## **NIKHIL M ABRAHAM**

USN: 1BM19CS101

# DATABASE MANAGEMENT SYSTEM 19CS4PCDBM

**LAB REPORT** 

### Lab-1: Insurance Database

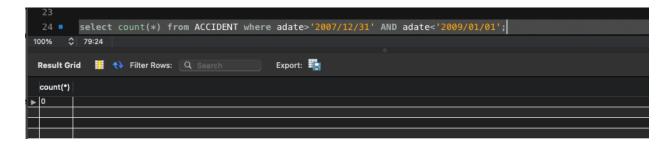
```
show databases:
create database Insurance;
use Insurance:
create table PERSON(driver id varchar(30) primary key, name varchar(30), address varchar(30));
create table CAR(Regno varchar(30) primary key, model varchar(30), year int);
create table ACCIDENT(report number int primary key, adate date, location 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';





#### 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;



#### 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");
```

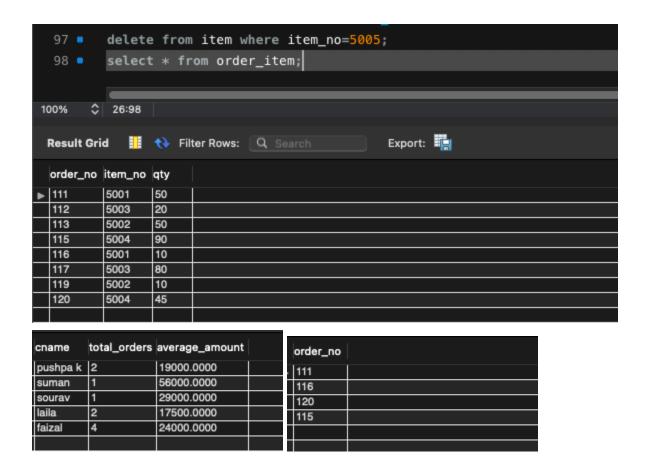
```
insert into warehouse values(7,"bombay");
insert into warehouse values(8,"chennai");
insert into warehouse values(9,"delhi");
insert into warehouse values(10,"bangalore");
select * from warehouse:
insert into shipment values(111,1,'2002-02-10');
insert into shipment values(112,5,'2002-09-10');
insert 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 theORDER ITEM table.

delete from item where item\_no=5005;

select \* from order item;



#### 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,

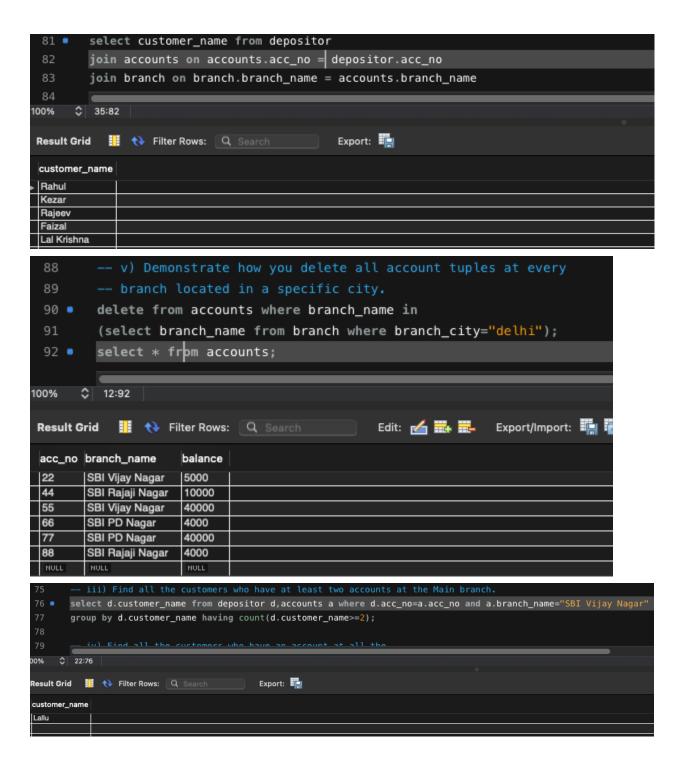
```
foreign key(branch name) references branch(branch name) on delete cascade);
create table borrower(customer_name varchar(20),loan_number int,
foreign key(customer name) references customer(customer name) on delete cascade.
foreign key(loan_number) references loan(loan_number) on delete cascade);
show 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 Ioan 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;



#### 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");

