

Development Test

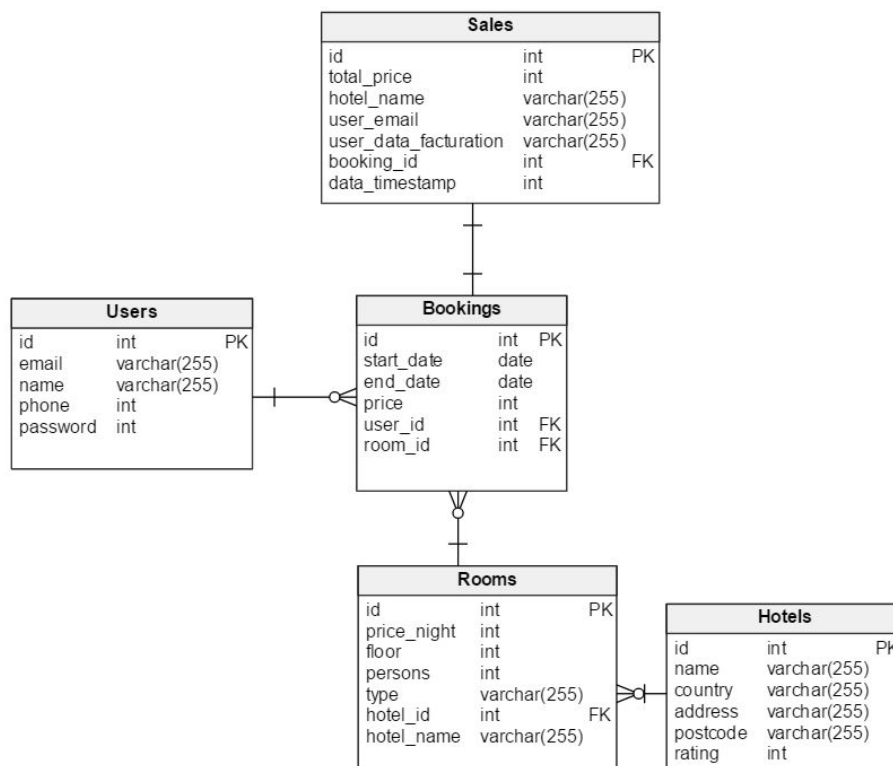
Yeray Brun Martin

Test

1. SQL and data layer

- a. Imagine a hotel reservation website such as booking.com. Define the minimum data model to support the basic case of web, that is, booking a room. Feel free to use SQL or NoSQL solutions. Just justify your decision and explain your decisions and what benefits you achieved.
- b. Given the previous data model, provide the table definition, indexes and any technical aspect you might consider important, depending on the data storage you decided to use. The performance of your solution will also taken into consideration. Please provide the way to obtain (either sql queries in case you chose a relational database, or any other solution to extract the data in case of any NoSQL solution):
 - i. Daily sales report.
 - ii. Best performer hotels of the day.
 - iii. Weekly accumulated sales

Modelo de datos mínimo. Hecho con la herramienta online de my.vertabelo.com/



En la tabla de "Sales" he puesto algunos datos que ya figuran en otras tablas como el hotel_name o el user_email, para evitar de hacer muchos joins y de esta forma no aumentar tanto el trafico en la DB.

Daily sales report

sql query:

```
SELECT Sales_id, Sales.total_price, Sales.hotel_name, Sales.user_email,
Sales.user_data_facturation, Bookings.start_date, Bookings.Bend_date,
Bookings.start_date, Bookings.id
FROM Sales
LEFT JOIN Bookings
ON Bookings.id = Sales.booking_id
WHERE Sales.data_timestamp >= today AND Sales.data_timestamp >
tomorrow
```

Los valores *today* y *tomorrow* serán variables de Php o del lenguaje que se este realizando, consultado el valor al servidor. Estos valores serán la fecha de hoy con timestamp y la fecha del día siguiente consultados a partir del servidor.

Best performer hotels of the day

Como no se especifica realmente a que se refiere con "Best performer" he supuesto que se refería a los hoteles que más hayan facturado en el día de hoy.

sql query:

```
SELECT SUM(Sales.total_price) AS total_price, Sales.hotel_name
FROM Sales
WHERE Sales.data_timestamp >= today AND Sales.data_timestamp >
tomorrow
GROUP BY Sales.hotel_name
```

Weekly accumulated sales

sql query:

```
SELECT Sales_id, Sales.total_price, Sales.hotel_name, Sales.user_email,
Sales.user_data_facturation, Bookings.start_date, Bookings.Bend_date,
Bookings.start_date, Bookings.id
FROM Sales
LEFT JOIN Bookings
ON Bookings.id = Sales.booking_id
WHERE Sales.data_timestamp >= ini_week AND Sales.data_timestamp >
final_week
```

Esta Query se tendría que lanzar con un cron cada lunes a las 00:00 donde los valores ini_week y final_week serán los valores de lunes pasado a las 00:00 y este lunes a las 00:00 que serán cogidos a partir de la hora del servidor, en formato timestamp.

