

NIKHIL M ABRAHAM

USN: 1BM19CS101

**DATABASE MANAGEMENT SYSTEM
19CS4PCDBM**

LAB REPORT

Lab-1 : Insurance Database

```
show databases;
create database Insurance;
use Insurance;
create table PERSON(driver_id varchar(30) primary key, name varchar(30), address varchar(30));
create table CAR(Regno varchar(30) primary key, model varchar(30), year int);
create table ACCIDENT(report_number int primary key, adate date, location varchar(30));
create table OWNS(driver_id varchar(30), Regno varchar(30), primary key(driver_id,Regno), foreign
key(driver_id) references PERSON(driver_id), foreign key(Regno) references CAR(Regno));
create table PARTICIPATED(driver_id varchar(30), Regno varchar(30), report_number int,
damage_amount int, primary key(driver_id, Regno), foreign key(driver_id, Regno) references
OWNS(driver_id, Regno));
show tables;
insert into PERSON values('05C','Bob','Jamaica');
insert into CAR values('1E','Hundai','2015');
insert into ACCIDENT values('5','1999/09/02','Hubli');
insert into OWNS values('05C','1E');
insert into PARTICIPATED values('01A','1A',1,96000);
select * from CAR;

-- 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_amount=25000 where Regno='01A' AND report_number=1;
select * from PARTICIPATED;

-- Add a new accident to the database.

insert into ACCIDENT values('6','1967/09/02','Kerala');
select * from ACCIDENT;
```

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

```
select count(*) from ACCIDENT where adate>'2007/12/31' AND adate<'2009/01/01';
```

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

```
select count(ACCIDENT.report_number) from ACCIDENT,PARTICIPATED,CAR where  
ACCIDENT.report_number=PARTICIPATED.report_number AND PARTICIPATED.Regno=CAR.Regno  
AND CAR.model='Chevy';
```

The screenshot shows a SQL query execution interface. The query is: `select count(*) from ACCIDENT where adate>'2007/12/31' AND adate<'2009/01/01';`. The result grid shows a single row with the value 0.

count(*)
0

The screenshot shows a SQL query execution interface. The query is: `select count(ACCIDENT.report_number) from ACCIDENT,PARTICIPATED,CAR where ACCIDENT.report_number=PARTICIPATED.report_number AND PARTICIPATED.Regno=CAR.Regno AND CAR.model='Chevy';`. The result grid shows a single row with the value 1.

count(ACCIDENT.report_number)
1

Lab-2 : Book Database

```
create database Book;
```

```
use Book;
```

```
create table AUTHOR( author_id int primary key, name varchar(30), city varchar(20), country  
varchar(20));
```

```
create table PUBLISHER( publisher_id int primary key, name varchar(30), city varchar(20), country  
varchar(20));
```

```
create table CATALOG( book_id int, title varchar(30), 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), foreign  
key(publisher_id) references PUBLISHER(publisher_id), foreign key(category_id) references  
CATEGORY(category_id));
```

```
create table CATEGORY( category_id int, description varchar(50), primary key(category_id));
```

```
create table ORDER_DETAILS( order_no int primary key, book_id int, quantity int, foreign key(book_id)  
references CATALOG(book_id));
```

```
insert into AUTHOR values(1005,'WILLIAMS STALLINGS','LAS VEGAS','USA');
```

```
insert into PUBLISHER values(5,'MGH','NEW YORK','USA');
insert into CATEGORY values(1005,'OPERATING SYSTEMS');
insert into CATALOG values(17,'COBOL Handbook',1005,4,1001,2000,658);
insert into ORDER_DETAILS values(2,17,10);
```

```
select A.name,C.title,C.price from AUTHOR A,CATALOG C where C.author_id=A.author_id and
C.year>=2000 and A.name=(select A.name from AUTHOR A,CATALOG C where
A.author_id=C.author_id group by C.author_id having count(*)>=2);
```

```
select A.name from AUTHOR A,CATALOG C,ORDER_DETAILS O where O.book_id=C.book_id and
A.author_id=C.author_id and O.book_id=(select book_id from ORDER_DETAILS where quantity=(select
max(quantity) from ORDER_DETAILS));
```

```
update CATALOG set price=1.10*price;
select * from CATALOG;
```

```

19
20 • select A.name from AUTHOR A,CATALOG C,ORDER_DETAILS O where O.book_id=C.book_id and A.author_id=C.author_id and O.book_id=(select b
21
22

```

1:22

Result Grid Filter Rows: Search Export:

name
KARTHIK B.P.

