

NAME : KARTHIK S

USN : 1BM19CS070

FINAL REPORT

LAB INTERNALS-1

NAME : KARTHIK S

USN : 1BM19CS070

DATE : 28/04/21

DBMS LAB PROGRAM – 1

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 THE PROGRAM OUTPUT :

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator Insurance ×

SCHEMAS

Filter objects

database1

Tables

Views

Stored Procedures

Functions

insurance

Tables

Views

Stored Procedures

Functions

sys

Tables

sys_config

Views

Stored Procedures

Functions

Result Grid Filter Rows: Export: Wrap Cell Content:

Tables_in_insurance

accident

car

owns

participated

person

Result Grid Form Editor Field

```
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);
```

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator Insurance ×

SCHEMAS

Filter objects

database1

Tables

Views

Stored Procedures

Functions

insurance

Tables

Views

Stored Procedures

Functions

sys

Tables

sys_config

Views

Stored Procedures

Functions

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

PERSON

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

Administration Schemas

PERSON

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator Insurance

SCHEMAS

- database1
- insurance
- sys

```

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

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

Result Grid Form Editor Field Types

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator Insurance

SCHEMAS

- database1
- insurance
- sys

```

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 | 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
25000	2000-06-11	RESIDENCY ROAD
26500	2001-08-12	RICHMOND ROAD
NULL	NULL	NULL

Result Grid Form Editor Field Types

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator Insurance*

SCHEMAS

Filter objects

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

Limit to 1000 rows

```

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 | Filter Rows: [] | Edit: [] | Export/Import: [] | Wrap Cell Content: []

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

Administration Schemas

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator Insurance*

SCHEMAS

Filter objects

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

Limit to 1000 rows

```

46 •   SELECT *FROM owns;
47 •   insert into PARTICIPATED(driver_id,regno,report_no,damage_amt)values('1111', 'KA04Q2301', '12', '20000');
48 •   insert into PARTICIPATED(driver_id,regno,report_no,damage_amt)values('2222', 'KA03L1234', '200', '500');
49 •   insert into PARTICIPATED(driver_id,regno,report_no,damage_amt)values('3333', 'KA03L9999', '300', '10000');
50 •   insert into PARTICIPATED(driver_id,regno,report_no,damage_amt)values('4444', 'KA01P4020', '25000', '2375');
51 •   insert into PARTICIPATED(driver_id,regno,report_no,damage_amt)values('1111', 'KA05P1000', '26500', '70000');
52 •   COMMIT;
53 •   desc PARTICIPATED ;
54 •   SELECT *FROM PARTICIPATED;
55

```

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

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

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

```

50 • insert into PARTICIPATED(driver_id,regno,report_no,damage_amt)values('4444','KA01P4020','25000 ','2375');
51 • insert into PARTICIPATED(driver_id,regno,report_no,damage_amt)values('1111','KA05P1000','26500','70000');
52 • COMMIT;
53 • desc PARTICIPATED ;
54 • SELECT *FROM PARTICIPATED;
55 • UPDATE PARTICIPATED SET DAMAGE_AMT=25000 WHERE REPORT_NO =12 AND REGNO='KA04Q2301';
56 • COMMIT;
57 • desc PARTICIPATED ;
58 • SELECT *FROM PARTICIPATED;
59

```

driver_id	regno	report_no	damage_amt
1111	KA04Q2301	12	25000
2222	KA03L1234	200	500
3333	KA03.9999	300	10000
4444	KA01P4020	25000	2375
1111	KA05P1000	26500	70000

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

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

```

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

```

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

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

Result Grid

count(*)

0

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

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

Result Grid

COUNT(A.REPORT_NO)

1

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

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';

Result Grid

Filter Rows: Export: Wrap Cell Content:

COUNT(A.REPORT_NO)

1

**** END OF WEEK1 PROGRAM ****

NAME : KARTHIK S

USN : 1BM19CS070

DATE : 05/05/21

DBMS LAB PROGRAM – 2

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)
);
show tables;
desc PUBLISHER;
SELECT *FROM PUBLISHER;
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,'Digit
al 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,'Serve
r 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;
/*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*/
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;
/*Find the author of the book which has maximum sales.*/
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));
/*Demonstrate how you increase the price of books published by a specific publisher
by 10%.*/
update CATALOG set price=1.1*price where publisher_id in(select publisher_id from
PUBLISHER where name='PEARSON');
select *from CATALOG;

