הפעולות באלגברת יחסים

1)

select book.name, inventory.count from book left join inventory on book.book_id = inventory.book_id where name='Robinson Crusoe' And inventory.count>0;

π B.name, I.count

B $\bowtie_{B.book\_id = I.book\_id}$ (I)

σ B.name='?' ∧ I.count>0

[ρB(Book)×*ρI*(Inventory)]

2)

select concat(first_name, " ", last_name) as name from customer order by date_created limit 1;

∏c.id (customer) –

∏C.customer_id((C)$\bowtie_{c.customer\_id = C2.book\_id}$(I)

σ I.count>0

[ρB(Book)x *ρI*(Inventory))*)*

3)

```
select book.*
 from book inner
 join inventory on book.book_id = inventory.book_id
where inventory.count > 0
order by time_stamp limit 1;
```

π-B.book_id(BOOKS) -

(πI1.book_id((I1)⋈I1. book _id = I2. book _id(I2)

σI1.count>0
I1.time_stamp>I2.time_stamp(ρI1(Inventory)×ρI2(Inventory)))

4)

```
select * from reservation
 inner join reservation_book on reservation.reservation_id =
reservation_book.reservation_id
 inner join book on book.book_id = reservation_book.book_id
inner join customer c on c.customer_id =
reservation.reservation_id
order by time_stamp;
```

(R) ⋈ R.reservation_id = RB.reservation_id (RB)

(B) ⋈ R.book_id = RB.book_id (RB)

(C) ⋈ C. reservation _id = R. customer _id (R)

[ρR(Reservation)]

5)

```
select book.name, sum(inventory.sold_counter) as
sold_books_amount
 from book
left join inventory on book.book_id = inventory.book_id
where name='Yaels House';
```

Π B.name, γsum(i.sold_counter)

B ⋈ B.book_id = I.book_id (I)

σ B.name='?'

[ρB(Book)]

6)

```
select a.first_name, a.last_name

from author a, purchase_customer pc, purchase_customer_book
pcb, author_book ab

where pc.purchase_id = pcb.purchase_id

      AND pcb.book_id = ab.book_id

      AND a.author_id = ab.author_id
```

Π A.first_name, A.last_name

σ pc.purchase_id = pcb.purchase_id ∧ pcb.book_id = ab.book_id

∧ a.author_id = ab.author_id

[ρA(Author)X *ρpc* (purchase_customer) X *ρpc*b
(purchase_customer_book) X *ρAB* (author_book)]

8)

```
select count(b.name) as translations_amount, b.name
from book b inner join inventory i on b.book_id = i.book_id
inner join book_edition be on b.book_id = be.book_id
where be.translator IS NOT NULL
group by b.name order by count(b.name) desc limit 1 ;
```

Π b.name –

(Π b.name

$B \bowtie_{B.book\_id= I.book\_id} (I)$

$B \bowtie_{D.book\_id= BE.book\_id} (BE)$

σ be.translator!=NULL

[ρB(Book)X *ρI*(Inventory) X *ρBE*(Book_Edition)])

γ (B.name)(*ρB*(Book) X *ρ*γcount (B.name)(BOOK)]

9)

select book.name, pc.purchase_date, pcb.price, pc.purchase_id

from book, purchase_customer pc, purchase_customer_book pcb, customer c

where c.customer_id = pc.customer_id AND pc.purchase_id = pcb.purchase_id AND book.book_id = pcb.book_id AND c.customer_id='5' order by pc.purchase_date;

Π b.name, pc.purchase_date, pcb.price, pc.purchase_id

$D \bowtie_{D.purchase\_id= pc.purchase\_id} (PC)$

$PC \bowtie_{D.purchase\_id= pcb.purchase\_id} (PCB)$

$B \bowtie_{D.book\_id= pcb.book\_id} (PCB)$

$D \bowtie_{D.\ company\ \_id= pc.company\_id} (DC)$

σ c.customer_id = pc.customer_id ∧ pc.purchase_id = pcb.purchase_id ∧book.book_id = pcb.book_id ∧c.customer_id='?'

