---ROLLUP

select department\_id, job\_id, sum(salary)

from employees

grup by department\_id, job\_id

select department\_id, null, sum(salary)

from employees

group by department\_id

order by 1, 2 nulls last

select null, null, sum(salary)

from employees

order by 1, 2, nulls last;

---echivalent cu

select department\_id, job\_id, sum(salary)

from employees

grup by rollup (department\_id, job\_id)

order by 1, 2 nuls last; -> dupa dep si job si dupa dep

--- CUBE

select department\_id, job\_id, sum(salary)

from employees

grup by department\_id, job\_id

union all

select department\_id, null, sum(salary)

from employees

group by department\_id

union all

select null, null, sum(salary)

from employees

group by job\_id

union all

select null null sum(salary)

from employees

order by 1, 2, nulls last

---echivalent cu

select department\_id, job\_id, sum(salary)

from employees

group by cube (department\_id, job\_id)

order by 1, 2 nulls last; --->mai multe moduri de a scoate rez

--- exemplu;

Pentru determinarea modului ìn care a fost obţinută o valoare totalizatoare cu ROLLUP sau

CUBE, se utilizează funcţia:

• GROUPING(expresie)

Aceasta întoarce:

- valoarea 0, dacă expresia a fost utilizată pentru calculul valorii agregat

- valoarea 1, dacă expresia nu a fost utilizată.

select department\_id, job\_id, sum(salary), grouping(department\_id), grouping(job\_id)

from employees

group by cube (department\_id, job\_id)

order by 1, 2, nulls last;

-- ex 1 Să se afişeze numele departamentelor, titlurile job-urilor şi valoarea medie a salariilor,

pentru:

- fiecare departament şi, în cadrul său pentru fiecare job;

- fiecare departament (indiferent de job);

- întreg tabelul.

ROLLUP = Select + group by pt mai multe cazuri conectate prin UNION

->n +1 rezultate ( ( ), ( ) ) ->3

CUBE = 2^n rez -> toate combinatiilede afisari

a)

\* paramntezele grupeaza in 3 cazuri rezultatele …

Select e.department\_id, e.job\_id, d.department\_name, j.job\_title, avg(salary)

From employees e join departmens d on (e.department\_id = d.department\_id)

Join jobs j on (e.job\_id = j.job\_id)

Group by rollup ((e.department\_id, d.department\_name), (e.job, j.job\_title))

Order by 1, 2;

b) Analog cu a), afişând şi o coloană care arată intervenţia coloanelor department\_name,

job\_title, în obţinerea rezultatului.

Select e.department\_id, e.job\_id, d.department\_name, j.job\_title, avg(salary),

Decode(grouping (e.department\_id), 0, Decode (grouping (e.job\_id), 0, 'medie dep si job',

1, 'medie dep'),

1, Decode (grouping (e.job\_id), 0, 'medie job',

1, 'medie employees')

)

if e.dep\_id = 0 -> if (e.job\_id = 0 ->medie dep si job else medie dep)

From employees e join departments d on (e.department\_id = d.department\_id)

Join jobs j on (e.job\_id = j.job\_id)

Group by rollup ((e.department\_id, d.department\_name), (e.job\_id, j.job\_title))

Order by 1, 2;

Dacă se doreşte obţinerea numai a anumitor grupări superagregat, acestea pot fi precizate

prin intermediul clauzei :

• GROUPING SETS ((expr\_11, expr\_12, …, expr\_1n), (expr\_21, expr\_22, …expr\_2m), …)

--ex3 Să se afişeze numele departamentelor, numele job-urilor, codurile managerilor, maximul şi suma salariilor pentru:

- fiecare departament şi, în cadrul său, fiecare job;

- fiecare job şi, în cadrul său, pentru fiecare manager;

- întreg tabelul.

Select d.department\_name, j.job\_title, m.last\_name, sum(e.salary)

From employees e join departments d on (e.department\_id = d.department\_id)

Join jobs j on (e.job\_id = j.job\_id)

Join employees m on (e.manager\_id = m. employee\_id)

Group by grouping sets ( (d.department\_name, j.job\_title), (j.bon\_title, m.last\_name), () )

--ex5 Să se afişeze informaţii despre angajaţii al căror salariu depăşeşte valoarea medie a

salariilor colegilor săi de departament.

select e.first\_name, e.salary, e.department\_id

from employees e join employees e2 on (e.department\_id = e2.department\_id)

//where e.employee\_id <> e2.employee\_id

group by e.department\_id, e.first\_name, e.salary

having e.salary > (select avg(salary) from employees where department\_id = e.department\_id);

--7. Sa se afiseze numele si salariul celor mai prost platiti angajati din fiecare departament

--sincronizata

select e.last\_name, e.salary

from employees e

where e.salary = (select (min) salary

from employees

where department\_id = e.department\_id);

--nesincronizata

select e.last\_name, e.salary

from employees e

where (e.department\_id, e.salary) in (select department\_id, (min) salary

from employees

group by department\_id);

--

select e.last\_name, e.salary

from employees e , (select department\_id, (min) minsal

from employees

group by department\_id) dep ;

where e.department\_id = dep.department\_Id and salary = dep.minsal;

-- ex9. Sa se obtina numele salariatilor care lucreaza intr-un departament in care exista cel putin un angajat cu salariul egal cu salariul maxim din departamentul 30 (operatorul exists).

