



Oracle 追加問題

1 目次

1 目次.....	1
1 ユーザー作成、テーブル作成.....	2
2 CRUD 処理.....	3
2.1 レコードの登録.....	3
2.2 レコードの検索.....	4
2.3 レコードの更新.....	5
2.4 レコードの削除.....	6
3 演算子と述語.....	7
4 集約.....	10
5 並べ替え.....	11
6 結合.....	12
7 その他の機能(シーケンス、サブクエリ).....	14
7.1 シーケンス.....	14
7.2 サブクエリ.....	15
8 JDBC.....	17

1 ユーザー作成、テーブル作成

(1) ユーザー作成

以下の条件を踏まえたうえで、ユーザーを作成しなさい。

※作成する前に、system ユーザーで「ALTER SESSION SET CONTAINER = xepdb1;」のコマンドを実行してセッションを PDB に切り替えること。

【条件】

ユーザー名：OraclePracticeUser

パスワード：systemsss

(2) GRANT 文で利用できるショートカットを利用し、

「SELECT ANY DICTIONARY」権限以外のすべてのシステム権限、オブジェクト権限を付与しなさい。

(3) (1),(2)で作成したユーザーで、snack テーブルを作成しなさい。

No	論理名称	物理名称	データ型	桁数	制約
1	id	snack_id	NUMBER	3	PRIMARY KEY
2	商品名	snack_name	VARCHAR2	40	NOT NULL
3	発売日	release_date	DATE		NOT NULL
4	製造番号	serial_number	NUMBER	4	NOT NULL

2 CRUD 処理

2.1 レコードの登録

(1) 以下の実行結果と同じになるように、snack テーブルに以下のデータを登録しなさい。

SNACK_ID	SNACK_NAME	RELEASE_DATE	SERIAL_NUMBER
1	キャラメル	20-10-19	1111
2	ドリアンチップス	20-10-20	2222
3	チョコ	20-10-21	1111
4	チューイングガム	20-10-22	3333
5	野菜チョコ	20-10-23	4444
6	クリームスティック	20-10-24	4444
7	プロテインバー	20-10-25	5555
8	7変化の飴	20-10-26	3333
9	カリアゲクン-焼肉味-	20-10-27	5555
10	醤油せんべい	20-10-27	5555

2.2 レコードの検索

(1)snack テーブルのすべてのレコードを検索し、実行結果と同じになるようにしなさい。

SNACK_ID	SNACK_NAME	RELEASE_DATE	SERIAL_NUMBER
1	キャラメル	20-10-19	1111
2	ドリアンチップス	20-10-20	2222
3	チョコ	20-10-21	1111
4	チューイングガム	20-10-22	3333
5	野菜チョコ	20-10-23	4444
6	クリームスティック	20-10-24	4444
7	プロテインバー	20-10-25	5555
8	7変化の飴	20-10-26	3333
9	カリアゲクン-焼肉味-	20-10-27	5555
10	醤油せんべい	20-10-27	5555

(2)serial_number が 1111 のレコードをすべて出力しなさい。

以下の実行結果を参考にすること。

SNACK_ID	SNACK_NAME	RELEASE_DATE	SERIAL_NUMBER
1	キャラメル	20-10-19	1111
3	チョコ	20-10-21	1111

(3)serial_number が 5555 の商品名を出力しなさい。

以下実行結果を参考にすること。

SNACK_NAME
プロテインバー
カリアゲクン-焼肉味-
醤油せんべい

2.3 レコードの更新

(1) シリアルナンバーが 1111 のレコードをすべて 9999 に書き換えて下さい。

以下の実行結果を参考にすること。

SNACK_ID	SNACK_NAME	RELEASE_DATE	SERIAL_NUMBER
1	キャラメル	20-10-19	9999
2	ドリアンチップス	20-10-20	2222
3	チョコ	20-10-21	9999
4	チューイングガム	20-10-22	3333
5	野菜チョコ	20-10-23	4444
6	クリームスティック	20-10-24	4444
7	プロテインバー	20-10-25	5555
8	7変化の飴	20-10-26	3333
9	カリアゲクン-焼肉味-	20-10-27	5555
10	醤油せんべい	20-10-27	5555

