סימונים בסיסיים	
AND	תנאים לוגיים
OR	
NOT	
BETWEEN	טווח
IN	רשימה של ערכים
LIKE	חיפוש ספציפי
IS NULL	בדיקה אם ערך ריק
Desc	מיון בסדר יורד
Asc	מיון בסדר עולה

סימונים בסיסיים	
>	גדול מ
<	קטן מ
>=	גדול שווה
<=	קטן שווה
=	שווה
<>	שונה
!=	לא שווה

פונקציות בסיסיות	
Sum()	חישוב סכום ערכים
Min()	הפונקציה מחזירה את הערך הנמוך ביותר
Max()	הפונקציה מחזירה את הערך הגבוה ביותר
Avg()	מחשב ממוצע
Count()	הפונקציה סופרת את מספר שורות

פונקציות שימושיות	
Now()	תאריך ושעה נוכחיים
Round(כמה ספרות להשאיר אחרי הנקודה, המספר)	עיגול מספר-
Concat(str1, str2)	חיבור מחרוזות
Length(str)	מחזיר את אורך המחרוזת
Lower/Upper	שינוי לאותיות קטנות/גדולות
Trim(str)	הסרת רווחים מיותרים
Substring(str, start, length)	קטע מהמחרוזת
Len()	מחזיר את מספר האותיות בביטוי

שימוש ב-%	
%abc%	טקסט שמכיל "abc" בכל מקום
%abc	טקסט שמתחיל ב- abc
abc%	טקסט שמסתיים ב- abc
%a_b%	טקסט מכיל "a", אחריו תו כלשהו ואחריו "b"
(% ללא)'abc'	בדיוק "abc"

- **כלל נרמול מסדר שלישי (3NF)** : בסיס נתונים הוא ב 3NF אם כל הטבלאות הן ב-2NF (כל שדה אטומי וכל שדה שאיננו במפתח תלוי במפתח) ובנוסף כל שדה בטבלה שאיננו במפתח תלוי **רק** במפתח.
  - אם לא כל השדות בטבלה תלויים בכל המפתח, הדבר מהווה בעיה בעדכון רשומות
- **בכדי להשיג נרמול 3NF**
  - עבור שדה שאינו תלוי רק בשדות המפתח ניצור טבלה חדשה שתכלול את השדות בהן תלוי השדה שאינם במפתח + תכונה לשדה התלוי

Except	Intersect	Union
רק הערכים שבשאילתה הראשונה ולא בשנייה	רק הערכים שמשותפים לשתי השאילתות	כל הערכים משתי השאילתות בלי כפילויות
SELECT field1,field2	SELECT field1,field2	SELECT field1,field2
FROM tbl1	FROM tbl1	FROM tbl1
EXCEPT	INTERSECT	UNION
SELECT field1,field2	SELECT field1,field2	SELECT field1,field2
FROM tbl1/2	FROM tbl1/2	FROM tbl1/2

מחבר בין טבלאות לפי עמודות משותפות – JOIN	
Inner join	מציג רק שורות שיש התאמה בשתי הטבלאות
Left join	הכל מהטבלה השמאלית+ התאמות מהימין
Right join	ההפך מ-left
Full join	כל מה שיש בכל הטבלאות

```
CREATE TABLE Orders (
    order_id INT PRIMARY KEY AUTO_INCREMENT,        -- מפתח ראשי
    customer_id INT,                                -- מפתח זר
    order_date DATE NOT NULL,
    total_amount DECIMAL(10,2),

    FOREIGN KEY (customer_id) REFERENCES Customers(customer_id)
);
```

PRIMARY KEY

מזהה ייחודי לכל שורה בטבלה

לא יכול להיות בו ערך כפול או ריק (null)

כל טבלה צריכה מפתח ראשי אחד לפחות

FOREIGN KEY

עמודה בטבלה שמצביעה על המפתח הראשי בטבלה אחרת

קיבוץ ומיון	
Group by	מקבץ לפי עמודה
Having	סינון לאחר קיבוץ
Order by	מיון על פי סדר עולה/ יורד
Limit n	הגבלת תוצאות
Distinct	מסיר כפילויות

```
SELECT
    [ALL | DISTINCT | DISTINCTROW ]
    select_expr [, select_expr] ...
    [FROM table_references
    [WHERE where_condition]
    [GROUP BY {col_name | expr | position}
    [HAVING where_condition]
        [ORDER BY {col_name | expr | position}
      [ASC | DESC], ..]
    [LIMIT {[offset,] row_count | row_count OFFSET offset}]
```

```
SELECT Column 1, Column 2,
CASE
        WHEN condition 1 THEN result1
        WHEN condition 2 THEN result2
        ELSE Else_result
END AS [Column Name for the
                Cases' results]
FROM Table_name;
```

דוגמא ליצירת טבלה ו-DATABASE

```
DROP DATABASE IF EXISTS DatabaseName;
```

```
USE DatabaseName;
```

```
CREATE TABLE TableOne (
    id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100) NOT NULL,
    email VARCHAR(100) UNIQUE,
    phone VARCHAR(20)
);
```

```
INSERT INTO Customers (name, email)
VALUES ('value for name', 'value for email');
```

