

DBMS LAB RECORD TEST-1

NAME:KUSUM M R

USN:1BM19CS077

COURSE NAME:DATABASE MANAGEMENT SYSTEM

COURSE CODE:19CS4PCDBM

DBMS LAB RECORD TEST-1

PROGRAM 1: INSURANCE DATABASE

QUESTION:

Consider the Insurance database given below. The primary keys are underlined and the data types are specified.

PERSON (driver-id #: String, name: String, address: String)

CAR (Regno: String, model: String, year: int)

ACCIDENT (report-number: int, adate: date, location: String)

OWNS (driver-id #: String, Regno: String)

PARTICIPATED (driver-id: String, Regno: String, report-number: int, damage-amount: int)

i. Create the above tables by properly specifying the primary keys and the foreign keys.

ii. Enter at least five tuples for each relation.

iii. Demonstrate how you

a. Update the damage amount for the car with a specific Regno in the accident with report number 12 to

25000.

b. Add a new accident to the database.

iv. Find the total number of people who owned cars that involved in accidents in 2008.

v. Find the number of accidents in which cars belonging to a specific model were involved.

PROGRAM CODE:

```
create database Insurance;
```

```
use Insurance;
```

```
CREATE TABLE PERSON(DRIVER_ID VARCHAR(10),NAME VARCHAR(20),ADDRESS  
VARCHAR(15),PRIMARY KEY(DRIVER_ID));
```

```
show tables;
```

```
desc PERSON;
```

```
SELECT *FROM PERSON;
```

```
create table car(Regno varchar(10),Model varchar(20),Year date,Primary key(Regno));
```

```

create table Accident(report_no int,ADATE DATE,Location varchar(15),Primary key(report_no));
create table owns(driver_id varchar(10),regno varchar(10),primary key(driver_id,regno),
foreign key(driver_id) references person(driver_id) on delete cascade, foreign key(regno) references
car(regno) on delete cascade);

CREATE TABLE PARTICIPATED(driver_id varchar(10),regno varchar(10),report_no int, damage_amt
float,
foreign key (driver_id,regno) references OWNS(driver_id,regno) ON DELETE CASCADE,
foreign key (REPORT_NO) references ACCIDENT(REPORT_NO) ON DELETE CASCADE);

show tables;

insert into PERSON(DRIVER_ID,NAME,ADDRESS)values('1111','RAMU', 'K.S.LAYOUT');

insert into PERSON(DRIVER_ID,NAME,ADDRESS)values('2222','JOHN', 'INDIRANAGAR');

insert into PERSON(DRIVER_ID,NAME,ADDRESS)values('3333','PRIYA','JAYANAGAR');

insert into PERSON(DRIVER_ID,NAME,ADDRESS)values('4444','GOPAL','WHITEFIELD');

insert into PERSON(DRIVER_ID,NAME,ADDRESS)values('5555','LATHA',' VIJAYANAGAR');

COMMIT;

desc PERSON;

SELECT *FROM PERSON;

insert into car(regno,Model,Year)values('KA04Q2301','MARUTHI-DX', '2000-10-11');

insert into car(regno,Model,Year)values('KA05P1000',' FORDICON', '2000-09-08');

insert into car(regno,Model,Year)values('KA03L1234','ZEN-VXI', '1999-07-06');

insert into car(regno,Model,Year)values('KA03L9999',' MARUTH-DX', '2002-06-05');

insert into car(regno,Model,Year)values('KA01P4020',' INDICA-VX', '2002-05-04');

COMMIT;

desc car;

SELECT *FROM car;

insert into Accident(report_no,ADATE,Location)values('12',' 2002-06-02',' M G ROAD');

insert into Accident(report_no,ADATE,Location)values('200',' 2002-12-10',' DOUBLEROAD');

insert into Accident(report_no,ADATE,Location)values('300',' 1999-07-10','M G ROAD');

insert into Accident(report_no,ADATE,Location)values('25000',' 2000-06-11',' RESIDENCY ROAD');

insert into Accident(report_no,ADATE,Location)values('26500',' 2001-08-12',' RICHMOND ROAD');

COMMIT;

desc Accident;

```

```

SELECT *FROM Accident;

insert into owns(driver_id,regno)values('1111', 'KA04Q2301');

insert into owns(driver_id,regno)values('1111','KA05P1000');

insert into owns(driver_id,regno)values('2222','KA03L1234');

insert into owns(driver_id,regno)values('3333','KA03L9999');

insert into owns(driver_id,regno)values('4444','KA01P4020');

COMMIT;

desc owns;

SELECT *FROM owns;

insert into PARTICIPATED(driver_id,regno,report_no,damage_amt)values('1111', 'KA04Q2301',' 12','20000');

insert into PARTICIPATED(driver_id,regno,report_no,damage_amt)values('2222','KA03L1234','200','500');

insert into PARTICIPATED(driver_id,regno,report_no,damage_amt)values('3333','KA03L9999','300','10000');

insert into PARTICIPATED(driver_id,regno,report_no,damage_amt)values('4444','KA01P4020','25000','2375');

insert into
PARTICIPATED(driver_id,regno,report_no,damage_amt)values('1111','KA05P1000','26500','70000');

COMMIT;

desc PARTICIPATED ;

SELECT *FROM PARTICIPATED;

/*
a. Update the damage amount for the car with a specific Regno in the accident with report number
12 to
25000.

*/
UPDATE PARTICIPATED SET DAMAGE_AMT=25000 WHERE REPORT_NO =12 AND
REGNO='KA04Q2301';

COMMIT;

desc PARTICIPATED ;

SELECT *FROM PARTICIPATED;

/*

```

b. Add a new accident to the database

```
*/
```

```
insert into Accident(report_no,ADATE,Location)values('500','2005-06-02','Mysore Road');
```

```
desc Accident;
```

```
SELECT *FROM Accident;
```

```
/*
```

iv. Find the total number of people who owned cars that involved in accidents in 2008

```
*/
```

```
select count(*) from Accident where year(ADATE)=2008;
```

```
/*
```

V. Find the number of accidents in which cars belonging to a specific model were involved

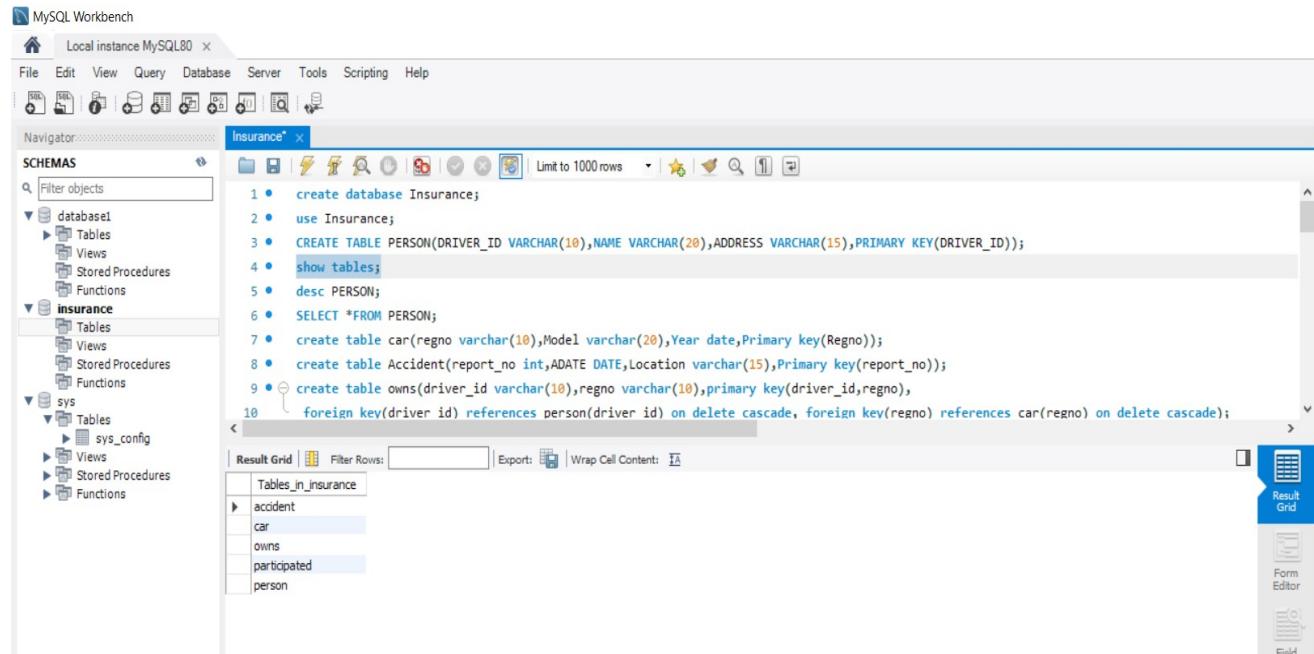
```
*/
```

```
SELECT COUNT(A.REPORT_NO) FROM ACCIDENT A, PARTICIPATED P, CAR C
```

```
WHERE A.REPORT_NO=P.REPORT_NO AND
```

```
P.REGNO=C.REGNO AND C.MODEL='MARUTHI-DX';
```

SCREENSHOTS OF OUTPUT:



The screenshot shows the MySQL Workbench interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The left sidebar displays the Navigator with Schemas (database1, insurance), Tables, Views, Stored Procedures, and Functions. The main area shows an SQL editor with the following code:

```
1 • create database Insurance;
2 • use Insurance;
3 • CREATE TABLE PERSON(DRIVER_ID VARCHAR(10),NAME VARCHAR(20),ADDRESS VARCHAR(15),PRIMARY KEY(DRIVER_ID));
4 • show tables;
5 • desc PERSON;
6 • SELECT *FROM PERSON;
7 • create table car(regno varchar(10),Model varchar(20),Year date,Primary key(Regno));
8 • create table Accident(report_no int,ADATE DATE,Location varchar(15),Primary key(report_no));
9 • create table owns(driver_id varchar(10),regno varchar(10),primary key(driver_id,regno),
10   foreign key(driver_id) references person(driver_id) on delete cascade, foreign key(regno) references car(regno) on delete cascade);
```

Below the SQL editor is a Results Grid titled "Tables_in_insurance" with the following data:

Tables_in_insurance
accident
car
owns
participated
person

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator: Insurance*

SCHEMAS

- database1
 - Tables
 - Views
 - Stored Procedures
 - Functions
- insurance
 - Tables
 - Views
 - Stored Procedures
 - Functions
- sys
 - Tables
 - sys_config
 - Views
 - Stored Procedures
 - Functions

Query Editor:

```

16 • insert into PERSON(DRIVER_ID,NAME,ADDRESS)values('2222','JOHN', 'INDIRANAGAR');
17 • insert into PERSON(DRIVER_ID,NAME,ADDRESS)values('3333','PRIYA', 'JAYANAGAR');
18 • insert into PERSON(DRIVER_ID,NAME,ADDRESS)values('4444','GOPAL', 'WHITEFIELD');
19 • insert into PERSON(DRIVER_ID,NAME,ADDRESS)values('5555','LATHA', 'VIJAYANAGAR');
20 • COMMIT;
21 • desc PERSON;
22 • SELECT *FROM PERSON;
23 • insert into car(regno,Model,Year)values('KA04Q2301', 'MARUTHI-DX', '2000-10-11');
24 • insert into car(regno,Model,Year)values('KA05P1000', 'FORDICON', '2000-09-08');
25 • insert into car(regno,Model,Year)values('KA03L1234', 'ZEN-VXI', '1999-07-06');

```

Result Grid:

DRIVER_ID	NAME	ADDRESS
1111	RAMU	K.S.LAYOUT
2222	JOHN	INDIRANAGAR
3333	PRIYA	JAYANAGAR
4444	GOPAL	WHITEFIELD
5555	LATHA	VIJAYANAGAR
NULL	NULL	NULL

