

FINAL REPORT

LAB INTERNAL 1

NAME: KANALA BHUVANA REDDY

USN : 1BM19CS069

BATCH: 4B B2 BATCH

LAB PROGRAM 1:

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.

CODE:

```
create database Insurance;
```

```
show databases;
```

```
use Insurance;
```

```
show tables;
```

```
create table PERSON(driverid varchar(20),dname varchar(20),address varchar(40),primary key(driverid));
```

```
desc PERSON;
```

```
create table CAR(regno varchar(10),model varchar(10),year int ,primary key(regno));
```

```
desc CAR;
```

```

create table ACCIDENT(report_no int,adate date,location varchar(20),primary
key(report_no));

desc ACCIDENT;

create table OWNS(driverid varchar(10),regno varchar(10),primary key(driverid,regno),
foreign key(driverid) references PERSON(driverid) on delete cascade,
foreign key(regno) references CAR(regno) on delete cascade);

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

desc PARTICIPATED;

insert into PERSON values('1111','Ramu','K.S.LAYOUT');

commit;

select* FROM PERSON;

insert into PERSON values('2222','John','INDIRANAGAR');

insert into PERSON values('3333','Priya','JAYANAGAR');

insert into PERSON values('4444','Gopal','WHITEFIELD');

insert into PERSON values('5555','Latha','VIJAYNAGAR');

commit;

insert into CAR values('KA04Q2301','MARUTHI-DX', 2000);

insert into CAR values('KA05P1000', 'FORDICON', 2000);

insert into CAR values('KA03L1234','ZEN-VXI',1999);

insert into CAR values('KA03L9999', 'MARUTHI-DX', 2002);

insert into CAR values('KA01P4020', 'INDICA-VX', 2002);

commit;

select * from CAR;


insert into ACCIDENT values(12,'2002-06-01', 'M G ROAD');

insert into ACCIDENT values(200, '2002-12-10', 'DOUBLEROAD');

insert into ACCIDENT values(300, '1999-07-23', 'M G ROAD');

```

```
insert into ACCIDENT values(25000, '2000-06-11', 'RESIDENCY ROAD');
insert into ACCIDENT values(26500, '2001-10-16', 'RICHMOND CIRCLE');
commit;
select * from ACCIDENT;
```

```
insert into OWNS values('1111','KA04Q2301');
insert into OWNS values('1111', 'KA05P1000');
insert into OWNS values('2222', 'KA03L1234');
insert into OWNS values('3333', 'KA03L9999');
insert into OWNS values('4444', 'KA01P4020');
commit;
select * from OWNS;
```

```
insert into PARTICIPATED values('1111', 'KA04Q2301', 12 ,20000);
insert into PARTICIPATED values('2222', 'KA03L1234', 200, 500);
insert into PARTICIPATED values('3333', 'KA03L9999', 300, 10000);
insert into PARTICIPATED values('4444', 'KA01P4020', 25000 ,2375);
insert into PARTICIPATED values('1111', 'KA05P1000', 26500 ,70000);
```

```
UPDATE PARTICIPATED SET DAMAGE_AMT=25000 WHERE REPORT_NO =12 AND
REGNO='KA04Q2301';
select * from PARTICIPATED;
```

```
select count(*) from ACCIDENT where Adate like '2002-__-__';
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";
show tables;
```

OUTPUT:
show tables;

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following commands:

```
54 * insert into PARTICIPATED values('1111', 'KA05P1000', 20500, 70000);
55
56 * UPDATE PARTICIPATED SET DAMAGE_AMT=25000 WHERE REPORT_NO =12 AND REGNO='KA04Q2301';
57 * select * from PARTICIPATED;
58
59 * select count(*) from ACCIDENT where Adate like '2002-__-__';
60 * select count(A.Report_no) from ACCIDENT A, PARTICIPATED P, CAR C where
61 A.Report_no=P.Report_no AND P.Regno=C.Regno AND C.Model='HARUTHI-DX';
62 * show tables;
```

The 'Result Grid' shows the output of the 'show tables;' command:

Tables_in_insurance
accident
car
owns
participated
person

The 'Action Output' pane shows the execution results of the queries, including the 'show tables;' command which returned 5 rows.

SELECT * FROM insurance.accident;

The screenshot shows the MySQL Workbench interface. The SQL editor contains the command:

```
1 * SELECT * FROM insurance.accident;
```

The 'Result Grid' shows the output of the 'SELECT * FROM insurance.accident;' command:

report_no	adate	location
12	2002-06-01	M G ROAD
200	2002-12-10	DOUBLERROAD
300	1999-07-23	M G ROAD
25000	2000-06-11	RESIDENCY ROAD
26500	2001-10-16	RICHMOND CIRCLE

The 'Action Output' pane shows the execution results of the queries, including the 'SELECT * FROM insurance.accident;' command which returned 5 rows.

SELECT * FROM insurance.car;

The screenshot shows the MySQL Workbench interface. The left sidebar displays the database schema for 'bhuvanadb', with the 'insurance' database selected. The 'car' table is highlighted under the 'insurance' database. The main query editor contains the SQL statement: `SELECT * FROM insurance.car;`. The 'Result Grid' shows the following data:

regno	model	year
KA01P4020	INDICA-VX	2002
KA03L1234	ZEN-VXI	1999
KA03L9999	MARUTHE-DX	2002
KA04Q2301	MARUTHE-DX	2000
KA05P1000	FORDICON	2000

The 'Output' pane at the bottom shows the execution log, including the query execution time and the number of rows returned.

SELECT * FROM insurance.owns;

The screenshot shows the MySQL Workbench interface. The left sidebar displays the database schema for 'bhuvanadb', with the 'insurance' database selected. The 'owns' table is highlighted under the 'insurance' database. The main query editor contains the SQL statement: `SELECT * FROM insurance.owns;`. The 'Result Grid' shows the following data:

driverid	regno
4444	KA01P4020
2222	KA03L1234
3333	KA03L9999
1111	KA04Q2301
1111	KA05P1000

The 'Output' pane at the bottom shows the execution log, including the query execution time and the number of rows returned.

SELECT * FROM insurance.participated;

The screenshot shows the MySQL Workbench interface with the 'participated' table selected in the Navigator. The SQL editor contains the query 'SELECT * FROM insurance.participated;'. The Results window displays the following data:

driverid	regno	report_no	damage_amt
1111	KA04Q2301	12	25000
2222	KA03L234	200	500
3333	KA03.9999	300	10000
4444	KA01P4020	25000	2375
1111	KA0SP1000	26500	70000

The Action Output window shows the execution of the query, indicating that 5 rows were returned.

SELECT * FROM insurance.person;

The screenshot shows the MySQL Workbench interface with the 'person' table selected in the Navigator. The SQL editor contains the query 'SELECT * FROM insurance.person;'. The Results window displays the following data:

driverid	dname	address
1111	Ramu	K.S.LAYOUT
2222	John	INDIRANAGAR
3333	Priya	JAYANAGAR
4444	Gopal	WHITEFIELD
5555	Latha	VJAYNAGAR

The Action Output window shows the execution of the query, indicating that 5 rows were returned.

LAB PROGRAM 2:

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%.

