<u>סיכום שיעור שני</u>

משעורי הבית:

** **חשוב מאד לשים לב** שהפעולות מתרחשות בסדר מסויים , AND מגיע לפני OR, אלא אם כן נשתמש בסוגריים ואז אלו יפתרו ראשונות.

** **יש לשים לב טוב** מה מבקשים מאיתנו אפילו לשרטט לנו מה אנחנו רוצים להשיג.

** <u>שאלה 6 תרגול 2:</u> נשתמש בשורה האחרונה ב **IN** כדי לדייק את התואות שלנו, כלומר לקבל רק את הרשומות הכוללות את הערכים שהזנו בסוגריים.

OR(au_salary>10000 AND au_sex IN ('M','F'))

במידה ויש לנו ערכי **NULL** או ערכים שאנחנו לא מחפשים כמו לדוגמה: מין - "U"(מהתרגול) אנחנו נשתמש בפעולה IN, ונקבל בדיוק את התוצאות שאנחנו מחפשים, למשל : 'M' 'F'.

<u>שאלה תרגול 2:</u> "רוצים לקבל את כל מי שלא למד פסיכולוגיה", אפשר לפתור בשני דרכים, האחת בעזרת **NOT IN שהוא פחות מתאים למקרה זה (יתאים יותר מרשימות), הדרך השניה תיהיה להשתמש בסימן שונה **<>,** פעולה עדיפה למקרה זה.

**חשוב לציין- יש פעולות בSQL שעושות את אותם הדברים ולפעמים ניתן להשתמש ביותר מדרך אחת לקבל את אותה תוצאה.

פעולות SELECT

****פעולות SELECT כשמם כן הן, נעשות בעמודת הSELECT.

TOP - נקבל את מספר הרשומות הראשונות, נקבע את הכמות של הרשומות שאנחנו רוצים למצוא, אם נרשום רק מספר אחד אחרי ה TOP , נקבל רק את הרשומות הראשונות הסיבה שנעשה את הTOP הזה, יכול לשמש אותנו בעת ביצועים של הרבה שינויים על השאילתא ובטבלה מאד גדולה ככה שיקח זמן רב לעבד על כל הרשומות את המידע החדש. הפתרון הלחיל את השינוים רק על כמות קטנה של רשומות בעזרת הTOP.

**אם נבצע את פעולת הSORT BY שנלמד בהמשך נוכל להביא את הערכים הגבוהיים ביותר בתלות לערכים. למשל לקבל את שלושת התלמידים עם הציונים הגבוהים ביותר.

דוגמה:

```
SELECT
TOP 10 UserID , FirstName , LastName
FROM Users
```

פלט:

מקבלים את 10 הערכים הממוקמים בראש הטבלה

דוגמה:

```
SELECT TOP 1 *
FROM Users
```

פלט:

נקבל את הרשומה הראשונה שהוזנה בDATA BASE שלנו. מי שהוא ID=1

DISTINCT - בחירת ערכים ייחודיים סינון כפולים, מחזיר לנו רק ערכים שהם שונים, אם אני מבצע שליפה של נתונים יש נתונים שהם זהים, אם נרצה לדעת רק מה הערכים היחודיים נשים את הDISTINCT .

הפעוה תמחוק רשומות זהות **מהתצוגה. הרשומות צריכות להיות זהות בSELECT, יכולים להיות צירופים שונים לנתונים שונים ולכן גם הם יתקבלו בתצוגה. לכן לעשות * ב SELECT יחד עם הDISTICNT יהיה מיותר כי כאשר נקרא לכל העמודות בהכרח יהיה שוני מסויים בניהן.

**דוגמה, עמודה עם ערכים: בן בת, ועמודה שניה עם גבהים:גבוה, בינוני, נמוך אם נעשה DISTINCT על המין, נקבל שני רשומות בלבד:בן,בת.

אם נשים גם את הגובהה בנוסף , נקבל שישה רשומות: בן גבוהה, בת גבוה, בן בינוני, בת בינונית, בן קטן, בת קטנה. **כל הצירופים הייחודיים.**

דוגמה:

```
SELECT DISTINCT FirstName
FROM Users
```

פלט:

נקבל רק את השמות הייחודיים, לא יהיו לנו כפילות של שמות ראשונים.

```
SELECT DISTINCT first name, maritalstuts
FROM users
WHERE firstname IN('Joe','Abbey','Adam')
```

פלט:

נקבל שלושה רשומות, על כל אחד מהשמות(ערכים ייחודיים).

נקבל רק את הערכים הייחודיים, לא יהיה לנו בכלל ערכים שרשומים פעמיים.