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator: Insurance*

SCHEMAS

- database1
 - Tables
 - Views
 - Stored Procedures
 - Functions
- insurance
 - Tables
 - Views
 - Stored Procedures
 - Functions
- sys
 - Tables
 - sys_config
 - Views
 - Stored Procedures
 - Functions

Query Editor:

```

28 • COMMIT;
29 • desc car;
30 • SELECT *FROM car;
31 • insert into Accident(report_no,ADATE,Location)values('12', ' 2002-06-02', ' M G ROAD');
32 • insert into Accident(report_no,ADATE,Location)values('200', ' 2002-12-10', ' DOUBLEROAD');
33 • insert into Accident(report_no,ADATE,Location)values('300', ' 1999-07-10', 'M G ROAD');
34 • insert into Accident(report_no,ADATE,Location)values('25000', ' 2000-06-11', ' RESIDENCY ROAD');
35 • insert into Accident(report_no,ADATE,Location)values('26500', ' 2001-08-12', ' RICHMOND ROAD');
36 • COMMIT;
37 • desc Accident;

```

Result Grid:

regno	Model	Year
KA01P4020	INDICA-VX	2002-05-04
KA03L1234	ZEN-VXI	1999-07-06
KA03L9999	MARUTHI-DX	2002-06-05
KA04Q2301	MARUTHI-DX	2000-10-11
KA05P1000	FORDICON	2000-09-08
NULL	NULL	NULL

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator: Insurance*

SCHEMAS

- database1
 - Tables
 - Views
 - Stored Procedures
 - Functions
- insurance
 - Tables
 - Views
 - Stored Procedures
 - Functions
- sys
 - Tables
 - sys_config
 - Views
 - Stored Procedures
 - Functions

Query Editor:

```

37 • desc Accident;
38 • SELECT *FROM Accident;
39 • insert into owns(driver_id,regno)values('1111', 'KA04Q2301');
40 • insert into owns(driver_id,regno)values('1111', 'KA05P1000');
41 • insert into owns(driver_id,regno)values('2222', 'KA03L1234');
42 • insert into owns(driver_id,regno)values('3333', 'KA03L9999');
43 • insert into owns(driver_id,regno)values('4444', 'KA01P4020');
44 • COMMIT;
45 • desc owns;
46 • SELECT *FROM owns;
    
```

Result Grid:

report_no	ADATE	Location
12	2002-06-02	M G ROAD
200	2002-12-10	DOUBLEROAD
300	1999-07-10	M G ROAD
25000	2000-06-11	RESIDENCY ROAD
26500	2001-08-12	RICHMOND ROAD
*	NULL	NULL

Toolbar:

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator: Insurance*

SCHEMAS

- database1
 - Tables
 - Views
 - Stored Procedures
 - Functions
- insurance
 - Tables
 - Views
 - Stored Procedures
 - Functions
- sys
 - Tables
 - sys_config
 - Views
 - Stored Procedures
 - Functions

Query Editor:

```

38 • SELECT *FROM Accident;
39 • insert into owns(driver_id,regno)values('1111', 'KA04Q2301');
40 • insert into owns(driver_id,regno)values('1111', 'KA05P1000');
41 • insert into owns(driver_id,regno)values('2222', 'KA03L1234');
42 • insert into owns(driver_id,regno)values('3333', 'KA03L9999');
43 • insert into owns(driver_id,regno)values('4444', 'KA01P4020');
44 • COMMIT;
45 • desc owns;
46 • SELECT *FROM owns;
47 • insert into PARTICIPATED(driver_id,regno,report_no,damage_amt)values('1111', 'KA04Q2301', '12', '20000');
    
```

Result Grid:

driver_id	regno
4444	KA01P4020
2222	KA03L1234
3333	KA03L9999
1111	KA04Q2301
1111	KA05P1000
*	NULL

Toolbar:

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator: Insurance*

SCHEMAS

- database1
 - Tables
 - Views
 - Stored Procedures
 - Functions
- insurance
 - Tables
 - Views
 - Stored Procedures
 - Functions
- sys
 - Tables
 - sys_config
 - Views
 - Stored Procedures
 - Functions

1. SELECT *FROM owns;

2. insert into PARTICIPATED(driver_id,regno,report_no,damage_amt)values('1111','KA04Q2301','12','20000');

3. insert into PARTICIPATED(driver_id,regno,report_no,damage_amt)values('2222','KA03L1234','200','500');

4. insert into PARTICIPATED(driver_id,regno,report_no,damage_amt)values('3333','KA03L9999','300','10000');

5. insert into PARTICIPATED(driver_id,regno,report_no,damage_amt)values('4444','KA01P4020','25000','2375');

6. insert into PARTICIPATED(driver_id,regno,report_no,damage_amt)values('1111','KA05P1000','26500','70000');

7. COMMIT;

8. desc PARTICIPATED ;

9. SELECT *FROM PARTICIPATED;

10.

Result Grid | Filter Rows: Export: Wrap Cell Content:

driver_id	regno	report_no	damage_amt
1111	KA04Q2301	12	20000
2222	KA03L1234	200	500
3333	KA03L9999	300	10000
4444	KA01P4020	25000	2375
1111	KA05P1000	26500	70000

Result Grid Form Editor Field Types

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator: Insurance*

SCHEMAS

- database1
 - Tables
 - Views
 - Stored Procedures
 - Functions
- insurance
 - Tables
 - Views
 - Stored Procedures
 - Functions
- sys
 - Tables
 - sys_config
 - Views
 - Stored Procedures
 - Functions

1. insert into PARTICIPATED(driver_id,regno,report_no,damage_amt)values('4444','KA01P4020','25000','2375');

2. insert into PARTICIPATED(driver_id,regno,report_no,damage_amt)values('1111','KA05P1000','26500','70000');

3. COMMIT;

4. desc PARTICIPATED ;

5. SELECT *FROM PARTICIPATED;

6. UPDATE PARTICIPATED SET DAMAGE_AMT=25000 WHERE REPORT_NO =12 AND REGNO='KA04Q2301';

7. COMMIT;

8. desc PARTICIPATED ;

9. SELECT *FROM PARTICIPATED;

10.

Result Grid | Filter Rows: Export: Wrap Cell Content:

driver_id	regno	report_no	damage_amt
1111	KA04Q2301	12	25000
2222	KA03L1234	200	500
3333	KA03L9999	300	10000
4444	KA01P4020	25000	2375
1111	KA05P1000	26500	70000

Result Grid Form Editor Field Types

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator Insurance

SCHEMAS

database1

- Tables
- Views
- Stored Procedures
- Functions

insurance

- Tables
- Views
- Stored Procedures
- Functions

sys

- Tables
- sys_config
- Views
- Stored Procedures
- Functions

```

59 • UPDATE PARTICIPATED SET DAMAGE_AMT=25000 WHERE REPORT_NO =12 AND REGNO='KA04Q2301';
60 • COMMIT;
61 • desc PARTICIPATED ;
62 • SELECT *FROM PARTICIPATED;
63 • /*
64 • b. Add a new accident to the database
65 */
66 • insert into Accident(report_no,ADATE,Location)values('500',' 2005-06-02','Mysore Road');
67 • desc Accident;
68 • SELECT *FROM Accident;
69

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

report_no	ADATE	Location
12	2002-06-02	M G ROAD
200	2002-12-10	DOUBLEROAD
300	1999-07-10	M G ROAD
500	2005-06-02	Mysore Road
25000	2000-06-11	RESIDENCY ROAD
26500	2001-08-12	RICHMOND ROAD
NULL	NULL	NULL

Result Grid Form Editor

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator Insurance

SCHEMAS

database1

- Tables
- Views
- Stored Procedures
- Functions

insurance

- Tables
- Views
- Stored Procedures
- Functions

sys

- Tables
- sys_config
- Views
- Stored Procedures
- Functions

```

66 • insert into Accident(report_no,ADATE,Location)values('500',' 2005-06-02','Mysore Road');
67 • desc Accident;
68 • SELECT *FROM Accident;
69
70 • /*
71 • iv. Find the total number of people who owned cars that involved in accidents in 2008
72 */
73 • select count(*) from Accident where year(ADATE)=2008;
74
75 • /*
76 • V. Find the number of accidents in which cars belonging to a specific model were involved

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

count(*)
0

Result Grid Form Editor

The screenshot shows the MySQL Workbench interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. Below the menu is a toolbar with various icons. The left sidebar, titled 'Navigator', shows the database schema with 'SCHEMAS' expanded to show 'database1' (Tables, Views, Stored Procedures, Functions), 'insurance' (Tables, Views, Stored Procedures, Functions), and 'sys' (Tables, Views, Stored Procedures, Functions). The main area is titled 'Insurance' and contains a query editor with the following SQL code:

```

70  /* iv. Find the total number of people who owned cars that involved in accidents in 2008
71  */
72  */
73 • select count(*) from Accident where year(ADATE)=2008;
74
75  /*
76  V. Find the number of accidents in which cars belonging to a specific model were involved
77  */
78 • SELECT COUNT(A.REPORT_NO) FROM ACCIDENT A, PARTICIPATED P, CAR C
    WHERE A.REPORT_NO=P.REPORT_NO AND
          P.REGNO=C.REGNO AND C.MODEL='MARUTHI-DX';
79
80

```

The results grid below shows the output of the last query:

COUNT(A.REPORT_NO)
1

PROGRAM-2:BOOK DEALER DATABASE

QUESTION:

The following tables are maintained by a book dealer:

AUTHOR(author-id: int, name: String, city: String, country: String)

PUBLISHER(publisher-id: int, name: String, city: String, country: String)

CATALOG(book-id: int, title: String, author-id: int, publisher-id: int, category-id: int, year: int, price: int)

CATEGORY(category-id: int, description: String)

ORDER-DETAILS(order-no: int, book-id: int, quantity: int)

i) Create the above tables by properly specifying the primary keys and the foreign keys.

ii) Enter at least five tuples for each relation.

iii) Give the details of the authors who have 2 or more books in the catalog and the price of the books in the catalog and the year of publication is after 2000.

iv) Find the author of the book which has maximum sales.

v) Demonstrate how you increase the price of books published by a specific publisher by 10%.

