

CSI – TP1 – SALEZ Nathan

Cours 1

BDD emp-dept

- 1) Quels sont les noms et les jobs des employés ?

Requête SQL :

```
SELECT ENAME, JOB FROM emp ;
```

ENAME	JOB
SMITH	CLERK
ALLEN	SALESMAN
WARD	SALESMAN
JONES	MANAGER
MARTIN	SALESMAN
BLAKE	MANAGER
CLARK	MANAGER
SCOTT	ANALYST
KING	PRESIDENT
TURNER	SALESMAN
ADAMS	CLERK
JAMES	CLERK
FORD	ANALYST
MILLER	CLERK

- 2) Quels sont les jobs (sans doublon) des employés ?

Requête SQL :

```
SELECT DISTINCT JOB FROM emp ;
```

JOB
CLERK
SALESMAN
MANAGER
ANALYST
PRESIDENT

- 3) Liste des employés (tous les champs) triés par Numéro de MGR croissant (NULL) (qui est le dernier employé ?)

Requête SQL :

SELECT * FROM emp

ORDER BY MGR;

EMPNO	ENAME	JOB	MGR	1	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT	NULL		1981-11-17	5000.00	NULL	10
7788	SCOTT	ANALYST	7566		1987-04-19	3000.00	NULL	20
7902	FORD	ANALYST	7566		1982-12-03	3000.00	NULL	20
7499	ALLEN	SALESMAN	7698		1981-02-20	1600.00	300.00	30
7521	WARD	SALESMAN	7698		1981-02-22	1250.00	500.00	30
7654	MARTIN	SALESMAN	7698		1981-09-28	1250.00	1400.00	30
7844	TURNER	SALESMAN	7698		1981-09-08	1500.00	0.00	30
7900	JAMES	CLERK	7698		1981-12-03	950.00	NULL	30
7934	MILLER	CLERK	7782		1982-01-23	1300.00	NULL	10
7876	ADAMS	CLERK	7788		1987-05-23	1100.00	NULL	20
7566	JONES	MANAGER	7839		1981-04-02	2975.00	NULL	20
7698	BLAKE	MANAGER	7839		1981-05-01	2850.00	NULL	30
7782	CLARK	MANAGER	7839		1981-06-09	2450.00	NULL	10
7369	SMITH	CLERK	7902		1980-12-17	800.00	NULL	20

- 4) Nom, Job, Deptno des employés triés par Deptn croissant et Job décroissant.

Requête SQL :

```
SELECT ENAME, JOB, DEPTNO FROM emp
ORDER BY DEPTNO ASC, JOB DESC;
```

ENAME	JOB ▾ 2	DEPTNO ▲ 1
KING	PRESIDENT	10
CLARK	MANAGER	10
MILLER	CLERK	10
JONES	MANAGER	20
SMITH	CLERK	20
ADAMS	CLERK	20
SCOTT	ANALYST	20
FORD	ANALYST	20
ALLEN	SALESMAN	30
WARD	SALESMAN	30
MARTIN	SALESMAN	30
TURNER	SALESMAN	30
BLAKE	MANAGER	30
JAMES	CLERK	30

- 5) Liste des Noms d'employés (sans doublon) triés par Nom décroissant.

Requête SQL :

SELECT DISTINCT ENAME FROM emp

ORDER BY ENAME DESC;

ENAME ▾ 1
WARD
TURNER
SMITH
SCOTT
MILLER
MARTIN
KING
JONES
JAMES
FORD
CLARK
BLAKE
ALLEN
ADAMS

- 6) Liste des Noms et Jobs d'employés (sans doublon) triés par Nom décroissant (sans doublon porte sur quoi ?).

Requête SQL :

SELECT DISTINCT ENAME, JOB FROM emp

ORDER BY ENAME DESC;

ENAME	JOB
WARD	SALESMAN
TURNER	SALESMAN
SMITH	CLERK
SCOTT	ANALYST
MILLER	CLERK
MARTIN	SALESMAN
KING	PRESIDENT
JONES	MANAGER
JAMES	CLERK
FORD	ANALYST
CLARK	MANAGER
BLAKE	MANAGER
ALLEN	SALESMAN
ADAMS	CLERK

-- Sans doublon porte sur le couple (ENAME,JOB)

- 7) Liste des Noms, Jobs, Salaire total (SALAIRE_TOTAL) des employés (Salaire et Commission).

Requête SQL :

```
SELECT ENAME,JOB, SAL+COMM AS SALAIRE_TOTAL FROM emp
```

```
WHERE COMM IS NOT NULL
```

```
UNION
```

```
SELECT ENAME,JOB,SAL AS SALAIRE_TOTAL FROM emp
```

```
WHERE COMM IS NULL;SELECT ENAME,JOB, SAL+COMM AS SALAIRE_TOTAL FROM emp;
```

ENAME	JOB	SALAIRE_TOTAL
ALLEN	SALESMAN	1900.00
WARD	SALESMAN	1750.00
MARTIN	SALESMAN	2650.00
TURNER	SALESMAN	1500.00
SMITH	CLERK	800.00
JONES	MANAGER	2975.00
BLAKE	MANAGER	2850.00
CLARK	MANAGER	2450.00
SCOTT	ANALYST	3000.00
KING	PRESIDENT	5000.00
ADAMS	CLERK	1100.00
JAMES	CLERK	950.00
FORD	ANALYST	3000.00
MILLER	CLERK	1300.00

-- On fait une disjonction de cas sur COMM.

- 8) Liste des Noms, Jobs, Salaire total sur l'année des employés

Requête SQL :

```
SELECT ENAME, JOB, (SAL+COMM)*12 AS SALAIRE_TOTAL_PAR_AN FROM emp
WHERE COMM IS NOT NULL
UNION
SELECT ENAME, JOB, SAL*12 AS SALAIRE_TOTAL_PAR_AN FROM emp
WHERE COMM IS NULL;
```

ENAME	JOB	SALAIRE_TOTAL_PAR_AN
ALLEN	SALESMAN	22800.00
WARD	SALESMAN	21000.00
MARTIN	SALESMAN	31800.00
TURNER	SALESMAN	18000.00
SMITH	CLERK	9600.00
JONES	MANAGER	35700.00
BLAKE	MANAGER	34200.00
CLARK	MANAGER	29400.00
SCOTT	ANALYST	36000.00
KING	PRESIDENT	60000.00
ADAMS	CLERK	13200.00
JAMES	CLERK	11400.00
FORD	ANALYST	36000.00
MILLER	CLERK	15600.00

- 9) Liste des couples Job, Salaire sans doublon (cf. ANALYST)

Requête SQL :

```
SELECT DISTINCT JOB, SAL FROM emp;
```

JOB	SAL
CLERK	800.00
SALESMAN	1600.00
SALESMAN	1250.00
MANAGER	2975.00
MANAGER	2850.00
MANAGER	2450.00
ANALYST	3000.00
PRESIDENT	5000.00
SALESMAN	1500.00
CLERK	1100.00
CLERK	950.00
CLERK	1300.00

BDD world

- 1) Liste de toutes les langues (sans doublon) triées par ordre alphabétique.

