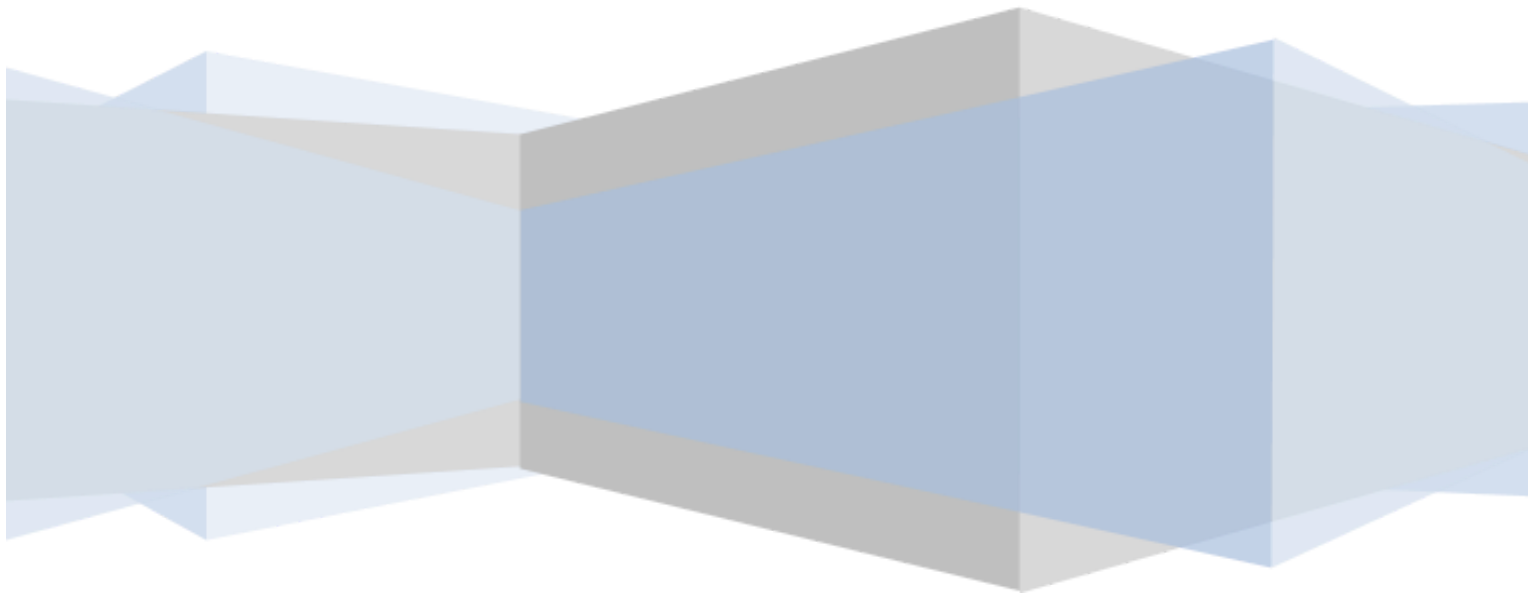


מיני פרויקט בבסיסי נתונים



מגישים: יחזקאל בן עטר 318834579

מיידן סולומון אלגרסי 303159628

תוכן עניינים

<u>3</u>	<u>מבוא</u>
<u>5</u>	<u>עבודת הכנה והכרת התוכנה</u>
5	תרשים ERD
5	תיאור הישויות והקשרים
5	ישויות
6	קשרים
6	נרמול הטבלאות
6	פרוקים
7	תרשים DSD
8	<u>הפרויקט שלנו</u>
8	תרשים ERD
8	תיאור הישויות והקשרים
8	ישויות
9	קשרים
9	נרמול הטבלאות
10	תרשים DSD
11	יצירת הטבלאות
12	הכנסת נתונים
14	שאלות SQL
14	בחירה - SELECT
22	מחיקה - DELETE
22	הכנסה - INSERT
23	אינדקסים
29	שאלות שנותנות מידע מחיבור של אגפים שונים
31	הרשאות
32	VIEWS
34	תרשימים
35	פונקציות
40	<u>נספחים</u>
40	נספח ראשון: שאלות ואינדקסים
43	נספח שלישי: שאלות אינטגרציה
43	נספח רביעי: VIEWS
44	נספח חמישי: פונקציות

מבוא

• מחלקת לוגיסטיקה

כלי רכב: מס' רכב, חברה, דגם, רישיון, שנת ייצור, צבע.

ציוד תפעול ותחזוקה: מזהה, שם, מחלקה.

מזון: ברקוד, שם, כמות, ערכים תזונתיים.

• מנהלה ועובדים

איש צוות: מזהה, מספר עובד, מחלקה, חשבון בנק, משכורת, ותק.

קבוצות: מזהה, שם, תיאור, תקציב, מזהה של מלון.

מחלקה: מזהה מחלקה, שם, מספר מנהל.

• מחלקת מבקרים

מבקר: ת.ז, מספר כרטיס, אופן תשלום.

כרטיס: מזהה, סוג, מחיר, תאריך הנפקה.

דוכני מכירות: שם, ת.ז מנהל, רישיון.

• מחלקת אספקה

ספק: מזהה, טלפון, רישיון.

הזמנה: מספר הזמנה, מוצר, כמות, מספר ספק, תאריך.

• מחלקת חיות הגן

כלוב: מזהה, מזהה מתחם, מיקום, גודל, מזהה חיה, כמות פרטים.

חיה: מס' זהות, שם, סיווג מדעי, תנאי מחיה, גיל, ארץ מוצא, מזון, תיק רפואי.

אזור: מס' אזור, מיקום אזור, שם אזור.

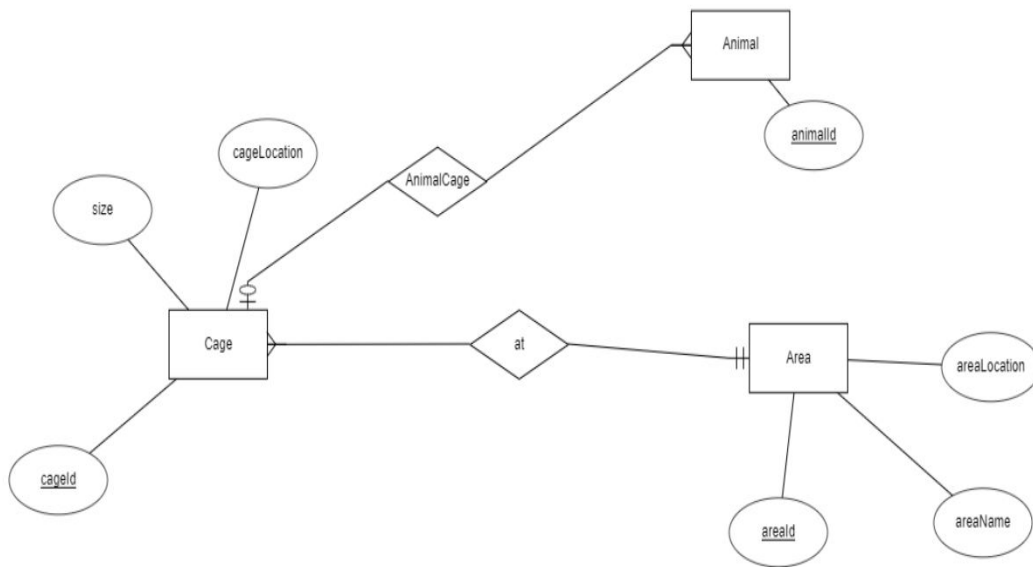
• מחלקת רפואה

תיק רפואי: מספר תיק, מחלות, תרופות שנוטל, עבר רפואי.

תרופה: מזהה, שם, תאריך תוקף.

עבודת הכנה והכרת התוכנה

ERD תרשים



תיאור הישויות והקשרים

ישויות

- Area - מאופיין ב- מספר אזור, שם אזור ומיקום אזור.
- Cage - מאופיין ב- מספר כלוב, מספר אזור, מיקום הכלוב וגודל הכלוב.
- AnimalCage - מאופיין ב- מספר חיה ומספר הכלוב.

קשרים

- לכל חיה- יש מקסימום כלוב אחד, זאת אומרת יתכן וחיה תהיה בכלוב ויתכן ולא תהיה בכלוב בכלל.
- לכל כלוב- 1. יכולים להיות כמה חיות, זאת אומרת שמספר חיות יכולות להיות בכלוב 2 . חייב להיות באזור אחד .
- לכל אזור- יכולים להיות כמה כלובים, זאת אומרת שמספר כלובים יכולים להיות באזור אחד.

