

Relazioni model i SQL

Relacija-osnovna informaciona struktura relacionog modela

- Dva pogleda na bazu :
- Naivni korisnici : tabela
- Projektanti: relacija
- S formalne strane osnovna informaciona struktura relacionog modela jeste relacija.
- U matematičkim terminima relaciju r definišemo kao podskup Dekartovog proizvoda domena atributa tj:

$$r \subset D_1 \times D_2 \times \dots \times D_n$$

gdje su D_i , $i=1..n$, domeni tj. konačni i beskonačni skupovi vrijednosti atributa.

$$\begin{array}{cc} A & B \\ D1 = \{a1, a2\} & D2 = \{b1, b2, b3\} \end{array}$$

$$D1 \times D2 = \{(a1, b1), (a1, b2), (a1, b3), (a2, b1), (a2, b2), (a2, b3)\}$$

$$r \subseteq D1$$

$$r = \{(a1, b1), (a1, b2), (a2, b1), (a2, b3)\}$$

$r(A, B)$ – relacija r sa relacionom šemom (A, B)

a1 b1

a1 b2

a2 b1 - n-torka relacije

a2 b3

Relacija kao tabela

- $r(R)$, $R=A_1,A_2,\dots,A_n$

A1	A2	...	An
v1	v2	...	vn

n-torka

Primjena formalizama u postupku projektovanja relacione baze: algoritmi normalizacije

$r(R)$ - polazna relacija



normalizacija 1NF, 2NF, 3NF



$r_1(R_1) \quad r_2(R_2) \quad \dots \quad r_n(R_n)$

$$R_1 \cup R_2 \cup \dots \cup R_n = R$$

E-R dijagram

relacije

normalizacija

Relacioni model

isporučilac (IME_I, ADRESA, #ARTIKLA, CIJENA)

- dupliciranje informacija
- anomalija ažuriranja
- anomalija unošenja
- anomalija brisanja

isporučilac (IME_I, ADRESA)

isporučuje (IME_I, #ARTIKLA, CIJENA)

isporučilac (IME_I, ADRESA, #ARTIKLA, CIJENA)

ALS	XXX	123	15
ALS	XXX	135	40
ALS	XXX	143	20
ALS	XXX	167	35
MNV	YYY	123	14
MNV	YYY	135	45
...

isporučilac (IME_I, ADRESA, #ARTIKLA, CIJENA)

ALS	XXX	123	15
ALS	ZZZ	135	40
ALS	XXX	143	20
ALS	XXX	167	35
MNV	YYY	123	14
MNV	YYY	135	45
...

isporučilac (IME_I, ADRESA, #ARTIKLA, CIJENA)

