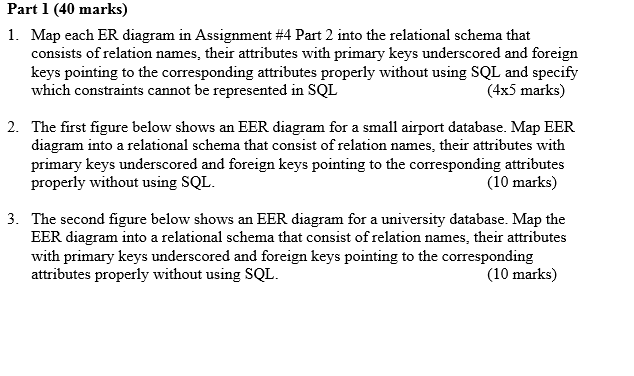
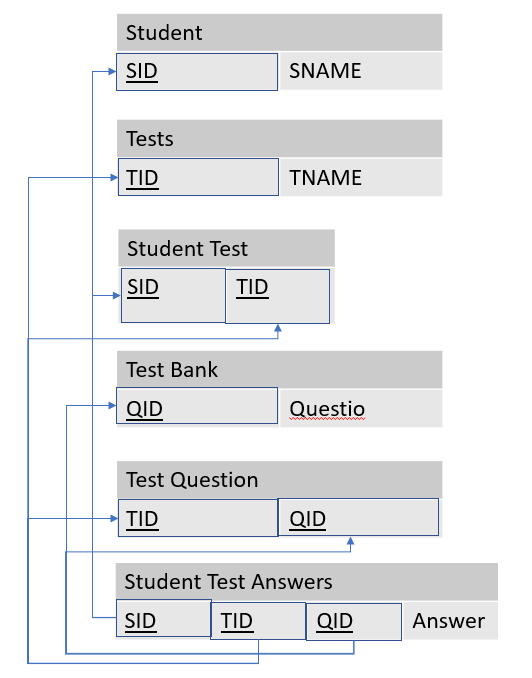
**COMP 3005**

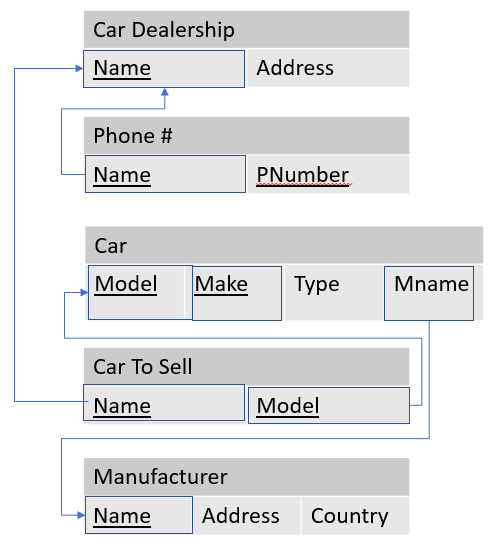
**Krystian Wojcicki**

**101001444**

**A5**

1a) 

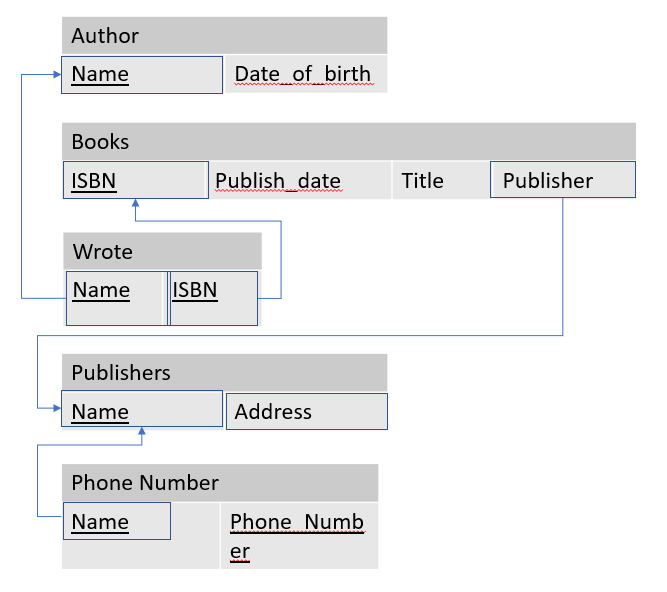
All constraints can be represented in sql.

