LAB ASSIGNMENT
ADVANCED DATABASE MANAGEMENT SYSTEMS LAB
TKM COLLEGE OF ENGINEERING

GROUP 1

1.Consider the following schema for a LibraryDatabase:

BOOK (Book_id, Title, Publisher_Name, Pub_Year)

BOOK_AUTHORS (Book_id, Author_Name)

PUBLISHER (Name, Address, Phone)

BOOK_COPIES (Book_id, Branch_id, No-of_Copies)

BOOK_LENDING (Book_id, Branch_id, Card_No, Date_Out, Due_Date)

LIBRARY_BRANCH (Branch_id, Branch_Name, Address)

Write SQL queries to

1. Retrieve details of all books in the library – id, title, name of publisher, authors, number of copies in each branch,etc.

2. Get the particulars of borrowers who have borrowed more than 3 books, but from Jan 2017 to Jun2017

3. Delete a book in BOOK table. Update the contents of other tables to reflect this data manipulationoperation.

4. Partition the BOOK table based on year of publication. Demonstrate its working with a simplequery.

5. Create a view of all books and its number of copies that are currently available in the Library.

SCRIPT

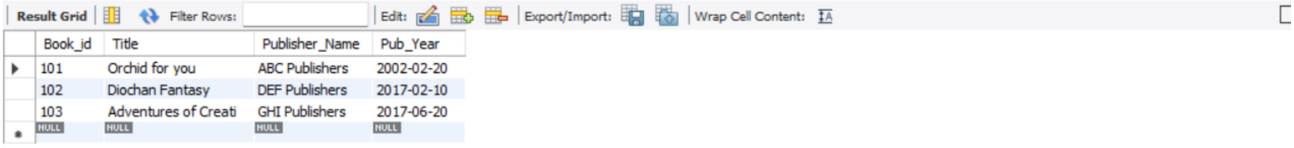
```
CREATE DATABASE LIBRARYDATABASE;

USE LIBRARYDATABASE;

CREATE TABLE BOOK (Book_id INT PRIMARY KEY,Title VARCHAR(20),Publisher_Name
VARCHAR(20),Pub_Year DATE);

CREATE TABLE BOOK_AUTHORS(Book_id INT,Author_Name VARCHAR(20) PRIMARY
KEY, FOREIGN KEY(Book_id) REFERENCES BOOK(Book_id) ON DELETE CASCADE);

CREATE TABLE PUBLISHER(Publisher_Name VARCHAR(20) PRIMARY KEY,Address
VARCHAR(20),Phone BIGINT);

CREATE TABLE LIBRARY_BRANCH(Branch_id INT PRIMARY KEY,Branch_Name
VARCHAR(20),Address VARCHAR(20));
```

```
CREATE TABLE BOOK_COPIES(Book_id INT,Branch_id INT,No_of_Copies INT,FOREIGN KEY(Book_id) REFERENCES BOOK(Book_id) ON DELETE CASCADE,FOREIGN KEY(Branch_id) REFERENCES LIBRARY_BRANCH(Branch_id) ON DELETE CASCADE);

CREATE TABLE BOOK_LENDING(Book_id INT,Branch_id INT,Card_No INT,Date_Out Date,Due_date DATE,FOREIGN KEY(Book_id) REFERENCES BOOK(Book_id) ON DELETE CASCADE,FOREIGN KEY(Branch_id) REFERENCES LIBRARY_BRANCH(Branch_id) ON DELETE CASCADE);

INSERT INTO BOOK(Book_id,Title,Publisher_Name,Pub_year) VALUES (101,'Orchid for you','ABC Publishers','2002-02-20');

INSERT INTO BOOK(Book_id,Title,Publisher_Name,Pub_year) VALUES (102,'Diochan Fantasy','DEF Publishers','2017-02-10');

INSERT INTO BOOK(Book_id,Title,Publisher_Name,Pub_year) VALUES (103,'Adventures of Creation','GHI Publishers','2017-06-20');

INSERT INTO BOOK(Book_id,Title,Publisher_Name,Pub_year) VALUES (104,'Never leave you','JKL Publishers','2020-02-20');

INSERT INTO BOOK_AUTHORS(Book_id,Author_Name) VALUES (101,'AMEEN KIRAN');

INSERT INTO BOOK_AUTHORS(Book_id,Author_Name) VALUES (102,'SHARAVAN S');

INSERT INTO BOOK_AUTHORS(Book_id,Author_Name) VALUES (103,'SREE PADHMARAJAN');

INSERT INTO BOOK_AUTHORS(Book_id,Author_Name) VALUES (104,'LALJIHADR');

INSERT INTO PUBLISHER(Publisher_Name,Address,Phone) VALUES ('ABC Publishers','123 LANE BHORIGARD',9878758810);

INSERT INTO PUBLISHER(Publisher_Name,Address,Phone) VALUES ('DEF Publishers','456 LANE BHORIGARD',9093943900);