(2) 以下の商品名を変更後のデータに更新して下さい。

商品名：ドリアンチップス → シリアルナンバー変更後：1111

クリームスティック → 発売日変更後：2022/07/28

SNACK_ID	SNACK_NAME	RELEASE_DATE	SERIAL_NUMBER
1	キャラメル	20-10-19	9999
2	ドリアンチップス	20-10-20	1111
3	チョコ	20-10-21	9999
4	チューイングガム	20-10-22	3333
5	野菜チョコ	20-10-23	4444
6	クリームスティック	22-07-28	4444
7	プロテインバー	20-10-25	5555
8	7変化の飴	20-10-26	3333
9	カリアゲクン-焼肉味-	20-10-27	5555
10	醤油せんべい	20-10-27	5555

2.4 レコードの削除

- (1) シリアルナンバーが 5555 のレコードをすべて削除しなさい。

以下の実行結果を参考にする。

SNACK_ID	SNACK_NAME	RELEASE_DATE	SERIAL_NUMBER
1	キャラメル	20-10-19	9999
2	ドリアンチップス	20-10-20	1111
3	チョコ	20-10-21	9999
4	チューイングガム	20-10-22	3333
5	野菜チョコ	20-10-23	4444
6	クリームスティック	22-07-28	4444
8	7変化の飴	20-10-26	3333

- (2) 発売日が 10/20 の商品を削除しなさい。

以下の実行結果を参考にする。

SNACK_ID	SNACK_NAME	RELEASE_DATE	SERIAL_NUMBER
1	キャラメル	20-10-19	9999
3	チョコ	20-10-21	9999
4	チューイングガム	20-10-22	3333
5	野菜チョコ	20-10-23	4444
6	クリームスティック	22-07-28	4444
8	7変化の飴	20-10-26	3333

3 演算子と述語

(1) 以下のテーブルの作成、データの挿入を行いなさい。

No	論理名称	物理名称	データ型	桁数	制約
1	商品番号	drink_id	NUMBER	3	PRIMARY KEY
2	商品名	drink_name	VARCHAR2	40	NOT NULL
3	発売日	release_date	DATE		
4	価格	sel_price	NUMBER	4	NOT NULL

--テーブル作成

```
CREATE TABLE drink(  
  drink_id NUMBER(3) PRIMARY KEY,  
  drink_name VARCHAR2(40) NOT NULL,  
  release_date DATE,  
  sel_price NUMBER(4) NOT NULL  
);
```

--データ挿入

```
INSERT INTO drink VALUES(1,'コーラ','2020/10/21',100);  
INSERT INTO drink VALUES(2,'オレンジ','2020/10/22',200);  
INSERT INTO drink VALUES(3,'おいしい水','2020/10/22',130);  
INSERT INTO drink VALUES(4,'ウーロン茶','2020/10/22',130);  
INSERT INTO drink VALUES(5,'微妙な味！レモン水','2020/10/25',190);  
INSERT INTO drink VALUES(6,'海洋ミネラル麦茶','2020/10/26',150);  
INSERT INTO drink VALUES(7,'スポーツドリンク','2020/10/26',150);  
COMMIT;
```


(2) 価格が 150 円以上のレコードをすべて表示しなさい。

実行結果を参考にする事。

DRINK_ID	DRINK_NAME	RELEASE_DATE	SEL_PRICE
2	オレンジ	20-10-22	200
5	微妙な味！レモン水	20-10-25	190
6	海洋ミネラル麦茶	20-10-26	150
7	スポーツドリンク	20-10-26	150

(3) 価格が 150 円以上、200 円以下の商品名と価格をすべて表示しなさい。

実行結果を参考にする事。

DRINK_NAME	SEL_PRICE
オレンジ	200
微妙な味！レモン水	190
海洋ミネラル麦茶	150
スポーツドリンク	150

(4) 10/25~31 に発売した商品名と発売日をすべて表示しなさい。

実行結果を参考にする事。

DRINK_NAME	RELEASE_DATE
微妙な味！レモン水	20-10-25
海洋ミネラル麦茶	20-10-26
スポーツドリンク	20-10-26

- (5) 「ー」が入っている商品名をすべて検索しなさい。
実行結果を参考にすること。

DRINK_NAME
コーラ
ウーロン茶
スポーツドリンク

-
- (6) 実行結果と同じになるように検索を行いなさい。

DRINK_NAME
コーラ
ウーロン茶

4 集約

- (1) Drink テーブルの中で、登録されているすべてのレコードの数を表示しなさい。

実行結果は以下を参考にする

COUNT(*)
7

- (2) drink テーブルの価格の平均を表示しなさい。

実行結果は以下を参考にする

AVG(SEL_PRICE)
150

- (3) snack テーブルの中で、serial_number ごとのレコード数を表示しなさい。

実行結果は以下を参考にする。 (表示順は考慮しない)

