

## דו"ח הפרויקט - שלב ג'

### ניתוח ביצועים של רופאים

יצרנו מערכת שמנתחת את ביצועי הרופאים במחלקות השונות. המערכת מחשבת אחוזונים של ביצועים, מעלה את השכר לרופאים מצטיינים, מסווגת את הרופאים בהתאם לתוצאותיהם, ומדווחת בצורה ברורה ומסודרת.

### מטרות העבודה:

1. חישוב ביצועי רופאים: חישוב ממוצע משך הזמן שרופא מטפל בחולה בהתבסס על תאריכי האשפוז והשחרור של החולים ביחס לכמות החולים שטיפל.
2. חלוקה לאחוזונים: סיווג הרופאים לאחוזוני ביצועים (Top 10%, Top 20%, Top 50%, ועוד), תוך מתן העלאות שכר למצטיינים ומתן חיווי למי שדורש שיפור.
3. מודולריות: בניית המערכת באופן מודולרי כדי לאפשר תחזוקה והרחבה קלים יותר.
4. טיפול בחריגות: התמודדות עם חריגות באופן ממוקד בכל פונקציה או פרוצדורה כדי להבטיח תהליך חלק וללא שגיאות בלתי צפויות.

### מהלך העבודה:

במהלך העבודה פיתחנו מספר פונקציות ופרוצדורות, שכל אחת מהן מבצעת חלק מוגדר מהמשימה:

- פונקציה לשליפת מחלקות (**get\_departments**): פונקציה זו אחראית לשליפת רשימת המחלקות מבסיס הנתונים באמצעות קורסור.

```
CREATE OR REPLACE FUNCTION get_departments
RETURN SYS_REFCURSOR
IS
    department_cursor SYS_REFCURSOR;
BEGIN
    -- Open a cursor to fetch department list
    OPEN department_cursor FOR
        SELECT DepartmentID, DepartmentName
        FROM Departments;
    RETURN department_cursor;

EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Error: No departments found.');
```

```
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error: Unable to fetch departments - ' || SQLERRM);
END;
```

- פרוצדורה לעיבוד מחלקה (**process\_department**): פרוצדורה זו מקבלת את מזהה המחלקה ומעבדת אותה, כולל חישוב אחוזונים והדפסת נתוני הרופאים.

```
CREATE OR REPLACE PROCEDURE process_department(department_id IN NUMBER, department_name IN VARCHAR2)
IS
    top_10_threshold NUMBER;
    top_20_threshold NUMBER;
    top_50_threshold NUMBER;
    low_20_threshold NUMBER;
    percentiles_cursor SYS_REFCURSOR;
BEGIN
    -- Print department name and ID
    DBMS_OUTPUT.PUT_LINE(CHR(10) || CHR(10) || 'Department: ' || department_name || ' (ID: ' || department_id || ')');

    -- Fetch percentiles for the department
    percentiles_cursor := get_department_percentiles(department_id);
    FETCH percentiles_cursor INTO top_10_threshold, top_20_threshold, top_50_threshold, low_20_threshold;

    -- Print calculated percentiles
    DBMS_OUTPUT.PUT_LINE('Percentiles calculated: Top 10%: ' || top_10_threshold ||
        ', Top 20%: ' || top_20_threshold ||
        ', Top 50%: ' || top_50_threshold ||
        ', Bottom 20%: ' || low_20_threshold);

    -- Print doctor performance for the department
    print_doctor_performance(department_id);

EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Error: No data found for department ' || department_name);
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error processing department ' || department_name || ' - ' || SQLERRM);
END;
```

