【V-UP】ORACLE MASTER Silver SQL 2019 講座

演習問題

ThinkethBank

WEB・プログラム・ネットワーク・OA・資格取得
KENスクール

■ 注意事項

・ 本書に記載されている内容に基づく設定や運用の結果について、著者、株式会社シンクスバンクは一切の責任を負いかねます。　あらかじめご了承ください。
・ 本書の内容は、改善のために事前連絡なしに変更されることがあります。

■ 商標に関して

・ 本書に記載されているすべての会社名およびロゴ、製品名などは、該当する各社の商標または登録商標です。
・ 本書では、TM、®、©マークは省略し、また一般に使われている名称を用いている場合があります。

目次

はじめに

ここでは、本書の目的、対象読者、構成、関連情報などを説明します。

目的

本書は、「 【V-UP】ORACLE MASTER Silver SQL 2019 講座 」 のテキストに対応した演習となります。

対象読者

このテキストは「【V-UP】ORACLE MASTER Silver SQL 2019 講座」のテキストについて学習中の方、または学習を終えた方を対象としています。

構成

本書は、「【V-UP】ORACLE MASTER Silver SQL 2019 講座 」のテキストに構成を合わせてあります。また、演習問題がない章もあります。

関連コンテンツ

・ 「【V-UP】ORACLE MASTER Silver SQL 2019 講座」テキスト：

関連情報

・ Oracle Live SQL
https://livesql.oracle.com/

Oracle Live SQL について

本演習で実行する SQL 文は、Oracle Live SQL 上で実施することを前提とします。
Oracle Live SQL を使用するには、Oracle アカウント（プロファイル）が必要です。
Oracle アカウントの作成は無償です。
Oracle アカウントを作成されていない方は、上記 Oracle Live SQL サイトから Oracle アカウントを作成して、Oracle Live SQL を使用できるようにしてください。

1. リレーショナル・データベースの概要

本章の演習問題はありません。
テキストの内容を確認してください。

2. SQL SELECT 文を使用したデータの取得

演習 2-1

以下の構成の表があります。

従業員表（employees）

Column	Null？	Type
emp_id	NOT NULL	VARCHAR2(5)
first_name	NOT NULL	VARCHAR2(20)
last_name	NOT NULL	VARCHAR2(20)
address		VARCHAR2(30)

住所（address）が’Tokyo’の従業員の中で、従業員名（first_name,last_name）が同姓同名の社員の重複行を除去して、従業員名を取り出す SELECT 文を作成してください。
なお、first_name と last_name の表示は連結して、間を半角スペースで区切ってください。

※ヒント：テキスト「重複の除去」「行の選択」「連結演算子」の項を参照しましょう。

演習 2-2

以下の構成の表があります。

従業員表（Employees）

Column	Null？	Type
emp_id	NOT NULL	VARCHAR2(5)
first_name	NOT NULL	VARCHAR2(20)
last_name	NOT NULL	VARCHAR2(20)
salary		NUMBER(10)

来年度人件費の試算のため、給与（salary）を全社員一律で 1.05 倍にアップするレポートを作成する必要があります。
従業員 ID（emp_id)、従業員名、新給与を表示する SELECT 文を作成してください。
表示は以下の条件とします。

・従業員名は first_name 列と last_name 列を半角スペースで連結
・表示結果の従業員名の列名を”EMP-NAME”、新給与の列名を”NEW-SALARY“として表示することとします。

※ヒント：テキスト「計算式の指定」「列別名」の項を参照しましょう。

演習 2-3

dual 表を用いて、以下の文字列が表示されるような SELECT 文を作成してください。

```
It's show time.
```

※ヒント：テキスト「Dual 表」「文字列と代替引用符」を参照しましょう。

演習 2-4

以下の構成の表があります。

従業員表（Employees）

Column	Null？	Type
emp_id	NOT NULL	VARCHAR2(5)
first_name	NOT NULL	VARCHAR2(20)
last_name	NOT NULL	VARCHAR2(20)
age		NUMBER(3)
salary		NUMBER(10)

年齢が 40 以上 60 以下、または給与が 400000 より大きい従業員について、全ての列情報を表示する SELECT 文を作成してください。なお、salary 列が NULL のデータは含まないものとします。

※ヒント：テキスト「論理演算子」「比較演算子」「演算子の優先順位」を参照しましょう。

演習解答例

演習 2-1

```
SELECT
  DISTINCT first_name || ' ' ||  last_name
FROM
  employees
WHERE
  address = 'Tokyo'
;
```

演習 2-2

```
SELECT
  first_name || ' ' || last_name AS "EMP-NAME",
  salary*1.05 AS "NEW-SALARY"
FROM
  employees
;
```

演習 2-3

```
SELECT
  'It''s show time.'
FROM
  dual
;
```

※別解

```
select
  q'*It's show time.*'
FROM
  dual
;
```

「*」は引用符デリミタ

演習 2-4

```
SELECT
  *
FROM
  employees
WHERE
  (
    age >= 40
    AND
```

```
    age <= 60
    OR salary > 400000
  )
  AND salary IS NOT NULL
;
```

3. データの制限とソート

演習 3-1

以下の構成の表があります。

従業員表（Employees）

Column	Null？	Type
emp_id	NOT NULL	VARCHAR2(5)
first_name	NOT NULL	VARCHAR2(20)
last_name	NOT NULL	VARCHAR2(20)
salary		NUMBER(10)

以下の条件を満たす SELECT 文を作成してください。

・emp_id 値と、salary の値を 1.1 倍にした”NEW-SALARY”という列別名の値を表示します。

・NEW-SALARY 列別名をキーに昇順に並び替えます。

・従業員表全行を表示します。

※ヒント：テキスト「列の値による並べ替え」の項を参照しましょう。

演習 3-2

以下の構成の表があります。

従業員表（Employees）

Column	Null？	Type
emp_id	NOT NULL	VARCHAR2(5)
first_name	NOT NULL	VARCHAR2(20)
last_name	NOT NULL	VARCHAR2(20)
salary		NUMBER(10)

last_name を降順、かつ salary を昇順に並び替えて、すべての列と行を表示する SELECT 文を作成してください。
ソートキーの順は last_name を第一ソートキー、salary を第二ソートキーとします。

