TSIDIANY Raharison Muriel L1 G2

-------------------------------------------------------------------------------------------------------

EXERCICE sur commandes SQL (Exercice page 9-10-11-12-13-14-15)

-------------------------------------------------------------------------------------------------------

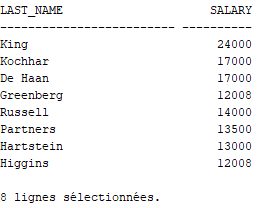
## RESTREINDRE ET TRIER DES DONNEES

# Affichons le nom et le salaire des employés qui gagnent plus de 12000 :

SELECT last\_name, salary

FROM employee

WHERE salary > 12000 ;

+

# Créons un état affichant le nom et le numéro de département correspondant à l’ID d’employé 176 :

SELECT department\_name, department\_id

FROM departments

WHERE department\_id =176;

NB : Cette numéro de département n’existe, donc aucune ligne sélectionnée !

# Affichons les noms, salaires des employés qui gagnent entre 6000 et 10000 :

SELECT last\_name, salary

FROM employees

WHERE salary BETWEEN 6000 AND 10000;

NB: On a 42 lignes sélectionnées !

# Affichons le nom, l’ID de post et la date d’embauche de Matos et Taylor tel que la date d’embauche est en ordre croissant :

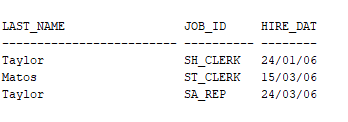
SELECT last\_name, job\_id, hire\_date

FROM employees

WHERE last\_name = 'Matos'

OR last\_name = 'Taylor'

ORDER BY hire\_date ASC;



# Affichons le nom et le numéro de département de tous les employés du département 20 ou 50 par ordre alphabétique croissant:

SELECT last\_name, department\_id

FROM employees

WHERE (department\_id = 20

OR department\_id = 50)

ORDER BY last\_name ASC;

NB: 47 lignes sélectionnées !

# Affichons le nom et salaire des employés qui gagnent entre 5000 et 12000 dans le département 20 ou 50 avec colonnes Employee et Monthly Salary :

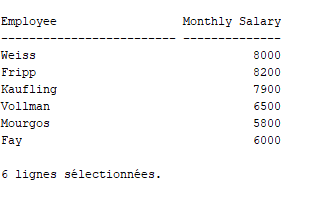
SELECT last\_name AS "Employee", salary "Monthly Salary"

FROM employees

WHERE (salary BETWEEN 5000 AND 12000)

AND (department\_id = 20

OR department\_id = 50);



# Affichons le nom et la date d’embauche de tous les employés embauchés en 1994 :

SELECT last\_name, hire\_date

FROM employees

WHERE hire\_date LIKE 'DD/Mon/94';

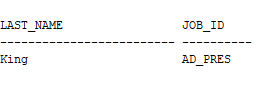
NB : Aucune ligne sélectionnée !

# Créons un état affichant le nom et l’intitulé de poste de tous les employés qui n’ont pas de manager.

SELECT last\_name, job\_id

FROM employees

WHERE manager\_id IS NULL;



# Affichons le nom et salaire des employés qui gagnent plus qu’un montant saisi par l’utilisateur en réponse à une invité :

SELECT last\_name, salary

FROM employees

WHERE salary > &MONTANT;

# Affichons le nom de tous les employés dont le nom comporte un « a » et un « e » :

SELECT last\_name

FROM employees

WHERE last\_name LIKE '%a%' OR last\_name LIKE '%e%';

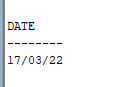
NB : 82 lignes sélectionnées !

## FONCTIONS MONOLIGNES

# Ecriture d’une interrogation pour afficher la date du système avec son intitulation :

SELECT sysdate AS "DATE"

FROM dual;



# Affichons le numéro d’employé, nom, salaire et salaire augmenté de 15.5% pour chaque employé avec nom New Salary :

SELECT employee\_id, last\_name, salary, (salary + salary\*(155/1000)) AS "New Salary"

FROM employees;

NB : 107 lignes sélectionnées !

# Ecrivons un interrogation pour afficher le nom (Premier lettre en Majuscule et les en minuscule) et la longueur de nom de tous les employés dont le nom commence par les lettres « J », « A», « M » avec chaque colonne avec des libellé. Avec triage d e nom des employés :

SELECT INITCAP (last\_name) AS "NAME",

LENGTH (UPPER(INSTR(last\_name, 'A%')))

FROM employees

ORDER BY NAME ASC;

# Une interrogation permettant d’afficher le nom et salaire de tous les employés avec format de 15 caractères :

SELECT last\_name, LPAD(salary,15,'$') AS "SALARY"

FROM employees;

## FONCTIONS DE CONVERSION ET DES EXPRESSIONS CONDITIONNELLES

# Affichons le nom, la date d’embauche et le jours de la semaine où l’employé a commencé avec l’intituler colonne DAY et avec triage en fonction de la date :

SELECT last\_name, hire\_date "Date",

TO\_CHAR(hire\_date, 'DD')

AS day

FROM employees

ORDER BY hire\_date ASC;

NB: 107 lignes sélectionnées !

# Une interrogation qui affiche le nom et montant de la commission de chaque employé, si un employé ne perçoit pas de commission, on indique « No Commission » et la colonne est COMM :

SELECT last\_name,

NVL(TO\_CHAR(commission\_pct),'No Commission') AS COMM

FROM employees;

NB: 107 lignes sélectionnées !

