

طراحی پایگاه داده ها

دانشکده مهندسی کامپیوتر

مریم رضایی
بهار ۱۴۰۳



تمرین دوم

تاریخ انتشار: ۱ اردیبهشت ۱۴۰۳

۱. پرسش های خود درمورد این تمرین را در سامانه کوثر مطرح کنید.

۲. سیاست ارسال با تاخیر: شما در مجموع در طول نیم سال می توانید از ۹ روز تاخیر استفاده کنید. همچنین هر تمرین را می توانید تا حداکثر ۴ روز با تاخیر تحویل دهید. تاخیرها با مقیاس ساعت محاسبه شده و به بالا گرد می شوند.

۳. سیاست مشارکت دانشجویان در حل کردن تمرین: دانشجویان می توانند در حل تمرین برای رفع ابهام و یا به دست آوردن ایده های کلی با یکدیگر مشورت و همفکری کنند. این کار مورد تایید و تشویق تیم ارائه ای درس می باشد؛ چرا که هم فکری و کار گروهی می تواند موجب تقویت یادگیری شود. اما به دست آوردن جزئیات راه حل و نگارش پاسخ باید تماماً توسط خود دانشجو انجام شود. حتماً در انتهای پاسخ های ارسالی خود نام افرادی که با آنها همفکری کردید را ذکر کنید.

۴. این تمرین متشکل از دو بخش تئوری و عملی با ددلاین یکسان می باشد. در فایل zip پاسخ خود یک pdf برای سوالات تئوری و یک پوشه با نام sql برای سوالات عملی قرار دهید. پوشه sql نیز در درون خود باید برای هر سوال عملی یک پوشه با نام شماره سوال داشته باشد. درون پوشه سوال نیز پاسخ بخش های مختلف سوال را قرار دهید. به طور مثال برای سوال چهارم عملی درون پوشه ۴، فایل های Q4-1.sql و Q4-2.sql را قرار دهید.

۵. کد های SQL باید در دیتابیس PostgreSQL قابل اجرا باشند

سوالات تئوری (۴۰ نمره)

تاریخ تحویل: ۱۸ اردیبهشت ۱۴۰۳

پرسش ۱ (۱۰ نمره)

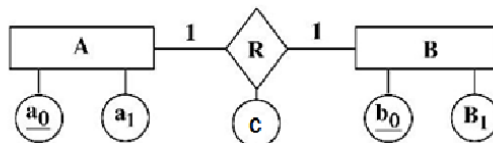
(آ) با توجه به دیتابیس معرفی شده در زیر که مربوط به یک بانک می باشد:

branch (branch_id, branch_name, branch_city, assets)
customer (customer_id, customer_name, customer_street, customer_city)
loan (loan_number, branch_id, amount)
borrower (customer_id, loan_number)
account (account_number, branch_id, balance)
depositor (customer_id, account_number)

i. مشخص کنید کلید های اصلی^۱ مناسب کدامند؟

ii. با توجه به کلید های اصلی که در بخش قبل معرفی کردید، کلید های خارجی^۲ را مشخص کنید.

(ب) برای نمودار ER زیر، سه طراحی ارائه شده است. برای هر کدام بیان کنید که در صورت صحت طراحی، در چه شرایطی موجه است. (کلید خارجی با علامت نقطه چین زیر خصیصه^۳ مشخص شده است.)



i. $ABR(\underline{a_0}, \underline{b_0}, a_1, b_1, c)$

ii. $A(\underline{a_0}, a_1, \underline{b_0}), BR(\underline{b_0}, b_1, c)$

iii. $A(\underline{a_0}, a_1), B(\underline{b_0}, b_1), R(\underline{a_0}, \underline{b_0}, c)$

۱. کلید اصلی: primary key
۲. کلید خارجی: foreign key
۳. خصیصه: attribute

پرسش ۲ (۱۵ نمره) فرض کنید که شما یک دیتابیس برای مدیریت کارمندان شامل جداول زیر است.

employees (employee_id, name, department_id)

departments (department_id, name)

tasks (task_id, description, employee_id)

هر کارمند مربوط به یک دپارتمان می باشد.

