SQL

- הקדמה ומילון מונחים
- יצירת יחסים והכנסת רשומות חדשות
- שליפת נתונים מתוך יחס אחד
- שליפת נתונים מתוך מספר יחסים
- פעולת הקבצה ופונקציות הקבצה
- ⇐ סיכום

SQL - סיכום

פסוקי מניפולציה בנתונים		
משפט	תחביר	הסבר
הכנסת רשומה	insert into *table_name* values ('*value*','*value*'...),(...),(...)...	בכדי להכניס רשומות חדשות לטבלה קיימת יש לרשום insert into *table_name* values, מיד לאחר ה-values יש לרשום את הערכים השונים בתוך סוגריים עגולים עם פסיק מפריד בין תכונה לתכונה. זהו כתיב מקוצר המחייב להכניס ערכים לכל התכונות לפי סדר הופעתן ביחס. אם רוצים להכניס מספר רשומות בבת אחת פשוט רושמים פסיק אחרי הסוגרים העגולים ופותחים סוגריים נוספים - כל סוגריים עגולים שכאלו הם שורה אחת ביחס.
עדכון רשומה	update *table_name* set *column_name*='*value*' where *column_name*='*value*'	עדכון של תכונות ספציפיות בתוך רשומה קיימת ביחס: update *table_name* set *column_name*='value'. אם לא משתמשים בתנאי where (המופיע מיד אחרי התכונות שרוצים לשנות) שבו מגדירים איזה שורות יושפעו מהשינוי, כל השורות ביחס ישתנו! אם יש כמה שורות ביחס המקיימות את התנאי - כולן ישתנו.
מחיקת רשומה	delete from *table_name* where *column_name*='*value*'	מחיקה של שורות שלמות הקיימות ביחס. אם לא משתמשים בתנאי where (המופיע מיד אחרי שם הטבלה ממנה מעוניינים למחוק) שבו מגדירים איזה שורות יימחקו, כל השורות ביחס יימחקו! אם יש כמה שורות ביחס המקיימות את התנאי - כולן יימחקו.

SQL - סיכום

שאילתות SQL		
מושג	**תחביר**	**הסבר**
הטלה	select	בחירה של עמודות מסוימות מתוך היחס הנמצא ב-from
מהיחס/ים	from	שם הטבלה/הטבלאות מהם השאילתה מבקשת לקרוא נתונים
תנאי על הרשומות	where	רק שורות המקיימות את התנאי יופיעו ביחס התוצאה (לא חובה להציב תנאי)
צמצום כפילויות	distinct	מצמצם שורות זהות. לדוגמא: **select distinct**(column_name) או לדוגמא: **select count** (**distinct** (column_name))
שרשור תנאים	and, or	and משמש לדרישה שכל התנאים יתקיימו, or משמש לדרישה שלפחות אחד מהתנאים יתקיים
צירוף טבעי	natural join	מחבר שני יחסים יחד לפי התכונה/ות בעלת/ות אותו השם. חיבור זה יוצר טבלה אחת גדולה המכילה את כל התכונות המופיעות בשני היחסים. תכונות בעלות אותו שם מקבלות רק עמודה אחת ולכן לא חייבים לכתוב בבחירה מאיזה יחס התכונה מבוקשת. תחביר: from table_A natural join table_B
מכפלה קרטזית	,	מחבר כל שורה ביחס הראשון עם כל שורה ביחס השני: **from** table_A , table_B. תכונות בעלות אותו שם לא מתבטלות וביחס התוצאה יש שתי עמודות שונות עבור תכונות אלו ולכן חייבים לכתוב מאיזה טבלה מבקשים את התכונה: **select** table_A.column_name
נתינת שם ליחס או לתכונה	as	מאפשר לקרוא ליחס בשם מקוצר. שימושי מאוד במכפלה קרטזית בין שני יחסים שהם אותו היחס: **select** A1.column_name **from** table_A **as** A1, table_A **as** A2 אפשר להשתמש גם אחרי שאילתה מקוננת ואז יש להגדיר את שמות העמודות של השאילתה: **Select** A.c1 **from** (**select** column_name1, column_name2 **from** table_A) **as** A(c1,c2))
מיון	order by	מיון התוצאות לפי תכונה/ות ספציפית/יות. ברירת המחדל היא מיון עולה (asc), למיון יורד (desc) יש לציין במפורש: **order by** column_name **desc**
פונקציות הקבצה (אסור להפעיל אחת על השניה)	count	מספר הערכים
	sum	סכום הערכים
	max	הערך המקסימלי
	min	הערך המינימלי
	avg	הערך הממוצע
פעולת הקבצה	group by	מחלק את הטבלה לקבוצות לפי תכונה/ות מסוימת/ות. כל התכונות המופיעות ב-select מחוץ לפונקציית הקבצה כלשהי, חייבות להופיע גם ב- group by. לא משתמשים בפעולת ההקבצה ללא שימוש באחת מפונקציות ההקבצה.
שאילתות מקוננות	all, any	**all** משמש לבדיקה אם ערך כלשהו גדול/קטן/שווה/שונה **מכל** הערכים שביחס המתקבל מהשאילתה המופיעה אחריו, **any** משמש לבדיקה דומה מערך **כלשהו** המופיע ביחס. לדוגמא: column_name >= **all**(**select** column_name **from** table_A)
	in, not in	משמש לבדיקה האם ערך כלשהו נמצא או לא נמצא ביחס המופיע אחריו. לדוגמא: column_name **in** (**select** column_name **from** table_A)
	exists, not exists	exists מחזיר "אמת" אם השאילתה המופיעה אחריו מכילה לפחות רשומה אחת. not exists מחזיר "אמת" אם השאילתה המופיעה אחריו ריקה לחלוטין.
פעולות על יחסים	union	איחוד שני יחסים יחד
	intersect	חיתוך שני יחסים (כל השורות המופיעות בשני היחסים גם יחד)
	except	חיסור יחסים (כל השורות המופיעות ביחס הראשון ולא מופיעות בשני)

שלושת הפעולות מצמצמות שורות כפולות. כדי להימנע מזה אפשר לכתוב לדוגמא: union all