

## DBak atzitzeko aplikazioak ARIKETAK

- Ikaslearen NAN zenbakia eskutu eta ikasle horrek projektuen aukeraketariak ez duela egin egiaz-tatu ondoren
- oraindik hautagai diren proiektuen zerrenda erakusten du, hots, 5 ikaslek baino gutxiagok hartu dituen proiektuen zerrenda. Projektu horiek sailaren arabera banatuta agertuko dira, indiar erakusen den moduan:

IRAKASLEAK(nan, izena, titulua, ..., bere-saila)  
 IAKASGAIAK(kodea, izena, kredituak, ..., bere-saila)  
 PROIEKTUA(pkotea izena, aurrekontua, ..., bere-saila)  
 IKASLEA(jan, izena, helbidea, ...)  
 IKAS:IRAKAS(jan, kodea, nota)

Aurreko domeinua kontuan hartuz:

- Irakasgaiaren izena eskutu eta jarraian bere saila, irakasgai hori hartu duten (bertan matrikulatura dauten) ikasleen izen-zerrrenda eta ikasle-kopurua pantailaratzenten dituen programa diseina ezazu.
- Gobunukoan sailaren izena dagoela, sail bakoltzaren irakasleak, irakasgaiak eta proiektuak erakutsiko dituen lstatua lortzen duen programa diseina ezazu.
- Hamar ikasle onenen izena lortzen duen programa diseina ezazu, hots, batezbesteko nota onena duten ikasleen izena. Ikasle batzuk matrikulatu berriak izango direla eta beraz bere batezbesteko nota null izango dela kontuan izan.

BEZEROA (nan, bizen, hiria, beherap)

SALTZAILEA (snan, sizen, hiria, portzenttua)

PRODUKTUA (kodea, pizca, hiria, kopurua, prezioa)

ESKAERA (eskenenb, hilabetea, bnan, snan, kodea, kantitatea, ordaintzeko)

Aurreko domeinua kontuan hartuz:

- Karrera amaterako proiektuen inguruko informazioa bildu nahi da. Lehen fasean ikasleek gustukoeneen dituzten hiru proiektuen izena eman beharko dute, azkenean batetik gelditzeko. Arin eta honetan lehen fase horretan zentratuko gara. Horretarako honako EER diagrama eta bere taulak kontuan hartuko ditugu:

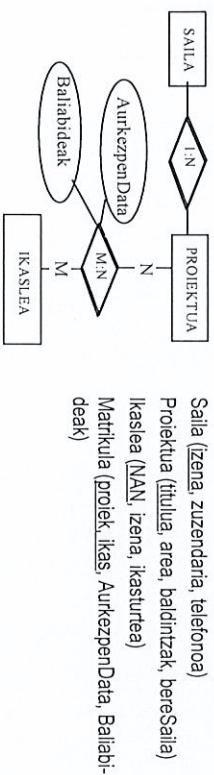


Diagrama horretatik honako taulak lortzen dira

Hiriak(Zena, Bizi-maila, InteresHistorikoa, PaisaienInteresa, HondartzakKalitatea)  
 Ostatua(Oizena, Bizi-maila, helbidea, tmoa,gela-kop, hiria, kategorkodea)

Bezeroa(nan, Bizen, helbidea, VSA-zentro)

Kategorial(kodea, prezioMax, prezioMin)

Erreserba(Oizena, nan, data)

Hiriak zembait irizpideren arabera sailkatzen dira: bizi-maila, bere historia, paisaien interesa, hondartzak kalitatea. Ezagarrirako bakoitzak 0 eta 10 balioak bat esleitzen zaio. Hiria bakoitzean ostatu desberdinak daude, baina bere kategoria. Ostatuetan bezeroak erreserba egiten dira eta bezero batek ostatu desberdinaren erreserba, erginda edukiztea geria daiteke. Sistemak bezeroaren ostatuetan erreserba egiteko aukera emango dio, honek aukeraturako hirietan dauden ostatuetan. Hirien aukera bere historiak, bizi-mailak, etabarren arabera egingo du. Sisteman pauso hauiek jarraituko dira:

- Hiriak bakoitzeko erabilizten den ezagarrirako bakoitzeko sistemak bezeroari balio bat eskatuko dira: Balio hori 0 eta 10 bitarteko izango da. Adibidez, honen antzeko pantaila bat aterako da:
- Honako ezagarririetan zure hiri ideak bete beharko lituzkeen balioak sartu, mesedez. Balioa 0 (ez du garrantziak) eta 10 (nabitaezkoa da) bitarteko izan behar da:

bizi-maila;

Historia:

Paisaien interesa:  
hondartzak kalitatea:

Honakoa egiten duen eta SQL erabilizten duen programa idatz ezazti:

```

    Honako SQL erabilizten duen programa idatz ezazti:

    CREATE TABLE IKASLEAK (
        ID INT PRIMARY KEY,
        NOMBRE VARCHAR(50),
        DIRECCION VARCHAR(100),
        TELEFONO VARCHAR(15)
    );

    CREATE TABLE PROIEKTUA (
        ID INT PRIMARY KEY,
        CODIGO VARCHAR(10),
        TITULO VARCHAR(100),
        PRECIO DECIMAL(10,2),
        CANTIDAD INT
    );

    CREATE TABLE BEZEROA (
        ID INT PRIMARY KEY,
        DIRECCION VARCHAR(100),
        HABITACIONES INT,
        Precio DECIMAL(10,2)
    );

    CREATE TABLE SALTA (
        ID INT PRIMARY KEY,
        PROIEKTUA_ID INT,
        IKASLEAK_ID INT
    );

    CREATE TABLE Matrikula (
        ID INT PRIMARY KEY,
        PROIEKTUA_ID INT,
        IKASLEAK_ID INT
    );
  
```

2.- Esandako baloak betetzen dituzten hirien zerrenda itzuliko oñ sistemak, hau da, ezaugarrir guzitza tan bezeroak esandako baino balio handiago bat duten hiria. Horrela, sistemak bigarren pantaila bat aterako du, hirien zerrendarekin, bat aukeratzeko eskatuz. Adibidez:

Zuk esandako baloak betetzen dituzten hirak honakoak dira:
Donostia
Santander
Gijón
A Coruña

Bat aukera ezazu:

3.- Sistemak aukeratutako hirien ostatuak zerrenda itzuliko du. Zerrenda horretan ostatuak kategorien arabera sailkatuta azaltuko dira, eta maila zehatz batean ostatuak EZ badago, mezu egoki bat aterako da. Adibidez:

(1) DONOSTIA
BOST IZARREKO HOTELAK
Maria Cristina
LAU IZARREKO HOTELAK
Costa Vasca
Aranzazu-Donostia
Londres
HIRU IZARREKO HOTELAK
Amara Plaza
Europa
BLIZARREKO HOTELAK
Hiri horretan ez dago kategoria horretako ostaturik
...

4.- Ondoren sistemak honako datuak eskatuko ditu: bezeroaren nan zenbakia, nahi den ostatuaren izena, sarrera-data eta egonaldiaren egun-kopurua.

5.- Sistema egun horietarako plaza libreak daudela ejiazatuko du, eta balekoan erresa egingo du. Adibidez, sarrera-data 99/12/9 izanik egonaldia 3 egunetako bada, sistemak erresa 99/12/9, 99/12/10 eta 99/12/11 egunetako egingo du. Plaza libreak egin GUZTIETARAKO ez badadeez da erresa egingo. Data batean plaza libenik egongo da egun horietarako egindako erre-serben kopurua hautatutako ostatu horrek dituen plazent kopuna (Ostatura taularen gel-a-kop atributua) baino txikiagoa denean.

