1) SELECT distinct(inventary_id), concept FROM inventary join book_details using(details_id) join book_author using(book_id) join book using(book_id) join author using(author_id) join stock using(stock_id); -- where title='X' or where first_name='X' and last_name='Y'	π distinct(inventory_id, concept (σ title='X' (first_name='X' and 'last_name='Y') (inventory)(inventory ⋈ book_details ⋈ book_author ⋈ author ⋈ stock)
2) Select customer_id,first_name,last_name from customer order by customerEntryDate limit 1;	Π customer_id,first_name,last_name(σ (customer))
3) Select title from inventary join book_details using(details_id) join book using(book_id) where stock_id!=3 order by purchase_date limit 1;	Π title(σ stock_id!=3(inventory) (inventory ⋈ book_details ⋈ book))
4) Select order_id,customer_id,order_date,details_id from orders join detail_order using(order_id) where orderBookStatus_id=1 order by order_date;	Π order_id, customer_id, order_date, details_id(σ orderBookStatus_id=1 (orders) (orders ⋈ detail_order))

5) Select count(*) from orders join detail_order using(order_id) join book_details using(details_id) where orderBookStatus_id=3 and details_id=111178;	Π count(*)(σ orderBookStatus_id=3 ∧ details_id='X' (orders) (orders ⋈ detail_order ⋈ Book_details))
6) SELECT first_name,last_name, COUNT(author_id) AS MaxAuthor FROM orders JOIN detail_order USING(order_id) JOIN book_details USING(details_id) JOIN book_author USING(book_id) JOIN author USING(author_id) WHERE orderBookStatus_id=3 AND order_date BETWEEN '2020-01-01' AND '2020-12-31' GROUP BY author_id ORDER BY MaxAuthor DESC LIMIT 1;	Π first_name, last_name, ρ(count(author_id)/MaxAuthor) (σ orderBookStatus_id=3 ∧ Order_date> 'X' ∧ order_date<'Y' (orders) (orders ⋈ detail_order ⋈ book_details ⋈ book_author ⋈ author))
7) SELECT customer_id,first_name,last_name, COUNT(customer_id) AS MAXCUSTOMER FROM orders JOIN detail_order USING(order_id) JOIN customer USING(customer_id) WHERE orderBookStatus_id=3 GROUP BY customer_id ORDER BY MAXCUSTOMER DESC LIMIT 3;	Π customer_id,first_name, last_name, ρ(count(customer_id)/MaxCustomer) (σ orderBookStatus_id=3 (orders) (orders ⋈ detail_order ⋈ customer))
8) SELECT title, COUNT(DISTINCT translator_id) NumOfTranslations FROM inventary JOIN book_details USING(details_id) join book using(book_id) WHERE stock_id!=3 AND translator_id IS NOT NULL GROUP BY book_id ORDER BY NumOfTranslations DESC LIMIT 1;	Π title, ρ(count(distinct(translator_id)))/NumOfTranslations) (σ stock_id!=3∧translator_id !=NULL (inventary) (inventary ⋈ book_details ⋈ book))

9) SELECT book_id,title takenBooks,order_date purchaseDate,price FROM orders JOIN detail_order USING(order_id) JOIN book_details USING(details_id) JOIN book_author USING(book_id) JOIN book USING(book_id) JOIN sale_price USING(details_id) WHERE customer_id=123456789 and orderBookStatus_id=3 ORDER BY order_date;	Π book_id, title,order_date (σ customer_id=’X’∧ orderBookStatus_id=3 (orders) (orders ⋈ detail_order ⋈ book_details ⋈ book_author ⋈ book ⋈ sale_price))
10. SELECT title,order_date,IF(orderBookStatus_id=1,'No','Yes') isInInventary,IF(orderBookStatus_id=3,'Yes','No') getTheBook FROM orders JOIN detail_order USING(order_id) JOIN book_details USING(details_id) JOIN book_author USING(book_id) JOIN book USING(book_id) WHERE customer_id=123456789 ORDER BY order_date;	Π title, order_date, IF(orderBookStatus_id=1 then No else yes), IF(orderBookStatus_id=3 then 'Yes' else 'No') (σ customer_id=’X’ (orders) (orders ⋈ detail_order ⋈ book_details ⋈ book_author ⋈ book))
11.	
12. select order_id,order_date,concept,title from delivery join detail_order using(detailOrder_id) join orders using(order_id) join delivery_types using(deliveryType_id) join book_details using(details_id) join book using(book_id) join customer using(customer_id) where order_id in(select order_id from delivery join detail_order using(detailOrder_id) join orders using(order_id) join customer using(customer_id) where customer_id=523456798 group by order_id having (count(distinct(deliveryType_id))>1) order by order_id);	Π order_id,order_date, concept, title (Π order_id σ customer_id=’X’ (delivery) (delivery ⋈ detail_order ⋈ orders ⋈ customer) having (count(distinct(deliveryType_id))>1) (σ order_id=’X’ (delivery) (delivery ⋈ orders ⋈ delivery_types ⋈ book_details ⋈ book ⋈ customer))