Result 2 Read Only

```

22
23 • update CATALOG set price=1.10*price;
24 • select * from CATALOG;

```

23:24

Result Grid Edit: Export/Import:

book_id	title	author_id	publisher_id	category_id	year	price
11	Unix System Prg	1001	1	1001	2000	276
12	Digital Signal	1002	2	1003	2001	468
13	Logic Design	1003	3	1002	1999	248
14	Server Prg	1004	4	1004	2001	366
15	Linux OS	1005	5	1005	2003	359
16	C++ Bible	1005	5	1001	2000	579
17	COBOL Handbook	1005	4	1001	2000	724
NULL	NULL	NULL	NULL	NULL	NULL	NULL

CATALOG 3 Apply

```

14
15
16 • select A.name,C.title,C.price from AUTHOR A,CATALOG C where C.author_id=A.author_id and C.year>=2000 and A.name=(select A.name from
17

```

139:16

Result Grid Filter Rows: Search Export:

name	title	price
WILLIAMS STALLINGS	Linux OS	326
WILLIAMS STALLINGS	C++ Bible	526
WILLIAMS STALLINGS	COBOL Handbook	658

Result 1 Read Only

Lab-3 : Orders Database

```

create database order_processing1;
use order_processing1;
create table customer(cust int primary key,cname varchar(20),city varchar(20));
create table order_(order_no int primary key,odate date,cust int ,ord_amt int,
foreign key(cust) references customer(cust) on delete cascade);
create table item(item_no int primary key,unit_price int);
create table order_item(order_no int,item_no int ,qty int,
foreign key(order_no) references order_(order_no)on delete cascade,
foreign key(item_no) references item(item_no)on delete cascade);
create table warehouse(warehouse_no int primary key,city varchar(20));

```

```
create table shipment(order_no int,warehouse_no int ,ship_date date,
foreign key(order_no) references order_(order_no) on delete cascade,
foreign key(warehouse_no) references warehouse(warehouse_no) on delete cascade);
show tables;
drop table order_item;
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");
select * from customer;
```

```
insert into order_ values(111,'2002-01-22',771,18000);
insert into order_ values(112,'2002-07-30',774,6000);
insert into order_ values(113,'2003-04-03',775,9000);
insert into order_ values(114,'2003-11-03',775,29000);
insert into order_ values(115,'2003-12-10',773,29000);
insert into order_ values(116,'2004-08-19',772,56000);
insert into order_ values(117,'2004-09-10',771,20000);
insert into order_ values(118,'2004-11-20',775,29000);
insert into order_ values(119,'2005-02-13',774,29000);
insert into order_ values(120,'2005-10-13',775,29000);
select * from order_;
```

```
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);
select * from item;
```

```
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);
select * from order_item;
```

```
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");
select * from warehouse;
```

```
insert into shipment values(111,1,'2002-02-10');
insert into shipment values(112,5,'2002-09-10');
insert into shipment values(113,8,'2003-02-10');
insert into shipment values(114,3,'2003-12-10');
insert into shipment values(115,9,'2004-01-19');
insert into shipment values(116,1,'2004-09-20');
insert into shipment values(117,5,'2004-09-10');
insert into shipment values(118,7,'2004-11-30');
insert into shipment values(119,7,'2005-04-30');
insert into shipment values(120,6,'2005-12-21');
select * from shipment;
```

-- 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.

```
select c.cname,count(o.order_no) as total_orders,avg(o.ord_amt) as average_amount from customer
c,order_ o
where c.cust=o.cust group by o.cust;
```

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

```
select s.order_no from shipment s,warehouse w
where s.warehouse_no=w.warehouse_no and w.city="delhi";
```

-- select s.order_no from shipment s where s.warehouse_no in(select w.warehouse_no from warehouse
w where w.city="delhi");

-- v) 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 order_item;
```

```

97 • delete from item where item_no=5005;
98 • select * from order_item;

```

100% 26:98

Result Grid Filter Rows: Search Export:

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

cname	total_orders	average_amount	
pushpa k	2	19000.0000	
suman	1	56000.0000	
sourav	1	29000.0000	
laila	2	17500.0000	
faizal	4	24000.0000	

order_no	
111	
116	
120	
115	

Lab-4 : Bank Database