CODE:

```
create database book_dealer;

use book_dealer;

create table author(author_id int,name varchar(15),city varchar(15),country
varchar(15),primary key(author_id));

create table publisher(publisher_id int,name varchar(15),city varchar(15),country
varchar(15),primary key(publisher_id));

create table category(category_id int,description varchar(15),primary key(category_id));

create table catalog(book_id int,title varchar(15),author_id int,publisher_id int,category_id
int,year int,price int,primary key(book_id),

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);
create table order_details(order_no int,book_id int,quantity int,primary key(order_no),
foreign key(book_id)references catalog(book_id) on delete cascade);
insert into author values('1001','TERAS CHAIN','CA','USA');
insert into author values('1002','STEVENS','ZOMBI','UGANDA');
insert into author values('1003','M MANO','CAIR','CANADA');
insert into author values('1004','KARTHIK B.P.','NEW YORK','USA');
insert into author values('1005','WILLIAM','LAS VEGAS','USA');
commit;
select*from author;
insert into publisher values('101','PEARSON','NEW YORK','USA');
insert into publisher values('102','EEE','NEW SOUTHVALES','USA');
insert into publisher values('103','PHI','DELHI','INDIA');
insert into publisher values('104','WILLEY','BERLIN','GERMANY');
insert into publisher values('105','MGH','NEW YORK','USA');
commit;
select*from publisher;
insert into category values('1011','CSE');
insert into category values('1022','ADA');
insert into category values('1033','ELECTRONICS');
insert into category values('1044','PROGRAMMING');
insert into category values('1055','OS');
commit;
select*from category;
insert into catalog values('11','UNIX SYSTEM','1001','101','1011','1998','235');
insert into catalog values('12','DIG ANALYSIS','1001','102','1033','1997','255');
insert into catalog values('13','LOGIC DESIGN','1002','103','1022','2001','352');
insert into catalog values('14','SERVER','1002','103','1011','2002','523');
insert into catalog values('15','LINUX OS','1003','104','1044','2003','124');

```

```
commit;
```

```
select*from catalog;
```

```
insert into order_details values('1','11','12');
```

```
insert into order_details values('1','12','2');
```

```
insert into order_details values('2','12','15');
```

```
insert into order_details values('3','13','22');
```

```
insert into order_details values('4','13','14');
```

```
insert into order_details values('5','15','7');
```

```
commit;
```

```
select*from order_details;
```

```
select a.author_id,a.name,a.city,a.country,c.price,year from author a join catalog c on  
a.author_id=c.author_id where c.year>=2000 group by a.author_id having  
count(c.author_id)>=2 ;
```

```
select c.author_id from catalog c where book_id=(select book_id from order_details od  
where quantity=(select max(quantity) from order_details having max(quantity)))) ;
```

```
update catalog set price=1.1*price where publisher_id=(select publisher_id from publisher  
where name='PEARSON');
```

```
select*from catalog;
```

OUTPUT:

SELECT * FROM book_dealer.order_details;

The screenshot shows the MySQL Workbench interface. The SQL Editor contains the query: `SELECT * FROM book_dealer.order_details;`. The left sidebar shows the database schema with the `book_dealer` database selected. The `order_details` table is highlighted. The bottom pane shows the execution results. The 'Action Output' tab is active, displaying a list of actions and their results. The 'Output' tab shows the results of the SELECT query, which is currently empty. The 'Message' pane shows the following messages:

#	Time	Action	Message	Duration / Fetch
71	14:15:06	Insert into catalog values('15','UNIX OS','1003','104','2003','124')	1 row(s) affected	0.000 sec
72	14:15:06	commit	0 row(s) affected	0.000 sec
73	14:15:06	select from catalog LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
74	14:15:06	insert into order_details values('1','11','12')	1 row(s) affected	0.000 sec
75	14:15:06	insert into order_details values('1','12','2')	Error Code: 1062. Duplicate entry '1' for key 'order_details PRIMARY'	0.000 sec
76	14:17:28	SELECT * FROM book_dealer.order_details LIMIT 0, 1000	1 row(s) returned	0.047 sec / 0.000 sec

SELECT * FROM book_dealer.author;

The screenshot shows the MySQL Workbench interface. The SQL Editor contains the query: `SELECT * FROM book_dealer.author;`. The left sidebar shows the database schema with the `book_dealer` database selected. The `author` table is highlighted. The bottom pane shows the execution results. The 'Action Output' tab is active, displaying a list of actions and their results. The 'Output' tab shows the results of the SELECT query, which is currently empty. The 'Message' pane shows the following messages:

#	Time	Action	Message	Duration / Fetch
72	14:15:06	commit	0 row(s) affected	0.000 sec
73	14:15:06	select from catalog LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
74	14:15:06	insert into order_details values('1','11','12')	1 row(s) affected	0.000 sec
75	14:15:06	insert into order_details values('1','12','2')	Error Code: 1062. Duplicate entry '1' for key 'order_details PRIMARY'	0.000 sec
76	14:17:28	SELECT * FROM book_dealer.order_details LIMIT 0, 1000	1 row(s) returned	0.047 sec / 0.000 sec
77	14:18:14	SELECT * FROM book_dealer.author LIMIT 0, 1000	5 row(s) returned	0.062 sec / 0.000 sec

SELECT * FROM book_dealer.catalog;

The screenshot shows the MySQL Workbench interface. The 'Schemas' pane on the left shows the 'book_dealer' database selected. The 'SQL File 1' editor contains the query 'SELECT * FROM book_dealer.catalog;'. The 'Result Grid' displays the following data:

book_id	title	author_id	publisher_id	category_id	year	price
11	UNIX SYSTEM	1001	101	1011	1998	235
12	DIG ANALYSIS	1001	102	1033	1997	255
13	LOGIC DESIGN	1002	103	1022	2001	352
14	SERVER	1002	103	1011	2002	523
15	LINUX OS	1003	104	1044	2003	124

The 'Action Output' pane shows the execution of the query, indicating that 5 rows were returned.

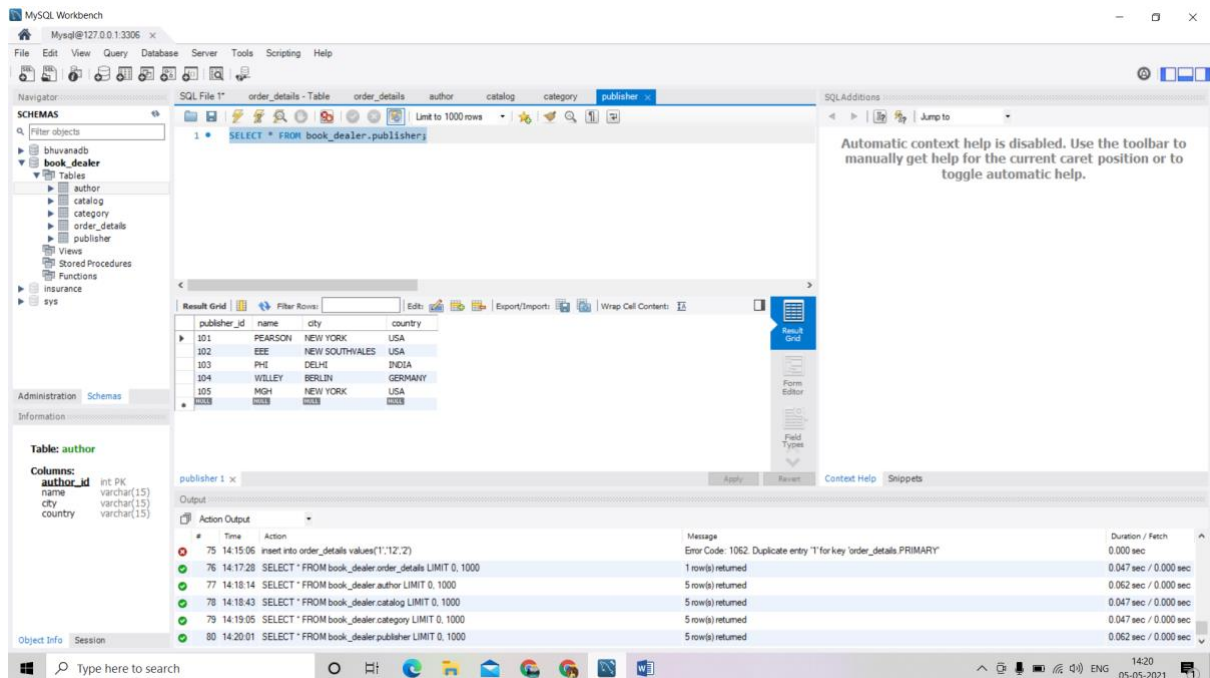
SELECT * FROM book_dealer.category;

The screenshot shows the MySQL Workbench interface. The 'Schemas' pane on the left shows the 'book_dealer' database selected. The 'SQL File 1' editor contains the query 'SELECT * FROM book_dealer.category;'. The 'Result Grid' displays the following data:

category_id	description
1011	CSE
1022	ADA
1033	ELECTRONICS
1044	PROGRAMMING
1055	OS

The 'Action Output' pane shows the execution of the query, indicating that 5 rows were returned.

SELECT * FROM book_dealer.publisher;



LAB PROGRAM 3:

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.

