



בית הספר הגבוה לטכנולוגיה בירושלים

מיני-פרוייקט בבסיסי נתונים – תשע"ג

מורים:

ד"ר משה גולדשטיין, מר עזרא דשט, מר אליעזר גנסבורגר, מר אריה ויזן
גב' עדינה מילסטון, גב' ג'ויס ווגל, גב' הדר סופבייב

הקדמה	▪
חומרים לשימוש בקורס	▪
הערות חשובות	▪
נוהלי עבודה	▪
נוהלי הגשה ובדיקה	▪
בקשר לציון	▪
תכנית הסמסטר	▪

הקדמה

ככל מיני פרוייקט, עיקר הקורס הזה יהיה מעשי-תכנותי. במשך הסמסטר נבנה מערכת תכנה מבוססת מסד נתונים (database application). בחירת תחום העיסוק של מערכת זו תהיה בהתאם לנטיות לבך, אבל לקוח מעולם המעשה, מעולם המציאות (למשל, מערכות לניהול נתוני תחרויות ספורט, תחזוקת ציוד מסוים, סוכנות נסיעות, תיווך דירות, בית ספר, בית חולים, ספריה, מרפאה, ...). בצורה כזאת אנו מאפשרים לך לעסוק במשך כל הסמסטר בתחום שמעניין אותך במקום שתעסוק במשהו שנוצר באופן מלאכותי, לצורכי הקורס הזה בלבד. למיני-פרוייקט שלנו יהיו ארבעה שלבים התלויים זה בזה; כלומר, שלב מסוים לא יוכל להתבצע עד השלמת השלב הקודם לו.

חשוב מאוד לציין שבגלל אופי הקורס הזה, שמתפתח במשך הסמסטר וכל שלב בנוי על השלבים הקודמים, לא יהיה ניתן להצטרף לקורס, וגם לא לעבור בין קבוצות תרגול, אחרי השבוע השלישי, זמן הגשת שלב מס' 1, שהוא הבסיס לכל המיני-פרוייקט.

חומרים לשימוש בקורס

• דוגמת מיני-פרוייקט משנים קודמות

• ערכות להתקנת תכנות של חברת Oracle, רלוונטיות לקורס, ניתן להוריד מהאתר

www.oracle.com

▪ Oracle Database 10g Express Edition

▪ Oracle Developer Tools for Visual Studio

• ערכת התקנה של הכלים שנשתמש ברוב המשימות של המיני-פרוייקט:

▪ PL/SQL developer – שימוש בתכנה הזאת דורש רישיון שמוגבל בזמן (עד ה-31/08/2013), אבל היא חינם. קוד הרישיון הזה יפורסם באתר הקורס, שב-Moodle. אם ברצונך להשתמש בתכנה הזאת במחשב האישי (או הנייד) שלך, חייבים קודם להתקין את Oracle Database 10g Express Edition (חשוב מאוד להיזכר בסיסמה שנתת למשתמש system בזמן התקנת ה-Express Edition; אם לא, לא תוכל אפילו להיכנס למערכת ניהול בסיסי הנתונים שהתקנת במחשב האישי שלך). תכנה זו תהיה מותקנת במעבדות.

▪ DDS-Lite – תכנה זו מיועדת לעיצוב הסכמה המושגית של בסיס הנתונים; היא מבוססת על המודל ישויות-קשרים (גרסת Chen). זאת אינה תכנת חינם. אם ברצונך להתקין אותה במחשב האישי/הנייד שלך, ניתן לקבל את ערכת ההתקנה מהמורה של קבוצת המעבדה שלך, להשתמש בה לתקופת הקורס, עם התחייבותך למחוק אותה מהחשב שלך מייד עם סיום הקורס. תכנה זו תהיה מותקנת במעבדות.

• מצגות Oracle - מצגות ppt שנשתמש מדי פעם במשך הסמסטר.

• ספרים רלוונטיים לנושאי הקורס שנמצאים בספריות של כל המכונים:

(1) Oracle PL/SQL Programming, 5th Edition מאת S. Feuerstein ו-B. Pribyl, 2009.

(2) Oracle Database 10g PL/SQL Programming מאת S. Urman ו-M. McLaughlin, 2004.

• ספרים רלוונטיים לנושאי הקורס נגישים בספריה האינטרנטית Safari שספרית המכון מנויה לה:

(1) Oracle PL/SQL Programming, 4th Edition מאת S. Feuerstein ו-B. Pribyl, 2005.

(2) Learning Oracle PL/SQL מאת B. Pribyl, 2001.

(3) Oracle PL/SQL by Example, 4th Edition מאת B. Rosenzweig, 2008.

(4) SQL and Relational Theory, 1st Edition מאת C. J. Date, 2009.

(5) The Art of SQL מאת Stephane Faroult ו-Peter Robson, 2006.

(6) Introduction to SQL: Mastering the Relational Database Language, 4th Edition, 20th Anniversary Ed מאת D. F. van der Lans, 2006.

