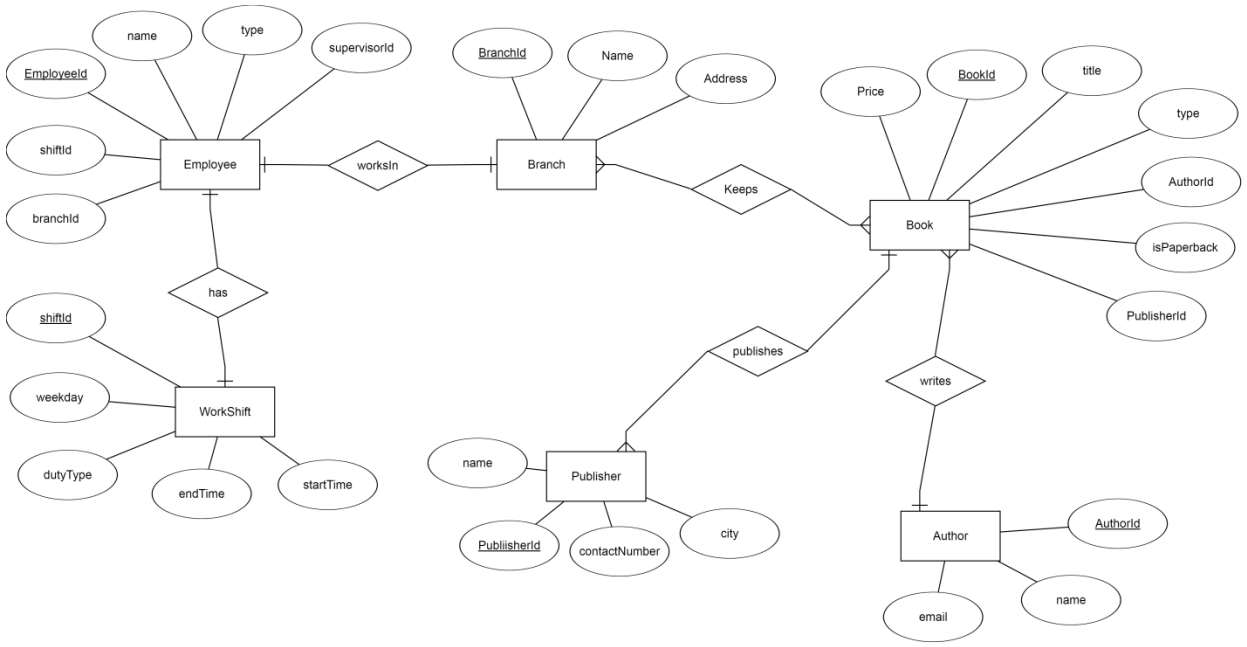
Task1 : ER Diagram



Task 2:

Tables would be as following:

Author

AuthorId	Name	Email

Publisher

PublisherId	Name	City	contactnumber

Branch

branchId	name	address

Book

BookId	title	type	price	isPaperback	AuthorId	PublisherId

WorkShift

shiftId	weekday	dutyType	startTime	endTime

Employee

EmployeeId	name	type	supervisorId	shiftId	branchid

BranchHasBooks

branchId	bookId	count

Below are the set of functional dependencies for the relation

AuthorId->name, email

publisherId->name, city, contactnumber

BranchId->name, address

BookId->title, price, type, authored, publisherId

Bookid->authored,

BookId->publisherId

Shiftd->weekday, dutytype, startTime, endTime

EmployeeId->name, type, supervisorId

EmployeeId-> shiftId

EmployeeId-> branchid

branchId, bookid -> count

1NF: Since all the attributes are atomic hence the relation is in 1NF

2NF: Since there are no partial functional dependency it is also in 2NF

3NF: As there are no partial dependency in the relation, hence it is in 3NF as well.

BCNF: As there are no multiple overlapping dependencies, this is in BCNF.

