

SCHEMA DIAGRAM

Student

snum	sname	major	level	age
------	-------	-------	-------	-----

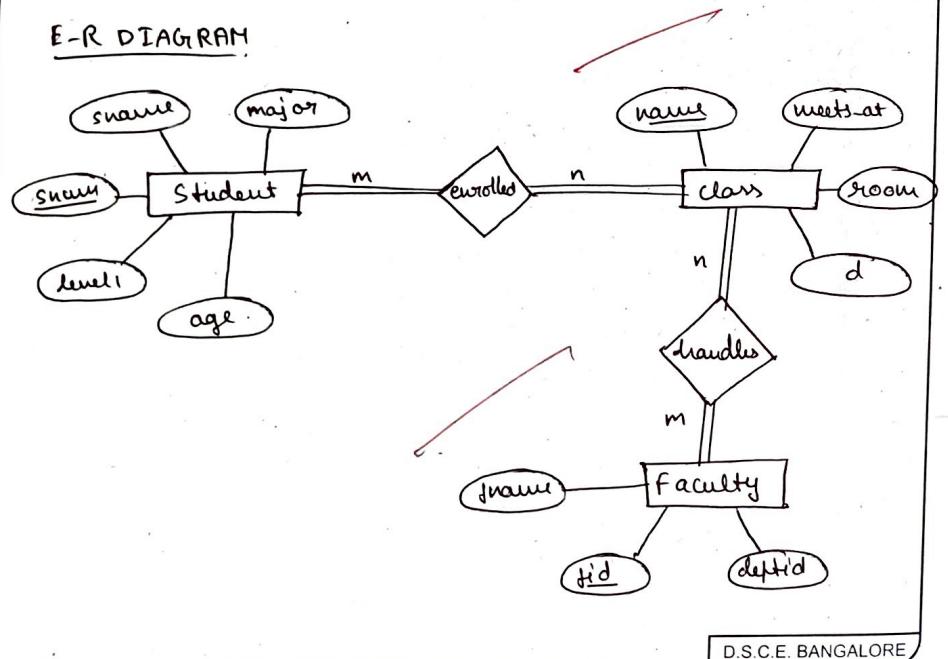
class

cname	meets-at	room	d
snum	cname		

enrolled

f_id	f_name	dept_id
------	--------	---------

E-R DIAGRAM



DD	MM	YY	YY
04	09	20	19

1. STUDENT DATABASE

Consider the following relations:

STUDENT (snum: integer, sname: string, major: string, level: string, age: integer)

CLASS (name: string, meets-at : string, room: string, d: integer)

ENROLLED (snum: integer, cname: string)

FACULTY (fid : integer, fname : string, deptid : integer)

Design an ER model and schema diagram for the relations.

Write the following queries in SQL. No duplicates should be printed in any of the answers.

- 1) Create the tables for the schemas provided with primary keys and foreign keys.
- 2) Insert five tuples of values to each table
- 3) Find the names of all juniors (level=jr) who are enrolled in a class taught by Prof. Harrith
- 4) Find the names of all classes that either meet in room R128 or have 5 or more students enrolled.
- 5) Find the names of faculty members for whom the

Student

SNUM	SNAME	MAJOR	LEVELI	AGE
001	Ayush	CSE	JM	20
002	Apoorva	ECE	SR	21
003	Barkha	CSF	JR	23
004	Viraj	ISE	SR	25
005	Ahsaas	ECE	JR	30

Faculty

FID	FNAME	DEPTID
304	Harshith	1
312	Anupama	2
215	Ram	3

Class

NAME	MEETSAT	ROOM	D
dbms	10:00	R128	304
java	09:00	L04	312
ME	11:00	S501	315

DD	MM	YY	YY
----	----	----	----

combined enrollment of the courses that they teach is less than 5.

- 1) > create table student (snum number primary key, sname varchar(15), major varchar(10), leveli varchar(5), age number(2));
- > create table class (cname varchar(20) primary key, meets_at varchar(10), room varchar(10), d number, foreign key (d) references faculty (fid));
- > create table faculty (fid number primary key, fname varchar(10), deptid number);
- > create table enrolled (snum number, cname varchar(20), primary key (snum, cname), foreign key (snum) references student (snum), foreign key (cname) references class (cname));

Enrolled.

