

# תרגיל 3 בסיסי נתונים

מגישות:

איילה חורי 213867963

ורטהימר לאה 325311355

הערה: הוספנו שדה injection\_age ל injection המתאר את הגיל בו אמורים לקבל חיסון זה.

כל קודי הפונקציות והפרוצדורות מצורפות בסוף הקובץ.

## תוכנית א:

תוכנית המקבלת תעודת זהות של ילד, מספר חיסון, תעודת זהות של אחות, תאריך, מספר חדר וקוראת לפונקציה המחשבת את גילו ולאחר מכן אם הילד קטן בגיל שבו נותנים את החיסון, קוראת לפרוצורה שמוסיפה את הפגישה לילד.

```
BEGIN
  DECLARE
    v_count NUMBER;
    v_child_age NUMBER;
    v_age_injection NUMBER;
    v_child_id NUMBER := &v_child_id;
    v_nurse_id NUMBER := &v_nurse_id;
    v_injection_id NUMBER := &v_injection_id;
    v_app_id NUMBER;
    v_hour VARCHAR2(10) := '&v_hour';
    v_app_date DATE := TO_DATE(&date, 'DD-MM-YYYY');
    v_room NUMBER := &v_room;
    message VARCHAR2(100);
    child_id_dont_exists EXCEPTION;
    PRAGMA EXCEPTION_INIT(child_id_dont_exists, -20006);
  BEGIN

    -- Check if the child exists
    SELECT COUNT(*)
    INTO v_count
    FROM child
    WHERE child_id = v_child_id;

    IF v_count = 0 THEN
      message := 'Child id ' || v_child_id || ' dont exists ' ;
      RAISE child_id_dont_exists;
    END IF;

    -- Calculate the child age
    v_child_age := calculate_child_age(v_child_id);
    DBMS_OUTPUT.PUT_LINE('Child age: ' || v_child_age);
```

```

SELECT age_injection
INTO v_age_injection
FROM injection
WHERE injection_id = v_injection_id;

-- Adding an appointment only if the child is younger than 18
IF v_child_age < 18 THEN
    -- Generate a new appointment ID
    SELECT NVL(MAX(appointment_id), 0) + 1 INTO v_app_id FROM
appointment;

    -- Add the appointment
    add_appointment(v_app_id, v_app_date, v_hour, v_room,
v_child_id, v_nurse_id, v_injection_id);
ELSE
    DBMS_OUTPUT.PUT_LINE('Child is too old for vaccination.');
```

```

END IF;
EXCEPTION
    WHEN child_id_dont_exists THEN
        RAISE_APPLICATION_ERROR(-20006, message);
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
END;
END;
```

ניתן לראות שהילד עם תעודת הזהות 108 בן שנה

SQL Output Statistics				
<pre>select * from child</pre>				
	CHILD_ID	NAME	BIRTH_DATE	
1	100	Tay	12/11/2017	
2	101	Lou	30/01/2019	
3	102	Mar	03/06/2017	
4	103	Tin	05/07/2020	
5	104	Jef	24/11/2013	
6	105	Ray	13/12/2019	
7	106	Emi	28/01/2020	
8	107	Chr	07/05/2017	
9	108	Isa	27/04/2023	
10	109	Jas	07/12/2022	
11	110	Joh	11/12/2021	
12	111	Mac	15/07/2020	
13	112	Ash	17/01/2015	
14	113	Gab	28/08/2020	
15	114	Cat	05/05/2016	

ניתן לראות שהחיסון עם מספר 72 ניתן בגיל שנה

SQL Window - select * from injection a where a.age_injection = 1					
SQL Output Statistics					
<pre>select * from injection a where a.age_injection = 1</pre>					
	INJECTION_ID	NAME	DISEASE	DOSAGE	AGE_INJECTION
1	72	Van	ALC	3.25	1
2	75	Ber	Ber	5.59	1
3	84	HCG	Hum	2.59	1
4	107	DOX	dox	6.34	1
5	149	Gli	Gli	1.91	1
6	198	Rop	Rop	1.89	1
7	203	Oxy	Oxy	5.74	1
8	265	RUE	ALC	3.56	1
9	286	GAB	tia	1.59	1
10	301	Sag	Sag	1.28	1
11	310	RIC	OCT	0.59	1
12	338	Lad	Alu	3.09	1
13	360	TON	CAM	4.3	1
14	379	Sal	Sal	4.9	1