```

SCREENSHOT OF THE PROGRAM OUTPUT :

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

```

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');

```

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

Tables_in_bookdealer			
	author	catalog	category
▶	author		
	catalog		
	category		
	order_details		
	publisher		

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

```

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');

```

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

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	NULL

Administration Schemas

Result Grid

Form Editor

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: Insurance Bookdealer

SCHEMAS

Filter objects

database1 insurance

Tables Views Stored Procedures Functions

sys

Limit to 1000 rows

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: |

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	NULL

Administration Schemas

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: Insurance Bookdealer

SCHEMAS

Filter objects

database1 insurance

Tables Views Stored Procedures Functions

sys

Limit to 1000 rows

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: |

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

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

```

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: |

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
• HULL	HULL	HULL	HULL	HULL	HULL	HULL

Result Grid Form Editor Field Types

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

```

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: |

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

Result Grid Form Editor Field Types

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

Limit to 1000 rows

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 AUTHOR.author_id;
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

Limit to 1000 rows

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 AUTHOR.author_id;
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 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

```

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 x

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
		NULL	NULL	NULL	NULL	NULL

Administration Schemas

**** END OF WEEK 2 PROGRAM ****

NAME : KARTHIK S

USN : 1BM19CS070

DATE : 05/05/21

DBMS LAB PROGRAM – 3

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
);
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;

```

SCREENSHOT OF THE PROGRAM OUTPUT :

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator Insurance Bookdealer OrderProcessing x

SCHEMAS bookdealer database1 insurance sys

```

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');

Tables_in_order_processing
customer
item
order_item
orders
shipment
warehouse

```

Administration Schemas Information

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator Insurance Bookdealer OrderProcessing x

SCHEMAS bookdealer database1 insurance sys

```

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);

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

```

Administration Schemas

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator Insurance Bookdealer OrderProcessing

SCHEMAS Filter objects

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

Limit to 1000 rows

```
59 • insert into ORDERS(order_no,odate,cust_no,ord_Amt)values(119,'13-02-05',774,29000);
60 • insert into ORDERS(order_no,odate,cust_no,ord_Amt)values(120,'13-10-05',775,29000);
61 • COMMIT;
62 • desc ORDERS;
63 • SELECT *FROM ORDERS;
64 • insert into ITEM(item_no,unit_price)values(5001,503);
```

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

order_no	odate	cust_no	ord_Amt
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

Administration Schemas

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator Insurance Bookdealer OrderProcessing

SCHEMAS Filter objects

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

Limit to 1000 rows

```
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);
```

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

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

Administration Schemas

Information

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator: Insurance Bookdealer OrderProcessing ×

SCHEMAS

Filter objects

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

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

Filter objects

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
*	NULL

Administration Schemas Information

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

Filter objects

- 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

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator Insurance Bookdealer OrderProcessing*

SCHEMAS

Filter objects

- 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,

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

114

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

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator Insurance Bookdealer OrderProcessing*

```

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: | Result Grid

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*

```

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;

```

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

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

Administration Schemas Information

**** END OF WEEK3 PROGRAM ****

NAME : KARTHIK S

USN : 1BM19CS070

DATE : 12/05/21

DBMS LAB PROGRAM – 4

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;
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;

```

SCREENSHOTS OF THE PROGRAM OUTPUT :

The screenshot shows the MySQL Workbench interface with the following details:

- SQL Editor:** Contains the following SQL code:

```

53   (1083,'C',7500),
54   (1084,'D',50000),
55   (1085,'D',75000),
56   (1086,'B',560),
57   (1087,'B',500),
58   (1088,"B",1500);
59 • select *from accounts;
60
61 • insert into customer(customer_name,customer_street,customer_city) VALUES

```
- Result Grid:** Displays a table with columns `accno`, `branch_name`, and `balance`. The data is:

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
- Session History:** Shows the execution history with actions like `select`, `insert`, and `delete` statements along with their execution time and message.
- Toolbar:** Includes standard MySQL Workbench icons for file operations, database management, and scripting.

