ALTA-PAY INTERVIEW

What is the result of the following code?

```java
public class ExceptionTest {
    public Integer divide(int a, int b) {
        try {
            return a / b;
        } finally {
            System.out.println("finally");
        }
    }

    public static void main(String[] args) {
        ExceptionTest test = new ExceptionTest();

        try {
            System.out.println(test.divide(10, 0));
        } catch (Exception ex) {
            System.out.println("Division by 0!");
        }
    }
}
```

1. ○ The program will print the following:

   Division by 0!
   finally

2. ○ The program will print the following:

   finally

3. ○ The program will print the following:

   Division by 0!

4. ○ The program will print the following:

   finally
   Division by 0!

5. ○ Compilation error on line 5 due to missing catch keyword

Question #2 (Java)

Which of the following statements are correct?

1. ☐ A class can inherit from a base class and implement three interfaces at the same time

```
public class DerivedClass extends BaseClass implements Interface1, Interface2, Interface3 {}
```

2. ☐ An abstract class can inherit from another abstract class

```
public abstract class AbstractClass extends AbstractClass2 {}
```

3. ☐ A class can inherit from two or more base classes

```
public class DerivedClass extends BaseClass1, BaseClass2, ... {}
```

4. ☐ A class can inherit from a normal class and an abstract class at the same time

```
public class DerivedClass extends BaseClass, AbstractClass {}
```

5. ☐ An interface can inherit from an abstract class

```
public interface Interface extends AbstractClass {}
```

6. ☐ A class can inherit from two or more abstract classes

```
public class DerivedClass extends AbstractClass, AbstractClass2, ... {}
```

Question #2 (Java)

Which of the following statements are correct?

1. ☐ A class can inherit from a base class and implement three interfaces at the same time

```
public class DerivedClass extends BaseClass implements Interface1, Interface2, Interface3 {}
```

2. ☐ An abstract class can inherit from another abstract class

```
public abstract class AbstractClass extends AbstractClass2 {}
```

3. ☐ A class can inherit from two or more base classes

```
public class DerivedClass extends BaseClass1, BaseClass2, ... {}
```

4. ☐ A class can inherit from a normal class and an abstract class at the same time

```
public class DerivedClass extends BaseClass, AbstractClass {}
```

5. ☐ An interface can inherit from an abstract class

```
public interface Interface extends AbstractClass {}
```

6. ☐ A class can inherit from two or more abstract classes

```
public class DerivedClass extends AbstractClass, AbstractClass2, ... {}
```

Question #3 (Java)

What is the output of the following code?

```
Set<Integer> set = new TreeSet<Integer>();

set.add(3);
set.add((int)3.0);
set.add(2);
set.add(2);
set.add(new Integer(2));
set.add(Integer.parseInt("2"));

System.out.println(set);
```

1. ◯ [3, 3, 2, 2, 2, 2]

2. ◯ [2, 3]

3. ◯ [3, 2]

4. ◯ [2, 2, 2, 2, 3, 3]

5. ◯ [3, 2, 2]

6. ◯ [2]

Submit

Your time: 4 min 39

Question #4 (Java)

Consider the following entity classes:

```
1  @Entity
2  @Table(name="DOOR")
3  // INSERT CODE HERE
4  public class Door implements Serializable {
5      ...
6  }
7
8  @Entity
9  @Inheritance(discriminatorValue="I")
10 public class IronDoor extends Door {
11     ...
12 }
13
14 @Entity
15 @Inheritance(discriminatorValue="W")
16 public class WoodDoor extends Door {
17     ...
18 }
```

Which of the following code snippets, inserted at line 3, will map all the entities to a single database table?