----ex Să se afişeze codul, numele şi prenumele angajaţilor care au subalterni.

Distinct – pt ca un ang poate avea mai multi subalterni

Select distinct e.employee\_id, e.first\_name, e.last\_name

From employees e join employees sub on ( e.employees\_id = sub.manager\_id)

Select distinct e.employee\_id, e.first\_name, e.last\_name

From employees e where e.employee\_id in select (mananger\_id from employees)

Select distinct e.employee\_id, e.first\_name, e.last\_name

From employees e where exists ( select ‚x’ from employees wheere mananger\_id = e.employee\_id) 'x' pt ca trb sa selectez ceva dar nu mi pasa ce anume, mai multe de formalitate

--de preferat sa fol exists in loc de where in

-- au subalterni cu salariu mai mare

Select distinct e.employee\_id, e.first\_name, e.last\_name

From employees e where e.employee\_id in (select mananger\_id from employees

Where salary>=e.salary)

--ex 11 Să se afişeze codul, numele şi prenumele angajaţilor care au cel puţin doi subalterni.

Select e.employee\_id, e.first\_name, e.last\_name

From employees e join on employees sub on ( e.employees\_id = sub.manager\_id)

Group by e.employee\_id, e.last\_name, e.first\_name

Having count(\*) >=2

--12. Să se determine locaţiile în care se află cel puţin un departament.

select location\_id

from locations l

where exists (select'x' from departments where location\_id = l.location\_id);

--21. Să se detemine primii 10 cei mai bine plătiţi angajaţi.

select rownum, last\_name, first\_name, salary

from employees

where rownum <=10

order by salary; // nu e bn , order by TREBUIE inainte de where !!! ---> DAR OBLIGATORIU IN SUBCERERE, ALTFEL -> PB DE SINTAXA

In cazul asta voi ordona primele 10 rez ...aleatoare

--- asa e bn

select last\_name, first\_name, salary

from ( practic fol o subcerere ca sa iau in ordine desc

select last\_name, first\_name, salary

from employees

order by salary desc

)

where rownum <=10; o variabila aleasa pt a calcula nr de rez returnate

--- sau asa

select last\_name, first\_name, salary

from employees e

where //count nr de ang care castiga mai mult decat e,

// sa nu existe mai multi ang

// e.last name e in top 10 daca nu exista mai mult de 9 ang cu salariul mai mare decat e.salary

(selec count (\*) from employees where salary > e.salary) <=9

--22 Să se determine cele mai prost plătite 3 job-uri, din punct de vedere al mediei salariilor.

select job\_title, medie\_job

from (select job\_id, avg(salary) medie\_job

from employees e join jobs j on (e.job\_id = j.job\_id)

group by e.job\_id, job\_title;

order by avg(salary) desc;

)

where rownum <=3;

--23. Să se afişeze informaţii despre departamente, în formatul următor: „Departamentul

<department\_name> este condus de {<manager\_id> | nimeni} şi {are numărul de salariaţi

<n> | nu are salariati}“.

select 'Dep' || d.department\_name ||'este condus' || nvl(e.last\_name, 'nimeni') || case count( e.employee\_id) when 0 then 'nu are salariati'

else 'si are' || count( e.employee\_id) end "info"

from departments d left join employees e on (d.manager\_id = e.employee\_id)

left join employees a on (d.department\_id = a.department\_id)

group by d.department\_id, d.department\_name;

-- 26

select

case when job\_title like 'S%' then sum(job\_title)

when (max\_salary = max(max\_salary) )then avg(min\_salary + max\_salary)

from jobs;

DECODE (expresie, val\_1, val\_2, val\_3, val\_4, …., val\_2n-1, val\_2n, default) – dacă expresie

= val\_1, întoarce val\_2; dacă expresie = val\_3, întoarce val\_4; …; altfel întoarce default.

• DECODE este echivalent cu CASE, a cărui structură este:

CASE expresie

WHEN val\_1 THEN val\_2

WHEN val\_3 THEN val\_4

…

ELSE default

END

CASE poate avea si forma: