

۱. کوئریهای ساخت جدول برای هر موجودیت در نمودار ER را ایجاد کنید.

```
CREATE TABLE TPL (
    TP_ID NUMERIC PRIMARY KEY,
    first_name VARCHAR(50) NOT NULL,
    last_name VARCHAR(50) NOT NULL
);

CREATE TABLE User (
    user_ID NUMERIC PRIMARY KEY,
    first_name VARCHAR(50) NOT NULL,
    last_name VARCHAR(50) NOT NULL,
    Email VARCHAR(100) UNIQUE NOT NULL,
    Password VARCHAR(100) NOT NULL,
    Age NUMERIC NOT NULL,
    Type VARCHAR(10) NOT NULL CHECK (Type IN ('simple', 'admin')),
    TP_ID NUMERIC NOT NULL,
    phone_number VARCHAR(15) NOT NULL CHECK (phone_number GLOB
'+[0-9]*'),
    FOREIGN KEY (TP_ID) REFERENCES TPL(TP_ID) ON DELETE CASCADE ON UPDATE
CASCADE
);

CREATE TABLE Warehouse (
    warehouse_ID NUMERIC PRIMARY KEY,
    warehouse_name VARCHAR(50) NOT NULL,
    is_open BOOLEAN NOT NULL,
    creation_date TIMESTAMP NOT NULL,
    country VARCHAR(50) NOT NULL,
    TP_ID NUMERIC NOT NULL,
    FOREIGN KEY (TP_ID) REFERENCES TPL(TP_ID) ON DELETE CASCADE ON UPDATE
CASCADE,
    FOREIGN KEY (country) REFERENCES Country(country_name)
);

CREATE TABLE Country (
    country_name VARCHAR(50) PRIMARY KEY
);

CREATE TABLE Catalog (
    catalog_ID NUMERIC PRIMARY KEY,
    product_type VARCHAR(50) NOT NULL,
    manufacturing_cost NUMERIC NOT NULL,
    selling_price NUMERIC NOT NULL,
    country_origin VARCHAR(50) NOT NULL,
    weight NUMERIC NOT NULL,
    length NUMERIC NOT NULL,
    description VARCHAR(255),
    product_maintenance_ID VARCHAR(50) UNIQUE NOT NULL,
    product_name VARCHAR(100) NOT NULL,
    TP_ID NUMERIC NOT NULL,
    registered_by_user_ID NUMERIC NOT NULL,
    ordered_by_user_ID NUMERIC,
    image BLOB,
    FOREIGN KEY (TP_ID) REFERENCES TPL(TP_ID) ON DELETE CASCADE ON UPDATE
CASCADE,
    FOREIGN KEY (registered_by_user_ID) REFERENCES User(user_ID) ON
DELETE CASCADE ON UPDATE CASCADE,
```

```

        FOREIGN KEY (ordered_by_user_ID) REFERENCES User(user_ID) ON DELETE
CASCADE ON UPDATE CASCADE,
        FOREIGN KEY (country_origin) REFERENCES Country(country_name),
        CHECK (manufacturing_cost < selling_price)
);

CREATE TABLE Orders (
    order_ID NUMERIC PRIMARY KEY,
    order_date TIMESTAMP NOT NULL,
    Total_cost NUMERIC NOT NULL,
    TP_ID NUMERIC NOT NULL,
    order_user_ID NUMERIC NOT NULL,
    warehouse_ID NUMERIC NOT NULL,
    order_status VARCHAR(50) NOT NULL CHECK (order_status IN
('Processing', 'Preparation', 'In stock', 'Sent', 'Received',
'Returned')),
    order_type VARCHAR(50) NOT NULL CHECK (order_type IN ('Online', 'In
person')),
    user_notes VARCHAR(255),
    is_gift BOOLEAN NOT NULL,
    FOREIGN KEY (TP_ID) REFERENCES TPL(TP_ID) ON DELETE CASCADE ON UPDATE
CASCADE,
    FOREIGN KEY (order_user_ID) REFERENCES User(user_ID) ON DELETE
CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (warehouse_ID) REFERENCES Warehouse(warehouse_ID) ON
DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE OrdersCatalogJoin (
    order_ID NUMERIC,
    catalog_ID NUMERIC,
    PRIMARY KEY (order_ID, catalog_ID),
    FOREIGN KEY (order_ID) REFERENCES Orders(order_ID) ON DELETE CASCADE
ON UPDATE CASCADE,
    FOREIGN KEY (catalog_ID) REFERENCES Catalog(catalog_ID) ON DELETE
CASCADE ON UPDATE CASCADE
);

CREATE TABLE Location (
    location_ID NUMERIC PRIMARY KEY,
    has_container BOOLEAN NOT NULL,
    TP_ID NUMERIC NOT NULL,
    warehouse_ID NUMERIC NOT NULL,
    notes VARCHAR(255),
    latitude NUMERIC NOT NULL,
    longitude NUMERIC NOT NULL,
    registered_by_user_ID NUMERIC NOT NULL,
    FOREIGN KEY (TP_ID) REFERENCES TPL(TP_ID) ON DELETE CASCADE ON UPDATE
CASCADE,
    FOREIGN KEY (warehouse_ID) REFERENCES Warehouse(warehouse_ID) ON
DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (registered_by_user_ID) REFERENCES User(user_ID) ON
DELETE CASCADE ON UPDATE CASCADE
);

