

# Baze de Date

Cap. 1. BD. SGBD. Algebra relațională



2025 UPT

Conf. Dan Pescaru

# Organizare curs

---

## 1. Curs BD

- Conf.dr.ing. Dan Pescaru
- 2C+I - Miercuri, 8:00-10:00 A106
- 2TI – Vineri 10:00-12:00 ASPC

## 2. Laborator

- B623 / B529
- Oracle Cloud

## 3. Evaluaare

- Nota Laborator (1/2)
- Nota Examen (1/2)

# Conținut

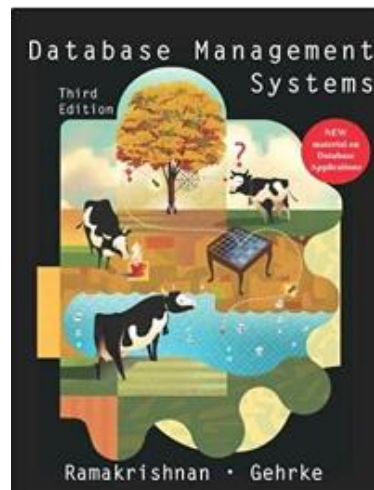
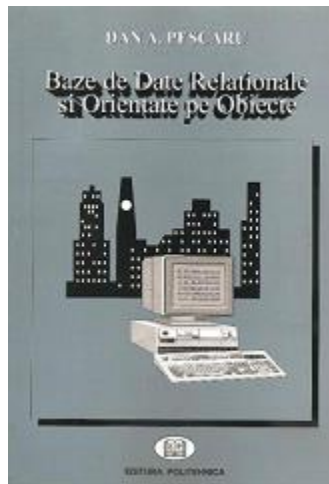
---

1. Introducere. BD. SGBD. Algebra relațională.
2. SQL utilizare DDL. Gestiune date. Interogari active.
3. SQL implementare Proiecție, Selecție și Join.
4. SQL utilizare subinterogări. Operații pe mulțimi.
5. SQL agregare date. Obținerea informațiilor statistice.
6. Implementare aplicațiilor client folosind Oracle APEX.
7. Implementarea unei BD Web folosind MySQL și PHP.
8. Introducere in modelarea si organizarea datelor intr-o BD relațională.
9. Optimizarea accesului la o BD. Utilizare indecși.
10. Optimizarea interogarilor SQL. Generare plan execuție
11. Introducere in administrare BD.

# Bibliografie

---

1. D. Pescaru, "Baze de Date Relationale si Orientate pe Obiecte", Editura Politehnica, Timisoara, 2001, ISBN 973-8247-53-5.
2. R- Ramakrishnan și J. Gehrke, "Database Management Systems", 3rd edition, 2003, ISBN 007-2465-63-8, McGraw-Hill.
3. Oracle® 23c Database SQL Language Reference, Dec. 2023.  
<https://docs.oracle.com/en/database/oracle/oracle-database/23/sqlrf/>





# Exemplele din curs/laborator

---

## 1. Tehnologii utilizate

- Oracle, Oracle XE, APEX
- MySQL Community Server
- Visual dBase Plus



## 2. Limbaje

- De interogare date: SQL, xBase
- De programare:  
Java, Javascript, PHP



# Managementul datelor

---

1. Soluția simplă: fișierul
2. Organizarea fișierelor: structuri de directoare
3. Avantaje
  1. Simplitate
  2. Acces liniar (text) sau secvențial (binar)
4. Dezavantaje:
  1. Căutarea datelor în fișiere – algoritmi complecși specifici fiecărei aplicații
  2. Slaba protecție a datelor
  3. Controlul accesului și securitate – la nivelul SO

# DB și SGBD

---

1. Soluția: utilizarea unei baze de date
2. Baza de date (**BD**) = o alternativă de stocare adecvată pentru colecții de date de mari dimensiuni. Include organizarea fizică eficientă pe suport extern, algoritmi avansați de căutare, regăsire, protecție și mecanisme de securitate a datelor
3. Sisteme de gestiune a bazelor de date (**SGBD**) = un sistem software proiectat să asigure stocarea și gestionarea unor baze de date

# De ce sunt importante bazele de date?

---

1. Largă răspândire in lume: majoritatea oamenilor le utilizează în fiecare zi
  1. Majoritatea site-urilor Web utilizează BD (magazine virtuale, forumuri, broadcasting etc.)
  2. Sistemele de telefonie
  3. Sistemele bancare
  4. Sistemele de facturare din magazine, etc.
2. Există o cerere constantă de specialiști în domeniul bazelor de date (administratori, proiectanți, analiști, programatori etc.)



