

# Sisteme de gestiune a bazelor de date

Lect. Univ. Dr. Claudia Dinuca

# Objective curs

- Limbajul Oracle SQL
- PL/SQL

# Examinare:

- Examen final
- Proiect final realizat si sustinut de student in ultimele intalniri din semestru +
- Activitatea pe parcursul anului

# Baza de date :

Definitie: **O baza de date** (BD, eng. DB) este **un ansamblu structurat de date** înregistrat pe suporturi accesibile calculatorului pentru a satisface simultan cerintele mai multor utilizatori într-un mod selectiv și în timp util.

Baza de date reprezintă o modalitate de stocare pe un suport extern a unei mulțimi de date care modelează un proces (sistem) din lumea reală, cu posibilitatea regăsirii acesteia.

# Definitii

**... un ansamblu structurat de date ...**

Regulile si conceptele care permit descrierea structurii unei BD formeaza modelul datelor.

În timp au fost definite astfel de modele precum:

- Modelul ierarhic în care datele erau organizate sub forma unui arbore, nodurile constând din înregistrări iar arcele referinte către alte noduri.
- Modelul retea în care datele erau organizate sub forma unui graf orientat. Nodurile si arcele au aceeași semnificație ca mai sus.
- Modelul relational în care, intuitiv, datele sunt organizate sub forma de tabele.

# Baze de date relationale

Conceptul de baze de date relaționale a fost introdus de E.F. Codd în anii '70 și modelează bazele de date sub formă de tabele și relații între acestea.

Coloană

Câmp

Linie

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	800	(null)	20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7566	JONES	MANAGER	7839	02-APR-81	2975	(null)	20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7698	BLAKE	MANAGER	7839	01-MAY-81	2850	(null)	30
7782	CLARK	MANAGER	7839	09-JUN-81	2450	(null)	10
7788	SCOTT	ANALYST	7566	09-DEC-82	3000	(null)	20
7839	KING	PRESIDENT	(null)	17-NOV-81	5000	(null)	10
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	(null)	30
7876	ADAMS	CLERK	7788	12-JAN-83	1100	(null)	20
7900	JAMES	CLERK	7698	03-DEC-81	950	(null)	30
7902	FORD	ANALYST	7566	03-DEC-81	3000	(null)	20
7934	MILLER	CLERK	7782	23-JAN-82	1300	(null)	10

# Definitie SGBD (Data Base Management System –DBMS)

Bazele de date sunt manipulate cu ajutorul sistemelor de gestiune a bazelor de date. Acestea, SGBD-urile, sunt responsabile cu crearea, manipularea și întreținerea unei baze de date

**Un sistem de gestiune a bazelor de date** SGBD/DBMS) este ansamblul de programe care permit utilizatorului sa interactioneze cu o baza de date.

# Definitie SGBD (Data Base Management System –DBMS)

Un **sistem de gestiune a bazelor de date**(*SGBD*, eng.*Data Base Management System -DBMS*) este un produs *software* care asigură interacțiunea cu o bază de date, permițând definirea, consultarea și actualizarea datelor din baza de date. Toate cererile de acces la baza de date sunt tratate și controlate de către SGBD.



# Rolul SGBD

Rolul unui SGBD Rolul unui SGBD într-un context de sistem de bază de date este de a:

1. defini și descrie structura bazei de date, printr-un limbaj propriu specific (LDD), conform unui anumit model de date;
2. încărca/valida datele în baza de date respectând niște restricțiile de integritate impuse de modelul de date utilizat;
3. realiza accesul la date pentru diferite operații (consultare, interogarea, actualizare etc) - operatorii modelului de date;
4. întreține BD cu ajutorul unor instrumente specializate (editoare, utilitare - shells, navigatoare – browsers etc.);
5. asigura protecția bazei de date sub cele două aspecte: securitatea și integritatea datelor.

# SGBD

## **Sistemul de gestiune a bazei de date (SGBD):**

- este un sistem complex de programe care asigură interfața între o bază de date și utilizatorii acesteia (exemple de programe: ACCESS, ORACLE, MySQL, SQL Server, PostgreSQL)
- este software-ul bazei de date care asigură:
  - 1) definirea structurii bazei de date;
  - 2) încărcarea datelor în baza de date;
  - 3) accesul la baza de date (interogare, actualizare);
  - 4) întreținerea bazei de date (refolosirea spațiilor goale, refacerea bazei de date în cazul unor incidente);
  - 5) reorganizarea bazei de date (restructurarea și modificarea strategiei de acces);
  - 6) securitatea datelor.

# Oracle SGBD

- <https://db-engines.com/en/ranking>
- Oracle este cel mai utilizat SGBD din lume

# Oracle SGBD

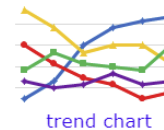
[Ranking](#) > Complete Ranking

[RSS](#) [RSS Feed](#)

## DB-Engines Ranking

The DB-Engines Ranking ranks database management systems according to their popularity. The ranking is updated monthly.

Read more about the [method](#) of calculating the scores.



415 systems in ranking, October 2023

Rank			DBMS	Database Model	Score		
Oct 2023	Sep 2023	Oct 2022			Oct 2023	Sep 2023	Oct 2022
1.	1.	1.	Oracle	Relational, Multi-model	1261.42	+20.54	+25.05
2.	2.	2.	MySQL	Relational, Multi-model	1133.32	+21.83	-72.06
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	896.88	-5.34	-27.80
4.	4.	4.	PostgreSQL	Relational, Multi-model	638.82	+18.06	+16.10
5.	5.	5.	MongoDB	Document, Multi-model	431.42	-8.00	-54.81
6.	6.	6.	Redis	Key-value, Multi-model	162.96	-0.72	-20.41
7.	7.	7.	Elasticsearch	Search engine, Multi-model	137.15	-1.84	-13.92
8.	8.	8.	IBM Db2	Relational, Multi-model	134.87	-1.85	-14.79
9.	9.	10.	SQLite	Relational	125.14	-4.06	-12.66
10.	10.	9.	Microsoft Access	Relational	124.31	-4.25	-13.85
11.	11.	13.	Snowflake	Relational	123.24	+2.35	+16.51
12.	12.	11.	Cassandra	Wide column, Multi-model	108.82	-1.24	-9.12
13.	13.	12.	MariaDB	Relational, Multi-model	99.66	-0.79	-9.65
14.	14.	14.	Splunk	Search engine	92.37	+0.98	-2.28
15.	15.	16.	Microsoft Azure SQL Database	Relational, Multi-model	80.93	-1.80	-4.03
16.	16.	15.	Amazon DynamoDB	Multi-model	80.91	+0.00	-7.44
17.	17.	20.	Databricks	Multi-model	75.82	+0.64	+18.21
18.	18.	17.	Hive	Relational	69.18	-2.65	-11.42
19.	19.	18.	Teradata	Relational, Multi-model	58.56	-1.77	-7.51
20.	20.	22.	Google BigQuery	Relational	56.57	+0.11	+4.12

# Oracle SGBD

- Pentru a interoga și prelucra informația stocată într-o bază de date, în Oracle se pot folosi următoarele limbaje de programare:
  - SQL : este un limbajul de programare dezvoltat special cu scopul de a procesa informația păstrată într-o bază de date (*set-oriented language*).
  - PL/SQL: este un limbaj procedural, construit ca extensie a limbajului SQL, care, pe lângă interogările SQL, folosește structuri de control (if...then, if...then...else, if...then...elsif...else, case ) și bucle (loop, while, for).
- Produsele Oracle s-au impus pe piață datorită instrumentelor de dezvoltare oferite, siguranței în exploatarea bazelor de date, gestionarea unui volum mare de date și mai ales datorită nivelurilor de securitate.

