Project 3_1 (55 points)

In this project you will implement the relational database DBActors using **SQL** from DBMS DB2.

1. Create all tables below. Use the information from the Project 2 concerning the data type for the attributes in tables below. Your tables must have the same name, same schema as presented below, and data types as in Project 2.
2. Insert data in your tables (same data as presented in examples below, with one difference: you will replace MK by your initials). Be aware that sometimes I used **Cast()** function to display a tuple in row with 80 characters. Hence, when you define the tables you must use the data type presented in Project 2 not the one displayed in examples.
3. Write the corresponding queries for the questions a to g.

1 and 2: Tables list

PLACE (PLACE_NO, place_town, place_country)
PK (PLACE_NO)

PLACE_NO	PLACE_TOWN	PLACE_COUNTRY
1	TORONTO	CANADA
2	Mississauga	CANADA
3	HAMILTON	CANADA
4	NEW YORK	USA
5	PARIS	FRANCE

PLAY (P_NO, p_TITLE, p_DESCR p_year smallint, p_date_P)
PK (P_NO)

P_NO	P_TITLE	P_DESCR	P_YEAR	P_DATE_P
1	MK_PLAY_1	MK_DESCR_1	1990	02/12/1991
2	MK_PLAY_2	MK_DESCR_2	1850	02/11/1859
3	MK_PLAY_3	MK_DESCR_3	1900	11/08/1901
4	MK_Comedy_4	MK_DESCR_4	1900	05/07/1913
5	MK_Tragedy_5	MK_DESCR_5	1800	02/09/1810
6	MK_Tragedy_6	MK_DESCR_6	2000	02/09/2001

THEATER_COMP_Place (tCOMP_NO, tCOMP_name, tCOMP_date, tComp_place)
PK (tComp_No)
FK (tComp_Place) references Place(place_No)

TCOMP_NO	TCOMP_NAME	TCOMP_DATE	TCOMP_PLACE
1	MK_Comp_1	02/11/2000	1
2	MK_Comp_2	12/01/1997	5
4	MK_Comp_4	02/04/2010	5
5	MK_Comp_5	05/05/2005	1

PERFORMER (Perf_NO, performer_start)
PK (Perf_no)
FK(perf_NO) references Actor_Born_Lives(A_no)

PERF_NO	PERFORMER_START
3	01/01/1995
5	01/01/1980
4	01/01/2000

Actor_Born_Lives (A_NO, A_NAME, A_BORN, A_DIED, A_B_place, A_L_place)
PK (A_NO)
FK (A_B_place) references Place(place_No)
FK (A_L_place) references Place(place_No)

A_NO	A_NAME	A_BORN	A_DIED	A_B_PLACE	A_L_PLACE
1	MK_actor_1	05/12/1980	-	1	1
2	MK_actor_2	05/03/1977	-	1	5
3	MK_actor_3	05/11/1965	-	2	2
4	MK_actor_4	05/09/1970	-	3	4
5	MK_actor_5	02/20/1956	-	5	4
6	MK_actor_6	05/03/1940	05/05/2010	1	4
7	MK_actor_7	07/11/1941	12/09/2012	4	5
8	MK_actor_8	05/12/1980	-	4	1

t_Director (TD_NO, td_start)
PK (TD_NO)
FK(td_NO) references Actor_Born_Lives(A_no)

TD_NO	TD_START
1	01/01/2005
2	01/01/2000

PERFORMS (Perf_NO, play_no, tcomp_no, pdate)
PK (perf_no, play_no, tcomp_no, pdate)
FK(perf_no) references PERFORMER_2015(perf_no)
FK(play_no) references PLAY(p_no)
FK(tcomp_no) references theater_Comp_Place(tComp_no)

PERF_NO	PLAY_NO	TCOMP_NO	PDATE
3	1	1	01/01/2003
3	1	2	01/01/2003
4	2	4	01/01/1998
4	2	4	01/01/2011
5	5	5	01/01/2006

DRAMA (d_p_no, d_type, d_main_n, d_main_p)
PK(d_p_no)
FK (d_p_no) references play (p_no)

