

# **Software Engineering Department**

מערכות קבצים ומסדי נתונים

**Lecturer: Itzhak Nudler**

## **Programming project**

פרויקט תכנות

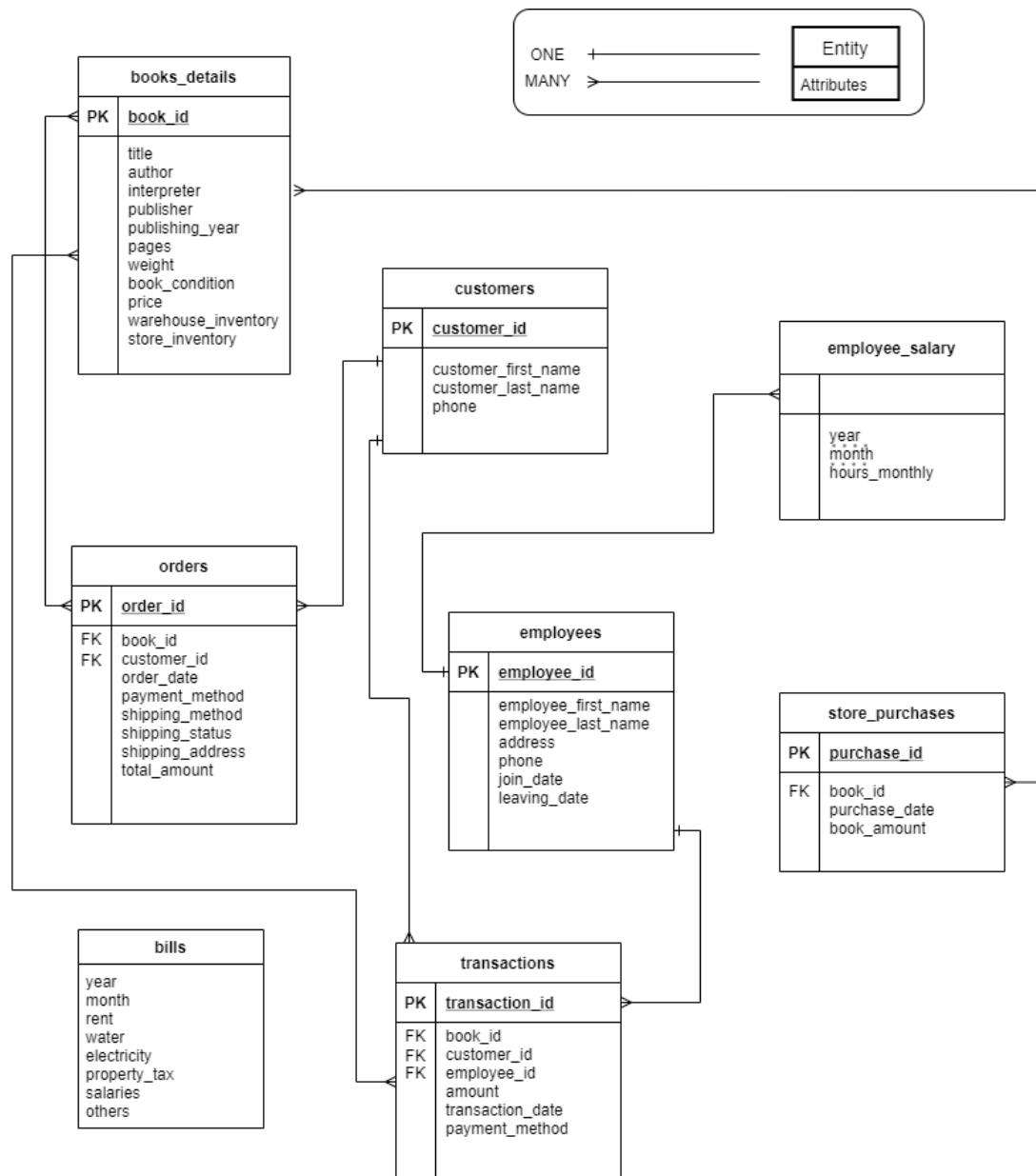
**Written by:**

**Ido Betesh – 307833822**

**Yahli Sofer – 205701212**

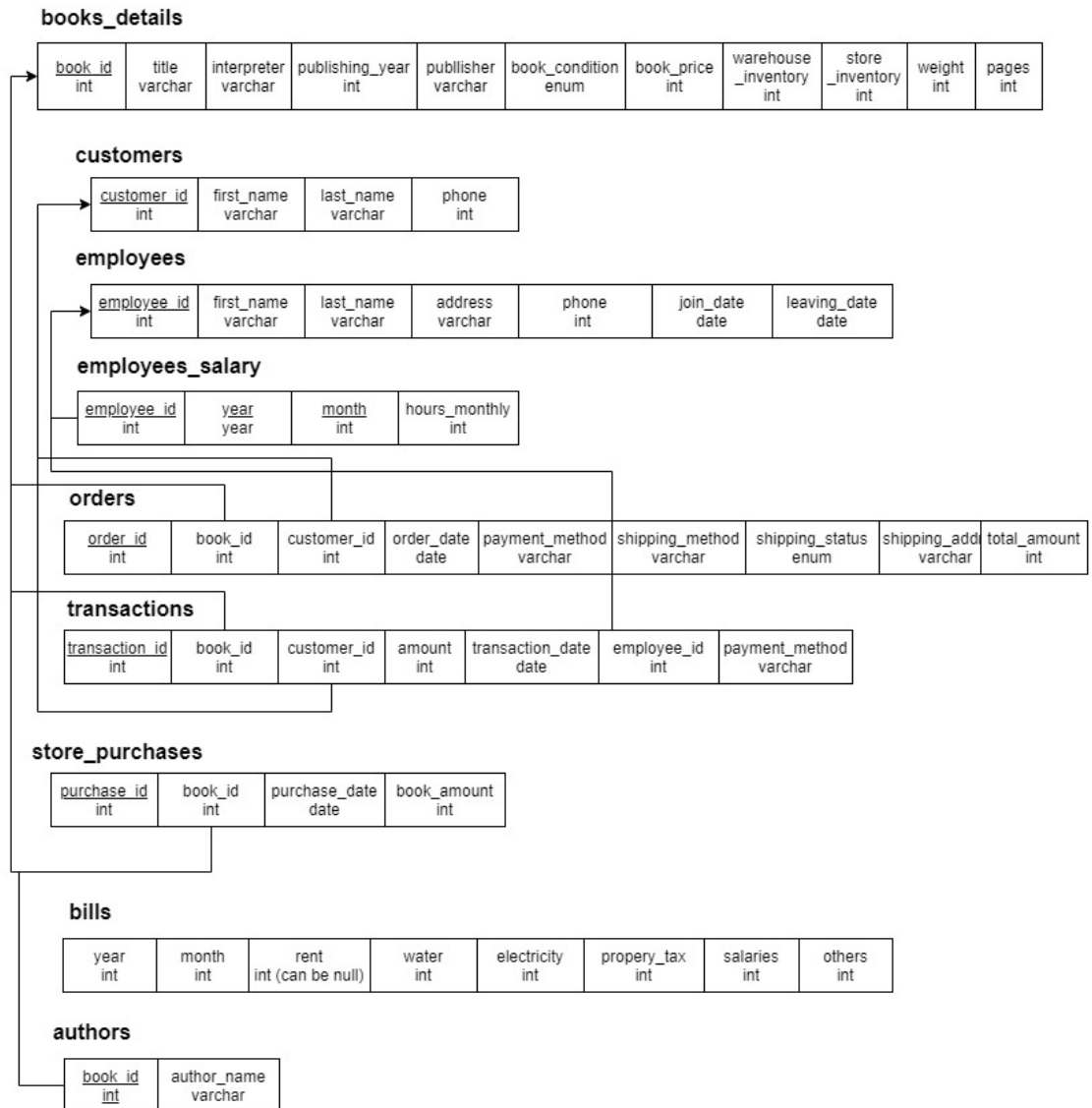
**05 August 2020**

## תרשים ERD ראשוני לפי תיאור העולם:



- קבוצת ישויות employee\_salary היא חלשה מכיוון שהמשכורת משויכת באופן חד ערכי לעובד מסוים אך רצינו להוריד כפילויות מידע מטבלת עובדים ולכן יצרנו טבלה נפרדת שמזוהה על ידי מספר עובד את המשכורות שלו ומכיוון שזוהי קבוצת ישויות חלשה יש לנו פה אילוץ יחס של אחד-לרבים. המפתחות החלקיים בטבלה הם month + year. בנוסף, עדכון משכורות העובדים תתבצע בצורה יעילה יותר מאחר ולא נצטרך לגשת לטבלה שמכילה את כל פרטי העובד אלא רק לטבלת המשכורות שמכילה רק מספר עובד, חודש, שנה ואת כמות השעות באותו חודש ושם נעדכן את כמות השעות.
- קבוצת ישויות employee\_salary היא קבוצת ישויות חלשה מכיוון שמשכורת עובד מזוהה לפי מיהו העובד ולכן היא 'תלויה' ב-employee\_id ובלי עובד לישות למשכורת אין זכות קיום.
- אילוץ יחס עובד-עסקה הוא אחד-לרבים מכיוון שלעובד אחד יכולות להיות מספר עסקאות שעליהן הוא חתום אך לא יכול להיות מספר עובדים שחתומים על אותה עסקה.
- כך גם אילוץ יחס לקוח-עסקה הוא יחס אחד לרבים, לקוח אחד יכול לבצע מספר עסקאות אך באותה עסקה יש רק לקוח אחד שביצע אותה.
- אילוץ יחס ספר-הזמנה הוא רבים-לרבים מכיוון שבהזמנה אחת יכולים להיות מספר ספרים וגם אותו ספר יכול להופיע במספר הזמנות.
- אילוץ יחס פרטי ספר-עסקה הוא יחס רבים-לרבים מכיוון שיכול להיות לנו מצב שבו יש מספר ספרים באותה עסקה והפוך, יכול להיות לנו מספר עסקאות על אותו מזהה של ספר.
- כך גם בצורה דומה יהיה לנו אילוץ יחס רבים-לרבים בין טבלאות פרטי ספר-רכישות חנות מכיוון שסביר שיהיה מצב שנבצע מספר רכישות חנות על אותו מזהה ספר והפוך.

