7.Bölüm Cevaplar

İnceleme Soruları ve Cevapları:

 **Tarih verilerini saklamak için karakter veri türü yerine DATE veri türünü kullanmanın neden tercih edileceğini açıklayın.**

**Cevap:** DATE veri türü, tarihsel veriler için optimize edilmiştir. Tarihlerle aritmetik işlemler yapmayı, sıralamayı ve karşılaştırmayı kolaylaştırır. Karakter veri türü ile saklanan tarih bilgileri üzerinde doğrudan matematiksel işlemler yapılamaz ve formatlama hatalarına açıktır.

 **Aşağıdaki komutun neden bir hata oluşturduğunu ve düzeltmek için ne gibi değişiklikler yapılabileceğini açıklayın:**

SEÇİNİZ V\_CODE, SUMP\_QOH ÜRÜNDEN;

**Cevap:**

 Hata nedeni: GROUP BY ifadesinin eksik olmasıdır. SUM gibi toplama fonksiyonları kullanılırken, gruplanmamış sütunlar için GROUP BY kullanmak gerekir.

 Düzeltme:

SELECT V\_CODE, SUMP\_QOH FROM ÜRÜN

GROUP BY V\_CODE;

 **Çapraz birleştirme CROSS JOIN nedir? Sözdizimine bir örnek verin.**

**Cevap:** Çapraz birleştirme, iki tabloyu koşulsuz olarak birleştirerek her satırı diğer tablodaki tüm satırlarla eşleştirir.

**Örnek:**

SELECT \* FROM T1

CROSS JOIN T2;

 **Dış birleştirme sınıflandırmasına hangi üç birleştirme türü dahildir?**

**Cevap:**

 LEFT OUTER JOIN

 RIGHT OUTER JOIN  FULL OUTER JOIN

 **Üç dış birleştirme türü için birer sorgu yazın.**

 LEFT OUTER JOIN SELECT \* FROM T1

LEFT JOIN T2 ON T1.C1  T2.C1;

 RIGHT OUTER JOIN SELECT \* FROM T1

RIGHT JOIN T2 ON T1.C1  T2.C1;

 FULL OUTER JOIN SELECT \* FROM T1

FULL JOIN T2 ON T1.C1  T2.C1;

 **Özyinelemeli Recursive) nedir?**

**Cevap:** Özyinelemeli, bir işlemin kendisini çağırarak çalışmasını ifade eder. SQLʼde özyinelemeli sorgular genellikle hiyerarşik veri yapıları (örneğin, çalışanların yöneticileriyle ilişkilendirilmesi) için kullanılır.

 **WHERE cümlesini IN operatörü olmadan yazın.**

WHERE V\_STATE  'TN' OR V\_STATE  'FL' OR V\_STATE  'GA'

 **ORDER BY ve GROUP BY arasındaki fark?**

**Cevap:**

 **ORDER BY** Sonuç kümesini belirtilen sütunlara göre sıralar.

 **GROUP BY** Verileri belirli sütunlara göre gruplar ve her grup için toplama işlemleri yapılmasını sağlar.

 **Aşağıdaki iki sorgu neden farklı sonuçlar üretir?**

SELECT DISTINCT COUNTV\_CODE FROM PRODUCT; SELECT COUNTDISTINCT V\_CODE FROM PRODUCT;

**Cevap:**

 İlk sorgu, COUNTV\_CODE değerini hesapladıktan sonra DISTINCT uygular.

 İkinci sorgu, V\_CODE sütunundaki benzersiz değerleri sayar.

 **COUNT ve SUM işlevleri arasındaki fark?**

**Cevap:**

 **COUNT** Satır sayısını döndürür.

 **SUM** Sayısal sütunların toplamını döndürür.

 **WHERE ve HAVING arasındaki fark?**

**Cevap:**

 **WHERE** Satırları filtreler, gruplama öncesi kullanılır.

 **HAVING** Gruplanmış sonuçları filtreler.

 **Alt sorgu nedir?**

**Cevap:** Ana sorgu içinde kullanılan, sonuç döndüren iç sorgudur.

 **Alt sorguların döndürebileceği üç sonuç türü?**

**Cevap:**

 Tek değer Scalar)

 Tek sütunlu liste Column)

 Çok satır ve sütun içeren sonuç kümesi Table)

 **İlişkili alt sorgu nedir?**

**Cevap:** Dış sorgudaki her satır için çalıştırılan alt sorgudur.

**Örnek:**

SELECT \* FROM EMPLOYEE E

WHERE SALARY  SELECT AVGSALARY FROM EMPLOYEE WHERE DEP T\_ID  E.DEPT\_ID;

 **Normal alt sorgu ile ilişkili alt sorgu arasındaki fark?**

**Cevap:** Normal alt sorgular bağımsızdır, ilişkili alt sorgular dış sorgunun her satırına bağlıdır.

 **SQL operatörleri küme yönelimli midir?**

**Cevap:** Evet, çünkü SQL, birden fazla satır ve sütunu aynı anda işler.

 **Union uyumluluk (union compatible) nedir?**

**Cevap:** UNION, INTERSECT, EXCEPT işlemlerinin çalışabilmesi için sütun sayısı ve veri türleri aynı olmalıdır.

 **UNION ve UNION ALL arasındaki fark?**

**Cevap:**

 **UNION** Tekrar eden satırları kaldırır.

 **UNION ALL** Tüm satırları gösterir.**Örnek:**

SELECT \* FROM EMPLOYEE UNION

SELECT \* FROM EMPLOYEE\_1;