# Meserii legate de BD

---

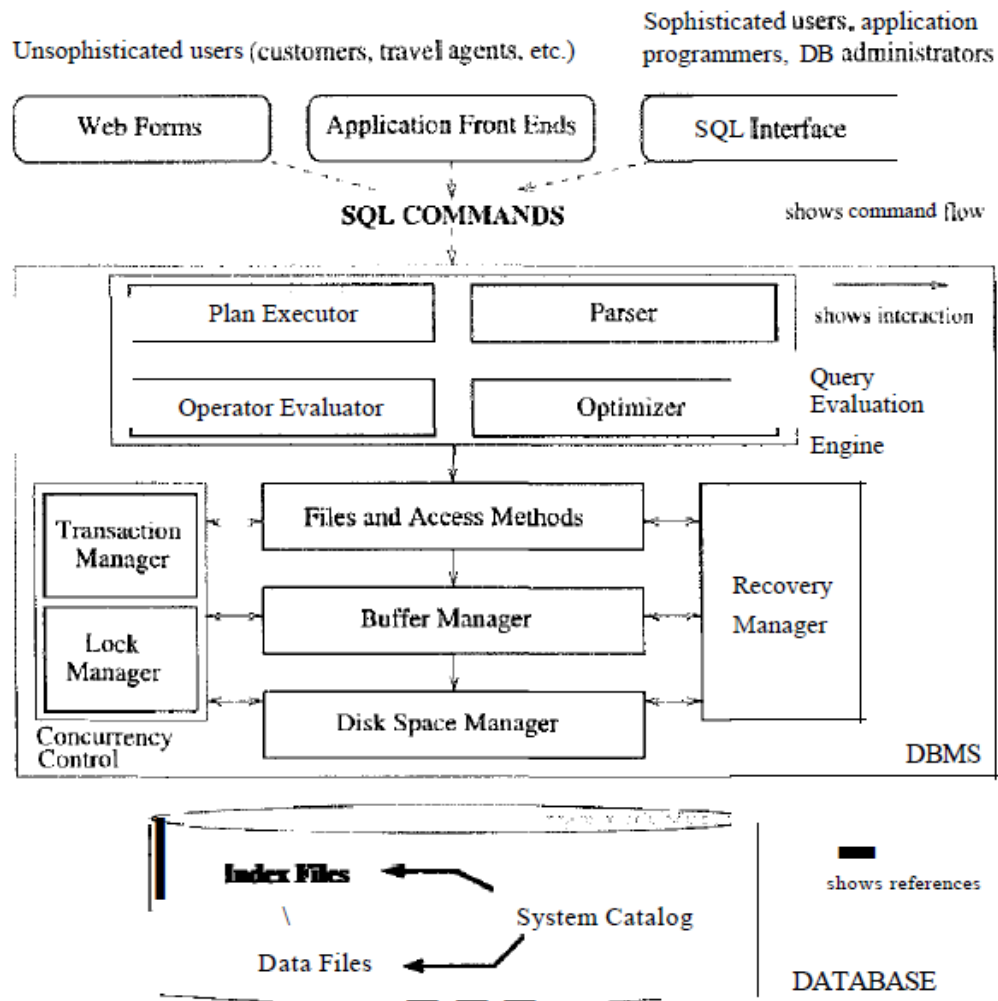
1. Implementator SGBD (angajați de Oracle, Microsoft, IBM etc.)
  - Creează sistemele de gestiune a bazelor de date
2. Utilizatori finali – un număr cvasi-infinit
  - Manageri, secretare, contabili, bancheri, etc.
3. Programatori de aplicații (SQL + limbaje imperative)
  - Dezvoltă aplicații prin care utilizatorii interacționează cu BD
4. Administratori BD
  - Proiectează schemele conceptuale și fizice ale BD
  - Gestionează securitatea datelor
  - Asigură recuperarea datelor și disponibilitatea lor
  - Optimizează performanța BD (DB benchmarking/tunning)

# Avantajele utilizării SGBD

---

1. Independența datelor de aplicații
2. Eficiența accesului la date
  - Utilizare sisteme complexe de indexare
  - Optimizarea interogărilor
3. Suport pentru tehnici RAD
4. Asigură integritatea datelor
5. Asigură securitatea datelor
6. Scalabilitate

# Aritectura unui SGBD



\*Ref: Ramakrishnan, Gehrke, "Database Management Systems".