## סכמה ראשונית



## הנחות שהנחנו לבניית ה-ERD לאחר נירמול:

בהינתן תיאור העולם (סעיף 2) במסמך 'פרויקט תכנות' וניתוח ההסבר ונרמול יצרנו את הטבלאות העיקריות הללו –

- Books\_details
- Authors
- Books\_prices
- Customers
- Employees\_details
- Employees\_salary
- Inventory
- Orders
- Order\_details
- Payment\_method
- Shipping\_types
- Store\_purchases
- Transactions
- Transaction\_details
- Bills

כל אחת מהטבלאות הנ"ל מכילות את התכונות הרלוונטיות לראייתנו בנוסף לדרישות המפורטות בסעיף 3.3 'דרישות מהאפליקציה'.

ההנחות שביצענו לצורך יצירת התרשים ERD היו:

- הגדרנו את שדה book\_condition כ-ENUM מכיוון שהוא מופיע רק בטבלה אחת (books\_details) כתכונה עם טווח ערכים ידוע מראש.
- בטבלת book\_details הגדרנו שדה book\_id PK ויצרנו טבלה חדשה book\_prices אשר בה יהיה פירוט המחיר של כל ספר גם בקנייתו על ידי החנות וגם מחיר מכירתו על ידי החנות, וכך book\_id הוא FK בטבלת book\_prices ונוצר קשר בין שתי הטבלאות, באופן זה קל יותר לעדכן מחירים מכיוון שנצטרך לגשת לטבלת המחירים ולא book\_details שמכילה את כל פרטי הספר.
- את טבלת orders פיצלנו ל-2 טבלאות: וזאת על מנת למנוע כפילויות מיותרות, יצרנו טבלת מקשרת order\_details שלוקחת את ה-PK של orders (order\_id) ואת ה-PK של book\_details (book\_id) ומשתמשת בהן בתור FK ליצירת טבלה מקשרת שמקטינה את כמות העמודות שהיו בטבלה orders ומכילה פרטי ספרים שונים באותה הזמנה.
- לקחנו מטבלת transactions את ה-PK שהוא transaction\_id ואת ה-PK book\_id ויצרנו את transaction\_details כטבלת עזר על מנת להכיל קניה של מספר ספרים שונים.
- לקחנו את ה-PK של book\_details ויצרנו טבלה חדשה inventory שמציגה את כמות הספרים שקיימים גם בחנות וגם בטבלה לפי ספר כך שה-FK הוא book\_id וכך יצרנו קשר בין הטבלאות.
- לקחנו מטבלת employees\_details את ה-PK employee\_id ויצאנו טבלה חדשה employee\_salary ששומרת את המידע על שעות העבודה של עובד ומקשרת כ-FK לטבלת מידע על עובדים.