6.- Erresa egingo balitz egin, bezeroari daggokion mezuia atera eta 4. pausora itzuliko lizateke (erresa egingo, gehienez ere, beste bi aukera emanez)

# 1. Arbeit

try {

:

#sql iterB1 = {select NA, <sup>Tfnotebilie</sup> from Haukello-Trango\_Bezero INNER JOIN  
Bezero ON HTB.NA = Bezero.NA where HTB.Helmage = 'Bilbo'  
as HTB}

Bezero ON HTB.NA = Bezero.NA where HTB.Helmage = 'Bilbo'

System.out.println("NA " + NA + " and HTB.InteroDate = '2012-05-31' and HTB.Zenbilie = 5");  
while (iterB1.next()) {

try {

na = iterB1.NAC();

telefono = iterB1.Tfnotebilie();

System.out.println("NA " + " telefono");

# if (telefono.equals(goldtel(telefono))) {

#sql iterB1 = {select NA from Bezero where NA = :na};

# else { #sql iterB1 = {select NA from Bezero where NA = :na};

#sql {set transaction isolation level read commit};

#sql {Delete from Haukello-Trango\_Bezero where NA = :na};

#sql {commit};

catch (SQLException e) {

#sql.rollback();

if (e.getErrorCode == "ConstraintViolated") {

System.out.println("Doppelte Sonnentags Begeben trangolko bezero vor! ");

continue;

8

■

#sql {set transaction isolation level read committed};  
#sql {insert into Haukello-Trango values ('Bilbo', '2012-05-31', 12, Gestid, 9.00, La\_Casilla, 60);}

#sql {lock table Haukello-Trango\_Bezero in shared mode};

while (iterB1.next()) {

na = iterB1.NAC();

#sql {UPDATE Haukello-Trango\_Bezero SET Zenbilie = 12 WHERE Helmage = Bilbo,  
and InteroDate = '2012-05-31' and Zenbilie = 5 and NA = :na};

while (iterB1.next()) {

na = iterB1.NAC();

#sql {Delete from Haukello-Trango\_Bezero WHERE Helmage = Bilbo and InteroDate = '2012-05-31'  
and Zenbilie = 5 and NA = :na};

8

#Sql { Delete from Hotelde-Braegge where Adresse = Bilbao  
and istestadt = 2012-05-31  
and zubelue = 54;  
#sql kommt h

4

{ catch (SQLException e) {

System.out.println ("Error: " + e.getErrorCode());

{ #Sql { rodtBode};  
exit(-1);

{ catch (Exception e) {

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

exit(1);

5

## 5. Anil Mehta

```

import java.io.*;
import java.sql.*;
import java.util.*;

public class informacionTurista {
    public static void main (String args[]) throws Exception {
        int bimail, Historia, paisaje, handkult, Gkip, gelak;
        Iteratorklasse iterB, iterBj;
        String Izena, Oizena, han;
        Date date;
        BufferedReader sarrera = new BufferedReader (new InputStreamReader (System.in));
        String Driver = "Oracle.jdbc.driver.OracleDriver";
        String DBURL = "jdbc:oracle:thin:@192.168.0.10:1512:Erreka";
        String UTD = "ErrekaTalde";
        String PWD = "pasahitza";
        try {
            System.out.println("Hautzko erangarriak zure hiri idealek bate beharko
liburueen balioko sar itza, mesedeb. Baloa 0 (ez du gerentzain) eta 10 (maitasuneko
degi bitartekoak izan behar da)");
            System.out.println(" bizi-mailer: ");
            bimail = sarrera.nextInt();
            System.out.println(" Historia: ");
            Historia = sarrera.nextInt();
            System.out.println(" Paisaien interes: ");
            paisaje = sarrera.nextInt();
            System.out.println(" Handortean kultatea: ");
            handkult = sarrera.nextInt();
            #Sql iterB = #Select izene from hiri where Bizi-mail >=:bimail and
            interestHistorialor >=:historia and PaisaienInteresa >=:paisaje and
            HandorteanKultatea >=:handkult;
        }
    }
}

```

System.out.println("Alle escondedas betreuen derivate liric  
hundert dira");

while (iterBet.next()) {

Izene = iterBet.getStr();

System.out.println(" " + Izene);

} iterBet.close();

System.out.println("Bet allere caoau: ")

Izene = Serre.readLine();

int i = 5;

~~Asql iterBi = {Select Oizene, libreglobes from Ostetue where liric = :Izene;~~

while (i > 0) {

~~Asql iterBi = {Select Oizene, from Ostetue where liric = :Izene and libreglobes >~~

System.out.println(" " + "Oizene");

is(iterBi.next());

while (iterBi.next()) {

Oizene = iterBi.Oizene();

System.out.println(" " + Oizene);

} else {

System.out.println(" Hin houeta es dego libreglobes kostetuki")

} -->

}

while (true) {

System.out.println("Sortu van Beaubhia");

no = Serre.readLine();

System.out.println("Sortu uhi den ostetueren (izene)");

Oizene = Serre.readLine();

System.out.println(" Serre-data: ");

data = Serre.readLine();

System.out.println(" Egin lepuru sortu");

Glop = Serre.nextInt();

~~Asql {Select getlop into getlop from Ostetue where Oizene = :Oizene;}~~

for (i = 0; i < Glop; i++) {

~~Asql Update carbur into :elop from Eisreserbe where Oizene = :Oizene and~~

~~libre = i & elop & getlop & libre & data = :data + i;}~~

} if (not libre) ... // berriak seihua. continue;

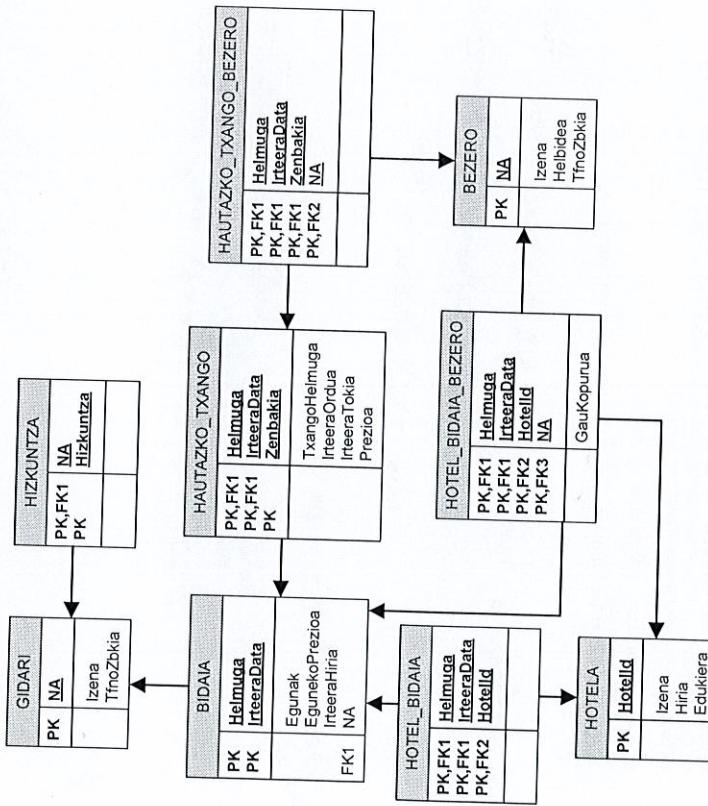
~~Asql transaction isolation level read committed;~~

break for (i = 0; i < gaudi; i++); Asql insert into Eisreserbe values (:no, :Oizene, :data);

## Datu-baseak. Programa SQLJn

### Datu-baseak. Programa SQLJn

Bedi honako UML diagrama:



Gidari (NA, Izena, TfnoZbkia)  
Hizkuntza (NA, Hizkuntza)

Bidaia (Helmuga, IrteeraData, Egunak, EgunekoPrezioa, IrteeraHiria, NA)  
Hotel (HotelId, Izena, Hiria, Edukiera)

Hotel\_Bidaia (Helmuga, IrteeraData, HotelId)

Bezero (NA, Izena, Helmidea, TfnoZbkia)

Hotel\_Bidaia\_Bezero (Helmuga, IrteeraData, HotelId, NA, GauKopurua)

Hautazko\_Txango (Helmuga, IrteeraData, Zenbakia, TxangoHelmuga, IrteeraOrdua, IrteeraTokia, Prezioa)

Hautazko\_Bezero (Helmuga, IrteeraData, Zenbakia, NA)

Lanetik erretiratuta dauden pertsonentzat egun bat baino gehiago irauten duten bidaiaak antolatzenten dituen ONDO-PASA empresak, datuak kudeatuko dituen aplikazio informatiko bat garatu du.