2. General

- a. **You have a live server behaving strange, you don't know what is happening but the requests are not responding. You see the number of apache threads increase until the server crashes. How would you start investigating this issue? What is the test plan?**

Intentaría diferenciar si se trata de algún tipo de ataque como de DDoS o que el propio servidor de apache se ha quedado "corto" para el tráfico de clientes de nuestra página. En este último caso sólo tendríamos que aumentar el máximo de clientes que soporta nuestro servidor en los archivos de configuración de apache.

Para ver si se trata de un ataque de DDoS podríamos mirar los logs del mod_evasive o syslog (del kernel). Si detectamos que es un ataque de DDoS, para solucionarlo podríamos limitar el máximo de conexiones por segundo a nuestra página en caso de superar este límite bloquear dicha ip.

- b. **Difference between Unit Test , Integration test and Acceptance test.**

Unit Test: Prueba de una unidad de código, por ejemplo para comprobar el funcionamiento de un módulo de código concreto.

Integration test: Prueba de un gran conjunto de partes, por ejemplo de diferentes módulos.

Acceptance test :Se podría considerar un proceso de revisión donde se comprueba que el sistema cumple todas las especificaciones planteadas.

- c. **What is a mock and how would you create them in PHP.**

Son objetos especiales que representan diferentes niveles de falsificar el comportamiento de un objeto. Son usados normalmente en los Unit Tests para imitar el comportamiento de un objeto real.

Los crearía con PHPUnit que es un framework open source para realizar pruebas unitarias y que permite fácilmente crear objetos mock en php.

<https://www.sitepoint.com/an-introduction-to-mock-object-testing/>

- d. **You need to design a storage architecture for php sessions that provides high scalability, failover, and high performance. What would you do?**

e. JSON vs XML. When / Why would you use instead of the other?

XML → para archivos de configuración y datos que requieran gran extensibilidad como podrían ser fotos, tablas, gráficos entre otros.

JSON → para intercambio de información entre servicios web y APIs REST, ya que en XML tendríamos que transformar los datos antes de que sean importados.

f. If you are using mysql command “explain”, what do you look at to know if you must optimize a query.

En principio miraría el tiempo total para devolver todas las tuplas, aunque en algunos casos podría ser importante también el tiempo que toma para devolver solo la primera tupla.

g. What advantages provides the usage of traits?

Permite la reutilización del código, permitiendo reutilizar conjuntos de funciones sobre varias clases independientes y pertenecientes a clases jerárquicas distintas.

<https://secure.php.net/manual/es/language.oop5.traits.php>

h. What is the benefit of using NameSpaces?

Tienen dos grandes beneficios por un lado permiten organizar de una manera clara y jerárquica el código y por otro lado permiten evitar la colisión de nombres, por ejemplo si hubiera dos clases con el mismo nombre.

i. RESTful interface vs SOAP. Benefits of each option

REST beneficios:

- Bajo consumo de recursos
- Instancias del proceso creadas explícitamente
- Fácil de crear y adoptar

SOAP

- Incrementa la privacidad
- Bastantes herramientas de desarrollo
- Operaciones complejas pueden ser escondidas

<http://users.dsic.upv.es/~rnavarro/NewWeb/docs/RestVsWebServices.pdf>

j. What is dependency injection about? What are some of the key benefits of using it?

Permite poner dentro de un objeto otros que pueden cambiar su comportamiento, sin tener que volver a crear dicho objeto.

Una de las claves es a la hora de realizar Tests, ya que al tener componentes más “desacoblados” estos son más independientes y como consecuencia se hace más fácil la implementación.

3. Problem solving

You receive a string that contains an undefined number of parenthesis, please create a class in OOP that validates that each pair that is opened is also closed. If not throw an exception notifying us of mismatched parenthesis.

```
<?php
class Validate{
    function check_data($string){
        echo $string;
        $cash = 0;
        $i = 0;
        $long_string = strlen($string);
        while($cash >= 0 AND $i<$long_string){
            $open = "(";
            $close = ")";

            if(strcmp($string[$i], $open) == 0){
                $cash++;
            }
            elseif(strcmp($string[$i], $close) == 0){
                $cash--;
            }
            $i++;
        }

        echo "</br>";
        if($cash != 0){
            echo "Exception Mismatched Parenthesis";
        }
        else{
            echo "Successful";
        }
        echo "</br>";
        echo "</br>";
    }
}

$cart = new Validate;
$eval_string = "Hello (my name) is (Josh";
$eval_string2 = ")Hello (my name) is (Josh";
$eval_string3 = "Hello (my name) is (())Josh";
$cart->check_data($eval_string);
$cart->check_data($eval_string2);
$cart->check_data($eval_string3);
?>
```