funTESTic part 2

בדיקות בסיסיות עבור | BASIC TEST for

DB

בדיקות בסיסיות לDB (ORACLE)  
  
הקדמה :

למרות שלא כולם עושים ולא כולם מעדיפים לעשות בדיקות DB, והטענה היא שדרך בדיקות פונקציונאליות של המערכת אפשר לראות שפעולות כגון הוספה/עריכה/מחיקה של נתונים מתבצעות כנדרש דרך שכבת התצוגה הרי שזה מספיק, גם במקרה הגרוע ביותר שפעולת מחיקה התבצעה רק בשכבת התצוגה ובבסיס הנתונים עדין הנתון נשאר, לא כל כך משנה לנו שהרי המשתמש עדין לא יכול לגשת לנתונים עוד (כפי שהוכח בבדיקות) והנתון תופס כמה קילובייט שזה שטח זעום ומקסימום נוסיף שרת בעוד כמה שנים טובות,

חשיבה זו נכונה למערכת שעיקרה פונקציונאליות ולא אוגרת מידע רב, מערכות מידע שאוגרות מידע רב, במערכות כאלה יהיו מספר מועט של תהליכים עיקריים לעיבוד המידע (ETL) ובליוני נתונים יזרמו דרכם, במערכות אלו ניתן לבנות בדיקות לפי זיהוי מקומות של העברת נתונים STTUC, STTWC, ונתונים אלו תופסים מקום רב ולכן חשוב לוודא שכל דבר מוגדר נכון כי טעות קטנה בתהליך פירושה כמות ענקית של נתונים לא נכונים (ושטח גדול מבוזבז)  
  
במידה ואתם עושים צעדים ראשונים תתחילו בהסברים בויקפדיה[[1]](#footnote-1) או בבלוגים[[2]](#footnote-2)  
ותמשיכו לבדיקות[[3]](#footnote-3)

כמו במסמך הקודם[[4]](#footnote-4) אין כאן תיאורים והסברים אלא בדיקות מעשיות בלבד.  
מסמך נכתב לאור ניסיוני בבדיקות BI והוא כולל שליפות עבור בדיקות DB בסיסיות, עם סינטקס ORACLE , ניתן לקחת את הקונספט של הבדיקות כאן וליישם בDB עם סינטקס שונה(MySQL, MONGO DB).

אשמח לכל תוספת ושיפור מסמך זה (הוא רק טיפה מתוך הים וצריך עוד להוסיף)  
[yosef771@gmail.com](mailto:yosef771@gmail.com)  
בברכה  
יוסף חזן   
מתכנת agile point

1. בדיקות ברמת טבלה:
2. בדיקת טבלאות קיימות (יתכן ומצריך הרשאה של DBA):  
   צעד 1: שליפת כל הטבלאות -

* ""OWNER – מציג את שם הסכמה

Select t.OWNER t.TABLE\_NAME  
from dba\_tables t;

--במידה ואין הרשאה מספיקה

Select t.OWNER t.TABLE\_NAME  
from all\_tables t;

--צפיה בטבלאות שהיוזר הנוכחי יצר

Select t.OWNER t.TABLE\_NAME  
from user\_tables t;

צעד 2: השוואת הטבלאות מהשליפה לטבלאות המופיעות באפיון

**תוצאה צפויה:** כל הטבלאות מהאפיון ומהשליפה תואמות.

1. בדיקת טבלאות VIEW קיימות (יתכן ומצריך הרשאה שלDBA ):  
   צעד 1: שליפת טבלאות VIEW -

* View - טבלה המאפשרת  
  הצגה ועריכה מוגבלת וחלקית  
  לנתונים היושבים בטבלה אחרת. להוסיף קישור למקום עם הסבר.

Select \*  
from dba\_views;

--במידה ואין הרשאה מספיקה

select \*  
from all\_ views;

-- צפיה בטבלאות שהיוזר הנוכחי יצר

select \*  
from user\_ views;

צעד 2: השוואה לטבלאות VIEW הקיימות באפיון.  
**תוצאה צפויה:** כל טבלאות VIEW מהאפיון ומהשליפה תואמות.

