

מסמך 7 - שאילתות SQL ואלגברת יחסים:

-- 1. is book X exist in the inventory?

```
select b.*, a.first_name, a.last_name from books b
inner join inventory i
using(book_id)
inner join authors a
using(book_name)
where b.book_name = '1984';
```

$\pi_{books.*, author.first_name, author.last_name}(\sigma_{book_name = 'X'}(inventory \bowtie books \bowtie author));$

-- 2. who is the oldest-customer?

```
select bk.order_date, c.customer_id, c.first_name, c.last_name from customer c
inner join book_order bk
using(customer_id) order by order_date limit 1;
```

```
select d.deal_date, c.customer_id, c.first_name, c.last_name from customer c
inner join books_sale
using (customer_id)
inner join deals d
using (deal_id) order by deal_date limit 1;
```

$\tau_{order_date} \pi_{customer.*, order_date}(customer \bowtie book_order);$

$\tau_{deal_date} \pi_{customer.*, deal_date}(customer \bowtie book_sale \bowtie deals);$

-- 3. what is the oldest book in store or warehouse?

```
select b.*, date_stored from books b
inner join inventory
using(book_id) order by date_stored limit 1;
```

$\tau_{date_stored} \pi_{books.*, date_stored}(books \bowtie inventory);$

```
-- 4.show all book orders sorted by order_date
select first_name, last_name, book_name, order_date, informed_date from book_order
inner join books using(book_id)
inner join customer using(customer_id)
where isnull(informed_date) order by order_date;
```

```
 $\tau$  order_date  $\pi$  first_name, last_name, book_name, order_date, informed_date
 $\sigma$ (book_order  $\bowtie$  books  $\bowtie$  customer);
```

```
-- 5. How many copies of book name (some Y) sold by book store
select count(b.book_name) from books b
inner join books_sale bs
using (book_id)
group by b.book_name having book_name = "ok";
```

```
 $\pi$  COUNT(book_name)( $\sigma$  book_name = 'Y'(books  $\bowtie$  book_sale)) book_name;
```

```
-- 6. Most read author between X and Y dates
select count(a.first_name) most_read , a.first_name, a.last_name from books b
inner join books_sale
using(book_id)
inner join deals
using(deal_id)
inner join authors a
using(book_name)
where deal_date between '2018-01-01' and '2020-07-30'
group by a.first_name
order by most read desc limit 1;
```

```
 $\tau$  most_read desc  $\pi$  count(author.first_name) most_read, author.first_name, author.last_name ( $\sigma$  deal_date between
(date1, date2) (books  $\bowtie$  books_sale  $\bowtie$  deals  $\bowtie$  authors)) author.first_name;
```

```
-- 7. Who are the top three customer in book buying?
select c.first_name, c.last_name, count(customer_id) top_customer from customer c
inner join books_sale bs
using (customer_id)
group by customer_id order by top_customer desc limit 3;
```

```
 $\tau$  top_customer desc  $\pi$  customer.first_name, customer.last_name, cont(customer_id) top_customer
```

```
(customer ⋈ books_sale) customer_id;
```

```
-- 8. What is the most translated book in storage?
```

```
select t.book_name ,count(book_name) Translations from translator t
inner join books b
using(book_name)
inner join inventory
using (book_id)
group by t.first_name, t.last_name order by Translations desc limit 1;
```

```
τ translations desc π translator.book_name, count(book_name) translations (translator ⋈ books ⋈ inventory)
translator.first_name, translator.last_name;
```

```
-- 9. what is the purchase history of customer X
```

```
select c.first_name, c.last_name, b.book_name, d.deal_date, bs.book_price from books_sale bs
inner join customer c
using(customer_id)
inner join books b
using(book_id)
inner join deals d
using(deal_id)
where c.first_name = "itamar" and c.last_name = "yarden"
order by deal_date desc;
```

```
τ deal_date desc π customer.first_name, customer.last_name, books.book_name, deals.deal_date,
books_sale.book_price (σ customer.first_name = 'X' ∧ customer.last_name = 'Y' (customer ⋈ books
⋈ books_sale ⋈ deals));
```

```
-- 10. what is the order history of customer X // if book_price is not null then we'll know if the book is sold
```

```
select b.book_id ,b.book_name,bo.order_date, i.date_stored ,bs.book_price from customer c
inner join book_order bo
using(customer_id)
left join inventory i
using(book_id)
inner join books b
using(book_id)
left join books_sale bs
using(book_id)
where c.first_name = 'Barak' and c.last_name= 'Daniel'
order by order_date;
```

```
τ order_date π books.book_id, books.book_name, book_order.order_date, inventory.date_stored,
books_sale.book_price (σ customer.first_name = 'X' ∧ customer.last_name = 'Y' (book_order ⋈
```

customer ⋈ *inventory* ⋈ *books* ⋈ *books_sale*));