Requête SQL :

```
SELECT DISTINCT Language FROM countrylanguage  
ORDER BY Language ASC;
```

Language ▲ 1
Abhyasi
Acholi
Adja
Afar
Afrikaans
Aimará
Ainu
Aizo
Akan
Albaniana
Ambo
Amhara
Ami
Ane
Arabic
Arabic-French
Arabic-French-English
Araucan
Arawakan
Armenian
Asami
Assyrian
Atayal
Avarian
Azerbaijani

- 2) Liste des Continents (sans doublon) triés par ordre inverse alphabétique
Requête SQL :

```
SELECT DISTINCT Continent FROM worldregion  
ORDER BY Continent DESC;
```

Continent ▼ 1
South America
Antarctica
Oceania
Africa
North America
Europe
Asia

- 3) Liste des langues parlées, code pays, officiel et pourcentage, triés par pays croissant puis par Officiel – d’abord les langues officielles puis les langues non officielles d’un pays

Requête SQL :

```
SELECT Language, CountryCode, isOfficial, Percentage
```

```
FROM countrylanguage
```

```
ORDER BY CountryCode ASC, isOfficial ASC
```

Language	CountryCode ▲ 1	isOfficial ▲ 2	Percentage
Dutch	ABW	T	5.3
English	ABW	F	9.5
Papiamentu	ABW	F	76.7
Spanish	ABW	F	7.4
Dari	AFG	T	32.1
Pashto	AFG	T	52.4
Balochi	AFG	F	0.9
Turkmenian	AFG	F	1.9
Uzbek	AFG	F	8.8
Ambo	AGO	F	2.4
Chokwe	AGO	F	4.2
Kongo	AGO	F	13.2
Luchazi	AGO	F	2.4
Luimbe-nganguela	AGO	F	5.4

-- On met d’abord les langues officiels (isOfficial : T), puis les non officiels (isOfficial : F) => décroissant (mais mySQL semble faire le contraire)

- 4) Liste des pays : nom, et population (en millions de personnes).

Requête SQL :

```
SELECT Name, population/1000000 AS Population_En_Millions_Personnes
```

```
FROM country;
```

Name	Population_En_Millions_Personnes
Aruba	0.1030
Afghanistan	22.7200
Angola	12.8780
Anguilla	0.0080
Albania	3.4012
Andorra	0.0780
Netherlands Antilles	0.2170
United Arab Emirates	2.4410
Argentina	37.0320
Armenia	3.5200
American Samoa	0.0680
Antarctica	0.0000
French Southern territories	0.0000
Antigua and Barbuda	0.0680
Australia	18.8860
Austria	8.0918
Azerbaijan	7.7340
Burundi	6.6950
Belgium	10.2390
Benin	6.0970
Burkina Faso	11.9370
Bangladesh	129.1550
Bulgaria	8.1909
Bahrain	0.6170
Bahamas	0.3070

- 5) Liste des pays : nom, Région, Gouvernement, Année d'indépendance triés par Région décroissante puis par Gouvernement croissant et par année d'indépendance croissante.

Requête SQL :

```
SELECT Name,Region,GovernmentForm,IndepYear
```

```
FROM country
```

```
ORDER BY Region DESC, GovernmentForm ASC, IndepYear ASC;
```


Name	Region ▼ 1	GovernmentForm ▲ 2	IndepYear ▲ 3
Netherlands	Western Europe	Constitutional Monarchy	1581
Liechtenstein	Western Europe	Constitutional Monarchy	1806
Monaco	Western Europe	Constitutional Monarchy	1861
Luxembourg	Western Europe	Constitutional Monarchy	1867
Belgium	Western Europe	Constitutional Monarchy, Federation	1830
Austria	Western Europe	Federal Republic	1918
Germany	Western Europe	Federal Republic	1955
Switzerland	Western Europe	Federation	1499
France	Western Europe	Republic	843
Saint Helena	Western Africa	Dependent Territory of the UK	NULL
Nigeria	Western Africa	Federal Republic	1960
Liberia	Western Africa	Republic	1847
Ghana	Western Africa	Republic	1957
Guinea	Western Africa	Republic	1958
Togo	Western Africa	Republic	1960
Niger	Western Africa	Republic	1960
Senegal	Western Africa	Republic	1960
Mali	Western Africa	Republic	1960
Mauritania	Western Africa	Republic	1960
Côte d'Ivoire	Western Africa	Republic	1960
Benin	Western Africa	Republic	1960
Burkina Faso	Western Africa	Republic	1960
Sierra Leone	Western Africa	Republic	1961
Gambia	Western Africa	Republic	1965
Guinea-Bissau	Western Africa	Republic	1974

- 6) Liste des villes (tous les champs) classées par Pays (Croissant), par District (croissant) et par Population (décroissante).

Requête SQL :

SELECT *

FROM city

ORDER BY CountryCode ASC, District ASC, Population DESC;

ID	Name	CountryCode ▲ 1	District ▲ 2	Population ▼ 3
129	Oranjestad	ABW	–	29034
4	Mazar-e-Sharif	AFG	Balkh	127800
3	Herat	AFG	Herat	186800
1	Kabul	AFG	Kabul	1780000
2	Qandahar	AFG	Qandahar	237500
58	Lobito	AGO	Benguela	130000
59	Benguela	AGO	Benguela	128300
57	Huambo	AGO	Huambo	163100
56	Luanda	AGO	Luanda	2022000
60	Namibe	AGO	Namibe	118200
61	South Hill	AIA	–	961
62	The Valley	AIA	–	595
34	Tirana	ALB	Tirana	270000
55	Andorra la Vella	AND	Andorra la Vella	21189
33	Willemstad	ANT	Curaçao	2345

- 7) Liste des pays (tous les champs) triés par PNB / habitant croissant (le critère de tri n'est pas forcément dans les champs affichés).

Requête SQL :

SELECT *,GNP/Population

FROM country

ORDER BY GNP/Population ASC;

