LABORATOR 1

BAZE DE DATE

CHESTIUNI ORGANIZATORICE

- Nota finala: (Nota design + Nota finala SQL)/2
- Nota finala SQL: 2/3*SQL laborator + 1/3*SQL examen
  - Nota de la laborator va avea o valoare de la 1p la 10p.
  - Nota de la SQL examen va avea o valoare de la 1p la 10p.
  - Nu este permisa o diferenta >=5p intre nota de la examenul scris si cea de la laborator (cu bonus inclus), daca una dintre cele doua note este <5 – in acest caz intreaga parte de SQL se considera nepromovata, altfel ramane nepromovata doar proba cu nota <5.
  - In urma rezultatelor obtinute la examen, studentilor li se va comunica proba care trebuie refacuta la restanta(test laborator sau SQL scris)
- Prezenta: 14 laboratoare (din care ultimul este Testul) **(maxim 5 absente** pentru a participa la testul de la laborator)
- Nota SQL laborator: Test final: 10p
  - Bonus:
    - Prezenta laborator + activitate minima: 2p
    - Activitate laborator constanta: 4p
    - Maxim 2 p obtinute din bonusuri se vor adauga notei de la laborator, daca aceasta este >= 5.
    - (Bonus nefolosit: 50% se adauga la SQL examen in limita a 2p, daca nota la SQL examen este >= 5)
- Contact: natalia.moanga@drd.unibuc.ro

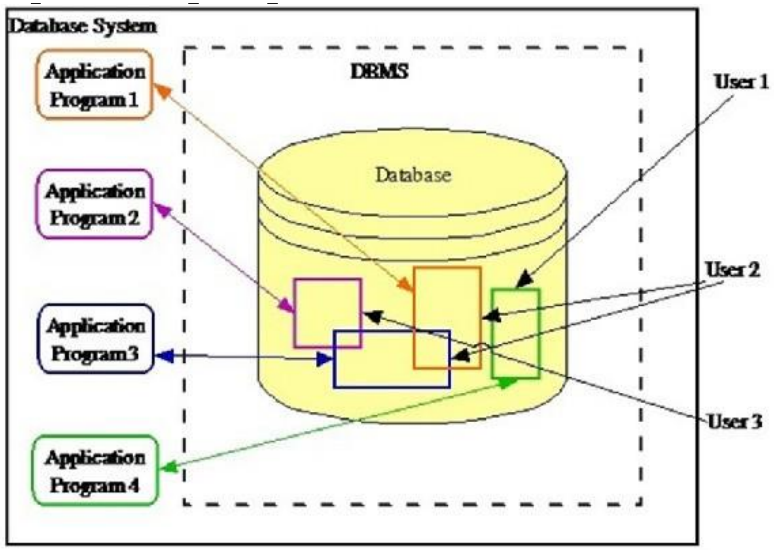
OBIECTIVE

- Introducere
- Ce este SQL
- Analiza comenzii SELECT
- Interogari simple mono-relatie

INTRODUCERE

Baza de date este un ansamblu structurat de date coerente, fără redondanţă inutilă, astfel încât acestea pot fi prelucrate eficient de mai mulţi utilizatori într-un mod concurent. Baza de date este o colectie de date persistente, care sunt folosite de catre sistemele de aplicatii ale unei anumite "intreprinderi".

Un **sistem de gestiune a bazei de date** (SGBD – *Data Base Management System*) este un produs *software* care asigură interacţiunea cu o bază de date, permiţând definirea, consultarea şi actualizarea datelor din



Retinem: un user/ o aplicatie nu poate accesa direct datele din BD,ci doar via SGBD

INTRODUCERE

In lucrul cu Baze de Date , se remarca activitatile urmatoare, pe care le veti studia in cadrul cursului de Baze de Date:

- Proiectarea Bazei de Date

  - numai la curs !!!

- Crearea Bazei de Date - Limbajul de definire a datelor

  -la curs si in partea a III-a a semestrului la lab.

- Prelucrarea Datelor din Baza de Date

  - la curs si tot semestrul la lab.

