פרויקט גמר SQL

נושא:

ניהול בית מלון

שם המרצה: אורית קלינגר

שם המגיש: אוריאל שבת

ת.ז: 328106844

תוכן עניינים

[**מבוא: 3**](#_sov5k8ypjkrv)

[**מטרות המערכת: 3**](#_3qrc7zioaoq7)

[**משתמשים: 4**](#_c2svjrkqndi9)

[**ERD - Entity-relationship model: 4**](#_c4mgupgmn8uj)

[**DSD- Data Structure Diagram: 5**](#_gnn187r6o1yz)

[**אילוצי מערכת: 6**](#_7cbr9o1fjpci)

[**טבלאות: 7**](#_v3q4hz1dzkth)

[**פירוט ומבנה ודוגמה לנתונים: 7**](#_cz5ztgoluuze)

[**רשימת שאילתות: 11**](#_f5cmti70zee3)

# מבוא:

ניהול בית מלון זה עבודה חשובה שמטרתה לספק חווית אירוח טובה לאורחים. מערכת ניהול היא כלי שמסייע לנו לנהל את המלון בצורה יעילה. היא כוללת תתי מערכות שמטרתן לטפל בדברים כמו הזמנות חדרים, ניהול כספים וחשבוניות, ניהול מלאי ועוד. כל אחת מתתי המערכות עוזרת לנו לנהל את המלון בצורה יעילה יותר ולשפר את השירות שאנו מספקים לאורחים.

תיאור המערכת:

מערכת ניהול בית המלון היא ערכת כלים ותהליכים המיועדת לייעל ולתמרץ את פעילות בית המלון. המערכת מכילה תתי מערכות המתמקדות בתחומים שונים כגון הזמנות חדרים, ניהול כספים וחשבונאות, ניהול מלאי, וניהול משאבי אנוש. כל אחת מתתי המערכות משרתת בתחום מסוים ומספקת פתרונות מותאמים לצרכי המלון.

לדוגמה, מערכת ההזמנות מטפלת לקבלת הזמנות מלקוחות לחדרי המלון ולשירותי האוכל והשתייה. מערכת הכספים והחשבונאות עוזרת לנהל את התפעול הכספי של המלון, כולל חשבוניות, תשלומים לספקים, ושכר עובדים. מערכת המלאי מקנה יכולת לנהל את המלאי של חומרי עזר המוצרים הנדרשים לפעילות המלון. באמצעות כלי אלה, המערכת מסייעת בהפעלת המלון בצורה חלקה ובעלת יעדים כלכליים ותפעוליים מוגדרים.

## מטרות המערכת:

1. ניהול אורחים ביעילות: מסד נתונים מאפשר למלון לנהל בדיוק את המידע של האורחים, כולל הזמנות, צ'ק-אין, העדפות ובקשות מיוחדות. זה מסייע בספק שירותים אישיים לאורחים, ומשפר את חוויתם הכוללת ושביעות רצונם.

2. ניהול מלאי ומשאבים: בעזרת מסד נתונים, המלון יכול לעקוב אחר המלאי שלו, כולל חדרים זמינים, ציוד וציוד. זה מבטיח שימוש אופטימלי במשאבים, מונע עליות מספר ההזמנות ועוזר בתכנון תחזוקה והחלפת משאבים באופן יעיל.

3. ניהול פיננסי: מסד נתונים מאפשר למלון לשפר את פעולותיו הפיננסיות על ידי ניהול חשבוניות, הפקת חשבוניות ועיבוד תשלומים בצורה יעילה. זה מספק תובנות בנושא זרמי הכנסות, הוצאות ורווחיות, ומקל על קבלת החלטות פיננסיות ותקציבות.

4. ניתוח ביצועים ושיפור: על ידי אחסון וניתוח נתונים על משוב מאורחים, שיעורי התיישנות, דפוסי הזמנות ושימוש בשירותים, המלון יכול לזהות מגמות, נקודות חוזק ואזורים לשיפור. זה מאפשר קבלת החלטות אסטרטגיות לשיפור פעולות, אסטרטגיות שיווק וביצועים כלליים.

