

A

School Case Study

EXERCISE 1 :

1. Start the sqlplus interaction.

Connect using ssh to one of these unix machines :

bootybay	astranaar
brill	auberdine
darkshire	bloodhoof
goldshire	crossroads
ironforge	darnassus
kargath	dolanaar
kharanos	feathermoon
menethil	gadgetzan
southshore	orgrimmar
stonard	ratchet
stormwind	senjin
undercity	thunderbluff

Then follow these instructions :

<http://perso.efrei.fr/~bernardi/oracle.html>

Essentially, from the shell type :

```
$$>
```

```
$$> ./opt/oracle/oracle.env
```

```
$$> sqlplus login@sgbd
```

Your password = login is your usual efrei account name.

2. Test these commands : run them and try to understand what is happening.

⇒⇒ First contact with SQL/Oracle : useful functions you should know about

```
SELECT RPAD('Soleil',17,'bla') "RPAD exemple" FROM DUAL;
```

```
SELECT LPAD('DESS EID',15,'*') "LPAD exemple" FROM DUAL;
```

```

SELECT SUBSTR('DESS EID',6,3) "SUBSTR exemple" FROM DUAL;
SELECT SUBSTR('ABCDEFGHJI',-5,4) "SUBSTR exemple" FROM DUAL;

SELECT TO_CHAR (SYSDATE, 'MM-DD-YYYY HH24:MI:SS') "Now" FROM DUAL;

SELECT LENGTH('WEB WAREHOUSE') "Longueur en caractères" FROM DUAL;

SELECT ROUND(17.0958,1) "ROUND exemple" FROM DUAL;
SELECT ROUND(17.58,2) "ROUND exemple" FROM DUAL;

SELECT TRUNC(1958.0917,1) "TRUNC exemple" FROM DUAL;
SELECT TRUNC(1958.0917,2) "TRUNC exemple" FROM DUAL;

SELECT ROUND(TO_DATE('17-SEP-2002'), 'YEAR') "New Year" FROM DUAL;

SELECT EXTRACT(YEAR FROM SYSDATE) FROM DUAL;

SELECT ADD_MONTHS(SYSDATE,7) FROM DUAL;

SELECT TRUNC(MONTHS_BETWEEN(SYSDATE, TO_DATE('19-JUN-2001'))) AS AGEBB FROM DUAL;

SELECT TO_NUMBER(TO_CHAR(SYSDATE, 'YYYY')) FROM DUAL;

```

First run the School creation script, then try these commands :

```

SELECT DECODE(YEAR, 1, 'First', 2, 'Second', 'Value not 1 or 2 !!')
AS STUDY_YEAR FROM STUDENTS;

SELECT UPPER(NAME) FROM STUDENTS;

SELECT LOWER(NAME) FROM STUDENTS;

SELECT NVL(SPECIALTY, 'Null value detected') FROM FACULTY;
SELECT NVL(SPECIALTY, ' Null value detected ') AS SPEC_FACULTY FROM FACULTY;

```

If you need to change the date format (default depends on locale settings) :

```
ALTER SESSION SET NLS_DATE_FORMAT = 'DD-MM-YYYY';
```

3. Test these instructions :

⇒⇒ **Discovering the Oracle dictionary**

show user ;

select * from **all_users**;

pause

desc user_tables ;

select table_name from user_tables;

pause

column table_name **heading** NAME_de_l_utilisateur

select table_name from user_tables;

pause

column object_name **format** a30

set pages 30

select object_name, object_type from **user_objects** order by object_type;