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- book_dealer
 - Tables
 - Views
 - Stored Procedures
 - Functions
- order_processing
 - Tables
 - Views
 - Stored Procedures
 - Functions
- sys

SQL Editor

order_processing SQL File 11*

```
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;
68
69 • insert into depositor(customer_name,accno) VALUES
```

Result Grid

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

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Output

Action Output

#	Time	Action	Message	Duration / Fetch
22	17:46:42	select customer_name from depositor join accounts on depositor.accno = accounts.accno where accounts.bra...	1 row(s) returned	0.000 sec / 0.000 sec
23	17:46:42	select customer_name from depositor join accounts on accounts accno = depositor accno join branch on bran...	1 row(s) returned	0.000 sec / 0.000 sec
24	17:46:42	delete from accounts where branch_name in (select branch_name from branch where branch_city='Delhi')	1 row(s) affected	0.000 sec
25	17:46:42	select *from accounts LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec
26	17:50:15	select *from accounts LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec
27	17:51:12	select *from customer LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec

MySQL Workbench

Local instance MySQL80

Navigator:

- SCHEMAS
 - book_dealer
 - Tables
 - Views
 - Stored Procedures
 - Functions
 - order_processing
 - sys

SQL Editor: SQL File 11*

```

45 ('C','Delhi',235344),
46 ('D','Chennai',1050560),
47 ('E','Chennai',678900);
48 • select *from branch;
49
50 • insert into accounts(acno,branch_name,balance) VALUES
51 (1001,'A',10000),
52 (1002,'B',5000),
53 (1003,'C',7500),

```

Result Grid:

branch_name	branch_city	assets
A	Bangalore	190000
B	Bangalore	200000
C	Delhi	235344
D	Chennai	1050560
E	Chennai	678900

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
23	17:46:42	select customer_name from depositor join accounts on accounts.acno = depositor.acno join branch on branch.branch_name = accounts.branch_name where branch_name in ('A')	1 row(s) returned	0.000 sec / 0.000 sec
24	17:46:42	delete from accounts where branch_name in (select branch_name from branch where branch_city='Delhi')	1 row(s) affected	0.000 sec
25	17:46:42	select *from accounts LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec
26	17:50:15	select *from accounts LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec
27	17:51:12	select *from customer LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
28	17:52:40	select *from branch LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

MySQL Workbench

Local instance MySQL80

Navigator:

- SCHEMAS
 - book_dealer
 - Tables
 - Views
 - Stored Procedures
 - Functions
 - order_processing
 - sys

SQL Editor: SQL File 11*

```

77 ("Vinay",1000)
78 • select *from depositor;
79
80
81 • insert into loan(loan_number,branch_name,amount) VALUES
82 (001,'A',10000),
83 (002,'B',25000),
84 (003,'B',250000),
85 (004,'C',5000),

```

Result Grid:

customer_name	acno
Ravi	1001
Ravi	1002
Seema	1004
Seema	1005
Arlita	1006
Viney	1007
Viney	1008

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
24	17:46:42	delete from accounts where branch_name in (select branch_name from branch where branch_city='Delhi')	1 row(s) affected	0.000 sec
25	17:46:42	select *from accounts LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec
26	17:50:15	select *from accounts LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec
27	17:51:12	select *from customer LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
28	17:52:40	select *from branch LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
29	17:53:36	select *from depositor LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

MySQL Workbench

Local instance MySQL80

Navigator:

- SCHEMAS
 - book_dealer
 - Tables
 - Views
 - Stored Procedures
 - Functions
 - order_processing
 - sys

SQL Editor: SQL File 11*

```

85   ('004','C',5000),
86   ('005','E',80000);
87 •   select *from loan;
88
89 •   insert into borrower(customer_name,loan_number) VALUES
90   ("Arpit",001),
91   ("Ravi",002),
92   ("Arpit",003),
93   ("Shyam",004),

```

Result Grid:

loan_number	branch_name	amount
1	A	10000
2	B	25000
3	B	250000
4	C	50000
5	E	90000
MAX	MAX	MAX

