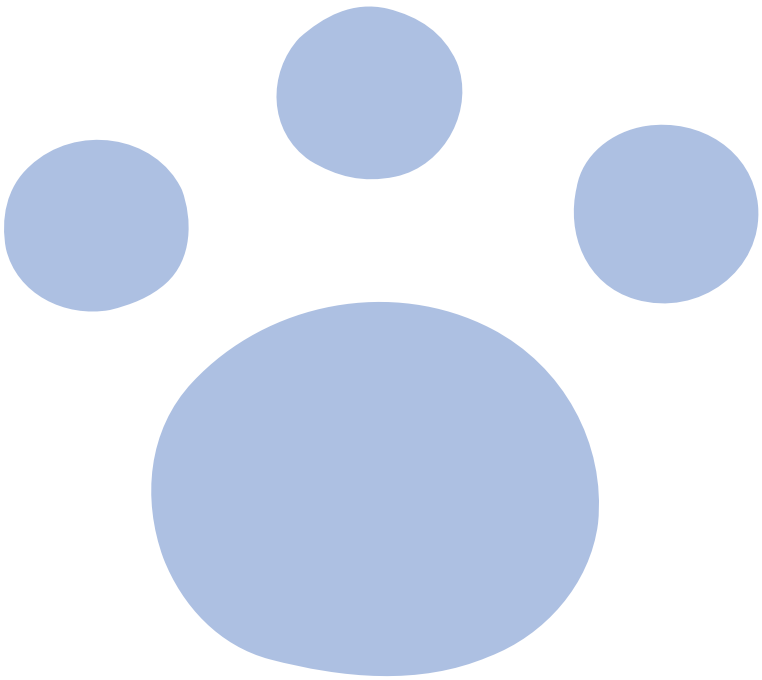
CD-ROM 収録コンテンツ

解答編

第2章～第4章「練習問題」、第5章「応用問題」の解答



第2章〜第4章「練習問題」、第5章「応用問題」の解答

解答は、ここに記載しているものが唯一の正解とは限りません。他の書き方ができる場合もありますので、ぜひ自分で別解も考えてみてください。

第2章　ひとつのテーブルを扱う

その1

第1問

```
SELECT
  Address
FROM
  Customers
;
```

第2問

```
SELECT
  ProductName
FROM
  Products
;
```

第3問

```
SELECT
  Price
FROM
  Products
;
```

第4問

```
SELECT
  EmployeeName
FROM
  Employees
;
```

第5問

```
SELECT
  Email
FROM
  Employees
;
```

その2

第1問

```
SELECT
  CustomerName
, Address
FROM
  Customers
;
```

第2問

```
SELECT
  ProductID
, ProductName
, Price
FROM
  Products
;
```

第3問

```
SELECT
  EmployeeName
, Email
, Height
FROM
  Employees
;
```

第4問

```
SELECT
  CustomerCode
, CustomerName
, CustomerCode
FROM
  Customers
;
```

第5問

```
SELECT
  ProductCode
, Price
, ProductName
, ProductCode
FROM
  Products
;
```

その3

第1問

```
SELECT
  EmployeeName AS 社員名
FROM
  Employees
;
```

第2問

```
SELECT
  CustomerCode AS 顧客コード
, CustomerName AS 顧客名
FROM
  Customers
;
```

第3問

```
SELECT
  ProductCode AS 商品コード
, ProductName AS 商品名
, Price AS 価格
FROM
  Products
;
```

第4問

```
SELECT
  CustomerName AS 顧客名
, CustomerName AS 得意先名
FROM
  Customers
;
```

第5問

```
SELECT
  EmployeeName AS 社員名
, Email AS メールアドレス
, Email AS 連絡先
FROM
  Employees
;
```

その4

第1問

```
SELECT
  Amount * 0.15 AS "給与の15%"
FROM
  Salary
;
```

第2問

```
SELECT
  Height * 0.5 AS 身長の半分
FROM
  Employees
;
```

第3問

```
SELECT
  Weight * 3 - 50 AS 体重の3倍引く50
FROM
  Employees
;
```

第4問

```
SELECT
  (Price + 100) * 0.3
                    AS "(価格+100)の30%"
FROM
  Products
;
```

第5問

```
SELECT
  (Quantity + 200 ) / 3
                      AS "(数量+200)÷3"
FROM
  Sales
;
```

その5

第1問

```
SELECT
  Height * 3 - Weight * 2.5 AS 結果
FROM
  Employees
;
```

第2問

```
SELECT
  HireFiscalYear / Weight + Height
                                AS 結果
FROM
  Employees
;
```

第3問

```
SELECT
  Quantity + CustomerID * ProductID
                  * EmployeeID AS 結果
FROM
  Sales
;
```

第4問

```
SELECT
  Price - ProductCode * CategoryID
                                AS 結果
FROM
  Products
;
```

第5問

```
SELECT
  CustomerID + CustomerClassID *
                        CustomerClassID *
                    CustomerClassID AS 結果
FROM
  Customers
;
```

その6

第1問

```
SELECT
  EmployeeName || 'さん' AS 社員名
FROM
  Employees
;
```

第2問

```
SELECT
   'E-MAIL:' || Email AS
                                メールアドレス
FROM
  Employees
;
```

第3問

```
SELECT
  EmployeeName || 'さんの'
               || 'E-MAIL:' || Email AS 連絡先
FROM
  Employees
;
```

第4問

```
SELECT
  CustomerName || '様のお住まいは'
                  || Address AS お得意様連絡先
FROM
  Customers
;
```

第5問

```
SELECT
  '社員' || EmployeeName
                || 'さんの血液型は' || BloodType
                    || '型' AS 社員血液型
FROM
  Employees
;
```

その7

第1問

```
SELECT
  COUNT( CustomerID ) AS お得意様数
FROM
  Customers
;
```

第2問

```
SELECT
  SUM( Weight ) AS 社員体重合計
FROM
  Employees
;
```

第3問

```
SELECT
  MAX( Price ) AS 最高額価格
FROM
  Products
;
```

第4問

```
SELECT
  MIN( Weight ) AS 最軽量体重
FROM
  Employees
;
```

第5問

```
SELECT
  AVG( Height ) AS 平均身長
, AVG( Weight ) AS 平均体重
FROM
  Employees
;
```

その8

第1問

```
SELECT
  ProductName
FROM
  Products
WHERE
  Price >= 2500
;
```

第2問

```
SELECT
  EmployeeName
, Weight
FROM
  Employees
WHERE
  Weight >= 70
;
```

第3問

```
SELECT
  EmployeeName
, Height
FROM
  Employees
WHERE
  Height BETWEEN 160 AND 180
;
```

第4問

```
SELECT
  SaleID
FROM
  Sales
WHERE
  SaleDate >= '2007-06-01'
;
```

第5問

```
SELECT
  EmployeeName
, Height
, Weight
FROM
  Employees
WHERE
  Height >= 170
  AND
  Weight >= 60
;
```

その9

第1問

```
SELECT
  CustomerName AS 会社名
FROM
  Customers
WHERE
  CustomerName LIKE '%株式会社%'
;
```

第2問

```
SELECT
  AVG( Height ) AS 平均身長
FROM
  Employees
WHERE
  EmployeeName LIKE '%一%'
;
```

第3問

```
SELECT
  COUNT( CustomerID ) AS 顧客数
FROM
  Customers
WHERE
  CustomerName NOT LIKE '%株式会社%'
;
```

第4問

```
SELECT
  EmployeeName
, Height
FROM
  Employees
WHERE
  EmployeeName  LIKE '%リ%'
  AND Height <= 160
;
```