נרמול הטבלאות

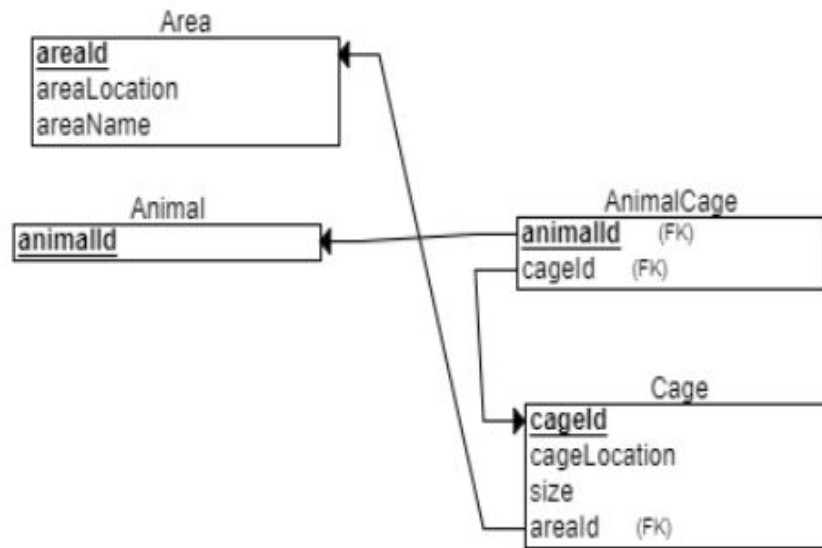
- Area (areaLocation, areaName, areald)
- Cage(cageID, cageLocation, cageSize, areald)
- AnimalCage (animakld, cageId)

פרקים

היחסים עומדים ב- NF3 וב- BCNF : מכיון שבכל טבלה, התלויות הפונקציונאליות הלא-טריוויאליות הן מהמפתח אל תכונות נוספות לכן מתקיים שלכל $X \rightarrow Y$, הוא מפתח ולכן הם עומד בתנאים.

DSD תרשים

כאן ניתן לראות את ה-DSD שהפקנו מתרשים ה-ERD שיצרנו.

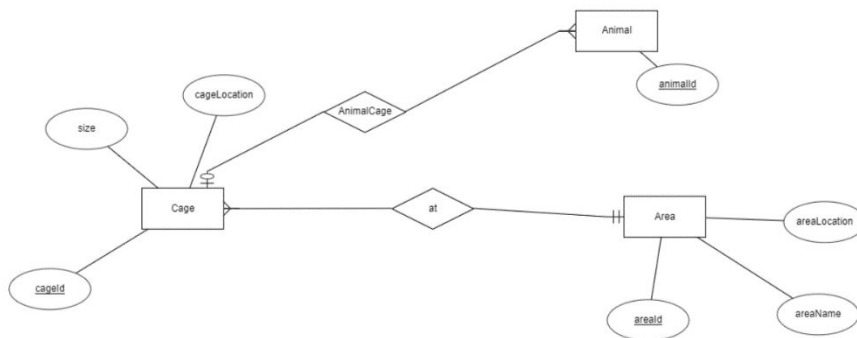


הפרויקט שלנו

הסבר אנחנו מתעסקים בחלק של אכלוס החיות בו אנחנו משבצים את החיות בכלובים ואזורים מסוימים.

תרשים ERD

כפי שהזכרנו, במחלקה שלנו, ישנן 3 ישויות: אזורים, כלובים וטבלה שמקשרת בין חיה וכלוב. בשלב הראשון יצרנו תרשים ERD שיתאר את הקשרים בין הישויות הללו ואת התכונות שלהן (הישות animal שייכת למחלקה אחרת היא רק הובאה כאן בשביל ההקשר בין הטבלאות הנ"ל).



תיאור הישויות והקשרים

ישויות

- Area – ישות זאת אחראית על כל האזורים הנמצאים במערכת. ישות זאת הינה חזקה, כיוון שיכולה להתקיים ללא תלות בישות אחרת.

- o area_id – מספר מזהה של האזור (PK)
- o area_name – שם האזור
- o area_location – מיקום האזור

- Cage – ישות זאת אחראית על כל הכלובים הנמצאים במערכת. ישות זאת הינה חזקה, כיוון שיכולה להתקיים ללא תלות בישות אחרת.

- o cage_id – מספר מזהה של הכלוב (PK)
- o cage_location – מיקום הכלוב
- o size – גודל בשטח מרובע של הכלוב
- o area_id – האזור שבו נמצא הכלוב (FK)

- AnimalCage – ישות זאת אחראית על שילוב של חיות בכלובים. ישות זאת הינה חלשה, כיוון שהיא תלויה בין היתר בחיות ובכלובים.

- animal_id – מספר מזהה של החיה (FK)(PK)
- cage_id – מספר מזהה של הכלוב (FK)

קשרים

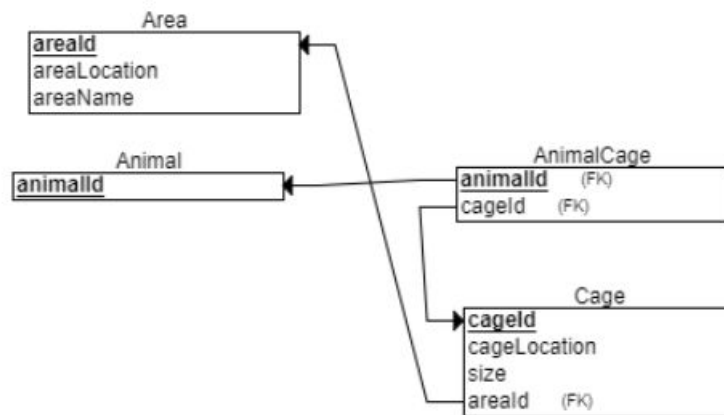
- At – הקשר בין cage לבין area. הקשר הינו חלש, כיוון שמקשר בין ישות חלשה לחזקה. הקשר הוא M:1 משום שיכול להיות לאזור אחד הרבה כלובים, אבל לכלוב אחד שייך לאזור אחד.
- AnimalCage – הקשר בין animal (שלא שייך למחלקה שלנו) לבין cage. הקשר הינו בעצם אחד מהטבלאות אליהם האגף אחראי. הקשר הוא M:1 משום שיכול להיות בכלוב אחד הרבה חיות, אבל חיה אחת שייכת לכלוב אחד בלבד.

נרמול הטבלאות

היחסים עומדים ב-NF3 וב-BCNF: מכיוון שבכל טבלה, התלויות הפונקציונאליות הלא-טריוויאליות הן מהמפתח אל תכונות נוספות לכן מתקיים שלכל $X \rightarrow Y$ הוא מפתח ולכן הם עומד בתנאים.

תרשים DSD

על פי תרשים ה- ERD ועל ידי הבנת הקשרים בין הישויות, יצרנו תרשים DSD עבור החלק שלנו במערכת: מחלקת מתקני הגן חיות(יש לציין שהישות animal הובאה רק לצורך הסבר התרשים אך היא אינה שייכת למחלקה).



יצירת הטבלאות

אחרי שהבנו כיצד בסיס הנתונים צריך להראות בצורה מדויקת, מה תכיל כל טבלה ומהם הקשרים בין כל הטבלאות, ניגשנו ליצירת הטבלאות בפועל בעזרת פקודות create table.

יצרנו קוד לייצור הטבלאות באמצעות export SQL של האתר erdPlus, יצרנו קובץ SQL ואז העתקנו את קוד ה-SQL של כל טבלה אל תוכנת ה-plsql לשם יצירת הטבלאות בפועל.

```
CREATE TABLE Area
(
    areaLocation varchar(30) NOT NULL,
    areaName varchar(30) NOT NULL,
    areaId INT NOT NULL,
    PRIMARY KEY (areaId)
);

CREATE TABLE Cage
(
    cageId INT NOT NULL,
    cageLocation varchar(30) NOT NULL,
    cageSize INT NOT NULL,
    areaId INT NOT NULL,
    PRIMARY KEY (cageId),
    FOREIGN KEY (areaId) REFERENCES Area(areaId)
);

CREATE TABLE AnimalCage
(
    animalId INT NOT NULL,
    cageId INT NOT NULL,
    PRIMARY KEY (animalId),
    FOREIGN KEY (animalId) REFERENCES shectma.animal(animalId),
    FOREIGN KEY (cageId) REFERENCES Cage(cageId)
);
```

הכנסת נתונים

על מנת לאכלס את הטבלאות שיצרנו בנתונים, השתמשנו ב data generator .