PROGRAM CODE:

```
create database bookdealer;
use bookdealer;
create table AUTHOR (
author_id int,
name varchar(20),
city varchar(15),
country varchar(15),
primary key(author_id)
);
show tables;
desc AUTHOR;
SELECT *FROM AUTHOR;
create table PUBLISHER (
publisher_id int,
name varchar(20),
city varchar(15),
country varchar(15),
primary key(publisher_id)
);
create table CATEGORY(
category_id int,
description varchar(20),
primary key(category_id)
);
show tables;
desc CATEGORY;
```

```
SELECT *FROM CATEGORY;

create table CATALOG (
    book_id int,
    title varchar(15),author_id int,publisher_id int,category_id int,
    foreign key(author_id) references AUTHOR(author_id) on delete cascade,
    foreign key(publisher_id) references PUBLISHER(publisher_id) on delete cascade,
    foreign key(category_id) references CATEGORY(category_id) on delete cascade,
    year int,
    price int,
    primary key(book_id)
);

show tables;

desc CATALOG;

SELECT *FROM CATALOG;

create table ORDER_DETAILS (
    order_no int,book_id int,
    foreign key(book_id) references CATALOG(book_id) on delete cascade,
    quantity int
);

show tables;

desc ORDER_DETAILS;

SELECT *FROM ORDER_DETAILS;

insert into AUTHOR(author_id,name,city,country)values(1001,'TERAS CHAN','CA','USA');

insert into
AUTHOR(author_id,name,city,country)values(1002,'STEVENS','ZOMBI','UGANDA');

insert into AUTHOR(author_id,name,city,country)values(1003,'M MANO','CAIR','CANADA');

insert into AUTHOR(author_id,name,city,country)values(1004,'KARTHIK B.P','NEW
YORK','USA');

insert into AUTHOR(author_id,name,city,country)values(1005,'WILLIAM STALLINGS','LAS
VEGAS','USA');
```

```
COMMIT;

desc AUTHOR;

SELECT *FROM AUTHOR;

insert into PUBLISHER(publisher_id,name,city,country)values(1,'PEARSON','NEW YORK','USA');

insert into PUBLISHER(publisher_id,name,city,country)values(2,'EEE','NEW SOUTH VALES','USA');

insert into PUBLISHER(publisher_id,name,city,country)values(3,'PHI','DELHI','INDIA');

insert into
PUBLISHER(publisher_id,name,city,country)values(4,'WILLEY','BERLIN','GERMANY');

insert into PUBLISHER(publisher_id,name,city,country)values(5,'MGH ','NEW YORK','USA');

COMMIT;

desc PUBLISHER;

SELECT *FROM PUBLISHER;

insert into CATEGORY(category_id,description)values(1001,'COMPUTER SCIENCE');

insert into CATEGORY(category_id,description)values(1002,'ALGORITHM DESIGN');

insert into CATEGORY(category_id,description)values(1003,'ELECTRONICS');

insert into CATEGORY(category_id,description)values(1004,'PROGRAMMING');

insert into CATEGORY(category_id,description)values(1005,'OPERATING SYSTEMS');

COMMIT;

desc CATEGORY;

SELECT *FROM CATEGORY;

insert into
CATALOG(book_id,title,author_id,publisher_id,category_id,year,price)values(11,'Unix System Prg',1001,1,1001,2000,251);

insert into
CATALOG(book_id,title,author_id,publisher_id,category_id,year,price)values(12,'Digital Signals',1002,2,1003,2001,425);

insert into
CATALOG(book_id,title,author_id,publisher_id,category_id,year,price)values(13,'Logic Design',1003,3,1002,1999,225);
```

```
insert into
CATALOG(book_id,title,author_id,publisher_id,category_id,year,price)values(14,'Server
Prg',1004,4,1004,2001,333);

insert into
CATALOG(book_id,title,author_id,publisher_id,category_id,year,price)values(15,'Linux
OS',1005,5,1005,2003,326);

insert into
CATALOG(book_id,title,author_id,publisher_id,category_id,year,price)values(16,'C++
Bible',1005,5 ,1001,2000,526);

insert into
CATALOG(book_id,title,author_id,publisher_id,category_id,year,price)values(17,'COBOL
Handbook',1005,4,1001,2000,658);

COMMIT;

desc CATALOG;

SELECT *FROM CATALOG;

insert into ORDER_DETAILS(order_no,book_id,quantity)values(1,11,5);

insert into ORDER_DETAILS(order_no,book_id,quantity)values(2,12,8);

insert into ORDER_DETAILS(order_no,book_id,quantity)values(3,13,15);

insert into ORDER_DETAILS(order_no,book_id,quantity)values(4,14,22);

insert into ORDER_DETAILS(order_no,book_id,quantity)values(5,15,3);

insert into ORDER_DETAILS(order_no,book_id,quantity)values(2,17,10);

COMMIT;

desc ORDER_DETAILS;

SELECT *FROM ORDER_DETAILS;

SELECT AUTHOR.author_id,name,city,country FROM AUTHOR,CATALOG where
AUTHOR.author_id=CATALOG.author_id group by CATALOG.author_id having
count(CATALOG.author_id)>=2;

SELECT PRICE FROM CATALOG where year>2000;

select name from AUTHOR,CATALOG where AUTHOR.author_id=CATALOG.author_id and
book_id in(select book_id from ORDER_DETAILS where quantity=(select max(quantity) from
ORDER_DETAILS));

update CATALOG set price=1.1*price where publisher_id in(select publisher_id from
PUBLISHER where name='PEARSON');

COMMIT;
```

SELECT *FROM CATALOG;

OUTPUT SCREENSHOTS:

This screenshot shows the MySQL Workbench interface. The query editor window displays the following SQL code:

```
43 | foreign key(book_id) references CATALOG(book_id) on delete cascade,
44 | quantity int
45 |
46 • show tables;
47 • desc ORDER_DETAILS;
48 • SELECT *FROM ORDER_DETAILS;
49 • insert into AUTHOR(author_id,name,city,country)values(1001,'TERAS CHAN','CA','USA');
50 • insert into AUTHOR(author_id,name,city,country)values(1002,'STEVENS','ZOMBI','UGANDA');
51 • insert into AUTHOR(author_id,name,city,country)values(1003,'M MANO','CAIR','CANADA');
52 • insert into AUTHOR(author_id,name,city,country)values(1004,'KARTHIK B.P','NEW YORK','USA');
```

The results pane shows the following table structure:

Tables_in_bookdealer			
author	catalog	category	order_details

This screenshot shows the MySQL Workbench interface. The query editor window displays the following SQL code:

```
52 • insert into AUTHOR(author_id,name,city,country)values(1004,'KARTHIK B.P','NEW YORK','USA');
53 • insert into AUTHOR(author_id,name,city,country)values(1005,'WILLIAM STALLINGS','LAS VEGAS','USA');
54 • COMMIT;
55 • desc AUTHOR;
56 • SELECT *FROM AUTHOR;
57 • insert into PUBLISHER(publisher_id,name,city,country)values(1,'PEARSON','NEW YORK','USA');
58 • insert into PUBLISHER(publisher_id,name,city,country)values(2,'EEE','NEW SOUTH VALES','USA');
59 • insert into PUBLISHER(publisher_id,name,city,country)values(3,'PHI','DELHI','INDIA');
60 • insert into PUBLISHER(publisher_id,name,city,country)values(4,'WILLEY','BERLIN','GERMANY');
61 • insert into PUBLISHER(publisher_id,name,city,country)values(5,'MGH ','NEW YORK','USA');
```

The results pane shows the following data grid:

author_id	name	city	country
1001	TERAS CHAN	CA	USA
1002	STEVENS	ZOMBI	UGANDA
1003	M MANO	CAIR	CANADA
1004	KARTHIK B.P	NEW YORK	USA
1005	WILLIAM STALLINGS	LAS VEGAS	USA
*	NULL	NULL	NULL

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator: Insurance Bookdealer

SCHEMAS

- database1
- insurance
- Tables
- Views
- Stored Procedures
- Functions
- sys

```

58 • insert into PUBLISHER(publisher_id,name,city,country)values(2,'EEE','NEW SOUTH VALES','USA');
59 • insert into PUBLISHER(publisher_id,name,city,country)values(3,'PHI','DELHI','INDIA');
60 • insert into PUBLISHER(publisher_id,name,city,country)values(4,'WILLEY','BERLIN','GERMANY');
61 • insert into PUBLISHER(publisher_id,name,city,country)values(5,'MGH','NEW YORK','USA');
62 • COMMIT;
63 • desc PUBLISHER;
64 • SELECT *FROM PUBLISHER;
65 • insert into CATEGORY(category_id,description)values(1001,'COMPUTER SCIENCE');
66 • insert into CATEGORY(category_id,description)values(1002,'ALGORITHM DESIGN');
67 • insert into CATEGORY(category_id,description)values(1003,'ELECTRONICS');

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Result Grid | Form Editor

publisher_id	name	city	country
1	PEARSON	NEW YORK	USA
2	EEE	NEW SOUTH VALES	USA
3	PHI	DELHI	INDIA
4	WILLEY	BERLIN	GERMANY
5	MGH	NEW YORK	USA
*	NULL	NULL	NULL

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator: Insurance Bookdealer

SCHEMAS

- database1
- insurance
- Tables
- Views
- Stored Procedures
- Functions
- sys

```

70 • COMMIT;
71 • desc CATEGORY;
72 • SELECT *FROM CATEGORY;
73 • insert into CATALOG(book_id,title,author_id,publisher_id,category_id,year,price)values(11,'Unix System Prg',10);
74 • insert into CATALOG(book_id,title,author_id,publisher_id,category_id,year,price)values(12,'Digital Signals',10);
75 • insert into CATALOG(book_id,title,author_id,publisher_id,category_id,year,price)values(13,'Logic Design',1003);
76 • insert into CATALOG(book_id,title,author_id,publisher_id,category_id,year,price)values(14,'Server Prg',1004,4);
77 • insert into CATALOG(book_id,title,author_id,publisher_id,category_id,year,price)values(15,'Linux OS',1005,5,10);
78 • insert into CATALOG(book_id,title,author_id,publisher_id,category_id,year,price)values(16,'C++ Bible',1005,5);
79 • insert into CATALOG(book_id,title,author_id,publisher_id,category_id,year,price)values(17,'COBOL Handbook',100);

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Result Grid | Form Editor

category_id	description
1001	COMPUTER SCIENCE
1002	ALGORITHM DESIGN
1003	ELECTRONICS
1004	PROGRAMMING
1005	OPERATING SYSTEMS
*	NULL

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Insurance Bookdealer x

SCHEMAS

Filter objects

- database1
- insurance
 - Tables
 - Views
 - Stored Procedures
 - Functions
- sys

```

76 • insert into CATALOG(book_id,title,author_id,publisher_id,category_id,year,price)values(14,'Server Prg',1004,4,^
77 • insert into CATALOG(book_id,title,author_id,publisher_id,category_id,year,price)values(15,'Linux OS',1005,5,10
78 • insert into CATALOG(book_id,title,author_id,publisher_id,category_id,year,price)values(16,'C++ Bible',1005,5,^
79 • insert into CATALOG(book_id,title,author_id,publisher_id,category_id,year,price)values(17,'COBOL Handbook',100
80 • COMMIT;
81 • desc CATALOG;
82 • SELECT *FROM CATALOG;
83 • insert into ORDER_DETAILS(order_no,book_id,quantity)values(1,11,5);

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Result Grid | Form Editor | Field Types

book_id	title	author_id	publisher_id	category_id	year	price
11	Unix System Prg	1001	1	1001	2000	251
12	Digital Signals	1002	2	1003	2001	425
13	Logic Design	1003	3	1002	1999	225
14	Server Prg	1004	4	1004	2001	333
15	Linux OS	1005	5	1005	2003	326
16	C++ Bible	1005	5	1001	2000	526
17	COBOL Handbook	1005	4	1001	2000	658
*	NULL	NULL	NULL	NULL	NULL	NULL

Administration Schemas

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Insurance Bookdealer x

SCHEMAS

Filter objects

- database1
- insurance
 - Tables
 - Views
 - Stored Procedures
 - Functions
- sys

```

84 • insert into ORDER_DETAILS(order_no,book_id,quantity)values(2,12,8);
85 • insert into ORDER_DETAILS(order_no,book_id,quantity)values(3,13,15);
86 • insert into ORDER_DETAILS(order_no,book_id,quantity)values(4,14,22);
87 • insert into ORDER_DETAILS(order_no,book_id,quantity)values(5,15,3);
88 • insert into ORDER_DETAILS(order_no,book_id,quantity)values(2,17,10);
89 • COMMIT;
90 • desc ORDER_DETAILS;
91 • SELECT *FROM ORDER_DETAILS;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Result Grid | Form Editor | Field Types

order_no	book_id	quantity
1	11	5
2	12	8
3	13	15
4	14	22
5	15	3
2	17	10

Administration Schemas

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Insurance Bookdealer x

SCHEMAS

Filter objects

database1 insurance

Tables Views Stored Procedures Functions

sys

```
83 • insert into ORDER_DETAILS(order_no,book_id,quantity)values(1,11,5);
84 • insert into ORDER_DETAILS(order_no,book_id,quantity)values(2,12,8);
85 • insert into ORDER_DETAILS(order_no,book_id,quantity)values(3,13,15);
86 • insert into ORDER_DETAILS(order_no,book_id,quantity)values(4,14,22);
87 • insert into ORDER_DETAILS(order_no,book_id,quantity)values(5,15,3);
88 • insert into ORDER_DETAILS(order_no,book_id,quantity)values(2,17,10);
89 • COMMIT;
90 • desc ORDER_DETAILS;
91 • SELECT *FROM ORDER_DETAILS;
92 • SELECT AUTHOR.author_id,name,city,country FROM AUTHOR,CATALOG where AUTHOR.author_id=CATALOG.author_id group by
93 • SELECT PRICE FROM CATALOG where year>2000;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

author_id	name	city	country
1005	WILLIAM STALLINGS	LAS VEGAS	USA

Administration Schemas

Result Grid

Form Editor

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Insurance Bookdealer x

SCHEMAS

Filter objects

database1 insurance

Tables Views Stored Procedures Functions

sys