```

۲. جدول را با مقادیری داده پر کنید و همچنین بررسی کنید که با داده‌هایی که هم‌نوع نیستند پر نمی‌شود.

پرکردن جداول در فایل generate_data.py قرار دارد. اما بررسی داده های غیر هم نوع، با توجه به آنکه ما از sqlite در این پروژه استفاده کرده ایم، این DBMS قید های محدودیت در تعاریف جداول را بررسی نمی کند و به اشتباه می توان تاپل هایی با صفت های غیر هم نوع وارد کرد. برای مدیریت این مشکل باید از check استفاده کرد.

۳. جدولها را با داده های غلط پر کنید و سپس آنها را درست نمایید و در آخر آنها را پاک کنید.

```
INSERT INTO Orders (order_ID, order_date, Total_cost, order_user_id,
TP_ID, order_user_ID, warehouse_ID, order_status, order_type, user_notes,
is_gift)
VALUES (3, '2023-03-20 09:00:00', 500, 103, 1, 203, 303, 'InvalidStatus',
'Online', 'Test invalid status', 0);
```

SQLite

SQLITE_CONSTRAINT_CHECK: sqlite3 result code 275: CHECK constraint failed: order_status IN ('Processing', 'Preparation', 'In stock', 'Sent', 'Received', 'Returned')

```
INSERT INTO Warehouse (warehouse_ID, warehouse_name, is_open,
creation_date, country, TP_ID)
VALUES (10001, 'mainWarehouse', 1, '2023-07-01 08:00:00', 'Italy',
10001);
```

SQLite

SQLITE_CONSTRAINT_FOREIGNKEY: sqlite3 result code 787: FOREIGN KEY constraint failed

```
INSERT INTO TPL (TP_ID, first_name, last_name)
VALUES (100001, NULL, NULL);
```

SQLite

SQLITE_CONSTRAINT_NOTNULL: sqlite3 result code 1299: NOT NULL constraint failed: TPL.first_name

```
INSERT INTO Country (country_name) VALUES
('ThisCountryNameIsWayTooLongAndShouldFailToInsertIntoThisTable');
```

```
Update Country
set country_name = 'IRAN'
where country_name =
'ThisCountryNameIsWayTooLongAndShouldFailToInsertIntoThisTable';
```

```
Delete from Country
where country_name = 'IRAN';
```