- על מנת להפריד כפילויות ותלויות יצרנו 2 טבלאות חדשות shipping\_types, payment\_method ולכל אחת מהן PK משלה payment\_method\_id -I shipping\_type\_id כך שבעת טבלת orders מושכת מהן מידע על ידי FK אצלה.
- שדה המתרגם בטבלת 'פרטי ספר' עלול להיות NULL משום שלא בכל ספר קיים מתרגם.
- מכיוון שיש לנו מצב של כמה סופרים לאותו הספר, לקחנו את ה-PK של book\_details ויצרנו טבלה חדשה authors שבה נשמור את שמות הסופרים לכל ספר.
- יצאנו מנקודת הנחה שאין מצב שבוא ללקוח/ עובד אין מספר טלפון, אם אין לו מספר טלפון הוא לא יכול לקנות/להזמין/לעבוד בחנות.
- יצרנו טבלת inventory שלה הגדרנו FK book\_id מאחר ואלה שני השדות שיתעדכנו תכופות בטבלת books\_details החלטנו להוציא את שדות warehouse\_inventory -I store\_inventory לטבלה נפרדת.

## רשימת ת"פ לפי תיאור עולם המתוארות במסמך העבודה הקרוי 'עבודת תכנות' –

### **books\_details:**

Fc = {book\_id -> title, book price, author, publish\_year, book\_condition, interpreter, publisher, weight, pages, store\_inventory, warehouse\_inventory}

{book\_id}+ = {book\_id, title, book price, author, publish\_year, book\_condition, interpreter, publisher, weight, pages, store\_inventory, warehouse\_inventory}

\*בהינתן book\_id אם קיים מתרגם לאותו הספר, בהכרח נגיע אליו. ואם לא שדה זה ישאר ריק (NULL).

### **customers:**

FC = { customer\_id -> first\_name, last\_name, phone}

{customer\_id} + = { customer\_id, first\_name, last\_name, phone}

### **Employee:**

Fc = {Employee\_id -> first\_name, last\_name, address, phone, join\_date, leaving\_date}

### **Employees\_salary:**

Fc = {Employee\_id, year, month -> hours\_monthly}

### **Orders:**

Fc = {Order\_id -> book\_id, customer\_id, order\_date, payment\_method, shipping\_method, shipping\_address, total\_amount}

### **Transactions:**

Fc = {Transactions\_id -> costumer\_id, amount, transaction\_date, employee\_id, payment\_method}