INSERT INTO PUBLISHER(Publisher_Name,Address,Phone) VALUES ('GHI Publishers','789 LANE BHORIGARD',8948939899);

INSERT INTO PUBLISHER(Publisher_Name,Address,Phone) VALUES ('JKL Publishers','453 LANE BHORIGARD',9849898999);

INSERT INTO LIBRARY_BRANCH(Branch_id,Branch_Name,Address) VALUES (10,'QUILION LIB','200 LANE Quiron');

INSERT INTO LIBRARY_BRANCH(Branch_id,Branch_Name,Address) VALUES (20,'STANDFORD LIB','300 LANE Standon');

INSERT INTO LIBRARY_BRANCH(Branch_id,Branch_Name,Address) VALUES (30,'PURKUSH LIB','200 LANE Purshk');

INSERT INTO LIBRARY_BRANCH(Branch_id,Branch_Name,Address) VALUES (40,'LAKFORD LIB','200 LANE Lateford');

INSERT INTO BOOK_COPIES(Book_id,Branch_id,No_of_Copies) VALUES (101,10,30);

INSERT INTO BOOK_COPIES(Book_id,Branch_id,No_of_Copies) VALUES (101,20,40);

INSERT INTO BOOK_COPIES(Book_id,Branch_id,No_of_Copies) VALUES (102,10,30);
INSERT INTO BOOK_COPIES(Book_id,Branch_id,No_of_Copies) VALUES (102,20,30);

INSERT INTO BOOK_COPIES(Book_id,Branch_id,No_of_Copies) VALUES (103,10,0);

INSERT INTO BOOK_COPIES(Book_id,Branch_id,No_of_Copies) VALUES (103,20,0);

INSERT INTO BOOK_COPIES(Book_id,Branch_id,No_of_Copies) VALUES (104,10,0);

INSERT INTO BOOK_COPIES(Book_id,Branch_id,No_of_Copies) VALUES (104,20,0);

INSERT INTO BOOK_LENDING(Book_id,Branch_id,Card_No,Date_Out,Due_date) VALUES
(101,20,1001,'2017-02-10','2017-04-10');

INSERT INTO BOOK_LENDING(Book_id,Branch_id,Card_No,Date_Out,Due_date) VALUES
(102,20,1001,'2017-02-10','2017-04-10');

INSERT INTO BOOK_LENDING(Book_id,Branch_id,Card_No,Date_Out,Due_date) VALUES
(103,20,1001,'2017-02-10','2017-04-10');

INSERT INTO BOOK_LENDING(Book_id,Branch_id,Card_No,Date_Out,Due_date) VALUES
(104,20,1001,'2017-02-10','2017-04-10');

INSERT INTO BOOK_LENDING(Book_id,Branch_id,Card_No,Date_Out,Due_date) VALUES
(102,20,1002,'2017-02-10','2017-04-10');

INSERT INTO BOOK_LENDING(Book_id,Branch_id,Card_No,Date_Out,Due_date) VALUES
(103,10,1020,'2017-04-10','2017-06-10');

INSERT INTO BOOK_LENDING(Book_id,Branch_id,Card_No,Date_Out,Due_date) VALUES
(104,10,1010,'2017-04-10','2017-06-10');

INSERT INTO BOOK_LENDING(Book_id,Branch_id,Card_No,Date_Out,Due_date) VALUES
(101,20,1030,'2017-06-10','2017-08-10');

INSERT INTO BOOK_LENDING(Book_id,Branch_id,Card_No,Date_Out,Due_date) VALUES
(101,20,1040,'2017-08-10','2017-10-10');

SELECT * FROM BOOK;

SELECT * FROM BOOK_AUTHORS;

SELECT * FROM PUBLISHER;

SELECT * FROM LIBRARY_BRANCH;

SELECT * FROM BOOK_COPIES;

SELECT * FROM BOOK_LENDING;

SELECT Book_id,Title,Publisher_Name,Author_Name,Branch_id,No_of_Copies FROM BOOk
NATURAL JOIN BOOK_AUTHORS NATURAL JOIN BOOK_COPIES WHERE
BOOK.Book_id=Book_Authors.Book_id AND Book.Book_id=BOOK_COPIES.Book_id;

SELECT * FROM BOOK_LENDING WHERE Date_Out BETWEEN '2017-01-01' AND '2017-06-
01' GROUP BY Card_No HAVING COUNT(*)>3;

DELETE FROM BOOK WHERE Book_id=104;

SELECT * , AVG(Pub_Year) OVER (PARTITION BY Pub_Year) FROM BOOK;
```