```
84 • insert into ORDER_DETAILS(order_no,book_id,quantity)values(2,12,8);
85 • insert into ORDER_DETAILS(order_no,book_id,quantity)values(3,13,15);
86 • insert into ORDER_DETAILS(order_no,book_id,quantity)values(4,14,22);
87 • insert into ORDER_DETAILS(order_no,book_id,quantity)values(5,15,3);
88 • insert into ORDER_DETAILS(order_no,book_id,quantity)values(2,17,10);
89 • COMMIT;
90 • desc ORDER_DETAILS;
91 • SELECT *FROM ORDER_DETAILS;
92 • SELECT AUTHOR.author_id,name,city,country FROM AUTHOR,CATALOG where AUTHOR.author_id=CATALOG.author_id group by
93 • SELECT PRICE FROM CATALOG where year>2000;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

PRICE
425
333
326

Administration Schemas

Result Grid

Form Editor

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator Insurance Bookdealer ×

SCHEMAS Filter objects

- database1
- insurance
 - Tables
 - Views
 - Stored Procedures
 - Functions
- sys

```

85 • insert into ORDER_DETAILS(order_no,book_id,quantity)values(3,13,15);
86 • insert into ORDER_DETAILS(order_no,book_id,quantity)values(4,14,22);
87 • insert into ORDER_DETAILS(order_no,book_id,quantity)values(5,15,3);
88 • insert into ORDER_DETAILS(order_no,book_id,quantity)values(2,17,10);
89 • COMMIT;
90 • desc ORDER_DETAILS;
91 • SELECT *FROM ORDER_DETAILS;
92 • SELECT AUTHOR.author_id,name,city,country FROM AUTHOR,CATALOG where AUTHOR.author_id=CATALOG.author_id group b
93 • SELECT PRICE FROM CATALOG where year>2000;
94 • select name from AUTHOR,CATALOG where AUTHOR.author_id=CATALOG.author_id and book_id in(select book_id from OR

```

Result Grid Filter Rows: Export: Wrap Cell Content:

name
KARTHIK.B.P

Administration Schemas

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator Insurance Bookdealer ×

SCHEMAS Filter objects

- database1
- insurance
 - Tables
 - Views
 - Stored Procedures
 - Functions
- sys

```

91 • SELECT *FROM ORDER_DETAILS;
92 • SELECT AUTHOR.author_id,name,city,country FROM AUTHOR,CATALOG where AUTHOR.author_id=CATALOG.author_id group b
93 • SELECT PRICE FROM CATALOG where year>2000;
94 • select name from AUTHOR,CATALOG where AUTHOR.author_id=CATALOG.author_id and book_id in(select book_id from OR
95 • update CATALOG set price=1.1*price where publisher_id in(select publisher_id from PUBLISHER where name='PEARSO
96 • COMMIT;
97 • SELECT *FROM CATALOG;
98

```

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:

book_id	title	author_id	publisher_id	category_id	year	price
11	Unix System Prg	1001	1	1001	2000	276
12	Digital Signals	1002	2	1003	2001	425
13	Logic Design	1003	3	1002	1999	225
14	Server Prg	1004	4	1004	2001	333
15	Linux OS	1005	5	1005	2003	326
16	C++ Bible	1005	5	1001	2000	526
17	COBOL Handbook	1005	4	1001	2000	658
*		HULL	HULL	HULL	HULL	HULL

Administration Schemas

PROGRAM-3:ORDER PROCESSING DATABASE

QUESTION:

Consider the following relations for an Order Processing database application in a company.

CUSTOMER (CUST #: int, cname: String, city: String)

ORDER (order #: int, odate: date, cust #: int, ord-Amt: int)

ITEM (item #: int, unit-price: int)

ORDER-ITEM (order #: int, item #: int, qty: int)

WAREHOUSE (warehouse #: int, city: String)

SHIPMENT (order #: int, warehouse #: int, ship-date: date)

i) Create the above tables by properly specifying the primary keys and the foreign keys and the

foreign

keys.

ii) Enter at least five tuples for each relation.

iii) Produce a listing: CUSTNAME, #oforders, AVG_ORDER_AMT, where the middle column

is the total

numbers of orders by the customer and the last column is the average order amount for that

customer.

iv) List the order# for orders that were shipped from all warehouses that the company has in a

specific city.

v) Demonstrate how you delete item# 10 from the ITEM table and make that field null in the

ORDER_ITEM

table.

PROGRAM CODE:

```
create database Order_processing;
use Order_processing;
CREATE TABLE CUSTOMER
(
    cust_no int,
    cname VARCHAR(15),
    city VARCHAR(15),
    PRIMARY KEY(cust_no)
);

CREATE TABLE ORDERS(
    order_no int,
    odate date,
    cust_no int,
    foreign key(cust_no) references CUSTOMER(cust_no) on delete cascade,
    ord_Amt int,
    primary key(order_no)
);

create table ITEM (
    item_no int,
    unit_price int,
    primary key(item_no)
);

create table ORDER_ITEM (
    order_no int,
    item_no int,
    qty int,
    foreign key(order_no) references ORDERS(order_no) on delete cascade,
```

```
foreign key(item_no) references ITEM(item_no) on delete SET NULL
);

create table WAREHOUSE(
warehouseno int,
city varchar(30),
primary key(warehouseno)
);

create table SHIPMENT(
order_no int,
warehouseno int,
ship_date date,
foreign key(order_no) references ORDERS(order_no) on delete cascade,
foreign key(warehouseno) references WAREHOUSE(warehouseno) on delete cascade
);

show tables;

insert into CUSTOMER(cust_no,cname,city)values(771,'PUSHPA K','BANGALORE');
insert into CUSTOMER(cust_no,cname,city)values(772,'SUMAN','MUMBAI');
insert into CUSTOMER(cust_no,cname,city)values(773,'SOURAV','CALICUT');
insert into CUSTOMER(cust_no,cname,city)values(774,'LAILA','HYDERABAD');
insert into CUSTOMER(cust_no,cname,city)values(775,'FAIZAL','BANGALORE');

COMMIT;

desc CUSTOMER;

SELECT *FROM CUSTOMER;

insert into ORDERS(order_no,odate,cust_no,ord_Amt)values(111,'22-01-02',771,18000);

insert into ORDERS(order_no,odate,cust_no,ord_Amt)values(112,'30-07-02',774,6000);

insert into ORDERS(order_no,odate,cust_no,ord_Amt)values(113,'03-04-03',775,9000);
```

```
insert into ORDERS(order_no,odate,cust_no,ord_Amt)values(114,'03-11-03',775,29000);

insert into ORDERS(order_no,odate,cust_no,ord_Amt)values(115,'10-12-03',773,29000);

insert into ORDERS(order_no,odate,cust_no,ord_Amt)values(116,'19-08-04',772,56000);

insert into ORDERS(order_no,odate,cust_no,ord_Amt)values(117,'10-09-04',771,20000);

insert into ORDERS(order_no,odate,cust_no,ord_Amt)values(118,'20-11-04',775,29000);

insert into ORDERS(order_no,odate,cust_no,ord_Amt)values(119,'13-02-05',774,29000);

insert into ORDERS(order_no,odate,cust_no,ord_Amt)values(120,'13-10-05',775,29000);

COMMIT;

desc ORDERS;

SELECT *FROM ORDERS;

insert into ITEM(item_no,unit_price)values(5001,503);

insert into ITEM(item_no,unit_price)values(5002,750);

insert into ITEM(item_no,unit_price)values(5003,150);

insert into ITEM(item_no,unit_price)values(5004,600);

insert into ITEM(item_no,unit_price)values(5005,890);

COMMIT;

desc ITEM;

SELECT *FROM ITEM;

insert into ORDER_ITEM(order_no,item_no,qty)values(111,5001,50);

insert into ORDER_ITEM(order_no,item_no,qty)values(112,5003,20);

insert into ORDER_ITEM(order_no,item_no,qty)values(113,5002,50);

insert into ORDER_ITEM(order_no,item_no,qty)values(114,5005,60);

insert into ORDER_ITEM(order_no,item_no,qty)values(115,5004,90);
```

```
insert into ORDER_ITEM(order_no,item_no,qty)values(116,5001,10);
insert into ORDER_ITEM(order_no,item_no,qty)values(117,5003,80);
insert into ORDER_ITEM(order_no,item_no,qty)values(118,5005,50);
insert into ORDER_ITEM(order_no,item_no,qty)values(119,5002,10);
insert into ORDER_ITEM(order_no,item_no,qty)values(120,5004,45);

COMMIT;

desc ORDER_ITEM;

SELECT *FROM ORDER_ITEM;

insert into WAREHOUSE(warehouseno,city)values(1,'DELHI');

insert into WAREHOUSE(warehouseno,city)values(2,'BOMBAY');

insert into WAREHOUSE(warehouseno,city)values(3,'CHENNAI');

insert into WAREHOUSE(warehouseno,city)values(4,'BANGALORE');

insert into WAREHOUSE(warehouseno,city)values(5,'BANGALORE');

insert into WAREHOUSE(warehouseno,city)values(6,'DELHI');

insert into WAREHOUSE(warehouseno,city)values(7,'BOMBAY');

insert into WAREHOUSE(warehouseno,city)values(8,'CHENNAI');

insert into WAREHOUSE(warehouseno,city)values(9,'DELHI');

insert into WAREHOUSE(warehouseno,city)values(10,'BANGALORE');

COMMIT;

desc WAREHOUSE;

SELECT *FROM WAREHOUSE;

insert into SHIPMENT(order_no,warehouseno,ship_date)values(111,1,'10-02-02');

insert into SHIPMENT(order_no,warehouseno,ship_date)values(112,5,'10-09-02');

insert into SHIPMENT(order_no,warehouseno,ship_date)values(113,8,'10-02-03');

insert into SHIPMENT(order_no,warehouseno,ship_date)values(114,3,'10-12-03');

insert into SHIPMENT(order_no,warehouseno,ship_date)values(115,9,'19-01-04');

insert into SHIPMENT(order_no,warehouseno,ship_date)values(116,1,'20-09-04');

insert into SHIPMENT(order_no,warehouseno,ship_date)values(117,5,'10-09-04');
```

```
insert into SHIPMENT(order_no,warehouseno,ship_date)values(118,7,'30-11-04');
insert into SHIPMENT(order_no,warehouseno,ship_date)values(119,7,'30-04-05');
insert into SHIPMENT(order_no,warehouseno,ship_date)values(120,6,'21-12-05');

COMMIT;

desc SHIPMENT;

SELECT *FROM SHIPMENT;

/*Produce a listing: CUSTNAME, #oforders, AVG_ORDER_AMT, where the middle
column
is the total numbers of orders by the customer and the last column is the average
order amount for that customer.*/

SELECT C.CNAME as CUSTNAME, COUNT(*) as no_of_orders,AVG(O.ord_Amt) as
AVG_ORDER_AMT FROM CUSTOMER C,
ORDERS O WHERE C.cust_no=O.cust_no GROUP BY C.CNAME;

/*List the order# for orders that were shipped from all warehouses that the company
has in a specific city.*/

SELECT order_no FROM WAREHOUSE W, SHIPMENT S WHERE
W.warehouseno=S.warehouseno AND CITY='BANGALORE';

/*Demonstrate how you delete item# 10 from the ITEM table and make that field
null in the ORDER_ITEM table.*/

delete from ITEM where item_no=5005;

select *from ITEM;

select *from ORDER_ITEM;
```