 **UNION çıktısı?**

**Cevap:** Çalışanların birleşik kümesi, tekrar eden kayıtlar kaldırılmış şekilde listelenir.

 **UNION ALL çıktısı?**

**Cevap:** Tüm çalışanlar listelenir, tekrar eden kayıtlar dahil edilir.

 **INTERSECT çıktısı?**

**Cevap:** Ortak çalışanlar listelenir.

 **EXCEPT MINUS çıktısı?**

**Cevap:** EMPLOYEEʼde olup EMPLOYEE\_1ʼde olmayan çalışanlar listelenir.

 **Verilen tablolar için UNION, UNION ALL, INTERSECT, EXCEPT çıktıları nedir?**

**Cevap:**

 **UNION** Her iki tablodaki benzersiz VEND\_CODEʼler

 **UNION ALL** Tüm VEND\_CODEʼler

 **INTERSECT** Ortak VEND\_CODEʼler

 **EXCEPT** PRODUCT tablosunda olup VENDORʼda olmayan VEND\_CODEʼler

 **EXCEPT MINUS sırası neden önemlidir?**

**Cevap:** İlk tablodaki kayıtlar ikinci tablodakilerden çıkarılır. UNIONʼda sıralama fark etmez.

 **MS Access ve SQL Server'da gün farkını hesaplayan fonksiyon?**

DATEDIFF(day, DoğumTarihi, GETDATE

 **Oracle'da gün farkını hesaplayan fonksiyon?**

SELECT SYSDATE  DoğumTarihi FROM DUAL;

 **İlk üç karakteri getiren fonksiyon?**

 **Oracle:**

**SQL Server:**

SELECT SUBSTREMP\_LNAME, 1, 3 FROM EMPLOYEE;

SELECT LEFTEMP\_LNAME, 3 FROM EMPLOYEE;

 **Bir SQL programcısı SELECT sorgusu oluşturmadan önce hangi iki şeyi anlamalıdır?**

**Cevap:**

 Veritabanı şeması (tablolar ve ilişkiler)  İstenen çıktının yapısı ve amacı

Problem Soruları ve Çözümleri:

1. **Soyadı Smith ile başlayan tüm çalışanları listeleyin.**

SELECT EMP\_NUM, EMP\_LNAME, EMP\_FNAME, EMP\_MNAME FROM EMPLOYEE

WHERE EMP\_LNAME LIKE 'Smith%' ORDER BY EMP\_NUM;

1. **EMPLOYEE ve PROJECT tablolarını birleştirerek listeleyin.**

SELECT E.EMP\_NUM, E.EMP\_LNAME, E.EMP\_FNAME, P.PROJ\_NAME FROM EMPLOYEE E

JOIN ASSIGNMENT A ON E.EMP\_NUM  A.EMP\_NUM JOIN PROJECT P ON A.PROJ\_NUM  P.PROJ\_NUM ORDER BY P.PROJ\_NAME;

1. **Problem 2ʼdeki sorguyu çalışanın soyadına göre sıralayın.**

SELECT E.EMP\_NUM, E.EMP\_LNAME, E.EMP\_FNAME, P.PROJ\_NAME FROM EMPLOYEE E

JOIN ASSIGNMENT A ON E.EMP\_NUM  A.EMP\_NUM JOIN PROJECT P ON A.PROJ\_NUM  P.PROJ\_NUM ORDER BY E.EMP\_LNAME;

1. **ASSIGNMENT tablosunda farklı proje numaralarını listeleyin.**

SELECT DISTINCT PROJ\_NUM FROM ASSIGNMENT

ORDER BY PROJ\_NUM;

1. **ASSIGNMENT tablosundaki ücretleri doğrulayan bir sorgu yazın.**

SELECT ASSIGN\_NUM, EMP\_NUM, PROJ\_NUM,

ASSIGN\_CHARGE, ASSIGN\_CHG\_HR \* ASSIGN\_HOURS AS CALCULATE D\_CHARGE

FROM ASSIGNMENT ORDER BY ASSIGN\_NUM;

1. **Çalışılan toplam saatleri ve ücretleri listeleyin.**

SELECT EMP\_NUM, SUMASSIGN\_HOURS AS TOTAL\_HOURS, SUMASSIGN\_CHG\_HR \* ASSIGN\_HOURS AS TOTAL\_CHARGE

FROM ASSIGNMENT GROUP BY EMP\_NUM ORDER BY EMP\_NUM;

1. **Proje bazında toplam saat ve ücretleri listeleyin.**

SELECT PROJ\_NUM, SUMASSIGN\_HOURS AS TOTAL\_HOURS, SUMASSIGN\_CHG\_HR \* ASSIGN\_HOURS AS TOTAL\_CHARGE

FROM ASSIGNMENT GROUP BY PROJ\_NUM ORDER BY PROJ\_NUM;

1. **Tüm çalışanlar için toplam saat ve masrafları hesaplayın.**

SELECT SUMASSIGN\_HOURS AS TOTAL\_HOURS, SUMASSIGN\_CHG\_HR \* ASSIGN\_HOURS AS TOTAL\_COST

FROM ASSIGNMENT;

1. **Fatura sayısını sayın.**

SELECT COUNT\* AS INVOICE\_COUNT FROM INVOICE;

1. **Bakiyesi 500$'dan fazla olan müşteri sayısını bulun.**

SELECT COUNT\* AS HIGH\_BALANCE\_CUSTOMERS FROM CUSTOMER

WHERE BALANCE  500;

1. **Müşterilerin yaptığı tüm satın alımları listeleyin.**

SELECT C.CUST\_CODE, I.INV\_NUM, P.PROD\_DESCRIPT FROM CUSTOMER C

JOIN INVOICE I ON C.CUST\_CODE  I.CUST\_CODE JOIN LINE L ON I.INV\_NUM  L.INV\_NUM

JOIN PRODUCT P ON L.PROD\_SKU  P.PROD\_SKU

ORDER BY C.CUST\_CODE, I.INV\_NUM, P.PROD\_DESCRIPT;

1. **Müşteri satın alımları için alt toplamları hesaplayın.**

SELECT C.CUST\_CODE, I.INV\_NUM, P.PROD\_DESCRIPT,

L.LINE\_UNITS \* L.LINE\_PRICE AS SUBTOTAL FROM CUSTOMER C

JOIN INVOICE I ON C.CUST\_CODE  I.CUST\_CODE JOIN LINE L ON I.INV\_NUM  L.INV\_NUM

JOIN PRODUCT P ON L.PROD\_SKU  P.PROD\_SKU

ORDER BY C.CUST\_CODE, I.INV\_NUM, P.PROD\_DESCRIPT;