SNUM	CNAME
001	dbms
002	dbms
003	dbms
004	dbms
005	jana
006	dbms
007	jana
008	me
009	me
010	dbms

3)

SNAME
Ayush
Barkha

4)

NAME
dbms

5)

FNAME
Anupama
Ram.

DBLIMMYY**

--	--	--	--

- 2) >insert into student values ('001', 'Ayush', 'CSE', 'Jr', 20);
>insert into faculty values ('304', 'Harshith', 1);
>insert into class values ('&name', 'Meet', 'at', '2 room', 'Bd'); or
>insert into class values ('dbms', '10:00', 'R128', '304');
>insert into enrolled values ('001', 'dbms');

3) >select sname from student s, faculty f, enrolled e, class c where s.snum = e.snum and e.cname = c.name and c.d = f.fid and s.level = 'Jr' and f.fname = 'Harshith';

4) >select name from class where room = 'R128'
union (select cname from enrolled group by cname having count(*) >= 5);

5) >select distinct f.fname from faculty f where
s > (select count(e.snum) from class c, enrolled e where c.name = e.cname and f.fid = c.d);

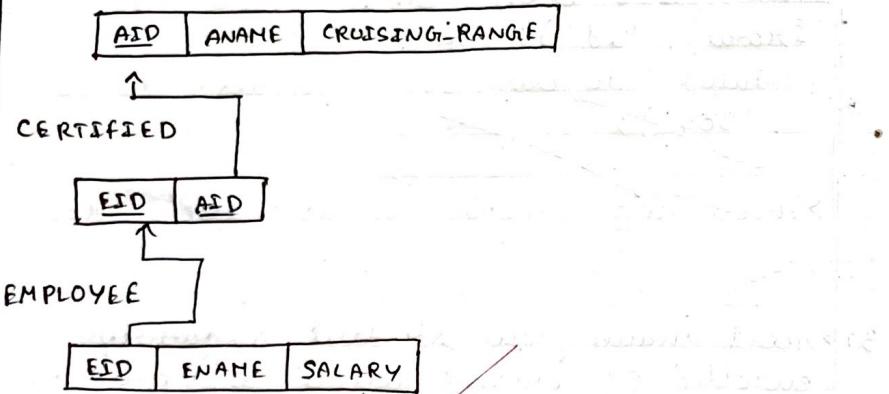
DD	MM	YY	YY
25	09	2019	

SCHEMA DIAGRAM

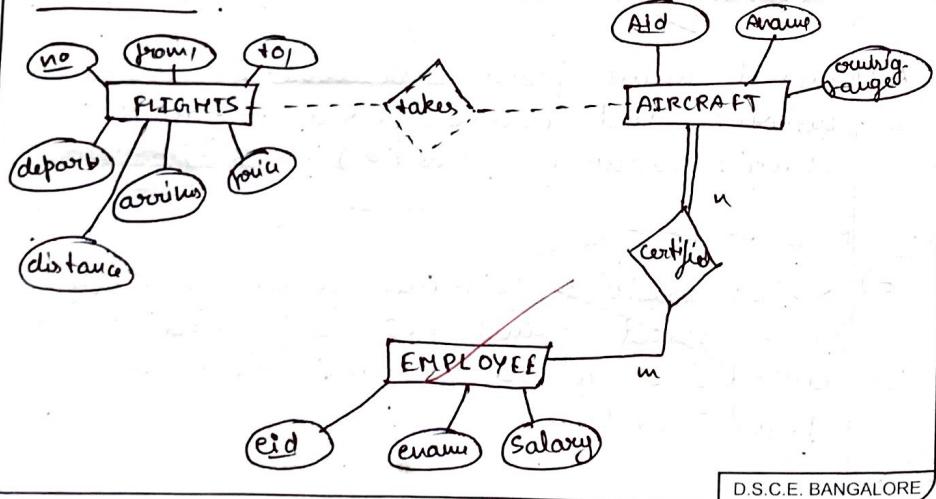
FLIGHTS

NO	FROM	TO	DISTANCE	DEPARTS	ARRIVES	PRICE
----	------	----	----------	---------	---------	-------

AIRCRAFT



ER DIAGRAM



2. FLIGHT DATABASE

The following relations keep track of airline flight information:

FLIGHTS (no:integer, from :String ,to :String , distance: integer, departs :varchar , arrives : time , price: real)..

