

```

#include <ctype.h>
#include <stdio.h>
#include <string.h>

EXEC SQL BEGIN DECLARE SECTION;
    VARCHAR user[20];
    VARCHAR pwd[20];
    VARCHAR n_equ[8];
    VARCHAR n_equ_w[8];
    VARCHAR nom_equ[30];
    VARCHAR n_proj[8];
    VARCHAR n_proj_lu[8];
    VARCHAR nom_proj[20];
    VARCHAR nom_proj_w[20];
    VARCHAR n_cher_resp[11];
    VARCHAR nom_cher_resp[10];
    VARCHAR n_cher[11];
    VARCHAR nom_cher[10];
    VARCHAR nom_cher_w[10];
    VARCHAR n_specialite[10];
    VARCHAR nom_spec[30];
    int nb_jour_sem;
    int nb_proj_w;
    int nb_cher_spec;
EXEC SQL END DECLARE SECTION;

EXEC SQL INCLUDE sqlca.h;

connection()
{
    /* entree du mot de passe et du nom */

    user.len=asks("Entrer votre user  : ",user.arr);
    pwd.len=asks("Entrer  votre pwd   : ",pwd.arr);

    /* connection a la base */
    EXEC SQL WHENEVER SQLERROR GOTO erreur;
    EXEC SQL CONNECT :user IDENTIFIED BY :pwd;
    printf("\n Connexion reussie ..... \n");
    return(1);
erreur:
    printf("\nErreur a la connection .... ");
    errpt();
    return(0);
    EXEC SQL WHENEVER SQLERROR continue;
}

selection()
{
    n_proj_lu.len =asks("\n\n Entrez le numero du projet ou 'fin': ",
n_proj_lu.arr);
    n_proj_lu.arr[n_proj_lu.len]='\0';

    while (strcmp((char *)n_proj_lu.arr,"fin"))
    {
/* saisie des informations */
    nom_proj.len =asks("\n\n Entrez le nom du projet: ", nom_proj.arr);

```

```

nom_proj.arr[nom_proj.len]='\0';

n_equ.len =asks("\n\n Entrez le numero de l'equipe: ", n_equ.arr);
n_equ.arr[n_equ.len]='\0';

n_cher_resp.len =asks("\n\n Entrez le numero du chercheur responsable: ",
n_cher_resp.arr);
n_cher_resp.arr[n_cher_resp.len]='\0';

n_specialite.len =asks("\n\n Entrez la specialite: ", n_specialite.arr);
n_specialite.arr[n_specialite.len]='\0';

EXEC SQL WHENEVER SQLERROR GOTO erreur2;

EXEC SQL DECLARE c1 CURSOR FOR
SELECT n_chercheur, nom_chercheur
FROM chercheur c
WHERE c.n_equipe=:n_equ
AND c.specialite=:n_specialite;

/* verification des informations saisies */

/* existence de l'equipe */
EXEC SQL WHENEVER NOT FOUND GOTO equipe_inconnue;

EXEC SQL SELECT nom_equipe INTO :nom_equ
      FROM equipe WHERE n_equipe=:n_equ;

EXEC SQL WHENEVER NOT FOUND continue;

/* existence de la specialite */
EXEC SQL WHENEVER NOT FOUND GOTO specialite_inconnue;

EXEC SQL SELECT nom_specialite INTO :nom_spec
      FROM specialite WHERE specialite=:n_specialite;

EXEC SQL WHENEVER NOT FOUND continue;

/* le chercheur responsable appartient a l'equipe */
EXEC SQL SELECT n_equipe INTO :n_equ_w
      FROM chercheur WHERE n_chercheur=:n_cher_resp;

n_equ_w.arr[n_equ_w.len]='\0';

if (strcmp((char *)n_equ_w.arr,(char *)n_equ.arr))
{
    goto mauvaise_equ_cher_resp;
}

/* le projet n'existe pas deja */
EXEC SQL SELECT count(*) INTO :nb_proj_w
      FROM projet WHERE n_projet=:n_proj_lu;

if (nb_proj_w==1)
{
    goto projet_existant;
}

EXEC SQL SELECT count(*) into :nb_cher_spec FROM chercheur c
      WHERE c.n_equipe=:n_equ

