

Dbms LAB6 CMDS

-V Venkataraman 106118106

Q1

USE MYSAMPLDB;

-- 1

```
create table Movie(mID int, title varchar(100), year int, director varchar(100));
create table Reviewer(rID int, name varchar(100));
create table Rating(rID int, mID int, stars int, ratingDate date);
alter table Movie add primary key (mID);
-- alter table Movie drop primary key;
alter table Movie add primary key (title,year);
alter table Reviewer add primary key (rID);
alter table Rating change stars stars int NOT NULL;
alter table Rating change stars stars int NOT NULL check(stars in (1, 2, 3, 4, 5));
alter table Rating change ratingDate ratingDate date check(ratingDate >=
"00-00-2000");
alter table Movie add constraint CHECK(YEAR>1900 AND CASE WHEN
DIRECTOR="STEVEN SPIELBERG" THEN YEAR<1990 WHEN
DIRECTOR="JAMES CAMERON" THEN YEAR>1990 END);
alter table Rating add constraint foreign key(rID) references Reviewer(rID);
alter table Rating add constraint foreign key(mID) references Movie(mID);
desc Movie;
```

Field	Type	Null	Key	Default	Extra
mID	int	YES			
title	varchar(100)	NO	PRI		
year	int	NO	PRI		
director	varchar(100)	YES			

desc Rating;

Field	Type	Null	Key	Default	Extra
rID	int	YES	MUL		
mID	int	YES			
stars	int	NO			
ratingDate	date	YES			

desc Reviewer;

Field	Type	Null	Key	Default	Extra
rID	int	NO	PRI		
name	varchar(100)	YES			

-- 2

```
insert into Movie values(101, 'Gone with the Wind', 1939, 'Victor Fleming');
insert into Movie values(102, 'Star Wars', 1977, 'George Lucas');
insert into Movie values(103, 'The Sound of Music', 1965, 'Robert Wise');
insert into Movie values(104, 'E.T.', 1982, 'Steven Spielberg');
insert into Movie values(105, 'Titanic', 1997, 'James Cameron');
insert into Movie values(106, 'Snow White', 1937, null);
insert into Movie values(107, 'Avatar', 2009, 'James Cameron');
```

```

insert into Movie values(108, 'Raiders of the Lost Ark', 1981, 'Steven Spielberg');
insert into Reviewer values(201, 'Sarah Martinez');
insert into Reviewer values(202, 'Daniel Lewis');
insert into Reviewer values(203, 'Brittany Harris');
insert into Reviewer values(204, 'Mike Anderson');
insert into Reviewer values(205, 'Chris Jackson');
insert into Reviewer values(206, 'Elizabeth Thomas');
insert into Reviewer values(207, 'James Cameron');
insert into Reviewer values(208, 'Ashley White');
insert into Rating values(201, 101, 2, '2011-01-22');
insert into Rating values(201, 101, 4, '2011-01-27');
insert into Rating values(202, 106, 4, null);
insert into Rating values(203, 103, 2, '2011-01-20');
insert into Rating values(203, 108, 4, '2011-01-12');
insert into Rating values(203, 108, 2, '2011-01-30');
insert into Rating values(204, 101, 3, '2011-01-09');
insert into Rating values(205, 103, 3, '2011-01-27');
insert into Rating values(205, 104, 2, '2011-01-22');
insert into Rating values(205, 108, 4, null);
insert into Rating values(206, 107, 3, '2011-01-15');
insert into Rating values(206, 106, 5, '2011-01-19');
insert into Rating values(207, 107, 5, '2011-01-20');
insert into Rating values(208, 104, 3, '2011-01-02');
select * from Movie;

```

mID	title	year	director
107	Avatar	2009	James Cameron
104	E.T.	1982	Steven Spielberg
101	Gone with the Wind	1939	Victor Fleming
108	Raiders of the Lost Ark	1981	Steven Spielberg
106	Snow White	1937	
102	Star Wars	1977	George Lucas
103	The Sound of Music	1965	Robert Wise

```
select * from Reviewer;
```

rID	name
201	Sarah Martinez
202	Daniel Lewis
203	Brittany Harris
204	Mike Anderson
205	Chris Jackson
206	Elizabeth Thomas
207	James Cameron

```
select * from Rating;
```

