**עבודת הגשה בתכנון וניהול מסדי נתונים ERD**

**סטודנטים:**

- מתן אלמליח 205625221

- דניאל פרץ 203550827

**שם המרצה:**

ניר רגב

**תאריך הגשה:**

28.8.2025

תוכן העניינים:

[**ישויות ומאפיינים נדרשים במערכת**](#_q70hfegnvfos) **2**

[**רשימת ישויות – מאפיינים, סוגי נתונים ומפתחות PK/FK**](#_poxnhlp032c8) **3**

[**קשרים בין ישויות ומידת היחסים הקרדינלים**](#_8n8hgn9xv2b0) **8**

[**קשרי הורשה IS A**](#_m6tv7j24kske) **11**

[**תרשים ERD – תרשים גרפי כולל כל הישויות והקשרים**](#_kpqr3qfo4g43) **12**

[**הקוד ליצירת הטבלאות**](#_i4wptqk1jspu) **13**



### 1. ישויות ומאפיינים נדרשים במערכת

המערכת מנהלת נתונים ארגוניים עבור חברה שמוכרת מוצרים ללקוחות. היא כוללת נתוני אנשים (כישות־על people שממנה נגזרות employees, customers, engineers), מבנה ארגוני (offices, sales\_rep) ותהליכי מסחר (product\_lines, products, orders, order\_details, payments). לכל ישות הוגדרו מאפיינים הכרחיים, סוגי נתונים ואילוצי תקינות (תחומי ערכים), כדי לשמור על עקביות ולמנוע כפילויות.

* **יישויות ואובייקטים מרכזיים**
  + people כיישות־על (ISA Total/Disjoint), שממנה נגזרות employees, customers ו־engineers.
  + יישויות ארגוניות: offices, sales\_rep (כתת־קבוצה של employees).
  + יישויות עסקיות: product\_lines, products, orders, order\_details, payments.
* **מאפיינים ותכונות נדרשים**
  + לכל יישות נרשמו מאפיינים חיוניים לניהול המערכת: מזהים (כגון personalId, employeeNumber, productCode), פרטים אישיים (firstName, lastName, email, phoneNumber), נתוני ארגון (officeCode, jobTitle, reportTo), נתוני לקוחות (creditLimit, salesRepEmployeeNumber), ונתוני מוצרים והזמנות (quantityInStock, price, quantityOrdered, status, paymentDate, amount).
* **סוגי נתונים**
  + מזהים ומספרים: INT (חיובי בלבד).
  + מחרוזות: VARCHAR באורך מוגבל (לדוגמה: שמות עד 50 תווים, אימייל עד 100).
  + סכומים כספיים: DECIMAL עם דיוק מתאים (amount > 0).
  + תאריכים: DATE עם אילוצים לוגיים (requiredDate ≥ orderDate, shippedDate ≥ orderDate).
  + ערכים סגורים: עמודות עם רשימת ערכים מוגדרת מראש (למשל status = Process, Shipped, Delivered, Cancelled).
* **תחומי ערכים (Constraints)**
  + כל מזהה חייב להיות ייחודי ולא ריק.
  + סכומים (price, amount, creditLimit) חייבים להיות ≥ 0.
  + ערכי מפתחות זרים חייבים להתאים לערכים קיימים בטבלאות אחרות (FK → PK).
  + קשרי ISA מחייבים שכל אדם יהיה בדיוק ב־subtype אחד (Total + Disjoint).

באמצעות מבנה זה ניתן להבטיח בהירות ואחידות במערכת, להימנע מכפילויות, ולוודא שכל נתון המוזן עומד בכלל תחום הערכים שנקבע לו.

### 2. רשימת ישויות – מאפיינים, סוגי נתונים ומפתחות (PK/FK)

בסעיף זה מפורטות ישויות המערכת ומאפייניהן, כולל סוגי הנתונים ותחומי הערכים, תוך ציון מפורש של המאפיינים העתידים להפוך למפתחות בשלב המימוש הפיזי (DDL): עמודות המסומנות כ-PK ישמשו כמפתח הראשי של הטבלה, ועמודות המסומנות כ-FK יהפכו למפתחות זרים המפנים לטבלת היעד ולעמודת הייחוס המתאימה, במבנה ה-ISA, המאפיין המצביע אל **people** יסומן בנוסף כ-UNIQUE (בכחול) כדי לאכוף יחס 1:1 בין יישות-הבן לבין הרשומה המקורית באנשים.