- פונקציה לחישוב אחוזונים (**get\_department\_percentiles**): פונקציה זו מחשבת את האחוזונים השונים על בסיס ביצועי הרופאים בכל מחלקה.

```
CREATE OR REPLACE FUNCTION get_department_percentiles(department_id IN NUMBER)
RETURN SYS_REFCURSOR
IS
    percentiles_cursor SYS_REFCURSOR;
BEGIN
    -- Open cursor for percentile calculations
    OPEN percentiles_cursor FOR
    SELECT ROUND(PERCENTILE_CONT(0.9) WITHIN GROUP (ORDER BY performance_score), 2) AS top_10,
        ROUND(PERCENTILE_CONT(0.8) WITHIN GROUP (ORDER BY performance_score), 2) AS top_20,
        ROUND(PERCENTILE_CONT(0.5) WITHIN GROUP (ORDER BY performance_score), 2) AS top_50,
        ROUND(PERCENTILE_CONT(0.2) WITHIN GROUP (ORDER BY performance_score), 2) AS low_20
    FROM (
        SELECT d.DoctorID,
            AVG(NVL(p.ReleaseDate, SYSDATE) - p.AdmissionDate) / COUNT(p.PatientID) AS performance_score
        FROM Doctors d
        LEFT JOIN PatientDoctor pd ON d.DoctorID = pd.DoctorID
        LEFT JOIN Patients p ON pd.PatientID = p.PatientID
        WHERE d.DepartmentID = department_id
        GROUP BY d.DoctorID
    );

    RETURN percentiles_cursor;

EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Error: No performance data found for department ' || department_id);
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error calculating percentiles for department ' || department_id || ' - ' || SQLERRM);
    RETURN NULL;
END;
```

- פרוצדורה להדפסת ביצועי רופאים (**print\_doctor\_performance**):  
פרוצדורה זו מציגה את הביצועים של הרופאים לפי הקטגוריות שהוגדרו (מצטיינים, בסדר, ודורשים שיפור).

```
-- Procedure to print doctor performance for a given department
CREATE OR REPLACE PROCEDURE print_doctor_performance(department_id IN NUMBER)
IS
    performance_cursor SYS_REFCURSOR;
    percentiles_cursor SYS_REFCURSOR;
    doctor_id NUMBER;
    doctor_name VARCHAR2(100);
    performance_score NUMBER;
    top_10_threshold NUMBER;
    top_20_threshold NUMBER;
    top_50_threshold NUMBER;
    low_20_threshold NUMBER;
    top_10_doctors VARCHAR2(2000) := '';
    top_20_doctors VARCHAR2(2000) := '';
    top_50_doctors VARCHAR2(2000) := '';
    low_20_doctors VARCHAR2(2000) := '';
    improvement_doctors VARCHAR2(2000) := '';
BEGIN
    -- Fetch percentiles from the external function
    percentiles_cursor := get_department_percentiles(department_id);
    FETCH percentiles_cursor INTO top_10_threshold, top_20_threshold, top_50_threshold, low_20_threshold;

    -- Fetch doctor performance sorted by score
    OPEN performance_cursor FOR
        SELECT d.DoctorID, d.FirstName || ' ' || d.LastName AS DoctorName,
            AVG(NVL(p.ReleaseDate, SYSDATE) - p.AdmissionDate) / COUNT(p.PatientID) AS performance_score
        FROM Doctors d
        LEFT JOIN PatientDoctor pd ON d.DoctorID = pd.DoctorID
        LEFT JOIN Patients p ON pd.PatientID = p.PatientID
        WHERE d.DepartmentID = department_id
        GROUP BY d.DoctorID, d.FirstName, d.LastName
        ORDER BY performance_score DESC;

    -- Loop through doctors and categorize based on performance
    LOOP
        FETCH performance_cursor INTO doctor_id, doctor_name, performance_score;
        EXIT WHEN performance_cursor%NOTFOUND;

        -- Round performance score
        performance_score := ROUND(performance_score, 2);

        -- Add doctor to the appropriate category
        IF performance_score >= top_10_threshold THEN
            top_10_doctors := top_10_doctors || CHR(10) || ' Doctor ' || doctor_name || ' (ID: ' || doctor_id || ') - Performance Score: ' || performance_score;
        ELSIF performance_score >= top_20_threshold THEN
            top_20_doctors := top_20_doctors || CHR(10) || ' Doctor ' || doctor_name || ' (ID: ' || doctor_id || ') - Performance Score: ' || performance_score;
        ELSIF performance_score >= top_50_threshold THEN
            top_50_doctors := top_50_doctors || CHR(10) || ' Doctor ' || doctor_name || ' (ID: ' || doctor_id || ') - Performance Score: ' || performance_score;
        ELSIF performance_score >= low_20_threshold THEN
            low_20_doctors := low_20_doctors || CHR(10) || ' Doctor ' || doctor_name || ' (ID: ' || doctor_id || ') - Performance Score: ' || performance_score;
        ELSE
            improvement_doctors := improvement_doctors || CHR(10) || ' Doctor ' || doctor_name || ' (ID: ' || doctor_id || ') - Performance Score: ' || performance_score;
        END IF;
    END LOOP;

    -- Print results for each category
    IF top_10_doctors IS NOT NULL THEN
        DBMS_OUTPUT.PUT_LINE(CHR(10) || 'Top 10% doctors with a salary raise of 1000: ' || top_10_doctors);
    END IF;

    IF top_20_doctors IS NOT NULL THEN
        DBMS_OUTPUT.PUT_LINE(CHR(10) || 'Top 20% doctors with a salary raise of 500: ' || top_20_doctors);
    END IF;

    IF top_50_doctors IS NOT NULL THEN
        DBMS_OUTPUT.PUT_LINE(CHR(10) || 'Top 50% doctors with a rating of "Very Good": ' || top_50_doctors);
    END IF;

    IF low_20_doctors IS NOT NULL THEN
        DBMS_OUTPUT.PUT_LINE(CHR(10) || 'Doctors with a rating of "Almost Good": ' || low_20_doctors);
    END IF;

    IF improvement_doctors IS NOT NULL THEN
        DBMS_OUTPUT.PUT_LINE(CHR(10) || 'Doctors needing improvement (Bottom 20%): ' || improvement_doctors);
    END IF;

    -- Close the performance cursor
    CLOSE performance_cursor;
END;
```

```

EXCEPTION
WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE('Error: No doctor performance data found for department ' || department_id);
WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Error printing doctor performance for department ' || department_id || ' - ' || SQLERRM);
    CLOSE performance_cursor;