• מדריכי Oracle למיניהם

[Oracle SQL Reference](#)

[Oracle PL/SQL User Guide and Reference](#)

[SQL Developer Tutorial](#)

[Stanford Oracle Tutorial](#)

[CUNY SQL*Plus Tutorial](#)

[Oracle 11g PL/SQL](#)

הערות חשובות

- (1) ייתכנו שינויים במשך הסמסטר. אם בפועל יהיו שינויים, הודעות תישלחנה בדוא"ל באמצעות ה-Moodle או בזמן השיעור. לכן, לא תתקבלנה תלונות על אי ידיעה (תלונות כגון: "לא ידעתי", "לא שמעתי", "לא הייתי בשיעור", וכו').
- (2) נלמד איך לכתוב יישום חלונאי שניגש לבסיס נתונים טבלאי (במקרה שלנו יהיה בסיס נתונים מנוהל ע"י מערכת ניהול בסיסי הנתונים Oracle). בלינקים דלעיל אפשר למצוא כלים לפיתוח יישומים מסוג זה, ותיעוד עליהם.
- (3) רצוי שלכל זוג יהיה מחשב נייד אחד כי מומלץ להוריד ולהתקין במחשב האישי שלך את Oracle Database 10g Express Edition ואת PL/SQL developer. הרישיון לשימוש ב-PL/SQL developer הוא חינם אבל דורש הכנסה חד-פעמית של קוד רישיון שיהיה בתוקף עד ה-31/08/2013. קוד הרישיון הזה יפורסם באתר הקורס, שב-Moodle.
- (4) חשוב לציין שלמרות שהכלי העיקרי לעבודה במיני-פרויקט (אחרי התכנון והעיצוב של סכמת בסיס הנתונים עם DDS-Lite) יהיה PL/SQL developer, מומלץ גם להתקין ולנסות את Oracle SQL Developer. קיים חומר הדרכה מחברת Oracle (ה-SQL Developer Tutorial) שבאמצעותו ניתן ללמוד בקלות איך משתמשים ב-Oracle SQL Developer.
- (5) ע"מ לראות את קובץ ה-Help של DDS צריך להוריד את תוכנית winhlp32.exe מ-<http://www.microsoft.com/en-us/download/details.aspx?id=91>

נוהלי עבודה

העבודה על המיני-פרויקט תיעשה בזוגות; באופן חריג נאפשר שלשות. לא יתאפשר מצב בו השותפים/השותפות יהיו/תהיינה משתי קבוצות תרגול שונות. במקרה כזה העבודה לא תיבדק. הנוכחות במעמדות תהיה חובה. תתבצע אכיפה של נוכחות כל שיעור. בלא נוכחות בקורס לא תתבצע הגנה על הפרויקט בסוף הקורס.

כל תלמיד/תלמידה חייב/חייבת לשמוע את הקורס מהמרצה שהוא/היא רשום/רשומה אצלו/אצלה. כדי לבצע את משימות המיני-פרויקט, חייבים להיות לכם/לכן חשבונות (login name ו-password) במערכת לניהול בסיסי נתונים Oracle. אנו מניחים שבסמסטר א' כבר קיבלתם חשבונות אלה. אם זה לא המצב, תיפנו למורה של הקבוצה שלכם על מנת להסדיר את עניין החשבונות.

אם תהינה בעיות כלשהן, ניתן יהיה לפנות למורי הקורס באמצעות דואר אלקטרוני, בהתאם לקבוצה שכל אחד מהם אחראי עליה.

הערה חשובה: לסיכום כל הנ"ל, ועל מנת ללמוד איך מתקנים את חבילת התכנה **Oracle** במחשב האישי שלכם, תסתכלו במסמך הזה.

נוהלי הגשה ובדיקה

בקשר להגשת שלבי המיני-פרויקט:

כל שלב במיני-פרויקט (ראה פרטים ב-תכנית הסמסטר) יתבצע בערך בשלושה שבועות. לכל שלב במיני-פרויקט יהיה תאריך הגשה ב-Moodle. כל זוג יעלה קובץ ZIP לתא ההגשה. קובץ ה-ZIP הזה יכיל דו"ח המתייחס לאותו שלב ביחד עם קבצי קוד למיניהם, קבצי פלט וכו'. כל מה שייכת במשך המיני-פרויקט, ב-SQL או ב-PL/SQL, חייב להיות מוגש בקבצי sql. תוכן קבצי ה-sql וקבצי הפלט יופיע גם בדו"ח מלווה בהסברים המתאימים, ותשובות לשאלות המופיעות כחלק מהמשימות של כל שלב.

בקשר להגשה סופית והגנה על המיני-פרויקט כולו:

המיני-פרויקט כולו יוגש בסוף הסמסטר ל-Moodle והסטודנטים יצטרכו להגן על עבודתם. תאריך ההגנה ייקבע על ידי מדור הבחינות כחלק מבחינות מועד א', או בתאום בין כל מורה והקבוצה לה הוא/היא אחראי/אחראית.