**- שורה כחולה = קשר הורשה is a**

**- שורה צהובה = קשרי ליבה לטבלה אחרת**

**- שורה ירוקה = קשר לערך אחר באותה הטבלה או קשר רקורסיבי**

**היישויות במערכת — פירוט לפי קטגוריות טבלאות ואילוצים**

# **אנשים ותפקידים** – people, employees, engineers, customers, sales\_rep

| **people** | | | |
| --- | --- | --- | --- |
| **Range of values** | **Key** | **Data type** | **Attribute name** |
| מספר חיובי ייחודי (=>1), **בעזרת המפתח הזה, נוריש מאפיינים לישויות employees, engineers, customers),** AUTO\_INCREMENT | PK | INT | personalID |
| - מחרוזת באורך של עד 50 תווים, NOT NULL |  | VARCHAR(30) | firstName |
| מחרוזת באורך של עד 50 תווים, NOT NULL |  | VARCHAR(30) | lastName |
| מחרוזת באורך של עד 50 תווים (נדרש פורמט תקין של אימייל name@domain.com) ייאכף באפליקציה, UNIQUE |  | VARCHAR(50) | email |
| מספר ישראלי תקין באורך 10 ספרות, עם כל אחד מהקידומות המוכרות בישראל(052/050/054 וכו…), ייאכף באפליקציה |  | INT | phoneNumber |

| **employees** | | | |
| --- | --- | --- | --- |
| **Range of values** | **Key** | **Data type** | **Attribute name** |
| מספר ייחודי חיובי (>=1), AUTO\_INCREMENT | PK | INT | employeeNumber |
| מספר חיובי ייחודי (1 >=), בעזרת המפתח הזה, נירש מאפיינים מהיישות people, חייב להופיע ב people.personalID, UNIQUE, ISA | FK | INT | personalID |
| שם התפקיד(אותיות בלבד, עד 50 תווים, אחד מתוך Developer, Manager, Engineer, SalesRep…), NOT NULL |  | VARCHAR(50) | jobTitle |
| קוד משרד ייחודי(למשל NY02, IL01), חייב להופיע ביישות offices.officeCode, NOT NULL | FK | VARCHAR(15) | officeCode |
| מספר עובד ייחודי, יכול להיות שייך למספר עובד אחר למשל בתפקיד מנהל, חוץ ממנכ״ל שיכול להיות NULL, **קשר רקורסיבי**. | FK | INT | reportTo |

| **engineers** | | | |
| --- | --- | --- | --- |
| **Range of values** | **Key** | **Data type** | **Attribute name** |
| מספר ייחודי חיובי (>=1), AUTO\_INCREMENT | PK | INT | engineerID |
| מספר חיובי ייחודי (1 >=), בעזרת המפתח הזה, נירש מאפיינים מהיישות people, חייב להופיע ב people.personalID, UNIQUE, ISA | FK | INT | personalID |
| אותיות בלבד, עד 50 תווים, אחד, תחום התמחות מתוך רשימה: (Software, Hardware, Network, DevOps, Data), ייאכף באפליקציה |  | VARCHAR(50) | expertise |

| **customers** | | | |
| --- | --- | --- | --- |
| **Range of values** | **Key** | **Data type** | **Attribute name** |
| מספר ייחודי חיובי (>=1), AUTO\_INCREMENT | PK | INT | customerNumber |
| מספר חיובי ייחודי (1 >=), בעזרת המפתח הזה, נירש מאפיינים מהיישות people, חייב להופיע ב people.personalID, UNIQUE, ISA | FK | INT | personalID |
| מספר חיובי ייחודי, חייב להופיע ב - employees.employeeNUmber או NULL | FK | INT | salesRepEmployeeNumber |
| ערך גדול מ-0, ברירת מחדל 0.00 |  | DECIMAL(10,2) | creditLimit |

| **sales\_rep** | | | |
| --- | --- | --- | --- |
| **Range of values** | **Key** | **Data type** | **Attribute name** |
| מספר ייחודי של איש המכירות, ≥1, UNIQUE, AUTO\_INCREMENT | PK | INT | salesRepId |
| קיים וייחודי ב־employees.employeeNumber (קשר 1:1 לעובד) | FK | INT | employeeNumber |
| שם אזור (North, South, Europe, APAC …), עד 50 תווים, ייאכף באפליקציה |  | VARCHAR(50) | region |
| ≤ תאריך היום |  | DATE | hireDate |
| סך הכל מכירות, דצימלית ≥ 0 (ברירת מחדל 0.00) |  | DECIMAL(10,2) | totalSales |