# Scurtă istorie a domeniului BD

## 1. Modelul ierarhic – 1960

- Structură arborescentă (relații 1-la-N)
- Ex. IBM IMS – Information Management System – utilizat în programul Apollo (inventarul pieselor pentru racheta Saturn V); American Airlines + IBM – SABRE



NASA: <http://www.hq.nasa.gov/>



## 1. Modelul rețea – 1962

- Structură de garf denormalizată (N-la-M)
- IDS (Integrated Data Store) - realizat de Charles Bachman pentru General Electric (în Cobol)
  - > A primit premiul ACM Turing ( $\Leftrightarrow$  Nobel) în 1970 pentru activitatea sa în domeniul bazelor de date

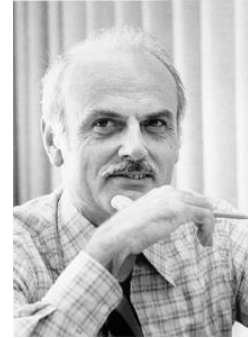


# Actual - modelul relațional

---

## 1. Modelul relațional – 1970

- Edgar Codd, la IBM's San Jose Research Laboratory
- A câștigat Turing Award in 1981
- Noutatea: are la bază un model matematic
- Sisteme navigaționale (imperative) pentru desktop
  - xBase: FoxPro, Clipper, Visual dBase **dBASE™**
- Sisteme declarative – SQL (IBM's System R project)
  - Oracle, IBM DB2, Ms SQL Server, MySQL







# Modele alternative

---

## 1. Modelul obiectual

- Obiecte persistente în limbajele OO
- ODMG - Object Query Language (OQL) 
- Ex: Jasmine (Fujitsu), Versant , O2, POET, ObjectStore, JADE 
- Soluție de compromis: model obiectual-relațional

## 2. Modele NoSQL (post-relaționale)

- Diverse: de la modele de tip cheie-valoare la BD orientate pe documente
- Ex: MongoDB, Redis, Apache Cassandra etc.)



**Cassandra**



**mongoDB**



# MODELUL RELAȚIONAL

---

1. BD relațională: o colecție de relații (tabele) și legăturile dintre ele

2. Relație (tabel):

- Schemă: nume relație, attribute (coloane) și domeniul atributelor (tipul datelor)
- Ex: `Studenti(marca:String, nume:String, an:Integer, nota:Real)`
- Instanță: tabelă fizică, având un număr fix de coloane și un număr variabil de rânduri (înregistrări)
  - Numărul de coloane = GRADUL relației (arietatea)
  - Numărul de rânduri = CARDINALITATEA relației
  - Obs.: Rândurile (înregistrările) NU sunt ordonate!  
Coloanele SUNT ordonate (ex. pt. SQL INSERT)

# Instanță (tabelă)

---

## Student

marca	nume	an	media
AC2153	Pop Angela	2	8.50
AC1078	Avram Ioan	1	9.35
AC2056	Ionescu Mihai	2	7.80
AC3098	Georgescu Ana	3	9.00
AC3023	Mihu Andrei	3	6.30

- Relația Student
  - Grad: 4
  - Cardinalitate: 5
- Obs: *Conține doar înregistrări distincte!*

# Proiectare unei BD

---

1. Analiza cerințelor: primul pas la proiectarea BD

2. Se utilizează un model semantic

- Abstract, de nivel înalt, capabil să descrie semantica datelor (indiferent de implementare)
- Servește la înțelegerea datelor și a rolului lor