※ヒント：テキスト「複数の条件を指定」の項を参照しましょう。

演習 3-3

以下の構成の表があります。

従業員表（Employees）

Column	Null？	Type
emp_id	NOT NULL	VARCHAR2(5)
first_name	NOT NULL	VARCHAR2(20)
last_name	NOT NULL	VARCHAR2(20)
salary		NUMBER(10)

salary を昇順に並び替えて、全行の 50%の行を表示する SELECT 文を作成してください。
表示する列は、全列とします。

※ヒント：テキスト「SQL 行制限の構文」の項を参照しましょう。

演習解答例

演習 3-1

```
SELECT
  emp_id, salary * 1.1 AS "NEW-SALARY"
FROM
  employees
ORDER BY
  "NEW-SALARY"
;
```

演習 3-2

```
SELECT
  *
FROM
  employees
ORDER BY
  last_name ASC , salary DESC
;
```

演習 3-3

```
SELECT
  *
FROM
  employees
ORDER BY salary DESC
FETCH FIRST 50 PERCENT ROWS ONLY
;
```

4. 単一行関数を使用した出力のカスタマイズ

演習 4-1

dual 表を用いて、以下の数値を四捨五入する SELECT 文を作成してください。

1250.175

なお、表示結果は、小数第二位までを表示することとします。

※ヒント：テキスト「数値関数」を参照しましょう。

演習 4-2

3 桁のナンバーの後に都道府県が記載されているデータがあります。
都道府県はアルファベット表記で、大文字小文字は統一されていません。

001-HOKKAIDOU
…
013-Tokyo
…
047-okinawa

都道府県の部分のみを切り出して、すべて大文字で表示する SELECT 文を作成してください。

※ヒント：テキスト「文字関数」を参照しましょう。

演習 4-3

以下の構成の表があります。

従業員表（Employees）

Column	Null？	Type
emp_id	NOT NULL	VARCHAR2(5)
first_name	NOT NULL	VARCHAR2(20)
last_name	NOT NULL	VARCHAR2(20)
start_date		DATE

直近 1 年間に入社した従業員の従業員 ID リストを作成する必要があります。
入社日（start_date）が現在日時から 1 年以内の従業員 ID（emp_id)を表示する SELECT 文を作成してください。

※ヒント：テキスト「日時の計算と日時関数」の項を参照しましょう。

演習 4-4

以下の構成の表があります。

従業員表（Employees）

Column	Null？	Type
emp_id	NOT NULL	VARCHAR2(5)
first_name	NOT NULL	VARCHAR2(20)
last_name	NOT NULL	VARCHAR2(20)
start_date		DATE

入社記念品を贈るため、入社日（start_date）から 1 年後の最初の月曜日の日付と、その従業員 ID（emp_id）のリストを作成する必要があります。このための SELECT 文を作成してください。

※ヒント：テキスト「日時の計算と日時関数」の項を参照しましょう。

演習解答例

演習 4-1

```
SELECT
  ROUND(1250.175,2)
FROM
  dual
;
```

演習 4-2

```
SELECT
  UPPER(SUBSTR(place_name,5))
FROM
  place
;
```

演習 4-3

```
SELECT
  emp_id
FROM
  employees
WHERE
  ADD_MONTHS(SYSDATE,-12) <= start_date
;
```

演習 4-4

```
SELECT
  NEXT_DAY(ADD_MONTHS(start_date,12),'MON'),emp_id
FROM
  employees
;
```

5. 変換関数と条件式の使用

演習 5-1

以下の構成の表があります。

従業員表（Employees）

Column	Null？	Type
emp_id	NOT NULL	VARCHAR2(5)
first_name	NOT NULL	VARCHAR2(20)
last_name	NOT NULL	VARCHAR2(20)
start_date		DATE

2020 年 1 月以降に入社した従業員の従業員 ID（emp_id）とその入社日を表示する SELECT 文を作成してください。
なお、入社日（start_date）は、'2021-08-01'のように、年 4 桁、月 2 桁、日 2 桁を'-'で区切る形式で表示する必要があります。

※ヒント：テキスト「データ型の変換関数」「日時の書式」の項を参照しましょう。

演習 5-2

以下の構成の表があります。

従業員表（Employees）

Column	Null？	Type
emp_id	NOT NULL	VARCHAR2(5)
first_name	NOT NULL	VARCHAR2(20)
last_name	NOT NULL	VARCHAR2(20)
salary		NUMBER(7)

給与が 5000 以上の従業員の、従業員 ID（emp_id）と給与（salary）を表示する SELECT 文を作成してください。なお、給与の表示は、最大 7 桁、3 桁ずつカンマ区切りで、先頭に'$'を付与して表示する必要があります。

※ヒント：テキスト「数値の書式」の項を参照しましょう。

演習 5-3

以下の構成の表があります。

従業員表（Employees）

Column	Null？	Type
emp_id	NOT NULL	VARCHAR2(5)
name	NOT NULL	VARCHAR2(30)
salary		NUMBER(10)
special_bonus		NUMBER(10)

各従業員の emp_id と年間給与総額（列別名：TOTAL）を表示する SQL を作成してください。年間給与総額は、給与（salary）12 か月分と、特別賞与（special_bonus）の合計です。特別賞与は一部従業員のみに年 1 回支払われます。

※ヒント：テキスト「汎用関数による NULL の処理」の項を参照しましょう。

演習 5-4

以下の構成の表があります。

従業員表（Employees）

Column	Null？	Type
emp_id	NOT NULL	VARCHAR2(5)
name	NOT NULL	VARCHAR2(30)
start_date		DATE

現在年から、従業員の入社年（start_date）の年数を引いた値により、経験値（列別名：GRADE）を表示する SQL を作成してください。
経験値には以下の条件で、"LOW","MIDDLE","HIGH"を表示します。
0 年～5 年未満：LOW
5 年～10 年未満：MIDDLE
10 年以上：HIGH
表示項目は、従業員 ID(emp_id)、従業員名(name)、経験値を表示してください。

※ヒント：テキスト「データ型の変換」「CASE 式」の項を参照しましょう。

演習解答例

演習 5-1

```
SELECT
  emp_id,TO_CHAR(start_date, 'YYYY-MM-DD')
FROM
  employees
WHERE
  start_date >= TO_DATE('01-JAN-21')
;
```

※別解

```
SELECT
  emp_id,TO_CHAR(start_date, 'YYYY-MM-DD')
FROM
  employees
WHERE
  start_date >= '01-JAN-21'
;
```