So the highest normal form the relation is in BCNF

Task 3

DDL

```
create table Author(
Authorid number,
name varchar2(40),
email varchar2(40),
CONSTRAINT Author_pk PRIMARY KEY (Authorid)
);

create table Publisher(
Publisherid number,
name varchar2(40),
city varchar2(40),
contactnumber varchar2(40),
CONSTRAINT Publisher_pk PRIMARY KEY (Publisherid)
);

create table Branch(
Branchid number,
name varchar2(40),
address varchar2(80),
```

```
CONSTRAINT Branch_pk PRIMARY KEY (Branchid)
);

create table Book(
BookId number,
title varchar2(40),
booktype varchar2(80),
price number(10, 2),
isPaperback varchar2(2),
AuthorId number,
PublisherId number,
CONSTRAINT Book_pk PRIMARY KEY (BookId),
CONSTRAINT fk_Book1
    FOREIGN KEY (AuthorId)
    REFERENCES Author(AuthorId),
CONSTRAINT fk_Book2
    FOREIGN KEY (PublisherId)
    REFERENCES Publisher(PublisherId)
);


create table WorkShift(
shiftId number,
weekday varchar2(40),
dutyType varchar2(80),
startTime varchar2(80),
endTime varchar2(80),
CONSTRAINT WorkShift_pk PRIMARY KEY (shiftId)
);


create table Employee(
EmployeeId number,
name varchar2(40),
type varchar2(80),
supervisorId number,
shiftId number,
branchid number,
CONSTRAINT Employee_pk PRIMARY KEY (EmployeeId),

CONSTRAINT fk_Employee1
    FOREIGN KEY (shiftId)
    REFERENCES WorkShift(shiftId),
CONSTRAINT fk_Employee2
    FOREIGN KEY (branchid)
    REFERENCES Branch(branchid),
CONSTRAINT fk_Employee3
    FOREIGN KEY (supervisorId)
    REFERENCES Employee(EmployeeId)
);


create table BranchHasBooks(
branchId number,
bookId number,
count number,
CONSTRAINT BranchHasBooks_pk PRIMARY KEY (branchId, bookId),
```

```
CONSTRAINT fk_BranchHasBooks1
    FOREIGN KEY (branchId)
    REFERENCES Branch(branchId),
CONSTRAINT fk_BranchHasBooks2
    FOREIGN KEY (bookId)
    REFERENCES book(bookId)
);
```

Task 4: Inserts

```
-- Author
insert into author
  (AUTHORID, NAME, EMAIL)
values
  (5, 'Robert Frost', 'robert.frost@email.com');

insert into author
  (AUTHORID, NAME, EMAIL)
values
  (4, 'William Shakespeare', 'william@email.com');

insert into author
  (AUTHORID, NAME, EMAIL)
values
  (3, 'Nicholoas Sparks', 'nicholar.sparks@email.com');

insert into author
  (AUTHORID, NAME, EMAIL)
values
  (2, 'J K Rowling', 'jkrowling@email.com');

insert into author
  (AUTHORID, NAME, EMAIL)
values
  (1, 'Agatha Christie', 'agatha@email.com');

--branch
insert into branch
  (BRANCHID, NAME, ADDRESS)
values
  (3,
   'Central business district',
   'Central business district, Sydney, New South Wales');

insert into branch
  (BRANCHID, NAME, ADDRESS)
values
  (2, 'Bankstown', 'Bankstown Sydney, New South Wales');

insert into branch
  (BRANCHID, NAME, ADDRESS)
values
```

```
  (1, 'Richmond', 'Richmond Sydney New South Wales');

--publisher
insert into publisher
  (PUBLISHERID, NAME, CITY, CONTACTNUMBER)
values
  (5, 'Kiran Publication', 'New Delhi', '9993182827');

insert into publisher
  (PUBLISHERID, NAME, CITY, CONTACTNUMBER)
values
  (4, 'SmokeHouse Publishing', 'Torronto', '6198890090');

insert into publisher
  (PUBLISHERID, NAME, CITY, CONTACTNUMBER)
values
  (3, 'Evergreen Publication', 'Calgary', '4038089273');

insert into publisher
  (PUBLISHERID, NAME, CITY, CONTACTNUMBER)
values
  (2, 'Big Machine', 'Nashvile', '5123889990');

insert into publisher
  (PUBLISHERID, NAME, CITY, CONTACTNUMBER)
values
  (1, 'Red House', 'Seattle', '5234567789');

--workshift
insert into WorkShift
  (SHIFTID, WEEKDAY, DUTYTYPE, STARTTIME, ENDTIME)
values
  (2, 'Monday', 'sales', '8.00', '4.00');

insert into WorkShift
  (SHIFTID, WEEKDAY, DUTYTYPE, STARTTIME, ENDTIME)
values
  (1, 'Sunday', 'cleaning', '7.30', '4.30');

insert into WorkShift
  (SHIFTID, WEEKDAY, DUTYTYPE, STARTTIME, ENDTIME)
values
  (3, 'Tuesday', 'security', '6.00', '5.00');

--employee
insert into Employee
  (EMPLOYEEID, NAME, TYPE, SUPERVISORID, SHIFTID, BRANCHID)
values
  (1, 'Michale Hussey', 'permanent', null, 2, 2);

insert into Employee
  (EMPLOYEEID, NAME, TYPE, SUPERVISORID, SHIFTID, BRANCHID)
values
  (2, 'Ricky ponting', 'contractual', 1, 2, 1);

insert into Employee
  (EMPLOYEEID, NAME, TYPE, SUPERVISORID, SHIFTID, BRANCHID)
```