OUTPUT SCREENSHOTS:

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas Insurance Bookdealer OrderProcessing

```

38 | ship_date date,
39 |     foreign key(order_no) references ORDERS(order_no) on delete cascade,
40 |     foreign key(warehouseno) references WAREHOUSE(warehouseno) on delete cascade
41 | );
42 • show tables;
43 • insert into CUSTOMER(cust_no,cname,city)values(771,'PUSHPA K','BANGALORE');
44 • insert into CUSTOMER(cust_no,cname,city)values(772,'SUMAN','MUMBAI');
45 • insert into CUSTOMER(cust_no,cname,city)values(773,'SOURAV','CALICUT');

```

Result Grid Filter Rows: Export: Wrap Cell Content:

Tables_in_order_processing		
	customer	item
▶	order_item	orders
	shipment	warehouse

Administration Schemas Information

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas Insurance Bookdealer OrderProcessing

```

44 • insert into CUSTOMER(cust_no,cname,city)values(772,'SUMAN','MUMBAI');
45 • insert into CUSTOMER(cust_no,cname,city)values(773,'SOURAV','CALICUT');
46 • insert into CUSTOMER(cust_no,cname,city)values(774,'LAILA','HYDERABAD');
47 • insert into CUSTOMER(cust_no,cname,city)values(775,'FAIZAL','BANGALORE');
48 • COMMIT;
49 • desc CUSTOMER;
50 • SELECT *FROM CUSTOMER;
51 • insert into ORDERS(order_no,odate,cust_no,ord_Amt)values(111,'22-01-02',771,18000);

```

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:

cust_no	cname	city
771	PUSHPA K	BANGALORE
772	SUMAN	MUMBAI
773	SOURAV	CALICUT
774	LAILA	HYDERABAD
775	FAIZAL	BANGALORE
*	NULL	NULL

Administration Schemas

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator Insurance Bookdealer OrderProcessing x

SCHEMAS

Filter objects

bookdealer database1 insurance

Tables Views Stored Procedures Functions

sys

111 2022-01-02 771 18000
112 2030-07-02 774 6000
113 2003-04-03 775 9000
114 2003-11-03 775 29000
115 2010-12-03 773 29000
116 2019-08-04 772 56000
117 2010-09-04 771 20000
118 2020-11-04 775 29000
119 2013-02-05 774 29000
120 2013-10-05 775 29000
NULL NULL NULL NULL

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content: Result Grid Form Editor Field Types

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator Insurance Bookdealer OrderProcessing x

SCHEMAS

Filter objects

bookdealer database1 insurance

Tables Views Stored Procedures Functions

sys

68 • insert into ITEM(item_no,unit_price)values(5005,890);
69 • COMMIT;
70 • desc ITEM;
71 • SELECT *FROM ITEM;
72 • insert into ORDER_ITEM(order_no,item_no,qty)values(111,5001,50);
73 • insert into ORDER_ITEM(order_no,item_no,qty)values(112,5003,20);
74 • insert into ORDER_ITEM(order_no,item_no,qty)values(113,5002,50);
75 • insert into ORDER_ITEM(order_no,item_no,qty)values(114,5005,60);

item_no unit_price

5001 503
5002 750
5003 150
5004 600
5005 890
NULL NULL

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content: Result Grid Form Editor Field Types

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator: Insurance Bookdealer OrderProcessing

SCHEMAS

- bookdealer
- database1
- insurance
 - Tables
 - Views
 - Stored Procedures
 - Functions
- sys

Limit to 1000 rows

```

79 • insert into ORDER_ITEM(order_no,item_no,qty)values(118,5005,50);
80 • insert into ORDER_ITEM(order_no,item_no,qty)values(119,5002,10);
81 • insert into ORDER_ITEM(order_no,item_no,qty)values(120,5004,45);
82 • COMMIT;
83 • desc ORDER_ITEM;
84 • SELECT *FROM ORDER_ITEM;
85 • insert into WAREHOUSE(warehouseno,city)values(1,'DELHI');
  
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

order_no	item_no	qty
111	5001	50
112	5003	20
113	5002	50
114	5005	60
115	5004	90
116	5001	10
117	5003	80
118	5005	50
119	5002	10
120	5004	45

Administration Schemas Information

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator: Insurance Bookdealer OrderProcessing

SCHEMAS

- bookdealer
- database1
- insurance
 - Tables
 - Views
 - Stored Procedures
 - Functions
- sys

Limit to 1000 rows

```

94 • insert into WAREHOUSE(warehouseno,city)values(10,'BANGALORE');
95 • COMMIT;
96 • desc WAREHOUSE;
97 • SELECT *FROM WAREHOUSE;
98 • insert into SHIPMENT(order_no,warehouseno,ship_date)values(111,1,'10-02-02');
99 • insert into SHIPMENT(order_no,warehouseno,ship_date)values(112,5,'10-09-02');
100 • insert into SHIPMENT(order_no,warehouseno,ship_date)values(113,8,'10-02-03');
  
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

warehouseno	city
1	DELHI
2	BOMBAY
3	CHENNAI
4	BANGALORE
5	BANGALORE
6	DELHI
7	BOMBAY
8	CHENNAI
9	DELHI
10	BANGALORE
11	NULL

Administration Schemas Information

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator: Insurance Bookdealer OrderProcessing

SCHEMAS

- bookdealer
- database1
- insurance**
 - Tables
 - Views
 - Stored Procedures
 - Functions
- sys

105 • insert into SHIPMENT(order_no,warehouseno,ship_date)values(118,7,'30-11-04');

106 • insert into SHIPMENT(order_no,warehouseno,ship_date)values(119,7,'30-04-05');

107 • insert into SHIPMENT(order_no,warehouseno,ship_date)values(120,6,'21-12-05');

108 • COMMIT;

109 • desc SHIPMENT;

110 • **SELECT *FROM SHIPMENT;**

111 /*Produce a listing: CUSTNAME, #oforders, AVG ORDER AMT. where the middle column

Result Grid | Filter Rows: Export: Wrap Cell Content:

order_no	warehouseno	ship_date
111	1	2010-02-02
112	5	2010-09-02
113	8	2010-02-03
114	3	2010-12-03
115	9	2019-01-04
116	1	2020-09-04
117	5	2010-09-04
118	7	2030-11-04
119	7	2030-04-05
120	6	2021-12-05

Administration Schemas

Result Grid Form Editor Field Types

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator: Insurance Bookdealer OrderProcessing*

SCHEMAS

- bookdealer
- database1
- insurance**
 - Tables
 - Views
 - Stored Procedures
 - Functions
- sys

107 • insert into SHIPMENT(order_no,warehouseno,ship_date)values(120,6,'21-12-05');

108 • COMMIT;

109 • desc SHIPMENT;

110 • **SELECT *FROM SHIPMENT;**

111 /*Produce a listing: CUSTNAME, #oforders, AVG_ORDER_AMT, where the middle column

112 is the total numbers of orders by the customer and the last column is the average order amount for that customer

113 • **SELECT C.CNAME as CUSTNAME, COUNT(*) as no_of_orders, AVG(O.ord_Amt) as AVG_ORDER_AMT FROM CUSTOMER C,**

114 **ORDERS O WHERE C.cust_no=O.cust_no GROUP BY C.CNAME;**

115 /*List the order# for orders that were shipped from all warehouses that the company has in a specific city.*/

Result Grid | Filter Rows: Export: Wrap Cell Content:

CUSTNAME	no_of_orders	AVG_ORDER_AMT
PUSHPA K	2	19000.0000
SUMAN	1	56000.0000
SOURAV	1	29000.0000
LAILA	2	17500.0000
FAIZAL	4	24000.0000

Administration Schemas

Information

Result Grid Form Editor

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator: Insurance Bookdealer OrderProcessing*

SCHEMAS

109 • desc SHIPMENT;

110 • SELECT *FROM SHIPMENT;

111 /*Produce a listing: CUSTNAME, #oforders, AVG_ORDER_AMT, where the middle column
112 is the total numbers of orders by the customer and the last column is the average order amount for that custom

113 • SELECT C.CNAME as CUSTNAME, COUNT(*) as no_of_orders,AVG(O.ord_Amt) as AVG_ORDER_AMT FROM CUSTOMER C,
114 ORDERS O WHERE C.cust_no=O.cust_no GROUP BY C.CNAME;

115 /*List the order# for orders that were shipped from all warehouses that the company has in a specific city.*/

116 • SELECT order_no FROM WAREHOUSE W, SHIPMENT S WHERE W.warehouseno=S.warehouseno AND CITY='BANGALORE';

117 /*Demonstrate how you delete item# 10 from the ITEM table and make that field null in the ORDER_ITEM table.*/

Result Grid | Filter Rows: Export: Wrap Cell Content:

order_no
112
117

Administration Schemas Information

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator: Insurance Bookdealer OrderProcessing*

SCHEMAS

111 /*Produce a listing: CUSTNAME, #oforders, AVG_ORDER_AMT, where the middle column
112 is the total numbers of orders by the customer and the last column is the average order amount for that custom

113 • SELECT C.CNAME as CUSTNAME, COUNT(*) as no_of_orders,AVG(O.ord_Amt) as AVG_ORDER_AMT FROM CUSTOMER C,
114 ORDERS O WHERE C.cust_no=O.cust_no GROUP BY C.CNAME;

115 /*List the order# for orders that were shipped from all warehouses that the company has in a specific city.*/

116 • SELECT order_no FROM WAREHOUSE W, SHIPMENT S WHERE W.warehouseno=S.warehouseno AND CITY='BANGALORE';

117 /*Demonstrate how you delete item# 10 from the ITEM table and make that field null in the ORDER_ITEM table.*/

118 • delete from ITEM where item_no=5005;

119 • select *from ITEM;

Result Grid | Filter Rows: Edit: Export/Import: Wrap Cell Content:

item_no	unit_price
5001	503
5002	750
5003	150
5004	600
HULL	HULL

Administration Schemas Information

The screenshot shows the MySQL Workbench interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. Below the menu is a toolbar with various icons. The left sidebar, titled 'Navigator', shows the database schema with 'bookdealer', 'database1', 'insurance' (selected), 'Tables', 'Views', 'Stored Procedures', and 'Functions'. The right pane contains a query editor with the following SQL code:

```

115  /*List the order# for orders that were shipped from all warehouses that the company has in a specific city.*/
116 •  SELECT order_no FROM WAREHOUSE W, SHIPMENT S WHERE W.warehouseno=S.warehouseno AND CITY='BANGALORE';
117  /*Demonstrate how you delete item# 10 from the ITEM table and make that field null in the ORDER_ITEM table.*/
118 •  delete from ITEM where item_no=5005;
119 •  select *from ITEM;
120 •  select *from ORDER_ITEM;

```

Below the code is a 'Result Grid' showing the output of the last query:

order_no	item_no	qty
111	5001	50
112	5003	20
113	5002	50
114	NULL	60
115	5004	90
116	5001	10
117	5003	80
118	NULL	50
119	5002	10
120	5004	45

PROGRAM-4:BANKING DATABASE

Question:

Consider the following database for a banking enterprise.