**# מבנה ארגוני** – offices.

| **offices** | | | |
| --- | --- | --- | --- |
| **Range of values** | **Key** | **Data type** | **Attribute name** |
| קוד משרד ייחודי(למשל NY02, IL01), ייאכף באפליקציה | PK | VARCHAR(15) | officeCode |
| שם עיר(אותיות בלבד, עד 50 תווים) |  | VARCHAR(50) | city |
| שם כתובת מגורים(אותיות בלבד, עד 50 תווים) |  | VARCHAR(50) | address |
| שם כתובת מגורים(אותיות בלבד, עד 50 תווים, מהרשימה הבינלאומית המוכרת, USA, UK…), ייאכף באפליקציה |  | VARCHAR(50) | country |

# **קטלוג מוצרים** – product\_lines, products.

| **products** | | | |
| --- | --- | --- | --- |
| **Range of values** | **Key** | **Data type** | **Attribute name** |
| קוד מוצר ייחודי, חייב להופיע ביישות **order\_details.productCode** | PK | VARCHAR(15) | productCode |
| שם מוצר עד 50 תווים |  | VARCHAR(50) | productName |
| חייב להופיע ביישות **product\_lines.productLine** | FK | VARCHAR(50) | productLine |
| מספר חיובי גדול מ-0 |  | INT | quantityInStock |
| מחיר דצימלי גדול מ–0 |  | DECIMAL(10,2) | price |

| **product\_lines** | | | |
| --- | --- | --- | --- |
| **Range of values** | **Key** | **Data type** | **Attribute name** |
| שם קטגוריה ייחודית (Electronics, Furniture, Vehicles, …), חייב להופיע ביישות **products.productLine** | PK | VARCHAR(50) | productLine |
| טקס חופשי המתאר את הקטגוריה של המוצר |  | TEXT | Description |

# **מכירות ותפעול** – orders, order\_details, payments.

| **orders** | | | |
| --- | --- | --- | --- |
| **Range of values** | **Key** | **Data type** | **Attribute name** |
| מספר ייחודי חיובי (>=1), חייב להופיע ביישות **order\_details.orderNumbe**r | PK | INT | orderNumber |
| תאריך חוקי (לא בעתיד רחוק מדי) |  | DATE | orderDate |
| חייב להיות ≥ orderDate |  | DATE | requiredDate |
| NULL או ≥ orderDate |  | DATE | shippedDate |
| חייב להיות (Pending Processed, Shipped, Cancelled) |  | VARCHAR(50) | status |
| חייב להופיע ביישות ב־**customers.customerNumber** | FK | INT | customerNumber |

| **order\_details** | | | |
| --- | --- | --- | --- |
| **Range of values** | **Key** | **Data type** | **Attribute name** |
| חייב להופיע בישות ב־**orders.orderNumber** | PK, FK | INT | orderNumber |
| קוד מוצר ייחודי, חייב להופיע ביישות **products.productCode** | FK | VARCHAR(15) | productCode |
| מספר חיובי גדול מ-0 |  | INT | quantityOrdered |
| מספר דצימלי חיובי גדול מ-0 |  | DECIMAL(10,2) | priceEach |

| **payments** | | | |
| --- | --- | --- | --- |
| **Range of values** | **Key** | **Data type** | **Attribute name** |
| חייב להופיע בישות **customers.customerNumber** | PK, FK | INT | customerNumber |
| מזהה ייחודי לשיק/תשלום | PK | VARCHAR(50) | checkNumber |
| תאריך חוקי ≤ מהיום הנוכחי |  | DATE | paymentDate |
| מספר דצימלי חיובי גדול מ-0 |  | DECIMAL(10,2) | amount |

### 3. קשרים בין ישויות ומידת היחסים הקרדינלים

**עובד ↔ משרדים**

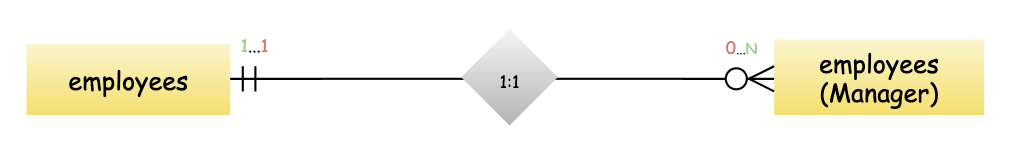
****

**תיאור:** כל משרד (office) בארגון יכול להכיל מספר רב של עובדים (employees) — למשל עובדים המחולקים לפי סניפים גאוגרפיים. עם זאת, כל עובד יכול להיות משויך למשרד אחד בלבד, כלומר לא ייתכן עובד שממוקם ביותר ממשרד אחד.