# Alte produse Oracle

- Pentru dezvoltarea de aplicații complexe pot fi folosite și alte produse Oracle, cum ar fi:
  - Forms Builder –pentru generarea de interfețe utilizator
  - Reports Builder –pentru generarea de rapoarte
- SQL Developer –IDE pentru dezvoltarea interogărilor SQL și programelor PL/SQL

# Organizare BD

Cele trei concepte de bază utilizate în organizarea bazei de date sunt:

- **entitatea**
- **atributul**
- **valoarea**

*Prin entitate se înțelege un obiect concret sau abstract reprezentat prin proprietățile sale. O proprietate a unui obiect poate fi exprimată prin perechea (ATRIBUT, VALOARE).*

**Exemplu:** În exemplul “Masa x are culoarea alba”, atributul este „culoare”, iar valoarea este reprezentată de cuvântul „albă”. Alte exemple ar putea fi: (Sex, Feminin), (Nume, POP), (Profesie, Medic), (Salariu, 200\$). **Observație:** Atributele pot caracteriza o clasă de entități, nu doar o entitate.

**Organizarea bazei de date – se referă la structura bazei de date și reprezintă un ansamblu de instrumente pentru descrierea datelor, relațiilor, restricțiilor la care sunt supuse.**

**Tranzacțiile efectuate asupra unei baze de date trebuie să respecte:**

### **Proprietăți ACID**

**A- atomicitate** -> tranzacția trebuie privită ca un atom

**C-consistentă**->după efectuarea tranz. , db trebuie să rămână în stare consistentă

**I-Izolare**->fiecare tranzacție este executată ca și când ar fi singură în sistem

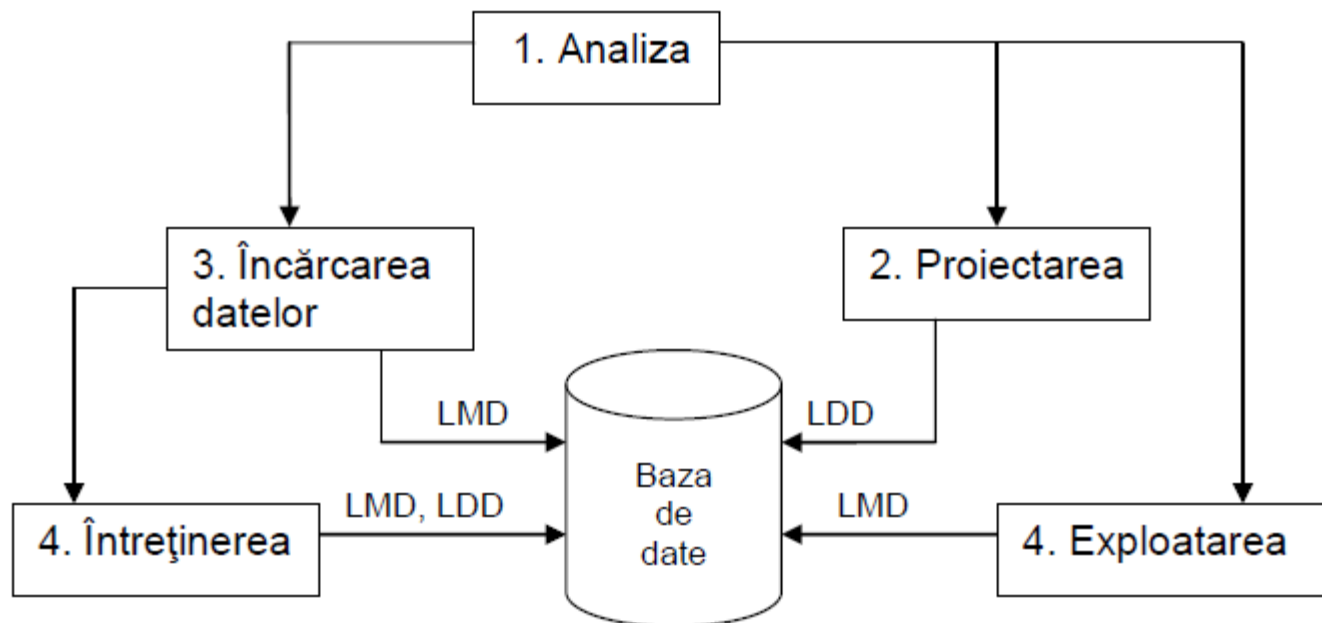
**D-durabilitate**-> baza de date trebuie să fie durabilă să păstreze toate modificările



# Realizarea unei BD

Realizarea unei baze de date presupune parcurgerea etapelor:

1. analiza domeniului (sistemului) pentru care se realizează baza de date;
2. proiectarea structurii bazei de date;
3. încărcarea datelor în baza de date;
4. exploatarea și întreținerea bazei de date.



# Realizarea BD

**1. Analiza sistemului** presupune stabilirea temei, analiza componentelor sistemului și analiza legăturilor (asocierilor) dintre aceste componente. Rezultatul analizei formează modelul conceptual al bazei de date.

## **2. Proiectarea structurii bazei de date**

Dacă etapa de analiză a modelului conceptual se realizează independent de un SGBD, prin etapa de proiectare a structurii bazei de date se trece la luarea în considerare a SGBD-ului cu ajutorul căruia va fi implementată și exploatată baza de date.

**Proiectarea structurii bazei de date** reprezintă transpunerea rezultatelor obținute în urma analizei modelului conceptual în termenii unui model al datelor suportat de un anumit SGBD. Compilatorul limbajului de descriere a datelor permite aducerea schemei bazei de date la nivelul la care să poată fi memorată în baza de date.

