

# מסדי נתונים – 89-281

## תרגול 7 - SQL 1 - פקודות בסיסיות

עמיעד רוזנברג

How many Microsoft programmers does it takes to change a light bulb?  
0. They will make darkness a standard and force everyone to use it.

# SQL (Structured Query Language)

- SQL היא שפת שאילתות המבוססת על המודל היחסי ומאפשרת גישה נוחה לבסיסי נתונים.
- שפת SQL מתחלקת ל-2 מרכיבים :
  - Data Definition Language – DDL.  
שפת הגדרת נתונים - משמשת לטיפול במבנה הנתונים.
  - Data Manipulation Language – DML.  
שפת הטיפול בנתונים – משמשת לטיפול בנתונים עצמם.
- בשלב זה נתעסק עם DML.

# SQL - מוסכמות

- כל פקודה מסתיימת עם ";" .
- מקובל להפריד את פקודות ה SQL משמות הטבלאות והשדות ע"י שימוש באותיות גדולות בשביל פקודות ה SQL (מלים שמורות) ובאותיות קטנות עבור שמות וערכים.

# SQL - מבנה בסיסי

- המבנה הבסיסי של בטוי ב SQL מכיל 3 פסוקים : SELECT, FROM ו - WHERE :
- הפסוק SELECT מקביל לפעולת ההטלה באלגברת היחסים ומשמש לפירוט התכונות המבוקשות בתוצאה.
- הפסוק FROM כולל רשימת יחסים שבהם יש להשתמש בזמן חישוב הבטוי (מקביל למכפלה קרטזית).
- הפסוק WHERE מקביל לפעולת הבחירה באלגברת היחסים. כולל פרדיקט שמציין אילו שורות לבחור.

# SQL - מבנה בסיסי

- שאלתא אופיינית ב-SQL תיראה כך :

- $\text{SELECT } a_1, a_2, \dots, a_n$   
 $\text{FROM } r_1, r_2, \dots, r_m$   
 $\text{WHERE } p;$

- שאלתא זו באלגברת יחסים תיראה כך :

- $\pi_{a_1, a_2, \dots, a_n}(\sigma_p(r_1 \times r_2 \times \dots \times r_m))$

# פסוקי SELECT \ FROM

- פסוקי SELECT \ FROM הם פסוקי חובה בשאילתת SQL.

- דוגמא:

שלוף את שמות כל הסניפים והערים:

- SELECT branch-name, branch-city  
FROM branch;

branch	
branch-name	branch-city
Downtown	Brooklyn
Redwood	Palo Alto
Perryridge	Horseneck
Mianus	Horseneck
Round Hill	Horseneck
Pownal	Bennington
North Town	Rye
Brighton	Brooklyn

# פסוקי SELECT \ FROM

- שימו לב!

בניגוד לאלגברת יחסים, SQL מאפשרת שורות כפולות בטבלה המוחזרת. לכן, אם נרצה לא לאפשר שורות כפולות נוסיף DISTINCT לאחר SELECT.

- דוגמא:

שלוף את שמות כל הערים שיש בהם סניפים:

- ```
SELECT DISTINCT branch-city  
FROM branch;
```

- שימו לב שהצימצום מתבצע על כל ה-tuple.

# SELECT \ FROM פוקי

- SELECT branch-city  
FROM branch;

| branch      |
|-------------|
| branch-city |
| Brooklyn    |
| Palo Alto   |
| Horseneck   |
| Horseneck   |
| Horseneck   |
| Bennington  |
| Rye         |
| Brooklyn    |

- SELECT DISTINCT branch-city  
FROM branch;

| branch      |
|-------------|
| branch-city |
| Brooklyn    |
| Palo Alto   |
| Horseneck   |
| Bennington  |
| Rye         |



# פסוקי SELECT \ FROM

- אם לא נרצה לבחור עמודות מסויימות להצגה, אלא נרצה להציג את כל העמודות של הטבלאות מהם אנחנו שולפים נתונים, נשתמש בסימן \*.