עבור הטבלה AnimalCage

ANIMALCAGE					
Owner		Table		Number of records	
ALGRSI		ANIMALCAGE		6600	
Name	Type	Size	Data	Master	
ANIMALID	NUMBER		List(select animalid from shechtma.animal)	---	▼
CAGEID	NUMBER		List(select cageid from algrsi.cage)	---	▼
*					

עבור הטבלה Cage

CAGE					
Owner		Table		Number of records	
ALGRSI		CAGE		10000	
Name	Type	Size	Data	Master	
CAGEID	NUMBER		Sequence(11111, [1], [99999])	---	▼
LOCATIONCAGE	VARCHAR2	30	Address2	---	▼
SIZECAGE	NUMBER		Random(10, 1000)	---	▼
AREAD	NUMBER		List(select areaid from area)	---	▼
*					

עבור הטבלה Area

AREA					
Owner	Table		Number of records		
ALGRSI	AREA		10000		
Name	Type	Size	Data		Master
AREALOCATION	VARCHAR2	30	Address1	...	▼
AREANAME	VARCHAR2	30	LastName	...	▼
AREAID	NUMBER		Sequence(11111, [1], [99999])	...	▼
*				...	▼

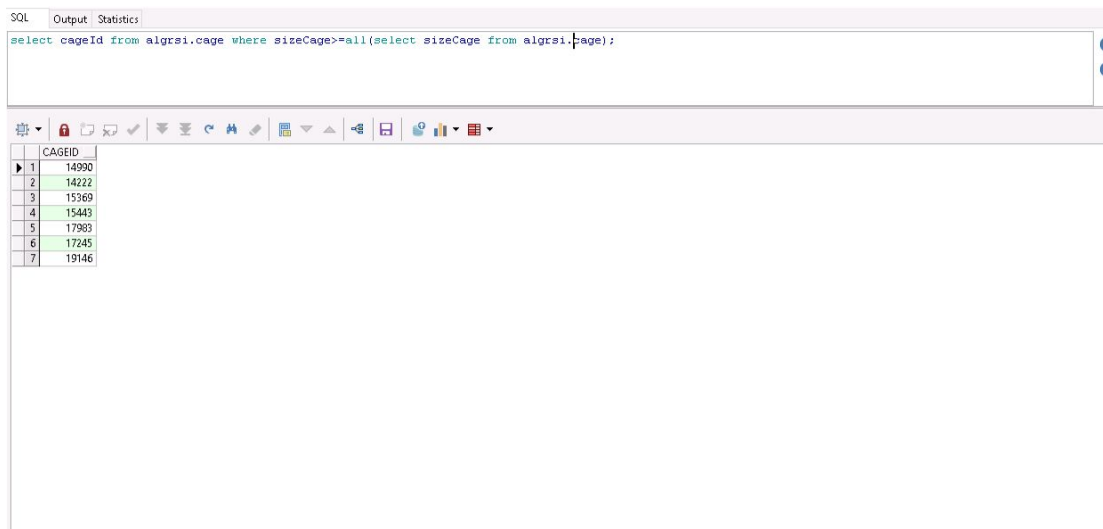
שאלות SQL

בחירה - SELECT

לאחר שיצרנו את בסיס הנתונים והכנסנו לתוכו מידע, כתבנו כמה שאלות מעניינות על מנת לתשאל אותו.

1. כחלק מהרצון של ההנהלה לדעת איפה צריך להשקיע את מירב המאמץ בכלובים, על החברה לדעת מי הם הכלובים עם השטח הכי גדול.

- לכן כתבנו שאלת המחזירה טבלה של כל מספרי הזיהוי של הכלובים עם השטח המקסימלי.



The screenshot shows a SQL query execution interface. The query is: `select cageId from algrs1.cage where sizeCage>=all(select sizeCage from algrs1.cage);`. The results are displayed in a table with two columns: CAGEID and sizeCage. The results are as follows:

CAGEID	sizeCage
1	14990
2	14222
3	15369
4	15443
5	17983
6	17245
7	19146

2. ההנהלה החליטה להוסיף עוד עובד נוסף בסופי שבוע לכל הכלובים מעל שטח של 800 מטר מרובע.
- לכן ההנהלה ביקשה רשימה של כל הכלובים עם שטח שגודלו מעל 800 מטר מרובע.

SQL	Output	Statistics
<pre>select cageId from algrsi.cage where sizeCage>800;</pre>		
	CAGEID	
1	11738	
2	11739	
3	11757	
4	11760	
5	11765	
6	11775	
7	11778	
8	11781	
9	11782	
10	11783	
11	11785	
12	11788	
13	11792	
14	11795	
15	11796	
16	11807	
17	11818	
18	11822	
19	11826	
20	11836	
21	11849	
22	11855	
23	11861	
24	11862	
25	11868	

1:51 ybenatar@labdbwin 28 rows selected in 0.043 seconds (more...)

3. ההנהלה רוצה את כל האזורים אשר שמם מתחיל באות מסוימת לדוגמה שמתחיל ב A.
 - לכן כתבנו שאילתה המחזירה את כל האזורים אשר שמם מתחיל באות גדולה A.

select areald from algrsi. ... X

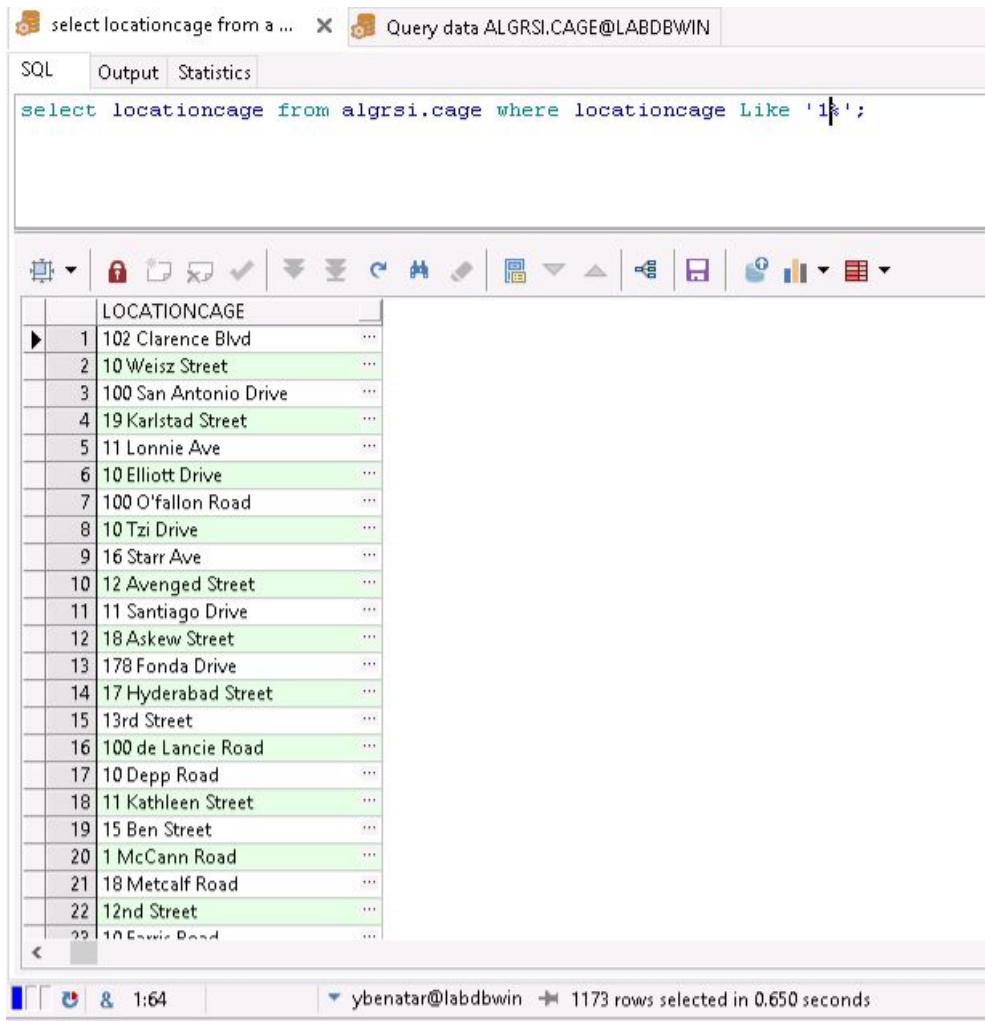
SQL Output Statistics

```
select areaId from algrsi.area where areaName Like 'A%';
```

	AREAID
1	176343
2	831720
3	982570
4	832043
5	731563
6	245574
7	191067
8	752264
9	737181
10	779610
11	424387
12	614993
13	127412
14	296771
15	208053
16	269254
17	148921
18	426357
19	259836
20	572542
21	389926
22	390612
23	893305
24	658968

1:57 ybenatar@labdbwin 941 rows selected in 0.729 seconds

4. בנוסף לשאילתה הקודמת ההנהלה רוצה את כל הכלובים אשר כתובתם מתחיל בספרה מסוימת לדוגמה 1 בשביל לעשות סדר כלשהו בהנהלת החשבונות.
- לכן כתבנו שאילתה המחזירה את כל הכלובים אשר נמצאים בכתובת שמתחילה ב1.



The screenshot shows a SQL query execution window. The query is: `select locationcage from algrsi.cage where locationcage Like '1%';`. The results are displayed in a table with 23 rows. The first column is an index (1-23) and the second column is the location name. The status bar at the bottom indicates that 1173 rows were selected in 0.650 seconds.