WHERE 句の比較で暗黙的データ変換が実行される

演習 5-2

```
SELECT
  TO_CHAR(salary, '$9,999,999')
FROM
  employees
WHERE
  salary >= 5000
;
```

演習 5-3

```
SELECT
  emp_id,
  salary*12+(NVL2 (special_bonus, special_bonus, 0)) AS TOTAL
FROM
  employees
;
```

演習 5-4

```
SELECT
  emp_id,name,
  CASE
    WHEN
      TO_NUMBER(TO_CHAR(SYSDATE,'YYYY')) -
      TO_NUMBER(TO_CHAR(start_date,'YYYY')) < 5
    THEN
      'LOW'
    WHEN
      TO_NUMBER(TO_CHAR(SYSDATE,'YYYY')) -
      TO_NUMBER(TO_CHAR(start_date,'YYYY')) >=6 AND
      TO_NUMBER(TO_CHAR(SYSDATE,'YYYY')) -
      TO_NUMBER(TO_CHAR(start_date,'YYYY')) <10
    THEN
      'MIDDLE'
    ELSE
      'HIGH'
  END
  AS "GRADE"
FROM
  employees
;
```

6. グループ関数を使用した集計データのレポート

演習 6-1

以下の構成の表があります。

従業員表（Employees）

Column	Null？	Type
department_id	NOT NULL	VARCHAR2(5)
emp_id	NOT NULL	VARCHAR2(5)
first_name		VARCHAR2(20)
last_name		VARCHAR2(20)
salary		NUMBER(10)

各部門（department_id）で給与（salary）が最も高い値と、部門毎の給与合計値を表示する SELECT 文を作成してください。表示順は部門 ID の昇順とします。

※ヒント：テキスト「グループ化」「グループ関数」の項を参照しましょう。

演習 6-2

以下の構成の表があります。

従業員表（Employees）

Column	Null？	Type
department_id	NOT NULL	VARCHAR2(5)
emp_id	NOT NULL	VARCHAR2(5)
first_name	NOT NULL	VARCHAR2(20)
last_name	NOT NULL	VARCHAR2(20)
salary		NUMBER(10)

部門 ID（department_id）が'10001'の各従業員の給与（salary）を、","区切りで 1 行に表示する SELECT 文を作成してください。表示順は給与の昇順、列別名は"SALARY"とします。

※ヒント：テキスト「グループ関数」の項を参照しましょう。

演習 6-3

以下の構成の表があります。

従業員表（employees）

Column	Null？	Type
emp_id	NOT NULL	VARCHAR2(5)
first_name	NOT NULL	VARCHAR2(20)
last_name	NOT NULL	VARCHAR2(20)
address		VARCHAR2(30)

従業員 ID（emp_id）が重複している行を確認する SQL を作成してください。
表示する項目は、重複している従業員 ID と、重複行の件数です。

※ヒント：テキスト「グループ化」「グループ化の制限」の項を参照しましょう。

演習解答例

演習 6-1

```
SELECT
  department_id, MAX(salary),SUM(salary)
FROM
  employees
GROUP BY
  department_id
ORDER BY
  department_id
;
```

演習 6-2

```
SELECT
  LISTAGG(salary, ',') WITHIN GROUP (ORDER BY salary)
"SALARY"
FROM
  employees
WHERE
  department_id = '10001'
;
```

演習 6-3

```
SELECT
  emp_id , COUNT(emp_id)
FROM
  employees
GROUP BY
  emp_id
HAVING
  COUNT(emp_id) > 1
;
```

7. 複数の表からのデータの表示

演習 7-1

以下の構成の表があります。

顧客表（Customer）

Column	Null？	Type
customer_id	NOT NULL	VARCHAR2(10)
first_name	NOT NULL	VARCHAR2(20)
last_name	NOT NULL	VARCHAR2(20)
address	NOT NULL	VARCHAR2(50)
email		VARCHAR2(50)
phone_number	NOT NULL	VARCHAR2(15)
age		NUMBER(3)
note		VARCHAR2(50)

購入履歴表（History）

Column	Null？	Type
history_id	NOT NULL	VARCHAR2(10)
customer_id	NOT NULL	VARCHAR2(10)
goods_id	NOT NULL	VARCHAR2(10)
purchase_date	NOT NULL	DATE

商品購入時に、顧客表の顧客 ID（customer_id）が購入履歴表の顧客 ID（customer_id）列に登録され、かつ購入日時が購入履歴表の購入日時列（purchase_date）が登録されます。

2021 年 3 月 1 日～2021 年 3 月 31 日までに商品を購入した顧客のリストを表示する SELECT 文を作成してください。

表示情報は、

顧客 ID（customer_id）、氏名（first_name と last_name を半角スペースで連結）、住所（address）、電話番号（phone_number）、購入日時（purchase_date）

とします。表示順は顧客 ID の昇順とします。

また、購入履歴表に存在する customer_id は、顧客表に必ず存在するものとします。

※ヒント：テキスト「内部結合」の項を参照しましょう。

演習 7-2

以下の構成の表があります。

顧客表（Customer）

Column	Null？	Type
customer_id	NOT NULL	VARCHAR2(10)
first_name	NOT NULL	VARCHAR2(20)
last_name	NOT NULL	VARCHAR2(20)
address	NOT NULL	VARCHAR2(50)
email		VARCHAR2(50)
phone_number	NOT NULL	VARCHAR2(15)
age		NUMBER(3)
note		VARCHAR2(50)

購入履歴表（History）

Column	Null？	Type
history_id	NOT NULL	VARCHAR2(10)
customer_id	NOT NULL	VARCHAR2(10)
goods_id	NOT NULL	VARCHAR2(10)
purchase_date	NOT NULL	DATE

商品購入時に、顧客表の顧客 ID（customer_id）が購入履歴表の顧客 ID（customer_id）列に登録され、かつ購入日時が購入履歴表の購入日時列（purchase_date）が登録されます。
顧客の購入履歴を調べるための SELECT 文を作成してください。
表示情報は、
顧客 ID（customer_id)、氏名（first_name と last_name を半角スペースで連結)、住所（address)、電話番号（phone_number)、購入日時（purchase_date）
とします。
また、顧客情報は登録されていますが、一度も購入履歴がない顧客の情報も表示します。

