

Disseny de bases de dades

PRA1: Llenguatge SQL

Presentació

L'objectiu principal d'aquesta activitat és adquirir els coneixements bàsics, tant teòrics com pràctics, per a la posada en marxa d'un SGBD relacional. Per aquest motiu serà necessari saber administrar el servidor MySQL amb programes clients tals com *phpMyAdmin*, *mysql.exe* i *MySQL Workbench*.

Competències

Les competències que es treballaran en aquesta activitat són:

- Conèixer els models de dades associats a la tecnologia de Base de Dades (BD).
- Saber accedir a una BD per a la seva consulta i/o modificació.
- Saber accedir a una BD des d'aplicacions Web.

Objectius

L'objectiu d'aquesta activitat és que els estudiants acabin d'afinar les seves habilitats en el disseny conceptual d'esquemes E-R, que aprenguin a crear bases de dades en un SGBD concret (MySQL) i a accedir a les dades, ja sigui directament des de comandes SQL com des d'una aplicació web.

Descripció de la pràctica a realitzar

L'activitat es divideix en tres parts:

- Treballar el correcte disseny d'un model E-R a partir d'un conjunt de consultes SQL que es realitzen sobre la base de dades que modela.
- Practicar amb el llenguatge SQL i la base de dades adjunta a l'enunciat resolent una sèrie d'exercicis proposats.
- Repasar aspectes bàsics de programació en PHP per accedir a una base de dades i realitzar consultes per tal de mostrar el resultat per pantalla.

Recursos

Per a realitzar la pràctica es facilitarà el fitxer **rutes.zip**, el qual conté un arxiu amb l'estructura de la base de dades amb què es treballarà en les preguntes de l'exercici B i el script que s'implementarà a l'exercici C.

D'altra banda, tal i com s'especifica a l'aula, els mòduls didàctics en què se centrarà la PRA són:

- Mòdul 1. Fonaments de Bases de Dades
- Mòdul 2. Llenguatge SQL
- Mòdul 5. MySQL Motors d'emmagatzematge
- Mòdul 6. Llenguatge PHP
- Mòdul 7. Accés a MySQL amb PHP

Criteris de valoració

L'apartat A tindrà un pes del 30% de la nota i la resta d'apartats tindran un pes del 35%.

En l'**apartat A** s'avaluarà la correcta identificació de les entitats, interrelacions, i atributs presents a la solució proposada. Es valorarà la correcta explicació i justificació de la resposta.

En l'**apartat B** cada pregunta aporta el mateix a la nota de l'exercici. Es valorarà la correcta formulació de les consultes i la validesa de la resposta obtinguda en la seva execució.

En l'**apartat C** es valorarà el correcte funcionament de l'aplicació. Es valorarà que l'aplicació implementi totes les funcionalitats indicades a l'enunciat així com el seu correcte funcionament i la claredat del codi PHP.

Total de puntuació de la PRA1: **10 punts**.

La traducció de punts a la nota per a aquesta PRA és:

- 10-9 punts = "A"
- 8-7 punts = "B"
- 6-5 punts = "C+"
- 4-3 punts = "C-"
- 2-1 punt = "D"

Format i data de lliurament

S'haurà de lliurar aquest mateix document, en format PDF amb les respostes degudament complimentades a l'àrea de Lliurament i registre d'AC.

A més d'aquest document, s'haurà d'adjuntar obligatòriament el fitxer **exercici_B.sql**, on es presenti la solució als problemes plantejats a l'exercici B, i l'*script* **rutes.php** amb la solució de l'exercici C.

Tanmateix, aquest *script* **rutes.php** s'haurà de publicar obligatòriament al vostre espai d'alumne per tal de comprovar el seu correcte funcionament.

Data límit de lliurament: **10 d'abril de 2017**

Nota: Propietat intel·lectual

Sovint és inevitable, en produir una obra multimèdia, fer ús de recursos creats per terceres persones. És per tant comprensible fer-ho en el marc d'una pràctica dels estudis del Grau Multimèdia, sempre i això es documenti clarament i no suposi plagi en la pràctica.

Per tant, en presentar una pràctica que faci ús de recursos aliens, s'ha de presentar juntament amb ella un document en què es detallin tots ells, especificant el nom de cada recurs, el seu autor, el lloc on es va obtenir i el seu estatus legal: si l'obra està protegida pel copyright o s'acull a alguna altra llicència d'ús (*Creative Commons*, llicència GNU, GPL ...). L'estudiant haurà d'assegurar-se que la llicència que sigui no impedeix específicament el seu ús en el marc de la pràctica. En cas de no trobar la informació corresponent haurà d'assumir que l'obra està protegida pel copyright.