هر کارمند میتواند چند تسک داشته باشد.

هر تسک به وسیله کلید خارجی employee_id به کارمند مربوط می شود.

هر کارمند به وسیله کلید خارجی department_id به دپارتمان مربوط می شود.

در هر سناریو، اتفاقاتی را که در دیتابیس رخ می دهد را تشریح کنید.

(آ) اگر یک کارمند استعفا دهد (رکورد هایش را حذف کنیم) و از قاعده حذف cascade استفاده کنیم.

(ب) اگر یک دپارتمان را بخواهیم حذف کنیم و از قاعده حذف restricted استفاده کنیم.

(ج) سناریو قسمت الف را برای قاعده حذف set to null تشریح کنید.

(د) سناریو قسمت ب را برای قاعده حذف set to default تشریح کنید.

پرسش ۳ (۱۵ نمره) تصور کنید یک پایگاه داده برای یک سیستم کتابخانه با جداولی همچون Authors، Title، Books و Checkouts وجود دارد.

جدول Books شامل ویژگی هایی نظیر BookID (PK)، Title و AuthorID (FK) است. جدول Authors شامل ویژگی های AuthorID (PK)، AuthorName و Genre است. جدول Checkouts حاوی CheckoutID (PK)، BookID (FK)، StudentID (FK) و DueDate است.

(آ) یک سناریو ارائه دهید که یک دید پذیرا^۳ با نام CheckedOutBooks برای کارکنان کتابخانه مفید باشد. با استفاده از اطلاعاتی که میتواند از جداول

Books و Checkouts به دست آورده شود، ویژگی هایی که این دید^۱ ممکن است شامل شود را مشخص کنید و توضیح دهید چگونه این دید فرآیند

مدیریت و پیگیری کتب امانت گرفته را ساده می کند.

(ب) چالش های ممکن در ارتباط با یکپارچگی^۲ داده در مواقعی که چندین transaction به صورت همزمان رخ می دهد را بررسی کنید. پیشنهاداتی برای

مدیریت یکپارچه داده ها ارائه کنید. چه مکانیزمی ممکن است باعث اطمینان از یکپارچگی داده ها شود؟

(ج) در زیر دو قطعه کد SQL دارید که هر یک مربوط به ایجاد یک دید است. کد را تجزیه و تحلیل کرده و مشخص کنید کدام یک پذیرا و کدام یک

ناپذیر است^۴. برای پاسخ خود دلیل بیاورید. توضیح دهید پذیرا و ناپذیرا بودن هر کدام به چه معناست.

View1:

```
CREATE VIEW ArchivedBooks AS
```

```
SELECT Books.BookID, Books.Title, Checkouts.DueDate, Checkouts.StudentID
```

```
FROM Books
```

```
JOIN Checkouts ON Books.BookID = Checkouts.BookID
```

```
WHERE Checkouts.DueDate <= CURRENT_DATE;
```

View2:

```
CREATE OR REPLACE VIEW CheckedOutBooks AS
```

```
SELECT Books.BookID, Books.Title, Checkouts.DueDate, Checkouts.StudentID
```

```
FROM Books
```

```
JOIN Checkouts ON Books.BookID = Checkouts.BookID
```

```
WHERE Checkouts.DueDate > CURRENT_DATE;
```

پاسخ تئوری ()

پاسخ ۱ (۱۰ نمره)

(آ) i. branch_id •

customer_id •

loan_number •

customer_id + loan_number •

account_number •

customer_id + account_number •

ندارد • ii.