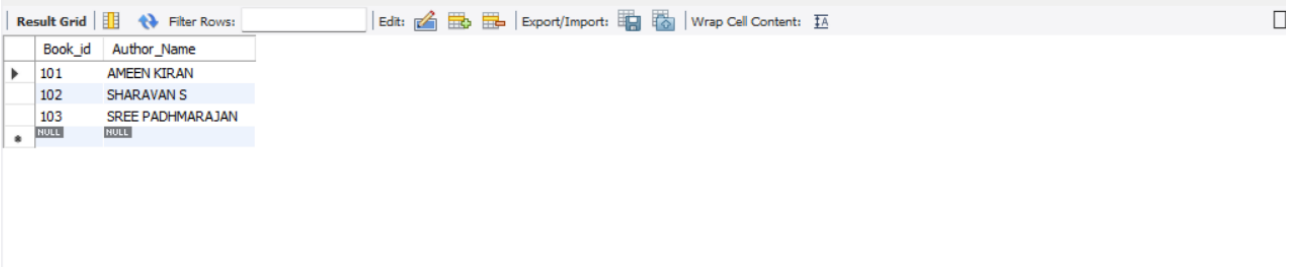
```
CREATE VIEW BOOKSS AS SELECT BOOK.Title,SUM(BOOK_COPIES.No_of_Copies) FROM
BOOK,BOOK_COPIES WHERE BOOK.Book_id=BOOK_COPIES.Book_id GROUP BY
BOOK.Title;

SELECT * FROM BOOKSS;
```

RESULT: Output obtained successfully

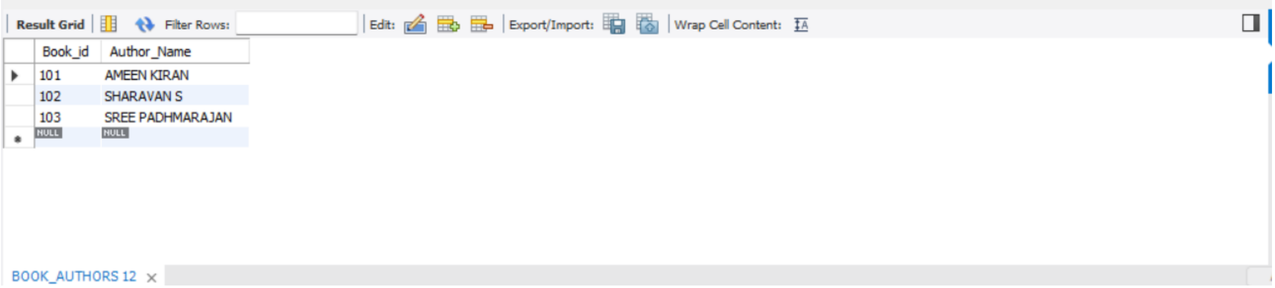
OUTPUT:



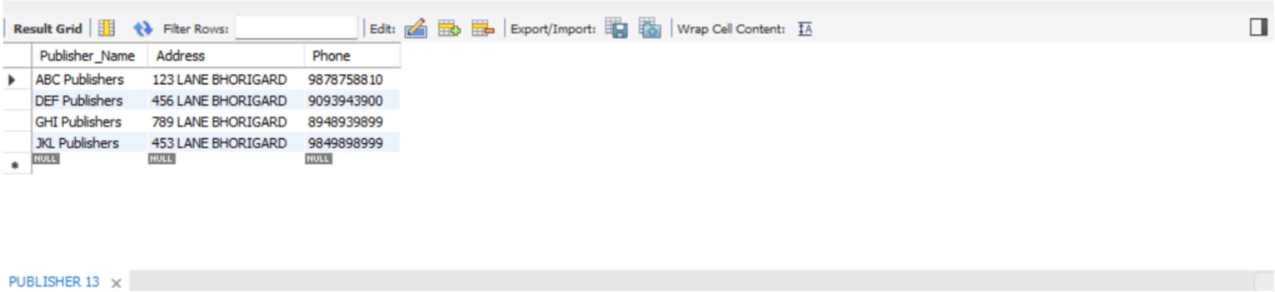
Book_id	Title	Publisher_Name	Pub_Year
101	Orchid for you	ABC Publishers	2002-02-20
102	Diochan Fantasy	DEF Publishers	2017-02-10
103	Adventures of Creati	GHI Publishers	2017-06-20
NULL	NULL	NULL	NULL