※ヒント：テキスト「外部結合」の項を参照しましょう。

演習 7-3

以下の構成の表があります。

顧客表（Customer）

Column	Null？	Type
customer_id	NOT NULL	VARCHAR2(10)
first_name	NOT NULL	VARCHAR2(20)
last_name	NOT NULL	VARCHAR2(20)
address	NOT NULL	VARCHAR2(50)
email		VARCHAR2(50)
phone_number	NOT NULL	VARCHAR2(15)
age		NUMBER(3)
note		VARCHAR2(50)

商品（Goods）

Column	Null？	Type
goods_id	NOT NULL	VARCHAR2(10)
goods_name	NOT NULL	VARCHAR2(50)
price	NOT NULL	NUMBER(10)
maker_id	NOT NULL	VARCHAR2(5)

購入履歴表（History）

Column	Null？	Type
history_id	NOT NULL	VARCHAR2(10)
customer_id	NOT NULL	VARCHAR2(10)
goods_id	NOT NULL	VARCHAR2(10)
purchase_date	NOT NULL	DATE

商品購入時に、顧客表の顧客 ID（customer_id）が購入履歴表の顧客 ID（customer_id）列に登録され、かつ購入日時が購入履歴表の購入日時列（purchase_date）が登録されます。
顧客の購入履歴を調べるための SELECT 文を作成してください。
表示情報は、
顧客 ID（customer_id)、氏名（first_name と last_name を半角スペースで連結)、商品名（goods_name)、購入日時（purchase_date）
とします。
また、顧客情報は登録されていますが、一度も購入履歴がない顧客の情報も表示します。

※ヒント：テキスト「外部結合」「三つ以上の表の結合」の項を参照しましょう。

演習解答例

演習 7-1

```
SELECT
  c.customer_id,
  c.first_name || ' ' || c.last_name,
  c.address,
  c.phone_number,
  h.purchase_date
FROM
  customer c
  INNER JOIN history h
    ON c.customer_id = h.customer_id AND
       h.purchase_date BETWEEN TO_DATE('2021-03-01','yyyy-mm-
dd') AND TO_DATE('2021-03-31','yyyy-mm-dd')
ORDER BY c.customer_id
;
```

演習 7-2

```
SELECT
  customer_id,
  first_name || ' ' || last_name,
  address,
  phone_number,
  purchase_date
FROM
  customer
  LEFT JOIN history
    USING(customer_id)
;
```

演習 7-3

```
SELECT
  customer_id,
  first_name || ' ' || last_name,
  goods_name,
  purchase_date
FROM
  customer
  LEFT JOIN history
    USING(customer_id)
  LEFT JOIN goods
    USING(goods_id)
;
```

8. 副問合せを使用した問合せの解決

演習 8-1

以下の構成の表があります。

商品（Goods）

Column	Null？	Type
goods_id	NOT NULL	VARCHAR2(10)
goods_name	NOT NULL	VARCHAR2(50)
price	NOT NULL	NUMBER(10)
maker_id	NOT NULL	VARCHAR2(5)

売上（Sales）

Column	Null？	Type
sales_id	NOT NULL	VARCHAR2(10)
goods_id	NOT NULL	VARCHAR2(10)
purchase_date	NOT NULL	DATE

売上があった商品は売上表に登録されます。
過去に一度も売上がない商品を表示する SELECT 文を作成してください。
表示情報は、商品 ID（goods_id）と商品名（goods_name）とします。

※ヒント：テキスト「副問合せとは」の項を参照しましょう。

演習 8-2

以下の構成の表があります。

従業員表（Employees）

Column	Null？	Type
department_id	NOT NULL	VARCHAR2(5)
emp_id	NOT NULL	VARCHAR2(5)
first_name	NOT NULL	VARCHAR2(20)
last_name	NOT NULL	VARCHAR2(20)
salary		NUMBER(10)

全部門の中で、平均給与が最も高い部門を表示する SELECT 文を作成してください。
表示情報は、部門 ID（department_id）とその部門の平均給与とします。表示順は部門 ID の昇順とします。

※ヒント：テキスト「グループ化と集計」「グループ関数」「単一行副問合せ」の項を参照

しましょう。

演習 8-3

以下の構成の表があります。

顧客表（Customer）

Column	Null？	Type
customer_id	NOT NULL	VARCHAR2(10)
first_name	NOT NULL	VARCHAR2(20)
last_name	NOT NULL	VARCHAR2(20)
address	NOT NULL	VARCHAR2(50)
email		VARCHAR2(50)
phone_number	NOT NULL	VARCHAR2(15)
age		NUMBER(3)
note		VARCHAR2(50)

購入履歴表（History）

Column	Null？	Type
history_id	NOT NULL	VARCHAR2(10)
customer_id	NOT NULL	VARCHAR2(10)
goods_id	NOT NULL	VARCHAR2(10)
purchase_date	NOT NULL	DATE

商品購入時に、顧客表の顧客 ID（customer_id）が購入履歴表の顧客 ID（customer_id）列に登録され、かつ購入日時が購入履歴表の購入日時列（purchase_date）が登録されます。
年齢（age）が 50 歳以上の顧客が購入した、商品 ID（goods_id）と購入日（purchase_date）を表示する SELECT 文を作成してください。
表示順は購入日時の降順とします。

※ヒント：テキスト「複数行副問合せ」の項を参照しましょう。

演習解答例

演習 8-1

```
SELECT
  goods_id,goods_name
FROM
  goods
WHERE
  NOT EXISTS
    (SELECT
       *
     FROM
       sales
     WHERE
       sales.goods_id = goods.goods_id
)
;
```

演習 8-2

```
SELECT
  department_id, AVG(salary)
FROM
  employees
GROUP BY
  department_id
HAVING
  AVG(salary) = (
    SELECT
      MAX(AVG(salary))
    FROM
      employees
    GROUP BY
      department_id
  )
ORDER BY
  department_id
;
```

演習 8-3

```
SELECT
  goods_id,purchase_date
FROM
  history
WHERE
  customer_id IN (
    SELECT
      customer_id
```

```
    FROM
      customer
    WHERE
      age >= 50
  )
ORDER BY purchase_date DESC
;
```

9. 集合演算子の使用

演習 9-1

以下の構成の表があります。

従業員表（Employees）

Column	Null？	Type
department_id	NOT NULL	VARCHAR2(5)
emp_id	NOT NULL	VARCHAR2(5)
first_name	NOT NULL	VARCHAR2(20)
last_name	NOT NULL	VARCHAR2(20)
salary	NOT NULL	NUMBER(10)
retirement_date		DATE