ندارد •

branch_id •

۱. دید: view

۲. یکپارچگی: integrity

۳. دید پذیرا: updatable view

۴. دید ناپذیرا: non-updatable view

- customer_id, loan_number
- branch_id
- customer_id, account_number

- (ب) i. صحت ندارد، زیرا مشارکت طرفین در رابطه الزامی نیست و همچنین اینکه این روش زمانی به کار می رود که تعداد صفات موجودیت ها کم باشد. همچنین با کلید اصلی در نظر گرفتن هر دو a, b ممکن است شرط یک به یک بودن رابطه نقض شود.
- ii. در صورتی که ارتباط از سمت A الزامی باشد یا صفت b در A به صورت *NotNullable* در نظر گرفته شده باشد، موجه و صحیح خواهد بود.
- iii. صحت ندارد، زیرا با کلید اصلی در نظر گرفتن هر دو a, b ، موجودیت R، شرط یک به یک بودن می تواند نقض شود.

پاسخ ۲ (۱۵ نمره)

- (آ) در cascade، اگر یک عملیات حذف در جدول والد انجام شود، رکورد های مربوطه نیز در جدول فرزند حذف خواهند شد. در اینجا، با استعفا هر کارمند، تسک هایی مربوط به آن کارمند نیز حذف خواهد شد.
- (ب) در قاعده restricted، اگر در جدول فرزند، رکورد هایی وابسته به رکوردی در جدول پدر وجود داشته باشد از حذف رکورد در جدول پدر جلوگیری می شود. در اینجا چون کارمندان وابسته به دپارتمان می باشند، از حذف دپارتمان جلوگیری می شود.
- (ج) در قاعده set to null، وقتی رکوردی در جدول والد حذف می شود، مقدار کلید خارجی را در رکورد های مربوطه در جدول فرزند را به null تغییر می دهیم. در اینجا، با استعفای کارمند، در جدول تسک ها، مقدار کلید های خارجی که منتسب به این کارمند بودند، به null تغییر پیدا می کند، به این معنی که این تسک ها دیگر منتسب به کارمندی نمی باشند.
- (د) در قاعده set to default، وقتی رکوردی در جدول والد حذف میشود، مقدار کلید خارجی را در رکورد های مربوطه در جدول فرزند را به مقدار default value تغییر می دهیم. در اینجا، اگر دپارتمانی حذف شود، مقدار department_id در جدول کارمندان به مقدار default تغییر می یابد تا مطمئن باشیم که همچنان هر کارمند به یک دپارتمان منتسب خواهد ماند.

پاسخ ۳ (۱۵ نمره)

- (آ) سناریویی را در نظر بگیرید که کارکنان کتابخانه میخواهند کتاب هایی که در حال حاضر امانت گرفته شده اند را به طور بهینه پیگیری کنند. دید پذیرای CheckedOutBooks می تواند ویژگی هایی نظیر DueDate Title, BookID و StudentID را شامل شود. این دید فرایند پیگیری کتاب های امانت گرفته شده را با ارائه کردن بی درنگ اطلاعات برای آیتم ها امانت گرفته شده ساده تر کرده و باعث می شود کارکنان بتوانند تمام موجودی کتاب ها و تواریخ بازگرداندن آن ها را مدیریت کنند.
- (ب) چالش هایی در زمینه حفظ یکپارچگی و یکدستی داده ها مخصوصا در حین transaction های همزمان می توانند به وجود بیایند. مشکلات احتمالی می تواند حالت های race را شامل شود که در آن ها چندین transaction تلاش می کنند به طور همزمان یک رکورد را تغییر دهند. این موضوع می تواند باعث بروز ناهنجاری هایی مانند به روزرسانی های از دست رفته و یا داده های commit نشده شود. برای مقابله با این چالش ها، به کارگیری سطوح ایزوله کردن transaction ها با استفاده از مکانیزم های قفل کردن مناسب و اضافه کردن فرایندهای error handling پیشنهاد می شود.
- (ج) دید اول ناپذیرا و دید دوم پذیراست. کلید واژه Create or Replace اشاره به پذیرا بودن دید دوم می کند. این به این معناست که ساختار دید قابل تغییر بوده و باعث می شود که این دید با تغییرهای مورد نیاز، سازگار باشد. این ویژگی به این دید اجازه پذیرش تغییرات پویا را می دهد. به همین ترتیب دید ArchivedBooks پس از ساخته شدن، قابل تغییر نخواهد بود و ساختاری ثابت دارد.