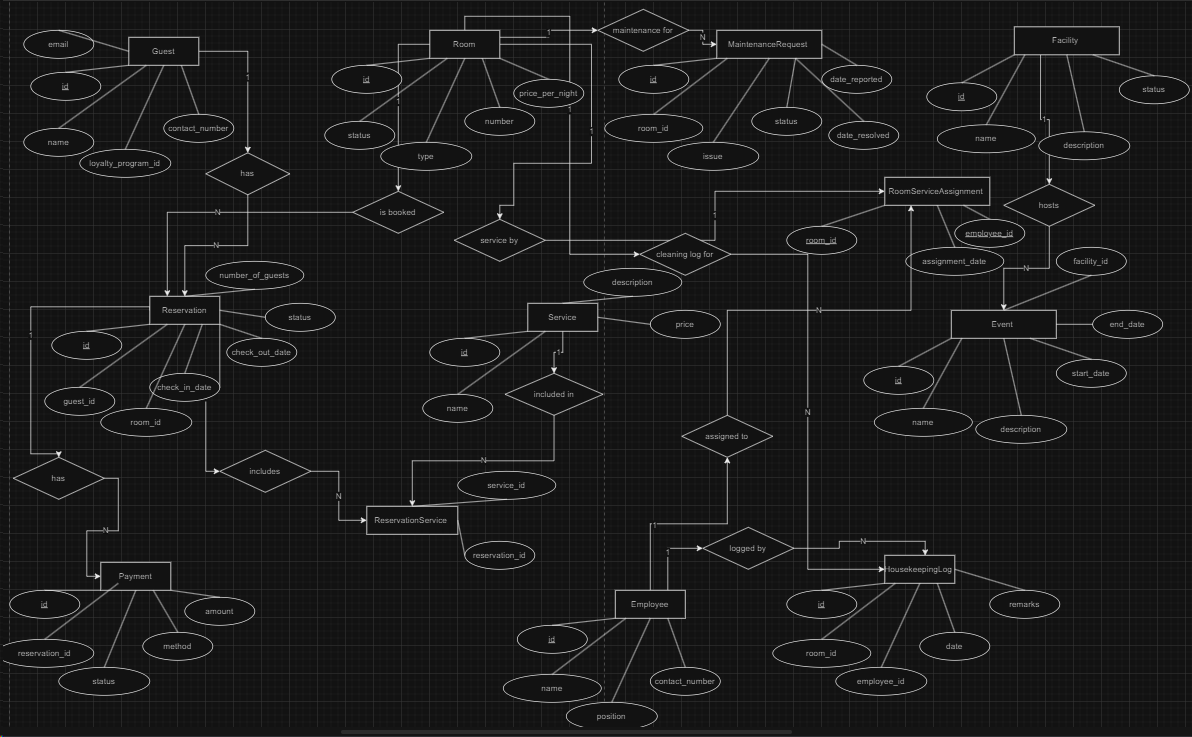
## משתמשים:

קהל היעד:תיירים ומטיילים: אנשים שמגיעים לבילוי, טיולים וחופשה באזור. עסקים וכנסים: קבוצות עסקיות או מטיילים המגיעים לצורך כנסים, פגישות עסקיות, אירועים או פעילויות עסקיות אחרות ואורחים פרטיים: אנשים המחפשים חווית אירוח ומפנקת, כגון זוגות לחתונות, חגיגות יומולדת או פגישות משפחתיות.

