Proiect Securitatea Bazelor de date

**Administrarea activității angajaților unei companii**

Student: Stancu Adriana, grupa 510

# 1.Introducere

## a) Modelul proiectat și regulile sale

Sistemul de administrare a activității angajaților are ca scop monitorizarea programului de lucru, învoirilor, concediilor și locațiilor în care își desfășoară activitatea angajații unei companii.

Angajatii pot depune cereri de concediu sau cereri de învoire. În cazul cererilor de concediu, angajatul trebuie să specifice un înlocuitor. În cazul cererilor de învoire, orele vor trebui recuperate. Ambele tipuri de cereri trebuie aprobate de managerul angajatului și trebuie să facă referire la unul din motivele standard de concediu (medical, odihnă, studii, etc.) sau de învoire (boală, probleme familiale, etc). Fiecare angajat are un manager cu excepția proprietarilor și administratorilor companiei.

În funcție de departamentul din care fac parte, angajații au diferite roluri care le oferă anumite privilegii. Datele de salarizare pot fi modificate doar de proprietarii firmei și de administratori. Aceste date sunt folosite de către departamentul de resurse umane pentru a genera o situație zilnică și o situație lunară a activității fiecărui angajat. Situațiile zilnice sunt utilizate pentru a contoriza orele lucrate și locațiile în care angajații își desfășoară activitatea. Situațiile lunare sunt utilizate pentru calculul salariului, bonurilor de masă și zilelor de concediu rămase. Datele de conectare și datele de salarizare ale fiecărui angajat vor fi criptate cu chei diferite.

## b) Diagramele bazei de date

Diagrama Entitate-Relație

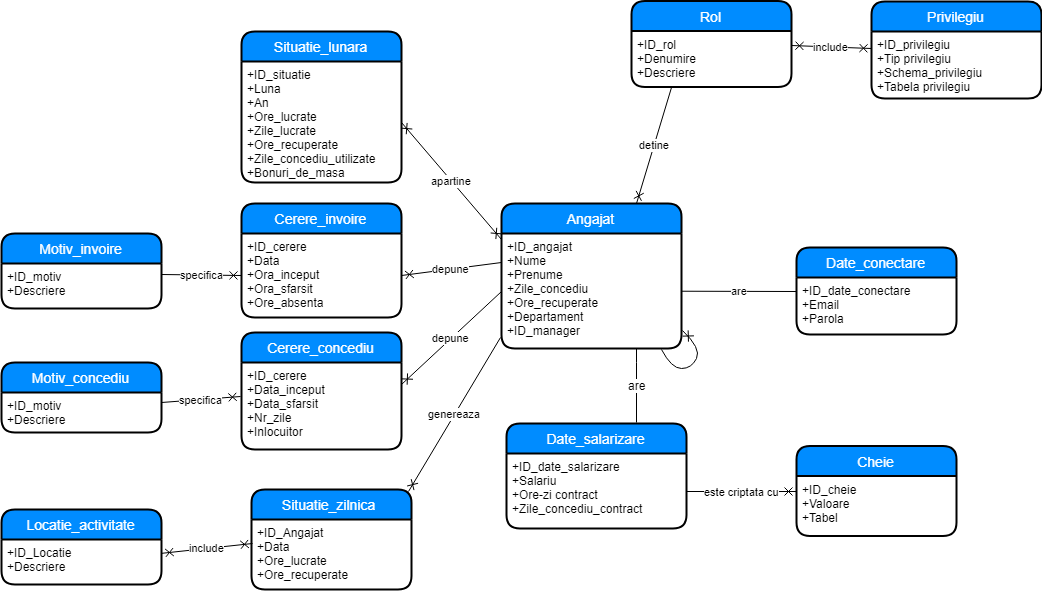
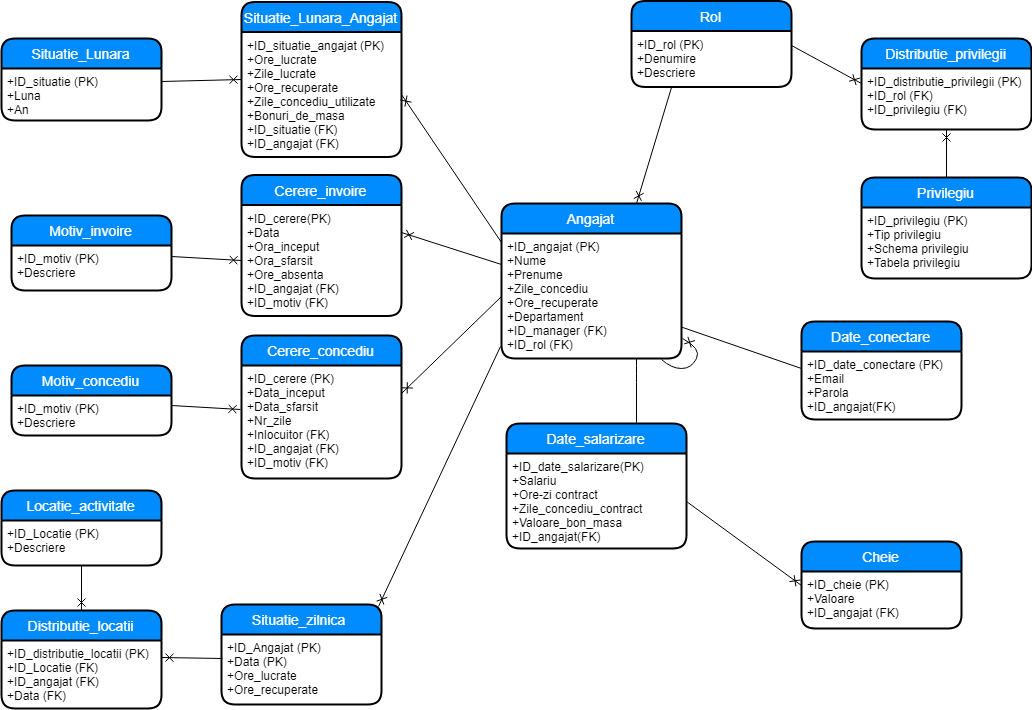