Code	Name	Region	SurfaceArea	IndepYear	Population	LifeExpectancy	GNP	GNPold	LocalName	GovernmentForm	HeadOfState	Capital	Code2	GNP/Population ≈ 1
IOT	British Indian Ocean Territory	Eastern Africa	78.00	NULL	0	NULL	0.00	NULL	British Indian Ocean Territory	Dependent Territory of the UK	Elisabeth II	NULL	IO	NULL
SGS	South Georgia and the South Sandwich Islands	Antarctica	3903.00	NULL	0	NULL	0.00	NULL	South Georgia and the South Sandwich Islands	Dependent Territory of the UK	Elisabeth II	NULL	GS	NULL
ATA	Antarctica	Antarctica	13120000.00	NULL	0	NULL	0.00	NULL	-	Co-administrated		NULL	AQ	NULL
ATF	French Southern territories	Antarctica	7780.00	NULL	0	NULL	0.00	NULL	Terres australes françaises	Nonmetropolitan Territory of France	Jacques Chirac	NULL	TF	NULL
HMD	Heard Island and McDonald Islands	Antarctica	359.00	NULL	0	NULL	0.00	NULL	Heard and McDonald Islands	Territory of Australia	Elisabeth II	NULL	HM	NULL
UMI	United States Minor Outlying Islands	Micronesia/Caribbean	16.00	NULL	0	NULL	0.00	NULL	United States Minor Outlying Islands	Dependent Territory of the US	George W. Bush	NULL	UM	NULL
BVT	Bouvet Island	Antarctica	59.00	NULL	0	NULL	0.00	NULL	Bouvetøya	Dependent Territory of Norway	Harald V	NULL	BV	NULL
FRO	Faroe Islands	Nordic Countries	1399.00	NULL	43000	78.4	0.00	NULL	Føroyar	Part of Denmark	Margrethe II	901	FO	0.000000
NIU	Niue	Polynesia	260.00	NULL	2000	NULL	0.00	NULL	Niue	Nonmetropolitan Territory of New Zealand	Elisabeth II	2805	NU	0.000000
FLK	Falkland Islands	South America	12173.00	NULL	2000	NULL	0.00	NULL	Falkland Islands	Dependent Territory of the UK	Elisabeth II	763	FK	0.000000
CCK	Cocos (Keeling) Islands	Australia and New Zealand	14.00	NULL	600	NULL	0.00	NULL	Cocos (Keeling) Islands	Territory of Australia	Elisabeth II	2317	CC	0.000000
VIR	Virgin Islands, U.S.	Caribbean	347.00	NULL	93000	78.1	0.00	NULL	Virgin Islands of the United States	US Territory	George W. Bush	4067	VI	0.000000
CXR	Christmas Island	Australia and New Zealand	135.00	NULL	2500	NULL	0.00	NULL	Christmas Island	Territory of Australia	Elisabeth II	1791	CX	0.000000
MNP	Northern Mariana Islands	Micronesia	464.00	NULL	78000	75.5	0.00	NULL	Northern Mariana Islands	Commonwealth of the US	George W. Bush	2913	MP	0.000000
NFK	Norfolk Island	Australia and New Zealand	36.00	NULL	2000	NULL	0.00	NULL	Norfolk Island	Territory of Australia	Elisabeth II	2806	NF	0.000000
TMP	East Timor	Southeast Asia	14874.00	NULL	885000	46.0	0.00	NULL	Timor Timur	Administrated by the UN	José Alexandre Gusmão	1522	TP	0.000000
PCN	Pitcairn	Polynesia	49.00	NULL	50	NULL	0.00	NULL	Pitcairn	Dependent Territory of the UK	Elisabeth II	2912	PN	0.000000
WLF	Wallis and Futuna	Polynesia	200.00	NULL	15000	NULL	0.00	NULL	Wallis-et-Futuna	Nonmetropolitan Territory of France	Jacques Chirac	3536	WF	0.000000
SJM	Svalbard and Jan Mayen	Nordic Countries	62422.00	NULL	3200	NULL	0.00	NULL	Svalbard og Jan Mayen	Dependent Territory of Norway	Harald V	938	SJ	0.000000
SHN	Saint Helena	Western Africa	314.00	NULL	6000	76.8	0.00	NULL	Saint Helena	Dependent Territory of the UK	Elisabeth II	3063	SH	0.000000
MYT	Mayotte	Eastern Africa	373.00	NULL	149000	59.5	0.00	NULL	Mayotte	Territorial Collectivity of France	Jacques Chirac	2514	YT	0.000000
GRL	Greenland	North America	2166090.00	NULL	56000	68.1	0.00	NULL	Kalaallit Nunaat/Groenland	Part of Denmark	Margrethe II	917	GL	0.000000
TKL	Tokelau	Polynesia	12.00	NULL	2000	NULL	0.00	NULL	Tokelau	Nonmetropolitan Territory of New Zealand	Elisabeth II	3333	TK	0.000000
SPM	Saint Pierre and Miquelon	North America	242.00	NULL	7000	77.6	0.00	NULL	Saint-Pierre-et-Miquelon	Territorial Collectivity of France	Jacques Chirac	3067	PM	0.000000
STP	Sao Tome and Principe	Central Africa	964.00	1975	147000	65.3	6.00	NULL	São Tomé e Príncipe	Republic	Miguel Trovoada	3172	ST	0.000041

8) Liste des pays : nom, PNB / habitant (GNP_per_h), triés par PNB / habitant croissant.

Requête SQL :

SELECT Name, GNP/Population AS GNP_per_h

FROM country

ORDER BY GNP_per_h ASC;

Name	GNP_per_h ▲ 1
British Indian Ocean Territory	NULL
South Georgia and the South Sandwich Islands	NULL
Antarctica	NULL
French Southern territories	NULL
Heard Island and McDonald Islands	NULL
United States Minor Outlying Islands	NULL
Bouvet Island	NULL
Faroe Islands	0.000000
Niue	0.000000
Falkland Islands	0.000000
Cocos (Keeling) Islands	0.000000
Virgin Islands, U.S.	0.000000
Christmas Island	0.000000
Northern Mariana Islands	0.000000
Norfolk Island	0.000000
East Timor	0.000000
Pitcairn	0.000000
Wallis and Futuna	0.000000
Svalbard and Jan Mayen	0.000000
Saint Helena	0.000000
Mayotte	0.000000
Greenland	0.000000
Tokelau	0.000000
Saint Pierre and Miquelon	0.000000
Sao Tome and Principe	0.000041

- 9) Liste des pays triés par Gouvernement. Pour les pays ayant le même Gouvernement afficher d'abord les pays non indépendants (comment apparaissent les pays non indépendants ?).

Requête SQL :

```
SELECT Name, GovernmentForm, IndepYear
FROM country
ORDER BY GovernmentForm ASC, IndepYear ASC;
```

Name	GovernmentForm ▲ 1	IndepYear ▲ 2
East Timor	Administrated by the UN	NULL
Palestine	Autonomous Area	NULL
Antarctica	Co-administrated	NULL
Northern Mariana Islands	Commonwealth of the US	NULL
Puerto Rico	Commonwealth of the US	NULL
Japan	Constitutional Monarchy	-660
Denmark	Constitutional Monarchy	800
Sweden	Constitutional Monarchy	836
United Kingdom	Constitutional Monarchy	1066
Thailand	Constitutional Monarchy	1350
Spain	Constitutional Monarchy	1492
Netherlands	Constitutional Monarchy	1581
Nepal	Constitutional Monarchy	1769
Liechtenstein	Constitutional Monarchy	1806
Monaco	Constitutional Monarchy	1861
Luxembourg	Constitutional Monarchy	1867
Norway	Constitutional Monarchy	1905
New Zealand	Constitutional Monarchy	1907
Jordan	Constitutional Monarchy	1946
Cambodia	Constitutional Monarchy	1953
Morocco	Constitutional Monarchy	1956
Jamaica	Constitutional Monarchy	1962
Lesotho	Constitutional Monarchy	1966
Barbados	Constitutional Monarchy	1966
Bahamas	Constitutional Monarchy	1973

-- Les pays non indépendants ont pour valeur de IndepYear NULL (valeur égale à 0 sur MySQL)

- 10) Liste des pays triés par nombre d'années d'indépendance croissantes (où se situent les pays non indépendants ?)

Requête SQL :

```
SELECT Name, 2018-IndepYear AS nb_years_indep
FROM country
ORDER BY nb_years_indep ASC;
```

United States Minor Outlying Islands	NULL
Virgin Islands, British	NULL
Virgin Islands, U.S.	NULL
Wallis and Futuna	NULL
Palau	24
Czech Republic	25
Eritrea	25
Slovakia	25
Bosnia and Herzegovina	26
Armenia	27
Azerbaijan	27
Belarus	27
Estonia	27
Georgia	27
Croatia	27
Kazakhstan	27
Kyrgyzstan	27
Lithuania	27
Latvia	27
Moldova	27
Macedonia	27
Russian Federation	27
Slovenia	27
Tajikistan	27
Turkmenistan	27
Ukraine	27
Uzbekistan	27
Micronesia, Federated States of	28
Marshall Islands	28
Namibia	28
Brunei	34
Saint Kitts and Nevis	35
Antigua and Barbuda	37
Belize	37
Vanuatu	38

-- Les pays dépendants se situent en premier (2018-NULL = NULL = 0)

Cours 2

BDD emp_dept

I/Requêtes de Sélection

- 1) Nom des employés commençant par un S.