הכנסת נתונים לתוך טבלה שיצרנו

```
SELECT DISTINCT column_name FROM table;
```

דוגמא ל- distinct

```
SELECT name AS full_name
```

דוגמא לאליאס

```
COUNT (distinct A)
```

ספירת ערכים ייחודיים

דוגמא ל- Subquery

```
SELECT t.name, t.total
FROM (
    SELECT name, SUM(amount) AS total
    FROM Transactions
    GROUP BY name
) AS t
WHERE t.total > 1000;
```

סוגי נתונים	
INT	מספר שלם
NUMERIC(p,s)	מספר עשרוני
VARCHAR(n)	מחרוזת באורך משתנה
DATE	תאריך בלבד
DATETIME	תאריך+שעה
BOOLEAN	FALSE/TRUE

Data Integrity

מה זה Data Integrity?
שמירה על כך שהנתונים במסד הנתונים:
נכונים – ערכים הגיוניים (למשל, לא קיים מוצר עם מחיר שלילי)
שלמים- אין רשומות חסרות או שבורות
עקביים – אין סתירות בין טבלאות (למשל, אין הזמנה ללקוח שלא קיים)
מותאמים לחוקים העסקיים- למשל : אסור להזמין מוצר שאין במלאי

שמירה על Data Integrity
1.Primary Key מזהה ייחודי לכל שורה בטבלה
2. Foreign Key יוצר קשר בין טבלאות, מוודא שאי אפשר להזין ערך שלא קיים בטבלה הראשית
3. Not Null מבטיח ששדה מסוים חייב להכיל ערך
4. Unique מונע כפילויות בשדה מסוים
5. Check מאפשר להגדיר תנאים חוקיים לשדה
6. Default מגדיר ערך ברירת מחדל לשדה שלא מולא

Normalization

דוגמאות לבעיות ב - Normalization
שדות חוזרים
ערכים שמופיעים כמה פעמים
טקסט חופשי – שדות שניתן להקליד בהן כל מחרוזת ללא הגבלה

Normalization	
אין רשומות/ערכים מרובים בתא אחד	1NF
כל עמודה תלויה בכל המפתח הראשי	2NF
איין תלות בין עמודות שאינן מפתחות	3NF

Normalization:
תהליך שמטרתו
למנוע כפילויות
לשמור על עקביות של נתונים
לארגן את הנתונים בצורה ברורה
נעשה זאת ע״ פירוק טבלאות גדולות עם מידע חוזר לטבלאות קטנות יותר עם קשרים ביניהן

יחסים בין טבלאות

טיפים:
כשאר רואים **טבלה עם כמה שדות שמצביעים לטבלאות אחרות** – בדיקת קישור N:N
כשאר רואים **שדה שמופיע הרבה עם אותו ערך** -N:1

יחסים בין טבלאות	
אחד לאחד- כל רשומה בטבלה א מקושרת לרשומה אחת בלבד בטבלה ב	1:1
אחד לרבים – רשומה אחת בטבלה א יכולה להיקשר לרבות בטבלה ב, אך כל רשומה בטבלה ב שייכת רק לאחת בטבלה א	N:1
רבים לרבים – רשומות רבות מטבלה א מקושרות לרשומות רבות בטבלה ב	N:N

למה צריך יחסים בין טבלאות?
נרמול (למנוע כפילויות)
קשרים הגיוניים בין ישויות
יכולת לבצע שאילתות חכמות שמחברות מידע ממקורות שונים

ייצוג ערכים – **כיצד לייצג מידע חוזר או קבוע** – כמו מדינות,מטבעות,קטגוריות,סטטוסים, מגדר,סוגים וכו׳

ייצוג לא נכון – טקסט חופשי
טעויות כתיב
קושי לסנן , לקבץ או לנתח
אין שליטה בערכים

ייצוג נכון – רשימת ערכים קבועים
ערכים מוגדרים מראש
שינוי שם של מדינה מתבצע ממקום אחד בלבד
קוד נקי וברור

תכנון סכמת מסד נתונים

1. האם יש כפילויות? → נרמל
2. האם יש ערכים שחוזרים על עצמם בטקסט? → הפוך אותם לטבלה נפרדת עם FK
3. האם אתה צריך לדעת אם שדה חסר או תקין? → `NOT NULL` או `DEFAULT`
4. האם יש יחסים בין טבלאות? → נסח אותם בבירור ואל תשכח `FOREIGN KEY`

אלו סוגי נתונים מתאימים?

שדה	טיפוס
מזהים	`BIGINT` + `AUTO_INCREMENT` או `INT`
טקסט קצר	`VARCHAR(n)`
טקסט ארוך	`TEXT`
כסף	`DECIMAL(10,2)`
תאריך	`DATETIME` או `DATE`
לוגי	`BOOLEAN`