Diagrama Conceptuală



## c) Schemele relaționale

**SITUATIE\_LUNARA** (ID\_SITUATIE#, LUNA, AN);

**SITUATIE\_LUNARA\_ANGAJAT** (ID\_SITUATIE\_ANGAJAT#, ID\_SITUATIE#, ID\_ANGAJAT#, ORE\_LUCRATE, ZILE\_LUCRATE, ORE\_RECUPERATE, ZILE\_CONCEDIU\_UTILIZATE, BONURI\_DE\_MASA);

**ROL** (ID\_ROL#, DENUMIRE, DESCRIERE);

**DISTRIBUTIE\_PRIVILEGII** (ID\_DISTRIBUTIE\_PRIVILEGII#, ID\_ROL#, ID\_PRIVILEGIU#);

**PRIVILEGIU** (ID\_PRIVILEGIU#, TIP PRIVILEGIU, TABELA PRIVILEGIU);

**LOCATIE\_ACTIVITATE** (ID\_LOCATIE#, DESCRIERE);

**DISTRIBUTIE\_LOCATII** (ID\_DISTRIBUTIE\_LOCATII#, ID\_LOCATIE# , ID\_ANGAJAT#, DATA#);

**SITUATIE\_ZILNICA** (ID\_ANGAJAT#, DATA#, ORE\_LUCRATE, ORE\_RECUPERATE);

**MOTIV\_CONCEDIU** (ID\_MOTIV#, DESCRIERE);

**CERERE\_CONCEDIU** (ID\_CERERE#, INLOCUITOR#, ID\_MOTIV#, ID\_ANGAJAT, DATA\_INCEPUT, DATA\_SFARSIT, NR\_ZILE);

**MOTIV\_INVOIRE** (ID\_MOTIV#, DESCRIERE);

**CERERE\_INVOIRE** (ID\_CERERE#, ID\_ANGAJAT#, ID\_MOTIV#, DATA, ORA\_INCEPUT, ORA\_SFARSIT, ORE\_ABSENTA);

**DATE\_SALARIZARE** (ID\_DATE\_SALARIZARE#, ID\_ANGAJAT#, SALARIU, ORE-ZI CONTRACT, ZILE\_CONCEDIU\_CONTRACT, VALOARE\_BON\_MASA);

**DATE\_CONECTARE** (ID\_DATE\_CONECTARE#, ID\_ANGAJAT#, EMAIL, PAROLA);

**CHEIE** (ID\_CHEIE#, ID\_ANGAJAT#, VALOARE, TABEL);

**ANGAJAT** (ID\_ANGAJAT#, ID\_MANAGER#, ID\_ROL#, NUME, PRENUME, ZILE\_CONCEDIU, ORE\_RECUPERATE, DEPARTAMENT);

## d) Crearea tabelelor

Scripturile pentru crearea tabelelor sunt următoarele:

CREATE TABLE SITUATIE\_LUNARA (

id\_situatie number(4) PRIMARY KEY,

luna number(4),

an number(4));

CREATE TABLE MOTIV\_INVOIRE (

id\_motiv number(4) PRIMARY KEY,

descriere varchar2(200));

CREATE TABLE MOTIV\_CONCEDIU (

id\_motiv number(4) PRIMARY KEY,

descriere varchar2(200));

CREATE TABLE LOCATIE\_ACTIVITATE (

id\_locatie number(4) PRIMARY KEY,

descriere varchar2(200));

CREATE TABLE ROL (

id\_rol number(4) PRIMARY KEY,

denumire varchar2(200),

descriere varchar2(200));

CREATE TABLE PRIVILEGIU (

id\_privilegiu number(4) PRIMARY KEY,

tip\_privilegiu varchar2(50),

tabela\_privilegiu varchar2(50));

CREATE TABLE DISTRIBUTIE\_PRIVILEGII (

id\_distributie\_privilegiu number(4) PRIMARY KEY,

id\_rol number(4),

id\_privilegiu number(4),

CONSTRAINT fk\_distributie\_privilegii\_rol

FOREIGN KEY (id\_rol) REFERENCES rol(id\_rol),

CONSTRAINT fk\_distributie\_privilegii\_priv

FOREIGN KEY (id\_privilegiu) REFERENCES privilegiu(id\_privilegiu));

CREATE TABLE ANGAJAT (

id\_angajat number(4) PRIMARY KEY,

nume varchar2(100),

prenume varchar2(100),

zile\_concediu number(4),

ore\_recuperate number(4),

departament varchar2(100),

id\_manager number(4),

id\_rol number(4),

CONSTRAINT fk\_angajat\_manager

FOREIGN KEY (id\_manager) REFERENCES angajat(id\_angajat),

CONSTRAINT fk\_angajat\_rol

FOREIGN KEY (id\_rol) REFERENCES rol(id\_rol));

CREATE TABLE DATE\_CONECTARE (

id\_date\_conectare number(4) PRIMARY KEY,

email varchar2(200),

parola varchar2(200),

id\_angajat number(4),

CONSTRAINT fk\_date\_conectare\_angajat

FOREIGN KEY (id\_angajat) REFERENCES angajat(id\_angajat));

CREATE TABLE DATE\_SALARIZARE (

id\_date\_salarizare number(4) PRIMARY KEY,

salariu number(8),

ore\_zi\_contract number(4),

zile\_concediu\_contract number(4),

valoare\_bon\_masa number(4),

id\_angajat number(4),

CONSTRAINT fk\_date\_salarizare\_angajat

FOREIGN KEY (id\_angajat) REFERENCES angajat(id\_angajat));