1. ◯ @Inheritance(strategy=SINGLE_TABLE_PER_HIERARCHY, Discriminator="DOOR_TYPE")

2. ◯ @Discriminator(name="DOOR_TYPE", strategy=SINGLE_TABLE_PER_HIERARCHY)

3. ◯ @Inheritance(strategy=SINGLE_TABLE)
   @DiscriminatorColumn(name="DOOR_TYPE")

4. ◯ @Inheritance(strategy=SINGLE_TABLE_PER_HIERARCHY)
   @Discriminator(name="DOOR_TYPE")

Question #5 (Java)

What is the output of the following code?

```java
public class Pet
{
    public String getName()
    {
        return "pet";
    }
}
```

```java
public class Cat extends Pet
{
    @Override
    public String getName()
    {
        return "cat";
    }
}
```

```java
public class Shape
{
    private String name;

    public Shape()
    {
        name = "shape";
    }

    public String getName()
    {
        return name;
    }
}
```

```java
public class Ball extends Shape
{
    private String name;

    public Ball()
    {
        name = "ball";
    }
}
```

```
public class Main
{
    public static void main(String[] args)
    {
        Pet myPet = new Cat();
        Cat johnsCat = new Cat();
        Ball ball = new Ball();

        System.out.println(String.format("My %s is playing with a %s. John's %s is sleeping",
                myPet.getName(), ball.getName(), johnsCat.getName()));
    }
}
```

1. ◯ My pet is playing with a ball. John's pet is sleeping

2. ◯ My cat is playing with a ball. John's cat is sleeping

3. ◯ My pet is playing with a shape. John's cat is sleeping

4. ◯ My cat is playing with a shape. John's cat is sleeping

5. ◯ My pet is playing with a ball. John's cat is sleeping

Submit

I Don

Question #6 (Java)

Which of the following code snippets shows the correct way of defining a synchronized method?

1. ☐
```
synchronized MyLockObject lockObject = new MyLockObject();
void method(MyLockObject lockObject){
    //synchronized method code
}
```

2. ☐
```
final MyLockObject lockObject = new MyLockObject();
void method(){
    synchronized (lockObject){
        //synchronized method code
    }
}
```

3. ☐
```
synchronized void method() {
    //synchronized method code
}
```

4. ☐
```
void method(){
    synchronized {
        //synchronized method code
    }
}
```

Question #7 (Java)

Consider the following JPA entity classes:

```
1  @Entity
2  public class Person {
3      ...
4      @ElementCollection
5      // INSERT HERE
6      public Set<Address> getAddresses() {...}
7      ...
8  }
9
10 // INSERT HERE
11 public class Address {
12     ...
13     // INSERT HERE
14     protected City city;
15     ...
16 }
17
18 // INSERT HERE
19 public class City {
20     ...
21     protected String name;
22     ...
23 }
```

Which of the following, inserted at lines 5, 10, 13, and 18 respectively, will give us a list of addresses sorted by city name when we load a Person entity?

1. ○ @OrderBy("city.name")
   @Entity
   @OneToOne
   @Entity

2. ○ @OrderBy("city.name")
@Embedded
@Embeddable
@Embedded

3. ○ @OrderColumn("city.name")
@Embeddable
@Embedded
@Embeddable

4. ○ @OrderBy("city.name")
@Embeddable
@Embedded
@Embeddable

5. ○ @OrderColumn("name")
@Embeddable
@Embedded
@Embeddable

Question #8 (Java)

Given that FileNotFoundException extends IOException, what is the output of the following code?

```
public static void main(String[] args) {
    try {
        throw new FileNotFoundException();
    } catch (FileNotFoundException e) {
        System.out.println("file not found");
    } catch (IOException e) {
        System.out.println("io");
    } catch (Exception e) {
        System.out.println("exception");
    } finally {
        System.out.println("finally");
    }
}
```

1. ○ file not found
   finally

2. ○ file not found
   io
   exception
   finally

3. ○ file not found
   exception

4. ○ file not found
   exception
   finally

5. ○ exception
   finally

Question #9 (Java)

Which of the following code snippets will return “true”?

1. ☐
```
new Integer(10) == new Integer(10)
```

2. ☐
```
10 == new Integer(10)
```

3. ☑
```
int i = 0;
```
```
1 == i++
```

4. ☐
```
true | false & false
```

5. ☐
```
10 == 10
```

Submit

Your time:

Question #10 (Java)

What is the result of the following code?

```java
class GenericsTest {
    public static void main(String[] args) {
        List<Integer> list = new ArrayList<Integer>();
        for (int i = 0; i < 5; i++) {
            list.add(i / 2);
        }
        System.out.println(list);
    }
}
```

1. ○ The program will compile and print the following:
   [0, 0.5, 1, 1.5, 2]

2. ○ The program will compile and print the following:
   [0, 0, 1, 1, 2]

3. ◉ The program will compile and print the following:
   [0, 0, 0, 0, 0]

4. ○ The program won't compile due to an error at line 5.