	LOCATIONCAGE
1	102 Clarence Blvd
2	10 Weisz Street
3	100 San Antonio Drive
4	19 Karlstad Street
5	11 Lonnie Ave
6	10 Elliott Drive
7	100 O'fallon Road
8	10 Tzi Drive
9	16 Starr Ave
10	12 Avenged Street
11	11 Santiago Drive
12	18 Askew Street
13	178 Fonda Drive
14	17 Hyderabad Street
15	13rd Street
16	100 de Lancie Road
17	10 Depp Road
18	11 Kathleen Street
19	15 Ben Street
20	1 McCann Road
21	18 Metcalf Road
22	12nd Street
23	10 Ennie Road

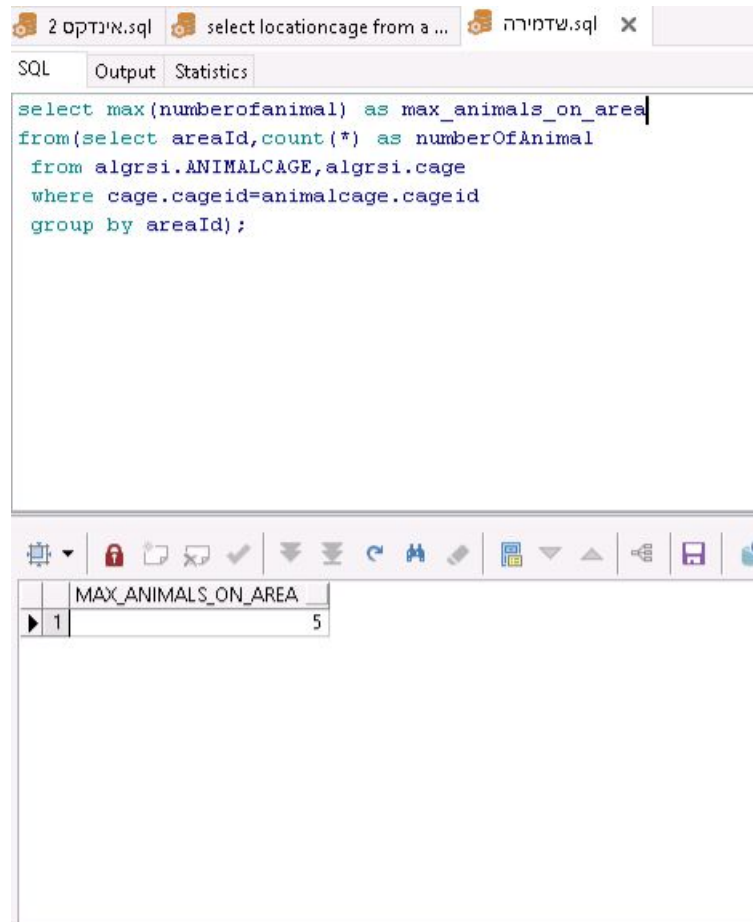
5. ההנהלה חוששת שבאזורים עם הכי הרבה חיות אין מספיק עובדי ניקיון מכיוון שהרבה מאוד מבקרים יש באזורים הללו.
- לכן כתבנו שאילתה אשר תיתן לנו את מספרי האזורים ומספרי החיות המקסימלי שיש בהם.

```
select areaId,numberofanimal
from(select areaId,count(*) as numberOfAnimal
from algrsi.ANIMALCAGE,algrsi.cage
where cage.cageid=animalcage.cageid
group by areaId)
where numberOfAnimal>=all(select numberOfAnimal from(select areaId,count(*) as numberOfAnimal
from algrsi.ANIMALCAGE,algrsi.cage
where cage.cageid=animalcage.cageid
group by areaId) );
```

	AREAIID	NUMBEROFANIMAL
1	915079	5
2	637302	5
3	505687	5
4	265490	5

ybenatar@labdbwin 4 rows selected in 0.037 seconds

6. ההנהלה יודעת כי מספר מקסימלי של חיות בכל אזור הוא 100 על כן ההנהלה רצתה לדעת מה מצבה ולכן היא רוצה שאילתה שתתן את מספר החיות המקסימלי שיש במתחם כלשהו.
- לכן כתבנו שאילתה שתתן את מספר החיות המקסימלי שיש במתחם כלשהו.



The screenshot shows a SQL IDE with three tabs: '2 אינדקס.sql', 'select locationcage from a ...', and 'שדמירה.sql'. The 'SQL' tab is active, displaying the following query:

```
select max(numberofanimal) as max_animals_on_area
from (select areaId, count(*) as numberOfAnimal
      from algrsi.ANIMALCAGE, algrsi.cage
      where cage.cageid=animalcage.cageid
      group by areaId);
```

Below the query editor is a toolbar with various icons. At the bottom, a result table is displayed with the following data:

	MAX_ANIMALS_ON_AREA
1	5

7. הנהלת הגן חיות קיבלה הזמנה של מים עבור כל החיות, ההנהלה רוצה לעדכן את המשאיות כמה לחלק בכל אזור, על כן היא צריכה לדעת כמה חיות יש בכל אזור אשר התחיל לאכלס (משמע קיים לפחות כלוב אחד באזור הזה עם חיה אחת לפחות).
- לכן כתבנו שאילתה המחזירה טבלה של מספרי אזור + כמה חיות יש בכל מתחם.

SQL	Output	Statistics
<pre>select areaId,count(*) as numberOfAnimal from algrsi.ANIMALCAGE,algrsi.cage where cage.cageid=animalcage.cageid group by areaId;</pre>		
AREAID	NUMBEROFANIMAL	
1	3	
2	2	
3	2	
4	1	
5	1	
6	1	
7	3	
8	3	
9	1	
10	1	
11	3	
12	1	

8. מס הכנסה פנה להנהלת גן החיות וביקש מידע כמה חיות יש בכל כלוב כדי לדעת כמה מס צריך לשלם על כל כלוב(המס נמדד על פי כמות החיות בכל כלוב).
- לשם כך כתבנו שאילתה הנותנת מידע כמה חיות יש בכל כלוב אשר התחיל לאכלס שם לפחות חיה אחת.

SQL Output Statistics

```
select cageId,Count(*) as NumberOfAnimals from algrsi.ANIMALCAGE group by cageId;
```

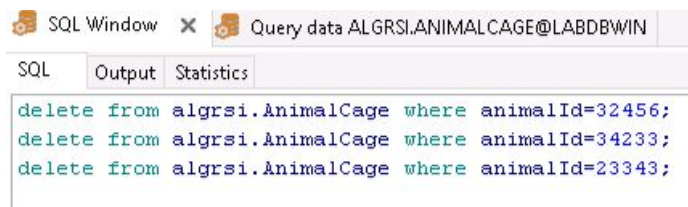
	CAGEID	NUMBEROFANIMALS
1	13070	2
2	11827	3
3	19485	1
4	20713	2
5	12864	1
6	20121	1
7	18856	1
8	15939	1
9	15983	2
10	19137	2
11	18349	2
12	18282	2

1:55 0:01 ybenatar@labdbwin 3085 rows selected in 1.527 seconds

מחיקה – DELETE

כתבנו שאילת מחיקה על מנת למחוק רשומות מבסיס הנתונים.

1. עקב המוות של שלושת האריות בעקבות מחלת הקורונה שבהם נדבקו החיות נאלצנו למחוק את החיות מהכלובים בהם היו רשומים. לכן מחקנו מבסיס הנתונים את כל האריות מהטבלה בהם היא מקשרת בינם לבין הכלוב.

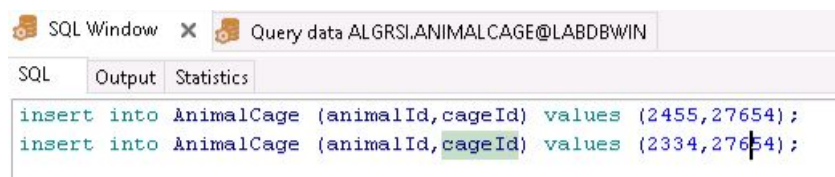


```
SQL Window x Query data ALGRSI.ANIMALCAGE@LABDBWIN
SQL Output Statistics
delete from algrsi.AnimalCage where animalId=32456;
delete from algrsi.AnimalCage where animalId=34233;
delete from algrsi.AnimalCage where animalId=23343;
```

הכנסה – INSERT

השתמשנו בשאילתות insert על מנת להכניס נתונים לטבלאות

1. בעקבות המוות הטראגי של שלושת האריות הוחלט להביא שני גורי אריה חדשים מאפריקה, על כן נצטרך להכניס אותם לכלוב כלשהו בבסיס הנתונים שלנו.



```
SQL Window x Query data ALGRSI.ANIMALCAGE@LABDBWIN
SQL Output Statistics
insert into AnimalCage (animalId,cageId) values (2455,27654);
insert into AnimalCage (animalId,cageId) values (2334,27654);
```

אינדקסים

אינדקסים עוזרים למצוא במהירות גדולה יותר נתונים שנשמרו בטבלאות בבסיס הנתונים. אפשר לדמות את האינדקסים כמו מראה מקום בספר. במקום שנקרא את כל הספר כדי למצוא את מה שאנחנו מחפשים נלך למראה מקום שיראה לנו את כל המקומות שבהם מוזכר הנושא שאנחנו מחפשים. השימוש באינדקסים יחסוך לנו זמן ויהפוך את תהליך החיפוש ליעיל יותר. מהבחינה הזו האינדקסים בטבלאות של ה-SQL זהים לאינדקס בספר.

במידה ולא נגדיר אינדקס לטבלה אז בכל שאילתה על הטבלה השאילתה תגרום למעבר על כל הרשומות בטבלה עד שתמצא את כל הרשומות העונות למה שחיפשנו. כשנגדיר אינדקס מתאים אז החיפוש יהיה מהיר יותר כי הפניה לבסיס הנתונים תגרום לזה שמנוע החיפוש בבסיס הנתונים יפנה קודם לאינדקס וילך לרשומות המתאימות על פי מה שרשום באינדקס.