pause

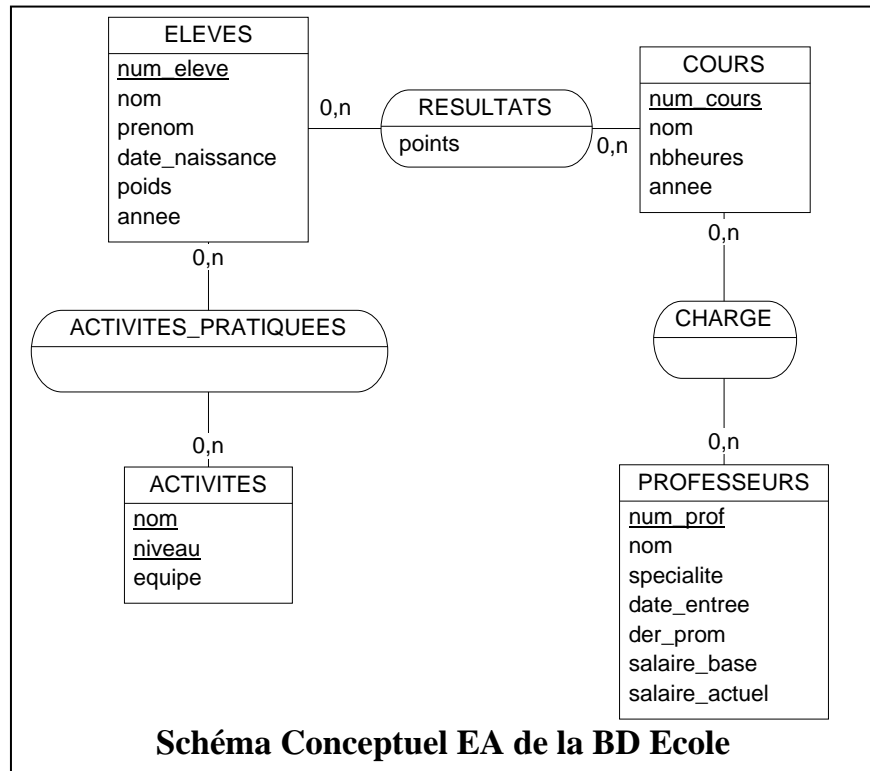
desc user_constraints;

column constraint_name format a30

select constraint_name from user_constraints where table_name='&matable';

EXERCISE 2 :

Let us consider the Engineering School case study. The following conceptual schema describes the database using entity-relationship notation (sorry about the naming in French):



2. The corresponding relational schema is described below :

STUDENTS(**SID**, NAME, FIRSTNAME, BIRTH_DATE, WEIGHT, YEAR)
 COURSE(**CID**, NAME, HOURS, YEAR)
 FACULTY(**FID**, NAME, SPECIALTY, DATE_HIRE, LAST_RAISE, BASE_SALARY, CURRENT_SALARY)
 ACTIVITY (**LEVEL,NAME** , TEAM)

 RESULTS(**SID, CID**, POINTS)
 OFFER(**FID, CID**)
 PRACTICE(**SID, LEVEL, NAME**)

Schéma relationnel de la BD Ecole

The SQL table creation **script**, automatically generated using a IDE tool such as PowerAMC is given here :

⇒ The tables that correspond to entities (rectangles in the diagram) are the following :

```
CREATE TABLE STUDENTS
(
    SID          NUMBER(4)          NOT NULL,
    NAME         VARCHAR2(25)       NOT NULL,
    FIRSTNAME    VARCHAR2(25)       NOT NULL,
    BIRTH_DATE   DATE               NOT NULL,
    WEIGHT       NUMBER,
    YEAR        NUMBER,
```

CONSTRAINT PK_STUDENTS PRIMARY KEY (SID));

CREATE TABLE COURSE

(CID NUMBER(2) NOT NULL,
 NAME VARCHAR(20) NOT NULL,
 HOURS NUMBER(2),
 ANNE NUMBER(1),
 CONSTRAINT PK_COURSE PRIMARY KEY (CID));

CREATE TABLE FACULTY

(FID NUMBER(4) NOT NULL,
 NAME VARCHAR2(25) NOT NULL,
 SPECIALTY VARCHAR2(20),
 DATE_ENTRÉE DATE,
 LAST_RAISE DATE,
 BASE_SALARYNUMBER,
 CURRENT_SALARY NUMBER,
 CONSTRAINT PK_FACULTY PRIMARY KEY (FID));

CREATE TABLE ACTIVITY

(LEVEL NUMBER(1) NOT NULL,
 NAME VARCHAR2(20) NOT NULL,
 TEAM VARCHAR2(32),
 CONSTRAINT PK_ACTIVITY PRIMARY KEY (LEVEL, NAME));