退職従業員表（Employees_retire）

Column	Null？	Type
department_id	NOT NULL	VARCHAR2(5)
emp_id	NOT NULL	VARCHAR2(5)
first_name	NOT NULL	VARCHAR2(20)
last_name	NOT NULL	VARCHAR2(20)
salary	NOT NULL	NUMBER(10)
retirement_date		DATE

従業員表には現在所属している従業員データが管理されています。退職従業員表には退職した従業員データが管理されています。従業員が退職した場合は、退職日の定時後に従業員表の退職日列（retirement_date）に退職日が設定されて、月末のバッチ処理で退職従業員表にデータが移動されます。
最終所属部署が総務部（department_id:10001）の従業員を、現従業員と退職従業員と合わせて表示する SELECT 文を作成してください。
表示情報は、従業員 ID（emp_id）、氏名（first_name と last_name を半角スペースで連結）、退職日（retirement_date）とします。
従業員表と退職従業員表に従業員 ID の重複は無いものとします。

※ヒント：テキスト「UNION 演算子」の項を参照しましょう。

演習 9-2

以下の構成の表があります。

従業員表（Employees）

Column	Null？	Type
department_id	NOT NULL	VARCHAR2(5)
emp_id	NOT NULL	VARCHAR2(5)
first_name	NOT NULL	VARCHAR2(20)
last_name	NOT NULL	VARCHAR2(20)
salary	NOT NULL	NUMBER(10)
retirement_date		DATE

退職従業員表（Employees_retire）

Column	Null？	Type
department_id	NOT NULL	VARCHAR2(5)
emp_id	NOT NULL	VARCHAR2(5)
first_name	NOT NULL	VARCHAR2(20)
last_name	NOT NULL	VARCHAR2(20)
salary	NOT NULL	NUMBER(10)
retirement_date		DATE

従業員表には現在所属している従業員データが管理されています。退職従業員表には退職した従業員データが管理されています。従業員が退職した場合は、退職日の定時後に従業員表の退職日列（retirement_date）に退職日が設定されて、月末のバッチ処理で退職従業員表にデータが移動されます。
システム異常で、月末実行バッチが作動せずに手動で退職従業員データの移動を行った際、誤って移動ではなくコピーをしてしまい、従業員表と退職従業員表に同一のデータが存在する状態となってしまいました。この重複データを抽出するための SELECT 文を作成してください。表示情報は、従業員 ID（emp_id）、氏名（first_name と last_name を半角スペースで連結）、退職日（retirement_date）とします。

※ヒント：テキスト「INTERSECT 演算子」の項を参照しましょう。

演習 9-3

以下の構成の表があります。

従業員表（Employees）

Column	Null？	Type
department_id	NOT NULL	VARCHAR2(5)
emp_id	NOT NULL	VARCHAR2(5)
first_name	NOT NULL	VARCHAR2(20)
last_name	NOT NULL	VARCHAR2(20)
salary	NOT NULL	NUMBER(10)
retirement_date		DATE

従業員バックアップ表（Employees_bak）

Column	Null？	Type
department_id	NOT NULL	VARCHAR2(5)
emp_id	NOT NULL	VARCHAR2(5)
first_name	NOT NULL	VARCHAR2(20)
last_name	NOT NULL	VARCHAR2(20)
salary	NOT NULL	NUMBER(10)
retirement_date		DATE

従業員表のデータを月末のバッチ処理で従業員バックアップ表にコピーしています。
正常コピーできたかどうかの確認のため、バッチに両方の表の件数が一致しているかを確認するためのコードを追加します。
集合演算子を使用して、件数差異確認用の SELECT 文を作成してください。
なお、従業員バックアップ表は、コピー前に空の状態にされているものとします。

※ヒント：テキスト「集合演算子の使用」の項を参照しましょう。

演習解答例

演習 9-1

```
SELECT
  emp_id,first_name || ' ' || last_name,retirement_date
FROM
  employees
WHERE
  department_id = '10001'
UNION
SELECT
  emp_id,first_name || ' ' || last_name,retirement_date
FROM
  employees_retire
WHERE
  department_id = '10001'
;
```

演習 9-2

```
SELECT
  emp_id,first_name || ' ' || last_name,retirement_date
FROM
  employees
INTERSECT
SELECT
  emp_id,first_name || ' ' || last_name,retirement_date
FROM
  employees_retire
;
```

演習 9-3

```
SELECT
  COUNT(*)
FROM
(
  SELECT * FROM employees
  MINUS
  SELECT * FROM employees_bak
)
;
```

10．DML 文を使用した表の管理

演習 10-1

以下の構成の表があります。

アンケート表（Questionnaire）

Column	Null？	Type
customer_id	NOT NULL	VARCHAR2(10)
questionnaire	NOT NULL	VARCHAR2(100)

売上表（Sales）

Column	Null？	Type
sales_id	NOT NULL	VARCHAR2(10)
goods_id	NOT NULL	VARCHAR2(10)
customer_id	NOT NULL	VARCHAR2(10)
questionnaire		VARCHAR2(100)
sales_date	NOT NULL	DATE

顧客は、商品購入時に、ショッピングサイトの操作性についてのアンケートを入力することができます。（アンケート入力は任意）
商品購入時にアンケートを入力した場合、アンケート表に対して、
・アンケート表に顧客 ID（customer_id）が存在する場合は、売上表のアンケート（questionnaire）を、アンケート表の該当顧客のアンケート（questionnaire）に上書き更新
・アンケート表に顧客 ID（customer_id）が存在しない場合は、売上表の顧客 ID（customer_id）とアンケート（questionnaire）を、アンケート表に新規登録
とするような DML を、一文で作成してください。
なお、同じ顧客が複数のアンケートを入力している場合は、売上日（sales_date）が新しい情報を更新するものとします。
また、作成した文の結果を確認するために、作成した文の実行前後の questionnaire 表の内容を確認する SELECT 文も合わせて作成してください（全列全行表示）。

※ヒント：テキスト「MERGE 文」の項を参照しましょう。

演習 10-2

以下の構成の表があります。

従業員表（Employees）

Column	Null？	Type
department_id	NOT NULL	VARCHAR2(5)
emp_id	NOT NULL	VARCHAR2(5)
first_name	NOT NULL	VARCHAR2(20)
last_name	NOT NULL	VARCHAR2(20)
salary	NOT NULL	NUMBER(10)