SERIAL_NUMBER	COUNT(*)
2222	1
4444	2
1111	2
3333	2
5555	3

- (4) snack テーブルの中で、serial_number が 2 行以上のレコード数を表示しなさい。

実行結果は以下を参考にする。 (表示順は考慮しない)

SERIAL_NUMBER	COUNT(*)
4444	2
1111	2
3333	2
5555	3

5 並べ替え

- (1) snack テーブルの発売日が新しい順(数値が大きい順)に並べ替えて表示しなさい。
実行結果は以下を参考にすること。

SNACK_ID	SNACK_NAME	RELEASE_DATE	SERIAL_NUMBER
6	クリームスティック	22-07-28	4444
8	7変化の飴	20-10-26	3333
5	野菜チョコ	20-10-23	4444
4	チューイングガム	20-10-22	3333
3	チョコ	20-10-21	9999
1	キャラメル	20-10-19	9999

- (2) drink テーブルの発売日が古い順に並べて表示しなさい。
同じ日付の商品がある場合は id が大きい順に表示される形にすること。

DRINK_ID	DRINK_NAME	RELEASE_DATE	SEL_PRICE
1	コーラ	20-10-21	100
4	ウーロン茶	20-10-22	130
3	おいしい水	20-10-22	130
2	オレンジ	20-10-22	200
5	微妙な味！レモン水	20-10-25	190
7	スポーツドリンク	20-10-26	150
6	海洋ミネラル麦茶	20-10-26	150

- (3) drink テーブルの id を降順に並べ替えなさい。
商品名は別名で「DrinkId」という名前に変更しなさい。

DRINKID	DRINK_NAME
7	スポーツドリンク
6	海洋ミネラル麦茶
5	微妙な味！レモン水
4	ウーロン茶
3	おいしい水
2	オレンジ
1	コーラ

6 結合

(1) 以下のテーブルの作成、データの挿入と drink テーブルに変更を行いなさい。

【drink_genre テーブル】

No	論理名称	物理名称	データ型	桁数	制約
1	商品番号	genre_id	NUMBER	3	PRIMARY KEY
2	商品名	genre_name	VARCHAR2	40	NOT NULL

```
--テーブル作成
CREATE TABLE drink_genre(
genre_id NUMBER(3) PRIMARY KEY,
genre_name VARCHAR2(40) NOT NULL);

-- 【INSERT 文】
INSERT INTO drink_genre VALUES(1,'炭酸');
INSERT INTO drink_genre VALUES(2,'果実水');
INSERT INTO drink_genre VALUES(3,'お茶');

--コミット
COMMIT;
```

Drink テーブル(変更)

```
--テーブル変更
ALTER TABLE drink ADD genre_id NUMBER(3) REFERENCES drink_genre (genre_id);

--テーブル内のレコードを一旦削除
DELETE FROM drink;

--レコードを再追加
INSERT INTO drink VALUES(1,'コーラ','2020/10/21',100,1);
INSERT INTO drink VALUES(2,'オレンジ','2020/10/22',200,2);
INSERT INTO drink VALUES(3,'おいしい水','2020/10/22',130,null);
INSERT INTO drink VALUES(4,'ウーロン茶','2020/10/22',130,3);
INSERT INTO drink VALUES(5,'微妙な味！レモン水','2020/10/25',190,2);
INSERT INTO drink VALUES(6,'海洋ミネラル麦茶','2020/10/26',150,3);
INSERT INTO drink VALUES(7,'スポーツドリンク','2020/10/26',150,null);
COMMIT;
```