CODE:

```

create database order_Processing;
use order_Processing;

create table customer(custno int,cname varchar(30),city varchar(30),primary key(custno));

create table order1(orderno int,odate date,custno int,ord_amt int,primary
key(orderno),foreign key(custno) references customer(custno));

create table item(itemno int,untpirice int,primary key(itemno));

create table order_item(orderno int,itemno int,quantity int,primary
key(orderno,itemno),foreign key(orderno) references order1(orderno),foreign key(itemno)
references item(itemno) on delete cascade);

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

create table shipment(orderno int,warehouseno int,ship_date date,primary
key(orderno,warehouseno),foreign key(orderno) references order1(orderno),foreign
key(warehouseno) references warehouse(warehouseno));

show tables;

insert into customer values (771,'PUSHPA K','BANGALORE');
insert into customer values (772,'SUMAN','MUMBAI');

```

```
insert into customer values (773,'SOURAV','CALICUT');  
insert into customer values (774,'LAILA','HYDERABAD');  
insert into customer values (775,'FAIZAL','BANGALORE');
```

```
insert into order1 values (111,'2002-1-22',771,18000);  
insert into order1 values (112,'2002-7-30',774,6000);  
insert into order1 values (113,'2003-4-3',775,9000);  
insert into order1 values (114,'2003-10-3',775,29000);  
insert into order1 values (115,'2003-11-3',773,29000);  
insert into order1 values (116,'22-8-4',772,56000);  
insert into order1 values (117,'22-9-4',771,20000);  
insert into order1 values (118,'22-11-4',775,29000);  
insert into order1 values (119,'22-2-5',774,29000);  
insert into order1 values (120,'22-10-5',775,29000);
```

```
insert into item values (5001,503);  
insert into item values (5002,750);  
insert into item values (5003,150);  
insert into item values (5004,600);  
insert into item values (5005,890);
```

```
insert into order_item values (111,5001,50);  
insert into order_item values (112,5003,20);  
insert into order_item values (113,5002,50);  
insert into order_item values (114,5005,60);  
insert into order_item values (115,5004,90);  
insert into order_item values (116,5001,10);  
insert into order_item values (117,5003,80);  
insert into order_item values (118,5005,50);
```

```
insert into order_item values (119,5002,10);  
insert into order_item values (120,5004,45);
```

```
insert into warehouse values (1,'DELHI');  
insert into warehouse values (2,'BOMBAY');  
insert into warehouse values (3,'CHENNAI');  
insert into warehouse values (4,'BANGALORE');  
insert into warehouse values (5,'BANGALORE');  
insert into warehouse values (6,'DELHI');  
insert into warehouse values (7,'BOMBAY');  
insert into warehouse values (8,'CHENNAI');  
insert into warehouse values (9,'DELHI');  
insert into warehouse values (10,'BANGALORE');
```

```
insert into shipment values (111,1,'2002-2-10');  
insert into shipment values (112,5,'2002-2-10');  
insert into shipment values (113,8,'2003-2-10');  
insert into shipment values (114,3,'2003-10-10');  
insert into shipment values (115,9,'2004-1-19');  
insert into shipment values (116,1,'2004-9-20');  
insert into shipment values (117,5,'2004-9-10');  
insert into shipment values (118,7,'2004-11-30');  
insert into shipment values (119,7,'2005-4-30');  
insert into shipment values (120,6,'2005-10-21');
```

```
select * from customer;  
select * from order1;  
select * from item;  
select * from order_item;  
select * from warehouse;  
select * from shipment;
```



```
/*query(1)*/
```

```
select c.custno,count(*) as No_of_orders,avg(o.ord_amt) as Avg_order_amount from customer c,  
order1 o where o.custno=c.custno group by c.custno;
```

```
/*query(3)*/
```

```
delete from item where itemno=5002;
```

```
/*query(2)*/
```

```
select orderno from shipment, warehouse where shipment.warehouseno =  
warehouse.warehouseno and  
warehouse.city = 'BANGALORE';
```

```
commit;
```

OUTPUT:

```
SELECT * FROM order_processing.customer;
```

The screenshot shows the MySQL Workbench interface. The 'Schemas' pane on the left lists the 'order_processing' schema. The 'Query' editor in the center contains the SQL query: `SELECT * FROM order_processing.customer;`. The 'Result Grid' pane displays the results of the query, showing a table with columns 'custno', 'cname', and 'city'. The results are as follows:

custno	cname	city
771	PUSHPA K	BANGALORE
772	SURPRI	MUMBAI
773	SOURAV	CALCUT
774	LAILA	HYDERABAD
775	FAIZAL	BANGALORE

The 'Output' pane at the bottom shows the execution log, including the time taken for each step and the number of rows returned. The final step shows that 5 rows were returned for the query.

```
SELECT * FROM order_processing.item;
```

MySQL Workbench

Navigator

SCHMAS

Filter objects

banking

book_dealer

insurance

order_processing

customer

item

order1

order_item

shipment

warehouse

Views

Stored Procedures

Functions

sys

Administration

Schemas

Information

Schema: order_processing

Object Info

Session

1 * SELECT * FROM order_processing.item

Result Grid

Itemno	unitprice
5001	503
5002	750
5003	150
5004	600
5005	890
5006	1000

Output

Action Output

#	Time	Action	Message	Duration / Fetch
63	15:19:58	select * from item LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
64	15:19:58	select * from order_item LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
65	15:19:58	select * from warehouse LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
66	15:19:58	select * from shipment LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
67	15:22:08	SELECT * FROM order_processing.customer LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
68	15:22:40	SELECT * FROM order_processing.item LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec

Search for anything

1522 12-05-2021

SELECT * FROM order_processing.order1;

MySQL Workbench

Navigator

SCHMAS

Filter objects

banking

book_dealer

insurance

order_processing

customer

item

order1

order_item

shipment

warehouse

Views

Stored Procedures

Functions

sys

Administration

Schemas

Information

Schema: order_processing

Object Info

Session

1 * SELECT * FROM order_processing.order1

Result Grid

orderno	odate	custno	ord_amt
111	2002-01-22	771	18000
112	2002-07-30	774	6000
113	2003-04-03	775	9000
114	2003-10-03	775	29000
115	2003-11-03	773	29000
116	2002-08-04	772	56000
117	2002-09-04	771	20000
118	2002-11-04	775	29000
119	2002-02-05	774	29000
120	2002-10-05	775	29000

Search for anything

1523 12-05-2021

SELECT * FROM order_processing.order_item;

The screenshot shows the MySQL Workbench interface. The query editor contains the SQL statement: `SELECT * FROM order_processing.order_item;`. The result grid displays the following data:

orderno	itemno	quantity
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

SELECT * FROM order_processing.shipment;

The screenshot shows the MySQL Workbench interface. The query editor contains the SQL statement: `SELECT * FROM order_processing.shipment;`. The result grid displays the following data:

orderno	warehouseno	ship_date
111	1	2003-02-10
112	5	2003-02-10
113	8	2003-02-10
114	3	2003-10-10
115	9	2004-01-19
116	1	2004-09-20
117	5	2004-09-10
118	7	2004-11-30
119	7	2005-04-30
120	6	2005-10-21

SELECT * FROM order_processing.warehouse;

The screenshot shows the MySQL Workbench interface. The query editor contains the SQL statement: `SELECT * FROM order_processing.warehouse;`. The result grid displays the following data:

warehouseid	city
1	DELHI
2	BOMBAY
3	CHENNAI
4	BANGALORE
5	BANGALORE
6	DELHI
7	BOMBAY
8	CHENNAI
9	DELHI
10	BANGALORE

QUERY 1.