1b) 

Constraints that cannot be represented in SQL:

Each manufacturer builds atleast one car

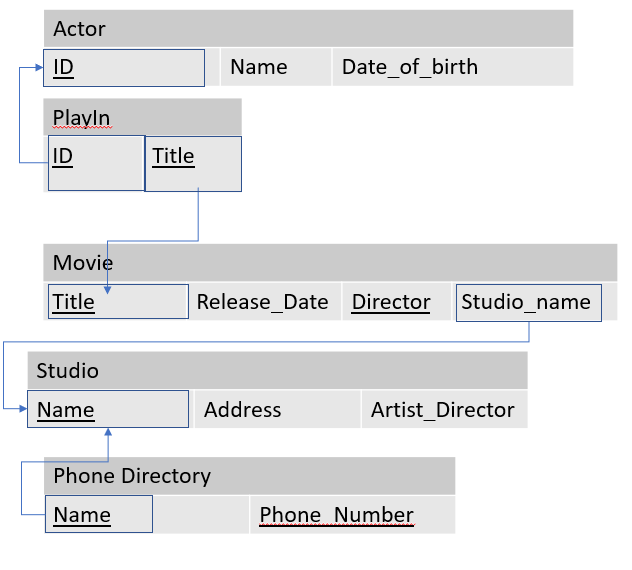
Each dealership sells atleast one car

1c)

Constraints that cannot be represented in SQL:

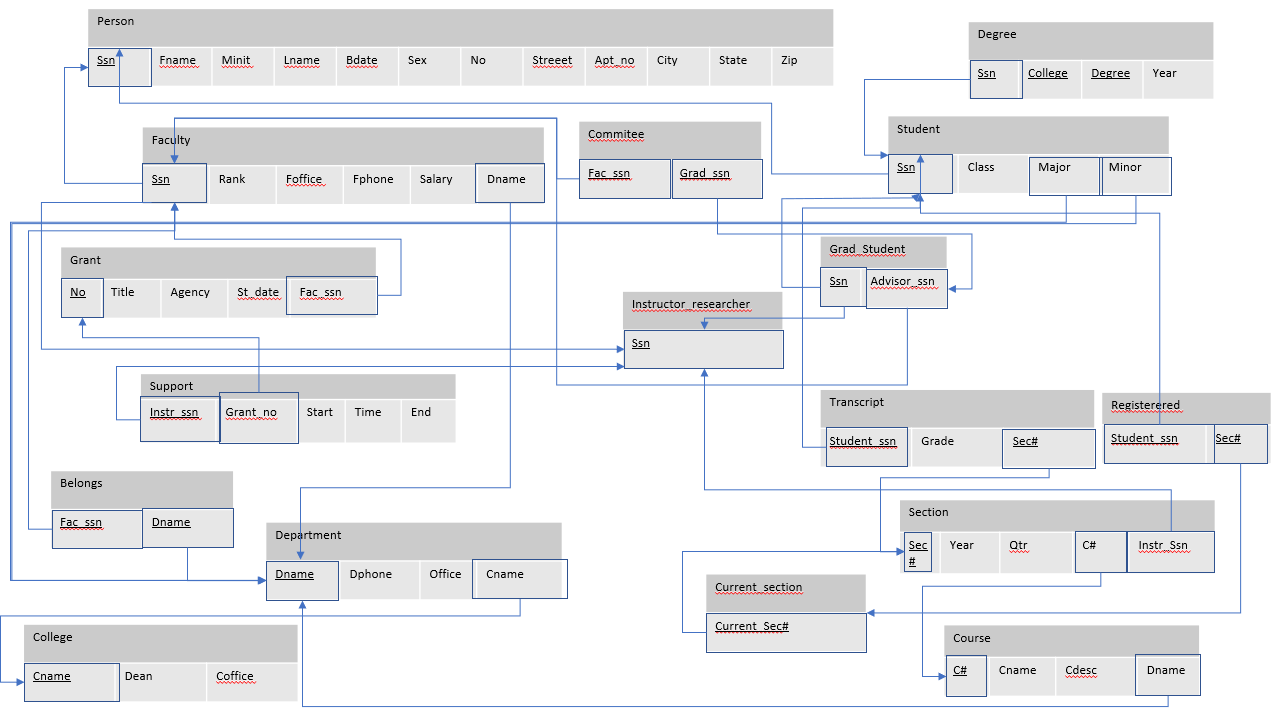
Each author must write atleast one book

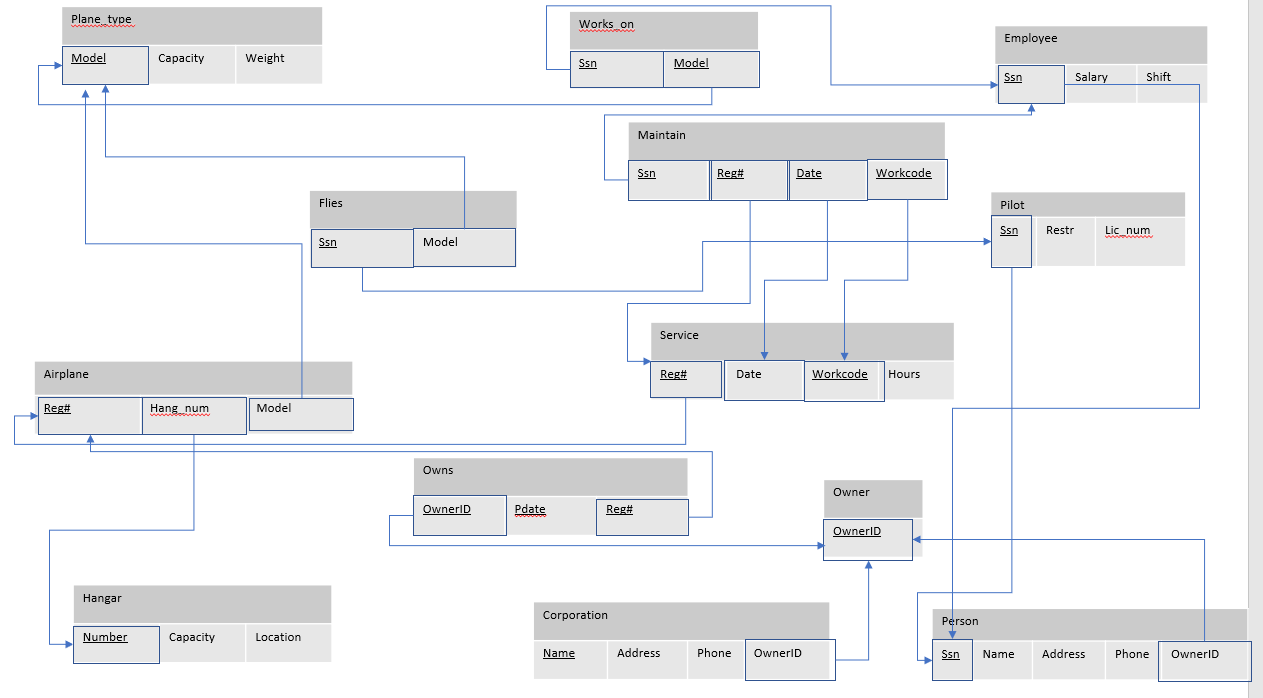
Each book must be written by atleast one author