- Bidaia bat (adibidez, Benidorm-era) data desberdinetan egin daitake. Bidaia bakoitzak helmuga eta irteera-data erabilzen dira. Gainera, bidaiaaren iraupena egunetan, eguneko prezioa, irteera-hiria, erabilgarri dauden hotelak eta bidaiai esleitutako gidari bakarra gordeko dira.
- Gidari bakoitzak bere NArekin identifikatzenten da eta bere izena, telefono-zenbakia eta hitz egiten dituen hizkuntzak ezagutzenean dira.
- Bezeroen, NArekin identifikatzak, izena, helbidea eta telefono-zenbakia ezagutzenean dira.
- Hotel bakoitzeko bere identifikazio zenbaki bakarra, izena, hiria eta edukiera gordetzen dira.
- Bidaia batean parte hartzen duen bezero bakoitzeko zenbat gau pasa dituen bezeroak bidaia horri esleitutako hoteletan gordetzen da. Bidaiai egun bat baino gehiago irauten duenez gerta daitake bidaia berean hotel bat baino gehiago esleitu izana.
- Bidaia bakoitzak hautazko txango bat edo gehiago eduki ditzake. Hautazko txango teera-tokia gordetzen da. Zein bezerok egin duen hautazko txango bakoitzak ere gordetzen da.

Gasteizko Artium museoan desinfekzio-tratamendu bat egingo da 2012-05-31an. Horren ondorioz, ezin izango da egin egun horretarako aurreikusita zegoen eta (Bilbo, 2012-05-31, 5) identifikazioa duen hautazko txangoa. Hautazko txango honek datu-basetik desagerantzera doan txangoan izena emanda zeukanen bezeroei. Txango berria ez dago datubasean gordeta: (Bilbo, 2012-05-31, 12, Gasteiz, 9:00, La Casilla, 60). "Hautazko txangoan izen-ematea eguneratu/ezabatu" erabilpen-kasua implementatu duen eta datubasea SQLJren bidez atitztuko duen Java programa idatz ezazu (2 puntu).

- (Bilbo, 2012-05-31, 5) hautazko txangoan izena eman duten bezeroak zein diren identifikatu:

NA	Telefonoa
1123	94356677
2234	688678901
3455	943500890

- Identifikatutako bezero bakoitzari telefonoa erabiliz galduen eta hautazko txango berrian izena eman nahi duen ala ez (erantzun posibleak: "bai", "ez") eta datubasean aldaketa guztiek jasotzeko behar diren ekintzak gauzatu.

Oharra: Kontuan izan une berean eta bulego desberdinietatik sistemaren gaineko atzipen konkurrentea egin ahal dela. Hori horrela izanik posible da, adibidez, hautazko txangoen izen-emateko prozesua hanbat erabiliztailek aldi berean egitea.

Izena:

## Datu-baseak. Programa SQLJn

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

class Ondopasa
{
    public static void main (String args[]) throws Exception
    {
        BufferedReader sarrera = new BufferedReader (new InputStreamReader (System.in));
        String DRIVER = "oracle.jdbc.driver.OracleDriver";
        String DBURL = "jdbc:oracle:thin:@000100.gi.ehu.es:1512:Errealia";
        String UID = "ERABILITZAILEA";
        String FWD = "PARAHITZA";
        try
        /* Datu-basearekin konexioa sortu eta konexio-testuinguru lehenetsia definitu */
        {
            Driver controlador = (Driver) Class.forName(DRIVER).newInstance();
            DriverManager.registerDriver(controlador);
            DefaultContext test = new DefaultContext(DBURL, UID, FWD, false);
            DefaultContext.setDefaultContext(test);
            System.out.println ("Konektioa eginda");
            catch (Exception e)
            {
                System.err.println("Errorea konexioan: " + e.getMessage());
                System.exit(-1);
            }
        }
        #sql iterDat = hSelect HTB.NA, telefono from HTB where NA := nahi;
        AND identificacion = "5";
        while (iterDat.next())
        {
            if (telefono != null)
            {
                System.out.println ("NA: " + NA + " telefono: " + telefono);
            }
        }
    }
}
```

#sql delete from HalkoBero\_trxos where Helmua = "Bilbo"  
AND TtxoData = "2012-05-31" AND Identificacion = "5";

#sql hInsert into HalkoBero\_trxos values ('Bilbo', 2012-05-31,  
12, Gasteiz, 9:00, 'Le Castillo', '60');

while (iterBsi.next()) { nom = iterBsi.get("NA");  
#sql hInsert into HalkoBero\_trxos\_Bezero  
values ('Bilbo', 2012-05-31, 12, nom); }

#sql hDelete from HalkoBero\_trxos where Helmua = "Bilbo"  
AND TtxoData = "2012-05-31" AND Identificacion = "5";

#sql hInsert into HalkoBero\_trxos\_Bezero values (HTB.Natural)  
Join Bezero where HTB.Helmua = "Bilbo" AND TtxoData = "2012-05-31"  
AND Identificacion = "5";

4

#Sql hSet transaction isolation level and commit;  
#sql hInsert into HalkoBero\_trxos\_Bezero values (HTB.Natural)  
Join Bezero from HalkoBero\_trxos where NA = nahi;  
#sql hDelete from HalkoBero\_trxos\_Bezero where NA = nahi;

## Ariketak: 5 → 4 bidean

Ustez libre zegoen zerbait jadanik ez dago libre. Nola tratatu?

### 5.- Ostatuen erreserba

```
void pauso2( ) throws Exception {
    BufferedReader sarrera = new BufferedReader (new InputStreamReader (System.in));
    String na, izena, data, egunak, oizena;
    int kont, gkop, ekop = 0;
    Boolean libre?;
    try {
        while (true) {
            kont = kont + 1;
            ... // Datuak eskatu eta irakurri
            try {
                libre? = true;
                #sql { select gelakop into :gkop from Ostatua where Oizena = :izena };
                for (int i=0; i=egunak; i++) {
                    #sql { select count(*) into :ekop from Erreserba where Oizena = :izena and data = :data + :i };
                    libre? = (libre? and (gkop > ekop));
                }
                if (not libre?) {
                    System.out.println("Ostatua data horietan erreserbatuta dago");
                    System.out.println("Berriztai!");
                    if (kont < 3) continue;
                }
                #sql { set transaction isolation level read committed };
                for (int i=0; i=egunak; i++) {#sql { insert into Erreserba values (:na, :izena, :data + :i) };
                #sql { commit };
                break
            } catch (Exception e) { System.out.println("errorea: " + e.getMessage()); }
        } catch (Exception e) {System.out.println("errorea: " + e.getMessage()); }
    }
```

### 4.- Proiektuak (murriztapen bat dauka definituta)

```
#sql {set transaction isolation level read committed};
for (i=0, j=2, i++) #sql {insert into matrikula values (:ident, :taula(i), null, null)};
#sql {commit};
break;
} catch (SQLException e) {
    #sql {rollback};
    System.err.print (" Aukeraketa prozesua berriztai hasi beharko da ");
    if (e.getErrorCode() == "CONSTRAINT_VIOLATED") {
        System.out.println ("Aukeratutako proiekturen bat jadanik ez dago libre.");
        if ( lag = 3) break;
        else continue;
    }
}
...
...
```



```
        System.out.println ("Egindako aukeraketa guzta gorde da ");
        break;
    } catch(e SQLException){
        #sql {rollback};
        System.err.print (" Aukeraketa prozesua berriz hasi beharko da ");
        if (e.getErrorCode( ) == "CONSTRAINT_VIOLATED") {
            System.out.println ("Aukeratutako proiekturen bat jadanik ez dago libre.");
            if ( lag = 4 ) break; /* Aukeraketa lau alditan bakarrik egin dezake*/
            else continue;
        }
    } // if
} // while
} catch(SQLException e){ System.err.print (" Errorea gertatu da "); }
}
}
```

```

import java.sql.*;

#sql iterator IterKlasea (String izena);           // iteratzalea SELECT galderentzat
class ProiekturenAukeraketa {
public static void main ( String args[] ) {
    IterKlasea iterBat, iterBi;
    String ident, proiek, sailBat;
    String[ ] taula = new String[3];
    int kont, lag;
    BufferedReader sarrera = new BufferedReader (new InputStreamReader (System.in));

