1.   
Consider the insurance database given below. The primary keys are made bold  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 , accd\_date:date , location:string )  
OWNS( **driver\_id**:string , **regno**:string )  
PARTICIPATED( **driver\_id**:string , **regno**:string , **report\_number**:int , damage\_amount:int)

1)Create the above tables by properly specifying the primary keys and foreign keys.  
2)Enter at least five tuples for each relation.  
3)Demonstrate how you

a.Update the damage amount for the car with specific regno in the accident with report number 12 to 25000.  
b.Add a new accident to the database.

4)Find the total number of people who owned cars that were involved in accidents in the year 2008.  
5)Find the number of accidents in which cars belonging to a specific model were involved.  
  
2.  
Consider the following relations for a order processing database application in a company.

CUSTOMER( **custno**:int , cname:string , city:string )  
ORDER( **orderno**:int , odate:date , custno:int , ord\_amt:int )  
ORDER\_ITEM( **orderno**:int , **itemno**:int , quantity:int )  
ITEM( **itemno**:int , unitprice:int )  
SHIPMENT( **orderno**:int , **warehouseno**:int , ship\_date:date )  
WAREHOUSE( **warehouseno**:int , city:string )

1)Create the above tables by properly specifying the primary keys and foreign keys.  
2)Enter at least five tuples for each relation.  
3)Produce a listing: custname , No\_of\_orders , Avg\_order\_amount ,  where the middle column is the total number of orders by the customer and the last column is the average order amount for that customer.  
4)List the orderno for orders that were shipped from ***all*** the warehouses that the company has in a specific city.  
5)Demonstrate the deletion of an item from the ITEM table and demonstrate a method of handling the rows in the ORDER\_ITEM table that contains this particular item.  
  
3.  
Consider the following database of student enrollment in courses and books adopted for that course.

STUDENT( **regno**:string , name:string , major:string , bdate:date )  
COURSE( **courseno**:int , cname:string , dept:string )  
ENROLL( **regno**:string , **courseno**:int , **sem**:int , marks:int )  
BOOK\_ADOPTION( **courseno**:int , **sem**:int , book\_isbn:int )  
TEXT( **book\_isbn**:int , book\_title:string , publisher:string , author:string )

1)Create the above tables by properly specifying the primary keys and foreign keys.  
2)Enter atleast five tuples for each relation.  
3)Demonstrate how you add a new text book to the database and make this book to be adopted by some department.  
4)Produce a list of text books ( includes courseno , book\_isbn , book\_title ) in the alphabetical order for courses offered by the 'CS' department that use more than two books.  
5)List any department that has ***all*** its books published by a specific publisher.  
  
4.  
The following are maintained by abook 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 )  
1)Create the above tables by properly specifying the primary keys and foreign keys.  
2)Enter at least five tuples for each relation.  
3)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 2000.  
4)Find the author of the book that has maximum sales.  
5)Demonstrate how you increase the price of books published by a specific publisher by 10%.  
  
5.  
Consider the following database for a banking enterprise.

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)Create the above tables by properly specifying the primary keys and foreign keys.  
2)Enter at least five tuples for each relation.  
3)Find ***all*** the customers who have at least two accounts at the ***main*** branch.  
4)Find all the customers who have an account at ***all*** the branches located in a specific city.  
5)Demonstrate how you delete all account tuples at every branch located in a specific city.