איך ייבדק כל שלב?

- הדגמה מהירה של כל שלב תתבצע במעבדה בשיעור מייד אחרי תאריך ההגשה. במשך ההדגמה המהירה המורה ירשום/תרשום בקובץ הציונים, ליד כל מרכיב של אותו שלב, האם הודגם או לא. המורה גם ירשום אם נכתב הדו"ח החלקי של אותו שלב. דו"ח זה חייב לכלול תשובות מפורטות ומנומקות לכל השאלות שכלולות במשימות של אותו שלב. כל דו"ח כזה חייב לכלול הסברים רלוונטיים לשלב הספציפי - הדפסות למיניהן (קוד, הרצות, מסכים...) וכל דבר שתמצאו לנכון לציין על מה שעשיתם באותו שלב. בזמן הבדיקה, המורה יוודא/תוודא שדו"ח כזה

נכתב ויסתכל עליו בצורה יחסית שטחית, בלי לבדוק את תוכנו ממש; אם בזמן הבדיקה במעבדה יתברר שאין דו"ח כזה להגשה באותו רגע, המורה ירשום שדו"ח כזה עדיין לא נכתב לאותו שלב.

- **בדיקה לעומק של כל שלב** תתבצע בסוף הסמסטר אחרי הגשת המיני-פרוייקט כולו והציון עליו ייקבע בהתאם לשני מרכיבים:

(א) ציון על הדו"ח החלקי של אותו שלב: ציון זה ייקבע בהתאם למשקלן של המשימות השונות של אותו שלב.

(ב) הגנה על השלב: תוך הדגמת המיני-פרוייקט כולו, המורה ישאל/תשאל שאלות בקשר למשימות של השלב - הציון לאותו שלב יהיה שבר בין 0 ל-1 ויהיה מבוסס על התשובות לשאלות הנ"ל. שבר זה יופעל על הציון שהמורה יעניק/תעניק לדו"ח החלקי של אותו שלב, והתוצאה תהיה הציון של אותו שלב.

דוגמאות של שאלות שהמורה יכול/יכולה לשאול:

- מה עשתה תכנת עיצוב הסכמה (ששמה DDS-Lite) בכל צעד של תהליך העיצוב - תכנון ה-ERD, תירגומו ל-DSD ולאחר מכן תרגומו ל-script של פקודות create table.
- איך הכנתם את הנתונים לטעינת הטבלאות של בסיס הנתונים ואיך תכנת ה-PL/SQL developer ביצעה את הטעינה – כלומר, באיזה מרכיבים של PL/SQL developer השתמשתם ואיך הטעינה יצאה לפועל.
- מה אמורה לעשות שאילתא מסוימת ואיך מערכת ניהול בסיסי הנתונים אמורה לבצע אותה.
- איך היית מבטא את השאילתא שכתבת ב-SQL כשאילתא באלגברה טבלאית (relational algebra).

▪ **בקשר לדו"ח והציון הסופי הכולל של המיני-פרוייקט:**

כפי שנאמר לעיל, לכל שלב, חייב להיות דו"ח חלקי שמתייחס לביצוע המשימות של אותו שלב. בסוף הקורס, כל זוג יידרש להגיש את כל הדו"חות החלקיים האלה ביחד, ערוכים כדו"ח סופי אחד. הדו"ח הזה יוגש בשיעור האחרון של הסמסטר או לכל המאוחר ביום שייקבע להגנת המיני-פרוייקט (כמובן שבקורס כזה אין מועד ב'). המורה יבצע/תבצע בדיקה לעומק של אותו דו"ח סופי והציון של כל שלב ייקבע בהתאם למה שנאמר לעיל. הציון הסופי של המיני-פרוייקט כולו ייקבע כסה"כ המשוקלל של ציוני כל השלבים יחד (ראה בהמשך).

▪ **בקשר לבדיקת השלב האחרון של המיני-פרוייקט:**

זאת תהיה בדיקת/הדגמת/הגנת המיני-פרוייקט כולו, תוך הרצה של כל המיני-פרוייקט באמצעות ה-GUI שיפותח באותו שלב.