- Administrarea Bazei de Date

  - la curs si in partea a III-a a semestrului la lab.
  - aprofundare la cursul de SGBD (an III) si
    la Master Baze de Date si Tehnologii
    Web

INTRODUCERE

Concepte si termeni:

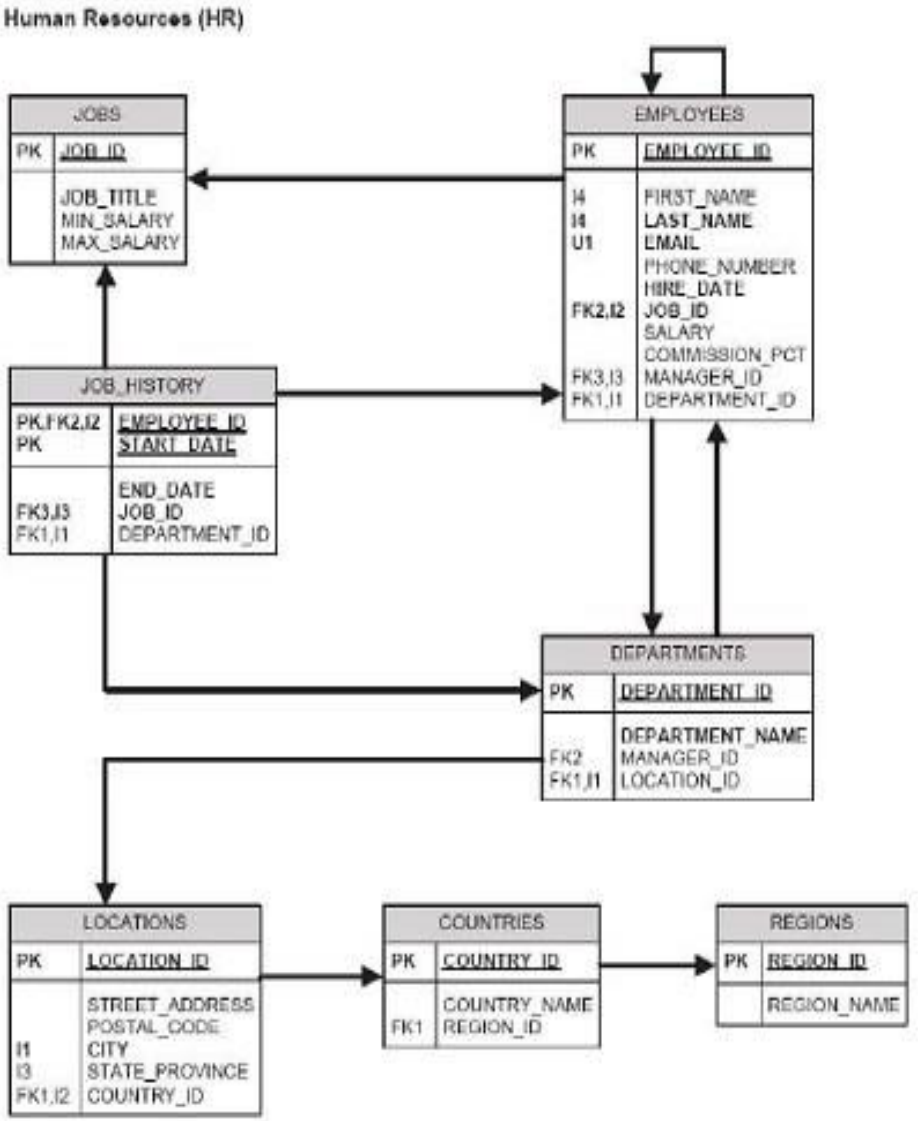
* **Dicţionarul datelor**, structurat şi administrat ca o bază de date (metabază de date), contine „date despre date“, furnizează descrierea tuturor obiectelor unei baze de date, starea acestor obiecte etc.
* Obiectele dintr-o baza de date pot fi: tabele, indecsi, subprograme stocate(proceduri,functii), pachete, triggeri, sinonime, view-uri, etc.
* La laboratorul de BD vom lucra cel mai mult cu tabele. Tabelele tin de modelul relational. Conceptele utilizate pentru a descrie formal, uzual sau fizic elementele de bază ale organizării datelor sunt date în următorul tabel:

Formal	Uzual	Fizic
relaţie	tablou / tabela	fişier
tuplu	linie	înregistrare
atribut	coloană	câmp
domeniu	tip de dată	tip de dată

