BCA304P: DATABASE MANAGEMENT SYSTEM LAB

PART - A

1. The STUDENT detail databases has a table with the following attributes. The primary keys are underlined. STUDENT(regno: int, name: string, dob: date, marks: int)

i) Create the above table.

ii) Remove the existing attributes from the table.

iii) Change the date type of regno from integer to string.

iv) Add a new attribute phoneno to the existing table.

v) Enter five tuples into the table.

vi) Display all the tuples in student table.

Answer

1. Create table studentdb (regno number(10) primary key , name varchar(20), dob date, marks number(10))
2. Alter table studentdb drop column marks
3. Alter table studentdb modify regno varchar2(20)
4. Alter table studentdb add phoneno number(10)
5. Insert into studentdb values(&regno,&name,&dob,&marks,&phoneno)
6. Select * from studentdb

****

2. A LIBRARY database has a table with the following attributes.

LIBRARY(bookid:int, title:string, author:string, publication:string, yearpub:int, price:real)

i) Create the above table.

ii) Enter the five tuples into the table

iii) Display all the tuples in library table.

iv) Display the different publishers from the list.

v) Arrange the tuples in the alphabetical order of the book titles.

vi) List the details of all the books whose price ranges between Rs. 100 and Rs. 300

Answer

1. Create table librarydb (bookid number(10) primary key, title varchar2(20),author varchar2(20), publication varchar2(20), year number(5),price number(6,2))
2. Insert into librarydb values(&bookid,&title,&author,&publication,&year,&price)
3. Select * from librarydb;
4. Select distinct publication from librarydb
5. Select * from librarydb order by title asc
6. Select * from librarydb where price between 100 and 300

****

3. The SALARY database of an organization has a table with the following attributes.

EMPSALARY(empcod:int, empnamee:string, dob:date, department:string, salary:real)

i) Create the above table.

ii) Enter the five tuples into the table

iii) Display all the number of employees working in each dapartment.

iv) Find the sum of the salaries of all employees.

v) Find the sum and average of the salaries of employees of a particular department.

vi) Find the least and highest salaries that an employee draws.

Answers

1. Create table salarydb(empcode number(10) primary key, empname varchar2(20), dob date, dept varchar2(15),salary number(10,2))
2. Insert into salarydb values(&empcode,&empname,&dob,&dept,&salary)
3. Select dept,count(*) from salarydb group by dept
4. Select sum(salary) from salarydb
5. Select sum(salary) , avg(salary) from salarydb where dept=”computer”
6. Select min(salary) from salarydb
7. Select max(salary) from salarydb

****

4. Consider the insurance database given below. The primary keys are underlined and

the data types are specified.

PERSON(driver-id-no: string, name: string, address:strong)

CAR(regno: string, model: string, year: int)

ACCIDENT(report-no: int, date: date, location: String)

OWNS(driver-id-no: string, regno: string)

PARTICIPATED(driver-id-no: string, regno: string, report-no: int, damage-amount: int)

i) Create the above tables by properly specifying the primary keys and the foreign keys

ii) Enter atleast 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 no 12 to 25000.

b) Add a new accident to the database.

iv) Find total number of people who owned cars that were involved in accidents in 2002

v) Find the number of accidents in which cars belonging to a specific model were involved

Answers

1. Create table persondb(driver_id varchar2(20) primary key,name varchar2(20) not null, address varchar2(30))
2. Create table cardb(regno varchar2(20) primary key, model varchar2(20) not null,year number(5))
3. Create table accidentdb(report_no number(10) primary key,accident_date date,location varchar2(20))
4. Create table ownsdb (driver_id varchar2(20) references persondb,,regno varchar2(20) references cardb)
5. Create table participateddb(driver_id varchar2(20) references persondb,regno varchar2(20) references cardb,report_no number(10) references accidentdb,damage_amount number(10))
6. Insert into persondb values(&driver_id,&name,&address)
7. Insert into cardb values(&regno,&model,&year)
8. Insert into accidentdb values(&report_no,&accident_date,&location)
9. Insert into ownsdb values(&driver_id,&reg_no)