第5問

```
SELECT
  *
FROM
  Customers
WHERE
  CustomerName NOT LIKE '%株式会社%'
  AND Address LIKE '%江戸川区%'
;
```

その10

第1問

```
SELECT
  EmployeeName AS 社員名
, CASE
    WHEN Height < 160 THEN 'A'
    WHEN Height < 170 THEN 'B'
    WHEN Height < 180 THEN 'C'
    ELSE 'D'
  END AS ランク
FROM
  Employees
;
```

第2問

```
SELECT
  SalaryID AS 給与ID
, CASE
    WHEN Amount < 150000 THEN 'D'
    WHEN Amount < 300000 THEN 'C'
    WHEN Amount < 500000 THEN 'B'
    ELSE 'A'
  END AS ランク
FROM
  Salary
;
```

第3問

```
SELECT
  EmployeeName AS 社員名
, CASE
    WHEN Weight < 60 THEN 'A'
    WHEN Weight < 70 THEN 'B'
    WHEN Weight < 80 THEN 'C'
    ELSE 'D'
  END AS ランク
FROM
  Employees
;
```

第4問

```
SELECT
  SaleID AS 販売ID
, CASE
    WHEN Quantity < 10 THEN 'B'
    ELSE 'A'
  END AS ランク
FROM
  Sales
;
```

第5問

```
SELECT
  EmployeeName AS 社員名
, Height AS 身長
, CASE
    WHEN Height < 160 THEN 'A'
    WHEN Height < 170 THEN 'B'
    WHEN Height < 180 THEN 'C'
    ELSE 'D'
  END AS ランク
FROM
  Employees
;
```

その11

第1問

```
SELECT
  CustomerID AS 顧客ID
, COUNT( * ) AS 件数
FROM
  Sales
GROUP BY
  CustomerID
;
```

第2問

```
SELECT
  EmployeeID AS 社員ID
, SUM( Amount ) AS 合計
FROM
  Salary
GROUP BY
  EmployeeID
;
```

第3問

```
SELECT
  CustomerID AS 顧客ID
, ProductID AS 商品ID
, SUM( Quantity ) AS 数量
FROM
  Sales
GROUP BY
  CustomerID
, ProductID
;
```

第4問

```
SELECT
  BloodType  AS 血液型
, AVG( Height ) AS 平均身長
, AVG( Weight ) AS 平均体重
FROM
  Employees
GROUP BY
  BloodType
;
```

第5問

```
SELECT
  EmployeeID AS 社員ID
, COUNT( * ) AS 支給回数
, AVG( Amount ) AS 平均支給額
FROM
  Salary
GROUP BY
  EmployeeID
;
```

その12

第1問

```
SELECT
  EmployeeID AS 社員ID
, COUNT( * ) AS 支給回数
FROM
  Salary
GROUP BY
  EmployeeID
HAVING
  COUNT( * ) < 12
;
```

第2問

```
SELECT
  PrefecturalID AS 県ID
, COUNT( * ) AS 顧客数
FROM
  Customers
GROUP BY
  PrefecturalID
HAVING
  COUNT( * )  > 1
;
```

第3問

```
SELECT
  ProductID AS 商品ID
, COUNT( * ) AS 売上レコード数
FROM
  Sales
GROUP BY
  ProductID
HAVING
  COUNT( * ) >= 10
                          AND COUNT( * ) <= 50
;
```

第4問

```
SELECT
  BloodType  AS 血液型
, COUNT( * )  AS データ件数
FROM
  Employees
GROUP BY
  BloodType
HAVING
  COUNT( * ) >=10
;
```

第5問

```
SELECT
  ProductID AS 商品ID
, SUM( Quantity ) AS 数量合計
FROM
  Sales
GROUP BY
  ProductID
HAVING
  SUM( Quantity ) >= 100
                     AND SUM( Quantity ) <= 200
;
```

その13

第1問

```
SELECT
  PrefecturalID AS 県ID
, COUNT( * ) AS 顧客数
FROM
  Customers
WHERE
  PrefecturalID >= 10
GROUP BY
  PrefecturalID
HAVING
  COUNT( * ) > 1
;
```

第2問

```
SELECT
  EmployeeID AS 社員ID
, COUNT( * ) AS 支給回数
FROM
  Salary
WHERE
  EmployeeID >= 20
GROUP BY
  EmployeeID
HAVING
  COUNT( * ) >= 12
;
```

第3問

```
SELECT
  ProductID AS 商品ID
, COUNT( * ) AS 売上レコード数
FROM
  Sales
WHERE
  ProductID >= 20
          AND ProductID <= 30
GROUP BY
  ProductID
HAVING
  COUNT( * ) >= 30
;
```

第4問

```
SELECT
  BloodType  AS 血液型
, COUNT( * )  AS データ件数
FROM
  Employees
WHERE
  Height >= 165
GROUP BY
  BloodType
HAVING
  COUNT( * ) >= 5
;
```

第5問

```
SELECT
  ProductID AS 商品ID
, SUM( Quantity ) AS 数量合計
FROM
  Sales
WHERE
  SaleDate >= '2007-06-01'
GROUP BY
  ProductID
HAVING
  SUM( Quantity ) >= 200
;
```

その14

第1問

```
SELECT
  HireFiscalYear AS 入社年度
, SUM(
      CASE
        WHEN Height <= 160 THEN 1
        ELSE 0
      END
    ) AS "160cm以下"
, SUM(
      CASE
        WHEN Height <= 170 THEN 1
        ELSE 0
      END
    ) AS "170cm以下"
, SUM(
      CASE
        WHEN Height <= 180 THEN 1
        ELSE 0
      END
    ) AS "180cm以下"
, SUM(
      CASE
        WHEN Height > 180 THEN 1
        ELSE 0
      END
    ) AS "181cm以上"
FROM
  Employees
GROUP BY
  HireFiscalYear
;
```

第2問

```
SELECT
  CategoryID AS 商品カテゴリID
, SUM(
      CASE
        WHEN Price < 100 THEN 1
        ELSE 0
      END
    ) AS "100円未満"
, SUM(
      CASE
        WHEN Price < 400 THEN 1
        ELSE 0
      END
    ) AS "400円未満"
, SUM(
      CASE
        WHEN Price < 1000 THEN 1
        ELSE 0
      END
    ) AS "1000円未満"
```

```
, SUM(
      CASE
        WHEN Price >= 1000 THEN 1
        ELSE 0
      END
     ) AS "1000円以上"
FROM
  Products
GROUP BY
  CategoryID
;
```

第3問

```
SELECT
  CustomerID AS 顧客ID
, SUM(
      CASE
           WHEN MONTH( SaleDate ) = 9 THEN
Quantity
        ELSE 0
      END
     ) AS "9月"
, SUM(
      CASE
           WHEN MONTH( SaleDate ) = 10 THEN
Quantity
        ELSE 0
      END
     ) AS "10月"
, SUM(
      CASE
           WHEN MONTH( SaleDate ) = 11 THEN
Quantity
        ELSE 0
      END
     ) AS "11月"
FROM
  Sales
WHERE
  YEAR( SaleDate ) = 2006
GROUP BY
  CustomerID
;
```

第4問

```
SELECT
  PrefecturalID AS 都道府県ID
, SUM(
      CASE
        WHEN CustomerClassID = 1 THEN 1
        ELSE 0
      END
     ) AS 法人
, SUM(
      CASE
        WHEN CustomerClassID = 2 THEN 1
        ELSE 0
      END
     ) AS 個人
FROM
  Customers
GROUP BY
  PrefecturalID
;
```