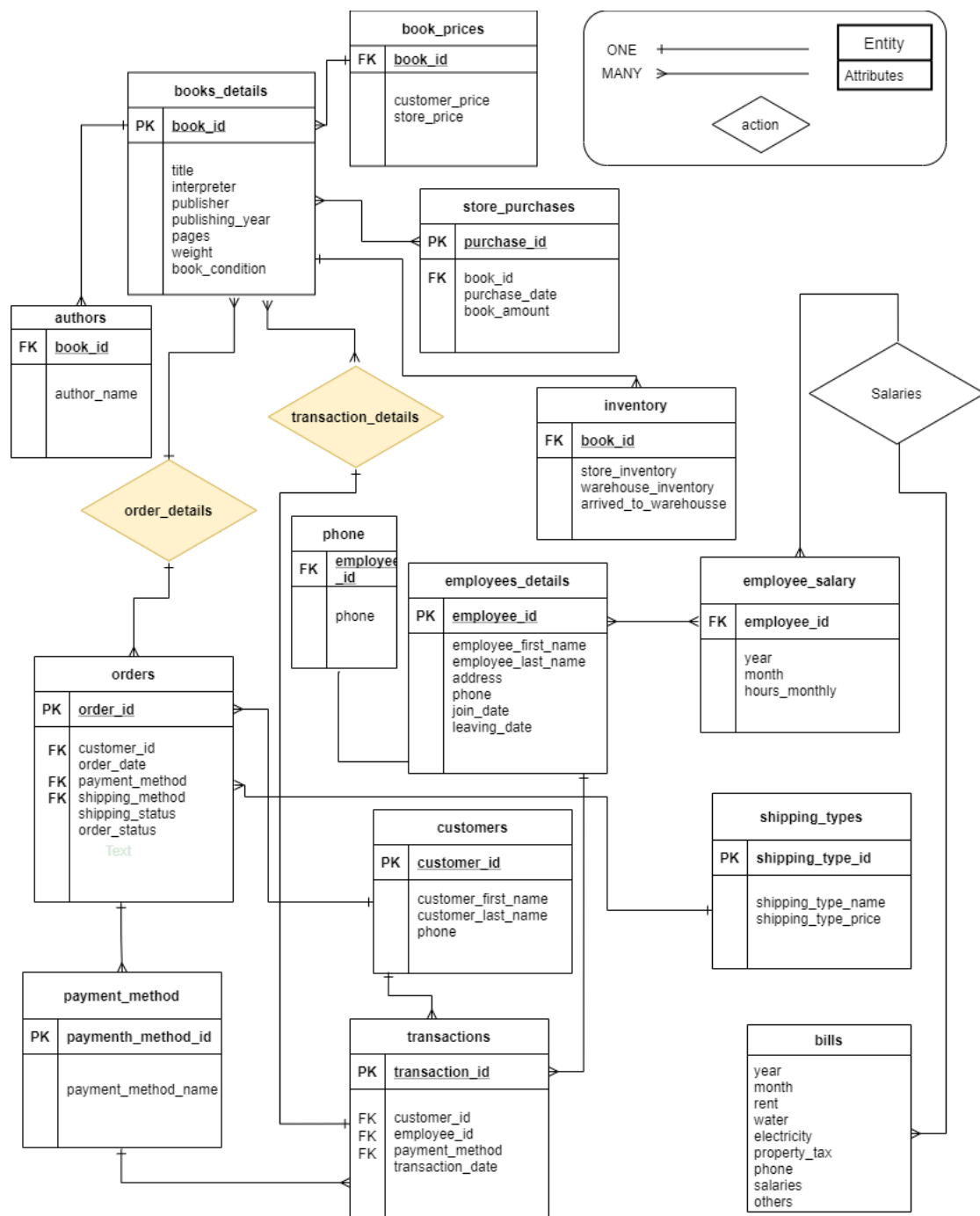
### **Store\_purchases:**

Fc = {Purchase\_id -> book\_id, purchase\_date, book\_amount}

### **Bills:**

Fc = {Year, month -> rent, water, electricity, property\_tax, salaries, others}

## ERD לאחר 3NF:



## מה מכילות טבלאות הActions –

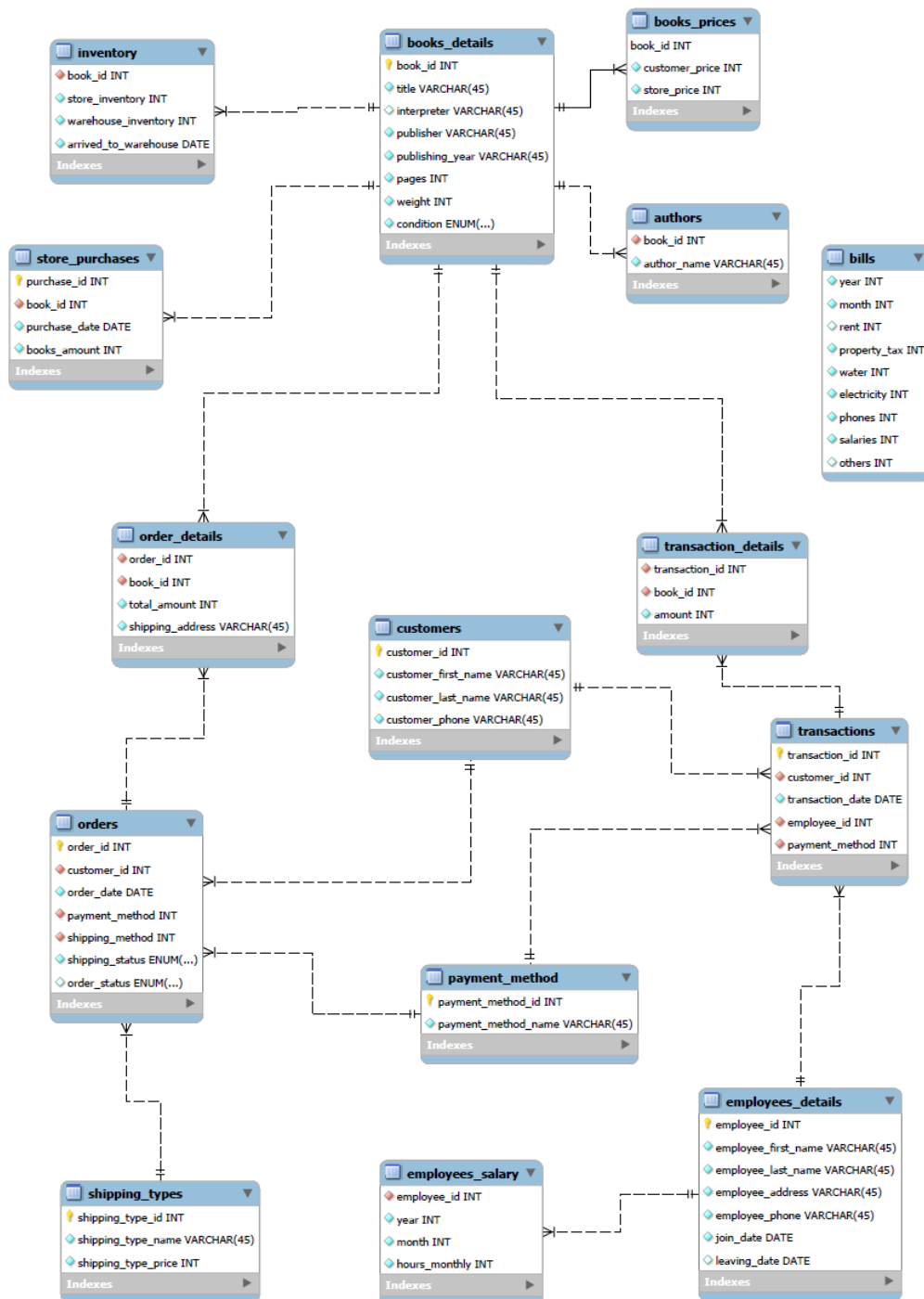
order_details	
FK	order_id
FK	book_id total_amount shipping_address

transaction_details	
FK	Transaction_id
FK	book_id amount

salaries	
	קישור קונספטואלי
	מצביע על הקשר שבין המשכורות בתשלומי החנות ומשכורות העובדים



## EERD לאחר נרמול ל-3NF (כולל טיפוסים המשתנים):



## תוכן כל הטבלאות:

### :Books\_details

book_id	title	interpreter	publisher	publishing_year	pages	weight	condition
1	The Doom	NULL	Sterling	2008	120	2	good
2	In Search of Lost Time	NULL	Penguin Random House	1913	300	2	new
3	Ulysses	arik levy	HarperCollins	1904	220	1	as new
4	Don Quixote	NULL	Penguin Random House	2000	80	1	good
5	The Great Gatsby	shlomo zeev	Simon & Schuster	1980	600	6	fair
6	One Hundred Years of Solitude	NULL	Hachette Book Group	1927	352	3	poor
7	Moby Dick	NULL	Workman	1950	124	2	as new
8	In Search of Lost Time	NULL	Penguin Random House	1913	300	2	new
9	War and Peace	fred boyer	Penguin Random House	1923	726	5	as new
10	Lolita	tom aharon	HarperCollins	1977	427	4	new
11	Hamlet	yoni rom	Hachette Book Group	1599	144	2	fair
12	The Odyssey	NULL	Kensington	1883	219	3	poor
13	Crime and Punishment	hen manor	Sterling	1492	485	5	good
14	Pride and Prejudice	lior dor	Simon & Schuster	2004	828	8	fair
15	Catch-22	NULL	Penguin Random House	1943	444	4	new
16	The Stranger	dor ariel	HarperCollins	1946	70	2	new
17	The Great Gatsby	zohar yefet	Sterling	1980	600	6	new
18	In Search of Lost Time	guy orpaz	Workman	1913	300	2	poor

### :inventory

book_id	store_inventory	warehouse_inventory	arrived_to_warehouse
2	12	8	2019-02-09
3	30	40	2019-03-04
4	90	20	2019-04-02
5	10	0	2019-05-18
6	4	2	2019-06-30
7	7	7	2019-07-12
8	24	12	2019-08-24
9	10	11	2019-09-13
10	1	23	2019-10-17
11	20	200	2019-11-28
12	22	190	2019-12-29
13	40	29	2019-03-14
14	5	78	2019-06-27
15	18	50	2019-03-05
16	12	100	2019-06-05
1	10	2	2019-01-05
17	20	11	2019-08-09
18	31	80	2019-12-07

### Book\_prices:

book_id	customer_price	store_price
1	89	70
2	80	65
3	120	90
4	69	47
5	99	70
6	110	90
7	85	70
8	79	40
9	99	64
10	29	10
11	49	20
12	99	39
13	102	73
14	109	92
15	67	30
16	129	100
17	119	80
18	100	65