10. Insert into participateddb values(&driver_id,&reg_no,&report_no,&damage_amount)

11. Update participateddb set damage_amount = 25000 where reg_no = 1001 and report_no = 12
12. Insert into participateddb values(1004,2004,10,3000)
13. Select count(*) from accidentdb where accident_date LIKE '%-%-02'
14. Select count(*) from cardb c ,participateddb p where c.regno=p.regno and c.model = 'scoda'

****

5. Consider the following database of students enrollment in courses and books adopted for each course.

STUDENT(regno: string, name: string, major: strong, bdate: date)

COURSE(course-no: int cname: string, dept: string)

ENROLL(reg-no: string, course-no: int, sem: int, marks: int)

BOOK-ADOPTION(course-no: 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 atleast 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-no, book-isbn, book-title) in the alphabetical order for courses offered by the 'Compute Science' department that use more than two books.

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

Answers

1. Create table student (regno varchar2(10) primary key ,name varchar2(20),major varchar2(20),dob date)
2. Create table course (courseno number(10) primary key,cname varchar2(20),dept varchar2(20))
3. Create table enroll(regno varchar2(10) references student , courseno number(10) references course, sem number(5),marks number(5))
4. Create table text(bkisbn number(5) primary key,book_title varchar2(20),publisher varchar2(20),author varchar2(20))
5. Create table book_adpt (course_no number (10) references course, bkisbn number(5) references text,sem number(5))
6. Insert into student values(&regno,&name,&major,&dob)
7. Insert into course values(&courseno,&cname,&dept)
8. Insert into enroll values(&regno,&courseno,&sem,&marks)
9. Insert into text values(&bkisbn,&book_title,&publisher,&author)
10. Insert into book_adpt values(&course_no,&bkisbn,&sem)
11. Insert into text values(2009,'vb','skyword','chitra ravi')
12. Insert into book_adpt values(104,2009,4)
13. create view compdept as (select c.dept , c.course_no, t.book_title, t.bkisbn from course c,book_adpt ba, text t where c.course_no = ba.course_no and ba.bkisbn=t.bkisbn and dept='cs') order by t.book_title
14. select * from compdept
15. select course_no ,bkisbn,book_title from compdept where dept in (select dept from compdept group by dept having count(*) > =2)
16. select c.dept , t.book_title,t.publisher from course c,text t,book_adpt b where t.publisher = 'himalaya' and c.course_no = b.course_no and b.bkisbn=t.bkisbn

****

6. 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)

CATLOG(book-id: int, title : string, author-id: int, publisher-id: int, category: int, year:

int, price: int)

CATEGORY(category-id: int, description: string)

ORDER-DETAILS(order-no: int, book-id: int, quantity: int)

i) Create above tables by properly specifying the primary keys and the foreign keys.

ii) Enter atleast 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 is greater than the average price of the books in the catalog and the year

of publication is after 2010.

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

v) Demonstrate how to increase price of books published by specific publisher by 10%

Answers

1. create table authordb (author_id number(5) primary key , name varchar2(15) not null,city varchar2(10),country varchar2(10))
2. create table publisherdb(publisher_id number(5) primary key,name varchar2(15) not null,city varchar2(15),country varchar2(15))
3. create table catalogdb(book_id number(5) primary key, title varchar2(15) not null,author_id number(5) references authordb,publisher_id number(5) references publisherdb,year number(5),price number(8))
4. create table categorydb(category_id number(5) primary key,description varchar2(15))
5. create table orderdb(order_id number(5) primary key,book_id number(5) references catalogdb,quantity number(5))
6. insert into authordb values(&author_id,&name,&city,&country)
7. insert into publisherdb values(&publisher_id,&name,&city,&country)
8. insert into catalogdb values(&book_id,&title,&author_id,&publisher_id,&year,&price)
9. insert into categorydb values(&category_id,&description)
10. insert into orderdb values(&order_id,&book_id,&quantity)
11. select c.author_id,a.name from catalogdb c,authordb a where a.author_id = c.author_id and c.price > (select avg(price) from catalogdb group by c.author_id , a.name having count(*) > 2)
12. update catalogdb set price = price*1.10 where publisher _id = 10;