第5問

```
SELECT
  HireFiscalYear AS 入社年度
, SUM(
      CASE
        WHEN Weight <= 50 THEN 1
        ELSE 0
      END
     ) AS "50kg以下"
, SUM(
      CASE
        WHEN Weight <= 60 THEN 1
        ELSE 0
      END
     ) AS "51～60kg"
, SUM(
      CASE
        WHEN Weight <= 80 THEN 1
        ELSE 0
      END
     ) AS "61～80kg"
, SUM(
      CASE
        WHEN Weight > 80 THEN 1
        ELSE 0
      END
     ) AS "80kg超"
FROM
  Employees
GROUP BY
  HireFiscalYear
;
```

その15

第1問

```
SELECT
  EmployeeID
, EmployeeName
, Birthday
FROM
  Employees
ORDER BY
  Birthday
;
```

第2問

```
SELECT
  SaleID
, Quantity
, CustomerID
, ProductID
, SaleDate
FROM
  Sales
ORDER BY
  CustomerID
, ProductID
, SaleDate DESC
;
```

第3問

```
SELECT
  CategoryID
, COUNT( * ) AS 商品数
FROM
  Products
WHERE
```

```
  Price <= 1000
GROUP BY
  CategoryID
HAVING
  COUNT( * ) < 5
ORDER BY
  CategoryID
;
```

第4問

```
SELECT
  EmployeeID AS 社員ID
, SUM( Amount ) AS 給与合計
FROM
  Salary
GROUP BY
  EmployeeID
ORDER BY
  給与合計 DESC
;
```

第5問

```
SELECT
  DepartmentID AS 部署ID
, COUNT( * ) AS レコード数
FROM
  BelongTo
WHERE
  EndDate IS NULL
GROUP BY
  DepartmentID
ORDER BY
  COUNT( EmployeeID ) DESC
;
```

その16

第1問

```
SELECT DISTINCT
  HireFiscalYear
FROM
  Employees
;
```

第2問

```
SELECT DISTINCT
  CustomerID
, ProductID
FROM
  Sales
;
```

第3問

```
SELECT DISTINCT
  CustomerClassID
, PrefecturalID
FROM
  Customers
;
```

第4問

```
SELECT DISTINCT
  CustomerID
, ProductID
, EmployeeID
FROM
  Sales
;
```

第5問

```
SELECT DISTINCT
  Price
, CategoryID
FROM
  Products
;
```

第3章　複数のテーブルを扱う

その1

第1問

```
SELECT
  EmployeeID
, EmployeeName
FROM
  Employees
WHERE
  EmployeeID IN
  (
   SELECT
     EmployeeID
   FROM
     Salary
   GROUP BY
     EmployeeID
   HAVING
     MAX( Amount ) >= 300000
  )
;
```

第2問

```
SELECT
  SaleID
, Quantity
, CustomerID
, (
   SELECT
     CustomerName
   FROM
     Customers
   WHERE
     CustomerID = Sales.CustomerID
  ) AS 顧客名
FROM
  Sales
WHERE
  Quantity >= 100
;
```

第3問

```
SELECT
  ProductID
, ProductName
FROM
  Products
WHERE
  ProductID IN
  (
   SELECT
     ProductID
   FROM
     Sales
   GROUP BY
     ProductID
   HAVING
     SUM( Quantity ) >= 100
  )
;
```

第4問

```
SELECT
  EmployeeID
, EmployeeName
, (
   SELECT
     MAX( Amount )
   FROM
     Salary
   WHERE
     EmployeeID
               = Employees.EmployeeID
  ) AS 最高給与額
FROM
  Employees
WHERE
  EmployeeID IN
  (
   SELECT
     EmployeeID
   FROM
     Salary
   GROUP BY
     EmployeeID
   HAVING
     MAX( Amount ) >= 300000
  )
;
```

第5問

```
SELECT
  SaleID
, Quantity
, CustomerID
, (
   SELECT
     CustomerName
   FROM
     Customers
   WHERE
     CustomerID = Sales.CustomerID
  ) AS 顧客名
, (
   SELECT
     ProductName
   FROM
     Products
   WHERE
     ProductID = Sales.ProductID
  ) AS 商品名
FROM
  Sales
WHERE
  Quantity >= 100
;
```

その2

第1問

```
SELECT
  B.EmployeeName
, A.PayDate
, A.Amount
FROM
  Salary AS A
    JOIN
  Employees AS B
    ON A.EmployeeID = B.EmployeeID
ORDER BY
  B.EmployeeID
;
```

第2問

```
SELECT
  A.Quantity
, B.CustomerName
, C.ProductName
, D.EmployeeName
FROM
  Sales AS A
    JOIN
  Customers AS B
    ON A.CustomerID = B.CustomerID
      JOIN
    Products AS C
      ON A.ProductID = C.ProductID
        JOIN
      Employees AS D
        ON A.EmployeeID = D.EmployeeID
WHERE
  A.Quantity >= 200
;
```

第3問

```
SELECT
  A.ProductID
, B.ProductName
, SUM( A.Quantity ) AS 数量合計
FROM
  Sales AS A
    JOIN
  Products AS B
    ON A.ProductID = B.ProductID
GROUP BY
  A.ProductID
, B.ProductName
HAVING
  SUM( A.Quantity ) >= 300
;
```

第4問

```
SELECT
  A.Quantity
, B.CustomerName
, C.ProductName
, D.EmployeeName
FROM
  Sales AS A
, Customers B
, Products C
, Employees D
WHERE
  A.Quantity >= 200
  AND
  A.CustomerID = B.CustomerID
  AND
  A.ProductID = C.ProductID
  AND
  A.EmployeeID = D.EmployeeID
;
```

第5問

```
SELECT
  A.CustomerName
, B.PrefecturalName
, C.CustomerClassName
FROM
  Customers A
, Prefecturals B
, CustomerClasses C
WHERE
  A.PrefecturalID = B.PrefecturalID
  AND
  A.CustomerClassID
                = C.CustomerClassID
ORDER BY
  A.PrefecturalID
;
```

その3

第1問

```
SELECT
  B.CategoryID
, MAX( C.CategoryName ) AS カテゴリ名
, SUM( A.Quantity ) AS 数量合計
FROM
  Sales AS A
    JOIN
  Products AS B
    ON A.ProductID = B.ProductID
      JOIN
    Categories AS C
      ON B.CategoryID = C.CategoryID
GROUP BY
  B.CategoryID
;
```

第2問

```
SELECT
  SUM( A.Quantity ) AS 合計数量
, B.PrefecturalID
, MAX( C.PrefecturalName ) AS 県名
FROM
  Sales AS A
    JOIN
  Customers AS B
    ON A.CustomerID = B.CustomerID
      JOIN
    Prefecturals AS C
      ON B.PrefecturalID = C.PrefecturalID
GROUP BY
  B.PrefecturalID
;
```

【解説】
グループ化を行う場合、選択リストで許可されるのは、グループ化のキーとなる列名か、集合関数のみであることを思い出してください（第2章その11）。PrefecturalIDにPrefecturalNameは1つしかないはずなのに、わざわざ MAX(C.PrefecturalName) AS県名 としているのはこのためです。PrefecturalNameは文字列のため、MAXを使いました。MINでも構いません。

第3問

```
SELECT
  MAX( A.Quantity ) AS 最大数量
, B.CustomerClassID
, MIN( C.CustomerClassName ) AS 顧客クラス名
FROM
  Sales AS A
    JOIN
  Customers AS B
    ON A.CustomerID = B.CustomerID
      JOIN
    CustomerClasses AS C
      ON B.CustomerClassID = C.CustomerClassID
GROUP BY
  B.CustomerClassID
;
```