לכן יצרנו אינדקסים (הקוד בנספח הראשון) שמקצרים את תהליך ביצוע השאילתות.

1. שאלנו שאילתה המחזירה את כל האזורים אשר שמם מתחיל באות A. האינדקס שיצרנו היה על הטבלה AREA בעמודה AREANAME כאשר האינדקס הוא על פי תווים המתחילים באות A. זמן השאילתה תרד כי מבחינה הגיונית כשנחפש את השמות שמתחילות ב A לא נצטרך לחפש בכל הטבלה אלא ישר נחפש באינדקס איפה שהשמות מתחילות ב A. לכן ניתן לראות כי זמן הריצה של השאילתה אחרי האינדקס ירד בצורה משמעותית.

לפני האינדקס

The screenshot shows a SQL query execution window with the following SQL query: `select areaId from algrsi.area where areaName Like 'A%';`. The results are displayed in a table with 24 rows, each containing an AREAID value. The status bar at the bottom indicates that 941 rows were selected in 0.729 seconds.

	AREAID
1	176343
2	891720
3	982570
4	832043
5	731563
6	245574
7	191067
8	752264
9	737181
10	779610
11	424387
12	614993
13	127412
14	296771
15	208053
16	269254
17	148921
18	426357
19	259836
20	572542
21	389926
22	390612
23	893305
24	658968

אחרי האינדקס

create index IDX_Area_name ... select areald from algrsi. ...

SQL Output Statistics

```
select areaId from algrsi.area where areaName Like 'A%';
```

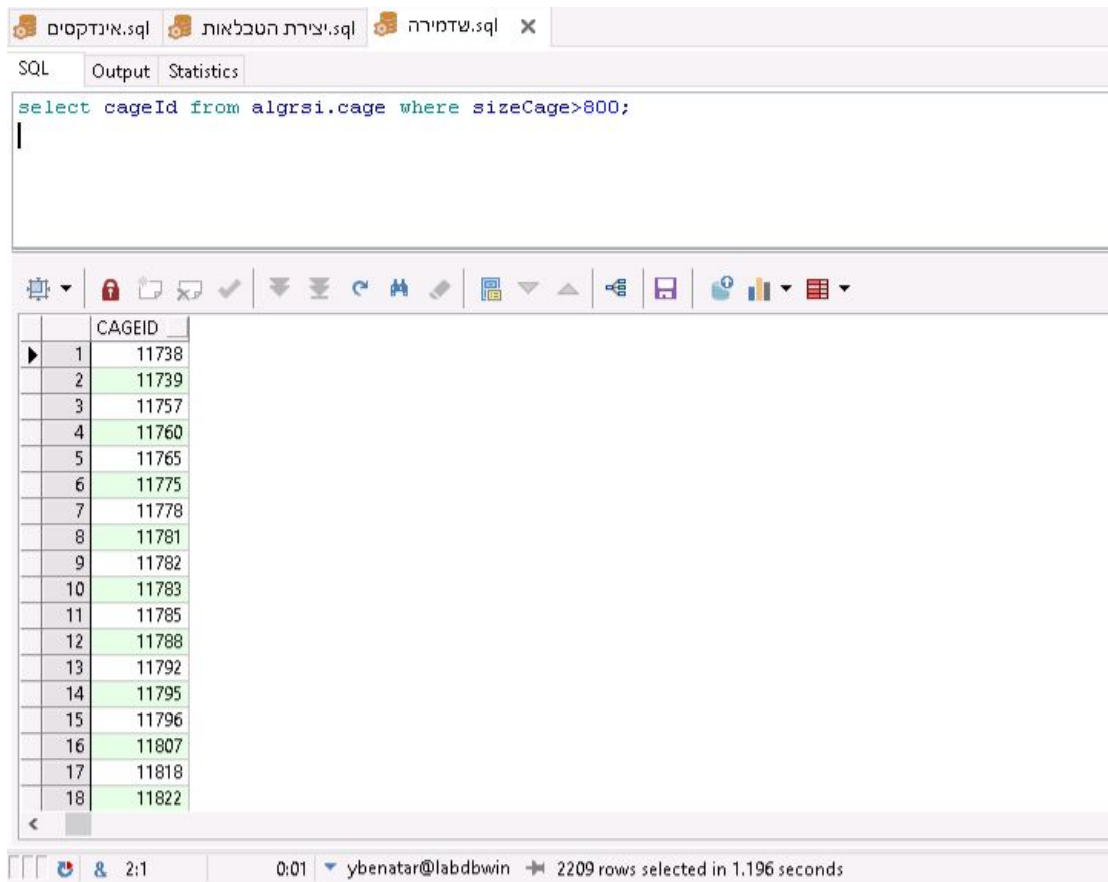
	AREALD
1	176343
2	831720
3	982570
4	832043
5	731563
6	245574
7	191067
8	752264
9	737181
10	779610
11	424387
12	614993
13	127412
14	296771
15	208053
16	269254
17	148921
18	426357
19	259836
20	572542
21	389926
22	390612
23	893305
24	658968

1:57 ybenatar@labdbwin 941 rows selected in 0.449 seconds

2. שאלנו שאילתה המחזירה את כל המספרים של הכלובים אשר גודלם הוא יותר מאשר 800 מטר מרובע.

האינדקס שיצרנו היה על הכלובים בעמודה של גודל הכלוב, מבחינה הגיונית כרגע נוצר לנו אינדקס שאומר איפה כל הכלובים עם גודל X מסוים. לכן השאילתה תלך לאינדקס ותחפש איפה כל הכלובים שגודלם הוא מעל 800. על כן זמן הריצה תפחת בצורה משמעותית.

לפני האינדקס



The screenshot shows a SQL IDE with three tabs: 'אינדקסים.sql', 'יצירת הטבלאות.sql', and 'שדמירה.sql'. The 'SQL' tab is active, displaying the query: `select cageId from algrsi.cage where sizeCage>800;`. Below the query, the 'Output' tab shows the results of the query, which is a table with two columns: 'CAGEID' and a list of 18 cage IDs. The status bar at the bottom indicates '2209 rows selected in 1.196 seconds'.

	CAGEID
1	11738
2	11739
3	11757
4	11760
5	11765
6	11775
7	11778
8	11781
9	11782
10	11783
11	11785
12	11788
13	11792
14	11795
15	11796
16	11807
17	11818
18	11822

אחרי האינדקס

SQL Output Statistics

```
select cageId from algrsi.cage where sizeCage>800;
```

	CAGEID
1	11738
2	11739
3	11757
4	11760
5	11765
6	11775
7	11778
8	11781
9	11782
10	11783
11	11785
12	11788
13	11792
14	11795
15	11796
16	11807
17	11818
18	11819

1:1 ybenatar@labdbwin 2209 rows selected in 0.753 seconds

3. שאלנו שאילתה המחזירה את כל שמות המיקומים של הכלובים אשר כתובתם הוא מתחיל ב 1. האינדקס שיצרנו היה על הטבלה CAGE בעמודה LOCATIONCAGE על האיברים אשר מתחילים ב 1, התברר כי הרוב המכריע של הנתונים היו כאלו שהתחילו בכתובת מספר 1 על כן האינדקס לא עזר אלא להיפך הוא עשה הרבה יותר גרוע כי עכשיו על השאילתה ללכת לחפש באינדקס והאינדקס לא ממש יכולן אותנו בצורה נכונה איפה להתמקד. על כן השאילתה אחרי האינדקס תעלה בזמן ריצה במעט.

לפני האינדקס

The screenshot shows a SQL query execution window with the following details:

- Query:** `select locationcage from algrsi.cage where locationcage Like '1%';`
- Results Table:**

	LOCATIONCAGE
1	102 Clarence Blvd
2	10 Weisz Street
3	100 San Antonio Drive
4	19 Karlstad Street
5	11 Lonnie Ave
6	10 Elliott Drive
7	100 O'fallon Road
8	10 Tzi Drive
9	16 Starr Ave
10	12 Avenged Street
11	11 Santiago Drive
12	18 Askew Street
13	178 Fonda Drive
14	17 Hyderabad Street
15	13rd Street
16	100 de Lancie Road
17	10 Depp Road
18	11 Kathleen Street
19	15 Ben Street
20	1 McCann Road
21	18 Metcalf Road
22	12nd Street
23	10 Erie Road
- Status Bar:** 1173 rows selected in 0.650 seconds

אחרי האינדקס

2 אינדקס.sql select locationcage from a ...

SQL Output Statistics

```
select locationcage from algrsi.cage where locationcage like '1%';
```

LOCATIONCAGE

1	102 Clarence Blvd	...
2	10 Weisz Street	...
3	100 San Antonio Drive	...
4	19 Karlstad Street	...
5	11 Lonnie Ave	...
6	10 Elliott Drive	...
7	100 O'fallon Road	...
8	10 Tzi Drive	...
9	16 Starr Ave	...
10	12 Avenged Street	...
11	11 Santiago Drive	...
12	18 Askew Street	...
13	178 Fonda Drive	...
14	17 Hyderabad Street	...
15	13rd Street	...
16	100 de Lancie Road	...
17	10 Depp Road	...
18	11 Kathleen Street	...
19	15 Ben Street	...
20	1 McCann Road	...
21	18 Metcalf Road	...