INTRODUCERE

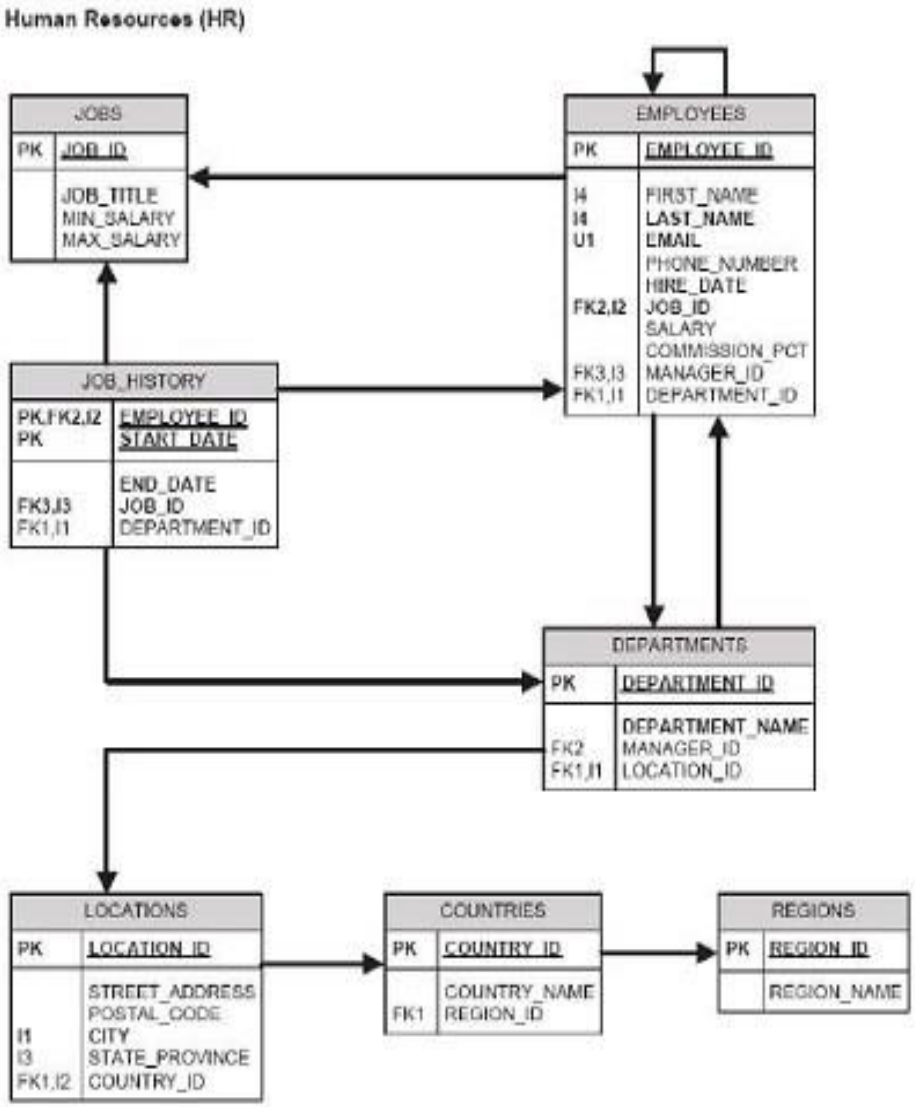
- Exercitiile din cadrul laboratorului vor folosi urmatoarele tabele:

Consideram modelarea datelor din domeniul gestiunii resurselor umane intr-o firma.



INTRODUCERE

- Exercitiile din cadrul laboratorului vor folosi urmatoarele tabele:



Explicatie:

JOBS este o **tabela**.

Job_id,job_title,min_salary, max_salary sunt **coloane** in tabela JOBS.