CREATE TABLE CHEIE (

id\_cheie number(4) PRIMARY KEY,

valoare varchar2(200),

id\_angajat number(4),

CONSTRAINT fk\_cheie\_angajat

FOREIGN KEY (id\_angajat) REFERENCES angajat(id\_angajat));

CREATE TABLE SITUATIE\_LUNARA\_ANGAJAT (

id\_situatie\_angajat number(4) PRIMARY KEY,

ore\_lucrate number(4),

zile\_lucrate number(4),

ore\_recuperate number(4),

zile\_concediu\_utilizate number(4),

bonuri\_de\_masa number(4),

id\_situatie number(4),

id\_angajat number(4),

CONSTRAINT fk\_situatie\_lunara\_angajat

FOREIGN KEY (id\_angajat) REFERENCES angajat(id\_angajat),

CONSTRAINT fk\_situatie\_lunara\_situatie

FOREIGN KEY (id\_situatie) REFERENCES situatie\_lunara(id\_situatie));

CREATE TABLE CERERE\_INVOIRE (

id\_cerere number(4) PRIMARY KEY,

data\_invoire date,

ora\_inceput number(4),

ora\_sfarsit number(4),

ore\_absenta number(2),

id\_angajat number(4),

id\_motiv number(4),

CONSTRAINT fk\_invoire\_motiv

FOREIGN KEY (id\_motiv) REFERENCES motiv\_invoire(id\_motiv),

CONSTRAINT fk\_invoire\_angajat

FOREIGN KEY (id\_angajat) REFERENCES angajat(id\_angajat));

CREATE TABLE CERERE\_CONCEDIU (

id\_cerere number(4) PRIMARY KEY,

ora\_inceput date,

ora\_sfarsit date,

nr\_zile number(2),

inlocuitor number(4),

id\_angajat number(4),

id\_motiv number(4),

CONSTRAINT fk\_concediu\_motiv

FOREIGN KEY (id\_motiv) REFERENCES motiv\_concediu(id\_motiv),

CONSTRAINT fk\_concediu\_angajat

FOREIGN KEY (id\_angajat) REFERENCES angajat(id\_angajat));

CREATE TABLE SITUATIE\_ZILNICA (

id\_angajat number(4),

data\_situatie date,

ore\_lucrate number(4),

ore\_recuperate number(4),

CONSTRAINT pk\_situatie\_zilnica

PRIMARY KEY (id\_angajat,data\_situatie),

CONSTRAINT fk\_situatie\_zilnica\_angajat

FOREIGN KEY (id\_angajat) REFERENCES angajat(id\_angajat));

CREATE TABLE DISTRIBUTIE\_LOCATII (

id\_distributie\_locatii number(4) PRIMARY KEY,

id\_locatie number(4),

id\_angajat number(4),

data\_situatie date,

CONSTRAINT fk\_distr\_locatii\_locatie

FOREIGN KEY (id\_locatie) REFERENCES locatie\_activitate(id\_locatie),

CONSTRAINT fk\_distr\_locatii\_situatie

FOREIGN KEY (id\_angajat,data\_situatie) REFERENCES situatie\_zilnica(id\_angajat,data\_situatie));

## e) Prezentarea regulilor de securitate care vor fi aplicate asupra modelului

Datele de salarizare vor fi criptate cu o cheie diferită pentru fiecare angajat. Salariu, orele de lucru, valoarea bonurilor de masa și zilele de concediu din contract vor fi criptate folosind algoritmi diferiți. Cheile vor fi generate random și vor fi memorate în tabela chei. Parolele de conectare a angajaților nu vor fi memorate, ci doar rezumatul hash al acestora.

Tabela privilegii va conține drepturi atomice asupra tabelelor și înregistrărilor din acestea. De exemplu, unul dintre privilegii este acela de vizualizare a datelor de salarizare, acest privilegiu va fi prezent in rolul angajatului, dar și în rolul de resurse umane. Angajații vor putea vizualiza doar detaliile de salarizare care le aparțin și nu vor putea modifica decât parola din tabela cu date de conectare. Cererile de învoire și concediu vor putea fi aprobate doar de managerul angajatului. Managerii vor mai putea modifica datele din situația zilnică a angajaților pe care îi au în subordonare.

# 2. Criptarea datelor

După fiecare introducere sau schimbare a datelor de conectare, parola va fi înlocuită cu valoarea hash MD5 a acesteia.

GRANT EXECUTE ON dbms\_crypto TO C##angajat1;

GRANT EXECUTE ON dbms\_crypto TO C##angajat2;

GRANT EXECUTE ON dbms\_crypto TO C##angajat3;

GRANT EXECUTE ON dbms\_crypto TO C##angajat4;

GRANT EXECUTE ON dbms\_crypto TO C##angajat5;

GRANT EXECUTE ON dbms\_crypto TO C##resurse\_umane1;

GRANT EXECUTE ON dbms\_crypto TO C##administrator\_firma1;

GRANT EXECUTE ON dbms\_crypto TO C##proprietar\_firma1;

GRANT EXECUTE ON dbms\_crypto TO C##app\_admin;

GRANT EXECUTE ON dbms\_crypto TO C##manager1;

GRANT EXECUTE ON dbms\_crypto TO C##manager2;

GRANT EXECUTE ON dbms\_crypto TO C##manager3;

CREATE OR REPLACE TRIGGER criptare\_date\_conectare

BEFORE INSERT OR UPDATE on date\_conectare

FOR EACH ROW

declare

hash\_parola RAW(100);

raw\_parola RAW(100);

mod\_operare NUMBER;

hash\_parola\_char VARCHAR2(200);

begin

raw\_parola := utl\_i18n.string\_to\_raw(:new.parola,'AL32UTF8');

mod\_operare := DBMS\_CRYPTO.HASH\_MD5;

hash\_parola := DBMS\_CRYPTO.Hash (raw\_parola, mod\_operare);

hash\_parola\_char := utl\_i18n.raw\_to\_char(hash\_parola,'AL32UTF');

dbms\_output.put\_line('Parola Hash este ' || hash\_parola\_char);

