1) Insurance Database

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)

Queries

Find the total number of people who owned cars that were involved in accidents in 1989

```
select count(distinct P.driverid)
from accident A, participated P
where A.reportno = P.reportno and A.accdate between
      '1998-01-01 00:00:00' and  '1998-12-31 00:00:00'
```

Find the number of accidents in which the cars belonging to "John Smith" were involved

```
select count(P.reportno) as NO_OF_ACC
from partcipated P, person PN
where P.driverid =  PN.driverid and   PN.fname = 'John Smith'
```

Update the damage amount for the car with reg number "KA-12" in the accident with report number "1" to $3000

```
update partcipated
set dmgamt=3000
where regno = 'KA-12' and reportno = 1
```

2) Order Processing Database

CUSTOMER (custid: int, cname: string, city: string)

ORDER (orderid: int, odate: date, custid: int, ord-Amt: int)

ORDER – ITEM (orderid: int, itemid int, qty: int)

ITEM (itemid: int, unit price: int)

SHIPMENT (orderid: int, warehouseid: int, ship-date: date)

WAREHOUSE (warehouseid: int, city: string)

Queries

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.orderid) as NO_OF_ORDR, avg(O.ordamt)
       as AVG_ORD_AMT
from CUSTOMER C, ORDER O
where C.custid = O.custid
group  by C.cname
```

For each item that has more than two orders , list the item, number of orders that are shipped from atleast two warehouses and total quantity of items shipped

```sql
select itemid, sum(qty) as Tot_Qty,count(*) as No_Of_Orders
from ORDER_ITEM
where orderid in( select orderid
                  from SHIPMENT
                  group by orderid
                  having count(*) >= 2
                )
group by itemid
having count(*) >= 2
```

List the customers who have ordered for every item that the company produces

```sql
select c.cname
from CUSTOMER c
where not exists( select i.itemid
                  from ITEM i
                  where i.itemid not in( select distinct(oi.itemid)
                                         from ORDER_ITEM oi,
                                              C_ORDER o
                                         where o.orderid = oi.orderid
                                               and o.custid = c.custid
                                       )
                )
```

3) Student Database

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

COURSE (course-id: int, cname: string, dept: string)

ENROLL (regno: string, course-id: int, sem: int marks: int)

BOOK _ ADOPTION (course-id: int, sem: int, book-ISBN: int)

TEXT (book-ISBN: int, book-title: string, publisher: string, author: string)

Queries

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 T.bookISBN,T.booktitle,C.courseid,C.cname
from TEXTBOOK T,COURSE C,BOOK_ADAPTION B
where  T.bookISBN = B.bookISBN and C.courseid=B.courseid
       and C.dept='CS' and C.courseid in ( select course
                                           from BOOK_ADAPTION
                                           group by courseid
                                           having count(*)>=2
                                         )
order by T.booktitle
```

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

```
select distinct(C.dept)
from COURSE C
where not exists( select bookISBN
                  from BOOK_ADAPTION
                  where courseid in( select courseid
                                     from  COURSE
                                     where dept = C.dept
                                   )
                        and bookISBN not in( select bookISBN
                                             from TEXTBOOK
                                             where publisher='McGraw'
                                           )
                )
```

List the bookISBNs and book titles of the department that has maximum number of students

```
select distinct(t.bookISBN),t.title,c.dept
from TEXTBOOK t, BOOK_ADAPTION b, COURSE c
where t.bookISBN = b.bookISBN and b.courseid = c.courseid
      and c.dept in (select c.dept
                     from COURSE c,ENROLL e
                     where c.courseid = e.courseid
                     group by c.dept
                     having count(*) >= all (select count(*)
                                             from COURSE c,ENROLL e
                                             where c.courseid =
e.courseid

                                             group by c.dept
                                            )
                    )
```

4) 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)

Queries

Find the author of the book which has maximum sales

```
select A.authorid, A.name, A.city, C.bookid,
       sum(O.quantity) as QTY_SUM
from AUTHOR A, CATALOG C, ORDER_DETAILS O
where A.authorid  =  C.authorid and C.bookid = O.bookid
group by A.authorid, C.bookid
having sum(O.quantity) >= all ( select sum(quantity)
                                from ORDER_DETAILS
                                group by bookid
                              )
```

Increase the price of the books published by a specific publisher by 10%

```
select * from CATALOG

update CATALOG set price = price * 1.1
where publisherid in ( select publisherid
                       from publisher
                       where name ='Pearson'
                     )
```

Find the number of orders for the book that has minimum sales

```
select count(o.orderno),c.title,c.bookid
from ORDERDETAILS o,CATALOG c
where c.bookid=o.bookid
group by o.bookid
having sum(o.quantity) <= all (select sum(quantity)
                        from ORDERDETAILS
                        group by bookid
                        )
```

5) Banking Database

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)

Queries

Find all the customers who have atleast 2 accounts at all the branches located in a specific city

```
select C.customername
from CUSTOMER C
where not exists( select B.branchname
                  from  BRANCH B
                  where B.branchcity = 'karkala' and branchname
                        not in( select A.branchname
                                from ACCOUNT A ,DEPOSITOR D
                                where D.accno = A.accno and
                                      A.branchname = B.branchname and
                                      D.customername = C.customername
                                group by A.branchname
                                having count(*) >= 2
                              )
                )
```

Find all the customers who have accounts in atleast 1 branch located in all the cities

```
select C.customername
from CUSTOMER  C
where not exists( select distinct(B.branchcity)
                  from  BRANCH B
                  where not exists( select A.branchname
                                    from ACCOUNT A ,DEPOSITOR D
                                    where  D.accno = A.accno
                                           and D.custname =C.custname
                                           and A.branchname
                                           in( select branchname
                                               from BRANCH
                                               where brcity = B.brcity
                                             )
                                  )
                )
```

Find all the customers who have accounts in atleast 2 branches located in a specific city

```
select C.customername
from CUSTOMER  C
where  exists( select  count( distinct B.branchname)
               from BRANCH B, ACCOUNT A ,DEPOSITOR D
               where A.branchname = B.branchname
                     and D.accno = A.accno
                     and B.branchcity = 'karkala'
                     and D.customername = C. customername
               group by B.branchcity
               having count(*) >=2
             )
```