سوالات عملی (۱۲۰ نمره)

پرسش ۱ (۴۰ نمره) شما به عنوان یک مدیر پایگاه داده برای یک شرکتی که یک شبکه از بیمارستان ها را مدیریت می کند، کار می کنید. پایگاه داده شامل اطلاعاتی درباره بیماراران، پزشکان، پرونده های پزشکی، ویزیت ها و داروها است. ساختار موجودیت ها و ویژگی های آن ها در ادامه آمده است. به سوالات داده شده پاسخ بدهید. در نهایت از دیتابیس خود backup گرفته و با نام 1.dump در پوشه مربوط به این سوال قرار دهید. برای backup گرفتن میتوانید از منابع موجود در اینترنت کمک بگیرید همچنین ابزار هایی مانند DBeaver این کار را راحت تر انجام میدهند.

Patients (patient_id, name, dob, gender, address)
 Doctors (doctor_id, name, specialization, hospital_id)
 Hospitals (hospital_id, name, address, phone)
 MedicalRecords (record_id, patient_id, doctor_id, diagnosis, treatment)
 Appointments (appointment_id, patient_id, doctor_id, appointment_date, status)
 Medications (medication_id, name, manufacturer, dosage, side_effects)

- (آ) کوثری هایی برای ساخت (CREATE) جدول های داده شده را بنویسید. باید کوثری هایتان به گونه ای باشد که اگر یک جدول وجود داشت دوباره ساخته نشود. تمام کلیدهای اصلی (PRIMARY KEY) و کلیدهای خارجی (FOREIGN KEY) را مشخص کنید.
- (ب) به هر یک از جدول ها تعدادی رکورد اضافه (INSERT) کنید.
- (ج) برای تمام ویزیت های برنامه ریزی شده قبل از تاریخ امروز که هنوز وضعیت "pending" دارند، وضعیت را به "completed" به روز رسانی (UPDATE) کنید.
- (د) مقدار دوز داروهای (Medications) که کلمه "severe" در عوارض جانبی آنها وجود دارد را ۵۰ درصد افزایش دهید، اما پس از تغییر داده شده دوز هیچ دارویی نباید بیشتر از ۱۰۰۰ میلی گرم شود.
- (ه) جدول Medications و Appointments را نمایش دهید و تغییراتی که در قسمت های قبل داده اید را مشخص کنید.

- (و) تعداد کل بیماران در هر بیمارستان را مشخص کنید، حتی بیمارستان‌هایی که هیچ بیماری ندارند. نتیجه باید شامل نام بیمارستان، تعداد کل بیماران و یک ستون که نشان دهنده این است که آیا بیمارستان خالی از بیمار است یا خیر.
- (ز) بیمارانی که به چندین بیمارستان مراجعه کرده‌اند (بیش از یک بیمارستان) را پیدا کرده و نام آنها و بیمارستان‌هایی که به آن مراجعه کرده‌اند (در یک ستون) را لیست کنید.

پرسش ۲ (۲۵ نمره) تحلیل تجارت الکترونیک شما در حال تحلیل پایگاه داده یک پلتفرم تجارت الکترونیک هستید تا بینش‌هایی در مورد رفتار مشتریان و عملکرد محصولات در طول سال گذشته به دست آورید. ساختار پایگاه داده شامل جداول زیر است:

Customers (CustomerID, CustomerName, Country)
 Orders (OrderID, CustomerID, OrderDate, Status)
 OrderDetails (OrderDetailID, OrderID, ProductID, Quantity, UnitPrice)
 Products (ProductID, ProductName, CategoryID)
 Categories (CategoryID, CategoryName)

