



HAUTE ECOLE
Provinciale de
H A I N A U T
CONDORCET

PROJET JEE

Dombou marius et Kamgo teuguia

2016-2017

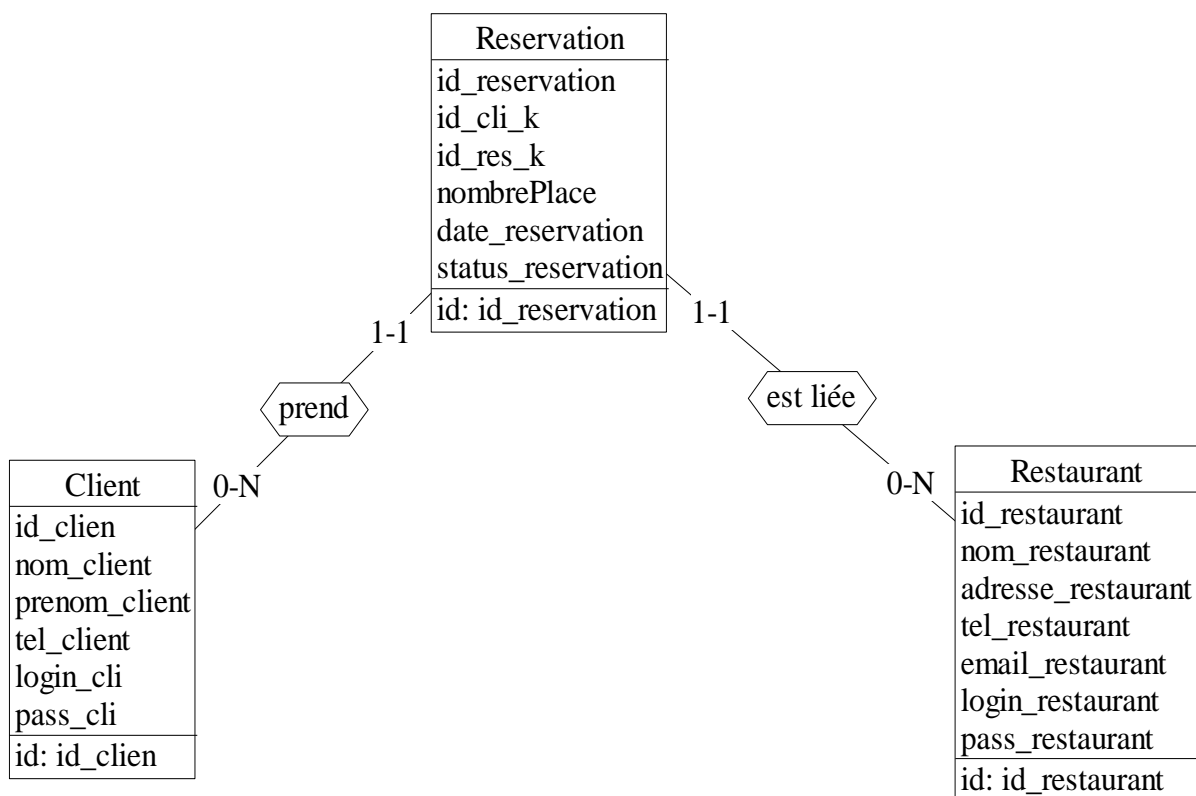
Énoncé

L'application développée est une plateforme qui permet aux personnes de s'y connecter après avoir créé un compte, et de prendre des rendez-vous à l'un ou plusieurs restaurants enregistrés dans la base de données.

L'utilisateur pourra non seulement prendre un ou plusieurs rendez-vous pour un ou plusieurs restaurants mais aussi consulter ses réservations.

Un restaurant pourra s'enregistrer et consulter ses rendez-vous.

Base de données



Explication :

Un client est caractérisé par son nom, prénom, téléphone, login et password, il pourra prendre un ou plusieurs rendez-vous caractérisés par le nombre de la place et la date de réservation.

Une réservation est liée à un et un seul restaurant qui sera caractérisé par un nom, adresse, téléphone, email, login et password.

script

```
/*=====*/
/* Table : Client                                     */
/*=====*/
create table Client (
    id_cli          INTEGER          not null,
    nom_cli         VARCHAR2(50),
    prenom_cli      VARCHAR2(50)     not null,
    adresse_cli     VARCHAR2(100)    not null,
    tel_cli         VARCHAR2(20),
    login_cli       VARCHAR2(21)     not null,
    pass_cli        VARCHAR2(31)     not null,
    constraint PK_Client primary key (id_cli)
);

/*=====*/
/* Table : Restaurant                                 */
/*=====*/
create table Restaurant (
    id_res          INTEGER          not null,
    nom_res         VARCHAR2(100)    not null,
    adresse_res     varchar2(40)     not null,
    tel_res         VARCHAR2(20)     not null,
    mail_res        VARCHAR2(100)    not null,
    login_res       VARCHAR2(100)    not null,
    pass_res        VARCHAR2(100)    not null,
    constraint PK_Restaurant primary key (id_res)
);
```

```

/*****
/* AUTO INCREMENT */
*****/
create sequence nextIdClient_seq start with 1;

create or replace trigger T_client
before insert on Client
for each row
begin
    select nextIdClient_seq.nextval
    into    :new.id_cli
    from    dual;
end;
/

create sequence nextIdRestaurant_seq start with 1;

create or replace trigger T_restaurant
before insert on Restaurant
for each row
begin
    select nextIdRestaurant_seq.nextval
    into    :new.id_res
    from    dual;
end;

create sequence nextIdReservation_seq START with 1;

create or replace trigger T_reservation
before insert on Reservation
for each row
begin
    select nextIdReservation_seq.nextval
    into    :new.id_reservation
    from    dual;
end;
/