    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);
        }

        System.out.print ("Ikaslearen NA: "); ident = sarrera.readLine();
        /* Ikaslek proiekturen aukeraketa egina al du? */
        #sql iterBat = {select proiek as izena from matrikula where ikas = :ident};
        if (!iterBat.next()) { /* Ikaslek oraindik ez du proiekturek aukeratu, proiekturen zerrenda atera */
            iterBat.close();
            lag = 0;
            while (true) {
                try {
                    lag = lag + 1;      /* Sail guztien zerrenda */
                    #sql iterBat = {select izena from saila order by izena};
                    while (iterBat.next()) {
                        sailBat = iterBat.izena();
                        System.out.println ("SAILA: " + sailBat);
                        /* 5 ikaslek baino gutxiagok aukeratu dituzten proiekturen zerrenda */
                        #sql iterBi = {select proiek as izena
                                      from matrikula right join proiekta on proiek = titulua
                                      where bereSaila = :sailBat
                                      group by proiek
                                      having count (ikas) < 5};
                        kont = 0;
                        while (iterBi.next()) {
                            System.out.println (" " + iterBi.izena());
                            kont = kont + 1;
                        }
                        if (kont == 0) System.out.println ("Sail honek ez du proiektu librerik");
                        iterBi.close();
                    }
                    iterBat.close();
                    /* Ikasleari bere proiektauak eskatu, bitarteko taulan gorde, eta
                     gero insert guziak bata bestearen ondoren egin*/
                    for (int i=0; i<2; i++) {
                        System.out.print ("Eskatutako proiektauren izena: ");
                        taula [i] = sarrera.readLine();
                    }
                }
                #sql {set transaction isolation level read committed};
                for (i=0, i<2, i++) #sql {insert into matrikula values (:ident, :taula(i), null, null)};
                #sql {commit};
            }
        }
    }
}

```

## TRANSAKZIOAK

1. Hasterako balioak: X=1000; N=300; M=100.

T1 transakzioa	T2 transakzioa
read(X)	set transaction isolation level read uncommitted
X:=X-N	
	read(X)
	X:=X+M
write(X)	
commit	
	write(X)
	commit

- (a) Bi transakzioak exekutatu ostean, ze balio edukitiko du X aldagaiak? \ \(\bigcirc\)
- (b) Eta ze balio beharko luke? \ \(\bigcirc\)
- (c) Ze kasuaren aurrean gaude:

i. Egunetarazte galdudaren arazoa

ii. Behin-behineko aldatzearen arazoa (irakurketa zikina)

iii. Kalkulu okerraren arazoa

iv. Irakurketa errepikazina

v. Ez da inolako arazorik egon

2. Hasterako balioak: X=1000; N=300; M=100.

T1 transakzioa	T2 transakzioa
read(X)	set transaction isolation level read uncommitted
X:=X-N	
	write(A)
commit	
	read(X)
	X:=X+M
	write(X)
	commit

- (a) Bi transakzioak exekutatu ostean, ze balio edukitiko du X aldagaiak? \ \(\bigcirc\)
- (b) Eta ze balio beharko luke? \ \(\bigcirc\)
- (c) Ze kasuaren aurrean gaude:

i. Egunetarazte galdudaren arazoa

ii. Behin-behineko aldatzearen arazoa (irakurketa zikina)

iii. Kalkulu okerraren arazoa

iv. Irakurketa errepikazina

v. Ez da inolako arazorik egon

3. Hasterako balioak: X=1000; N=300; M=100.

T1 transakzioa	T2 transakzioa
read(X)	set transaction isolation level read uncommitted
X:=X-N	
	read(X)
write(X)	
commit	
	write(X)
	commit

- (a) Bi transakzioak exekutatu ostean, ze balio edukitiko du X aldagaiak? \ \(\bigcirc\)
- (b) Eta ze balio beharko luke? \ \(\bigcirc\)
- (c) Ze kasuaren aurrean gaude:

i. Egunetarazte galdudaren arazoa

ii. Behin-behineko aldatzearen arazoa (irakurketa zikina)

iii. Kalkulu okerraren arazoa

iv. Irakurketa errepikazina

v. Ez da inolako arazorik egon

5. ERABILTZIALE

T1 transakzioa	T2 transakzioa
set transaction isolation level read committed	
(a) select id from erabiltziale	insert into erabiltziale values (3, Axun, 23)
read(X)	
X:=X-N	
	commit
write(X)	
commit	

Zer bueltatuko du SELECT bakoitzak:

- (a) select id from erabiltziale 1, 2  
 (b) select id from erabiltziale 1, 2  
 (c) select id from erabiltziale 1, 2, 3
- (a) Bi transakzioak exekutatu ostean, ze balio edukitiko du X aldagaiak? \ \(\bigcirc\)

6. ERABILTZAILLE

id	izena	adina
1	Jon	20
2	Mander	25

T1 transakzioa	T2 transakzioa
set transaction isolation level read uncommitted	
(a) select id from erabiltzaile	insert into erabiltzaile values (3, Axun, 23)
(b) select id from erabiltzaile	commit
(c) select id from erabiltzaile	
commit	

Zer bueltatuko du SELECT bakoitzak:

- (a) select id from erabiltzaile  $\backslash, 12$
- (b) select id from erabiltzaile  $\backslash, 17 \Rightarrow$
- (c) select id from erabiltzaile  $\backslash, 23$

7. Ze gertatzen da honako exekutazio honetan:

T1 transakzioa	T2 transakzioa
lock x (B)	
read (B)	
B := B+50	
write (B)	
	lock s (A)
	read (A)
	lock s (B)
lock x (A)	
...	

Etxe bloke bat dago

↳ Solo Bain  
↳ Pentsu aurreko jatorrizko  
↳ RollBack

8. Honako sententziak exekutatu ostean...

```
DefaultContext ing = new DefaultContext(  
    "jdbc:odbc:adibidea", "Ni", "auskalo", false);  
  
DefaultContext.setDefauleContext(ing);  
  