Requête SQL :
SELECT ENAME
FROM emp
WHERE ENAME LIKE 'S%';

ENAME
SMITH
SCOTT

- 2) Nom des employés comportant au moins un S.

Requête SQL :
SELECT ENAME
FROM emp
WHERE ENAME LIKE '%S%';

ENAME
SMITH
JONES
SCOTT
ADAMS
JAMES

- 3) Nom des employés comportant au moins deux L.

Requête SQL :
SELECT ENAME
FROM emp
WHERE ENAME LIKE '%L%L%';

ENAME
ALLEN
MILLER

- 4) Nom des employés dont le salaire est supérieur strictement à 1000

Requête SQL :
SELECT ENAME
FROM emp
WHERE SAL > 1000;

ENAME
ALLEN
WARD
JONES
MARTIN
BLAKE
CLARK
SCOTT
KING
TURNER
ADAMS
FORD
MILLER

- 5) Nom des employés dont le salaire est compris entre 1300 et 1000 (2 requêtes différentes).

Requête SQL 1 :

```
SELECT ENAME  
FROM emp  
WHERE SAL BETWEEN 1000 AND 1300;
```

Requête SQL 2 :

```
SELECT ENAME  
FROM emp  
WHERE 1000 <= SAL AND SAL <= 1300;
```

ENAME
WARD
MARTIN
ADAMS
MILLER

- 6) Nom des employés embauchés après le 21/04/1981.

Requête SQL :

```
SELECT ENAME  
FROM emp  
WHERE HIREDATE > '1981-04-21';
```


ENAME

MARTIN

BLAKE

CLARK

SCOTT

KING

TURNER

ADAMS

JAMES

FORD

MILLER

- 7) Nom des employés ayant une commission (NDLR : une commission supérieure à 0 et qui existe.)

Requête SQL :

SELECT ENAME

FROM emp

WHERE COMM IS NOT NULL AND COMM > 0;

ENAME

ALLEN

WARD

MARTIN

- 8) Nom des employés n'ayant pas une commission (NDLR : une commission égale à 0 ou inexistante.).

Requête SQL :

SELECT ENAME

FROM emp

WHERE COMM IS NULL OR COMM = 0;

ENAME

SMITH

JONES

BLAKE

CLARK

SCOTT

KING

TURNER

ADAMS

JAMES

FORD

MILLER

- 9) Nom des employés qui n'ont pas de MGR

Requête SQL :

SELECT ENAME

```
FROM emp
WHERE MGR IS NULL;
ENAME
KING
```

10) Nom des employés qui touchent plus de commission que de salaire

```
Requête SQL :
SELECT ENAME
FROM emp
WHERE COMM IS NOT NULL AND COMM > SAL;
ENAME
MARTIN
```

11) Nom des employés qui sont dans le département 20 ou 30 (2 requêtes différentes)

```
Requête SQL 1 :
SELECT ENAME
FROM emp
WHERE DEPTNO IN (20,30);
```

```
Requête SQL 2 :
SELECT ENAME
FROM emp
WHERE DEPTNO=20 OR DEPTNO=30;
```

```
ENAME
SMITH
ALLEN
WARD
JONES
MARTIN
BLAKE
SCOTT
TURNER
ADAMS
JAMES
FORD
```

II/ Requetes de Jointure et de Sélection

1) Nom des employés et nom des départements d'affectation des employés.

```
Requête SQL :
SELECT emp.ENAME, dept.DNAME
FROM emp, dept
WHERE emp.DEPTNO = dept.DEPTNO;
```

ENAME	DNAME
CLARK	ACCOUNTING
KING	ACCOUNTING
MILLER	ACCOUNTING
SMITH	RESEARCH
JONES	RESEARCH
SCOTT	RESEARCH
ADAMS	RESEARCH
FORD	RESEARCH
ALLEN	SALES
WARD	SALES
MARTIN	SALES
BLAKE	SALES
TURNER	SALES
JAMES	SALES

- 2) Nom des employés dont le département d'affectation se situe à Dallas.

Requête SQL :

```
SELECT emp.ENAME
```

```
FROM emp, dept
```

```
WHERE emp.DEPTNO = dept.DEPTNO AND LOC = "DALLAS";
```

ENAME
SMITH
JONES
SCOTT
ADAMS
FORD

- 3) Nom des employés et nom de leur Manager (dans cet ordre).

Requête SQL :

```
SELECT emp.ENAME AS EMPLOYE_NAME, mgr.ENAME AS MANAGER_NAME
```

```
FROM emp, emp mgr
```

```
WHERE emp.MGR = mgr.EMPNO;
```

EMPLOYEE_NAME	MANAGER_NAME
SMITH	FORD
ALLEN	BLAKE
WARD	BLAKE
JONES	KING
MARTIN	BLAKE
BLAKE	KING
CLARK	KING
SCOTT	JONES
TURNER	BLAKE
ADAMS	SCOTT
JAMES	BLAKE
FORD	JONES
MILLER	CLARK

- 4) Nom des employés embauchés après l'employé SMITH.

Requête SQL :

SELECT ENAME

FROM emp

WHERE HIREDATE > (SELECT HIREDATE

FROM emp

WHERE ENAME = 'SMITH');

ENAME
ALLEN
WARD
JONES
MARTIN
BLAKE
CLARK
SCOTT
KING
TURNER
ADAMS
JAMES
FORD
MILLER

- 5) Nom des employés, nom de leur département, nom de leur Manager et nom du département de leur Manager.

Requête SQL :

SELECT emp.ENAME, dept_emp.DNAME, mgr.ENAME, dept_mgr.DNAME

FROM emp, emp mgr, dept dept_emp, dept dept_mgr

WHERE emp.MGR = mgr.EMPNO

AND dept_emp.DEPTNO = emp.DEPTNO

AND dept_mgr.DEPTNO = mgr.DEPTNO;

ENAME	DNAME	ENAME	DNAME
CLARK	ACCOUNTING	KING	ACCOUNTING
MILLER	ACCOUNTING	CLARK	ACCOUNTING
JONES	RESEARCH	KING	ACCOUNTING
BLAKE	SALES	KING	ACCOUNTING
SMITH	RESEARCH	FORD	RESEARCH
SCOTT	RESEARCH	JONES	RESEARCH
ADAMS	RESEARCH	SCOTT	RESEARCH
FORD	RESEARCH	JONES	RESEARCH
ALLEN	SALES	BLAKE	SALES
WARD	SALES	BLAKE	SALES
MARTIN	SALES	BLAKE	SALES
TURNER	SALES	BLAKE	SALES
JAMES	SALES	BLAKE	SALES

- 6) Nom des employés qui ont un salaire strictement plus petit que celui de KING.

Requête SQL :

SELECT ENAME

FROM emp

WHERE SAL < (SELECT SAL
FROM emp
WHERE ENAME = 'KING');

ENAME
SMITH
ALLEN
WARD
JONES
MARTIN
BLAKE
CLARK
SCOTT
TURNER
ADAMS
JAMES
FORD
MILLER

- 7) Nom des employés qui ont un salaire plus petit ou égal à la commission de MARTIN.

