

## 8. 情報の保存：データベース

- リレーショナルデータベース(RDB)の説明として正しいものを選択してください。(2つ選択)
  - SQLキーワードでは大文字と小文字を明確に区別する
  - データベース製品が異なってもSQLの記述方法は同じである
  - データベースにアクセス後、SQLではINSERT文、UPDATE文、DELETE文SELECT文を利用できる
  - 値のないセルがあっても表を作成することができる
  - スプレッドシートのようにデータベーステーブルの行の順番は決められている
- PDOの説明として誤っているものを選択してください。(1つ選択)
  - PHPの拡張モジュールとして実装されている
  - DBMSに対して共通のインターフェースを提供するため、移植性が高い
  - PDOを利用した場合でも、データベース特有の機能の中にはデータベース独自の関数でしかアクセスできないものがある
  - ブレースフォルダを用いたプリペアドステートメントをサポートしていない
- PDOによるデータベース接続を行う場合の記述として正しいものを選択してください。ただし、\$DSN変数には適切なDSN接続文字列が代入されるものとします。(1つ選択)
  - `db = pdo_connect(DSN);`
  - `db = new PDO(DSN, 'user-id', 'user-pass');`
  - `db = PDO :: connect(DSN, 'user-id', 'user-pass');`
  - `db = connect(DSN);`
- PDOのエラーモードの設定として誤っているものを選択してください。(1つ選択)
  - PDOには例外、サイレント、警告の3つのエラーモードがある
  - デフォルトは警告モードである
  - 次のメソッド呼び出しを行うと例外モードを設定できる  
`setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION)`
  - 次のメソッド呼び出しを行うと警告モードを設定できる  
`setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_WARNING)`
- 次のemployeesテーブルに新たな行を追加するSQL文として正しいものを選択してください。(1つ選択)

**【employeesテーブル】**

id	name
1	Tanaka
2	Yamada
3	Takahashi
4	Saito

- `INSERT INTO employees VALUES(5, 'Ito');`
- `INSERT INTO employees SET(5, 'Ito');`
- `ADD employees VALUES(5, 'Ito');`
- `ADD employees SET(5, 'Ito');`

6. foodsテーブルに対して以下のコードを実行したときの出力結果として正しいものを選択してください。(1つ選択)

**[foodsテーブル]**

id	name	price	order
1	apple	200	10
2	banana	300	30
3	carrot	150	25
4	durian	500	5

```
try{
    $db = new PDO('mysql:host=localhost;dbname=testsample',
        'user-id', 'user-pass');

    $db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    $sql = "UPDATE foods SET price = price - 50 where price >= 200";
    $result = $db->exec($sql);

    print "Changed the price of " . $result . " rows.";

}catch(PDOException $e){
    print "connect error". $e->getMessage();
}
```

- A. Changed the price of 1 rows.
- B. Changed the price of 2 rows.
- C. Changed the price of 3 rows.
- D. Changed the price of 4 rows.

7. foodsテーブルに対して以下のSQL文を実行したとき、実行後のfoodsテーブルの状況として正しいものを選択してください。(1つ選択)

**[foodsテーブル]**

id	name	price	order
1	apple	200	10
2	banana	300	30
3	carrot	150	25
4	durian	500	5

```
DELETE FROM foods WHERE price < 200;
```

A.

id	name	price	order
1	apple	200	10
3	carrot	150	25

B.

id	name	price	order
2	banana	300	30
4	durian	500	5

C.

id	name	price
1	apple	200
2	banana	300
4	durian	500

D.

id	name	price	order
1	apple	200	10
2	banana	300	30
4	durian	500	5

8. fruitsテーブルに対して以下のSQL文を実行したときの出力結果として正しいものを選択してください。(1つ選択)

【fruitsテーブル】

id	name	price	stock
1	plum	240	40
2	kiwi	120	80
3	lemon	180	50
4	melon	360	30

```
SELECT name , price , stock FROM fruits ORDER BY price;
```

A.

name	price	stock
kiwi	120	80
lemon	180	50
melon	360	30
plum	240	40

B.

name	price	stock
melon	360	30
plum	240	40
lemon	180	50
kiwi	120	80

C.

name	price	stock
kiwi	120	80
lemon	180	50
plum	240	40
melon	360	30

D.

name	price	stock
plum	360	40
kiwi	240	80
lemon	180	50
melon	120	30

9. 次のコードを実行したときの出力結果として正しいものを選択してください。(1つ選択)

```
try{
    $db = new PDO('mysql:host=localhost;dbname=testsample',
        'user-id', 'user-pass');

    $db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    $sql = "SELECT name , price , stock
        FROM fruits ORDER BY stock LIMIT 3";
    $q = $db->query($sql);

    while ($row = $q->fetch()){
        print "$row[name] \t $row[price] \t $row[stock] \n";
    }

} catch(PDOException $e){
    print "connect error". $e->getMessage();
}
```

【fruitsテーブル】

id	name	price	stock
1	plum	240	40
2	kiwi	120	80
3	lemon	180	50
4	melon	360	30

- A.    lemon        180        50
- B.    kiwi        120        80  
      lemon        180        50  
      plum        240        40  
      melon       360        30
- C.    melon        360        30  
      plum        240        40  
      lemon        180        50
- D.    lemon        180        50  
      plum        240        40  
      melon       360        30

10. 次のコードを実行したときに以下の結果が得られるようにしたい場合、空欄①に当てはまる記述として正しいものを選択してください。(1つ選択)

```
try{
    $db = new PDO('mysql:host=localhost;dbname=testsample',
                  'user-id', 'user-pass');

    $db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    $sql = "SELECT id , name FROM employees";
    $q = $db->query($sql);

    $rows = $q-><div data-bbox="261 228 296 241" data-label="Text">

①


```

**【employeesテーブル】**

id	name
1	Tanaka
2	Yamada
3	Takahashi
4	Saito

**【実行結果】**

```
1 Tanaka
2 Yamada
3 Takahashi
4 Saito
```

- A. getAll
- B. get
- C. fetchAll
- D. fetch

11. 次のコードでは、PDOのプリペアドステートメントを使用して、外部からのデータを用いた検索を行っています。空欄①、②に当てはまる記述の組み合わせとして正しいものを選択してください。

```
try{
    $db = new PDO('mysql:host=localhost;dbname=testsample',
                  'user-id', 'user-pass');

    $db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $db->setAttribute(PDO::ATTR_DEFAULT_FETCH_MODE, PDO::FETCH_
                      OBJ);

    $sql = "SELECT name, price
            FROM foods WHERE price >= ? AND price <= ?";

    $q = $db->①($sql);
    $q->②(array($_POST['min_price'], $_POST['max_price']));

    $foods = $q->fetchAll();

    if(count($foods) == 0){
        print 'No foods matched';
    }else{
        foreach($foods as $row){
            print_r($row);
        }
    }

} catch(PDOException $e){
    print "connect error". $e->getMessage();
}
```

- A. ①bind            ②exec
- B. ①bind            ②execute
- C. ①query           ②exec
- D. ①query           ②execute
- E. ①prepare        ②exec
- F. ①prepare        ②execute