AIRCRAFT (aid :integer, aname :String ,cruising_range: integer)

CERTIFIED (eid :integer , aid :integer)

EMPLOYEE (eid :integer , ename :String , salary :integer)

Design an ER diagram & schema diagram.

a) Create the tables for the schemas provided with primary keys & foreign keys.

b) Insert 5 tuples of values to each table.

c) Find the names of aircraft such that all pilots certified to operate them have salaries more than Rs. 80,000.

d) For each pilot, who is certified for more than three aircraft for which she or he is certified.

e) Find the names of pilots whose salary is less than the price of the cheapest route from Bangalore to Frankfurt.

FLIGHTS

NO	FROM	TO	DIST	DEPARTS	ARRIVES	PRICE
1	Bangalore	Frankfurt	2000	11 am	5 pm	10000
2	Bangalore	Frankfurt	2000	12 pm	6 pm	13000
3	Delhi	Germany	3000	1 am	5 am	40000
4	Calcutta	Bangalore	600	5 am	8 am	7000
5	Gujrat	Bengal	400	9 am	11 am	4000

AIRCRAFT

AID	ANAME	CRUISING-RANGE
101	Airlines	700
112	AirIndia	800
115	Indigo	500
104	Jetlines	700

EMPLOYEE

EID	FNAME	SALARY
1	Maira	9000
2	Sourab	90000
3	Harsha	160000
4	Lakshmi	50000
5	Richa	80000

DD	MM	YY	YY

a) create table flights (no number primary key , from varchar(15) , to varchar(15) , distance number, departs varchar(6) , price number(10, 2) , arrives varchar(10));

create table aircraft (aid number primary key , aname varchar(10) , cruising range number(10, 2));

create table employee (eid number primary key , varchar(10) , salary number(10));

create table certified (eid number, aid number primary key (eid, aid), foreign key (eid) references employee(eid), foreign key (aid) references aircraft(aid));

b) >insert into flights values ('&price', '&no', '&from', '&to', '&distance', '&departs', '&arrives');

>insert into aircraft values (&aid, '&aname', &cruising-range);

>insert into employee values (&eid, &salary);

>insert into certified values (&eid, &aid);

DD	MM	YY	YY
----	----	----	----

CERTIFIED

EID	AID
5	101
5	112
5	115
5	104
2	112
3	112

a. ANAME

Airlindia

b. EID nofaircrafts man(cruisingrange)

5	4	800
---	---	-----

c. ENAME

Maira

- a. > select distinct a.aname from aircraft a where a.aid not in (select c.aid from certified c, employee e where e.eid = c.cid and e.salary < 80000) ;
- b. > Select c.cid , max(cruising range) , count(*) nofaircrafts from certified c, aircraft a where a.aid = c.cid group by c.cid having count(c.cid) > 3 ;
- c. > select distinct ename from employee e , certified c where e.cid = c.cid and salary < (select min(price) from flights f where from = 'Bangalore' and to = 'frankfurt') ;

(1)

9/10/19

SCHEMA DIAGRAM

Author

<u>aid</u>	name	city	country
------------	------	------	---------

Publisher

<u>pid</u>	name	city	country
------------	------	------	---------

Catalog.

book-id	publisher-id	author-id	category-id	title	price	year
---------	--------------	-----------	-------------	-------	-------	------

Order-details.

<u>ono</u>	book-id	quantity
------------	---------	----------

Category.

<u>category-id</u>	description
--------------------	-------------

DD	MM	YY	YY
09	10	2019	

3. The following table 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)

Design an ER model for the schemas provided with primary keys and foreign keys.

a) Insert five tuples of values to each table

b) Create the tables for the schemas.