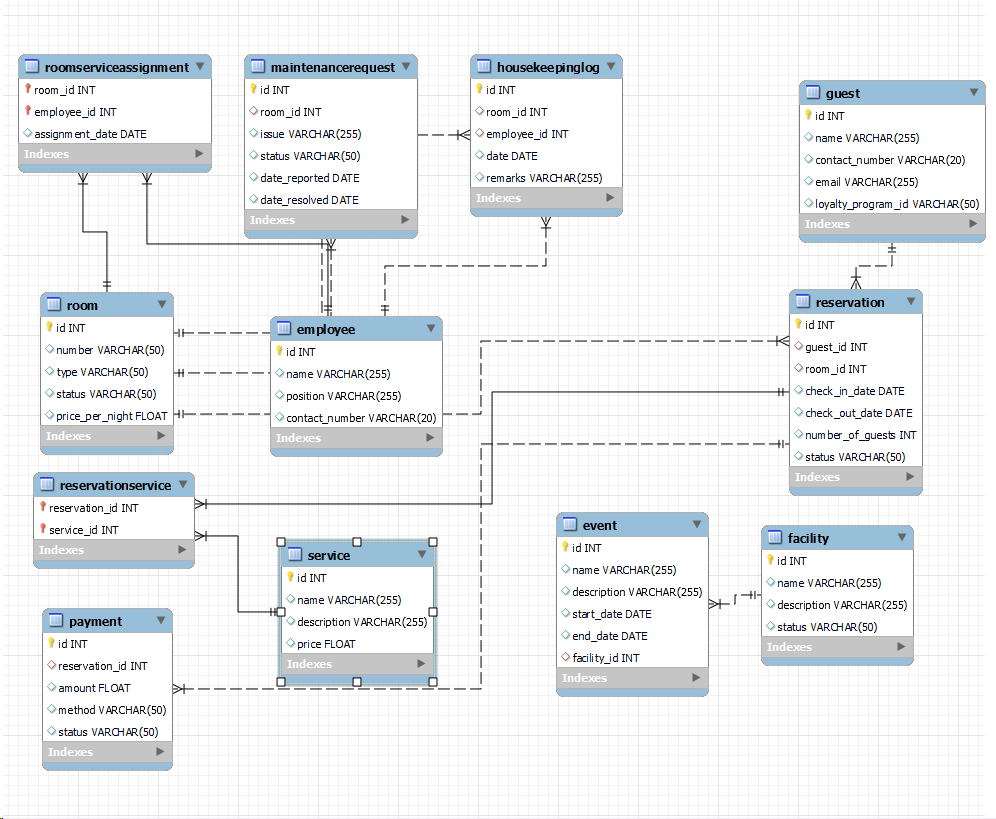
# 

# 

# ERD - Entity-relationship model:



# DSD- Data Structure Diagram:



# אילוצי מערכת:

-- Unique Email for Guests

1. ALTER TABLE Guest ADD CONSTRAINT UC\_Guest\_Email UNIQUE (email);

-- Unique Room Numbers

1. ALTER TABLE Room ADD CONSTRAINT UC\_Room\_Number UNIQUE (number);

-- Valid Reservation Dates

1. ALTER TABLE Reservation ADD CONSTRAINT CHK\_Reservation\_Dates CHECK (check\_out\_date > check\_in\_date);

-- Non-Negative Payment Amounts

1. ALTER TABLE Payment ADD CONSTRAINT CHK\_Payment\_Amount CHECK (amount >= 0);

-- Unique Contact Numbers for Employees

1. ALTER TABLE Employee ADD CONSTRAINT UC\_Employee\_ContactNumber UNIQUE (contact\_number);

-- Referential Integrity for Reservations

1. ALTER TABLE Reservation ADD CONSTRAINT FK\_Reservation\_Guest FOREIGN KEY (guest\_id) REFERENCES Guest(id);

ALTER TABLE Reservation ADD CONSTRAINT FK\_Reservation\_Room FOREIGN KEY (room\_id) REFERENCES Room(id);

-- Referential Integrity for Payments

1. ALTER TABLE Payment ADD CONSTRAINT FK\_Payment\_Reservation FOREIGN KEY (reservation\_id) REFERENCES Reservation(id);

-- Room Service Assignment Constraints

1. ALTER TABLE RoomServiceAssignment ADD CONSTRAINT FK\_RoomServiceAssignment\_Room FOREIGN KEY (room\_id) REFERENCES Room(id);

ALTER TABLE RoomServiceAssignment ADD CONSTRAINT FK\_RoomServiceAssignment\_Employee FOREIGN KEY (employee\_id) REFERENCES Employee(id);

-- Housekeeping Log Constraints

1. ALTER TABLE HousekeepingLog ADD CONSTRAINT FK\_HousekeepingLog\_Room FOREIGN KEY (room\_id) REFERENCES Room(id);

ALTER TABLE HousekeepingLog ADD CONSTRAINT FK\_HousekeepingLog\_Employee FOREIGN KEY (employee\_id) REFERENCES Employee(id);