(2) drink テーブルと genre テーブルから以下の実験結果と同じになるように検索を行いなさい。

DRINKNAME	SEL_PRICE	GENRE_NAME	GENRE_ID
コーラ	100	炭酸	1
オレンジ	200	果実水	2
ウーロン茶	130	お茶	3
微妙な味！レモン水	190	果実水	2
海洋ミネラル麦茶	150	お茶	3

(3) drink テーブルと genre テーブルから以下の実験結果と同じになるように検索を行いなさい。

DRINKNAME	SEL_PRICE	GENRE_NAME	DRINK_ID
コーラ	100	炭酸	1
微妙な味！レモン水	190	果実水	5
オレンジ	200	果実水	2
海洋ミネラル麦茶	150	お茶	6
ウーロン茶	130	お茶	4
スポーツドリンク	150	(null)	7
おいしい水	130	(null)	3

7 その他の機能(シーケンス、サブクエリ)

7.1 シーケンス

No	論理名称	物理名称	データ型	桁数	制約
1	ジャンル id	genre_id	NUMBER	3	PRIMARY KEY
2	ジャンル名	genre_name	VARCHAR2	40	NOT NULL

(1) 以下のテーブルの作成を行いなさい。

```
--書籍テーブル作成
CREATE TABLE book(
book_id NUMBER(3) PRIMARY KEY,
book_name VARCHAR2(40) NOT NULL,
release_date DATE,
sel_price NUMBER(4) NOT NULL
);
```

(2) book_id にシーケンスをつけて、レコードが追加されるたびに番号が自動採番されるようにしなさい。

【シーケンス名】 book_id_seq

【オプション】 NOCACHE

(3)(2)で作成したシーケンスを使って以下のレコードを追加しなさい。

BOOK_ID	BOOK_NAME	RELEASE_DATE	SEL_PRICE
1	犬と僕の2000の約束	22-11-16	400
2	初めてのJavaSilver	00-04-10	480
3	おすすめの国内旅行	22-07-29	650
4	クラウド基礎/応用	21-10-30	420

(4) 作成したシーケンスを削除しなさい。

7.2 サブクエリ

(1) 以下のテーブルの作成、データの挿入を行いなさい。

Book_genre テーブル

No	論理名称	物理名称	データ型	桁数	制約
1	ジャンル id	genre_id	NUMBER	3	PRIMARY KEY
2	ジャンル名	genre_name	VARCHAR2	40	NOT NULL

```
--シーケンス作成
```

```
CREATE SEQUENCE genre_id_seq NOCACHE;
```

```
--書籍ジャンルテーブル
```

```
CREATE TABLE book_genre(  
genre_id NUMBER(3) PRIMARY KEY,  
genre_name VARCHAR2(40) NOT NULL  
);
```

```
--INSERT 文
```

```
INSERT INTO book_genre VALUES(genre_id_seq.NEXTVAL,'絵本');  
INSERT INTO book_genre VALUES(genre_id_seq.NEXTVAL,'技術');  
INSERT INTO book_genre VALUES(genre_id_seq.NEXTVAL,'娯楽');  
INSERT INTO book_genre VALUES(genre_id_seq.NEXTVAL,'ビジネス');  
INSERT INTO book_genre VALUES(genre_id_seq.NEXTVAL,'音楽');  
COMMIT;
```



```
--book テーブル変更
ALTER TABLE book ADD genre_id NUMBER(3) REFERENCES book_genre(genre_id);

--シーケンス削除
DROP SEQUENCE book_id_seq;
--テーブル内のレコードを一旦削除
DELETE FROM book;

--シーケンス再作成
CREATE SEQUENCE book_id_seq NOCACHE;

--レコードを再追加
INSERT INTO book VALUES(book_id_seq.NEXTVAL,'犬と僕の 2000 の約束','2022/11/16',400,1);
INSERT INTO book VALUES(book_id_seq.NEXTVAL,'初めての JavaSilver','2000/04/10',480,2);
INSERT INTO book VALUES(book_id_seq.NEXTVAL,'おすすめの国内旅行','2022/07/29',650,3);
INSERT INTO book VALUES(book_id_seq.NEXTVAL,'クラウド基礎/応用','2021/10/30',420,2);
COMMIT;
```

(2) 次のサブクエリを読みなさい。(解答はメモ等に記述しなさい。)

```
SELECT book_name
FROM (
  SELECT
    b.book_name,
    g.genre_id,
    g.genre_name
  FROM book b INNER JOIN book_genre g
  ON b.genre_id=g.genre_id
);
```

