BRANCH(branch-name:string, branch-city:string, assets:real)

ACCOUNT(accno:int, branch-name:string, balance:real)

DEPOSITOR(customer-name:string, accno:int)

CUSTOMER(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)

1. Find the loan number of those loans with loan amount between 50000 and 100000.

select loanno from loan where amount between 100000 and 500000;

2.For all customers, who have a loan from the bank find their names, loan numbers & loan amount.

select c.customername,l.loanno,l.amount from borrower c,loan l

where l.loanno=c.loanno;

3.Find the customer names, loan numbers loan amount for all loans at Brooklyn Branch.

select b.customername,l.loanno,l.amount from borrower b,loan l

where l.branchname='mgr' and l.loanno=b.loanno;

4.Find the names of all branches that have assets greater than at least one branch located in Frankfurt.

select branchname from branch where assets > any (select assets

from branch where branchcity='hyderabad');

5.List in alphabetical order all customers who have a loan at Perryridge branch.

select b.customername from borrower b,loan l where l.branchname='mgr' and l.loanno=b.loanno order by b.customername;

6.Find the names of all customers who street address includes the substring “Main”.

select customername from customer1 where customerstreet like '%road%';

7.List the loan numbers in descending order of amount

select loanno,amount from loan order by amount desc;

8.Find all customers having a loan, an account, or both at the bank(union, & union all).

(select d.customername from depositor d) union (select customername from borrower);

9.To find all customers who have both a loan and an account at the bank.

select unique borrower.customername from borrower,depositor where borrower.customername=depositor.customername;

10.To find all customers who have an account but no loan at the bank.

select customername from depositor where customername not in(select customername

from borrower);

11.Find the average account balance at the SOMAJIGUDA branch.

SELECT AVG(BALANCE) FROM ACCOUNT WHERE BRANCHNAME='SOMAJIGUDA';

12.Find the average account balance at each branch

SELECT AVG(BALANCE),BRANCHNAME FROM ACCOUNT GROUP BY BRANCHNAME;

13.Find the number of depositor for each branch.

SELECT A.BRANCHNAME,COUNT(\*) FROM ACCOUNT A,DEPOSITOR D

WHERE A.ACCNO=D.ACCNO GROUP BY A.BRANCHNAME;

14.Find the branch where the average account balance is more than Rs.13000 (group by, having)

SELECT BRANCHNAME FROM ACCOUNT HAVING AVG(BALANCE)>13000

GROUP BY BRANCHNAME;

15.Find the average balance for each customer who lives in Horriron & has at least 3 account.

SELECT AVG(A.BALANCE) FROM CUSTOMER1 C,ACCOUNT A,DEPOSITOR D

WHERE C.CUSTOMERCITY='BANGALORE' AND A.ACCNO=D.ACCNO

HAVING COUNT(\*)>=3 GROUP BY D.CUSTOMERNAME;

**16. Find bank accounts with a balance under $700:**

SELECT account\_number, balance FROM account WHERE balance < 700;

**17. Find bank accounts with a balance under $700 with increasing order of bank balance:**

SELECT account\_number, balance FROM account WHERE balance < 700 ORDER BY balance;

“Retrieve a list of all bank branch details, ordered by branch city, with each city’s branches listed in reverse order of holdings.” SELECT \* FROM branch ORDER BY branch\_city ASC, assets DESC;

**18. “Find the average loan amount for each branch.**

” SELECT branch\_name, AVG(amount) AS avg\_amt FROM loan GROUP BY branch\_name;

**19. “Find all customers with more than one loan**.

” SELECT customer\_name, COUNT(\*) AS num\_loans FROM borrower GROUP BY customer\_name HAVING COUNT(\*) > 1;

20.“**Select all customer names from depositor relation, that also appear somewhere in borrower relation.”**

SELECT DISTINCT customer\_name FROM depositor WHERE customer\_name IN ( SELECT customer\_name FROM borrower)