【解説】
第2問と同じ理由で MIN(C.CustomerClassName) を使いました。

第4問

```
SELECT
  SUM( A.Quantity ) AS 合計数量
, B.PrefecturalID
, MAX( C.PrefecturalName ) AS 県名
FROM
  Sales AS A
, Customers AS B
, Prefecturals AS C
WHERE
  A.CustomerID = B.CustomerID
  AND
  B.PrefecturalID = C.PrefecturalID
GROUP BY
  B.PrefecturalID
;
```

第5問

```
SELECT
  MAX( A.Quantity ) AS 最大数量
, B.CustomerClassID
, MIN( C.CustomerClassName ) AS 顧客クラス名
FROM
  Sales AS A
, Customers AS B
, CustomerClasses AS C
WHERE
  A.CustomerID = B.CustomerID
  AND
  B.CustomerClassID = C.CustomerClassID
GROUP BY
  B.CustomerClassID
;
```

その4

第1問

```
SELECT
  A.CustomerName
, SUM(
      CASE
        WHEN B.CustomerID IS NULL
                              THEN 0
        ELSE B.Quantity
      END
    ) AS 販売数量合計
FROM
  Customers AS A
    LEFT OUTER JOIN
  Sales AS B
    ON A.CustomerID = B.CustomerID
GROUP BY
  A.CustomerName
;
```

第2問

```
SELECT
  A.EmployeeID
, MAX( A.EmployeeName ) AS 社員名
, SUM(
       CASE
         WHEN B.EmployeeID IS NULL
                               THEN 0
         ELSE 1
       END
    ) AS 販売件数
FROM
  Employees AS A
    LEFT OUTER JOIN
  Sales AS B
    ON A.EmployeeID = B.EmployeeID
GROUP BY
  A.EmployeeID
;
```

【解説】
「チー」という同じ名前の社員が2名います。もちろんEmployeeIDは異なります。もし、EmployeeNameでGROUP BYすると2人の「チー」の販売件数が合算されてしまいます。EmployeeIDでGROUP BYして、MAX(A.EmployeeName) AS 社員名により社員名を得ます。

第3問

```
SELECT
  A.PrefecturalName
, SUM(
     CASE
       WHEN B.PrefecturalID
                   IS NULL THEN 0
       ELSE 1
     END
    ) AS 顧客数
FROM
  Prefecturals AS A
    LEFT OUTER JOIN
  Customers AS B
    ON A.PrefecturalID
                 = B.PrefecturalID
GROUP BY
```

```
  A.PrefecturalName
;
```

第4問

```
SELECT
  A.EmployeeID
, CASE
    WHEN B.CNT IS NULL THEN 0
    ELSE B.CNT
  END AS 販売件数
FROM
  Employees AS A
    LEFT OUTER JOIN
  (
   SELECT
     EmployeeID
   , COUNT( * ) AS CNT
   FROM
     Sales
   GROUP BY
     EmployeeID
  ) AS B
    ON A.EmployeeID = B.EmployeeID
;
```

第5問

```
SELECT
  A.EmployeeName
, (
   CASE
     WHEN B.Amount IS NULL THEN 0
     ELSE B.Amount
   END
  ) AS 支給額
FROM
  Employees AS A
    LEFT OUTER JOIN
  (
   SELECT
     EmployeeID
   , Amount
   FROM
     Salary
   WHERE
     PayDate = '2007-02-25'
  ) AS B
    ON A.EmployeeID = B.EmployeeID
;
```

その5

第1問

```
SELECT
  p1.ProductName AS 商品名1
, p2.ProductName AS 商品名2
FROM
  Products AS p1
    JOIN
  Products AS p2
    ON p1.ProductID < p2.ProductID
    AND p1.CategoryID = p2.CategoryID
;
```

第2問

```
SELECT
  c1.CustomerName AS 顧客名1
, c2.CustomerName AS 顧客名2
FROM
  Customers AS c1
    JOIN
  Customers AS c2
    ON c1.CustomerID < c2.CustomerID
    AND c1.PrefecturalID = c2.PrefecturalID
    AND c1.CustomerClassID = c2.CustomerClassID
;
```

第3問

```
SELECT
  e1.EmployeeName AS 従業員名1
, e2.EmployeeName AS 従業員名2
FROM
  Employees AS e1
    JOIN
  Employees AS e2
    ON e1.Birthday > e2.Birthday
;
```

第4問

```
SELECT
  c1.CategoryName AS カテゴリ名1
, c2.CategoryName AS カテゴリ名2
FROM
  Categories AS c1
    JOIN
  Categories AS c2
    ON c1.CategoryID < c2.CategoryID
;
```

第5問

```
SELECT
  c1.CustomerName AS 顧客名1
, c2.CustomerName AS 顧客名2
FROM
  Customers AS c1
    JOIN
  Customers AS c2
    ON c1.CustomerID < c2.CustomerID
    AND c1.PrefecturalID = c2.PrefecturalID
    AND c1.CustomerClassID = c2.CustomerClassID
WHERE
  c1.PrefecturalID <> 13
;
```

その6

第1問

```
SELECT DISTINCT
  A.ProductID
, B.ProductName
, A.Quantity
FROM
  Sales AS A
    JOIN
  Products AS B
    ON A.ProductID = B.ProductID
WHERE
  A.Quantity =
  (
   SELECT
     MAX( Quantity )
   FROM
     Sales AS C
   WHERE
     A.ProductID = C.ProductID
  )
ORDER BY
  A.ProductID
;
```

第2問

```
SELECT
  ProductID
, ProductName
FROM
  Products AS A
WHERE
  EXISTS
  (
   SELECT
     'X'
   FROM
     Sales AS B
   WHERE
     A.ProductID = B.ProductID
  )
;
```

第3問

```
SELECT
  ProductID
, ProductName
FROM
  Products AS A
WHERE
  NOT EXISTS
  (
   SELECT
     'X'
   FROM
     Sales AS B
   WHERE
     A.ProductID = B.ProductID
  )
;
```

第4問

```
SELECT
  A.ProductID
, B.ProductName
, A.Quantity
FROM
  (
   SELECT
     ProductID
   , MAX( Quantity ) AS Quantity
   FROM
     Sales
   GROUP BY
     ProductID
  ) AS A
    JOIN
  Products AS B
    ON A.ProductID = B.ProductID
ORDER BY
  A.ProductID
;
```

第5問

```
SELECT
  ProductID
, ProductName
FROM
  Products AS A
WHERE
  ProductID IN
  (
   SELECT
     ProductID
   FROM
     Sales AS B
   WHERE
     A.ProductID = B.ProductID
   HAVING
     AVG( Quantity )
           <= MAX( Quantity ) / 10
  )
ORDER BY
  ProductID
;
```

その7

第1問

```
SELECT
  DepartmentID AS ID
, DepartmentName AS 名前
FROM
  Departments
UNION ALL
SELECT
  CategoryID AS ID
, CategoryName AS 名前
FROM
  Categories
;
```

第2問

```
SELECT
  'Departments' AS テーブル名
, DepartmentID AS ID
, DepartmentName AS 名前
FROM
```

```
  Departments
UNION ALL
SELECT
  'Categories' AS テーブル名
, CategoryID AS ID
, CategoryName AS 名前
FROM
  Categories
ORDER BY
  テーブル名
, ID
;
```

第3問

```
SELECT
  DepartmentID AS ID
, DepartmentName AS 名前
FROM
  Departments
UNION ALL
SELECT
  CustomerClassID AS ID
, CustomerClassName AS 名前
FROM
  CustomerClasses
UNION ALL
SELECT
  CategoryID AS ID
, CategoryName AS 名前
FROM
  Categories
UNION ALL
SELECT
  PrefecturalID AS ID
, PrefecturalName AS 名前
FROM
  Prefecturals
;
```