給与ランク表（Salary_rank_high,Salary_rank_middle,Salary_rank_low）

Column	Null？	Type
department_id	NOT NULL	VARCHAR2(5)
emp_id	NOT NULL	VARCHAR2(5)
salary	NOT NULL	NUMBER(10)

以下の条件でランク付けをします。

・従業員表の給与（salary）が 500000 以上
　従業員表から該当従業員情報を salary_rank_high 表に登録
・従業員表の給与（salary）が 300000 以上、500000 未満
　従業員表から該当従業員情報を salary_rank_middle 表に登録
・従業員表の給与（salary）が 300000 未満
　従業員表から該当従業員情報を salary_rank_low 表に登録

上記の条件を満たすための SQL を一文で作成してください。
また、作成した文の結果を確認するために、作成した文の実行前後の salary_rank_high 表、salary_rank_middle 表、salary_rank_low 表の内容を確認する SELECT 文も合わせて作成してください（全列全行表示）。

※ヒント：テキスト「マルチテーブル・インサート」の項を参照しましょう。

演習 10-3

以下の構成の表があります。

従業員表（Employees）

Column	Null？	Type
department_id	NOT NULL	VARCHAR2(5)
emp_id	NOT NULL	VARCHAR2(5)
first_name	NOT NULL	VARCHAR2(20)
last_name	NOT NULL	VARCHAR2(20)
salary		NUMBER(10)

従業員 ID（emp_id）：10003 の社員の評価が高かったため、給与アップ更新用 DML を実行しましたが、実行文に誤りがあり、salary の値を誤って更新し、確定してしまいました。
正しい値（270000）に更新し直すための DML を作成してください。
また、誤りを防止するため、更新文の実行前後に確認用問合せ文を実行し、更新前後を目視確認した上で、最後に確定する文を合わせて作成してください。

※ヒント：テキスト「行の更新」「トランザクションの開始と終了」の項を参照しましょう。

演習解答例

演習 10-1

```
SELECT
  *
FROM
  questionnaire
;

MERGE INTO
  questionnaire q
USING (
  SELECT
    customer_id,questionnaire
  FROM
    sales
  WHERE
    questionnaire IS NOT NULL
) s
ON (
  q.customer_id = s.customer_id
)
WHEN MATCHED
  THEN
    UPDATE
      SET
        q.questionnaire = s.questionnaire
WHEN NOT MATCHED
  THEN
    INSERT (
      q.customer_id, q.questionnaire
    )
    VALUES (
      s.customer_id, s.questionnaire
    )
;
SELECT
  *
FROM
  questionnaire
;
```

演習 10-2

```
SELECT
  *
FROM
  salary_rank_high
;

SELECT
```

```
  *
FROM
  salary_rank_middle
;

SELECT
  *
FROM
  salary_rank_low
;

INSERT
  ALL
    WHEN salary >= 500000
      THEN INTO salary_rank_high values(department_id,
emp_id, salary)
    WHEN salary >= 300000 AND salary < 500000
      THEN INTO salary_rank_middle values(department_id,
emp_id, salary)
    WHEN salary < 300000
      THEN INTO salary_rank_low values(department_id, emp_id,
salary)
  SELECT
    department_id,emp_id,salary
  FROM
    employees
;
SELECT
  *
FROM
  salary_rank_high
;

SELECT
  *
FROM
  salary_rank_middle
;

SELECT
  *
FROM
  salary_rank_low
;
```

演習 10-3

```
SELECT
  emp_id,salary
FROM
  employees
WHERE
  emp_id = '10003'
;
```

```
UPDATE
  employees
SET
  salary = '280000'
WHERE
  emp_id = '10003'
;

SELECT
  emp_id,salary
FROM
  employees
WHERE
  emp_id = '10003'
;

COMMIT
;
```

11．DDL による表とその関係の管理

演習 11-1

商品管理システムを新たに構築するため、以下の表を作成する DDL 文を作成してください。列名はローマ字でも英単語でも構いませんが、長くても 20 文字以内の列名としてください。

仕入先（Suppliers）

カラム	型	型のサイズ	制約
仕入先 ID	可変長文字データ型	5	主キー制約
仕入先名	可変長文字データ型	50	NOT NULL 制約
住所	可変長文字データ型	40	-
電話番号	可変長文字データ型	11	-

商品（Products）

カラム	型	型のサイズ	制約
商品 ID	可変長文字データ型	5	主キー制約
商品名	可変長文字データ型	50	NOT NULL 制約
仕入先 ID	可変長文字データ型	5	外部キー制約（Suppliers 表の仕入先 ID）
価格	数値型	10	-

また、上記の２つの表を作成した後に、仕入先表を削除しようとした場合の結果を確認してください。

※ヒント：テキスト「表の作成：CREATE TABLE（制約付き）」の項を参照しましょう。

演習 11-2

以下の構成の表があります。

顧客表（Customer）

Column	Null？	Type
customer_id	NOT NULL	VARCHAR2(10)
first_name	NOT NULL	VARCHAR2(20)
last_name	NOT NULL	VARCHAR2(20)
address	NOT NULL	VARCHAR2(50)
email		VARCHAR2(50)
phone_number	NOT NULL	VARCHAR2(15)
age		NUMBER(3)
note		VARCHAR2(50)

将来的な顧客予想情報を管理するため、顧客表から年齢（age）が 18 以上 30 以下の顧客情報を抽出し、新たに「Customer_future」表を作成するための DDL を作成してください。
なお、Customer_future 表の構成は、Customer 表と同じとします。

※ヒント：テキスト「表の作成：CREATE TABLE（副問合せ）」の項を参照しましょう。

演習 11-3

売上表（Sales）から売上日時バックアップ表にデータをバックアップした後、バックアップ後の件数チェックで問題なければ、売上表データを全消去する夜間日次バッチを作成します。
このバッチ内で使用する SQL 文について、性能面を考慮して、売上表データの全消去を行う最適な文を作成してください。
なお、表の制約や索引情報は変更したくないため、DROP→CREATE 文は使用しないこととします。
また、全消去前後の売上表の件数を確認する文を合わせて作成してください。