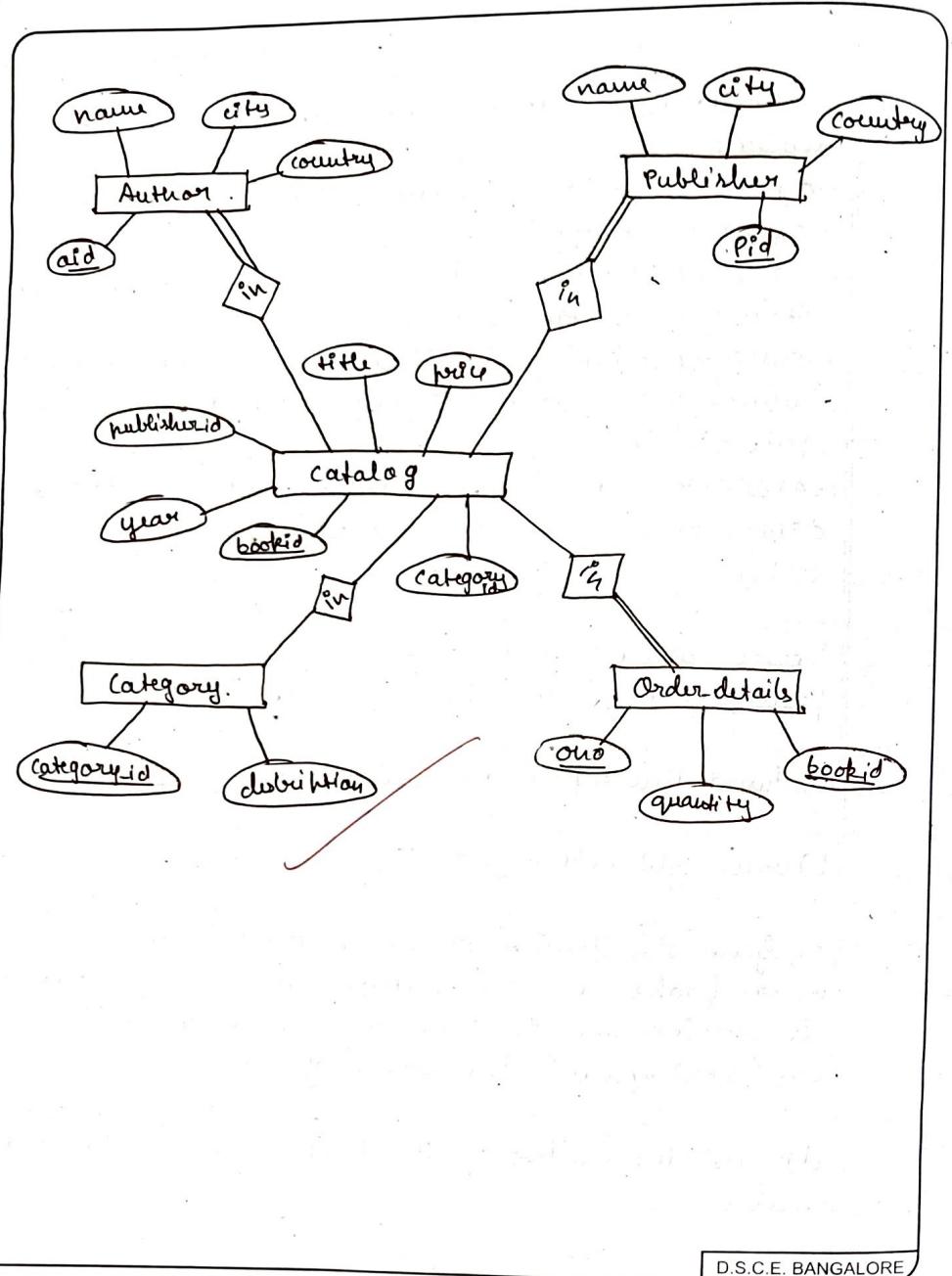
c) List the details of the authors who have 2 or more books in the catalog and price of the book is greater than the average of price of books in the catalog and the year of 2000.

