

# דוח פרויקט בבסיסי נתונים

גרסה: 1.3

תאריך: 28 ליוני 2020, ו' תמוז התש"ף

קישור לקוד: <https://github.com/Sarieldv/DB-mini-project>

מנחה: מר יאיר גולדשטיין

מגישים: שמואל גרבר 323947747, שריאל סיגל 212413165

קורס: מיני פרויקט בבסיסי נתונים (150225)

מספר קבוצה: 49

מוסד אקדמי: המרכז האקדמי לב

## תוכן עניינים

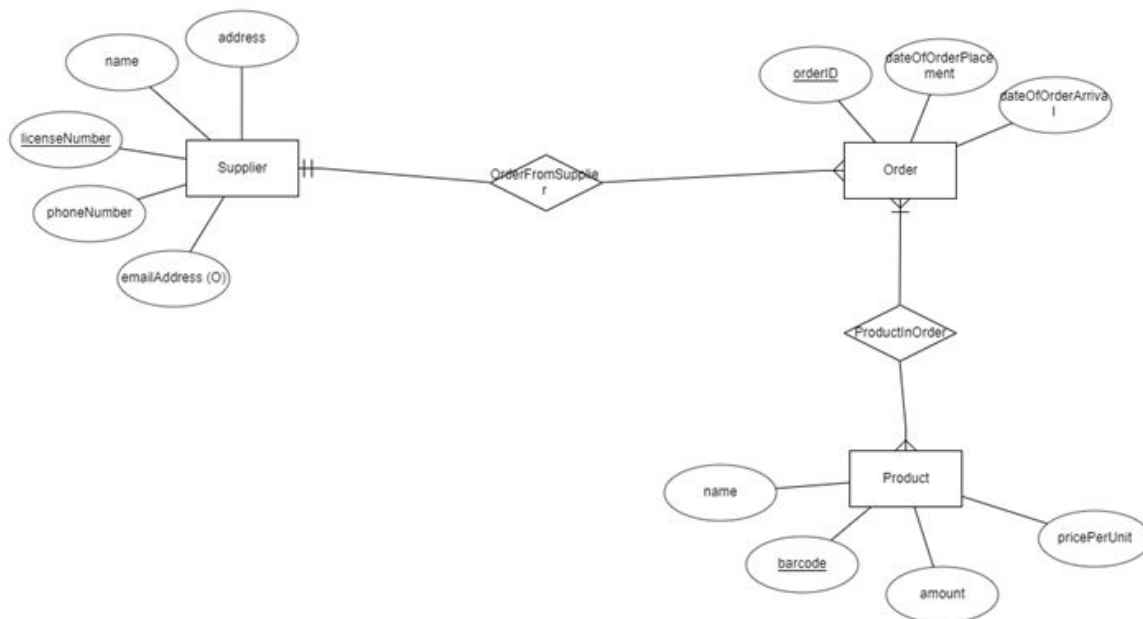
0	עמוד שער
2	הפרויקט שלנו
2	תרשים ERD
3	תיאור הישויות והקשרים
3	ישויות
3	קשרים
4	תרשים DSD
4	יצירת הטבלאות
5	אכלוס הטבלאות
7	שאלות SQL
13	אינדקסים
17	אינטגרציה והרשאות
18	שאלות האינטגרציה
21	תרשימים
23	Views
27	פונקציות
28	פרוצדורות
31	נספחים

## הפרויקט שלנו

כחלק מפרויקט בניית בסיס הנתונים עבור גן חיות, אנו לקחנו על עצמנו את אחריות טיפול בתחום הספקים וההזמנות. תפקיד מערכת זו כחלק מהמערכת הכללית של גן החיות הינה אספקת מוצרים לכל אגפי הגן כגון מוצרים לחיות עצמם, מוצרים עבור מלאי דוכני המכירה וכן כל אספקה אחרת הנדרשת לגן החיות.

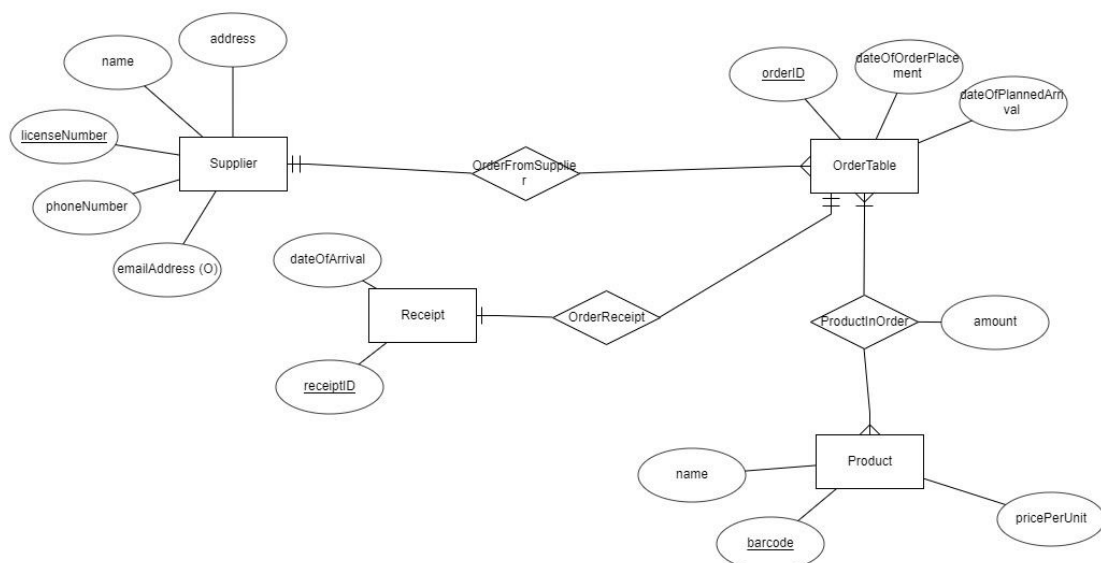
## תרשים ERD

כפי שאמרנו אנחנו אחראים על הספקים ההזמנות והמוצרים ועל כן בתור סקיצה ראשונית יצרנו את תרשים ה ERD הבא:



הרעיון הינו שלכל הזמנה יהיה ספק יחיד אך לכל ספק יתכנו מספר הזמנות. ולכל הזמנה יהיו מספר מוצרים והמוצרים יוכלו להיות בכמה הזמנות.

אך, כאשר התבוננו בסקיצה ראינו בה מספר בעיות. ראשית, לפי הסקיצה הראשונה, מוצר הינו בעצם שורה בהזמנה, כלומר שלכל כמות של מוצר נצטרך ברקוד שונה וזה ממש לא הגיוני! עוד עניין הינו



שרצינו להוסיף עוד מידע בתרשים שלנו על זמן ההגעה האמיתי שבו הגיעה ההזמנה ולכן יצרנו את התרשים הבא:

כפי שניתן לראות, בתרשים השני והסופי העברנו את amount מלהיות תכונה של product ללהיות תכונה של הקשר productInOrder וזה גורם לכך שלכל מוצר יהיה כמות רק בהקשר של הזמנה והכמות לא תהיה בתכונות של המוצר עצמו. זה יתן לנו שאותו מוצר יוכל להיות בכמויות שונות בהזמנות שונות. בנוסף, בתרשים השני הוספנו את הישות Receipt המתארת קבלה ובה תהיה תכונה של תאריך ההגעה האמיתי.

## תיאור הישויות והקשרים

### ישויות

- Supplier – ישות זאת מייצגת את הספקים.
  - licenseNumber - מספר מזהה של רישיון הספק (המפתח).
  - name - שם הספק.
  - phoneNumber - מספר הטלפון של הספק.
  - address - כתובתו של הספק.
  - emailAddress - כתובת המייל של הספק (אופציונלי - לא לכל הספקים חייבת להיות כתובת מייל)
- OrderTable – ישות המתארת הזמנה.
  - orderID - המספר המזהה של ההזמנה (מפתח).
  - dateOfOrderPlacement - התאריך בו הוזמנה ההזמנה.
  - dateOfOrderPlannedArrival - התאריך בו ההזמנה מתוכננת להגיע.
- Product – ישות המייצגת מוצר.
  - barcode - הברקוד של המוצר (המפתח).
  - name - שם המוצר
  - pricePerUnit - מחיר ליחידה
- Receipt – הישות שמתארת קבלה.
  - receiptID - מספר הזיהוי של הקבלה (המפתח).
  - dateOfArrival - תאריך ההגעה האמיתי של ההזמנה.

### קשרים

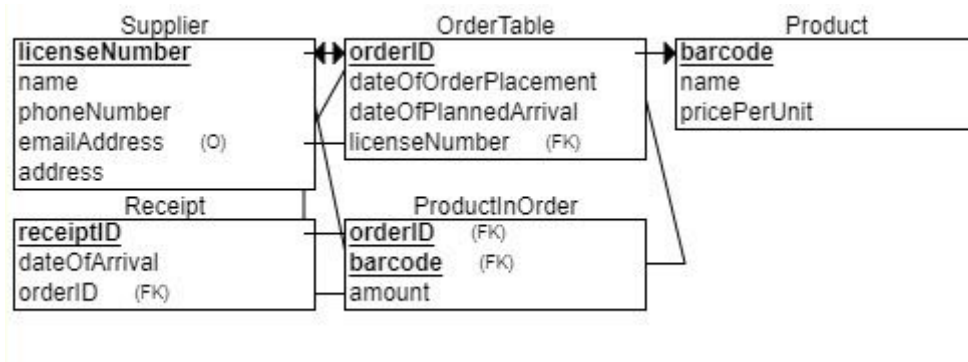
- OrderFromSupplier – הקשר בין OrderTable לבין Supplier. הקשר הוא  $M : 1$  משום שלכל הזמנה ישנו ספק אחד ויחיד אך לכל ספק יכולות להיות מספר הזמנות.
- ProductInOrder – הקשר בין Product לבין OrderTable. הקשר הוא  $M:M$  משום שיכול להיות להזמנה אחת הרבה מוצרים וכל מוצר יכול להיות בכמה הזמנות. לקשר זה ישנה תכונה -

amount. תכונה זו אומרת עבור כל קשר של הזמנה מסוימת ומוצר מסוים כמה יש מהמוצר הספציפי הזה בהזמנה הספציפית הזאת.