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
25	17:46:42	select *from accounts LIMIT 0, 1000	7row(s) returned	0.000 sec / 0.000 sec
26	17:50:15	select *from accounts LIMIT 0, 1000	7row(s) returned	0.000 sec / 0.000 sec
27	17:51:12	select *from customer LIMIT 0, 1000	5row(s) returned	0.000 sec / 0.000 sec
28	17:52:40	select *from branch LIMIT 0, 1000	5row(s) returned	0.000 sec / 0.000 sec
29	17:53:36	select *from depositor LIMIT 0, 1000	7row(s) returned	0.000 sec / 0.000 sec
30	17:54:08	select *from loan LIMIT 0, 1000	5row(s) returned	0.000 sec / 0.000 sec

Object Info Session

MySQL Workbench

Local instance MySQL80

Navigator:

- SCHEMAS
 - book_dealer
 - Tables
 - Views
 - Stored Procedures
 - Functions
 - order_processing
 - sys

SQL Editor: SQL File 11*

```

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 = "0"
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
105 •   select customer_name from depositor

```

Result Grid:

customer_name
Seema

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
27	17:51:12	select *from customer LIMIT 0, 1000	5row(s) returned	0.000 sec / 0.000 sec
28	17:52:40	select *from branch LIMIT 0, 1000	5row(s) returned	0.000 sec / 0.000 sec
29	17:53:36	select *from depositor LIMIT 0, 1000	7row(s) returned	0.000 sec / 0.000 sec
30	17:54:08	select *from loan LIMIT 0, 1000	5row(s) returned	0.000 sec / 0.000 sec
31	17:54:34	select *from borrower LIMIT 0, 1000	5row(s) returned	0.000 sec / 0.000 sec
32	17:56:07	select customer_name from depositor join accounts on depositor.acctno = accounts.acctno where accounts.bra...	1row(s) returned	0.000 sec / 0.000 sec

Object Info Session

MySQL Workbench

Local instance MySQL80

SQL File 11*

```

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')
  
```

Result Grid

customer_name
Ravi

Result 18

Action Output

#	Time	Action	Message	Duration / Fetch
28	17:52:40	select from branch LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
29	17:53:36	select from depositor LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec
30	17:54:08	select from loan LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
31	17:54:34	select from borrower LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
32	17:56:07	select customer_name from depositor join accounts on depositor.acctno = accounts.acctno where accounts.bra...	1 row(s) returned	0.000 sec / 0.000 sec
33	18:01:18	select customer_name from depositor join accounts on depositor.acctno = depositor.acctno join branch on bran...	1 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Type here to search

18:01 21-05-2021 ENG

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

MySQL Workbench

Local instance MySQL80

SQL File 11*

```

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');

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

acctno	branch_name	balance
1001	A	10000
1002	B	5000
1004	D	50000
1005	D	75000
1006	E	500
1007	B	500
1008	B	1500
1009	NULL	NULL

accounts 19

Action Output

#	Time	Action	Message	Duration / Fetch
29	17:53:36	select from depositor LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec
30	17:54:08	select from loan LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
31	17:54:34	select from borrower LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
32	17:56:07	select customer_name from depositor join accounts on depositor.acctno = accounts.acctno where accounts.bra...	1 row(s) returned	0.000 sec / 0.000 sec
33	18:01:18	select customer_name from depositor join accounts on accounts.acctno = depositor.acctno join branch on bran...	1 row(s) returned	0.000 sec / 0.000 sec
34	18:02:07	select from accounts LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Type here to search

18:02 21-05-2021 ENG

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: Local instance MySQL80

SCHEMAS: book_dealer, order_processing, sys

SQL Editor: SQL File 11* (order_processing)