****

7. Consider the following database for BANK.

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-no: int, branch-name: string, amount: real)

BORROWER(customer-name: string, loan-no: int)

i) Create the above tables by properly specifying the primary keys and foreign keys.

ii) Enter atleast five tuples for each relation.

iii) Find all the customers who have atleast two accounts at the main branch.

iv) Find all customer who have an account at all the branches located in a specific city.

v) Demonstrate how t0 delete all account tuples at every branch located in specific city.

Answers

1. Create table branchdb (bname varchar2(15) primary key,bcity varchar2(15) not null,asset number(8,4))

2. Create table accountdb (accno number(10) primary key , bname varchar2(15) references branchdb,balance number(10,2))

3. create table depositordb(cname varchar2(15) not null,accno number(5) references accountdb primary key(cname,accno))

4.create table customerdb (cname varchar2(15) primary key, cstreet varchar2(15),ccity varchar(15))

5.create table loandb (loanno number(10) primary key,bname varchar2(15) references branchdb ,amount number(10,2))

6.create table borrowerdb (cname varchar2(20) not null,loanno number(10) references loandb,primary key(cname,loanno)

7.insert into branchdb values(&bname,&bcity,&asset)

8.insert into accountdb values(&accno,&bname,&balance)

9.insert into depositordb values(&cname,&accno)

10.insert into customerdb values(&cname,&cstreet,&ccity)

11.insert into loandb values(&loanno,&bname,&amount)

12. insert into borrowerdb values(&cname,&loanno)

13. create view decacc as (selelct b.bname,a.accno,d.cname from branchdb b,accountdb a,depositordb d where a.accno = d.accno and a.bname = b.bname and b.bname = ‘jayanagar’)

14. select bname,accno,cname from decacc where cname in(select cname from decacc group by cname having count(*)>-2)

15. select d.cname,a.accno,b.bname,b.bcity from depositordb d,accountdb a,branchdb b where b.bcity=’bangalore’ and d.accno = a.ccno and a.bname=b.bname

16.delete from accountdb where bname in (select bname from branchdb where bcity=’bangalore’)

****

8. Consider the following database for ORDER PROCESSING.

CUSTOMER(cust-no: int, cname: string, city: string)

ORDER(orderno: int, odate: date, ord-amt: real)

ORDER_ITEM(orderno: int, itemno:int, qty: int)

ITEM(itemno: int, unitprice: real)

SHIPMENT(orderno: int, warehouseno: int, ship-date: date)

WAREHOUSE(warehouseno: int, city: string)

i) Create the above tables by properly specifying the primary keys and the foreign keys

ii) Enter atleast five tuples for each relation.

iii) List the order number and ship date for all orders shipped from particular warehouse

iv) Produce a listing: customer name, no of orders, average order amount

v) List the orders that were not shipped within 30 days of ordering

Answers

1. create table customerdb (cust_id number(6) primary key,cname varchar2(20) not null,ccity varchar2(20))
2. create table custorderdb(orderno number(6) primary key,orderdate date,customer_no number(6),order_amt number(6))