Requête SQL :

SELECT ENAME

FROM emp

WHERE SAL < (SELECT COMM
FROM emp
WHERE ENAME = 'MARTIN');

ENAME
SMITH
WARD
MARTIN
ADAMS
JAMES
MILLER

-- Remarque : Martin possède un salaire plus petit que sa commission, il est donc normal que son nom se retrouve dans la table réponse.

- 8) Nom des employés qui ont un salaire plus grand ou égal à celui de leur Manager.

Requête SQL :

```
SELECT emp.ENAME
FROM emp, emp mgr
WHERE      mgr.EMPNO = emp.MGR
AND        mgr.SAL <= emp.SAL;
```

ENAME
SCOTT
FORD

- 9) Nom des employés qui ont comme Manager BLAKE.

Requête SQL :

```
SELECT emp.ENAME
FROM emp, emp mgr
WHERE mgr.EMPNO = emp.MGR
AND mgr.ENAME LIKE 'BLAKE';
```

ENAME
ALLEN
WARD
MARTIN
TURNER
JAMES

- 10) Nom des employés qui ont le même manager que BLAKE

Requête SQL :

```
SELECT emp.ENAME
FROM emp, emp mgr, emp selected_emp, emp selected_emp_mgr
WHERE mgr.EMPNO = emp.MGR
AND selected_emp.ENAME = 'BLAKE'
AND selected_emp_mgr.EMPNO = selected_emp.MGR
AND mgr.EMPNO = selected_emp_mgr.EMPNO
AND emp.ENAME != 'BLAKE' ;
```

ENAME
JONES
CLARK

-- On exclut BLAKE des résultats.

- 11) Nom des employés qui ont comme Manager un employé qui a lui-même comme Manager KING.

Requête SQL :

```
SELECT emp.ENAME
FROM emp, emp emp_mgr_one, emp selected_mgr_two
WHERE emp_mgr_one.EMPNO = emp.MGR
AND selected_mgr_two.EMPNO = emp_mgr_one.MGR
AND selected_mgr_two.ENAME = 'KING';
```

ENAME
SCOTT
FORD
ALLEN
WARD
MARTIN
TURNER
JAMES
MILLER

- 12) Nom des employés qui gagnent le même salaire qu'un autre employé.

Requête SQL :

```
SELECT emp.ENAME
FROM emp, emp other_emp
WHERE emp.EMPNO != other_emp.EMPNO
AND emp.SAL = other_emp.SAL;
```

ENAME
MARTIN
WARD
FORD
SCOTT

-- Puisqu'un employé gagne le même salaire que lui-même, on l'enlève des résultats.

-- Par ailleurs, on n'observe pas de doublons sur cette requête, car aucun employé ne possède le même nom de famille.

- 13) Même question que 12 mais sans doublon

Requête SQL :

```
SELECT DISTINCT emp.ENAME
FROM emp, emp other_emp
WHERE emp.EMPNO != other_emp.EMPNO
AND emp.SAL = other_emp.SAL;
```

ENAME
MARTIN
WARD
FORD
SCOTT

-- on obtient le même résultat qu'au 12 car la table emp contient que noms différents.

14) Nom des employés qui sont le Manager d'un employé qui est CLERK (CLERK est un JOB).

Requête SQL :

```
SELECT mgr.ENAME
FROM emp selected_emp_job, emp mgr
WHERE selected_emp_job.MGR = mgr.EMPNO
AND selected_emp_job.JOB = "CLERK";
```

ENAME

FORD

SCOTT

BLAKE

CLARK

15) Nom des employés qui sont Manager d'un employé se trouvant dans le département SALES ou RESEARCH.

Requête SQL :

```
SELECT DISTINCT mgr.ENAME
FROM emp, emp mgr
WHERE emp.MGR = mgr.EMPNO
AND emp.DEPTNO IN (SELECT DEPTNO
                    FROM dept
                    WHERE DNAME IN('SALES', 'RESEARCH'));
```

ENAME

FORD

KING

JONES

SCOTT

BLAKE

-- Un manager peut être manager de plusieurs employés, on peut donc avoir des doublons si l'on ne met pas « DISTINCT »

BDD world

I/ Requêtes de Sélection

1) Nom des Pays se trouvant en Western Europe.

Requête SQL :

```
SELECT Name FROM country
WHERE REGION = "WESTERN EUROPE";
```


Name
Austria
Belgium
Switzerland
Germany
France
Liechtenstein
Luxembourg
Monaco
Netherlands

- 2) Nom des Pays dont le PIB par habitant > 10000€ (PIB de la table est exprimé en million d'euros).

Requête SQL :

SELECT Name FROM country

WHERE (GNP*1000000)/Population > 10000;

Name
Andorra
United Arab Emirates
Australia
Austria
Belgium
Bahrain
Bahamas
Bermuda
Brunei
Canada
Switzerland
Cayman Islands
Cyprus
Germany
Denmark
Spain
Finland
France
United Kingdom
Gibraltar
Greece
Hong Kong
Ireland
Iceland
Israel

-- on considère que GNP est le PIB ici, et qu'il regroupe le PIB de toute la population

- 3) Nom des Pays devenu indépendants entre 1900 et 2000.

Requête SQL :

```
SELECT Name FROM country
```

```
WHERE IndepYear BETWEEN 1900 AND 2000 ;
```

Name

Afghanistan

Angola

Albania

United Arab Emirates

Armenia

Antigua and Barbuda

Australia

Austria

Azerbaijan

Burundi

Benin

Burkina Faso

Bangladesh

Bulgaria

Bahrain

Bahamas

Bosnia and Herzegovina

Belarus

Belize

Barbados

Brunei

Bhutan

Botswana

Central African Republic

Côte d'Ivoire

- 4) Nom des Pays pas encore indépendants.

Requête SQL :

```
SELECT Name FROM country
```

```
WHERE IndepYear IS NULL ;
```

Name
Aruba
Anguilla
Netherlands Antilles
American Samoa
Antarctica
French Southern territories
Bermuda
Bouvet Island
Cocos (Keeling) Islands
Cook Islands
Christmas Island
Cayman Islands
Western Sahara
Falkland Islands
Faroe Islands
Gibraltar
Guadeloupe
Greenland
French Guiana
Guam
Hong Kong
Heard Island and McDonald Islands
British Indian Ocean Territory
Macao
Northern Mariana Islands

- 5) Nom des Langues parlées par plus de 50% d'un pays.
 SELECT Language FROM countrylanguage
 WHERE Percentage > 50.0 ;

Language
Papiamento
Pashto
Albaniana
Papiamento
Spanish
Armenian
Samoan
Creole English
English
German
Azerbaijani
Kirundi
Dutch
Mossi
Bengali
Bulgariana
Arabic
Creole English
Serbo-Croatian
Belorussian
English
English
Spanish
Portuguese
Bajan

6) Idem 5 mais sans doublon.

Requête SQL :

```
SELECT DISTINCT Language FROM countrylanguage
WHERE Percentage > 50.0 ;
```

Language
Papiamentu
Pashto
Albaniana
Spanish
Armenian
Samoan
Creole English
English
German
Azerbaijani
Kirundi
Dutch
Mossi
Bengali
Bulgariana
Arabic
Serbo-Croatian
Belorussian
Portuguese
Bajan
Tswana
Chinese
Kongo
Comorian
Crioulo

- 7) Nom des Villes se trouvant dans un pays dont le code est BEL OU FRA OU MCO.