1. השוואת נתונים משני טבלאות **באותה** סכמה.  
   צעד 1: הרצת השאילתה הבאה -

* יש לשנות את TABLE\_NAME\_A לשם הטבלה הראשונה.
* יש לשנות את TABLE\_NAME\_B לשם הטבלה השניה.

SELECT \*  
from TABLE\_NAME\_A   
MINUS  
SELECT \*  
from TABLE\_NAME\_B ;

**תוצאה צפויה:** נתונים לא תואמים יוצגו, במידה ואין רשומה כל שהיא - הכל תקין

צעד 2:

* בצעד זה השאילתה בסדר הפוך - טבלה B מינוס טבלה A.  
  ההסבר לכך: ברגע שטבלה A מכילה יותר מ- B לא מוצג שיש הפרש בין טבלת A לטבלת B.

SELECT \*  
from TABLE\_NAME\_B   
MINUS  
SELECT \*  
from TABLE\_NAME\_A ;

**תוצאה צפויה:** נתונים לא תואמים יוצגו, במידה ואין רשומה כל שהיא - הכל תקין.

1. השוואות נתונים בין שני טבלאות בעלות אותו שם ובסכמות **שונות**

* את "OWNER\_ a" יש להחליף בשם הסכמה הראשונה
* את "OWNER\_ b" יש לחליף בשם הסכמה השנייה
* השליפה מוצאת לבד את הטבלאות בעלות אותו השם
* יש לעשות שליפה שאפשר לבדוק טבלאות עם שמות שונים

Select X.TABLE\_NAME, X.cnt, Y.TABLE\_NAME, Y.cnt  
from (select TABLE\_NAME, count(\*) cnt  
 from all\_tab\_columns  
 where owner in ('OWNER\_a')  
 group by TABLE\_NAME) X,  
 (select TABLE\_NAME, count(\*) cnt  
 from all\_tab\_columns  
 where owner in ('OWNER\_b')  
 group by TABLE\_NAME) Y  
where X.TABLE\_NAME = Y.TABLE\_NAME  
and X.cnt = Y.cnt;

**תוצאה צפויה:** הטבלאות המופיעות בשני הסכמות תואמות.

1. DBlink – קישור לטבלה הנמצאת ב DB אחר (קישור זה מוגדר ע"י DBA)

* יעיל להשוואת 2 טבלאות תוך שימוש בפקודת MINUS כמפורט בדוגמא   
  + יש לשנות את TABLE\_NAME\_A לשם הטבלה המופיעה בDB החיצוני
  + יש לשנות את DB\_NAME לשם הDB החיצוני
  + יש לשנות את TABLE\_NAME\_B השני לשם הטבלה המופיעה בDB הנוכחי

צעד 1:

Select a.\*  
from TABLE\_NAME\_A@DB\_NAME a  
MINUS  
Select a.\*  
from TABLE\_NAME\_B b;

**תוצאה צפויה:** נתונים לא תואמים יוצגו, במידה ואין רשומה כל שהיא הכל תקין**.**

**צעד 2:  
הרצה הפוכה**

Select a.\*  
from TABLE\_NAME\_B b  
MINUS  
Select a.\*  
from TABLE\_NAME\_A@DB\_NAME a;

**תוצאה צפויה:** נתונים לא תואמים יוצגו, במידה ואין רשומה כל שהיא הכל תקין**.**

1. בדיקת הגדרות ברמת עמודה:
2. DATA\_TYPE – מציג את סוג **העמודה** (תאריך, האם הערך הוא מספרי או מחרוזתי)
3. במידה והשדה מספרי כמות הספרות שהשדה יכול להכיל יוצג ב DATA\_PRECISION
4. במידה והשדה מחרוזת כמות התווים שהשדה יכול להכיל יוצג ב CHAR\_LENGTH