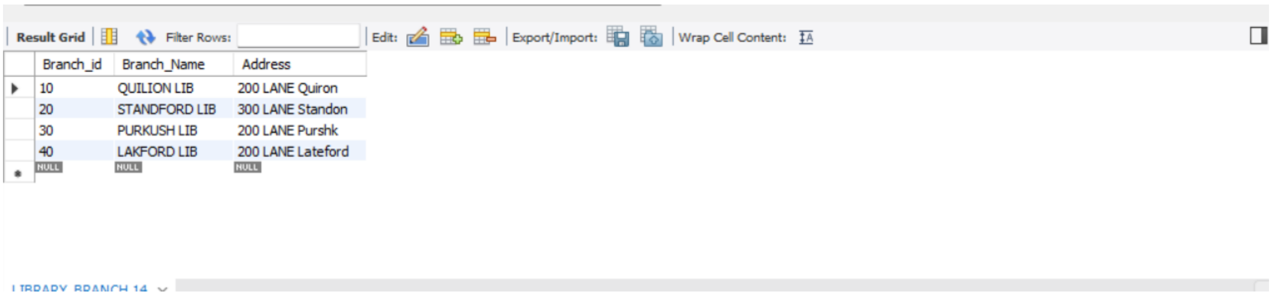
Book_id	Author_Name
101	AMEEN KIRAN
102	SHARAVAN S
103	SREE PADHMARAJAN
NULL	NULL



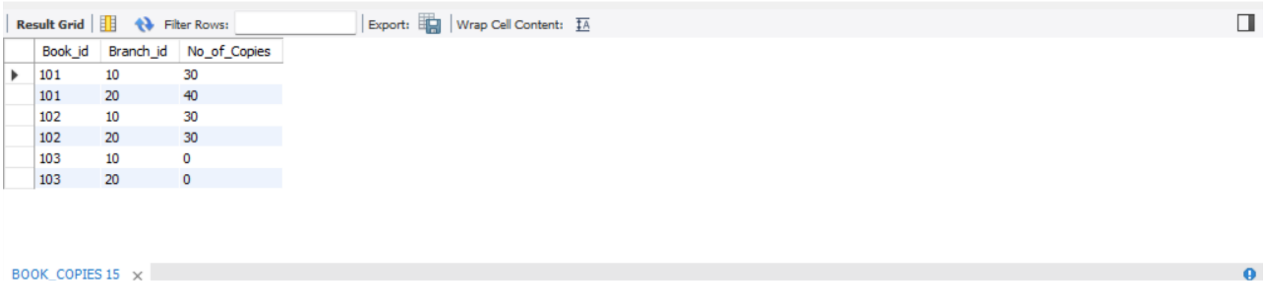
Book_id	Author_Name
101	AMEEN KIRAN
102	SHARAVAN S
103	SREE PADHMARAJAN
NULL	NULL



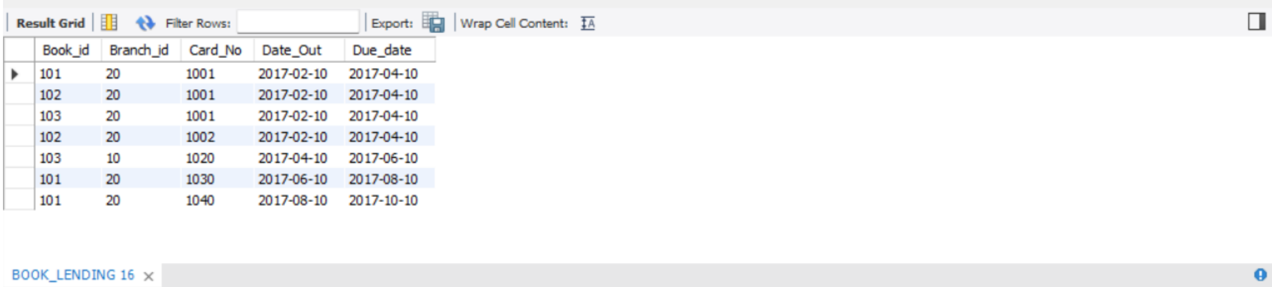
Publisher_Name	Address	Phone
ABC Publishers	123 LANE BHORIGARD	9878758810
DEF Publishers	456 LANE BHORIGARD	9093943900
GHI Publishers	789 LANE BHORIGARD	8948939899
JKL Publishers	453 LANE BHORIGARD	9849898999
NULL	NULL	NULL