# Realizare unei BD

## **3. Încărcarea datelor în baza de date**

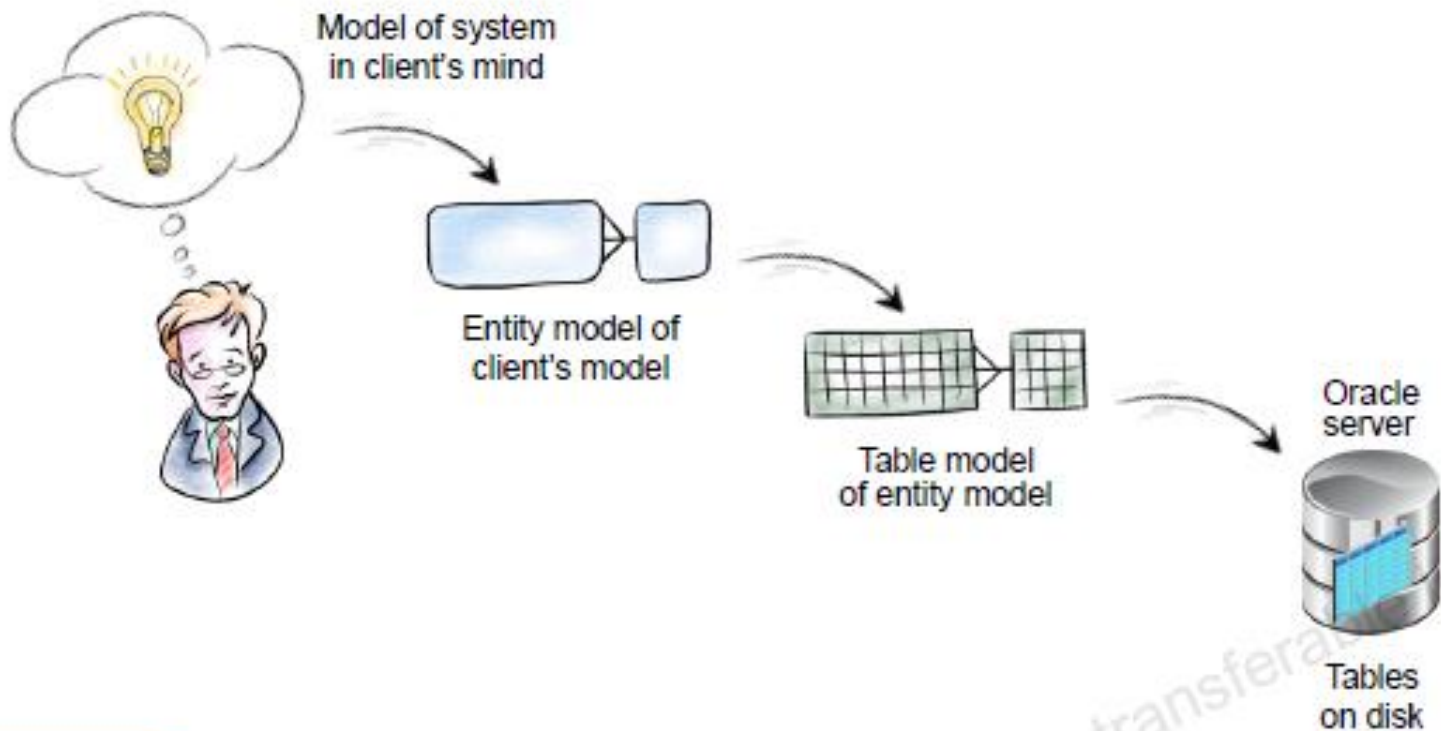
Este etapa în care se realizează popularea masivă cu date a bazei de date, activitate care trebuie să se efectueze cu un minim de efort.

## **4. Exploatarea și întreținerea bazei de date**

Exploatarea bazei de date de către diferiți utilizatori finali este realizată în scopul satisfacerii cerințelor de informare ale acestora. SGBD sprijină utilizatorii finali în exploatarea bazei de date, oferind o serie de mecanisme și instrumente cum ar fi limbajele de manipulare a datelor (LMD).

Întreținerea bazei de date reprezintă o activitate complexă, realizată, în principal, de către administratorul bazei de date și care se referă la actualizarea datelor din cadrul bazei de date.

## Data Models



# Construirea de diagrame entitate relatie (ER)

**Prima etapă pentru realizarea unei baze de date constă în *analiza sistemului*.**

***Se cunosc mai multe tehnici de analiză, dar cea mai des întâlnită este tehnica entitate-relație (entity-relationship).***

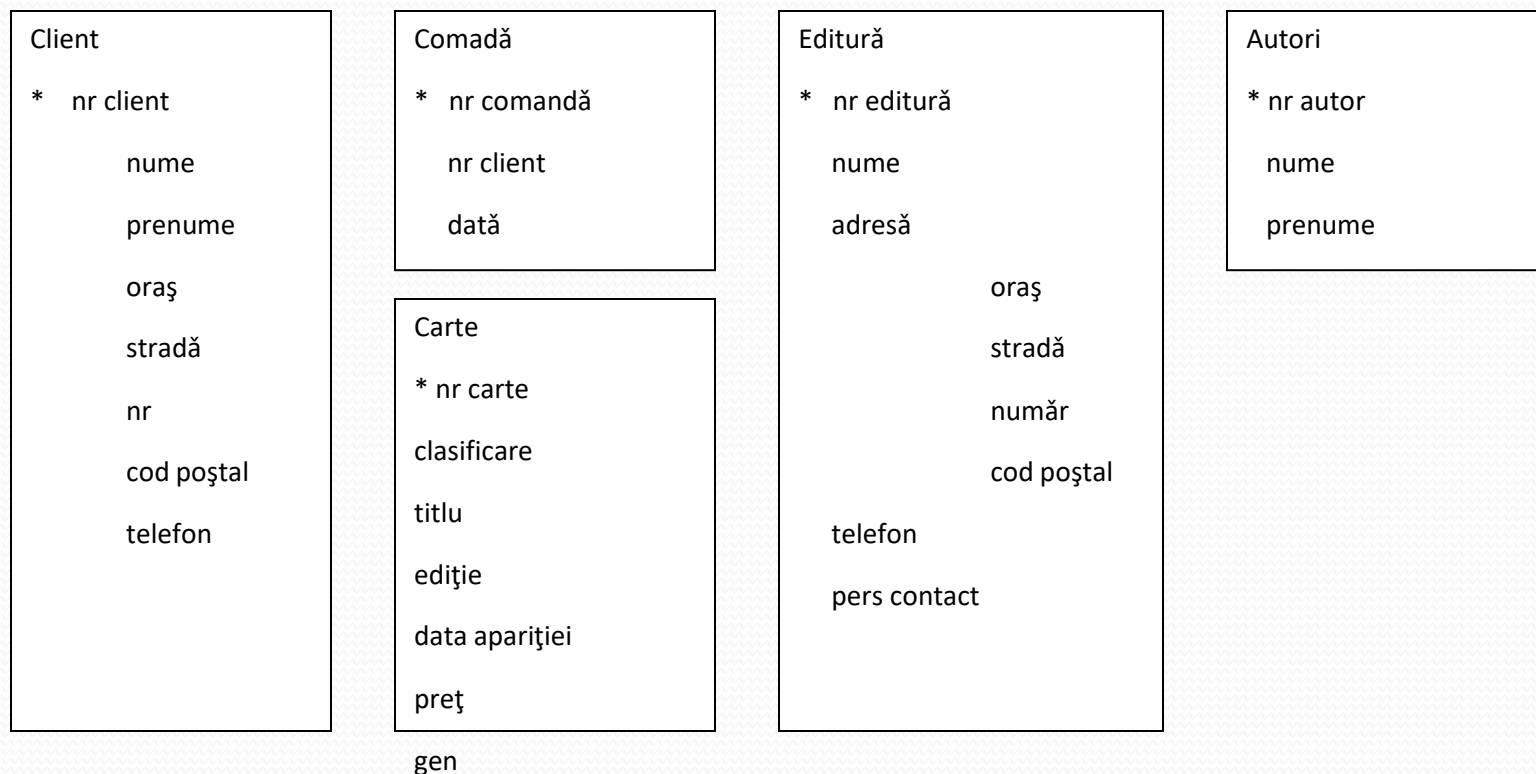
Prin tehnica entitate-relație (denumită și entitate-asociere) se construiește o diagramă entitate-relație (notată E-R) prin parcurgerea următorilor pași:

- a) identificarea entităților (componentelor) din sistemul proiectului;
- b) identificarea asocierilor (relațiilor) dintre entități și calificarea lor;
- c) identificarea atributelor corespunzătoare entităților;
- d) stabilirea atributelor de identificare a entităților.

# Diagrama ER

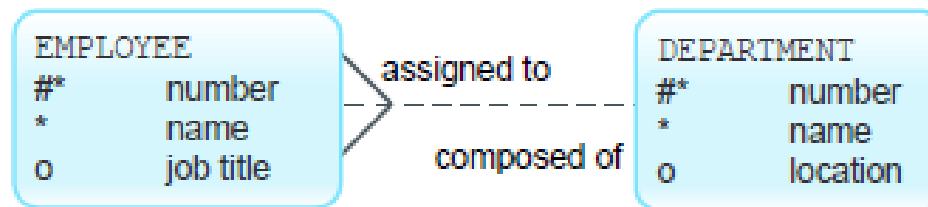
Entitatea se reprezintă sub forma unui dreptunghi în care sunt listate atributele și în care este pusă în evidență cheia.

Apoi vom vedea cum se reprezintă și relațiile între entități. societatea “Lectura inteligentă”.



## Entity Relationship Model

- Create an entity relationship diagram from business specifications or narratives:



- Scenario:
  - "... Assign one or more employees to a department. . ."
  - "... Some departments do not yet have assigned employees. . ."

## Relationships

Each direction of the relationship contains:

- A label: For example, *taught by* or *assigned to*
- An optionality: Either *must be* or *maybe*
- A degree: Either *one and only one* or *one or more*

Symbol	Description
Dashed line	Optional element indicating “maybe”
Solid line	Mandatory element indicating “must be”
Crow’s foot	Degree element indicating “one or more”
Single line	Degree element indicating “one and only one”



## **Tipuri de legături între tabele:**

În baza de date se vor introduce și relații între entitățile.

Vom enumera în continuare tipurile de relații care se pot stabili între entități:

### **Relații 1 la 1**

Este cazul relației din căsătoria dintre două posibile entități “bărbați” și “femei” (nu în țările islamice)

### **Relații 1 la n**

Sunt cele mai frecvente. Exemplu la o editură pot fi mai multe cărți, dar o anumită carte provine de la o singură editură.

### **Relații n la m**

Între comenzi și cărți. Într-o comandă pot fi mai multe cărți și aceeași carte poate să apară în mai multe comenzi. Acest tip de relații nu este ușor de mânuit și o astfel de relație este transformată în două relații de 1 la n introducând o nouă entitate. În cazul nostru noua entitate va fi “detaliu comandă”. Este obligatoriu ca în această entitate să avem cheile celor două entități între care există relația de n la m.

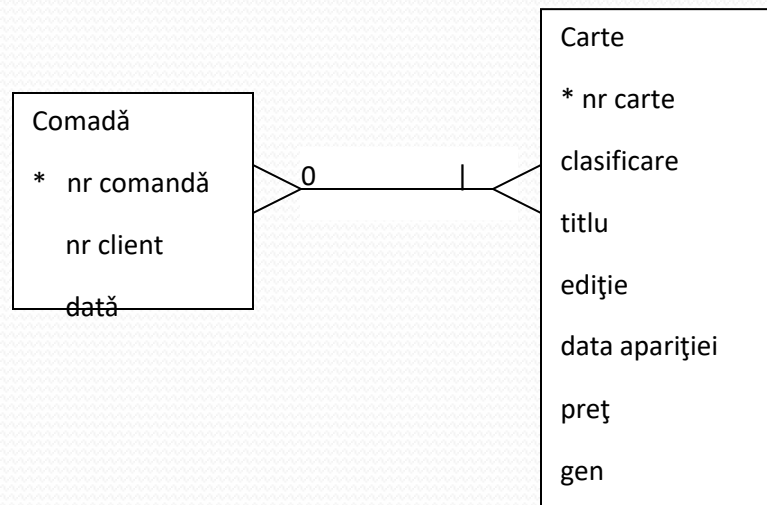
Relațiile vor fi descrise în diagramele E-R cu linii între entități care vor fi întretăiate de simboluri ale tipurilor. Aceste simboluri sunt descrise mai jos.

| | pentru una și numai una (instanță)