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

D	D	M	M	Y	Y	Y	Y
---	---	---	---	---	---	---	---

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

- a) > create table author (author_id number(5) primary key, name varchar(10), city varchar(20), country varchar(20));
- > create table publisher (publisher_id number(5) primary key, name varchar(10), city varchar(20), country varchar(20));
- > create table category (category_id number(5) primary key, description varchar(20));
- > create table catalog (book_id number(5) primary key, title varchar(10), author_id number(5), publisher_id number(5), category_id number(5), year number(4), price number(6,2), 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 order_details (ono number(10), book_id number(5), quantity number(4), primary key (ono, book_id));



DD	MM	YY	YY
----	----	----	----

Author:

AUTHOR-ID	NAME	CITY	COUNTRY
1	Dan Brown	London	UK
2	Sudeep	Gujarat	India
4	Sheldon	Sidney	Australia
5	Kalan	Tamil Nadu	India
3	Ann Frank	Berlin	Germany

Publisher:

PUBLISHER-ID	NAME	CITY	COUNTRY
11	Pearson	Mumbai	India
12	Sapna	Karnataka	India
14	Dmark	Copen Hagen	Den Mark

CATEGORY:

CATEGORY-ID	DESCRIPTION
1001	Romantic
1002	Mystery
1003	Motivation

Catalog.

BOOK-ID	TITLE	AUTHOR-ID	PUB-ID	CATID	YEAR	PRICE
2001	DaVinci	1	11	1002	2002	700
2002	Inferno	1	12	1002	2005	500
2004	Diary	3	14	1003	1999	400
2005	Password	2	11	1001	2001	200
2006	Wings of fire	5	12	1003	1978	300

- b) > insert into author values (&author_id, '&name', '&city', '&country');
 > insert into publisher values (&publisher_id, '&name', '&city', '&country');
 > insert into category values (&book_id, '&title', &author_id, &publisher_id, &category_id, '&year', &price);
 > insert into category values (&category_id, '&description');
 > insert into category values order_details values (&ono, &book_id, &quantity);
 > select * from author;
 > select * from publisher;
 > select * from category;
 > select * from catalog;
 > select * from order_details;

Order details:

ONO	BOOK-ID	QUANTITY
20	2001	1
25	2002	2
26	2004	1
27	2006	1

AUTHOR-ID	NAME	CITY	COUNTRY
1	Dan Brown	London	UK
1	Dan Brown	London	UK

d) NAME
Anne Frank

e) BOOK-ID TITLE PRICE PRICE * 0.1
2006 Wings of fire 300 30.

