

Employés

EE(EID, ENAME, JOB, HIRED, SAL, COMM, #UID, #EMGR)
 BU(UID, UNAME, CITY, #UMGR)
 PRJ(PID, PNAME, STARTD, ENDD, #PMGR)
 IN(#EID, #PID, PCT)

Potentiellement null; EMGR reference EE(EID)
 UMGR " "
 PMGR " "

Exercice 1

→ lister tous les Business Unit

select * from BU;

→ Nom et salaire : Attention à l'ordre demandé !

select ENAME, SAL from EE;

→ Nom et salaire commissioné :

select ENAME, SAL + COMM as Total Sal from EE;

Δ Comm est potentiellement null → $2000 + null = null$

→ select ENAME, SAL + case when Comm is null then 0 else Comm end as Total Sal from EE;

→ Nom et salaire en € (à la place de \$)

```
select ENAME, SAL * 1,2 as Sal_In_Euro from EE;
```

→ Nom des employés contenant deux "M"

pas à la
'%1%1%'

```
select distinct ENAME from EE where ENAME like '%mm%';
```

distinct

```
upper(ENAME) like '%111%';
```

↳ upper car tout en majuscules pour prendre en compte tous les cas: mM, Mm...

→ Manager qui gagnent + de 2000.

```
select ENAME from EE
```

```
where JOB = 'Manager' and SAL > 2000;
```

→ Non employés embauchés après 2010 et le salaire inclus entre 2000 et 5000.

```
select ENAME from EE
```

```
where HIRED > DATE '2010-12-31'
```

```
and SAL >= 2000 and SAL <= 5000;
```

SQL ↳ SAL between 2000 and 5000;

Heure 3.

→ Employé avec commission.

select * from EE where Comm $\neq$ 0 [?] or Comm is not null;

→ Trouver le nom des BU où l'UID n'est pas spécifié.

select UNAME from BU where UID is null;

↳ comede, mais on est sûr qu'elle produit $\emptyset$.

→ Trouver les top managers

select ENAME from EE where EMGR is null;

→ Trouver les villes dans laquelle il y a une BU.

select CITY from BU;
distinct

→ Employés par ordre alphabétique de nom

select * from EE order by ENAME;

→ Job et salaire des employés triés par job et par salaire décroissant.

select ENAME, JOB, SAL from EE order by JOB, SAL desc;

Exercice 3.

→ Produit cartésien EE et BU

select * from EE cross join BU; (select * from EE, BU;)
possible, mais "obsoleter"

→ Produit et égalité d'un attribut.

① select ENAME, UNAME from EE cross join BU
where EE.UID = BU.UID;

② select ENAME, UNAME from EE join BU on EE.UID = BU.UID;

③ select ENAME, UNAME from EE natural join BU;

→ Employés qui travaillent à PARIS.

select ENAME from EE natural join BU where CITY = 'Paris';

→ Employés avec le nom du manager.

select e1.ENAME, e2.ENAME from EE e1, EE e2
where e1.MGR = e2.EID;

↳ select e1.ENAME, e2.ENAME from EE e1 join EE e2 on e1.MGR = e2.EID;

→ Trouver les ^{couple} employés avec le père manager.

select e1.ENAME, e2.ENAME, e1.MGR from EE e1 cross join EE e2
where e1.MGR = e2.EID and
e1.ENAME < e2.ENAME;

→ Employés qui ont le même job que John.

```
select e1.ENAME from EE e1 inner join EE e2 where e1.job = e2.job  
and e2.ENAME = 'John';  
and e1.EID <> e2.EID.
```

→ Employés dans une BU...

```
select e1.ENAME  
from EE e1, EE e2  
where e1.UID = e2.UID and  
e2.job = 'Analyst';
```

→ Salaire...

```
select distinct e1.ENAME  
from EE e1, EE e2  
where e1.SAL > e2.SAL and  
e2.job = 'Analyst';
```