تحلیل شما شامل دو بخش است:

بخش اول: دسته بندی مشتریان بر اساس ارزش سفارش
 هدف: مشتریان را بر اساس ارزش کل سفارشاتشان (مقدار * قیمت واحد) در طول سال گذشته به دسته های مختلف طبقه بندی کنید. دسته ها به شرح زیر تعریف می شوند:

- ارزش پایین: ارزش کل سفارش در ۳۳٪ پایین
- ارزش متوسط: ارزش کل سفارش در ۳۳٪ میانی
- ارزش بالا: ارزش کل سفارش در ۳۳٪ بالا

- (آ) یک کوئری SQL بنویسید تا ارزش کل سفارش برای هر مشتری در طول سال گذشته را محاسبه کند.
- (ب) یک کوئری SQL بنویسید تا مشتریان را بر اساس ارزش کل سفارش همانطور که توضیح داده شد، دسته بندی کند.

بخش دوم: تحلیل عملکرد محصول
 هدف: عملکرد دسته بندی های محصول را بر اساس حجم فروش کل جمع مقادیر فروخته شده و تعداد مشتریان منحصر به فردی که در طول سال گذشته از هر دسته خرید کرده اند، تحلیل کنید.

- (آ) یک کوئری SQL بنویسید تا حجم فروش کل برای هر دسته محصول را محاسبه کنید.
- (ب) یک کوئری SQL بنویسید تا تعداد مشتریان منحصر به فردی که محصولات از هر دسته خریده اند را بشمارید.

پرسش ۳ (۳۰ نمره) فرض کنید پایگاه داده یک مسافرخانه با جداول زیر موجود است.

hosts(host_id, host_name, host_contact)
 guests(guest_id, guest_name, guest_contact)
 rooms(room_id, room_number, room_type, room_price, owner)
 reservations(reservation_id, guest_id, room_id, check_in_date, check_out_date, reservation_date)

حال از شما خواسته شده است تا با نوشتن Query مناسب اطلاعات زیر را لیست کنید. برای بررسی صحت عملکرد کد خود از فایل 3.dump که در اختیار شما قرار داده شده است، استفاده نمایید. چنانچه کد شما بر روی دیتابیس داده شده به درستی عمل نکند، نمره ای به آن تعلق نخواهد گرفت.

- (آ) سطر مربوط به اتاق با شماره ۵۳۱ در جدول Rooms را با کوئری مناسب بیابید.
- (ب) تمام مهمان‌هایی که در این مسافرخانه حساب ساخته‌اند را لیست کنید.
- (ج) تمامی رزروهای مهمان با آیدی ۱۲۳ را لیست کنید.
- (د) تعداد رزروهای هر نوع اتاق (RoomType) را بیابید و به صورت نزولی نمایش دهید.
- (ه) آیدی و شماره اتاق‌هایی را بیابید که در بازه ۲۰۲۴-۰۳-۱۵ الی ۲۰۲۴-۰۳-۱۸ رزرو نشده‌اند.
- (و) اطلاعات ۵ کاربری که بیشترین رزرو را داشته‌اند را نمایش دهید.
- (ز) درآمدی که هر میزبان (هر میزبان می‌تواند بیش از یک اتاق داشته باشد) در ماه گذشته داشته است را بیابید، توجه کنید که این کوئری می‌بایست در هر زمانی معتبر باشد و نباید زمان فعلی در آن هاردکد شود. (مهمان در زمان check_in هزینه را پرداخت می‌کند).

پرسش ۴ (۲۵ نمره) با توجه به جداول مسابقات المپیک که در زیر آمده است به سوالات پاسخ دهید. برای بررسی صحت عملکرد کد خود از فایل 4.dump که در اختیار شما قرار داده شده است، استفاده نمایید. چنانچه کد شما بر روی دیتابیس داده شده به درستی عمل نکند، نمره ای به آن تعلق نخواهد گرفت.

sport(sport_id, sport_name)
 athlete(athlete_id, athlete_name, score, sport_id)

- (آ) با یک کوئری، نفر برتر هر رشته را مشخص کنید. توجه کنید که در این سوال در کوئری خود حق استفاده از تابع MAX و یا MIN را ندارید.