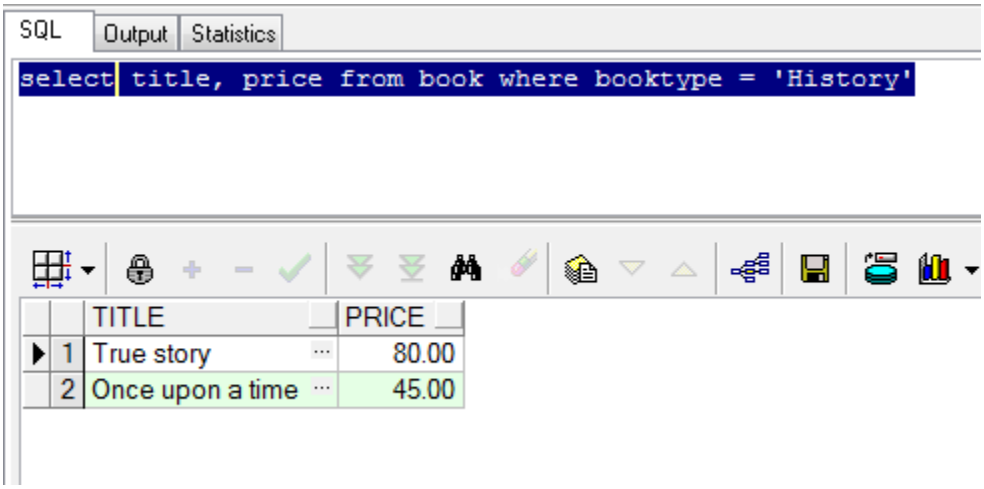
```
values
  (3, 'Glen maxwel', 'casual', 2, 3, 3);

insert into Employee
  (EMPLOYEEID, NAME, TYPE, SUPERVISORID, SHIFTID, BRANCHID)
values
  (4, 'Steve Smith', 'casual', 2, 3, 2);

insert into Employee
  (EMPLOYEEID, NAME, TYPE, SUPERVISORID, SHIFTID, BRANCHID)
values
  (5, 'Rafael Nadal', 'permanent', 3, 2, 2);

insert into Employee
  (EMPLOYEEID, NAME, TYPE, SUPERVISORID, SHIFTID, BRANCHID)
values
  (6, 'Hugh Jackman', 'permanent', 2, 2, 3);

insert into Employee
  (EMPLOYEEID, NAME, TYPE, SUPERVISORID, SHIFTID, BRANCHID)
values
  (8, 'Steve Waugh', 'contractual', 5, 2, 1);

insert into Employee
  (EMPLOYEEID, NAME, TYPE, SUPERVISORID, SHIFTID, BRANCHID)
values
  (9, 'Brett Lee', 'casual', 2, 3, 2);

insert into Employee
  (EMPLOYEEID, NAME, TYPE, SUPERVISORID, SHIFTID, BRANCHID)
values
  (10, 'Andy Flower', 'permanent', 3, 1, 3);

--BranchHasBooks

insert into BranchHasBooks (BRANCHID, BOOKID, COUNT) values (3, 1, 134);

insert into BranchHasBooks (BRANCHID, BOOKID, COUNT) values (2, 7, 134);

insert into BranchHasBooks (BRANCHID, BOOKID, COUNT) values (2, 6, 34);

insert into BranchHasBooks (BRANCHID, BOOKID, COUNT) values (2, 5, 21);

insert into BranchHasBooks (BRANCHID, BOOKID, COUNT) values (2, 4, 245);

insert into BranchHasBooks (BRANCHID, BOOKID, COUNT) values (2, 3, 24);

insert into BranchHasBooks (BRANCHID, BOOKID, COUNT) values (2, 2, 54);

insert into BranchHasBooks (BRANCHID, BOOKID, COUNT) values (2, 1, 33);

insert into BranchHasBooks (BRANCHID, BOOKID, COUNT) values (1, 7, 12);

insert into BranchHasBooks (BRANCHID, BOOKID, COUNT) values (1, 6, 344);

insert into BranchHasBooks (BRANCHID, BOOKID, COUNT) values (1, 5, 12);
```

```
insert into BranchHasBooks (BRANCHID, BOOKID, COUNT) values (1, 4, 10);
insert into BranchHasBooks (BRANCHID, BOOKID, COUNT) values (1, 3, 30);
insert into BranchHasBooks (BRANCHID, BOOKID, COUNT) values (1, 2, 20);
insert into BranchHasBooks (BRANCHID, BOOKID, COUNT) values (1, 1, 10);
insert into BranchHasBooks (BRANCHID, BOOKID, COUNT) values (3, 2, 22);
insert into BranchHasBooks (BRANCHID, BOOKID, COUNT) values (3, 3, 43);
insert into BranchHasBooks (BRANCHID, BOOKID, COUNT) values (3, 4, 234);
insert into BranchHasBooks (BRANCHID, BOOKID, COUNT) values (3, 5, 22);
insert into BranchHasBooks (BRANCHID, BOOKID, COUNT) values (3, 6, 1);
insert into BranchHasBooks (BRANCHID, BOOKID, COUNT) values (3, 7, 21);
```