```

Chaque classe possède un package pour contenir les fonctions liées à la classe.

Dans chaque package contient une directive pragma qui permet d'enclencher une exception qui sera traitée dans l'application.

Package Client

Contient les fonctions suivantes :

createClient: qui permet d'insérer un nouveau client

loginUser: qui permet à un client de se connecter

findById : qui permet de recherche un client par son id

rdvClient : permet d'obtenir les rendez-vous du client

```
CREATE OR REPLACE PACKAGE PKG_Client
AS
/* VARIABLE GLOBAL */
TYPE cur_login IS REF CURSOR;
/* DETAILS PROCEDURE */
PROCEDURE createClient (
    nom IN client.nom_cli%TYPE,
    prenom IN CLIENT.PRENOM_CLI%TYPE,
    adresse IN client.adresse_cli%TYPE,
    tel IN client.tel_cli%type,
    login IN client.login_cli%TYPE,
    pass IN client.pass_cli%TYPE
);

PROCEDURE loginUser(
    login IN client.login_cli%TYPE,
    pass IN client.pass_cli%TYPE,
    p_ref OUT cur_login
);

procedure findById(id_f IN CLIENT.ID_CLI%TYPE,cur_res out cur_login);

function rdvClient(id_f IN CLIENT.ID_CLI%TYPE)
    return cur_login;

ex_violationContraite EXCEPTION;
PRAGMA EXCEPTION_INIT(ex_violationContraite,-2291);

END PKG_Client;
```

```

CREATE OR REPLACE PACKAGE BODY PKG_Client
AS
    PROCEDURE createClient (
        nom IN client.nom_cli%TYPE,
        prenom IN CLIENT.PRENOM_CLI%TYPE,
        adresse IN client.adresse_cli%TYPE,
        tel IN client.tel_cli%type,
        login IN client.login_cli%TYPE,
        pass IN client.pass_cli%TYPE
    )
    IS
    /* DECLARATION VARAIBLES */
    BEGIN
        insert into client (NOM_CLI,PRENOM_CLI,ADRESSE_CLI,TEL_CLI,LOGIN_CLI,PASS_CLI)
        values(nom,prenom,adresse,tel,login,pass);
        commit;
    END createClient;

    PROCEDURE loginUser(
        login IN client.login_cli%TYPE,
        pass IN client.pass_cli%TYPE,
        p_ref OUT cur_login
    )
    is
    begin
        OPEN p_ref for
        SELECT *
        FROM CLIENT
        where LOGIN_CLI=login and PASS_CLI=pass;
    end loginUser;

```

```

procedure findById(id_f IN CLIENT.ID_CLI%TYPE,cur_res out cur_login)
is
begin
    OPEN cur_res for
        SELECT *
        FROM CLIENT
        where ID_CLI=id_f;
end findById;

```

```

function rdvClient(id_f IN CLIENT.ID_CLI%TYPE)
    return cur_login
is
    cur_ref cur_login;
begin
    OPEN cur_ref for
        select ID_RESERVATION,ID_RES_K,nombreplace,date_reservation from
            RESERVATION WHERE ID_CLI_K=id_f;
    return cur_ref;
end rdvClient;
END PKG_Client;

```

Package réservation

createReservation : permet de d'insérer une nouvelles réservations.