Hauran, a més, d'adjuntar els fitxers originals quan les obres utilitzades siguin digitals, i el seu codi font si correspon.

Un altre punt a considerar és que qualsevol pràctica que faci ús de recursos protegits pel copyright no podrà en cap cas publicar-se en Mosaic, la revista del Graduat en Multimèdia a la UOC, a no ser que els propietaris dels drets intel·lectuals donin la seva autorització explícita.

A.- Identificació dels elements d'un model E-R

A partir de les següents consultes SQL:

```
--consulta 1
SELECT p.nom, a.nom
FROM PAÍS p
      JOIN AEROLÍNIA a ON a.país_id = p.id
      JOIN cobrir c ON c.aerolínia_id = a.id
WHERE c.ruta_codi LIKE 'BCN -%' OR c.ruta_codi LIKE '% -PAR';

--consulta 2
SELECT r.codi, COUNT(DISTINCT c.aerolínia_id), AVG (a.capacitat), MIN(a.any)
FROM RUTA r
      JOIN cobrir c ON c.ruta_codi = r.codi
      JOIN AVIÓ a ON a.codi = c.avió_codi
WHERE r.codi LIKE '%BCN%'
GROUP BY r.codi
HAVING c.hora_arribada - c.hora_sortida < 3;
```

Expliqueu quin serà el resultat de cadascuna, tenint en compte que el camp codi de ruta conté informació de les ciutats per les que passarà una aerolínia, per exemple “BCN-PAR” o “BCN-MAD”.

■ Consulta 1

La consulta torna el **nom dels països i el nom de les aerolínies** que tenen trajectes com a **origen** la ciutat de Barcelona (BCN) o com a **destí** la ciutat de París (PAR)