:new.parola = has\_parola\_char

end;

/

Actualizarea datelor de salarizare se realizează prin intermediul unei vizualizări a tabelei cu aceste date. Operațiile de inserare și modificare a datelor vor actualiza în același timp și cheile de criptare:

CREATE VIEW VIEW\_DATE\_SALARIZARE AS SELECT \* FROM DATE\_SALARIZARE;

create or replace trigger criptare\_date\_salarizare

INSTEAD OF INSERT OR UPDATE ON VIEW\_DATE\_SALARIZARE

FOR EACH ROW

declare

raw\_salariu RAW(100);

raw\_ore\_zi\_contract RAW(100);

raw\_zile\_concediu\_contract RAW(100);

raw\_valoare\_bon\_masa RAW(100);

criptat\_salariu varchar2(100);

criptat\_ore\_zi\_contract varchar2(100);

criptat\_zile\_concediu\_contract varchar2(100);

criptat\_valoare\_bon\_masa varchar2(100);

raw\_cheie RAW (200);

cheie\_char VARCHAR2(200);

mod\_operare1 NUMBER;

mod\_operare2 NUMBER;

mod\_operare3 NUMBER;

mod\_operare4 NUMBER;

hash\_parola\_char VARCHAR2(200);

begin

raw\_salariu := utl\_i18n.string\_to\_raw(:new.salariu,'AL32UTF8');

raw\_ore\_zi\_contract := utl\_i18n.string\_to\_raw(:new.ore\_zi\_contract,'AL32UTF8');

raw\_zile\_concediu\_contract := utl\_i18n.string\_to\_raw(:new.zile\_concediu\_contract,'AL32UTF8');

raw\_valoare\_bon\_masa := utl\_i18n.string\_to\_raw(:new.valoare\_bon\_masa,'AL32UTF8');

mod\_operare1 := DBMS\_CRYPTO.ENCRYPT\_DES + DBMS\_CRYPTO.PAD\_ZERO + DBMS\_CRYPTO.CHAIN\_CBC;

mod\_operare2 := DBMS\_CRYPTO.ENCRYPT\_3DES + DBMS\_CRYPTO.PAD\_NONE + DBMS\_CRYPTO.CHAIN\_CFB;

mod\_operare3 := DBMS\_CRYPTO.ENCRYPT\_AES256 + DBMS\_CRYPTO.PAD\_NONE + DBMS\_CRYPTO.CHAIN\_ECB;

mod\_operare4 := DBMS\_CRYPTO.ENCRYPT\_RC4 + DBMS\_CRYPTO.PAD\_ZERO + DBMS\_CRYPTO.CHAIN\_OFB;

IF INSERTING THEN

raw\_cheie:= DBMS\_CRYPTO.randombytes (200);

criptat\_salariu := utl\_i18n.raw\_to\_char(DBMS\_CRYPTO.ENCRYPT(raw\_salariu, mod\_operare4, raw\_cheie), 'AL32UTF8');

criptat\_ore\_zi\_contract := utl\_i18n.raw\_to\_char(DBMS\_CRYPTO.ENCRYPT(raw\_ore\_zi\_contract, mod\_operare3, raw\_cheie), 'AL32UTF8');

criptat\_zile\_concediu\_contract:=utl\_i18n.raw\_to\_char(DBMS\_CRYPTO.ENCRYPT(raw\_zile\_concediu\_contract, mod\_operare2, raw\_cheie), 'AL32UTF8');

criptat\_valoare\_bon\_masa :=utl\_i18n.raw\_to\_char(DBMS\_CRYPTO.ENCRYPT(raw\_valoare\_bon\_masa, mod\_operare1, raw\_cheie), 'AL32UTF8');

dbms\_output.put\_line('Criptat\_salariu este ' || criptat\_salariu);

INSERT INTO VIEW\_DATE\_SALARIZARE VALUES (secv\_id\_date\_salarizare.nextval, criptat\_salariu, criptat\_ore\_zi\_contract, criptat\_zile\_concediu\_contract, criptat\_valoare\_bon\_masa, :new.id\_angajat);

INSERT INTO CHEIE VALUES (secv\_id\_cheie.nextval, utl\_i18n.raw\_to\_char(raw\_cheie, 'AL32UTF8'), :new.id\_angajat);

END IF;

IF UPDATING THEN

SELECT valoare INTO cheie\_char FROM cheie WHERE id\_angajat = :new.id\_angajat;

raw\_cheie:= utl\_i18n.string\_to\_raw (cheie\_char, 'AL32UTF8');

criptat\_salariu := utl\_i18n.raw\_to\_char(DBMS\_CRYPTO.ENCRYPT(raw\_salariu, mod\_operare4, raw\_cheie), 'AL32UTF8');

criptat\_ore\_zi\_contract := utl\_i18n.raw\_to\_char(DBMS\_CRYPTO.ENCRYPT(raw\_ore\_zi\_contract, mod\_operare3, raw\_cheie), 'AL32UTF8');

criptat\_zile\_concediu\_contract:=utl\_i18n.raw\_to\_char(DBMS\_CRYPTO.ENCRYPT(raw\_zile\_concediu\_contract, mod\_operare2, raw\_cheie), 'AL32UTF8');

criptat\_valoare\_bon\_masa:= utl\_i18n.raw\_to\_char(DBMS\_CRYPTO.ENCRYPT(raw\_valoare\_bon\_masa, mod\_operare1, raw\_cheie), 'AL32UTF8');

dbms\_output.put\_line('Criptat\_salariu este ' || criptat\_salariu);