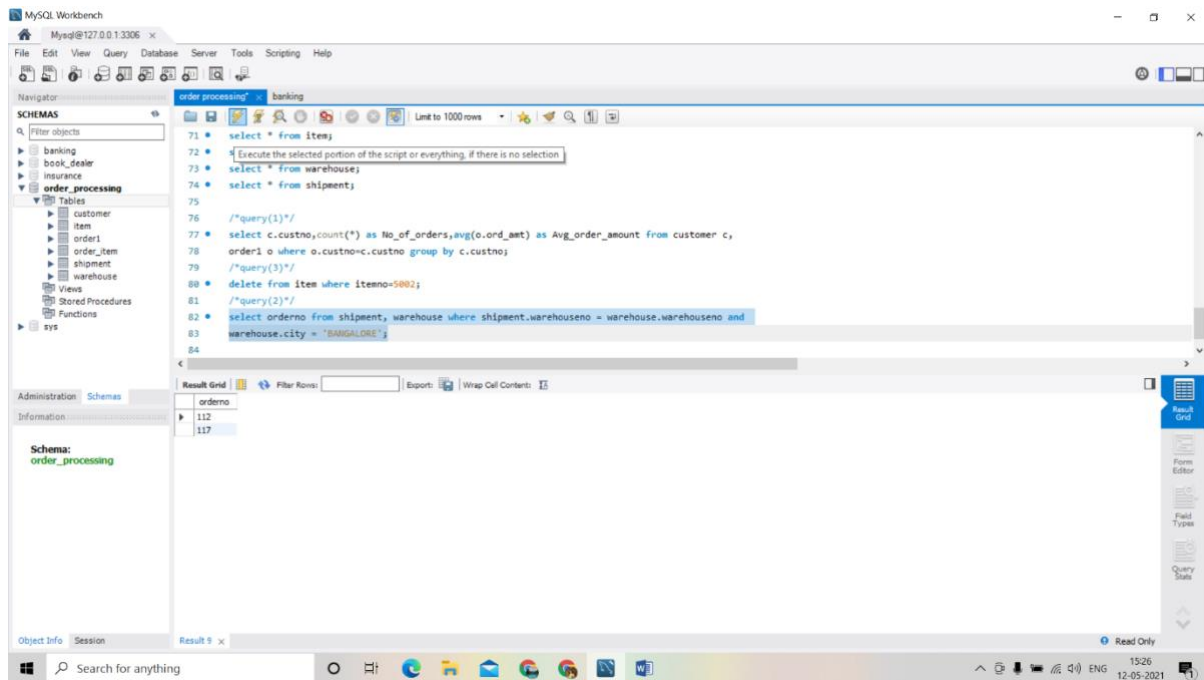
The screenshot shows the MySQL Workbench interface with a complex SQL query. The query editor contains the following SQL statements:

```
66 insert into shipment values (119,7,'2005-4-30');
67 insert into shipment values (120,6,'2005-10-21');
68
69 select * from customer;
70 select * from order1;
71 select * from item;
72 select * from order_item;
73 select * from warehouse;
74 select * from shipment;
75
76 /*query(1)*/
77 select c.custno,count(*) as No_of_orders,avg(o.ord_amt) as Avg_order_amount from customer c,
order1 o where o.custno=c.custno group by c.custno;
78
79 /*query(3)*/
```

The result grid displays the following data:

custno	No_of_orders	Avg_order_amount
771	2	19000.0000
772	1	56000.0000
773	1	29000.0000
774	2	17500.0000
775	4	24000.0000

QUERY 2.



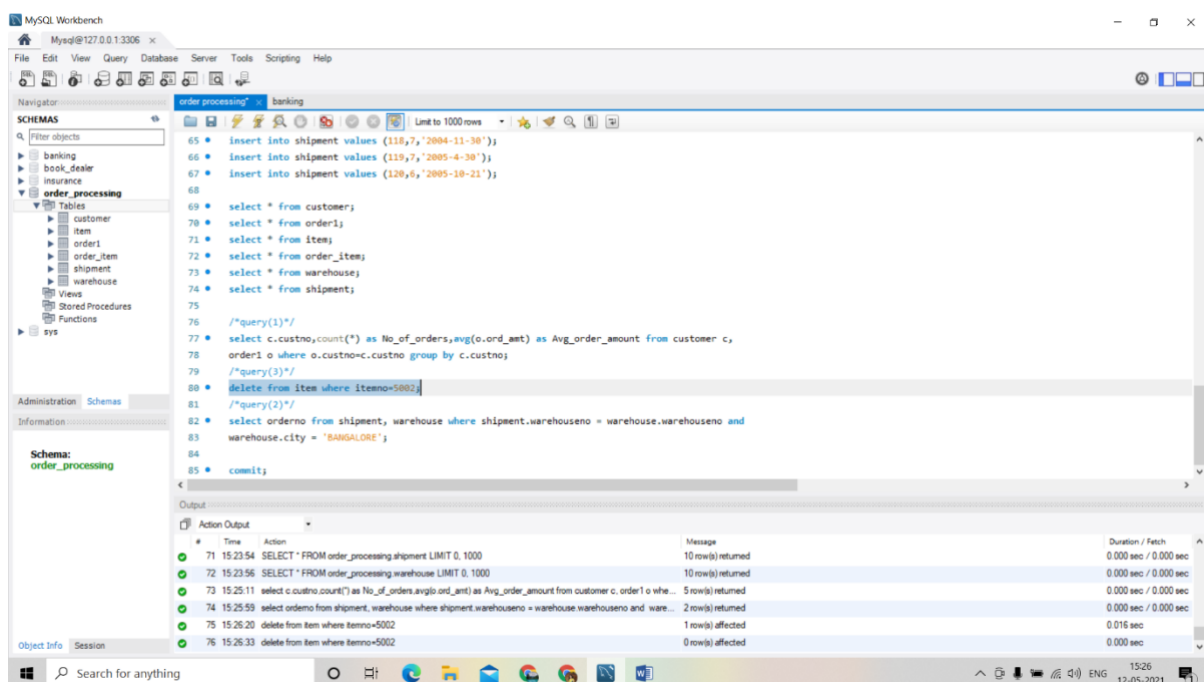
MySQL Workbench interface showing Query 2. The query editor contains the following SQL code:

```
71 select * from item;
72
73 select * from warehouse;
74 select * from shipment;
75
76 /*query(1)*/
77 select c.custno,count(*) as No_of_orders,avg(o.ord_amt) as Avg_order_amount from customer c,
78 order1 o where o.custno=c.custno group by c.custno;
79
80 /*query(2)*/
81 delete from item where itemno=5002;
82
83 /*query(3)*/
84 select orderno from shipment, warehouse where shipment.warehouseno = warehouse.warehouseno and
85 warehouse.city = 'BANGALORE';
```

The result grid shows two rows of data:

orderno
112
117

QUERY 3.



MySQL Workbench interface showing Query 3. The query editor contains the following SQL code:

```
65 insert into shipment values (118,7,'2004-11-30');
66 insert into shipment values (119,7,'2005-4-30');
67 insert into shipment values (120,6,'2005-10-21');
68
69 select * from customer;
70 select * from order1;
71 select * from item;
72 select * from order_item;
73 select * from warehouse;
74 select * from shipment;
75
76 /*query(1)*/
77 select c.custno,count(*) as No_of_orders,avg(o.ord_amt) as Avg_order_amount from customer c,
78 order1 o where o.custno=c.custno group by c.custno;
79
80 /*query(2)*/
81 delete from item where itemno=5002;
82
83 /*query(3)*/
84 select orderno from shipment, warehouse where shipment.warehouseno = warehouse.warehouseno and
85 warehouse.city = 'BANGALORE';
86
87 commit;
```

The output window shows the execution results of the queries:

#	Time	Action	Message	Duration / Fetch
71	15:23:54	SELECT * FROM order_processing.shipment LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
72	15:23:56	SELECT * FROM order_processing.warehouse LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
73	15:25:11	select c.custno,count(*) as No_of_orders,avg(o.ord_amt) as Avg_order_amount from customer c, order1 o where o.custno=c.custno group by c.custno	5 row(s) returned	0.000 sec / 0.000 sec
74	15:25:59	select orderno from shipment, warehouse where shipment.warehouseno = warehouse.warehouseno and warehouse.city = 'BANGALORE'	2 row(s) returned	0.000 sec / 0.000 sec
75	15:26:20	delete from item where itemno=5002	1 row(s) affected	0.016 sec
76	15:26:33	delete from item where itemno=5002	0 row(s) affected	0.000 sec

LAB PROGRAM 4:

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.