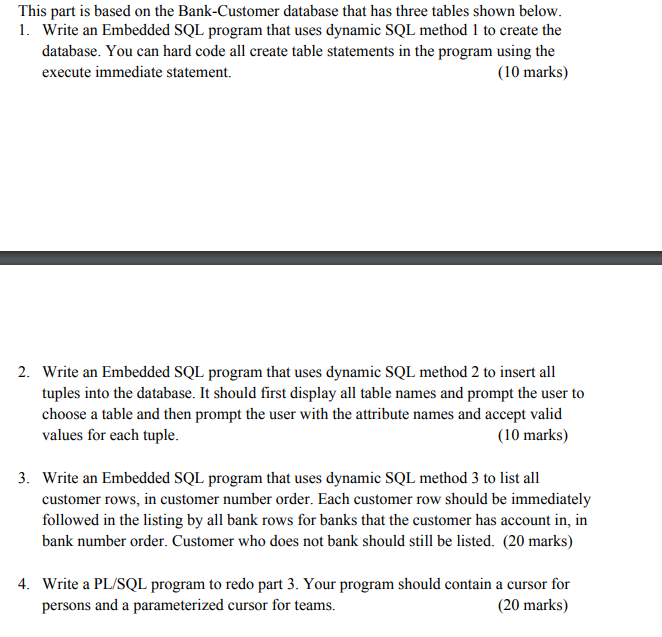
1d) Constraints that cannot be represented in SQL:

Each movie has atleast one actor

Each actor must have played in atleast one movie

2) 

3) 



1)

#include <stdio.h>

int main(){

exec sql include sqlca;

exec sql begin declare section;

char sql1[1024];

char sql2[1024];

char sql3[1024];

char \*LOGIN= "fedora/oracle";

exec sql end declare section;

exec sql whenever sqlerror goto error;

exec sql whenever not found goto finish;

exec sql connect :LOGIN;

printf("Connected to db\n");

printf("Creating tables:\n");

strcpy(sql1, "create table Bank(B# varchar(4), Name varchar(10) NOT NULL, City varchar(10) NOT NULL, primary key (B#))" );

strcpy(sql2, "create table Customer(C# varchar(4), Name varchar(10) NOT NULL, Age int NOT NULL, City varchar(10) NOT NULL, primary key (C#))" );

strcpy(sql3, "create table Account (C# varchar(4), B# varchar(4), Balance int NOT NULL, primary key (C#,B#), foreign key (C#) references Customer(C#) ON DELETE CASCADE, foreign key (B#) references Bank(B#) ON DELETE CASCADE)");

printf("%s\n", sql1);

exec sql execute immediate :sql1;

printf("%s\n", sql2);

exec sql execute immediate :sql2;

printf("%s\n", sql3);

exec sql execute immediate :sql3;

finish:

printf("Created all tables\n");

exit(0);

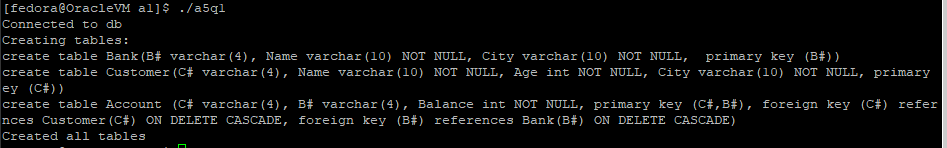
error:

printf("Error found\n");

exec sql rollback release;

exit(1);

}



2)

#include <stdio.h>

int main() {

exec sql include sqlca;

exec sql begin declare section;

char cnum[5];

char name[20];

int age;

char city[20];

char bnum[5];

int balance;

char sql[1024];

char pick[2];

char \*LOGIN= "fedora/oracle";

exec sql end declare section;

exec sql whenever sqlerror goto error;

exec sql whenever not found goto finish;

exec sql connect :LOGIN;

printf("Connected to db\n");

printf("There are 3 tables in this db: \n");

printf("Please enter number 1 (Bank), 2 (Customer) or 3 (Account) to select a table or enter q to quit. \n");

scanf("%s",&pick);

while(strcmp(pick, "q") != 0){

if(strcmp(pick, "1") == 0){

while(strcmp(pick, "q") != 0) {

printf("input B#:\n");

scanf("%s", &bnum);

printf("input Name:\n");

scanf("%s", &name);

printf("input City:\n");

scanf("%s", &city);

strcpy(sql,"insert into Bank values (:v1, :v2, :v3)");

exec sql prepare s from :sql;

exec sql execute s using :bnum, :name, :city;

printf("'q' to quit Bank table else, other key to continue.\n");

scanf("%s",&pick);