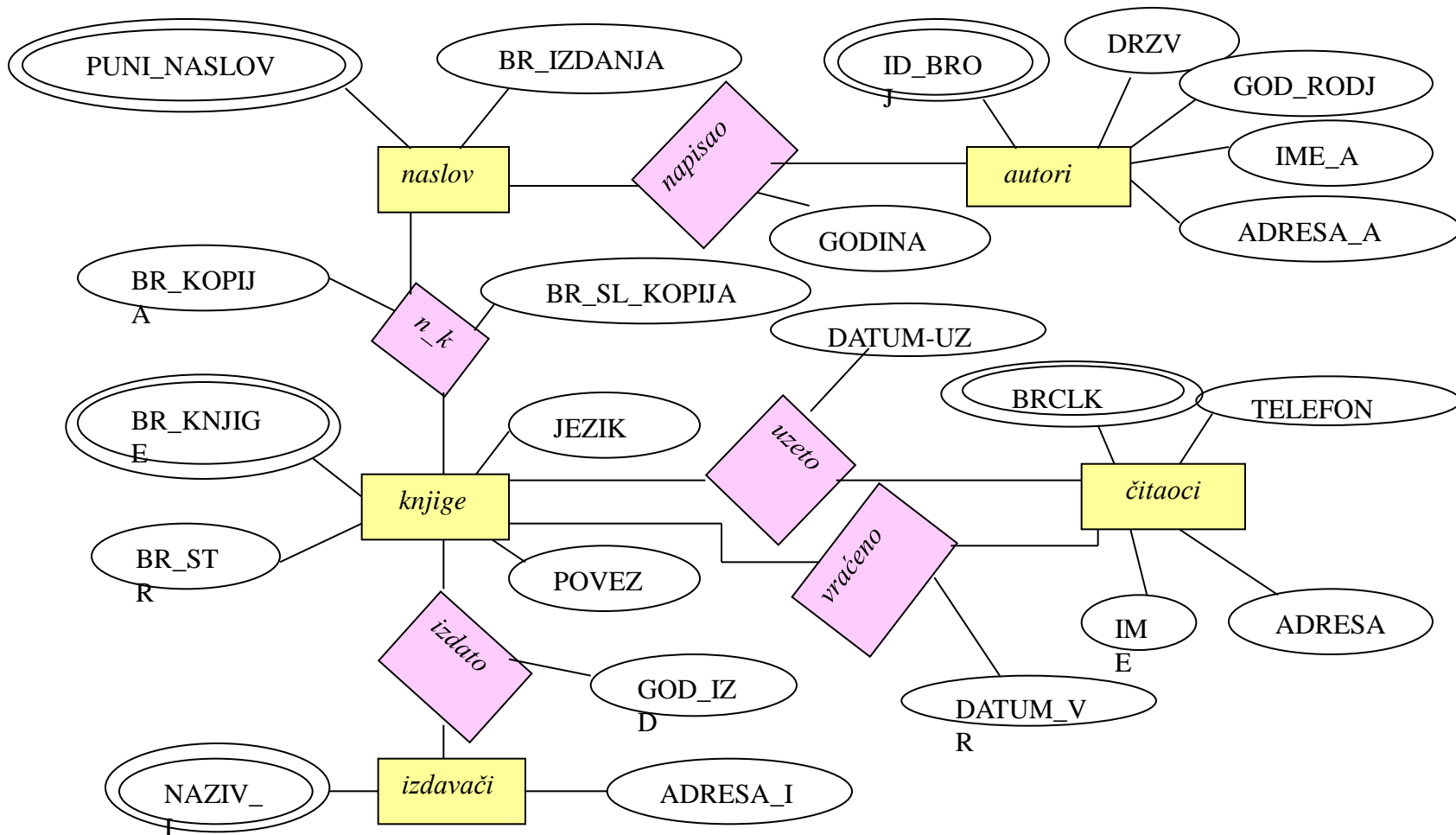
ALS	XXX	123	15
ALS	XXX	135	40
ALS	XXX	143	20
ALS	XXX	167	35
MNV	YYY	123	14
MNV	YYY	135	45
...

SCD **SSS** **?**

isporučilac (IME_I, ADRESA, #ARTIKLA, CIJENA)

MNV	YYY	123	14
MNV	YYY	135	45
...
SCD	SSS	?	

Primjer Biblioteka



Primjer *Biblioteka*

naslovi(PUNI NASLOV, BR_IZDANJA)

autori(ID BROJ, IME_A, ADRESA_A,
DRZV,GOD_RODJ)

napisao(ID BROJ, PUNI NASLOV, GODINA)

knjige(BR KNJIGE, BR_STR, POVEZ, JEZIK)

n-k(PUNI NASLOV, BROJ KNJIGE, BROJ_KOPIJA,
BR_SL_KOPIJA)

čitaoci(BRCL, IME, ADRESA, TELEFON)

uzeto(BR KNJIGE, BRCL,DATUM_UZ, VRIJEME)