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 values('SBI PD NAGAR','BANGALORE', 200000);

insert into branch values('SBI RAJAJI NAGAR','BANGALORE', 500000);

insert into branch values('SBI JAYANAGAR','DELHI', 660000);

insert into branch values('SBI VIJAY NAGAR','CHENNAI', 870000);

insert into branch values('SBI HOSAKEREHALLI','CHENNAI', 550000);

select *from branch;
```

```
insert into accounts values(1001, 'SBI PD NAGAR',15000);

insert into accounts values(1002, 'SBI RAJAJI NAGAR',25000);

insert into accounts values(1003 , 'SBI JAYANAGAR',500);

insert into accounts values(1004, 'SBI VIJAY NAGAR',10000);

insert into accounts values(1005,'SBI VIJAY NAGAR',40000);

insert into accounts values(1006 , 'SBI HOSAKEREHALLI',4000);

insert into accounts values(1007 , 'SBI RAJAJI NAGAR',7000);

insert into accounts values(1008 , 'SBI RAJAJI NAGAR',7000);
```

```
select * from accounts;
```

```
insert into customer values('KEZAR','M G ROAD','BANGALORE');
```

```
insert into customer values('LAL KRISHNA','ST MKS ROAD','DELHI');
```

```
insert into customer values('RAHUL','AUGSTEN ROAD','CHENNAI');
```

```
insert into customer values('FAIZAL','RESEGENCY ROAD','BANGALORE');
```

```
insert into customer values('RAJEEV','DICKNSN ROAD','CHENNAI');
```

```
select * from customer;
```

```
insert into depositor values('KEZAR',1001);
```

```
insert into depositor values('KEZAR',1002);
```

```
insert into depositor values('LAL KRISHNA',1003);
```

```
insert into depositor values('RAHUL',1004);
```

```
insert into depositor values('RAHUL',1005);
```

```
insert into depositor values('FAIZAL',1006);
```

```
insert into depositor values('RAJEEV',1007);
```

```
insert into depositor values('RAJEEV',1008);
```

```
select * from depositor;
```

```
insert into loan values(10011,'SBI PD NAGAR',10000);
```

```
insert into loan values(10012,'SBI RAJAJI NAGAR',5000);
```

```
insert into loan values(10013,'SBI RAJAJI NAGAR',20000);
```

```
insert into loan values(10014,'SBI JAYANAGAR',15000);
```

```
insert into loan values(10015,'SBI HOSAKEREHALLI',25000);
```

```
select * from loan;
```

```
insert into borrower values('FAIZAL', 10011);
```

```
insert into borrower values('KEZAR', 10012);
```

```
insert into borrower values('FAIZAL', 10013);
```



```
insert into borrower values('LAL KRISHNA', 10014);
```

```
insert into borrower values('RAJEEV', 10015);
```

```
select *from borrower;
```

```
/*1*/
```

```
select customer_name from depositor d, accounts a where d.accno=a.accno and  
a.branch_name="SBI VIJAY NAGAR" group by d.customer_name having  
count(d.customer_name) >= 2;
```

```
/*2*/
```

```
select d.customer_name from accounts a, branch b, depositor d where  
b.branch_name=a.branch_name and a.accno=d.accno and b.branch_city="BANGALORE"  
group by d.customer_name having count(distinct b.branch_name)=(select  
count(branch_name) from branch where branch_city="BANGALORE");
```

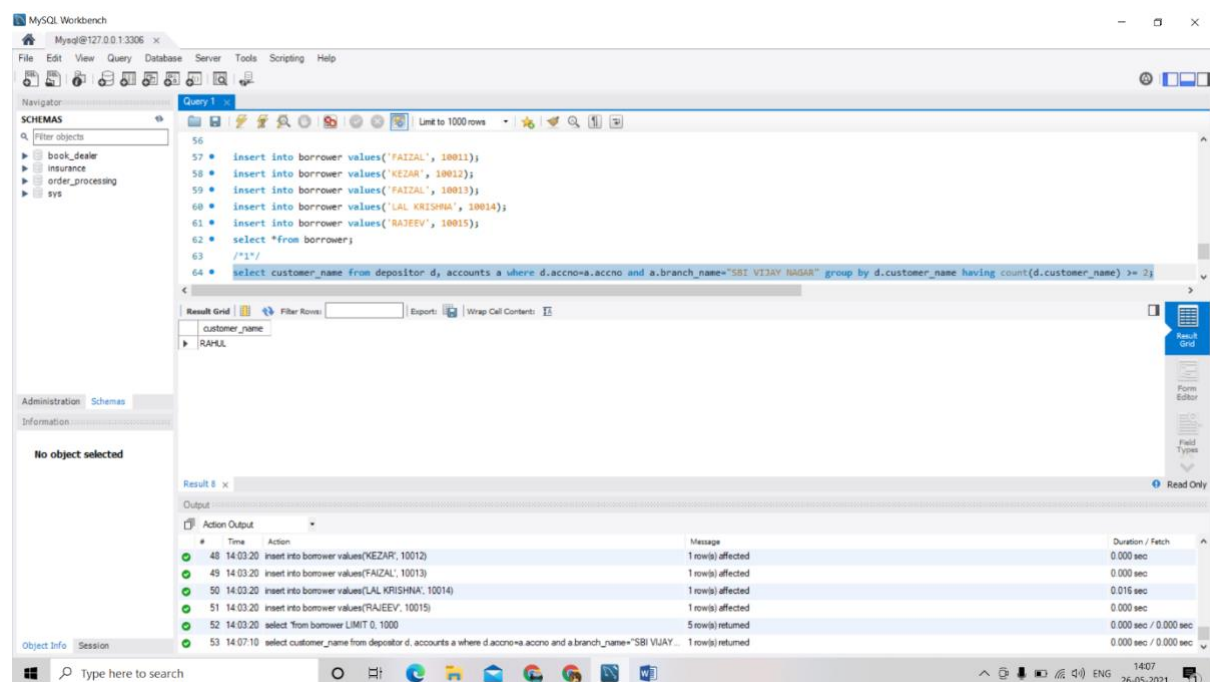
```
/*3*/
```

```
delete from accounts where branch_name in(select branch_name from branch where  
branch_city="BANGALORE");
```

```
select *from accounts;
```

OUTPUT:

Query 1



The screenshot shows the MySQL Workbench interface. The 'Query 1' window displays the following SQL script:

```
56  
57 * insert into borrower values('FAIZAL', 10011);  
58 * insert into borrower values('KEZAR', 10012);  
59 * insert into borrower values('FAIZAL', 10013);  
60 * insert into borrower values('LAL KRISHNA', 10014);  
61 * insert into borrower values('RAJEEV', 10015);  
62 select *from borrower;  
63 /*1*/  
64 select customer_name from depositor d, accounts a where d.accno=a.accno and a.branch_name="SBI VIJAY NAGAR" group by d.customer_name having count(d.customer_name) >= 2;
```

The 'Result Grid' shows the output of the queries. The first query returns 5 rows, and the second query returns 1 row.

#	Time	Action	Message	Duration / Fetch
48	14.03.20	insert into borrower values('KEZAR', 10012)	1 row(s) affected	0.000 sec
49	14.03.20	insert into borrower values('FAIZAL', 10013)	1 row(s) affected	0.000 sec
50	14.03.20	insert into borrower values('LAL KRISHNA', 10014)	1 row(s) affected	0.016 sec
51	14.03.20	insert into borrower values('RAJEEV', 10015)	1 row(s) affected	0.000 sec
52	14.03.20	select *from borrower LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
53	14.07.10	select customer_name from depositor d, accounts a where d.accno=a.accno and a.branch_name="SBI VIJAY NAGAR" group by d.customer_name having count(d.customer_name) >= 2	1 row(s) returned	0.000 sec / 0.000 sec

Query 2

MySQL Workbench

Navigator

SCHEMAS

Filter objects

- book_dealer
- insurance
- order_processing
- sys

Administration Schemas

Information

No object selected

Query 1

```

59 * insert into borrower values('FAIZAL', 10013);
60 * insert into borrower values('LAL KRISHNA', 10014);
61 * insert into borrower values('RAJEEV', 10015);
62 * select *from borrower;
63 /**/
64 * select customer_name from depositor d, accounts a where d.accto=a.accto and a.branch_name='SBI VIJAY NAGAR' group by d.customer_name having count(d.customer_name) >= 2;
65 /*2*/
66 * select d.customer_name from accounts a, branch b, depositor d where b.branch_name=a.branch_name and a.accto=d.accto and b.branch_city='BANGALORE'
67 group by d.customer_name having count(distinct b.branch_name)=(select count(branch_name) from branch where branch_city='BANGALORE');

```

Result Grid

customer_name
KEZAR

Output

Action Output

#	Time	Action	Message	Duration / Fetch
49	14.03.20	insert into borrower values('FAIZAL', 10013);	1 row(s) affected	0.000 sec
50	14.03.20	insert into borrower values('LAL KRISHNA', 10014);	1 row(s) affected	0.016 sec
51	14.03.20	insert into borrower values('RAJEEV', 10015);	1 row(s) affected	0.000 sec
52	14.03.20	select *from borrower LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
53	14.07.10	select customer_name from depositor d, accounts a where d.accto=a.accto and a.branch_name='SBI VIJAY NAGAR' group by d.customer_name having count(d.customer_name) >= 2;	1 row(s) returned	0.000 sec / 0.000 sec
54	14.07.34	select d.customer_name from accounts a, branch b, depositor d where b.branch_name=a.branch_name and a.accto=d.accto and b.branch_city='BANGALORE' group by d.customer_name having count(distinct b.branch_name)=(select count(branch_name) from branch where branch_city='BANGALORE');	1 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Type here to search

1407 26-05-2021

Query 3

MySQL Workbench

Navigator

SCHEMAS

Filter objects

- book_dealer
- insurance
- order_processing
- sys

Administration Schemas

Information

No object selected

Query 1