- OrderReceipt - הקשר בין OrderTable לבין Receipt. הקשר הוא 1 : 1 משום שלכל הזמנה ישנה קבלה אחת ויחידה וכן להפך.

## תרשים DSD

על פי תרשים ה- ERD ועל ידי הבנת הקשרים בין הישויות, יצרנו תרשים DSD עבור החלק שלנו במערכת:



## יצירת הטבלאות

לאחר שיצרנו את תרשים ה- ERD, הפקנו באתר ERDPlus קוד ב- SQL ליצירת הטבלאות שלנו. פתחנו את התוכנה PLSQL ויצרנו דף SQL. העתקנו את הקוד שקיבלנו לדף והרצנו וכך יצרנו את כל הטבלאות שלנו.



## אכלוס הטבלאות

באכלוס הטבלאות שלנו בפרויקט, השתמשנו בכלים מגוונים. הכלי המרכזי שהשתמשנו בו הינו אתר MOCKAROO המאפשר בחירה מקטגוריות רבות של נתוני סרק. לדוגמא, עבור הסכמה PRODUCT ישנם שלושה שדות: ברקוד (מפתח), שם המוצר ומחיר ליחידה. באתר MOCKAROO ניתן להגדיר את הסכמה, ולבחור מאיזה מאגר נתונים לבחור את השדה הרצוי:

Product Save Changes

Field Name	Type	Options
barcode	Row Number	blank: 0 % fx ×
name	Product (Grocery)	blank: 0 % fx ×
pricePerUnit	Number	min: 1 max: 500 decimals: 2 blank: 0 % fx ×

לאחר מכן ניתן לייצר עד אלף נתונים (במנוי חינמי) ולהוריד אותם כפקודות INSERT או כקובץ CSV. באתר ישנה אפשרות אף להעלות קובץ CSV של נתונים ולהשתמש בערכים ספציפיים מתוך בסיסי הנתונים שמעלים.

ordertable Save Changes

Field Name	Type	Options
orderId	Row Number	blank: 0 % fx ×
dateOfOrderPlacement	Datetime	6/7/2015 to 6/7/2018 in SQL datetime blank: 0 % fx ×
dateOfOrderArrival	Datetime	6/7/2018 to 6/7/2021 in SQL datetime blank: 0 % fx ×
licenseNumber	Dataset Column	Suppliers licenseNumber random blank: 0 % fx ×

למטרת הפרויקט היו סכמות שרצינו ליצור להם לפחות עשרים אלף נתונים, לכן, מכיוון שרצינו להשתמש בפלאטפורמה של MOCKAROO אך מסיבות טכניות (מנוי חינמי בלבד) לא יכולנו לאכלס טבלה ביותר מאלף נתונים בבת אחת. בסופו של דבר נדרשנו להשתמש באפשרות המוצעת באתר MOCKAROO של APPEND DATA. דבר זה מאפשר לנו להעלות לאתר נתונים כקובץ CSV ולהוסיף עוד אלף כל פעם.

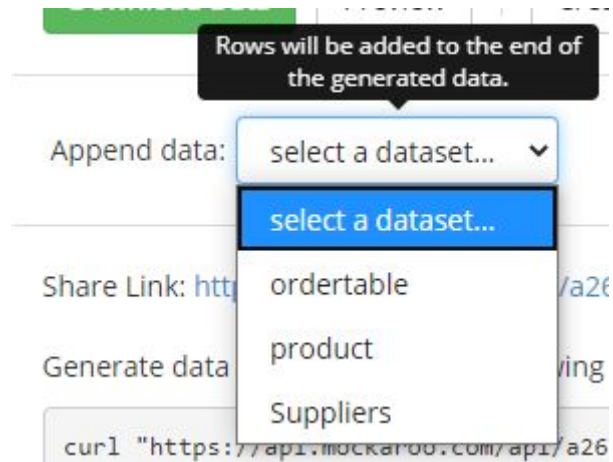
## My Datasets

Have existing data you want reference in your schemas?

Upload your CSV datasets here, then import columns into your schemas using the **Dataset Column** type.

Upload a New Dataset

Name	File	Last Modified
ordertable	ordertable.csv	11 days ago
product	Product.csv	10 days ago
Suppliers	suppliers.csv	14 days ago



כדי להגיע לעשרים אלף נתונים נדרש לחזור על תהליך זה עשרים פעם. דבר זה גרם לעיכוב רציני ביצירת הנתונים. בעייתיות נוספת היתה שכאשר יוצרים את המידע בצורה הזאת, המפתח שקבענו אותו להיות מספר רץ חוזר על עצמו, כלומר בסוף התהליך היה לנו קובץ CSV שבעמודת הID היו את המספרים 1-1000 בצורה מחזורית עשרים פעם. לכן, היה נדרש לפתוח את הקובץ בטרמינל פייתון ולשנות את כל העמודה הראשונה למספר רץ מ1 עד 20000.

כלי נוסף שהשתמשנו בו לאכלוס הטבלאות הינו DATA GENERATOR המובנה בPLSQL. השתמשנו בכלי זה ספציפית לאכלוס טבלת PRODUCTINORDER. DATA GENERATOR מאפשר יצירה מהירה מאוד של נתונים רבים. לקח כמות קצרה מאוד של זמן לייצר כשלושים ושש אלף נתונים. בסכמה PRODUCTINORDER ישנם שלושה שדות: ID של ההזמנה בה המוצר מופיע, ברקוד המוצר (ביחד, שני אלו מהווים מפתח) וכמות יחידות המוצר בהזמנה. DATA GENERATOR מאפשר לייצר נתונים מתוך טבלאות אחרות:

Owner	Table	Number of records
SASEGAL	PRODUCTINORDER	30000,40000

Name	Type	Size	Data
AMOUNT	NUMBER		Random(1, 100)
ORDERID	CHAR	10	List(select orderid from ordertable)
BARCODE	CHAR	15	List(select barcode from product)

בעיה העולה משימוש בכלי זה הינה שלא יכולנו לבחור לא ליצור נתונים שלא יופיעו בהם זוגות של ORDERID וברקוד שכבר הופיעו ביחד. לכן, בזמן יצירת הנתונים, פעמים רבות הופיעה על המסך הודעת שגיאה שסעיף UNIQUE הופר. לכן במציאות כדי לייצר נתונים רבים, פשוט לחצנו OK לאחר שהודעה כזו הופיעה, ויצירת הנתונים המשיכה עד ששוב הייתה הפרה של UNIQUE. המשכנו כך עד שייצרנו כשלושים ושש אלף נתונים.

## שאלות SQL

1. כחלק מסיכום הזמנות הנהלת גן החיות מעוניינת לבדוק אילו ספקים איחרו בהבאת ההזמנות ביחס לתאריך בו הם התחייבו, ככה תדע ההנהלה עם איזה עסקים כדאי לעבוד.

```
select distinct name as supplierName, licenseNumber
from supplier natural join receipt r natural join ordertable o
where r.dateofarrival > o.dateofplannedarrival
group by name, licenseNumber
order by name;
```

	SUPPLIERNAME	LICENSENUMBER
1	Abbott and Sons	390
2	Abbott-Lubowitz	520
3	Abernathy Group	832
4	Abernathy, Waters and Murazik	438
5	Abernathy-Mertz	833
6	Abshire Inc	56
7	Abshire, Torp and Nader	440
8	Adams Inc	939
9	Adams, Bauch and Sipes	521
10	Adams, Lind and Welch	33
11	Altenwerth and Sons	195
12	Altenwerth, Smitham and Brekke	652
13	Anderson Group	513
14	Anderson-Altenwerth	200
15	Anderson-Emmerich	407
16	Anderson-Wilkinson	55
17	Ankunding, Brakus and Lowe	528
18	Ankunding, Kuvalis and Windler	178
19	Ankunding, Lakin and Deckow	109

2. ההנהלה מעוניינת בסיכום של כל הזמנה כמה היא עלתה, כי בתרשים הפרטים הינם רק כמה עלה כל מוצר בהזמנה ולא כמה עלתה ההזמנה בכללותה ולכן צריך את השאלתה שתחשב



זאת.

```
select orderid, sum(price) as totalPrice
from (select priceperunit * amount as price, orderid from product natural join productinorder)
group by orderid;
```

	ORDERID	TOTALPRICE
1	17869	7600.52
2	12921	90450.61
3	18205	40164.43
4	6026	513.15
5	5694	8693.88
6	5958	6723.47
7	7750	45436.59
8	8614	63639.96
9	4061	18602.57
10	2742	2638.93
11	2394	64939.34
12	9638	68691.47
13	16224	42156.33
14	16432	29323
15	12052	44813.46
16	7691	43975.95
17	19273	49330.99
18	13400	38503.05
19	17125	50766.42
20	19158	92800.4
21	14819	93529.28

0:07 Sasegal@labdbwin 16845 rows selected in 7.805 seconds

3. ההנהלה מעוניינת לדעת בכמה כסף היא הזמינה מוצרים מכל ספק בשנת 2018. על מנת לכתוב את דו"ח ההוצאות השנתי שלה לשנת 2018.

```
select supplierName, licensenumber, sum(totalprice) as totalyearlyprice
from (select dateofarrival, name as suppliername, licensenumber, totalprice
from (select orderid, sum(price) as totalPrice
from (select priceperunit * amount as price, orderid from product natural join productinorder)
group by orderid) natural join ordertable natural join supplier natural join receipt
where TO_DATE('2018/01/01 00:00:00', 'yyyy/mm/dd hh24:mi:ss') <= dateofarrival and
TO_DATE('2019/01/01 00:00:00', 'yyyy/mm/dd hh24:mi:ss') >= dateofarrival)
group by suppliername, licensenumber;
```