<u>DISTINCT על שני ערכים או יותר</u>

דוגמה:

```
SELECT DISTINCT first name, maritalstuts
FROM users
WHERE firstname IN('Joe','Abbey','Adam')
```

פלט:

נקבל את כל אחד מהשלושה שמות בשילוב של הסוגי maritalstuts שיש, במקרה שלנו זה 2 , נקבל שישה רשומות.

** אם נרשום * SELECT DISTICT first_name נקבל את כל השמות כנראה כיוון שהייחודיות צריכה להתרחש בכל העמודות , צריך שהיו שני אנשים בדיוק אותו דבר בכל הפרמטרים על מנת שהDISTINCT יבטל אחד מהם.

AS- לתת כינויים לראש של עמודות, רלוונטי לפונקציות, אני יכול לשנות גם שם של טבלה עם הפעולה הזאת. בסיכומו של דבר היא מאפשרת לנו לשנות רק **לקוד** את השם של הטבלה או העמודה, יהיה שימושי בפונקציות.

**לאחר ביצוע שינוי שם של עמודה או של טבלה עם פעולת ה AS (והוא נמצא תחת הSELECT) נשתמש בהמשך הקוד עם השם המקורי של העמודה או הטבלה, כיוון שפעולת הSELECT מבוצעת אחרונה בקוד לאחר כל השאילתא, לכן ה AS יוצא לפועל רק בסוף סדר הפעולות ולא ישפיע על הקוד שתחתיו.

****לשים לב** מבחינת סדר ביצוע הקוד, הSELECT מתבצע **בסוף** ועל כן לא נוכל להשתמש בשם חדש שהגדרנו לעמודה בWHERE , כיוון שהוא מתבצע לפני.

** ניתן להגדיר לטבלה שם חדש או קיצור על מנת להשתמש בו בקוד, לא שפיע כל הטבלה ב DATA BASE

פונקציות המבוצעות בשורת ה- SELECT וגם בשורת ה- WHERE

פונקיציות שורה - עובדות על כל שורה בנפרד, יבצעו את השינוי על כל השורות. ישנם פונקציות אשר מבצעות פעולה תצוגתית אבל גם ניתן להתנות עליהם, מטרתן העיקרית של השימוש בפונקציות הוא לבצע מניפולציות על הDATA BASE לפני התצוגה שלו, כמו חישובים או שליפות ספציפויות שנרצה לבצע.

** את פונקציות השורה ניתן להשתמש בSELECT על מנת להשיג תצוגה של מידע מסויים, וכן ניתן להשתמש בתור התניה בעזרת פעולת הWHERE, מאפשר לנו לשלוף רק את המידע שעונה על השאלה המסומיית שלנו, למשל תן לי את כל האנשים שנולדו בינואר, ואז נוכל לבצע את החיפוש בתוך פעולת הWHERE?.

להלן הפונקציות:

YEAR , MONTH, DAY – פונקציות המחלצות מרכיב מתוך תאריך

LEN - פונקציה אשר מחזירה אורך של טקסט

LEFT, RIGHT,MID – פונקציות אשר חותכות מספר תווים מתוך טקסט

-LEN בודק לנו את האורך של הסטרינג,(שם עמודה או טקסט)len ואז נוסיף עמודה חדשה לקוד, ולא לDB, היא תתן לי כמה תווים יש בכל אחת המערכים במעמודה שעליה עשית LEN, אם נשים טקסט זה יהיה בגרשיים " ונקבל כמה אותיות יש במילה.

**יש לשים לב שזה יפתח לנו עמודה חדשה אז למען הסדר נשתמש ב AS להגדיר את ראש העמודה. כוון שבפונקציות שורה הכותרת משתנה כי הנתונים כבר עוברים עיבוד וזה כיביכול מידע חדש לגמרי לטבלה ולכן אין לעמודה הזו שם. נקפיד להגדיר שמות בעזרת הAS

** רווח נחשב גם כתו בחיפוש של LEN

```
SELECT
FirstName , LEN (FirstName) AS Length
FROM Users
```

פלט:

נקבל את מספר התויים בתוך כל סטרינג, כלומר את מספר האותיות בתוך כל שם פרטי במקרה הזה

דוגמה:

```
SELECT
FirstName , LEN (FirstName) AS Length
FROM users
WHERE LEN (FirstName  > 3)
```

פלט:

נקבל את כל הרשומות שבהם שם הפרטי הוא שלושה אותיות.

פונקציית LEFT ו- RIGHT

פונקציות אשר חותכות מספר תווים מצד מוגדר מצד ימין או

מצד שמאל של הטקסט. כמה תוים להביא מצד שמאל, או כמה תוים מצד ימין, יש להזין שני ערכים

בפונקציה הזו, LEFT(firstname,2) - נקבל את שני האותיות משמאל לערך.