```

37  loan_number int,
38  primary key(customer_name, loan_number),
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',150000),
44  ('B','Bangalore',200000),
45  ('C','Delhi',235344),

```

Result Grid: Tables_in_branch

	Tables_in_branch
▶	accounts
▶	borrower
▶	branch
▶	customer
▶	depositor
▶	loan

Result 20: Action Output

#	Time	Action	Message	Duration / Fetch
30	17:54:08	select from loan LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
31	17:54:34	select from borrower LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
32	17:56:07	select customer_name from depositor join accounts on depositor.acctno = accounts.acctno where accounts.bra...	1 row(s) returned	0.000 sec / 0.000 sec
33	18:01:18	select customer_name from depositor join accounts on accounts.acctno = depositor.acctno join branch on bran...	1 row(s) returned	0.000 sec / 0.000 sec
34	18:02:07	select from accounts LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec
35	18:02:42	show tables	6 row(s) returned	0.015 sec / 0.000 sec

Object Info Session

Type here to search

**** END OF WEEK4 PROGRAM ****

NAME : KARTHIK S

USN : 1BM19CS070

DATE : 26/05/21

DBMS LAB PROGRAM – 5

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)

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 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 STUDENT_ENROLLMENT;
USE STUDENT_ENROLLMENT;
CREATE TABLE STUDENT(REGNO VARCHAR(10), NAME VARCHAR(30), MAJOR VARCHAR(10), BDATE DATE, PRIMARY KEY(REGNO));
CREATE TABLE COURSE(COURSE_NO INT, CNAME VARCHAR(30), DEPT VARCHAR(4), PRIMARY KEY(COURSE_NO));
CREATE TABLE ENROLL(REGNO VARCHAR(10), COURSE_NO INT, SEM INT, MARKS INT, FOREIGN KEY(REGNO) REFERENCES STUDENT(REGNO) ON DELETE CASCADE ON UPDATE CASCADE, FOREIGN KEY(COURSE_NO) REFERENCES COURSE(COURSE_NO) ON DELETE CASCADE ON UPDATE CASCADE);
CREATE TABLE TEXTBOOK(ISBN INT, TITLE VARCHAR(30), PUBLISHER VARCHAR(30), AUTHOR VARCHAR(30), PRIMARY KEY(ISBN));
CREATE TABLE ADOPTION(COURSE_NO INT, SEM INT, ISBN INT, FOREIGN KEY(COURSE_NO)
```

```

REFERENCES COURSE(COURSE_NO) ON DELETE CASCADE ON UPDATE CASCADE, FOREIGN KEY(ISBN)
REFERENCES TEXTBOOK(ISBN) ON DELETE CASCADE ON UPDATE CASCADE);
INSERT INTO STUDENT VALUES("CS01", "RAM" , "DS" , "1986-03-12"),
("IS02" , "SMITH" , "USP" , "1987-12-23"),
("EC03" , "AHMED" , "SNS" , "1985-04-17"),
("CS03" , "SNEHA" , "DBMS" , "1987-01-01"),
("TC05" , "AKHILA" , "EC" , "1986-10-06");
INSERT INTO COURSE VALUES(11 , "DS" , "CS"),
(22 , "USP" , "IS"),
(33 , "SNS" , "EC"),
(44 , "DBMS" , "CS"),
(55 , "EC" , "TC");
INSERT INTO ENROLL VALUES("CS01" , 11 , 4 , 85),
("IS02" , 22, 6, 80),
("EC03", 33, 2, 80),
("CS03", 44, 6, 75),
("TC05", 55, 2, 8);
INSERT INTO TEXTBOOK VALUES(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");
INSERT INTO ADOPTION VALUES(11 , 4 , 1),
(11 , 4 , 2),
(44 , 6 , 3),
(44 , 6 , 4),
(55 , 2 , 5),
(22, 6, 6);
#Demonstrate how you add a new text book to the database and make this book be
adopted by some department.
INSERT INTO TEXTBOOK VALUES(7, "Operating System Concepts", "Wiley", "Silberschatz-
Galvin-Gagne");
INSERT INTO ADOPTION VALUES(55, 2, 7);
SELECT * FROM TEXTBOOK;
/*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_NO , T.ISBN, T.TITLE FROM COURSE C, TEXTBOOK T, ADOPTION A WHERE
C.DEPT = "CS" AND C.COURSE_NO = A.COURSE_NO AND A.ISBN = T.ISBN;
/*List any department that has all its adopted books published by a specific
publisher.*/

```

```

SELECT DISTINCT C.DEPARTMENT FROM COURSE C
WHERE C.DEPARTMENT IN (SELECT C.DEPARTMENT FROM COURSE C, TEXTBOOK T, ADOPTION A
WHERE T.PUBLISHER = "PRINCETON" AND C.COURSE_NO = A.COURSE_NO AND A.ISBN = T.ISBN)
AND C.DEPARTMENT NOT IN (SELECT C.DEPARTMENT FROM COURSE C, TEXTBOOK T, ADOPTION A
WHERE T.PUBLISHER != "PRINCETON" AND C.COURSE_NO = A.COURSE_NO AND A.ISBN =
T.ISBN);