1:67 ybenatar@labdbwin 1173 rows selected in 0.893 seconds

שאלות שנותנות מידע מחיבור של אגפים שונים

- (1) שאלתא המייצגת כל שמות החיות בכלובים מעל מספר מסויים (בדוגמא: 20000), השאלתא מורכבת משימוש בטבלת כלוב-חיה וטבלת חיה (שהתקבלה ע"י אינטגרציה עם חבר כיתה).

```
select animalName from shechtma.animal, animalcage
where shechtma.animal.animalid = animalcage.animalid and animalcage.cageId>20000;
```

	ANIMALNAME
1	Crocodile
2	Koala
3	Albatross
4	Bandicoot
5	Snail
6	Fox
7	Bullfrog
8	Gull
9	Viper
10	Affenpinscher
11	Gibbon
12	Seahorse

- (2) שאלתא המייצגת כל מספרי הארצות-מוצא של חיות בכלובים מעל מספר מסויים (בדוגמא: 20000), השאלתא מורכבת משימוש בטבלת כלוב-חיה וטבלת חיה (שהתקבלה ע"י אינטגרציה עם חבר כיתה).

```
select COUNTRYID from shechtma.animal, animalcage
where shechtma.animal.animalid = animalcage.animalid and animalcage.cageId>20000;
```

	COUNTRYID
1	113
2	129
3	101
4	184
5	70
6	22
7	187
8	46
9	115
10	21
11	115
12	87

3) שאילתא המייצגת כל המאכלים של חיות בכלובים מעל מספר מסויים (בדוגמא: 20000), השאילתא מורכבת משימוש בטבלת כלוב-חיה וטבלת חיה (שהתקבלה ע"י אינטגרציה עם חבר כיתה).

```
select FOOD from shechtma.animal, animalcage
where shechtma.animal.animalid = animalcage.animalid and animalcage.cageId>20000;
```

	FOOD	
1	Magnotta Bel Paese Red	...
2	Wine - Chateau Bonnet	...
3	Taro Leaves	...
4	Soap - Mr.clean Floor Soap	...
5	Muskox - French Rack	...
6	Neckerchief Blck	...
7	Asparagus - Frozen	...
8	Flower - Dish Garden	...
9	Turkey - Whole, Fresh	...
10	Pasta - Bauletti, Chicken White	...
11	Oil - Macadamia	...
12	Bread - Corn Muffaletta Onion	...

הרשאות

כדי שנוכל לעבוד ככיתה שלמה המפתחת בסיס נתונים כאשר כל קבוצה בונה טבלאות שונות לבסיס הנתונים ואנו צריכים לגשת לטבלאות שלהם כדי לקחת מידע אנו צריכים לקבל מיוצרי הטבלאות הרשאה כדי שנוכל להשתמש בטבלאות שלהם.

במקרה שלנו כמעט ולא היינו צריכים לתת הרשאות לאף אחד מהסיבה שהכלובים והמתחמים מתייחסים בעיקר אחד לשני ובתכונה אחת לחיות.

SQL	Output	Statistics
<pre>grant ALL PRIVILEGES on algrsi.cage to shechtma</pre>		

SQL Window X

SQL	Output	Statistics
<pre>grant all privileges on algrsi.animalcage to dinhaim; grant all privileges on algrsi.area to dinhaim; grant all privileges on algrsi.cage to dinhaim;</pre>		

Views

VIEWS הם טבלאות וירטואליות. VIEWS מכילים הגדרות של עמודות וסוגי מידע שאותן עמודות יכולות להכיל. ההבדל בין הטבלאות לבין ה-VIEWS הוא שבטבלאות נשמרים נתונים באופן פיזי ואילו ב-VIEWS הנתונים לא נשמרים באופן פיזי בתוכם אלא הם רק מציגים נתונים הנשמרים בטבלאות. לכן לא ניתן לעדכן או להוסיף נתונים ל-VIEWS כפי שעושים לטבלאות.

התייחסנו לשני משתמשים עיקריים: (1) מטפלי-חיות (2) אנשי אבטחה.

(1) מטפלי-חיות;

- כיוון שהרבה שאילות מתייחסות למספר הזהות של חיה ומיקום החיה, כי מטפלים שונים צריכים לאתר מיקום של חיה מסויימת, יעיל יותר להפריד את עמודות אלה מחיפוש בטבלה המלאה.

```
select * from Animal_location_id t
```

	LOCATIONCAGE	ANIMALID
1	53 Shaw	13277
2	620 Freddy Street	3789
3	102 Clarence Blvd	10785
4	70 MacNeil Road	1807
5	70 MacNeil Road	4293
6	80 McDormand Road	10343
7	44 Tustin Road	15670
8	86 MacLachlan Road	12557
9	6 Hector Blvd	10531
10	55 Warley Road	14627
11	55 Warley Road	8720
12	55 Warley Road	8097
13	52 Mendoza Road	5570
14	52 Maebashi Street	4690

- חלק מעבודת המטפלים היא מעקב אחרי מספר החיות בכלובים, שלא יהיה עומס בכלוב, לכן ייצוג של עמודת גודל הכלוב עם עמודה של כמות החיות בכלוב שימושית באופן קבוע.

```
select * from Number_AnimalForCage t
```

	SIZECAGE	ANIMALAMOUNT
1	819	1
2	173	1
3	773	1
4	299	2
5	759	1
6	479	1
7	738	1
8	945	1
9	141	3
10	299	1
11	168	1
12	492	1
13	685	2
14	772	2

(2)

- בעת אזעקת אנשי האבטחה של גן-החיות למקרה, המידע הספציפי שנדרש הוא שם המתחם והמיקום, לכן הפרדנו את עמודות אלה משאר הטבלה במטרה לייעל וקצר את זמן החיפוש של אנשי האבטחה בעת הצורך.

```
select * from Location_Name t
```

	AREALOCATION	AREANAME
1	156 Faye	Rebecca
2	97 Lesley Drive	Harrison
3	48 San Dimas Road	Claire
4	44 Holliston Street	Nils
5	81 Kutcher Road	Balthazar
6	20 Gallagher Drive	Harvey
7	55 Shannon Street	Johnny
8	62 Ice Drive	Gordie
9	440 Cedar Park Blvd	Hikaru
10	855 Richmond Street	LeVar
11	88 Herbie Street	Paula
12	758 Whitman Blvd	Regina
13	795 Mills Ave	Lauren
14	988 Baker	Angie

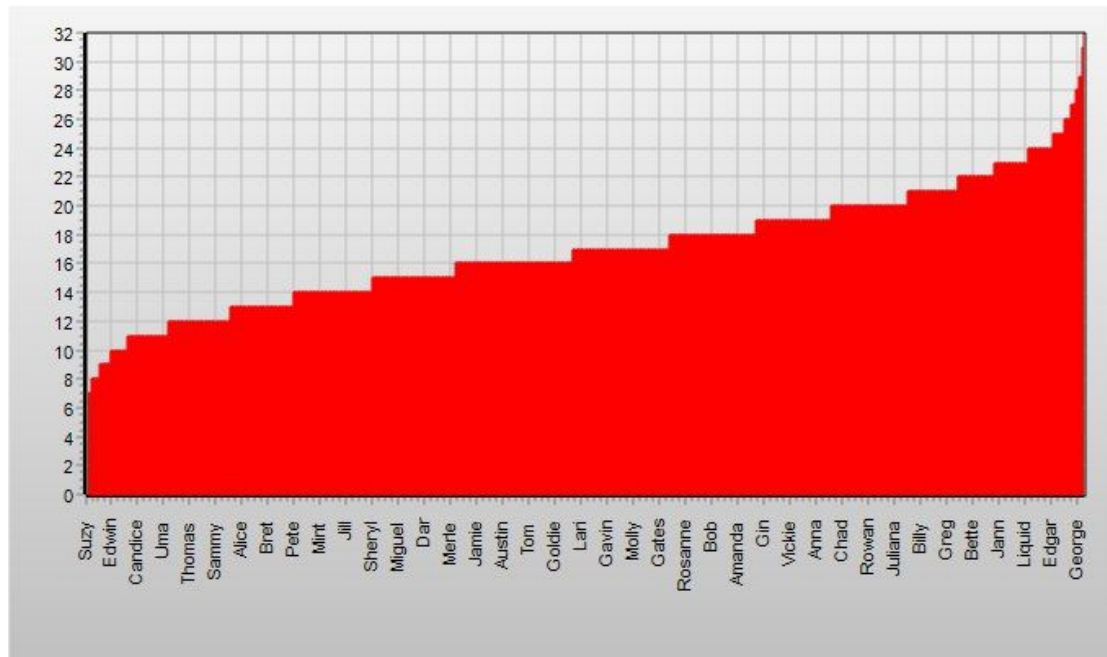
- כחלק משגרת הבדיקה של אנשי האבטחה בגן החיות, בודקים מאבטחי הסיור את שאכן מספר הכלובים הרשומים במערכת עבור מתחם מסויים נכון וסופרים את מספר הכלובים כל סיור. לכן טבלה שבה עמודת המתחמים וכמות הכלובים בכ"א מהם מסייעת לשגרה יעילה.

```
select * from Number_CageForArea t
```