D_P_NO	D_TYPE	D_MAIN_N	D_MAIN_P
1	medieval	MK_Neg_1	MK_Pos_1
2	modern	MK_Neg_2	MK_Pos_2
3	renaissance	MK_Neg_3	MK_Pos_3

COMEDY (c_p_no, c_type, c_main, c_second)
PK(c_p_no)
FC (c_p_no) references play (p_no)

C_P_NO	C_TYPE	C_MAIN	C_SECOND
4	farce	MK_main_4	MK_Secnd_4

TRAGEDY (t_p_no, t_type, t_main, t_second)
PK (t_p_no)
FK (t_p_no) references play (p_no)

T_P_NO	T_TYPE	T_MAIN	T_SECOND
5	Roman	MK_main_5	MK_Secnd_5
6	Greek	MK_main_7	MK_Secnd_7

TH_HAS_DIR (tcomp_no, td_no, hasdate)
PK (tcomp_no, td_no, hasdate)
FK(tcomp_no) references THEATER_COMP_Place (tCOMP_NO)
FK (td_no) references t_Director (TD_NO)

TCOMP_NO	TD_NO	HASDATE
1	2	01/01/2000
1	2	01/01/2006
4	1	01/01/2011
4	1	01/01/2014
5	1	01/01/2005

DRAMATIST (dramatist_no, dramatist_name)
PK(dramatist_no)

DRAMATIST_NO	DRAMATIST_NAME
1	MK_Drmst_1
2	MK_Drmst_2
3	MK_Drmst_3
4	MK_Drmst_4

WRITES (dramatist_no, play_no)
PK (dramatist_no, play_no)
FK (dramatist_no) references DRAMATIST(dramatist_no)
FK (play_no) references play(p_no)

DRAMATIST_NO	PLAY_NO
1	2
1	5
2	3
2	4
3	1
4	6

ACTOR_DRAM (dramatist_no, A_no)
PK(dramatist_no) or PK(A_no)
FK (dramatist_no) references DRAMATIST(dramatist_no)
FK (a_no) references Actor_Born_Lives(A_no)

DRAMATIST_NO	A_NO
4	2

HIRED (A_no, tcomp_no, hire_year)
PK (A_no, tcomp_no, hire_year)
FK (A_no) references Actor_Born_Lives(A_no)
FK (tcomp_no) references THEATER_COMP_Place (tCOMP_N)

A_NO	TCOMP_NO	HIRE_YEAR
1	1	2000
5	2	1997
1	2	1998
7	2	1997
3	2	1997
2	2	1997
4	4	2010
4	4	2012
6	5	2005

3. **Use SQL language to write queries that will obtain the following lists**:

a. List actors that are dramatist. Attributes list: dramatist_no, a_no, dramatist_name, a_name, play_no, p_title

D_NO	A_NO	D_NAME	A_NAME	PLAY_NO	P_TITLE
4	2	MK_Drmst_4	MK_actor_2	6	MK_Tragedy_6

Figure 1. An example of answer table.

b. **List theater companies and actors hired there order by company, actor.** Attributes list: tcomp_no, tcomp_name, tcomp_place, hire_year, a_no, a_name, A_l_place.

TCOMP_NO	TCOMP_NAME	TCOMP_PLACE	HIRE_YEAR	A_NO	A_NAME	A_L_PLACE
1	MK_Comp_1	1	2000	1	MK_actor_1	1
2	MK_Comp_2	5	1998	1	MK_actor_1	1
2	MK_Comp_2	5	1997	2	MK_actor_2	5
2	MK_Comp_2	5	1997	3	MK_actor_3	2
2	MK_Comp_2	5	1997	5	MK_actor_5	4
2	MK_Comp_2	5	1997	7	MK_actor_7	5
4	MK_Comp_4	5	2010	4	MK_actor_4	4
4	MK_Comp_4	5	2012	4	MK_actor_4	4
5	MK_Comp_5	1	2005	6	MK_actor_6	4

Figure 2. An example of answer table.

c. **List how many comedies, dramas, and tragedies are stored in the database.** Attributes list: Nr_comedies, Nr_dramas, Nr_tragedies

NR_COMEDIES	NR_DRAMAS	NR_TRAGEDIES
1	3	2

Figure 3. An example of answer table

d. **List the plays in the database.** Attributes list p_no, p_title, p_year, type (play type, see figure 4).