	SUPPLIERNAME	LICENSENUMBER	TOTALYEARLYPRICE
1	Botsford-Prohaska	30	65672.05
2	Runte, Beer and Breitenberg	928	5549.06
3	Steuber, Durgan and Padberg	542	52907.04
4	Smith-Yost	756	37631.7
5	Sporer and Sons	99	38793.52
6	Champlin, Hagenes and Will	937	109618.88
7	Hickle-Schiller	388	68579.95
8	Schneider-Lueilwitz	314	129456.11
9	Wunsch Group	798	15277.62
10	Spinka Group	984	65276.25
11	Shanahan-Skiles	201	125637.42
12	Kunde, Russel and Auer	138	82810.84
13	Miller, Romaguera and Adams	877	19419.91
14	Schultz-Little	840	18178.52
15	Mertz, Koss and Krajcik	608	19222.73
16	Towne Inc	385	40166.53

1:1 Sasegal@labdbwin 545 rows selected in 0.306 seconds

4. ההנהלה מעוניינת לדעת אילו הזמנות הגיעו לפני הזמן שנקבע להגעתן כדי שתוכל לשער מראש את זמן ההגעה של ההזמנות בעתיד.

```
select orderid, receiptid, dateofplannedarrival, dateoforderplacement, dateofarrival
from ordertable natural join receipt
where dateoforderplacement<=dateofarrival and dateofarrival<=dateofplannedarrival
group by orderid, receiptid, dateofplannedarrival, dateoforderplacement, dateofarrival
order by dateofarrival;
```

	ORDERID	RECEIPTID	DATEOFPLANNEDARRIVAL	DATEOFORDERPLACEMENT	DATEOFARRIVAL
1	4751	4751	23/11/2018 20:40:25	18/07/2015 02:13:59	07/06/2018 02:09:29
2	841	841	10/07/2020 13:40:09	12/02/2016 19:32:58	07/06/2018 02:51:28
3	1170	1170	27/03/2019 00:23:36	14/09/2015 09:52:19	07/06/2018 08:25:07
4	2796	2796	03/10/2019 04:56:54	08/10/2015 08:01:52	07/06/2018 14:40:57
5	2860	2860	19/09/2020 20:48:26	09/03/2018 14:43:08	07/06/2018 14:48:00
6	4638	4638	25/01/2021 01:20:37	05/02/2016 12:27:57	08/06/2018 03:03:12
7	1829	1829	13/11/2020 07:56:28	09/05/2017 17:51:24	08/06/2018 03:51:25
8	869	869	12/05/2021 08:42:56	25/03/2017 18:20:13	08/06/2018 08:36:05
9	1712	1712	19/02/2020 12:51:39	22/09/2016 05:32:11	08/06/2018 18:00:41
10	271	271	24/12/2019 11:12:35	30/09/2015 06:28:58	08/06/2018 23:08:42
11	1585	1585	25/09/2018 19:59:26	19/01/2018 13:37:29	09/06/2018 05:48:22
12	2100	2100	29/10/2019 08:09:14	27/11/2017 21:43:25	09/06/2018 06:34:40
13	3618	3618	09/07/2019 11:06:52	02/08/2016 10:36:03	09/06/2018 07:49:29
14	2132	2132	25/09/2020 10:36:28	16/07/2017 19:21:07	09/06/2018 12:24:03
15	2929	2929	01/03/2020 12:24:51	06/01/2017 23:31:55	09/06/2018 13:03:40
16	2461	2461	11/09/2018 08:40:44	24/05/2018 18:41:05	09/06/2018 20:03:49
17	2533	2533	23/11/2020 17:44:10	07/06/2016 18:19:01	10/06/2018 03:33:09
18	181	181	21/10/2019 19:35:03	26/04/2017 12:19:58	10/06/2018 08:26:08

5. ההנהלה צריכה רשימה של כל המידע על כל ההזמנות שהגיעו, על מנת לוודא שכל המוצרים שהגיעו נקנו או מופיעים במלאי.

	ORDERID	SUPPLIERNAME	LICENSENUMBER	DATEOFARRIVAL	DATEOFORDERPLACEMENT
1	508	O'Conner Inc	560	29/08/2019 16:22:15	03/06/2016 10:36:28
2	510	Kozey-MacGyver	439	07/07/2018 22:40:49	07/08/2015 21:10:46
3	519	Sipes, MacGyver and Graham	167	29/05/2019 17:13:25	10/08/2015 07:10:28
4	523	Luetngen Inc	24	12/08/2020 12:48:03	27/02/2016 18:44:47
5	528	Terry-Schroeder	693	02/08/2020 22:57:36	23/01/2016 20:49:06
6	540	Wuckert-Cartwright	581	22/06/2019 23:01:15	20/03/2018 12:16:07
7	557	Luehlwitz-Kohler	360	16/12/2019 22:05:12	01/10/2017 07:08:38
8	592	Bernhard-Mills	918	24/04/2020 21:12:32	06/11/2017 12:43:24
9	593	Langosh LLC	659	16/09/2020 23:17:21	31/05/2017 11:37:11
10	595	Willms LLC	349	02/03/2019 21:04:16	22/11/2016 14:02:25
11	597	Zieme Group	765	03/10/2019 14:48:55	04/03/2018 16:01:37
12	603	Rempel, Wunsch and Jerde	763	18/04/2020 06:29:27	05/03/2018 10:00:22
13	606	Bartell, Goldner and Dickens	187	24/11/2018 06:45:55	30/04/2017 10:50:45
14	618	Klein and Sons	910	13/02/2020 18:03:26	05/03/2016 23:21:09
15	630	Larkin and Sons	674	19/03/2021 00:54:53	26/08/2015 18:28:36
16	636	Carroll and Sons	952	10/11/2020 23:06:33	19/05/2017 05:20:25
17	646	Corkery-Rodriguez	293	07/11/2019 23:18:42	17/06/2017 04:26:54
18	649	Leffler LLC	11	01/11/2018 09:18:39	18/11/2016 02:23:56
19	650	Hickle, Hand and Gibson	65	26/04/2020 11:11:14	14/05/2018 08:20:57
20	658	Daniel, Heller and Kertzmann	412	10/06/2018 23:28:32	05/07/2015 04:43:51

6. הספקים מעוניינים לדעת איזה מוצר הינו הכי יקר על מנת לחשב רווחים במקרי קיצון (מוצרים יקרים מאוד) עבור המוצרים שלהם.

The screenshot shows a SQL IDE with the following query in the SQL pane:

```
select *
from product
where priceperunit>=all(select distinct priceperunit from product);
```

The Output pane displays the result of the query, which is a single row representing the most expensive product:

BARCODE	NAME	PRICEPERUNIT
355	Chinese Foods - Chicken	499.96

The status bar at the bottom indicates: 1 row selected in 0.029 seconds.

7. ההנהלה מעוניינת לדעת כמה הזמנות היא הזמינה מכל ספק, כדי לכתוב בדוח ההוצאות שלה.

The screenshot shows a SQL IDE with the following query in the SQL pane:

```
select name as suppliername, licenseNumber, count(orderid) as numberOfOrders
from supplier natural join ordertable
group by name, licenseNumber;
```

The Output pane displays the result of the query, which is a list of suppliers and the number of orders they have placed:

SUPPLIERNAME	LICENSENUMBER	NUMBEROFORDERS
1 Morissette and Sons	51	20
2 Anderson-Wilkinson	55	25
3 Klein, Stehr and Prosacco	57	13
4 Olson Group	62	20
5 Hickie, Hand and Gibson	65	25
6 Dooley-Crist	66	21
7 Jakubowski-Schoen	75	20
8 Rempel, Heidenreich and Hand	174	15
9 Nienow, Watsica and Robel	192	23
10 Jakubowski-Collins	194	17
11 Shanahan-Skiles	201	17
12 Cummerata LLC	91	21
13 Sporer and Sons	99	22
14 Boyer-McGlynn	119	21
15 Crooks Inc	126	27
16 Lesch, Friesen and Blanda	133	20
17 Kunde, Russel and Auer	138	19
18 Rath, Daniel and Willms	156	26

The status bar at the bottom indicates: 1000 rows selected in 0.405 seconds.

8. ההנהלה מעוניינת לדעת את כל המוצרים שהוזמנו בהזמנה כלשהיא עם כמות יותר מ 10 ופחות מ - 100 משום שמוצרים בכמות כזו יש תשלום מופחת בדמי השירות .

```
select orderid,amount,priceperunit, barcode, name
from (select * from productinorder where amount > 10 and amount < 100) natural join ordertable natural join product;
```

	ORDERID	AMOUNT	PRICEPERUNIT	BARCODE	NAME
1	8024	23	290.32	609	Bagel - Plain
2	2785	44	181.22	848	Initation Crab Meat
3	2997	74	311.01	629	Quail - Whole, Boneless
4	142	42	347.81	500	Cheese - Fontina
5	19221	81	354.71	400	Rice - Arborio
6	3646	45	289.89	683	Cheese - Shred Cheddar / Mozza
7	10220	73	455.50	898	Cake - Lemon Chiffon
8	8153	76	46.29	922	Pasta - Fett Alfredo, Single Serve
9	15461	57	455.05	682	Ocean Spray - Kiwi Strawberry
10	18269	87	402.14	282	Wine - Ice Wine
11	8646	87	90.38	811	Coffee - Beans, Whole
12	11205	95	64.48	741	Worcestershire Sauce
13	17869	56	82.77	25	Bread - Multigrain Oval
14	2885	63	81.42	406	Quail - Jumbo Boneless
15	2570	23	324.55	528	Shrimp - Black Tiger 6 - 8
16	5159	49	27.52	9	Beef - Tenderloin
17	9245	50	104.93	711	Wine - Jafflin Bourgongone
18	18235	55	9.99	140	Gatorade - Fruit Punch
19	18783	77	484.26	7	Pumpkin - Seed

0:34 Sasegal@labdbwin 32881 rows selected in 34.237 seconds



## אינדקסים

1. שאלנו שאילתה המחזירה לנו את כל המוצרים שהוזמנו בהזמנה מסוימת שכמותם הינו יותר מ10 אך פחות מ100. ללא אינדקס שאילה זו לקחה כ 34 שניות.