ציון

שיקולול ציונים של כל מרכיבי שלב מס' 1 של המיני-פרוייקט

הצעת תיאור האירגון	ERD DSD	נרמול הסכמה הלוגית והגדרתה בשפת SQL	יצירת טבלאות	אכלוס טבלאות	הגנה
10%	40%	20%	20%	10%	[1.0 ... 0.0]

שיקולול ציונים של כל מרכיבי שלב מס' 2 של המיני-פרוייקט

שאליות	אינדקסים	אילוצים מוגדרים עם ALTER	UPDATE DELETE GRANT	ROLLBACK COMMIT GRANT	VIEWS	הגנה
40%	10%	10%	15%	10%	10%	[1.0 ... 0.0]

שיקולול ציונים של כל מרכיבי שלב מס' 3 של המיני-פרוייקט

שאליות עם משתני החלפה והפקת דו"חות	PL/SQL	הגנה
25%	75%	[1.0 ... 0.0]

שיקולול ציונים של כל מרכיבי שלב מס' 4 של המיני-פרוייקט

DB application	הגנה
100%	[1.0 ... 0.0]

שיקולול ציונים של כל מרכיבי המיני-פרוייקט

שלב מס' 1	שלב מס' 2	שלב מס' 3	שלב מס' 4	איכות עריכתו של הדו"ח הסופי	בנוס על תוספת ומעוף
27%	37%	13%	13%	10%	5%

תכנית הסמסטר

שלב מס' 1 (שלושה שבועות) – מועד הגשה: 14/3/2012 – ג' ניסן תשע"ג

הרצאה:

1. עקרונות עיצוב הסכמה של בסיס נתונים.
2. המודל ישויות-קשרים (ER), איך משתמשים בו לעיצוב הסכמה המושגית של בסיס נתונים, ואיך ממירים דיאגרמות ER לסכמה לוגית במודל הטבלאי.
(מצגות משנים קודמות אפשר למצוא [כאן](#), [כאן](#), [כאן](#), [כאן](#), [כאן](#)).

הצעת תאור האירגון:

עליך לבחור בארגון בו אתה מעוניין לעסוק במסגרת המיני-פרוייקט.
משימתך לכתוב תיאור של הארגון שבחרת, תוך כדי ציון היבטים מסוימים שאתה חושב מיוחדים.

ניתוח הבעיה, ותרשים ERD:

על בסיס התיאור מהסעיף הקודם, בצע ניתוח מבנה המידע של הארגון הנידון. כלומר, זהה את כל סוגי הישויות, תכונותיהן מפתחותיהן הראשיים, סוגי הקשרים ביניהן, וכו'. משימתך לצייר תרשים ERD מתאים לניתוח שלך. לצורך זה, עומדת לרשותנו התכנה DDS Lite. תכנה זו מיועדת לתכנון סכמה של בסיס נתונים תוך ציור תרשים ERD. יתרון השימוש בתכנה זו הוא שהתרשים ERD שאנו מציירים מומר באופן (כמעט) אוטומטי לתרשים DSD ול-script של פקודות create table מתאים לאותו תרשים ERD.

הערות:

1. התרשים ERD שלך יכול את רוב המושגים שנתמכים ע"י המודל ERD, תוך שמירה על גודל סביר מבחינת מספר סוגי ישויות המשתתפים בתרשים (לא כולל סוגי ישויות שנוצרים מהאוסף המקורי ממנו התחלת את תהליך עיצוב הסכמה). קנה מידה "סביר" לגודל התרשים יהיה שלושה סוגי ישויות לכל תלמיד/תלמידה. כלומר, מכיוון שעובדים בזוגות, לתרשים חייבים להיות בסביבות שישה טיפוסים ישויות. במקרה החריג שתהיה שלשה, נדבר על תשעה סוגי ישויות.

2. במפגש השני בשלב הזה של המיני-פרוייקט, כל זוג חייב לבוא לכיתה עם מסמך בן עמוד אחד בו יתואר באופן מילולי הארגון שהזוג בחר לעסוק בו וטיטוט ניתוח/תרשים ERD מוצע.

המרת תרשים ה-ERD לסכמה לוגית המיוצגת כתרשים DSD וכ-script של פקודות create table:

ה-DDS Lite ממיר באופן אוטומטי את תרשים ה-ERD שציירת בסעיף הקודם לתרשים DSD ול-script של פקודות create table.

משימתך לכתוב הסבר מפורט של המרת תרשים ה-ERD שה-DDS Lite מבצע בשבילנו באופן אוטומטי. ההסבר הזה מהווה חלק הכרחי של השלב הזה וחייב להופיע בדו"ח החלקי שמתייחס לשלב הזה של המיני-פרוייקט. אם ההסבר הזה לא יופיע בדו"ח, הציון של השלב הזה יהיה 0.

נירמול הסכמה:

כל יחס בסכמה המוצעת לעיל צריך להיות לפחות ב-3NF.

משימתך (א) לציין את כל התלויות הפונקציונליות הלא-טריוויאליות שאמורות להתקיים בכל אחד מסכמות היחסים הנ"ל.

(ב) לענות לשאלה האם כל אחת מהן נמצאת לפחות ב-3NF (ולנמק את תשובתך).

(ג) בכל אחד מהמקרים שהתשובה ל-(ב) שלילית, משימתך לנרמל את סכמות היחסים האלה כך שיהיו לפחות ב-3NF.

הערה: נושא נירמול הסכמה הלוגית של בסיס נתונים טבלאי כבר נלמד בסמסטר א', בקורס "בסיסי נתונים". בכל זאת, חומר על הנושא אפשר למצוא [כאן](#), [כאן](#), [וכאן](#).