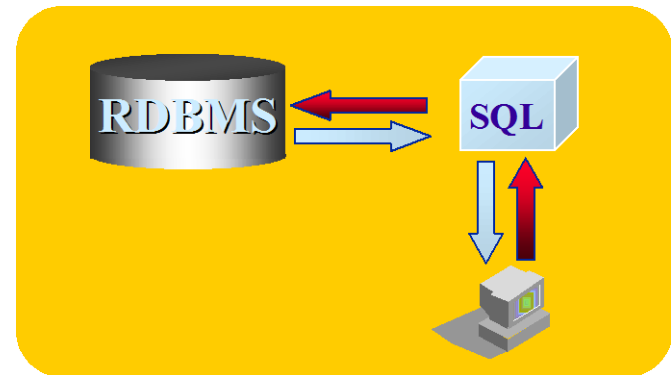
vraćeno(BR KNJIGE, BRCL, DATUM_VR)

izdavači(NAZIV_I, ADRESA_I)

izdato(BR KNJIGE, NAZIV_I, GOD_IZD)

SQL (Structured Query Language)

- SQL je standardni upitni jezik relacionih SUBP.
- **DDL** (Data Definition Language)
- **DML** (Data Manipulation Language)
 - pretraživanje podataka
 - ažuriranje podataka



Primjeri DDL naredbi

```
CREATE TABLE Customer  
(SID integer PRIMARY KEY,  
Last_Name varchar(30),  
First_Name varchar(30));
```

```
ALTER table customer ADD INDEX  
IDX_First_Name (First_Name);
```

```
DROP TABLE customer;
```

Pretraživanje podataka

```
SELECT A1, A2, ..., An  
FROM r1, r2, ..., rk  
WHERE P
```

$\Pi_{A_1, A_2, \dots, A_n} (\sigma_P(r_1 \times r_2 \times \dots \times r_k))$ - relaciona algebra

SELECT klauzula

U SELECT klauzuli navodi se **spisak atributa** rezultirajuće relacije koje treba izlistati.

SELECT klauzula može da sadrži i neke **specijalne funkcije** kao što su:

- * koja zamjenjuje spisak svih atributa rezultirajuće relacije;
- **DISINCT** koja selektuje samo različite n-torke rezultujuće relacije
- Derivacione funkcije, kao što su: **SUM, MAX, MIN, AVG i COUNT.**

FROM klauzula

U FROM klauzuli specificiraju se **relacije** koje učestvuju u upitu, a može biti navedena i jedna ili više promjenljivih tipa n-torke. Ove promjenljive se nazivaju često **nadimci** relacije.

WHERE klauzula

Pod izrazom WHERE specificira se **selekcija n-torki** rezultujuće relacije koje zadovoljavaju **predikat P**.

U predikatu P figurišu:

- **Atributi** relacija iz podizraza FROM;
- **Konstante** iz domena tih atributa;
- **Relacije** ili **njihovi nadimci** iz podizraza FROM;

- Znaci poređenja iz skupa: {=,<,>,<>,<=,>=}
- Operacije poređenja alfanumerika kao što su: **{LIKE, CONTAINING, STARTING WITH};**
- Logički operatori iz skupa **{AND, OR, NOT};**

primjer predikata

student.GOD_STUDIJA < 3 AND student.IME STARTING WITH "R"

relacija

atribut

znak
poređenja

konstanta
iz domena
atributa

logički
operator

- Operator pripadnosti skupu **IN** (\in);
- Operator podskupa **CONTAINS** (\subseteq);
- Operatori **NOT IN** i **NOT CONTAINS**;
- Operatori za ispitivanje nula polja: **IS NULL, IS NOT NULL**;

- Konstrukcije **ANY** i **ALL** koje omogućavaju da se jedna vrijednost uporedi sa bilo kojom ili sa svim vrijednostima iz nekog skupa;
- Operator pripadnosti intervalu **BETWEEN**;
- Tačka, otvorena mala i zatvorena mala zagrada { ., (,) }

Osim ovog, SQL koristi i ključne riječi **UNION**, **INTERSECT** i **MINUS** za skupovne operacije unija, presjek i razlika.