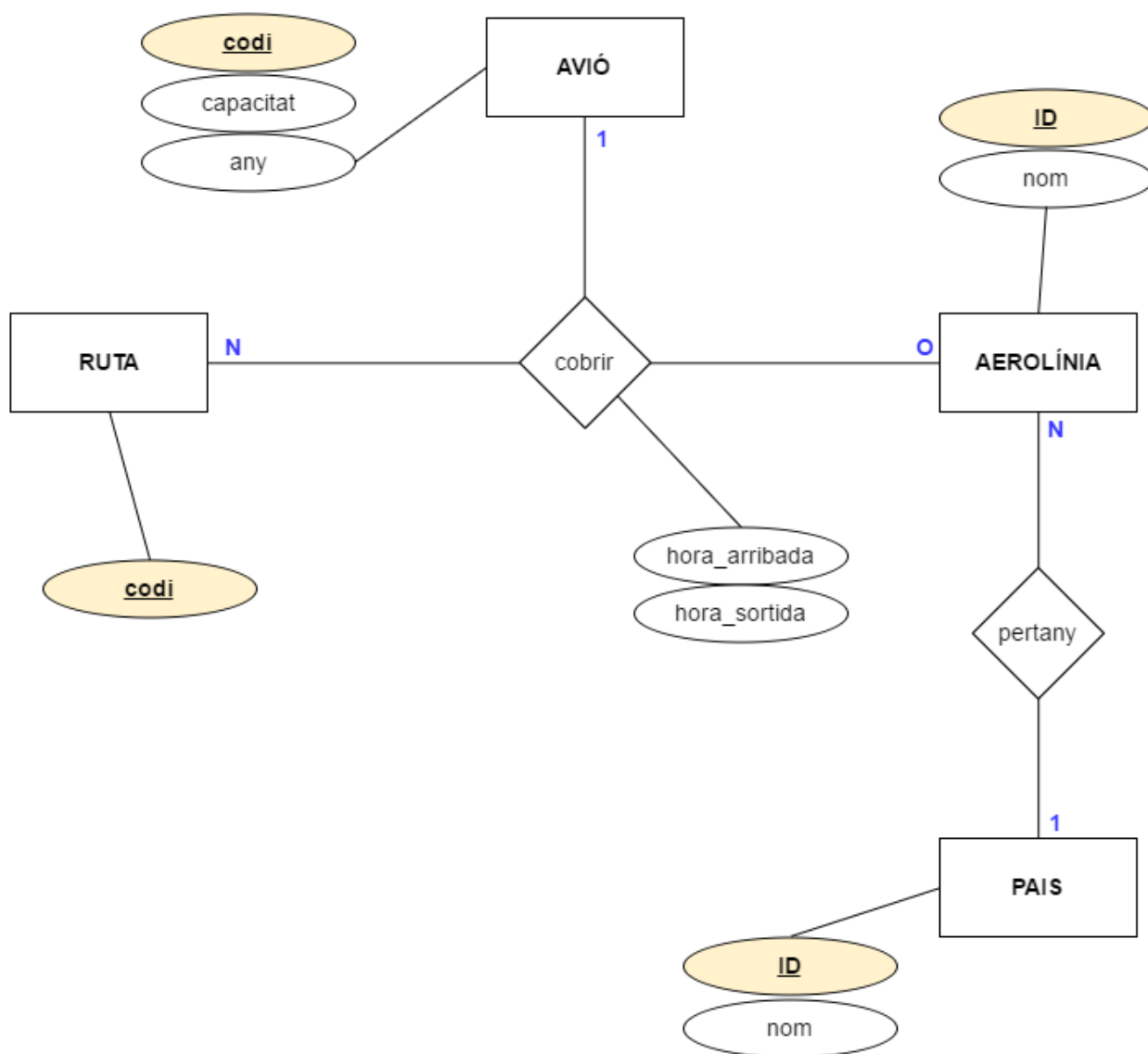
■ Consulta 2

Aquesta consulta torna el **codi de la ruta**, el **nombre d'aerolínies** que cobreixen la ruta, la **mitja de capacitat** que tenen els avions i l'**any de l'avió més vell**.

Les dades tornades seran de totes les rutes origen o destí Barcelona (BCN) i que la durada del vol sigui de **menys de 3 hores**.

Completeu, a partir de les consultes anteriors, el següent esquema de model E-R:

Il·lustració 1 Diagrama E-R a completar



Entitats

AVIÓ

- **codi** : **Identificador**. Codi que identifica el avió.
- **capacitat**: atribut que indica la capacitat de viatgers que té l'avió
- **any**: atribut que indica l'any de construcció de l'avió

AEROLÍNIA

- **ID** : **Identificador**. Codi que identifica la aerolínia.
- **nom**: atribut que indica el nom de l'aerolínia

PAIS

- **ID** : **Identificador**. Codi que identifica el país.
- **nom**: atribut que indica el nom del país.

RUTA

- **codi** : **Identificador**. Codi únic de ruta

Relacions

PERTANY : PAIS / AEROLÍNIA (1:N)

Una aerolínia pertany a un país (1:1)

A un país poden haver diferents aerolínies (1:N)

COBRIR : AVIO / RUTA / AEROLÍNIA (1:N:M)

Un avió pertany a una aerolínia (1:1)

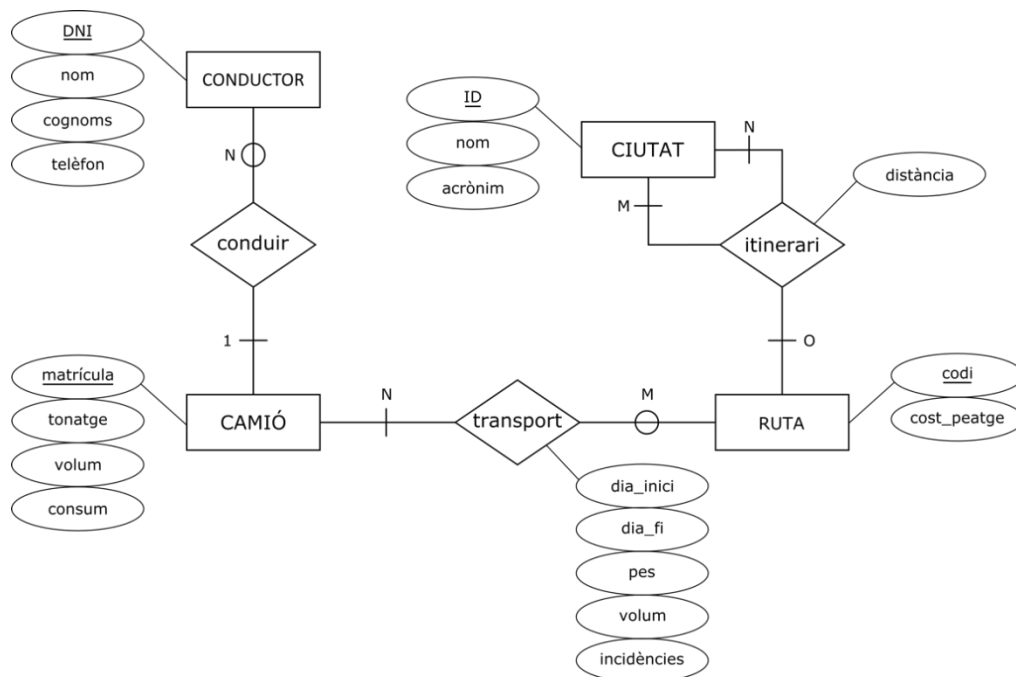
Un avió pot volar en diferents rutes (1:N)