הגדרת הסכמה ב-Oracle:

משימתך להכניס ל-Oracle הגדרת כל היחסים של הסכמה שהגעת אליה אחרי הנימול, כולל מפתחות ראשיים ותנאי אמינות (integrity constraints, I-domain constraints).

הערה חשובה בקשר לסכמת בסיס הנתונים שלך על כל רמותיה:

תרשימי ה-ERD ו-DSD היחידים שיתקבלו יהיו רק אלו שצוירו באמצעות ה-DDS Lite. חשוב מאוד לציין שה-script של פקודות create table שמיישם את הסכמה הלוגית, ומופק באופן אוטומאטי על ידי ה-DDS-Lite, חייב לעבור עריכה סופית: (א) חייבים למחוק דברים מיותרים שה-DDS-Lite מוסיף ל-script, כגון פקודות grant ופקודות commit.

(ב) חייבים למחוק את השורות שבהן כל מה שמופיע הוא התוו "/", ולהוסיף את התוו ";". מייד אחרי הסוף של כל פקודת create table.

חשוב מאוד לציין גם שלפני העריכה הסופית של ה-script של פקודות create table, חייבים לבדוק שהסכמה הטבלאית מנורמלת ב-3NF – אם מגיעים למסקנה שהיא לא מנורמלת ב-3NF, חייבים לערוך את ה-script בהתאם כך שהטבלאות שיוגדרו באמצעותו תהינה כולן מנורמלות ב-3NF.

הערה: אם יש לך צורך לרענן את הידע שלך בקשר לפקודת create table, תסתכל [כאן](#).

הפקת דו"ח על הסכמה הלוגית שהוגדרה במערכת ניהול בסיסי הנתונים:

אחרי ביצוע הגדרת הסכמה ב-Oracle, משימתך לוודא שכל ההגדרות בהתאם למה שהתכוונת; לצורך זה תפיק דו"ח עם המידע המתאים לכל אחת מהטבלאות שהגדרת. הפקת הדו"ח הזו תיעשה באמצעות סביבת הפיתוח PL/SQL developer.

הערות:

1. מובן מאליו שכל אחת מהתלויות הפונקציונאליות שצינת בסכמה חייבת להיות מתאימה למציאות הבעיה הנדונה.
2. אם ברצונך לשנות משהו בסכמה, עשה את זה עכשיו.

טעינת נתונים לבסיס הנתונים:

- באמצעות פקודות insert: משימתך לכתוב script של פקודות insert שטוענות טבלה אחת של בסיס הנתונים, לפי בחירתך, ולהריץ אותו מתוך PL/SQL developer.
 - הערה: אם יש לך צורך לרענן את הידע שלך בקשר לפקודות insert, תסתכל [כאן](#).
 - ל-PL/SQL developer יש שלושה כלים לטעינת נתונים בכמויות (bulk loading): משימתך לטעון את יתר הטבלאות תוך שימוש בכלים אלה:
 - לטעינת חלק מהטבלאות, תשתמש ב-Data Generator של PL/SQL developer שבאופן אוטומאטי, וכמעט רנדומאלי, ייצור נתונים לטעינה.
 - לטעינת חלק אחר מהטבלאות, תיצור קבצי טקסט (בפורמט csv) עם הנתונים שברצונך לטעון, ותשתמש ב-Text Importer של PL/SQL developer.
 - לטעינת חלק אחר מהטבלאות, תיצור קבצי Excel עם הנתונים שברצונך לטעון, ותשתמש ב-ODBC Importer של PL/SQL developer.
- הערה: אם לצורך המיני-פרוייקט שלך יש לך נתונים אמיתיים, בכמות וגם בתוכן, אפשר לדלג על ה-Data Generator ולבצע את הטעינה באמצעות ה-Text Importer ו/או ה-ODBC Importer.

השוואה בין נתוני הקלט לבין תוכן הטבלאות:

משימתך לכתוב ולהריץ פקודות `select *` על כל הטבלאות, כדי לוודא שהנתונים הוכנסו בצורה נכונה.

ביצוע יצוא (מעין גיבוי) ויבוא (מעין שיחזור) של בסיס הנתונים שלך:

אחרי טעינת הנתונים, עליך לגבות את כל מה שיש בבסיס הנתונים שלך עד עכשיו, באמצעות פונקציית ה-Export שניתן לבצע מתוך ה-PL/SQL developer. לאחר מכן, תכתוב script של פקודות delete ותבצע אותו באמצעות PL/SQL developer על מנת לרוקן את כל הטבלאות של בסיס הנתונים שלך, ותשחזר אותו למצבו של בסיס הנתונים שהיה לפני בצוע ה-delete-ים באמצעות פונקציית ה-Import שניתן לבצע מתוך ה-PL/SQL developer.