1. **Her müşteri için toplam satın alma değerini bulun.**

SELECT C.CUST\_CODE, C.BALANCE,

SUML.LINE\_UNITS \* L.LINE\_PRICE AS TOTAL\_PURCHASE FROM CUSTOMER C

JOIN INVOICE I ON C.CUST\_CODE  I.CUST\_CODE JOIN LINE L ON I.INV\_NUM  L.INV\_NUM

GROUP BY C.CUST\_CODE, C.BALANCE ORDER BY C.CUST\_CODE;

1. **Müşteri satın alımlarının sayısını listeleyin.**

SELECT C.CUST\_CODE, COUNTL.LINE\_UNITS AS TOTAL\_PURCHASES FROM CUSTOMER C

JOIN INVOICE I ON C.CUST\_CODE  I.CUST\_CODE JOIN LINE L ON I.INV\_NUM  L.INV\_NUM

GROUP BY C.CUST\_CODE ORDER BY C.CUST\_CODE;

1. **Müşterilerin ortalama satın alma tutarlarını bulun.**

SELECT C.CUST\_CODE, SUML.LINE\_UNITS \* L.LINE\_PRICE AS TOTAL\_PURC HASE,

COUNTL.LINE\_UNITS AS TOTAL\_ORDERS, AVGL.LINE\_UNITS \* L.LINE\_PRICE AS AVG\_PURCHASE

FROM CUSTOMER C

JOIN INVOICE I ON C.CUST\_CODE  I.CUST\_CODE JOIN LINE L ON I.INV\_NUM  L.INV\_NUM

GROUP BY C.CUST\_CODE ORDER BY C.CUST\_CODE;

1. **Her fatura için toplam satın alma değerini hesaplayın.**

SELECT I.INV\_NUM, SUML.LINE\_UNITS \* L.LINE\_PRICE AS TOTAL\_INVOICE\_ AMOUNT

FROM INVOICE I

JOIN LINE L ON I.INV\_NUM  L.INV\_NUM

GROUP BY I.INV\_NUM ORDER BY I.INV\_NUM;

1. **Müşteriye göre fatura sayısı ve toplam satın alma tutarlarını üretin.**

SELECT C.CUST\_CODE, COUNTI.INV\_NUM AS TOTAL\_INVOICES, SUML.LINE\_UNITS \* L.LINE\_PRICE AS TOTAL\_PURCHASE

FROM CUSTOMER C

JOIN INVOICE I ON C.CUST\_CODE  I.CUST\_CODE JOIN LINE L ON I.INV\_NUM  L.INV\_NUM

GROUP BY C.CUST\_CODE ORDER BY C.CUST\_CODE;

1. **Tüm faturaların toplamı, en küçük, en büyük ve ortalama satışları hesaplayın.**

SELECT COUNTI.INV\_NUM AS TOTAL\_INVOICES, SUML.LINE\_UNITS \* L.LINE\_PRICE AS TOTAL\_SALES, MINL.LINE\_UNITS \* L.LINE\_PRICE AS MIN\_PURCHASE, MAXL.LINE\_UNITS \* L.LINE\_PRICE AS MAX\_PURCHASE, AVGL.LINE\_UNITS \* L.LINE\_PRICE AS AVG\_PURCHASE

FROM INVOICE I

JOIN LINE L ON I.INV\_NUM  L.INV\_NUM;

1. **Faturalama döneminde satın alma yapmış müşterilerin bakiyelerini listeleyin.**

SELECTDISTINCTC.CUST\_CODE,C.BALANCE FROM CUSTOMER C

JOIN INVOICE I ON C.CUST\_CODE  I.CUST\_CODE ORDER BY C.CUST\_CODE;

1. **Satın alma yapmayan müşterileri listeleyin.**

SELECT C.CUST\_CODE, C.BALANCE FROM CUSTOMER C

LEFT JOIN INVOICE I ON C.CUST\_CODE  I.CUST\_CODE WHERE I.INV\_NUM IS NULL

ORDER BY C.CUST\_CODE;

1. **Envanterdeki tüm ürünlerin toplam değerini hesaplayın.**

SELECT SUMPROD\_QOH \* PROD\_PRICE AS TOTAL\_INVENTORY\_VALUE FROM PRODUCT;

1. **Departman 300ʼde çalışan veya "CLERK I" unvanına sahip çalışanları listeleyin.**

SELECT EMP\_NUM, EMP\_FNAME, EMP\_LNAME, EMP\_PHONE, EMP\_TITLE, DE PT\_NUM

FROM EMPLOYEE

WHERE DEPT\_NUM  300 OR EMP\_TITLE  'CLERK I' ORDER BY EMP\_LNAME, EMP\_FNAME;

1. **Belirli çalışanların maaş geçmişini görüntüleyin.**

SELECT EMP\_NUM, EMP\_LNAME, SAL\_FROM, SAL\_END, SAL\_AMOUNT FROM EMPLOYEE E

JOIN SALARY\_HISTORY S ON E.EMP\_NUM  S.EMP\_NUM WHERE E.EMP\_NUM IN 83731, 83745, 84039

ORDER BY E.EMP\_NUM, SAL\_FROM;

1. **Satıcıların sattığı marka ve ürün sayısını görüntüleyin.**

SELECT V.VEND\_ID, V.VEND\_NAME, B.BRAND\_NAME, COUNTP.PROD\_SKU AS TOTAL\_PRODUCTS

FROM VENDOR V

JOIN SUPPLIES S ON V.VEND\_ID  S.VEND\_ID JOIN PRODUCT P ON S.PROD\_SKU  P.PROD\_SKU JOIN BRAND B ON P.BRAND\_ID  B.BRAND\_ID

GROUP BY V.VEND\_ID, V.VEND\_NAME, B.BRAND\_NAME ORDER BY V.VEND\_NAME, B.BRAND\_NAME;

1. **En pahalı ürünü ve fiyatını görüntüleyin.**

SELECT PROD\_SKU, PROD\_DESCRIPT, PROD\_PRICE FROM PRODUCT

ORDER BY PROD\_PRICE DESC LIMIT 1;