3. create table itemdb(item_no number(6) primary key , unitprice number(6))
4. create table warehousedb(warehouseno number(5) primary key,city varchar2(12))
5. create table shipmentdb(orderno number(6) references custorderdb,warehouse_no number(5) references warehouse_db,shipdate date);
6. insert into customerdb values(&cust_id,&cname,&ccity)
7. insert into custorderdb values(&orderno,&orderdate,&customer_no,&order_amt)
8. insert into itemdb values(&item_no,&unitprice)
9. insert into warehousedb values(&warehouseno,&city)
10. insert into shipmentdb values(&orderno,&warehouse_no,&shipdate)
11. select orderno,shipdate from shipmentdb where warehouse_no=1002
12. select c.cname , count(co.orderno),avg(co.order_amt) from customerdb c,customer_db co where c.cust_id = co.customer_no group by c.cname,co.customer_no
13. select c.cname ,co.orderno,co.orderdate,sh.shipdate from customerdb c,custorder co,shipmentdb sh where c.cust_id =co.customer_no and co.orderno = sh.orderno and (to_date (sh.shipdate)-to_date(co.orderdate))>30

****

PART – B

1. Write a PL/SQL program to find the largest of three numbers

```
declare
   a number;
   b number;
   c number;
begin
   a:=&a;
   b:=&b;
   c:=&c;
if (a>b and a>c) then
     dbms_output.put_line('a is largest' || a);
elsif (b>a and b>c) then
     dbms_output.put_line('b is largest' || b);
else
    dbms_output.put_line('c is the largest'||c);
endif;
end;
```

2. Write a PL/SQL program to generate reverse for given number

```
declare
n number(4) := &n;
s number(4) := 0;
r number(4);
begin
while n > 0
loop
r:= mod(n,10);
s:=(s*10)+r;
n:=trunc(n/10);
end loop;
dbms_output.put_line('the reverse number is');
dbms_output.put_line(s);
end;
```

3. Write a PL/SQL program to find the factorial of a given number

```
declare
i number(4) :=1;
n number(4) := &n;
f number(4) :=1;
begin
for i in 1..n
loop
f:=f*i;
end loop;
dbms_output.put_line('factorial of a number is'|| f);
end;
```

4. Write a PL/SQL program to check whether given number is prime or not

```
declare
  num number;
  i number := 1;
  c number :=0;
begin
  num := &num;
```

```
  for i in 1..num
loop
if ((mod(num,i))=0)
then
c:=c+1;
end if;
end loop;
if (c>2)
then
 dbms_output.put_line(num|| 'not prime');
else
 dbms_output.put_line(num || 'is prime');
end if;
end;
```

5. Write a PL/SQL program to generate Fibonacci series upto N

```
declare
  a number(3) := 1;
  b number(3) := 1;
  c number(3);
  n number(3);
begin
  n:=&n;
  dbms_output.put_line('the Fibonacci series is:')
  while a<=n
  loop
    dbms_output.put_line(a);
    c:=a+b;
    a:=b;
    b:=c;
 end loop;
end;
```

6. Write a PL/SQL program for inserting a row into employee table

```
Create table employee (emp_id number(5) primary key,emp_name varchar2(30),
Emp_dept varchar2(10),emp_salary  varchar2(8));
```

```
Declare
begin
Insert into employee values(10,'chitra','hr',40000);
End;
```

7. Write a pl/sql program to handle a predefined exception

```
declare
 n number(4);
 d number(4);
begin
  n:=&n;
  d:=n/0;
  exception
 when zero_divide then
  dbms_output.put_line('divide by error exception is caught');
end;
```

8. Write a pl/sql program for creating a procedure for calculating sum of two numbers.

```
Create or replace procedure "sum"(n1 in number,n2 in number) is
Total number(6);
Begin
Total:= n1+n2;
Dbms_output.put_line('the sum is '||total);
End;

Execution
SQL>exec sum(10,20);
The sum is : 30
```

9. Write a procedure to check the given year is leap year or not

```
Create or replace procedure leapyear(y in number) is
Begin
  If y mod 4 = 0 and y mod 100<>0 or y mod 400 =0 then
   Dbms_output.put_line ('the given year is leap year');
Else
```

```
Dbms_output.put_line('the given year is not leap year');
End if;
End;

Calling a above procedure
Begin
  Leapyear(2012);
End;
```