}

}

if(strcmp(pick, "2") == 0){

while(strcmp(pick, "q") != 0) {

printf("input C#:\n");

scanf("%s", &cnum);

printf("input Name:\n");

scanf("%s", &name);

printf("input Age:\n");

scanf("%d", &age);

printf("input City:\n");

scanf("%s", &city);

strcpy(sql, "insert into Customer values (:v1, :v2, :v3, :v4)");

exec sql prepare s from :sql;

exec sql execute s using :cnum, :name, :age, :city;

printf("('q' to quit Customer table else, other key to continue.)\n");

scanf("%s",&pick);

}

}

if(strcmp(pick, "3") == 0) {

while(strcmp(pick, "q") != 0) {

printf("input C#:\n");

scanf("%s", &cnum);

printf("input B#:\n");

scanf("%s", &bnum);

printf("input Balance (In integer):\n");

scanf("%d", &balance);

strcpy(sql,"insert into Account values (:v1, :v2, :v3)");

exec sql prepare s from :sql;

exec sql execute s using :cnum, :bnum, :balance;

printf("'q' to quit Bank table else, other key to continue.)\n");

scanf("%s", &pick);

}

}

printf("Please enter number 1 (Bank), 2 (Customer) or 3 (Account) to select a table or enter q to quit. \n");

scanf("%s",&pick);

exec sql execute immediate "commit";

}

finish:

printf("Finished inserting \n");

exit(0);

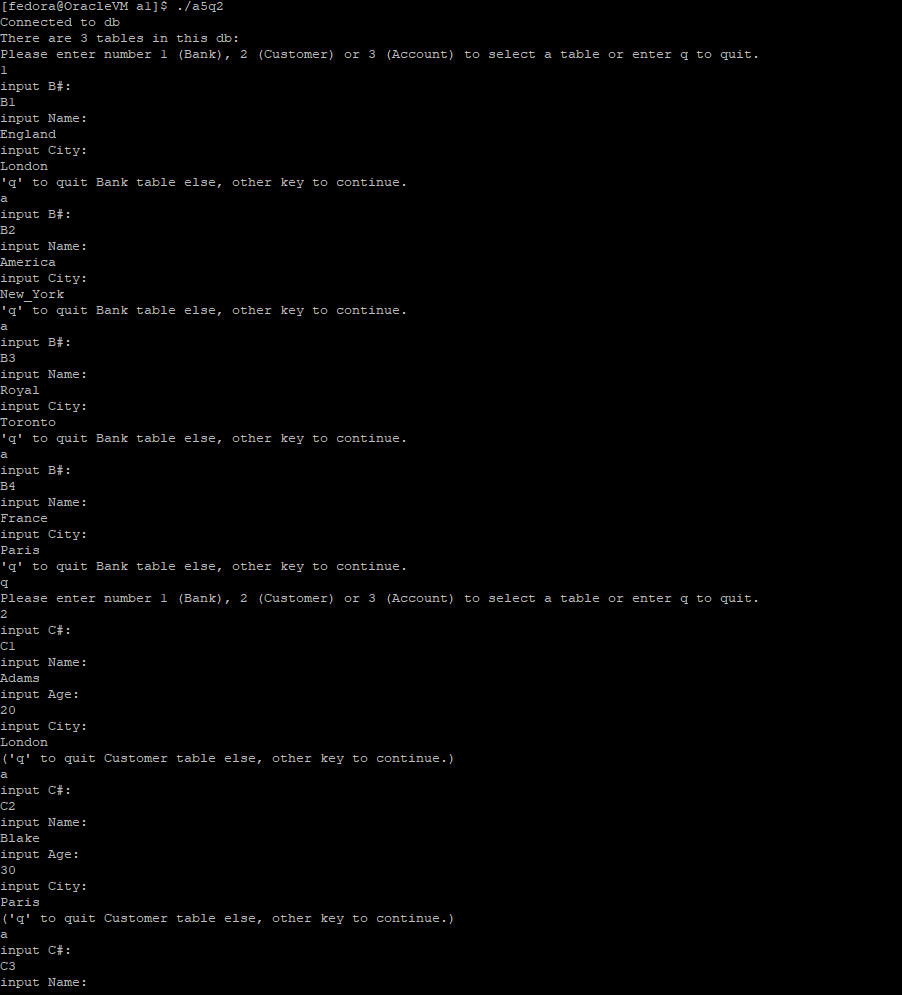
error:

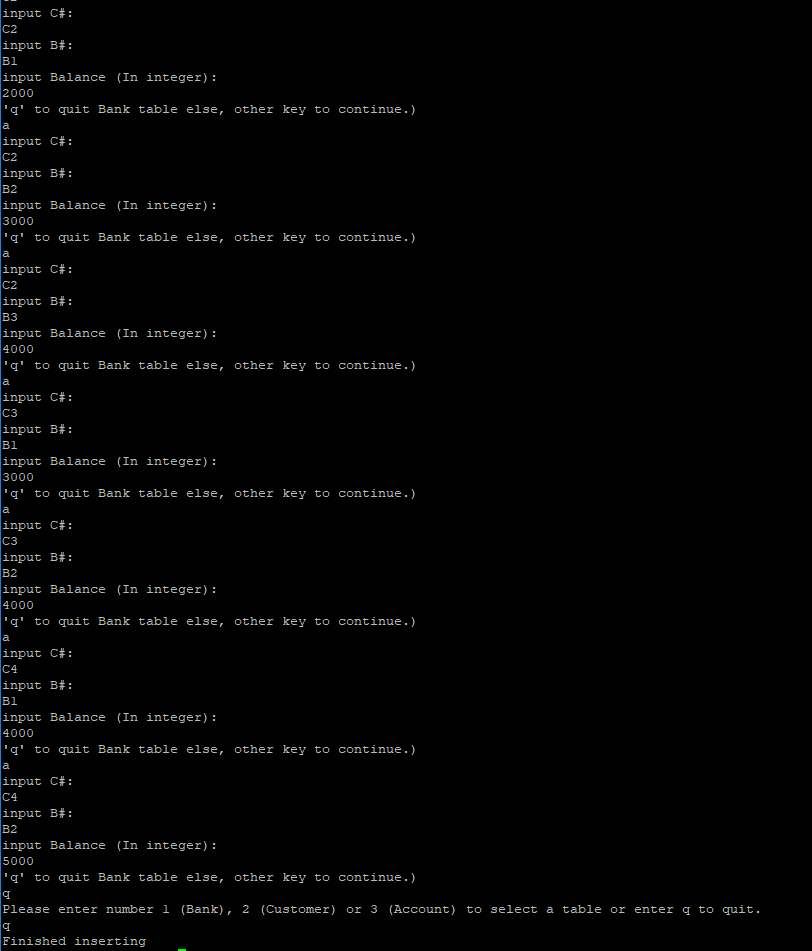
printf("SQL error found \n");

exec sql rollback release;

exit(1);

}





3)

#include <stdio.h>

int main(){

exec sql include sqlca;

exec sql begin declare section;

char sql1[1024];

char sql2[1024];

char cnum[3];

char cname[10];

int age;

char ccity[10];

char bnum[3];

char bname[10];

char bcity[10];

int balance;

char \*LOGIN= "fedora/oracle";

exec sql end declare section;

exec sql whenever sqlerror goto error;

exec sql whenever not found do break;

exec sql connect :LOGIN;

printf("Connected to db\n");

strcpy(sql1, "select \* from customer order by C#");

exec sql prepare c from :sql1;

exec sql declare c\_cursor cursor for c;

strcpy(sql2, "select \* from Bank where exists (select \* from account A where A.C#=:cnum and A.B#=Bank.B#) order by Bank.B#");

exec sql prepare b from :sql2;

exec sql declare b\_cursor cursor for b;

exec sql open c\_cursor;

printf(" C# | Name | Age | City |\n");

printf("+----------------------------------------------------+\n");

while(1) {

exec sql fetch c\_cursor into :cnum, :cname, :age, :ccity;

printf("| %s | %s | %d | %s |\n", cnum, cname, age, ccity);

exec sql open b\_cursor using :cnum;

printf("+----------------------------------------------------+\n");

printf("| +-----------------------------------------------+ |\n");

printf("| | B# | Name | City | |\n");

printf("| +-----------------------------------------------+ |\n");

while(1){

exec sql fetch b\_cursor into :bnum, :bname, :bcity;

printf("| | %s | %s | %s | |\n", bnum, bname, bcity);

printf("| +-----------------------------------------------+ |\n");