BRANCH (branch-name: String, branch-city: String, assets: real)

ACCOUNTS (accno: int, branch-name: String, balance: real)

DEPOSITOR (customer-name: String, customer-street: String, customer-city: String)

LOAN (loan-number: int, branch-name: String, amount: real)

BORROWER (customer-name: String, loan-number: int)

i) Create the above tables by properly specifying the primary keys and the foreign keys.

ii) Enter at least five tuples for each relation.

iii) Find all the customers who have at least two accounts at the Main branch.

iv) Find all the customers who have an account at all the branches located in a specific city.

- v) Demonstrate how you delete all account tuples at every branch located in a specific city.
- vi) Generate suitable reports.
- vii) Create suitable front end for querying and displaying the results.

PROGRAM CODE:

```
create database banking;
```

```
use banking;
```

```
create table branch(  
branch_name varchar(30) primary key,  
branch_city varchar(30),  
assets real);
```

```
create table accounts(  
accno int primary key,  
branch_name varchar(30),  
balance real,  
foreign key (branch_name) references branch(branch_name) on delete cascade on update  
cascade);
```

```
create table customer(  
customer_name varchar(30) primary key,  
customer_street varchar(20),  
customer_city varchar(20));
```

```
create table depositor(  
customer_name varchar(30),  
accno int,
```

```
primary key(customer_name ,accno),  
foreign key (accno) references accounts(accno) on delete cascade on update cascade,  
foreign key (customer_name) references customer(customer_name) on delete cascade on  
update  
cascade);
```

```
create table loan(  
loan_number int primary key,  
branch_name varchar(30),  
amount real,  
foreign key (branch_name) references branch(branch_name)  
);
```

```
create table borrower (  
customer_name varchar(30),  
loan_number int,  
primary key(customer_name, loan_number),  
foreign key (customer_name) references customer(customer_name) on delete cascade on  
update cascade,  
foreign key (loan_number) references loan(loan_number) on delete cascade on update  
cascade);  
show tables;  
insert into branch(branch_name,branch_city,assets) values  
('A','Bangalore',190000),  
('B','Bangalore',200000),  
('C','Delhi',235344),  
('D','Chennai',1050560),  
('E','Chennai',678909);  
select *from branch;
```

```
insert into accounts(accno,branch_name,balance) VALUES  
(1001,'A',10000),  
(1002,'B',5000),  
(1003,'C',7500),  
(1004,'D',50000),  
(1005,'D',75000),  
(1006,'E',560),  
(1007,"B",500),  
(1008,"B",1500);  
select *from accounts;
```

```
insert into customer(customer_name,customer_street,customer_city) VALUES  
("Ravi","Dasarahalli","Bangalore"),  
("Shyam","Indiranagar","Delhi"),  
("Seema","Vasantnagar","Chennai"),  
("Arpita","Church Street","Bangalore"),  
("Vinay","MG Road","Chennai");  
select *from customer;
```

```
insert into depositor(customer_name,accno) VALUES  
("Ravi",1001),  
("Ravi",1002),  
("Shyam",1003),  
("Seema",1004),  
("Seema",1005),  
("Arpita",1006),  
("Vinay",1007),  
("Vinay",1008);  
select *from depositor;
```

```
insert into loan(loan_number,branch_name,amount) VALUES  
(001,'A',10000),  
(002,'B',25000),  
(003,'B',250000),  
(004,'C',5000),  
(005,'E',90000);  
select *from loan;
```

```
insert into borrower(customer_name,loan_number) VALUES  
("Arpita",001),  
("Ravi",002),  
("Arpita",003),  
("Shyam",004),  
("Vinay",005);  
select *from borrower;
```

/*iii. Find all the customers who have at least two accounts at the Main branch */

```
select customer_name from depositor  
join accounts on depositor.accno = accounts.accno where accounts.branch_name = "D"  
group by depositor.customer_name having count(depositor.customer_name) >=2;
```

/* iv. Find all the customers who have an account at all the branches located in a specific city.*/

```
select customer_name from depositor  
join accounts on accounts.accno = depositor.accno  
join branch on branch.branch_name = accounts.branch_name
```

```

where branch.branch_city = "Bangalore"
GROUP BY depositor.customer_name
having count(DISTINCT branch.branch_name) = (SELECT COUNT(branch_name)
FROM branch
WHERE branch_city = 'Bangalore');

```

*/*v. Demonstrate how you delete all account tuples at every branch located in a specific city.*/*

```

delete from accounts where branch_name in
(select branch_name from branch where branch_city="Delhi");
select *from accounts;

```

OUTPUT SCREENSHOTS:

The screenshot shows the MySQL Workbench interface with the 'BankingDatabase' selected. In the SQL editor, the following query is being run:

```

97  /*iii. Find all the customers who have at least two accounts at the Main branch */
98
99 • select customer_name from depositor
100 join accounts on depositor.acctno = accounts.acctno where accounts.branch_name = "D"
101 group by depositor.customer_name having count(depositor.customer_name) >=2;
102
103 /* iv. Find all the customers who have an account at all the branches located in a specific city.*/
104

```

The result grid shows one row with the customer name 'Seema'.

The screenshot shows the MySQL Workbench interface with the 'BankingDatabase' selected. In the SQL editor, the following query is being run:

```

104
105 • select customer_name from depositor
106 join accounts on accounts.acctno = depositor.acctno
107 join branch on branch.branch_name = accounts.branch_name
108 where branch.branch_city = "Bangalore"
109 GROUP BY depositor.customer_name
110 having count(DISTINCT branch.branch_name) = (SELECT COUNT(branch_name)
111 FROM branch
112 WHERE branch_city = 'Bangalore');
113

```

The result grid shows one row with the customer name 'Ravi'.

The screenshot shows the MySQL Workbench interface with the 'BankingDatabase' selected. In the SQL editor, the following query is being run:

```

111
112 FROM branch
113 WHERE branch_city = 'Bangalore');
114 /*v. Demonstrate how you delete all account tuples at every branch located in a specific city.*/

```

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator Insurance Bookdealer OrderProcessing* BankingDatabase x

SCHEMAS Filter objects

- bookdealer
- database1
- insurance
- order_processing
 - Tables
 - Views
 - Stored Procedures
 - Functions
- sys

111 FROM branch
112 WHERE branch_city = 'Bangalore');

113
114 /*v. Demonstrate how you delete all account tuples at every branch located in a specific city.*/
115 • delete from accounts where branch_name in
116 (select branch_name from branch where branch_city="Delhi");
117 • select *from accounts;

Result Grid Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

accno	branch_name	balance
1001	A	10000
1002	B	5000
1004	D	50000
1005	D	75000
1006	E	560
1007	B	500
1008	B	1500
• NULL	NULL	NULL

Result Grid Result Grid Form Editor

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator Insurance Bookdealer OrderProcessing* BankingDatabase x

SCHEMAS Filter objects

- bookdealer
- database1
- insurance
- order_processing
 - Tables
 - Views
 - Stored Procedures
 - Functions
- sys

39 foreign key (customer_name) references customer(customer_name) on delete cascade on update cascade,
40 foreign key (loan_number) references loan(loan_number) on delete cascade on update cascade;
41 • show tables;
42 • insert into branch(branch_name,branch_city,assets) values
43 ('A','Bangalore',190000),
44 ('B','Bangalore',200000),
45 ('C','Delhi',235344),

Result Grid Filter Rows: | Export: | Wrap Cell Content: |

Tables_in_banking
accounts
borrower
branch
customer
depositor
loan

Result Grid Result Grid Form Editor

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator Insurance Bookdealer OrderProcessing* BankingDatabase x

SCHEMAS Filter objects

- bookdealer
- database1
- insurance
- order_processing
 - Tables
 - Views
 - Stored Procedures
 - Functions
- sys

43 ('A','Bangalore',190000),
44 ('B','Bangalore',200000),
45 ('C','Delhi',235344),
46 ('D','Chennai',1050560),
47 ('E','Chennai',678909);
48 select *from branch;

50 insert into accounts(accno,branch name,balance) VALUES

Result Grid Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

branch_name	branch_cty	assets
A	Bangalore	190000
B	Bangalore	200000
C	Delhi	235344
D	Chennai	1050560
E	Chennai	678909
• NULL	NULL	NULL

Result Grid Result Grid Form Editor

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator Insurance Bookdealer OrderProcessing* BankingDatabase x

SCHEMAS Filter objects

- bookdealer
- database1
- insurance
- order_processing
 - Tables
 - Views
 - Stored Procedures
 - Functions
- sys

55 (1005,'D',75000),
56 (1006,'E',560),
57 (1007,"B",500),
58 (1008,"B",1500);
59 select *from accounts;

60

Result Grid Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

accno	branch_name	balance
1001	A	10000
1002	B	5000
1003	C	7500
1004	D	50000
1005	D	75000
1006	E	560
1007	B	500
1008	B	1500

Result Grid Result Grid Form Editor

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Insurance Bookdealer OrderProcessing* BankingDatabase x

SCHEMAS: Filter objects bookdealer database1 insurance order_processing sys

Limit to 1000 rows

```
55 (1005,'D',75000),
56 (1006,'E',560),
57 (1007,"B",500),
58 (1008,"B",1500);
59 select *from accounts;
60
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Result Grid | Form Editor | Field Types

accno	branch_name	balance
1001	A	10000
1002	B	5000
1003	C	7500
1004	D	50000
1005	D	75000
1006	E	560
1007	B	500
1008	B	1500
NULL	NULL	NULL

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Insurance Bookdealer OrderProcessing* BankingDatabase x

SCHEMAS: Filter objects bookdealer database1 insurance order_processing sys

Limit to 1000 rows

```
61 insert into customer(customer_name,customer_street,customer_city) VALUES
62 ("Ravi","Dasarahalli","Bangalore"),
63 ("Shyam","Indiranagar","Delhi"),
64 ("Seema","Vasantnagar","Chennai"),
65 ("Arpita","Church Street","Bangalore"),
66 ("Vinay","MG Road","Chennai");
67 select *from customer;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Result Grid | Form Editor

customer_name	customer_street	customer_city
Arpita	Church Street	Bangalore
Ravi	Dasarahalli	Bangalore
Seema	Vasantnagar	Chennai
Shyam	Indiranagar	Delhi
Vinay	MG Road	Chennai
NULL	NULL	NULL

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Insurance Bookdealer OrderProcessing* BankingDatabase x

SCHEMAS: Filter objects bookdealer database1 insurance order_processing sys

Limit to 1000 rows

```
76 ("Vinay",1007),
77 ("Vinay",1008);
78 select *from depositor;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Result Grid | Form Editor

customer_name	accno
Ravi	1001
Ravi	1002
Shyam	1003
Seema	1004
Seema	1005
Arpita	1006
Vinay	1007
Vinay	1008
NULL	NULL

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: Insurance Bookdealer OrderProcessing* BankingDatabase x

SCHEMAS: Filter objects bookdealer database1 insurance order_processing sys

Limit to 1000 rows

```
82 (001,'A',10000),
83 (002,'B',25000),
84 (003,'B',250000),
85 (004,'C',5000),
86 (005,'E',90000);
87 select *from loan;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Result Grid | Form

loan_number	branch_name	amount
1	A	10000
2	B	25000
3	B	250000
4	C	5000

The image shows two screenshots of MySQL Workbench. Both screenshots have a top navigation bar with tabs for Insurance, Bookdealer, OrderProcessing, and BankingDatabase. The left sidebar shows the 'Schemas' tree with 'order_processing' selected, revealing tables like 'Tables', 'Views', 'Stored Procedures', and 'Functions'. The right pane displays the results of SQL queries.