پاسخ عملی ()

پاسخ ۱ (۴۰ نمره)

(آ)

```
CREATE TABLE Patients (  
patient_id SERIAL PRIMARY KEY,  
name VARCHAR(255) NOT NULL,  
dob DATE NOT NULL,  
gender VARCHAR(10) NOT NULL,  
address TEXT NOT NULL );
```

```
CREATE TABLE Hospitals (  
hospital_id SERIAL PRIMARY KEY,  
name VARCHAR(255) NOT NULL,  
address TEXT NOT NULL,  
phone VARCHAR(20) NOT NULL );
```

```
CREATE TABLE Doctors (  
doctor_id SERIAL PRIMARY KEY,  
name VARCHAR(255) NOT NULL,  
specialization VARCHAR(255) NOT NULL,  
hospital_id INT NOT NULL REFERENCES Hospitals(hospital_id) );
```

```
CREATE TABLE MedicalRecords (  
record_id SERIAL PRIMARY KEY,  
patient_id INT NOT NULL REFERENCES Patients(patient_id),  
doctor_id INT NOT NULL REFERENCES Doctors(doctor_id),  
diagnosis TEXT,  
treatment TEXT );
```

```
CREATE TABLE Appointments (  
appointment_id SERIAL PRIMARY KEY,  
patient_id INT NOT NULL REFERENCES Patients(patient_id),  
doctor_id INT NOT NULL REFERENCES Doctors(doctor_id),  
appointment_date DATE NOT NULL,  
status VARCHAR(20) NOT NULL );
```

```
CREATE TABLE Medications (  
medication_id SERIAL PRIMARY KEY,  
name VARCHAR(255) NOT NULL,  
manufacturer VARCHAR(255) NOT NULL,  
dosage DECIMAL(10,2) NOT NULL,  
side_effects TEXT);
```

– Inserting multiple patients

```
INSERT INTO Patients (name, dob, gender, address) VALUES  
( 'John Doe', '1980-01-15', 'Male', '123 Main St, Anytown, USA'),  
( 'Jane Smith', '1975-05-20', 'Female', '456 Oak St, Othertown, USA'),  
( 'Alice Johnson', '1990-09-10', 'Female', '789 Elm St, Another town, USA');
```

– Inserting multiple doctors

```
INSERT INTO Doctors (name, specialization, hospital_id) VALUES  
( 'Dr. Meredith Grey', 'General Surgery', 1),  
( 'Dr. Alex Karev', 'Orthopedic Surgery', 2),  
( 'Dr. Miranda Bailey', 'Cardiology', 1);
```

– Inserting multiple hospitals

```
INSERT INTO Hospitals (name, address, phone) VALUES  
( 'Seattle Grace Hospital', '123 Hospital St, Seattle, WA', '555-1234'),  
( 'Pacific Northwest Hospital', '456 Hospital Ave, Portland, OR', '555-5678'),  
( 'Seattle Presbyterian Hospital', '789 Hospital Rd, Seattle, WA', '555-9101');
```

(ب)

– Inserting multiple medical records

```
INSERT INTO MedicalRecords (patient_id, doctor_id, diagnosis, treatment) VALUES  
(1, 1, 'Pneumonia', 'Antibiotics'),  
(2, 2, 'Broken Leg', 'Surgery and Physical Therapy'),  
(3, 3, 'Heart Disease', 'Medication and Lifestyle Changes');
```

– Inserting multiple appointments

```
INSERT INTO Appointments (patient_id, doctor_id, appointment_date, status) VALUES  
(1, 1, '2024-02-15', 'Completed'),  
(2, 2, '2024-02-16', 'Pending'),  
(3, 3, '2024-02-17', 'Completed');
```