[ρB(Book)X *ρC*(Customer) X *ρPC*(Purchase_customer) X *ρPCB*(Purchase_customer_book)]

10)

```
select book.name, rb.count, reservation.time_stamp, inventory.*,
pcb.*, pc.purchase_date from customer inner join reservation on
customer.customer_id = reservation.customer_id
        inner join reservation_book rb on rb.reservation_id =
reservation.reservation_id
        inner join book on book.book_id = rb.book_id
        left join inventory on inventory.book_id = book.book_id
    left join purchase_customer_book pcb on pcb.book_id =
book.book_id
    left join purchase_customer pc on pc.purchase_id =
pcb.purchase_id AND pc.customer_id = customer.customer_id
    where customer.customer_id = 1
    order by reservation.time_stamp;
```

Π b.name, rb.count, r.time_stamp, i*, pcb.*, pc.purchase_date

PCB $\bowtie_{PCB.purchase\_id= pc.purchase\_id}$(PC)

PC $\rtimes_{PC.book\_id= PCB.book\_id}$(PCB)

PC $\rtimes_{PC.customer\_id= C.Customer\_id}$(C)

B $\rtimes_{B.book\_id = I.book\_id}$ (I)

B $\bowtie_{B.book\_id= R.book\_id}$(RB)

B $\bowtie_{B.book\_id= I.book\_id}$(I)

R $\bowtie_{R.reservation\_id= R.reservation\_id}$(RB)

σ c.customer_id='?'

[ρB(Book)X *ρI*(Inventory) X *ρPC*(Purchase_customer) X *ρPCB*(Purchase_customer_book) X *ρR*(Reservation) X *ρR*B(Reservation_Book) X *ρC*(customer)]

11)

select sum((d.price * book.weight)+pcb.price)

from purchase_customer pc

inner join deliveries d on pc.purchase_id = d.purchase_id

inner join purchase_customer_book pcb on pcb.purchase_id = pc.purchase_id

inner join book on book.book_id = pcb.book_id

where pc.purchase_id = 5;

Π γsum(d.price * book.weight)+pcb.price)

D $\bowtie_{D.purchase\_id= pc.purchase\_id}$(PC)

PC $\bowtie_{D.purchase\_id= pcb.purchase\_id}$(PCB)

B $\bowtie_{D.book\_id= pcb.book\_id}$(PCB)

σ pc.purchase_id='?'

[ρB(Book)X *ρD*(Deliveries) X *ρPC*(Purchase_customer) X *ρPC*B(Purchase_customer_book) X *ρ*γsum(d.price * book.weight)+pcb.price) (delivery_cost)]

12)

```
select * from deliveries d
left join purchase_customer pc  on pc.purchase_id =
d.purchase_id
 where pc.customer_id = '2' and d.purchase_id in
(select d.purchase_id from deliveries d
group by d.purchase_id
having count(d.purchase_id)>1)
 order by d.purchase_id;
```

Π D.*

/

σ c.customer_id =

((Π C,customer_id

σ pc.customer_id '?' ∧ d.purchase_id =

(Π D.purchase_id,

σ D.purchase_id)

[ρC(Customer)X *ρPC*(Purchase_customer])

/ (Π C,customer_id

σ pc.customer_id = c.customer_id ∧
pc.purchase_date>'last_year_date'

[ρC(Customer)X *ρPC*(Purchase_customer])

[(C.Customer_idγ*ρC*(Customer))

ρC(Customer)X *ρPC*(Purchase_customer])

13)

select d.status from deliveries d, purchase_customer pc where d.purchase_id = pc.purchase_id And d.purchase_id=2;

Π d.status

σwhere d.purchase_id = pc.purchase_id ∧d.purchase_id='?';

[ρD(Deliveries)x , *ρpc*(purchase_customer)]

14)

```
select sum((d.price * book.weight)) from purchase_customer pc
inner join deliveries d on pc.purchase_id = d.purchase_id
inner join purchase_customer_book pcb on pcb.purchase_id = pc.purchase_id
inner join book on book.book_id = pcb.book_id
 inner join delivery_company dc on dc.company_id = d.delivery_company
 where dc.company_name = 'Xpress' AND month(d.delivery_date) = 7 AND year(d.delivery_date) = 2019;
```

Π pc.purchase_customer

$D \bowtie_{D.purchase\_id= pc.purchase\_id} (PC)$

$PC \bowtie_{D.purchase\_id= pcb.purchase\_id} (PCB)$

$B \bowtie_{D.book\_id= pcb.book\_id} (PCB)$

$D \bowtie_{D.\ company\ \_id= pc.company\_id} (DC)$

σdc.company_name='Xpress' ∧month(d.delivery_date)= '?' ∧year(d.delivery_date) = '?'