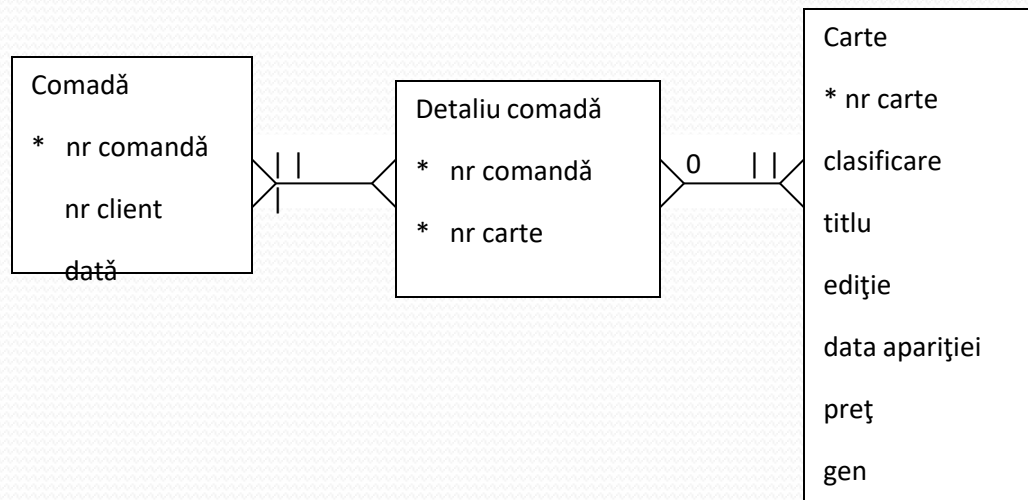
0 | pentru zero sau una

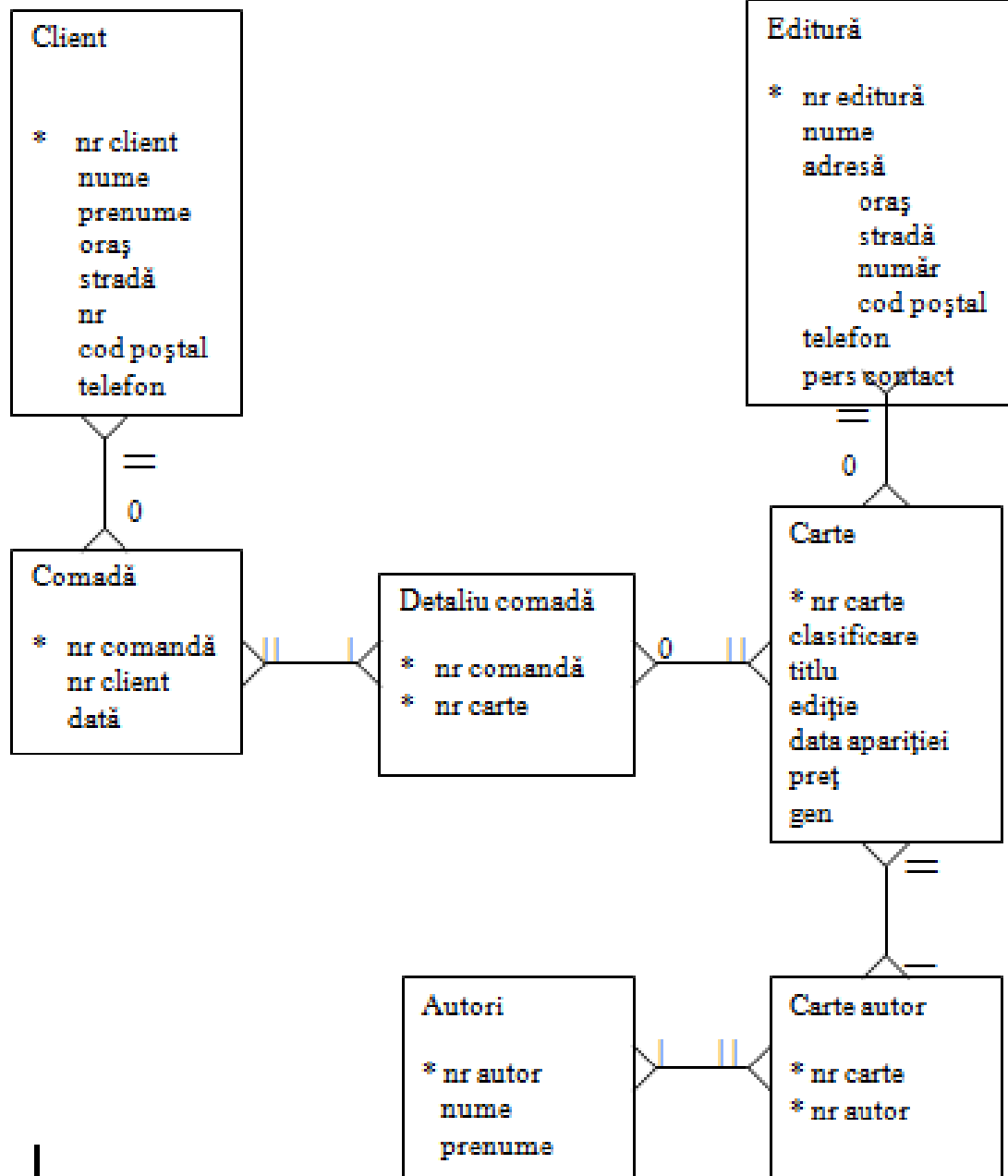
> | pentru una sau mai multe

> 0 pentru zero, una sau mai multe



O comandă conține cel puțin o carte, dar o carte poate să nu fie comandată de nimeni, totuși este o relație de tip n la m.  
Aceasta poate fi transformată prin introducerea entității “detaliu comandă”  
în:





- Să observăm că:
- cheia în entitatea comandă este numărul de comandă care apare în mod natural pe acest document.
- cheia în entitățile de legătură “detaliu comandă” și “carte autor”, introduce pentru a “distruge” relațiile n la m, este formată din concatenarea celor două chei ale entităților legate și nu crează probleme de actualizare.
- O bază de date relațională are la bază relația, care poate fi considerată ca un tabel (reprezentarea unei entități) cu linii (instanțe ale entităților), și coloane (atribute).
- Bineînțeles că între tabele trebuie să existe legături. Aceste legături sunt realizate prin disciplina:
- cheie primară → cheie străină

- Cheia primară este cheia unei relații, iar cheia străină este atributul (de obicei cu același nume) de același tip cu cheia primară și cu valori care se pun în corespondență cu cele ale cheii primare.
- Pentru ca o bază de date relațională să fie corectă, trebuie ca baza de date să îndeplinească anumite restricții:
- **restricția de unicitate a cheii**
- **restricția referențială** valorile cheii străine trebuie să figureze printre valorile cheii primare sau să aibă valoarea NULL
- **restricția entității** valorile cheii primare sunt unice și nu pot fi NULL
- **restricția de domeniu** valorile atributelor pot fi NULL sau din domeniul de definiție.

# SQL Statements Used in the Course

Data manipulation language  
(DML)

SELECT  
INSERT  
UPDATE  
DELETE  
MERGE

Data definition language (DDL)

CREATE  
ALTER  
DROP  
RENAME  
TRUNCATE  
COMMENT

Data control language (DCL)

GRANT  
REVOKE

Transaction control

COMMIT  
ROLLBACK  
SAVEPOINT

# Limbajul SQL

Limbajul SQL are următoarele comenzi, grupate în 4 categorii:

- –Data Manipulation Language (DML): SELECT, INSERT, UPDATE, DELETE, MERGE
- –Data Definition Language (DDL): CREATE, ALTER, DROP, RENAME, TRUNCATE, COMMENT
- –Data Control Language (DCL): GRANT, REVOKE
- –Transaction Control Language (TCL): COMMIT, ROLLBACK, SAVEPOINT