## :Orders

order_id	customer_id	order_date	payment_method	shipping_method	shipping_status	order_status
1	1	2020-03-05	2	4	Collected	NULL
2	1	2020-07-25	1	1	Preparing for shipping	NULL
3	2	2019-07-21	2	2	Collected	NULL
4	3	2020-07-05	3	3	Arrived and wait for pick up	NULL
5	4	2020-07-19	3	3	Arrived and wait for pick up	NULL
6	5	2020-06-25	1	2	Sent	NULL
7	5	2019-12-25	1	4	Collected	NULL
8	6	2020-05-05	3	5	Sent	NULL
9	7	2020-07-27	3	1	Preparing for shipping	NULL
10	8	2020-06-21	2	5	Sent	NULL
11	8	2020-06-21	1	4	Sent	NULL
12	9	2019-02-10	1	3	Collected	NULL
13	9	2019-02-10	2	3	Collected	NULL
14	1	2020-07-01	1	4	Pending	Contacted client 14 days ago or more
15	2	2020-06-14	2	1	Pending	Contacted client 14 days ago or more
16	5	2019-01-09	2	3	Collected	NULL
17	8	2019-03-20	3	2	Collected	NULL
18	3	2019-04-17	2	4	Collected	NULL
19	11	2019-05-13	2	1	Collected	NULL
20	5	2019-06-14	1	2	Collected	NULL
21	8	2019-08-19	2	3	Collected	NULL
22	10	2019-09-11	3	4	Collected	NULL
23	4	2019-10-10	2	3	Collected	NULL
24	7	2019-11-08	3	1	Collected	NULL

## :order\_details

order_id	book_id	total_amount	shipping_address
1	3	4	Yael 7 Tel Aviv
2	4	2	Katz 37 Ramat Gan
3	1	1	Noya 63 Kfat Sabba
4	6	1	Yarkon 10 Tel Afek
5	8	7	Hagefen 137 Raanna
6	3	8	Pinchas 19 Tel Aviv
7	9	2	Gur 64 Nahariyya
8	5	1	Tufin 29 Beer Sheva
9	11	5	Frug 26 Tel Aviv
10	12	2	Begin 5 Bat Yam
11	13	1	Halutzim 13 Bnei Brak
12	2	2	Doron 22 Reut
13	18	1	Amnon 122 Maccabim
15	4	1	Drorit 231 Shlomi
16	4	1	Fogel 12 Arad
17	4	1	Haon 18 Dimona
18	7	1	Alonim 56 Jaffa
19	8	1	Knaan 132 Tel Aviv
20	2	1	Farid 121 Ramat Gan
21	9	1	Harif 12 Eilat
22	10	1	HaOren 221 Raanana
23	12	1	Poalim 90 Beer Sheva
24	14	1	Levanon 12 Yerucham
16	3	1	Fogel 12 Arad
14	3	1	Arizona 17 Bnei Brak
14	4	1	Arizona 17 Bnei Brak
22	5	1	HaOren 221 Raanana
12	5	1	Doron 22 Reut
13	2	1	Amnon 122 Maccabim

## :Shipping\_types

shipping_type_id	shipping_type_name	shipping_type_price
1	Israel Post registered	30
2	Israel Post courier	40
3	Israel Post fast courier	50
4	Xpress	10
5	Xpress courier	20

## :transactions

transaction_id	customer_id	transaction_date	employee_id	payment_method
1	1	2018-06-28	11	1
2	2	2020-05-28	4	1
3	2	2020-05-28	4	2
4	3	2014-01-01	3	3
5	4	2020-05-30	5	1
6	5	2020-02-22	5	3
7	6	2020-04-28	5	1
8	7	2011-11-01	8	2
9	8	2019-12-10	7	3
10	9	2018-11-28	7	1
11	10	2019-06-28	10	1
12	11	2019-03-21	10	3
13	7	2018-06-28	9	2
14	12	2019-12-17	5	2
15	10	2019-10-18	5	3

## :Payment\_methods

payment_method_id	payment_method_name
1	bit
2	bank tranfer
3	credit card

## :store\_purchases

purchase_id	book_id	purchase_date	books_amount
1	2	2019-01-07	30
2	12	2019-01-07	5
3	10	2020-02-20	25
4	1	2019-10-21	10
5	6	2020-05-10	5
6	4	2018-12-29	2
7	11	2020-07-17	7
8	12	2020-01-30	1
9	14	2019-11-11	9
10	9	2020-05-24	11

## :Transaction\_details

transaction_id	book_id	amount
1	3	2
2	4	2
3	6	1
4	11	2
5	12	1
6	13	2
7	3	1
8	8	3
9	1	1
10	14	2
11	15	3
12	1	2
13	4	2
14	7	1
15	10	1

## :Bills

year	month	rent	property_tax	water	electricity	phones	salaries	others
2019	1	NULL	750	69	450	60	25000	100
2019	2	3000	750	100	430	70	25000	800
2019	3	NULL	750	89	400	60	25000	500
2019	4	3000	750	49	450	100	25000	200
2019	5	NULL	750	169	600	68	25000	100
2019	6	3000	750	99	350	48	25000	720
2019	7	NULL	750	89	450	80	25000	900
2019	8	3000	750	99	550	140	25000	500
2019	9	NULL	750	99	450	100	25000	400
2019	10	3000	750	69	450	220	25000	140
2019	11	NULL	750	99	350	162	25000	1000
2019	12	3000	750	69	450	111	25000	1200
2020	1	NULL	750	99	450	60	25000	100
2020	2	3000	750	140	430	70	25000	880
2020	3	NULL	750	89	400	60	25000	1500
2020	4	3000	750	49	440	100	25000	1250
2020	5	NULL	750	199	350	68	25000	700
2020	6	3000	750	99	350	98	25000	720

## :Customers

customer_id	customer_first_name	customer_last_name	customer_phone
1	Roy	Dor	0503387291
2	Shay	Loren	0523997291
3	Roni	David	0503915231
4	Doron	Mor	0541928374
5	Harel	Toms	0527768122
6	Arik	Dadon	0523344789
7	Oren	Cohen	0546633916
8	Idan	Gur	0534490872
9	Nir	Tal	0509864112
10	Yoni	Ztur	0523987541
11	Gabi	fein	0529183659
12	Reut	Lev	0545411098

## :Authors

book_id	author_name
1	Shei Eyla
2	Marcel Proust
3	James Joyce
4	Miguel de Cervantes
5	Scott Fitzgerald
6	Gabriel Garcia Marquez
7	Herman Melville
8	Leo Tolstoy
9	Marcel Proust
10	Vladimir Nabokov
11	William Shakespeare
12	Homer
13	Fyodor Dostoyevsky
14	Jane Austen
15	Joseph Heller
16	The Stranger
17	Scott Fitzgerald
18	Marcel Proust
9	Yahli HaSofer