(در واقع sqlite محدودیت های طول را در varchar اعمال نمی کند و نیاز به check می باشد) این رویکرد برای

numeric(a,c) نیز می باشد که به طور خاص برای مثال بالا، دستور check آن را می نویسیم:

```
CREATE TABLE Country ( country_name VARCHAR(50) PRIMARY KEY
CHECK(length(country_name) <= 50) );
```

۴. کوئری بنویسید که اسم و ایمیل تمام کاربرانی که سفارش ثبت کرده اند را نمایش بدهد.

```
select u.first_name, u.last_name, u.Email
from user as u
where u.user_ID in (
select order_user_id from Orders
);
```

۵. کوئری بنویسید که سود حاصل از فروش هر کاتالوگ را نمایش بدهد.

```
select c.catalog_ID, c.product_name, (c.selling_price -  
    c.manufacturing_cost) as profit  
from Catalog c;
```

۶. کوئری بنویسید که تعداد سفارش داده شده از هر کاتالوگ را نمایش بدهد.

```
select product_name, product_maintenance_ID, count(catalog_id) as  
times_of_order  
from Catalog  
where ordered_by_user_id is not null  
GROUP by product_name, product_maintenance_ID;
```

۷. کوئری بنویسید که تعداد مکانهای هر مخزن را نمایش بدهد.

```
select W.warehouse_ID, W.warehouse_name, count(L.location_ID) as  
Location_numbers  
from Warehouse W  
join Location L on W.warehouse_ID = L.warehouse_ID  
GROUP by W.warehouse_ID, W.warehouse_name;
```

۸. کوئری بنویسید که مقدار هزینه ساخت هر سفارش را برای تمامی فروشگاهها نمایش دهد.

```
select O.order_ID, T.TP_ID, sum(C.manufacturing_cost) as  
manufacturing_cost  
from Orders O  
NATURAL join OrdersCatalogJoin OC, Catalog C, TPL T  
where C.catalog_ID = OC.catalog_ID and T.TP_ID = C.TP_ID  
GROUP by O.order_ID, T.TP_ID;
```

۹. کوئری بنویسید که میانگین سود کل را در یک ماه خاص برای هر فروشگاه نمایش بدهد.

```
select order_profits.TP_ID, avg(profit) as average_profit  
from (  
    select T.TP_ID, O.order_ID, sum(C.selling_price - C.manufacturing_cost)  
as profit  
    from TPL As T  
    left outer join Orders O on O.TP_ID = T.TP_ID  
    join OrdersCatalogJoin OC on OC.order_ID = O.order_ID  
    join Catalog C on OC.catalog_ID = C.catalog_ID  
    WHERE STRFTIME('%m', o.order_date) = 'specificMonth' AND STRFTIME('%Y',  
o.order_date) = 'specificYear'  
    GROUP by T.TP_ID, O.order_ID  
) as order_profits  
GROUP by order_profits.TP_ID;
```

۱۰. کوئری بنویسید که تمامی محصولات موجود به همراه شناسه نگهداری محصول آنها را نمایش بدهد.

```
select C.product_name, C.product_maintenance_ID, count(C.catalog_ID) as  
count  
from Catalog as C  
where C.ordered_by_user_id is NULL  
GROUP by C.product_name, C.product_maintenance_ID;
```

۱۱. کوئری بنویسید که هزینه کل و تعداد سفارشات هر مشتری را در یک بازه زمانی خاص نمایش بدهد.

```
select U.user_ID, COUNT(O.order_ID), sum(O.Total_cost)  
from User U  
left outer join Orders O on U.user_ID = O.order_user_ID  
WHERE O.order_date BETWEEN :start_date AND :end_date  
group by U.user_ID;
```

۱۲. کوئری بنویسید که تعداد کانتینرهای هر مخزن را نمایش بدهد.

```
select W.warehouse_ID, W.warehouse_name, sum(L.has_container) as  
Location_numbers  
from Warehouse W  
left OUTER join Location L on W.warehouse_ID = L.warehouse_ID  
GROUP by W.warehouse_ID, W.warehouse_name;
```