#sql {insert into SAILA values (100, 'PAYROLL');  
#sql {insert into SAILA values (200, 'ACCOUNTING');  
#sql {insert into SAILA values (300, 'MARKETING');  
#sql {insert into SAILA values (400, 'BRAND');  
#sql {insert into SAILA values (500, 'CLIENT ACCOUNT');  
#sql {commit};  
#sql {insert into SAILA values (500, 'CLIENT ACCOUNT');  
#sql {rollback};  
#sql
```

- a) Zenbat tupla dago datu-basean?  $3 \text{ tuple}$
- b) Zein dira tupla horiek?

200 Marketing

400

Brand

500 Client Account

#### 49. Arikete

Select Helmuge, IntercDate

From Bidarie

Where Egundek \* EgundekPresia <= ALL (Select Egundek \* EgundekPresia

From Bidarie)

#### 50. Arikete

Select Helmuge, IntercDate, Zenbaki

From Hecteko-Txango-Bedero AS HBB

Group By Helmuge, IntercDate, Zenbaki

Having Count(\*) >= ALL (Select Count(\*)

#### 51. Arikete

Select NA, Izena

From ~~Bedero Natural Join~~ AS HBB

(Bedero Natural Join Hotel, Bidarie, Bedero) AS HBB Group By HBB.Helmuge, HBB.IntercDate, HBB.Zenbaki

INNER JOIN Bidarie ON

HBB.Helmuge = B.Helmuge AND HBB.IntercDate = B.IntercDate

Where Egundek \* EgundekPresia <= 3000

Except

Select NA

From H.B.B ~~AND~~ INNER JOIN Bidarie

Where Egundek \* EgundekPresia <= 3000

#### 52. Arikete

Select NA, Izena

From Bidarie AS G

Where Not Exists (Select \*

From Bidarie AS B

B.NA <> G.NA)

Where Egundek > 50 AND NotExists (

(Select \*

From Bidarie AS B1

Where G.NA =

### 53. Arikete

Select NA, Izera

From Gideri AS G Natural Join Bidear AS B

Where B.Helvage like 'Errone' AND B.InteractionDate = '12-01-01'

Except

Select NA,

From Bidear

Where B.Helvage like 'Errone' AND B.InteractionDate = '12-02-01'

### 54. Arikete

Select NA, Izera

From Gideri AS G Natural Join Bidear AS B

Where Helvage like 'Errone'

Except

Select NA

From Bidear

Where Helvage <> 'Errone'

### 55. Arikete

Select Helvage, InteractionDate, Zeibalki

From Hezkaklo\_txago AS Ht left Join Hezkaklo\_txago\_Bezero AS HTB

ON Ht.Helvage = HTB.Helvage AND Ht.InteractionDate = HTB.InteractionDate AND

Ht.Zeibalki = HTB.Zeibalki

Where NA IS Null

## 56. Arillete

Select NA, Izene

From Gideri AS G

Where H.Helvige Lille 'Italiere' AND Exist (Select \*

From Bideie AS B

Where 'Helvige Lille Errou' AND G.NA = B.NA)

## 57. Arillete

Select NA, Izene

From Bezero AS B

Where Exist (Select \*

From Hotel-Bideie-Bezero AS HBB

Where HBB.NA = HBB.HA AND HBB.Helvige Lille 'Paris'  
AND Not Exist (Select \*

From Hotel AS H

H.Hinie < 'Paris' AND H.HelId = HBB.Helvige

Select Helvige, IztereDate, Iztere

From Bideie AS B

INNER JOIN (Hotel-Helvige AS HT ON B.Helvige = HT.Helvige AND

B.IztereDate = HT.IztereDate)

Where Count (NA)

Select Helvige, IztereDate, Count (NA)

From Hotel-Helvige AS HT Natural Join Hotel-Helvige-Bezero  
where

Group By Helvige, IztereDate, Republic

Having Count (NA) / >= All / Select Count (NA)

= (Select Max from

From Hotel-Helvige Natural Join Hotel-Helvige-Bezero  
where HT.Hel = HT1.Hel AND HT.Iz > HT1.Iz  
Group

## 58. Arillete

### 53. Arikete

Select NA, Izene

From Gideri AS G Natural Join Bidcie AS B

where B.Helvuge Like 'Erone' AND B.IzereDate Like '12-01-01'

Except

Select NA, Izene

From Gideri AS G1 Natural Join Bidcie AS B1

where B.Helvuge Like 'Erone' AND B.IzereDate Like '12-02-01'

### 54. Arikete

S 51

Select NA, Izene

From Gideri AS G ~~Natural Join Bidcie AS B~~ Right Join Bidcie AS B

where IzereDate ~~(Select)~~

Helvuge IN (From Bidcie  
Select Helvuge

From Bidcie

where Helvuge Like 'Erone' AND G.NA = B.NA

### 55. Arikete

Select ~~HtB.~~ Helvuge, IzereDate, Bebelin

From Hercules\_trigo\_Bebeli <sup>AS HtB</sup> Right Join Hercules\_trigo <sup>AS HT</sup> ON  
HtB.Helvuge = HT.Helvuge AND HtB.IzereDate = HT.IzereDate AND HtB.Bebelin = HT.

where NA IS NULL

#### 49. Ariketa

Select Helmuga, IntercetData, Egunel \* Egunello Precio  
 From Bidaia

Where

Egunel \* Egunello Precio <= ALL (Select

From Bidaia)

#### 50. Ariketa

Select Helmuga, IntercetData, <sup>Zerbeltia</sup>Count (NA)  
 From Hotelak\_Bango\_Bederro

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

From Hotelak\_Bango\_Bederro

Group By Helmuga, IntercetData, Zerbeltia

#### 51. Ariketa

Select NA, Izene

From Bederro <sup>AS B</sup>

Where Not Exists (Select \*

> 3000

EXCET - NAT IN -

< 3000

From Hotel-Bideia-Bederro AS HBB

Where B.NA = HBB.NA AND

Not Exists (Select \*

From Bideia AS B1

Where B1.Helmuga = HBB.Helmuga AND

B1.IntercetData = HBB.IntercetData AND

Egunel \* Egunello Precio > 3.000))

#### 52 Ariketa

Select NA, Izene

From Gideri AS G

Where Not Exists (Select \*

From Bidaia AS B

Where Egunel > 50 AND

Not Exists (Select \*

From Gideri AS G1

Where B.NA = G.NA)

Biderrik Izenak the AND ID

### 39. Ariketa

Select NA, Izene

From Bezero ASB

where Exist (Select\*)

From Hotel\_Bideia\_Bezero AS HBB Natural Join Hotel AS H

where ~~Izene~~ Izene = Peris AND HBB.NA = B.NA AND

Not Exist (Select\*)

From Hotello\_txango\_Bezero AS HTB

where HTB.Helmuge = HBB.Helmuge AND

HTB.Isteerdale = HBB.Isteerdale AND B.NA = HBB.NA

### 47. Ariketa

Select HT.Helmuge, HT.Isteerdale, HT.Zenubalde, Count(HTB.NA)

From Hotello\_txango AS HT Left Join Hotello\_txango\_Bezero AS HTB ON

HT.Helmuge = HTB.Helmuge AND HT.Isteerdale = HTB.Isteerdale AND

HT.Zenubalde = HTB.Zenubalde

Group By HT.Helmuge, HT.Isteerdale, HT.Zenubalde

### 48. Ariketa

Select HT.Helmuge, HT.Isteerdale, HT.Zenubalde, Count(HTB.NA)

From ~~From~~ Hotello\_txango AS HT Left Join Hotello\_txango\_Bezero AS

ON HT.Helmuge = HTB.Helmuge AND HT.Isteerdale = HTB.Isteerdale

AND HT.Zenubalde = HTB.Zenubalde

Where ~~Count~~ Count(HT.NA)  $\leq 5$

Group By HT.Helmuge, HT.Isteerdale, HT.Zenubalde

### 45. Arilleta

Select \*

From Bidaia<sup>N</sup> Left JOIN Hauteko - Txango AS HTB ON

HTB.Helmuge = HTB.Helmuge AND HTB.IsteeraDate = HTB.IsteeraDate

Group By HTB.Helmuge, HTB.IsteeraDate

### 46. Arilleta

From HTB

Select \* HTB, HTB.NA, HTB.Isteera

From Hauteko - Txango AS HT left Join (Hauteko - Txango - Bezero<sup>HTB</sup> Natural Join

Bezero AS B)

ON HTB.Helmuge = HTB.Helmuge AND HTB.IsteeraDate = HTB.IsteeraDate AND

HTB.Zeebelie = HTB.Zeebelie

Group By HTB.Helmuge, HTB.IsteeraDate, HTB.Zeebelie

### 38. Arilleta

Select NA, Isteera

From Bezero AS B

Where Not Exist (Select \*

From Hotel-Bideia - Bezero AS HBB Natural Join Hotel AS H

Where H.Hirie = Paris AND HBB.NA = B.NA

AND Not Exist (Select \*

From Hauteko - Txango - Bezero AS HTB

