תשובות למבחן- שרה לוין

**SQL**

מצורף קוד בקובץ נפרד,

**הסבר**:

* הפרוצדורה מקבלת שנה וחודש עבורו אנו רוצים להפיק את הדוח.

@CurrentMonthStart ו-@NextMonthStart מסמנים את תחילת וסוף החודש הנוכחי.

@PrevMonthStart ו-@PrevMonthEnd מסמנים את טווח החודש הקודם.

@LastYearMonthStart ו-@LastYearNextMonthStart מסמנים את אותו חודש בשנה שעברה.

כל טווח משמש לסינון פניות לפי תאריך.

* השאילתה יוצרת שלוש טבלאות זמניות:

#CurrentMonth: סיכום פניות לפי מחלקה עבור החודש הנוכחי.

#PrevMonth: סיכום פניות לפי מחלקה עבור החודש הקודם.

#LastYear: סיכום פניות לפי מחלקה עבור אותו חודש בשנה שעברה.

השאילתה סופרת פניות לכל מחלקה בטווח התאריכים המבוקש.

* לבסוף המחלקות מוצגות עם סך הפניות לכל תקופה.

משתמשים ב-LEFT JOIN כדי להציג את כל המחלקות גם אם אין להן פניות.

מחשבים אחוזי שינוי מול החודש הקודם והשנה שעברה.

אם אין נתונים לחודש הקודם או לשנה שעברה (0 פניות), מחזירים NULL באחוז השינוי.

**דרכים לשפר ביצועים**

**1. . אינדקסים על עמודות התאריך והמחלקה**

הוספת אינדקס על טבלת  Inquiries לפחות על העמודות (InquiryDate, DepartmentId).

זה מאפשר לסנן בצורה יעילה על טווחי תאריכים ולבצע GROUP BY מהיר יותר.

**2**. ניתן להקטין שימוש ביצירת **טבלאות זמניות**, אפשר לנסות לשלוף את הנתונים בפס אחד, למשל באמצעות LEFT JOIN עם תנאי טווח תאריכים בתת-שאילתות או SUM(CASE WHEN ...). זה יכול להקטין את כמות הכתיבה והקריאה לדיסק.

**3. שימוש בטבלאות מצטברות (Aggregated Tables) או חומרת דוח (materialized views)**

אם הנתונים גדולים והדוח מבוצע לעיתים קרובות על אותם חודשים:

אפשר ליצור טבלה מצטברת (כמו MonthlyInquirySummary) שמכילה את סך הפניות בכל חודש ובכל מחלקה.

לעדכן את הטבלה הזו באופן אוטומטי בהוספה או שינוי של הפניות.

הפקודה תוכל לשלוף ישירות מהטבלה המצטברת במקום לחשב כל פעם מחדש.

**4**. **שיפור חישוב אחוזי שינוי**

אם יש סיכוי לפניות רבות מאוד, עדיף לחשב אחוזי שינוי מחוץ ל-SQL (ביישום) כדי לחסוך עומס חישובי.