第4問

```
SELECT
  'Departments' AS テーブル名
, DepartmentID AS ID
, DepartmentName AS 名前
FROM
  Departments
UNION ALL
SELECT
  'CustomerClasses' AS テーブル名
, CustomerClassID AS ID
, CustomerClassName AS 名前
FROM
  CustomerClasses
UNION ALL
SELECT
  'Categories' AS テーブル名
, CategoryID AS ID
, CategoryName AS 名前
FROM
  Categories
UNION ALL
SELECT
  'Prefecturals' AS テーブル名
, PrefecturalID AS ID
, PrefecturalName AS 名前
FROM
  Prefecturals
ORDER BY
  テーブル名
, ID
;
```

第5問

```
SELECT
  A.SaleID
, A.ProductID
, A.Quantity
, B.CustomerClassID
, B.CustomerName
FROM
  Sales A
    JOIN
  Customers B
    ON A.CustomerID = B.CustomerID
WHERE
  A.Quantity >= 10
  AND
  B.CustomerClassID = 2
UNION ALL
SELECT
  A.SaleID
, A.ProductID
, A.Quantity
, B.CustomerClassID
, B.CustomerName
FROM
  Sales A
    JOIN
  Customers B
    ON A.CustomerID = B.CustomerID
WHERE
  A.Quantity >= 100
  AND
  B.CustomerClassID = 1
;
```

その8

第1問

```
SELECT
  CustomerID AS ID
, CustomerName AS 名前
FROM
  Customers
UNION
SELECT
  EmployeeID AS ID
, EmployeeName AS 名前
FROM
  Employees
ORDER BY
  ID
;
```

第2問

```
SELECT
  EmployeeID AS ID
, EmployeeName AS 名前
FROM
  Employees
UNION
SELECT
  EmployeeID AS ID
, EmployeeName AS 名前
```

```
FROM
  Employees
ORDER BY
  ID
;
```

第3問

```
SELECT
  ProductID
FROM
  Products
UNION
SELECT
  ProductID
FROM
  Sales
ORDER BY
  ProductID
;
```

第4問

```
SELECT
  CustomerID
, ProductID
FROM
  Sales
WHERE
  SaleDate BETWEEN '2006-10-01'
                   AND '2006-12-31'
  AND
  Quantity >= 10
UNION
SELECT
  CustomerID
, ProductID
FROM
  Sales
WHERE
  SaleDate BETWEEN '2007-01-01'
                   AND '2007-03-31'
  AND
  Quantity >= 10
UNION
SELECT
  CustomerID
, ProductID
FROM
  Sales
WHERE
  SaleDate BETWEEN '2007-04-01'
                   AND '2007-06-30'
  AND
  Quantity >= 10
ORDER BY
  CustomerID
, ProductID
;
```

第5問

```
SELECT
  A.ProductID
FROM
  Sales A
    JOIN
  Customers B
    ON A.CustomerID = B.CustomerID
WHERE
  A.Quantity >= 10
  AND
  B.CustomerClassID = 2
UNION
SELECT
  A.ProductID
FROM
  Sales A
    JOIN
  Customers B
    ON A.CustomerID = B.CustomerID
WHERE
  A.Quantity >= 100
  AND
  B.CustomerClassID = 1
ORDER BY
  ProductID
;
```

その9

第1問

```
SELECT
  CustomerID AS ID
, CustomerName AS 名前
FROM
  Customers
INTERSECT
SELECT
  EmployeeID AS ID
, EmployeeName AS 名前
FROM
  Employees
ORDER BY
  ID
;
```

第2問

```
SELECT
  EmployeeID AS ID
, EmployeeName AS 名前
FROM
  Employees
INTERSECT
SELECT
  EmployeeID AS ID
, EmployeeName AS 名前
FROM
  Employees
ORDER BY
  ID
;
```

第3問

```
SELECT
  ProductID
FROM
  Products
INTERSECT
SELECT
  ProductID
FROM
  Sales
ORDER BY
  ProductID
;
```

第4問

```
SELECT
  CustomerID
, ProductID
FROM
  Sales
WHERE
  SaleDate BETWEEN
                '2006-10-01'
                 AND '2006-12-31'
  AND
  Quantity >= 10
INTERSECT
SELECT
  CustomerID
, ProductID
FROM
  Sales
WHERE
  SaleDate BETWEEN
                '2007-01-01'
                AND '2007-03-31'
  AND
  Quantity >= 10
INTERSECT
SELECT
  CustomerID
, ProductID
FROM
  Sales
WHERE
  SaleDate BETWEEN
                '2007-04-01'
                AND '2007-06-30'
  AND
  Quantity >= 10
ORDER BY
  CustomerID
, ProductID
;
```

第5問

```
SELECT
  A.ProductID
FROM
  Sales A
    JOIN
  Customers B
    ON A.CustomerID
                = B.CustomerID
WHERE
  A.Quantity >= 10
  AND
  B.CustomerClassID = 2
INTERSECT
SELECT
  A.ProductID
FROM
  Sales A
    JOIN
  Customers B
    ON A.CustomerID
                = B.CustomerID
WHERE
  A.Quantity >= 100
  AND
  B.CustomerClassID = 1
ORDER BY
  ProductID
;
```

その10

第1問

```
SELECT
  CustomerID AS ID
, CustomerName AS 名前
FROM
  Customers
EXCEPT
SELECT
  EmployeeID AS ID
, EmployeeName AS 名前
FROM
  Employees
ORDER BY
  ID
;
```

第2問

```
SELECT
  EmployeeID AS ID
, EmployeeName AS 名前
FROM
  Employees
EXCEPT
SELECT
  EmployeeID AS ID
, EmployeeName AS 名前
FROM
  Employees
ORDER BY
  ID
;
```

第3問

```
SELECT
  ProductID
FROM
  Products
EXCEPT
SELECT
  ProductID
FROM
  Sales
ORDER BY
  ProductID
;
```

第4問

```
SELECT
  A.ProductID
FROM
  Sales A
    JOIN
  Customers B
    ON A.CustomerID = B.CustomerID
WHERE
  A.Quantity >= 100
  AND
  B.CustomerClassID = 1
EXCEPT
SELECT
```

```
  A.ProductID
FROM
  Sales A
    JOIN
  Customers B
    ON A.CustomerID = B.CustomerID
WHERE
  A.Quantity >= 10
  AND
  B.CustomerClassID = 2
ORDER BY
  ProductID
;
```

第5問

```
SELECT
  A.ProductID
FROM
  Sales A
    JOIN
  Customers B
    ON A.CustomerID = B.CustomerID
WHERE
  A.Quantity >= 10
  AND
  B.CustomerClassID = 2
EXCEPT
SELECT
  A.ProductID
FROM
  Sales A
    JOIN
  Customers B
    ON A.CustomerID = B.CustomerID
WHERE
  A.Quantity >= 100
  AND
  B.CustomerClassID = 1
ORDER BY
  ProductID
;
```

第4章　データを変更する

その1

第1問

```
INSERT
INTO Employees
(
  EmployeeID
, EmployeeName
, Height
, Weight
, Email
, HireFiscalYear
, Birthday
, BloodType
)
VALUES
(
 31
, 'モクモク'
, 170
, 60
, 'moku@nekoyasudo'
, 2007
, '1989-08-08'
, 'AB'
)
;
```

第2問

```
INSERT
INTO BelongTo
(
  BelongID
, EmployeeID
, DepartmentID
, StartDate
)
VALUES
(
  35
, 31
, 1
, '2007-09-01'
)
;
```