58

Result Grid

rID	mID	stars	ratingDate
201	101	2	2011-01-22
201	101	4	2011-01-27
202	106	4	
203	103	2	2011-01-20
203	108	4	2011-01-12
203	108	2	2011-01-30
204	101	3	2011-01-09

Rating 13

Output

#	Time	Action	Message	Duration / Fetch
113	17:15:07	select * from Reviewer LIMIT 0, 1000	8 row(s) returned	0.000 sec / 0.000 sec
114	17:15:24	select * from Rating LIMIT 0, 1000	14 row(s) returned	0.000 sec / 0.000 sec

-- 3

insert into Reviewer values(209, null);

-- 4

insert into Rating values(201, 101, null, '2011-01-22');

-- 5

insert into Rating values(201, 101, 7, '2011-01-22');

-- 6

insert into Rating values(201, 101, 7, '1999-01-22');

-- 7

insert into Movie values(1000, 'Titanic', 1889, 'James Cameron');

-- 8

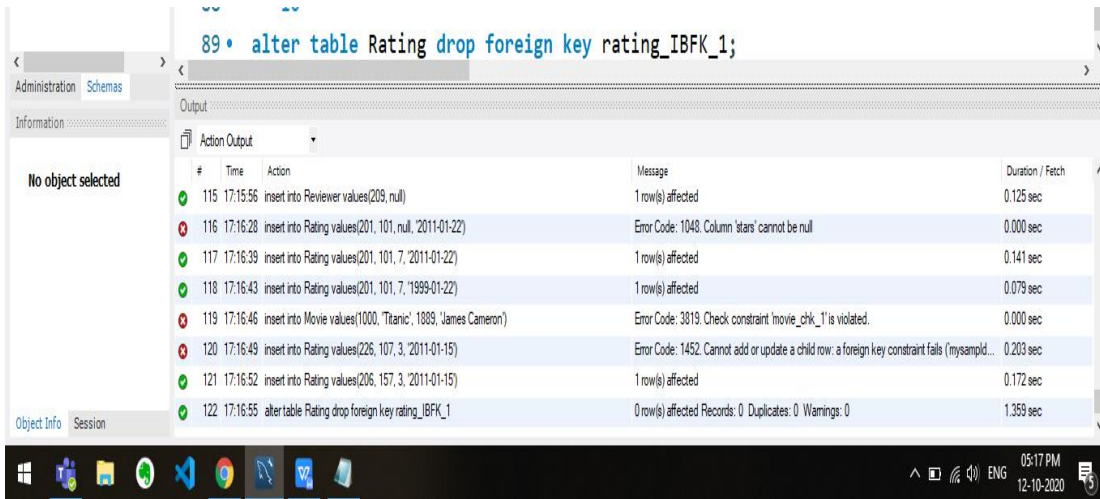
insert into Rating values(226, 107, 3, '2011-01-15');

-- 9

insert into Rating values(206, 157, 3, '2011-01-15');

-- 10

alter table Rating drop foreign key rating_IBFK_1;



Q2

USE MYSAMPLDB;

```
CREATE TABLE BOOK(BID INT PRIMARY KEY,TITLE VARCHAR(100),PUBLISHER
VARCHAR(100));
```

```
CREATE TABLE BOOK_AUTHOR(AID INT PRIMARY KEY,BID INT,AUTHOR_NAME
VARCHAR(100),FOREIGN KEY (BID) REFERENCES BOOK(BID));
```

```
CREATE TABLE BORROWER(CARDNO INT PRIMARY KEY,NAME VARCHAR(100),ADDRESS
VARCHAR(100),PHONE INT);
```

```
CREATE TABLE BOOK_LOANS(TRANSID INT PRIMARY KEY,BID INT,CARDNO
INT,DATE_OUT DATE,DUE_DATE DATE);
```

```
INSERT INTO BOOK VALUES (100,'MATHS','SHARMA');
INSERT INTO BOOK VALUES (101,'ENGLISH','ALICE');
INSERT INTO BOOK VALUES (102,'SCIENCE','RAJU');
INSERT INTO BOOK VALUES (103,'SOCIAL','NEEL');
INSERT INTO BOOK VALUES (104,'HINDI','RAM');
```