(3) book テーブルの、平均販売価格よりも高い書籍名を答えてください。
(ヒント：AVG 関数も使用します)

8 JDBC

(前提)

本問題では、1章で作成した OraclePracticeUser と、7章で作成した book,book_genre テーブルを使用していく。

Eclipse で以下のプロジェクトとクラスを作成しなさい。

プロジェクト名 : JDBC_practice

パッケージ名 : jp.co.sss.jdbc

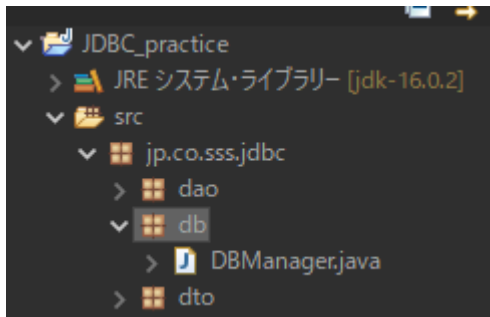
DB 周りのクラスをまとめるパッケージ : db(jp.co.sss.jdbc のサブパッケージとして作成しなさい)

DB に接続するクラス : DBManager(パッケージ db の中に作成すること)

CRUD 処理を行うクラスをまとめるパッケージ : dao(jp.co.sss.jdbc のサブパッケージ)

データを管理するクラスをまとめるパッケージ : dto(jp.co.sss.jdbc のサブパッケージ)

【パッケージ構成】



- (1)以下の条件に従いながら JDBC を用いて、book テーブルの id、書籍名、発売日、価格と genre テーブルからジャンル名を合わせて画面(コンソール)に表示しなさい。実行結果を参考にする事。
コンソールでの表示について、文字のずれは気にしないものとする。

<条件>

- ①作成するメソッド名は、selectAll()とすること。
- ②実行クラスは JdbcPractice1
- ③DB に接続する処理と SQL を実行する処理はすべてそのメソッド内に記述し、メインメソッドからは呼び出す形式で実行しなさい。

【実行結果】

```
コンソール × 🔍 検索
終了> JdbcPractice01 [Java アプリケーション] C:\Program Files\Java\jdk-16.0.2\bin\javaw.exe (2022/08/01 14:27:10 - 14:27:12)
書籍の検索を行います。
検索の形式を選んでください。
1:全件検索>1
書籍ID      書籍名      発売日      価格      ジャンル
1      犬と僕の2000の約束      2022-11-16      400      絵本
4      クラウド基礎/応用      2021-10-30      420      技術
2      初めてのJavaSilver      2000-04-10      480      技術
3      おすすめの国内旅行      2022-07-29      650      娯楽
```

(2) 以下の条件に従いながら JDBC を用いて、book テーブルにレコードを 1 件登録しなさい。
実行結果を参考にすること。コンソールでの表示について、文字のずれは気にしないものとする。

<条件>

- ①作成するメソッド名は、insertBook()とすること。
- ②実行クラスは JdbcPractice2
- ③DB に接続する処理と SQL を実行する処理はすべてそのメソッド内に記述し、メインメソッドからは呼び出す形式で実行しなさい。
- ④登録するデータは自由とする。