UPDATE VIEW\_DATE\_SALARIZARE SET salariu = criptat\_salariu, ore\_zi\_contract = criptat\_ore\_zi\_contract,

zile\_concediu\_contract = criptat\_zile\_concediu\_contract, valoare\_bon\_masa = criptat\_valoare\_bon\_masa

WHERE id\_angajat = :new.id\_angajat;

END IF;

end;

/

# 3. Auditarea activităților asupra bazei de date

## a. Auditare standard

Auditarea standard se face în baza de date cu stocarea tuturor modificărilor de adăugare și ștergere a angajaților. Încercările eșuate de vizualizare a datelor de conectare vor fi de asemenea memorate.

alter system set audit\_trail=DB,EXTENDED scope=SPFILE;

show PARAMETER audit\_trail;

GRANT AUDIT ANY to SYS;

AUDIT INSERT, DELETE ON C##app\_admin.ANGAJAT;

AUDIT SELECT ON C##app\_admin.DATE\_CONECTARE WHENEVER NOT SUCCESSFUL;

SELECT \* FROM DBA\_AUDIT\_TRAIL;

NOAUDIT ALL;

## b. *Trigger*-i de auditare

Fiecare modificare asupra privilegiilor va fi memorată într-o tabelă separată. În această tabelă se vor memora vechea și noua valoare a privilegiului înainte de modificarea oricărei înregistrări din tabelă. Pentru fiecare modificare se vor stoca utilizatorul, host-ul de pe care a executat modificarea și momentul de timp când a fost executată.

create table tab\_audit\_privilegii (

id\_secv number(4) primary key,

utilizator varchar2(20),

host varchar2(100),

timp date,

privilegiu\_vechi varchar2(200),

privilegiu\_nou varchar2(200));

drop SEQUENCE secv\_aud\_modificari\_salariu;

create sequence secv\_aud\_modificari\_privilegii start with 1 increment by 1;

create or replace trigger auditare\_adaugare\_privilegii

before update on C##app\_admin.privilegiu

for each row

declare

privilegiu\_old varchar2(200);

begin

SELECT tip\_privilegiu into privilegiu\_old from C##app\_admin.privilegiu WHERE id\_privilegiu = :new.id\_privilegiu;

insert into tab\_audit\_privilegii values(secv\_aud\_modificari\_privilegii.nextval, sys\_context('userenv', 'session\_user'),

sys\_context('userenv', 'host'), sysdate,

privilegiu\_old, :new.tip\_privilegiu);

end;

/

## c. Politici de auditare

Memorarea modificărilor asupra privilegiilor vor fi memorate în tabela de audit prin intermediul unei politici de audit.

create or replace procedure politica\_audit

as

begin

DBMS\_FGA.ADD\_POLICY (

object\_schema=> 'C##app\_admin',

object\_name=>'C##app\_admin.privilegiu',

policy\_name=>'Politica auditare privilegii',

audit\_column=>'tip\_privilegiu',

enable=>true,

statement\_types=>'select,insert,update,delete');

end;

/

execute politica\_audit;

# 4. Gestiunea utilizatorilor unei baze de date si a resurselor computaționale

Toți utilizatorii aplicației sunt identificați prin adresa de mail și parolă:

* Angajați
* Manageri
* Resurse umane
* Administratori firmă
* Administratori aplicație
* Proprietari firmă

Procesele din cadrul aplicației sunt:

1. Depunerea cererilor de concediu
2. Depunerea cererilor de invoire
3. Actualizarea situatiilor zilnice
4. Actualizarea situatiilor lunare
5. Vizualizarea situatiilor lunare
6. Vizualizarea situatiilor zilnice
7. Actualizarea datelor de conectare
8. Actualizarea datelor de salarizare
9. Vizualizarea datelor angajatilor
10. Crearea, acordarea si revocarea de privilegii si roluri
11. Înregistrare în sistem
12. Logare în sistem

Fiecare utilizator va avea o schemă proprie cu următoarele tabele:

* datele de salarizare vor fi stocate in scema proprietarului firmei;
* angajații, managerii, resurse umane si administratorii firmei vor avea propria lor tabelă cu cereri de învoire și de concediu, situații zilnice, lunare și distribuție locații pentru situațiile zilnice;
* restul tabelelor vor fi stocate în schema administratorului aplicației.

## a. Proiectarea configurației de management a identităților în baza de date (matricile proces-utilizator, entitate-proces, entitate-utilizator)

**Matricea proces-utilizator**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Procese**  **Utilizatori** | **Angajat** | **Manager** | **Resurse**  **umane** | **Administrator**  **firmă** | **Administrator**  **Aplicație** | **Proprietar**  **firmă** |
| **P1** | **X** | **X** | **X** | **X** |  |  |
| **P2** | **X** | **X** | **X** | **X** |  |  |
| **P3** |  | **X** | **X** |  |  |  |
| **P4** |  |  | **X** | **X** |  |  |
| **P5** | **X** | **X** | **X** | **X** |  | **X** |
| **P6** | **X** | **X** | **X** | **X** |  |  |
| **P7** | **X** | **X** | **X** | **X** | **X** | **X** |
| **P8** |  |  | **X** | **X** |  | **X** |
| **P9** | **X** | **X** | **X** | **X** | **X** | **X** |
| **P10** |  |  |  |  | **X** |  |
| **P11** | **x** | **x** | **x** | **x** | **x** | **x** |
| **P12** | **x** | **x** | **x** | **x** | **x** | **x** |

