**PROIECT INTEGRARE INFORMAȚIONALĂ**

**STUDENȚI: BĂLĂȘEL MARIANA-RODICA**

**HENEA LAURA-ELENA**

**MANOLACHE PAULA-CRISTINA**

DS\_1**: Doctori** **-** conține date generale despre doctori

* TIP: CSV
* Implementare sursă de date externă: doctori.csv
* Implementare acces la sursa externă de date:

1. Tip\_acces: Oracle\_Loader
2. Nume tabelă locală: doctori\_table
3. Implementare:

Ne-am conectat ca sys și am rulat următoarele comenzi:

--DROP DIRECTORY external\_data;

CREATE OR REPLACE DIRECTORY external\_data AS 'C:\App\external\_data';

GRANT ALL ON DIRECTORY external\_data TO PUBLIC;

GRANT READ,WRITE ON DIRECTORY external\_data TO integrare;

De pe userul căruia i-am dat drepturi:

DROP TABLE doctori\_table;

CREATE TABLE doctori\_table (

iddoctor VARCHAR2(2),

numedoctor VARCHAR2(30),

specialitate VARCHAR2(20),

datanasterii date

)

ORGANIZATION EXTERNAL (

TYPE ORACLE\_LOADER

DEFAULT DIRECTORY external\_data

ACCESS PARAMETERS (

RECORDS DELIMITED BY NEWLINE SKIP 1

FIELDS TERMINATED BY ','

MISSING FIELD VALUES ARE NULL

)

LOCATION ('doctori.csv')

)

REJECT LIMIT UNLIMITED;

SELECT \* FROM doctori\_table;

DS\_2: **Pacienți conține date generale despre pacienți**

* Tip: XLS
* Implementare sursă de date externă: pacienti.xls
* Implementare acces la sursa externă de date:

1. Tip acces: ExcelTable Library
2. Nume tabelă locală:
3. Implementare:

RUN scrpit install.sql:

--DROP PACKAGE EXCELTABLE;

--DROP PACKAGE XUTL\_CDF;

--DROP PACKAGE XUTL\_OFFCRYPTO;

--DROP PACKAGE XUTL\_XLS;

--DROP PACKAGE XUTL\_XLSB;

--DROP TYPE EXCELTABLECELLLIST;

--DROP TYPE EXCELTABLECELL;

--DROP TYPE EXCELTABLEIMPL;

prompt Creating package XUTL\_CDF ...

@@C:\App\external\_data\ExcelTable-master\MSUtilities\MSUtilities-master/CDFManager/xutl\_cdf.pks

@@C:\App\external\_data\ExcelTable-master\MSUtilities\MSUtilities-master/CDFManager/xutl\_cdf.pkb

prompt Creating package XUTL\_OFFCRYPTO ...

@@C:\App\external\_data\ExcelTable-master\MSUtilities\MSUtilities-master\OfficeCrypto\xutl\_offcrypto.pks

@@C:\App\external\_data\ExcelTable-master\MSUtilities\MSUtilities-master\OfficeCrypto\xutl\_offcrypto.pkb

prompt Creating type ExcelTableSheetList ...

@@C:\App\external\_data\ExcelTable-master\ExcelCommons\ExcelCommons-main\plsql\ExcelTableSheetList.tps

prompt Creating type ExcelTableCell ...

@@C:\App\external\_data\ExcelTable-master\ExcelCommons\ExcelCommons-main\plsql\ExcelTableCell.tps

prompt Creating type ExcelTableCellList ...

@@C:\App\external\_data\ExcelTable-master\ExcelCommons\ExcelCommons-main\plsql\ExcelTableCellList.tps

prompt Creating package ExcelTypes ...

@@C:\App\external\_data\ExcelTable-master\ExcelCommons\ExcelCommons-main\plsql\ExcelTypes.pks

@@C:\App\external\_data\ExcelTable-master\ExcelCommons\ExcelCommons-main\plsql\ExcelTypes.pkb

prompt Creating package XUTL\_XLS ...

@@C:\App\external\_data\ExcelTable-master\ExcelCommons\ExcelCommons-main\plsql\xutl\_xls.pks

@@C:\App\external\_data\ExcelTable-master\ExcelCommons\ExcelCommons-main\plsql\xutl\_xls.pkb

prompt Creating package XUTL\_XLSB ...

@@C:\App\external\_data\ExcelTable-master\ExcelCommons\ExcelCommons-main\plsql\xutl\_xlsb.pks

@@C:\App\external\_data\ExcelTable-master\ExcelCommons\ExcelCommons-main\plsql\xutl\_xlsb.pkb

prompt Creating package XUTL\_FLATFILE ...

@@C:\App\external\_data\ExcelTable-master\ExcelCommons\ExcelCommons-main\plsql\xutl\_flatfile.pkb

@@C:\App\external\_data\ExcelTable-master\ExcelCommons\ExcelCommons-main\plsql\xutl\_flatfile.pks

prompt Creating type ExcelTableImpl ...

@@C:\App\external\_data\ExcelTable-master\plsql\ExcelTableImpl.tps

prompt Creating package ExcelTable ...