第3問

```
INSERT
INTO Sales
(
  SaleID
, Quantity
, CustomerID
, ProductID
, EmployeeID
, SaleDate
)
VALUES
(
  1006
, 10
, 1
, 40
, 31
, '2007-09-01'
)
;
```

第4問

```
INSERT
INTO Salary
(
 SalaryID
, EmployeeID
, PayDate
, Amount
)
VALUES
(
  354
, 31
, '2007-09-05'
, 100000
)
;
```

第5問

```
INSERT
INTO Customers
(
  CustomerID
, CustomerName
, Address
, CustomerClassID
, PrefecturalID
)
VALUES
(
  31
, '有限会社貉商会'
, '和歌山県吉野郡'
, 1
, 30
)
;
```

その2

第1問

```
INSERT
INTO Salary
(
  SalaryID
, EmployeeID
, PayDate
, Amount
)
SELECT
  EmployeeID + 20000
, EmployeeID
, '2007-10-01'
, 20000
FROM
```

```
  Employees
WHERE
  HireFiscalYear <=1993
;
```

第2問

```
INSERT
INTO Customers
(
  CustomerID
, CustomerCode
, CustomerName
, Address
, CustomerClassID
, PrefecturalID
)
SELECT
  EmployeeID + 10000
, EmployeeID + 10000
, EmployeeName
, '江戸川区西小岩'
, 2
, 13
FROM
  Employees
WHERE
  HireFiscalYear <= 1988
;
```

第3問

```
INSERT
INTO Sales
(
  SaleID
, Quantity
, CustomerID
, ProductID
, EmployeeID
, SaleDate
)
SELECT
  EmployeeID + 30000
, 10
, 10
, 20
, EmployeeID
, '2007-09-01'
FROM
  Employees
WHERE
  BloodType = 'O'
;
```

第4問

```
INSERT
INTO Sales
(
  SaleID
, Quantity
, CustomerID
, ProductID
, EmployeeID
, SaleDate
)
SELECT
  CustomerID + 40000
, 20
, CustomerID
, 21
, 5
, '2007-09-05'
FROM
  Customers
WHERE
  PrefecturalID = 8
;
```

第5問

```
INSERT
INTO Sales
(
  SaleID
, Quantity
, CustomerID
, ProductID
, EmployeeID
, SaleDate
)
SELECT
  ProductID + 50000
, 30
, 15
, ProductID
, 2
, '2007-09-10'
FROM
  Products
WHERE
  CategoryID = 5
;
```

その3

第1問

```
UPDATE
  Customers
SET
  CustomerCode = CustomerCode + 1000
;
```

第2問

```
UPDATE
  Employees
SET
  Email = Email || '.co.jp'
;
```

第3問

```
UPDATE
  Employees
SET
  Height = Height + 2
, Weight = Weight - 5
;
```

第4問

```
UPDATE
  Departments
SET
  DepartmentName = DepartmentName || '部'
;
```

第5問

```
UPDATE
  Customers
SET
  CustomerName =
    CASE
      WHEN CustomerClassID = 1 THEN
        CustomerName || '御中'
      ELSE
        CustomerName || '様'
    END
;
```

その4

第1問

```
UPDATE
  Employees
SET
  Height = Height + 5
WHERE
  EmployeeID = 10
;
```

第2問

```
UPDATE
  Salary
SET
  Amount = Amount + 20000
WHERE
  EmployeeID = 5
  AND
  PayDate = '2007-03-25'
;
```

第3問

```
UPDATE
  Employees
SET
  Height = Height - 2
, Weight = Weight + 3
WHERE
  BloodType = 'AB'
;
```

第4問

```
UPDATE
  Sales
SET
  Quantity = Quantity +10
WHERE
  CustomerID = 10
  AND
  ProductID = 5
  AND
  SaleDate >= '2007-05-31'
;
```

第5問

```
UPDATE
  Products
SET
  Price =
    CASE WHEN Price >=2000
                    THEN Price * 0.8
         ELSE Price * 0.9
    END
WHERE
  CategoryID = 7
  AND
  Price >=1000
;
```

その5

第1問

```
UPDATE
  Products
SET
  Price = Price * 0.97
WHERE
  ProductID NOT IN
 (
  SELECT
    ProductID
  FROM
   Sales
 )
;
```

第2問

```
UPDATE
  Salary
SET
  Amount = Amount * 0.95
WHERE
  PayDate = '2007-10-01'
  AND
  EmployeeID IN
 (
  SELECT
    EmployeeID
  FROM
    Sales
  GROUP BY
    EmployeeID
  HAVING
    COUNT ( * ) < 10
 )
;
```

第3問

```
UPDATE
  Salary
SET
  Amount = Amount * 1.1
WHERE
  PayDate = '2007-10-01'
  AND
  EmployeeID IN
 (
  SELECT
    EmployeeID
  FROM
    Sales
```

```
  GROUP BY
    EmployeeID
  HAVING
    COUNT ( * ) >=  50
 )
;
```

第4問

```
UPDATE
  Salary
SET
  Amount = Amount * 0.9
WHERE
  PayDate = '2007-08-25'
  AND
  EmployeeID NOT IN
 (
  SELECT
    EmployeeID
  FROM
    Sales
  WHERE
    SaleDate < '2007-08-25'
 )
;
```

第5問

```
UPDATE
  Salary
SET
  Amount = Amount * 1.1
WHERE
  PayDate = '2007-08-25'
  AND
  EmployeeID IN
 (
  SELECT
    A.EmployeeID
  FROM
    Sales AS A
      JOIN
    Customers AS B
      ON A.CustomerID = B.CustomerID
  WHERE
    B.CustomerClassID = 1
    AND
    A.SaleDate < '2007-08-25'
 )
;
```

その6

第1問

```
UPDATE
  Customers
SET
  Address
  =
  (
   SELECT
     PrefecturalName
   FROM
     Prefecturals
   WHERE
     Customers.PrefecturalID
       = Prefecturals.PrefecturalID
  )
  || Address
WHERE
  EXISTS
  (
   SELECT
     'X'
   FROM
     Prefecturals
   WHERE
     Customers.PrefecturalID
       = Prefecturals.PrefecturalID
  )
;
```

第2問

```
UPDATE
  Salary
SET
  Amount
  = Amount +
  (
   SELECT
     SUM( Sales.Quantity
          * Products.Price ) * 0.03
   FROM
     Sales
       JOIN
     Products
       ON Sales.ProductID
              = Products.ProductID
   WHERE
     Sales.SaleDate < '2007-08-25'
     AND
     Salary.EmployeeID
              = Sales.EmployeeID
  )
WHERE
  PayDate = '2007-08-25'
  AND
  EXISTS
  (
   SELECT

'X'
   FROM
     Sales
   WHERE
     Sales.SaleDate < '2007-08-25'
   AND
     Salary.EmployeeID
              = Sales.EmployeeID
  )
;
```

第3問

```
UPDATE
  Products
SET
  Price
  =
  (
   SELECT
     AVG( Sales.Quantity
              * Products.Price )
   FROM
```

```
    Sales
  WHERE
    Products.ProductID
                = Sales.ProductID
 )
WHERE

  EXISTS
  (
   SELECT
     'X'
   FROM
     Sales
   WHERE
     Products.ProductID
                = Sales.ProductID
  )
;
```

第4問

```
UPDATE
  Products
SET
  ProductName
  =  ProductName || '(' ||
  (
   SELECT
     CategoryName
   FROM
     Categories
   WHERE
     Products.CategoryID
          = Categories.CategoryID
  )
  || ')'
WHERE
  EXISTS
  (
   SELECT
     'X'
   FROM
     Categories
   WHERE
     Products.CategoryID
          = Categories.CategoryID
  )
;
```