Screenshot 1:

```

82 (001,'A',10000),
83 (002,'B',25000),
84 (003,'B',250000),
85 (004,'C',5000),
86 (005,'E',90000);
87 select *from loan;
88

```

Result Grid:

loan_number	branch_name	amount
1	A	10000
2	B	25000
3	B	250000
4	C	5000
5	E	90000
*	HULL	HULL

Screenshot 2:

```

94 ("Vinay",005);
95 • select *from borrower;
96
97 /*iii. Find all the customers who have at least two accounts at the Main branch */
98
99 • select customer_name from depositor
100 join accounts on depositor.acctno = accounts.acctno where accounts.branch_name = "D"
101 group by depositor.customer_name having count(depositor.customer_name) >=2;

```

Result Grid:

customer_name	loan_number
Arpta	1
Ravi	2
Arpta	3
Shyam	4
Vinay	5
*	HULL

PROGRAM 5-STUDENT ENROLLMENT DATABASE

QUESTION:

Consider the following database of student enrollment in courses and books adopted for each course.

STUDENT (regno: String, name: String, major: String, bdate: date)

COURSE (course #: int, cname: String, dept: String)

ENROLL (regno: String, cname: String, sem: int, marks: int)

BOOK_ADOPTION (course #: int, sem: int, book-ISBN: int)

TEXT(book-ISBN:int, book-title: String, publisher:String, author:String)

- Create the above tables by properly specifying the primary keys and the foreign keys.
- Enter at least five tuples for each relation.

iii) Demonstrate how you add a new text book to the database and make this book be adopted by some department.

iv) Produce a list of text books (include Course #, Book-ISBN, Book-title) in the alphabetical order for courses offered by the 'CS' department that use more than two books.

v) List any department that has all its adopted books published by a specific publisher.

PROGRAM CODE:

```
CREATE DATABASE STUDENTENROLLMENT;
```

```
USE STUDENTENROLLMENT;
```

```
CREATE TABLE STUDENT(
    REG_NO VARCHAR(30),
    SNAME VARCHAR(30),
    MAJOR VARCHAR(30),
    BDATE DATE,
    PRIMARY KEY(REG_NO)
);
```

```
CREATE TABLE COURSE(
    COURSE_ID INT,
    CNAME VARCHAR(30),
    DEPT VARCHAR(30),
    PRIMARY KEY(COURSE_ID)
);
```

```
CREATE TABLE ENROLL(
```

```
REG_NO VARCHAR(30),  
COURSE_ID INT,  
SEM INT,  
MARKS INT,  
FOREIGN KEY(REG_NO) REFERENCES STUDENT(REG_NO) ON DELETE CASCADE ON UPDATE  
CASCADE,  
FOREIGN KEY(COURSE_ID) REFERENCES COURSE(COURSE_ID) ON DELETE CASCADE ON  
UPDATE CASCADE  
);
```

```
CREATE TABLE BOOK_ADOPTION(  
COURSE_ID INT,  
SEM INT,  
BOOK_ISBN INT,  
PRIMARY KEY(BOOK_ISBN),  
FOREIGN KEY(COURSE_ID) REFERENCES COURSE(COURSE_ID) ON DELETE CASCADE ON  
UPDATE CASCADE  
);
```

```
CREATE TABLE TEXT(  
BOOK_ISBN INT,  
BOOK_TITLE VARCHAR(30),  
PUBLISHER VARCHAR(30),  
AUTHOR VARCHAR(30),  
FOREIGN KEY(BOOK_ISBN) REFERENCES BOOK_ADOPTION(BOOK_ISBN) ON DELETE  
CASCADE ON UPDATE CASCADE  
);
```

```
show tables;  
INSERT INTO STUDENT(REG_NO, SNAME, MAJOR, BDATE) VALUES ('CS01', 'RAM', 'DS',  
'1986-03-12');
```

```
INSERT INTO STUDENT(REG_NO, SNAME, MAJOR, BDATE) VALUES ('IS02', 'SMITH', 'USP',  
'1987-12-23');  
  
INSERT INTO STUDENT(REG_NO, SNAME, MAJOR, BDATE) VALUES ('EC03', 'AHMED', 'SNS',  
'1985-04-17');  
  
INSERT INTO STUDENT(REG_NO, SNAME, MAJOR, BDATE) VALUES ('CS03', 'SNEHA', 'DBMS',  
'1987-01-01');  
  
INSERT INTO STUDENT(REG_NO, SNAME, MAJOR, BDATE) VALUES ('TC05', 'AKHILA', 'EC',  
'1986-10-06');  
  
SELECT * FROM STUDENT;
```

```
INSERT INTO COURSE(COURSE_ID, CNAME, DEPT) VALUES (11, 'DS', 'CS');  
  
INSERT INTO COURSE(COURSE_ID, CNAME, DEPT) VALUES (22, 'USP', 'IS');  
  
INSERT INTO COURSE(COURSE_ID, CNAME, DEPT) VALUES (33, 'SNS', 'EC');  
  
INSERT INTO COURSE(COURSE_ID, CNAME, DEPT) VALUES (44, 'DBMS', 'CS');  
  
INSERT INTO COURSE(COURSE_ID, CNAME, DEPT) VALUES (55, 'EC', 'TC');  
  
SELECT * FROM COURSE;
```

```
INSERT INTO ENROLL(REG_NO, COURSE_ID, SEM, MARKS) VALUES ('CS01', 11, 4, 85);  
  
INSERT INTO ENROLL(REG_NO, COURSE_ID, SEM, MARKS) VALUES ('IS02', 22, 6, 80);  
  
INSERT INTO ENROLL(REG_NO, COURSE_ID, SEM, MARKS) VALUES ('EC03', 33, 2, 80);  
  
INSERT INTO ENROLL(REG_NO, COURSE_ID, SEM, MARKS) VALUES ('CS03', 44, 6, 75);  
  
INSERT INTO ENROLL(REG_NO, COURSE_ID, SEM, MARKS) VALUES ('TC05', 55, 2, 8);  
  
SELECT * FROM ENROLL;
```

```
INSERT INTO BOOK_ADOPTION(COURSE_ID,SEM,BOOK_ISBN) VALUES (11,4,1);  
  
INSERT INTO BOOK_ADOPTION(COURSE_ID,SEM,BOOK_ISBN) VALUES (11,4,2);  
  
INSERT INTO BOOK_ADOPTION(COURSE_ID,SEM,BOOK_ISBN) VALUES (44,6,3);  
  
INSERT INTO BOOK_ADOPTION(COURSE_ID,SEM,BOOK_ISBN) VALUES (44,6,4);  
  
INSERT INTO BOOK_ADOPTION(COURSE_ID,SEM,BOOK_ISBN) VALUES (55,2,5);  
  
INSERT INTO BOOK_ADOPTION(COURSE_ID,SEM,BOOK_ISBN) VALUES (22,6,6);  
  
INSERT INTO BOOK_ADOPTION(COURSE_ID,SEM,BOOK_ISBN) VALUES (55,2,7);
```

```
SELECT * FROM BOOK_ADOPTION;
```

```
INSERT INTO TEXT(BOOK_ISBN, BOOK_TITLE, PUBLISHER, AUTHOR) VALUES (1, 'DS and C', 'Princeton', 'Padma Reddy');
```

```
INSERT INTO TEXT(BOOK_ISBN, BOOK_TITLE, PUBLISHER, AUTHOR) VALUES (2, 'Fundamentals of DS', 'Princeton', 'Godse');
```

```
INSERT INTO TEXT(BOOK_ISBN, BOOK_TITLE, PUBLISHER, AUTHOR) VALUES (3, 'Fundamentals of DBMS', 'Princeton', 'Navathe');
```

```
INSERT INTO TEXT(BOOK_ISBN, BOOK_TITLE, PUBLISHER, AUTHOR) VALUES (4, 'SQL', 'Princeton', 'Foley');
```

```
INSERT INTO TEXT(BOOK_ISBN, BOOK_TITLE, PUBLISHER, AUTHOR) VALUES (5, 'Electronic circuits', 'TMH', 'Elmasri');
```

```
INSERT INTO TEXT(BOOK_ISBN, BOOK_TITLE, PUBLISHER, AUTHOR) VALUES (6, 'Adv unix prog', 'TMH', 'Stevens');
```

```
SELECT * FROM TEXT;
```

-- Demonstrate how you add a new text book to the database and make this book be adopted by some department.

```
INSERT INTO TEXT VALUES(7, "TREES & GRAPHS", "PRINCETON", "SADGE");
```

```
INSERT INTO BOOK_ADOPTION VALUES(11, 4, 8);
```

```
SELECT * FROM BOOK_ADOPTION;
```

```
SELECT * FROM TEXT;
```

-- Produce a list of text books (include Course #, Book-ISBN, Book-title) in the alphabetical order for courses offered by the 'CS' department that use more than two books.

```
SELECT C.COURSE_ID,T.BOOK_ISBN,T.BOOK_TITLE FROM TEXT T,COURSE  
C,BOOK_ADOPTION B WHERE T.BOOK_ISBN=B.BOOK_ISBN AND  
B.COURSE_ID=C.COURSE_ID AND C.DEPT="CS" AND (SELECT COUNT(B.BOOK_ISBN) FROM  
BOOK_ADOPTION B WHERE  
C.COURSE_ID=B.COURSE_ID)>=2 ORDER BY T.BOOK_TITLE;
```

-- List any department that has all its adopted books published by a specific publisher.

```
SELECT DISTINCT C.DEPT  
FROM COURSE C  
WHERE C.DEPT IN  
( SELECT C.DEPT  
FROM COURSE C,BOOK_ADOPTION B,TEXT T  
WHERE C.COURSE_ID=B.COURSE_ID  
AND T.BOOK_ISBN=B.BOOK_ISBN  
AND T.PUBLISHER='Princeton')  
AND C.DEPT NOT IN  
(SELECT C.DEPT  
FROM COURSE C,BOOK_ADOPTION B,TEXT T  
WHERE C.COURSE_ID=B.COURSE_ID  
AND T.BOOK_ISBN=B.BOOK_ISBN  
AND T.PUBLISHER != 'Princeton');
```

OUTPUT SCREENSHOTS:

The screenshot shows the MySQL Workbench interface. The top navigation bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. Below the bar is a toolbar with various icons. The main area is divided into several panes: a Navigator pane on the left listing Schemas (banking, bookdealer, database1, insurance, order_processing) and Tables (customer, item, order_item, orders, shipment, warehouse); a SQL pane containing the query code; and a Result Grid pane at the bottom displaying the results of the executed query. The results show a list of book adoptions with columns: COURSE_ID, SEM, and BOOK_ISBN.