※ヒント：テキスト「表の削除と内容消去」の項を参照しましょう。

演習 11-4

以下の構成の表があります。

顧客表（Customer）

Column	Null？	Type
customer_id	NOT NULL	VARCHAR2(10)
first_name	NOT NULL	VARCHAR2(20)
last_name	NOT NULL	VARCHAR2(20)
address	NOT NULL	VARCHAR2(40)
email		VARCHAR2(50)
phone_number	NOT NULL	VARCHAR2(15)
age		NUMBER(3)
note		VARCHAR2(50)

顧客管理システムのリリース後、顧客情報の入力時に、顧客表の住所列のサイズが足りないというケースが多発したため、サイズを拡張します。
顧客表の住所列サイズを、40 から 60 に変更するための DDL 文を作成してください。
また、拡張文の実行前後の表構造とデータ内容についての差異確認を行いますので、その文も合わせて作成してください。
なお、事前に顧客表のバックアップは実施済とします。

※ヒント：テキスト「表の構造の変更」「表構造の表示」の項を参照しましょう。

演習解答例

演習 11-1

```
CREATE TABLE suppliers
(
  supplier_id  VARCHAR2(5) PRIMARY KEY,
  suppliers_name VARCHAR2(50) NOT NULL,
  address  VARCHAR2(40),
  phone_number  VARCHAR2(11)
)
;
```

```
CREATE TABLE products
(
  product_id  VARCHAR2(5) PRIMARY KEY,
  products_name VARCHAR2(50) NOT NULL,
  supplier_id  VARCHAR2(5),
  price NUMBER(10),
  FOREIGN KEY(supplier_id) REFERENCES suppliers(supplier_id)
)
;
```

上記２表が存在している場合に仕入先表を削除する場合の確認文

```
DROP TABLE suppliers;

ORA-02449: unique/primary keys in table referenced by foreign
keys
```

演習 11-2

```
CREATE TABLE
  customer_future
AS
SELECT
  *
FROM
  customer
WHERE
  age BETWEEN 18 AND 30
;
```

演習 11-3

```
SELECT
  COUNT(*)
FROM
  sales
;

TRUNCATE TABLE
  sales
;

SELECT
  COUNT(*)
FROM
  sales
;
```

演習 11-4

```
SELECT
  *
FROM
  customer
;

DESC customer;

ALTER TABLE customer MODIFY (
  address VARCHAR2(60)
)
;

DESC customer;

SELECT
  *
FROM
```

```
  customer
;
```

12．ビューの管理

演習 12-1

以下の構成の表があります。

商品（Goods）

Column	Null？	Type
goods_id	NOT NULL	VARCHAR2(10)
goods_name	NOT NULL	VARCHAR2(50)
price	NOT NULL	NUMBER(10)
maker_id	NOT NULL	VARCHAR2(5)

売上（Sales）

Column	Null？	Type
sales_id	NOT NULL	VARCHAR2(10)
goods_id	NOT NULL	VARCHAR2(10)
num_of_sales	NOT NULL	NUMBER(5)
purchase_date	NOT NULL	DATE

売上があった商品は売上表に登録されます。
日次バッチで、売上があった商品名（goods_name）、その商品の価格（price）、その商品の売上数（num_of_sales）、その商品の価格（price）と売上数（num_of_sales）を掛けた値（列別名：amount）をレポートとして出力します。
上記レポートを表示するためのビューを作成してください（ビュー名：v_goods_sales）。
作成するビューは読み取り専用とします。
また、作成したビューに対する SELECT 文を作成してください。表示情報はビューの全行全列とします。

※ヒント：テキスト「ビューの作成」「ビューへのアクセス」の項を参照しましょう。

演習 12-2

以下の構成の表があります。

従業員表（Employees）

Column	Null？	Type
department_id	NOT NULL	VARCHAR2(5)
emp_id	NOT NULL	VARCHAR2(5)
first_name	NOT NULL	VARCHAR2(20)
last_name	NOT NULL	VARCHAR2(20)
salary		NUMBER(10)

経理部（department_id：10001）のみの情報を抽出するビューを作成してください（ビュー名：v_employees_accounting）。
作成するビューには更新チェックオプションを付与するものとします。
また、作成したビューに対する SELECT 文を作成してください。表示情報はビューの全行全列とします。
また、上記ビューに対して、department_id：10002 のデータを Insert する SQL 文を作成して、実行結果について確認してください。

※ヒント：テキスト「ビューの作成」「ビューへのアクセス」の項を参照しましょう。

演習解答例

演習 12-1

```
CREATE OR REPLACE VIEW
  v_goods_sales
AS
  SELECT
    g.goods_name,
    g.price,
    s.num_of_sales,
    g.price * s.num_of_sales amount
  FROM
    goods g
    JOIN sales s USING(goods_id)
WITH READ ONLY
;
```

```
SELECT
  *
FROM
  v_goods_sales
;
```

演習 12-2

```
CREATE OR REPLACE VIEW
  v_employees_accounting
AS
  SELECT
    *
  FROM
    employees
  WHERE
    department_id = '10001'
WITH CHECK OPTION
;
```

```
SELECT
  *
FROM
  v_employees_accounting
;
```

```
INSERT INTO
  v_employees_accounting
VALUES ('10002', '10003','Taro','Suzuki',300000)
;

ORA-01402: view WITH CHECK OPTION where-clause violation…
```

13．索引とシノニムとシーケンスの管理

演習 13-1

以下の構成の表があります。

商品（Goods）

Column	Null？	Type
goods_id	NOT NULL	VARCHAR2(10)
goods_name	NOT NULL	VARCHAR2(50)
price	NOT NULL	NUMBER(10)
maker_id	NOT NULL	VARCHAR2(5)

商品名に対して索引を付与する文を作成してください。
また、索引の作成後、今後問合せ文の性能チューニングを行う予定です。
この準備のため、上記索引を不可視にする文も作成してください。

※ヒント：テキスト「索引の作成」の項を参照しましょう。

演習 13-2

以下の構成の表があります。

従業員表（Employees）

Column	Null？	Type
department_id	NOT NULL	VARCHAR2(5)
emp_id	NOT NULL	VARCHAR2(5)
first_name	NOT NULL	VARCHAR2(20)
last_name	NOT NULL	VARCHAR2(20)
salary	NOT NULL	NUMBER(10)
retirement_date		DATE