1. **Bir müşteri için toplam satın alım sayısını listeleyin.**

SELECT C.CUST\_CODE, COUNTDISTINCT I.INV\_NUM AS TOTAL\_ORDERS FROM CUSTOMER C

JOIN INVOICE I ON C.CUST\_CODE  I.CUST\_CODE GROUP BY C.CUST\_CODE

ORDER BY C.CUST\_CODE;

1. **Her markanın ürünlerinin ortalama fiyatını listeleyin.**

SELECT B.BRAND\_NAME, ROUNDAVGP.PROD\_PRICE, 2 AS AVG\_PRICE FROM BRAND B

JOIN PRODUCT P ON B.BRAND\_ID  P.BRAND\_ID GROUP BY B.BRAND\_NAME

ORDER BY B.BRAND\_NAME;

1. **Her departmandaki çalışan sayısını görüntüleyin.**

SELECT DEPT\_NUM, COUNTEMP\_NUM AS TOTAL\_EMPLOYEES FROM EMPLOYEE

GROUP BY DEPT\_NUM ORDER BY DEPT\_NUM;

29. 1 Mayıs 2011 ile 31 Aralık 2012 arasında işe alınan çalışanları listeleyin.

SELECT EMP\_FNAME, EMP\_LNAME, EMP\_EMAIL FROM EMPLOYEE

WHERE EMP\_HIREDATE BETWEEN '20110501' AND '20121231' ORDER BY EMP\_LNAME, EMP\_FNAME;

1. **Bir departmanda çalışan en yüksek maaşlı çalışanları listeleyin.**

SELECT EMP\_NUM, EMP\_FNAME, EMP\_LNAME, MAXSAL\_AMOUNT AS HIG HEST\_SALARY

FROM EMPLOYEE E

JOIN SALARY\_HISTORY S ON E.EMP\_NUM  S.EMP\_NUM WHERE DEPT\_NUM  200

GROUP BY EMP\_NUM, EMP\_FNAME, EMP\_LNAME ORDER BY HIGHEST\_SALARY DESC;

1. **Bir kitap ödünç alan farklı kullanıcıların sayısını görüntüleyin.**

SELECT COUNTDISTINCT PATRON\_ID AS TOTAL\_BORROWERS FROM CHECKOUT;

1. **Fiyatı premium markalardan daha pahalı olan premium olmayan ürünleri listeleyin.**

SELECT P.PROD\_SKU, P.PROD\_DESCRIPT, P.PROD\_PRICE, B.BRAND\_TYPE FROM PRODUCT P

JOIN BRAND B ON P.BRAND\_ID  B.BRAND\_ID WHERE B.BRAND\_TYPE ! 'Premium'

AND P.PROD\_PRICE  SELECT MAXPROD\_PRICE FROM PRODUCT WHERE BRAND\_ID IN

SELECT BRAND\_ID FROM BRAND WHERE BRAND\_TYPE  'Premi

um'))

ORDER BY P.PROD\_PRICE DESC;

1. **İş unvanı "ASSOCIATE" ile biten çalışanları listeleyin.**

SELECT EMP\_NUM, EMP\_LNAME, EMP\_EMAIL, EMP\_TITLE, DEPT\_NAME FROM EMPLOYEE E

JOIN DEPARTMENT D ON E.DEPT\_NUM  D.DEPT\_NUM WHERE EMP\_TITLE LIKE '%ASSOCIATE'

ORDER BY DEPT\_NAME, EMP\_TITLE;

1. **Her markanın veritabanında bulunan ürün sayısını listeleyin.**

SELECT B.BRAND\_NAME, COUNTP.PROD\_SKU AS PRODUCT\_COUNT FROM BRAND B

JOIN PRODUCT P ON B.BRAND\_ID  P.BRAND\_ID GROUP BY B.BRAND\_NAME

ORDER BY B.BRAND\_NAME;

1. **Her kategorideki su bazlı ürünlerin sayısını görüntüleyin.**

SELECT PROD\_CATEGORY, COUNTPROD\_SKU AS PRODUCT\_COUNT FROM PRODUCT

WHERE PROD\_BASE  'Su' GROUP BY PROD\_CATEGORY ORDER BY PROD\_CATEGORY;

1. **Her baz ve tür kombinasyonundaki ürün sayılarını listeleyin.**

SELECT PROD\_BASE, PROD\_TYPE, COUNTPROD\_SKU AS PRODUCT\_COUN T

FROM PRODUCT

GROUP BY PROD\_BASE, PROD\_TYPE ORDER BY PROD\_BASE, PROD\_TYPE;

1. **Her markaya ait toplam envanteri görüntüleyin.**

SELECT B.BRAND\_NAME, SUMP.PROD\_QOH AS TOTAL\_INVENTORY FROM BRAND B

JOIN PRODUCT P ON B.BRAND\_ID  P.BRAND\_ID GROUP BY B.BRAND\_NAME

ORDER BY TOTAL\_INVENTORY DESC;

1. **Her markanın ürünlerinin ortalama fiyatını listeleyin.**

SELECT B.BRAND\_NAME, ROUNDAVGP.PROD\_PRICE, 2 AS AVG\_PRICE FROM BRAND B

JOIN PRODUCT P ON B.BRAND\_ID  P.BRAND\_ID GROUP BY B.BRAND\_NAME

ORDER BY B.BRAND\_NAME;

1. **Her departmandaki en son işe alınan kişiyi görüntüleyin.**

SELECT DEPT\_NUM, MAXEMP\_HIREDATE AS LAST\_HIRED FROM EMPLOYEE

GROUP BY DEPT\_NUM ORDER BY DEPT\_NUM;

1. **Departman 200ʼde çalışanların en yüksek maaşlarını listeleyin.**

SELECT EMP\_NUM, EMP\_FNAME, EMP\_LNAME, MAXSAL\_AMOUNT AS HIG HEST\_SALARY

FROM EMPLOYEE E

JOIN SALARY\_HISTORY S ON E.EMP\_NUM  S.EMP\_NUM

WHERE DEPT\_NUM  200

GROUP BY EMP\_NUM, EMP\_FNAME, EMP\_LNAME ORDER BY HIGHEST\_SALARY DESC;