第5問

```
UPDATE
  Products
SET
  ProductName
  =
  (
   SELECT
     SUM( Quantity )
   FROM
     Sales
   WHERE
     Products.ProductID
                = Sales.ProductID
  )
  || '個も売れてるヒット商品！'
                || ProductName
WHERE
  (
   SELECT
     SUM( Quantity )
   FROM
     Sales
   WHERE
     Products.ProductID
                = Sales.ProductID
  ) >= 500
;
```

その7

第1問

```
DELETE
FROM
  BelongTo
;
```

第2問

```
DELETE
FROM
  Customers
;
```

第3問

```
DELETE
FROM
  Sales
;
```

第4問

```
DELETE
FROM
  Products
;
```

第5問

```
DELETE
FROM
  Employees
;
```

その8

第1問

```
DELETE
FROM
  Sales
WHERE
  SaleID = 1006
;
```

第2問

```
DELETE
FROM
  Salary
WHERE
  EmployeeID = 10
  AND
  PayDate = '2007-10-01'
;
```

第3問

```
DELETE
FROM
  Customers
WHERE
  CustomerID >= 10000
;
```

第4問

```
DELETE
FROM
  Products
WHERE
  CategoryID = 1
;
```

第5問

```
DELETE
FROM
  Customers
WHERE
  CustomerClassID = 2
  AND
  PrefecturalID IN (3, 5, 7, 13)
;
```

その9

第1問

```
DELETE
FROM
  Salary
WHERE
  EmployeeID NOT IN
 (
  SELECT
    EmployeeID
  FROM
    Sales
 )
;
```

第2問

```
DELETE
FROM
  Products
WHERE
  ProductID NOT IN
 (
  SELECT
    ProductID
  FROM
    Sales
  GROUP BY
    ProductID
  HAVING
    SUM( Quantity ) >= 20
 )
;
```

第3問

```
DELETE
FROM
  Customers
WHERE
  CustomerID NOT IN
 (
  SELECT
    CustomerID
  FROM
    Sales
 )
;
```

第4問

```
DELETE
FROM
  Employees
WHERE
  EmployeeID NOT IN
 (
  SELECT
    EmployeeID
  FROM
    Sales
  GROUP BY
    EmployeeID
  HAVING
    COUNT ( * ) > 5
 )
;
```

第5問

```
DELETE
FROM
  Sales
WHERE
  EmployeeID IN
 (
  SELECT
    EmployeeID
  FROM
    BelongTo
  WHERE
    EndDate IS NULL
    AND
    DepartmentID = 3
 )
;
```

第5章　応用問題

その1

```
SELECT
  'INSERT INTO Pref_Back VALUES ('
  || PrefecturalID
  || ','''
  || PrefecturalName
  || ''');' AS 都道府県のINSERT文
FROM
  Prefecturals
;
```

【解説】

まず、元になるテーブルはPrefecturalsです。そして選択リストはINSERT文という「ひとつの文字列」になります。後はテーブルから取り出した値と自分で指定する文字列を組み合わせればいいことになります。なお、シングルクオーテーションを文字列として指定する場合はエスケープしてやる必要があります。今回はシングルクオーテーションを2回続けて書くことでエスケープを指定していますが、RDBMSによってはバックスラッシュ（日本語環境だと¥マーク）を使うこともあります。

その2

```
SELECT
  年月
, SUM( 販売金額 )  AS 販売合計金額
FROM
 (
  SELECT
    SUBSTR(CAST( SaleDate AS VARCHAR ), 1, 7)
                                        AS 年月
  , s.Quantity * p.Price AS 販売金額
  FROM
    Sales AS s
      JOIN
    Products AS p
      ON s.ProductID = p.ProductID
 )
GROUP BY
  年月
ORDER BY
  年月
;
```

【解説】

簡単ですね。このレベルはスラスラと書けるようになりましょう。日付の扱いはRDBMSごとに異なります。

その3

```
SELECT
  e.EmployeeID
, e.EmployeeName
, 年月
, SUM(
      CASE WHEN 販売金額 IS NULL THEN 0
           ELSE 販売金額
      END
    )  AS 販売合計金額
FROM
  Employees AS e
    LEFT OUTER JOIN
 (
  SELECT
    s.EmployeeID
  , SUBSTR(CAST( SaleDate AS VARCHAR ), 1, 7)
                                        AS 年月
  , s.Quantity * p.Price AS 販売金額
  FROM
    Sales AS s
      JOIN
    Products AS p
      ON s.ProductID = p.ProductID
 ) AS x
   ON e.EmployeeID = x.EmployeeID
GROUP BY
  e.EmployeeID
, e.EmployeeName
, 年月
ORDER BY
  e.EmployeeID
, e.EmployeeName
, 年月
;
```

【解説】

すべての社員を出す必要があります。一方である社員に販売データが存在しない可能性もあります。そこで外部結合を使います。外部結合を使うと結合対象のレコードが見つからない場合はNULLが返ってきます。そこでNULLへの対応をつけておく必要があります。

その4

```
SELECT
  p.ProductID
, p.ProductName
, 年月
, SUM( s.Quantity * p.Price ) AS 販売合計金額
FROM
 (
  SELECT
    ProductID
  , SUBSTR(CAST( SaleDate AS VARCHAR ), 1, 7)
                                        AS 年月
  , Quantity
  FROM
    Sales
 ) AS s
    JOIN
  Products AS p
    ON s.ProductID = p.ProductID
WHERE
  p.CategoryID IN (1, 3, 9)
```

```
GROUP BY
  p.ProductID
, p.ProductName
, 年月
HAVING
  SUM( s.Quantity * p.Price ) > 5000
ORDER BY
  p.ProductID
, p.ProductName
, 年月 DESC
;
```

【解説】

問題の言い回しにひっかからなければ特に難しいところはないでしょう。最後のDESCによる降順指定を忘れると古い順に並んでしまう点に注意しましょう。条件がWHERE句とHAVING句の両方必要なのも気をつけるポイントです。

その5

```
SELECT
  c.CustomerID
, c.CustomerName
, p.ProductName
, SUM( s.Quantity * p.Price ) 販売合計金額
FROM
  Sales AS s
    JOIN
  Products AS p
    ON s.ProductID = p.ProductID
      JOIN
    Customers AS c
      ON s.CustomerID = c.CustomerID
GROUP BY
  c.CustomerID
, c.CustomerName
, p.ProductName
ORDER BY
  c.CustomerID
, c.CustomerName
, p.ProductName
;
```

【解説】

結合が数珠つなぎになってたぐっていくという形ではなく、Salesテーブルを中心にしてそれぞれくっついている点に注目してください。つまり、Customersテーブルと Productsテーブルが別々の飾りになっているということです。基本形は次のSQLになります。

```
SELECT
  CustomerID
, ProductID
, SUM( s.Quantity )
FROM
  Sales AS s
GROUP BY
  CustomerID
, ProductID
;
```

ここに順番に飾りをつけていくのだという手順をしっかりと習得してください。

その6

```
SELECT
  pr.PrefecturalID
, pr.PrefecturalName
, p.ProductName
, SUM( s.Quantity * p.Price ) 販売合計金額
FROM
  Sales AS s
    JOIN
  Products AS p
    ON s.ProductID = p.ProductID
      JOIN
   Customers AS c
      ON s.CustomerID = c.CustomerID
        JOIN
      Prefecturals AS pr
        ON c.PrefecturalID = pr.PrefecturalID
GROUP BY
  pr.PrefecturalID
, pr.PrefecturalName
, p.ProductName
ORDER BY
  pr.PrefecturalID
, pr.PrefecturalName
, p.ProductName
;
```