⇒ The tables corresponding to relationships (ellipse in diagram) are the following :

```
CREATE TABLE RESULTS
(   SID          NUMBER(4)          NOT NULL,
    CID          NUMBER(4)          NOT NULL,
    POINTS       NUMBER,
    CONSTRAINT PK_RESULTS PRIMARY KEY (SID, CID) );
```

```
CREATE TABLE OFFER
(   NUM_PRO      NUMBER(4)          NOT NULL,
    CID          NUMBER(4)          NOT NULL,
    CONSTRAINT PK_OFFER PRIMARY KEY (CID, FID) );
```

```
CREATE TABLE PRACTICE
(   SID          NUMBER(4)          NOT NULL,
    LEVEL        NUMBER(1)          NOT NULL,
    NAME         VARCHAR2(20)       NOT NULL,
    CONSTRAINT PK_PRACTICE PRIMARY KEY (SID, LEVEL, NAME) );
```

⇒⇒☒ **Attention : The foreign key constraints are defined separately here !!!**

```
ALTER TABLE RESULTS
ADD CONSTRAINT FK_RESULTAT_STUDENTS FOREIGN KEY (SID)
REFERENCES STUDENTS (SID) ;
```

```
ALTER TABLE RESULTS
ADD CONSTRAINT FK_RESULTAT_COURSE FOREIGN KEY (CID)
REFERENCES COURSE (CID) ;
```

```
ALTER TABLE OFFER
ADD CONSTRAINT FK_OFFER_COURSE FOREIGN KEY (CID)
REFERENCES COURSE (CID) ;
```

```
ALTER TABLE OFFER
ADD CONSTRAINT FK_OFFER_FACULTY FOREIGN KEY (FID)
REFERENCES FACULTY (FID) ;
```

```
ALTER TABLE PRACTICE
ADD CONSTRAINT FK_ACTIVITE_STUDENTS FOREIGN KEY (SID)
REFERENCES STUDENTS (SID) ;
```

```
ALTER TABLE PRACTICE
ADD CONSTRAINT FK_ACTIVITEPR_ACTIVITE FOREIGN KEY (NAME, LEVEL)
REFERENCES ACTIVITY (NAME, LEVEL) ;
```

Data insertion is done using :

INSERT INTO table (attributs) VALUES (values) ;

To validate your work : **COMMIT ;**

To undo your actions since the last commit :

ROLLBACK ;

Please execute the command : set autocommit on

This executes a commit after each successful instruction, and prevents overloading the server when the whole class is working on the same database.

- 2.1. Run the table creation script. (file create_school.sql)
- 2.2. List the structure of table STUDENTS and list its contents.

2.3. Modify the structure of table STUDENTS.

Add the attributes :

ZipCode : a 5 digit number type, and
Town a string of maximum 20 characters

2.4. Update the address of students SID 1, 2, 5 et 7 (respectively) with the following data :

75013 ; paris
93800 ; EPINAY / seine
93800 ; EPINAY SUR SEINE
91000 ; EPINAY / ORGE

2.5. Create a new table CITIES with the following schema :

CITIES (ZipCode, CITY_NAME)

Define the constraint « A city name should be upper case ».

2.6. Fill this CITIES with correct data :

Warning : EPINAY / seine is a different string than EPINAY SUR SEINE

ZIPCODE	CITY_NAME
75001	PARIS
75013	PARIS
93800	EPINAY SUR SEINE
93430	Villetaneuse <i>Remark :This line should provoke a constraint violation error !!!</i>
91000	EPINAY SUR ORGE
Etc...	

```
INSERT INTO CITIES (ZIPCODE, CITY_NAME)
```

```
VALUES (75001, 'PARIS') ;
```

```
INSERT INTO CITIES (ZIPCODE, CITY_NAME)
```

```
VALUES (75013, 'PARIS') ;
```

2.7. Update table STUDENTS. Use a single request to do so. Display the updated contents.

```
UPDATE Table1
SET AttribToUpdate = (SELECT Attribute FROM Table2
WHERE JoinCondition);
```