## :Employees\_details

employee_id	employee_first_name	employee_last_name	employee_address	employee_phone	join_date	leaving_date
1	Ravit	Shir	Hapaamon 5 Haifa	0529918811	2006-03-20	2010-01-11
2	Miki	Even	Yarden 15 Tel Aviv	0529333331	2002-04-10	2008-02-14
3	Mirna	Evron	Dganit 23 Raanana	0532953868	2012-11-20	2014-01-22
4	Amit	David	Lenon 12 Haifa	0507716842	2014-12-20	NULL
5	Shaul	Snir	Chen 8 Dimona	0529714523	2004-03-20	NULL
6	Roy	Heler	Sofit 29 Yerucham	0521298310	2018-01-20	2020-01-14
7	Shir	Hoff	Danit 77 Arad	0541210926	2006-09-29	NULL
8	Or	Nisan	Yam 21 Haifa	0548915263	2006-03-20	2011-11-11
9	Idan	Gefen	Gilad 31 Ramat Gan	0541345363	2010-06-20	NULL
10	Natan	Nave	Dovnov 122 Tel Aviv	0522290765	2017-03-20	2019-01-18
11	Sagi	Levi	Noach 51 Rehovot	0547729184	2001-03-20	2006-03-19



## :Employees\_salary

9	2019	10	50
9	2019	11	30
9	2019	12	70
9	2020	1	90
9	2020	2	70
9	2020	3	90
9	2020	4	80
9	2020	5	50
9	2020	6	90
1	2010	1	80
1	2009	12	50
1	2009	11	30
2	2008	2	30
2	2008	1	70
2	2007	12	90
3	2014	1	30
3	2013	12	40
3	2013	11	50
6	2014	1	50
6	2013	12	50
6	2013	11	50
8	2011	11	30
8	2011	10	20
8	2011	9	30
11	2019	1	30
11	2018	12	10
11	2018	11	30

employee_id	year	month	hours_monthly
4	2019	10	60
4	2019	11	30
4	2019	12	40
4	2020	1	80
4	2020	2	70
4	2020	3	50
4	2020	4	60
4	2020	5	40
4	2020	6	90
5	2019	10	40
5	2019	11	30
5	2019	12	40
5	2020	1	70
5	2020	2	20
5	2020	3	50
5	2020	4	30
5	2020	5	40
5	2020	6	80
7	2019	10	80
7	2019	11	40
7	2019	12	20
7	2020	1	90
7	2020	2	60
7	2020	3	40
7	2020	4	30
7	2020	5	60
7	2020	6	70

הנחות לשאלות:

שאלתה	הנחה
6.מי הסופר הכי נקרא בטווח תאריכים X עד Y.	הספרים של איזה סופר נרכשו הכי הרבה בטווח התאריכים הנתונים.
11. חישוב עלות משלוח.	בהינתן מספר order_id התוכנית תחזיר את עלות המשלוח תוך התחשבות בעלות שיטת המשלוח, וכן בעלות כללית של 3 שקלים לכל קילו.
12. האם לקוח X פיצל אי פעם רכישת ספרים למספר משלוחים, ואם כן מהם נתוני המשלוחים?	נזהה פיצול רכישה בכך שתאריך ההזמנה ומספר הלקוח זהים.
13. מהו הסטטוס הנוכחי של משלוח מסוים?	נזהה משלוח על ידי מספר זהות (order_id).
17. כמה משלוחים בוצעו במהלך 12 החודשים האחרונים באמצעות דואר ישראל וכמה באמצעות חברת express?	השאלתה תחזיר את כמות המשלוחים שנשלחו באמצעות דואר ישראל(שלוש אפשרויות) וכן משלוחים שנשלחו באמצעות חברת express(שתי אפשרויות)
25. המשכורת ברוטו של עובד Z בחודש X.	כל עובד מקבל שכר שעתי של \$35.



## שאלות ב-SQL:

```
-- 1
select (warehouse_inventory + store_inventory) as in_stock from books_details bd
left join inventory i on i.book_id = bd.book_id
where bd.title = "The Doom";

-- 2
select customer_first_name, customer_last_name, MIN(transaction_date) as 'date' from transactions t
left join transaction_details td on td.transaction_id = t.transaction_id
left join customers c on c.customer_id = t.customer_id
left join books_details bd on bd.book_id = td.book_id;

-- 3
select title from
(
  (select transaction_date, bd.book_id, title from
    (select transaction_date, td.book_id from transactions t
      left join transaction_details td on td.transaction_id = t.transaction_id
      group by transaction_date order by transaction_date desc) PURCHASE
    left join books_details bd on bd.book_id = PURCHASE.book_id
  group by bd.book_id) as allbks order by transaction_date asc limit 1;

-- 4
select order_id, customer_id, order_date, payment_method_name, shipping_type_name from orders o
join payment_method pm on pm.payment_method_id = o.payment_method
join shipping_types st on st.shipping_type_id = o.shipping_method
where shipping_status != 'Collected'
order by order_date;

-- 5
select sum(amount) as num_of_copys from transactions t
right join transaction_details td on td.transaction_id = t.transaction_id
right join books_details bd on bd.book_id = td.book_id
where title = "Don Quixote";

-- 6
select author_name from
(select td.transaction_id, book_id, amount from transactions t
left join transaction_details td on td.transaction_id = t.transaction_id
where transaction_date between '2012-11-01' and '2020-01-01') times_of_transactions
left join books_details on books_details.book_id = times_of_transactions.book_id
left join authors a on a.book_id = books_details.book_id
group by author_name order by amount desc limit 1;

-- 7
select customer_first_name, customer_last_name, SUM(amount) as total, customers.customer_id from
(select transaction_details.transaction_id, book_id, amount, customer_id from transactions t
left join transaction_details on transaction_details.transaction_id = t.transaction_id) as sold_bet
left join customers on customers.customer_id = sold_bet.customer_id
group by customer_id order by total desc limit 3;
```

```

-- 8
select bd.title from books_details bd
join books_details s_bd on bd.title = s_bd.title
where bd.interpreter != s_bd.interpreter
group by title order by count(bd.interpreter) desc LIMIT 1;

-- 9
select concat(customer_first_name, ' ', customer_last_name) as full_name, transaction_date, title, amount,
customer_price, (customer_price * amount) as total
from transactions t
join customers c on t.customer_id = c.customer_id
join transaction_details td on t.transaction_id = td.transaction_id
join books_details bd on bd.book_id = td.book_id
join books_prices bp on bp.book_id = td.book_id
where customer_first_name = 'Shay' and customer_last_name = 'Loren' order by transaction_date;

-- 10
select concat(customer_first_name, ' ', customer_last_name) as full_name, order_date, title,
total_amount, customer_price, (total_amount*customer_price) as total
from orders o
join customers c on o.customer_id = c.customer_id
join order_details od on o.order_id = od.order_id
join books_details bd on bd.book_id = od.book_id
join books_prices bp on bp.book_id = od.book_id
where customer_first_name = 'Harel' and customer_last_name = 'Toms' order by order_date;

-- 11
select sum(total_amount*weight) as total_weight , shipping_type_price, shipping_type_name,
( sum(total_amount*weight) * 3 + shipping_type_price) as total_price, sum(total_amount*weight) * 3
from books_details bd
join order_details od on od.book_id = bd.book_id
join orders o on o.order_id = od.order_id
join shipping_types st on st.shipping_type_id = o.shipping_method
where o.order_id = 12;

-- 12
select concat(customer_first_name, ' ', customer_last_name) as full_name, o.order_id, o.order_date,
title, total_amount, o.shipping_status, payment_method_name, shipping_type_name
from orders a, orders o
join customers c on o.customer_id = c.customer_id
join order_details od on o.order_id = od.order_id
join shipping_types st on o.shipping_method = st.shipping_type_id
join second_hand_books.payment_method pm on o.payment_method = pm.payment_method_id
join books_details bd on bd.book_id = od.book_id
where customer_first_name = 'Idan' and customer_last_name = 'Gur' and o.order_date = a.order_date and o.order_id != a.order_id;

-- 13
select o.order_id, concat(customer_first_name, ' ', customer_last_name) as full_name,
o.order_date, shipping_type_name, shipping_status
from orders o
join customers c on o.customer_id = c.customer_id
join order_details od on o.order_id = od.order_id
join shipping_types st on o.shipping_method = st.shipping_type_id
where o.order_id = 10;

-- 14
select count(order_id) as total_shipping_by_Xpress from orders o
where (shipping_method between 4 and 5) and (month(order_date) = 8);

-- 15
select transaction_date, t.transaction_id , sum(customer_price * amount) as total_by_bit from transaction_details td
join books_prices bp on td.book_id = bp.book_id
join transactions t on t.transaction_id = td.transaction_id
where (payment_method = 1) and (month(transaction_date) = 6);