Una ruta pot ser operada per diferents aerolínies (1:M)

- **hora_arribada** : atribut que indica la hora en la que arriba el vol.
- **hora_sortida**: atribut que indica la hora de enlairament del vol.

B.- Practicar amb el llenguatge SQL

Restaeu la còpia de seguretat de la base de dades guardada a l'arxiu `rutes.sql` a una base de dades buida amb el nom **Rutes**. Aquesta còpia de seguretat conté una instància del model E-R que es presenta a la il·lustració.



Il·lustració 2 Diagrama E-R de la base de dades Rutes

No és necessari mostrar en aquest document el resultat de la solució.

B.1 – Presenteu un llistat de camions ordenat de forma descendent pel seu volum de càrrega. Presenteu només els 3 primers resultats.

```
SELECT *
FROM camió
ORDER BY volum DESC
LIMIT 3;
```

Per ordenar en forma descendent es va servir la clàusula **DESC**, i amb **LIMIT** podem indicar quants registres retorna.

B.2 – Seleccionar el nom i el cost en peatges de les rutes que passen per la ciutat de “Lleida”.

```
SELECT DISTINCT r.nom, r.cost_peatge
FROM itinerari i
  INNER JOIN ruta r ON (i.codi_ruta = r.codi_ruta)
  INNER JOIN ciutat c
    ON (c.id = i.ciutat_destí OR c.id = i.ciutat_partida)
WHERE c.nom = 'Lleida';
```

Comparem el nom de la ciutat amb el literal ‘Lleida’, es important fer la unió de les taules tant quan la ciutat és origen com destí.

Gràcies a la instrucció **DISTINCT** aconseguim només els valors diferents

B.3 – Mostreu un llistat on es presentin les matrícules dels camions que han realitzat, o estan realitzant algun transport, el nom de la ruta que segueixen i les dates en les que transcorre el transport.

```
SELECT t.matrícula,
       r.nom,
       t.data_inici,
       t.data_finalització
FROM transport t INNER JOIN ruta r ON (t.codi_ruta = r.codi_ruta)
ORDER BY r.codi_ruta, t.matrícula;
```

A l’entitat **TRANSPORT** tenim les dates de cada servei i la ruta. El nom de la ruta l’aconseguim unint l’entitat amb **RUTA** a través de l’atribut **codi_ruta**

B.4 – Calculeu el número de camions que no han realitzat cap transport o en tenen algun pendent de finalitzar.

```
SELECT count(*) as camions
FROM camió c LEFT JOIN transport t ON c.matrícula = t.matrícula
WHERE t.data_finalització IS NULL;
```

Un servei no estarà finalitzat si no té data de finalització, és a dir el camp està buit. Per contar els registres fem servir la instrucció **COUNT**. És important que la unió sigui un **LEFT JOIN** perquè si no si un camió no te cap transport no apareixeria en el llistat.

B.5 – Mostreu el nom dels conductors dels camions que han informat d’alguna incidència al realitzar un transport

```
SELECT DISTINCT concat(c.nom, ' ', c.cognoms) AS conductor
FROM conductor c
  INNER JOIN transport t ON (c.camió_matrícula = t.matrícula)
WHERE t.data_finalització is not null and
      (t.incidències is null or t.incidències = 'Cap incidència');
```

Per ajuntar dos atributs, es fa servir la instrucció **CONCAT**, per saber si un conductor té alguna incidència, s’ha de cercar a la taula **TRANSPORT**, el text ‘Cap incidència’ o el valor a **NULL**, indiquen que no tingut cap incidència.