# LDD-limbaj definire datelor

- Limbajul de definire a datei SQL (LDD) ne permite crearea si distrugerea obiectelor baza de data (scheme, domenii, tabele, view-uri, indexuri). In aceasta sectiune, examinam, pe scurt, cum sa cream si sa distrugem scheme, tabele si indexuri.
- Principalele declaratii de definirii datei in SQL sunt:
- CREATE SCHEMA DROP SCHEMA
- CREATE DOMAIN ALTER DOMAIN DROP DOMAIN
- CREATE TABLE ALTER TABLE DROP TABLE
- CREATE VIEW DROP VIEW
- De asemenea, urmatoarele doua declaratii au fost prevazute de multe SGBD:
- CREATE INDEX DROP INDEX
- Aceste declaratii sunt folosite pentru a crea, schimba sau distruge structurile care alcatuiesc schema conceptuala.
-

# Create table

```
CREATE TABLE nume_tabel  
    (nume_coloana tip_data[NULL | NOT NULL][,...])
```

```
CREATE TABLE cadre_didactice(  
    nr_mat          CHAR(4)  NOT NULL,  
    nume            CHAR(15) NOT NULL,  
    prenume         CHAR(15) NOT NULL,  
    functia         CHAR(10));
```

# Create table

```
create table <table_name> (  
    <column1> <data type>,  
    <column2> <data type>,  
    <column3> <data type>,  
    ...  
);
```

Se observă imediat că trebuie să spunem ce valori pot lua attributele și cât loc o să ocupe aceste valori.

Aceasta înseamnă că trebuie stabilim **domeniul** atributelor.

Practic se utilizează următoarele simboluri:

TIP	Caracteristici
VARCHAR2(size)	sir de caractere de lungime variabila
NUMBER(p,s)	Numeric: p (precision)
LONG	sir de caractere de lung>2GB
DATE	data calendaristica
RAW(size)	date de tip binar. Maxim size este 2000 bytes.
LONG RAW	lungime variabila > 2GB se stocheaza grafice, sunete, documente
ROWID	adresa fiecarui rind din tabela
CHAR(size)	sir de caractere de lungime fixa
BLOB	binary large object (max 4GB) se stocheaza date nestructurate (text, imagine, video, date spatiale)
BFILE	contine un pointer catre un fisier binar stocat in afara bazei de date

# Drop Table

- DROP TABLE nume\_tabel [RESTRICT | CASCADE]  
DROP TABLE cadre\_didactice;
- ALTER TABLE nume\_tabel
- [ADD [COLUMN] nume\_coloana tip\_data[NOT NULL][UNIQUE]
- [DEFAULT optiune\_implicita][CHECK(conditie\_cautare)]]
- [DROP [COLUMN] nume\_coloana[RESTRICT | CASCADE]]
- [ADD [CONSTRAINT [nume\_restrictie]] definitie\_restrictie\_tabel]
- [DROP CONSTRAINT nume\_restrictie [RESTRICT | CASCADE]]
- [ALTER [COLUMN] SET DEFAULT optiune\_implicita]
- [ALTER [COLUMN] DROP DEFAULT]

# Modificare structura tabel

- ALTER TABLE cadre\_didactice  
ADD vechime INTEGER(2);

# Creare vederi - view-uri

- Un view este un tabel virtual, care nu exista din punct de vedere fizic. Este creat pe baza datelor din una sau mai multe tabele :
- Sintaxa :
- `CREATE VIEW nume-view AS`  
`SELECT coloane`  
`FROM tabele`  
`[WHERE conditii];`

# Update vederi - view-uri

- Se poate modifica definitia unui view fara al sterge (drop) folosind sintaxa CREATE OR REPLACE VIEW
- Sintaxa :  
  
CREATE OR REPLACE VIEW view\_name AS  
SELECT columns  
FROM table  
WHERE conditions;



# Stergere vederi - view-uri

- Se poate sterge un view folosind comanda DROP
- Sintaxa :
- `DROP VIEW view_name;`

# DML (Data Manipulation Language)

- SELECT interogarea datelor din baza de date,
- INSERT inserarea datelor intr-un tabel,
- UPDATE actualizarea datelor intr-un tabel,
- DELETE stergerea datelor dintr-un tabel.

## Exemple tabele

- cadre\_didactice (nr\_mat, nume, prenume, functia, vechime, salariu)
- sectii (cod\_sec, denumire, initiale)
- grupe (cod\_sec, cod\_gr, nr\_semigrupe, nr\_studenti)
- orase (cod, denumire);
- municipii(cod\_m, denum\_m);

## Basic SELECT Statement


- `SELECT` identifies the columns to be displayed.
- `FROM` identifies the table containing those columns.

```
SELECT  *|{[DISTINCT] column [alias],...}  
FROM    table;
```

# COMANDA SELECT

**SELECT** – realizeaza selectia si regasirea datelor din tabele


```
SELECT [DISTINCT] { * , tabelă1.câmp1 [alias] , expresii AS ALIAS ...}  
FROM tabelă1, tabelă2,....  
WHERE {condiții, precizarea legăturilor dintre tabele}  
GROUP BY tabelă .câmp  
HAVING {condiții impuse valorilor de grup}  
ORDER BY tabelă .câmp ASC/DESC;
```



SELECT	specifică attributele selectate;
DISTINCT	suprimă valorile duplicate;
*	selectează toate attributele;
atribut	selectează coloana numită;
expresie	permite construirea de expresii si valori noi
alias	denumiri pentru attributele selectate;
FROM tabele	specifică tabelele ce conțin coloanele selectate.
WHERE	clauza permite specificarea condițiilor si a criteriilor de selectie a datelor
GROUP BY	se precizeaza campul dupa care vor fi grupate datele in cazul expresiilor si functiilor de grup (SUM(), AVG(), COUNT(), MIN(), MAX())
HAVING	in cazul functiilor de grup conditiile impuse acestora se precizeaza in clauza HAVING
ORDER BY	precizeaza ordonarea in functie un anumite campuri ascendent (ASC) –implicit sau descendent (DESC)

# Comanda Select

- SELECT [DISTINCT]{\* | [expresie\_coloana [AS nume\_nou]][,...]}
- FROM nume\_tabel [alias][,...]
- [WHERE conditie]
- [GROUP BY lista\_coloane] [HAVING conditie]
- [ORDER BY lista\_coloane]

- 
- **Frazele SQL:**
  - Nu sunt case sensitive;
  - Pot fi scrise pe mai multe linii;
  - Cuvintele cheie nu pot fi prescurtate sau scrise pe mai multe linii.