- דוגמא :

הצג את כל פירטי הלקוחות :

- SELECT \*  
FROM customer;

| customer      |                 |               |
|---------------|-----------------|---------------|
| customer-name | customer-street | customer-city |
| Jones         | Main            | Harrison      |
| Smith         | North           | Rye           |
| Hayes         | Main            | Harrison      |
| Curry         | North           | Rye           |
| Lindsay       | Park            | Pittsfield    |
| Turner        | Putnam          | Stamford      |
| Williams      | Nassau          | Princeton     |
| Adams         | Spring          | Pittsfield    |
| Johnson       | Alma            | Palo Alto     |
| Glenn         | Sand Hill       | Woodside      |
| Brooks        | Senator         | Brooklyn      |
| Green         | Walnut          | Stamford      |

# פסוקי SELECT \ FROM

- פסוק SELECT יכול להכיל גם ביטויים אריתמטיים למיניהם... הפעולות +, -, \*, \ יכולות להתבצע על קבועים ותכונות של n-יות.

- דוגמא:

נניח שבסוף כל חודש יורדים ללקוחות דמי ניהול חשבון בסך 10 שקלים.  
נרצה לדעת מה יהיה סכום הכסף בחשבונות השונים בחודש הבא:

- SELECT account-number, balance - 10  
FROM account;

| account        |              |
|----------------|--------------|
| account-number | Balance - 10 |
| A-101          | 490          |
| A-215          | 690          |
| A-102          | 390          |
| A-305          | 340          |
| A-201          | 890          |
| A-222          | 690          |
| A-217          | 740          |

# פסוקי SELECT \ FROM

- ניתן ליצור עמודה המכילה מחרוזת קבועה כלשהי.  
כותרת העמודה תהיה זהה למחרוזת הקבועה.

דוגמא:

- SELECT branch-name, 'is in', branch-city  
FROM branch;

| branch      |       |             |
|-------------|-------|-------------|
| branch-name | is in | branch-city |
| Downtown    | is in | Brooklyn    |
| Redwood     | is in | Palo Alto   |
| Perryridge  | is in | Horseneck   |
| Mianus      | is in | Horseneck   |
| Round Hill  | is in | Horseneck   |
| Pownal      | is in | Bennington  |
| North Town  | is in | Rye         |
| Brighton    | is in | Brooklyn    |

# פסוק WHERE

- בחלק זה נוכל לבחור שורות המקיימות תנאי מסויים.
- במקום הקשרים הלוגיים שהיו לנו באלגברת יחסים –  
ב-SQL נשתמש ב-AND, OR, NOT.
- בתנאים, נשתמש באופרטורי ההשוואה: <, <=, >, >=, =, <, >.  
LIKE, BETWEEN ... AND ... (נדבר עליהם בקרוב).

# פסוק WHERE

- דוגמא :

מצא את כל מספרי ההלוואות שבוצעו בסניף Round Hill שסכום ההלוואה שלהם היה לפחות 1200.

- ```
SELECT loan-number  
FROM loan  
WHERE branch-name='Round Hill' AND amount>=1200;
```

- מחרוזות :

- מקיפים בגרש.

- ההשוואה נעשית ע"פ סדר לקסיקוגרפי.

# פסוק WHERE

- האופרטור BETWEEN ... AND ... :

- אופרטור זה מחזיר ערכים בתחום מסויים.

- WHERE a BETWEEN  $a_1$  AND  $a_2$

שקול ל:

WHERE  $a \geq a_1$  AND  $a \leq a_2$

- דוגמא:

הצג את מספרי ההלוואות בסכום שבין 1000 ל-2000 שקלים.

- SELECT loan-number  
FROM loan

WHERE amount BETWEEN 1000 AND 2000;

# פסוק WHERE

## • האופרטור LIKE

- אופרטור זה מאפשר להשוות בין מחרוזות.
- האופרטור ישווה בין המחרוזת לבין תבנית שנגדיר לו כאשר:
- '%' – מתאים למספר כלשהו של תווים (0 או יותר).
- '\_' – מתאים לתו אחד בדיוק.