על שאילתה זו יצרנו 2 אינדקסים נפרדים:

(1) אינדקס ראשון יצרנו על כמות המוצר שהוזמן השמור בamount שבטבלת ProductInOrder. בחרנו באינדקס זה מכיוון שהוא ממין את התכונה עליה רצינו לעבוד, ובשאילתה זו סדר החיפוש מקל רבות. ההסתברות לקבל מוצרים בעלי אותו שם היא כ - 0.01, אך עיקר השיפור נובע מהמיון. אינדקס זה הוריד כ 9 שניות מזמן ביצוע השאילתה.

(2) אינדקס אחר שייצרנו הינו אינדקס על שם המוצר name השמור בproduct. האינדקס מקל על חיפוש בטבלת תוצאה כה ארוכה. ההסתברות לקבל מוצרים בעלי אותו שם היא כ - 0.0017. אינדקס זה הוריד את זמן ביצוע השאילתה לכ 12 שניות.

להלן תוצאות הוספת האינדקסים:

● ללא אינדקס:

```
select orderid,amount,priceperunit, barcode, name
from (select * from productinorder where amount > 10 and amount < 100) natural join ordertable natural join product;
```

	ORDERID	AMOUNT	PRICEPERUNIT	BARCODE	NAME
1	8024	23	290.32	609	Bagel - Plain
2	2785	44	181.22	848	Initation Crab Meat
3	2997	74	311.01	629	Quail - Whole, Boneless
4	142	42	347.81	500	Cheese - Fontina
5	19221	81	354.71	400	Rice - Arborio
6	3646	45	289.89	683	Cheese - Shred Cheddar / Mozza
7	10220	73	455.50	898	Cake - Lemon Chiffon
8	8153	76	46.29	922	Pasta - Fett Alfredo, Single Serve
9	15461	57	455.05	682	Ocean Spray - Kiwi Strawberry
10	18269	87	402.14	282	Wine - Ice Wine
11	8646	87	90.38	811	Coffee - Beans, Whole
12	11205	95	64.48	741	Worcestershire Sauce
13	17869	56	82.77	25	Bread - Multigrain Oval
14	2885	63	81.42	406	Quail - Jumbo Boneless
15	2570	23	324.55	528	Shrimp - Black Tiger 6 - 8
16	5159	49	27.52	9	Beef - Tenderloin
17	9245	50	104.93	711	Wine - Jafflin Bourgongone
18	18235	55	9.99	140	Gatorade - Fruit Punch
19	18783	77	484.26	7	Pumpkin - Seed

0:34 Sasegal@labdbwin 32881 rows selected in 34.237 seconds

• עם אינדקס על amount:

SQL Output Statistics

```
select orderid,amount,priceperunit, barcode, name
from (select * from productinorder where amount > 10 and amount < 100) natural join ordertable natural join product;
```

	ORDERID	AMOUNT	PRICEPERUNIT	BARCODE	NAME
1	8024	23	290.32	609	Bagel - Plain
2	2785	44	181.22	848	Initiation Crab Meat
3	2997	74	311.01	629	Quail - Whole, Boneless
4	142	42	347.81	500	Cheese - Fontina
5	19221	81	354.71	400	Rice - Arborio
6	3646	45	289.89	683	Cheese - Shred Cheddar / Mozza
7	10220	73	455.50	898	Cake - Lemon Chiffon
8	8153	76	46.29	922	Pasta - Fett Alfredo, Single Serve
9	15461	57	455.05	682	Ocean Spray - Kiwi Strawberry
10	18269	87	402.14	282	Wine - Ice Wine
11	8646	87	90.38	811	Coffee - Beans, Whole
12	11205	95	64.48	741	Worcestershire Sauce
13	17869	56	82.77	25	Bread - Multigrain Oval
14	2885	63	81.42	406	Quail - Jumbo Boneless
15	2570	23	324.55	528	Shrimp - Black Tiger 6 - 8
16	5159	49	27.52	9	Beef - Tenderloin
17	9245	50	104.93	711	Wine - Jafflin Bourgongone
18	18235	55	9.99	140	Gatorade - Fruit Punch
19	18783	77	484.26	7	Pumpkin - Seed
20	6448	65	407.29	415	Island Oasis - Mango Daiquiri
21	18870	52	489.50	307	Pepper - Paprika, Spanish
22	12921	76	452.34	613	Ice Cream Bar - Hageen Daz To

1:1 0:25 Sasegal@labdbwin 32881 rows selected in 25.028 seconds

• עם אינדקס על name:

```
select orderid,amount,priceperunit, barcode, name
from (select * from productinorder where amount > 10 and amount < 100) natural join ordertable natural join product;
```

	ORDERID	AMOUNT	PRICEPERUNIT	BARCODE	NAME
1	8024	23	290.32	609	Bagel - Plain
2	2785	44	181.22	848	Initiation Crab Meat
3	2997	74	311.01	629	Quail - Whole, Boneless
4	142	42	347.81	500	Cheese - Fontina
5	19221	81	354.71	400	Rice - Arborio
6	3646	45	289.89	683	Cheese - Shred Cheddar / Mozza
7	10220	73	455.50	898	Cake - Lemon Chiffon
8	8153	76	46.29	922	Pasta - Fett Alfredo, Single Serve
9	15461	57	455.05	682	Ocean Spray - Kiwi Strawberry
10	18269	87	402.14	282	Wine - Ice Wine
11	8646	87	90.38	811	Coffee - Beans, Whole
12	11205	95	64.48	741	Worcestershire Sauce
13	17869	56	82.77	25	Bread - Multigrain Oval
14	2885	63	81.42	406	Quail - Jumbo Boneless
15	2570	23	324.55	528	Shrimp - Black Tiger 6 - 8
16	5159	49	27.52	9	Beef - Tenderloin
17	9245	50	104.93	711	Wine - Jafflin Bourgongone
18	18235	55	9.99	140	Gatorade - Fruit Punch
19	18783	77	484.26	7	Pumpkin - Seed
20	6448	65	407.29	415	Island Oasis - Mango Daiquiri
21	18870	52	489.50	307	Pepper - Paprika, Spanish
22	12921	76	452.34	613	Ice Cream Bar - Hageen Daz To

1:1 0:12 Sasegal@labdbwin 32881 rows selected in 12.678 seconds

2. שאלנו שאילתה המחזירה טבלה המכילה את רשימת כל הספקים שהוזמנו מהם הזמנות בשנת 2018 וכן את מחיר סך ההזמנות שהוזמנו מכל ספק. יצרנו אינדקס על תכונת התאריך של הקבלה של כל הזמנה מכיוון שחשבנו שמיון של התאריכים יביא לשיפור הביצוע. כמעט ואין ערכים שחוזרים על עצמם. אולם, בשל מהירות השאילתה, נראה כי דבר זה לא הביא לאף שיפור. בין כך ובין כך מהירות השאילתה הייתה 0.33 שניות.

• בלי אינדקס:

```
select supplierName, licensenumber, sum(totalprice) as totalyearlyprice
from (select dateofarrival, name as suppliername, licensenumber, totalprice
from (select orderid, sum(price) as totalPrice
from (select priceperunit * amount as price, orderid from product natural join productinorder)
group by orderid) natural join ordertable natural join supplier natural join receipt
where TO_DATE('2018/01/01 00:00:00', 'yyyy/mm/dd hh24:mi:ss') <= dateofarrival and
TO_DATE('2019/01/01 00:00:00', 'yyyy/mm/dd hh24:mi:ss') >= dateofarrival)
group by suppliername, licensenumber;
```

	SUPPLIERNAME	LICENSENUMBER	TOTALYEARLYPRICE
1	Botsford-Prohaska	30	65672.05
2	Runte, Beer and Breitenberg	928	5549.06
3	Steuber, Durgan and Padberg	542	52907.04
4	Smith-Yost	756	37631.7
5	Sporer and Sons	99	38793.52
6	Champlin, Hagenes and Will	937	109618.88
7	Hickle-Schiller	388	68579.95
8	Schneider-Lueilwitz	314	129456.11
9	Wunsch Group	798	15277.62
10	Spinka Group	984	65276.25
11	Shanahan-Skiles	201	125637.42
12	Kunde, Russel and Auer	138	82810.84
13	Miller, Romaguera and Adams	877	19419.91
14	Schultz-Little	840	18178.52
15	Mertz, Koss and Krajcik	608	19222.73
16	Towne Inc	385	40166.53

Sasegal@labdbwin 545 rows selected in 0.306 seconds



• עם אינדקס על dateofarrival:

```
select supplierName, licensenumber, sum(totalprice) as totalyearlyprice
from (select dateofarrival, name as suppliername, licensenumber, totalprice
from (select orderid, sum(price) as totalPrice
from (select priceperunit * amount as price, orderid from product natural join productinorder)
group by orderid) natural join ordertable natural join supplier natural join receipt
where TO_DATE('2018/01/01 00:00:00', 'yyyy/mm/dd hh24:mi:ss') <= dateofarrival and
TO_DATE('2019/01/01 00:00:00', 'yyyy/mm/dd hh24:mi:ss') >= dateofarrival)
group by suppliername, licensenumber;
```

	SUPPLIERNAME	LICENSENUMBER	TOTALYEARLYPRICE
1	Botsford-Prohaska	30	65672.05
2	Runte, Beer and Breitenberg	928	5549.06
3	Steuber, Durgan and Padberg	542	52907.04
4	Smith-Yost	756	37631.7
5	Sporer and Sons	99	38793.52
6	Champlin, Hagenes and Will	937	109618.88
7	Hickle-Schiller	388	68579.95
8	Schneider-Lueilwitz	314	129456.11
9	Wunsch Group	798	15277.62
10	Spinka Group	984	65276.25
11	Shanahan-Skiles	201	125637.42
12	Kunde, Russel and Auer	138	82810.84
13	Miller, Romaguera and Adams	877	19419.91
14	Schultz-Little	840	18178.52
15	Mertz, Koss and Krajcik	608	19222.73

Sasegal@labdbwin 545 rows selected in 0.238 seconds

## אינטגרציה והרשאות

בשלב באינטגרציה הישויות שהיו הכי הגיוניות עבורנו להתחבר אליהן היו הישויות של דוכני המכירה שיש בגן החיות. דוכני המכירה הם מהצרכנים העיקריים של שירות הספקים וההזמנות בגן החיות. לכן, היה מאוד טבעי שנבצע אינטגרציה עם ישות SALES\_STAND אשר ברשות ASLEVI.

