrikula_Aut	NA_Gid	

0. ataza. Zerbitzarria konektatzeara.

Lanean hasteko, zerbitzarria konektatu behar dugu. Horretarako, konekta zaitez <https://2004601.giehme.es:12322/zerbitzarria>. Ondoren, irakasleak adicrazi dizun erabiltailearen beharko dituzu.

1. ataza. DB bat sortzea.

Oharra: Laborategi honekarako erabiliko duzuen datu-base eskema dagoeneko sortuta daukazue. Zuen erabiltaile-izena dararam datu-base huts bat daukazue eta bertan klikatu baino ez dizue egin beharko sortzen hasteko. Hau segurtasun-arazoak ekiditeko egin da.

¹ MySQLn gako arrozkak era hauen gaineko murrizapenak erabiltzeko. Beharrezkoak da InnoDB motorea erabilizte: Beste edozin motoreek ez du gakoak ongi erabiliko direnik bermatzenten (sistema instalatzenten denean aukeratzen da InnoDB motorea erabiliztea).

2.1 Taulak sortzea

DBan, eredu erlazionalari dagozkieta taulak sor itzazu.

Atributo bakoitzari eskuera dituzun datu-motetikatik egokiena iruditzen zaizuna esle iezaozu. Datu-mota batzuek zehaztzen handia behar izaten du. Adibidez, attributua VARCHAR motakoa aukeratuz gero, karaktere-katearen luzera maxima zehaztu beharko dugu.

Taulako attributo guztiak definitu ondoren, oinarriko gakoak definitu behar duzu. Azter ezazu nola definitzen den oinarriko gakoak, *unique* (gako hautagai) ezaugarria e.a.

Gogora ezazu gako arrozen porteraera zuzena izan dadin, taulak **InnoDB** motorearekin definitu behar direla.

2.2 Taulak oldutzeara

Entitate/erlazio creduan, GIDATZEN_DU taulan "noiztik" attributua definitzea ahaztu zait. Gehi ezazu attributu hau dagozkieta taulan.

2.3 Taulak etxartzeara

-DBan PROBA_TAULA izeneko taula sor ezazu nahi dituzun attributuekin.
-Taula hori ez da beharrezko eta ezabatu nahi dugu. DBan taula hori ezabatzeko ekintza identifika ezazu eta, ondoren, taula ezaba ezazu.

3. ataza. Integratze erreferentzialeko (IE) erregeletak definitzea

Integritate erreferentzialak existitzen ez diren tuplei erreferentzialik egindo ez disgula bermatzenten. IEa, gako arroza eta erreferentzialatzen duen oinarriko gakoaren artean definitzen da. Gako porteraera ere definitzen da.

3.1 Taulak InnoDB motorea erabilizten dutela egiazatzea

Lehenago esan dugu, gako arrozketin lan egiteko eta integratze erreferentzialeko murriztapenak definitzeko, tauliek InnoDB motorea erabiliztea beharrezko dela. Egiaztatu ezazu definitutako taula guztiak motore hau dutela, horretarako, taula bakoitzeko MOTA zutabean InnoDB motoreak azaldutu behar du, eta ez, MyISAM motoreak (sisteman hau esleitzen du).

Taularen batean motorea aldaizteko taula horretan sartu behar dugu. Taula bateko taula irrikita duzula, menuko ERAGIKETAK botoia sakatu behar dugu. Taulako zehazten diren AJOAN botoia. Aldaketa gantzki definitutako taula bakoitzean egin beharko duzu.

3.2 Integratze erreferentzialeko murriztapenen definitzia

Defini itzazu beharrezkoak diren integratze erreferentzialaren murrizpena definitziko. ERLAZIOEN IKUSPEGIA botoia sakatu behar da eta gako attributuekin erlazionalen aukeratzen denean aukeratzen da. Gako arroza eta oinarriko gako attributuak dagozkin taulako oinarrizko direnean, aurerra eman beharreko ekintza definitu da. Gure kastan, eta erabakia egin dugun

aukera bat dela jakinda, ezabaketaren eta eguneraketen hedapena (cascade) aukeratuko ditugu.