INSERT INTO BOOK VALUES (105,'TELUGU','RAMHARAN');

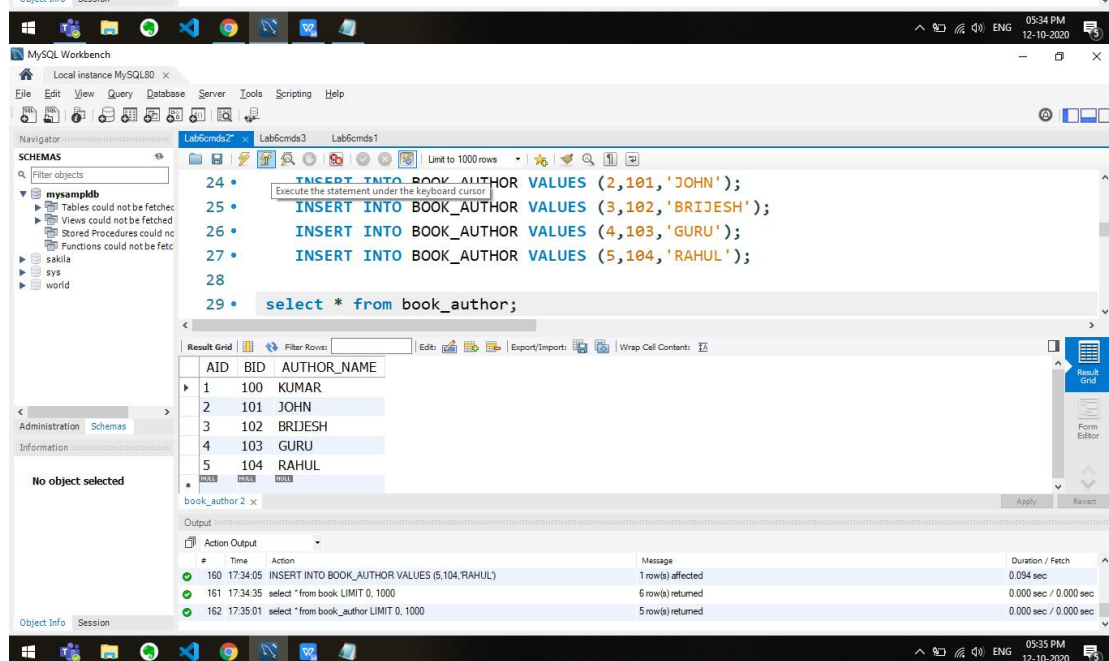
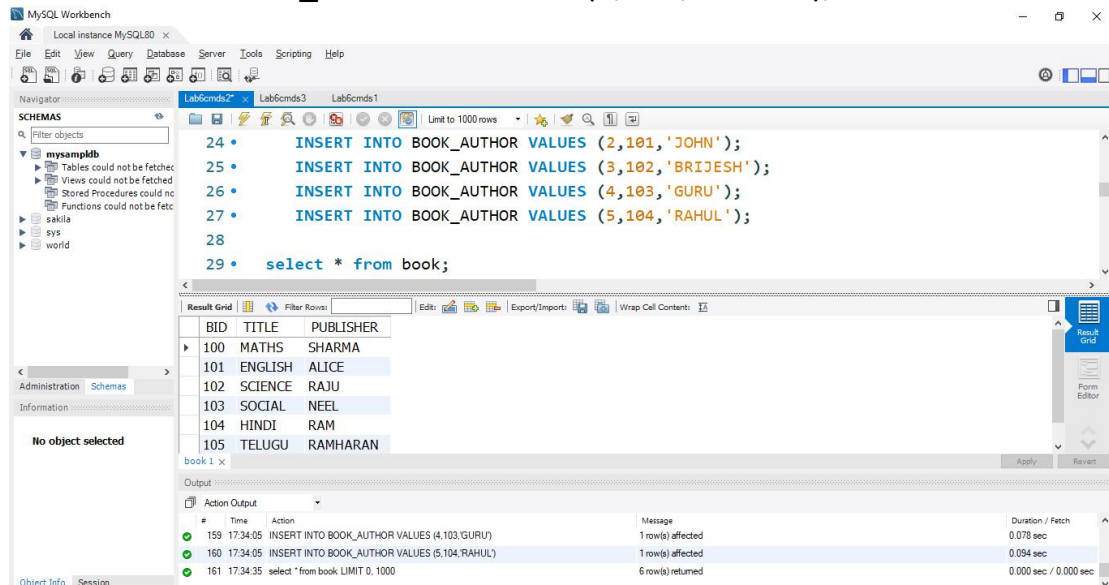
INSERT INTO BOOK_AUTHOR VALUES (1,100,'KUMAR');