O **linie** din tabela JOBS

JOB_ID	JOB_TITLE	MIN_SALARY	MAX_SALARY
ST_MAN	Stock Manager	5500	8500

INTRODUCERE

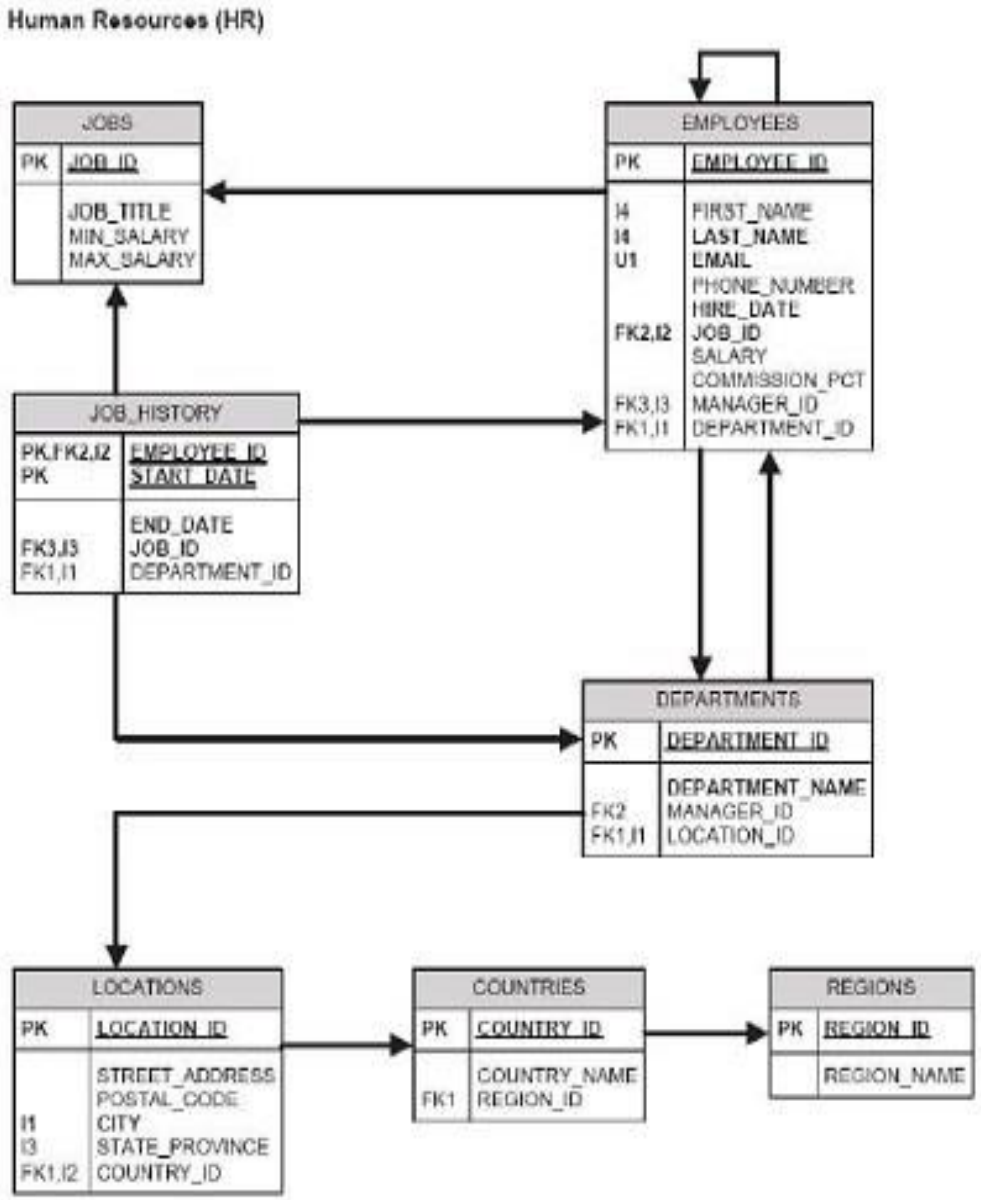
Concepte si termeni (cont):

Valoarea null intr-o coloana a unei linii din tabela semnifica faptul ca valoarea nu este cunoscuta sau atributul nu este aplicabil pentru linia respectiva.

Ex: in tabela DEPARTMENTS, in linia pentru departamentul

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
120	Treasury	NULL	1700

codul managerului (coloana Manager_id)



INTRODUCERE

Concepte si termeni (continuare):

Cheia primara este o mulţime minimală de atribute ale căror valori identifică unic un tuplu într-o relaţie.

Legatura intre 2 tabele este realizata prin intermediul **cheii externe**. Cheia externa dintr-o tabela refera cheia primara din cealalta tabela.

Modelul relaţional respectă **3 reguli de integritate structurală**. Regula 1 – unicitatea cheii:

Cheia primară trebuie să fie **unic**ă **şi minimală.**

Regula 2 – integritatea entităţii:

Atributele cheii primare trebuie să fie **diferite de valoarea null**.

Regula 3 – integritatea referirii.

O **cheie externă** trebuie **să fie ori null în întregime**, **ori să corespundă unei valori a cheii primare asociate**.