5. ○ The program will compile but it will raise a runtime exception at line 5.

Submit

I D

Question #11 (Java)

Consider the following code:

```
class ThreadTest {
    public static void main(String[] args) {
        Thread counter = new Thread(new XXX() {
            public void YYY() {
                for (int i = 0; i < 3; i++) {
                    System.out.print(i);
                }
            }
        });
        counter.ZZZ();
    }
}
```

Which of the following, inserted in place of XXX, YYY, and ZZZ respectively, will make the program compile and print "012"?

1. ○ Invocable
   start
   run

2. ○ Thread
   start
   execute

3. ○ Runnable
   start
   run

4. ○ Thread
   execute
   run

5. ○ Runnable
   run
   start

Question #12 (Java)

What is the result of the following code?

```java
import java.util.*;

class TodoList {
    public static void main(String[] args) {
        Map<Todo, String> todos = new HashMap<Todo, String>();
        Todo t1;
        todos.put(t1 = new Todo("Monday"), "Task1");
        todos.put(new Todo("Monday"), "Task2");
        todos.put(new Todo("Tuesday"), "Task3");
        System.out.println("Size: " + todos.size() + ", t1: " + todos.get(t1));
    }
}

class Todo {
    String day;

    Todo(String d) {
        day = d;
    }

    public boolean equals(Object o) {
        return ((Todo) o).day == this.day;
    }

    public int hashCode() {
        return 0;
    }
}
```

1. ◯ The program will print the following:

   Size: 2, t1: Task1

2. ◯ The program will print the following:

   Size: 3, t1: Task2

3. ◯ The program will print the following:

   Size: 2, t1: Task2

4. ◯ The program will print the following:

   Size: 3, t1: Task1

5. ◯ The program will throw an exception.

Submit

You can select only one answer

Question #13 (Java)

Consider the following classes:

```java
public class Parent
{
    public Parent() {
        System.out.println("constructor Parent");
    }

    private void privateMethod() {
        System.out.println("Parent.privateMethod");
    }

    protected void protectedMethod() {
        System.out.println("Parent.protectedMethod");
        privateMethod();
    }

    public void publicMethod() {
        System.out.println("Parent.publicMethod");
        protectedMethod();
    }
}
```

```java
public class Child extends Parent
{
    public Child() {
        System.out.println("constructor Child");
    }

    private void privateMethod() {
        System.out.println("Child.privateMethod");
    }

    @Override
    protected void protectedMethod() {
        System.out.println("Child.protectedMethod");
        privateMethod();
    }

    @Override
    public void publicMethod() {
        System.out.println("Child.publicMethod");
        protectedMethod();
    }
}
```

What is the output of the following code?

```
Parent object = new Child();
object.publicMethod();
```

1. ○ constructor Parent
   constructor Child
   Child.publicMethod
   Parent.protectedMethod
   Parent.privateMethod

2. ○ constructor Parent
   constructor Child
   Child.publicMethod
   Child.protectedMethod
   Child.privateMethod

3. ○ constructor Parent
   constructor Child
   Child.publicMethod
   Child.protectedMethod
   Parent.privateMethod

4. ○ constructor Child
   constructor Parent
   Child.publicMethod
   Child.protectedMethod
   Parent.privateMethod

5. ○ constructor Child
   constructor Parent
   Child.publicMethod
   Parent.protectedMethod
   Parent.privateMethod

What is the output of the following code?

```
class Tree {

    int leaves = 100;

    public static void main(String[] args) {
        Tree tree1 = new Tree();
        tree1.leaves = 200;
        Tree tree2 = doubleTreeLeaves(tree1);
        System.out.println(tree1 == tree2);
        System.out.println(tree1.leaves);
        System.out.println(tree2.leaves);
    }

    static Tree doubleTreeLeaves(Tree tree1) {
        tree1 = new Tree();
        tree1.leaves = 2 * tree1.leaves;
        return tree1;
    }
}
```

1. ○ true
   200
   400

2. ○ true
   400
   400

3. ○ true
   200
   200

4. ○ false
   200
   200

5. ○ false
   200
   400

Question #15 (Java)