```
create database banking_enterprise;
```

```
use banking_enterprise;
```

```
create table branch(branch_name varchar(20) primary key,branch_city varchar(20),assets real);
```

```
create table accounts(acc_no int primary key,branch_name varchar(20),balance real, foreign
key(branch_name)
references branch(branch_name) on delete cascade);
```

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

```
create table depositor(customer_name varchar(20),acc_no int,
foreign key(customer_name) references customer(customer_name) on delete cascade,
foreign key(acc_no) references accounts(acc_no) on delete cascade);
```

```
create table loan(loan_number int primary key,branch_name varchar(20),amount int,
```



```
foreign key(branch_name) references branch(branch_name) on delete cascade);
```

```
create table borrower(customer_name varchar(20),loan_number int,  
foreign key(customer_name) references customer(customer_name) on delete cascade,  
foreign key(loan_number) references loan(loan_number) on delete 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","Bangalore",550000);  
select * from branch;
```

```
insert into accounts values(11,"SBI Hosakerehalli",5000);  
insert into accounts values(22,"SBI Vijay Nagar",5000);  
insert into accounts values(33,"SBI Jayanagar",5000);  
insert into accounts values(44,"SBI Rajaji Nagar",10000);  
insert into accounts values(55,"SBI Vijay Nagar",40000);  
insert into accounts values(66,"SBI PD Nagar",4000);  
insert into accounts values(77,"SBI PD Nagar",40000);  
insert into accounts values(88,"SBI Rajaji Nagar",4000);  
select * from accounts;
```

```
insert into customer values("Kezar","MG road","Bangalore");  
insert into customer values("Lal Krishna","ST MKS road","Bangalore");  
insert into customer values("Rahul","Augsten road","Bangalore");  
insert into customer values("Lallu","V S road","Bangalore");  
insert into customer values("Faizal","Resedency road","Bangalore");  
insert into customer values("Rajeev","Dicknsn road","Bangalore");  
select * from customer;
```

```
insert into depositor values("Rahul",11);  
insert into depositor values("Lallu",22);  
insert into depositor values("Rahul",33);  
insert into depositor values("Faizal",44);  
insert into depositor values("Lallu",55);  
insert into depositor values("Kezar",66);  
insert into depositor values("Rajeev",77);  
insert into depositor values("Lal Krishna",88);  
select * from depositor;
```

```
insert into loan values(10011,"SBI Jayanagar",10000);  
insert into loan values(10012,"SBI Vijay Nagar",5000);  
insert into loan values(10013,"SBI Hosakerehalli",20000);  
insert into loan values(10014,"SBI PD Nagar",15000);  
insert into loan values(10015,"SBI Rajaji Nagar",25000);
```

```
select * from loan;
```

```
insert into borrower values("Kezar",10011);  
insert into borrower values("Lal Krishna",10012);  
insert into borrower values("Rahul",10013);  
insert into borrower values("Lallu",10014);  
insert into borrower values("Lal Krishna",10015);  
select * from borrower;
```

```
-- iii) Find all the customers who have at least two accounts at the Main branch.  
select d.customer_name from depositor d,accounts a where d.acc_no=a.acc_no and  
a.branch_name="SBI Vijay Nagar"  
group by d.customer_name having count(d.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.acc_no = depositor.acc_no  
join branch on branch.branch_name = accounts.branch_name  
where branch.branch_city = "Bangalore"  
GROUP BY depositor.customer_name;
```

```
-- 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;
```

81 • `select customer_name from depositor`
82 `join accounts on accounts.acc_no = depositor.acc_no`
83 `join branch on branch.branch_name = accounts.branch_name`
84

100% 35:82

Result Grid Filter Rows: Search Export:

customer_name
Rahul
Kezar
Rajeev
Faizal
Lal Krishna

88 `-- v) Demonstrate how you delete all account tuples at every`
89 `-- branch located in a specific city.`
90 • `delete from accounts where branch_name in`
91 `(select branch_name from branch where branch_city="delhi");`
92 • `select * from accounts;`

100% 12:92

Result Grid Filter Rows: Search Edit: Export/Import:

acc_no	branch_name	balance
22	SBI Vijay Nagar	5000
44	SBI Rajaji Nagar	10000
55	SBI Vijay Nagar	40000
66	SBI PD Nagar	4000
77	SBI PD Nagar	40000
88	SBI Rajaji Nagar	4000
NULL	NULL	NULL

75 `-- iii) Find all the customers who have at least two accounts at the Main branch.`
76 • `select d.customer_name from depositor d,accounts a where d.acc_no=a.acc_no and a.branch_name="SBI Vijay Nagar"`
77 `group by d.customer_name having count(d.customer_name)>=2;`
78
79 `-- iv) Find all the customers who have an account at all the`

100% 22:76

Result Grid Filter Rows: Search Export:

customer_name
Lallu

Lab-5 : Student Enroll Database