* במקום XXX יש לרשום את שם הטבלה הנוכחית בבדיקה

נריץ את השאילתה הבאה:

Select t.OWNER, t.TABLE\_NAME, t.COLUMN\_NAME, t.DATA\_TYPE, t.DATA\_PRECISION, t.CHAR\_LENGTH   
from all\_tab\_cols t  
where t. TABLE\_NAME = 'XXX';

**תוצאה צפויה:** התוצאות המוצגות תואמות לנכתב באפיון.

1. NULLABLE – האם מותר להשאיר שדה זה ללא ערך ("null").

* במקום XXX יש לרשום את שם הטבלה הנוכחית בבדיקה

נריץ את השאילתה הבאה:

Select t.OWNER, t.TABLE\_NAME, t.COLUMN\_NAME, t.NULLABLE  
from all\_tab\_cols t  
where t. TABLE\_NAME = 'XXX';

**תוצאה צפויה:** התוצאות המוצגות תואמות לנכתב באפיון

1. CONSTRAINT\_TYPE – מציג את סוג **הערך**

ישנם 3 סוגי תוצאות :  
'P' – מורה שהנתון הוא מפתח ראשי ('PRIMARY KEY')[[5]](#footnote-5)

'R' – מורה שהנתון הוא מפתח משני ('FOREIGN KEY ')[[6]](#footnote-6)

'U' – UNIQUE)) הנתון הוא חד חד ערכי

במידה והתוצאה היא 'R' (מפתח משני):

* מציג את טבלת היעד שלו ב FK\_TABLE\_TARGET
* ובתוך טבלת היעד מציג את עמודת היעד בFK\_COLUME\_TARGET
* יש להחליף את 'XXX' בשם הטבלה הנוכחית

SELECT a.CONSTRAINT\_NAME  
 c.R\_CONSTRAINT\_NAME,  
 a.COLUMN\_NAME,  
 c.CONSTRAINT\_TYPE,  
 cc.TABLE\_NAME FK\_TABLE\_TARGET  
 cc.COLUMN\_NAME FK\_COLUMN\_TARGET,  
FROM all\_cons\_columns a  
 JOIN all\_constraints c  
 ON a.CONSTRAINT\_NAME = c.CONSTRAINT\_NAME  
 AND a.OWNER = c.OWNER  
 LEFT OUTER JOIN all\_cons\_columns cc  
 ON c.OWNER = cc.OWNER  
 AND c.R\_CONSTRAINT\_NAME = cc.CONSTRAINT\_NAME  
WHERE c.TABLE\_NAME = 'XXX'  
AND ( c.CONSTRAINT\_TYPE = 'P'  
 OR c.CONSTRAINT\_TYPE = 'R'  
 OR c.CONSTRAINT\_TYPE = 'U');

1. בדיקה אופציונאלית - שליפת הערות שנכתבו עבור עמודה

* COLUMN\_NAME - העמודה
* COMMENTS – ההערה
* יש להחליף את 'XXX' בשם הטבלה הנוכחית

SELECT a.COLUMN\_NAME, a.COMMENTS  
FROM all\_col\_comments a  
WHERE a.TABLE\_NAME = 'XXX';

3. בדיקת הנתונים בעמודה

* כאשר אנו רוצים לבדוק נתונים אנו צריכים לזכור שמדובר פעמים רבות באלפי רשומות ולכן אנו לא נשלוף את כל הערכים ונתחיל לעבור אליהם אלא נכתוב שאילתה אשר תציג לנו נתונים שגוים   
  בעצם אנו נכתוב שאילתה הדורשת נתונים שלא עומדים בתנאים שנכתבו באפיון או נתון שהוא חריג מהצריך להיות בעמודה הנוכחית
* לאחר שהגדרנו שהשאילתה תציג את השונה והחריג אנו נגדיר שהשאילתה תציג לנו את הכמות נכתוב את זה בתוך שבלונה

1. שבלונה המציגה כמות

SELECT count(\*)  
FROM (השליפה המבוקשת);