What is the output of the following code?

```
public class Parent {
    {
        System.out.print("A ");
    }

    static {
        System.out.print("B ");
    }
}
```

```
class Child extends Parent {
    {
        System.out.print("C ");
    }

    static {
        System.out.print("D ");
    }

    public static void main(String[] args) {
        Child child = new Child();
    }
}
```

1. ◯ A C B D

2. ◯ B D A C

3. ◯ A B C D

4. ◯ C D A B

Question #16 (Java)

What is the output of the following code?

```java
public class Vehicle
{
    public int maxSpeed;

    public static String country;

    public Vehicle(int maxSpeed)
    {
        this.maxSpeed = maxSpeed;
        country = "UK";
    }
}
```

```java
public class Car extends Vehicle
{
    public String transmission;

    public Car(int maxSpeed, String transmission)
    {
        super(maxSpeed);
        this.maxSpeed = 100;
        this.transmission = transmission;
    }

    static
    {
        country = "USA";
    }
}
```

```java
public class Main {

    public static void main(String[] args) {
        Car car = new Car(200, "Manual");

        System.out.println(String.format("Max speed is %d, Country is %s",
                car.maxSpeed, Vehicle.country));
    }
}
```

1. ◯ Max speed is 100, Country is USA

2. ◯ Max speed is 100, Country is UK

3. ◯ Max speed is 200, Country is USA

4. ◯ Max speed is 200, Country is UK

5. ◯ The program will throw an exception.

Submit

Your time:

Question #17 (Java)

What is the output of the following code?

```java
public class Forecast
{
    public int temperature;
    public int pressure;
}
```

```java
public class Main
{
    public static void changeTheString(String weather)
    {
        weather = "sunny";
    }

    public static void changeTheArray(String[] rainyDays)
    {
        rainyDays[1] = "Sunday";
    }

    public static void changeTheObject(Forecast forecast)
    {
        forecast.temperature = 35;
    }

    public static void main(String[] args)
    {
        String weather = "rainy";
        changeTheString(weather);
        System.out.println("The weather is " + weather);

        String[] rainyDays = new String[] {"Monday", "Friday"};
        changeTheArray(rainyDays);
        System.out.println("The rainy days were on " + rainyDays[0] + " and " + rainyDays[1]);

        Forecast forecast = new Forecast();
        forecast.pressure = 700;
        forecast.temperature = 20;
        changeTheObject(forecast);
        System.out.println("The temperature is " + forecast.temperature + "°C");
    }
}
```

1. The weather is rainy
   The rainy days were on Monday and Friday
   The temperature is 20°C

2. The weather is sunny
   The rainy days were on Monday and Sunday
   The temperature is 35°C

3. The weather is rainy
   The rainy days were on Monday and Sunday
   The temperature is 35°C

4. The weather is sunny
   The rainy days were on Monday and Sunday
   The temperature is 20°C

5. The weather is rainy
   The rainy days were on Monday and Friday
   The temperature is 35°C

Submit

Your time: 13

Question #18 (Java)

Which of the following code snippets shows the correct implementation of the Singleton pattern?

1.

```
public class Singleton
{
    private Singleton instance;

    public static Singleton getInstance()
    {
        return (new Singleton()).instance;
    }

    public Singleton()
    {
        instance = new Singleton();
    }
}
```

2.

```
public class Singleton
{
    private static Singleton instance;

    private Singleton()
    {
    }

    public static Singleton getInstance()
    {
        return instance != null ? instance : (instance = new Singleton());
    }
}
```

3.

```
public class Singleton
{
    private Singleton instance;

    public Singleton getInstance()
    {
        return instance;
    }
```

3.

```
public class Singleton
{
    private Singleton instance;

    public Singleton getInstance()
    {
        return instance;
    }

    public Singleton()
    {
        instance = new Singleton();
    }
}
```

4.

```
public class Singleton
{
    private static Singleton instance;

    public static Singleton getInstance()
    {
        return instance == null ? new Singleton() : instance;
    }
}
```

Submit

Your

Question #19 (Java)

Which of the following code snippets will remove all even numbers from a list of random integers?

```
List<Integer> list = GetSomeRandomValues();
```

