

Proiectarea unei aplicatii cu baze de date

Specificatii produs (interfata utilizator, fluxuri de functionare, conditii functionale);

Documentatie de proiectare (de nivel inalt, detaliata);

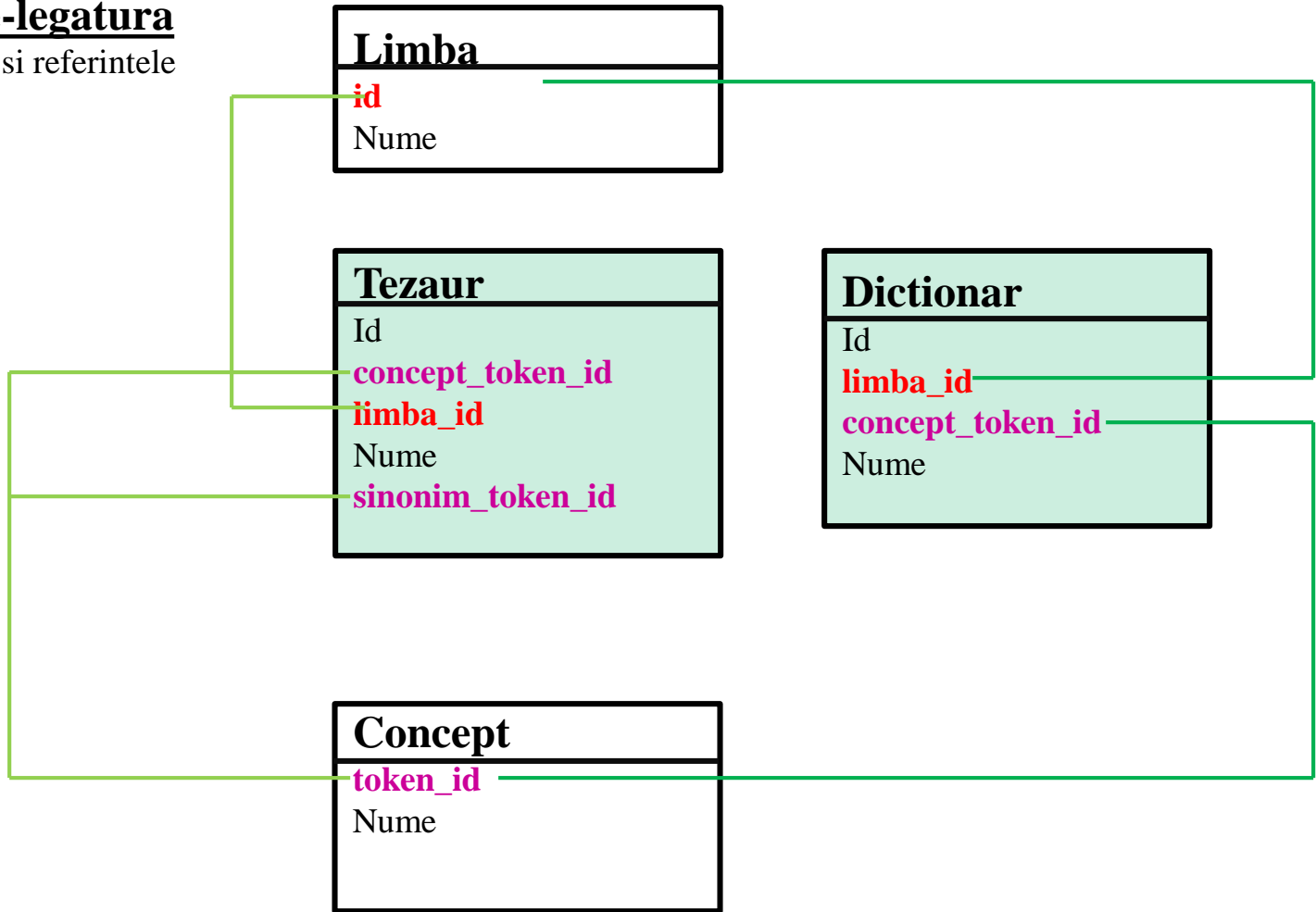
Implementarea solutiei software; validarea si asigurarea calitatii;

Mentenabilitate, robustete, flexibilitate

Specificatie produs: baza de date

Diagrama entitate-legatura

Se precizeaza campurile si referintele



Implementarea solutiei software

Definirea tabelelor

In tabela 'limba' sunt grupate limbile in care sunt disponibile traduceri; tabela 'concept' cuprinde toate conceptele (sau notinunile) pentru care exista traduceri in diferite limbi; in tabela 'tezaur' sunt grupate, in jurul unui concept dintr-o anume limba, acele notiuni care sunt, intr-un grad mai mare sau mai mic, sinonime cu acesta; cum aria semantica a unui termen poate diferi de la o limba la alta, sinonimele nu sunt universal valabile, ci numai pentru o anume limba.

In tabela 'dictionar' sunt furnizate traduceri pentru toate conceptele, in toate limbile disponibile.

E nevoie sa se adauge automat, la adaugarea unei limbi sau a unui nou concept, inregistrari in tabela 'dictionar'; acest proces este automatizat folosind doua triggere (declansatoare).

```
CREATE TABLE IF NOT EXISTS limba
(id int unique auto_increment primary key,
nume char(20));
```

```
CREATE TABLE IF NOT EXISTS concept
(token_id char(20) unique primary key,
descriere char(50));
```

```
CREATE TABLE IF NOT EXISTS tezaur
(id int unique auto_increment primary key,
concept_token_id char(20),
limba_id int,
sinonim_token_id char(20),
FOREIGN KEY (concept_token_id) REFERENCES concept(token_id),
FOREIGN KEY (limba_id) REFERENCES limba(id),
FOREIGN KEY (sinonim_token_id) REFERENCES concept(token_id));
```

```
CREATE TABLE IF NOT EXISTS dictionar
(id int unique auto_increment primary key,
limba_id int,
concept_token_id char(20),
nume char(30),
FOREIGN KEY (limba_id) REFERENCES limba(id),
FOREIGN KEY (concept_token_id) REFERENCES concept(token_id));
```

Implementarea solutiei software

```
## - Trigger AFTER INSERT tabela concept
## Este folosit pentru a adauga automat traduceri in dictionar, dupa fiecare inserare a unui concept
## Pentru fiecare limba existenta, se adauga cate o traducere; traducerea va contine numai un cod
## care indica faptul ca acel termen nu e inca tradus
DROP PROCEDURE IF EXISTS CURSOR_LIMBA;

DELIMITER //

CREATE PROCEDURE CURSOR_LIMBA(new_token_id char(20))

BEGIN

    DECLARE done INT DEFAULT 0;
    DECLARE nId INT;
    DECLARE cur1 CURSOR FOR SELECT id FROM limba;
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

    OPEN cur1;

    REPEAT
        FETCH cur1 INTO nId;
        IF NOT done THEN
            INSERT INTO dictionar (limba_id, concept_token_id, nume)
            VALUES(nId, new_token_id, '<enter text>');
        END IF;

    UNTIL done END REPEAT;

    CLOSE cur1;

END //
DELIMITER ;

DROP TRIGGER IF EXISTS insert_trad_dictionar;

delimiter //

CREATE TRIGGER insert_trad_dictionar AFTER INSERT ON concept

FOR EACH ROW BEGIN
    CALL CURSOR_LIMBA(NEW.token_id);
END; //

delimiter ;
```

Acest trigger se utilizeaza pentru a adauga inregistrari in tabela dictionar la adaugarea unui nou concept, cate o inregistrare pentru fiecare limba definita in tabela 'limba'.

Procedura implementeaza cursorul pentru tabela 'limba'; se parcurg toate inregistrarile tablei, si pentru fiecare inregistrare se adauga o noua inregistrare in tabela 'dictionar'. In campul 'nume' unde se adauga traducerea propriu-zisa se introduce un cod cu semnificatia 'traducere inexistentă'.

Trigger-ul apeleaza procedura CURSOR_LIMBA.