**21.find the largest loan at each branch**

SELECT branch\_name, MAX(amount) FROM loan GROUP BY branch\_name

**22. “Find customers with an account but not a loan.**”

SELECT DISTINCT customer\_name FROM depositor d WHERE NOT EXISTS ( SELECT \* FROM borrower b WHERE b.customer\_name = d.customer\_name);

**23. “Find customers with an account but not a loan**.”

SELECT DISTINCT customer\_name FROM depositor d WHERE NOT EXISTS ( SELECT \* FROM borrower b WHERE b.customer\_name = d.customer\_name);

**24. “Find all branches with assets greater than at least one branch in Brooklyn.”**

SELECT branch\_name FROM branch WHERE assets > SOME ( SELECT assets FROM branch WHERE branch\_name='Brooklyn');

**25. “Find branches with assets greater than all branches in Brooklyn.”**

SELECT branch\_name FROM branch WHERE assets > ALL ( SELECT assets FROM branch WHERE branch\_name='Brooklyn');

Or

SELECT branch\_name FROM branch WHERE assets > (SELECT MAX(assets) FROM branch WHERE branch\_name='Brooklyn');

**26. “Find all cities with more than two customers living in the city.”**

SELECT customer\_city, COUNT(\*) AS num\_customers FROM customer GROUP BY customer\_city HAVING COUNT(\*) > 2;

Or, can write:

SELECT customer\_city, num\_customers FROM (SELECT customer\_city, COUNT(\*) FROM customer GROUP BY customer\_city) AS counts (customer\_city, num\_customers) WHERE num\_customers > 2;

**27. Add 2% interest to all bank account balances with a balance of $500 or less.**

UPDATE account SET balance = balance \* 1.02 WHERE balance <= 500;

28. Delete all account tuples at every branch located at BANGALORE

DELETE FROM ACCOUNT WHERE BRANCHNAME IN( SELECT BRANCHNAME

FROM BRANCH WHERE BRANCHCITY='BANGALORE');

29. Find customers who hare borrowers are account holder.

SELECT CUSTOMERNAME FROM BORROWER WHERE CUSTOMERNAME IN (SELECT CUSTOMERNAME FROM DEPOSITOR);

30. Find all customers who have an both an account & loan at the bank

(SELECT CUSTOMERNAME FROM DEPOSITOR) INTERSECT

(SELECT CUSTOMERNAME FROM BORROWER);

**BOOK DEALER DATABASE**

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)

1. Get the titles of all the books that are not in the FICTION category

select c.title from catalog c,category ca

where ca.description!='fiction' and ca.categoryid=c.categoryid;

-----------------------------------------------------------------------------------------------------------------

1. Get the names of all books whose price is greater than the maximum of the category averages.

select title from catalog where price > (select max(avg(price)) from catalog

group by categoryid);

-----------------------------------------------------------------------------------------------------------------

1. Get the names of all books that are in the book table and for which an order is placed.

select c.title from catalog c,orderdetails o where c.bookid=o.bookid;

-----------------------------------------------------------------------------------------------------------------

1. Get the names of all books that are not in the order table.

select c.title from catalog c,orderdetails o where c.bookid not exists

(select bookid from orderdetails);

-----------------------------------------------------------------------------------------------------------------

1. Get the title, author name and publisher name where publisher name has an underscore.

select p.name,a.name,c.title from catalog c,author a,publisher p

where instr(p.name,'\_')>0 and c.authorid=a.authorid and p.publisherid=c.publisherid;

-----------------------------------------------------------------------------------------------------------------

1. Get all the book names where price is null.

select c.title from catalog c where c.price is null;

-----------------------------------------------------------------------------------------------------------------

1. Get the publisher, the average,maximum and minimum book prices of all the publisher other that “McGraw-Hill”

select p.name,max(c.price),avg(c.price),min(c.price) from catalog c,publisher p

where p.publisherid=c.publisherid and p.name!=' McGraw-Hill ' group by p.name;