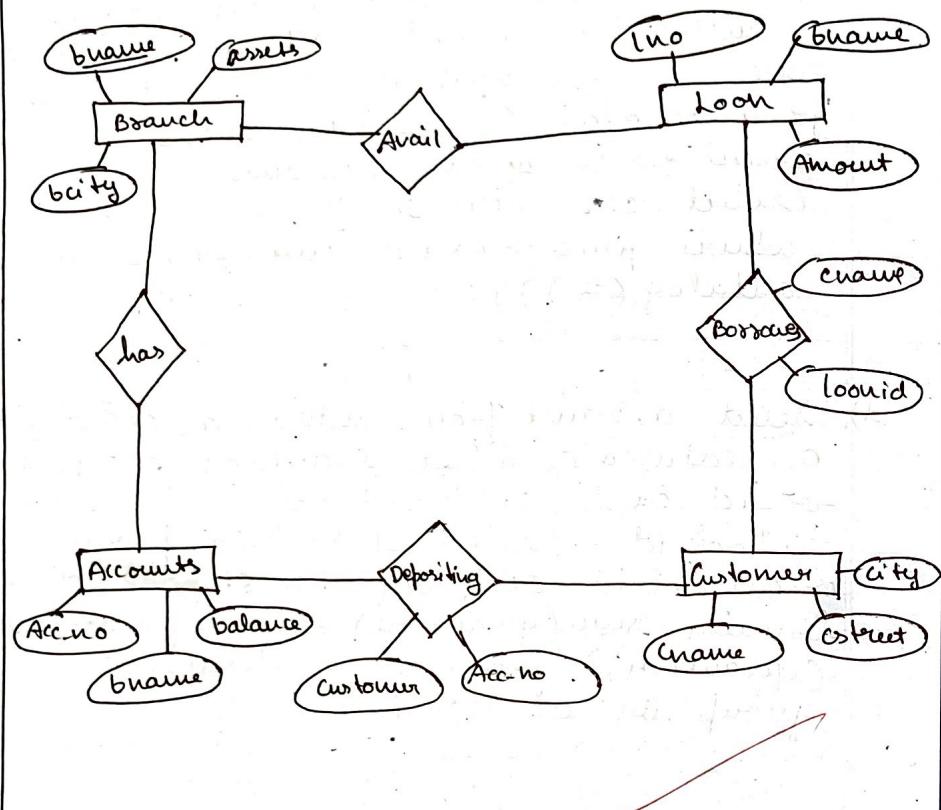
DD	MM	YY	YY	YY
----	----	----	----	----

c) > select * from author a, category c where a.author_id = c.author_id and year > 2000 and c.author_id in (select c.author_id from catalog c, group by author_id having count(*) >= 2 and c.author_id in (select c2.author_id from catalog c2 where price > (select avg(price) from catalog (3))));

d) select a.name from author a, order_details o, catalog c where a.author_id = c.author_id and c.book_id = o.book_id and o.book_id = (select o1.book_id from order_details o1 group by o1.book_id having sum(quantity) = (select max(sum(quantity)) from order_details o2 group by o2.book_id));

e) select book_id title, price, price + 1.1 from catalog publisher where catalog.publisher_id = publisher.publisher_id and name = 'Shane'

6/11/19

SCHEMA DIAGRAM:

DD	MM	YY	YY
13	11	2019	

- 4) Consider the following database for a banking enterprise:

```

Branch(bname: String, bcity: String, assets: real)
Accounts(accno: Int, bname: String, bal: real)
Depositor(cname: String, accno: int)
Customer(cname: String, cstreeet: string, city: String)
Loan(lno: Int, bname: String, amount: real)
Borrower(cname: String, lno: int)

```

- Create the tables for the schemas provided with primary keys and foreign keys.
- Insert five tuples of values to each table.
- Find all the customers who have at least two accounts at the main branch.
- Demonstrate how you delete all account tuples at every branch located in specific city.
- ~~Find all the customers who have an account at all the branches located in specific city.~~

SCHEMA DIAGRAM

BRANCH

Bname	Bcity	Arrears
-------	-------	---------

ACCOUNT

Ancno	Bname	Balance
-------	-------	---------

DEPOSITOR

Cname	Ancno
-------	-------

CUSTOMER

Cname	Cstreet	City
-------	---------	------

LOAN

Lno	Bname	Amount
-----	-------	--------

BORROWER

Cname	Lno
-------	-----

23

DD	MM	YY	YY	YY
----	----	----	----	----

24

a) > create table branch (Bname varchar(20) primary key, Bcity varchar(10), arrears number(10)) ;

> create table account (Ancno number(10) primary key, Bname varchar(20), balance int(10), foreign key (Bname) references branch (Bname)) ;

> create table customer (Cname varchar(20) primary key, Cstreet varchar(20), custcity varchar(20)) ;

> create table depositor (Cname varchar(20), Ancno number(10), primary key (Cname, Ancno), foreign key (Cname) references customer (Cname), foreign key (Ancno) references account (Ancno)) on delete cascade;

> create table loan (Lno number(10) primary key, Bname varchar(20), amount number(10), foreign key (Bname) references branch (Bname)) ;

> create table borrower (Cname varchar(20), Lno number(10), foreign key (Cname) references customer (Cname), foreign key (Lno) references loan (Lno)) ;

BRANCH.