Where HTB.Helmuge = HBB.Helmuge

AND HTB.IsteeraDate = HBB.IsteeraDate AND

HTB.NA = B.NA))

AND Exists (Select \*

From Hotel-Bideia - Bezero AS HBB Natural Join Hotel AS H

Where Hirie = Paris AND B.NA = HBB.NA)

#### 42. Arillete

Select NA

From Baserro Hotel\_Bideia - Bezero

where Helmuga Like 'Funchal'

Group By NA, Helmuga, IrteeraDate

Having Count(\*) >= (Select Count(HotelID)

From Hotel

where Helmuga Like 'Funchal'

#### 43. Arillete

Select HotelId, count(\*)

From Hotel\_Bideia - Bezero AS HBB Natural JOIN Hotel

Group By HotelID

Having Count(\*) >= ALL (Select Count(\*)

From Hotel\_Bideia - Bezero AS HBB Natural JOIN Hotel

Group By HotelID

Helmuga / Sum(Precio))

#### 44. Arillete

Select NA, Izena, ~~Count~~<sup>Sum</sup>(Precio)

From (Hautzalbo - Txango - Bezero AS HTB Natural JOIN Bideia) Natural JOIN

Bideria 16

Group By Helmuga, IrteeraDate

Having Sum(Precio) >= ALL (Select Sum(Precio))

From Hautzalbo - Txango - Bezero Natural JOIN  
Bideia )

Group By Helmuga, IrteeraDate

Helmuga / Sum(Precio))

### 36. Arikete

Select NA

From Bidaii\_Gidori

where Unique (Select NA

From Bidaii\_NA

where Helmoga Lille Paris AND Gidori.NA ≠ Bidaii.NA)

### 37. Arikete

Select NA

From Gidori

where Not Unique (Select NA

From Bidaii\_NA

where Helmoga Lille 'Paris' AND Gidori.NA = Bidaii.NA)

### 40 Arikete

Select Helmoga, Irteeradeta, Zerbaltia

From Hantzu\_Bangu\_Bezero

Group By Helmoga, Irteeradeta, Zerbaltia

Having Count(\*) > 30

### 41. Arikete

Select Helmoga, Irteeradeta, Zerbaltia

From Hantzu\_Bangu\_Bezero Bidaii

Group By Helmoga, Irteeradeta

Having Helmoga, Irteeradeta IN (Select Helmoga, Irteeradeta

From Hantzu\_Bangu\_Bezero  
where

From Hantzu\_Bangu\_Bezero

Group By Helmoga, Irteeradeta

Having Count(Distinct transHelmoga) = 1

### 32. Athlete

Select Helmuge, IrteeraDate

From Bidaiia

Where EgunekoPrezio > Any (Select EgunekoPrezio

From Bidaiia

Where Helmuge Like 'Eromal')

### 33. Athlete

Select Helmuge, IrteeraDate

From Bidaiia

Where Prezio \* EgunekoPrezio >= ALL (Select Prezio \* EgunekoPrezio  
From Bidaiia)  
Where

### 34. Athlete

Select Helmuge, IrteeraDate.

From Bidaiia AS B1

Where EgunekoPrezio <= ALL (Select Prezio

From Hautzaile\_Bezero AS HT

Where B1.Helmuge = HT.Helmuge

AND B1.IrteeraDate = HT.IrteeraDate)

### 35. Athlete

Select NA

From Bezero

Where NA <> ALL (Select NA,

From Hotel\_Bidaiia\_Bezero)

Where Helmuge Like 'Asilah' AND IrteeraDate ~~= 11/01/01~~

## 26. Aritletə

Select İzne

From Bezero Natural JOIN Hotel\_Bideci\_Bezero

Where

Where HotelID IN (Select HotelId

From Hotel

Where Hizie Like 'Durango')

## 27. Aritletə

Select NA

From Bideci\_Bezero HotelNo\_Fango\_Bezero

Where NA IN Not (Select NA

From HotelNo\_Fango\_Bezero)

## 29. Aritletə

Select Helmuge, InteresData

From Bideci

Where Egnoklup, <sup>Helmuge</sup> InteresHizie IN (Select <sup>Helmuge</sup> Egnoklup, <sup>Helmuge</sup> InteresHizie

From Bideci

Where InteresHizie Like 'Travice' AND

InteresData = '11/07/07')

## ALL ETƏ & ANY

## 31. Aritletə

Select Helmuge, InteresData

From Bideci

Where Egnoklup\_Precice > ALL (Select Egnoklup\_Precice

From Bideci

Where Helmuge Like 'Ermeni')

## 16. Arikete

Select Count (Distinct NA)

From Bidic

where Helmuge Like 'Paris'

## 17. Arikete

Select NA, Count(HotelCount)

From HotelCount

Group BY NA

## 18. Arikete

Select Helmuge, Count(Distinct B.NA)

From (Bidic AS B Natural JOIN Hotel\_Bidic) Natural JOIN  
(Hotel\_Bidic Natural JOIN Hotel\_Hire)  
where Hotel\_Hire.HotelID IN (Select HotelID  
from Hotel\_Hire  
where Hotel\_Hire.HotelName Like 'Italie' AND Hotel\_Hire.HotelName Like 'France')

Group By B.Helmuge

25.

## Exists etc IN

### 25. Arikete

Select Distinct NA, Scene

From Gideri

where NA IN (Select NA

From Bidic

where Eguncile Prezice \* Eguncile > 100

## Is NULL, IN, Like etc Between

### 7. Arikete

Select Helmuge, Interredate

From Bidcie

Where NA is NULL

### Union, except and intersect

### 11. Arikete

Select ~~Iden~~. Dene

From Gideri

Union

Select ~~Bzzer~~. Dene

From Bzzero

### 13. Arikete

Select Iden

From Gideri

Except

Select Iden

From Bzzero

### Function aggregative etc Group By

### 14. Arikete

Select AVG (Prciye \* Egnde Prciye)

From Bidcie

where Interredire like 'Domestic'

Datu-baseak: SQL Ariketa

INNER JOIN eta Auto JOIN

3. Ariketa

Select Helmuga, IrteeraData

From Bidaria As B1 INNER JOIN Hautsak - Txango As HT ON

B1.Helmuga = HT.Helmuga AND B1.IrteeraData = HT.IrteeraData

where HT.TxangoHelmuga like 'toledo' AND B1.Helmuga like 'Madril'

AND B1.IrteeraHiria like 'Domestic'

5. Ariketa

Select Helmuga, IrteeraData, NA Izene

From (Bidaria As B1 INNER JOIN Gideri AS G ON B1.NA = G.NA)

INNER JOIN (Bezero As Be INNER JOIN Hotel\_Bidaria\_Bezero AS HBB ON

HBB.NA = Be.NA) ON B1.Helmuga = HBB.Helmuga AND B1.IrteeraData =  
HBB.IrteeraData

~~Gideri~~

6. Ariketa

Select Izene NA

From (Hotel\_Bidaria\_Bezero AS HBB INNER JOIN Hotel AS H

ON HBB.HotelId = H.HotelId) INNER JOIN (Hotel\_Bidaria\_Bezero NATURAL JOIN  
Hotel) ON H.Gekklopurua < Hotel.Gekklopurua

where HBB.IrteeraData like '2011-0503' AND HBB.NA like '15238432'

AND Hotel.Izene like 'Gabrielle'

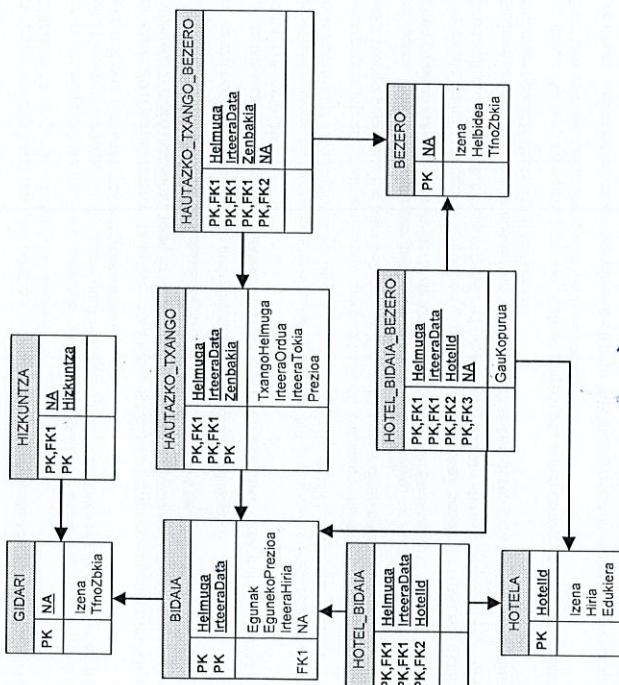
## Datu-baseak: SQL ariketak

### Datu-baseak: SQL ariketak

#### BIDAIAK

Lanetik erreirratuta dauden personentzat egun bat baino gehiago irauten duten bidaia antolatzenten dituen ONDO-PASA enpresak, datuak kudeanteko dituen aplikazio informatibo bat geratu du.

- Bidaia bat adibidez, Benidormera data desberdinaren egun dantekoa. Bidaia bakoitzeko helmuga eta irteera-data erabilizten dira. Gainera, bidaiaaren irapena eginetan, eguneko prezioa, irteera-hiria, erabiltarri dauden hotelak eta bidaia esleitutako gildari bakarra gordetzen dira.
- Gidari bakoitzak bere NA-arekin identifikatzen da eta bere izena, telefono-zenbakia eta hitz egiten dituen hizkuntzak ezagutzen dira.
- Hotel bakoitzeko bere identifikazio zenbakia, izena, hiria eta edukiera gordetzen dira.
- Bezeroen, NA-arekin identifikatzen, izena, helbidea eta telefono-zenbakia czaguntzen dira.
- Bidaia batean parte hartzen duen bezero bakoitzeko zenbat gau pasa dituen bezeroak bidaia horri esleitutako hautazko (1., 2., 3., ...) helmuga, prezioa, irteera-ordua eta irteera-tokiak gordetzen dira. Zein bezeroak egin duten hautazko txango bakoitzak ere gordetzen dira.