## • למשל:

- התבנית 'Perry%' מתאימה ל -
- התבנית '%you%' מתאימה ל -
- התבנית '\_\_\_' מתאימה ל -
- התבנית '%\_\_\_' מתאימה ל -

# פסוק WHERE

## • האופרטור LIKE

• דוגמא :

מציאת שמות כל הלקוחות הגרים ברחוב ששמו מתחיל באות S :

- ```
SELECT customer-name  
FROM customer  
WHERE customer-street LIKE 'S%';
```

| customer      |                 |               |
|---------------|-----------------|---------------|
| customer-name | customer-street | customer-city |
| Adams         | Spring          | Pittsfield    |
| Glenn         | Sand Hill       | Woodside      |
| Brooks        | Senator         | Brooklyn      |



# פסוק WHERE

- האופרטור LIKE - הערות

- ההשוואה בברירת מחדל איננה case sensitive. כדי שההשוואה תהיה case sensitive יש להוסיף את המילה BINARY לפני התבנית.  
בדוגמא שלנו:  
מציאת שמות כל הלקוחות הגרים ברחוב שהאות השנייה שלו היא S:

- ```
SELECT customer-name  
FROM customer  
WHERE customer-street LIKE BINARY '_S%';
```

# פסוק WHERE

- האופרטור LIKE - הערות

- ב MySQL אפשר להשתמש ב LIKE גם עבור ערכים מספריים (זה בניגוד ל LIKE הסטנדרטי של SQL).  
דוגמא:

מצא את כל החשבונות שסכום הכסף בהם הוא בין 10,000 לבין 19,999

- ```
SELECT account-number  
FROM account  
WHERE balance LIKE '1 _ _ _ _';
```

- כדי לחפש מופעים של התווים המיוחדים במחרוזות נשתמש ב '\' לפני התו. לדוגמא: '%' מתאים לכל תת מחרוזת. '\%' מתאים לתו %.

- ניתן להשתמש גם ב- NOT LIKE.

# פסוק FROM עם מספר טבלאות

- בפסוק זה נרשום את שמות הטבלאות המשתתפות בשאילתא מופרדים בפסיק. כזכור, בין הטבלאות הללו מתבצעת מכפלה קרטזית.

- דוגמא:

שמות כל הלקוחות שלקחו הלוואה וסכום ההלוואה שלקחו:

- ```
SELECT customer-name, amount  
FROM borrower,loan  
WHERE borrower.loan-number=loan.loan-number;
```

# פסוק FROM עם מספר טבלאות

- פעולת AS

- מאפשר לתת שם חדש לטבלה.

- דוגמא:

הצג את מספרי כל ההלוואות שנלקחו ע"י לפחות 2 אנשים שונים.

- ```
SELECT DISTINCT b1.loan-number
FROM borrower    b1, borrower    b2
WHERE b1.loan-number=b2.loan-number AND
      b1.customer-name<>b2.customer-name;
```

- הערה – ניתן להשמיט את המילה AS.

# פסוק ORDER BY

- ניתן למיין את התוצאה המוחזרת ע"י שימוש בפסוק ORDER BY לאחר פסוק WHERE. צורת השימוש היא:

ORDER BY F<sub>1</sub> [ASC/DESC], F<sub>2</sub> [ASC/DESC].....

- ASC – מיון בסדר עולה. DESC – מיון בסדר יורד.  
אם לא נציין אף אחד מהאפשרויות – המיון יתבצע בסדר עולה.

# פיוק ORDER BY

• דוגמא:

הצג רשימה ממויינת ע"פ שם של כל הלקוחות שלקחו הלוואה  
בסניף "Perryridge"

- ```
SELECT DISTINCT customer-name  
FROM borrower, loan  
WHERE branch-name='Perryridge'  
      AND borrower.loan-number=loan.loan-number  
ORDER BY customer-name;
```

# פסוק ORDER BY

- אם נרצה – נוכל למיין גם לפי יותר מעמודה אחת.

- דוגמא:

הצג את רשימת פרטי הלוואות לפי סדר יורד של סכום ההלוואה.  
אם יש מספר הלוואות באותו סכום, מיין לפי סדר עולה של מספר  
ההלוואה

- ```
SELECT *  
FROM loan  
ORDER BY amount DESC, loan-number ASC;
```

# אז מה היה לנו?

- ראינו מבנה בסיסי

- SELECT

- FROM

- AS

- WHERE

- LIKE

- BETWEEN ... AND ...

- מיון – ORDER BY

- בתירגול הבא : פונקציות הקבצה.