→ BU qui n'ont pas d'employés

```
select UID, UNAME, CITY, UINGR  
from BU natural left outer join EE  
where EID is null,  
UID, UNAME, CITY...  
250, 'New Develop', 'Paris', ...
```

→ Employés employés avant tous ceux de la BU n° 10.

select e1.ENAME

from EE e1 left outer join (EE e2 join EE e3 on e2.HIRED >= e3.HIRED and e3.UD = 10)

employés embauchés après
un employé de BU 10

on e1.UD = e2.UD
where e2.UD is null;

doublage de e1, e2, e3

Exercice 4.

- 1) select count (*) from EE;
- 2) select count (COMM) from EE;
↳ bis) select count (*)
from EE
where COMM is not null;
- 3) select min (SAL), avg (SAL), max (SAL) from EE
where JOB = 'Clerk';
- 4) select min (HIRED), max (HIRED) from EE
where JOB = 'Salesman';
- 5) select count (*) from EE where JOB = 'Manager';
↳ bis) select count (distinct MGR) from EE;
- 6) select count (distinct UD) from EE;
- 7) select count (*) from EE where COMM is null;
select count (*) - count (COMM) from EE;
- 8) select min (COMM), max (COMM) from EE where COMM <> 0;

Exercice 5.

4) Trouver les employés avec les plus haut salaire.

```
select * from EE
where (select max(SAL) from EE) = SAL;
on SAL = (select ...);
```

3) select * from EE
where SAL < (select avg(SAL) from EE);
~~SAL < avg(SAL);~~

3) select * from EE where SAL < (select min(SAL) from EE
where JOB = 'Manager');
↳ bis) select * from EE where SAL < all(select SAL from EE
where JOB = 'Manager');

4) select * from EE where SAL > (select min(SAL) from EE
where JOB = 'Analyst');

bis) SAL > any(select SAL from EE
where JOB = 'Analyst');

6) select * from BU where VID not in (select distinct VID from EE);

↳ bis) (select * from BU)
except (-- en Oracle)
(select VID, NAME, CITY...)

quand pas scalaire; " = " sinon.

session 11 TP1

7) select ENAME from EE where EID in (select distinct EID from IN);

8) select ENAME from EE where (select count(*) from PRJ) =
(select count(*) from IN where IN.EID = EE.EID)

ation: stats sur un ensemble

percent: ~

9. select * from BU where (select count(*)
from EE
where UID = BU.UID)
>= 3;

10. select * from EE natural join BU where CITY in
(select CITY from EE natural join BU
where ENAME = 'Blake')

Exercice 6.

1. select avg(SAL), JOB from EE group by JOB;
select sum(SAL), HIRED from EE.

2. select max(SAL), ENAME from EE natural join BU
group by ENAME;

3. select max(AVG-SAL) from (select avg(SAL) from EE group by UID) T
as AVG-SAL
renommage de la colonne qu'on veut.
having UID is not null.

4. select count(*), e2. ENAME from EE e1 join EE e2
on e1. ENGR = e2. EN
group by e2. ENAME;

5. select ENAME from BU natural join EE group by UID, ENAME
having count(*) >= 3;

6. select avg(SAL), ENAME, JOB from EE : natural join BU
group by ENAME, JOB;

7. select CITY from BU group by CITY having count (*) >= 3;

8. select count(*) , ENAME from EE ^{left outer} natural join ID group by EID, ENAME;

9. select ENAME from IN natural join EE group by EID, ENAME having sum(PCT) > 100;

Exercise 7.

create table EE (

EID int,

ENAME varchar(30);

JOB varchar(20);

HIRED date;

SAL decimal(7, 2);

COMM decimal(6, 2);

UID int;

EMGR int;

primary key (EID);

foreign key (UID) references BU(UID);

foreign key (EMGR) references EE(EID);

non total de chiffre

chiffres après la virgule.