[איתמר שושן, שני לוי]

שאילתות באלגברה יחסים

1

σ title = ? AND amount>0 (books);

2

σ Join_date>=2000-01-01 (customers);

3

σ available_date>='2000-01-01' (books);

4

π order_id,order_date,order_complited, first_name,last_name
(σ order_date >='2020-06-01' AND order_date < now() (orders o⋈ o.customer_id = customers c.customer_id));

5

π count() (σ b.title = ? (book_list_for_orders bfo ⋈ bfo.book_id = books b.book_id ⋈ bfo.order_id = orders o.oderd_od));

6

π a.first_name, a.last_name, count(ba.author_id)>1 (σ o.order_date >= ? AND o.order_date < ? (book b ⋈ b.book_id = book_list_for_orders bfo.book_id ⋈ bro.order_id = orders o.order_id ⋈ b.book_id = book_authors ba.book_id ⋈ authors a.author_id = ba.author_id));

 7

π first_name, last_name, sum(amount) (σ order_complited = 'yes' AND rownum() > 0 AND rownum() <= 3 (orders o ⋈ o.customer_id = customers c.customer_id);

8

π title, count(b.book_id)>0 (σ b.amount>0 AND rownum() > 0 AND rownum() <= 1 (book_translators bt ⋈ books b.book_id = bt.book_id ⋈ translators t.translator_id = bt.translator_id));

9

π o.order_date, o.pyment_date, o.order_status, b.title, b.price (σ c.first_name = ? AND c.last_name = ? (book_list_for_orders bo ⋈ books b.book_id = bo.book_id ⋈ bo.order_id = orders o.order_id ⋈ o.customer_id = customers c.customer_id));

10

[איתמר שושן, שני לוי]

π o.order_date, o.order_status, b.title,b.amount, o.oder_complited, b.price (σ c.first_name=? AND c.last_name=?(book _list_for_orders bo ⋈ b.book_id=bo.book_id ⋈ bo.order_id=orders o.order_id ⋈ o.customer_id=customers c.customer_id));

11

π d.adress, city, d.delivery_id, sum(b.price+b.weight+ExtraXpress) (σ d.address = ? AND num_address = ? AND city = ? (book_list_for_orders bo ⋈ orders o.order_id = bo.order_id ⋈ books b.book_id = bo.book_id ⋈ o.customer_id = customers c.customer_id ⋈ delivery d.delivery_id = bo.delivery_id ⋈ delivery_for_orders dd.delivery_id = d.delivery_id));

12

π dfo.delivery_id, o.order_id, o.order_date, d.city, d.address, b.title (σ c.first_name = ? AND c.last_name = ? (book_list_for_order bo ⋈ bo.order_od = orders o.order_id ⋈ book b.book_id = bo.book_id ⋈ o.customer_id = customers c.customer_id ⋈ delivery_for_order dfo.delivery_id = bo.delivery_id ⋈ dfo.delivery_id = delivery d.delivery_id));

13

π dfo.delivery_status (σ c.mobile = ? AND d.address = ? AND d.city = ?
(delivery_for_orders dfo ⋈ delivery_id d.delivery_id = dfo.delivery_id ⋈ orders o.order_id = dfo.order_id ⋈ o.customer_id = customers c.customer_id));

14

π count(month(delivery_date)) (σ dfo.shipment = 'Xpress' AND month(delivery_date) = ? AND year(delivery_date) = ? (delivery_for_orders dfo ⋈ orders o.order_id = dfo.order_id));

15

π sum(price) (σ o.payment_method = 'Transfer by Bit' ND month(o.pyment_date) = ? AND year(o.pyment_date) = ? (book_list_for_orders blfo ⋈ orders o.order_id = blfo.order_id ⋈ books b.book_id = blfo.book_id));

16

π price_for_order > sum (price_for_order/12), order_id (σ o.order_date >= '2019-06-01' AND o.order_date< now() AND order_complited = 'yes' AND pyment date >=
' 2019-06-01' AND pyment_date< now()(orders o));

17

π count(dfo.shipent) >=1 , dfo.shipment (σ dfo.delivery_date >='2019-07-01' and dfo.delivery_date<now() (delivery_for_orders dfo ⋈ delivery d.delivery_id = dfo.delivery_id ⋈ order o.order_id = dfo.irder_id));

18

π address, city, dfo.delivery_date, d.delivery_id (σ max(ps.edition) > min(ps.edition) (delivery_for_orders dfo ⋈ delivery d.delivery_id = dfo.delivery_id ⋈ orders o.order_id = dfo.order_id ⋈ o.order_id =

book_list_for_orders blo.order_id ⋈ publisher p.publisher_book_id = blo.publisher_book_id ⋈ p.publisher_id = publishers ps.publisher_id));

19

π count(mobile)>0->mobile, last_name, first_name (σ o.order_date >= '2000-01-01' AND o.order_date < '2018-08-01' AND order_complited = 'yes' (orders o ⋈ o.customer_id = customers c.customer_id);

π c.customer_id (σ order_Date >= '2018-08-01' AND order_date < now() AND o.order_complited = 'no' (o ⋈o.customer_id = c.customer_id)));

20

π c.mobile, c.first_name, c.last_name, cc.days, cc.purchased (σ cc.days_pass<14 AND cc.purchased = "not purchased" AND count(o.customer_id)>0 (orders o ⋈ o.customer_id = customers c.customer_id ⋈ contacts cc.order_id = o.order_id ⋈ o.order_id=book_list_for_orders bl.order_id ⋈ bl.book_id = books b.book_id));

21

π sum(set_amount_in_storage), year (purchase_date), MONTH(purchas_date) (orders_from_provider);

22;

π sum(amount)->total_amount, sum(purchas_price)->total_paid (σ purchas_date>=? AND purchas date < ? (orders_from_provider));

23

π (sum(price_for_order)-sum(purchas_price))->profit (σ m.month = ? AND m.year = ? (manager_control t ⋈t.managment_id = management m.management_id ⋈ t.order_provider_id = orders_from_provider p.order_provider_id ⋈ t.order_id = orders o.order_id));

24

π avg(price_for_order), year(o.order_date), month(o.order_date) (σ order_complited = 'yes' AND count(o.order_date)>0 (orders o));

25

π gross_salary (σ e.first_name = ? AND e.last_name = ? AND month(paycheck_date) = ? AND year (paycheck_date) = ? (employees_paycheck ep ⋈ ep.employee_id = employees e.employee_id)) ;

26

π count(e.phone) > 0, e.last_name, e.first_name (σ order_complited = 'yes' AND month(o.pyment_date) = ? AND year(o.pyment_date) = ? AND rownum() > 0 AND rownum() <=1 (orders o ⋈ o.employee_id = employees e.employee_id));