**Matricea entitate-proces**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Entitate**  **Proces** | **P1** | **P2** | **P3** | **P4** | **P5** | **P6** | **P7** | **P8** | **P9** | **P10** | **P11** | **P12** |
| Situatie lunara |  |  |  | **I,U,D** | **S** |  |  |  |  |  |  |  |
| Situatie lunara angajat |  |  |  | **I,U,D** | **S** |  |  |  |  |  |  |  |
| Rol |  |  |  | **S** | **S** |  |  |  |  | **I,U,D** |  |  |
| Distributie privilegii |  |  |  | **S** |  |  |  |  |  | **I,U,D** |  |  |
| Privilegiu |  |  |  | **S** |  |  |  |  |  | **I,U,D** |  |  |
| Locatie activitate |  |  | **S** |  |  | **S** |  |  |  |  |  |  |
| Distributie locatii |  |  | **I,U,D** | **S** |  | **S** |  |  |  |  |  |  |
| Situatie zilnica |  |  | **I,U,D** | **S** |  | **S** |  |  |  |  |  |  |
| Motiv concediu | **S** |  |  |  |  |  |  |  |  |  |  |  |
| Cerere concediu | **I,U,D** |  |  |  |  |  |  |  |  |  |  |  |
| Motiv invoire |  | **S** |  |  |  |  |  |  |  |  |  |  |
| Cerere invoire |  | **I,U,D** |  |  |  |  |  |  |  |  |  |  |
| Date salarizare |  |  |  | **S** |  |  |  | **I,U,D** | **S** |  | **I,U,D** |  |
| Date conectare |  |  |  |  |  |  | **I,U,D** |  | **S** |  | **I,U,D** | **S** |
| Cheie |  |  |  |  |  |  | **S** | **S** |  |  | **I,U,D** | **S** |
| Angajat | **S** | **S** |  | **S** |  |  | **S** | **S** | **S** | **S** | **I,U,D** | **S** |

**Matricea entitate-utilizator**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Entitate**  **Proces** | **Angajat** | **Manager** | **Resurse**  **umane** | **Administrator**  **firmă** | **Administrator**  **Aplicație** | **Proprietar**  **firmă** |
| Situatie lunara |  |  | **S** | **S** |  | **S** |
| Situatie lunara angajat |  |  | **I,U,D,S** | **S** |  | **S** |
| Rol |  |  |  |  | **I,U,D,S** |  |
| Distributie privilegii |  |  |  |  | **I,U,D,S** |  |
| Privilegiu |  |  |  |  | **I,U,D,S** |  |
| Locatie activitate |  |  | **S** | **S** | **I,U,D** | **S** |
| Distributie locatii |  | **I,U,D** | **S** | **S** |  | **S** |
| Situatie zilnica |  | **I,U,D** | **S** | **S** |  | **S** |
| Motiv concediu | **S** | **S** |  |  | **I,U,D** |  |
| Cerere concediu | **I,U,D,S** | **I,U,D,S** | **S,U** | **S,U** |  | **S** |
| Motiv invoire | **S** | **S** |  |  | **I,U,D** |  |
| Cerere invoire | **I,U,D,S** | **U,S** | **U,S** |  |  |  |
| Date salarizare | **S** | **S** | **S** | **I,U,D** |  | **I,U,D** |
| Date conectare | **S,I,U** | **S,I,U** | **S,I,U** | **S,I,U** | **I,U,D,S** | **S,U** |
| Cheie |  |  |  |  | **I,U,D,S** |  |
| Angajat | **S** | **S** | **S** | **I,U,D,S** | **I,U,D,S** | **S** |

## b. Implementarea configurației de management a identităților în baza de date

# Au fost creați utilizatorii bazei de date și rolurile acestora. Pentru utilizatorii cu rolul de angajat, manager, resurse umane, administrator firmă s-au alocat 50MB pentru schema proprie. Deoarece în schema proprietarului firmei se gasesc doar datele de conectare, s-au alocat doar 10MB.

# --crearea utilizatorilor

# CREATE USER C##app\_admin IDENTIFIED BY app\_admin;

# CREATE USER C##angajat1 IDENTIFIED BY angajat;

# CREATE USER C##angajat2 IDENTIFIED BY angajat;

# CREATE USER C##angajat3 IDENTIFIED BY angajat;

# CREATE USER C##angajat4 IDENTIFIED BY angajat;

# CREATE USER C##angajat5 IDENTIFIED BY angajat;

# CREATE USER C##manager1 IDENTIFIED BY manager;

# CREATE USER C##manager2 IDENTIFIED BY manager;

# CREATE USER C##manager3 IDENTIFIED BY manager;

# CREATE USER C##resurse\_umane1 IDENTIFIED BY resurse;

# CREATE USER C##administrator\_firma1 IDENTIFIED BY fadmin;

# CREATE USER C##proprietar\_firma1 IDENTIFIED BY fproprietar;

# --crearea rolurilor

# CREATE ROLE C##administrator\_aplicatie;

# CREATE ROLE C##administrator\_firma;

# CREATE ROLE C##angajat;

# CREATE ROLE C##manager;

# CREATE ROLE C##resurse\_umane;

# CREATE ROLE C##proprietar\_firma;

--acordarea rolurilor catre utilizatori

GRANT C##angajat to C##angajat1;

GRANT C##angajat to C##angajat2;

GRANT C##angajat to C##angajat3;

GRANT C##angajat to C##angajat4;

GRANT C##angajat to C##angajat5;

GRANT C##manager to C##manager1;

GRANT C##manager to C##manager1;

GRANT C##manager to C##manager2;

GRANT C##angajat to C##angajat3;

GRANT C##resurse\_umane to C##resurse\_umane1;

GRANT C##administrator\_aplicatie to C##app\_admin;

GRANT C##administrator\_firma to C##administrator\_firma1;

GRANT C##proprietar\_firma to C##proprietar\_firma1;

ALTER USER C##ANGAJAT1 DEFAULT ROLE C##ANGAJAT;

--acordarea drepturilor de conectare

GRANT CONNECT, CREATE SESSION TO C##angajat1;