```

SCREENSHOTS OF PROGRAM OUTPUT :

This screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** Local instance MySQL80, File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Toolbar:** Standard MySQL Workbench toolbar.
- Navigator:** Schemas (banking, banking_enterprise, bms_db, book_dealer, book_dealer_database, bookdealer, books_adopted_db, books_insurance, sys) and Administration tab.
- SQL Editor:** Titled "STUDENT ENROLLMENT", containing the following SQL code:


```

10 • INSERT INTO STUDENT VALUES("CS01", "RAH", "DS", "1986-03-12"),
11 ("IS02", "SMITH", "USP", "1987-12-23"),
12 ("EC03", "AHMED", "SNS", "1985-04-17"),
13 ("CS03", "SNEHA", "DBNIS", "1987-01-01"),
14 ("TC05", "AKHILA", "EC", "1986-10-06");
15
16 • SHOW TABLES;
17 • INSERT INTO COURSE VALUES(11 , "DS" , "CS"),
      
```
- Result Grid:** Shows the results of the SHOW TABLES command, listing tables: adoption, course, enroll, student, textbook.
- Output Tab:** Action Output table showing two entries:

#	Time	Action	Message	Duration / Fetch
1	16:57:20	SHOW TABLES	5 row(s) returned	0.234 sec / 0.000 sec
2	16:57:46	SHOW TABLES	5 row(s) returned	0.000 sec / 0.000 sec
- Information Tab:** No object selected.

This screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** Local instance MySQL80, File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Toolbar:** Standard MySQL Workbench toolbar.
- Navigator:** Schemas (banking, banking_enterprise, bms_db, book_dealer, book_dealer_database, bookdealer, books_adopted_db, books_insurance, sys) and Administration tab.
- SQL Editor:** Titled "STUDENT ENROLLMENT", containing the following SQL code:


```

16 • SHOW TABLES;
17 • INSERT INTO COURSE VALUES(11 , "DS" , "CS"),
18 (22 , "USP" , "IS"),
19 (33 , "SNS" , "EC"),
20 (44 , "DBNIS" , "CS"),
21 (55 , "EC" , "TC");
22 • SHOW TABLES;
23 • INSERT INTO ENROLL VALUES("CS01" , 11 , 4 , 85),
      
```
- Result Grid:** Shows the results of the SHOW TABLES command, listing tables: adoption, course, enroll, student, textbook.
- Output Tab:** Action Output table showing three entries:

#	Time	Action	Message	Duration / Fetch
1	16:57:20	SHOW TABLES	5 row(s) returned	0.234 sec / 0.000 sec
2	16:57:46	SHOW TABLES	5 row(s) returned	0.000 sec / 0.000 sec
3	16:58:31	SHOW TABLES	5 row(s) returned	0.015 sec / 0.000 sec
- Information Tab:** No object selected.

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

STUDENT ENROLLMENT*

SCHEMAS

Filter objects

banking
banking_enterprise_...
bms_db
book_dealer
book_dealer_data...
bookdealer
books_adopted_db
insurance
sys

21 (55 , 'EC' , 'TC');
22 • SHOW TABLES;
23 • INSERT INTO ENROLL VALUES("CS01" ,11 ,4 ,85),
24 ("IS02" ,22 , 6 ,80),
25 ("EC03" ,33 , 2 ,80),
26 ("CS03" ,44 , 6 ,75),
27 ("TC05" ,55 , 2 ,8);
28 • SHOW TABLES;

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

Tables_in_student_enrollment
adoption
course
enroll
student
textbook

No object selected

Result 8

Action Output

#	Time	Action	Message	Duration / Fetch
1	16:57:20	SHOW TABLES	5 row(s) returned	0.234 sec / 0.000 sec
2	16:57:46	SHOW TABLES	5 row(s) returned	0.000 sec / 0.000 sec
3	16:58:31	SHOW TABLES	5 row(s) returned	0.015 sec / 0.000 sec

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

STUDENT ENROLLMENT*

SCHEMAS

Filter objects

banking
banking_enterprise_...
bms_db
book_dealer
book_dealer_data...
bookdealer
books_adopted_db
insurance
sys