**קרדינליות:** employee 1..1 ↔ office 0..N

**סוג קשר:** דרגה 2 (קשר בינארי).

**עובד(מנהל - עובדים הכפופים לו) ↔ עובד**

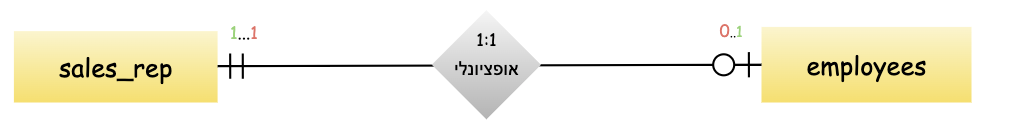
****

**תיאור:** במבנה הארגוני, עובד בתפקיד מנהל (employee) יכול לנהל אפס עובדים או עובדים רבים, כלומר להיות אחראי על מספר כפיפים. כל עובד (שאינו מנכ״ל) חייב לדווח למנהל אחד בלבד. זה קשר רקורסיבי בתוך אותה ישות employees, המייצג את ההיררכיה הארגונית.

**קרדינליות:** employee 1..1 ↔ manager 0..N

**סוג קשר:** דרגה 1 (קשר רקורסיבי)

**איש מכירות ↔ עובדים**

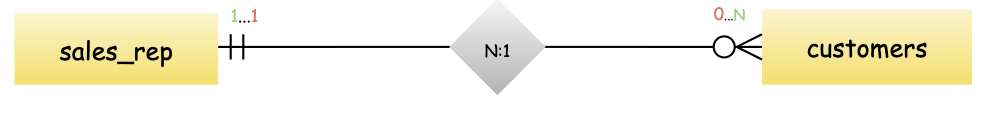


**תיאור:** איש מכירות (**sales\_rep**) הוא עובד מיוחד מתוך ה־employees. כל איש מכירות חייב להיות משויך לעובד אחד בלבד, אך לא כל עובד בהכרח מוגדר כאיש מכירות.

**קרדינליות:** sales\_rep 1..1 ↔ employee 0..1

**סוג קשר:** דרגה 2 (קשר בינארי).

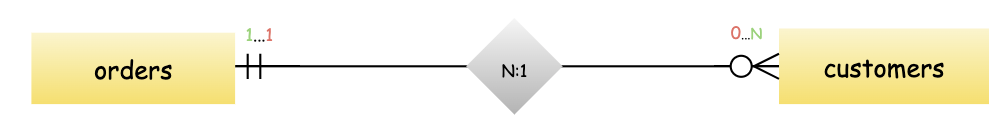
**איש מכירות ↔ לקוחות**



**תיאור:** איש מכירות (sales\_rep) הוא עובד מיוחד מתוך ה־employees, שתפקידו לשרת לקוחות. איש מכירות אחד יכול להיות אחראי על הרבה לקוחות (customers), לטפל בצרכיהם ולבצע עסקאות. כל לקוח חייב להיות משויך לאיש מכירות אחד בלבד.

**קרדינליות:** sales\_rep 1..1 ↔ customer 0..N

**סוג קשר:** דרגה 2 (קשר בינארי).

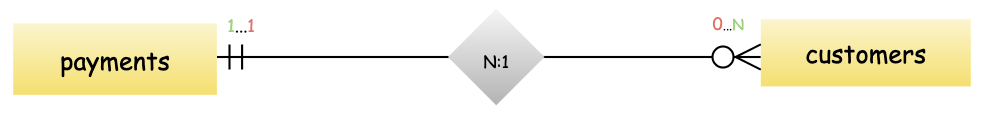
**הזמנות ↔ לקוח**

**תיאור:** כל לקוח (customer) יכול לבצע מספר הזמנות (orders) לאורך זמן – למשל רכישות חוזרות. לעומת זאת, כל הזמנה חייבת להיות קשורה ללקוח מסוים אחד בלבד (הלקוח שביצע אותה בפועל).