1. ☐

```
Iterator<Integer> iterator = list.iterator();
while (iterator.hasNext()) {
    Integer element = iterator.next();
    if (element % 2 == 0) {
        iterator.remove();
    }
}
```

2. ☑

```
List<Integer> tempList = new ArrayList<Integer>();
for (Integer element : list) {
    if (element % 2 == 0) {
        tempList.add(element);
    }
}
list.removeAll(tempList);
```

3. ☐

```
for (Integer element : list) {
    if (element % 2 == 0) {
        list.remove(element);
    }
}
```

Submit

Your time: 13 min 4

Question #20 (Java)

Consider the following code:

```
1  Department department = null;
2  Query query = entityManager.createQuery("SELECT d FROM Department d " +
3          "left join fetch d.employees WHERE d.id = :id");
4  query.setParameter("id", 99);
5
6  department = (Department) query.getSingleResult();
7
8  Collection<Employee> employees = department.getEmployees();
9
10 for(Employee employee : employees)
11 {
12     System.out.println(employee.getName());
13 }
```

Which of the following statements is correct regarding the above code?

1. ○ A database query occurs only once, at line 6.

2. ○ A database query occurs only once, at line 8.

3. ○ A database query occurs several times: once at line 6 and once for each employee inside the "for" loop at line 12.

4. ○ A database query occurs only twice, at line 6 and line 8.

5. ○ A database query occurs several times: once at line 6, once at line 8, and once for each employee inside the "for" loop at line 12.

Submit

I Don't Kn

Your time: 13 mi

Question #21 (MySQL)

Which of the following SQL queries will retrieve people whose minimum loan amount is more than $1000?

1. ○
```
SELECT p.first_name, p.last_name
FROM people AS p, loans AS l
WHERE p.id = l.people_id
GROUP BY p.id, p.first_name, p.last_name
HAVING MIN(l.amount) > 1000
```

2. ●
```
SELECT p.id, p.first_name, p.last_name
FROM people AS p, loans AS l
GROUP BY p.id, p.first_name, p.last_name
WHERE p.id = l.people_id AND MIN(l.amount) > 1000
```

3. ○
```
SELECT p.id, p.first_name, p.last_name
FROM people AS p, loans AS l
WHERE l.people_id = p.id AND MIN(l.amount) > 1000
```

4. ○ None of the above.

Submit

I Don't

You can select only one answer

Your time: 14 min 9 sec

Question #22 (MySQL)

Which of these will complete the following SQL query so that it returns the list of people who have no loans?

```
SELECT p.id, p.first_name, p.last_name
FROM people AS p
WHERE ... (SELECT * FROM loans AS l WHERE l.people_id = p.id)
```

1. ○ p.id IN

2. ○ NOT EXISTS

3. ○ EXISTS

4. ○ p.id NOT IN

Submit

I Don't Know

You can select only one answer

Your time:

Question #23 (MySQL)

We need to find all people who do not have children. Which SQL query should we use?

1. ○
```
SELECT *
FROM people AS p1
JOIN people AS p2 ON p1.id = p2.father_id OR p1.id = p2.mother_id
WHERE p1.id IS NULL;
```

2. ●
```
SELECT *
FROM people AS p
WHERE father_id IS NULL AND mother_id IS NULL;
```

3. ○
```
SELECT *
FROM people AS p
WHERE NOT EXISTS
  (SELECT *
   FROM people AS p1
   WHERE p1.mother_id = p.id OR p1.father_id = p.id);
```

4. ○ None of the above.

Submit

You can select only one answer

Your time: 1

Question #24 (MySQL)

When we need to concatenate the results of two independent queries with an equal set of fields, e.g.

```
(SELECT 1, 2 FROM A1) ... (SELECT 1, 2 FROM A2)
```

we should use the keyword:

1. ○ UNION
2. ○ JOIN
3. ○ PLUS
4. ○ CONCATENATE

Submit

ID

You can select only one answer

Your time: 14 min 46 sec

Question #25 (MySQL)

Which function is used to get the current time in MySQL?

1. ○ Now()
2. ○ GetTime()
3. ○ DateTime()
4. ○ DateTime.Now()
5. ○ Time()
6. ○ CurTime()
7. ○ None of the above

Submit

I Don't Know

You can select only one answer

Question #26 (MySQL)

Which of the following will be the result of the following SQL query?

```
ALTER TABLE _table_ ADD UNIQUE(_field_);
```

1. ◯ The "_field_" field in the "_table_ " table will get the unique property.