```

```

        AND c.specialite=:n_specialite;

if (nb_cher_spec==0)
{
    goto aucun_cher_spec;
}

/* insertion du projet */

EXEC SQL INSERT INTO PROJET
    VALUES (:n_proj_lu, :nom_proj, :n_equ, :n_cher_resp, 0, 0);

EXEC SQL OPEN c1;

EXEC SQL WHENEVER NOT FOUND GOTO fin_c1;

while(1)
{
    EXEC SQL FETCH c1 into :n_cher, :nom_cher;
    n_cher.arr[n_cher.len]='\0';
    nom_cher.arr[nom_cher.len]='\0';
    printf("\n Chercheur : %s ", nom_cher.arr);
    printf("\n entrez le nb de jours sur ce projet: ");
    scanf("%d", &nb_jour_sem);
    EXEC SQL INSERT INTO TRAVAILLER
        VALUES (:n_proj_lu, :n_cher, :nb_jour_sem);
}

fin_c1:
    EXEC SQL CLOSE c1;
    EXEC SQL COMMIT ;
    printf ("\n transaction reussie...");
    getchar();
    n_proj_lu.len =asks("\n\n Entrez le numero du projet ou 'fin': ",
n_proj_lu.arr);
    n_proj_lu.arr[n_proj_lu.len]='\0';
    continue;

equipe_inconnue: printf("Cette equipe n'existe pas..");
    printf("\n transaction annulee.....\n");
    getchar();
    n_proj_lu.len =asks("\n\n Entrez le numero du projet ou 'fin': ",
n_proj_lu.arr);
    n_proj_lu.arr[n_proj_lu.len]='\0';
    continue;

specialite_inconnue: printf("Cette specialite n'existe pas..");
    printf("\n transaction annulee.....\n");
    getchar();
    n_proj_lu.len =asks("\n\n Entrez le numero du projet ou
'fin': ", n_proj_lu.arr);
    n_proj_lu.arr[n_proj_lu.len]='\0';
    continue;

mauvaise_equ_cher_resp:
    printf("\n le chercheur responsable n'appartient pas a l'equipe...");
    printf("\n transaction annulee.....\n");
    getchar();
    n_proj_lu.len =asks("\n\n Entrez le numero du projet ou 'fin': ",
n_proj_lu.arr);

```

```

        n_proj_lu.arr[n_proj_lu.len]='\0';
        continue;

projet_existant:
    printf("\n le projet existe deja ...");
    printf("\n transaction annulee.....\n");
    getchar();
    n_proj_lu.len =asks("\n\n Entrez le numero du projet ou 'fin': ",
n_proj_lu.arr);
    n_proj_lu.arr[n_proj_lu.len]='\0';
    continue;

aucun_cher_spec:
    printf("\n aucun chercheur de l'equipe n'a cette specialite ...");
    printf("\n transaction annulee.....\n");
    getchar();
    n_proj_lu.len =asks("\n\n Entrez le numero du projet ou 'fin': ",
n_proj_lu.arr);
    n_proj_lu.arr[n_proj_lu.len]='\0';
    continue;

}
return;

erreur2:
    EXEC SQL ROLLBACK;
    printf("erreur lors de la transaction...");
    errpt();
    return(0);
    EXEC SQL WHENEVER SQLERROR continue;
}

main()
{
if (connection())
    {
        selection();
        sortie();
    }
    else
        arret();
}

sortie()
{
    EXEC SQL COMMIT RELEASE;
}

arret()
{
    EXEC SQL ROLLBACK RELEASE;
}

errpt()
{
    printf("\n%.70s (%d)\n",sqlca.sqlerrm.sqlerrmc, -sqlca.sqlcode);
}

```

```
int askn(text,variable)
char text[];
int *variable;
{
char s[20];
printf(text);
if ( gets(s) == (char *)0 )
return(EOF);
*variable = atoi(s);
return(1);
}
```

```
int asks(text,variable)
char text[],variable[];
{
printf(text);
return( gets(variable) == (char *)0 ? EOF : strlen(variable) );
}
```