	Branch	City	Assets
Bname	Bangalore		100,000
Nagarbhavi	Bangalore		20,00,000
RR Nagar	Hospet		10,00,000
Kengeri	Belgum		40,00,000
Majestic	CKM		30,00,000
KS Layout			

ACCOUNTS

	Branch	Balance
Ano	Bname	
001	Kengeri	5,00,000
002	Nagarbhavi	10,00,000
003	RR Nagar	15,00,000
004	Kengeri	20,00,000
005	KS Layout	25,00,000

CUSTOMER

	Customer	City
Anu	G street	Bangalore
Priya	7 th Gross	
Pragna	8 th Gross	Hospet
Ramya	15 th Gross	Bangalore
Midhu	14 th Gross	Belgum
Jaya	18 th Gross	CKM

DD	MM	YY	YY
----	----	----	----

- b) > insert into branch values ('&Bname', '&City')
- > insert into accounts values ('&Ano', '&Branch', '&balance')
- > insert into customer values ('&cname', '&street', '&city');
- > insert into depositor values ('& cname', '&Aho', '&Account');
- > insert into loan values ('&ano', '&Bname', '&Account');
- > insert into borrower values ('&cname', '&lno');
- ~~> select * from branch;~~
- > select * from accounts;
- > select * from customer;
- > select * from depositor;
- > select * from loan;

DEPOSITOR

CNAME	ACNO	LOAN	BLNAME	AMOUNT
Priya	005	1101	RS Layout	15,00,000
Prajna	001	1102	Majestic	16,00,000
Ramya	003	1103	R R Nagar	17,00,000
Midhi	002	1104	Vijaynagar	19,00,000
Priya.				
Prajna.	004			

BORROWER

CNAME	LNO
Ram	1102
Rajesh	1103
Jay	1104
Deepa	1101

OUTPUTa) CNAME

Prajna.

d) CNAME

Ramya

Midhi

e) 2 rows deleted for bcity = "Bangalore"

27

LOAN

LNO	BLNAME	AMOUNT
1101	RS Layout	15,00,000
1102	Majestic	16,00,000
1103	R R Nagar	17,00,000
1104	Vijaynagar	19,00,000

28

DD MM YY YY

> select * from borrower;

c) > select cname from depositor where accno in (select accno from depositor where in
 (select accno from account where blname in
 (select blname from account where blname =
 "Kengeri" group by blname count(*) > 1))
 group by cname having count(*) > 1;

d) select cname from branch b, account a,
 depositor d where b.blname = a.blname and
 a.accno = d.accno and b.bcity = "Bangalore"
 group by cname having count (distinct
 b.blname) = (select count(blname) from
 branch where bcity = "Bangalore") ;

e) delete from account where blname in (select
 blname from branch where bcity = "B city");

SCHEMA DIAGRAM

CUSTOMER

cust-no	cname	city
---------	-------	------

ORDER

order-no	odate	cust-no	ord-amt
----------	-------	---------	---------

ITEM

item-no	price
---------	-------

ORDER-ITEM

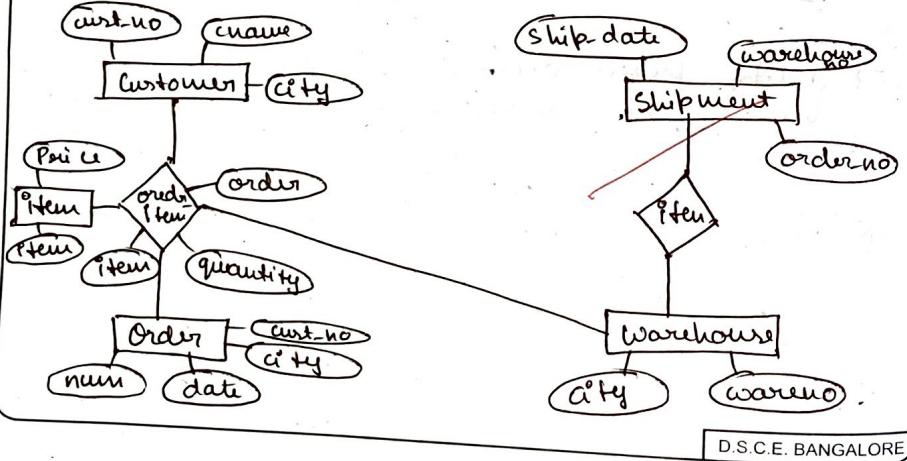
order-no	item-no	city
----------	---------	------

WAREHOUSE

ware-no	city
---------	------

SHIPMENT

orderno	ware-no	ship-date
---------	---------	-----------

ER DIAGRAM

DD	MM	YY	YY
13	11	2019	

- 5) Consider the following relations for an order processing database application in a company.