Primjer *Banka*

stranka (BR_LK, IME_S, GRAD)

poslovnica (IME_P, GRAD_P)

stедnja (BR_LK, BR_RAC, IME_P,
STANJE)

zajam (BR_LK, BR_ZAJMA, IME_P,
IZNOS)

Primjer *Banka*

1. Naći ukupan iznos uloga štediše Jovanović Marka.

```
SELECT SUM(STANJE)
FROM stednja U, stranka S
WHERE U.BR_LK=S.BR_LK AND
S.IME_S="Jovanović Marko";
```

Primjer *Banka*

2. Naći imena svih stranaka koje štede u ovoj banci.

```
SELECT DISTINCT IME_S  
FROM stranka X, stednja Y  
WHERE X.BR_LK=Y.BR_LK;
```


Primjer *Banka*

3. Naći broj štediša ove banke.

```
SELECT COUNT (DISTINCT BR_LK)  
FROM stednja;
```

4. Naći najveći zajam koji je banka dala.

```
SELECT MAX(IZNOS)  
FROM zajam;
```

Primjer *Banka*

5. Naći minimalno stanje na nekom računu poslovnice Blok V.

```
SELECT MIN(STANJE)  
FROM stednja  
WHERE IME_P="Blok V";
```

Primjer *Banka*

6. Naći imena poslovnica koje imaju štediše iz Podgorice.

```
SELECT DISTINCT IME_P  
FROM stranka, stednja  
WHERE stranka.BR_LK=stednja.BR_LK  
AND stranka.GRAD_S="Podgorica";
```

Primjer *Banka*

7. Naći sve štediše čiji broj računa ima prefiks 12/3-.

```
SELECT DISTINCT BR_LK
```

```
FROM stednja
```

```
WHERE BR_RAC STARTING WITH  
"12/3-";
```

Primjer *Banka*

8. Naći sve stranke kojima grad u kojem žive nije evidentiran.

```
SELECT BR_LK, IME_S  
FROM stranka  
WHERE GRAD_S IS NULL;
```

Primjer *Banka*

9. Naći sve štediša čiji uložci su između 20.000 i 30.000.

```
SELECT BR_LK  
FROM stednja  
WHERE STANJE BETWEEN 20.000  
AND 30.000;
```

Primjer *Banka*

10. Naći one štediše koji nemaju uloge u iznosu 20.000 i 30.000.

```
SELECT BR_LK  
FROM stednja  
WHERE STANJE NOT IN (20.000,  
30.000);
```

Ažuriranje podataka

1. Unošenje podataka

```
INSERT INTO naziv_relacije  
[(nazivi_atributa)]  
VALUES (spisak_vrijednosti);
```


Primjer *Banka*

- Stari štediša sa brojem lične karte 121 je u poslovnici Blok V uložio 45000 dinara, na računu broj 12/3-003. Ažurirati bazu podataka.

```
INSERT INTO stednja ( BR_LK,  
BR_RAC, IME_P, STANJE)  
VALUES ( 121, “12/3-003”, “Blok V”,  
45.000 );
```

Ažuriranje podataka

2. Brisanje podataka

```
DELETE FROM naziv_relacije  
WHERE uslov_selekcije;
```

Primjer *Banka*

- Štediša sa brojem lične karte 353 je ugasio račun broj 12/4-222.

```
DELETE FROM stednja  
WHERE BR_LK=353 AND  
BR_RAC="12/4-222";
```

Ažuriranje podataka

3. Ispravka podataka

UPDATE naziv_relacije

SET

(naziv_atributa=vrijednost_atributa,...)