B.6 – Modifiqueu la consulta anterior per mostrar, per cada conductor, el número d'incidències que han informat.

```
SELECT concat(c.nom, ' ', c.cognoms) AS conductor, count(*) AS incidències
FROM conductor c
INNER JOIN transport t ON (c.camió_matrícula = t.matrícula)
WHERE
    t.data_finalització is not null and
    not (t.incidències is null or t.incidències = 'Cap incidència')
GROUP BY c.nom, c.cognoms;
```

En aquest cas volem les que tenen incidència, per fer-ho neguem la comparació anterior amb NOT, per obtenir el nombre agrupem pel conductor i contem els registres.

B.7 – Mostreu un llistat on es vegin les dades dels camions que tenen més d'un conductor assignat.

```
SELECT c.*
FROM camió c inner join conductor n on c.matrícula = n.camió_matrícula
GROUP BY c.matrícula
HAVING count(c.matrícula) > 1;
```

La unió de camió i conductor ens permet saber quina camions condueixen cada conductor, si agrupem per matrícula i fem un COUNT, podem saber el nombre de conductors. Per filtrar els que tenen més d'un conductor, fem servir la instrucció HAVING count > 1 és a dir els que el resultat del COUNT sigui més gran de 1.

B.8 – Mostreu un llistat de les ciutats per les que passa l'itinerari en el que s'han reportat més incidències. Indiqueu, a més, la distància fins a la següent ciutat.

```
SELECT c.nom, i.distància
FROM itinerari i INNER JOIN ciutat c ON (c.id = i.ciutat_partida)
INNER JOIN (SELECT t.codi_ruta, count(t.incidències)
FROM transport t
WHERE t.incidències IS NOT NULL
AND t.incidències <> 'Cap incidència'
GROUP BY t.codi_ruta
ORDER BY 2 DESC
LIMIT 1) inc ON (i.codi_ruta = inc.codi_ruta);
```

Primer fem una consulta inicial (subconsulta) per trobar tots les rutes que tenen incidència i el nombre d'incidències, les dades les ordenem pel segon camp (nombre d'incidències) en ordre descendent ORDER BY 2 DESC, per la qual cosa el primer registre serà el que té més incidències, i posteriorment indiquem que només es tingui en compte el primer registres (LIMIT 1)

Un cop tenim aquestes dades les creuem amb la taula CIUTAT i ITINERARI per obtenir les ciutats que té la ruta i la distància.

B.9 – Seleccionar el DNI, nom i cognoms dels conductors que condueixen el camió en què s’ha fet el transport on millor s’ha aprofitat el pes de la càrrega en relació a la seva capacitat, de manera que la diferència entre el pes a transportar i el pes que pot transportar el camió al llarg de tot el recorregut hagi estat mínim. S’ha de recuperar, a més, la matrícula del camió i la *ratio*.

```
SELECT r.DNI,
       concat(r.nom, ' ', r.cognoms) AS conductor,
       c.matrícula,
       (c.tonatge - t.pes) / i.distància AS ratio
FROM   camió c
       INNER JOIN transport t ON t.matrícula = c.matrícula
       INNER JOIN conductor r ON r.camíó matrícula = c.matrícula
       INNER JOIN (SELECT codi_ruta, sum(distància) AS distància
                   FROM itinerari
                   GROUP BY codi_ruta) i
       ON i.codi_ruta = t.codi_ruta
ORDER BY 4
LIMIT 1;
```

Primer fem una subconsulta per obtenir per cada ruta el quilometratge total, que el farem servir per fer el càlcul del ratio. Les dades del conductor les hem d’obtenir a partir de la taula CONDUCTOR que la enllacem amb la taula CAMIÓ per saber la matrícula que ha conduït. Els diferents trajectes estan emmagatzemats a la taula TRANSPORT que la ajuntem amb la resta a partir de la clau matrícula.

Per retornar el més òptim ordenem el resultat en sentit ascendent per el camp calculat *RATIO* que està a la posició 4 (*ORDER BY 4*) i tornem un únic registre (*LIMIT 1*)