3. Query the database to obtain the following information :

- 1- Obtain the list (NAME, firstname , birth date) of all students.
- 2- Obtain all information available about activities.
- 3- Obtain the list of specialties from faculty.
- 4- Obtain the NAME and first name of students that weigh **less than 45 kilos** and enrolled in first year or of students in second year.
- 5- Obtain the NAME of students who weigh between **60 and 80 kilos**.
- 6- Obtain the NAME of FACULTY whose specialty is '**poésie**' or **SQL**.
- 7- Obtain the NAME of students whose NAME **starts by 'L'**.
- 8- Obtain the NAME of FACULTY whose specialty is unknown.
- 9- Obtain the NAME and first name of students that weigh less than **45 kilos and enrolled in first year**.
- 10- Obtain, for each FACULTY, his NAME and his specialty. If the specialty is unknown, display the string: '********'.
- 11- What are the names and first names of students the practice **surf at LEVEL 1**. Write this request in 5 different ways.
- 12- Obtain the NAME of students in team **AMC INDUS**.
- 13- Obtain the pair of faculty names that have the same specialty.
- 14- For each faculty specialized in **sql**, obtain the NAME, the current **monthly** salary, and his raise per month with respect to his base salary.
- 15- Obtain the NAME of FACULTY whose raise with respect to the base salary exceeds **25%**.
- 16- Display the points **Tsuno** has obtained in each course using a total out of **100 rather than out of 20**.
- 17- Obtain the average weight of students in first year.
- 18- Obtain the total points of student of sid 3.
- 19- Obtain the smallest and largest result points of student **Brisefer**.
- 20- Obtain the number of students enrolled in second year.
- 21- What is the average monthly raise of salary of FACULTY specialized in **SQL** ?
- 22- Obtain the year of FACULTY **Pucette's last raise**.
- 23- For each FACULTY, display the hire date, his last raise year and the number of years between these two dates.
- 24- Display the average age of students. This average should be displayed as an integer.
- 25- Display the NAME of FACULTY for whom 50 months at least separate the hire date and the last raise date.
- 26- Obtain the list of students who will be 24 within the next 4 months (24 is wrong, adapt the request to obtain some results).
- 27- Obtain a list of students in alphabetical name order.
- 28- Display in decreasing order the points obtained by student **Tsuno**, with totals out of 100 rather than 20.
- 29- Obtain for each first year student his name and his average points.
- 30- Obtain the average points of students in first year whose total points is at least 40.
- 31- Obtain the student who has the best total points.
- 32- Obtain the NAME of students who play in team **AMC INDUS**.
- 33- Find the first year students whose average points is superior to the global average of students in first year.
- 34- Obtain the NAME and weight of first year students heavier than all second year student.
- 35- Obtain the NAME and weight of first year students heavier than at least one second year student.
- 36- Obtain the NAME of FACULTY that do not offer course of cid 1.

37- Obtain the NAME of first year students that obtained an average of at least 60% and that play tennis.

38- list FACULTY who OFFER all COURSE of second year ; we ask for FID and name.

39- Students that practice all activities. We ask for SID and name.

Remark : Request 38 requires a division ; an answer is given here to serve as template for other division queries. You can also try the oracle specific DIVIDE sql keyword.

TITLE ' FACULTY who OFFER all COURSE of second year '

```
SELECT FID, NAME
FROM FACULTY
WHERE NOT EXISTS
  (SELECT * FROM COURSE WHERE YEAR = 2
  AND NOT EXISTS
    (SELECT * FROM OFFER
     WHERE FACULTY.FID = OFFER.FID
     AND OFFER.CID = COURSE.CID));
```

Note that you can create views to help you answer complex queries :

```
CREATE OR REPLACE VIEW view_name (attributes if you want to rename them)
AS (SELECT etc FROM ...);
```

4. Create an index to improve performance

Create an index over student name.

```
CREATE INDEX NDXSTUDENTS ON STUDENTS (NAME ASC);
```

5. Test using :

```
SET TIMING ON      &      SET TIMING OFF
```