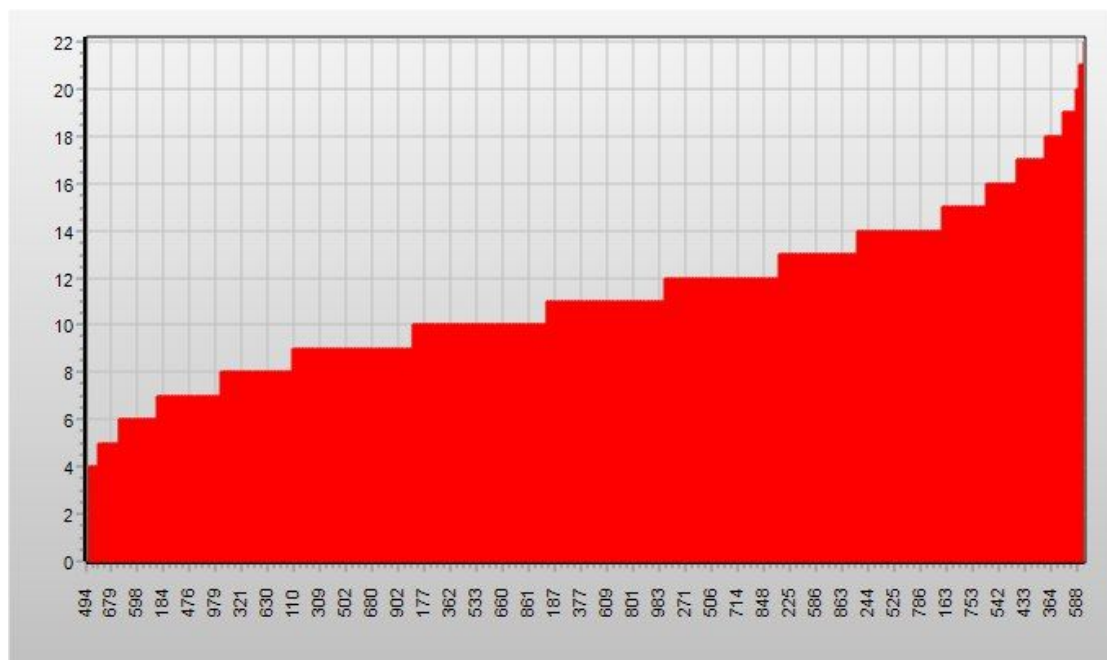
	AREAID	CAGEAMOUNT
1	428291	2
2	734838	1
3	341944	2
4	510171	2
5	741612	2
6	714891	4
7	133783	2
8	759450	1
9	469718	3
10	200359	2
11	237451	2
12	743117	2
13	463214	1
14	932399	2

תרשימים

תרשים 1. מייצג את מספר המיקומים הקיימים תחת שם מתחם אחד.



תרשים 2. מייצג את מספר הכלובים באותו גודל.



פונקציות

פונקציה בשפת SQL היא צורה מיוחדת של פקודה אשר מבצעת פעולות שונות על הנתונים בבסיס הנתונים.

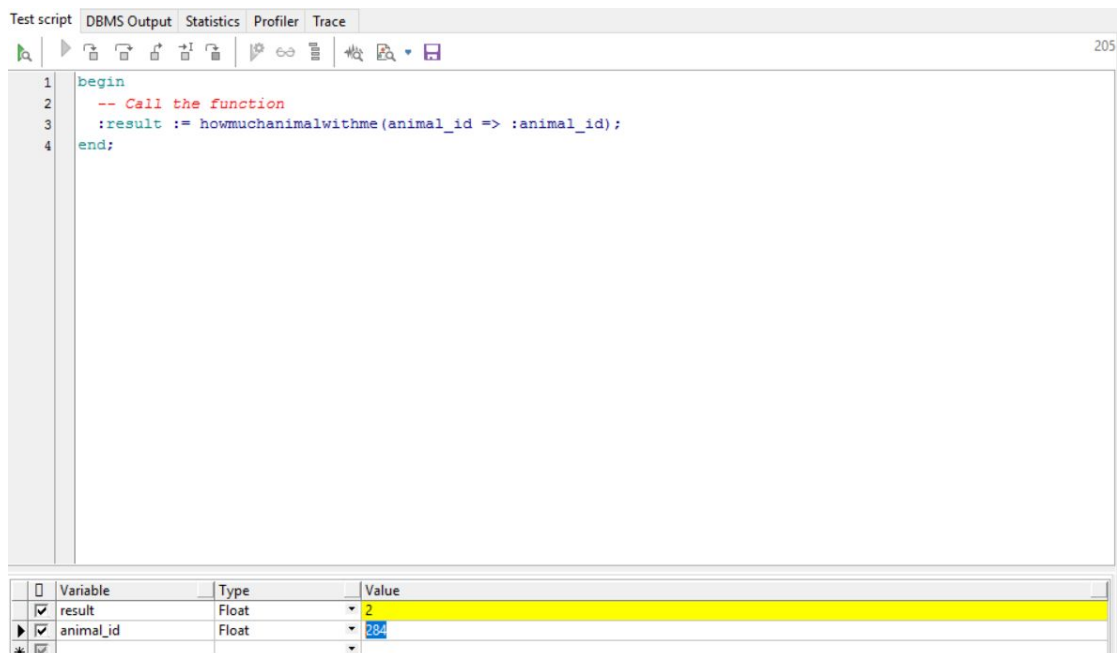
1.

הנהלת גן החיות קיבלה רשימה של חיות עם בעיות רפואיות. על ההנהלה לדעת כמה חיות חיים בכלוב עם אותה חיה בכדי לדעת האם להשאיר את החיה בכלוב או לחלופין להעביר אותה אגף. על כן ההנהלה יצרה פונקציה אשר כאשר מכניסים מס זהות של חיה הפונקציה תחזיר כמה חיות יש בכלוב שלה.

(הקוד נמצא בנספח הרביעי)

```
create or replace function HowMuchAnimalWithMe(animal_id int) return int is
  FunctionResult int;
begin
  select count(*) into FunctionResult
  from algrsi.animalcage
  where cageid=(select cageId from algrsi.animalcage where animal_id=animalid );
  return(FunctionResult);
end;
```

דוגמת הרצה



The screenshot shows a database IDE interface. At the top, there are tabs for 'Test script', 'DBMS Output', 'Statistics', 'Profiler', and 'Trace'. Below the tabs is a toolbar with various icons. The main area displays a SQL script:

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

At the bottom, there is a table showing the execution results:

Variable	Type	Value
result	Float	2
animal_id	Float	384

2.

הנהלת גן החיות מקבלת מדי חודש מס כלובים אשר פולטות זיהום אוויר לכל האזור שבו ממוקם הכלוב. על ההנהלה להביא וטרינרים מדי חודש אשר יטפלו באזורים הללו, על כן בהינתן מס כלוב בעייתי על ההנהלה לדעת כמה כלובים נוספים יש באותו אזור על כן יצרנו פונקציה המקבלת מס מזהה כלוב וכאשר מריצים את הפונקציה מקבלים כמה כלובים יש באזור שאותו הכלוב נמצא.

(הקוד נמצא בנספח הרביעי)

```

1 create or replace function how_much_cages_in_my_area(cage_id int) return int is
2   number_of_cages int;
3   my_area_id int;
4   begin
5     select areaid into my_area_id
6     from algrsi.cage
7     where cageid=cage_id;
8
9     select count(*) into number_of_cages
10    from algrsi.cage
11    where areaid=my_area_id;
12    return(number_of_cages);
13  end;
```

הרצה של הפונקציה

Test script DBMS Output Statistics Profiler Trace 205

```

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

Variable	Type	Value
result	Float	2
cage_id	Float	11924

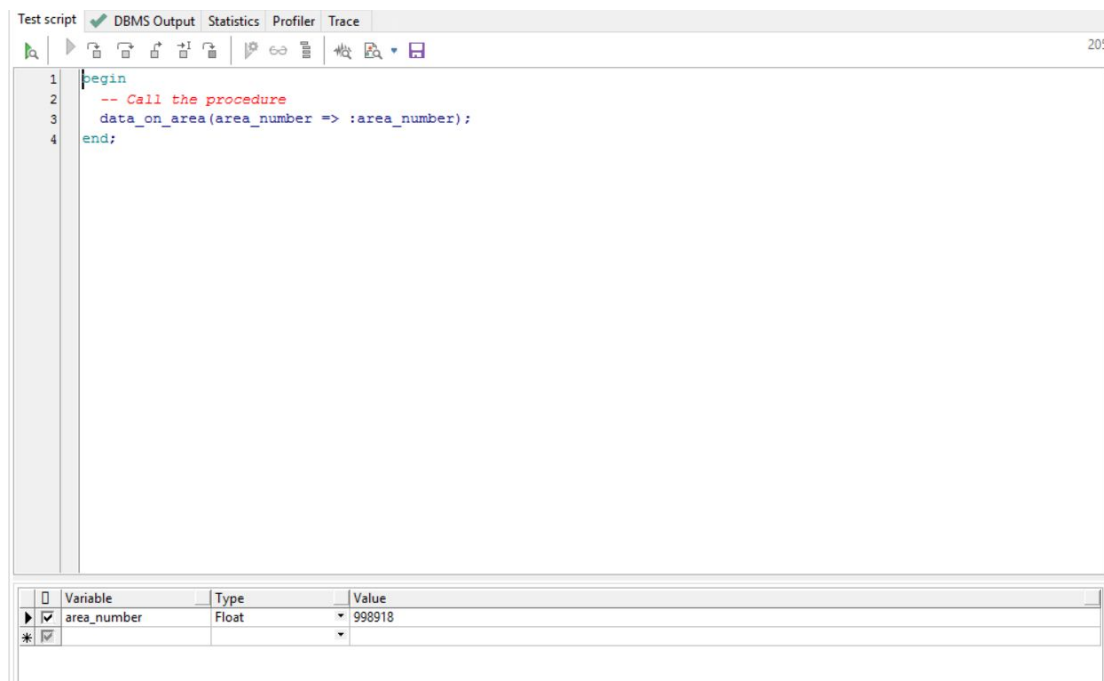
3.

מדי חודש על המחלקה לתת דוח פירוט על כל האזורים כמה כלובים וכמה חיות יש בכל אזור וכמובן את שמות האזורים הללו. על כן המחלקה החליטה לעשות פרוצדורה שתדפיס נתונים על אזור מסוים.