26 ("CS03" ,44 , 6 ,75),
27 ("TC05" ,55 , 2 ,8);
28 • SHOW TABLES;
29 • INSERT INTO TEXTBOOK VALUES(1 , "DS and C" , "Princeton" , "Padma Reddy"),
30 (2 , "Fundamentals of DS" , "Princeton" , "Goswami"),
31 (3 , "Fundamentals of DBMS" , "Princeton" , "Navathe"),
32 (4 , "SQL" , "Princeton" , "Foley"),
33 (5 , "Electronic circuits" , "TMH" , "Elmasri"),

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

Tables_in_student_enrollment
adoption
course
enroll
student
textbook

No object selected

Result 8

Action Output

#	Time	Action	Message	Duration / Fetch
1	16:57:20	SHOW TABLES	5 row(s) returned	0.234 sec / 0.000 sec
2	16:57:46	SHOW TABLES	5 row(s) returned	0.000 sec / 0.000 sec
3	16:58:31	SHOW TABLES	5 row(s) returned	0.015 sec / 0.000 sec

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

MySQL Workbench

Local instance MySQL80

STUDENT ENROLLMENT*

```

36 • INSERT INTO ADOPTION VALUES(11 ,4 ,1),
37   (11 ,4 ,2),
38   (44 ,6 ,3),
39   (44 ,6 ,4),
40   (55 ,2 ,5),
41   (22 ,6 ,6);
42 • SHOW TABLES;
43 #Demonstrate how you add a new text book to the database and make this book be adopted by some department.

```

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

Tables_in_student_enrollment

- adoption
- course
- enroll
- student
- textbook

Result 8 | Context Help | Snippets

Action Output

#	Time	Action	Message	Duration / Fetch
1	16:57:20	SHOW TABLES	5 row(s) returned	0.234 sec / 0.000 sec
2	16:57:46	SHOW TABLES	5 row(s) returned	0.000 sec / 0.000 sec
3	16:58:31	SHOW TABLES	5 row(s) returned	0.015 sec / 0.000 sec

No object selected

Object Info | Session

MySQL Workbench

Local instance MySQL80

STUDENT ENROLLMENT*

```

44
45 • INSERT INTO TEXTBOOK VALUES(7, "Operating System Concepts", "Wiley", "Silberschatz-Galvin-Gagne");
46 • INSERT INTO ADOPTION VALUES(55, 2, 7);
47
48 • SELECT * FROM TEXTBOOK;
49
50 /*Produce a list of text books (include Course #, Book-ISBN, Book-title) in the
51 alphabetical order for courses offered by the 'CS' department that use more than two

```

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

ISBN	TITLE	PUBLISHER	AUTHOR
1	DS and C	Princeton	Padma Reddy
2	Fundamentals of DS	Princeton	Godee
3	Fundamentals of DBMS	Princeton	Navathe
4	SQL	Princeton	Foley
5	Electronic circuits	TMH	Elmasri
6	Adv unix prog	TMH	Stevens
7	Operating System Concepts	Wiley	Silberschatz-Galvin-Gagne
*	NULL	NULL	NULL

No object selected

Object Info | Session

MySQL Workbench

Local instance MySQL80

STUDENT ENROLLMENT*

```

44
45 • INSERT INTO TEXTBOOK VALUES(7, "Operating System Concepts", "Wiley", "Silberschatz-Galvin-Gagne");
46 • INSERT INTO ADOPTION VALUES(55, 2, 7);
47
48 • SELECT * FROM TEXTBOOK;
49
50 /*Produce a list of text books (include Course #, Book-ISBN, Book-title) in the
51 alphabetical order for courses offered by the 'CS' department that use more than two

```

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

ISBN	TITLE	PUBLISHER	AUTHOR
1	DS and C	Princeton	Padma Reddy
2	Fundamentals of DS	Princeton	Godee
3	Fundamentals of DBMS	Princeton	Navathe
4	SQL	Princeton	Foley
5	Electronic circuits	TMH	Elmasri
6	Adv unix prog	TMH	Stevens
7	Operating System Concepts	Wiley	Silberschatz-Galvin-Gagne
*	NULL	NULL	NULL