B.10 – Seleccionar el cost de tots els transports ja realitzats i el de tots els transports que queden per realitzar encara, suposant que el preu del gasoil és de 0.84€ el litre.

```
SELECT 'cost no finalitzat' as descripció,
       sum((rr.km * c.consum / 100 * 0.84) + rr.cost_peatge) AS cost
FROM   transport t, camió c,
       (SELECT i.codi_ruta, sum(i.distància) AS km, r.cost_peatge
        FROM itinerari i, ruta r
        WHERE i.codi_ruta = r.codi_ruta
        GROUP BY i.codi_ruta) rr
WHERE  c.matrícula = t.matrícula
       AND rr.codi_ruta = t.codi_ruta
       AND t.data_finalització IS NULL
UNION ALL
SELECT 'cost finalitzat',
       sum((rr.km * c.consum / 100 * 0.84) + rr.cost_peatge) AS cost
FROM   transport t, camió c,
       (SELECT i.codi_ruta, sum(i.distància) AS km, r.cost_peatge
        FROM itinerari i, ruta r
        WHERE i.codi_ruta = r.codi_ruta
        GROUP BY i.codi_ruta) rr
WHERE  c.matrícula = t.matrícula
       AND rr.codi_ruta = t.codi_ruta
       AND t.data_finalització IS NOT NULL;
```

Solució alternativa fent servir un CASE per saber el tipus I agrupant per aquest concepte

```
SELECT CASE
WHEN t.data_finalització IS NULL THEN 'cost no finalitzat'
ELSE 'cost finalitzat'
END AS descripció,
sum((rr.km*c.consum/100*0.84) + rr.cost_peatge) as cost
FROM transport t,
camió c,
(SELECT i.codi_ruta, sum(i.distància) AS km, r.cost_peatge
FROM itinerari i, ruta r
WHERE i.codi_ruta = r.codi_ruta
GROUP BY i.codi_ruta) rr
WHERE c.matrícula = t.matrícula AND rr.codi_ruta = t.codi_ruta
group by 1;
```

En aquest cas apporto dos possibles solucions, per una banda una UNIÓN filtrant per cadascun dels conceptes data_finalització igual a NULL per saber les que no s'han completat i diferent de NULL per les completades.

En totes dues solucions, primer es fa una subconsulta per aconseguir per cada ruta la distància total (sum(distància)) i el cost del peatge. Aquestes dades les creuem amb CAMIO i TRANSPORT filtrant per la data de finalització, fent un sumatori per calcular el consum de combustible + el peatge

```
sum((rr.km * c.consum / 100 * 0.84) + rr.cost_peatge)
```

rr.km	Quilometratge del recorregut
c.consum	Consum del camió cada 100km
0.84	Preu del litre de benzina
rr.cost_peatge	Preu del peatge

La segona proposta afegeix un camp calculat comparant la data de finalització i agrupant per aquest camp

La resposta a aquest exercici s'ha de presentar en un fitxer anomenat exercici_B.sql adjunt a la solució de la pràctica. En aquest fitxer s'han de presentar les consultes degudament indicades segons la plantilla que us presenta adjunta.

Se us facilita, també, un fitxer amb les respostes que s'obtidran de l'execució de les consultes per tal de que us serveixi de joc de proves.

El fitxer que entregueu amb les vostres respostes no ha de donar error en la seva execució. **Recordeu finalitzar les consultes amb un “;”.**

C.- Repassar aspectes de programació bàsica de PHP

Repasseu aspectes bàsics de la programació en PHP fent el següent exercici:

Creeu un document PHP `rutes.php` que, donat el DNI d'un conductor, recuperi de la base de dades la darrera ruta que va realitzar.

A més de la informació bàsica del conductor s'haurà de mostrar el llistat de ciutats per les que passa la ruta, de manera que, de forma dinàmica, es generarà un *carousel* de fotos.

En cas de no indicar el paràmetre, o passar un conductor que no existeixi a la base de dades o no hagi realitzat cap transport, l'*script* n'haurà de seleccionar un a l'atzar i mostrar un missatge d'alerta a l'usuari per a notificar sobre l'error.

Exemple:

Estimat usuari, el conductor sol·licitat no es troba dins de la nostra base de dades. Alternativament, s'ha seleccionat un a l'atzar:

EL CONDUCTOR VISUALITZAT ÉS Joan Muntada AMB DNI 52111111A

Exemples de crida:

- Mostrar la darrera ruta realitzada pel conductor Joan Muntada:

`http://eimtdbd.uoc.edu/~groomete/rutes.php?DNI=52111111A`

- Mostrar la darrera ruta realitzada per un conductor escollit aleatòriament per no haver-se passat cap dada com a paràmetre:

`http://eimtdbd.uoc.edu/~groomete/rutes.php`

Afegiu aquí la URL d'enllaç al vostre script `rutes.php` dins del vostre espai d'alumne:

<http://eimtdbd.uoc.edu/~ppuertase/rutes.php>

Crida conductor inexistent

<http://eimtdbd.uoc.edu/~ppuertase/rutes.php?DNI=46322426R>

Crida conductor existent

<http://eimtdbd.uoc.edu/~ppuertase/rutes.php?DNI=52333333C>

Crida conductor existent sense rutes

<http://eimtdbd.uoc.edu/~ppuertase/rutes.php?DNI=52222222B>

Afegiu aquí les consultes SQL que feu servir a l'script `rutes.php` indicant el seu objectiu dins de l'script i els paràmetres que se'ls passarà:

Consultes

La consulta uneix les taules **CONDUCTOR** i **TRANSPORT** a través de l'identificador **matrícula**, es fa un **LEFT JOIN** per saber si té o no transport, per mostrar un missatge o un altre. Es filtren les dades per la clau única **DNI** de l'entitat **CONDUCTOR**. Com només volem obtenir un conductor, limitem el nombre de resultats a 1. Per si el **DNI** el passen en minúscules el convertim a majúscules amb la funció **UPPER**

```
SELECT c.*, t.data_finalització
FROM conductor c LEFT JOIN transport t ON (c.camio_matrícula = t.matrícula)
WHERE dni = UPPER(?)
ORDER BY t.data_finalització DESC LIMIT 1
```

La consulta que busca el conductor aleatòriament, es similar a l'anterior, però s'elimina la condició **DNI = ?** i es fa una **INNER JOIN**, s'ordenen els registres fent servir la funció **rand()**. Fem servir **LIMIT 1** per tornar un únic conductor.

```
SELECT c.*
FROM conductor c, transport t
WHERE c.camio_matrícula = t.matrícula
ORDER BY rand() LIMIT 1
```

Un cop hem obtingut el conductor sigui per paràmetre o aleatòriament, cerquem totes les dades que volem mostrar a la pantalla. En el meu cas, he afegit a part de les ciutats de l'última ruta, les dades del consum, incidències, data del trajecte, etc.

Per obtenir les dades, primer creem una subconsulta que ens retorna l'últim transport que ha fet el conductor, això ho fem ordenant per la data de finalització en ordre descendent, es a dir que no es mostraran els trajectes en curs, i tornem el primer registre. El resultat el creuem amb les taules **RUTA** per obtenir el nom de la ruta, **ITINERARI** per obtenir la distància que hi en cada trajecte, amb **CIUTAT** per aconseguir el nom de les ciutats i l'acrònim per poder mostrar les imatges. Finalment es creua també amb **CAMIO** per obtenir el consum del camió i poder calcular la benzina que ha consumit.

En aquest cas la forma de fer les unions ha estat a la part del **WHERE** en comptes de fer-ho amb **INNER JOINS**, no hi ha diferències a fer-ho d'una forma o un altre ja que els **SGBD** ho gestionen indistintament.

Per donar format a les dates, s'ha fet servir la funció **DATE_FORMAT**

```
SELECT ci.*, r.codi_ruta, ru.nom AS nom_ruta,
DATE_FORMAT(r.data_inici, '%d/%m/%Y') AS data_inici,
DATE_FORMAT(r.data_finalització, '%d/%m/%Y') AS data_finalització,
it.distància, ru.cost_peatge, r.incidències, c.consum
FROM ciutat ci, itinerari it, ruta ru, camio c,
(SELECT t.*
FROM conductor c, transport t
WHERE t.matrícula = c.camio_matrícula AND c.DNI = ?
ORDER BY t.data_finalització DESC LIMIT 1) r
WHERE it.codi_ruta = r.codi_ruta
AND ci.id = it.ciutat_partida
AND ru.codi_ruta = r.codi_ruta
AND c.matrícula = r.matrícula;
```

Subconsulta per trobar
l'últim transport

En format de JOINS seria

```
SELECT ci.*, r.codi_ruta, ru.nom AS nom_ruta,
       DATE_FORMAT(r.data_inici, '%d/%m/%Y') AS data_inici,
       DATE_FORMAT(r.data_finalització, '%d/%m/%Y') AS data_finalització,
       it.distància, ru.cost_peatge, r.incidències, c.consum
FROM (SELECT t.*
      FROM conductor c, transport t
      WHERE t.matrícula = c.camió_matrícula AND c.DNI = ?
      ORDER BY t.data_finalització DESC LIMIT 1) r
      INNER JOIN camió c ON (c.matrícula = r.matrícula)
      INNER JOIN ruta ru ON (ru.codi_ruta = r.codi_ruta)
      INNER JOIN itinerari it ON (it.codi_ruta = r.codi_ruta)
      INNER JOIN ciutat ci ON (ci.id = it.ciutat_partida);
```