-----------------------------------------------------------------------------------------------------------------

1. Get publisher, the average, maximum and minimum book prices of all publisher who have more that 2 books listed in catalog table.

select p.name,avg(c.price),max(c.price),min(c.price) from catalog c,publisher p

where p.publisherid=c.publisherid group by c.publisherid,p.name having count(\*)>2;

-----------------------------------------------------------------------------------------------------------------

1. Get all the books details whose price is greater than 300 in descending order of price.

select \* from catalog where price > 300 order by price desc.

-----------------------------------------------------------------------------------------------------------------

1. Get the titles of all the books in the CATALOG table whose price is greater than the average price.

select title from catalog where price > (select avg(price) from catalog);

-----------------------------------------------------------------------------------------------------------------

1. Get the names of all books for which an order has been placed.

select c.title from catalog c,orderdetails o where o.bookid=c.bookid;

-----------------------------------------------------------------------------------------------------------------

1. get the title and price of all the books whose price is less than the average price of the books.

select title,price from catalog where price < (select avg(price) from catalog);

-----------------------------------------------------------------------------------------------------------------

1. Get the names of all authors who have more than two books in the catalog.

select a.name,count(\*) from author a,catalog c where a.authorid=c.authorid

group by a.name having count(\*)>2;

-----------------------------------------------------------------------------------------------------------------

1. Get the title and price of all the books whose price is greater than the maximum of the category averages.

select c.title,c.price from catalog c where c.price > (select max(avg(price))

from catalog group by categoryid);

-----------------------------------------------------------------------------------------------------------------

1. Get the details of the authors whose books are being sold from the book house

select a.name from author a,catalog c,orderdetails o

where o.bookid=c.bookid and c.authorid=a.authorid;

-----------------------------------------------------------------------------------------------------------------

1. Get the title and price of all the books whose price is greater than the average price of the books in the “Business”category.

select c.title,c.price from catalog c where c.price > (select avg(c.price)

from catalog c,category co where co.description='business' and co.categoryid=c.categoryid);

-----------------------------------------------------------------------------------------------------------------

1. Get the details of the titles whose price is greater than the average price and whose year of publishing is greater than the average year of publishing.

select c.title from catalog c where c.price > (select avg(price) from catalog)

and c.year > (select avg(year) from catalog);

-----------------------------------------------------------------------------------------------------------------

1. Get the title,year and price of all the books in the ascending order of year of publishing for which an order is placed.

select c.title,c.year,c.price from catalog c,orderdetails o where o.bookid=c.bookid

order by c.year asc;

-----------------------------------------------------------------------------------------------------------------

1. Get the book details and category details of all books whose price is greater than 1000.

select c.title,c.categoryid from catalog c where c.price > 1000;

-----------------------------------------------------------------------------------------------------------------

1. Find out the titles that have the same price.

select distinct t1.title from catalog t1,catalog t2 where t1.price = t2.price and t1.title != t2.title;

-----------------------------------------------------------------------------------------------------------------

1. Get the details of all authors and publishers in India ordered by name.

(select name from author where country='india') union

(select name from publisher where country='india') order by name;

-----------------------------------------------------------------------------------------------------------------

1. Find the author of the book which has maximum sales.

select name, authored from author where authored in( select authored from catlog where bookid in

(select bookid from odetails where quantity in (select max(quantity) from odetails));

-----------------------------------------------------------------------------------------------------------------

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

Select \* from catlog; **(before updation)**

update catlog set price= price +(price \*0.10) where publisherid =’& pubid’;

**[OR]**

update catalog set price=price\*1.10 where publisherid=44;

1. 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, date: date, location: string)

OWNS (driver-id #:string, Regno:string)

PARTICIPATED (driver-id: string, Regno:string, report-number:int, damage amount:int)