[ρB(Book)X *ρD*(Deliveries) X *ρPC*(Purchase_customer) X *ρPC*B(Purchase_customer_book) X *ρD*(Deliveries_company) X *ρ*γsum (d.price * book.weight)]

15)

```
select sum(pcb.price) from purchase_customer pc
 inner join purchase_customer_book pcb on pc.purchase_id =
pcb.purchase_id
where pc.payment_method='bit'
        AND month(pc.purchase_date) = 7
        AND year(pc.purchase_date) = 2020;
```

Π sum((d.price * book.weight))

PC $\bowtie_{D.purchase\_id= pcb.purchase\_id}$(PCB)

σ pc.payment_method='bit' ∧month(pc.purchase_date) = '?' ∧ year(pc.purchase_date)= '?'

[ρB(Book)X *ρD*(Deliveries) X *ρ*γsum (d.price * b.weight)]

16)

```
select pcb.purchase_id,sum(pcb.price) as sum_purchase,
pc.purchase_date
 from purchase_customer pc, purchase_customer_book pcb where
pc.purchase_id = pcb.purchase_id AND pc.purchase_date>'2018-
08-01'
group by pc.purchase_date
having sum_purchase>
(select avg(pcb.price) as annual_avg from purchase_customer pc
inner join purchase_customer_book pcb
where pc.purchase_id=pcb.purchase_id AND
pc.purchase_date>'2018-08-01');
```

17)

```
select count(*), dc.company_name
from deliveries d, delivery_company dc
 where dc.company_id = d.delivery_company AND
d.delivery_date>'2019-06-02'
group by dc.company_name;
```

Π dc.company_name

(count (*)γ as delivered_books

(σ dc.company_id = d.delivery_company ∧ d.delivery_date>'?'

(*dc.delivery_company* γ*ρDC*(Delivery_Company)X

ρDC(Deliveries))

18)

```
select d.*, d.deliveries_id, b.name from deliveries d
inner join deliveries_books db on d.deliveries_id = db.deliveries_id
inner join book b on db.book_id = b.book_id
group by d.deliveries_id, b.name having count(*)>=2;
```

Π d.*, b.name

/(Π D.deliveries_id (σ dc.company_id = d.delivery_company

DB ⋈ DB.Deliveries_books= D.Deliveries_books (D)

DB ⋈ DB.Book_id= B.book_id(B)

(*dc.delivery_company* count(*)>2,d.deliveries_idγ *ρDC*(Deliveries))*XρB*(Book))

19)

select c.*, pc.purchase_date

from purchase_customer pc, customer c

where pc.customer_id = c.customer_id

AND pc.purchase_date<'2019-07-27'

and c.customer_id NOT IN

(select c.customer_id from purchase_customer pc, customer c

where pc.customer_id = c.customer_id AND
pc.purchase_date>'2019-07-27') group by c.customer_id;

Π C.*, pc.purchase_date

σ c.customer_id =

((Π C,customer_id

σ pc.customer_id = c.customer_id ∧
pc.purchase_date<'last_year_date'

[ρC(Customer)X *ρPC*(Purchase_customer])

- (Π C,customer_id

σ pc.customer_id = c.customer_id ∧
pc.purchase_date>'last_year_date'

[ρC(Customer)X *ρPC*(Purchase_customer])

[(C.Customer_idγ*ρC*(Customer))

ρC(Customer)X *ρPC*(Purchase_customer])

20)

```
select c.*,r.contact_customer from customer c, reservation r,
reservation_book rb

  where r.customer_id = c.customer_id AND rb.reservation_id =
r.reservation_id AND r.contact_customer

          IS NOT NULL and r.contact_customer <= curdate() - 14

AND NOT EXISTS

( select * from purchase_customer pc, purchase_customer_book
pcb

where pc.purchase_id = pcb.purchase_id AND pc.customer_id =
c.customer_id AND pcb.book_id = rb.book_id);
```