2. ◯ The new field "_field_" will be created in the "_table_" table.

3. ◯ The new unique field "_field_" will be created in the "_table_" table.

4. ◯ None of the above.

Submit

You can select only one answer

Your time: 15 min 10 sec

Question #27 (MySQL)

Which SQL query will retrieve people that are namesakes for the person whose id = 9?

1. ○
```
SELECT *
FROM people AS p
WHERE
(p.first_name, p.last_name) = (SELECT first_name, last_name FROM people WHERE id = 9)
```

2. ○
```
SELECT *
FROM people AS p
WHERE p.id IN (SELECT id FROM people AS p2 WHERE id = 9)
AND p2.first_name = p.first_name AND p2.last_name = p.last_name
```

3. ○
```
SELECT *
FROM people AS p
WHERE EXISTS (SELECT * FROM people AS p2 WHERE p2.id = 9)
```

4. ○ None of the above queries will work.

Submit

I Don't Know

You can select only one answer

Question #28 (MySQL)

What will the following SQL query do?

```
UPDATE loans AS l
INNER JOIN people AS p ON p.id = l.people_id
SET l.end_date = DATE_ADD(l.end_date, INTERVAL -1 MONTH)
WHERE p.id = @ID
```

1. ○ Delete all loans with the end_date less than the current date + 1 month for the person whose id = @ID

2. ○ Decrease the loan end date by 1 month for the person whose id = @ID

3. ○ Increase the loan end date by 1 month for the person whose id = @ID

4. ○ Add a new loan with end date = current date + 1 month for the person whose id = @ID

Submit

I Don't Know

You can select only one answer

Question #29 (MySQL)

Which of these will complete the following SQL query so that it returns the number of loans for each person in the PEOPLE table?

```
SELECT p.first_name, p.last_name, count(*) AS loans_number
FROM people AS p
... JOIN loan l ON p.id = l.people_id
GROUP BY p.first_name, p.last_name
```

1. ◯ INNER

2. ◯ RIGHT

3. ◯ LEFT

4. ◯ Empty space

Submit

I Don't Know

You can select only one answer

Your time: 15 min 45 s

Question #30 (MySQL)

Under what circumstances will the following SQL query return TRUE?

```
SELECT
  (SELECT YEAR(p.birth_date)
   FROM people AS p
   WHERE p.id = 5) >
   ANY (SELECT YEAR(p.birth_date)
        FROM people AS p
        WHERE p.last_name = 'McCarthy')
```

1. ○ If the person whose id = 5 is older than any people with last_name = "McCarthy".

2. ○ If the person whose id = 5 is younger than any people with last_name = "McCarthy".

3. ○ This query will never return TRUE.

4. ○ This query will always return TRUE.

Submit

I Don't Know

You can select only one answer

Your time: 15 min 54 sec

Question #31 (MySQL)

The following SQL query

```
SELECT
p.id,
p.first_name,
p.last_name,
SUM(case when l.end_date < CURDATE() then l.amount else 0 end) total
FROM people AS p, loans AS l
WHERE l.people_id = p.id
GROUP BY p.id, p.first_name, p.last_name
```

will return a list of people ...

1. ○ with the number of their overdue payments
2. ○ with the sum of their remaining payments
3. ○ with the number of their remaining payments
4. ○ with the sum of their overdue payments

Submit

I Don't Know

You can select only one answer

Question #32 (MySQL)

Given the following two tables:

tbl1:

id	name
1	A
2	B
3	C

tbl2:

id	letter
2	D
3	E
4	F

Which SQL query will return the following result?

id	name	id	letter
1	A	NULL	NULL
2	B	2	D
3	C	3	E

1. ☐

```
SELECT *
FROM tbl1
LEFT JOIN tbl2 USING(id);
```

2. ☐

```
SELECT *
FROM tbl1
RIGHT JOIN tbl2 ON tbl1.id = tbl2.id;
```

3. ☐

```
SELECT *
FROM tbl1
JOIN tbl2 ON tbl1.id = tbl2.id;
```

4. ☐

```
SELECT *
FROM tbl1
LEFT JOIN tbl2 ON tbl1.id = tbl2.id;
```

Submit

Question #33 (MySQL)

Which of the following SQL queries are valid?