4. ataza. *Datu-basearen sortkuntza egiazatea, SQL kodea azterutz. LANA igo.*

Sorutako datu-baseari dagokion SQL kodea interfazearen bidez ikusi ahal izateko, datu-basea SQL

koderia exportatu behar da. Egin izazu hau, eta egiaza ezazu DBa ongi sortu dela taula bakoitzak bere oinarritzko gakoa definitua duela, gidartak autobusekin erlazionatzen dituen taulak gako arrozak definituta dituela...

Gorde ezazu sortutako SQL kodea fitxategi bat sortuko da. Fitxategi hau egela-ra hautauz gero, ordenadorera jaitsi ahal izango duzu fitxategi bat sortutako (Abizena1.sql).

5. ataza. *Integritate erregelen definiziok diuen ondorioak egiazatea.*

Teklautik honakoak egiazatu ahal izateko datuak sat izazu (zuk asmatu behar diuzu datuak):

5.1. Gakoaren integritate murriztapena.

5.2. Entitate integritatea manutenen dela egiazatea (kontuan izan datuak interfaze grafikotik sartzen direnean, MySQLk "null" karaktere-katea esteitez diela VARCHAR motako cremuci, eta ez null balioa. Zenbakizko erremuei 0 esteitez die hauek ingurune grafikoan hutisk uzten badira. Benetan ez badio gu inolako baliorik esteitu nahi cremuri, interfaze grafikotik sortzen den SQL galdera aldatu egin behar dugu). Ez da beharrezkoa hau egitea.

5.3. Integritate erreferentziala betetzen dela egiazatea. Ezabatzek hedatzen direla egiazatu behar duzu.

6. ataza. *Datu-baseak importatzea eta datu-baseak ezabatza.*

Ataza honenkin hasi aurretik ziurtatu irakasleari bidali beharreko fitxategia dagoeneko bidali duzula. Ordutan, ezaba itzazu zure datu-baseko taulak banan-banan hutsik geratu arte (**kontuz!**, ez ezabatu datu-basea bera).

Egela-tik jaitsitako “zip” dokumentuan “importatu_bidaia.sql” fitxategia topatuko duzu bidaia datu-basearen SQL egiturarekin. Importa ezazu datu-base hau MySQL-tik. Honetarako, fitxategia importatu behar duzu. Nahikoan da fitxategia aukeratu eta *Juan* sakatzearekin.

7. ataza. *SQL kontsultak MySQLn. LANA igo.*

Egin izazu honako SQL kontsultak importatu berri duzun datu-basean. Gehi izazu egindako konsultak entregatu beharreko 2. fitxategian (Abizena2.txt).

7.1 Lor izazu honako baldintza betetzen duten bezeroen NA-a, izena eta helbidea: bezeroak bidaiaaren batean parte hartu du non bidaiaaren egoitza Donostia hiriko Amara izeneko hotelean zen (IN eragileta erabilta)

Emaitzia: Furia avenida8 (arketa orriko 26. ariketaren parekoa da)

7.2 Lor izazu honako baldintza betetzen duten bezeroen NA-a, izena eta helbidea: bezeroak parte hartu Donostia hiriko Amara izeneko hotelean egoitza izan duten bidaietan (IN eragileaz baliatu)

Emaitzia: Ez du ateria behar 10000009 bezeroa

7.3 Lor izazu honako baldintza betetzen duten “corona” hiriko hotelen identifikazioa eta izena: hotelaren edukiera eta “Corecatown”, hiriko “Nini” izeneko hotelaren edukiera berdinak dira (IN eragileaz baliatu edo AUTOJOIN bat eginez)

Emaitzia: h14 burn (arketa orriko 28. ariketaren parekoa da)

Emaitzia: (arketa orriko 31. ariketaren parekoa da)