```

- פרוצדורה לטיפול כללי בשגיאות (**handle\_general\_error**): פרוצדורה זו מטפלת בשגיאות כלליות שלא נלכדו בפרוצדורות האחרות.

```

-- Procedure to handle errors and exceptions
CREATE OR REPLACE PROCEDURE handle_general_error(error_code IN NUMBER, error_message IN VARCHAR2)
IS
BEGIN
    -- Print the error code and message
    DBMS_OUTPUT.PUT_LINE('Error Code: ' || error_code);
    DBMS_OUTPUT.PUT_LINE('Error Message: ' || error_message);
    -- Rollback the transaction if needed
    ROLLBACK;
END;

```

## דוגמת הרצה:

Department: Pediatrics (ID: 1)  
 Percentiles calculated: Top 10%: 162.49, Top 20%: 132.45, Top 50%: 70.28, Bottom 20%: 27

Top 10% doctors with a salary raise of 1000:

Doctor Linda Williams (ID: 309043170) - Performance Score: 276.6  
 Doctor Alice Smith (ID: 304078104) - Performance Score: 236.55  
 Doctor James Jones (ID: 207136636) - Performance Score: 169.99

Top 20% doctors with a salary raise of 500:

Doctor Linda Taylor (ID: 206367479) - Performance Score: 160.62  
 Doctor David Smith (ID: 207448768) - Performance Score: 156.24  
 Doctor James Jones (ID: 301135367) - Performance Score: 150.38

Top 50% doctors with a rating of "Very Good":