1. **Kümülatif fatura toplamı 1.500$'dan büyük olan müşterileri listeleyin.**

SELECT C.CUST\_CODE, C.CUST\_FNAME, C.CUST\_LNAME, SUML.LINE\_UNITS \* L.LINE\_PRICE AS TOTAL\_SPENT

FROM CUSTOMER C

JOIN INVOICE I ON C.CUST\_CODE  I.CUST\_CODE JOIN LINE L ON I.INV\_NUM  L.INV\_NUM

GROUP BY C.CUST\_CODE, C.CUST\_FNAME, C.CUST\_LNAME HAVING TOTAL\_SPENT  1500

ORDER BY TOTAL\_SPENT DESC;

1. **Her departman yöneticisinin bilgilerini listeleyin.**

SELECT D.DEPT\_NUM, D.DEPT\_NAME, D.DEPT\_PHONE, E.EMP\_NUM, E.EMP\_LNAME AS MANAGER\_LNAME

FROM DEPARTMENT D

JOIN EMPLOYEE E ON D.MANAGER\_ID  E.EMP\_NUM ORDER BY D.DEPT\_NAME;

1. **Satıcıların tedarik ettiği marka ve ürün sayılarını listeleyin.**

SELECT V.VEND\_ID, V.VEND\_NAME, B.BRAND\_NAME, COUNTP.PROD\_SKU AS PRODUCT\_COUNT

FROM VENDOR V

JOIN SUPPLIES S ON V.VEND\_ID  S.VEND\_ID JOIN PRODUCT P ON S.PROD\_SKU  P.PROD\_SKU JOIN BRAND B ON P.BRAND\_ID  B.BRAND\_ID

GROUP BY V.VEND\_ID, V.VEND\_NAME, B.BRAND\_NAME ORDER BY V.VEND\_NAME, B.BRAND\_NAME;

1. **Her çalışan tarafından tamamlanan faturaların toplam değerini listeleyin.**

SELECT E.EMP\_NUM, E.EMP\_LNAME, E.EMP\_FNAME, SUMI.INV\_TOTAL AS TOTAL\_SALES

FROM EMPLOYEE E

JOIN INVOICE I ON E.EMP\_NUM  I.EMPLOYEE\_ID GROUP BY E.EMP\_NUM, E.EMP\_LNAME, E.EMP\_FNAME ORDER BY TOTAL\_SALES DESC;

1. **En büyük ortalama ürün fiyatına sahip markayı bulun.**

SELECT B.BRAND\_NAME, ROUNDAVGP.PROD\_PRICE, 2 AS AVG\_PRICE FROM BRAND B

JOIN PRODUCT P ON B.BRAND\_ID  P.BRAND\_ID GROUP BY B.BRAND\_NAME

ORDER BY AVG\_PRICE DESC LIMIT 1;

1. **En yüksek ortalama fiyatlı markayı listeleyin.**

SELECT B.BRAND\_ID, B.BRAND\_NAME, B.BRAND\_TYPE, ROUNDAVGP.PROD

\_PRICE, 2 AS AVG\_PRICE FROM BRAND B

JOIN PRODUCT P ON B.BRAND\_ID  P.BRAND\_ID

GROUP BY B.BRAND\_ID, B.BRAND\_NAME, B.BRAND\_TYPE ORDER BY AVG\_PRICE DESC

LIMIT 1;

1. **Soyadı "Hagan" olan müşteriye 18 Mayıs 2021'de satış yapan çalışanı ve yöneticisini listeleyin.**

SELECT MGR.EMP\_FNAME AS MANAGER\_FNAME, MGR.EMP\_LNAME AS MA NAGER\_LNAME,

D.DEPT\_NAME, D.DEPT\_PHONE, SLS.EMP\_FNAME AS SALES\_FNAME, SLS.EMP\_LNAME AS SALES\_LNAME, C.CUST\_FNAME, C.CUST\_LNAME, I.INV\_DATE, I.INV\_TOTAL

FROM CUSTOMER C

JOIN INVOICE I ON C.CUST\_CODE  I.CUST\_CODE

JOIN EMPLOYEE SLS ON I.EMPLOYEE\_ID  SLS.EMP\_NUM JOIN DEPARTMENT D ON SLS.DEPT\_NUM  D.DEPT\_NUM JOIN EMPLOYEE MGR ON D.MANAGER\_ID  MGR.EMP\_NUM

WHERE C.CUST\_LNAME  'Hagan' AND I.INV\_DATE  '20210518';

1. **Departman 300ʼdeki çalışanların geçerli maaşlarını listeleyin.**

SELECT E.EMP\_NUM, E.EMP\_FNAME, E.EMP\_LNAME, S.SAL\_AMOUNT FROM EMPLOYEE E

JOIN SALARY\_HISTORY S ON E.EMP\_NUM  S.EMP\_NUM WHERE E.DEPT\_NUM  300 AND S.SAL\_END IS NULL ORDER BY S.SAL\_AMOUNT DESC;

1. **Her çalışan için başlangıç maaşını listeleyin.**

SELECT EMP\_NUM, EMP\_FNAME, EMP\_LNAME, MINSAL\_AMOUNT AS STA RTING\_SALARY

FROM SALARY\_HISTORY

GROUP BY EMP\_NUM, EMP\_FNAME, EMP\_LNAME ORDER BY EMP\_NUM;

1. **Aynı faturada aynı markanın sızdırmazlık ve son kat ürünleri**

SELECT L1.Inv\_Num, L1.Line\_Num AS Sealer\_Line, L2.Line\_Num AS Topcoat\_Li ne,

L1.Prod\_SKU AS Sealer\_SKU, L2.Prod\_SKU AS Topcoat\_SKU,

L1.Prod\_Descript AS Sealer\_Desc, L2.Prod\_Descript AS Topcoat\_Desc, L1.Brand\_ID

FROM LGLINE L1 JOIN LGLINE L2

ON L1.Inv\_Num  L2.Inv\_Num AND L1.Brand\_ID  L2.Brand\_ID WHERE L1.Prod\_Descript LIKE '%Sealer%'

AND L2.Prod\_Descript LIKE '%Topcoat%'