**קרדינליות:** order 1..1 ↔ customer 0..N

**סוג קשר:** דרגה 2 (קשר בינארי).

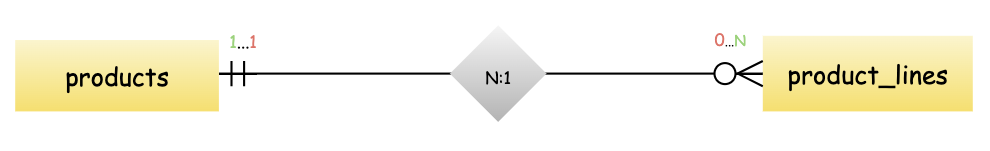
**תשלומים ↔ לקוח**



**תיאור:** כל לקוח (customer) יכול לבצע כמה תשלומים (payments), עבור הזמנות שונות או חלקיות. כל תשלום חייב להיות שייך ללקוח אחד בלבד – זהותו נשמרת במערכת הנהלת החשבונות.  
 **קרדינליות:** payment 1..1 ↔ customer 0..N

**סוג קשר:** דרגה 2 (קשר בינארי).

**מוצרים ↔ קו מוצרים**

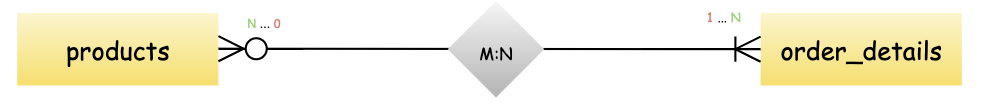


**תיאור:** כל קו מוצרים (product\_line) מגדיר קטגוריה עסקית, כמו "מחשבים ניידים" או "טלפונים ניידים". בתוך כל קו כזה יכולים להיכלל מוצרים רבים (products). עם זאת, כל מוצר חייב להשתייך לקו מוצרים אחד בלבד — לא ניתן לשייך מוצר לשני קווים שונים בו־זמנית.

**קרדינליות:** product 1..1 ↔ product line 0..N

**סוג קשר:** דרגה 2 (קשר בינארי).

**הזמנות ↔ מוצרים**



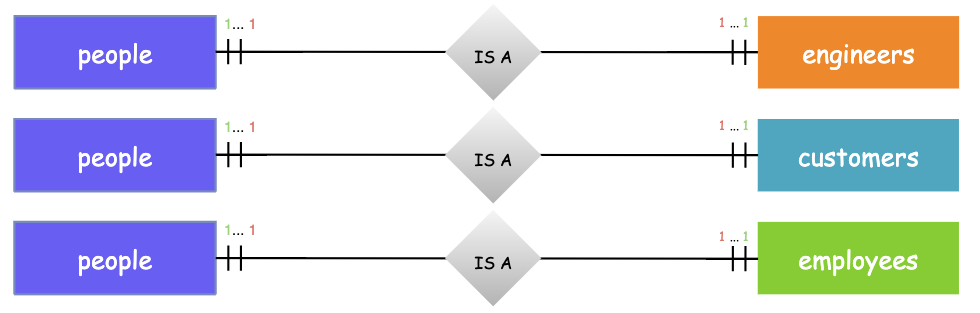
**תיאור:** לקוח שמבצע הזמנה יכול לכלול בה מספר מוצרים שונים (לדוגמה: טלפון + אוזניות + כיסוי). מצד שני, אותו מוצר יכול להופיע בהרבה הזמנות של לקוחות שונים — אך ייתכן שגם לא יופיע כלל בהזמנות (למשל מוצר חדש במלאי שטרם נרכש). זהו קשר רבים לרבים (M:N), הממומש באמצעות טבלת ביניים בשם **order\_details**, שמכילה גם את כמות המוצר ואת המחיר.