הרצנו את התוכנית לקביעת החיסון 72 לילד 108

```

DECLARE
    v_child_age NUMBER;
    v_age_injection NUMBER;
    v_child_id NUMBER := &v_child_id;
    v_nurse_id NUMBER := &v_nurse_id;
    v_app_date NUMBER := &v_app_injection_id;
    v_app_id NUMBER;
    v_hour VARCHAR2(10) := '&v_hour';
    v_app_date DATE := TO_DATE(&date, 'DD-MM-YYYY');
    v_room NUMBER := &v_room;

BEGIN
    -- Calculate the child age
    v_child_age := calculate_child_age(v_child_id);
    DBMS_OUTPUT.PUT_LINE('Child age: ' || v_child_age);

    SELECT age_injection
    INTO v_age_injection
    FROM injection
    WHERE injection_id = v_injection_id;

    -- Adding an appointment only if the child is young
    IF v_child_age < 18 THEN
        -- Generate a new appointment ID
        SELECT NVL(MAX(appointment_id), 0) + 1 INTO v_app_id
        FROM appointment;

        -- Add the appointment
        add_appointment(v_app_id, v_app_date, v_hour, v_nurse_id);

    ELSE
        DBMS_OUTPUT.PUT_LINE('Child is too old for vaccination');
    END IF;

    EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);

END;

```

The 'Variables' dialog box in RStudio is shown. It contains a table with the following data:

Name	Value
v_child_id	108
v_nurse_id	104
v_injection_id	72
v_hour	10:35
date	'06/07/2023'
v_room	109

At the bottom of the dialog, there are three buttons: 'OK', 'Cancel', and 'Clear'. The 'OK' button is highlighted.

## קִיבְּלָנוּ שְׂגִיָּאָה שֶׁלֹּא קִיִּימַת הָאָחוֹת הַנְּתוּנָה

```
Child age: 1
Error: ORA-20002: Nurse id 104 not found.
```

## שינינו את האחות

```

BEGIN
DECLARE
    v_child_age NUMBER;
    v_age_injection NUMBER;
    v_child_id NUMBER := &v_child_id;
    v_nurse_id NUMBER := &v_nurse_id;
    v_injection_id NUMBER := &v_injection_id;
    v_app_id NUMBER;
    v_hour VARCHAR2(10) := '&v_hour';
    v_app_date DATE := TO_DATE(&date, 'DD-MM-YYYY');
    v_room_number := &v_room;

BEGIN
    -- Calculate the child age
    v_child_age := calculate_child_age(v_child_id);
    DBMS_OUTPUT.PUT_LINE('Child age: ' || v_child_age);

    SELECT age_injection
    INTO v_age_injection
    FROM injection
    WHERE injection_id = v_injection_id;

    -- Adding an appointment only if the child is young
    IF v_child_age < 15 THEN
        -- Generate a new appointment ID
        SELECT NVL(MAX(appointment_id), 0) + 1 INTO v_app_id
        FROM appointment;

        -- Add the appointment
        add_appointment(v_app_id, v_app_date, v_hour, v_nurse_id);

        DBMS_OUTPUT.PUT_LINE('Child is too old for vaccination');
    ELSE
        DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
    END IF;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
END;
END;
```

**Variables**

Name	Value
v_child_id	108
v_nurse_id	102
v_injection_id	72
v_hour	10.35
date	06/07/2023
v_room	109

OK Cancel Clear

קיבלנו

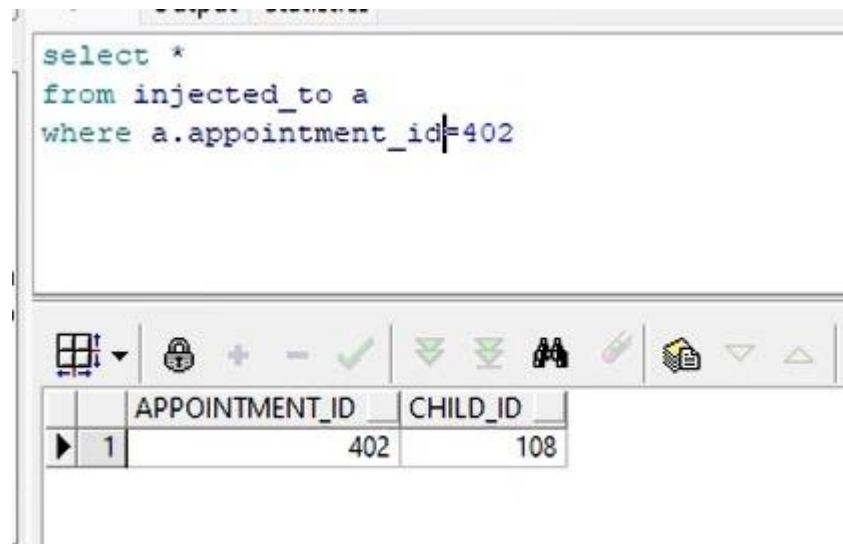
```
Child age: 1
Appointment added.
```

**נראה שמסד הנתונים השתנה בטבלאות הרצויות**

401 | 402 06/07/2023 10:35 109

הפגישה:

## הקשר בין פגישה לילד:



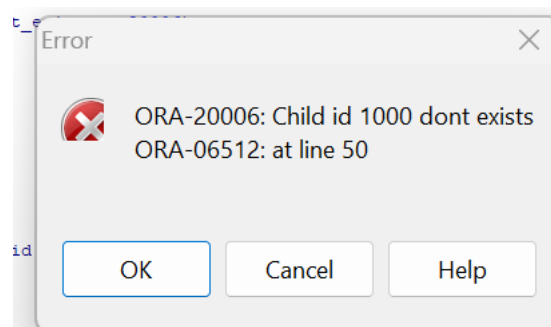
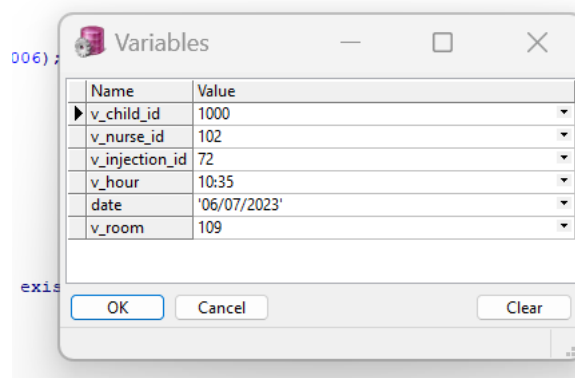
הקשר בין חיסון לפגישה:

	APPOINTMENT_ID	INJECTION_ID
1	402	72

הקשר בין אחות לפגישה:

	APPOINTMENT_ID	INJECTION_ID
1	402	72

אם הילד לא קיים ישנה שגיאה (כמו כן לגבי מספר חדר, אחות, חיסון וכן אם האחות או החדר תפוסים בשעה בה רוצים לקבוע פגישה)



## תוכנית ב:

התוכנית קולטת תז של ילד ומראה את תוכנית החיסונים שלו, כלומר חיסונים שחוסן עד עכשיו וחיסונים שעליו לקבל בעתיד ובאיזה גיל

```
DECLARE
  v_child_id NUMBER := &child_id;
BEGIN
  show_vaccination_schedule(v_child_id);
END;
```

הערה לבודקים: כאשר מריצים את התוכנית צריך להגיל את גודל הבאפר בחלון output אחרת ישנה שגיאה ולא מקבלים את כל הפלט הרצוי

Variables

Name	Value
child_id	106

OK

Cancel

Clear

Clear

Buffer size 104000

☒ Enabled

injection that the child recived:

injection\_name: Elt, disease: zin, date: 29-11-2024

injections the child need to get:

injection name: Fer, disease: Fer, age required: 4

injection name: Mes, disease: Mes, age required: 4

injection name: Fre, disease: Tri, age required: 4

injection name: Equ, disease: alu, age required: 4

injection name: Cal, disease: Arn, age required: 4

injection name: Aml, disease: Aml, age required: 4

injection name: Met, disease: met, age required: 4

injection name: Nam, disease: mem, age required: 4

injection name: Rit, disease: ZIN, age required: 4

injection name: Pan, disease: Pan, age required: 4

injection name: Alt, disease: Dex, age required: 4

injection name: Eve, disease: Tit, age required: 4

injection name: Mix, disease: Mix, age required: 4

injection name: LBE, disease: OCT, age required: 4

injection name: Hev, disease: ANA, age required: 4

injection name: Dil, disease: Dil, age required: 4

injection name: PAS, disease: Bah, age required: 4

injection name: Hyd, disease: Hyd, age required: 4

injection name: Esc, disease: esc, age required: 4

injection name: Jet, disease: ACO, age required: 4

injection name: Mou, disease: Mou, age required: 4

injection name: Spa, disease: Chl, age required: 4

injection name: Ter, disease: Ter, age required: 4

injection name: Eye, disease: wat, age required: 4

injection name: Bup, disease: Bup, age required: 5

injection name: ZES, disease: Lis, age required: 5

injection name: Lan, disease: lan, age required: 5

injection name: Bet, disease: Bet, age required: 5

injection name: ENA, disease: ENA, age required: 5

injection name: Lis, disease: lis, age required: 5

injection name: car, disease: Ace, age required: 5

injection name: LBE, disease: Oct, age required: 5

injection name: SHO, disease: OXY, age required: 5

injection name: Tem, disease: Tem, age required: 5

injection name: Gon, disease: FOL, age required: 5

injection name: Vel, disease: Vel, age required: 5

injection name: Irb, disease: IRB, age required: 5

injection name: SER, disease: QUE, age required: 5

injection name: Ant, disease: Eth, age required: 5

injection name: Dio, disease: TIT, age required: 5

## פונקציות

```
CREATE OR REPLACE FUNCTION calculate_child_age(p_child_id NUMBER)
RETURN NUMBER IS
    v_birth_date DATE;
    v_age NUMBER;
    CURSOR c_child IS
        SELECT birth_date
        FROM child
        WHERE child_id = p_child_id;
BEGIN
    OPEN c_child;
    FETCH c_child INTO v_birth_date;
    CLOSE c_child;

    IF v_birth_date IS NULL THEN
        RETURN NULL;
    ELSE
        v_age := TRUNC(MONTHS_BETWEEN(SYSDATE, v_birth_date) / 12);
        RETURN v_age;
    END IF;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RETURN NULL;
    WHEN OTHERS THEN
        RETURN NULL;
END calculate_child_age;
```

```
CREATE OR REPLACE FUNCTION get_child_vaccinations(its_child_id
NUMBER)
RETURN SYS_REFCURSOR IS
    v_cursor SYS_REFCURSOR;
BEGIN
    OPEN v_cursor FOR
    SELECT i.name, i.disease, a.app_date
    FROM injection i
    JOIN in_stock ins ON i.injection_id = ins.injection_id
    JOIN injected inj ON ins.stock_id = inj.stock_id
    JOIN appointment a ON inj.appointment_id = a.appointment_id
    JOIN injected_to it ON a.appointment_id = it.appointment_id
    WHERE a.app_date < sysdate AND it.child_id = its_child_id;
    RETURN v_cursor;
END;
```

## פרוצדורות

```
CREATE OR REPLACE PROCEDURE add_appointment(
    p_app_id NUMBER,
    p_app_date DATE,
    p_hour VARCHAR2,
    p_room NUMBER,
    p_child_id NUMBER,
    p_nurse_id NUMBER,
    p_injection_id NUMBER
) IS
    v_count NUMBER;
```



```

v_amount NUMBER;
message VARCHAR2(100);
Room_already_booked EXCEPTION;
PRAGMA EXCEPTION_INIT(Room_already_booked, -20000);
Appointment_id_already_exists EXCEPTION;
PRAGMA EXCEPTION_INIT(Appointment_id_already_exists, -20001);
Nurse_not_found EXCEPTION;
PRAGMA EXCEPTION_INIT(Nurse_not_found, -20002);
Nurse_already_busy EXCEPTION;
PRAGMA EXCEPTION_INIT(Nurse_already_busy, -20003);
Injection_not_found EXCEPTION;
PRAGMA EXCEPTION_INIT(Injection_not_found, -20004);
Injection_out_of_stock EXCEPTION;
PRAGMA EXCEPTION_INIT(Injection_out_of_stock, -20005);
BEGIN
    -- Check if appointment_id already exists
    SELECT COUNT(*)
    INTO v_count
    FROM appointment
    WHERE appointment_id = p_app_id;

    IF v_count != 0 THEN
        message := 'Appointment id ' || p_app_id || ' already
exists.';
        RAISE Appointment_id_already_exists;
    END IF;

    -- Check if the room is already booked for the given date and
hour
    SELECT COUNT(*)
    INTO v_count
    FROM appointment
    WHERE app_date = p_app_date AND hour = p_hour AND room = p_room;

    IF v_count != 0 THEN
        message := 'Room ' || p_room || ' is already booked.';
        RAISE Room_already_booked;
    END IF;

    -- Check if nurse exists
    SELECT COUNT(*)
    INTO v_count
    FROM nurse
    WHERE nurse_id = p_nurse_id;

    IF v_count = 0 THEN
        message := 'Nurse id ' || p_nurse_id || ' not found.';
        RAISE Nurse_not_found;
    END IF;

    -- Check if nurse is already busy at the given date and hour
    SELECT COUNT(*)
    INTO v_count
    FROM injected_by ib
    JOIN appointment a ON ib.appointment_id = a.appointment_id
    WHERE a.app_date = p_app_date AND a.hour = p_hour AND ib.nurse_id
= p_nurse_id;

    IF v_count != 0 THEN
        message := 'Nurse id ' || p_nurse_id || ' is already busy at
this time.';