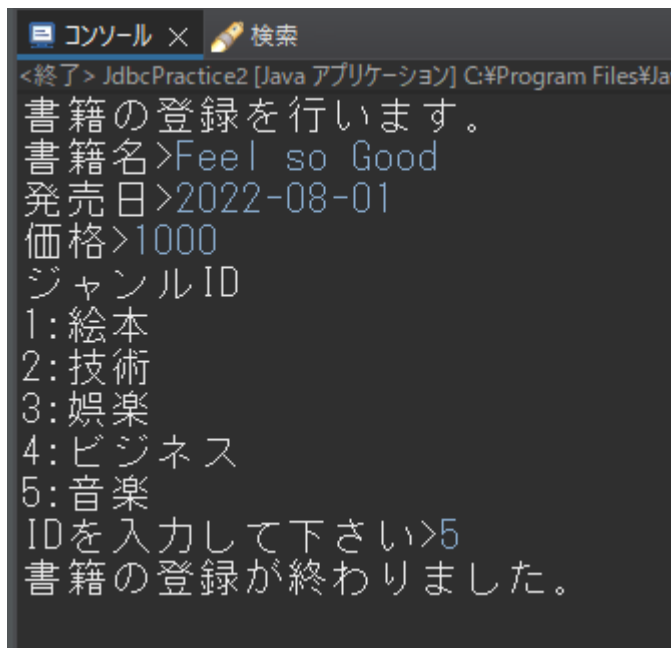
<ヒント>

日付登録について

DB に登録するため、Date クラスは java.sql.date をインポートして使用すること。
コンソールに入力する際にはすべて String 型になるためデータ登録の際は
Date 型に変換する必要がある。
文字列を java.sql の Date 型に変換する方法はインターネットで調べてみることにしよう。

【実行結果】

コンソール



```
<終了> JdbcPractice2 [Java アプリケーション] C:\Program Files\Ja
書籍の登録を行います。
書籍名>Feel so Good
発売日>2022-08-01
価格>1000
ジャンルID
1:絵本
2:技術
3:娯楽
4:ビジネス
5:音楽
IDを入力して下さい>5
書籍の登録が終わりました。
```

DB(表示順は順不同とする。気になる場合は、並べ替えをすること)

BOOK_ID	BOOK_NAME	RELEASE_DATE	SEL_PRICE	GENRE_ID
1	犬と僕の2000の約束	22-11-16	400	1
2	初めてのJavaSilver	00-04-10	480	2
5	Feel so Good	22-08-01	1000	5
3	おすすめの国内旅行	22-07-29	650	3
4	クラウド基礎/応用	21-10-30	420	2

(3)以下の条件に従いながら JDBC を用いて、book テーブルにレコードを 1 件更新しなさい。

<条件>

- ① 作成するメソッド名は、updateBook()とすること。
- ② 実行クラスは JdbcPractice3
- ③DB に接続する処理と SQL を実行する処理はすべてそのメソッド内に記述し、メインメソッドからは呼び出す形式で実行しなさい。
- ④更新するデータは book_id が 1 のデータとする。
以下の実行結果のように変更を行いなさい。

```
書籍の更新を行います。  
変更する書籍のIDを入力してください。>1  
書籍名>犬と私の1000の約束  
発売日>2022-08-02  
価格>1050  
ジャンルID  
1:絵本  
2:技術  
3:娯楽  
4:ビジネス  
5:音楽  
IDを入力して下さい>1  
書籍情報の更新が終わりました。
```

(4) 以下の条件に従いながら JDBC を用いて、book テーブルにレコードを 1 件削除しなさい。

<条件>

- ① 作成するメソッド名は、deleteBook()とすること。
- ② 実行クラスは JdbcPractice4
- ③DB に接続する処理と SQL を実行する処理はすべてそのメソッド内に記述し、メインメソッドからは呼び出す形式で実行しなさい。
- ④削除するデータは book_id が 4 のデータとする。
以下の実行結果のように変更を行いなさい。

```
書籍の削除を行います。  
削除する書籍のIDを入力してください。>4  
書籍情報の削除が終わりました。
```

【2】

(前提)

本問題では、1章で作成した OraclePracticeUser と、3~6章で作成した drink,genre テーブルを使用していく。

【drink テーブル】

No	論理名称	物理名称	データ型	桁数	制約
1	商品番号	drink_id	NUMBER	3	PRIMARY KEY
2	商品名	drink_name	VARCHAR2	40	NOT NULL
3	発売日	release_date	DATE		
4	価格	sel_price	NUMBER	4	NOT NULL

【genre テーブル】

No	論理名称	物理名称	データ型	桁数	制約
1	ジャンル id	genre_id	NUMBER	3	PRIMARY KEY
2	ジャンル名	genre_name	VARCHAR2	40	NOT NULL

(1)

以下の条件に従いながら JDBC を用いて、飲料名から検索をかけて発売日、価格、ジャンル名を画面(コンソール)に表示しなさい。実行結果を参考にすること。
コンソールでの表示について、文字のずれは気にしないものとする。

<条件>

- ① 作成するクラス(Dao) →DrinkDao
- ② 実行クラス →JdbcPractice5
- ③ 作成するメソッド名 →findByName()
- ④ DB に接続する処理と SQL を実行する処理はすべてそのメソッド内に記述し、メインメソッドからは呼び出す形式で実行しなさい。