הערה חשובה:

כסביבות פיתוח של שפות תכנות למיניהן, ל-PL/SQL developer יש אמצעי לניהול פיתוח של פרויקטים (ראה [פרק 13 של המדריך למשתמש של PL/SQL developer](#)). מומלץ כבר עכשיו ליצור פרויקט ב-PL/SQL developer ובמסגרתו ללוות את כל תהליך הפיתוח של המיני-פרוייקט שלך (לא כולל שלב מס' 4).

הערה כללית:

תשמור את פתרונותיך לכל הסעיפים של כל השלבים של המיני-פרויקט הזה, כולל תשובות מפורטות ומנומקות לכל השאלות המופיעות במשימות השונות. כל זה יהיה חלק של דו"ח המיני-פרויקט.

שלב מס' 2 – מועד הגשה: 2/5/2013 – כ"ב אייר תשע"ג

שאלות:

בסיס הנתונים שבנית בשלב הקודם של המיני-פרויקט שלך אמור לשמש מקור מידע למשתמשים שונים בארגון. **משימתך:**

- (1) לכתוב באופן מילולי שמונה (8) שאלות **לפחות**, שאמורות לספק את צרכי המשתמשים.
- (2) לכתוב בשפת SQL את כל השאלות שכתבת באופן מילולי ב-(1), ולהריץ אותן מתוך PL/SQL Developer.
- (3) למדוד את זמן ביצוע של כל השאלות שלך. באופן סטנדרטי, מקבלים את הנתון הזה ב- **SQL Window** של PL/SQL developer, אבל ניתן גם להשתמש ב- **Command Window** (שהוא דימוי האפליקציה הבסיסית ביותר של Oracle, שנקראת SQL*Plus), תוך שימוש בפקודת "**set timing on**" (כדי לבטל את ה-timing, השתמש בפקודת "**set timing off**").

הערה מאוד חשובה: השאלות צריכות להיות רלוונטיות לנושא המיני-פרויקט וגם מורכבות; כלומר, הן צריכות להיות עם צרופים (joins), עם תת-שאלות (sub-queries), עם מיונים (עם order by), עם פונקציות סטטיסטיות (aggregate functions כגון sum, average, וכו'), עם group by – having, וכו'.

הערה: כתיבת שאלות ב-SQL כבר נלמד בסמסטר א' בקורס "בסיסי נתונים". בכל זאת, אם יש לך צורך לרענן את הידע שלך בשפת SQL, תסתכל [כאן](#), [כאן](#), [כאן](#), [כאן](#), ו**כאן**.

אינדקסים:

ידוע לך שהשימוש באינדקסים יכול ליעל באופן משמעותי את ביצוען של שאלות (וגם של עדכונים). אם אנו מגדירים אינדקס על עמודה מסוימת בטבלה מסוימת, זה יאפשר ל-DBMS למצוא מהר יותר את כל השורות בטבלה שבאותה העמודה יש להן ערך מסוים (זה שימושי במיוחד כאשר מחשבים תנאי חיפוש או תנאי צירוף בהם מעורבת העמודה הזאת). באמצעות פקודת create index ניתן להגדיר אינדקס על כל עמודה בטבלה, או על צירוף של עמודות (זה ברור שכאשר אנו מגדירים עמודה - או צירוף של עמודות - כמפתח ראשי, המערכת בונה באופן אוטומטי אינדקס על העמודה הזאת). **משימתך:**

- (1) להגדיר **לפחות** שלושה אינדקסים שאתה מעריך שיוסיפו יעילות לביצוען של השאלות שכתבת בסעיף "שאלות" דלעיל, וזאת בנוסף למפתחות הראשיים שבשבילם המערכת יוצרת אינדקס באופן אוטומטי.
- (2) לבצע את אותן השאלות שכתבת בסעיף "שאלות", תוך כדי מדידת זמן ביצוען.
- (3) להשוות בין מדידת זמני הביצוע של השאלות בלי האינדקסים לעומת זמני הביצוע עם האינדקסים (בהנחה שהגדרת אינדקסים רלוונטיים לשאלות שלך). לטובת איזה משני הביצועים יש הבדל? (אם אין הבדל במדידות, הוסיף יותר נתונים לטבלאות הרלוונטיות ובצע את כל התהליך מחדש).

עידכון בסיס הנתונים:

בשימוש היומיומי של בסיסי נתונים מתבצעים הוספה, ביטול ועדכון של מופעים. **משימתך:**

- (1) לבצע שני עדכונים לבסיס הנתונים (כלומר, כתוב שתי פקודות **update**).
- (2) לבצע שני ביטולים מבסיס הנתונים (כלומר, כתוב שתי פקודות **delete**).