@@C:\App\external\_data\ExcelTable-master\plsql\ExcelTable.pks

@@C:\App\external\_data\ExcelTable-master\plsql\ExcelTable.pkb

prompt Creating type ExcelTableImpl ...

@@C:\App\external\_data\ExcelTable-master\plsql\ExcelTableImpl.tpb

--DROP VIEW pacienti\_to\_view;

CREATE OR REPLACE VIEW pacienti\_to\_view AS

select t.\*

from TABLE(

ExcelTable.getRows(

ExcelTable.getFile('EXTERNAL\_DATA','pacienti.xls')

, 'pacienti'

, '"idpacienti" VARCHAR2(10)

, "numepacient" VARCHAR2(20)

, "CNP" VARCHAR2(13)

, "adresa" VARCHAR2(100)

, "loc" VARCHAR2(40)

, "judet" VARCHAR2(30)

, "tara" VARCHAR2(40)

, "serie\_nr\_act\_de\_identitate" VARCHAR2(10)'

, 'A2')

)t;

SELECT \* FROM pacienti\_to\_view;

DS\_3: **Gărzi** conține date despre gărzile doctorilor

* Tip: XML
* Implementare sursă de date externă: garzi.xml
* Implementare acces la sursa externă de date:

1. Tip\_acces: Remote view utilizând bfilename() și xmltable()
2. Nume tablă locală: garzi\_view
3. Implementare:

DROP VIEW garzi\_view;

CREATE OR REPLACE VIEW garzi\_view AS

select x.iddoctor, x.inceput\_garda, x.sfarsit\_garda

from XMLTABLE('/garzi/garda'

passing xmltype(

BFILENAME('EXTERNAL\_DATA', 'garzi.xml')

, nls\_charset\_id('AL32UTF8')

)

columns

iddoctor varchar2(20) path 'iddoctor'

, inceput\_garda varchar2(20) path 'inceput\_garda'

, sfarsit\_garda varcHAR2(20) path 'sfarsit\_garda'

) x;

select \* from garzi\_view;

DS\_4: **TRIAJ** conține detalii detaliate despre pacienți legate de vizita la spital

* Tip: SQL
* Implementare sursă de date externă: TRIAJ.sql
* Implementare acces la sursa externă de date:

1. Tip acces: DB\_LINK
2. Nume tabelă remote: Triaj
3. Implementare

create user TRIAJ identified by "TRIAJ";

GRANT CREATE SESSION, CREATE TABLE,CREATE SEQUENCE, CREATE VIEW, CREATE PROCEDURE TO TRIAJ;

alter user TRIAJ quota unlimited on users;

dupa in oracle face o conexiune triaj in pune create tabele si inserturile

CREATE TABLE triaj (

idexaminare varchar2(12),

dataora\_examinare TIMESTAMP,

idpacient INTEGER ,

simptome VARCHAR2(500) ,

tratament\_imediat VARCHAR2(500),

sectie\_destinatie VARCHAR2(30)

);

INSERT INTO triaj VALUES (1, TIMESTAMP'2008-01-03 7:18:00', 1, 'dureri de stomac intense', NULL, 'boli interne');

INSERT INTO triaj VALUES (2, TIMESTAMP'2008-01-03 8:45:00', 2, 'febra puternica, varsaturi', 'penicilina', 'boli interne');

INSERT INTO triaj VALUES (3, TIMESTAMP'2008-01-03 12:45:00', 3, 'deranjament stomacal', 'scobutil', NULL);

INSERT INTO triaj VALUES (4, TIMESTAMP'2008-01-03 20:45:00', 4, 'palpitatii cardiace', 'linistin', 'cardiologie');

INSERT INTO triaj VALUES (5, TIMESTAMP'2008-01-04 1:28:00', 5, 'plaga profunda picior drept', 'antibiotice, pansament', NULL);

INSERT INTO triaj VALUES (6, TIMESTAMP'2008-01-04 10:45:00', 3, 'contractii stomacale, varsaturi', NULL, 'boli interne');

INSERT INTO triaj VALUES (7, TIMESTAMP'2008-01-04 11:20:00', 7, 'fata de culoare galbena, ameteli', NULL, 'hepatologie');

INSERT INTO triaj VALUES (8, TIMESTAMP'2008-01-04 22:45:00', 8, 'dureri articulare', 'scobutil', NULL);

INSERT INTO triaj VALUES (9, TIMESTAMP'2008-01-05 6:18:00', 5, 'febra puternica, delir', 'penicilina', 'chirurgie');

DROP DATABASE LINK triajSpitalDB;

CREATE DATABASE LINK triajSpitalDB CONNECT TO TRIAJ identified by TRIAJ

USING '//localhost:1521/orcl';

SELECT \* FROM user\_db\_links;

DROP VIEW TRIAJ\_VIEW;

CREATE OR REPLACE VIEW triaj\_view as

Select idexaminare, dataora\_examinare,idpacient,simptome,

tratament\_imediat, sectie\_destinatie

from triaj@triajSpitalDB;

select \* from triaj\_view;

**TEMA 2**

Analiza datelor privind triajul unui spital

SCHEMA ANALITICĂ ROLAP

1. Olap Fact VIEW

* OLAP\_FACT\_VIEW\_HEALTH\_STATE