13. select deliverState from delivery join detail_order using(detailOrder_id) join orders using(order_id) join delivery_status using(deliveryStatus_id) join delivery_types using(deliveryType_id) where customer_id=223456789 and order_date='2016-03-20' and concept='Xp Sheliaj' group by deliveryType_id,order_id;	Π deliveryState_id(σ customer_id=’X’ ∧ order_date=’Y’ ∧ concept=’Z’ (delivery) (delivery ⋈ orders ⋈ delivery_status ⋈ delivery_types ⋈ customer_id))
14. Select COUNT(*) from delivery join delivery_types using(deliveryType_id) join detail_order using(detailOrder_id) join orders using(order_id) where concept like 'Xp%' and month(order_date)=4 and year(order_date)=2020;	Π count(*)(σ concept=’Xp%’ ∧ month(order_date)=’Y’ ∧ year(order_date)=’Z’ (delivery_types) (delivery_types ⋈ detail_order ⋈ orders))
15. Select SUM(price+weight*pricePerKg) totalBitEntry from payment join payment_method using(paymentMethod_id) join orders using(order_id) join detail_order using(order_id) join sale_price using(details_id) join delivery using(detailOrder_id) join delivery_types using(deliveryType_id) where payment_concept='bit' and month(order_date)=4 and year(order_date)=2020;	Π SUM(price+weight*pricePerKg),payment(σ payment_concept=’Bit’ ∧ month(order_date)=’Y’ ∧ year(order_date)=’Z’ (payment) (payment ⋈ payment_method ⋈ orders ⋈ detail_order ⋈ sale_price ⋈ delivery ⋈ delivery_types))
16.	

17. select COUNT(CASE WHEN concept LIKE 'Doar%' THEN 1 END) AS Doar, COUNT(CASE WHEN concept LIKE 'XP%' THEN 1 END) AS Xpress from delivery join delivery_types using(deliveryType_id) join detail_order using(detailOrder_id) join orders using(order_id) WHERE order_date>DATE_SUB(now(),INTERVAL 12 month);	Π COUNT(price+weight*pricePerKg), ρ((COUNT(CASE WHEN concept LIKE 'Doar%' THEN 1 END)/Doar), ρ((COUNT(CASE WHEN concept LIKE 'XP%' THEN 1 END)/Xpress), payment(σ order_date>Sub(now(),INTERVAL 12 month month(order_date)) (delivery) (delivery ⋈ delivery_types ⋈ detail_order ⋈ orders ⋈ delivery_types))
18. select delivery_id,deliveryStatus_id,order_id,inventary_id from delivery join detail_order using(detailOrder_id) join book_details using(details_id) join book using(book_id) where title in(select title from delivery join detail_order using(detailOrder_id) join book_details using(details_id) join book using(book_id) group by title having (count(distinct(details_id))>1) order by title);	Π delivery_id, deliveryStatus_id, order_id, inventary_id (Π title (σ (delivery) (delivery ⋈ detail_order ⋈ book_details ⋈ book) having (count(distinct(details_id))>1)) (σ title (delivery) (delivery ⋈ detail_order ⋈ book_details ⋈ book))
19. SELECT * FROM orders JOIN customer USING(customer_id) WHERE LAST_DAY(order_date)< DATE_SUB(now(),INTERVAL 24 month) GROUP BY customer_id ORDER BY customer_id;	σ LAST_DAY(order_date)< DATE_SUB(now(),INTERVAL 24 month) (orders) (orders ⋈ customer))
20. select customer_id,first_name,last_name from detail_order join orders using(order_id) join customer using(customer_id) join inventary using(inventary_id) where orderBookStatus_id=2 and (purchase_date)<DATE_SUB(now(),INTERVAL 14 day);	Π customer_id, first_name, last_name(σ orderBookStatus_id=2 ∧ (purchase_date)<DATE_SUB(now(),INTERVAL 14 day) (detail_order) (detail_order ⋈ orders ⋈ customer ⋈ inventary))
21. select month(purchase_date),year(purchase_date),count(*) from inventary where stock_id=2 group by year(purchase_date),month(purchase_date) order by year(purchase_date);	Π month(purchase_date),year(purchase_date),count(*) (σ stock_id=2 (detail_order))

22. SELECT COUNT(*) totalBooks,SUM(purchase_price) totalSum FROM inventary WHERE purchase_date BETWEEN '2020-01-01' AND now();	Π COUNT(*) totalBooks,SUM(purchase_price) (σ purchase_date> 'X' ∧ purchase_date< now() (inventary))
23.	
24.	
25. SELECT staff_id,salaryPerHour*hours salary FROM staff JOIN hours_staff USING(staff_id) WHERE staff_id=1 AND month_id=1;	Π staff_id,salaryPerHour*hours (σ staff_id='x' ∧ month_id='Y' (staff))
26. SELECT staff_id, COUNT(staff_id) AS bestSeller FROM orders WHERE Month(order_date)=7 GROUP BY staff_id ORDER BY bestSeller DESC LIMIT 1;	Π staff_id, ρ(COUNT(staff_id)/ bestseller) (σ Month(order_date)='x' (orders))