Helmuga	IrteteraData
Barcelona	2015-05-04
bulevard	2020-11-11
Donostia	2015-01-01
Donostia	2015-01-02
Erosky	2001-11-11
Hernani	2015-01-03
India	2015-12-01
Madrid	2015-05-01
marte	2020-04-11
Martucos	2015-05-03
MediaMarkt	2000-11-11
Ongkong	2015-01-05
Pekin	2015-01-08
Piordjane	2015-01-06
sol	2035-11-11
Urbil	2015-06-02

7.5 Lor izazu bidaia garostenaren kodea

Emaitzia: sol 2035-11-11 (arketa orriko 33. ariketaren parekoa da)

25.Arilete

Select Gideri, NA, Izena

From Gideri INNER JOIN Bidai ON Gideri.NA = Bidai.NA
where EguinelloPrecision > 100

IN trabilba

Select NA, Izena

From Gideri IN / Select Distinct NA

where NA IN From Bidai

Select Distinct NA From Bidai where EguinelloPrecision > 100

From Bidai where EguinelloPrecision > 100

26.Arilete

Select Bedero, Izena

From Bedero

where NA IN / Select Distinct NA

From Hotel_Bidai_Bedero

where HotelId IN / Select HotelId

From Hotel

where Hiric Like Durego

27.Arilete

Select Izena

From Bedero

where NA NOT IN / Select NA

From HotelBtxango_Bedero

Estare

28.Arilete

Select Bidai, Helmuga, Bidai, InterneRate
From Bidai AS B1

Where B1.Helmuga, B1.EguinelloPuru (Select Helmuge, EguinelloPuru
From Bidai AS B2

Where B2.InterneRate = 2014/06/21 AND

B2.Helmuge, InterneHiric + 110 Days

31. Arikete

Select Bl.Helmige, Bl.IsteerDate

From Bideic AS B1

where Bl.Eguinello.PreBic > Any (Select ^{B2} Eguinello.PreBic

From Bideic AS B2

where B2.Helmige Like Errone)

32. Arikete

Select Bl. Helmige, Bl.IsteerDate

From Bideic AS B1

where Bl. Eguinello.PreBic > Any / Select B2. Eguinello.PreBic

From Bideic AS B2

where B2.Helmige Like Errone

33. Arikete

Select Bl. Helmige, Bl.IsteerDt.

From Bideic AS B1

Where Bl. Eguinello.PreBic >= All (Select B2. Eguinello.PreBic

From Bideic AS B2)

34. Arikete

Select Helmige, Isteer.Dt.

From ~~Bideic~~ Bideic AS B1

Where ~~Eguinello.PreBic~~

Eguinello.PreBic <= ALL (Select #PreBic

From HeterBb_txced AS HT1

Where Bl.Helmige = HT1.Helmige AND

Bl.Isteer_Dt = HT1.IsteerDt

35. Arikete

Select NL

From Bideic

Where H~~NA~~ NA <= ALL (Select NA

From H_A_B

Where Helmige like Asilch AND

IsteerDate like #2011/1/1#

49

Select B.Helvuga, B.IrteeraData

From Bidiee ~~ASZBZ INNER JOIN~~

Where Eguude * Eguude.Rate IN (Select

Min(Eguude * Eguude.Rate)

From Bidiee)

50

Select Helvuga, IrteeraData, Zerbikie

From

Select Helvuga, IrteeraData, Zerbikie

From Bidiee Helvuga - Txango

Where Zerbikie ASZ / Select

Helvuga, IrteeraData, Zerbikie IN (Select Helvuga

IrteeraData, Zerbikie

From Helvuga-Txango-Bidiee

Where Count(NA)

51

Select NA, Izene

From Bezero

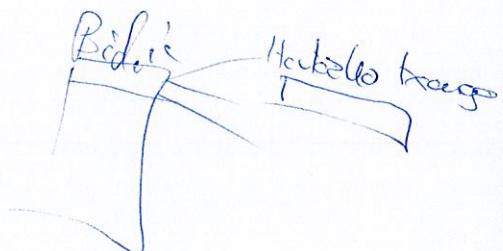
INNER JOIN (Hotel-Bidiee-Bedero ON

Bedero.NA = Hotel-Bidiee-Bedero.NA) INNER JOIN Bidiee

ON ~~Hotel~~.Bidiee.Helvuga = H-B-B.Helvuga AND B.IrteeraData =
H-B-B.IrteeraData.