【解説】

ここでのポイントは、都道府県を引っ張ってくるためにCustomersテーブルをブリッジとして結合していることです。このように結合するテーブルが増えてくると、ついつい混乱して結合するほうに気を取られてしまいがちですが、基本となるSELECT文は先ほどの「顧客別商品別販売額一覧」と同じです。

その7

```
SELECT
  d.DepartmentID
, d.DepartmentName
, 年月
, AVG( Amount ) AS 平均給与
FROM
 (
  SELECT
    EmployeeID
  , PayDate
  , SUBSTR(CAST( PayDate AS VARCHAR ), 1, 7)
                                        AS 年月
  , Amount
  FROM
    Salary
  WHERE
    SUBSTR(CAST( PayDate AS VARCHAR ), 1, 4)
                                        = '2007'
 ) AS s
    JOIN
  BelongTo AS b
    ON s.EmployeeID = b.EmployeeID
    AND s.PayDate >= b.StartDate
    AND s.PayDate <
```

```
        CASE WHEN b.EndDate IS NULL
                              THEN '9999-12-31'
             ELSE b.EndDate
         END
       JOIN
    Departments AS d
      ON b.DepartmentID = d.DepartmentID
GROUP BY
  d.DepartmentID
, d.DepartmentName
, 年月
ORDER BY
  d.DepartmentID
```

```
, d.DepartmentName
, 年月
;
```

【解説】

社員は部門を異動しますので、SalaryテーブルのPayDate時点に、社員がどの部門に所属していたかを考慮しなければなりません。これを怠ると1社員の給与が複数の部門に反映されてしまいます。

BelongToテーブルを参照すると、社員がいつから(StartDate)いつまで(EndDate)どの部門に所属していたかがわかりますので、これを利用します。

その8

```
SELECT
  年月
, SUM(CASE
      WHEN p.CategoryID = 1 THEN s.Quantity
                                 * p.Price
       ELSE 0
      END ) AS Ct1
, SUM(CASE
      WHEN p.CategoryID = 2 THEN s.Quantity
                                 * p.Price
       ELSE 0
      END ) AS Ct2
, SUM(CASE
      WHEN p.CategoryID = 3 THEN s.Quantity
                                 * p.Price
       ELSE 0
      END ) AS Ct3
, SUM(CASE
      WHEN p.CategoryID = 4 THEN s.Quantity
                                 * p.Price
       ELSE 0
      END ) AS Ct4
, SUM(CASE
      WHEN p.CategoryID = 5 THEN s.Quantity
                                 * p.Price
       ELSE 0
      END ) AS Ct5
, SUM(CASE
      WHEN p.CategoryID = 6 THEN s.Quantity
                                 * p.Price
       ELSE 0
      END ) AS Ct6
, SUM(CASE
      WHEN p.CategoryID = 7 THEN s.Quantity
                                 * p.Price
       ELSE 0
      END ) AS Ct7
, SUM(CASE
      WHEN p.CategoryID = 8 THEN s.Quantity
                                 * p.Price
       ELSE 0
      END ) AS Ct8
, SUM(CASE
      WHEN p.CategoryID = 9 THEN s.Quantity
                                 * p.Price
       ELSE 0
      END ) AS Ct9
, SUM(CASE
      WHEN p.CategoryID = 10 THEN s.Quantity
                                  * p.Price
       ELSE 0
      END ) AS Ct10
FROM
 (
  SELECT
    ProductID
  , SUBSTR(CAST( SaleDate AS VARCHAR ), 1, 7)
                                       AS 年月
  , Quantity
  FROM
    Sales
 ) AS s
    JOIN
  Products AS p
    ON s.ProductID = p.ProductID
GROUP BY
  年月
ORDER BY
  年月
;
```

【解説】

いわゆるクロス集計です。グループ化を行うキー項目と横の振り分けを行うキー項目の区別をしっかりと考えれば、特に難しいところはないでしょう。

その9

```
SELECT
  ProductID
, ProductName
, "6月販売金額"
, "7月販売金額"
, CASE WHEN "6月販売金額" < "7月販売金額"
                              THEN '↑'
       WHEN "6月販売金額" = "7月販売金額"
                              THEN '→'
       ELSE '↓'
  END AS 対6月増減
, "8月販売金額"
```

```
, CASE WHEN "7月販売金額" < "8月販売金額"
                              THEN '↑'
       WHEN "7月販売金額" = "8月販売金額"
                              THEN '→'
       ELSE '↓'
  END AS 対7月増減
FROM
 (
  SELECT
    p.ProductID
  , p.ProductName
  , SUM(
```

```
      CASE WHEN s.SaleDate IS NULL THEN 0
           WHEN SUBSTR( CAST( s.SaleDate
              AS VARCHAR), 1, 7) = '2007-06'
                THEN s.Quantity * p.Price
           ELSE 0
      END
     ) AS "6月販売金額"
  , SUM(
      CASE WHEN s.Quantity IS NULL THEN 0
           WHEN SUBSTR( CAST( s.SaleDate
            AS VARCHAR), 1, 7) = '2007-07'
                THEN s.Quantity * p.Price
           ELSE 0
      END
     ) AS "7月販売金額"
  , SUM(
      CASE WHEN s.Quantity IS NULL THEN 0
           WHEN SUBSTR( CAST( s.SaleDate
             AS VARCHAR), 1, 7) = '2007-08'
                THEN s.Quantity * p.Price
           ELSE 0
      END
     ) AS "8月販売金額"
  FROM
    Products AS p
      LEFT OUTER JOIN
        Sales s
          ON p.ProductID = s.ProductID
    GROUP BY
      p.ProductID
    , p.ProductName
  )
ORDER BY
  ProductID
;
```

【解説】

Productsテーブルが主体となります。これに6月、7月および8月の売上を外部結合します。

1つのSalesテーブルから6月分のみを抽出した表式、7月分のみを抽出した表式、8月分のみを抽出した表式を副問い合わせによって作成して、Productsテーブルに順次外部結合していく方法が考えられます。この方法は、自己結合をしかも外部結合で行うことになります。

しかし、第3章その5「自己結合を使う」のワンポイントレッスンに記しましたが、自己結合は一般的にかなり重い処理となり、パフォーマンスが低下する場合が多く見受けられます。

そこでここでは、ProductsテーブルにSalesテーブルを1回だけ外部結合して、その結果をCASE式によって6月、7月、8月に振り分ける方法をとりました。

また対前月増減の判定前までを副問い合わせで行い、判定のみを問い合わせ本体で行っています。

その10

```
UPDATE
  Customers
SET
  CustomerCode =
  CustomerClassID * 1000000 + PrefecturalID
                        * 10000 + CustomerID
;
```

【解説】

実は問題文にトリックがあります。前ゼロで埋める、と書かれているので文字列連結をイメージする方も多いと思われます。たとえばPostgreSQLだと以下のように書いたりすることもあるのではないでしょうか。

```
REPLACE( TO_CHAR( CustomerClassID,'0' ) ||
TO_CHAR( PrefecturalID,'00' ) || TO_CHAR(
CustomerID,'0000' ),' ','' )
```

しかし、CustomerCode列のデータ型はINTEGERです。そのため文字列型の値はセットできません。ここで、上記の文字列連結のものをさらに数値型に変換する関数で囲んだりすることもあるのですが、今回の例のように計算でゼロの場所をコントロールすることも可能です。

今回のような場合は、いきなりUPDATE文を書くのではなく、以下のようなSELECT文を書いて期待したとおりの値になるかどうかを確認してから、UPDATE文を組み立てるようにしましょう。

```
SELECT
    (CustomerClassID * 1000000) +
(PrefecturalID
                        * 10000) + CustomerID
FROM
  Customers
;
```