}

printf("+----------------------------------------------------+\n");

exec sql close b\_cursor;

}

exec sql close c\_cursor;

exit(0);

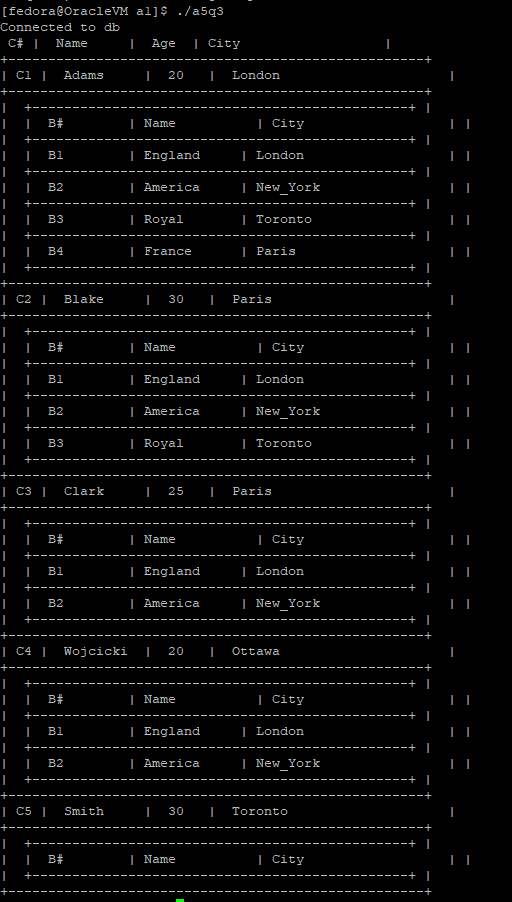
error:

printf("Error found in SQL. \n");

exec sql rollback release;

exit(1);

}



4)

DECLARE

b\_tuple Bank%rowtype;

c\_tuple Customer%rowtype;

CURSOR c\_cursor IS select \* from Customer ORDER BY C# ASC;

CURSOR b\_cursor(cnum Customer.c#%type)

IS select B.\*

FROM Bank B, Account A

WHERE B.B# = A.B#

AND A.C# = cnum

ORDER BY B.B# ASC;

BEGIN

dbms\_output.put\_line(' Customer ');

dbms\_output.put\_line('C# |Name |Age |City ');

dbms\_output.put\_line('+----------------------------------------------------+');

OPEN c\_cursor;

LOOP

fetch c\_cursor into c\_tuple;

exit when c\_cursor%NOTFOUND;

dbms\_output.put\_line('|' || rpad(to\_char(c\_tuple.c#),10) ||

'|' || rpad(c\_tuple.Name,15) ||

'|' || rpad(c\_tuple.Age,10) ||

'|' || rpad(c\_tuple.City,14) ||

'|');

dbms\_output.put\_line('+----------------------------------------------------+');

dbms\_output.put\_line('| +-----------------------------------------------+ |');

dbms\_output.put\_line('| |B# |Name |City | |');

dbms\_output.put\_line('| +-----------------------------------------------+ |');

OPEN b\_cursor(c\_tuple.c#);

LOOP

FETCH b\_cursor into b\_tuple;

exit when b\_cursor%NOTFOUND;

dbms\_output.put\_line('| |' ||rpad(to\_char(b\_tuple.b#),10) ||

'|' || rpad(b\_tuple.Name,15) ||

'|' || rpad(b\_tuple.City,10) ||

' | |' );

END LOOP;

CLOSE b\_cursor;

dbms\_output.put\_line('| +-----------------------------------------------+ |');

dbms\_output.put\_line('+----------------------------------------------------+');

END LOOP;

CLOSE c\_cursor;

END;