--acordarea dreturilor de creare tabel in propria schema

GRANT CREATE TABLE, CREATE VIEW TO C##ANGAJAT;

GRANT CREATE TABLE, CREATE VIEW TO C##RESURSE\_UMANE;

GRANT CREATE TABLE, CREATE VIEW TO C##MANAGER;

GRANT CREATE TABLE, CREATE VIEW TO C##administrator\_firma1;

GRANT CREATE TABLE, CREATE VIEW TO C##app\_admin;

GRANT CREATE TABLE, CREATE VIEW TO C##proprietar\_firma1;

ALTER USER C##angajat1 QUOTA 50M ON USERS;

ALTER USER C##angajat2 QUOTA 50M ON USERS;

ALTER USER C##angajat3 QUOTA 50M ON USERS;

ALTER USER C##angajat4 QUOTA 50M ON USERS;

ALTER USER C##angajat5 QUOTA 50M ON USERS;

ALTER USER C##manager1 QUOTA 50M ON USERS;

ALTER USER C##manager2 QUOTA 50M ON USERS;

ALTER USER C##manager3 QUOTA 50M ON USERS;

ALTER USER C##app\_admin QUOTA unlimited ON USERS;

ALTER USER C##resurse\_umane1 QUOTA 50M ON USERS;

ALTER USER C##administrator\_firma1 QUOTA 50M ON USERS;

ALTER USER C##proprietar\_firma1 QUOTA 10M ON USERS;

GRANT SELECT ON SITUATIE\_LUNARA TO C##RESURSE\_UMANE, C##ADMINISTRATOR\_FIRMA, C##PROPRIETAR\_FIRMA;

GRANT SELECT, INSERT, UPDATE, DELETE ON C##APP\_ADMIN.SITUATIE\_LUNARA\_ANGAJAT TO C##RESURSE\_UMANE;

GRANT SELECT ON C##APP\_ADMIN.SITUATIE\_LUNARA\_ANGAJAT TO C##administrator\_firma;

GRANT SELECT ON C##APP\_ADMIN.SITUATIE\_LUNARA\_ANGAJAT TO C##proprietar\_firma1;

GRANT SELECT ON LOCATIE\_ACTIVITATE TO C##RESURSE\_UMANE;

GRANT SELECT ON LOCATIE\_ACTIVITATE TO C##administrator\_firma;

GRANT SELECT ON LOCATIE\_ACTIVITATE TO C##proprietar\_firma;

GRANT SELECT ON DISTRIBUTIE\_LOCATII TO C##RESURSE\_UMANE;

GRANT SELECT ON DISTRIBUTIE\_LOCATII TO C##administrator\_firma;

GRANT SELECT ON DISTRIBUTIE\_LOCATII TO C##proprietar\_firma;

GRANT INSERT, UPDATE, DELETE ON DISTRIBUTIE\_LOCATII TO C##manager;

GRANT SELECT ON C##APP\_ADMIN.distributie\_locatii TO C##administrator\_firma;

GRANT SELECT ON C##APP\_ADMIN.distributie\_locatii TO C##resurse\_umane;

GRANT SELECT ON C##APP\_ADMIN.distributie\_locatii TO C##proprietar\_firma;

GRANT SELECT ON C##APP\_ADMIN.situatie\_zilnica TO C##administrator\_firma;

GRANT SELECT ON C##APP\_ADMIN.situatie\_zilnica TO C##resurse\_umane;

GRANT SELECT ON C##APP\_ADMIN.situatie\_zilnica TO C##proprietar\_firma;

GRANT SELECT ON MOTIV\_CONCEDIU TO C##ANGAJAT, C##MANAGER;

GRANT SELECT ON MOTIV\_INVOIRE TO C##ANGAJAT, C##MANAGER;

GRANT SELECT, UPDATE ON C##APP\_ADMIN.cerere\_concediu TO C##administrator\_firma;

GRANT SELECT, UPDATE ON C##APP\_ADMIN.cerere\_concediu TO C##resurse\_umane;

GRANT SELECT, UPDATE ON C##APP\_ADMIN.cerere\_concediu TO C##proprietar\_firma;

GRANT SELECT, INSERT, UPDATE ON DATE\_CONECTARE TO C##ANGAJAT;

GRANT SELECT, INSERT, UPDATE ON DATE\_CONECTARE TO C##MANAGER;

GRANT SELECT, INSERT, UPDATE ON DATE\_CONECTARE TO C##resurse\_umane;

GRANT SELECT, INSERT, UPDATE ON DATE\_CONECTARE TO C##administrator\_firma;

GRANT SELECT, UPDATE ON DATE\_CONECTARE TO C##proprietar\_firma;

GRANT SELECT ON ANGAJAT TO C##ANGAJAT;

GRANT SELECT ON ANGAJAT TO C##MANAGER;

GRANT SELECT ON ANGAJAT TO C##resurse\_umane;

GRANT SELECT,INSERT, UPDATE, DELETE ON ANGAJAT TO C##administrator\_firma;

GRANT SELECT ON ANGAJAT TO C##proprietar\_firma;

GRANT SELECT, INSERT, UPDATE, DELETE ON C##ADMINISTRATOR\_FIRMA1.SITUATIE\_LUNARA\_ANGAJAT TO C##RESURSE\_UMANE;

GRANT SELECT ON C##ADMINISTRATOR\_FIRMA1.SITUATIE\_LUNARA\_ANGAJAT TO C##proprietar\_firma1;

GRANT SELECT ON C##ADMINISTRATOR\_FIRMA1.SITUATIE\_LUNARA\_ANGAJAT TO C##administrator\_firma;

GRANT SELECT ON C##ADMINISTRATOR\_FIRMA1.distributie\_locatii TO C##proprietar\_firma;

GRANT SELECT ON C##ADMINISTRATOR\_FIRMA1.distributie\_locatii TO C##resurse\_umane;