אולם, בתכנון המקורי של הישויות בבעלותנו וכן בבעלות ASLEVI לא היו ישויות הקשורות אחת לשנייה. לכן, נדרש היה להוסיף אצל ASLEVI (כמובן בשיתוף פעולה שלנו ושלהם) ישות המייצגת את המלאי של פריט מסויים, בדוכן מסוים ברגע נתון. לאחר מחשבה הגענו למסקנה שנדרש להוסיף את סכמת SUPPLY:

- barcode - הברקוד של המוצר (מתוך PRODUCT).
- ID - מספר הזיהוי של הדוכן (מתוך SALES\_STAND).
- upToDate - תאריך זמן בו המלאי עודכן עבור מוצר והדוכן המתאימים הספציפיים.
- amount - כמות היחידות במלאי.

ביחד (barcode, ID, upToDate) מהווים מפתח עבור SUPPLY מכיוון שאנו רוצים לאפשר שאותו מוצר יוכל להיות בדוכנים רבים וכן שבדוכן יוכלו להיות מוצרים רבים. בנוסף אנו רוצים לשמור את היסטוריית המלאי שלנו על מנת שנוכל לדעת איך המלאי שלנו נראה בכל שלב בעבר. לכן, נדרש לצרף תאריך למפתח כדי לזהות כל INSTANCE בצורה ייחודית.

על מנת שיוכלו ליצור סכמה שעושה REFERENCE לסכמה שלנו נדרש שניתן הרשאת REFERENCE ל- ASLEVI:

```
grant references on Product to aslevi;
```

כדי לאכלס טבלה זו השתמשנו ב-DATA GENERATOR. בשביל האכלוס, היה נדרש לשלוף ערכים מתוך PRODUCT (סכמה שלנו). לכן נתנו גם הרשאת SELECT:

```
grant select on Product to aslevi;
```

## שאלות האינטגרציה

כמובן שעל מנת לבצע את שאלות אלו ASLEVI נתן לנו הרשאת SELECT לטבלת SUPPLY המאוכסת אצלו.

1. מעניין את הספקים לדעת אילו מוצרים נמכרו בגן החיות בשלב כלשהו על מנת להציע מוצרים נוספים הדומים למוצרים שהיו בגן החיות בעבר. לכן, השאלתה מחזירה את השם והברקוד של כל מוצר שאוכלס ב - SUPPLY:

```
select distinct name, barcode
from product natural join aslevi.supply
group by name, barcode
order by barcode;
```

	NAME	BARCODE
1	Five Alive Citrus	1
2	Irish Cream - Baileys	10
3	Rice - Long Grain	100
4	Lambcasing	1000
5	Boogies	101
6	Potatoes - Idaho 100 Count	102
7	Island Oasis - Ice Cream Mix	103
8	Cornstarch	104
9	Toothpick Filled	105
10	Beets - Golden	106
11	Soup - Campbells Chili Veg	107
12	Coke - Classic, 355 Ml	108
13	Crab - Dungeness, Whole, live	109
14	Basil - Seedlings Cookstown	11
15	Grapes - Black	110
16	Sauce - Rosee	111
17	Wine - Red, Marechal Foch	112
18	Chutney Sauce	113
19	Eggwhite Frozen	114
20	Shrimp, Dried, Small / Lb	115

Sasegal@labdbwin 1000 rows selected in 0.327 seconds

2. הספקים מעוניינים לדעת אילו מוצרים עודכנו במלאי בשנת 2018 על מנת שתוכל להחליט כמה מהמוצר להזמין מהמפעלים. לכן, השאילתה מחזירה את כל המוצרים שעודכנו ב- 2018 וכן את תאריך עדכוןם:

```
select name, barcode, upToDate
from product natural join aslevi.supply
where TO_DATE('2018/01/01 00:00:00', 'yyyy/mm/dd hh24:mi:ss') <= upToDate and
TO_DATE('2019/01/01 00:00:00', 'yyyy/mm/dd hh24:mi:ss') >= upToDate
group by upToDate, barcode, name
order by upToDate, barcode;
```

	NAME	BARCODE	UPTODATE
4	Sorrel - Fresh	37	01/01/2018 04:45:15
5	Ocean Spray - Kiwi Strawberry	682	01/01/2018 04:46:16
6	Bread Foccacia Whole	837	01/01/2018 05:44:18
7	Extract - Raspberry	32	01/01/2018 06:25:37
8	Horseradish Root	184	01/01/2018 06:34:33
9	Ice Cream - Chocolate	757	01/01/2018 11:55:06
10	Hipnotiq Liquor	877	01/01/2018 12:34:50
11	Wine - Chenin Blanc K.w.v.	248	01/01/2018 13:10:07
12	Parsley - Fresh	853	01/01/2018 16:44:41
13	Jam - Raspberry	293	01/01/2018 17:05:42
14	Iced Tea - Lemon, 340ml	317	01/01/2018 17:14:12
15	Nori Sea Weed - Gold Label	411	01/01/2018 17:29:28
16	Sauce - Chili	363	01/01/2018 17:30:02
17	Champagne - Brights, Dry	858	01/01/2018 19:58:48
18	Quail - Whole, Boneless	629	01/01/2018 21:21:01
19	Cake - Lemon Chiffon	321	01/01/2018 21:41:45
20	Crab - Claws, 26 - 30	170	01/01/2018 21:56:35

3. הספקים מעוניינים לאמוד את הפופולריות של מוצרים ברחבי דוכנים בגן החיות. לכן, השאלתה מחזירה את המוצרים ביחד עם כמות הדוכנים בהם כל אחד הופיע בעבר:

```
select p.name, barcode, count(id) as amountofstands
from (select distinct id, barcode from aslevi.supply) natural join product p
group by barcode, p.name
order by barcode;
```

	NAME	BARCODE	AMOUNTOFSTANDS
1	Five Alive Citrus	1	15
2	Irish Cream - Baileys	10	20
3	Rice - Long Grain	100	12
4	Lambcasing	1000	20
5	Boogies	101	23
6	Potatoes - Idaho 100 Count	102	19
7	Island Oasis - Ice Cream Mix	103	8
8	Cornstarch	104	18
9	Toothpick Filled	105	17
10	Beets - Golden	106	17
11	Soup - Campbells Chili Veg	107	10
12	Coke - Classic, 355 MI	108	14
13	Crab - Dungeness, Whole, live	109	14
14	Basil - Seedlings Cookstown	11	20
15	Grapes - Black	110	17
16	Sauce - Rosee	111	21
17	Wine - Red, Marechal Foch	112	15
18	Chutney Sauce	113	25
19	Eggwhite Frozen	114	21
20	Shrimp, Dried, Small / Lb	115	18
21	Baking Powder	116	18
22	Beef - Cooked, Corned	117	21
23	Split Peas - Yellow, Dry	118	20
24	Beef - Cooked, Corned	119	22

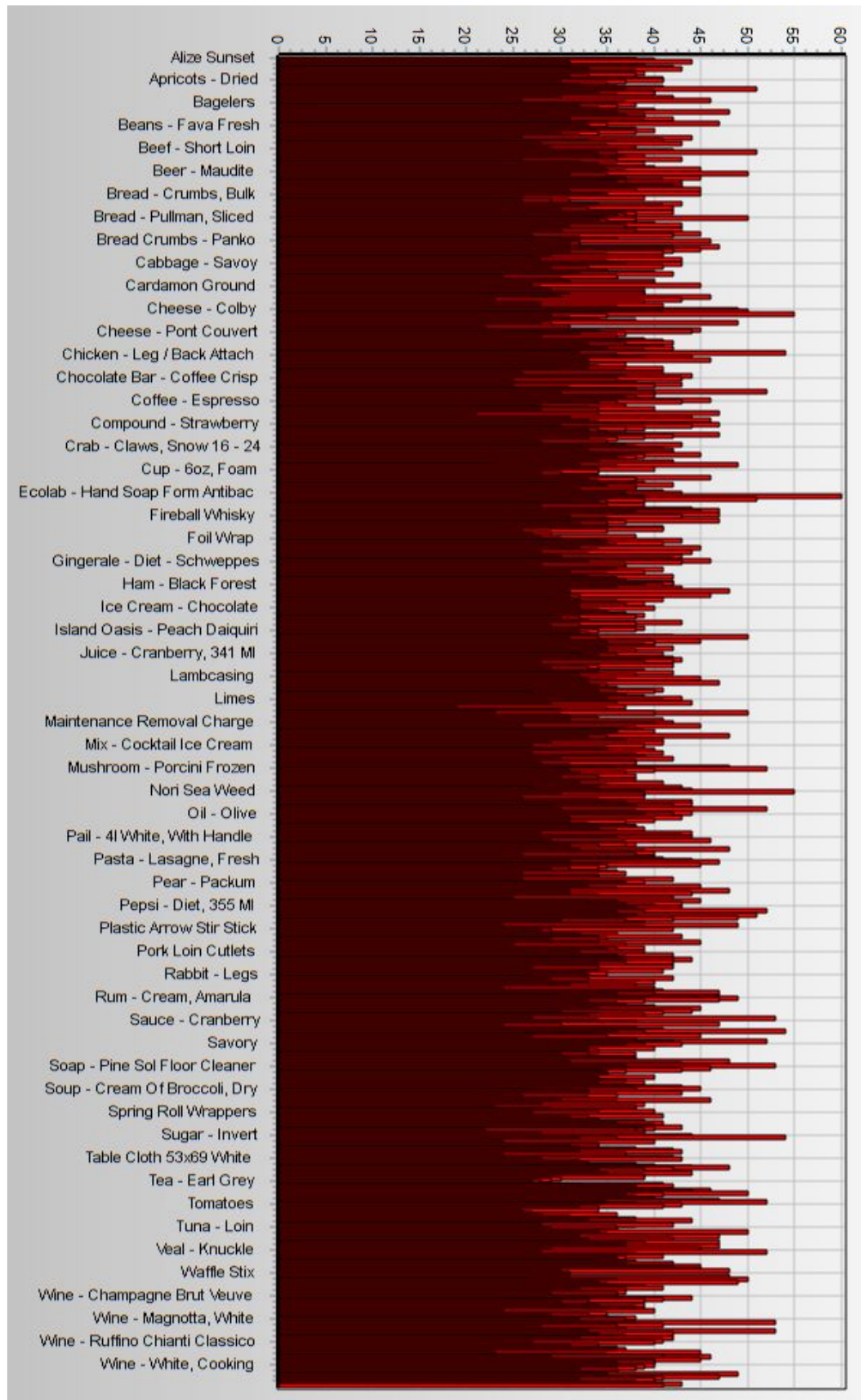
1 of 1000 | Sasegal@labdbwin | 1000 rows selected in 0.404 seconds