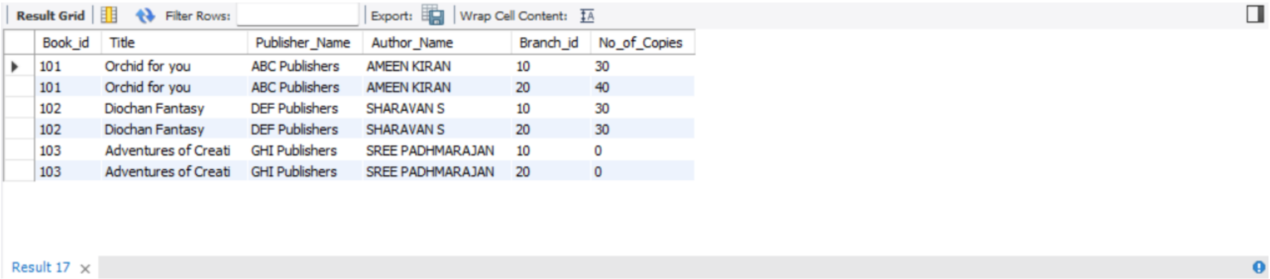
Branch_id	Branch_Name	Address
10	QUILION LIB	200 LANE Quiron
20	STANDFORD LIB	300 LANE Standon
30	PURKUSH LIB	200 LANE Purshk
40	LAKFORD LIB	200 LANE Lateford
NULL	NULL	NULL



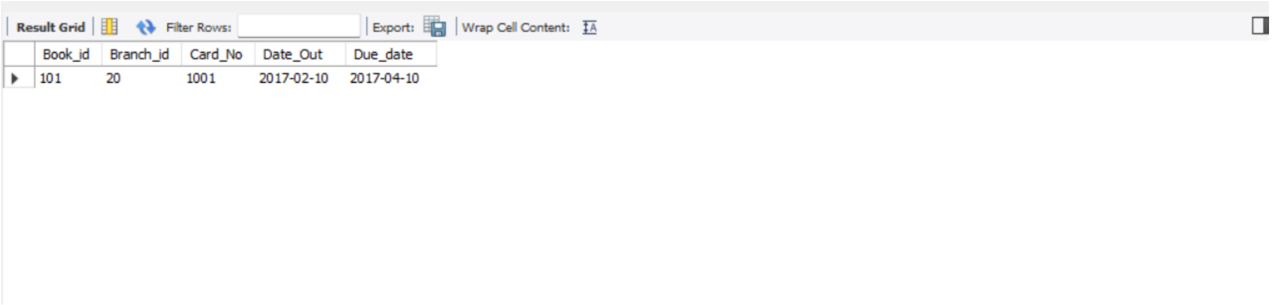
Book_id	Branch_id	No_of_Copies
101	10	30
101	20	40
102	10	30
102	20	30
103	10	0
103	20	0



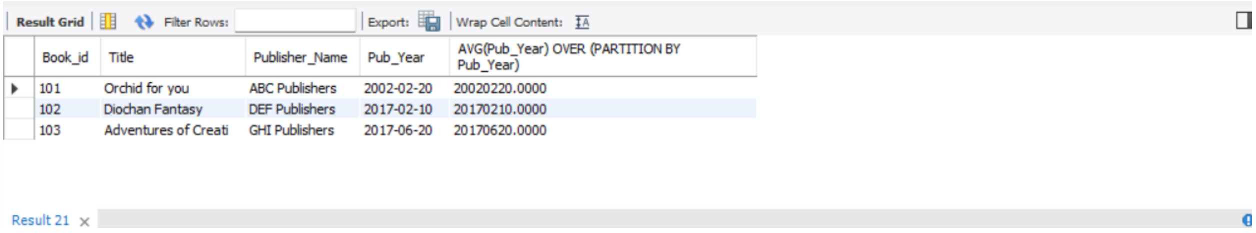
Book_id	Branch_id	Card_No	Date_Out	Due_date
101	20	1001	2017-02-10	2017-04-10
102	20	1001	2017-02-10	2017-04-10
103	20	1001	2017-02-10	2017-04-10
102	20	1002	2017-02-10	2017-04-10
103	10	1020	2017-04-10	2017-06-10
101	20	1030	2017-06-10	2017-08-10
101	20	1040	2017-08-10	2017-10-10