– Inserting multiple medications

```
INSERT INTO Medications (name, manufacturer, dosage, side_effects) VALUES  
(‘Amoxicillin’, ‘Pfizer’, 500, ‘Nausea, Vomiting’),  
(‘Ibuprofen’, ‘Advil’, 200, ‘Stomach pain, Heartburn’),  
(‘Lisinopril’, ‘Novartis’, 10, ‘Dizziness, Cough’);
```

```
UPDATE Appointments  
SET status = ‘completed’  
WHERE appointment_date < CURDATE() AND status = ‘pending’;
```

(ج)

```
UPDATE Medications  
SET dosage = LEAST(dosage * 1.5, 1000)  
WHERE side_effects LIKE ‘%severe%’;
```

(د)

```
SELECT * FROM Medications;  
SELECT * FROM Appointments;
```

(هـ)

```
SELECT h.name AS hospital_name, COUNT(p.patient_id) AS total_patients,  
CASE  
WHEN COUNT(p.patient_id) = 0 THEN ‘Yes’  
ELSE ‘No’  
END AS has_zero_patients  
FROM Hospitals h LEFT JOIN Doctors d ON h.hospital_id = d.hospital_id LEFT JOIN Appointments a ON  
d.doctor_id = a.doctor_id LEFT JOIN Patients p ON a.patient_id = p.patient_id GROUP BY h.hospital_id;
```

(و)

```
SELECT p.name AS patient_name, GROUP_CONCAT(DISTINCT h.name) AS visited_hospitals  
FROM Patients p  
JOIN Appointments a ON p.patient_id = a.patient_id  
JOIN Doctors d ON a.doctor_id = d.doctor_id  
JOIN Hospitals h ON d.hospital_id = h.hospital_id  
GROUP BY p.patient_id  
HAVING COUNT(DISTINCT h.hospital_id) > 1;
```

(ز)

پاسخ ۲ (۲۵ نمره)
بخش اول: دسته بندی مشتریان بر اساس ارزش سفارش

```
SELECT c.CustomerID, c.CustomerName, SUM(od.Quantity * od.UnitPrice) AS TotalOrderValue  
FROM Customers c  
JOIN Orders o ON c.CustomerID = o.CustomerID  
JOIN OrderDetails od ON o.OrderID = od.OrderID WHERE  
o.OrderDate >= DATEADD(year, -1, GETDATE())  
GROUP BY c.CustomerID, c.CustomerName;
```

(آ)

```
SELECT  
C.CustomerName,  
CASE  
WHEN (  
SELECT SUM(OD2.Quantity * OD2.UnitPrice)  
FROM Orders O2
```

(ب)

```

JOIN OrderDetails OD2 ON O2.OrderID = OD2.OrderID
WHERE O2.CustomerID = C.CustomerID
AND O2.OrderDate > CURRENT_DATE - INTERVAL '1 year'
AND O2.OrderDate < CURRENT_DATE
) < 0.33 * (
SELECT SUM(OD3.Quantity * OD3.UnitPrice)
FROM Orders O3
JOIN OrderDetails OD3 ON O3.OrderID = OD3.OrderID
WHERE O3.CustomerID = C.CustomerID
AND O3.OrderDate > CURRENT_DATE - INTERVAL '1 year'
AND O3.OrderDate < CURRENT_DATE
) THEN 'HIGH'
WHEN (
SELECT SUM(OD2.Quantity * OD2.UnitPrice)
FROM Orders O2
JOIN OrderDetails OD2 ON O2.OrderID = OD2.OrderID
WHERE O2.CustomerID = C.CustomerID
AND O2.OrderDate > CURRENT_DATE - INTERVAL '1 year'
AND O2.OrderDate < CURRENT_DATE
) < 0.66 * (
SELECT SUM(OD3.Quantity * OD3.UnitPrice)
FROM Orders O3
JOIN OrderDetails OD3 ON O3.OrderID = OD3.OrderID
WHERE O3.CustomerID = C.CustomerID
AND O3.OrderDate > CURRENT_DATE - INTERVAL '1 year'
AND O3.OrderDate < CURRENT_DATE
) THEN 'MEDIUM'
ELSE 'LOW'
END AS customerValue
FROM
Customers C
JOIN
Orders O ON C.CustomerID = O.CustomerID
JOIN
OrderDetails OD ON O.OrderID = OD.OrderID
WHERE
O.OrderDate > CURRENT_DATE - INTERVAL '1 year'
AND O.OrderDate < CURRENT_DATE
GROUP BY
C.CustomerName;