-- 11. What is the price of delivery number n?

```
select company, bs.delivery_id ,sum(b.weight) as total_weight, sp.pay_rate, sum(b.weight)*pay_rate as price
from books b
inner join books_sale bs
using(book_id)
inner join delivery d
using(delivery_id)
inner join shipping_rates sp
using(shipping_method)
where delivery_id = 420
group by delivery_id;
```

π company, books_sale.delivery_id, sum(books.weight) total_weight, shpping_rates.pay_rate
sum(books.weight)*pay_rate price (σ delivery_id = 'X' (books ⋈ books_sale ⋈ delivery ⋈
shipping_rates)) γ delivery_id;

-- 12. Is there a customer X that splited deliveries in the same deal?

```
select d.*, da.city, da.street from (select c.last_name,c.first_name,bs.deal_id, count(deal_id) multy_shipping from delivery
inner join books_sale bs
using(delivery_id)
inner join customer c
using(customer_id)
group by deal_id
having multy_shipping > 1 and c.last_name ="Buzaglo" and first_name = "Itamar" ) AS mid_tb
inner join books_sale bs
using(deal_id)
inner join delivery d
using(delivery_id)
inner join delivery_address da
using(delivery_id);
```

π delivery.*, delivery_address.city, delivery_address.street ((π customer.last_name, customer.first_name,
books_sale.deal_id, count(deal_id) multi_shipping (σ multi_shipping > 1 \wedge customer.last_name = 'X' \wedge
customer.first_name = 'Y') deal_id ⋈ books_sale ⋈ delivery ⋈ delivery_address);

-- 13. what is the current status of specific delivery

```
select shipping_status from delivery where delivery_id = 13;
```

π shipping_status (σ delivery_id = 'X' (delivery));


```
-- 14. what is the amount of payments made by specific shipping company at some month?
select sum(sr.pay_rate * b.weight) total_pay from books b
inner join books_sale bs
using(book_id)
inner join delivery d
using(delivery_id)
inner join shipping_rates sr
using(shipping_method)
where company = "Xpress" and month(delivery_date) = '07' and year(delivery_date) = '2020';
```

$\pi \text{ SUM}(\text{shipping_rates.pay_rate} * \text{books.weight}) \text{ total_pay } (\sigma \text{ company} = 'Xpress' \wedge \text{month}(\text{delivery_date}) = 'X' \wedge \text{year}(\text{deal_date}) = 'Y' (\text{books} \bowtie \text{books_sale} \bowtie \text{delivery} \bowtie \text{shipping_rates})))$;

```
-- 15. what is the total sum of money transfered to the store using 'bit' service in specific month?
select sum(total_pay) bit_pay from deals where payment_method = 'bit'
and month(deal_date) = 07 and year(deal_date) = 2020;
```

$\pi \text{ sum}(\text{total_pay}) \text{ bit_pay } (\sigma \text{ payment_method} = 'bit' \wedge \text{month}(\text{deal_date}) = 'X' \wedge \text{year}(\text{deal_date}) = 'Y' (\text{deals}))$;

```
-- 16. what are the deals that occurred during the past year that yield more than the average profit in the past year?
-- we will use these answer to calc the average and find the values that are bigger than it.
select d.deal_id, d.deal_date, sum(bs.book_price) Profit from deals d
inner join books_sale bs
using(deal_id)
inner join books b
using(book_id)
group by deal_id
having d.deal_date between DATE_SUB(current_date(), INTERVAL 12 MONTH) and current_date();
```

$\pi \text{ deals.deal_id, deals.deal_date, sum}(\text{books_sale.book_price}) \text{ profit } (\sigma \text{ deals.deal_date} \text{ between } ((\text{current_date}(), \text{interval}(12 \text{ months}), \text{current_date}())) (\text{deals} \bowtie \text{books_sale} \bowtie \text{books})))$;

```
-- 17. how many deliveries were made during the past year distribute to each company?
select count(company) distribution, company from delivery
inner join shipping_rates using(shipping_method) where delivery_date
between DATE_SUB(current_date(), INTERVAL 12 MONTH) and current_date()
group by(company);
```

$\pi \text{ count}(\text{company}) \text{ distribution, company } (\sigma \text{ delivery_date} \text{ between } ((\text{current_date}, \text{interval}(12 \text{ months}), \text{current_date}())) (\text{delivery} \bowtie \text{shipping rates})) \text{ shipping_company};$

```
-- 18. how many deliveries were made with more than two different types of the same book?
-- The table that returns as answer will provide us information about books in deliveries and will be solved in app.
select b.book_id, b.book_name, d.delivery_id, brand_name as publisher, publish_year, t.first_name, t.last_name
from books_sale bs
inner join delivery d
using(delivery_id)
inner join books b
using(book_id)
inner join publisher p
using(book_name)
left join translator t
using(book_name)
order by delivery_id, book_name;
```