```

```

-- 16
select td.transaction_id, transaction_date, (amount * customer_price) as income from transaction_details td
join books_prices bp on td.book_id = bp.book_id
join transactions t on t.transaction_id = td.transaction_id
where (amount * customer_price) > (
    select avg(amount * customer_price) as _avg from transaction_details td
    join books_prices bp on td.book_id = bp.book_id
    join transactions t on t.transaction_id = td.transaction_id
    where transaction_date between date_sub(now(), interval 12 month) and now())
and (transaction_date between date_sub(now(), interval 12 month) and now());

-- 17
select sum(shipping_method between 4 and 5) as total_orders_Xpress,
       sum(shipping_method between 1 and 3) as total_orders_IRpost from orders
where (order_date between date_sub(now(), interval 12 month) and now());
-- shipping_method between 4 and 5 -> Xpress
-- shipping_method between 1 and 3 -> Israel Post

-- 18
select distinct bd.publisher, oo.order_id, bd.title, o.order_date, o.customer_id
from books_details bd
right join order_details od on od.book_id = bd.book_id
right join orders o on o.order_id = od.order_id
right join orders oo on oo.order_date = o.order_date and (oo.order_id = o.order_id)
-- right join customers c on c.customer_id = oo.order_date
right join books_details bdd on bdd.publisher != bd.publisher and bd.book_id != bdd.book_id and bd.publishing_year = bdd.publishing_year
where bd.title = bdd.title and o.order_id = od.order_id and o.order_id = oo.order_id and bdd.publisher != bd.publisher
and o.order_id = 13 order by o.order_id;

-- 19
select c.customer_id, concat(customer_first_name, ' ', customer_last_name) as full_name, transaction_date from customers c
join transactions t on t.customer_id = c.customer_id
where transaction_date not between date_sub(now(), interval 24 month) and now()
group by customer_id;

-- 20
select concat(customer_first_name, ' ', customer_last_name) as full_name, order_date from customers c
join orders o on o.customer_id = c.customer_id
where shipping_status = 'Pendding' and order_status = 'Contacted client 14 days ago or more'
order by order_date;

-- 21
select (sum(warehouse_inventory) - books_sold) as in_stock from inventory i -- will run in loop for 12 times
join (select sum(total_amount) as books_sold from order_details od
left join orders o on od.order_id = o.order_id
where month(order_date) = 12 and year(order_date) = 2019) as p;

-- 22
select sum(books_amount) as books_amount, sum(books_amount * store_price) as total_spend
from store_purchases sp -- total books the store purchased in given date range
join books_prices bp on bp.book_id = sp.book_id
join books_details bd on bd.book_id = sp.book_id
where purchase_date between '2019-01-01' and '2019-12-30';

-- 23
select (sum(amount * customer_price) - (select sum(books_amount * store_price) as total_spend
from store_purchases sp -- total books the store purchased in given date range
join books_prices bp on bp.book_id = sp.book_id
join books_details bd on bd.book_id = sp.book_id
where purchase_date between '2020-05-01' and '2020-05-30')) as profit
from transaction_details td -- total books sold in given date range
join books_prices bp on bp.book_id = td.book_id
join books_details bd on bd.book_id = td.book_id
join transactions t on t.transaction_id = td.transaction_id
where transaction_date between '2020-05-01' and '2020-05-30';

```

```
-- 24
select month(transaction_date) as _month, avg(amount*customer_price) from transaction_details td
join transactions t on td.transaction_id = t.transaction_id
join books_prices bp on td.book_id = bp.book_id
group by _month order by _month;

-- 25
select concat(employee_first_name, ' ',employee_last_name) as full_name,
hours_monthly, es.month, es.year, (hours_monthly * 35) as salary
from employees_salary es
join employees_details ed on ed.employee_id = es.employee_id
where employee_first_name = 'Amit' and employee_last_name = 'David' and month = 7 and year = 2020;

-- 26
select e.employee_id, employee_first_name, employee_last_name, count(transaction_id) as _sum from employees_details e
join transactions t on t.employee_id = e.employee_id
where month(transaction_date) = 2 -- from user
group by(e.employee_id) order by _sum desc limit 1;
```