INSERT INTO BOOK_AUTHOR VALUES (2,101,'JOHN');

INSERT INTO BOOK_AUTHOR VALUES (3,102,'BRIJESH');

INSERT INTO BOOK_AUTHOR VALUES (4,103,'GURU');

INSERT INTO BOOK_AUTHOR VALUES (5,104,'RAHUL');



-- 3

UPDATE BOOK SET BID=6 WHERE BID=5;

-- 4

ALTER TABLE BOOKAUTHOR DROP FOREIGN KEY BOOKAUTHOR_ibfk_1;

-- 5

```
ALTER TABLE BOOKAUTHOR ADD FOREIGN KEY (BID) REFERENCES BOOK(BID) ON UPDATE  
CASCADE;
```

-- 6

```
UPDATE BOOK SET BID=6 WHERE BID=5;
```

-- 7

```
DELETE FROM BOOK WHERE BID=6;
```

-- 8

```
ALTER TABLE BOOKAUTHOR ADD FOREIGN KEY (BID) REFERENCES BOOK(BID) ON DELETE  
CASCADE;
```

-- 9

```
DELETE FROM BOOK WHERE BID=6;
```

-- 10

```
ALTER TABLE BOOKLOAN ADD FOREIGN KEY (BID) REFERENCES BOOK(BID) ON UPDATE  
RESTRICT ON DELETE SET NULL;
```

-- 11

```
insert into Borrower values ( 51 , "Ayush" , "Chennai" , 109823) , ( 52 , "Pinky" , "Madurai" ,  
903951) , ( 53 , "Harshit" , "Mumbai" , 980980) , ( 54 , "Nitin" , "Duabii" , 12323) , (55 ,  
"Priyanka" , "Bangalore" , 99819);
```