**Execute the following queries**

1. Create PERSON table with the following attributes: Driver-id#: string Name : string
2. Add New column address : string to PERSON table
3. Create CAR, ACCIDENT, OWNS, PARTICIPATED tables
4. Add a Foreign key constraint to link owns table and person table (Similarly add Foreign key constraint for other tables)
5. Enter atleast five tuples for each relation.
6. List all tables in the database.
7. List PERSON table structure.
8. List all the records of PERSON table.
9. SELECT the ACCIDENT record where report-number equal to 1.
10. List all ACCIDENT records where report-number is greater than 2.
11. Produce a list of drivers whose driver-id is between 2 & 10.
12. Display the driver details whose names are ‘john’ and ‘antony’ (using or & in)
13. Find the total number of people who owned car ‘Maruthi’
14. Update the manufacturing year of a specific CAR model to 2004.
15. DELETE the CAR whose Regno is 5.
16. DELETE the accident record where report-number is 4.
17. COUNT the no. of cars participated in accident.
18. Find the total number of people who owned cars that were involved in accidents in 2003.
19. Display the driver name with their car models.
20. Display the accident location where accident Regno is 5.
21. Display unique car Regnos involved in accidents.
22. Display the car Regno and model which has the maximum damage amount.
23. Display no. of cars owned by each driver.
24. Display driver-id & total no. of accidents in which each driver is involved.
25. Find the number of accidents in which cars belonging to a specific model were involved.
26. ***ORDER PROCESSING DATABASE***

CUSTOMER (cust #: int , cname: string, city: string)

ORDER (order #: int, odate: date, cust #: int, ord-Amt: int)

ORDER – ITEM (order #: int, Item #: int, qty: int)

ITEM (item # : int, unit price: int)

SHIPMENT (order #: int, warehouse#: int, ship-date: date)

WAREHOUSE (warehouse #: int, city: string)

1. List the no. of order placed by customer no. 5

SELECT COUNT(CO.ORDERNO) FROM CORDER CO WHERE CO.CUSTNO = 5;

------------------------------------------------------------------------------------------------------------------

2. List item nos and and its quantity of order no. 5

SELECT ITEMNO,QTY FROM ORDERITEM WHERE ORDERNO = 5;

------------------------------------------------------------------------------------------------------------------

3. List the average order amount where "for the current year"

SELECT AVG(ORDAMT) FROM CORDER WHERE ODATE LIKE '%13';

------------------------------------------------------------------------------------------------------------------

4. List the no. of items shipped for warehouse "bangalore"

SELECT COUNT(O.ITEMNO) FROM ORDERITEM O,SHIPMENT S,WAREHOUSE W

WHERE O.ORDERNO = S.ORDERNO AND S.WAREHOUSENO = W.WAREHOUSENO AND W.CITY = 'BANGALORE';

------------------------------------------------------------------------------------------------------------------

5. List the no. of orders placed by each customer.

SELECT COUNT(\*) FROM CORDER GROUP BY CUSTNO;

------------------------------------------------------------------------------------------------------------------

6. List the customer names who have not ordered for item no. 10.

SELECT C.ENAME FROM CUSTOMER C,CORDER CO

WHERE C.CUSTNO = CO.CUSTNO AND CO.ORDERNO != 10;

------------------------------------------------------------------------------------------------------------------

7. List customer details who has the largest order amount.

SELECT C.ENAME,C.CUSTNO,C.CITY FROM CUSTOMER C,CORDER CO

WHERE C.CUSTNO = CO.CUSTNO AND CO.ORDAMT =

(SELECT MAX(CO.ORDAMT) FROM CORDER CO);

------------------------------------------------------------------------------------------------------------------

8. List customer details whose order has been shipped from warehouse "bangalore".

select c.ename,c.custno,c.city from customer c,shipment s,warehouse w,corder co

where c.custno = co.custno and co.orderno = s.orderno and s.warehouseno = w.warehouseno and w.city = 'bangalore';

------------------------------------------------------------------------------------------------------------------

9.List the customer names who have placed more than 2 orders.