従業員表を所有していない他ユーザーに、従業員表へのアクセスを許可する必要がありますが、セキュリティの観点から、従業員表の名称を別名で公開します。別名オブジェクトを作成するための文を作成してください（別名称：emp）。なお、別名オブジェクトは上書き更新オプション付きで作成してください。
また、作成した別名オブジェクトに対する SELECT 文を作成してください。表示情報は別名オブジェクトの全行全列とします。

※ヒント：テキスト「シノニムの作成」の項を参照しましょう。

演習 13-3

以下の構成の表があります。

商品（Goods）

Column	Null？	Type
goods_id	NOT NULL	VARCHAR2(10)
goods_name	NOT NULL	VARCHAR2(50)
price	NOT NULL	NUMBER(10)
maker_id	NOT NULL	VARCHAR2(5)

商品 ID（goods_id）の値は、一意の値となり、かつ大量な値を生成する想定のため、性能を考えて「シーケンス」機能を使用して値を付与します。
このシーケンスを作成する文を作成してください。
シーケンスの仕様は以下の通りとします。

シーケンス名：goods_id_seq
上書き更新オプション指定有
初期値：101
増分値：1
最大値：1000000000
最小値：1
サイクル指定：無し
キャッシュ数：50

また、商品表に、作成したシーケンスを使用して、行を追加してください。
商品表に追加する内容は、以下の通りとします。
goods_id：goods_id_seq の新しい値（10 桁で、桁数が足りない場合は先頭からゼロ埋め）
goods_name：'apron’
price：980
maker_id：’10002’

また、追加した行の結果の確認を行いますので、追加前後で商品表の情報を表示する文も合わせて作成してください。表示情報は商品表の全行前列とします。

※ヒント：テキスト「シーケンスの作成」の項を参照しましょう。

演習解答例

演習 13-1

```
CREATE INDEX
  good_name_idx
ON
  goods(goods_name)
;
```

```
ALTER INDEX
  good_name_idx
INVISIBLE
;
```

演習 13-2

```
CREATE OR REPLACE SYNONYM
  emp
FOR
  employees
;
```

```
SELECT
  *
FROM
  emp
;
```

演習 13-3

```
CREATE SEQUENCE goods_id_seq
  START WITH 101
  INCREMENT BY 1
  MAXVALUE 1000000000
  MINVALUE 1
  NOCYCLE
  CACHE 50
;
```

```
SELECT
  *
FROM
  goods
;

INSERT INTO
```

```
  goods
VALUES (
  TO_CHAR(goods_id_seq.NEXTVAL,'FM0000000000'),
  'Apron',
  980,
  '10002'
)
;

SELECT
  *
FROM
  goods
;
```

14．ユーザーアクセスの制御

本章の演習問題はありません。
テキストの内容を確認してください。

15．データ・ディクショナリ・ビューを使用したオブジェクトの管理

演習 15-1

自ユーザーが所有する索引情報の一覧を表示する文を作成してください。

※ヒント：テキスト「データ・ディクショナリ・ビューの種類」「データ・ディクショナリ・ビュー一覧」の項を参照しましょう。

演習 15-2

自ユーザーが所有するオブジェクト情報の一覧を表示する文を作成してください。

※ヒント：テキスト「データ・ディクショナリ・ビューの種類」「データ・ディクショナリ・ビュー一覧」の項を参照しましょう。

演習 15-3

自ユーザーから参照可能な表情報の一覧を表示する文を作成してください。

※ヒント：テキスト「データ・ディクショナリ・ビューの種類」「データ・ディクショナリ・ビュー一覧」の項を参照しましょう。

演習解答例

演習 15-1

```
SELECT
  *
FROM
  user_indexes
;
```

※別解

```
SELECT
  *
FROM
  user_objects
WHERE
  OBJECT_TYPE = 'INDEX'
;
```

演習 15-2

```
SELECT
  *
FROM
  user_objects
;
```

演習 15-3

```
SELECT
  *
FROM
  all_tables
;
```

16．異なるタイムゾーンでのデータの管理

演習 16-1

以下の構成の表があります。

商品（Goods）

Column	Null？	Type
goods_id	NOT NULL	VARCHAR2(10)
goods_name	NOT NULL	VARCHAR2(30)
price	NOT NULL	NUMBER(10)
maker_id	NOT NULL	VARCHAR2(5)
made_date		TIMESTAMP

製造日（made_date）のデータを’YYYY/MM/DD’の形式で表示する SELECT 文を作成してください。表示情報は、名称（goods_name）、価格（price）、製造日（made_date）とします。

※ヒント：テキスト「データ型の変換関数」「日時データ型」の項を参照しましょう。

演習 16-2

DUAL 表から TIMESTAMP WITH TIME ZONE を表示する SELECT 文を作成してください。この際、表示する列別名を"TIMESTAMP WITH TIME ZONE"としてください。

※ヒント：テキスト「日時データ型」「日時データ型データ用タイムゾーン関数」の項を参照しましょう。

演習 16-3

以下の構成の表があります。

商品（Goods）

Column	Null？	Type
goods_id	NOT NULL	VARCHAR2(10)
goods_name	NOT NULL	VARCHAR2(30)
price	NOT NULL	NUMBER(10)
maker_id	NOT NULL	VARCHAR2(5)
made_date		DATE
warranty_period		INTERVAL YEAR TO MONTH

商品表に以下のデータを登録する INSERT 文を作成してください。

Column	登録する値	備考
goods_id	0000000001	
goods_name	TV	
price	69980	
maker_id	10001	
made_date	21-08-01	YY-MM-DD
warranty_period	1 年 6 ヶ月	

※ヒント：テキスト「日時データ型」「期間データ型」の項を参照しましょう。

演習解答例

演習 16-1

```
SELECT
  goods_name,price,TO_CHAR(made_date, 'YYYY/MM/DD')
FROM
  goods
;
```

演習 16-2

```
SELECT
  CURRENT_TIMESTAMP AS "TIMESTAMP WITH TIME ZONE"
FROM
  dual
;
```

演習 16-3

```
INSERT INTO
  goods
VALUES (
  '0000000001',
  'TV',
  69980,
  '10001',
  TO_DATE('21-08-01', 'YY-MM-DD'),
  INTERVAL '1-6' YEAR TO MONTH
)
;
```

【V-UP】ORACLE MASTER Silver SQL 2019 講座

演習問題

初版発行日：　2021年08月31日
最終更新日：
著　　作：　株式会社シンクスバンク
発　行　者：　株式会社シンクスバンク