# Avec la fonction de DECODE, écrivons une interrogation qui affiche le niveau de tous employés sur la base de valeur de la colonne JOB\_ID a l’aide de données donné dans le tableau :

SELECT last\_name,

DECODE (job\_id, 'AD\_PRES','A',

'ST\_MAN','B',

'IT\_PROG','C',

'SA\_REP','D',

'ST\_CLERK','E',

0)

AS GRADE\_EMPLOYER

FROM employees;

NB: 107 lignes sélectionnées !

# Réécrivons l’instruction avec l’instruction CASE :

SELECT last\_name,

CASE job\_id WHEN 'AD\_PRES' THEN 'A'

WHEN 'ST\_MAN' THEN 'B'

WHEN 'IT\_PROG' THEN 'C'

WHEN 'SA\_REP' THEN 'D'

WHEN 'ST\_CLERK' THEN 'E'

ELSE TO\_CHAR (0) END AS GRADE

FROM employees;

## FONCTIONS DE GROUPE

# C’est vrai, les fonctions de groupe opèrent sur plusieurs lignes et produisent un résultat par groupe

# C’est faux, les fonctions de groupe ne prennent pas en compte les valeurs NULL dans un Calculs.

# C’est vrai, le clause WHERE limite les lignes avant inclusion dans un calculs.

# Déterminons le salaire le plus élevé, le salaire le plus bas, le salaire cumulé et le salaire moyen pour tous les employés :

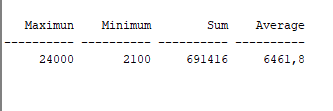
SELECT MAX(salary) "Maximun",

MIN(salary) "Minimum",

SUM(salary) "Sum",

ROUND(AVG(salary),1)"Average"

FROM employees;



# Ecrivons une interrogation permettant d’afficher le nombre de personnes occupant le même poste :

SELECT job\_id, COUNT (DISTINCT employee\_id ) AS "NB EMPLOYE"

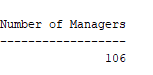
FROM employees

GROUP BY job\_id;

# Le nombre de manager :

SELECT COUNT (manager\_id) "Number of Managers"

FROM employees;



# Trouvons la différence entre le salaire le plus élevé et le salaire le plus le plus bas :

SELECT MAX (salary) "MAX",

MIN (salary) "MIN",

(MAX (salary) – MIN (salary)) "DIFFERENCE"

FROM employees;

## DONNEES PROVENANT DE PLUSIEUR TABLE

# Ecrivons une interrogation produisant l'adresse de tous les départements en utilisez les tables LOCATIONS et COUNTRIES tout en affichant dans les résultats l'ID de lieu, la rue, la ville, le département et le pays :

SELECT l.country\_id COUNTRIE, l.street\_address RUE, l.city CITY, department\_name DEP, c.country\_name COUNTRY

FROM countries c

JOIN locations l

ON c.country\_id = l.country\_id

JOIN departments d

ON d.location\_id = l.location\_id;

# Une interrogation permettant d’afficher le nom, ainsi que le numéro et le nom de départements, pour tous les employés :

SELECT last\_name NAME, employee\_id NUMERO , department\_name DEPARTEMENT

FROM employees NATURAL JOIN departments ;

# Affichons le nom, le poste, ainsi que le numéro et le nom de département, pour tous les employés qui travaillent à Toronto :

SELECT e.last\_name NAME, e.job\_id POSTE, d.location\_id "NUM DEP", d.department\_name DEP

FROM employees e

JOIN departments d

ON d.department\_id = e.department\_id

JOIN locations l

ON l.city ='Toronto';

# Créons un état permettant d'afficher le nom et l'ID des employés, ainsi que le nom et l'ID de leur manager. Intitulez respectivement les colonnes Employee, Emp#, Manager et Mgr#.

SELECT e.last\_name "EMPPLOYEE", e.employee\_id "EMP#", e.manager\_id "MGR#"

FROM employees e;

## SOUS-INTERROGATION

# Ecrivons une interrogation pour afficher le nom et la date d’embauche de tous les employés travaillant dans le même département que l’employé indiqué :

SELECT last\_name AS "NOM",hire\_date "DATE", department\_id

FROM employees

WHERE department\_id =

(SELECT department\_id

FROM employees

WHERE last\_name = '&&NON')

AND last\_name <> (SELECT last\_name

FROM employees

WHERE last\_name='&NON');

# Créons un état qui affiche le numéro d’employé, le nom et le salaire de tous les employés qui gagnent plus que le salaire moyen et trions les résultats par ordre croissant sur la base du salaire :

SELECT employee\_id "NUM", last\_name "NAME", salary

FROM employees

WHERE salary >

(SELECT AVG(salary)

FROM employees);

NB : 51 lignes sélectionnées !

# Une interrogation qui affiche le numéro d’employé et le nom de tous les employés qui travaillent dans un département comprenant un employé dont le nom contient la lettre « u » :

# Créons un état affichant la liste de tous les employés dont le salaire est plus que le salaire de tout employé du département 60 :

SELECT last\_name

FROM employees

WHERE salary >

(SELECT SUM(salary)

FROM employees

WHERE department\_id=60);

## OPERATEURS ENSEMBLISTES

# Affichons les listes des ID des départements qui ne contiennent pas l’ID de poste ST\_CLERK en utilisant les opérateurs ensemblistes pour créer cet état

SELECT department\_id AS "ID"

FROM departments

UNION

SELECT department\_id

FROM employees

WHERE job\_id <> 'ST\_CLERK';

NB : 28 lignes sélectionnées !

# Production de la liste des postes des départements 10, 50 et 20 dans l’ordre donné au tableau :

# Affichons l’ID et le nom des pays sans départements avec ensemblistes :

SELECT country\_id, country\_name

FROM locations

MINUS

SELECT country\_id, city

FROM