Implementarea solutiei software

```
##-----  
DROP PROCEDURE IF EXISTS CURSOR_CONCEPT;  
  
DELIMITER //  
  
CREATE PROCEDURE CURSOR_CONCEPT(new_limba_id INT)  
  
BEGIN  
  
    DECLARE done INT DEFAULT 0;  
    DECLARE sId CHAR(20);  
    DECLARE cur2 CURSOR FOR SELECT token_id FROM concept;  
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;  
  
    OPEN cur2;  
  
    REPEAT  
        FETCH cur2 INTO sId;  
        IF NOT done THEN  
            INSERT INTO dictionar (limba_id, concept_token_id, nume)  
            VALUES(new_limba_id, sId, '<enter text>');  
        END IF;  
  
    UNTIL done END REPEAT;  
  
    CLOSE cur2;  
  
END //  
DELIMITER ;  
  
## - Trigger AFTER INSERT tabela limba  
DROP TRIGGER IF EXISTS insert_limba_dictionar;  
  
delimiter //  
  
CREATE TRIGGER insert_limba_dictionar AFTER INSERT ON limba  
  
FOR EACH ROW BEGIN  
    CALL CURSOR_CONCEPT(NEW.id);  
END; //  
  
delimiter ;
```

Acest trigger se utilizeaza pentru a adauga inregistrari in tabela 'dictionar' la adaugarea unei noi limbi, cate o inregistrare pentru fiecare concept definit in tabela 'concept'.

Procedura implementeaza cursorul pentru tabela 'concept'; se parcurg toate inregistrarile tabelii, si pentru fiecare inregistrare se adauga o noua inregistrare in tabela 'dictionar'. In campul 'nume' unde se adauga traducerea propriu-zisa se introduce un cod cu semnificatia 'traducere inexistentă'.

Trigger-ul apeleaza procedura CURSOR_CONCEPT.

Implementarea solutiei software

```
# Verificare exista traducere din limba sursa in limba destinatie pentru termenul 'denumire'.
DROP PROCEDURE IF EXISTS EXISTA_TRADUCERE;

DELIMITER //

CREATE PROCEDURE EXISTA_TRADUCERE(traducere_din_limba char(20), traducere_in_limba char(20), denumire char(30))
BEGIN
    SET @limba_din_id = NULL, @limba_in_id = NULL, @ctid = NULL, @nume_traducere = NULL;

    SELECT @limba_din_id := id FROM limba WHERE nume = traducere_din_limba;
    SELECT @limba_in_id := id FROM limba WHERE nume = traducere_in_limba;

    SELECT @ctid := concept_token_id from dictionar
    WHERE nume = denumire AND limba_id = @limba_din_id;

    IF @ctid IS NULL THEN
        SELECT CONCAT('Nu exista cuvantul ',denumire, ' in ',traducere_din_limba,')
        AS Rezultat;
    ELSE
        BEGIN
            SELECT @nume_traducere := nume FROM dictionar
            WHERE concept_token_id = @ctid AND limba_id = @limba_in_id;

            IF @nume_traducere <> '<enter text>' THEN
                SELECT CONCAT('Traducerea ',denumire,' in ',traducere_in_limba, ' este:',@nume_traducere,')
                AS Rezultat;
            ELSE SELECT CONCAT('Nu exista traducerea ',denumire, ' in limba ',traducere_in_limba,')
            AS Rezultat;
            END IF;
        END;
    END IF;

END //
DELIMITER ;
```

Procedura verifica daca exista traducere din limba sursa in limba destinatie.

Daca exista traducerea, aceasta este returnata.

Daca nu exista cuvantul, in limba sursa, se returneaza un tip de eroare.

Daca exista cuvantul dar nu exista o traducere in limba destinatie, pentru acel cuvant (furnizat ca parametru de intrare pentru procedura) se returneaza alt tip de eroare.