Where

Egude * 3.000



LAB2

Kontsultak MS-ACCESSen

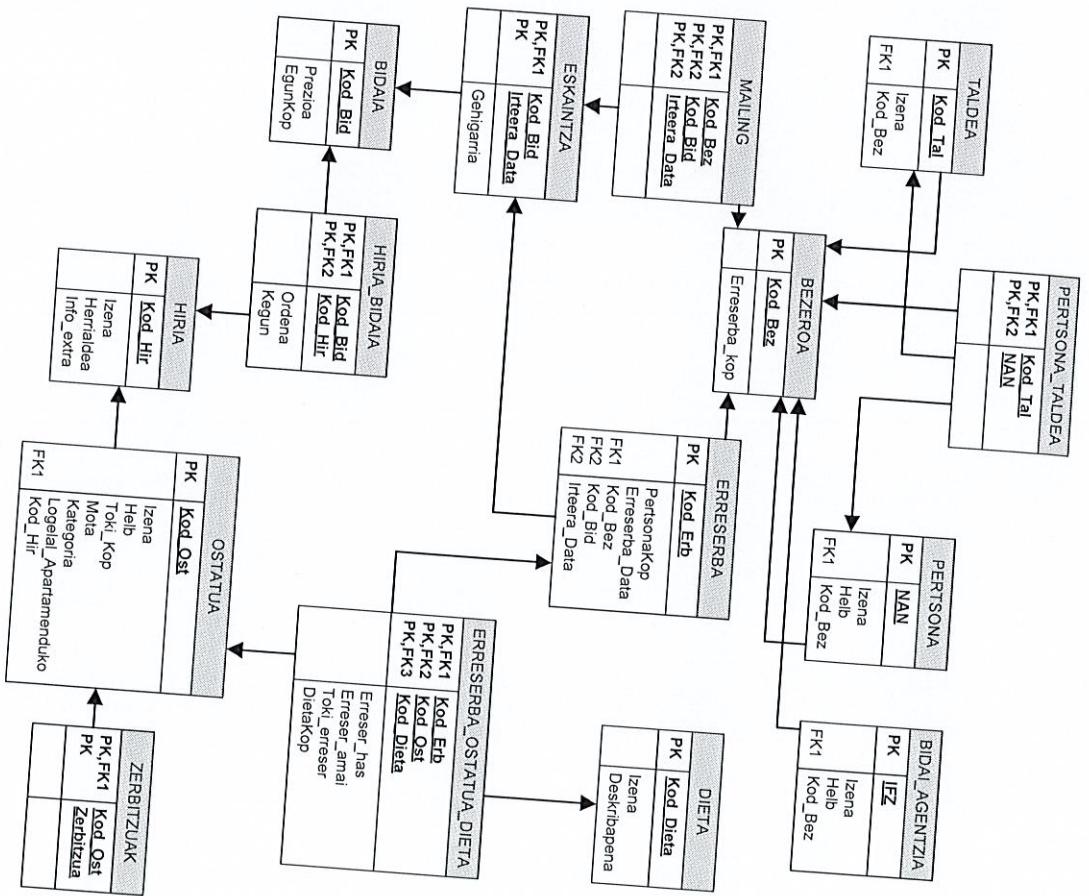
Eskema

Helburuak

Laborategi honetan egin beharrekooa ondoren zehaztuko ditugun kontsultak definitzean eta ondo eredu erlazionaleetik.