```

62 * select *from borrower;
63 /**/
64 * select customer_name from depositor d, accounts a where d.accto=a.accto and a.branch_name='SBI VIJAY NAGAR' group by d.customer_name having count(d.customer_name) >= 2;
65 /*2*/
66 * select d.customer_name from accounts a, branch b, depositor d where b.branch_name=a.branch_name and a.accto=d.accto and b.branch_city='BANGALORE'
67 group by d.customer_name having count(distinct b.branch_name)=(select count(branch_name) from branch where branch_city='BANGALORE');
68 /*3*/
69 * delete from accounts where branch_name in(select branch_name from branch where branch_city='BANGALORE');
70 * select *from accounts;

```

Result Grid

accto	branch_name	balance
1003	SBI JAYANAGAR	500
1004	SBI VIJAY NAGAR	10000
1005	SBI VIJAY NAGAR	40000
1006	SBI HOSANURHALLI	4000

accounts 10

Output

Action Output

#	Time	Action	Message	Duration / Fetch
51	14.03.20	insert into borrower values('RAJEEV', 10015);	1 row(s) affected	0.000 sec
52	14.03.20	select *from borrower LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
53	14.07.10	select customer_name from depositor d, accounts a where d.accto=a.accto and a.branch_name='SBI VIJAY NAGAR' group by d.customer_name having count(d.customer_name) >= 2;	1 row(s) returned	0.000 sec / 0.000 sec
54	14.07.34	select d.customer_name from accounts a, branch b, depositor d where b.branch_name=a.branch_name and a.accto=d.accto and b.branch_city='BANGALORE' group by d.customer_name having count(distinct b.branch_name)=(select count(branch_name) from branch where branch_city='BANGALORE');	1 row(s) returned	0.000 sec / 0.000 sec
55	14.07.58	delete from accounts where branch_name in(select branch_name from branch where branch_city='BANGALORE');	4 row(s) affected	0.063 sec
56	14.08.09	select *from accounts LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Type here to search

1408 26-05-2021

SELECT * FROM banking.accounts;

The screenshot shows the MySQL Workbench interface. The 'Query' tab is active, displaying the query: `SELECT * FROM banking.accounts;`. The 'Result Grid' shows the following data:

acno	branch_name	balance
1003	SBI JAYANAGAR	500
1004	SBI VIJAY NAGAR	10000
1005	SBI VIJAY NAGAR	40000
1006	SBI HOSAKEREHALLI	4000

The 'Action Output' pane shows the execution log, including the query execution and the number of rows returned (4 rows).

SELECT * FROM banking.borrower;

The screenshot shows the MySQL Workbench interface. The 'Query' tab is active, displaying the query: `SELECT * FROM banking.borrower;`. The 'Result Grid' shows the following data:

customer_name	loan_number
FAIZAL	10011
KEZAR	10012
FAIZAL	10013
LAL KRISHNA	10014
RAJEEV	10015

The 'Action Output' pane shows the execution log, including the query execution and the number of rows returned (5 rows).

SELECT * FROM banking.branch;

MySQL Workbench

Query 1: `SELECT * FROM banking.branch`

Result Grid:

branch_name	branch_city	assets
SRI HOSAKEREHALLI	CHENNAI	550000
SRI JAYANAGAR	DELHI	660000
SRI PD NAGAR	BANGALORE	200000
SRI RAJAJI NAGAR	BANGALORE	500000
SRI VIJAY NAGAR	CHENNAI	870000

Action Output:

#	Time	Action	Message	Duration / Fetch
54	14:07:34	select d.customer_name from accounts a, branch b, depositor d where b.branch_name=a.branch_name and a...	1 row(s) returned	0.000 sec / 0.000 sec
55	14:07:58	delete from accounts where branch_name in(select branch_name from branch where branch_city='BANGALO...	4 row(s) affected	0.063 sec
56	14:08:09	select * from accounts LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
57	14:08:51	SELECT * FROM banking.accounts LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
58	14:09:28	SELECT * FROM banking.borrower LIMIT 0, 1000	5 row(s) returned	0.047 sec / 0.000 sec
59	14:09:54	SELECT * FROM banking.branch LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec

`SELECT * FROM banking.customer;`

MySQL Workbench

Query 1: `SELECT * FROM banking.customer;`

Result Grid:

customer_name	customer_street	customer_city
FAIZAL	RESEDENCY ROAD	BANGALORE
KEZAR	M G ROAD	BANGALORE
LAL KRISHNA	ST MRS ROAD	DELHI
RAJAL	AUGUSTEN ROAD	CHENNAI
RAJEEV	DISCONSH ROAD	CHENNAI

Action Output:

#	Time	Action	Message	Duration / Fetch
55	14:07:58	delete from accounts where branch_name in(select branch_name from branch where branch_city='BANGALO...	4 row(s) affected	0.063 sec
56	14:08:09	select * from accounts LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
57	14:08:51	SELECT * FROM banking.accounts LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
58	14:09:28	SELECT * FROM banking.borrower LIMIT 0, 1000	5 row(s) returned	0.047 sec / 0.000 sec
59	14:09:54	SELECT * FROM banking.branch LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
60	14:10:18	SELECT * FROM banking.customer LIMIT 0, 1000	5 row(s) returned	0.047 sec / 0.000 sec

`SELECT * FROM banking.depositor;`

MySQL Workbench

Query 1: `SELECT * FROM banking.depositor;`

Result Grid:

customer_name	accno
LAL KRISHNA	1003
RAHUL	1004
RAHUL	1005
PAIZAL	1006

depositor1 x

Output:

#	Time	Action	Message	Duration / Fetch
56	14:08:09	select * from accounts LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
57	14:08:51	SELECT * FROM banking.accounts LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
58	14:09:28	SELECT * FROM banking.borrower LIMIT 0, 1000	5 row(s) returned	0.047 sec / 0.000 sec
59	14:09:54	SELECT * FROM banking.branch LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
60	14:10:18	SELECT * FROM banking.customer LIMIT 0, 1000	5 row(s) returned	0.047 sec / 0.000 sec
61	14:10:40	SELECT * FROM banking.depositor LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec

SELECT * FROM banking.loan;

MySQL Workbench

Query 1: `SELECT * FROM banking.loan;`

Result Grid:

loan_number	branch_name	amount
10011	SBI PO NAGAR	10000
10012	SBI RAJAJI NAGAR	5000
10013	SBI RAJAJI NAGAR	20000
10014	SBI JAYANAGAR	15000
10015	SBI HOSAKEREHALLI	25000

loan1 x

Output:

#	Time	Action	Message	Duration / Fetch
57	14:08:51	SELECT * FROM banking.accounts LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
58	14:09:28	SELECT * FROM banking.borrower LIMIT 0, 1000	5 row(s) returned	0.047 sec / 0.000 sec
59	14:09:54	SELECT * FROM banking.branch LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
60	14:10:18	SELECT * FROM banking.customer LIMIT 0, 1000	5 row(s) returned	0.047 sec / 0.000 sec
61	14:10:40	SELECT * FROM banking.depositor LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
62	14:11:07	SELECT * FROM banking.loan LIMIT 0, 1000	5 row(s) returned	0.062 sec / 0.000 sec

LAB PROGRAM 5:

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.

CODE:

```
create
database
student_enrol;

use student_enrol;
create table student(regno varchar(20),name VARCHAR(50),major
varchar(20),bdate date,primary key (regno));
create table course(courseno int,cname varchar(20),dept
VARCHAR(20),primary key(courseno));
create table enroll(regno varchar(20),courseno int,sem int,marks
int,primary key(regno,courseno),foreign key(regno) references
student(regno),
foreign key(courseno) references course(courseno));
create table text(book_isbn int,book_title varchar(50),publisher
varchar(50),author varchar(50),primary key(book_isbn));
create table book_adoption(courseno int,sem int,book_isbn int,primary
key(courseno,book_isbn),foreign key(courseno) references
course(courseno),
```

```

foreign key(book_isbn) references text(book_isbn));
insert into student values('CS01','RAM','DS','1986-03-12');
insert into student values('IS02','SMITH','USP','1987-12-23');
insert into student values('EC03','AHMED','SNS','1985-04-17');
insert into student values('CS03','SNEHA','DBMS','1987-01-01');
insert into student values('TC05','AKHILA','EC','1986-10-06');
commit;
select*from student;
insert into course values('11','DS','CS');
insert into course values('22','USP','IS');
insert into course values('33','SNS','EC');
insert into course values('44','DBMS','CS');
insert into course values('55','EC','TC');
commit;
select*from course;
insert into enroll values('CS01','11','4','85');
insert into enroll values('IS02','22','6','80');
insert into enroll values('EC03','33','2','80');
insert into enroll values('CS03','44','6','75');
insert into enroll values('TC05','55','2','56');
commit;
select*from enroll;
insert into text values('1','DS AND C REDDY','PRINCETON','PADMA');
insert into text values('2','FUNDAMENTALS OF DS','PRINCETON','GODSE');
insert into text values('3','FUNDAMENTALS OF
DBMS','PRINCETON','NAVATHE');
insert into text values('4','SQL','PRINCETON','FOLEY');
insert into text values('5','ELECTRONICS CIRCUITS','TMH','ELMASRI');
insert into text values('6','ADV UNIX PROG','TMH','STEVENS');
commit;
select*from text;
insert into book_adoption values('11','4','1');
insert into book_adoption values('11','4','2');
insert into book_adoption values('44','6','3');
insert into book_adoption values('44','6','4');
insert into book_adoption values('55','2','5');
insert into book_adoption values('22','6','6');
commit;
select*from book_adoption;

insert into text values('7','C PROGRAMMING','TMH','THAREJA');
insert into book_adoption values('55','2','7');
insert into book_adoption values(11,4,3);

```



```
select c.courseno,t.book_isbn,t.book_title from course c,book_adoption
ba,text t where c.courseno=ba.courseno and ba.book_isbn=t.book_isbn and
c.dept='CS'
```

```
and 2<(select count(book_isbn) from book_adoption b where
c.courseno=b.courseno) order by t.book_title;
```

```
select distinct c.dept from course c where c.dept in(select c.dept from
course c,book_adoption b,text t where c.courseno=b.courseno
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.courseno=b.courseno and t.book_isbn=b.book_isbn
and t.publisher !='PRINCETON');
```

OUTPUT:

The screenshot shows the MySQL Workbench interface. The 'Query' tab is active, displaying a SQL script with several INSERT statements and a SELECT statement. The 'Result Grid' shows the results of the SELECT statement, which is a self-join on the book_adoption table. The 'Output' tab shows the execution log, indicating that 6 rows were returned by the final SELECT statement.

Query 1:

```
39 * insert into book_adoption values('11','4','1');
40 * insert into book_adoption values('11','4','2');
41 * insert into book_adoption values('44','6','3');
42 * insert into book_adoption values('44','6','4');
43 * insert into book_adoption values('55','2','5');
44 * insert into book_adoption values('22','6','6');
45 * commit;
46 * select*from book_adoption;
47
```

Result Grid:

courseno	sem	book_isbn
11	4	1
11	4	2
22	6	6
44	6	3
44	6	4
55	2	5

Output:

#	Time	Action	Message	Duration / Fetch
39	11:54:00	insert into book_adoption values('11','4','1')	1 row(s) affected	0.000 sec
40	11:54:00	insert into book_adoption values('11','4','2')	1 row(s) affected	0.000 sec
41	11:54:00	insert into book_adoption values('44','6','3')	1 row(s) affected	0.000 sec
42	11:54:00	insert into book_adoption values('44','6','4')	1 row(s) affected	0.000 sec
43	11:54:00	insert into book_adoption values('55','2','5')	1 row(s) affected	0.000 sec
44	11:54:00	insert into book_adoption values('22','6','6')	1 row(s) affected	0.000 sec
45	11:54:00	commit	0 row(s) affected	0.000 sec
46	11:54:10	select*from book_adoption LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec

MySQL Workbench

Navigator

SCHEMAS

Filter objects

- banking
- book_dealer
- insurance
- order_processing
- sys

Administration Schemas

Information

No object selected

Query 1

```

45 * commit;
46 * select*from book_adoption;
47
48 * insert into text values('7','C PROGRAMMING','TMH','THAREJA');
49 * insert into book_adoption values('55','2','7');
50 * insert into book_adoption values('11','4','3');
51
52 * select c.courseno,t.book_isbn,t.book_title from course c,book_adoption ba,text t where c.courseno=ba.courseno and ba.book_isbn=t.book_isbn and c.dept='CS'
53 and 2(select count(book_isbn) from book_adoption b where c.courseno=b.courseno) order by t.book_title;

```

Result Grid

courseno	book_isbn	book_title
11	1	DS AND C REDDY
11	3	FUNDAMENTALS OF DBMS
11	2	FUNDAMENTALS OF DS

Output

Action Output

#	Time	Action	Message	Duration / Fetch
43	11:54:00	commit	0 row(s) affected	0.000 sec
44	11:54:10	select from book_adoption LIMIT 0.1000	6 row(s) returned	0.000 sec / 0.000 sec
45	11:55:25	insert into text values('7','C PROGRAMMING','TMH','THAREJA')	1 row(s) affected	0.000 sec
46	11:55:25	insert into book_adoption values('55','2','7')	1 row(s) affected	0.015 sec
47	11:55:25	insert into book_adoption values('11','4','3')	1 row(s) affected	0.000 sec
48	11:55:32	select c.courseno,t.book_isbn,t.book_title from course c,book_adoption ba,text t where c.courseno=ba.courseno...	3 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

MySQL Workbench

Navigator

SCHEMAS

Filter objects

- banking
- book_dealer
- insurance
- order_processing
- sys

Administration Schemas

Information

No object selected

Query 1