INTRODUCERE

Aplicatie:

Cheia primara din tabela

EMPLOYEES:

EMPLOYEE_ID

Cheia primara din tabela

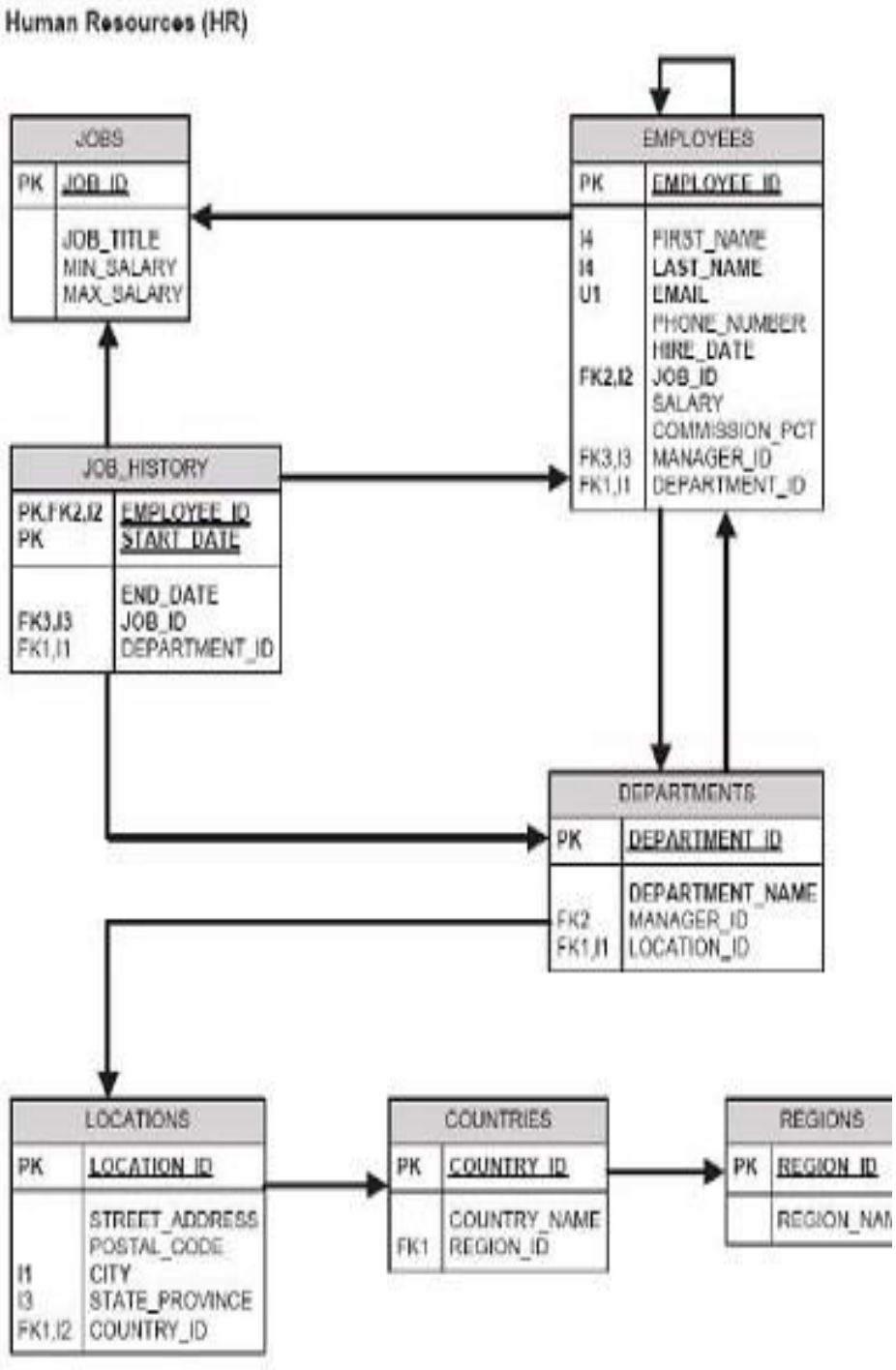
JOB_HISTORY: este compusa din

2 coloane (EMPLOYEE_ID,

START_DATE)

In tabela EMPLOYEES, job_id este cheie externa, face legatura cu tabela JOBS

=> coloana JOB_ID din tabela EMPLOYEES refera cheia primara din tabela JOBS, care este ……



CE ESTE SQL?

SQL (Structured Query Language) este un limbaj neprocedural pentru interogarea şi prelucrarea informatiilor din baza de date.

Compilatorul limbajului SQL generează automat o procedură care accesează baza de date şi execută comanda dorită.