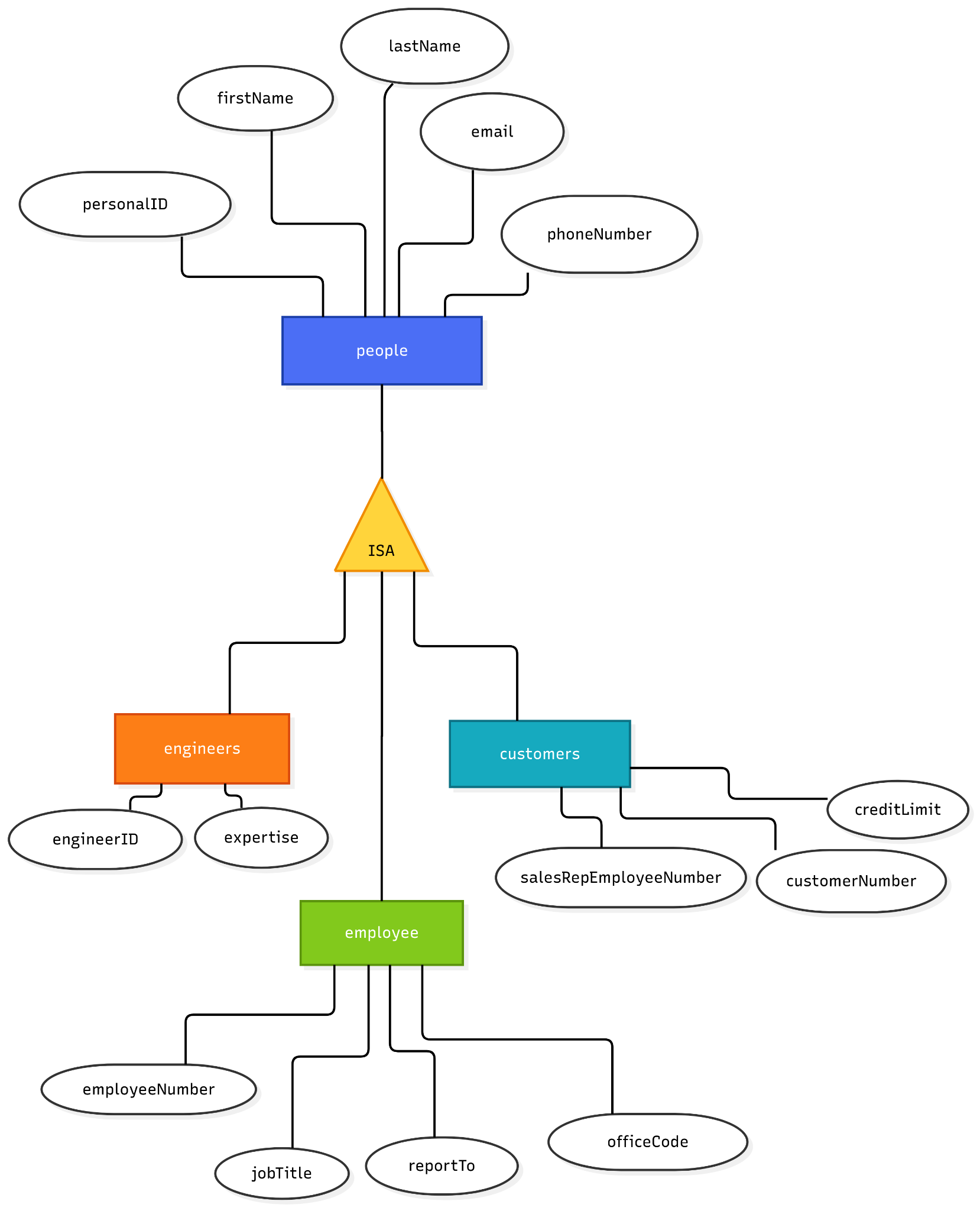
**קרדינליות:** product N..0 ↔ order\_details 1..N