הערה: אם יש לך צורך לרענן את הידע שלך בקשר לפקודות delete ו-update, תסתכל [כאן](#).

commit ו-rollback:

כל השינויים שנעשים בבסיס הנתונים הם שינויי ביניים. ניתן להפוך שינויים אלה לסופיים ובלתי חוזרים באמצעות **commit**. ראוי לציין שה-**commit** מתבצע (א) באופן אוטומטי בסוף ה-session, או (ב) כתגובה לבקשה מפורשת של המשתמש באמצעות ביצוע פקודת commit. לעומת זאת, ניתן לבטל בכלל את כל השינויים שנעשו בבסיס הנתונים מאז ה-commit האחרונה באמצעות **rollback**. ראוי לציין שה-rollback מתבצע (א) באופן אוטומטי במקרה של תקלת מערכת כלשהי, או (ב) כתגובה לבקשה מפורשת של המשתמש באמצעות ביצוע פקודת rollback. ובכן, **משימתך:**

- (1) לבצע שליפת נתונים כלשהי (כלומר, select ... from ... where)
- (2) לבצע מספר שינויים בנתונים שעליהם בצעת את השאילתא ב-(1).
- (3) לבצע פקודת rollback.
- (4) לבצע שוב את אותה השאילתא שבצעת ב-(1).
- (5) להשוות בין תוצאות השאילתא ב-(1) לבין תוצאותיה ב-(4). תאר בקצרה את מה שקרה לבסיס הנתונים; פרש ונמק את התופעה הזאת.
- (6) תחזור על כל השלבים, מ-(1) ל-(4), אבל השתמש בפקודת commit במקום rollback. גם כאן תשווה בין תוצאות השאילתא השונות; כלומר, תאר בקצרה את מה שקרה לבסיס הנתונים, ופרש ונמק את התופעה.
- (7) בשפת SQL קיימת פקודה המאפשרת לך לתת למשתמשי Oracle אחרים גישה לטבלאות שבבעלותך; פקודה זו היא grant. באמצעות פקודת grant תרשה למשתמש oracle00 לשלוף נתונים מכל הטבלאות שמשתתפות בשאילתא שכתבת ב-(1).
- (8) במשולב עם השימוש ב-rollback וב-commit, תכנס ל-Oracle כ-oracle00 עם סיסמה ora00x; בתור שכזה, תבצע את השאילתאות שביצעת ב-(1) עד (6) דלעיל, תוך שאתה מבצע מתוך החשבון שלך את פקודות העידכון שביצעת קודם. מה קורה בתוצאת השאילתאות האלה לעומת התוצאות שאתה רואה כשאתה מריץ אותן מתוך החשבון שלך?

הגדרת תנאי אמינות (constraints) נוספים:

- בזמן הגדרת הסכמה הלוגית (logical schema) של בסיס הנתונים שלך, הגדרת מספר תנאי אמינות (מפתחות ראשיים, מפתחות זרים, וכו'). בטוח יש תנאי אמינות שבסיס הנתונים שלך חייב לקיים בנוסף לאלה שכבר הגדרת. בשפת SQL, הכלי העומד לרשותנו כדי לשנות הגדרה של טבלה (בין היתר, כדי להוסיף תנאי אמינות - כגון check constraint, unique constraint, וכו') הוא פקודת alter table. ובכן, משימתך:
- תוסיף תנאי אמינות שלשיקול דעתך חייבים להיות מוגדרים על טבלאות בסיס הנתונים שלך ושעדיין לא הוגדרו.
 - תבחר באחת מהטבלאות שעברו שינוי, ותבצע עליה פקודות הכנסת, עדכון וביטול נתונים (insert, update, delete) שאתה מצפה לקבל הודעות שגיאה מהמערכת בגלל התנגדותן לתנאי האמינות שהגדרת.
 - תן הסבר לאחת מהודעות השגיאה שקיבלת מהמערכת.

הגדרת מבטים (views):

- הסכמה הלוגית של בסיס הנתונים שלך, היא מבט כולל של מבנה המציאות שבסיס הנתונים אמור לשקף. למעשה היא שילוב של מבטים חלקיים (sub-schemata) שכל אחד מהם מבטא השקפה של אוכלוסיית משתמשים מסוימת על אותה המציאות; מבט חלקי מהווה את בסיס הנתונים מנקודת מבטה של אותה אוכלוסיית משתמשים. בשפת SQL, הכלי העומד לרשותנו כדי להגדיר מבטים הוא פקודת create view. ובכן, משימתך:
- להגדיר מבטים רלוונטיים לצרכי המידע של שתי אוכלוסיות משתמשים מסוימות - לפחות שני מבטים לכל אחת מהאוכלוסיות האלה.
 - לכתוב ולבצע לפחות שתי שאילתאות על המבטים האלה.
 - לנסות לבצע, על כל אחד מהמבטים האלה, שתי פקודות הכנסת, עדכון וביטול נתונים, כך שבחלק מהן המערכת תוציא הודעת שגיאה על אי יכולתה לבצע אותן.
 - תן הסבר לאחת מהודעות השגיאה שקיבלת מהמערכת.
- הערה: אם יש לך צורך לרענן את הידע שלך בקשר לפקודת create view, תסתכל [כאן](#).