SQL permite atât definirea, prelucrarea şi interogarea datelor, cît şi controlul accesului la acestea. Comenzile SQL pot fi integrate în programe scrise în alte limbaje,

de exemplu Cobol, C, C++, Java etc.

Comenzile SQL se termina cu ;

CE ESTE SQL?

În functie de tipul actiunii pe care o realizează, instructiunile SQL se împart în mai multe categorii. Datorită importantei pe care o au comenzile componente, unele dintre aceste categorii sunt evidentiate ca limbaje în cadrul SQL, şi anume:

· limbajul de definire a datelor (LDD)

– comenzile CREATE, ALTER, DROP;

· limbajul de prelucrare a datelor (LMD)

– comenzile INSERT, UPDATE,DELETE, SELECT;

· limbajul de control al datelor (LCD)

– comenzile COMMIT, ROLLBACK.

Pe lângă comenzile care alcătuiesc aceste limbaje, SQL cuprinde:

· instructiuni pentru controlul sesiunii;

· instructiuni pentru controlul sistemului;

· instructiuni SQL încapsulate.

COMANDA
DESCRIBE

- Aceasta comanda poate fi folosita pentru a afla informatii despre structura unei tabele cu nume dat, tipul de date al fiecarei coloane din tabela, precum si a constrangerilor NOT NULL asupra coloanelor ce nu pot lua valoarea aceasta.

- Comanda DESCRIBE **nu** ne arata cheia primara si nici cheile externe(daca ar exista chei externe).

Exemplu:

DESCRIBE JOBS;

```
Name                 Null?    Type
-------------------- -------- -------------
JOB_ID               NOT NULL VARCHAR2(10)
JOB_TITLE            NOT NULL VARCHAR2(35)
MIN_SALARY                    NUMBER(6)
MAX_SALARY                    NUMBER(6)
```

ANALIZA COMENZII SELECT

```
SELECT { [ {DISTINCT | UNIQUE} | ALL] lista_campuri | *}
FROM [nume_schemă.]nume_obiect  ]
        [, [nume_schemă.]nume_obiect  …]
[WHERE condiție_clauza_where]
[START WITH condiție_clauza_start_with
 CONNECT BY condiție_clauza_connect_by]
[GROUP BY expresie [, expresie …]
  [HAVING condiție_clauza_having] ]
[ORDER BY {expresie | poziție} [, {expresie | poziție} …] ]
[FOR UPDATE
    [OF [ [nume_schemă.]nume_obiect.]nume_coloană
        [, [ [nume_schemă.]nume_obiect.]nume_coloană] …]
    [NOWAIT | WAIT număr_întreg] ];
```

Bibliografie: pag 170 – 173 , Popescu I & al (2004), BIBLIOTECA DE MATEMATICA : II 40038

ANALIZA COMENZII SELECT

***Minimal**, o cerere SELECT are **clauza SELECT si clauza FROM** .*

***In clauza FROM** specificam din ce tabela (tabele) sa se extraga datele.*

***In clauza SELECT** precizam lista coloanelor/expresii pe baza coloanelor ce se doresc regasite in BD*

- *Daca dorim informatii complete – selectam toate coloanele tabelei cu* *******

```
SELECT *
FROM  JOBS;
```

JOB_ID	JOB_TITLE	MIN_SALARY	MAX_SALARY
AD_PRES	President	20000	40000
AD_VP	Administration Vice President	15000	30000
AD_ASST	Administration Assistant	3000	6000
FI_MGR	Finance Manager	8200	16000
FI_ACCOUNT	Accountant	4200	9000
AC_MGR	Accounting Manager	8200	16000
AC_ACCOUNT	Public Accountant	4200	9000

- *Daca dorim numai anumite coloane (proiectie) , le enumeram separate de virgula*

```
SELECT JOB_ID, MIN_SALARY, MAX_SALARY
FROM  JOBS;
```

JOB_ID	MIN_SALARY	MAX_SALARY
AD_PRES	20000	40000
AD_VP	15000	30000
AD_ASST	3000	6000
FI_MGR	8200	16000
FI_ACCOUNT	4200	9000
AC_MGR	8200	16000
AC_ACCOUNT	4200	9000

- *Putem folosi alias-uri si expresii in campurile din select:*

```
SELECT JOB_ID, (MAX_SALARY - MIN_SALARY) AS “diferenta”
FROM  JOBS;
```

JOB_ID	DIFERENTA
AD_PRES	20000
AD_VP	15000
AD_ASST	3000
FI_MGR	7800
FI_ACCOUNT	4800
AC_MGR	7800
AC_ACCOUNT	4800

EXERCITII – SETUL 1

Ex3&4)

a)Afisati si comentati structura tabelei EMPLOYEES.

b)Care este aritatea tabelei? (Aritate=nr de coloane)

c)Afisati date complete despre angajatii firmei.

d)Care este cardinalitatea tabelei? (Cardinalitate=nr linii)

EXERCITII – SETUL 1: Rasp.

a) Afisati si comentati structura tabelei EMPLOYEES.

