

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

```
EXEC SQL BEGIN DECLARE SECTION;
    VARCHAR user[20];
    VARCHAR pwd[20];
    VARCHAR n_equ_lu[8];
    VARCHAR nom_equ[30];
    VARCHAR n_proj[8];
    VARCHAR nom_proj[20];
    VARCHAR n_cher_resp[11];
    VARCHAR nom_cher_resp[10];
    VARCHAR n_cher[11];
    VARCHAR nom_cher[10];
    VARCHAR n_specialite[10];
    VARCHAR nom_spec[30];
    int nb_jour_sem;
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()
```

```
{
    /* fonction de selection */
    EXEC SQL WHENEVER SQLERROR GOTO erreur1;
    printf("\n Debut de selection ..... \n");

    EXEC SQL DECLARE c1 CURSOR FOR
    SELECT p.n_projet,p.nom_projet,NVL(p.n_cher_resp,'NULL')
    FROM projet p
    WHERE p.n_equipe=:n_equ_lu;

    EXEC SQL DECLARE c2 CURSOR FOR
    SELECT c.nom_chercheur,s.nom_specialite,NVL(t.nb_jour_sem, 0)
    FROM chercheur c, specialite s, travailler t
    WHERE t.n_projet=:n_proj
    AND t.n_chercheur=c.n_chercheur
```

```

AND c.specialite=s.specialite;

n_equ_lu.len =asks("\n\n Entrez un numero d'equipe ou 'fin' : ",
n_equ_lu.arr);

while ( strcmp(n_equ_lu.arr,"fin"))
{
EXEC SQL WHENEVER NOT FOUND GOTO non_trouve;

EXEC SQL SELECT nom_equipe INTO :nom_equ
FROM equipe WHERE n_equipe=:n_equ_lu;
nom_equ.arr[nom_equ.len]='\0';
printf ("\nNom de l'equipe : %s", nom_equ.arr);

EXEC SQL OPEN c1;
EXEC SQL WHENEVER NOT FOUND GO TO fin_c1;

while (1)
{
EXEC SQL FETCH c1 INTO :n_proj, :nom_proj, :n_cher_resp;
n_proj.arr[n_proj.len]='\0';
nom_proj.arr[nom_proj.len]='\0';
n_cher_resp.arr[n_cher_resp.len]='\0';

/* si le chercheur responsable non null alors le selectionner*/
if (strcmp(n_cher_resp.arr,"NULL"))
{
EXEC SQL select c.nom_chercheur into :nom_cher_resp
from chercheur c
where c.n_chercheur=:n_cher_resp;

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

printf("\n\n\t Nom du projet          Nom du chercheur responsable");
printf("\n\t -----");
printf("\n\t %-20s  %-10s", nom_proj.arr,nom_cher_resp.arr);
}
else
{
printf("\n\n\t Nom du projet          Pas de chercheur responsable");
printf("\n\t -----");
printf("\n\t %-20s          ", nom_proj.arr);
};

EXEC SQL OPEN c2;
EXEC SQL WHENEVER NOT FOUND GOTO fin_c2;

EXEC SQL FETCH c2 INTO :nom_cher, :nom_spec, :nb_jour_sem;
nom_cher.arr[nom_cher.len]='\0';
nom_spec.arr[nom_spec.len]='\0';

printf("\n\n\t\t Nom du chercheur    Nom specialite          Nb
jours semaine");
printf("\n\t\t -----
-----");

while (1)
{
if (nb_jour_sem==0)

```

```

        printf("\n\t\t %-10s          %-30s inconnu", nom_cher.arr,
nom_spec.arr);
    else
        printf("\n\t\t %-10s          %-30s  %-2d", nom_cher.arr,
nom_spec.arr, nb_jour_sem);

    EXEC SQL FETCH c2 INTO :nom_cher,:nom_spec,:nb_jour_sem;
    nom_cher.arr[nom_cher.len]='\0';
    nom_spec.arr[nom_spec.len]='\0';

}

fin_c2:          EXEC SQL CLOSE c2;
                continue;

}
fin_c1:
    EXEC SQL CLOSE c1;
    n_equ_lu.len =asks("\n\n Entrez un numero d'equipe ou 'fin' : ",
n_equ_lu.arr);
    continue;

non_trouve:
    printf("\n equipe inconnue");
    n_equ_lu.len =asks("\n\n Entrez un numero d'equipe ou 'fin' : ",
n_equ_lu.arr);

}
return;

erreurl:
    printf("\nErreur lors de la selection.....");
    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) );
    }
```