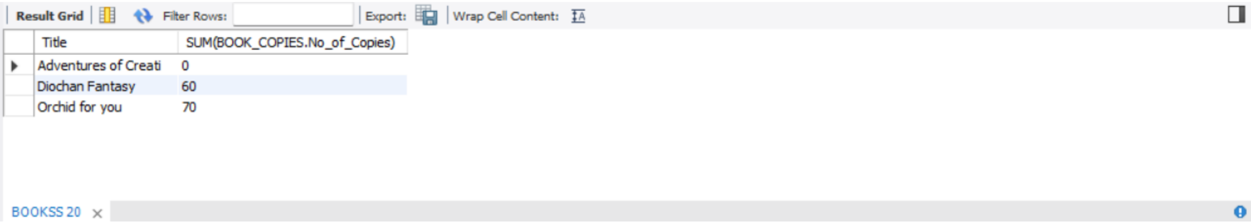
Book_id	Title	Publisher_Name	Author_Name	Branch_id	No_of_Copies
101	Orchid for you	ABC Publishers	AMEEN KIRAN	10	30
101	Orchid for you	ABC Publishers	AMEEN KIRAN	20	40
102	Diochan Fantasy	DEF Publishers	SHARAVAN S	10	30
102	Diochan Fantasy	DEF Publishers	SHARAVAN S	20	30
103	Adventures of Creati	GHI Publishers	SREE PADHMARAJAN	10	0
103	Adventures of Creati	GHI Publishers	SREE PADHMARAJAN	20	0



Book_id	Branch_id	Card_No	Date_Out	Due_date
101	20	1001	2017-02-10	2017-04-10



Book_id	Title	Publisher_Name	Pub_Year	AVG(Pub_Year) OVER (PARTITION BY Pub_Year)
101	Orchid for you	ABC Publishers	2002-02-20	20020220.0000
102	Diochan Fantasy	DEF Publishers	2017-02-10	20170210.0000
103	Adventures of Creati	GHI Publishers	2017-06-20	20170620.0000



Title	SUM(BOOK_COPIES.No_of_Copies)
Adventures of Creati	0
Diochan Fantasy	60
Orchid for you	70

2.Queries using aggregate functions(COUNT,AVG,MIN,MAX,SUM),Group by,Order by,Having.

E_ID	E_NAME	AGE	SALARY
101	ANU	22	9000
102	Shane	29	8000
103	Rohan	34	6000
104	Scott	44	10000
105	Tiger	35	8000
106	Alex	27	7000
107	Abhi	29	8000

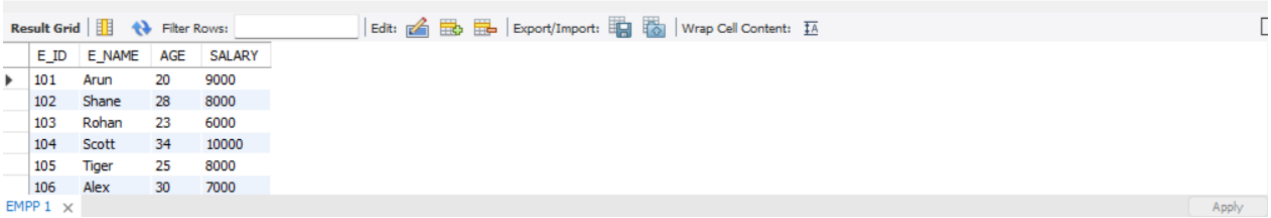
(i) Create Employee table containing all Records.
(ii) Count number of employee names from employee table.
(iii) Find the Maximum age from employee table
(iv) Find the Minimum age from employee table.
(v) Display the Sum of age employee table.
(vi) Display the Average of age from Employee table
(vii) Create a View for age in employee table
(viii) Display views
(ix) Find grouped salaries of employees.
(x) Find salaries of employee in Ascending Order
(xi) Find salaries of employee in Descending Order

Scripts:

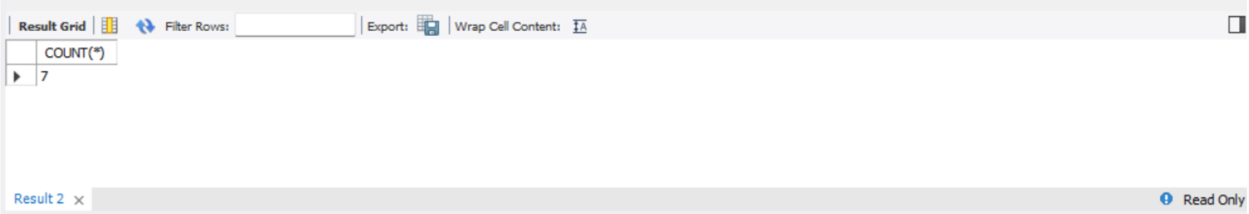
```
CREATE DATABASE EMP;


USE EMP;


CREATE TABLE EMPP(

        E_ID int not null,

  E_NAME varchar(20) not null,

  AGE int not null,

  SALARY int not null,

  PRIMARY KEY(E_ID)

  );


INSERT INTO EMPP(E_ID,E_NAME,AGE,SALARY) VALUES(101,"Arun",20,9000);

INSERT INTO EMPP(E_ID,E_NAME,AGE,SALARY) VALUES(102,"Shane",28,8000);

INSERT INTO EMPP(E_ID,E_NAME,AGE,SALARY) VALUES(103,"Rohan",23,6000);

INSERT INTO EMPP(E_ID,E_NAME,AGE,SALARY) VALUES(104,"Scott",34,10000);
INSERT INTO EMPP(E_ID,E_NAME,AGE,SALARY) VALUES(105,"Tiger",25,8000);
INSERT INTO EMPP(E_ID,E_NAME,AGE,SALARY) VALUES(106,"Alex",30,7000);
INSERT INTO EMPP(E_ID,E_NAME,AGE,SALARY) VALUES(107,"Abhi",29,8000);

SELECT * FROM EMPP;

SELECT COUNT(*) FROM EMPP;

SELECT MAX(AGE) FROM EMPP;

SELECT MIN(AGE) FROM EMPP;

SELECT SUM(AGE) FROM EMPP;

SELECT AVG(AGE) FROM EMPP;

CREATE VIEW AGES AS SELECT AGE FROM EMPP;

SELECT * FROM AGES;

SELECT SALARY FROM EMPP GROUP BY SALARY;

SELECT SALARY FROM EMPP ORDER BY SALARY ASC;

SELECT SALARY FROM EMPP ORDER BY SALARY DESC;
```