`CUSTOMER(id int, cname: String, city: String)`

`ORDER(order-no: int, odate: date, cust-no: int, ord-amt: int)`

`ITEM(item-no: int, uprice: int)`

`OITEM(order-no: int, item-no: int, quantity: int)`

`WAREHOUSE(ware-no: int, city: String)`

`SHIPMENT(order-no: int, ware-no: int, ship-date: date)`

- Create the table for the schemas provided with primary keys and foreign keys.
- Insert 5 tuples of values to each table
- Produce a listing: `curname, # of orders, avg. order-amount` where the middle column is the total number of orders by customers and last column is the average `order-amt` for that.
- List the ones for orders that were shifted from all warehouses that the company has in a specific city.
- Demonstrate how you delete `item 10` from the `item` table and make that filled null in the `oitem` table.

CUSTOMER

cid	cname	city
101	Priya	Bangalore
102	Pragna	Tumkur
103	Ranya	Bangalore
104	Nidhi	Hyderabad
105	Jaya	Delhi

ORDERS

ono	date	cid	count
1101	05-Aug-19	102	500
1102	4-Aug-19	104	700
1103	1-Sep-19	105	800
1104	6-Sep-19	101	900
1105	10-Nov-19	103	300

ITEM

iuno	uprice
10	50
20	70
30	80
40	35
50	80

DD	MM	YY	YY	YY
----	----	----	----	----

- 6) > create table customer (cid number(5), cname varchar(10), city varchar(10), primary key (cid))
- > create table order (ono number(5), odate date, cid number(5), count number(5), primary key (ono) , foreign key (cid) references customer (cid));
- > create table item (iuno number(5), uprice number(5), primary key (iuno));
- > create table oitem (ono number(5), iuno number(5), qty number(5), primary key (ono,iuno) , foreign key (ono) reference order (ono), foreign key (iuno) references item (iuno) on delete cascade);
- > create table warehouse (wareno number(5), city varchar(20), primary key (wareno));
- > create table shipment (ono number(5), wareno number(5), shiptdate date, primary key (ono,wareno) , foreign key (ono) references order (ono), foreign key (wareno) references warehouse (wareno));

ITEM

ono	bno	quantity
1103	10	5
1104	20	6
11025	30	7
11052	40	8
1101	500	9

WAREHOUSE

wareno	city
201	Hyderabad
202	Bangalore
203	Delhi
204	Tumkur
205	Delhi

SHIPMENT

ono	wareno	slipdate
1101	201	10-Aug-19
1102	202	14-Aug-19
1103	203	10-Sep-19
1104	204	21-Sep-19
1105	205	14-Mar-19

DD MM YY YY YY

- > insert into ~~customer~~ customer values ('&cid', '&name', '&city');
- > insert into orders values ('&ono', '&odate', '&cid', '&quant');
- > insert into item values ('&ino', '&uprice')
- > insert into item values ('&ono', '&ino', '&quantity');
- > insert into warehouse values ('&wareno', '&city');
- > insert into shipment values ('&ono', '&wareno', '&slipdate');
- > select * from customer;
- > ~~select * from orders;~~
- > select * from item;
- > select * from item;
- > select * from warehouse;
- > select * from shipment;

c)	cid	cname	orderno	avg(oi.quantity * i.price)
	103	Ramya	1	560
	105	Jaya	1	250
	104	Midhi	1	70
	102	Prajna	1	747
	101	Reiya	1	420

d) ono

1101

e) 1 row deleted

0 rows updated

DD	MM	YY	YY
----	----	----	----

- c) > select c.cid, c.cname, count(*) as orderno, avg(oi.quantity * i.price) from customer c, order at, item i, oitem oi, where c.cid = ot.cid and ot.ono = oi.ono and oi.iino = i.iino group by (c.cid, c.cname);
- b) > select s.ono from shipment s, warehouse w where s.wareno = w.wareno and w.city = "Hyderabad";
- c) > delete from item where iino = 10;
> update oitem set iino = null where iino = 10;

X
20/11/19