```

49 * insert into book_adoption values('55','2','7');
50 * insert into book_adoption values('11','4','3');
51
52 * select c.courseno,t.book_isbn,t.book_title from course c,book_adoption ba,text t where c.courseno=ba.courseno and ba.book_isbn=t.book_isbn and c.dept='CS'
53 and 2(select count(book_isbn) from book_adoption b where c.courseno=b.courseno) order by t.book_title;
54
55 * select distinct c.dept from course c where c.dept in(select c.dept from course c,book_adoption b,text t where c.courseno=b.courseno
56 and t.book_isbn=b.book_isbn and t.publisher='PRINCETON')
57 and c.dept not in(select c.dept from course c,book_adoption b,text t where c.courseno=b.courseno and t.book_isbn=b.book_isbn and t.publisher != 'PRINCETON');

```

Result Grid

dept
CS

Output

Action Output

#	Time	Action	Message	Duration / Fetch
44	11:54:10	select from book_adoption LIMIT 0.1000	6 row(s) returned	0.000 sec / 0.000 sec
45	11:55:25	insert into text values('7','C PROGRAMMING','TMH','THAREJA')	1 row(s) affected	0.000 sec
46	11:55:25	insert into book_adoption values('55','2','7')	1 row(s) affected	0.015 sec
47	11:55:25	insert into book_adoption values('11','4','3')	1 row(s) affected	0.000 sec
48	11:55:32	select c.courseno,t.book_isbn,t.book_title from course c,book_adoption ba,text t where c.courseno=ba.courseno...	3 row(s) returned	0.000 sec / 0.000 sec
49	11:55:49	select distinct c.dept from course c where c.dept in(select c.dept from course c,book_adoption b,text t where c.courseno=b.courseno...	1 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

MySQL Workbench

MySql@127.0.0.1:3306

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHMAS

Filter objects

- banking
- book_dealer
- insurance
- order_processing
- student_enrol
- sys

Tables

- book_adoption
- course
- enroll
- student
- text

Views

Stored Procedures

Functions

Administration Schemas

Information

Schema: student_enrol

Count: 5 Maintenance

Inspect Table Refresh

Output

Action Output

#	Time	Action	Message	Duration / Fetch
140	11:57:39	select distinct c.dept from course c where c.dept in(select c.dept from course c,book_adoption b, text t where c.	1 row(s) returned	0.000 sec / 0.000 sec
141	11:57:47	SELECT * FROM student_enrol,book_adoption LIMIT 0, 1000	8 row(s) returned	0.000 sec / 0.000 sec
142	11:57:49	SELECT * FROM student_enrol, course LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
143	11:57:50	SELECT * FROM student_enrol, enroll LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
144	11:57:51	SELECT * FROM student_enrol, student LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
145	11:57:53	SELECT * FROM student_enrol, text LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Type here to search

11:58 02-06-2021

MySQL Workbench

MySql@127.0.0.1:3306

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHMAS

Filter objects

- banking
- book_dealer
- insurance
- order_processing
- student_enrol
- sys

Tables

- book_adoption
- course
- enroll
- student
- text

Views

Stored Procedures

Functions

Administration Schemas

Information

Schema: student_enrol

Count: 18 Refresh

Output

Action Output

#	Time	Action	Message	Duration / Fetch
140	11:57:39	select distinct c.dept from course c where c.dept in(select c.dept from course c,book_adoption b, text t where c.	1 row(s) returned	0.000 sec / 0.000 sec
141	11:57:47	SELECT * FROM student_enrol,book_adoption LIMIT 0, 1000	8 row(s) returned	0.000 sec / 0.000 sec
142	11:57:49	SELECT * FROM student_enrol, course LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
143	11:57:50	SELECT * FROM student_enrol, enroll LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
144	11:57:51	SELECT * FROM student_enrol, student LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
145	11:57:53	SELECT * FROM student_enrol, text LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Type here to search

11:58 02-06-2021

MySQL Workbench

Navigator

Schemas

Filter objects

banking

book_dealer

insurance

order_processing

student_enrol

Tables

book_adoption

course

enroll

student

text

Views

Stored Procedures

Functions

sys

Administration

Schemas

Information

Schema: student_enrol

1 * SELECT * FROM student_enrol.book_adoption;

Result Grid

courseno	sem	book_idbn
11	4	1
11	4	2
11	4	3
22	6	6
44	6	3
44	6	4
55	2	5
55	2	7

Output

Action Output

#	Time	Action	Message	Duration / Fetch
140	11:57:39	select distinct c.dept from course c where c.dept in(select c.dept from course c,book_adoption b, text t where c...	1 row(s) returned	0.000 sec / 0.000 sec
141	11:57:47	SELECT * FROM student_enrol.book_adoption LIMIT 0, 1000	8 row(s) returned	0.000 sec / 0.000 sec
142	11:57:49	SELECT * FROM student_enrol.course LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
143	11:57:50	SELECT * FROM student_enrol.enroll LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
144	11:57:51	SELECT * FROM student_enrol.student LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
145	11:57:53	SELECT * FROM student_enrol.text LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec

Object Info

Session

Type here to search

11:58 02-06-2021

MySQL Workbench

Navigator

Schemas

Filter objects

banking

book_dealer

insurance

order_processing

student_enrol

Tables

book_adoption

course

enroll

student

text

Views

Stored Procedures

Functions

sys

Administration

Schemas

Information

Schema: student_enrol

1 * SELECT * FROM student_enrol.course;

Result Grid

courseno	cname	dept
11	DS	CS
22	USP	IS
33	SHS	EC
44	DBMS	CS
55	EC	TC

Output

Action Output

#	Time	Action	Message	Duration / Fetch
140	11:57:39	select distinct c.dept from course c where c.dept in(select c.dept from course c,book_adoption b, text t where c...	1 row(s) returned	0.000 sec / 0.000 sec
141	11:57:47	SELECT * FROM student_enrol.book_adoption LIMIT 0, 1000	8 row(s) returned	0.000 sec / 0.000 sec
142	11:57:49	SELECT * FROM student_enrol.course LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
143	11:57:50	SELECT * FROM student_enrol.enroll LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
144	11:57:51	SELECT * FROM student_enrol.student LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
145	11:57:53	SELECT * FROM student_enrol.text LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec

Object Info

Session

Type here to search

11:58 02-06-2021

MySQL Workbench

Navigator

Schemas

Filter objects

banking

book_dealer

insurance

order_processing

student_enrol

Tables

book_adoption

course

enroll

student

text

Views

Stored Procedures

Functions

sys

Administration

Schemas

Information

Schema: student_enrol

1 * SELECT * FROM student_enrol.enroll;

Result Grid

regno	courseno	sem	marks
CS01	11	4	85
CS03	44	6	75
EC03	33	2	80
IS02	22	6	80
TC05	55	2	56

enroll 1 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
140	11:57:39	select distinct c.dept from course c where c.dept in(select c.dept from course c,book_adoption b, text t where c...	1 row(s) returned	0.000 sec / 0.000 sec
141	11:57:47	SELECT * FROM student_enrol,book_adoption LIMIT 0, 1000	8 row(s) returned	0.000 sec / 0.000 sec
142	11:57:49	SELECT * FROM student_enrol, course LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
143	11:57:50	SELECT * FROM student_enrol, enroll LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
144	11:57:51	SELECT * FROM student_enrol, student LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
145	11:57:53	SELECT * FROM student_enrol, text LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec

Object Info

Session

Type here to search

11:58 02-06-2021

MySQL Workbench

Navigator

Schemas

Filter objects

banking

book_dealer

insurance

order_processing

student_enrol

Tables

book_adoption

course

enroll

student

text

Views

Stored Procedures

Functions

sys

Administration

Schemas

Information

Schema: student_enrol

1 * SELECT * FROM student_enrol.student;

Result Grid

regno	name	major	bdate
CS01	RAM	DS	1986-03-12
CS03	SHRHA	DBMS	1987-01-01
EC03	AHMED	SHS	1985-04-17
IS02	SMITH	UGP	1987-12-23
TC05	AKHILA	EC	1988-10-06

student 1 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
140	11:57:39	select distinct c.dept from course c where c.dept in(select c.dept from course c,book_adoption b, text t where c...	1 row(s) returned	0.000 sec / 0.000 sec
141	11:57:47	SELECT * FROM student_enrol,book_adoption LIMIT 0, 1000	8 row(s) returned	0.000 sec / 0.000 sec
142	11:57:49	SELECT * FROM student_enrol, course LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
143	11:57:50	SELECT * FROM student_enrol, enroll LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
144	11:57:51	SELECT * FROM student_enrol, student LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
145	11:57:53	SELECT * FROM student_enrol, text LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec

Object Info

Session

Type here to search

11:59 02-06-2021

MySQL Workbench

MySql@127.0.0.1:3306

File Edit View Query Database Server Tools Scripting Help

Navigator

book_adoption course enroll student text

Limit to 1000 rows

1 • SELECT * FROM student_enroll;text;

Result Grid

book_idbn	book_title	publisher	author
1	DS AND C REDDY	PRINCETON	PADMA
2	FUNDAMENTALS OF DS	PRINCETON	GODSE
3	FUNDAMENTALS OF DBMS	PRINCETON	NAVATHE
4	SQL	PRINCETON	POLEY
5	ELECTRONICS CIRCUITS	TMH	ELMASRI
6	ADV UNIX PROG	TMH	STEVENS
7	C PROGRAMMING	TMH	THAREJA

Schema: student_enroll

text 1 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
140	11:57:39	select distinct c.dept from course c where c.dept in(select c.dept from course c,book_adoption b;text t where c...	1 row(s) returned	0.000 sec / 0.000 sec
141	11:57:47	SELECT * FROM student_enroll;book_adoption LIMIT 0, 1000	8 row(s) returned	0.000 sec / 0.000 sec
142	11:57:49	SELECT * FROM student_enroll;course LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
143	11:57:50	SELECT * FROM student_enroll;enroll LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
144	11:57:51	SELECT * FROM student_enroll;student LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
145	11:57:53	SELECT * FROM student_enroll;text LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Type here to search

11:59 02-06-2021