21)

```
select sum(i2.count) as
AccumelatedNoOfBooks,month(i1.time_stamp),year(i1.time_stamp
)
from inventory i1
join inventory i2 on month(i2.time_stamp) <=month(i1.time_stamp)
and year(i2.time_stamp)<=year(i1.time_stamp)
 where i1.location = 'storage'
and i1.count>0
and i2.location='storage'
and i2.count>0
and i2.time_stamp<=i1.time_stamp
group by month(i1.time_stamp),year(i1.time_stamp)
order by month(i1.time_stamp),year(i1.time_stamp);
```

∏ month(i1.time_stamp),year(i1.time_stamp) ,sum(i2.count)

I1 ⋈ month(i2.time_stamp) <=month(i1.time_stamp) (I2)

I1 ⋈ year(i2.time_stamp)<=year(i1.time_stamp) (I2)

(month(i1.time_stamp),year(i1.time_stamp) sum(i2.count) γ as AccumelatedNoOfBooks

(σ i1.location = 'storage' ∧ i1.count>0 ∧ i2.location='storage' ∧ i2.count>0 ∧ i2.time_stamp<=i1.time_stamp

(*ρI1*(Inventory)x *ρI2*(Inventory))))

22)

```
select count(*) as books_bought_by_store,
sum(ps.count*ps.book_price) as books_payment
 from purchase_store ps
where ps.purchase_date between '2008-01-01' and '2020-01-29';
```

(count (*)γ as books_bought_by_store
sum(ps.count*ps.book_price) γ as books_payment

(σ '?'<ps.purchase_date<'?' (*ρPS*(purchase_store))

(*ρPS*(Purchase_customer))))

23)

```
select avg(pcb.price), year(pc.purchase_date) from
purchase_customer pc
inner join purchase_customer_book pcb
where pc.purchase_id=pcb.purchase_id
group by year(pc.purchase_date)
order by year(pc.purchase_date);
```

year(pc.purchase_date) γAVG(price) (pcb)

PC ⋈ (PCB)

σ pc.purchase_id=pcb.purchase_id

(pc.year(purchase_date)γ*ρpc*(purchase_customer))

24)

```
select wmh.hour_payment*wmh.hours
from working_month_hours wmh
inner join employees e on e.employee_id = wmh.employee_id
where e.employee_id=1
AND wmh.month = 6 AND wmh.year = 2018;
```

E ⋈ e.employee_id =wmh.employee_id (WMH)

σ e.employee_id='?' ∧ wmh.month = '?'

∧ wmh.year= '?'

((wmh.hour_payment*wmh.hours as hour_calcγ*ρWMH*(working_month_hoursX *ρE*(Employee))

25)

```
select e.first_name, e.last_name
from employees e
 inner join purchase_customer pc on e.employee_id =
pc.employee_id
 where month(pc.purchase_date)= 7
     AND year(pc.purchase_date) = 2020
     group by e.employee_id order by count(*) desc limit 1;
```

X<-Π e.id

E ⋈ e.employee_id = pc.employee_id (PC)

σ month(pc.purchase_date)= '?'

∧ year(pc.purchase_date) = '?'

(*ρE*(Employee)*XρPC*(Purchase_customer)))

Y<-Π pc.employee_id

(ρPC(Purchase_customer))

Z<- Π e.first_name,e.last_name(X/Y)

(e.employee_idγ*ρE*(Employee))

search book by name:

select * from book b, inventory i where b.book_id = i.book_id AND b.name = 'Single for the Summer;'

Πbook b,inventory i σ b.book_id = i.book_id ∧ b.name ='?'*[ρB*(Book)×*ρI*(Inventory)]

search book by author:

SELECT b.book_id, b.name,

  GROUP_CONCAT(CONCAT(a.first_name,' ' , a.last_name)) AS author_names,

  GROUP_CONCAT(a.author_id) as author_ids

FROM book b

LEFT JOIN author_book ab ON b.book_id = ab.book_id

LEFT JOIN author a ON ab.author_id = a.author_id

WHERE a.first_name= 'Laurence'

GROUP BY b.book_id;

Π B.book_id, B.name,A.author_id,A.author_id

B ⋈ B.book_id = AB.book_id (AB)

AB ⋈ AB.author_id = A.author_id (A)

σ A.first_name='?' or A.last_name='?'

[ρA(Author)X *ρAB* (author_book) X *ρ*γbook_id(book)]