```
create database Student_Enrollment;
use Student_enrollment;
create table student(regno varchar(10) primary key,name varchar(10),major varchar(10),bdate date);
```

```

create table course(course_no int primary key,cname varchar(10),dept varchar(10));
create table enroll(regno varchar(10),course_no int,sem int, marks int,
foreign key(regno) references student(regno) on delete cascade,
foreign key(course_no) references course(course_no) on delete cascade);
create table text_book(book_isbn int primary key,book_title varchar(20),publisher varchar(10),author
varchar(10));
create table book_adoption(course_no int,sem int,book_isbn int ,
foreign key(course_no) references course(course_no) on delete cascade,
foreign key(book_isbn) references text_book(book_isbn) on delete cascade);

```

```

insert into student(regno,name,major,bdate) 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');
select * from student;

```

```

insert into course(course_no,cname,dept) values
(11,"ds","cs"),
(22,"usp","is"),
(33,"sns","ec"),
(44,"dbms","cs"),
(55,"ec","tc");
select * from course;

```

```

insert into enroll(regno,course_no,sem,marks) values
("cs01",11,4,85),
("is02",22,6,80),
("ec03",33,2,80),
("cs03",44,6,75),
("tc05",55,2,80);
select * from enroll;

```

```

insert into text_book(book_isbn,book_title,publisher,author) values
(1,"ds and c","princeton","padma"),
(2,"fundamentals of ds","princeton","godse"),
(3,"fundamentals of dbms","princeton","navathe"),
(4,"sql","princeton","foley"),
(5,"electronic circuits","tmh","elmarsi"),
(6,"adv unix program","tmh","stevens");
select * from text_book;

```

```

insert into book_adoption(course_no,sem,book_isbn) values
(11,4,1),(11,4,2),(44,6,3),(44,6,4),(55,2,5),(22,6,6);
select * from book_adoption;

```

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

```
insert into text_book values(7,"database basics","princeton","shawn");
insert into book_adoption values(11,4,7);
```

-- 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.book_isbn,t.book_title from course c, text_book t,book_adoption b
where t.book_isbn=b.book_isbn and b.course_no=c.course_no and c.dept="cs" and
(select count(b.book_isbn) from book_adoption b where c.course_no=b.course_no)>2 order by
t.book_title;
```

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

```
select distinct c.dept from course c where c.dept in (select c.dept
from course c,book_adoption b,text_book t where c.course_no=b.course_no
and t.book_isbn=b.book_isbn and t.publisher="tmh")
and c.dept not in (select c.dept
from course c,book_adoption b,text_book t where c.course_no=b.course_no
and t.book_isbn=b.book_isbn and t.publisher!="tmh");
```

```
56 • select c.course_no,t.book_isbn,t.book_title from course c, text_book t,book_adoption b
57 where t.book_isbn=b.book_isbn and b.course_no=c.course_no and c.dept="cs" and
58 (select count(b.book_isbn) from book_adoption b where c.course_no=b.course_no)>2 order by t.book_title;
59
```

course_no	book_isbn	book_title
11	7	database basics
11	1	ds and c
11	2	fundamentals of ds

```
60 -- List any department that has all its adopted books published by a specific publisher.
61 • select distinct c.dept from course c where c.dept in (select c.dept
62 from course c,book_adoption b,text_book t where c.course_no=b.course_no
63 and t.book_isbn=b.book_isbn and t.publisher="tmh")
64 and c.dept not in (select c.dept
65 from course c,book_adoption b,text_book t where c.course_no=b.course_no
66 and t.book_isbn=b.book_isbn and t.publisher!="tmh");
```

dept
is
tc

```
50 -- Demonstrate how you add a new text book to the database and make this book be adopted by some department.
51 • insert into text_book values(7,"database basics","princeton","shawn");
52 • insert into book_adoption values(11,4,7);
```