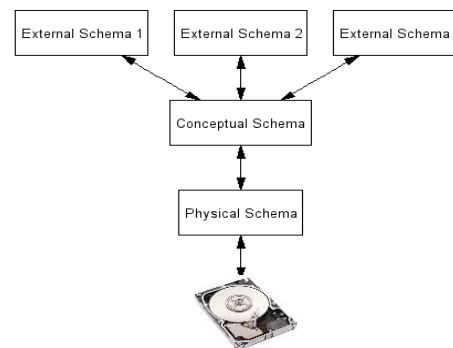
3. Limbaje de modelare semantice

- Entity Relationship (ER) – utilizează imagini pentru a descrie entități și relațiile dintre ele (modelare la nivel logic)
- UML – tot grafic dar mai general decât ER (modelare business-operare, sistem-integrare, logic, fizic-stocare, hardware)

# Nivele de abstractizare la implementare

1. Datele sunt descrise într-un SGBD pe trei nivele de abstractizare

- **Fizic**: cum și unde sunt stocate fizic datele
- **Conceptual**: modelul datelor
- **Extern**: model simplificat destinat unei anumite aplicații





# Constrângeri de integritate

---

1. O colecție de condiții (expresii) logice care trebuie să fie îndeplinite de orice instanță a relației
2. Constrângerile de integritate:
  - Specificate la crearea schemei BD (tabelelor)
  - Verificate automat la orice modificare a datelor
3. O instanță este legală ("într-o stare consistentă") dacă satisface toate constrângerile de integritate. Verificarea consistenței se face automat de SGBD
4. Obs.: Verificarea constrângerilor previne în mare măsură introducerea de date eronate

# Chei

---

1. Un set de attribute formează o cheie a unei relații dacă:

- a. Nu există două înregistrări care să aibă toate valorile atributelor identice (asigură unicitatea)
- b. Orice subset nu respectă proprietatea anterioară (cheia este minimală)

2. Supercheie: un set de attribute care conține cel puțin o cheie

# Cheie Primară (PK)

---

1. Cheie primară: o cheie aleasă de proiectant din setul de chei a unei relații, a.î. să satisfacă condițiile:

- Este scurtă (număr minim de attribute – ideal unul singur)
  - Obs: (neuzual) o singură cheie cu toate câmpurile relației
- Este reprezentativă în contextul problemei (ex. *marca* vs. *cnp*)
- Nu admite valori NULL

2. Alternative: crearea unei chei artificiale (de obicei numerică de tip *AutoIncrement*)

# Integritatea referențială

---

1. Cheie externă (FK) – leagă logic două tabele (va corespunde cheii primare din cea de a doua tabelă)
2. Utilizată pentru verificarea integrității referențiale
  - La adăugare în tabelă: valoarea cheii trebuie să existe în tabela referită
  - La ștergerea unei înregistrări din tabela referită: se va evita crearea de înregistrări "orfane"
  - La modificarea valorii unei chei externe sau a unei chei primare în tabela referită

# Asigurarea integrității referențiale

---

1. La adăugare (INSERT)
  - Se blochează adăugarea unei chei fără corespondent
2. La ștergere (DELETE) se poate alege
  - "Cascade delete related records"
  - "Avoiding operation that breaks constraints"
  - "Voiding the reference" (nu prea des)
3. La modificare (UPDATE) se poate alege
  1. "Cascade update related records"
  2. "Avoiding operation that breaks constraints"
4. Probleme: blocaje (deadlocks – referințe încrucișate),  
Soluție: utilizarea tranzacțiilor (verificare întârziată)



# Integritate referențială. Example

**Student**

marca	nume	an	media
AC2153	Pop Angela	2	8.50
AC1078	Avram Ioan	1	9.35
AC2056	Ionescu Mihai	2	7.80

**Contract**

cid	marca	lab	ex
PLA2	AC2153	9	8
UC1	AC1078	10	9
SO2	AC2056	8	7

1. **INSERT** un nou contract: verifică **marca** in tabela Student
2. **DELETE** șterge un student: verifică **marca** in Contract
3. **UPDATE** modifică **marca** in Student: verifică **marca** in Contract / modifică **marca** in Contract – verify **marca** in Student