DESCRIBE EMPLOYEES;

b) Care este aritatea tabelei? (**Aritate=nr de coloane**) 11 coloane

c) Afisati date complete despre angajatii firmei.

```
SELECT *
  FROM EMPLOYEES;
```

d) Care este cardinalitatea tabelei?

(**Cardinalitate=nr linii**) 107 linii

EXERCITII –SETUL 2

EX 7 &10 & 11)

e) Să se afişeze codul angajatului, numele, codul job-ului, data angajarii pentru toti angajatii din firma

f) Să se afişeze pentru fiecare angajat numele concatenat cu job_id-ul, separate prin virgula si spatiu, si etichetati coloana “Angajat si titlu”

(**Concatenarea sirurilor este ||**ex: 'a'||'b'||first_name)

g) Creati o cerere prin care sa se afiseze toate datele din tabelul EMPLOYEES. Separati fiecare coloană printr-o virgulă. Etichetati coloana ”Informatii complete”

EXERCITII –SETUL 2 – Rasp.

e) Să se afişeze codul angajatului, numele, codul job-ului, data angajarii pentru toti angajatii din firma

```
SELECT EMPLOYEE_ID, LAST_NAME, JOB_ID, HIRE_DATE
FROM EMPLOYEES;
```

f) Să se afişeze pentru fiecare angajat numele concatenat cu job_id-ul, separate prin virgula si spatiu, si etichetati coloana “Angajat si titlu”

(**Concatenarea sirurilor este ||** ex: 'a'||'b'||first_name)

```
SELECT  LAST_NAME||', '||JOB_ID AS "Angajat si titlu"
FROM EMPLOYEES;
```

g) Creati o cerere prin care sa se afiseze toate datele din tabelul EMPLOYEES. Separati fiecare coloană printr-o virgulă. Etichetati coloana ”Informatii complete”

```
SELECT
    EMPLOYEE_ID||','||FIRST_NAME||','||LAST_NAME||','||EMAIL||','||PHON
    E_NUMBER||','||HIRE_DATE||','||JOB_ID||','||SALARY||','||COMMISSION
    _PCT||','||MANAGER_ID||','||DEPARTMENT_ID AS "Informatii complete"
FROM EMPLOYEES;
```

FORMATAREA DATELOR CALENDARISTICE

- Datele calendaristice pot fi formatate cu ajutorul funcţiei TO_CHAR(data, format), unde formatul poate fi alcătuit dintr-o combinaţie a următoarelor elemente:

Element	***Semnificaţie***
D	Numărul zilei din săptămâna (duminica=1; luni=2; …sâmbătă=6)
DD	Numărul zilei din lună.
DDD	Numărul zilei din an.
DY	Numele zilei din săptămână, printr-o abreviere de 3 litere (MON, THU etc.)
DAY	Numele zilei din săptămână, scris în întregime.
MM	Numărul lunii din an.
MON	Numele lunii din an, printr-o abreviere de 3 litere (JAN, FEB etc.)
MONTH	Numele lunii din an, scris în întregime.
Y	Ultima cifră din an
YY, YYY, YYYY	Ultimele 2, 3, respectiv 4 cifre din an.
YEAR	Anul, scris în litere (ex: *two thousand four*).
HH12, HH24	Orele din zi, între 0-12, respectiv 0-24.
MI	Minutele din oră.
SS	Secundele din minut.
SSSSS	Secundele trecute de la miezul nopţii.