-- Event Facility Constraints

1. ALTER TABLE Event ADD CONSTRAINT FK\_Event\_Facility FOREIGN KEY (facility\_id) REFERENCES Facility(id);

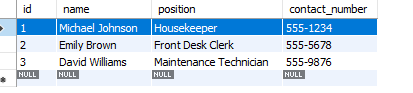
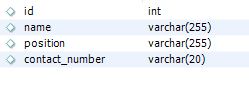
# 

# 

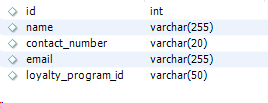
# טבלאות:

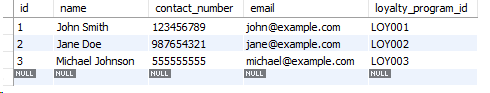
## פירוט ומבנה ודוגמה לנתונים:

Employee



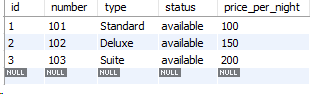
Guest



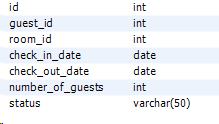


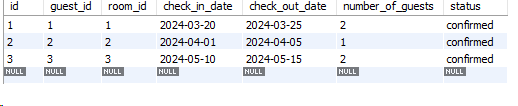
Room



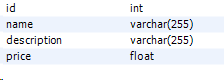


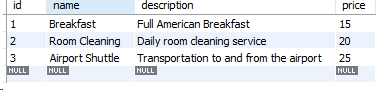
Reservation





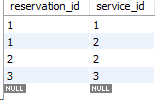
Service



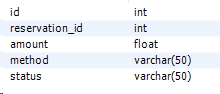


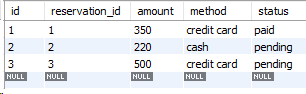
ReservationService





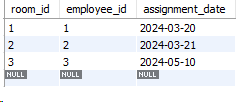
Payment



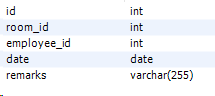


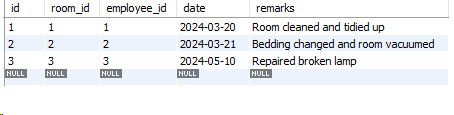
RoomServiceAssignment





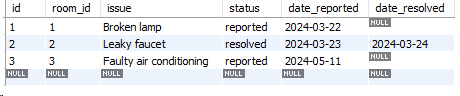
HousekeepingLog



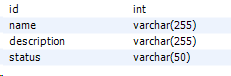


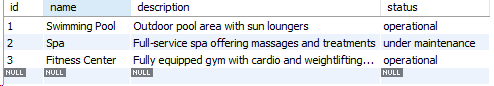
MaintenanceRequest



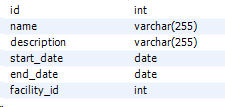


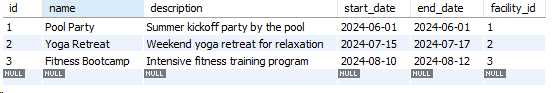
Facility





Event





# 

# 

# רשימת שאילתות:

1. מטרת השאילתה

חשב את משך השהייה הממוצע עבור כל סוג חדר.

SELECT r.type, AVG(DATEDIFF(check\_out\_date, check\_in\_date)) AS avg\_stay\_duration

FROM Room r

INNER JOIN Reservation res ON r.id = res.room\_id

GROUP BY r.type;

1. מטרת השאילתה

רשימה של כל האורחים שנשארו יותר משבוע:

SELECT g.name, res.check\_in\_date, res.check\_out\_date

FROM Guest g

INNER JOIN Reservation res ON g.id = res.guest\_id

WHERE DATEDIFF(res.check\_out\_date, res.check\_in\_date) > 7;

1. מטרת השאילתה

מוצא את חמשת השירותים המבוקשים ביותר

SELECT s.name, COUNT(rs.service\_id) AS total\_requests

FROM Service s

INNER JOIN ReservationService rs ON s.id = rs.service\_id

GROUP BY s.name

ORDER BY total\_requests DESC

LIMIT 5;

1. מטרת השאילתה

מוצאת את העובד עם המספר הגבוה ביותר של מטלות שירות חדרים

SELECT e.name, COUNT(rsa.employee\_id) AS total\_assignments

FROM Employee e

INNER JOIN RoomServiceAssignment rsa ON e.id = rsa.employee\_id

GROUP BY e.name

ORDER BY total\_assignments DESC

LIMIT 1;