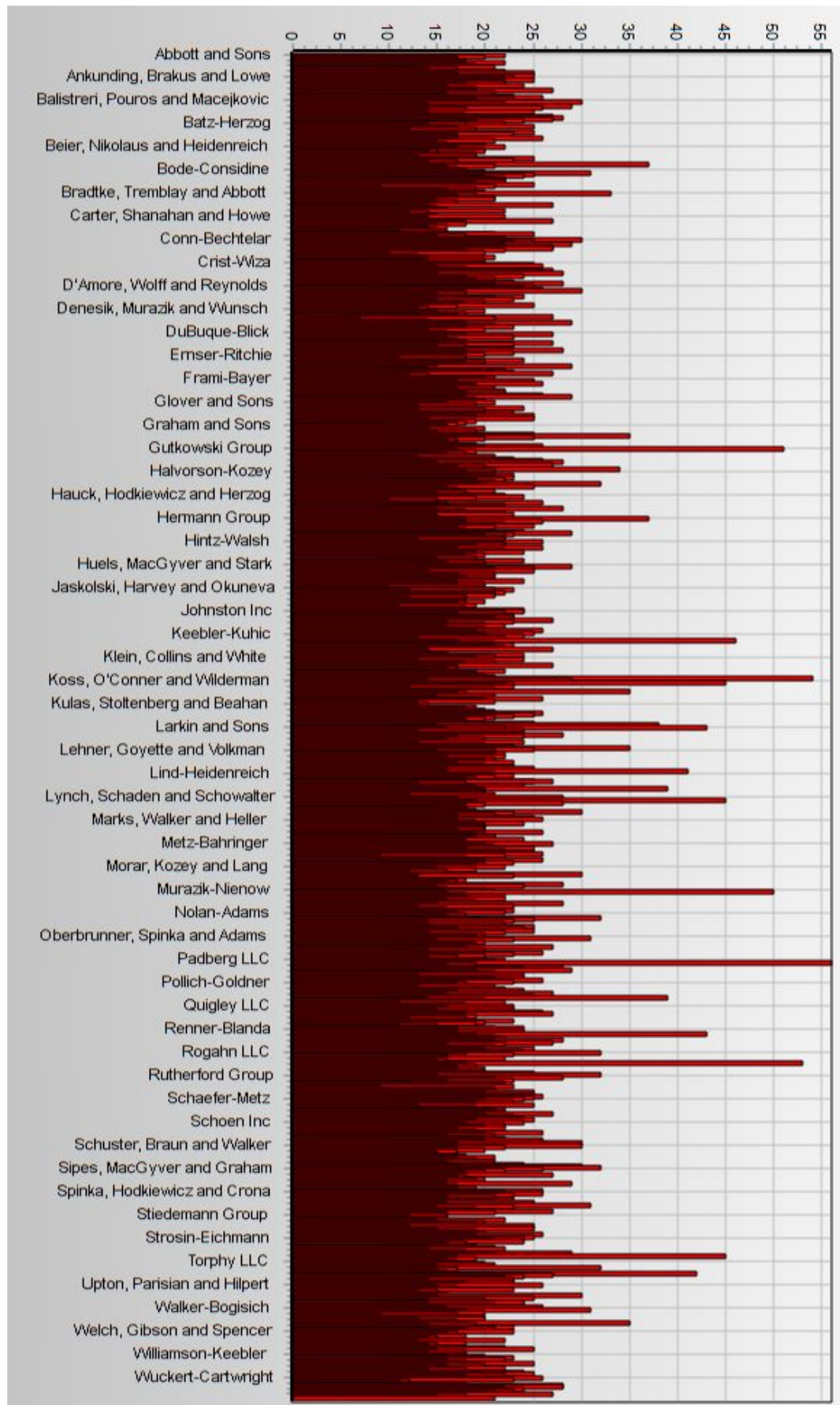
## תרשימים

יצרנו שני תרשימים:

1. תרשים המראה כמה פעמים כל מוצר מופיע בהזמנה כלשהיא:



2. תרשים המראה כמה הזמנות הוזמנו מכל ספק:



## Views

אנחנו התייחסנו לשני משתמשים: אחראי רכש מטעם הנהלת גן החיות שהוא אחראי לעבודה עם הספקים, והמשתמש השני הינו הספק עצמו. לכל משתמש יצרנו שני views.

### 1. אחראי הרכש:

- לא לכל הספקים ישנה כתובת מייל, ולכן כאשר אחראי הרכש מעוניין לשלוח מייל לספקים הוא זקוק לרשימת הספקים שכן יש להם כתובת מייל. לכן יצרנו view שמחזיר רשימה של כל הספקים שיש להם כתובת מייל ואת כתובת המייל שלהם. אין צורך לספק את כל המידע על ההספקים (כגון מספר הטלפון שלהם, הכתובת שלהם וכו') מפני שאחראי הרכש מעוניין רק לשלוח להם מייל בפניה ל view הזה, ולכן לא צירפנו את התכונות הללו ב view זה:

```
SELECT * FROM suppliers_with_email;
```

	NAME	LICENSENUMBER	EMAILADDRESS
1	Shanahan, McClure and Cassin	10	qlumsdale9@howstuffworks.com
2	Runolfsdottir-Hirthe	100	dbondy2r@gov.uk
3	Marks, Walker and Heller	1000	varboinrr@vinaora.com
4	Bartell-Mann	101	mbarabich2s@blogspot.com
5	Ebert-Dickinson	102	cguyet2t@businessweek.com
6	Yost Inc	104	garnoud2v@cisco.com
7	Kilback Inc	105	csonley2w@netvibes.com
8	Rice LLC	106	acaesman2x@narod.ru
9	Mohr-Nolan	108	dstaddart2z@live.com
10	Ankunding, Lakin and Deckow	109	kmattiassi30@reverbnation.com
11	Leffler LLC	11	tpontefracta@unesco.org
12	Halvorson and Sons	110	csommerlin31@usgs.gov
13	Bartoletti, Brakus and Strosin	111	nburborough32@addtoany.com
14	Gutmann-Emmerich	112	lvanderhoven33@ucoz.ru
15	Gerhold-Konopelski	113	gduddin34@who.int
16	Goodwin-Steuber	114	ocheetam35@miibeian.gov.cn
17	Stokes Inc	115	gmoakes36@ehow.com
18	Glover-Feeney	117	cmarzele38@ezinearticles.com
19	Von Group	118	slapworth39@purevolume.com
20	Boyer-McGlynn	119	gbrookhouse3a@army.mil
21	Dicki LLC	12	cwaywellb@t-online.de
22	Gleason-Lindgren	120	bmowett3b@youku.com
23	Green-O'Conner	121	sbather3c@meetup.com
24	Emard Inc	122	atuson3d@usa.gov
25	Daniel, Schultz and Friszon	124	educkat3f@a.co

Sasegal@labdbwin 820 rows selected in 0.306 seconds



- אחראי הרכש מעוניין לדעת אילו הזמנות הוזמנו על ידו ועדיין לא הגיעו, כדי שיוכל לנהל את המלאי השוטף של גן החיות. לכן יצרנו view שמחזיר רשימה של כל ההזמנות שעדיין לא הגיעו:

```
SELECT * FROM not_arrived_orders;
```

	ORDERID	DATEOFORDERPLACEMENT	DATEOFPLANNEDARRIVAL	LICENSENUMBER
1	10000	16/05/2017 19:07:31	27/09/2020 17:41:07	472
2	10001	23/05/2017 00:10:25	14/07/2019 21:29:46	71
3	10002	01/08/2016 21:52:59	21/07/2020 18:50:43	173
4	10003	07/09/2015 12:41:45	11/12/2018 19:38:37	55
5	10004	22/11/2015 10:36:05	14/04/2021 18:06:33	834
6	10005	18/07/2015 13:46:29	26/04/2021 02:00:01	394
7	10006	25/02/2016 00:25:24	10/11/2018 08:29:43	413
8	10007	11/12/2017 04:09:22	14/12/2019 07:24:21	378
9	10008	29/10/2016 12:18:20	09/11/2018 02:06:08	433
10	10009	09/06/2016 06:32:07	26/06/2020 17:46:04	434
11	10010	23/04/2018 17:51:12	11/03/2021 05:32:57	352
12	10011	03/10/2017 11:36:17	23/03/2021 05:33:00	709
13	10012	13/08/2015 08:40:31	11/07/2020 11:54:23	333
14	10013	21/03/2018 12:52:05	09/08/2019 00:29:31	679
15	10014	05/08/2015 22:53:43	11/05/2019 04:13:21	584
16	10015	04/05/2016 19:23:46	25/02/2020 19:25:41	865
17	10016	15/01/2016 17:53:22	05/09/2020 16:27:02	240
18	10017	11/02/2016 15:20:59	29/06/2019 17:10:20	311
19	10018	10/06/2015 18:43:42	31/12/2020 05:57:42	221
20	10019	04/10/2016 10:43:44	06/03/2020 14:15:12	103
21	10020	14/02/2018 18:55:38	19/11/2018 10:35:38	763
22	10021	20/06/2017 04:33:03	01/12/2018 04:10:10	160
23	10022	14/01/2017 03:19:44	06/03/2020 07:17:20	344

1:34 Sasegal@labdbwin 23 rows selected in 0.049 seconds (more...)