Requête SQL :

```
SELECT Name FROM city
```

```
WHERE CountryCode IN('BEL','FRA','MCO');
```

Name
Antwerpen
Gent
Charleroi
Liège
Bruxelles [Brussel]
Brugge
Schaerbeek
Namur
Mons
Paris
Marseille
Lyon
Toulouse
Nice
Nantes
Strasbourg
Montpellier
Bordeaux
Rennes
Le Havre
Reims
Lille
St-Étienne
Toulon
Grenoble

- 8) Nom des Régions du Monde de trouvant dans le continent Africa.

Requête SQL :

```
SELECT Region FROM worldregion
WHERE Continent = 'AFRICA' ;
```

Region
Central Africa
Eastern Africa
Northern Africa
Southern Africa
Western Africa

- 9) Nom des pays se trouvant dans le continent Europe.

Requête SQL :

```
SELECT country.Name
FROM country, worldregion selected_continent
WHERE selected_continent.Region = country.Region
AND selected_continent.Continent = "Europe";
```

Name

Estonia
Lithuania
Latvia
United Kingdom
Ireland
Bulgaria
Belarus
Czech Republic
Hungary
Moldova
Poland
Romania
Russian Federation
Slovakia
Ukraine
Denmark
Finland
Faroe Islands
Iceland
Norway
Svalbard and Jan Mayen
Sweden
Albania
Andorra
Bosnia and Herzegovina

- 10) Code des pays où le Français est parlé, et (est une langue officielle ou parlé par plus de 55,5% de la population)

Requête SQL :

```
SELECT CountryCode FROM countrylanguage
WHERE Language = "FRENCH" AND( isOfficial = 'T' OR Percentage >= 55.5) ;
```

CountryCode

BDI

BEL

CAN

CHE

FRA

GLP

HTI

LUX

MCO

MDG

MTQ

MYT

NCL

PYF

RWA

SPM

SYC

VUT

II/ Sélection et Jointure

- 1) Nom des Pays où le Français est parlé par plus de 1.000.000 d'habitants.

Requête SQL :

```
SELECT country.Name
```

```
FROM countrylanguage language_selected, country
```

```
WHERE country.Code = language_selected.CountryCode
```

```
AND language_selected.Language = "French"
```

```
AND language_selected.Percentage*country.Population > 1000000;
```

Name

Belgium

Canada

Switzerland

France

Italy

Luxembourg

Monaco

Mauritius

Mayotte

New Caledonia

French Polynesia

United States

Vanuatu

- 2) Nom des Pays se trouvant dans le Continent Antarctique.

Requête SQL :

```
SELECT country.Name
FROM country, worldregion selected_continent
WHERE selected_continent.Region = country.Region
AND selected_continent.Continent = "Antarctica";
```

Name
Antarctica
French Southern territories
Bouvet Island
Heard Island and McDonald Islands
South Georgia and the South Sandwich Islands

- 3) Nom de la Capitale du pays où se trouve la ville de Monte-Carlo

Requête SQL :

```
SELECT capital.Name
FROM city capital, country, city selected_city
WHERE selected_city.CountryCode = country.code
AND country.Capital = capital.ID
AND selected_city.Name = "Monte-Carlo";
```

Name
Monaco-Ville

- 4) Nom des Pays (sans doublon) qui ont au moins une même langue officielle que la Belgique.

Requête SQL :

```
SELECT DISTINCT country.Name
FROM country selected_country, countrylanguage selected_country_language, country,
countrylanguage
WHERE country.Name != "Belgium"
AND selected_country.Name = "Belgium"
AND selected_country.Code = selected_country_language.CountryCode
AND country.Code = countrylanguage.CountryCode
AND selected_country_language.isOfficial = 'T'
AND countrylanguage.isOfficial = 'T'
AND selected_country_language.Language = countrylanguage.Language;
```

Name
Aruba
Netherlands Antilles
Netherlands
Burundi
Canada
Switzerland
France
Guadeloupe
Haiti
Luxembourg
Monaco
Madagascar
Martinique
Mayotte
New Caledonia
French Polynesia
Rwanda
Saint Pierre and Miquelon
Seychelles
Vanuatu
Austria
Germany
Liechtenstein

- 5) Nom des Pays (sans doublon) qui ont au moins deux mêmes langues officielles que la Belgique.

Requête SQL :

```
SELECT DISTINCT country.Name
FROM country selected_country,
country,
countrylanguage selected_country_language_one,
countrylanguage selected_country_language_two,
countrylanguage countrylanguage_official_one,
countrylanguage countrylanguage_official_two
WHERE
    country.Name != "Belgium"
AND    selected_country.Name = "Belgium"
AND    selected_country.Code = selected_country_language_one.CountryCode
AND    selected_country.Code = selected_country_language_two.CountryCode
AND    selected_country_language_one.Language = countrylanguage_official_one.Language
AND    selected_country_language_two.Language = countrylanguage_official_two.Language
AND    selected_country_language_one.isOfficial = 'T'
AND    countrylanguage_official_one.isOfficial = 'T'
AND    selected_country_language_two.isOfficial = 'T'
```

```

AND    countrylanguage_official_two.isOfficial = 'T'
AND    country.Code = countrylanguage_official_one.CountryCode
AND    country.Code = countrylanguage_official_two.CountryCode
AND    countrylanguage_official_one.Language != countrylanguage_official_two.Language
AND    selected_country_language_one.Language != selected_country_language_two.Language;

```

Name
Switzerland
Luxembourg

- 6) Nom des Villes se trouvant dans le même Pays que la ville de Schaerbeek.

Requête SQL :

```

SELECT city.Name
FROM city selected_city, city
WHERE selected_city.Name != city.Name
AND    city.CountryCode = selected_city.CountryCode
AND    selected_city.Name = "Schaerbeek";

```

Name
Antwerpen
Gent
Charleroi
Liège
Bruxelles [Brussel]
Brugge
Namur
Mons

- 7) Nom du Continent de la ville de Schaerbeek.

Requête SQL :

```

SELECT worldregion.Continent
FROM worldregion, country, city selected_city
WHERE selected_city.Name = "Schaerbeek"
AND    country.Code = selected_city.CountryCode
AND    worldregion.Region = country.Region;

```

Continent
Europe

- 8) Quel est le nom de la Capitale du Pays de la ville de Mons ?