```

1 create or replace procedure data_on_area(area_number int) is
2 cursor animals is
3     select * from algrsi.animalcage natural join algrsi.cage natural join shechtma.animal
4     where area_number=algrsi.cage.areaaid;
5 num_of_cages int;
6 num_of_animals int;
7 begin
8     select count(*) into num_of_animals
9     from algrsi.animalcage natural join algrsi.cage
10    where area_number=algrsi.cage.areaaid;
11
12    select count(*) into num_of_cages
13    from algrsi.cage
14    where area_number=algrsi.cage.areaaid;
15
16    dbms_output.put_line('Number of cages: ' || num_of_cages);
17    dbms_output.put_line('Number of animals: ' || num_of_animals);
18
19    for i in animals
20    loop
21        dbms_output.put_line('Animal id: ' || i.animalid || ' | Animal name: ' || i.animalname);
22    end loop;
23
24 end;
```

הרצת הפרוצדורה



The screenshot shows a SQL IDE interface with a 'Test script' tab. The script contains the following code:

```

1 begin
2     -- Call the procedure
3     data_on_area(area_number => :area_number);
4 end;
```

Below the script, there is a table showing the execution results:

Variable	Type	Value
area_number	Float	998918
*		

פלט הפרוצדורה

```

Test script DBMS Output Statistics Profiler Trace
Clear Buffer size 10000 [ ] Enabled
Number of cages: 2
Number of animals: 1
Animal id: 15198; Animal name: Pony
|

```

.4

לעיתים קרובות על המחלקה להעביר חיות מכלוב לכלוב מסויים. על כן החליטה המחלקה להכין פרוצדורה אשר מקבלת מס זהות של הכלוב אליו רוצים להעביר את החיות ואת מספר הכלוב שממנו מוציאים את החיות, והפרוצדורה מבצעת את העברה.

```

united_cages
Code section Update
1 create or replace procedure united_cages(cage_id_in int, cage_id_out int) is
2 begin
3   update algrsi.animalcage
4     set cageid=cage_id_in
5     where cageid=cage_id_out;
6 end;

```

הרצת הפרוצדורה

```

Test script DBMS Output Statistics Profiler Trace
1 begin
2   -- Call the procedure
3   united_cages(cage_id_in => :cage_id_in,
4               cage_id_out => :cage_id_out);
5 end;

```

Variable	Type	Value
cage_id_in	Float	11112
cage_id_out	Float	11112

0:01 ybenatar@labdbwin Executed in 1.44 seconds

הנתונים לפני ואחרי הרצת הפרוצדורה

	ANIMALID	CAGEID
532	8130	11112
3503	15103	11112
267	14802	11115
1013	14299	11116
775	19192	11117
1701	10581	11122
2078	15377	11123
2750	1204	11124
1535	18101	11126
3484	9649	11128
2172	4842	11131
859	5429	11134
3590	7323	11135
1481	5386	11138
1110	3457	11142
122	14448	11146
3403	284	11146
2653	9526	11148
78	15842	11151
1548	3339	11151
246	12480	11151
1897	1242	11154
1940	8401	11159

לפני

	ANIMALID	CAGEID
267	14802	11112
532	8130	11112
3503	15103	11112
1013	14299	11116
775	19192	11117
1701	10581	11122
2078	15377	11123
2750	1204	11124
1535	18101	11126
3484	9649	11128
2172	4842	11131
859	5429	11134
3590	7323	11135
1481	5386	11138
1110	3457	11142
122	14448	11146
3403	284	11146

אחרי

נספחים

נספח ראשון: שאלות ואינדקסים

(האינדקסים מודגשים)

```
1. select cageId  
  
   from cage  
  
  where sizeCage>=all(select sizeCage from cage);
```

```
2. select cageId  
  
   from cage  
  
  where sizeCage>800;
```

create index IDX_cage_size on algrsi.Cage(sizeCage);

```
3. select areaId  
  
   from area  
  
  where areaName Like 'A%';
```

create index IDX_area_name on algrsi.area('A%');

```
4. select locationCage  
  
   from cage  
  
  where locationCage Like '1%';
```

create index IDX_cage_locationCage on algrsi.Cage('1%');

5. select areald,numberofanimal

from(select areald,count(*) as numberOfAnimal

from ANIMALCAGE,cage

where cage.cageid=animalcage.cageid

group by areald)

where numberOfAnimal>=all(select numberOfAnimal from(select areald,count(*) as
numberOfAnimal

from ANIMALCAGE,cage

where cage.cageid=animalcage.cageid

group by areald));

6. select max(numberofanimal) as max_animals_on_area

from(select areald,count(*) as numberOfAnimal

from ANIMALCAGE,cage

where cage.cageid=animalcage.cageid

group by areald);

7. select areald,count(*) as numberOfAnimal

from ANIMALCAGE,cage

where cage.cageid=animalcage.cageid

group by areald;

```
8. select cageId,Count(*) as NumberOfAnimals  
  
from ANIMALCAGE  
  
group by cageId;
```

נספח שלישי: שאלות אינטגרציה

- 1) select animalName from shechtma.animal,animalcage
where shechtma.animal.animalid = animalcage.animalid and
animalcage.cageid>20000;

- 2) select COUNTRYID from shechtma.animal,animalcage
where shechtma.animal.animalid = animalcage.animalid and
animalcage.cageid>20000;

- 3) select FOOD from shechtma.animal,animalcage
where shechtma.animal.animalid = animalcage.animalid and
animalcage.cageid>20000;

נספח רביעי: views

create or replace view Animal_location_id as

select LOCATIONCAGE,ANIMALID

from cage,animalcage

where cage.cageid = animalcage.cageid ;

create or replace view Number_AnimalForCage as

select SIZECAGE, AnimalAmount

from cage,(select cageid,count(*) as AnimalAmount

from animalcage

group by cageid) T

where T.cageId=cage.cageid;

create or replace view Location_Name as

select AREALLOCATION,AREANAME

from area ;

```
create or replace view Number_CageForArea as  
select AREAID,count(*) as CageAmount  
from cage  
group by areaid;
```

נספח חמישי: פונקציות

1.

```
create or replace function HowMuchAnimalWithMe(animal_id int) return int is  
FunctionResult int;  
begin  
select count(*) into FunctionResult  
from algrsi.animalcage  
where cageid=(select cageid from algrsi.animalcage where animal_id=animalid );  
return(FunctionResult);  
end;
```

2.

```
create or replace function how_much_cages_in_my_area(cage_id int) return int is  
number_of_cages int;  
my_area_id int;  
begin  
select areaid into my_area_id  
from algrsi.cage  
where cageid=cage_id;  
  
select count(*) into number_of_cages  
from algrsi.cage  
where areaid=my_area_id;
```

```
return(number_of_cages);
```

```
end;
```

3.

create or replace procedure data_on_area(area_number int) is

cursor animals is

```
select * from algrsi.animalcage natural join algrsi.cage natural join shechtma.animal
```

```
where area_number=algrsi.cage.areaid;
```

```
num_of_cages int;
```

```
num_of_animals int;
```

```
begin
```

```
select count(*) into num_of_animals
```

```
from algrsi.animalcage natural join algrsi.cage
```

```
where area_number=algrsi.cage.areaid;
```

```
select count(*) into num_of_cages
```

```
from algrsi.cage
```

```
where area_number=algrsi.cage.areaid;
```

```
dbms_output.put_line('Number of cages: ' || num_of_cages);
```

```
dbms_output.put_line('Number of animals: ' || num_of_animals);
```

```
for i in animals
```

```
loop
```

```
dbms_output.put_line('Animal id: ' || i.animalid || '; Animal name: ' || i.animalname);
```

```
end loop;
```

```
end;
```

4.

create or replace procedure united_cages(cage_id_in int, cage_id_out int) is

begin

 update algrsi.animalcage

 set cageid=cage_id_in

 where cageid=cage_id_out;

end;