## 2. ספק:

- על הספק לדעת אילו הזמנות עדיין לא סיפק לגן החיות על מנת לספקן בעתיד, ליצור קשר עם גן החיות בנושא, ולמנוע מצב של תביעה מצד גן החיות. לכן, יצרנו view המראה את כל שמות הספקים ואת מספר הרישיון שלהם ביחד עם מספרי הזמנותיהם שעדיין לא סופקו. אין צורך לספק לכולם את כל המידע על ההזמנות, וספק יוכל לבדוק מה תוכן ההזמנה לפי מספר ההזמנה:

```
SELECT * FROM not_arrived_orders_suppliers;
```

	NAME	LICENSENUMBER	ORDERID
1	Homenick, Purdy and Dicki	1	10790
2	Homenick, Purdy and Dicki	1	13744
3	Homenick, Purdy and Dicki	1	15371
4	Homenick, Purdy and Dicki	1	15770
5	Homenick, Purdy and Dicki	1	16610
6	Homenick, Purdy and Dicki	1	16902
7	Homenick, Purdy and Dicki	1	16928
8	Homenick, Purdy and Dicki	1	18958
9	Homenick, Purdy and Dicki	1	19155
10	Homenick, Purdy and Dicki	1	19994
11	Homenick, Purdy and Dicki	1	5470
12	Homenick, Purdy and Dicki	1	6495
13	Homenick, Purdy and Dicki	1	7227
14	Homenick, Purdy and Dicki	1	7951
15	Homenick, Purdy and Dicki	1	8505
16	Homenick, Purdy and Dicki	1	8546
17	Shanahan, McClure and Cassin	10	10964
18	Shanahan, McClure and Cassin	10	11444
19	Shanahan, McClure and Cassin	10	11617
20	Shanahan, McClure and Cassin	10	14748
21	Shanahan, McClure and Cassin	10	16630
22	Shanahan, McClure and Cassin	10	17484
23	Shanahan, McClure and Cassin	10	17632
24	Shanahan, McClure and Cassin	10	18198
25	Shanahan, McClure and Cassin	10	18626

1:44 0:05 Sasegal@labdbwin 15001 rows selected in 5.684 seconds

- הספק צריך לדעת כמה הזמנות סיפק לגן החיות על מנת להכניס לדוח החודשי שלו בנושא. לכן, יצרנו view המחזיר את כל שמות הספקים ואת מספר הרשיון שלהם ביחד עם כמות ההזמנות שסיפקו:

```
SELECT * FROM suppliers_with_amount;
```

	NAME	LICENSENUMBER	AMOUNTOFORDERS
1	Homenick, Purdy and Dicki	1	22
2	Shanahan, McClure and Cassin	10	18
3	Runolfsdottir-Hirthe	100	17
4	Marks, Walker and Heller	1000	17
5	Bartell-Mann	101	18
6	Ebert-Dickinson	102	23
7	Tillman, Kuvalis and Larkin	103	21
8	Yost Inc	104	28
9	Kilback Inc	105	16
10	Rice LLC	106	16
11	Grady, Yundt and Swaniawski	107	14
12	Mohr-Nolan	108	18
13	Ankunding, Lakin and Deckow	109	25
14	Leffler LLC	11	23
15	Halvorson and Sons	110	14
16	Bartoletti, Brakus and Strosin	111	14
17	Gutmann-Emmerich	112	18
18	Gerhold-Konopelski	113	18
19	Goodwin-Steuber	114	20
20	Stokes Inc	115	16
21	Rice, Lang and Fay	116	19
22	Glover-Feeney	117	16
23	Von Group	118	20
24	Boyer-McGlynn	119	21
25	Dick LLC	12	12

Sasegal@labdbwin 1000 rows selected in 0.415 seconds

## פונקציות

- amountOfOrders - הפונקציה מקבלת מספר הרישיון של ספק (string) ומחזירה את כמות ההזמנות שהוזמנו ממנו (integer). אם זה אינו מספר של ספק במערכת או שאין הזמנות מהספק הנ"ל, הפונקציה מחזירה 0. מידע זה רלוונטי הן לספק והן לגן החיות כמו שהסברנו בקובץ למעלה:

```

1 begin
2   -- Call the function
3   :result := amountoforders(licensenum1 => :licensenum1);
4 end;
```

Variable	Type	Value
result	Float	23
licensenum1	String	43

5:1 Sasegal@labdbwin Executed in 0.01 seconds

- priceByOrder - הפונקציה מקבלת מספר הזמנה (string), ומחזירה את עלות ההזמנה (float). אם אין הזמנה בעלת המספר או שההזמנה ריקה, הפונקציה מחזירה 0. חשוב לגן החיות (הלקוח) וגם לספק (מוכר) לדעת מהי העלות של כל הזמנה:

```

1 begin
2   -- Call the function
3   :result := pricebyorder(id1 => :id1);
4 end;
```

Variable	Type	Value
result	Float	52033.24
id1	String	1

1:1 Sasegal@labdbwin Executed in 0.017 seconds



## פרוצדורות

יצרנו 2 פרוצדורות שונות המשקפות שני שימושים שניתן לממש על ידי פרוצדורה:

- `addSupplier` - הפרוצדורה הנ"ל משקפת את היכולת לשנות את בסיס הנתונים ע"י פרוצדורה, וזאת לעומת הפונקציה שאינה משנה את נתוני בסיס הנתונים. פרוצדורה זו מקבלת את כל ערכי ההכנסה הנדרשים עבור יצירת ספק (שם, מספר רישיון, מספר טלפון, כתובת מייל וכתובת) ומנסה להוסיף לבסיס הנתונים. במצב הצלחה, בערך `added` (מסוג `out`) יוחזר `true`. במצב של שגיאה (מספר רישיון כבר קיים או אי התאמת נתונים) בערך `added` יוחזר `false` וכן אנו מדפיסים הודעת שגיאה למסך ה-DBMS output:  
 ○ מצב הצלחה:

```

1 declare
2   -- Boolean parameters are translated from/to integers:
3   -- 0/1/null <--> false/true/null
4   added boolean;
5 begin
6   -- Call the procedure
7   addsupplier(suppliername => :suppliername,
8               supplierlnum => :supplierlnum,
9               phnum => :phnum,
10              email => :email,
11              addr => :addr,
12              added => added);
13   -- Convert false/true/null to 0/1/null
14   :added := sys.diutil.bool_to_int(added);
15 end;
```

	Variable	Type	Value
<input checked="" type="checkbox"/>	suppliername	String	Mr. Yair Goldshtein
<input checked="" type="checkbox"/>	supplierlnum	String	2121212
<input checked="" type="checkbox"/>	phnum	String	0556467803
<input checked="" type="checkbox"/>	email	String	DBProject@g.jct.ac.il
<input checked="" type="checkbox"/>	addr	String	Hava'ad Haleumi 21
<input checked="" type="checkbox"/>	added	Integer	1
*			

ניתן לראות כי ערכים אלו אכן נוספו לבסיס הנתונים (שורה 129):

127	Pfeffer LLC	...	211	7561435364	fgorey5u@theatlantic.com	...	5 Basil Lane
128	Frami-Bayer	...	212	5792106652	jcuffe5v@mit.edu	...	78 La Follette Road
129	Mr. Yair Goldshtein	...	2121212	0556467803	DBProject@gjct.ac.il	...	Hava'ad Haleumi 21
130	Rohan, Streich and Bartoletti	...	213	9938601683	lbattey5w@fda.gov	...	304 Rowland Lane
131	Schaden and Sons	...	214	7822750227	adoggrell5x@edublogs.org	...	5 Sundown Park
132	Cormier, Stehr and Tillman	...	215	7885845635	lclemence5y@exblog.jp	...	5784 Canary Circle
133	Nicolas and Sons	...	216	9523747214	dgoalby5z@fda.gov	...	40695 Valley Edge Circle
134	Kuphal and Sons	...	217	2915367357	mantowski60@gov.uk	...	67644 Forest Dale Alley
135	Strosin LLC	...	218	7461291148	rhabron61@fastcompany.com	...	53 Meadow Valley Crossing

○ מצב תקלה (ספק בעל מספר רישיון 33333333 כבר קיים בבסיס הנתונים):

```

1 declare
2   -- Boolean parameters are translated from/to integers:
3   -- 0/1/null <--> false/true/null
4   added boolean;
5 begin
6   -- Call the procedure
7   addsupplier(suppliername => :suppliername,
8               supplierlnum => :supplierlnum,
9               phnum => :phnum,
10              email => :email,
11              addr => :addr,
12              added => added);
13   -- Convert false/true/null to 0/1/null
14   :added := sys.diutil.bool_to_int(added);
15 end;
```

<input type="checkbox"/>	Variable	Type	Value
<input checked="" type="checkbox"/>	suppliername	String	jkhhn
<input checked="" type="checkbox"/>	supplierlnum	String	33333333
<input checked="" type="checkbox"/>	phnum	String	hjj
<input checked="" type="checkbox"/>	email	String	
<input checked="" type="checkbox"/>	addr	String	uigiukb
<input checked="" type="checkbox"/>	added	Integer	0

1:1 Sasegal@labdbwin Executed in 0.004 seconds

למך מודפס הודעת שגיאה:

Test script DBMS Output Statistics Profiler Trace

Clear Buffer size 10000 ☒ Enabled

ERROR: input was invalid or unique constraint violated

- priceFromYear - פרוצדורה זו מקבלת מחרוזת של שנה ומחזירה דרך ערך - price (מסוג out) את הסך הכולל של מחירי ההזמנות שהגיעו בשנה זו והלאה. בפרוצדורה זו אנו עושים שימוש בפונקציה priceOrderBy. דבר זה משקף את היכולת להשתמש בפונקציות כחלק מפרוצדורה (לעומת זאת, לא ניתן לקרוא לפרוצדורה מפונקציה):

```

1 begin
2   -- Call the procedure
3   pricefromyear(year => :year,
4                 price => :price);
5 end;
```