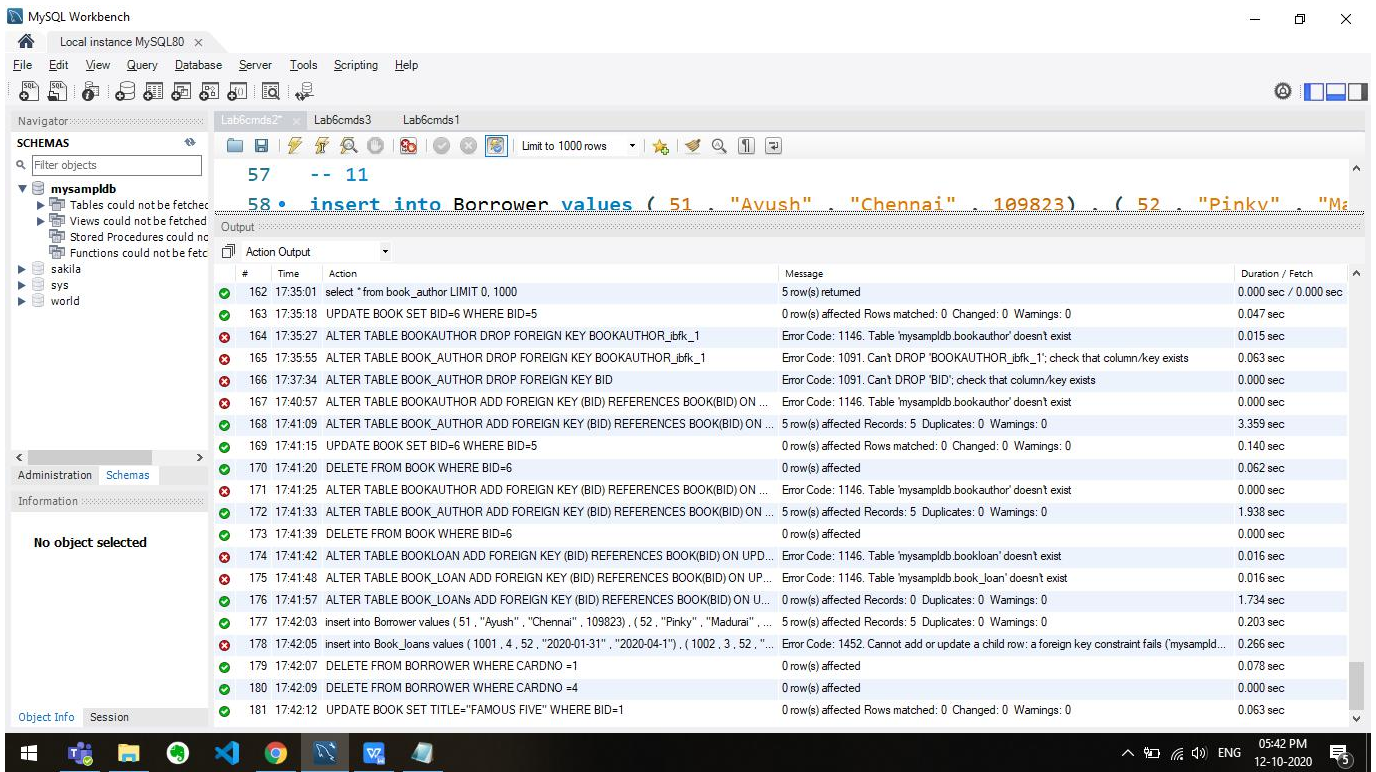
```
insert into Book_loans values ( 1001 , 4 , 52 , "2020-01-31" , "2020-04-1" ) , ( 1002 , 3 , 52 ,  
"2020-01-03" , "2020-08-01" ) , ( 1003 , 10 , 53 , "2020-09-01" , "2020-12-12" ) , ( 1004 , 5 ,  
54 , "2020-09-01" , "2020-12-12" );
```

-- 12

```
DELETE FROM BORROWER WHERE CARDNO =1;  
DELETE FROM BORROWER WHERE CARDNO =4;
```

-- 13

```
UPDATE BOOK SET TITLE="FAMOUS FIVE" WHERE BID=1
```

Q3

USE MYSAMPLDB;
 DROP TABLE IF EXISTS OFFSITESECTION;
 DROP TABLE IF EXISTS ENROLLEDIN;
 DROP TABLE IF EXISTS SECTION;
 DROP TABLE IF EXISTS STUDENT;
 DROP TABLE IF EXISTS PROFESSOR;
 DROP TABLE IF EXISTS COURSE;

CREATE TABLE COURSE

(

COURSENUM INT,
 COURSENAME VARCHAR(100),
 PRIMARY KEY(COURSENUM)

```
);  
CREATE TABLE PROFESSOR  
(  
    PROFNUM INT,  
    PROFNAME VARCHAR(50),  
    PRIMARY KEY(PROFNUM)  
);
```

```
CREATE TABLE STUDENT  
(  
    STUDENTNUM INT,  
    STUDENTNAME VARCHAR(50),  
    GPA INT,  
    PRIMARY KEY(STUDENTNUM)  
);
```