τ delivery_id, book_name π books.book_id, books.book_name, delivery.delivery_id, brand_name publisher, publish_year, translator.first_name, translator.last_name (books_sale \bowtie delivery \bowtie books \bowtie publisher \bowtie translator);

```
-- 19. customers who purchased at least one copy in past but not in the last 24 months?
select c.* from customer c
inner join books_sale using(customer_id)
inner join deals d using(deal_id)
where c.customer_id not in (select c.customer_id from books_sale
inner join customer c using(customer_id)
inner join deals d using(deal_id)
where d.deal_date between DATE_SUB(current_date(), INTERVAL 2 YEAR) and current_date())
group by c.customer_id ;
```

π customer.* $((\pi$ customer.customer_id $(\sigma$ deals.deal_date between $((current_date(), interval(2\ year)), current_date()))$ customer $-$ (customer \bowtie books_sale \bowtie deals))) customer.customer_id;

```
-- 20. how many books were ordered and informed about arrival to the customer two weeks and were not bought?
select c.*, bo.book_id, b.book_name, bo.informed_date from book_order bo
inner join books b
using(book_id)
inner join inventory
using(book_id)
inner join customer c
using(customer_id)
where informed_date < DATE_SUB(current_date(), INTERVAL 14 DAY);
```

π customer.*, book_order.book_id, bookd.book_name, book_order.informed_date $(\sigma$ informed_date < $(current_date(), interval(14\ day))$ (book_order \bowtie books \bowtie inventory \bowtie customer));

```
-- 21. How many books are in the warehouse in specific month?
select * from warehouse
where (((year(date_in) = '2019' and month(date_in) <= '07' ) or year(date_in) < '2020')
and (((year(date_out) = '2020' and month(date_out) >= '07') or year(date_out) > '2020') or isnull(date_out))));
```

π customer.*, book_order.book_id, bookd.book_name, book_order.informed_date (σ informed_date < (current_date(), interval(14 day)) (book_order \bowtie books \bowtie inventory \bowtie customer));

```
-- 22. how many books did the store purchased between two dates and how much did it cost?
select count(*) book_purchase, sum(price) total_price from store_purchase
where purchase_date>='2019-01-01' and purchase_date<='2020-07-31';
```

π count(*) book_purchase, sum(price) total_price (σ purchase_date >= 'X' \wedge purchase_date <= 'Y');

```
-- 23.what is the retail in a specific month?
select sum(bs.book_price - sp.price) store_income, month(deal_date) Month, year(deal_date) Year from books_sale bs
inner join deals
using(deal_id)
inner join store_purchase sp
using(book_id)
where month(deal_date) = '06' and year(deal_date) = '2017';
```

π sum(books_sale.book_price - store_purchase.price) store_income, month(deal_date) month, year(deal_date) year (σ month(deal_date) = 'X' \wedge year(deal_date) = 'Y') (books_sale \bowtie deals \bowtie store_purchase));

```
-- 24. what is deals average in year due to monthly cut?
select count(*) number_of_deals_per_year, sum(total_pay)/(12) Avg_monthly_cut
from deals where year(deal_date) = '2016';
```

π count(*) number_of_deals_per_year, sum(total_pay)/12 avg_monthly_cut (σ year(deal_date) = 'X' (deals));

```
-- 25. what is the salary of Z in month Y?
select emp_id id, pay_date, total_hours, first_name, last_name, total_hours*30 salary from salaries
inner join employee using(emp_id)
where month(pay_date) = '07' and year(pay_date) = '2020'
and (emp_id = 'itzik' or first_name = 'itzik');
```

$\pi emp_id, pay_date, total_hours, first_name, last_name, total_hours * 30 salary (\sigma month(pay_date) = 'X' \wedge year(pay_date) = 'Y' \wedge (emp_id = 'Z' \vee first_name = 'W')) (salaries \bowtie employee);$

```
-- 26. who is highest selling employee?
select emp_id, first_name, last_name, count(emp_id) sales_number from employee
inner join sales
using(emp_id)
inner join deals
using(deal_id) where month(deal_date) = '07' and year(deal_date) = '2020'
group by emp_id limit 1;
```

$\pi emp_id, first_name, last_name, count(emp_id) sales_number (\sigma month(deal_date) = 'X' \wedge year(deal_date) = 'Y') (employee \bowtie sales \bowtie deals) emp_id;$