שלב מס' 3 – מועד הגשה: 23/5/2013 – י"ד סיוון תשע"ג

שימוש ביכולות נוספות של PL/SQL developer:

PL/SQL developer מכיל יכולות נוספות לאלו של שפת השאילתאות SQL. ובכן, משימתך: לבחור בארבע מהשאילתאות שכתבת במסגרת המיני-פרויקט שלך, ולהדגים בהן את השימוש ביכולות הנוספות האלו, כגון שימוש ב**משתני החלפה (substitution variables)** המאפשרים כתיבת שאילתאות שמקבלות פרמטרים, פונקציות המאפשרות עיבוד ערכים למיניהם, פקודות המאפשרות הצגת תוצאת חישוב שאילתא כדו"ח לכל דבר, וכו'. לצורך משימות אלו, תשתמש ב-SQL Window ובפונקציית ה-Report שב-PL/SQL developer.

כתיבת תכניות בשפת PL/SQL:

תוצאה של שאילתא ב-SQL היא תמיד טבלה; כלומר, היא תמיד *קבוצת שורות*. באופן דומה, פקודת ביטול או עדכון מבטאת ביטול או עדכון של *קבוצת שורות*. בנוסף לזה, לכל פקודת SQL, כעקרון, אין שום קשר עם הפקודות שהתבצעו לפניה ועם אלה שתתבצענה אחריה. לפעמים נהיה מעוניינים לבצע פעולות כרצף לוגי אחד, או לטפל בכל שורה בנפרד, מה שב-SQL לא ניתן לעשות. לעומת זאת, שפת התכנות PL/SQL, המכילה את SQL כחלק ממנה, מאפשרת לנו להגדיר תכניות ופרוצדורות/פונקציות כרצף לוגי של פקודות בהן ניתן לטפל בכל שורה בנפרד. ובכן, *משימתך*:

- (1) לכתוב תכניות PL/SQL לצורך ביצוע פעולות על בסיס הנתונים שלך; כל אחת מהתכניות האלה חייבות להיות מספיק מסובכות כדי שיהיה לך צורך להשתמש במשתנה מקומי אחד, יותר מפקודת SQL (שאילתא, עדכון או ביטול) אחת; בנוסף לזה, בלפחות אחת מהתכניות האלו אתה חייב להשתמש ב-cursor אחד לפחות, בלולאה/לולאות מתאימה/מתאימות, ובהסתעפויות (אם יש צורך בזה).
 - (2) לכתוב לפחות שתי פרוצדורות/פונקציות שתשרתנה את התכניות שכתבת ב-(1).
 - (3) לכתוב לפחות שני triggers למיניהם, לפי צורך הפעולות שברצונך לבצע במשולב על מספר טבלאות בבסיס הנתונים שלך.
- לצורך משימות אלו, תיצור project-ים ב-PL/SQL developer (אם עד עכשיו לא עשית את זה) ובמסגרתם תפתח, תקמפל ותריץ את התכניות שתכתוב. ברור שכאן יהיה צורך בשימוש בהרבה מכלי הפיתוח הכלולים ב-PL/SQL developer.

שלב מס' 4 – מועד הגשה: 13/6/2013 – ה' תמוז תשע"ג

הכרת Embedded SQL:

משימתך לכתוב אפליקציה (חלונאית ב-Windows או בכל פלטפורמה אחרת, או web-ית) שניגשת לבסיס הנתונים שבניתם בשלבים הקודמים. אפליקציה זו תשמש להפעלת שאילתות, פקודות DML, פרוצדורות ופונקציות PL/SQL, ו-script-ים שכתבתם בשלבים קודמים.

בקשר ל-GUI של האפליקציה:

נאפשר חופש מוחלט לעיצוב ה-GUI בתנאי שהוא ישרת את צורכי המידע של כל משתמש סביר של בסיס הנתונים שבניתם. דוגמה ב-Java ניתן למצוא [כאן](#). דוגמה ב-.NET ניתן למצוא [כאן](#) [וכאן](#).

בקשר לשפת תכנות למימוש השלב הזה:

נאפשר בחירה חופשית של שפת תכנות למימוש שלב זה של המיני-פרוייקט בהתאם למגבלות הבאות:

- (1) שתי השפות העדיפות למימוש השלב הזה הן Java ו-C#, בגלל העובדה ששתיהן נלמדו בקורסים קודמים או נלמדות בקורסים מקבילים לקורס הזה. גם ++C יכולה לבוא בחשבון.
- (2) אם תבחרו לכתוב אפליקציה web-ית, יהיה ניתן להשתמש בשפות כגון php, asp, jsp, ...
- (3) נאשר שימוש בשפה אחרת בתנאי שהזוג שיבחר להשתמש בה יוכיח בקיאות בה.

הערה חשובה:

שלב זה יוגש (ויודגם) בזמן הגנת המיני-פרוייקט.