**תוצאה צפויה**: מוצגת לנו כמות הנתונים השגויים

* לדוגמא:

SELECT count(\*)  
FROM (

Select a.\*  
from TABLE\_NAME\_A a  
MINUS  
Select a.\*  
from TABLE\_NAME\_B b;

);

**תוצאה צפויה**: כמות הרשומות שלא הועתקו מטבלה A לטבלה B

B. שבלונה המציגה האם עבר או נכשל

* מומלץ להשתמש בדוגמא זו רק למתקדמים זוהי הדגמה חלקית

SELECT CASE WHEN xxx IS NOT NULL THEN "test over" ELSE "test not over" END AS result  
FROM ();

**תוצאה צפויה**: בדוגמא זאת רשומה שבעמודה XXX הערך לא ריק יוצג "test over"

C. בדיקת שדה תאריך יצירה לא מאוחר משדה עידכון ולהיפך:

- XXX\_ID

SELECT XXX\_ID, create\_date, update\_date  
FROM table\_name   
WHERE ( create\_date > update\_date  
 OR update\_date < create\_date) ;

**תוצאה צפויה:** יוצגו שדות תאריך שיצירה מאוחר מעידכון ולהיפך לצידם יוצג המזהה של הרשומה(נתון חד חד ערכי).

1. ויקפדיה BI

   ויקפדיה BIG DATA  
   ויקפדיה מערכות מידע [↑](#footnote-ref-1)
2. (זימזומים)  
   מה זה BI

   מה זה BIG DATA [↑](#footnote-ref-2)
3. בדיקות BI

   <http://www.dwh.co.il/blogs/entry/30-%D7%91%D7%93%D7%99%D7%A7%D7%95%D7%AA-bi-%D7%95-dwh-%D7%9C%D7%A2%D7%95%D7%9E%D7%AA-%D7%94%D7%91%D7%93%D7%99%D7%A7%D7%95%D7%AA-%D7%91%D7%AA%D7%97%D7%95%D7%9E%D7%99%D7%9D-%D7%90%D7%97%D7%A8%D7%99%D7%9D>

   בדיקות: השוואת כמות אוכלוסיה בין טבלאות המקור ליעד.

   1. בדיקת גודל, סוג שדה ומבנה תנאים.
   2. בדיקות תנאים – שגויים ותנאי סינון.
   3. אכלוס נתונים לפי חוקי טעינה והמרה ,חישוב לפי נוסחאות, שדות תפעוליים שהתוספו לטבלאות היעד וכן טיפול במקרי קצה.
   4. שלמות נתונים – ספירה והשוואת כמות, סכומים, טעינת נתונים ונתונים היסטוריים (SCD).
   5. תקינות מפתחות.
   6. זמני ביצוע.
   7. בדיקה ויזואלית של סכימת ה- ERD ולודא שלא קיימים self join שלא על פי האפיון.
   8. שמות ותאורים לאוביקטים – שמות משמעותיים, פורמט מובנה לכל האוביקטים והפילטרים, ההגדרות העסקיות המפורטות באפיון מיושמות.
   9. בדיקות אינטגרצה.
   10. בדיקת חשיפה והחבאה של אוביקטים.
   11. אגריגציה.

   התאמה עם טבלאות הפיענוח  
   בדיקות BIG DATA [↑](#footnote-ref-3)
4. קישור למסמך הקודם בגיטאב [↑](#footnote-ref-4)
5. מפתח ראשי = מוגדר על סוג נתונים חד חד ערכי ( חד חד ערכי = נתון שאין לו כפילות, כמו שאין שני אנשים עם אותו מספר של תעודת זהות) והדבר עוזר לקשר נתונים אחרים לנתון הנוכחי [↑](#footnote-ref-5)
6. מפתח משני = מוגדר על שדה המקושר לשדה מפתח ראשי בטבלה אחרת להחליף בקישור למקום עם הסבר [↑](#footnote-ref-6)