RESULT: Output obtained successfully

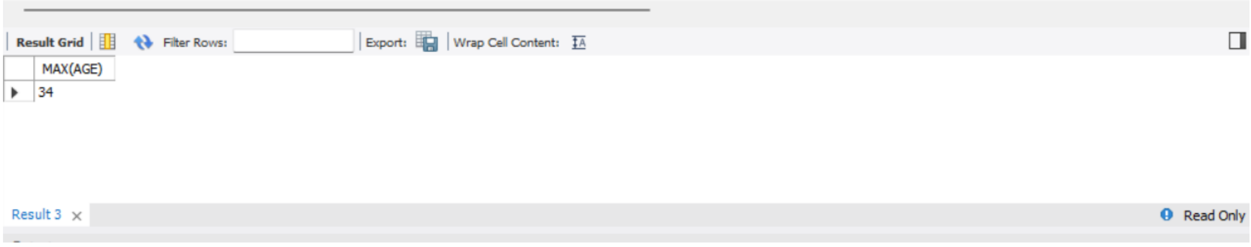
OUTPUT:



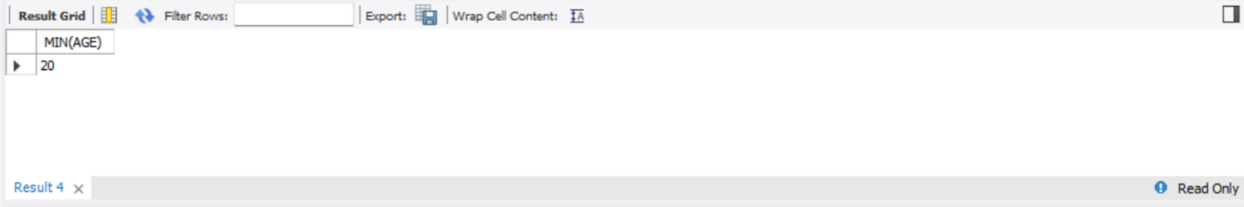
E_ID	E_NAME	AGE	SALARY
101	Arun	20	9000
102	Shane	28	8000
103	Rohan	23	6000
104	Scott	34	10000
105	Tiger	25	8000
106	Alex	30	7000



