Identify the query that will list all SCREWs.

Question 1 options:

- `SELECT * FROM Practice WHERE Part = 'BOLT';`
- `SELECT * FROM Practice WHERE Material = 'SCREW';`
- `SELECT * FROM Practice WHERE Part = 'WASHER';`
- `SELECT * FROM Practice WHERE Part = 'SCREW';`

Question 2 (Mandatory) (1 point)

Identify the query that will list all parts made from BRASS.

Question 2 options:

- ```
  SELECT *
  FROM Practice
  WHERE Material != 'BRASS';
  ```
- ```
  SELECT *
  FROM Practice
  WHERE Part = 'BRASS';
  ```
- ```
  SELECT *

  FROM Practice

  WHERE Material = 'BRASS';
  ```
- ```
  SELECT *

  FROM Practice

  WHERE Material = 'BOLT';
  ```

Question 3 (Mandatory) (1 point)

Identify the query that will list all SCREWs, all WASHERSs.

Question 3 options:

```
SELECT *

FROM Practice

WHERE Part = 'SCREW'
OR Part = 'NUT';
```

```
SELECT *
FROM Practice

WHERE Part = 'SCREW'
AND Part = 'WASHER';
```

```
SELECT *
FROM Practice

WHERE Part = 'SCREW'
OR Material = 'WASHER';
```

```
SELECT *

FROM Practice

WHERE Part = 'SCREW'

OR Part = 'WASHER';
```

Question 4 (Mandatory) (1 point)

Count the number of parts made from STEEL

Question 4 options:

```
SELECT COUNT( * )

FROM Practice

WHERE STEEL = 'Material';
```

```
SELECT COUNT( * )

FROM Practice

WHERE Material = 'STEEL';
```

```
SELECT COUNT( * )
FROM Practice
WHERE Part = 'STEEL'
AND Material = 'STEEL';
```

```
SELECT COUNT( * )
FROM Practice
WHERE Part = 'STEEL';
```

Question 5 (Mandatory) (1 point)

Identify the query that will list all parts **not** made from BRASS

Question 5 options:

```
SELECT *
FROM Practice
WHERE Part != 'ALUMINIUM';
```

```
SELECT COUNT(*)
FROM Practice
WHERE Part = 'ALUMINIUM'
AND Material = 'ALUMINIUM';
```

```
SELECT *
FROM Practice
WHERE Material != 'BRASS';
```

- SELECT *

  FROM Practice

  WHERE Material != 'ALUMINIUM';

Question 6 (Mandatory) (3 points)

Study the following SQL statement. Then select the three correct options.

```
SELECT *

FROM Practice

WHERE Part = 'WASHER';
```

Question 6 options:

- [ ] FROM in the SELECT clause specifies the tables that should be used for the query
- [ ] The * in the SELECT clause implies all tables in the database
- [ ] WHERE in the SELECT clause specifies the condition to filter columns in the query
- [ ] The * in the SELECT clause implies all columns
- [ ] FROM in the SELECT clause specifies the attributes that should be used for the query
- [ ] WHERE in the SELECT clause specifies the condition to filter rows in the query

Question 7 (Mandatory) (1 point)

How many rows does the `Practice` table have?

Question 7 options:

- 12 * 2 = 24

- [x] `12`
- [ ] `12 / 2 = 6`
- [ ] `2`

Question 8 (Mandatory) (1 point)

How many columns does the `Practice` table have?

Question 8 options:

- [ ] `12 * 2 = 24`
- [ ] `12 / 2 = 6`
- [ ] `24`
- [ ] `12`
- [x] `2`

Question 9 (Mandatory) (1 point)

Saved

Study the file `Practice-DDL-DML.sql`

What is the data type for `Part`.

Question 9 options:

- [ ] `DECIMAL`
- [ ] `CHAR`

- INTEGER
- VARCHAR
- DATE

Question 10 (Mandatory) (1 point)

What does the following statement do

```
INSERT INTO Practice( Part, Material )
VALUES( 'WASHER', 'ALUMINIUM' );
```

Question 10 options:

- It removes data from the Practice tables
- Adds data to the Practice Table. It adds WASHER as Part and ALUMINIUM as Material.
- It creates a table called Practice
- Adds data to the Practice Table. It adds WASHER as Material and ALUMINIUM as Part

Question 11 (Mandatory) (1 point)

What does the following statement do

```
CREATE TABLE Practice(
   Part VARCHAR( 20 ),
   Material VARCHAR( 20 )
);
```

Question 11 options:

- Adds data to the Practice Table. It adds WASHER as Materialand ALUMINIUM as Part.
- It removes data from the Practice tables
- It creates a table called **Practice** with two columns Part and Material
- Adds data to the Practice Table. It adds WASHER as Part and ALUMINIUM as Material.
- It removes the table Practice from the database.

Question 12 (Mandatory) (1 point)

Saved

The notation **!=** implies **not equal to**

Question 12 options:

- True
- False