- Bidaia bakoitzak hautazko txango bat edo gehiago eduki ditzake. Hautazko txango bakoitzaren ordenazioa (1., 2., 3., ...) helmuga, prezioa, irteera-ordua eta irteera-tokiak gordetzen dira. Zein bezeroak egin duten hautazko txango bakoitzak ere gordetzen dira.
- Bidaia bakoitzak hautazko txango bat edo gehiago eduki ditzake. Hautazko txango bakoitzaren ordenazioa (1., 2., 3., ...) helmuga, prezioa, irteera-ordua eta irteera-tokiak gordetzen dira. Zein bezeroak egin duten hautazko txango bakoitzak ere gordetzen dira.



- Gidari (NA, Izena, TñoZbka)  
Hizkuntza (NA, Hizk)
- Bidaia (Helmuga, IterraData, EgunkeroPrezioa, IteeraHiria, NA)
- Hotela (HotelId, Izena, Hiria, Edukiera)
- Bezero (NA, Izena, Helbidea, TñoZbka)
- Hotel\_Bidaia\_Bezero (Helmuga, IterraData, HotelId, NA, GauKopurua)
- Hautazko\_Txango\_Bezero (Helmuga, IterraData, Zenbakia, TxangoHelmuga, IteeraOrdua, IteeraTokia, Prezioa)

#### INNER JOIN eta NATURAL JOIN

- Lor itzazu italeraz hitz egiteko gaitasuna duten gidarien NA-a eta izena.
- Lortu Sevilla hiriko hotelak eskeintzen dituzten bidaien kodea.
- Lor ezazu honako baldintza betetzan duten bidaien kodea: Donostiatik ateria eta Madrid helmuga gisa eduki, eta Toledo hirira hautazko txangoren bat eskaintzea.
- Lortu bidaiaren batean hautazko txangoko aukeratu dituzten bezeroen NA-a eta izena.
- Lor itzazu honako baldintza hau betetzan duten bidaien identifikazioa, eta bezeroen NA-a eta izena: bezeroaren izenak eta egindako bidaiaren gidariaren izenak berdinak izan behar dute (adibidez, Mikel, Mikel).

#### AUTOJOIN

6. Lortu 2011-05-03 datan Errromara egindako bidaian NA=152238432 daukan bezeroak Gabriella izena, eta ez nahitaez Gabriella hotelean) egondako bezeroen izena eta helbidea.

#### IS NULL, IN, LIKE eta BETWEEN eragileak

7. Lor itzazu gidariak Gabek bidaia kodeak.
8. Hotelaren egoitza Miarritzen edo Baiionean duten bidaiai lortu hotelaren izenarekin batera.
9. Lor ezazu, errepi kaperik gabe, Zarautzen bizi eta bidaiai bar egin duten bezercen izena.
10. Lor ezazu 2010. urtean zehar Donostiatik Madrilera egindako bidaia guztien izteera-data.

#### UNION, EXCEPT eta INTERSECT

11. Loru gidarien eta bezeroen izenak.
12. Loru izen bereko bezeroen eta gidarien izenak.
13. Lor ezazu gidarien izenak, bezero baten izenarekin bat ez datozenak.
- Funtzio agregatua (COUNT, SUM, MAX...) eta GROUP BY
14. Lor ezazu Donostiatik ateratzan diren bidaiai batez besteko prezioa.
15. Lor ezazu 2011ko apirilean Parisera egindako bidaiai kopurua.
16. Lor ezazu Parisera bidaiaiaren bat egin duten gidari desberdinen kopurua.
17. Gidari bakoitzeko kalkuluak ezazu zenbat hizkuntza menperatzen dituen.
18. Helmuga bakoitzeko kalkulu zenbat gidari dagoen, italiaraz hitz egiteko gaitasuna izanik, helmuga horretara bidaiaiaren bat gidatuko duena, baldin eta bidaia horretan Errromako hotelen batean ostatu hartzeko aukera badago.
19. Loru hautazko txango garestienari dagokion zenbakia.

#### [NOT] EXISTS eta [NOT] IN

20. Lor ezazu bidaiaien batetan ere parte hartu ez duten gidarien NA-a.
21. Lor ezazu honako baldintza hau betetzan duten gidarien NA-a eta izena: euskaraz hitz egiteko gaitasuna izanik, Lekumberriko hotelen batean ostatu hartzeko aukera duten gidaria izatea.
22. Lor itzazu ingelesez hitz egiten ez duten gidarien NA-a eta izena.
23. Lor itzazu Madrileko Ritz izeneko hotelean egoitza harri duten bezeroen NA-a, izena eta helbidea.
24. Lor itzazu Madrileko Ritz izeneko hotelean egoitza harri ez duten bezeroen NA-a, izena eta helbidea.

## Datu-baseak: SQL ariketak

## Datu-baseak: SQL ariketak

### OUTER JOIN eta UNION JOIN

45. Lor izazu bidaia guztien datu guztiak (BIDATA taulan jasotzen diren), eta, bidaia hautazko txangoak baditu, lortu txangoen datu guztiak ere (txangorik ez badu, NULL atertua beharko litzateke txangoari lotutako datuetan).

46. Lor izazu hautazko txangoen datu guztiak, eta, txango horiek bezerezek aukeratuak izan badira, lor izazu, halaber, bezereen NA-a eta izena (ez badira aukeratuak izan, bezeroen bi zutabe horietan NULL atertua beharko litzake).

47. Lor izazu hautazko txango guztien identifikazioa eta zenbat bezerez hautatu duen txango bakoitza. Best bezero baino guxtagoko hautazko txango bakoitzeko, lortu, txangoaren identifikazioaz gain, tako bidaiaren helmuga bera eta egun kopuru berdina izatea.

48. Best bezero baino guxtagoko hautazko txango bakoitzeko, lortu, txangoaren kopurua 0 dela. atetako da, kopurua 0 dela.

### ALL eta ANY

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

32. Lor ezazu Erronara egindako bidaia guztien duten bidaien kodea.

33. Lor ezazu bidaia garestienaren kodea.

34. Lor ezazu honako baldintza hau betetzen duten bidaien kodea: bidaia hautazko txango guztien prezioa bidaiaren eguneko prezioaren bidera edo handiagoa da.

35. Lor ezazu honako baldintza hau betetzen duten bidaien kodea: bidaia hautazko txango guztien prezioa bidaiaren eguneko prezioaren bidera edo handiagoa da. Lor ezazu honako baldintza hau betetzen duten bidaioen NA-a: bezerezak ez zuen parte hartu 2011ko urtarrilaren 1ean Asiala helmuga gisa zeukan bidaia.

### [NOT] UNIQUE

36. Lor ezazu Erronara egindako bidaia bat gidaia bat gidaia duten gidaireni NA-a.

37. Lor ezazu bidaiaren helmuga Paris izanik bidaia bi edo gehiago gidaia dituzten gidaireni NA-a.

### [NOT] EXISTS kabiattuak

38. Lor izazu honako baldintza betetzen duten bezeroen NA-a eta izena: bezerezak hautazko txangoak hartu ditu hotelaren egotza Parisen duten bere bidaia guztietan, zeinetan hotelaren egotza Parisen.

39. Lor izazu honako baldintza hau betetzen duten bezeroen NA-a eta izena: bezerezak ez du inolako hautazko txangoik hartu hotelaren egotza Parisen duten bere bidaia hautazko txangoak hartu bere bidaia. Baldintza beste era batera esanda: bezerezak ez du inolako hautazko txangoik hartu bere bidaiaren bidaia, zeinetan hotelaren egotza Parisen baitago.

### HAVING

40. Lor ezazu 30 bezero baino gehiago dauzkaten hautazko txangoen kodea.

41. Lor ezazu honako baldintza hau betetzen duten bidaien kodea: bidaia hautazko txango guztiek helmuga berbera daukate.

42. Lor ezazu honako baldintza betetzen duten bezeroen NA-a: bezerezak Funchalera egin dituen bi-

43. Bezerrek egin dituzten bidaia guztiek kontuan hartuta, lortu hotel hautatuenaren izena eta esan zenbat aldiiz izan den hautatua.

44. Lor izazu honako baldintza betetzen duten bidaia gidatzten duten gidaireni NA-a eta izena, eta bidaia lortuko hautazko txangoen prezioaren batura: bidaia garestiena izatea, hautazko txangoen prezioen baturari erreparratu.

### BESTE BATZUK

49. Lor ezazu bidaia merkeenaren kodea.