ORDER BY L1.Inv\_Num ASC, L1.Line\_Num ASC, L2.Line\_Num DESC;

1. **Binder Prime markalı ürünleri en çok satan çalışan**

SELECT E.Emp\_Num, E.Emp\_Fname, E.Emp\_Lname, E.Emp\_Email, SUML.Line\_Qty) AS Total\_Units\_Sold

FROM LGEMPLOYEE E

JOIN LGINVOICE I ON E.Emp\_Num  I.Employee\_ID JOIN LGLINE L ON I.Inv\_Num  L.Inv\_Num

JOIN LGPRODUCT P ON L.Prod\_SKU  P.Prod\_SKU

WHERE P.Brand\_ID  SELECT Brand\_ID FROM LGBRAND WHERE Brand\_Nam e = 'Binder Prime')

AND I.Inv\_Date BETWEEN '20211101' AND '20211205'

GROUP BY E.Emp\_Num, E.Emp\_Fname, E.Emp\_Lname, E.Emp\_Email ORDER BY Total\_Units\_Sold DESC, E.Emp\_Lname ASC;

1. **83649 ve 83677 çalışanları tarafından tamamlanan faturalarla müşteriler**

SELECT DISTINCT C.Cust\_Code, C.Cust\_Fname, C.Cust\_Lname FROM LGCUSTOMER C

JOIN LGINVOICE I1 ON C.Cust\_Code  I1.Cust\_Code AND I1.Employee\_ID  83 649

JOIN LGINVOICE I2 ON C.Cust\_Code  I2.Cust\_Code AND I2.Employee\_ID  8 3677

ORDER BY C.Cust\_Lname ASC, C.Cust\_Fname ASC;

1. **Alabama'daki müşterilerin en büyük satın alımları**

SELECT C.Cust\_Code, C.Cust\_Fname, C.Cust\_Lname, C.Cust\_Street, C.Cust\_City, C.Cust\_State, I.Inv\_Date, COALESCEI.Inv\_Total, 0 AS Inv\_Total

FROM LGCUSTOMER C

LEFT JOIN LGINVOICE I ON C.Cust\_Code  I.Cust\_Code AND C.Cust\_State = 'AL'

ORDER BY C.Cust\_Lname ASC, C.Cust\_Fname ASC;

1. **Markalara göre ürünlerin ortalama fiyatı ve toplam satılan birimler**

SELECT B.Brand\_Name, B.Brand\_Type, ROUNDAVGP.Prod\_Price), 2) AS Avg\_Price, SUML.Line\_Qty) AS Total\_Units\_Sold

FROM LGBRAND B

JOIN LGPRODUCT P ON B.Brand\_ID  P.Brand\_ID JOIN LGLINE L ON P.Prod\_SKU  L.Prod\_SKU

GROUP BY B.Brand\_Name, B.Brand\_Type ORDER BY B.Brand\_Name ASC;

1. **Premium olmayan ama premium ürünlerden daha pahalı ürünler**

SELECT B.Brand\_Name, B.Brand\_Type, P.Prod\_SKU, P.Prod\_Descript, P.Prod\_Price

FROM LGPRODUCT P

JOIN LGBRAND B ON P.Brand\_ID  B.Brand\_ID WHERE B.Brand\_Type ! 'Premium'

AND P.Prod\_Price > SELECT MAXProd\_Price) FROM LGPRODUCT

JOIN LGBRAND ON LGPRODUCT.Brand\_ID  LGBRAND.Brand\_ID

WHERE Brand\_Type = 'Premium') ORDER BY P.Prod\_Price DESC;

1. **Her kitap için kitap adı, maliyet ve yayın yılı**

SELECT Book\_Title, Book\_Cost, Publish\_Year FROM BOOK

ORDER BY Book\_Title ASC;

1. **Müşterilerin ad ve soyadları (büyük/küçük harfe duyarsız sıralama)**

SELECT Cust\_Fname, Cust\_Lname FROM LGCUSTOMER

ORDER BY UPPERCust\_Lname), UPPERCust\_Fname);

1. **Ödünç verme bilgileri (ödünç verme numarasına göre sıralı)**

SELECT Loan\_Num, Loan\_Date, Due\_Date FROM LOAN

ORDER BY Loan\_Num ASC;

1. **Kitap numarası, kitap adı ve konusu**

SELECT Book\_Num, Book\_Title, Book\_Subject FROM BOOK

ORDER BY Book\_Num ASC;

1. **Farklı yayın yıllarını listeleme**

SELECT DISTINCT Publish\_Year FROM BOOK

ORDER BY Publish\_Year ASC;

1. **Farklı kitap konuları**

SELECT DISTINCT Book\_Subject FROM BOOK

ORDER BY Book\_Subject ASC;

1. **Kitap numarası, adı ve maliyeti**

SELECT Book\_Num, Book\_Title, Book\_Cost FROM BOOK

ORDER BY Book\_Num ASC;

1. **Tüm çıkışlar: Çıkış numarası, kitap numarası, kullanıcı kimliği, çıkış ve son tarih**

SELECT Loan\_Num, Book\_Num, Patron\_ID, Loan\_Date, Due\_Date FROM LOAN

ORDER BY Due\_Date DESC, Loan\_Num ASC;

1. **Kitap adı, yılı ve konusu (birkaç kriterle sıralama)**

SELECT Book\_Title, Publish\_Year, Book\_Subject FROM BOOK

ORDER BY Book\_Subject ASC, Publish\_Year DESC, Book\_Title ASC;

1. **Fiyatı $59.95 olan kitaplar**

SELECT Book\_Num, Book\_Title, Book\_Cost FROM BOOK

WHERE Book\_Cost  59.95 ORDER BY Book\_Num ASC;

1. **"Veritabanı" konulu kitaplar**

SELECT Book\_Num, Book\_Title, Book\_Cost FROM BOOK

WHERE Book\_Subject = 'Veritabanı' ORDER BY Book\_Num ASC;

1. **5 Nisan 2021'den önce ödünç verilen kitaplar**

SELECT Loan\_Num, Book\_Num, Loan\_Date FROM LOAN

WHERE Loan\_Date < '20210405' ORDER BY Loan\_Num ASC;