DBan gordetako taulak

Pertsona (M/M, Izena, Helb, Kod_Bez)
Taldea (Kod_Tal, Izena, Kod_Bez)
Pertsona_Taldea (M/N, Kod_Tal)
Bidai_Agentzia (IFZ, Izena, Helb, Kod_Bez)
Bezeroa (Kod_Bez, Erreserba_kop)
Eskainiza (Kod_Bid, Irteera_Data, Gehigarria)
Mailing (Kod_Bez, Kod_Bid, Irteera_dag)
Bidai (Kod_Bez, Kod_Bid, Prezia, Egunkop)
Hiria (Kod_Hir, Izena, Herriadea, Info_extra)
Ostaria (Kod_Ost, Izena, Helb, Toki_kop, Mota, Kategoria, Logelak_Apartamentuko, Kod_Hir)
Zerbitzuak (Kod_Ost, Zerbizua)
Erreserba (Kod_Erb, Pertsonakop, Erreserba_Data, Kod_Bez, Kod_Bid, Irteera_Data)
Erreserba_Ostaria_Dieta (Kod_Erb, Kod_Ost, Kod_Dieta, Erreser_has, Erreser_amai, Toki_ereser, Dietakop)
Dieta (Kod_Dieta, Izena, Deskribapena)



KONTSULTAK

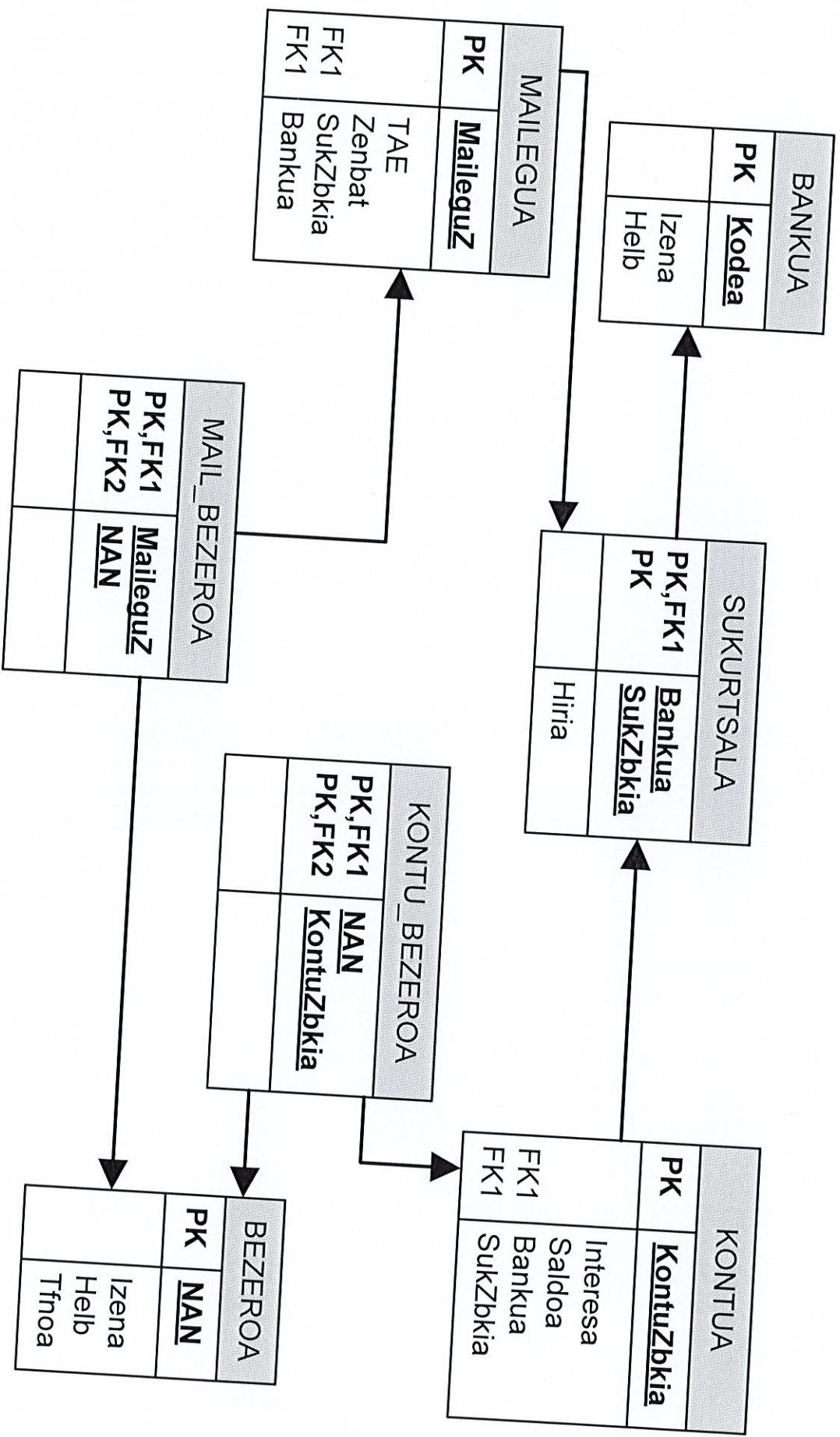
1. Lortu osatu guztiuen izena.
2. Lortu bidaietan bisitatzeten diren hiri guztiuen kodea errepikapenik gabe.
3. Lortu DBan gordeta dauden hiri guztiuen datuak izenaren arabera alfabetikoki ordenatuta.
4. Lortu 1.000 € baino gutxiago balio eta sei egun baino gehiago irauten duten bidaien zerrenda.