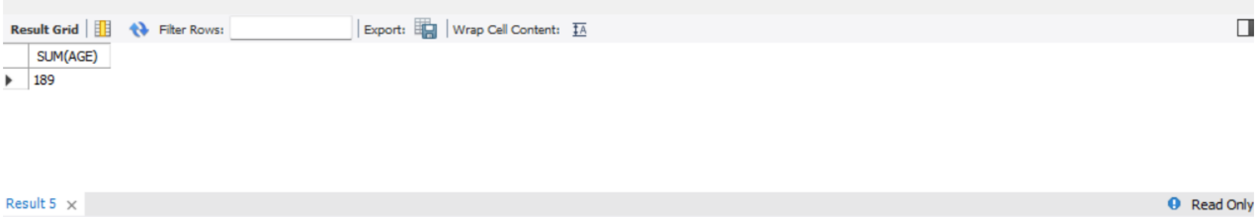
COUNT(*)
7



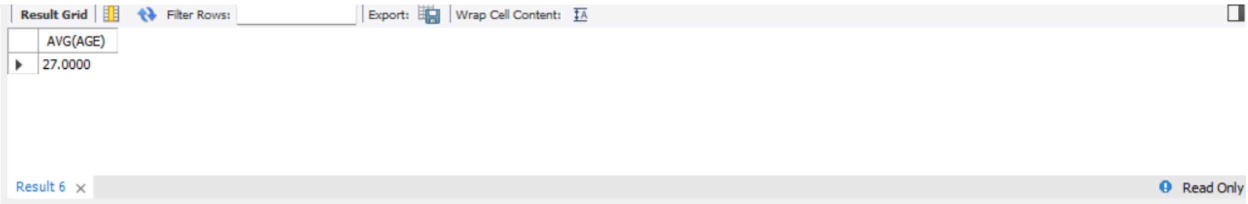
MAX(AGE)
34



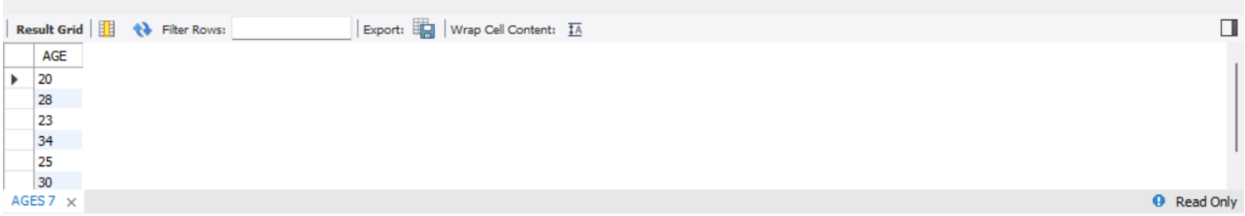
MIN(AGE)
20



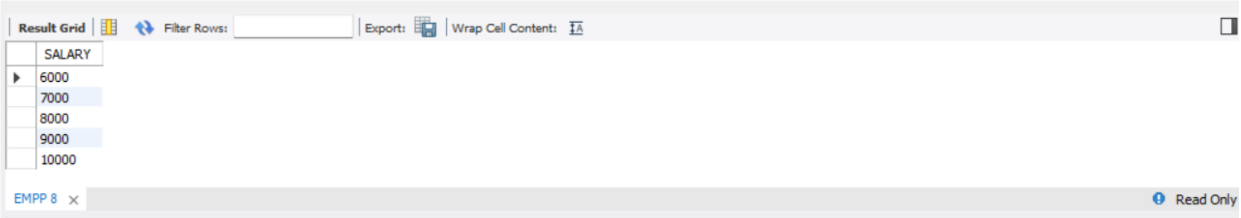
SUM(AGE)
189



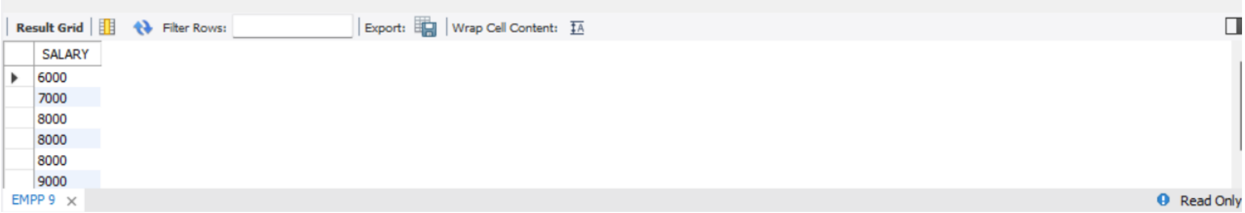
AVG(AGE)
27.0000



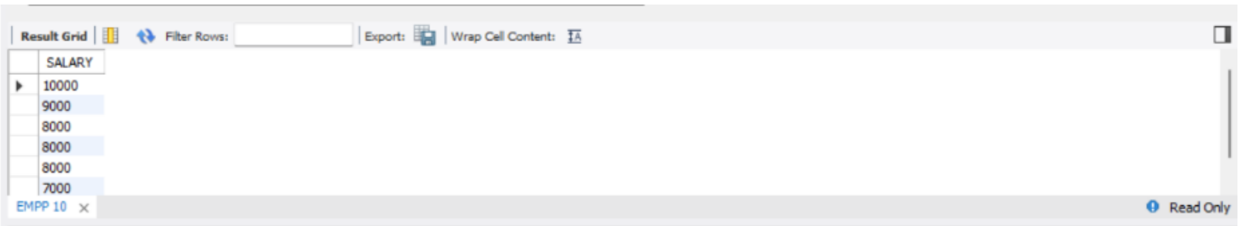
AGE
20
28
23
34
25
30



SALARY
6000
7000
8000
9000
10000



SALARY
6000
7000
8000
8000
8000
9000



SALARY
10000
9000
8000
8000
8000
7000