1. מטרת השאילתה

רשימה של כל החדרים שלא עברו ניקיון בשלושת הימים האחרונים

SELECT r.id, r.number

FROM Room r

WHERE NOT EXISTS (

SELECT 1

FROM HousekeepingLog h

WHERE r.id = h.room\_id

AND h.date >= DATE\_SUB(CURDATE(), INTERVAL 3 DAY));

1. מטרת השאילתה

מחזיר את סך ההכנסות בכל חודש בשנה האחרונה

SELECT YEAR(paid\_date) AS year, MONTH(paid\_date) AS month, SUM(amount) AS total\_revenue

FROM Payment

WHERE YEAR(paid\_date) = YEAR(CURRENT\_DATE - INTERVAL 1 YEAR)

GROUP BY YEAR(paid\_date), MONTH(paid\_date)

ORDER BY year, month;

1. מטרת השאילתה

מוצאת אילו מתקנים אירחו הכי הרבה אירועים

SELECT f.name, COUNT(e.id) AS total\_events

FROM Facility f

INNER JOIN Event e ON f.id = e.facility\_id

GROUP BY f.name

ORDER BY total\_events DESC;

1. מטרת השאילתה

מוצאת את האורחים שביצעו מספר הזמנות בחודש

SELECT g.name, COUNT(res.id) AS total\_reservations

FROM Guest g

INNER JOIN Reservation res ON g.id = res.guest\_id

WHERE res.check\_in\_date >= DATE\_SUB(CURDATE(), INTERVAL 1 MONTH)

GROUP BY g.name

HAVING COUNT(res.id) > 1;

1. מטרת השאילתה

מחשבת את שיעור התפוסה עבור כל סוג חדר עבור החודש הנוכחי

SELECT r.type,

COUNT(res.id) AS total\_reservations,

AVG(DATEDIFF(res.check\_out\_date, res.check\_in\_date)) AS avg\_stay\_duration,

(COUNT(res.id) \* 100 / COUNT(r.id)) AS occupancy\_rate

FROM Room r

LEFT JOIN Reservation res ON r.id = res.room\_id

WHERE res.check\_in\_date >= DATE\_SUB(CURDATE(), INTERVAL 1 MONTH)

GROUP BY r.type;

1. מטרת השאילתה

מוצאת את סוג החדר הרווחי ביותר בהתבסס על סך ההכנסות

SELECT r.type, SUM(p.amount) AS total\_revenue

FROM Room r

INNER JOIN Reservation res ON r.id = res.room\_id

INNER JOIN Payment p ON res.id = p.reservation\_id

GROUP BY r.type

ORDER BY total\_revenue DESC

LIMIT 1;

1. מטרת הטריגר

מעדכן אוטומטית את הסטטוס של החדר

CREATE TRIGGER after\_reservation\_insert

AFTER INSERT ON Reservation

FOR EACH ROW

BEGIN

UPDATE Room

SET status = 'occupied'

WHERE id = NEW.room\_id;

END;

1. מטרת הטריגר

מעדכן תוכנית נקודות אחרי התשלום

CREATE TRIGGER after\_payment\_insert

AFTER INSERT ON Payment

FOR EACH ROW

BEGIN

UPDATE Guest

SET loyalty\_program\_points = loyalty\_program\_points + ROUND(NEW.amount / 10)

WHERE id = NEW.guest\_id;

END;

1. מטרת הפונקציה

מחשבת את מחיר החדר לפי הסוג

CREATE FUNCTION CalculateRoomPriceByType(room\_type VARCHAR(50)) RETURNS FLOAT

BEGIN

DECLARE room\_price FLOAT;

SELECT price\_per\_night

INTO room\_price

FROM Room

WHERE type = room\_type;

RETURN room\_price;

END

1. מטרת הפונקציה

מחשבת את המחיר הממוצע של סוג שירות

CREATE FUNCTION CalculateAverageServicePrice() RETURNS FLOAT

BEGIN

DECLARE avg\_price FLOAT;

SELECT AVG(price)

INTO avg\_price

FROM Service;

RETURN avg\_price;

END