TEXTBOOK 9 | Context Help | Snippets

Action Output

#	Time	Action	Message	Duration / Fetch
1	16:57:20	SHOW TABLES	5 row(s) returned	0.234 sec / 0.000 sec
2	16:57:46	SHOW TABLES	5 row(s) returned	0.000 sec / 0.000 sec
3	16:58:31	SHOW TABLES	5 row(s) returned	0.015 sec / 0.000 sec
4	17:00:36	SELECT * FROM TEXTBOOK LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec

No object selected

Object Info | Session

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator STUDENT ENROLLMENT ×

SCHEMAS Filter objects

- banking
- banking_enterprise
- bms_db
- book_dealer
- book_dealer_data
- bookdealer
- books_adopted_db
- insurance
- sys

```

47
48 •   SELECT * FROM TEXTBOOK;
49
50 /*Produce a list of text books (include Course #, Book-ISBN, Book-title) in the
51 alphabetical order for courses offered by the 'CS' department that use more than two
52 books.*/
53 •   SELECT C.COURSE_NO ,T.ISBN,T.TITLE FROM COURSE C, TEXTBOOK T, ADOPTION A WHERE C.DEPT = "CS" AND C.COURSE_NO = A.COURSE_NO AND
54
55
56
57
58 /*List any department that has all its adopted books published by a specific publisher.*/
59
60 •   SELECT DISTINCT C.DEPARTMENT FROM COURSE C

```

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

COURSE_NO	ISBN	TITLE
11	1	DS and C
11	2	Fundamentals of DS
44	3	Fundamentals of DBMS
44	4	SQL

Administration S ↴ ↵ Information

No object selected

Result 11 ×

Output

Action Output

#	Time	Action	Message	Duration / Fetch
3	16:58:31	SHOW TABLES	5 row(s) returned	0.015 sec / 0.000 sec
4	17:00:36	SELECT * FROM TEXTBOOK LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec
5	17:01:05	SELECT C.COURSE_NO ,T.ISBN,T.TITLE FROM COURSE C, TEXTBOOK T, ADOPTION A WH... 4 row(s) returned	4 row(s) returned	0.000 sec / 0.000 sec
6	17:01:10	SELECT C.COURSE_NO ,T.ISBN,T.TITLE FROM COURSE C, TEXTBOOK T, ADOPTION A WH... 4 row(s) returned	4 row(s) returned	0.000 sec / 0.000 sec

Context Help Snippets

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator STUDENT ENROLLMENT ×

SCHEMAS Filter objects

- banking
- banking_enterprise
- bms_db
- book_dealer
- book_dealer_data
- bookdealer
- books_adopted_db
- insurance
- sys

```

50 /*Produce a list of text books (include Course #, Book-ISBN, Book-title) in the
51 alphabetical order for courses offered by the 'CS' department that use more than two
52 books.*/
53 •   SELECT C.COURSE_NO ,T.ISBN,T.TITLE FROM COURSE C, TEXTBOOK T, ADOPTION A WHERE C.DEPT = "CS" AND C.COURSE_NO = A.COURSE_NO AND
54
55
56
57
58 /*List any department that has all its adopted books published by a specific publisher.*/
59
60 •   SELECT DISTINCT C.DEPARTMENT FROM COURSE C

```

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

DEPT
CS

Administration S ↴ ↵ Information

No object selected

COURSE 12 ×

Output

Action Output

#	Time	Action	Message	Duration / Fetch
4	17:00:36	SELECT * FROM TEXTBOOK LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec
5	17:01:05	SELECT COURSE_NO ,T.ISBN,T.TITLE FROM COURSE C, TEXTBOOK T, ADOPTION A WH... 4 row(s) returned	4 row(s) returned	0.000 sec / 0.000 sec
6	17:01:10	SELECT COURSE_NO ,T.ISBN,T.TITLE FROM COURSE C, TEXTBOOK T, ADOPTION A WH... 4 row(s) returned	4 row(s) returned	0.000 sec / 0.000 sec
7	17:01:36	SELECT DISTINCT C.DEPARTMENT FROM COURSE C WHERE C.DEPARTMENT IN (SELECT C.DEPARTMENT FROM COURSE C WHERE C.DEPARTMENT = "CS") 1 row(s) returned	1 row(s) returned	0.000 sec / 0.000 sec

Context Help Snippets

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

**** END OF WEEK5 PROGRAM ****