## אלגברת יחסים:

1.  $\pi$  (warehouse\_inventory + store\_inventory)  $\rho$  in\_stock  
(Books\_details  $\bowtie$  inventory  $\sigma$ (title = ulysses))
2.  $\pi$  (customer\_first\_name,  
customer\_last\_name, MIN(transaction\_date)  $\rho$  date (transations  $\bowtie$   
transaction\_details  $\bowtie$  customers  $\bowtie$  books\_details))
3.  $\pi$  title (  $\sigma$  transaction\_date, bd.book\_id, title ( $\sigma$  transaction\_date,  
td.book\_id )(transactions  $\bowtie$  transaction\_details) )
4.  $\sigma$  \* (orders) , [where shipping\_status != 'Collected']
5.  $\pi$  sum(amount) (transaction\_details  $\bowtie$  books\_details)
6.  $\pi$  author\_name ( $\sigma$ (transaction\_details td  $\bowtie$  books\_details bd  $\bowtie$   
authors a) bd.book\_id = a.book\_id)
7.  $\pi$  customer\_first\_name, customer\_last\_name, sum(amount),  
customer\_id ( $\sigma$ (transactions t  $\bowtie$  transactions\_details td  $\bowtie$  customers  
c) c.customer\_id = t.customer\_id AND t.transaction\_id =  
td.transaction\_id )
8.  $\pi$  title ( $\sigma$ (books\_details bd  $\bowtie$  books\_details s\_bd) bd.title = s\_bd.title)
9.  $\pi$  concat(customer\_first\_name, customer\_last\_name),  
transaction\_date, title, amount, (customer\_price \* amount)  
( $\sigma$ (transactions t  $\bowtie$  customers c  $\bowtie$  transactions\_details td  $\bowtie$   
books\_prices bp) t.customer\_id = c.customer\_id AND t.transaction\_id =  
td.transaction\_id AND bd.book\_id = td.book\_id AND bp.book\_id =  
td.book\_id)
10.  $\pi$  concat(customer\_first\_name, customer\_last\_name),  
order\_date, title, total\_amount, (customer\_price \* total\_amount)  
( $\sigma$ (orders o  $\bowtie$  customers c  $\bowtie$  order\_details od  $\bowtie$  books\_details bd  $\bowtie$   
books\_prices bp) c.customer\_id = o.customer\_id AND o.order\_id =

od.order\_id AND bd.book\_id = od.book\_id AND bp.book\_id =  
od.book\_id)

11.  $\pi$  sum(total\_amount \* weight), shipping\_type\_price,  
shipping\_type\_name, ( sum(total\_amount\*weight) \* 3 +  
shipping\_type\_price) as total\_price, sum(total\_amount\*weight) \* 3  
( $\sigma$ (books\_details bd  $\bowtie$  order\_details od  $\bowtie$  orders o  $\bowtie$  shipping\_types  
st) od.book\_id = bd.book\_id AND o.order\_id = od.order\_id AND  
st.shipping\_type\_id = o.shipping\_method)
12.  $\pi$  concat(customer\_first\_name, ' ', customer\_last\_name)  
order\_id, order\_date, title, total\_amount, shipping\_status,  
payment\_method\_name, shipping\_type\_name ( $\sigma$ (books\_details bd  $\bowtie$   
order\_details od  $\bowtie$  orders o  $\bowtie$  shipping\_types st  $\bowtie$  payment\_method  
pm) o.customer\_id = c.customer\_id AND o.order\_id = od.order\_id AND  
o.shipping\_method = st.shipping\_type\_id AND o.payment\_method =  
pm.payment\_method\_id AND bd.book\_id = od.book\_id)
13.  $\pi$  order\_id, concat(customer\_first\_name, ' ',  
customer\_last\_name), order\_date, shipping\_type\_name,  
shipping\_status ( $\sigma$ (customers c  $\bowtie$  order\_details od  $\bowtie$  shipping\_types  
st  $\bowtie$  orders o) o.customer\_id = c.customer\_id AND o.order\_id =  
od.order\_id AND o.shipping\_method = st.shipping\_type\_id)
14.  $\pi$  count(order\_id)  $\sigma$ (orders) shipping\_method between 4 and 5
15.  $\pi$  transaction\_date, transaction\_id , sum(customer\_price \*  
amount) ( $\sigma$ (transaction\_details td  $\bowtie$  books\_prices bp  $\bowtie$  transactions t)  
td.book\_id = bp.book\_id AND t.transaction\_id = td.transaction\_id)
16.  $\pi$  transaction\_id, transaction\_date, (amount \* customer\_price)  
( $\sigma$ (transaction\_details td  $\bowtie$  books\_prices bp  $\bowtie$  transactions t >  
( $\sigma$ (books\_prices bp  $\bowtie$  transactions t) td.book\_id = bp.book\_id AND  
t.transaction\_id = td.transaction\_id)

17.  $\pi$  sum(shipping\_method between 4 and 5), sum(shipping\_method between 1 and 3)  $\sigma$ (orders )
18.  $\pi$  publisher, order\_id, title, order\_date, customer\_id ( $\sigma$ (books\_details bd  $\bowtie$  order\_details od  $\bowtie$  orders o) od.book\_id = bd.book\_id AND o.order\_id = od.order\_id AND bdd.publisher != bd.publisher)
19.  $\pi$  customer\_id, concat(customer\_first\_name, ' ',customer\_last\_name), transaction\_date ( $\sigma$ (customers c  $\bowtie$  transactions t) t.customer\_id = c.customer\_id)
20.  $\pi$  concat(customer\_first\_name, ' ', customer\_last\_name), order\_date ( $\sigma$ (customers c  $\bowtie$  orders o) o.customer\_id = c.customer\_id))
21.  $\pi$  warehouse\_inventory)  $\sigma$ (inventory i  $\bowtie$  orders o  $\bowtie$  order\_details od) od.order\_id = o.order\_id)
22.  $\pi$  sum(books\_amount), sum(books\_amount \* store\_price) ( $\sigma$ (store\_purchases  $\bowtie$  books\_prices bp  $\bowtie$  books\_details bd) bp.book\_id = sp.book\_id AND bd.book\_id = sp.book\_id)
23.  $\pi$  sum(amount \* customer\_price) –  $\pi$  (sum(books\_amount \* store\_price) ( $\sigma$  store\_purchases sp  $\bowtie$  books\_details bd  $\bowtie$  books\_prices bp  $\bowtie$  transactions t) bp.book\_id = sp.book\_id AND bd.book\_id = sp.book\_id AND bp.book\_id = td.book\_id AND bd.book\_id = td.book\_id AND t.transaction\_id = td.transaction\_id)
24.  $\pi$  month(transaction\_date), avg(amount\*customer\_price) ( $\sigma$ (transaction\_details td  $\bowtie$  books\_prices bp  $\bowtie$  transactions t) td.transaction\_id = t.transaction\_id AND td.book\_id = bp.book\_id)
25.  $\pi$  concat(employee\_first\_name, ' ',employee\_last\_name), hours\_monthly, month, year, (hours\_monthly \* 35) ( $\sigma$ (employees\_salary es  $\bowtie$  employees\_details ed) ed.employee\_id = es.employee\_id)

26.  $\pi$  employee\_id, employee\_first\_name, employee\_last\_name,  
count(transaction\_id) ( $\sigma$ (employees\_details e  $\bowtie$  transactions t)  
t.employee\_id = e.employee\_id