GRANT SELECT ON C##ADMINISTRATOR\_FIRMA1.distributie\_locatii TO C##administrator\_firma;

GRANT SELECT ON C##ADMINISTRATOR\_FIRMA1.situatie\_zilnica TO C##resurse\_umane;

GRANT SELECT ON C##ADMINISTRATOR\_FIRMA1.situatie\_zilnica TO C##administrator\_firma;

GRANT SELECT ON C##ADMINISTRATOR\_FIRMA1.situatie\_zilnica TO C##proprietar\_firma;

GRANT SELECT ON C##ADMINISTRATOR\_FIRMA1.cerere\_concediu TO C##proprietar\_firma;

GRANT SELECT, UPDATE ON C##ADMINISTRATOR\_FIRMA1.cerere\_concediu TO C##resurse\_umane;

GRANT SELECT, UPDATE ON C##ADMINISTRATOR\_FIRMA1.cerere\_concediu TO C##administrator\_firma;

GRANT SELECT ON C##PROPRIETAR\_FIRMA1.DATE\_SALARIZARE TO C##resurse\_umane;

GRANT INSERT, UPDATE, DELETE ON C##PROPRIETAR\_FIRMA1.DATE\_SALARIZARE TO C##administrator\_firma;

GRANT SELECT, INSERT, UPDATE, DELETE ON C##MANAGER1.SITUATIE\_LUNARA\_ANGAJAT TO C##RESURSE\_UMANE;

GRANT SELECT ON C##MANAGER1.SITUATIE\_LUNARA\_ANGAJAT TO C##administrator\_firma;

GRANT SELECT ON C##MANAGER1.SITUATIE\_LUNARA\_ANGAJAT TO C##proprietar\_firma1;

GRANT SELECT ON C##MANAGER1.distributie\_locatii TO C##administrator\_firma;

GRANT SELECT ON C##MANAGER1.distributie\_locatii TO C##resurse\_umane;

GRANT SELECT ON C##MANAGER1.distributie\_locatii TO C##proprietar\_firma;

GRANT SELECT ON C##MANAGER1.situatie\_zilnica TO C##proprietar\_firma;

GRANT SELECT ON C##MANAGER1.situatie\_zilnica TO C##administrator\_firma;

GRANT SELECT ON C##MANAGER1.situatie\_zilnica TO C##resurse\_umane;

GRANT SELECT ON C##MANAGER1.cerere\_concediu TO C##proprietar\_firma;

GRANT SELECT, UPDATE ON C##MANAGER1.cerere\_concediu TO C##resurse\_umane;

GRANT SELECT, UPDATE ON C##MANAGER1.cerere\_concediu TO C##administrator\_firma;

# 5. Privilegii și roluri

# Crearea utilizatorilor și rolurilor s-a realizat din schema sys. Rolurile și utilizatorii bazei de date sunt și cei ai aplicației. Acordarea privilegiilor asupra anumitor tabele din diferite scheme se realizeaza printr-un trigger apelare la toate operațiile de modificare asupra tabelei DISTRIBUȚIE\_PRIVILEGII.

# CREATE OR REPLACE TRIGGER ACTUALIZAR\_PRVILEGII

# BEFORE INSERT OR UPDATE OR DELETE ON DISTRIBUTIE\_PRIVILEGII

# FOR EACH ROW

# DECLARE

# var\_rol varchar2(200);

# var\_schema varchar2(100);

# var\_tabel varchar2(100);

# var\_privilegiu varchar2(50);

# old\_id\_privilegiu number(4);

# old\_id\_rol number(4);

# old\_rol varchar2(200);

# old\_schema varchar2(100);

# old\_tabel varchar2(100);

# old\_privilegiu varchar2(50);

# granting varchar2(400);

# revoking varchar2(400);

# BEGIN

# SELECT denumire INTO var\_rol FROM rol WHERE id\_rol = :new.id\_rol;

# SELECT tip\_privilegiu,schema\_privilegiu, tabela\_privilegiu

# INTO var\_privilegiu,var\_schema,var\_tabel

# FROM privilegiu where id\_privilegiu = :new.id\_privilegiu;

# IF INSERTING THEN

# granting := 'GRANT ' || var\_privilegiu ||' ON '||var\_schema ||'.'|| var\_tabel||' TO '||var\_rol;

# EXECUTE IMMEDIATE granting;

# END IF;

# IF DELETING THEN

# revoking := 'REVOKE ' || var\_privilegiu ||' ON '||var\_schema ||'.'|| var\_tabel||' FROM '||var\_rol;

# EXECUTE IMMEDIATE revoking;

# END IF;

# IF UPDATING THEN

# SELECT id\_rol, id\_privilegiu into old\_id\_privilegiu, old\_id\_rol

# from distributie\_privilegii where id\_distributie\_privilegiu = :new.id\_distributie\_privilegiu;

# SELECT denumire INTO old\_rol FROM rol WHERE id\_rol = old\_id\_rol;

# SELECT tip\_privilegiu,schema\_privilegiu, tabela\_privilegiu

# INTO old\_privilegiu,old\_schema,old\_tabel

# FROM privilegiu where id\_privilegiu = old\_id\_privilegiu;

# revoking := 'REVOKE ' || old\_privilegiu ||' ON '||old\_schema ||'.'|| old\_tabel||' FROM '||old\_rol;

# granting := 'GRANT ' || var\_privilegiu ||' ON '||var\_schema ||'.'|| var\_tabel||' TO '||var\_rol;

# EXECUTE IMMEDIATE revoking;

# EXECUTE IMMEDIATE granting;

# END IF;

# END;

# /

# 6. Aplicațiile pe baza de date si securitatea datelor

## a. Contextul aplicației

## b. SQL Injection

# 7. Mascarea datelor in Oracle