P_NO	P_TITLE	P_YEAR	TYPE
5	MK_Tragedy_5	1800	Roman
2	MK_PLAY_2	1850	modern
3	MK_PLAY_3	1900	renaissance
4	MK_Comedy_4	1900	farce
1	MK_PLAY_1	1990	medieval
6	MK_Tragedy_6	2000	Greek

Figure 4. An example of answer table

e. **List the actors in the database.** Attributes list: a_no, a_name, a_born, town of birth, town where actor lives. See in Figure 5 the names to be used in table's header.

A_NO	A_NAME	A_BORN	BORN IN:	LIVES IN:
1	MK_actor_1	05/12/1980	TORONTO	TORONTO
2	MK_actor_2	05/03/1977	TORONTO	PARIS
3	MK_actor_3	05/11/1965	Mississauga	Mississauga
4	MK_actor_4	05/09/1970	HAMILTON	NEW YORK
5	MK_actor_5	02/20/1956	PARIS	NEW YORK
6	MK_actor_6	05/03/1940	TORONTO	NEW YORK
7	MK_actor_7	07/11/1941	NEW YORK	PARIS
8	MK_actor_8	05/12/1980	NEW YORK	TORONTO

Figure 5. An example of answer table

f. **List all plays performed.** Attributes list: a_name, p_title, t_comp_name, pdate.

PERF_NO	A_NAME	TCOMP_NAME	P_TITLE	PDATE
3	MK_actor_3	MK_Comp_1	MK_PLAY_1	01/01/2003
3	MK_actor_3	MK_Comp_2	MK_PLAY_1	01/01/2003
4	MK_actor_4	MK_Comp_4	MK_PLAY_2	01/01/1998
4	MK_actor_4	MK_Comp_4	MK_PLAY_2	01/01/2011
5	MK_actor_5	MK_Comp_5	MK_Tragedy_5	01/01/2006

Figure 6. An example of answer table

g. **List the plays written after 1900.** Attributes list: p_no, p_title, p_year, type (play type).

P_NO	P_TITLE	P_YEAR	TYPE
1	MK_PLAY_1	1990	medieval
3	MK_PLAY_3	1900	renaissance
4	MK_Comedy_4	1900	farce
6	MK_Tragedy_6	2000	Greek

Figure 7. An example of answer table

h. **List all theater companies, director name, date when the director was there, order by company, actor, start date.** Attributes list: tcomp_name, a_name, hasdate.

TCOMP_NAME	A_NAME	Start:
MK_Comp_1	MK_actor_2	01/01/2000
MK_Comp_1	MK_actor_2	01/01/2006
MK_Comp_4	MK_actor_1	01/01/2011
MK_Comp_4	MK_actor_1	01/01/2014
MK_Comp_5	MK_actor_1	01/01/2005

Figure 8. An example of answer table

What to upload for Project 3_1 two files: P3_1_1and2.txt and P3_1_3.txt.

File P3_1_1and2.txt. In this file you present the SQL commands to create all tables listed in 1 and 2, and to insert data in your tables as presented in 1 and 2. For each table, after you insert all data in it present also a **SELECT** for all attributes (use if necessary **cast()** function to display the data from a row only on 80 characters).

File P3_1_3.txt In this file you present the SQL queries to obtain the lists a to g. For each query you have to present the query and the result of it. Hence, you have to use also cast() function in your SELECT such as the query result will look as in examples presented in 3.

You may use the commands:
update command options using v ON
update command options using z on P3_1_1and2.txt
or
update command options using z on P3_1_3.txt
to obtain the files requested for the project.