SELECT C.ENAME FROM CUSTOMER C WHERE (SELECT COUNT(CO.ORDERNO)

FROM CORDER CO WHERE C.CUSTNO = CO.CUSTNO) > 2 ;

------------------------------------------------------------------------------------------------------------------

10. List the average order amount of all customers.

SELECT AVG(ORDAMT) AS AVERAGE FROM CORDER;

------------------------------------------------------------------------------------------------------------------

11. List no. of warehouses in each city.

SELECT COUNT(W.WAREHOUSENO) FROM WAREHOUSE W GROUP BY W.CITY;

------------------------------------------------------------------------------------------------------------------

12. List name of customer, no. of orders placed by each customer residing in bangalore city.

SELECT C.ENAME,COUNT(CO.ORDERNO) FROM CUSTOMER C,CORDER CO

WHERE C.CUSTNO = CO.CUSTNO AND C.CITY = 'BANGALORE' GROUP BY C.ENAME;

------------------------------------------------------------------------------------------------------------------

13. List the customers who have purchased items of maximum amt shipped from "abc" warehouse.

SELECT C.ENAME FROM CUSTOMER C,CORDER CO

WHERE CO.ORDAMT = (SELECT MAX(CO.ORDAMT)

FROM CORDER CO,CUSTOMER C,SHIPMENT S,WAREHOUSE W

WHERE C.CUSTNO = CO.CUSTNO AND CO.ORDERNO = S.ORDERNO

AND W.WAREHOUSENO= S.WAREHOUSENO AND W.CITY = 'abc');

------------------------------------------------------------------------------------------------------------------

14. List the No. of items shipped on the same day.

SELECT COUNT(O.ITEMNO) FROM SHIPMENT S,ORDERITEM O

WHERE S.ORDERNO = O.ORDERNO GROUP BY S.SHIPDATE;

------------------------------------------------------------------------------------------------------------------

15. List the names of customers who have ordered at least 10 items

SELECT C.ENAME,COUNT(\*) FROM CUSTOMER C,CORDER CO

WHERE C.CUSTNO = CO.CUSTNO GROUP BY C.ENAME HAVING COUNT(\*) >= 10;

------------------------------------------------------------------------------------------------------------------

16. List the items which are ordered by at least 3 customers.

SELECT COUNT(O.ITEMNO) FROM ORDERITEM O,CUSTOMER C,CORDER CO

WHERE C.CUSTNO = CO.CUSTNO AND CO.ORDERNO = O.ORDERNO

GROUP BY C.CUSTNO HAVING COUNT(O.ITEMNO) >= 3;

------------------------------------------------------------------------------------------------------------------

17. Find the total order amount for each day.

SELECT ODATE,SUM(ORDAMT) FROM CORDER GROUP BY ODATE;

------------------------------------------------------------------------------------------------------------------

18. Find the item which is ordered maximum no. of times.

SELECT UNIQUE I.ITEMNO FROM ORDERITEM I WHERE I.ITEMNO = (SELECT MAX(COUNT(\*)) FROM ORDERITEM I GROUP BY I.ITEMNO);

19. List the names of all customers who have ordered item no. 10.

SELECT C.ENAME FROM CUSTOMER C,ORDERITEM O,CORDER CO WHERE

O.ITEMNO = 10 AND CO.ORDERNO = O.ORDERNO AND CO.CUSTNO = C.CUSTNO;

20. List all the order numbers which have more than 4 items.

SELECT CO.ORDERNO,COUNT(\*) FROM ORDERITEM O, ITEM I, CORDER CO

WHERE O.ORDERNO = CO.ORDERNO AND I.ITEMNO = O.ITEMNO

GROUP BY CO.ORDERNO HAVING COUNT(\*) >= 4;

21. Delete the items with no orders