Requête SQL :

```

SELECT capital.Name
FROM city capital, country, city selected_city
WHERE selected_city.CountryCode = country.code
AND    country.Capital = capital.ID
AND    selected_city.Name = "Mons";

```

Name
Bruxelles [Brussel]

-- identique à la question 3

Cours 3

BDD emp_dept

I/ Regroupement sur toute la table

- 1) Nombre total d'employés ?
Requête SQL :
`SELECT COUNT(*) AS NB_EMPLOYES FROM emp ;`
Résultat : 14
- 2) Somme de tous les salaires ?
Requête SQL :
`SELECT SUM(SAL) AS SUM_SALARIES FROM emp ;`
Résultat : 29025.00
- 3) Masse salariale totale (Salaire et commission) ?
Requête SQL :
`SELECT SUM(SAL)+SUM(COMM) AS MASSE_SALARIALE FROM emp;`
Résultat : 31225.00
- 4) Moyenne, Ecart type, Min, Max des Salaires ?
Requête SQL :
`SELECT AVG(SAL) AS MOYENNE, STDDEV(SAL) AS ECART_TYPE, MIN(SAL) AS MINIMUM, MAX(SAL) AS MAXIMUM FROM emp;`
--On ne prend pas en compte les commissions.
- 5) Nombre de départements ?
Requête SQL :
`SELECT COUNT(*) AS NB_DEPTS FROM dept ;`
Résultat : 4
- 6) Nombre de départements avec des employés?
Requête SQL :
`SELECT COUNT(DISTINCT DEPTNO) AS NB_DEPTS_NON_VIDES FROM emp ;`
Résultat : 3
- 7) Nombre de Jobs différents ?
Requête SQL :
`SELECT COUNT(DISTINCT JOB) AS NB_JOBS FROM emp;`
Résultat : 5
- 8) Nombre de Managers ? (NDLR : nombre de personnes ayant un salarié à une hiérarchie au dessous)
Requête SQL :
`SELECT COUNT(DISTINCT MGR) AS NB_MANAGERS FROM emp;`
Résultat : 6

II/ Regroupement par critère

- 1) Nombre d'employés par Job (Afficher le Job sinon inutile).

Requête SQL :

```
SELECT COUNT(*) AS NB_EMPLOYES, JOB from emp
GROUP BY JOB;
```

NB_EMPLOYES	JOB
2	ANALYST
4	CLERK
3	MANAGER
1	PRESIDENT
4	SALESMAN

- 2) Nombre d'employés par couple (Job, Département).

Requête SQL :

```
SELECT COUNT(*) AS NB_EMPLOYES, JOB,DEPTNO from emp
GROUP BY JOB,DEPTNO;
```

NB_EMPLOYES	JOB	DEPTNO
2	ANALYST	20
1	CLERK	10
2	CLERK	20
1	CLERK	30
1	MANAGER	10
1	MANAGER	20
1	MANAGER	30
1	PRESIDENT	10
4	SALESMAN	30

- 3) Maximum des salaires par Job.

Requête SQL :

```
SELECT JOB, MAX(SAL) FROM emp
GROUP BY JOB;
```

JOB	MAX(SAL)
ANALYST	3000.00
CLERK	1300.00
MANAGER	2975.00
PRESIDENT	5000.00
SALESMAN	1600.00

- 4) Somme des salaires par Département.

Requête SQL :

```
SELECT DEPTNO, SUM(SAL) FROM emp
GROUP BY DEPTNO;
```

DEPTNO	SUM(SAL)
10	8750.00
20	10875.00
30	9400.00

III/ Regroupement par critère + sélection après regroupement

- 1) Numéro de Département dans lequel il y a au moins 2 employés (pensez à vérifier).

Requête SQL :

```
SELECT DEPTNO FROM emp
```

```
GROUP BY DEPTNO
```

```
HAVING COUNT(*) >= 2 ;
```

DEPTNO
10
20
30

- 2) Somme des Salaires par Département pour les Départements dont la Somme des Salaires est supérieur strictement à 3000.

Requête SQL :

```
SELECT DEPTNO, SUM(SAL) FROM emp
```

```
GROUP BY DEPTNO
```

```
HAVING SUM(SAL) > 3000;
```

DEPTNO	SUM(SAL)
10	8750.00
20	10875.00
30	9400.00

- 3) Nombre de Jobs (avec doublon) par Département pour les Départements à plus (strictement) de 3 Jobs.

Requête SQL :

```
SELECT DEPTNO, COUNT(JOB) FROM emp
```

```
GROUP BY DEPTNO
```

```
HAVING COUNT(JOB) > 3;
```

DEPTNO	COUNT(JOB)
20	5
30	6

- 4) Idem mais sans doublon (résultat différent avec requête 3 ?)

Requête SQL :

```
SELECT DEPTNO, COUNT(DISTINCT JOB) FROM emp
```

```
GROUP BY DEPTNO
```

```
HAVING COUNT(DISTINCT JOB) > 3;
```

DEPTNO	COUNT(DISTINCT JOB)
--------	---------------------

-- Résultat différent de la requête précédente (les départements 20 et 30 ont chacun 3 jobs distincts, donc n'apparaissent pas)

IV-A/ Sélection avant regroupement + regroupement sur toute la table

- 1) Nombre d'employés sans MGR.

Requête SQL :

```
SELECT COUNT(*) AS NB_EMP_SANS_MGR FROM emp
WHERE MGR IS NOT NULL;
```

Résultat : 1

- 2) Nombre d'employés ayant un salaire supérieur à 1000.

Requête SQL :

```
SELECT COUNT(*) AS EMPLOYES_SAL_SUP_1000 FROM emp
WHERE SAL > 1000;
```

Résultat : 12.

- 3) Nombre d'employés embauchés après le 1^{er} janvier 1982.

Requête SQL :

```
SELECT COUNT(*) AS NB_EMPLOYES_EMBAUCHES_APRES_01011982
FROM emp
WHERE HIREDATE > '1982-01-01';
```

Résultat : 4.

IV-B/ Sélection avant regroupement + regroupement par critère

- 1) Nombre d'employés ayant un salaire supérieur à 1000 par Département.

Requête SQL :

```
SELECT DEPTNO, COUNT(*) FROM emp
WHERE SAL > 1000
GROUP BY DEPTNO;
```

DEPTNO	COUNT(*)
10	3
20	4
30	5

- 2) Nombre d'employés avec MGR par Job (quid si sans MGR ?).

Requête SQL :

```
SELECT JOB, COUNT(*) AS NB_EMP_MGR
FROM emp
WHERE MGR IS NOT NULL
GROUP BY JOB;
```

JOB	NB_EMP_MGR
ANALYST	2
CLERK	4
MANAGER	3
SALESMAN	4

-- Il n'y a qu'un seul job où il n'y a pas de MGR : Président.

- 3) Nombre d'employés par Job embauchés après le 1^{er} janvier 1982

Requête SQL :

```
SELECT JOB, COUNT(*) AS NB_EMPLOYES_EMBAUCHES_APRES_01011982
FROM emp
WHERE HIREDATE > '1982-01-01'
GROUP BY JOB;
```

JOB	NB_EMPLOYES_EMBAUCHES_APRES_01011982
ANALYST	2
CLERK	2

V/ Sélection avant regroupement + regroupement par critère + sélection après regroupement

- 1) Par département Nombre d'employés embauchés après le 1^{er} janvier 1982 pour les départements ayant plus de 2 employés embauchés après le 1^{er} janvier 1982.

Requête SQL :

```
SELECT DEPTNO, COUNT(*)
FROM emp
WHERE HIREDATE > '1982-01-01'
GROUP BY DEPTNO
HAVING COUNT(*) > 2;
```

DEPTNO	COUNT(*)
20	3

- 2) Nombre de Managers par Job (du Manager) pour Job à plus de 1 (strictement) Manager.

Requête SQL :

```
SELECT mgr.JOB, COUNT(DISTINCT mgr.EMPNO)
FROM emp mgr, emp
WHERE emp.MGR = mgr.EMPNO
GROUP BY mgr.JOB
HAVING COUNT(DISTINCT mgr.EMPNO) > 1;
```

JOB	COUNT(DISTINCT mgr.EMPNO)
ANALYST	2
MANAGER	3

BDD world

- 1) Somme des populations des villes par pays pour les pays à plus de 1.000.000 habitants dans les villes.