5. Lortu 1.000 € baino gutxiago balio eta sei egun baino gehiago irauten duten bidaien zerrenda.
6. Lortu 1998 urteko hiru aurreneko hilabetetan egin diten erreserba guztiuen datuak. ACCESSen data bat errepresentatzeko "#" karakterea jarri behar da bai hastearan bai bukaieran (#97-04-14#) kenketa edo zatiakera eragiketarekin ez nahasteko.
7. Lortu izenean hitz bat baino gehiago daukaten hirien zerrenda. ACCESSen '%' erabili beharrean '*' erabilizten da eta '_ ordez '?'.
8. Lortu irteera-daten goranzko sekuentzia jarraituz ordenatuta 1997ko udan (iztaia, abuztua, iraila) eskainti ziren bidaien datuak.
9. Bidaiia baten kode konkretu bat emanda, lortu bista ordenaren arabera ordenatuta bidaian bisitatzeten diren hirien izenak.
10. Lortu 1998ko apirileko eskaintzak eta eskaintza bakoitzaren prezio orokorra (prezioa + gehigarria).
11. Lortu emandako kodea duen bidatak baino egun gehiago dirauten bidaia guztiuen eskaintzak.
12. PERTSONA ∪ BIDAI-AGENTZIA.
13. Lortu emandako bi kodeei dagozkien taldetako pertsoneak.