Implementarea solutiei software

```
# Adauga traducere din limba sursa in limba destinatie pentru termenul 'denumire'.
DROP PROCEDURE IF EXISTS ADAUGA_TRADUCERE;

DELIMITER //

CREATE PROCEDURE ADAUGA_TRADUCERE(traducere_din_limba char(20), traducere_in_limba char(20),
    denumire_sursa char(30), denumire_destinatie char(30))
BEGIN
    SET @limba_din_id = NULL, @limba_in_id = NULL, @ctid = NULL, @nume_traducere = NULL;

    SELECT @limba_din_id := id FROM limba WHERE nume = traducere_din_limba;
    SELECT @limba_in_id := id FROM limba WHERE nume = traducere_in_limba;

    SELECT @ctid := concept_token_id from dictionar
    WHERE nume = denumire_sursa AND limba_id = @limba_din_id;

    IF @ctid IS NULL THEN
        SELECT CONCAT('Nu exista cuvantul ',denumire_sursa, ' in ',traducere_din_limba,',')
        AS Rezultat;
    ELSE
        BEGIN
            SELECT @nume_traducere := nume FROM dictionar
            WHERE concept_token_id = @ctid AND limba_id = @limba_in_id;

            IF (@nume_traducere = '<enter text>') THEN
                UPDATE dictionar SET nume = denumire_destinatie
                WHERE concept_token_id = @ctid AND limba_id = @limba_in_id;

            ELSE SELECT CONCAT('Traducerea deja existenta. Operatie abandonata.') As Rezultat;
            END IF;

        END;
    END IF;

END //
DELIMITER ;
```

Procedura adauga o traducere din limba sursa in limba destinatie pentru un cuvnt.

Daca nu exista cuvantul, in limba sursa, se returneaza un tip de eroare.

Daca exista cuvantul dar exista o traducere in limba destinatie, pentru acel cuvnt (furnizat ca parametru de intrare pentru procedura) se returneaza alt tip de eroare.

Altfel, se adauga traducerea.

Implementarea solutiei software

```
## Adauga un concept nou in tezaur; se introduc ca parametri traducerile si nu conceptele
DROP PROCEDURE IF EXISTS ADAUGA_SINONIM;

DELIMITER //

CREATE PROCEDURE ADAUGA_SINONIM(limba_ume char(20), denumire_concept char(30), denumire_sinonim char(30))
BEGIN
    SET @limbaid = NULL, @concept_id = NULL, @sinonim_id = NULL;

    SELECT @limbaid := id FROM limba WHERE nume = limba_ume;

    SELECT @concept_id := concept_token_id from dictionar
    WHERE nume = denumire_concept AND limba_id = @limbaid;

    SELECT @sinonim_id := concept_token_id from dictionar
    WHERE nume = denumire_sinonim AND limba_id = @limbaid;

    IF @concept_id IS NULL THEN SELECT CONCAT('Nu exista cuvantul ',denumire_concept, ' in ',limba_ume, '!')
        AS Rezultat;
    END IF;
    IF @sinonim_id IS NULL THEN SELECT CONCAT('Nu exista cuvantul ',denumire_sinonim, ' in ',limba_ume, '!')
        AS Rezultat;
    END IF;
    IF @concept_id IS NOT NULL AND @sinonim_id IS NOT NULL THEN
        BEGIN
            ## Adaugam inregistrari doar daca nu exista deja
            SET @selectat = NULL;
            SELECT @selectat := id FROM tezaur WHERE concept_token_id = @concept_id AND sinonim_token_id = @sinonim_id
            AND limba_id = @limbaid;
            IF @selectat IS NULL THEN
                BEGIN
                    INSERT INTO tezaur (concept_token_id, limba_id, sinonim_token_id)
                    VALUES (@concept_id,@limbaid, @sinonim_id);
                END;
            END IF;
        END;
    END IF;
END IF;
```

Procedura adauga un sinonim nou in tezaur, pentru un concept deja existent. Ca parametru de intrare se introduce traducerea intr-o anumita limba a conceptului respectiv.

Daca nu exista cuvantul traducere a conceptului, in limba respectiva, se returneaza un tip de eroare.

Daca nu exista cuvantul traducere a sinonimului, in limba respectiva, se returneaza alta eroare.

Altfel, se va adauga sinonimul daca, bineinteles, acest sinonim nu este deja adaugat.