**סוג קשר:** דרגה 2 (בינארי - רבים לרבים).

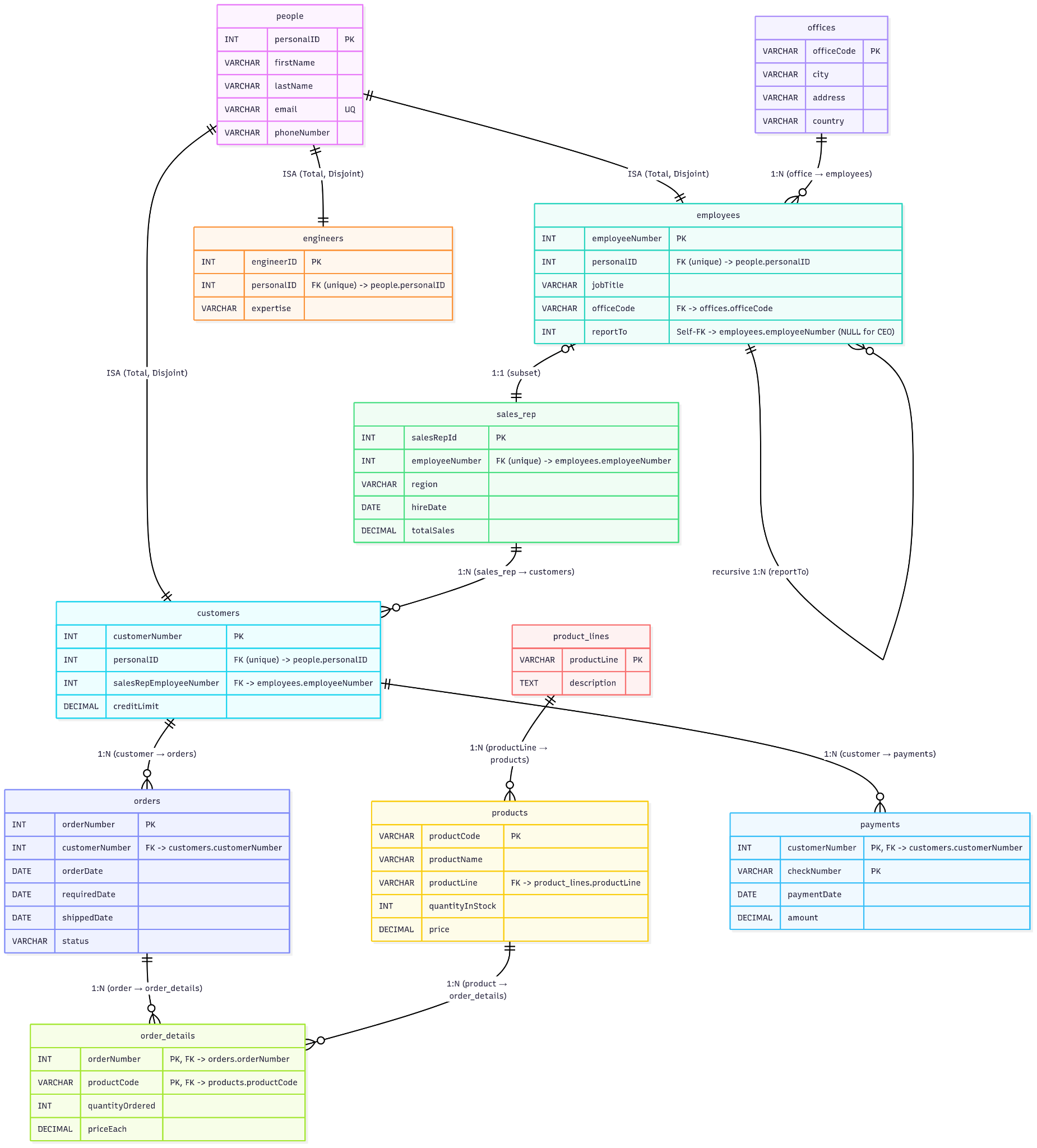
### 4. קשרי הורשה **IS A**

במערכת מוגדרת יישות־על **people** הכוללת את המאפיינים הבסיסיים של כל אדם (מזהה, שם, טלפון ודוא״ל). ממנה נגזרות שלוש יישויות־בן: **employees** (עם מספר עובד, תפקיד, משרד ומנהל ישיר), **customers** (עם מספר לקוח, איש מכירות אחראי ותקרת אשראי) ו־**engineers** (עם מזהה מהנדס ותחום התמחות). קשר ה־ISA ביניהן הוא **Total ו־Disjoint** – כל אדם חייב להשתייך לאחת מהיישויות בלבד, וללא חפיפה ביניהן.





### 5. **תרשים ERD** – תרשים גרפי כולל כל הישויות והקשרים



### 6. הקוד ליצירת הטבלאות

**יצירת הטבלאות לאנשים ותפקידים** – people, employees, engineers, customers, sales\_rep

CREATE TABLE people (

personalID INT PRIMARY KEY AUTO\_INCREMENT,

firstName VARCHAR(30) NOT NULL,

lastName VARCHAR(30) NOT NULL,

email VARCHAR(50) UNIQUE,

phoneNumber VARCHAR(10),

CHECK (CHAR\_LENGTH(firstName) BETWEEN 1 AND 30),

CHECK (CHAR\_LENGTH(lastName) BETWEEN 1 AND 30)

);

CREATE TABLE employees (

employeeNumber INT PRIMARY KEY AUTO\_INCREMENT,

personalID INT NOT NULL UNIQUE,

jobTitle VARCHAR(50) NOT NULL,

officeCode VARCHAR(15) NOT NULL,

reportTo INT NULL,

FOREIGN KEY (personalID) REFERENCES people(personalID) ON DELETE CASCADE,

FOREIGN KEY (officeCode) REFERENCES offices(officeCode) ON DELETE RESTRICT,

FOREIGN KEY (reportTo) REFERENCES employees(employeeNumber) ON DELETE SET NULL,

CHECK (CHAR\_LENGTH(jobTitle) BETWEEN 1 AND 50)