Doctor Alice Williams (ID: 200096911) - Performance Score: 120.5  
 Doctor Sara Williams (ID: 208771808) - Performance Score: 111.09  
 Doctor Alice Wilson (ID: 304942290) - Performance Score: 107.66  
 Doctor Jane Taylor (ID: 304539706) - Performance Score: 104.76  
 Doctor John Jones (ID: 200498237) - Performance Score: 97.95  
 Doctor James Moore (ID: 305317262) - Performance Score: 91.13  
 Doctor John Moore (ID: 306780302) - Performance Score: 89.56  
 Doctor David Jones (ID: 307401220) - Performance Score: 87.97  
 Doctor James Brown (ID: 208132669) - Performance Score: 70.28

Doctors with a rating of "Almost Good":

Doctor Robert Davis (ID: 207328765) - Performance Score: 66.29  
 Doctor Jane Taylor (ID: 203022445) - Performance Score: 56.65  
 Doctor Sara Williams (ID: 202537975) - Performance Score: 47.7  
 Doctor Jane Smith (ID: 202876381) - Performance Score: 39.06  
 Doctor Michael Johnson (ID: 300417897) - Performance Score: 37.85

## דוגמה לשינוי בבסיס הנתונים לרופאים שקיבלו העלאה עבור מחלקה 1:

### לפני הריצה

	DOCTORID	FIRSTNAME	LASTNAME	SALARY
1	200498237	John	Jones	138064
2	304942290	Alice	Wilson	80317
3	204350645	Alice	Taylor	94645
4	206367479	Linda	Taylor	170623
5	209929760	Michael	Smith	140521
6	207136636	James	Jones	56611
7	307401220	David	Jones	67109
8	208771808	Sara	Williams	177472
9	204901193	Linda	Moore	75839
10	300417897	Michael	Johnson	175967
11	208132669	James	Brown	160632
12	209531628	Linda	Moore	57191
13	309043170	Linda	Williams	73364
14	309815512	Michael	Moore	84983
15	207328765	Robert	Davis	133693
16	306780302	John	Moore	182203
17	305317262	James	Moore	167570
18	309925512	David	Wilson	59893
19	203022445	Jane	Taylor	90835
20	200096911	Alice	Williams	58587
21	304539706	Jane	Taylor	155402
22	202876381	Jane	Smith	180792
23	301135367	James	Jones	183456
24	304078104	Alice	Smith	199447
25	202537975	Sara	Williams	156153
26	307844344	Jane	Wilson	141892
27	302232987	Linda	Williams	111419
28	207448768	David	Smith	175872
29	302291201	Sara	Johnson	191253

### לאחר הריצה

	DOCTORID	FIRSTNAME	LASTNAME	SALARY
▶ 1	200498237	John	Jones	138064
2	304942290	Alice	Wilson	80317
3	204350645	Alice	Taylor	94645
4	206367479	Linda	Taylor	171123
5	209929760	Michael	Smith	140521
6	207136636	James	Jones	57611
7	307401220	David	Jones	67109
8	208771808	Sara	Williams	177472
9	204901193	Linda	Moore	75839
10	300417897	Michael	Johnson	175967
11	208132669	James	Brown	160632
12	209531628	Linda	Moore	57191
13	309043170	Linda	Williams	74364
14	309815512	Michael	Moore	84983
15	207328765	Robert	Davis	133693
16	306780302	John	Moore	182203
17	305317262	James	Moore	167570
18	309925512	David	Wilson	59893
19	203022445	Jane	Taylor	90835
20	200096911	Alice	Williams	58587
21	304539706	Jane	Taylor	155402
22	202876381	Jane	Smith	180792
23	301135367	James	Jones	183956
24	304078104	Alice	Smith	200447
25	202537975	Sara	Williams	156153
26	307844344	Jane	Wilson	141892
27	302232987	Linda	Williams	111419
28	207448768	David	Smith	176372
29	302291201	Sara	Johnson	191253