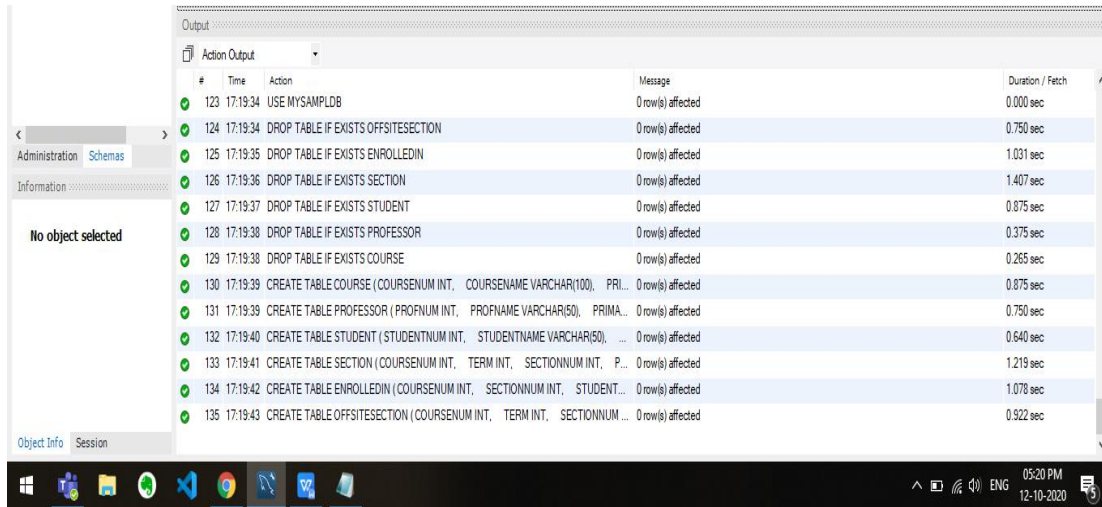
```
CREATE TABLE SECTION  
(  
    COURSENUM INT,  
    TERM INT,  
    SECTIONNUM INT,  
    PROFNUM INT,  
    PRIMARY KEY(COURSENUM,TERM,SECTIONNUM),  
    FOREIGN KEY(COURSENUM) REFERENCES COURSE(COURSENUM),  
    FOREIGN KEY(PROFNUM) REFERENCES PROFESSOR(PROFNUM)  
);
```

```
CREATE TABLE ENROLLEDIN  
(  
    COURSENUM INT,  
    SECTIONNUM INT,  
    STUDENTNUM INT,  
    TERM INT,  
    PRIMARY KEY(COURSENUM,SECTIONNUM,STUDENTNUM,TERM),  
    FOREIGN KEY(COURSENUM) REFERENCES SECTION(COURSENUM),  
    MARK INT  
  
);
```

```
CREATE TABLE OFFSITESECTION  
(  
    COURSENUM INT,  
    TERM INT,  
    SECTIONNUM INT,
```


PRIMARY KEY(COURSENUM,TERM,SECTIONNUM),
FOREIGN KEY(COURSENUM) REFERENCES SECTION(COURSENUM),
LOCATION VARCHAR(100)

);



#	Time	Action	Message	Duration / Fetch
123	17:19:34	USE MYSAMPLDB	0 row(s) affected	0.000 sec
124	17:19:34	DROP TABLE IF EXISTS OFFSITESECTION	0 row(s) affected	0.750 sec
125	17:19:35	DROP TABLE IF EXISTS ENROLLEDIN	0 row(s) affected	1.031 sec
126	17:19:36	DROP TABLE IF EXISTS SECTION	0 row(s) affected	1.407 sec
127	17:19:37	DROP TABLE IF EXISTS STUDENT	0 row(s) affected	0.875 sec
128	17:19:38	DROP TABLE IF EXISTS PROFESSOR	0 row(s) affected	0.375 sec
129	17:19:38	DROP TABLE IF EXISTS COURSE	0 row(s) affected	0.265 sec
130	17:19:39	CREATE TABLE COURSE (COURSENUM INT, COURSENAME VARCHAR(100), PRI...	0 row(s) affected	0.875 sec
131	17:19:39	CREATE TABLE PROFESSOR (PROFNUM INT, PROFNAME VARCHAR(50), PRIMA...	0 row(s) affected	0.750 sec
132	17:19:40	CREATE TABLE STUDENT (STUDENTNUM INT, STUDENTNAME VARCHAR(50), ...	0 row(s) affected	0.640 sec
133	17:19:41	CREATE TABLE SECTION (COURSENUM INT, TERM INT, SECTIONNUM INT, P...	0 row(s) affected	1.219 sec
134	17:19:42	CREATE TABLE ENROLLEDIN (COURSENUM INT, SECTIONNUM INT, STUDENT...	0 row(s) affected	1.078 sec
135	17:19:43	CREATE TABLE OFFSITESECTION (COURSENUM INT, TERM INT, SECTIONNUM ...	0 row(s) affected	0.922 sec