In construirea frazelor SQL se utilizeaza urmasorii operatori :

<, >, =, >=, <=, NOT	Operatori de comparatie
BETWEEN ... AND ...	între două valori (inclusiv).
<b>SELECT codprodus, cant FROM rindcom WHERE cant BETWEEN 10 AND 15;</b>	
IN(listă)	egal cu oricare valoare din listă
<b>SELECT codagent, numeagent FROM agenti WHERE codagent IN ('1', '3');</b> <b>SELECT codprod, cant FROM rindcom WHERE cant IN (400,500,1000);</b>	
LIKE	similar cu un șablon % - oricâte caractere; _ - un caracter;
<b>SELECT denfirma, loc FROM firme WHERE loc LIKE 'B%';</b> <b>SELECT denfirma, loc FROM firme WHERE loc LIKE '_I%';</b>	
IS NULL	are valoarea NULL
<b>SELECT denfirma, loc FROM firme WHERE loc IS NULL;</b>	



# SELECT Statement Syntax

```
SELECT column1, column2, column3 [, . . .]  
FROM table1 [JOIN table2 ON col1 relational_operator col2]  
    [. . . JOIN . . .]  
WHERE condition_1 [AND | OR] condition_2 [. . .]  
GROUP BY col1 [, col2 . . .]  
HAVING groupby_expression relational_operator  
    [constant | expression]  
ORDER BY [column_number | column1 | groupby_expr1 ], . . .
```

# Exemplu :

- create table Student

```
( id_student number constraint student_pk primary  
key,
```

```
  Nume varchar2(30),
```

```
  Prenume varchar2(30),
```

```
  Data_nasterii date);
```

-putem adauga in plus ce date consideram semnificative pentru student

# Comanda Insert – inserare date in tabel

- insert into Student values (2, 'Popescu','Claudia','01-jan-1984');

## Quiz

Identify the **SELECT** statements that execute successfully.

- a. 

```
SELECT first_name, last_name, job_id, salary*12,  
       AS Yearly Sal  
FROM   employees;
```
- b. 

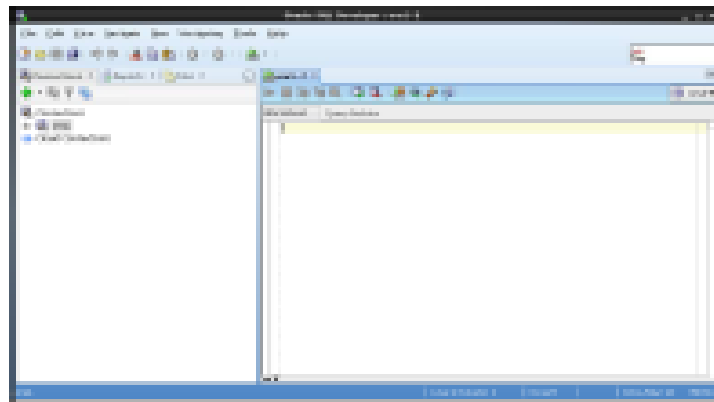
```
SELECT first_name, last_name, job_id, salary*12  
       "yearly sal"  
FROM   employees;
```
- c. 

```
SELECT first_name, last_name, job_id, salary AS  
       "yearly sal"  
FROM   employees;
```
- d. 

```
SELECT first_name+last_name AS name, job_id,  
       salary*12 yearly sal  
FROM   employees;
```

# Mediu de lucru:

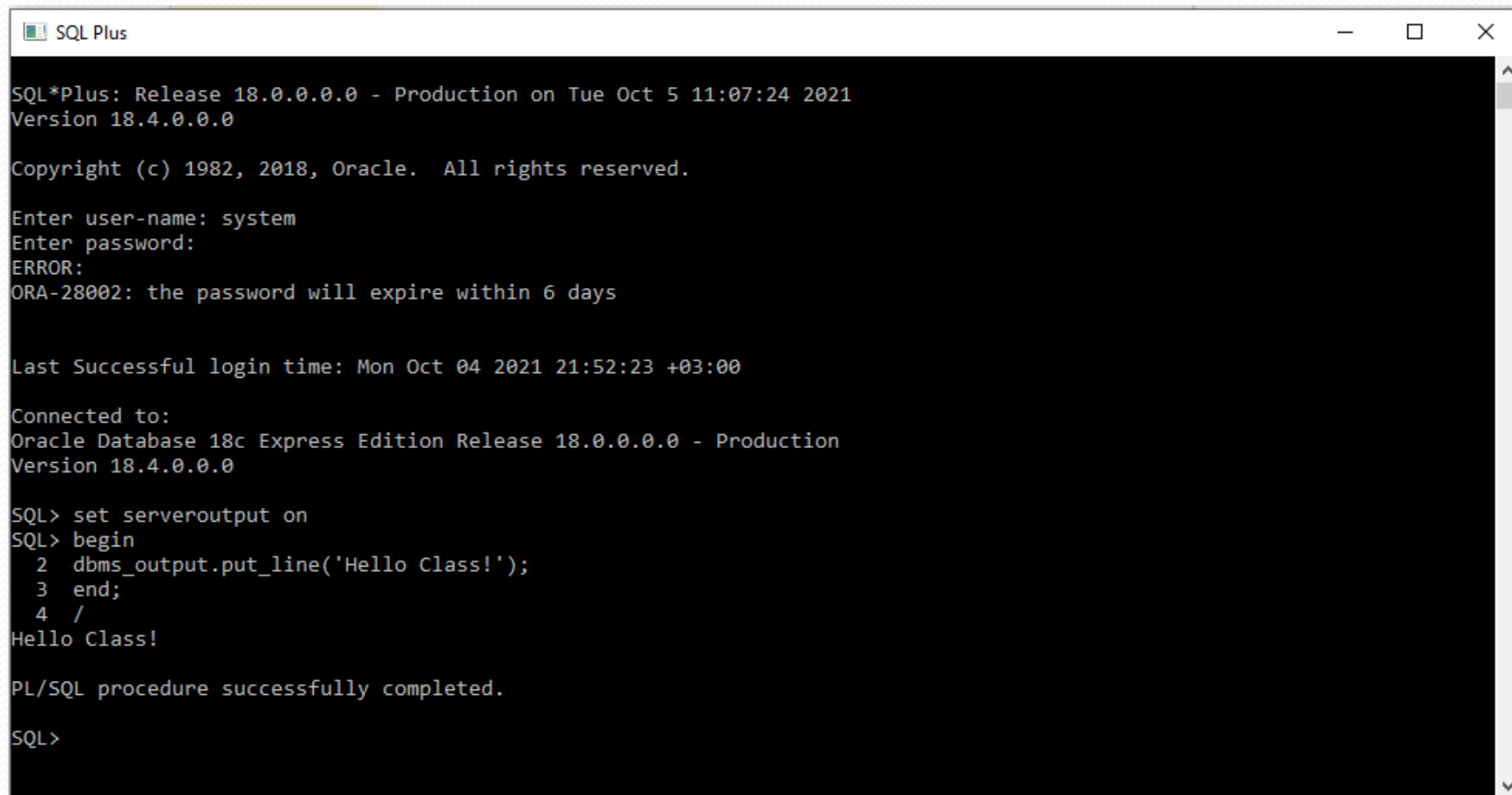
- -se poate instala baza de date Oracle local si SQL Developer
- SQL Plus, interfata in linie de comanda