1. ☐

```
SELECT id, COUNT(id)
FROM products
WHERE id > 2
GROUP BY id
ORDER BY id;
```

2. ☐

```
SELECT id, COUNT(id)
FROM products
GROUP BY id
HAVING id > 2
ORDER BY id;
```

3. ☐

```
SELECT id, COUNT(id) AS total
FROM products
WHERE total > 5
GROUP BY id
ORDER BY id;
```

4. ☐

```
SELECT id, COUNT(id) AS total
FROM products
GROUP BY id
HAVING total > 5
ORDER BY id;
```

Submit

Question #34 (MySQL)

What is the definition of INNER JOIN in MySQL?

1. ○ Selects records from both database tables with NO matching values.

2. ◉ Selects all records from the first database table and only those records from the second table that have matching values.

3. ○ Selects records from both database tables with matching values.

4. ○ None of the above.

Submit

You can select only one answer

Question #35 (MySQL)

What is the difference between the following two statements?

```
DELETE * FROM [simple_table_without_indexes_and_keys]
```

```
TRUNCATE TABLE [simple_table_without_indexes_and_keys]
```

1. ◯ The TRUNCATE statement will drop and recreate the same table while the DELETE statement will just delete all data.

2. ◯ The TRUNCATE statement will delete the table while the DELETE statement will delete all data.

3. ◯ The DELETE statement will drop and recreate the same table while the TRUNCATE statement will just delete all data.

4. ◯ These two statements are functionally identical.

Submit

You can select only one answer

Your time: 17 min 33 s

Question #36 (MySQL)

Which of these will complete the following SQL query so that it returns all people whose last_name contains only two instances of the letter "a" (case-insensitive)?

```
SELECT * FROM people AS p WHERE ...
```

1. ○
```
p.last_name LIKE '%a%a%'
```

2. ○
```
LENGTH(p.last_name) - LENGTH(REPLACE(LOWER(p.last_name), 'a', '')) = 2
```

3. ○
```
LENGTH(p.last_name) - INSTR(p.last_name, 'a')
```

4. ○
```
LOWER(p.last_name) LIKE '%a%a%'
```

Submit

I Don't Know

You can select only one answer

Your time

Question #37 (MySQL)

Which SQL query will return the last value of the column with the attribute AUTO_INCREMENT?

1. ○ SELECT LAST_ID();

2. ○ SELECT AUTO_INCREMENT();

3. ○ SELECT LAST_INSERT_ID();

4. ○ None of the above

Submit

You can select only one answer

Your time: 17 min 57 sec

Question #38 (MySQL)

Which SQL query will retrieve the three people with the highest total loan amounts?

1.
```
SELECT p.first_name, p.last_name, COUNT(l.amount) AS amount
FROM people AS p
LEFT JOIN loans AS l ON l.people_id = p.id
GROUP BY p.first_name, p.last_name
ORDER BY l.amount DESC LIMIT 3
```

2.
```
SELECT p.first_name, p.last_name, SUM(l.amount) AS amount
FROM people AS p
LEFT JOIN loans AS l ON l.people_id = p.id
GROUP BY p.first_name, p.last_name
ORDER BY l.amount DESC LIMIT 3
```

3.
```
SELECT *
FROM (SELECT
      p.first_name,
      p.last_name,
      SUM(l.amount) AS amount
      FROM people AS p
      LEFT JOIN loans AS l ON l.people_id = p.id
      GROUP BY p.first_name, p.last_name) AS total
ORDER BY total.amount DESC LIMIT 3
```

4.
```
SELECT p.first_name, p.last_name, l.amount
FROM people AS p
LEFT JOIN loans AS l ON l.people_id = p.id
ORDER BY l.amount LIMIT 3
```

Submit

I Don't Know

Your time: 18 min 8 sec

Question #39 (MySQL)

Which of the following is the result of performing a CROSS JOIN of table A to table B?

1. ○ The first row from table B is joined to every row of table A.
2. ○ Every row of table B is joined to every row of table A.
3. ○ All rows of table A and all rows of table B are retrieved sequentially.
4. ○ None of the above.

Submit

I Don't Know

You can select only one answer