# Algebra Relațională

---

1. Derivată din Algebra "clasică"
2. Operatori de bază:
  - Proiecție ( $\Pi$ ) – selectează coloanele specificate
  - Selecție ( $\sigma$ ) – selectează înregistrările specificate
  - Produs Cartezian ( $\times$ ) – combină două relații
  - Diferență mulțimi ( $\setminus$ ) – înregistrările din  $R_1$ , care nu  $\in R_2$
  - Reuniune ( $\cup$ ) – înregistrările din  $R_1$  și  $R_2$  (fără duplicate)
3. Operatori adiționali:
  - JOIN ( $\bullet$ ), Intersecție ( $\cap$ ), Divizare ( $\div$ ), Redenumire
4. Din moment ce orice operator returnează o relație, ei pot fi compuși în expresii (Algebra este "închisă")

# Proiecția

---

1. Proiecția ( $\Pi$ ) – Selectează doar attributele specificate în lista de proiecție
2. Schema rezultatului va conține exact câmpurile din lista de proiecție, cu același nume ca și cel din relația de intrare
3. Operatorul de proiecție trebuie să elimine duplicatele! (*De ce ??*)
4. Obs: sistemele reale în general nu elimină duplicatele decât dacă se specifică de către programator. (*De ce nu ??*)
5. Ex:  $\Pi_{\text{media}}(\text{Student}) = \{10.0, 9.5, 8, 6.5\}$

# Selecția

---

1. Selecția ( $\sigma$ ) – selectează rândurile care satisfac condiția de selecție
2. Condiția de selecție: simplă (compară attribute cu valori) sau complexă (utilizează conectori logici precum *AND*, *OR*, *NOT*)
3. Schema rezultatului va fi identică cu cea a relației de intrare
4. Obs: Nu apar duplicate în rezultat! (*De ce??*)
5. Cardinalitate rezultat  $\leq$  Cardinalitate intrare
6. Ex:  $\sigma_{\text{media} < 9.0}(\text{Student})$

# Reuniunea, Intersecția, Diferența

1. Toate au doi operanzi (relații) compatibile ("union-compatible"):

- Au același număr de atribute
- Atributele corespondente au același tip

**S1**

marca	nume	an	media
AC2034	Popescu Ion	2	9.25
AC2056	Ionescu Vasile	2	8.70

marca	nume	an	media
AC2056	Ionescu Vasile	2	8.70

**$S1 \cap S2$**

marca	nume	an	media
ET3045	Vasilescu Ana	3	9.00
ME1078	Pop Angela	1	8.25
AC1012	Ticu Gelu	1	9.50

**$S2 / S1$**

**S2**

marca	nume	an	media
AC2056	Ionescu Vasile	2	8.70
ET3045	Vasilescu Ana	3	9.00
ME1078	Pop Angela	1	8.25
AC1012	Ticu Gelu	1	9.50

marca	nume	an	media
AC2034	Popescu Ion	2	9.25
AC2056	Ionescu Vasile	2	8.70
ET3045	Vasilescu Ana	3	9.00
ME1078	Pop Angela	1	8.25
AC1012	Ticu Gelu	1	9.50

**$S1 \cup S2$**



# Produsul cartezian

1. Fiecare rând din R1 este împerecheat cu fiecare rând din R2
2. Schema rezultatului va avea câte un atribut pentru fiecare atribut din cele două relații, cu același nume (conflictele rezolvate prin redenumire)

**Student**

marca	nume	an	media
AC2056	Ionescu Vasile	2	8.70
ET3045	Vasilescu Ana	3	9.00
ME1078	Pop Angela	1	8.25
AC1012	Ticu Gelu	1	9.50

**Masina**

marca	nrm	tip
AC2056	TM06ABC	Dacia Logan
ET3045	TM01AAA	Renault Megane
ME1078	TM08BBB	Ford Focus
AC1012	TM09PPP	Opel Corsa

(marca)	nume	an	media	(marca)	nrm	tip
AC2056	Ionescu Vasile	2	8.70	AC2056	TM06ABC	Dacia Logan
AC2056	Ionescu Vasile	2	8.70	ET3045	TM01AAA	Renault Megane
AC2056	Ionescu Vasile	2	8.70	ME1078	TM08BBB	Ford Focus
AC2056	Ionescu Vasile	2	8.70	AC1012	TM09PPP	Opel Corsa
ET3045	Vasilescu Ana	3	9.00	AC2056	TM06ABC	Dacia Logan
ET3045	Vasilescu Ana	3	9.00	ET3045	TM01AAA	Renault Megane
...	...	...	...	...	...	...