Kontulta bat sortzeko ACCESSen SQL kengoarekin

1. Datu-base bat ireki
2. Kontulta berri bat sortu (*Crear + Diseño de consulta*)
3. Mostrar tabla leihoa itxi
4. Ver + Vista SQL hautatu eta leihoaan SQL sententzia idatzi
5. ↑ botia sakau sententzia exekutatzeko

Kontulta gorde nahi bada emaitzen leihoa ixteko orduan izen bat emango zaio.

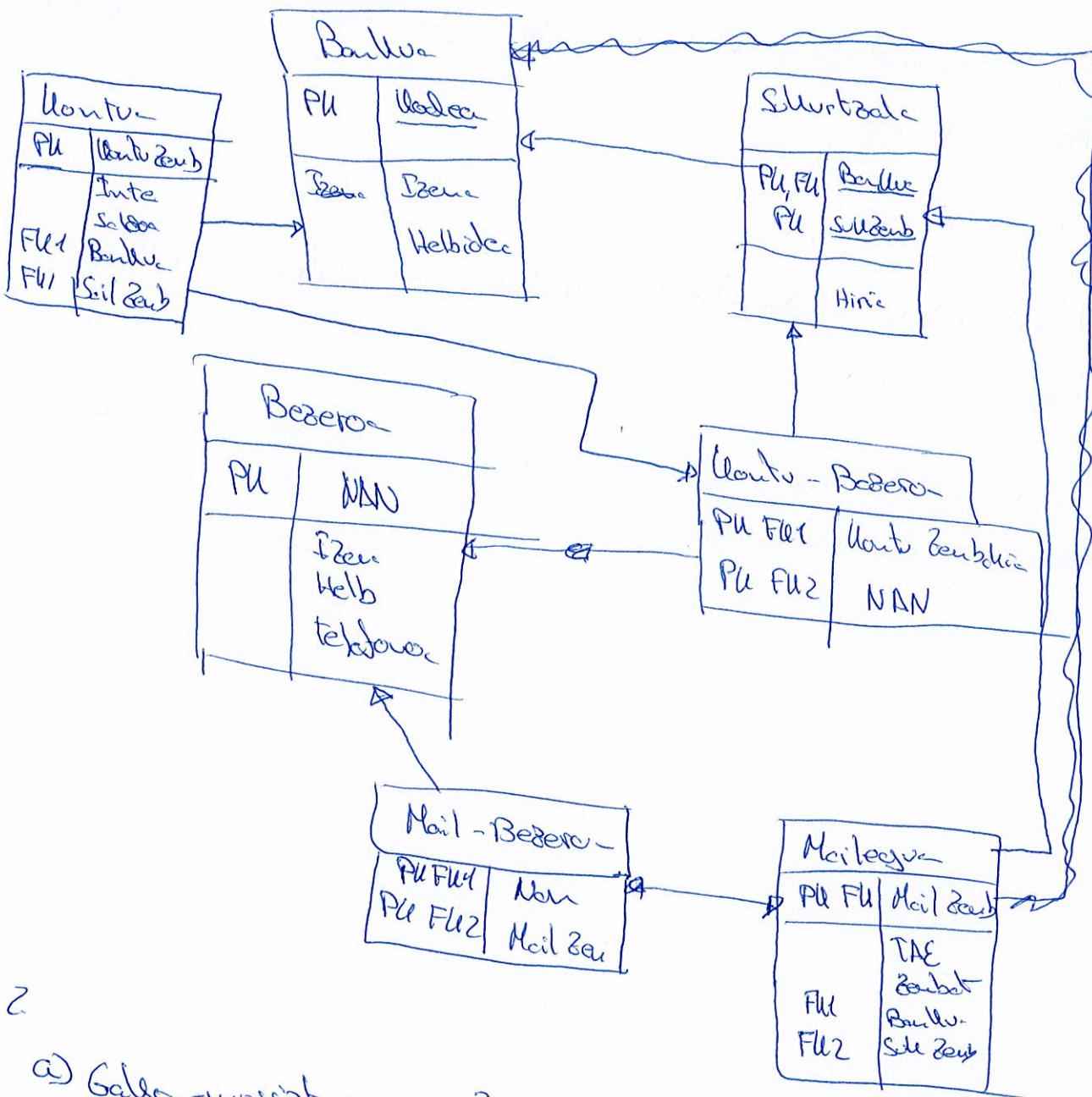
Integritate-murriztapenak – BANKETXEA (ARIKETA 2)





2. Arbeitsteilung

1.



2.

- Galler-moristik-punkt: 2 zentralisierte dezentrale Logos
(laut Zeichn. beide)
- andere ex.
- Galler-moristik Entitäten-integrat. Oftmals geben es in der Welt nicht
- Andere ex.
- Integrierte - referentielle. Nur zentralisierte beste Werte erhalten d.
- Andere ex.

- g) Arazorik ez
h) geltso morristapen. Bi bideratzea geltso oin dir errepakat.
- i) Arazorik ez
j) Arazorik ez
k) Integritate erreferentziale. Suturak ez litzelak existitzen
l) Arazorik ez.
m) Integritate erreferentziale, bestea erabazio batetako oinarrizko
geltso oin de null izan.