<input type="checkbox"/>	Variable	Type	Value
<input checked="" type="checkbox"/>	year	String	2018
<input checked="" type="checkbox"/>	price	String	120518596.19
<input checked="" type="checkbox"/>			

 1:1      0:01      ▼ Sasegal@labdbwin      ➦ Executed in 1.831 seconds

## נספחים

קוד יצירת הטבלאות:

```
CREATE TABLE Supplier
```

```
(
  name CHAR(50) NOT NULL,
  licenseNumber CHAR(10) NOT NULL,
  phoneNumber CHAR(10) NOT NULL,
  emailAddress CHAR(50),
  address CHAR(50) NOT NULL,
  PRIMARY KEY (licenseNumber)
```

```
);
```

```
CREATE TABLE OrderTable
```

```
(
  orderID CHAR(10) NOT NULL,
  dateOfOrderPlacement DATE NOT NULL,
  dateOfPlannedArrival DATE NOT NULL,
  licenseNumber CHAR(10) NOT NULL,
  PRIMARY KEY (orderID),
  FOREIGN KEY (licenseNumber) REFERENCES Supplier(licenseNumber)
```

```
);
```

```
CREATE TABLE Product
```

```
(
  barcode CHAR(15) NOT NULL,
  name CHAR(50) NOT NULL,
  pricePerUnit NUMERIC(12, 2) NOT NULL,
  PRIMARY KEY (barcode)
```

```
);
```

```
CREATE TABLE Receipt
```

```
(
  receiptID CHAR(10) NOT NULL,
  dateOfArrival DATE NOT NULL,
  orderID CHAR(10) NOT NULL,
  PRIMARY KEY (receiptID),
  FOREIGN KEY (orderID) REFERENCES OrderTable(orderID)
```

```
);
```

```
CREATE TABLE ProductInOrder
```

```
(
  amount INT NOT NULL,
  orderID CHAR(10) NOT NULL,
  barcode CHAR(15) NOT NULL,
  PRIMARY KEY (orderID, barcode),
  FOREIGN KEY (orderID) REFERENCES OrderTable(orderID),
  FOREIGN KEY (barcode) REFERENCES Product(barcode)
```

```
);
```



קוד השאילתות:

```
SELECT DISTINCT name AS supplierName, licenseNumber
FROM supplier NATURAL JOIN receipt r NATURAL JOIN ordertable o
WHERE r.dateofarrival > o.dateofplannedarrival
GROUP BY name, licenseNumber
ORDER BY name;
```

```
SELECT orderid, sum(price) AS totalPrice
FROM (SELECT priceperunit * amount AS price, orderid FROM product NATURAL
JOIN productinorder)
GROUP BY orderid;
```

```
SELECT supplierName, licensenumber, sum(totalprice) AS totalyearlyprice
FROM(SELECT dateofarrival, name AS suppliername, licensenumber, totalprice
FROM(SELECT orderid, sum(price) AS totalPrice
FROM (SELECT priceperunit * amount AS price, orderid FROM product NATURAL
JOIN productinorder)
GROUP BY orderid) NATURAL JOIN ordertable NATURAL JOIN supplier NATURAL
JOIN receipt
WHERE TO_DATE('2018/01/01 00:00:00', 'yyyy/mm/dd hh24:mi:ss') <=
dateofarrival AND
TO_DATE('2019/01/01 00:00:00', 'yyyy/mm/dd hh24:mi:ss') >= dateofarrival)
GROUP BY suppliername, licensenumber;
```

```
SELECT orderid, receiptid, dateofplannedarrival, dateoforderplacement,
dateofarrival
FROM ordertable NATURAL JOIN receipt
WHERE dateoforderplacement<=dateofarrival AND
dateofarrival<=dateofplannedarrival
GROUP BY orderid, receiptid, dateofplannedarrival, dateoforderplacement,
dateofarrival
ORDER BY dateofarrival;
```

```
SELECT DISTINCT orderid, name AS suppliername, licensenumber, dateofarrival,
dateoforderplacement
FROM receipt NATURAL JOIN ordertable NATURAL JOIN supplier;
```

```
SELECT *
FROM product
WHERE priceperunit>=ALL(SELECT DISTINCT priceperunit FROM product);
```

```
SELECT name AS suppliername, licenseNumber, count(orderid) AS
numberOfOrders
FROM supplier NATURAL JOIN ordertable
GROUP BY name, licenseNumber;
```

```
SELECT orderid,amount,priceperunit, barcode, name
FROM (SELECT * FROM productinorder WHERE amount > 10 AND amount < 100)
```

NATURAL JOIN ordertable NATURAL JOIN product;

קוד האינדקסים:

CREATE INDEX amount_index ON ProductInOrder(amount);	DROP INDEX amount_index ON ProductInOrder;
CREATE INDEX productName_index ON Product(name);	DROP INDEX productName_index ON Product;
CREATE INDEX date_index ON Receipt(dateOfArrival);	DROP INDEX date_index ON Receipt(dateOfArrival);

קוד פקודות GRANT:

GRANT SELECT ON Product TO ASLEVI;
GRANT REFERENCES ON Product TO ASLEVI;

קוד יצירת טבלה חדשה אצל ASLEVI בשלב האינטגרציה:

<pre>CREATE TABLE Supply (   amount INT NOT NULL,   ID NUMBER(10) NOT NULL,   barcode CHAR(15) NOT NULL,   upToDate DATE NOT NULL,   PRIMARY KEY (ID, barcode, upToDate),   FOREIGN KEY (ID) REFERENCES SALES_STAND(ID),   FOREIGN KEY (barcode) REFERENCES SASEGAL.Product(barcode) );</pre>
---

קוד שאילתות האינטגרציה:

<pre>SELECT DISTINCT name, barcode FROM product NATURAL JOIN ASLEVI.supply GROUP BY name, barcode ORDER BY barcode;</pre>
<pre>SELECT name, barcode, upToDate FROM product NATURAL JOIN ASLEVI.supply WHERE TO_DATE('2018/01/01 00:00:00', 'yyyy/mm/dd hh24:mi:ss') &lt;= upToDate AND TO_DATE('2019/01/01 00:00:00', 'yyyy/mm/dd hh24:mi:ss') &gt;= upToDate GROUP BY upToDate, barcode, name ORDER BY upToDate, barcode;</pre>

```
SELECT p.name, barcode, count(id) AS amountofstands
FROM (SELECT DISTINCT id, barcode FROM ASLEVI.supply) NATURAL JOIN product p
GROUP BY barcode, p.name
ORDER BY barcode;
```

קוד יצירת VIEWS:

```
CREATE OR REPLACE VIEW suppliers_with_email AS
SELECT name, licenseNumber, emailAddress
FROM supplier
WHERE emailAddress IS NOT NULL
GROUP BY licenseNumber, name, emailAddress
ORDER BY licenseNumber;
```

```
CREATE OR REPLACE VIEW not_arrived_orders AS
SELECT orderid, dateoforderplacement, dateofplannedarrival, licensenumber
FROM ordertable
WHERE orderid NOT IN(SELECT orderid
FROM ordertable NATURAL JOIN receipt)
GROUP BY orderid, dateoforderplacement, dateofplannedarrival, licensenumber
ORDER BY orderid;
```

```
CREATE OR REPLACE VIEW not_arrived_orders_suppliers AS
SELECT name, licenseNumber, orderid
FROM ordertable NATURAL JOIN supplier
WHERE orderid NOT IN(SELECT orderid
FROM ordertable NATURAL JOIN receipt)
GROUP BY name, licenseNumber, orderid
ORDER BY licenseNumber, orderid;
```

```
CREATE OR REPLACE VIEW suppliers_with_amount AS
SELECT name, licenseNumber, COUNT(*) AS amountOfOrders
FROM supplier NATURAL JOIN ordertable
GROUP BY licenseNumber, name
ORDER BY licenseNumber;
```

קוד יצירת הפונקציות:

```
CREATE OR REPLACE FUNCTION amountOfOrders(licenseNumber1 IN CHAR) RETURN
INTEGER IS
    FunctionResult INTEGER;
BEGIN
    SELECT COUNT(*)
    INTO FunctionResult
    FROM supplier NATURAL JOIN ordertable
    WHERE licenseNumber = licenseNumber1;
    RETURN(FunctionResult);
```

```
END amountOfOrders;
/
```

```
CREATE OR REPLACE FUNCTION priceByOrder(id1 in char) RETURN NUMBER IS
  FunctionResult NUMBER;
BEGIN
  SELECT SUM(priceperunit * amount)
  INTO FunctionResult
  FROM productinorder NATURAL JOIN product
  WHERE orderid = id1;
  RETURN(FunctionResult);
END priceByOrder;
/
```

קוד יצירת הפרוצדורות:

```
CREATE OR REPLACE PROCEDURE addSupplier(supplierName IN CHAR, supplierLNum IN
CHAR, phNum IN CHAR, email IN CHAR, addr IN CHAR, added OUT BOOLEAN) IS
BEGIN
  INSERT INTO supplier VALUES (supplierName, supplierLNum, phNum, email, addr);
  added := TRUE;
EXCEPTION
  WHEN OTHERS THEN dbms_output.put_line('ERROR: input was invalid or unique
constraint violated');
  added := FALSE;
END addSupplier;
/
```

```
CREATE OR REPLACE PROCEDURE priceFromYear(year IN CHAR, price OUT CHAR) IS
BEGIN
  SELECT SUM(pricebyorder(orderid))
  INTO price
  FROM receipt NATURAL JOIN ordertable
  WHERE dateofarrival >= TO_DATE(CONCAT(year,'/01/01 00:00:00'), 'yyyy/mm/dd
hh24:mi:ss');
END priceFromYear;
/
```