WHERE uslov_selekcije;

Primjer *Banka*

- Stranka sa brojem lične karte 323 se preselila u Podgoricu. Ažurirati bazu podataka.

```
UPDATE stranka  
SET GRAD_S="Podgorica"  
WHERE BR_LK=323;
```

Primjer *Studentska služba*

student (BR_IND, IME, PREZIME,
KUĆNI BROJ, TELEFON)

predmet (SIF_PR, NAZIV_PR)

prijave (BR_IND, SIF_PR, DATUM,
OCJENA)

Primjer *Studentska služba*

1. Prikaži sve podatke o studentima koji se zovu Jovan ili Luka, a stanuju u Jerevanskoj ulici.

```
SELECT *  
FROM student  
WHERE (IME="Jovan" OR IME="Luka")  
AND ULICA="Jerevanska";
```

Primjer *Studentska služba*

2. Prikaži ime, prezime i adresu studenata koji stanuju u ulicama čiji nazivi sadrže riječ “knez”.

```
SELECT  IME,      PREZIME      ULICA,  
        KUCNI_BROJ  
FROM studenti  
WHERE ULICA CONTAINING “knez”;
```


Primjer *Studentska služba*

3. Navedi sve podatke o studentima čije prezime počinje sa Jov, a stanuju u Knez Mihajlovoj ulici.

SELECT *

FROM studenti

WHERE PREZIME STARTING WITH
"Jov" AND

ULICA="Knez Mihajlova";

Primjer *Studentska služba*

4. Prikazati sve podatke o predmetima čiji naziv počinje riječju teorija.

```
SELECT *
```

```
FROM predmeti
```

```
WHERE NAZIV_PR LIKE "teorija%";
```

Primjer *Studentska služba*

5. Prikaži sve podatke o studentima čije prezime počinje sa Hadži, a završava se saić.

```
SELECT *
```

```
FROM studenti
```

```
WHERE PREZIME LIKE "Hadzi%ic";
```

Primjer *Studentska služba*

6. Prikaži imena svih studenata koji nemaju telefon.

```
SELECT IME, PREZIME  
FROM studenti  
WHERE TELEFON IS NULL;
```

Primjer *Studentska služba*

7. Izračunati srednju ocjenu studenata na ispitu iz Baza podataka u januarском roku.

```
SELECT AVG (OCJENA)
FROM prijave
WHERE SIF_PR="4BP" AND
      DATUM="januar10";
```

Primjer *Studentska služba*

8. Izračunati ukupan broj kandidata na ovom ispitu.

```
SELECT COUNT (*)  
FROM prijave  
WHERE SIF_PR="4BP" AND  
       DATUM="januar10";
```

Primjer *Studentska služba*

9. Prikaži ime, prezime i broj indeksa studenata koji su položili neki ispit u januarskom roku.

```
SELECT IME, PREZIME, BR_IND
```

```
FROM studenti
```

```
WHERE BR_IND=ANY
```

```
(SELECT BR_IND
```

```
FROM prijave
```

```
WHERE OCJENA>5 AND
```

```
DATUM='januar10';
```