Task 5:

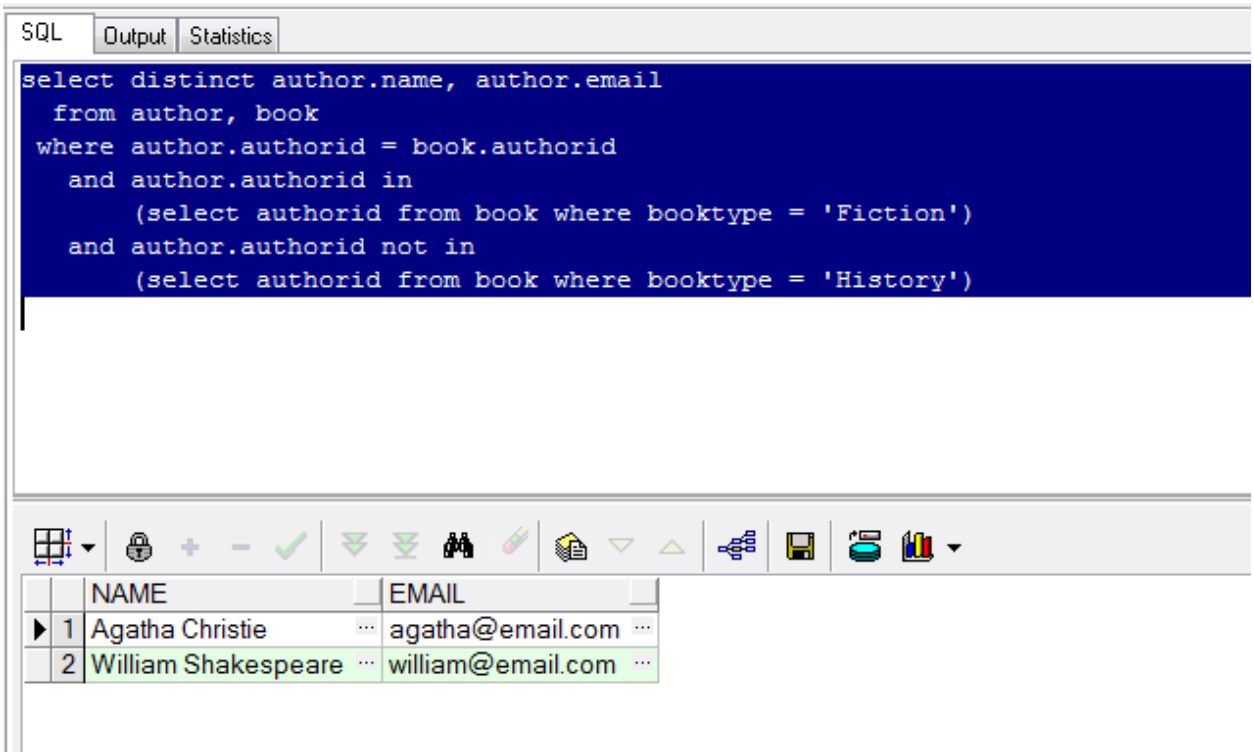
a) Find the book title and price of historical books.

```
Select title, price from book where booktype = 'History';
```