LEFT \ RIGHT- יש לשים שני ערכים (מספר החיתוך, שם עמודה)LEN

ניתן גם לצור התניה עם הפונצקציה :

LEFT(firstname,2) = 'Jo'

פלט:

נקבל את כל האנשים ששני האותיות שלהם משמאל הם "Jo", ישנה חשיבות לאותיות גדולות או

קטנות לכן ננבה לדייק את המידע לזה הקיים בDB(DATA BASE)

**עושה שינוי לכל הנתון של העמודות.

**לשים לב שים ב SELECT את שם העמודה פעמיים זה עושים כדי שנוכל לראות את השם ואת

האורך השם לידו, אחרת נקבל רק את האורך.

פונקציית MID -פונקציות אשר חותכות מספר תווים מוגדר מנקודת התחלה מוגדרת. צריכים

שלושה ערכים בסוגריים החלק הראשון העמודה עליה נרצה לבצע את הפונקציה, חלק שני מאיזה תו

אנחנו רוצים לחתוך, חלק שלישי כמה אנחנו רוצים לחתוך.

הפונקציהעובדת בתבנית הבאה:

Mid(string, start_position, number_ to_ slice)

דוגמה:

```
SELECT Email, MID (Email, 4 , 3) AS Mid_ 4 3
FROM Users
WHERE MID (Email, 4 , 3 ) Like  '%@%'
```

פלט:

נקבל רק את הרשומות בהן יש לאימייל @ באחת האותיות ה 4 5 6 חתכנו 3 תוים ממספר 4.

**ניתן בנוסף להשתמש בפונקציה זו בפעולת הWHERE ואז ליצור התניה עם הפונקציה.

דוגמה:

```
SELECT Email, MID (Email, 5 , 1) AS Mid
FROM Users
WHERE MID (Email, 5, 5 ) = a
```

פלט:

** אם שמים * בפונקציה זה לא יעבוד, אם נרצה את כל העמודות נשים **users.*** בפעולת ה SELECT. ואז נקבל את הכל העמודות.

**כל פונקציות השורה הנ"ל ניתן להשתמש בהן בSELECT, וגם להתנות אותם בWHERE

** העמודה של הפונקציות תצורף בסוף הטבלה.

פונקציות DAY, MONTH, YEAR

פונקציות המחלצות מרכיב מתוך תאריך – YEAR , MONTH, DAY לתוך הפונקציה נזין תאריך והתוצאה שנקבל תהיה הרכיב מאותו תאריך.

פונקציות שמחלצות מידע מסויים בעמודות תאריך:

DAY- נותן את המספר של היום בתאריך

MONTH- נותן את החודש

YEAR- נותן את השנה.

דוגמה:

```
SELECT Birthday, YEAR (Birthday) AS Bday_Year
FROM Users
```

פלט:

נקבל את השנה של כל יום הולדת

דוגמה:

```
SELECT Birthday, DAY (Birthday) AS Bday_Day

FROM Users
```

פלט:

נקבל את היום בכל יום הולדת של כל הרשומות.

```
SELECT Birthday, MONTH (Birthday) AS Bday_Month

FROM Users

WHERE MONTH (Birthday ) = 1
```

פלט:

אקבל את כל הרשומות בהן החודש הוא החמישי.

תשובות שעורי בית:

--Question No.3

```
--Q1
SELECT TOP 3 `Grade`
FROM `Grades`

--Q2
SELECT `Course_name` AS Name
FROM `Courses`

--Q3
SELECT DISTINCT  `Grade`
FROM `Grades`

--Q4
SELECT DISTINCT `species`, `house_id`
FROM `teachers`
```

--Question No.4

```
/*Q1*/
SELECT teacher_name ,
        DAY(Date_of_birth) AS BDay,
        MONTH (Date_of_birth) AS BMonth,
        YEAR (Date_of_birth) AS BYear,
        Date_of_birth
FROM `teachers`
```

```
/*Q2*/
SELECT *
FROM `teachers`
WHERE YEAR(`Date_of_birth`) > 1930

/*Q3*/
SELECT *
FROM `teachers`
WHERE DAY(`Date_of_birth`) BETWEEN 10 AND 20

/*Q4*/
SELECT *
FROM `teachers`
WHERE MONTH (`Date_of_birth`) <> 5

/*Q5*/
SELECT *
FROM `Students`
WHERE LENGTH(`Student_Name`) = 3
```

```
/*Q6*/
SELECT `phone` , MID(`phone`,5,3)
FROM `Students`
WHERE `Address` IS NOT NULL

/*Q7*/
SELECT DISTINCT LEFT(`EMail`,1)
FROM `Students`

/*Q8*/
SELECT s.ID
,s.student_name
,s.address
,s.phone
,s.email
,s.houseid,
CASE WHEN `EMail` LIKE '%Hogwarts.ac'
THEN '1'
ELSE '0'
END AS is_hogwarts_email
FROM `Students` AS s
```