```

بخش دوم: تحلیل عملکرد محصول

```

SELECT cat.CategoryName, SUM(od.Quantity) AS TotalSalesVolume
FROM Categories cat JOIN Products p ON cat.CategoryID = p.CategoryID
JOIN OrderDetails od ON p.ProductID = od.ProductID
JOIN Orders o ON od.OrderID = o.OrderID
WHERE o.OrderDate >= DATEADD(year, -1, GETDATE())
GROUP BY cat.CategoryName;

```

(آ)

```

SELECT cat.CategoryName, COUNT(DISTINCT o.CustomerID) AS UniqueCustomers
FROM Categories cat
JOIN Products p ON cat.CategoryID = p.CategoryID
JOIN OrderDetails od ON p.ProductID = od.ProductID
JOIN Orders o ON od.OrderID = o.OrderID
WHERE o.OrderDate >= DATEADD(year, -1, GETDATE())
GROUP BY cat.CategoryName;

```

(ب)

پاسخ ۳ (۳۰ نمره)

```
SELECT * FROM rooms WHERE room_number = '531';
```

(آ)

```
SELECT * FROM guests;
```

(ب)

SELECT * FROM reservations WHERE guest_id = 123; (ج)

SELECT room_type, COUNT(*) AS ReservationCount (د)
FROM rooms
JOIN reservations ON rooms.room_id = reservations.room_id
GROUP BY room_type
ORDER BY ReservationCount DESC;

SELECT * (ه)
FROM rooms
WHERE room_id NOT IN (
SELECT room_id
FROM reservations
WHERE (check_in_date, check_out_date) OVERLAPS ('2024-03-15'::DATE, '2024-03-18'::DATE)
);

SELECT g.guest_id, g.guest_name, COUNT(r.reservation_id) AS reservation_count (و)
FROM guests g
JOIN reservations r ON g.guest_id = r.guest_id
GROUP BY g.guest_id, g.guest_name
ORDER BY reservation_count DESC
LIMIT 5;

SELECT (ز)
h.host_id,
h.host_name,
COALESCE(SUM(ro.room_price), 0) AS total_income
FROM
hosts h
LEFT JOIN
rooms ro ON h.host_id = ro.owner
LEFT JOIN
reservations r ON ro.room_id = r.room_id
AND r.check_in_date >= CURRENT_DATE - INTERVAL '1 month'
AND r.check_in_date < CURRENT_DATE
GROUP BY
h.host_id, h.host_name
ORDER BY
total_income DESC;

پاسخ ۴ (۲۵ نمره)

SELECT (آ)
a.athlete_name,
a.score,
s.sport_id
FROM
athlete a
JOIN
sport s ON a.sport_id = s.sport_id
WHERE
NOT EXISTS (
SELECT 1
FROM
athlete a2
WHERE
a2.sport_id = a.sport_id
AND a2.score > a.score
);

SELECT a.athlete_name, a.score, s.sport_name (ب)
FROM athlete a


```
JOIN sports s ON a.sport_id = s.sport_id
WHERE a.score < (
SELECT AVG(ae.score)
FROM athlete ae
WHERE a.sport_id = ae.sport_id
)
AND a.score > (
SELECT AVG(score)
FROM athlete
);
```