```

```

        RAISE Nurse_already_busy;
    END IF;

    -- Check if injection_id exists and is in stock
    SELECT COUNT(*)
    INTO v_count
    FROM in_stock
    WHERE injection_id = p_injection_id;

    IF v_count = 0 THEN
        message := 'Injection id ' || p_injection_id || ' not
found.';
        RAISE Injection_not_found;
    END IF;

    SELECT s.amount
    INTO v_amount
    FROM stock s
    JOIN in_stock c ON s.stock_id = c.stock_id
    WHERE c.injection_id = p_injection_id;

    IF v_amount <= 0 THEN
        message := 'Injection id ' || p_injection_id || ' is out of
stock.';
        RAISE Injection_out_of_stock;
    END IF;

    -- Insert into appointment table
    INSERT INTO appointment (appointment_id, app_date, hour, room)
    VALUES (p_app_id, p_app_date, p_hour, p_room);

    -- Insert into injected_to table
    INSERT INTO injected_to (appointment_id, child_id)
    VALUES (p_app_id, p_child_id);

    -- Insert into injected_by table
    INSERT INTO injected_by (appointment_id, nurse_id)
    VALUES (p_app_id, p_nurse_id);

    -- Insert into injected table
    INSERT INTO injected (appointment_id, stock_id)
    VALUES (p_app_id, (SELECT stock_id FROM in_stock WHERE
injection_id = p_injection_id));

    -- Update stock amount
    UPDATE stock
    SET amount = amount - 1
    WHERE stock_id = (SELECT stock_id FROM in_stock WHERE
injection_id = p_injection_id);

    COMMIT;
    DBMS_OUTPUT.PUT_LINE('Appointment added.');
```

```

EXCEPTION
    WHEN Room_already_booked THEN
        RAISE_APPLICATION_ERROR(-20000, message);
    WHEN Appointment_id_already_exists THEN
        RAISE_APPLICATION_ERROR(-20001, message);
    WHEN Nurse_not_found THEN
        RAISE_APPLICATION_ERROR(-20002, message);
    WHEN Nurse_already_busy THEN
```

```

        RAISE_APPLICATION_ERROR(-20003, message);
    WHEN Injection_not_found THEN
        RAISE_APPLICATION_ERROR(-20004, message);
    WHEN Injection_out_of_stock THEN
        RAISE_APPLICATION_ERROR(-20005, message);
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
        ROLLBACK;
END add_appointment;

```

```

CREATE OR REPLACE PROCEDURE show_vaccination_schedule(its_child_id
NUMBER) IS
    its_child_age NUMBER;
    v_vaccinations SYS_REFCURSOR;
    v_name injection.name%TYPE;
    v_disease injection.disease%TYPE;
    v_app_date appointment.app_date%TYPE;

    CURSOR c_vaccinations IS
        SELECT name, disease, age_injection
        FROM injection
        ORDER BY age_injection;

    v_inj_rec c_vaccinations%ROWTYPE;
BEGIN
    its_child_age:= calculate_child_age(its_child_id);

    v_vaccinations := get_child_vaccinations(its_child_id);

    DBMS_OUTPUT.PUT_LINE('injection that the child recived:');
    LOOP
        FETCH v_vaccinations INTO v_name, v_disease, v_app_date;
        EXIT WHEN v_vaccinations%NOTFOUND;
        DBMS_OUTPUT.PUT('injection_name: ' || v_name);
        DBMS_OUTPUT.PUT_LINE(', disease: ' || v_disease || ',
date: ' || v_app_date);
    END LOOP;
    CLOSE v_vaccinations;

    DBMS_OUTPUT.PUT_LINE('');
    DBMS_OUTPUT.PUT_LINE('injections the child need to get:');
    OPEN c_vaccinations;
    LOOP
        FETCH c_vaccinations INTO v_inj_rec;
        EXIT WHEN c_vaccinations%NOTFOUND;
        IF v_inj_rec.age_injection >= its_child_age THEN
            DBMS_OUTPUT.PUT_LINE('injection name: ' || v_inj_rec.name
|| ', disease: ' || v_inj_rec.disease || ', age required: ' ||
v_inj_rec.age_injection);
        END IF;
    END LOOP;
    CLOSE c_vaccinations;
END;

```