);

CREATE TABLE engineers (

engineerID INT PRIMARY KEY AUTO\_INCREMENT,

personalID INT NOT NULL UNIQUE,

expertise VARCHAR(50) NOT NULL,

FOREIGN KEY (personalID) REFERENCES people(personalID) ON DELETE CASCADE,

CHECK (CHAR\_LENGTH(expertise) BETWEEN 1 AND 50)

);

CREATE TABLE sales\_rep (

salesRepId INT PRIMARY KEY AUTO\_INCREMENT,

employeeNumber INT NOT NULL UNIQUE,

region VARCHAR(50) NOT NULL,

hireDate DATE NOT NULL,

totalSales DECIMAL(15,2) DEFAULT 0.00,

FOREIGN KEY (employeeNumber) REFERENCES employees(employeeNumber) ON DELETE CASCADE,

CHECK (totalSales >= 0),

CHECK (CHAR\_LENGTH(region) BETWEEN 1 AND 50),

CHECK (hireDate <= CURRENT\_DATE)

);

CREATE TABLE customers (

customerNumber INT PRIMARY KEY AUTO\_INCREMENT,

personalID INT NOT NULL UNIQUE,

salesRepEmployeeNumber INT NULL,

creditLimit DECIMAL(10,2) DEFAULT 0.00,

FOREIGN KEY (personalID) REFERENCES people(personalID) ON DELETE CASCADE,

FOREIGN KEY (salesRepEmployeeNumber) REFERENCES employees(employeeNumber) ON DELETE SET NULL,

CHECK (creditLimit >= 0)

);

**יצירת הטבלאות למבנה ארגוני** – offices

CREATE TABLE offices (

officeCode VARCHAR(15) PRIMARY KEY,

city VARCHAR(50) NOT NULL,

address VARCHAR(50),

country VARCHAR(50) NOT NULL

);

**יצירת הטבלאות לקטלוג המוצרים** – product\_lines, products.

CREATE TABLE products (

productCode VARCHAR(15) PRIMARY KEY,

productName VARCHAR(50) NOT NULL,

productLine VARCHAR(50) NOT NULL,

quantityInStock INT NOT NULL DEFAULT 0,

price DECIMAL(10,2) NOT NULL,

FOREIGN KEY (productLine) REFERENCES product\_lines(productLine) ON DELETE RESTRICT,

CHECK (quantityInStock >= 0),

CHECK (price >= 0),

CHECK (CHAR\_LENGTH(productName) BETWEEN 1 AND 50)

);

CREATE TABLE product\_lines (

productLine VARCHAR(50) PRIMARY KEY,

description TEXT

);

**יצירת הטבלאות למכירות ותפעול** – orders, order\_details, payments.

CREATE TABLE payments (

customerNumber INT NOT NULL,

checkNumber VARCHAR(50) NOT NULL,

paymentDate DATE NOT NULL,

amount DECIMAL(10,2) NOT NULL,

PRIMARY KEY (customerNumber, checkNumber),

FOREIGN KEY (customerNumber) REFERENCES customers(customerNumber) ON DELETE RESTRICT,

CHECK (amount > 0),

CHECK (paymentDate <= CURRENT\_DATE)

);

CREATE TABLE orders (

orderNumber INT PRIMARY KEY AUTO\_INCREMENT,

customerNumber INT NOT NULL,

orderDate DATE NOT NULL,

requiredDate DATE NOT NULL,

shippedDate DATE NULL,

status VARCHAR(50) NOT NULL,

FOREIGN KEY (customerNumber) REFERENCES customers(customerNumber) ON DELETE RESTRICT,

CHECK (status IN ('Pending','Processed','Shipped','Cancelled')),

CHECK (requiredDate >= orderDate),

CHECK (shippedDate IS NULL OR shippedDate >= orderDate),

CHECK (orderDate <= CURRENT\_DATE)

);

CREATE TABLE order\_details (

orderNumber INT NOT NULL,

productCode VARCHAR(15) NOT NULL,

quantityOrdered INT NOT NULL,

priceEach DECIMAL(10,2) NOT NULL,

PRIMARY KEY (orderNumber, productCode),

FOREIGN KEY (orderNumber) REFERENCES orders(orderNumber) ON DELETE CASCADE,

FOREIGN KEY (productCode) REFERENCES products(productCode) ON DELETE RESTRICT,

CHECK (quantityOrdered > 0),

CHECK (priceEach >= 0)

);