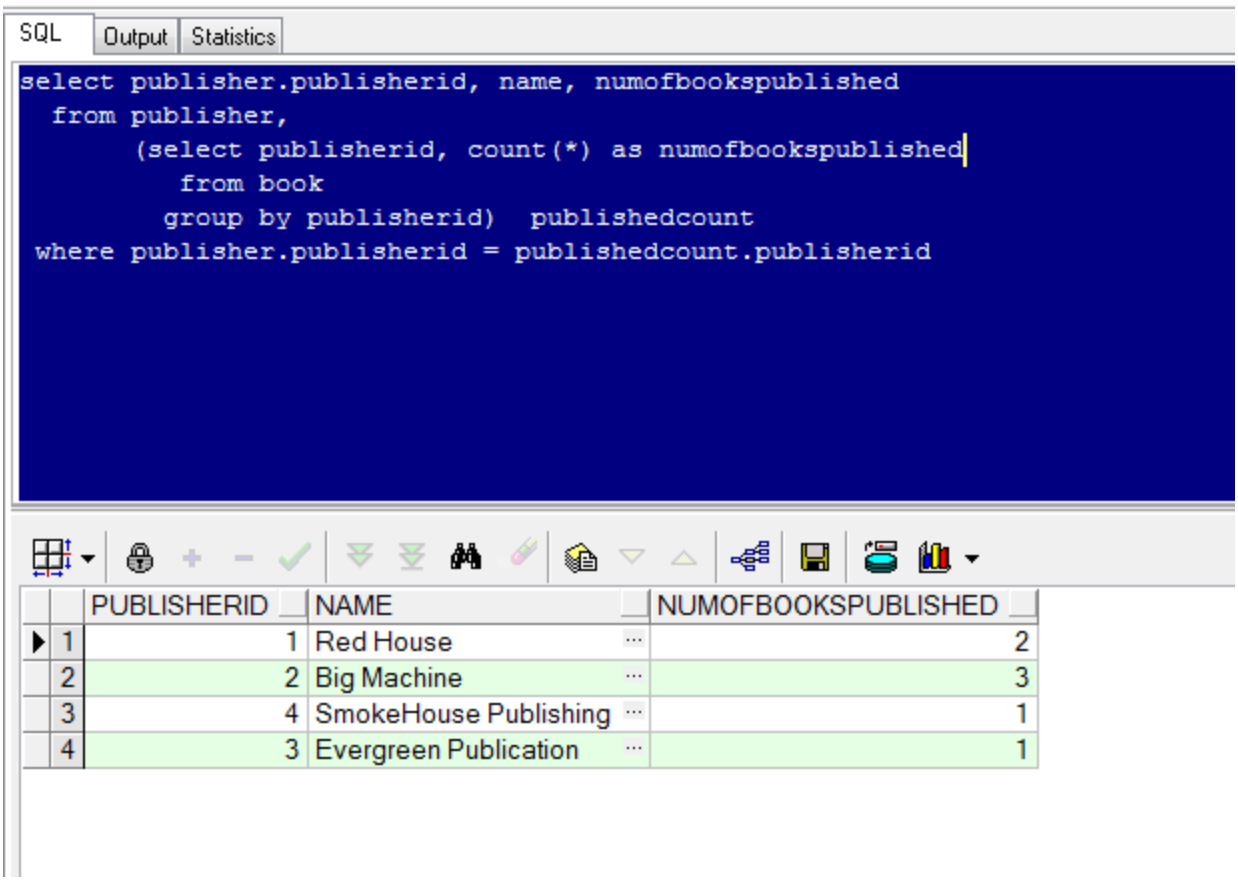
b) Find the name and email address of all authors who wrote at least one fiction but no historical book.

```
select distinct author.name, author.email
  from author, book
 where author.authorid = book.authorid
   and author.authorid in
       (select authorid from book where booktype = 'Fiction')
   and author.authorid not in
       (select authorid from book where booktype = 'History');
```



c) Find the total number of books published by each publisher.