50. Hautazko txangoak eskaintzen dituen bidaia bakoitzeko lor ezazu bidaia hornetan bezero gehien dauzkin hautazko txangoaren kodea.

51. Lor izazu honako baldintza hau betetzen duten bezeroen NA-a eta izena: beti hautatu izan dituzte 3000 euro baino gehiago baloitutu duten bidaiai.

52. Lor izazu honako baldintza hau betetzen duten gidaireni NA-a eta izena: gidaia horrek iraupen luzeoa (50 egun baino gehiago) bidaia guztiak gidaia ditu (gerta liteke bidaia horretaz gain iraupen luzeoa ez den bidaiai bat ere gidaia izana).

53. Lor izazu 2012ko urtarrilaren 1ean Erronara egindako bidaia gidaia zuen gidaireni NA-a eta izena, baldin eta 2012ko otsailaren 1ean Erronara egindako bidaia gidaia ez bazuen.

54. Lor izazu Erronara egindako bidaiai baino gidaia ez dituzten gidaireni NA-a eta izena

55. Lor ezazu bezerrik gabe geratu diren hautazko txangoen identifikazioa.

56. Lor izazu Italiara hitz egiteko gaitasuna izanik Erronara egindako bidaiai bat gidaia ez duten gidaireni NA-a eta izena.

57. Lor izazu honako baldintza betetzen duten bezeroen NA-a eta izena: Pariserako bidaiaiaren bidaia partea hartzea, eta bidaia horretako bakoitzean hiri horretako hotelaren baino ez egotea.

58. Hautazko txangoak bakoitzan bidaia bakoitzeko lortu bidaiaiaren identifikazioa, bidaia hornetan hau-

59. Hautazko txangoak ingelesa hitz egiteko gaitasuna izanik, ez Kanadarako ez Australiarako bidaiaiak gidaia ez duten gidaireni NA-a eta izena (not: ez Kanadarako eta ez Australiarako).

60. Hiru bidaia baino gutxiago gidaia dituen gidaia bakoitzeko, lortu gidaireni NA-a eta gidaia dituen bidaiaiaren kopurua. Gidaranak bidaiaiak gidaia ez badu ere, emaitzan aterako da, kopurua 0 dela.

61. Gidaran artean gutxien hitz egiten den hizkuntza hitz egiten duten gidaireni NA-a eta izena lor izazu.
62. Lor izazu Parisera egindako bidaiaiaren parte hartu ez badute.
63. Lor ezazu guztienak Erronara egindako bidaia guztiak gidaia dituzten gidaireni NA-a eta izena.
64. Lor ezazu honako baldintza betetzen duten bidaia identifikazioa: bidaiai lotutako hautazko txangoen batez besteko prezioa datu-basearen jasota dauden hautazko txango guztien batez besteko prezioa baino handiagoa da. Emaitzan bidaia gidaia dituzten gidaireni izena ere agertuko da.

### 61. Aritlett

Select NA, ~~Hiburte~~ Hizkuntza  
 From Gideri AS G ~~left join~~ Hiburte AS H  
~~where Group By~~  
~~Group By Hizkuntza~~  
~~Count~~  
 Having Count (NA) <= ALL (Select Count (NA))

Select NA, ~~Izena~~ Hizkuntza  
 From Gideri AS Natural Join Hiburte AS H  
~~where NA IN (Select @NA)~~  
~~From~~  
~~Group By NA~~

### 62. Aritlett

Select MA From  
 From Bezero ~~INTER~~ Natural Join Hotel\_Bideia\_Bezero  
~~where HB.B.Helmesa title 'Paris' AND Not Exists (Select \*~~  
~~From Hotel\_Bideia\_Bezero~~  
~~where B.NA = HB.B1.NA AND~~  
~~HB.B1.Helmesa title 'Girona'~~

### 63. Aritlett

Not Exist

Bideia

NA = NA AND Girone

Exist

Bideia

Ertzain

### 64. Aritlett

Select Helmesa, Interaketa, NA  
 From Bideia  
~~where Egunola \* Egunola Prazia > & AVG (Egunola \* Egunola Prazia)~~

$\gamma = \text{AVG} (\text{Egunola} * \text{Egunola Prazia})$   
 $\gamma = \text{AVG} (\text{Egunola} * \text{Egunola Prazia})$   
~~Select AVG (Egunola \* Egunola Prazia)~~  
~~From Bideia~~

#### 64. Archete

Select Henuge, Intercede, BNA, Bore

From Bideie Natural Join Gideri

Where E.Egurde \* Eguale.Predice >= AKG (Egurde \* Eguale.Predice)

### 54. Arikete

Select NA, Isen

From Giderie AS G Natural Join ~~Bidie~~ Bidie AS B

Where B.Helmuge = 'Erone' AND G.NA = B.NA

Except

Select, NA

From Bidie

Where B.Helmuge <> 'Erone' AND G.NA ~~=~~ Bidie.NA

### 55. Arikete

Select Helmuge, Isendate, ~~Zebalio-~~

From Hzebalio-Txango-Bedero AS HTB

Existe

Group By Helmuge, Isendate, ~~Zebalio~~

Having Count(NA) >= ALL (Select Count(NA))

From Hzebalio-Txango-Bedero AS HTB1

Where HTB.Helmuge = HTB1.Helmuge AND  
HTB.Isendate = HTB1.Isendate

Group By Helmuge, Isendate, ~~Zebalio~~)

### 60. Arikete

Select NA, Count(B.Helmuge)

From Giderie AS G left Join Bidie AS B ON G.NA = B.NA

Where Count(Helmuge) < 3

## 51. Aritlect

Select NA, Izene

From (Bidic AS B Natural Join Hotel-Bidic - Bezero AS HBB) Natural Join  
Bezero AS Be;

~~Exists~~ ~~Not Exists~~

where Eguncle \* Egunko Precio > 3000

Except

Select NA

From Bidic Natural Join Hotel-Bidic - Bezero

where Bezero.Eguncle \* Egunko Precio ≤ 3000

## 52. Aritlect

Select NA, Izene

From Gideric AS G

where Not Exists (Select \*)

From Bidic AS B

where B.Eguncle > 50 AND B.NA < G.NA)

Edo

Select NA, Izene

From Gideric AS G

where Not Exists (Select \*)

From Bidic AS B

where B.Eguncle > 50 AND

Not Exists (Select \*)

From Gideric AS G1 Natural Join Bidic AS B1

where G1.NA = B.NA AND B1.Helminge = B.Hel  
AND B1.InteractionDate = B.InteractionDate))

### 56. Arikete

Select NA, Izene

From Giederi ASG Natural Join Hialentza AS H

Where H.Hialentza.Lille = 'Hakier'

AND

~~Not Exists~~ (Select →)

Hekawaga AS H (Select Hekawaga)

From Bidic ASB

Where B.Hekawaga.Lille = 'Bisone' AND G.NA = B.NA

### 57. Arikete

Select NA, Izene

From Bezoar ASB

Where HBB.B.Hekawaga.Lille = 'Paris'

AND ~~Not Exists~~ (Select →)

From Hotel ASB

Where HBB.Hekawaga = Paris AND

B.NA = HBB.NA AND

Not Exists (Select →)

From Hotel

Where HBB.HotelId = All.Hotel.Id AND

Select Hekawaga, InterDate, Zerbikia, NA, HirietParis  
From Hotel ASB

From Hotello\_Bxingo\_Bezero ASO

Group By Hekawaga, InterDate, Zerbikia

Hekawaga Count(NA) >= All.(Count(NA))

From Hotello\_Bxingo\_Bezero AS 1

other O, H = 1..4

Group By Hekawaga, InterDate, Zerbikia

### 58. Arikete

Select Hekawaga, InterDate, Zerbikia, NA, HirietParis  
From Hotel ASB

From Hotello\_Bxingo\_Bezero ASO

Group By Hekawaga, InterDate, Zerbikia

From Hotello\_Bxingo\_Bezero AS 1

other O, H = 1..4

Group By Hekawaga, InterDate, Zerbikia

### 59. ArelNet

Select NA, Tren

From Gideri AS G Natural Join Hilateral AS H

where H.Hilateral Like 'Tigalee' AND

Not Exists (Select \*)

From Bidic AS B

where B.Hilateral  $\leftrightarrow$  'Australia' AND

B.B.NA = B.G.NA)

AND NOT Exist (Select \*)

From Bidic AS B

where B.Hilateral  $\leftrightarrow$  'Kavde' AND

B.NA = G.NA)

### 60. ArelNet

Select NA, Count(Helmege)

From Bidic AS B

Count  
Group By NA

Having  $\leq 3 >=$  (Count(Helmege))

From Bidic AS B

where B.NA = Bl.NA

Group By NA)

Select NA, Count(Helmege)

From Bidic left Join Gideri AS G

where Count(Helmege)  $< 3$