【実行結果】

```

お好きな飲み物を注文してください。
飲み物>コーラ
飲料名          発売日          価格          ジャンル
コーラ          2020-10-21        100          炭酸
  
```

(2)

以下の条件に従いながら JDBC を用いて、飲料名と価格で検索をかけて飲料名、発売日、価格、ジャンル名を画面(コンソール)に表示しなさい。実行結果を参考にすること。
コンソールでの表示について、文字のずれは気にしないものとする。

<条件>

- ① 使用するクラス(Dao) →DrinkDao
- ② 実行クラス →JdbcPractice6
- ③ 作成するメソッド名 →findByNameAndPrice
- ④ DB に接続する処理と SQL を実行する処理はすべてそのメソッド内に記述し、メインメソッドからは呼び出す形式で実行しなさい。
- ⑤飲料水はあいまい検索できる。価格は、希望価格以上の金額で検索できるようにする。

【茶というキーワードで 100 円以上の飲料水の検索結果↓】

```

あなたにおすすめの飲み物をご提案します。
飲んでみたい飲料名を入力してください。>茶
希望価格を入力してください。>100
飲料名          発売日          価格          ジャンル
ウーロン茶          2020-10-22        130          お茶
海洋ミネラル麦茶    2020-10-26        150          お茶
  
```

(3)

以下の条件に従いながら JDBC を用いて、価格が 190~200 の商品をすべて 150 円に変更しなさい。
画面(コンソール)に表示しなさい。実行結果を参考にすること。
コンソールでの表示について、文字のずれは気にしないものとする。

<条件>

- ① 使用するクラス(Dao) → DrinkDao
- ② 実行クラス → JdbcPractice7
- ③ 作成するメソッド名 → updateDrinkPrice()
- ④ DB に接続する処理と SQL を実行する処理はすべてそのメソッド内に記述し、メインメソッドからは呼び出す形式で実行しなさい。

【実行結果】

```
飲料の価格変更を行います
変更する価格の範囲を入力してください。
最小金額>190
最大金額>200
190円から200円の範囲の価格を一括変更します。
変更金額を入力してください。>150
更新が完了しました。
```

【DB の結果(黄色→元値 190~200 円の飲料水)】

DRINK_ID	NAME	RELEASE_DATE	SEL_PRICE	GENRE_ID
1	コーラ	20-10-21	100	1
2	オレンジ	20-10-22	150	2
3	おいしい水	20-10-22	130	(null)
4	ウーロン茶	20-10-22	130	3
5	微妙な味！レモン水	20-10-25	150	2
6	海洋ミネラル麦茶	20-10-26	150	3
7	スポーツドリンク	20-10-26	150	(null)

(4)

以下の条件に従いながら JDBC を用いて、発売日が 10/26 の商品をすべて削除しなさい。

実行結果を参考にすること。

コンソールでの表示について、文字のずれは気にしないものとする。

<条件>

- ① 使用するクラス(Dao) → DrinkDao
- ② 実行クラス → JDBCPractice8
- ③ 作成するメソッド名 → deleteDrink()
- ④ DB に接続する処理と SQL を実行する処理はすべてそのメソッド内に記述し、メインメソッドからは呼び出す形式で実行しなさい。

<ヒント>

日付削除について

コンソールに入力する際にはすべて String 型になるためデータ登録の際は Date 型に変換する必要がある。

文字列を java.sql の Date 型に変換する方法はインターネットで調べてみることにしよう。

【実行結果】

```
<終了> JdbcPractice8 [Java アプリケーション] C:\Program Files\Java\jdk-16.0.2\bin\javaw.exe
販売終了の商品を削除します。
削除する日付入力してください。>2020-10-26
削除が完了しました。
```

【DB の結果(2020-10-26 のデータが消えていればよい)】

DRINK_ID	NAME	RELEASE_DATE	SEL_PRICE	GENRE_ID
1	コーラ	20-10-21	100	1
2	オレンジ	20-10-22	150	2
3	おいしい水	20-10-22	130	(null)
4	ウーロン茶	20-10-22	130	3
5	微妙な味！レモン水	20-10-25	150	2