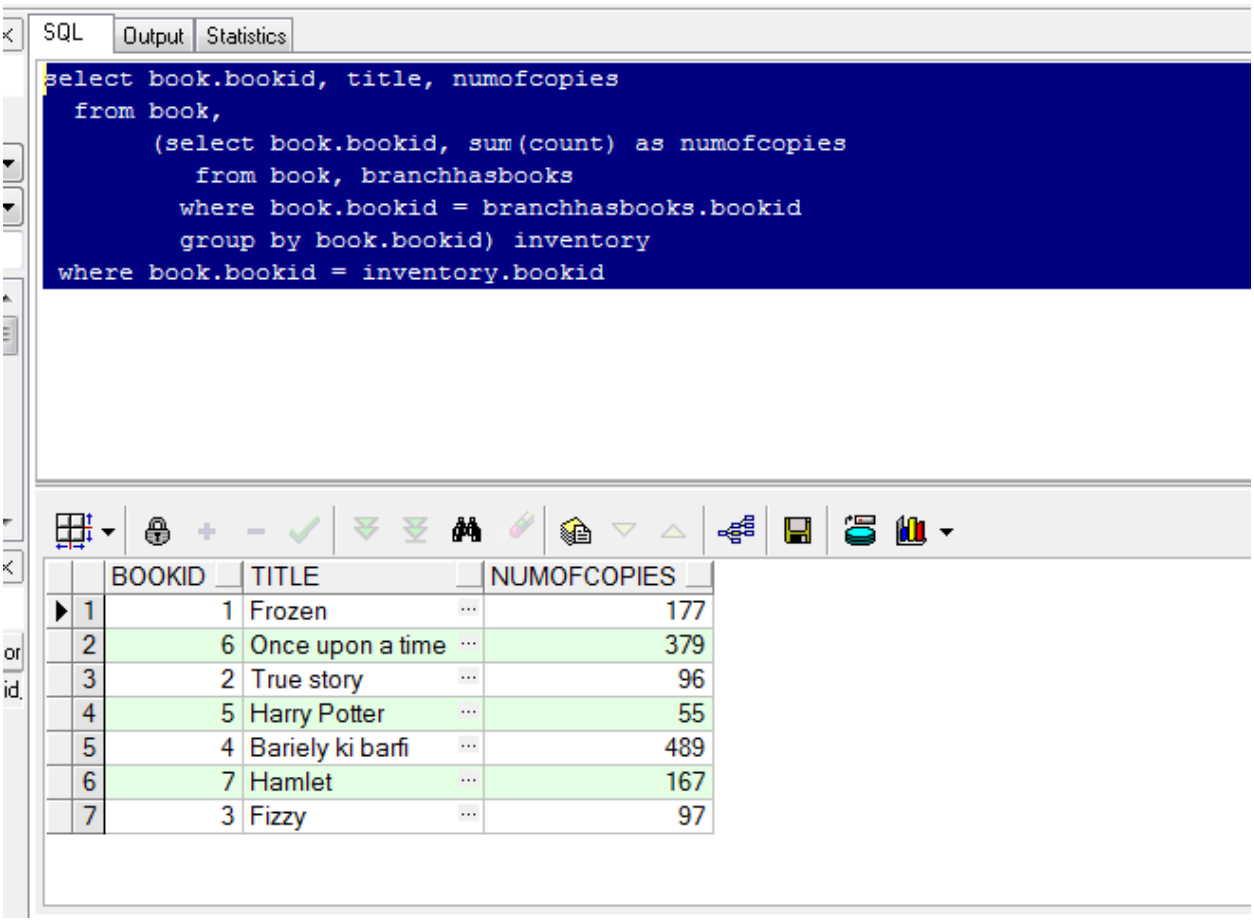
```
select publisher.publisherid, name, numofbookspublished
  from publisher,
       (select publisherid, count(*) as numofbookspublished
          from book
         group by publisherid)  publishedcount
 where publisher.publisherid = publishedcount.publisherid
```



	PUBLISHERID	NAME	NUMOFBOOKSPUBLISHED
1	1	Red House	2
2	2	Big Machine	3
3	4	SmokeHouse Publishing	1
4	3	Evergreen Publication	1

d) Find the total number of copies in the inventory (including all branches) for each book. Include only the top five books based on the number of copies available in the result set.

```
select book.bookid, title, numofcopies
  from book,
       (select book.bookid, sum(count) as numofcopies
          from book, branchhasbooks
         where book.bookid = branchhasbooks.bookid
         group by book.bookid) inventory
 where book.bookid = inventory.bookid
```


SQL
Output
Statistics
select book.bookid, title, numofcopies
from book,
(select book.bookid, sum(count) as numofcopies
from book, branchhasbooks
where book.bookid = branchhasbooks.bookid
group by book.bookid) inventory
where book.bookid = inventory.bookid
BOOKID
TITLE
NUMOFCOPIES
1 1 Frozen 177
2 6 Once upon a time 379
3 2 True story 96
4 5 Harry Potter 55
5 4 Bariely ki barfi 489
6 7 Hamlet 167
7 3 Fizzy 97