**SxM**

# Join

1. JOIN = selecție cu o condiție de Join peste un produs cartezian
2.  $R1 \circ_{Jcond} R2 = \sigma_{Jcond}(R1 \times R2)$

**Student**

marca	nume	an	media
AC2056	Ionescu Vasile	2	8.70
ET3045	Vasilescu Ana	3	9.00
ME1078	Pop Angela	1	8.25
AC1012	Ticu Gelu	1	9.50

**Masina**

marca	nrm	tip
AC2056	TM06ABC	Dacia Logan
ET3045	TM01AAA	Renault Megane
ME1078	TM08BBB	Ford Focus
AC1012	TM09PPP	Opel Corsa

(marca)	nume	an	media	(marca)	nrm	tip
AC2056	Ionescu Vasile	2	8.70	AC2056	TM06ABC	Dacia Logan
ET3045	Vasilescu Ana	3	9.00	ET3045	TM01AAA	Renault Megane
ME1078	Pop Angela	1	8.25	ME1078	TM08BBB	Ford Focus
AC1012	Ticu Gelu	1	9.50	AC1012	TM09PPP	Opel Corsa

**Student**  $\circ_{Student.marca=Masina.marca}$  **Masina**

# Join

---

1. Aceiași schemă a rezultatului ca și la produs cartezian dar mai puține înregistrări. Se poate procesa mai eficient. (*De ce?*)
2. Tipuri de JOIN:
  - Theta-join – condiție generală de Join ( $=, <, <>$ )
  - Equi-join – caz special care testează doar egalitate ( $=$ )
  - Natural-join – Equi-join pe toate câmpurile comune (implicit în lipsa condiției de  $\bowtie$ )
  - Self-join – R1 și R2 sunt același relație

# Example

Rezervări bărci într-un port – Schema BD apare in ref. [1]

**Marinar (M)**

<u>mid:int</u>	nume:string	rang:int	varsta:real
31	Horatiu	1	33.0
...			
58	Oana	8	54.0

**Barca (B)**

<u>bid:int</u>	bnume:string	culoare:string
103	Poseidon	verde
...		
127	Neptun	albastru

**Rezervare (R)**

<u>rid:int</u>	<u>mid:int</u>	<u>bid:int</u>	<u>data:date</u>
22	31	103	02/11/2023
...	...		
231	58	127	09/02/2023

[1] Ramakrishnan, Gehrke, "Database Management Systems".

# Exemple interogări

---

- Problemă: aflați toți marinarii care au rezervat o barcă verde
- Soluții:

1.  $\pi_{\text{mid,nume}}(\sigma_{\text{culoare}='verde'}((M \circ_{M.\text{mid}=R.\text{mid}} R) \circ_{R.\text{bid}=B.\text{bid}} B))$

2.  $\pi_{\text{mid,nume}}((S \circ_{M.\text{mid}=R.\text{mid}} R) \circ_{R.\text{bid}=B.\text{bid}}(\sigma_{\text{culoare}='verde'}(B)))$

- Două planuri de execuție diferite
- Care credeți că este cel mai eficient?