1. **2015 sonrası yayınlanan "Programlama" konulu kitaplar**

SELECT Book\_Num, Book\_Title, Publish\_Year FROM BOOK

WHERE Publish\_Year  2015 AND Book\_Subject = 'Programlama' ORDER BY Book\_Num ASC;

1. **"Middleware" veya "Cloud" konulu ve fiyatı $70'dan fazla olan kitaplar**

SELECT Book\_Num, Book\_Title, Book\_Subject, Book\_Cost FROM BOOK

WHERE Book\_Subject = 'Middleware' OR Book\_Subject = 'Cloud')

AND Book\_Cost  70 ORDER BY Book\_Num ASC;

1. **1980'lerde doğan yazarlar**

SELECT Author\_ID, Author\_Fname, Author\_Lname, Birth\_Year FROM AUTHOR

WHERE Birth\_Year BETWEEN 1980 AND 1989 ORDER BY Author\_ID ASC;

1. **Başlığında "Veritabanı" geçen kitaplar (büyük/küçük harf duyarsız)**

SELECT Book\_Num, Book\_Title, Book\_Subject FROM BOOK

WHERE UPPERBook\_Title) LIKE '%VERİTABANI%'

ORDER BY Book\_Num ASC;

1. **Öğrenci olan tüm kullanıcılar**

SELECT Patron\_ID, Patron\_Fname, Patron\_Lname FROM PATRON

WHERE Patron\_Type = 'Student' ORDER BY Patron\_ID ASC;

1. **Soyadı "C" harfiyle başlayan kullanıcılar**

SELECT Patron\_ID, Patron\_Fname, Patron\_Lname, Patron\_Type FROM PATRON

WHERE Patron\_Lname LIKE 'C%' ORDER BY Patron\_ID ASC;

1. **Doğum yılı bilinmeyen yazarlar**

SELECT Author\_ID, Author\_Fname, Author\_Lname FROM AUTHOR

WHERE Birth\_Year IS NULL ORDER BY Author\_ID ASC;

1. **Doğum yılı bilinen yazarlar**

SELECT Author\_ID, Author\_Fname, Author\_Lname FROM AUTHOR

WHERE Birth\_Year IS NOT NULL ORDER BY Author\_ID ASC;

1. **Henüz iade edilmemiş tüm ödemeler**

SELECT Loan\_Num, Book\_Num, Patron\_ID, Loan\_Date, Due\_Date FROM LOAN

WHERE Return\_Date IS NULL ORDER BY Book\_Num ASC;

1. **Yazarlar (doğum yılına göre azalan, soyada göre artan sıralama)**

SELECT Author\_ID, Author\_Fname, Author\_Lname, Birth\_Year FROM AUTHOR

ORDER BY Birth\_Year DESC, Author\_Lname ASC;

1. **Sistemdeki kitap sayısı**

SELECT COUNT\* AS Total\_Books FROM BOOK;

1. **Farklı kitap konularının sayısı**

SELECT COUNTDISTINCT Book\_Subject) AS Unique\_Subjects FROM BOOK;

1. **Ödünç verilmemiş kitapların sayısı**

SELECT COUNT\* AS Available\_Books FROM BOOK

WHERE Book\_Num NOT IN SELECT DISTINCT Book\_Num FROM LOAN;

1. **En pahalı kitap fiyatı**

SELECT MAXBook\_Cost) AS Max\_Book\_Cost FROM BOOK;

1. **En ucuz kitap fiyatı**

SELECT MINBook\_Cost) AS Min\_Book\_Cost FROM BOOK;

1. **Bir kitabı ödünç alan farklı kullanıcı sayısı**

SELECT COUNTDISTINCT Patron\_ID AS Unique\_Borrowers FROM LOAN;

1. **Konu başına kitap sayısı**

SELECT Book\_Subject, COUNT\* AS Book\_Count FROM BOOK

GROUP BY Book\_Subject

ORDER BY Book\_Count DESC, Book\_Subject ASC;

1. **Yazar başına yazılan kitap sayısı**

SELECT A.Author\_ID, COUNTB.Book\_Num) AS Book\_Count FROM AUTHOR A

JOIN BOOK\_AUTHOR BA ON A.Author\_ID  BA.Author\_ID JOIN BOOK B ON BA.Book\_Num  B.Book\_Num

GROUP BY A.Author\_ID

ORDER BY Book\_Count DESC, A.Author\_ID ASC;

1. **Kütüphanedeki tüm kitapların toplam değeri**

SELECT SUMBook\_Cost) AS Total\_Book\_Value FROM BOOK;

1. **Her ödeme için kullanıcı, kitap ve kaç gün ödünç tutulduğu**

SELECT Patron\_ID, Book\_Num, DATEDIFFReturn\_Date, Loan\_Date) AS Days\_ Held

FROM LOAN

WHERE Return\_Date IS NOT NULL

ORDER BY Days\_Held DESC, Patron\_ID ASC, Book\_Num ASC;

1. **Kullanıcılar ve kullanıcı türleri (kullanıcı kimliğine göre sıralı)**

SELECT Patron\_ID, CONCATPatron\_Fname, ' ', Patron\_Lname) AS Full\_Name, Patron\_Type

FROM PATRON

ORDER BY Patron\_ID ASC;

1. **Kitaplar, yılları ve konuları**

SELECT Book\_Num, Publish\_Year, Book\_Title, Book\_Subject FROM BOOK

ORDER BY Book\_Num ASC;

1. **Yazarlar ve yazdıkları kitaplar**

SELECT A.Author\_Lname, A.Author\_Fname, B.Book\_Num FROM AUTHOR A

JOIN BOOK\_AUTHOR BA ON A.Author\_ID  BA.Author\_ID JOIN BOOK B ON BA.Book\_Num  B.Book\_Num

ORDER BY A.Author\_Lname ASC, A.Author\_Fname ASC, B.Book\_Num ASC;

