

```

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

EXEC SQL BEGIN DECLARE SECTION;
    VARCHAR user[20];
    VARCHAR pwd[20];
    VARCHAR nomproj[20];
    VARCHAR nomequipe[30];
    VARCHAR nomcherresp[10];
    VARCHAR numcherresp[11];
    VARCHAR numproj[8];
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 */
    printf("\n Debut de selection ..... \n");

EXEC SQL WHENEVER SQLERROR GOTO erreurl;
    numproj.len=asks("\n Entrez le numero du projet (ou fin): ", numproj.arr);
    numproj.arr[numproj.len]='\0';
    while (strcmp(numproj.arr,"fin"))
    {
        /* tester l'existence du projet */
EXEC SQL select nom_projet from projet p
            where p.n_projet=:numproj;

        EXEC SQL WHENEVER NOT FOUND GOTO non_trouve;

        /* selectionner les informations recherchees */

EXEC SQL select p.nom_projet, e.nom_equipe,
NVL(p.n_cher_resp, 'NULL')
            into :nomproj, :nomequipe, :numcherresp
            from projet p, equipe e
            where p.n_projet =:numproj
            and p.n_equipe=e.n_equipe;
    }
}

```

```

/* terminer les chaines de caracteres par \0 */
nomproj.arr[nomproj.len]='\0';
nomequipe.arr[nomequipe.len]='\0';
numcherresp.arr[numcherresp.len]='\0';

/* si le chercheur responsable est non null */
if (strcmp(numcherresp.arr,"NULL"))
{
    EXEC SQL WHENEVER NOT FOUND GOTO non_trouve;

    /* selectionner le chercheur responsable */

    EXEC SQL select c.nom_chercheur
        into :nomcherresp
        from chercheur c
        where c.n_chercheur=:numcherresp;

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

    printf("\nNom du projet          Nom de l'equipe          Nom chercheur
responsable");
    printf("\n-----");
    printf("\n %s          %s          %s", nomproj.arr, nomequipe.arr,
nomcherresp.arr);
}

else

{
    printf("\nNom du projet          Nom de l'equipe          pas de
chercheur responsable");
    printf("\n-----");
    printf("\n %s          %s ", nomproj.arr, nomequipe.arr);
};

    numproj.len=asks("\n Entrez le numero du projet (ou fin): ",
numproj.arr);

    continue;

non_trouve:
    printf("\n Ce projet n existe pas !!!\n");
    getchar();
    numproj.len=asks("\n Entrez le numero du projet (ou fin): ",
numproj.arr);
}

printf("\n Fin de selection ..... \n");
return;
erreurl:
    printf("\n erreur a la selection... \n");
    errpt();
    arret();
    exit();
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) );
}

```