3. Arillets

a)

P_{R1-1} Gehänge $\leftarrow t_{P2} = 1$ And $o_{P2,1}$ (Lone-horn)

P_{R1-1} Gehänge

LGS2	P2	OND
11	1	5
22	1	3

$E_1 \leftarrow t_{LGS2}$ ($\leftarrow P_{R1-1}$ Gehänge)

LGS2
11
22

b)

$E_2 \leftarrow t_{P2}$ (Canc-horn)

P2
1
2
B

c)

P_{R0} -Guszhale $\leftarrow t_{P2}$ (PROJ)

Pro-Guszhale

P2
1
2
3
4

$E_3 \leftarrow \text{Pro-Guszhale} - E_2$

P2
4

d)

$E_4 \leftarrow \text{Lang}$ $\Delta S_2 = 3.8 \text{mic fail}$

Izene	G52	S2	Size
iller	11	1 1	LSI
Anc	22	1 1	LSI
5an	33	2 2	UAT
Khudic	44	2 2	UAT

e)

$E_5 \leftarrow t_{S2 \text{ aus Size}} (E_4)$

Izene	Size
iller	LSI
Anc	LSI
5an	UAT
Khudic	UAT

5) PERT \leftarrow Itizens GS2 (LANG)

Lane (GS2, P2) \leftarrow ItzLGS2 P2 Lane now

↓ Pert

IZone	GS2
Ilver	n
Ave	22
Jon	11, 22, 33
Uermelte	44

Lane

GS2	P2
11	1
11	2
22	1
22	3
44	3

6) E6 \leftarrow PERT * Lane

IZone	GS2	P2
Ilver	11	1
Ilver	11	2
Ave	22	1
Ave	22	3
Uermelte	44	3

7)

PRO (P2, PIZZa) \leftarrow PRO_P

PRO

P2	PIZZa
1	.net
2	xm1
3	ETB
4	Wncl

PRO - PERT \leftarrow E6 * PRO

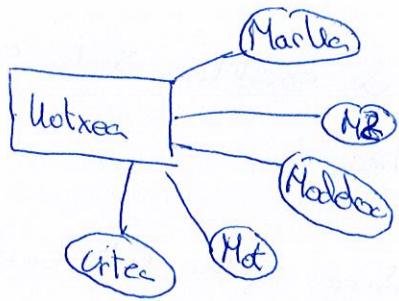
PRO - PERT

IZone	GS2	P2	PIZZa
Ilver	n	1	.net
ilver	11	2	xm1
Ave	22	1	.net
Ave	22	3	ETB
Uermelte	44	3	ETB Wncl

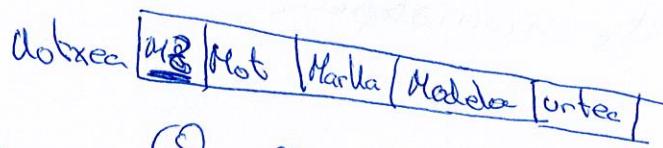
E7 \leftarrow Ilver, PIZZa (PRO - Pert)

IZone	PIZZa
Ilver	.net
ilver	xm1
Ave	.net
Ave	ETB
Uermelte	ETB Wncl

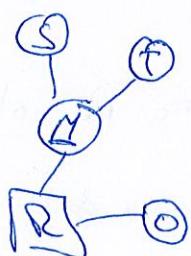
Galearen murrizapenak



M2
Mot } 2 gales hautagaia



Adibide -



Gaso lehorra, aurrikile esan behar cain denez
atxilikale ez diren beharkoak tauko s eta
jarrila ditugu. Baino ~~ba~~-ek bi osagai dituenek
bide batzuk ditugu eta galo behar esaniko.

Arriketak

1. Bai gehitu daitze
2. Ezin da gehitu, Galko arrazoa ez de existitza, Sile existitzaez batzuk. Integritate murrizpenen apurtzen da.
3. Ezin de gehitu, Dagoeneko Idegog gale bat berdin bat, Galko murrizpenen gerbatzen da.
4. Ezin da gehitu, Entitate-murrizpenen
5. Bai gehitu daitze
6. ~~Ezin da eman do integritate erreferentzia Bai al de~~
7. Ezin da, Integritate erreferentzia ~~ez do~~ apurtzen da.
8. ~~Ez do ego arazorrik Integritate erreferentziala~~
9. Ez degog arazorrik
10. ~~Ez degog arazorrik Integritate erreferentziala~~
11. Ez degog arazorrik.