Requête SQL :

```
SELECT country.Name, SUM(city.Population)
FROM city, country
WHERE city.CountryCode = country.Code
GROUP BY country.Name
HAVING SUM(city.Population > 1000000);
```


Name	SUM(city.Population)
Afghanistan	2332100
Algeria	5192179
Angola	2561600
Argentina	19996563
Armenia	1633100
Australia	11313666
Austria	2384273
Azerbaijan	2464000
Bangladesh	8569906
Belarus	4741000
Brazil	85876862
Bulgaria	2696915
Cameroon	3522554
Canada	12673840
Chile	9717970
China	175953614
Colombia	20250990
Congo, The Democratic Republic of the	9864615
Côte d'Ivoire	3191137
Cuba	4629925
Czech Republic	2634711
Dominican Republic	2438276
Ecuador	5744142
Egypt	20083079
Ethiopia	3190334

2) PNB / Continent

Requête SQL :

```
SELECT worldregion.Continent, SUM(country.GNP) AS PNB
```

```
FROM worldregion,country
```

```
WHERE country.Region = worldregion.Region
```

```
GROUP BY worldregion.Continent;
```

Continent	PNB
Asia	7655392.00
Europe	9498865.00
North America	9688627.20
Africa	580375.00
Oceania	419774.70
Antarctica	0.00
South America	1511874.00

3) Nombre de langues parlées par Pays.

Requête SQL :

```
SELECT country.Name, COUNT(countrylanguage.Language)
FROM country, countrylanguage
WHERE country.Code = countrylanguage.CountryCode
GROUP BY country.Name;
```

Name	COUNT(countrylanguage.Language)
Afghanistan	5
Albania	3
Algeria	2
American Samoa	3
Andorra	4
Angola	9
Anguilla	1
Antigua and Barbuda	2
Argentina	3
Armenia	2
Aruba	4
Australia	8
Austria	8
Azerbaijan	4
Bahamas	2
Bahrain	2
Bangladesh	7
Barbados	2
Belarus	4
Belgium	6
Belize	4
Benin	7
Bermuda	1
Bhutan	3
Bolivia	4

- 4) Par Pays Population totale parlant une langue non officielle (problème ?)

Requête SQL :

```
SELECT country.Name, SUM(language.Percentage*country.Population/100) AS POP_SPEAK-
ING_UNOFFICIAL_LANGUAGE
FROM country, countrylanguage language
WHERE country.Code = language.CountryCode AND language.isOfficial = 'F'
GROUP BY country.Name;
```

Name	POP_SPEAKING_UNOFFICIAL_LANGUAGE
Afghanistan	2635520.03250
Albania	64622.79843
Algeria	4405940.00000
American Samoa	2107.99994
Andorra	48047.99881
Angola	12285612.13510
Antigua and Barbuda	65075.99792
Argentina	740640.02207
Armenia	91519.99664
Aruba	96407.99695
Australia	1491994.02927
Austria	412681.80555
Azerbaijan	564581.99631
Bahamas	306999.99122
Bahrain	0.00000
Bangladesh	1291550.01925
Barbados	256769.99588
Belarus	194483.99756
Belgium	501711.00976
Belize	115680.00230
Benin	5658015.92441
Bhutan	1061999.97974
Bolivia	8329.00012
Botswana	1542522.00232
Brazil	2551725.02535
Brunei	135135.99781
Bulgaria	1285971.26485
Burkina Faso	8928876.02277
Burundi	0.00000
Cambodia	1228480.00000
Cameroon	12173595.18702

- 5) Pays où le nombre de langues officielles est supérieur ou égal à 3 (code d'abord puis Nom des Pays)

Requête SQL :

```
SELECT country.Code, country.Name, COUNT(countrylanguage.Language)
FROM country, countrylanguage
WHERE country.Code = countrylanguage.CountryCode AND countrylanguage.isOfficial = 'T'
GROUP BY country.Name
HAVING COUNT(countrylanguage.Language) >= 3
ORDER BY country.Code ASC, country.Name ASC;
```

Code	Name	COUNT(countrylanguage.Language)
BEL	Belgium	3
BOL	Bolivia	3
CHE	Switzerland	4
LUX	Luxembourg	3
PER	Peru	3
SGP	Singapore	3
VUT	Vanuatu	3
ZAF	South Africa	4

Cours 4

BDD emp_dept

- 1) Pour les employés travaillant à DALLAS ou CHICAGO :

regroupement dans le département de BOSTON

salaire = moyenne des salaires du département actuel + 10%

commission = moyenne des commissions du département actuel + 20%

date = date du jour

Requête SQL (1 - faux):

```
/*UPDATE emp
SET DEPTNO = (SELECT DEPTNO FROM dept WHERE LOC LIKE "Boston"),
    SAL = (SELECT AVG(SAL)*1.1 FROM emp E2 WHERE E2.DEPTNO = emp.DEPTNO),
    COMM = (SELECT AVG(COMM)*1.1 FROM emp E2 WHERE E2.DEPTNO =
emp.DEPTNO),
    HIREDATE = CURRENT_DATE
WHERE DEPTNO IN (SELECT DEPTNO FROM dept WHERE LOC IN("Dallas","Chicago"));*/
-- On ne peut pas utiliser la table emp dans la clause FROM, puisqu'il s'agit de la table modi-
fiée. Une solution serait de copier la table emp dans une table annexe_emp.
```

Script SQL (2 - bon) :

```
CREATE TABLE emp_annexe AS SELECT * FROM emp;
UPDATE emp
SET DEPTNO = (SELECT DEPTNO FROM dept WHERE LOC LIKE "Boston"),
    SAL = (SELECT AVG(IFNULL(SAL,0))*1.1 FROM emp_annexe WHERE emp_an-
nexe.DEPTNO = emp.DEPTNO),
    COMM = (SELECT AVG(IFNULL(COMM,0))*1.2 FROM emp_annexe WHERE emp_an-
nexe.DEPTNO = emp.DEPTNO),
    HIREDATE = CURRENT_DATE
WHERE DEPTNO IN (SELECT DEPTNO FROM dept WHERE LOC IN("Dallas","Chicago"));
DROP TABLE emp_annexe;
--IFNULL(value,1) renvoie value si value != NULL, 1 sinon.
```

- 2) Affecter aux employés gagnant moins que le salaire moyen de leur emploi, un salaire égal

au salaire moyen de leur emploi.

Requête SQL (1) :

```
/*UPDATE emp
SET emp.SAL = ( SELECT AVG(e1.SAL) FROM emp e1
                WHERE e1.JOB = emp.JOB )
WHERE emp.SAL < ( SELECT AVG(e1.SAL) FROM emp e1
                WHERE e1.JOB = emp.JOB );*/
-- Même remarque que la question précédente.
```

Script SQL (2) :

```
CREATE TABLE emp_annexe AS SELECT * FROM emp;
UPDATE emp
SET SAL = ( SELECT AVG(IFNULL(SAL,0)) FROM emp_annexe
            WHERE emp_annexe.JOB = emp.JOB )
WHERE SAL < ( SELECT AVG(IFNULL(SAL,0)) FROM emp_annexe
            WHERE emp_annexe.JOB = emp.JOB );
DELETE FROM emp_annexe;
```