findById : recherche une réservation avec son id

```
CREATE OR REPLACE PACKAGE PKG_Reservation
AS

/* DETAILS PROCEDURE */
TYPE cur_reser IS REF CURSOR;
/* ***** */
/*      fonction de creation d'une nouvelle reservation      */
/* ***** */
PROCEDURE createReservation (
    id_clien  in client.id_cli%TYPE,
    id_resto  in restaurant.ID_RES%TYPE,
    nbr       in reservation.NOMBREPLACE%TYPE,
    date_r    in reservation.DATE_RESERVATION%TYPE,
    statut    in reservation.STATUT_RESERVATION%TYPE
);
/* ***** */
/*      fonction qui peret de recherche une reservation avec son id      */
/* ***** */
PROCEDURE findById(id_f in reservation.id_reservation%type,p_ref OUT cur_reser);
/* ***** */
/*      fonction de mis a jour d'une reservation      */
/* ***** */
function updateReseration(id_reser in reservation.id_reservation%type,
    stat in reservation.statut_reservation%type
)
return number;

ex_violationContraite EXCEPTION;
PRAGMA EXCEPTION_INIT(ex_violationContraite,-2291);
END PKG_Reservation;
```

```

CREATE OR REPLACE PACKAGE BODY PKG_Reservation
AS
PROCEDURE createReservation (
    id_clien  in client.id_cli%TYPE,
    id_resto  in restaurant.ID_RES%TYPE,
    nbr       in reservation.NOMBREPLACE%TYPE,
    date_r    in reservation.DATE_RESERVATION%TYPE,
    statut    in reservation.STATUT_RESERVATION%TYPE
)
IS
BEGIN
    insert into reservation (ID_CLI_K, ID_RES_K, NOMBREPLACE, DATE_RESERVATION, STATUT_RESERVATION)
        VALUES(id_clien, id_resto, nbr, date_r, statut);
    commit;
END createReservation;

PROCEDURE findById(
    id_f in reservation.id_reservation%type,
    p_ref OUT cur_reser
)
is
begin
    OPEN p_ref for
        SELECT *
        FROM reservation
        where ID_RESERVATION=id_f;
end findById;

function updateReseration(...)
END PKG_Reservation;

```


Package restaurant

Contient les fonctions suivantes :

createRestaurant : qui permet d'insérer un nouveau restaurant

loginRestaurant : qui permet à un restaurant de se connecter

findRd : permet de recherche les réservation d'un restaurant

```
CREATE OR REPLACE PACKAGE PKG_Restaurant
AS
TYPE cur_login IS REF CURSOR;
type rdv is record(id_re RESERVATION.ID_RESERVATION%type,
nbr RESERVATION.NOMBREPLACE%TYPE, date_r reservation.date_reservation%type);
type tab_rdv is table of rdv not null index by binary_integer;
/* DETAILS PROCEDURE */
PROCEDURE createRestaurant (
    nom    IN restaurant.nom_res%type,
    tel    in restaurant.tel_res%TYPE,
    email  in restaurant.mail_res%TYPE,
    login  in restaurant.login_res%TYPE,
    pass   in restaurant.pass_res%TYPE,
    adresse in restaurant.adresse_res%TYPE
);
PROCEDURE loginRestaurant(
    login in restaurant.login_res%TYPE,
    pass  in restaurant.pass_res%TYPE,
    p_ref OUT cur_login
);
PROCEDURE findById(id_f IN RESTAURANT.ID_RES%TYPE,p_ref OUT cur_login);
PROCEDURE findAll(p_ref OUT cur_login);
/*FUNCTION findRdv(
    id_r in RESERVATION.ID_CLI_K%TYPE
) return tab_rdv;*/

procedure findRd(id_r in RESERVATION.ID_CLI_K%TYPE, p_ref OUT cur_login);

ex_violationContraite EXCEPTION;
PRAGMA EXCEPTION_INIT(ex_violationContraite,-2291);
END PKG_Restaurant;
```

```

CREATE OR REPLACE PACKAGE BODY PKG_Restaurant
AS
/*****/
    PROCEDURE createRestaurant (
        nom    IN restaurant.nom_res%type,
        tel    in restaurant.tel_res%TYPE,
        email  in restaurant.mail_res%TYPE,
        login  in restaurant.login_res%TYPE,
        pass   in restaurant.pass_res%TYPE,
        adresse in restaurant.adresse_res%TYPE
    )
    is
    begin
        insert into restaurant (nom_res,tel_res,mail_res,login_res,pass_res,ADRESSE_RES)
            values (nom,tel,email,login,pass,adresse);
        commit;

    END createRestaurant;

/*****/
    PROCEDURE loginRestaurant(
        login in restaurant.login_res%TYPE,
        pass  in restaurant.pass_res%TYPE,
        p_ref OUT cur_login
    )
    is
    begin
        OPEN p_ref for
            SELECT *
            FROM RESTAURANT
            where LOGIN_res=login and PASS_res=pass;
    end loginRestaurant;

```

```

PROCEDURE findById(id_f in RESTAURANT.ID_RES%TYPE,p_ref OUT cur_login)
is
begin
    OPEN p_ref for
        SELECT *
            FROM RESTAURANT
            where ID_RES=id_f;
end findById;
PROCEDURE findAll(p_ref OUT cur_login)
is
begin
    OPEN p_ref for
        SELECT *
            FROM RESTAURANT;
end findAll;

procedure findRd(id_r in RESERVATION.ID_CLI_K%TYPE, p_ref OUT cur_login)
is
begin
OPEN p_ref for
    select ID_RESERVATION,ID_CLI_K,nombreplace,date_reservation from
        RESERVATION WHERE ID_RES_K=id_r;
end findRd;
END PKG_Restaurant;

```