COURSE_ID	SEM	BOOK_ISBN
11	4	1
11	4	2
44	6	3
44	6	4
55	2	5
22	6	6
55	2	7
11	4	8
NULL	NULL	NULL

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- Filter objects
- banking
- bookdealer
- database1
- insurance
- order_processing
- Tables
- customer
- item
- order_item
- orders
- shipment
- warehouse
- Views
- Stored Procedures
- Functions
- sys

Administration Schemas Information

OrderProcessing BankingDatabase StudentEnrollment*

```

86 • INSERT INTO TEXT VALUES(7, "TREES & GRAPHS", "PRINCETON", "SADGE");
87 • INSERT INTO BOOK_ADOPTION VALUES(11, 4, 8);
88
89 • SELECT * FROM BOOK_ADOPTION;
90
91 • SELECT * FROM TEXT;
92
93 -- Produce a list of text books (include Course #, Book-ISBN, Book-title) in the alphabetical order for courses offered by t

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Result Grid | Form Editor

BOOK_ISBN	BOOK_TITLE	PUBLISHER	AUTHOR
1	DS and C	Princeton	Padma Reddy
2	Fundamentals of DS	Princeton	Gode
3	Fundamentals of DBMS	Princeton	Navathe
4	SQL	Princeton	Foley
5	Electronic circuits	TMH	Elmasri
6	Adv unix prog	TMH	Stevens
7	TREES & GRAPHS	PRINCETON	SADGE

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- Filter objects
- banking
- bookdealer
- database1
- insurance
- order_processing
- Tables
- customer
- item
- order_item
- orders
- shipment
- warehouse
- Views
- Stored Procedures
- Functions
- sys

Administration Schemas Information

OrderProcessing BankingDatabase StudentEnrollment*

```

89 • SELECT * FROM BOOK_ADOPTION;
90
91 • SELECT * FROM TEXT;
92
93 -- Produce a list of text books (include Course #, Book-ISBN, Book-title) in the alphabetical order for courses offered by t
94 • SELECT C.COURSE_ID,T.BOOK_ISBN,T.BOOK_TITLE FROM TEXT T,COURSE C,BOOK_ADOPTION B WHERE T.BOOK_ISBN=B.BOOK_ISBN AND
95 B.COURSE_ID=C.COURSE_ID AND C.DEP="CS" AND (SELECT COUNT(B.BOOK_ISBN) FROM BOOK_ADOPTION B WHERE
96 C.COURSE_ID=B.COURSE_ID)>=2 ORDER BY T.BOOK_TITLE;
97
98

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Result Grid | Form Editor

COURSE_ID	BOOK_ISBN	BOOK_TITLE
11	1	DS and C
44	3	Fundamentals of DBMS
11	2	Fundamentals of DS
44	4	SQL

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- Filter objects
- banking
- bookdealer
- database1
- insurance
- order_processing
- Tables
- customer
- item
- order_item
- orders
- shipment
- warehouse
- Views
- Stored Procedures
- Functions
- sys

Administration Schemas Information

OrderProcessing BankingDatabase StudentEnrollment*

```

103 ( SELECT C.DEP
104 FROM COURSE C,BOOK_ADOPTION B,TEXT T
105 WHERE C.COURSE_ID=B.COURSE_ID
106 AND T.BOOK_ISBN=B.BOOK_ISBN
107 AND T.PUBLISHER='Princeton')
108 AND C.DEP NOT IN
109 (SELECT C.DEP
110 FROM COURSE C,BOOK_ADOPTION B,TEXT T
111 WHERE C.COURSE_ID=B.COURSE_ID
112 AND T.BOOK_ISBN=B.BOOK_ISBN
113 AND T.PUBLISHER != 'Princeton');

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Result Grid | Form Editor

DEPT
CS

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator: OrderProcessing BankingDatabase StudentEnrollment ×

SCHEMAS Filter objects

- banking
- bookdealer
- database1
- insurance
- order_processing
- Tables
 - customer
 - item
 - order_item
 - orders
 - shipment
 - warehouse
- Views
- Stored Procedures
- Functions
- sys

Administration Schemas Information

```

55 • INSERT INTO COURSE(COURSE_ID, CNAME, DEPT) VALUES (22, 'USP', 'IS');
56 • INSERT INTO COURSE(COURSE_ID, CNAME, DEPT) VALUES (33, 'SNS', 'EC');
57 • INSERT INTO COURSE(COURSE_ID, CNAME, DEPT) VALUES (44, 'DBMS', 'CS');
58 • INSERT INTO COURSE(COURSE_ID, CNAME, DEPT) VALUES (55, 'EC', 'TC');
59 • SELECT * FROM COURSE;
60
61 • INSERT INTO ENROLL(REG_NO, COURSE_ID, SEM, MARKS) VALUES ('CS01', 11, 4, 85);
62 • INSERT INTO ENROLL(REG_NO, COURSE_ID, SEM, MARKS) VALUES ('IS02', 22, 6, 80);
63 • INSERT INTO ENROLL(REG_NO, COURSE_ID, SEM, MARKS) VALUES ('EC03', 33, 2, 80);

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Result Grid | Form Editor

COURSE_ID	CNAME	DEPT
11	DS	CS
22	USP	IS
33	SNS	EC
44	DBMS	CS
55	EC	TC
NULL	NULL	NULL

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator: OrderProcessing BankingDatabase StudentEnrollment ×

SCHEMAS Filter objects

- banking
- bookdealer
- database1
- insurance
- order_processing
- Tables
 - customer
 - item
 - order_item
 - orders
 - shipment
 - warehouse
- Views
- Stored Procedures
- Functions
- sys

Administration Schemas Information

```

39 • CREATE TABLE TEXT(
    BOOK_ISBN INT,
    BOOK_TITLE VARCHAR(30),
    PUBLISHER VARCHAR(30),
    AUTHOR VARCHAR(30),
    FOREIGN KEY(BOOK_ISBN) REFERENCES BOOK_ADOPTION(BOOK_ISBN) ON DELETE CASCADE ON UPDATE CASCADE
);
46 • show tables;
47 • INSERT INTO STUDENT(REG_NO, SNAME, MAJOR, BDATE) VALUES ('CS01', 'RAM', '1986-03-12');

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Result Grid | Form Editor

Tables_in_studentenrollment
book_adoption
course
enroll
student
text

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator: OrderProcessing BankingDatabase StudentEnrollment ×

SCHEMAS Filter objects

- banking
- bookdealer
- database1
- insurance
- order_processing
- Tables
 - customer
 - item
 - order_item
 - orders
 - shipment
 - warehouse
- Views
- Stored Procedures
- Functions
- sys

Administration Schemas Information

```

49 • INSERT INTO STUDENT(REG_NO, SNAME, MAJOR, BDATE) VALUES ('EC03', 'AHMED', 'SNS', '1985-04-17');
50 • INSERT INTO STUDENT(REG_NO, SNAME, MAJOR, BDATE) VALUES ('CS03', 'SNEHA', 'DBMS', '1987-01-01');
51 • INSERT INTO STUDENT(REG_NO, SNAME, MAJOR, BDATE) VALUES ('TC05', 'AKHILA', 'EC', '1986-10-06');
52 • SELECT * FROM STUDENT;
53
54 • INSERT INTO COURSE(COURSE_ID, CNAME, DEPT) VALUES (11, 'DS', 'CS');
55 • INSERT INTO COURSE(COURSE_ID, CNAME, DEPT) VALUES (22, 'USP', 'IS');
56 • INSERT INTO COURSE(COURSE_ID, CNAME, DEPT) VALUES (33, 'SNS', 'EC');
57 • INSERT INTO COURSE(COURSE_ID, CNAME, DEPT) VALUES (44, 'DBMS', 'CS');

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Result Grid | Form Editor

REG_NO	SNAME	MAJOR	BDATE
CS01	RAM	DS	1986-03-12
CS03	SNEHA	DBMS	1987-01-01
EC03	AHMED	SNS	1985-04-17
IS02	SMITH	USP	1987-12-23
TC05	AKHILA	EC	1986-10-06
NULL	NULL	NULL	NULL

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- Filter objects
- banking
- bookdealer
- database1
- insurance
- order_processing
- Tables
- customer
- item
- order_item
- orders
- shipment
- warehouse
- Views
- Stored Procedures
- Functions
- sys

OrderProcessing BankingDatabase StudentEnrollment*

```

58 • INSERT INTO COURSE(COURSE_ID, CNAME, DEPT) VALUES (55, 'EC', 'TC');
59 • SELECT * FROM COURSE;
60
61 • INSERT INTO ENROLL(REG_NO, COURSE_ID, SEM, MARKS) VALUES ('CS01', 11, 4, 85);
62 • INSERT INTO ENROLL(REG_NO, COURSE_ID, SEM, MARKS) VALUES ('IS02', 22, 6, 80);
63 • INSERT INTO ENROLL(REG_NO, COURSE_ID, SEM, MARKS) VALUES ('EC03', 33, 2, 80);
64 • INSERT INTO ENROLL(REG_NO, COURSE_ID, SEM, MARKS) VALUES ('CS03', 44, 6, 75);
65 • INSERT INTO ENROLL(REG_NO, COURSE_ID, SEM, MARKS) VALUES ('TC05', 55, 2, 8);
66 • SELECT * FROM ENROLL;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Result Grid | Form Editor

REG_NO	COURSE_ID	SEM	MARKS
CS01	11	4	85
IS02	22	6	80
EC03	33	2	80
CS03	44	6	75
TC05	55	2	8

Administration Schemas Information

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- Filter objects
- banking
- bookdealer
- database1
- insurance
- order_processing
- Tables
- customer
- item
- order_item
- orders
- shipment
- warehouse
- Views
- Stored Procedures
- Functions
- sys

OrderProcessing BankingDatabase StudentEnrollment*

```

70 • INSERT INTO BOOK_ADOPTION(COURSE_ID, SEM, BOOK_ISBN) VALUES (44, 6, 3);
71 • Execute the selected portion of the script or everything, if there is no selection.
72 • INSERT INTO BOOK_ADOPTION(COURSE_ID, SEM, BOOK_ISBN) VALUES (44, 6, 4);
73 • INSERT INTO BOOK_ADOPTION(COURSE_ID, SEM, BOOK_ISBN) VALUES (55, 2, 5);
74 • INSERT INTO BOOK_ADOPTION(COURSE_ID, SEM, BOOK_ISBN) VALUES (22, 6, 6);
75 • INSERT INTO BOOK_ADOPTION(COURSE_ID, SEM, BOOK_ISBN) VALUES (55, 2, 7);
76
77 • INSERT INTO TEXT(BOOK_ISBN, BOOK_TITLE, PUBLISHER, AUTHOR) VALUES (1, 'DS and C', 'Princeton', 'Padma Reddy');
78 • INSERT INTO TEXT(BOOK_ISBN, BOOK_TITLE, PUBLISHER, AUTHOR) VALUES (2, 'Fundamentals of DS', 'Princeton', 'Godse');

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Result Grid | Form Editor

COURSE_ID	SEM	BOOK_ISBN
11	4	1
11	4	2
44	6	3
44	6	4
55	2	5
22	6	6
55	2	7
HULL	HULL	HULL

BOOK_ADOPTION 5 x

Apply | Revert

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- Filter objects
- banking
- bookdealer
- database1
- insurance
- order_processing
- Tables
- customer
- item
- order_item
- orders
- shipment
- warehouse
- Views
- Stored Procedures
- Functions
- sys

OrderProcessing BankingDatabase StudentEnrollment*

```

79 • INSERT INTO TEXT(BOOK_ISBN, BOOK_TITLE, PUBLISHER, AUTHOR) VALUES (3, 'Fundamentals of DBMS', 'Princeton', 'Navathe');
80 • INSERT INTO TEXT(BOOK_ISBN, BOOK_TITLE, PUBLISHER, AUTHOR) VALUES (4, 'SQL', 'Princeton', 'Foley');
81 • INSERT INTO TEXT(BOOK_ISBN, BOOK_TITLE, PUBLISHER, AUTHOR) VALUES (5, 'Electronic circuits', 'TMH', 'Elmasri');
82 • INSERT INTO TEXT(BOOK_ISBN, BOOK_TITLE, PUBLISHER, AUTHOR) VALUES (6, 'Adv unix prog', 'TMH', 'Stevens');
83 • SELECT * FROM TEXT;
84
85 -- Demonstrate how you add a new text book to the database and make this book be adopted by some department.
86 • INSERT INTO TEXT VALUES(7, "TREES & GRAPHS", "PRINCETON", "SADGE");
87 • INSERT INTO BOOK_ADOPTION VALUES(11, 4, 8);

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Result Grid | Form Editor

BOOK_ISBN	BOOK_TITLE	PUBLISHER	AUTHOR
1	DS and C	Princeton	Padma Reddy
2	Fundamentals of DS	Princeton	Godse
3	Fundamentals of DBMS	Princeton	Navathe
4	SQL	Princeton	Foley
5	Electronic circuits	TMH	Elmasri
6	Adv unix prog	TMH	Stevens

Administration Schemas Information