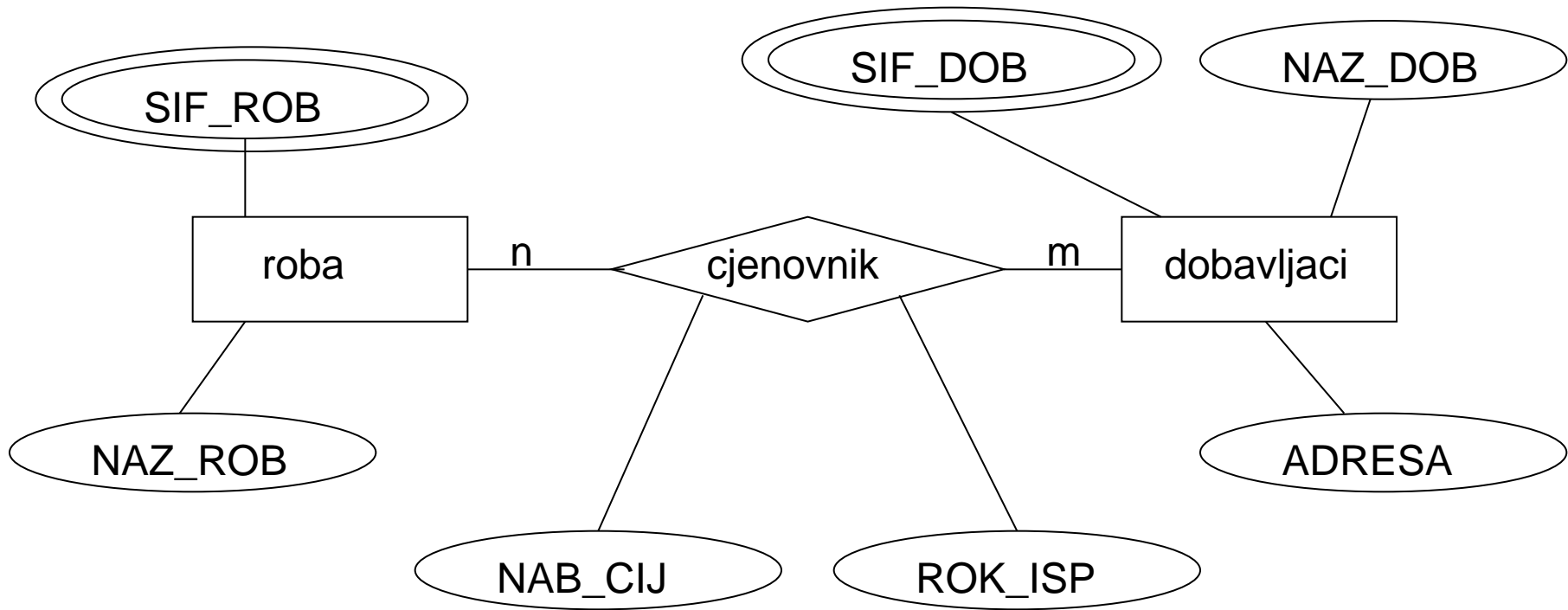
E-R dijagrami i SQL

Zadaci

1. Dio baze podataka nabavke jednog preduzeća treba da sadrži sljedeće podatke:

- Šifra robe
- Naziv robe
- Šifra dobavljača
- Naziv dobavljača
- Adresa
- Nabavna Cijena
- Rok isporuke

Sastaviti odgovarajući E-R dijagram i predstaviti tabelama.



roba

<u>SIF_ROB</u>	NAZ_ROB
123	Šećer u kristalu
124	So morska

cjenovnik

<u>SIF_ROB</u>	<u>SIF_DOB</u>	NAB_CIJ	ROK_ISP
123	1/223	1.33	2
124	1/225	0.75	4

dobavljači

<u>SIF_DOB</u>	NAZ_DOB	ADRESA
1/223	Mex	Pg
1/225	Fenix	Pg

2. Putem SQL-a izraziti sljedeće upite:

- a) Naći naziv i nabavnu cijenu robe koju dobavljač *Mex* isporučuje za vrijeme kraće od 3 dana.
- b) Naći koliko različitih roba se dobavlja od dobavljača *Fenix*.

a)

```
SELECT  NAZ_ROB, NAB_CIJ
FROM    roba R, cjenovnik C, dobavljac D
WHERE   R.SIF_ROB = C.SIF_ROB AND
        D.SIF_DOB = C.SIF_DOB AND
        D.NAZ_DOB = "Mex" AND
        C.ROK_ISP < 3;
```

b)

```
SELECT COUNT (DISTINCT SIF_ROB)
FROM   dobavljac D, cjenovnik C
WHERE  D.SIF_DOB = C.SIF_DOB AND
       D.NAZ_DOB = "Fenix";
```