1. **Her kitap için yazarlar ve kitap bilgileri**

SELECT A.Author\_ID, B.Book\_Num, B.Book\_Title, B.Book\_Subject FROM AUTHOR A

JOIN BOOK\_AUTHOR BA ON A.Author\_ID  BA.Author\_ID JOIN BOOK B ON BA.Book\_Num  B.Book\_Num

ORDER BY B.Book\_Num ASC, A.Author\_ID ASC;

1. **Yazarlar, kitap adları ve değiştirme maliyetleri**

SELECT A.Author\_Lname, A.Author\_Fname, B.Book\_Title, B.Book\_Cost FROM AUTHOR A

JOIN BOOK\_AUTHOR BA ON A.Author\_ID  BA.Author\_ID JOIN BOOK B ON BA.Book\_Num  B.Book\_Num

ORDER BY B.Book\_Num ASC, A.Author\_ID ASC;

1. **Şu anda ödünç verilmiş kitaplar**

SELECT L.Patron\_ID, B.Book\_Num, CONCATP.Patron\_Fname, ' ', P.Patron\_Lna me) AS Patron\_Name, B.Book\_Title

FROM LOAN L

JOIN BOOK B ON L.Book\_Num  B.Book\_Num JOIN PATRON P ON L.Patron\_ID  P.Patron\_ID WHERE L.Return\_Date IS NULL

ORDER BY P.Patron\_Lname ASC, B.Book\_Title ASC;

1. **Kullanıcılar (tam ad ve türüne göre sıralama)**

SELECT Patron\_ID, CONCATPatron\_Fname, ' ', Patron\_Lname) AS Full\_Name, Patron\_Type

FROM PATRON

ORDER BY Patron\_Type ASC, Patron\_Fname ASC;

1. **Kaç kez ödünç verilen kitaplar (hiç verilmemiş olanlar hariç)**

SELECT Book\_Num, COUNT\* AS Loan\_Count FROM LOAN

GROUP BY Book\_Num HAVING COUNT\*  0

ORDER BY Loan\_Count DESC, Book\_Num DESC;

1. **"Bulut" konulu kitaplar ve yazarları**

SELECT A.Author\_ID, A.Author\_Fname, A.Author\_Lname, B.Book\_Num, B.Book

\_Title

FROM BOOK B

JOIN BOOK\_AUTHOR BA ON B.Book\_Num  BA.Book\_Num JOIN AUTHOR A ON BA.Author\_ID  A.Author\_ID

WHERE B.Book\_Subject = 'Bulut'

ORDER BY B.Book\_Title ASC, A.Author\_Lname ASC;

1. **En çok kitap ödünç alan kullanıcılar**

SELECT Patron\_ID, COUNTBook\_Num) AS Loan\_Count FROM LOAN

GROUP BY Patron\_ID

ORDER BY Loan\_Count DESC, Patron\_ID ASC;

1. **2021ʼde ödünç alınan kitaplar**

SELECT Book\_Num, Loan\_Date FROM LOAN

WHERE YEARLoan\_Date)  2021 ORDER BY Loan\_Date ASC;

1. **Aynı anda 2ʼden fazla kitap ödünç alan kullanıcılar**

SELECT Patron\_ID, COUNTBook\_Num) AS Loan\_Count FROM LOAN

GROUP BY Patron\_ID, Loan\_Date HAVING COUNTBook\_Num)  2 ORDER BY Loan\_Count DESC;

1. **Kullanıcı başına toplam borç**

SELECT Patron\_ID, SUMBook\_Cost) AS Total\_Cost FROM LOAN L

JOIN BOOK B ON L.Book\_Num  B.Book\_Num GROUP BY Patron\_ID

ORDER BY Total\_Cost DESC;

1. **2021ʼde ödünç alınan ve iade edilen kitaplar**

SELECT Book\_Num, Loan\_Date, Return\_Date FROM LOAN

WHERE YEARLoan\_Date)  2021 AND Return\_Date IS NOT NULL ORDER BY Loan\_Date ASC;

1. **Bir kitap için tüm ödünç alma kayıtları**

SELECT Book\_Num, Patron\_ID, Loan\_Date, Return\_Date FROM LOAN

WHERE Book\_Num = 'B001' ORDER BY Loan\_Date ASC;

1. **2020ʼden beri hiç ödünç verilmemiş kitaplar**

SELECT Book\_Num, Book\_Title FROM BOOK

WHERE Book\_Num NOT IN 

SELECT DISTINCT Book\_Num FROM LOAN WHERE Loan\_Date >= '202001

-01'

)

ORDER BY Book\_Num ASC;

1. **Aynı kullanıcı tarafından birden fazla kez ödünç alınan kitaplar**

SELECT Book\_Num, Patron\_ID, COUNT\* AS Loan\_Count FROM LOAN

GROUP BY Book\_Num, Patron\_ID HAVING COUNT\*  1

ORDER BY Loan\_Count DESC;

1. **Her konu için ortalama kitap maliyeti**

SELECT Book\_Subject, ROUNDAVGBook\_Cost), 2) AS Avg\_Cost FROM BOOK

GROUP BY Book\_Subject ORDER BY Avg\_Cost DESC;

1. **En son ödünç alınan kitaplar**

SELECT Book\_Num, Loan\_Date FROM LOAN

ORDER BY Loan\_Date DESC LIMIT 5;

1. **Bir kullanıcının ödünç aldığı en pahalı kitap**

SELECT Patron\_ID, MAXBook\_Cost) AS Max\_Cost FROM LOAN L

JOIN BOOK B ON L.Book\_Num  B.Book\_Num GROUP BY Patron\_ID

ORDER BY Max\_Cost DESC;

1. **Bir kullanıcının ödünç aldığı kitap sayısı**

SELECT Patron\_ID, COUNTBook\_Num) AS Book\_Count FROM LOAN

GROUP BY Patron\_ID

ORDER BY Book\_Count DESC;

1. **En çok ödünç alınan konu başlığı**

SELECT B.Book\_Subject, COUNTL.Book\_Num) AS Loan\_Count FROM LOAN L

JOIN BOOK B ON L.Book\_Num  B.Book\_Num GROUP BY B.Book\_Subject

ORDER BY Loan\_Count DESC LIMIT 1;