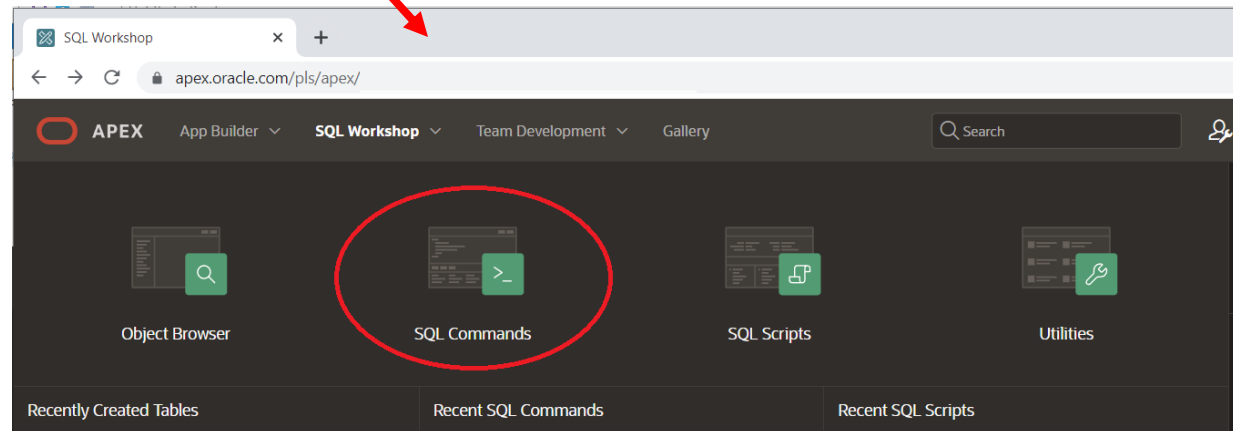
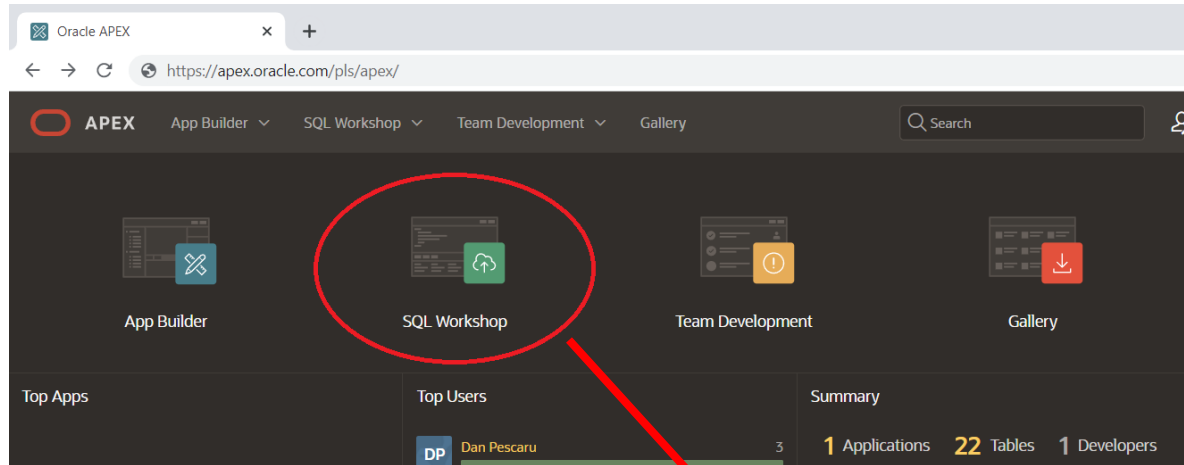
# Limbaje de interogare relaționale

---

1. Limbaje de interogare relaționale: proiectate pentru a interoga, modifica și gestiona date
2. Modelul relațional permite limbaje de interogare simple și puternice (ex. SQL,QBE):
  - Fundație puternică bazată pe algebra relațională
  - Permite optimizarea automată a interogărilor
3. Limbajele de interogare nu sunt limbaje de programare:
  1. Nu sunt Turing-complete (SQL92), dar extensii ale lor pot fi (ex. Oracle PL/SQL)
  2. Ex. Inchiderea recursivă nu poate fi exprimată
4. Dar ele asigură accesul la date complexe!

# Rulare example curs in Oracle APEX

1. Exemplele pot fi rulate folosind APEX SQL Workshop / SQL Commands



# Rulare cod SQL

1. Codul se introduce in fereastra de editare (1) si apoi se apasă butonul [Run] (2).  
Obs: tabelele trebuie create in prealabil

The screenshot shows the APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The user is logged in as 'Dan Pescaru'. The 'SQL Commands' section is active, showing a schema of 'WKSP\_DPESCARU'. The 'Language' is set to 'SQL' and 'Rows' to '10'. The 'Run' button is circled in red and labeled with a red '2'. The SQL command 'SELECT \* FROM Student;' is entered in the editor, with the first line circled in red and labeled with a red '1'. Below the editor, the 'Results' tab is selected, displaying a table with 4 rows and 4 columns: MARCA, NUME, AN, and MEDIA.

MARCA	NUME	AN	MEDIA
2	Ana Olteanu	3	9.25
4	Nicoleta Coman	1	7.75
3	Eduard Stoica	2	9.5
1	Alexandru Popescu	1	8.45

4 rows returned in 0.01 seconds [Download](#)