SQL Developer



SQL\*Plus

A screenshot of a Windows-style application window titled "SQL Plus". The window has a white title bar with standard minimize, maximize, and close buttons. The main content area has a black background with white text. The text shows the SQL\*Plus startup sequence: release information, copyright notice, login prompts, an error message about password expiration, connection details, and the execution of a PL/SQL block that prints "Hello Class!".

```
SQL*Plus: Release 18.0.0.0.0 - Production on Tue Oct 5 11:07:24 2021
Version 18.4.0.0.0

Copyright (c) 1982, 2018, Oracle. All rights reserved.

Enter user-name: system
Enter password:
ERROR:
ORA-28002: the password will expire within 6 days

Last Successful login time: Mon Oct 04 2021 21:52:23 +03:00

Connected to:
Oracle Database 18c Express Edition Release 18.0.0.0.0 - Production
Version 18.4.0.0.0

SQL> set serveroutput on
SQL> begin
  2  dbms_output.put_line('Hello Class!');
  3  end;
  4  /
Hello Class!

PL/SQL procedure successfully completed.

SQL>
```

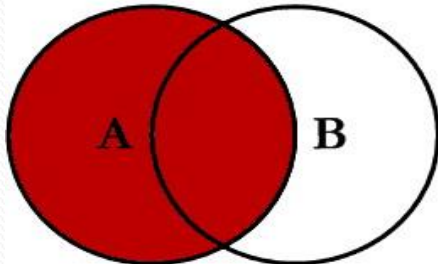
# https://livesql.oracle.com

The screenshot shows the Oracle Live SQL web application. The browser address bar displays `https://livesql.oracle.com/apex/?p=590:1000`. The application has a dark sidebar on the left with a menu containing: Home, SQL Worksheet, My Session, Schema, Quick SQL, My Scripts, My Tutorials, and Code Library. The main content area has a dark header with the "Live SQL" logo and "Help" and "Sign In" links. Below the header, there is a red icon with a white prompt character (>\_). The main heading is "Learn and share SQL" with the subtitle "Now running on Oracle Database 19c". A search bar with the placeholder "Search scripts and tutorials" is present. Below the search bar are two buttons: "Start Coding Now" (green) and "View Scripts and Tutorials" (grey). A section titled "Featured Scripts and Tutorials" contains four items in a 2x2 grid:

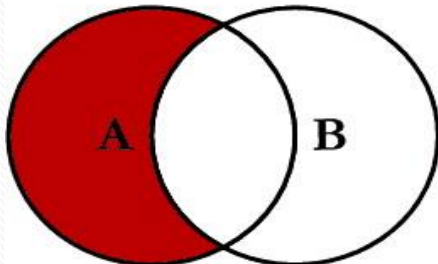
Featured Scripts and Tutorials	
<b>Introduction to SQL</b> This tutorial provides an introduction to the Structured Query Language (SQL), learn how to create tables with primary keys, columns, constraints, ind...	<b>Simple Explain Plan</b> This script explains the plan for a query of the sh.sales and sh.products tables.
<b>19C LISTAGG DISTINCT</b> The LISTAGG aggregate function now supports duplicate elimination by using the new DISTINCT keyword. The LISTAGG aggregate function orders the rows...	<b>19c JSON_OBJECT</b> Syntax simplifications are offered for SQL/JSON path expressions, SQL/JSON generation with function json_object, and field projection with SQL/JSON ne...

# Tipuri de join-uri

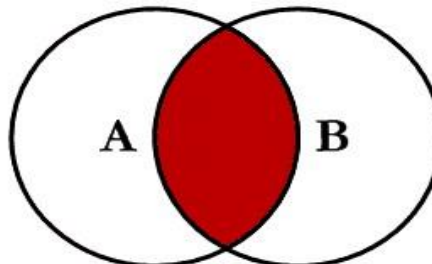
## SQL JOINS



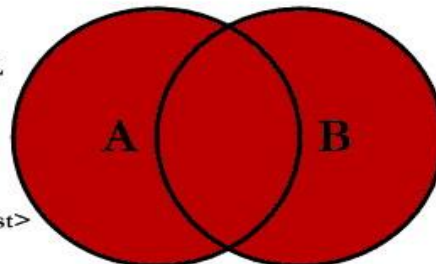
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key
```



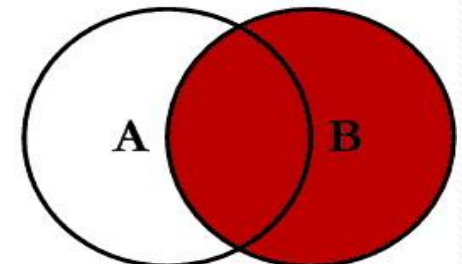
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key  
WHERE B.Key IS NULL
```



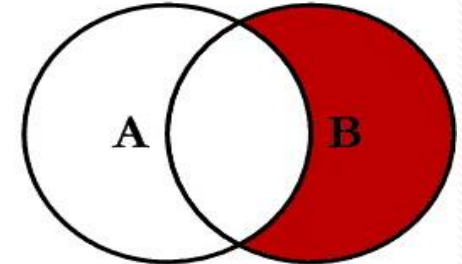
```
SELECT <select_list>  
FROM TableA A  
INNER JOIN TableB B  
ON A.Key = B.Key
```



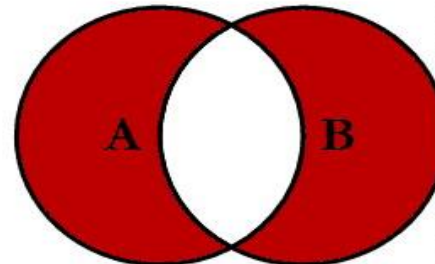
```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key
```



```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key
```



```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key  
WHERE A.Key IS NULL
```



```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key  
WHERE A.Key IS NULL  
OR B.Key IS NULL
```



## **Bibliografie :**

- 1) Connolly Thomas, Carolyn Begg, Database Systems – A practical approach to design, implementation and management, sixth edition, Pearson 2015
- 2) Fotache M.– *SQL. Dialecte DB2, Oracle, PostgreSQL si SQL Server*, Editia a II-a, revazuta si adaugitaed. Polirom, 2009
- 3) Geschwinde Ewald și Schönig Hans-Jürgen, *PostgreSQL Developer's Handbook*, PostgreSQL București, Teora, 2003.
- 4) Forta Ben, *SQL in lectii de 10 minute*, ed. Teora, 2005
- 5) Iacob Paul, Baze de date, Universitatea Transilvania din Brasov, note de curs
- 6) Lungu I., C. Bodea ș. a. - “*Baze de date – organizare, proiectare și implementare*“, ed. ALL, 1995.
- 7) Lungu I., I. Iacob – “*Aplicații informatice utilizând SGBD Oracle*“, ed. ProUniversalis, 2005
- 8) Oracle Student Guides, Activity Guides , Introduction to SQL
- 9) Oracle Academy