In baza de date exista **o tabela speciala, numita DUAL**, cu o singura coloana numita DUMMY avand ca tip de date VARCHAR2(1), si o singura linie. Aceasta tabela speciala poate fi folosita pentru a interoga pseudo-coloane precum SYSDATE (data curenta), expresii aritmetice :

SELECT SYSDATE FROM DUAL;

```
SYSDATE
---------
11-FEB-13
```

SELECT TO_CHAR(SYSDATE,'MONTH') FROM DUAL;

```
TO_CHAR(S
---------
FEBRUARY
```

SELECT 2+3 AS "SUMA" FROM DUAL;

```
 SUMA
-----
    5
```

ORDONAREA REZULTATULUI UNEI CERERI

Pentru o ordona rezultatul unei interogari, la partea minimala formata din clauzele SELECT si FROM, adaugam clauza ORDER BY. Aceasta va fi mereu ultima clauza dintr-o cerere! Implicit, ordoneaza ascendent (ASC), mic -> mare.

- Ordonare dupa un camp din clauza SELECT, scrieri echivalente:

Specific prin nume	**Specific prin alias**	**Specific prin pozitie**
SELECT FIRST_NAME,SALARY as "s1" **FROM** EMPLOYEES **ORDER BY SALARY DESC;**	SELECT FIRST_NAME,SALARY as "s1" FROM EMPLOYEES **ORDER BY "s1" DESC**;	SELECT FIRST_NAME,SALARY FROM EMPLOYEES **ORDER BY 2 DESC**;

- Ordonare dupa un camp ce nu apare in SELECT:

SELECT FIRST_NAME, SALARY

FROM employees

ORDER BY COMMISSION_PCT;

!!! Implicit , valorile NULL sunt considerate cele mai mari

EXERCITII - SETUL 3

EX 18 & propuse)

h) Să se afişeze data şi ora curentă, pana la precizie de minut.

i) Ce efect are comanda:

```
SELECT SYSDATE
FROM
EMPLOYEES;
```

j) Pentru fiecare angajat sa se scrie codul, numele si ziua din saptamana in litere in care a fost angajat

k) Ordonati angajatii din firma crescator dupa anul angajarii in firma, iar la egalitate de an crescator dupa ziua din an in care au fost angati.

CLAUZA WHERE

Clauza *WHERE* poate fi folosită pentru a impune anumite condiţii liniilor din care se vor extrage coloanele specificate în clauza *SELECT*.

Clauza WHERE este pozitionata mereu dupa clauza FROM:

```
SELECT ….
 FROM ….
 WHERE ….
 ORDER BY  …  ;
```

Ordinea in care se vor executa clauzele din cererea mono-relatie:

1) From ∈ stabileste din ce tabela se extrag datele
2) Where ∈ filtreaza datele, numei unele linii sunt selectate
3) Select ∈ proiectie a.i. numai unele coloane sunt alese, iar alias-urile (daca exista) stabilesc etichetarea coloanelor in rezultatul final
4) Order By ∈ sorteaza rezultatul final.

Morala: in clauza WHERE NU putem folosi alias-urile campurilor din clauza SELECT

CLAUZA WHERE

Exemple de conditii posibile in clauza WHERE:

- De testarea egalitatii/inegalitatii cu o constanta
- De testarea egalitatii/inegalitatii cu o alta coloana din aceeasi linie
- De testarea valorii NULL cu IS NULL
- De testarea apartenentei la o lista de valori cu IN (10,20,30)
- De testarea apartenentei la un interval cu BETWEEN 10 AND 15
- La siruri, de testarea conformitatii cu un sablon cu LIKE '_A%'

In clauza WHERE putem avea conditii compuse prin AND si/sau OR Seva tine cont de prioritatea operatorilor: ORsparge conditia compusa in subconditii

WHERE a=1 AND b=2 OR c=3 AND d=4 #

WHERE (a=1 AND b=2) OR (c=3 AND d=24)

EXERCITII - SETUL 4

EX 12&15&16&25&20&26)

l) Sa se listeze numele si salariul angajatilor care câştigă mai mult de 2850 $.

m) Să se afişeze numele, job-ul şi data la care au început lucrul salariatii angajati între 20 Februarie 1987 şi 1 Mai 1989. Rezultatul va fi ordonat crescător după data de început.

n) Să se afişeze numele salariatilor şi codul departamentelor pentru toti angajatii din departamentele 10, 30 şi 50 în ordine alfabetică a numelor

o) Să se afiseze numele, job-ul si salariul pentru toti salariatii al caror job contine şirul "clerk" sau "rep" si salariul nu este egal cu 1000, 2000 sau 3000 $.

p) Să se afişeze numele şi job-ul pentru toti angajatii care nu au manager

q) Afisati numele, salariul si comisionul pentru toti angajatii al caror salariu este mai mare decat comisionul (salary*commission_pct) marit de 5 ori.