Estructura

1. Definició d'una classe ruta per emmagatzemar les dades
2. Declaració de variables
3. Inicialització de les dades de connexió i apertura de la connexió contra la base de dades
4. Analitzar si tenim paràmetre DNI, si el tenim mirem de trobar un conductor amb rutes, si no es troba el conductor o no té cap ruta o no ens han passat el paràmetre executem una consulta per un conductor aleatori
5. Recuperem els registres i omplim la variable ruta amb les dades
6. Generem les línies amb les dades obtingudes
7. Generem les línies amb les dades obtingudes
8. Alliberem recursos i tanquem la connexió a la base de dades

Codi

```
<!DOCTYPE html>
<?php
// Classe per emmagatzemar les dades obtingudes de la base de dades
class ruta {
    public $conductor = null;
    public $destins = null;
    public $nom_ruta = null;
    public $km = 0;
    public $inici = null;
    public $fin = null;
    public $consum = null;
    public $peatge = null;
    public $incidencia = null;
}
// Iniciem un objecte buit
$info_ruta = new ruta();

// Declaració de variables
$resultat = null; // Variable auxiliar per recuperar els registres
$aleatori = ""; // Emmagatzemarà el text a pintar si el conductor és aleatori
$destins = null; // Llista dels diferents destins
$bloc_li = null; // Bloc per compondre els tags <li> del carrussell
$bloc_img = null; // Bloc per compondre els tags de l'ordre de les imatges

// Variables de connexió
$mysql_server = "localhost";
$mysql_user = "root";
$mysql_pass = "";
$mysql_db = "rutes";
```

```
// Si estem al servidor de la UOC canviem les credencials
if ($_SERVER['HTTP_HOST'] == 'eimtdbd.uoc.edu') {
    $mysql_server = "localhost";
    $mysql_user   = "ppuertase";
    $mysql_pass   = "K0dLrh5W";
    $mysql_db     = "ppuertase";
}

// Generem el header de la pàgina
header ( 'Content-Type: text/html; charset="UTF-8"' );

// CODI PHP

// En aquest bloc haureu d'introduir les instruccions PHP necessàries per
// · connectar-vos a la base de dades
// · recuperar el DNI del conductor passat per GET (o seleccionar un de forma aleatòria si no es passa)
// · recuperar la darrera ruta seguida pel conductor

// Conectem a la base de dades indicant les credencials
if (! ($iden = mysqli_connect ( $mysql_server, $mysql_user, $mysql_pass )))
    die ( "Error: No se pudo conectar" );

// Selecciona la base de dades
if (! mysqli_select_db ( $iden, $mysql_db ))
    die ( "Error: No existe la base de datos" );

// Important indicar que les dades estan en UTF-8
mysqli_set_charset ( $iden, "utf8" );

// Comprovem si tenim el paràmetre DNI per consultar les dades
if (isset ( $_GET ['DNI'] )) {
    // Seleccionar el conductor, si no te rutes el camp t.data_finalització estarà a null
    $sentencia = "select c.*, t.data_finalització from conductor c left join transport t on (c.camión_matrícula =
t.matrícula) where dni = UPPER('". $_GET ['DNI'] . "')";
    "order by t.data_finalització desc limit 1 ";
    $resultat = mysqli_query ( $iden, $sentencia );
    if ($resultat && $resultat->num_rows > 0) {
        $info_ruta->conductor = mysqli_fetch_assoc ( $resultat );
        // Si no te data de finalització significa que no te cap ruta
        if ($info_ruta->conductor['data_finalització'] == null) {
            $aleatori = "Estimat usuari, el conductor sol·licitat ".
            $info_ruta->conductor ['nom'] . " " . $info_ruta->conductor ['cognoms'] .
            " (".$info_ruta->conductor ['DNI'] .")" .
            "no te cap ruta. Hem triat un altre conductor a l'atzar: <br><br> ";
            $resultat = null; // Posem la variable a nul perquè busqui un aleatori
        }
    }
}

// Si no hem obtingut un conductor o no han passat el paràmetre DNI la variable $resultat serà NULL o el nombre
// de registres serà zero

if (!$resultat || $resultat->num_rows == 0) {
    // Si no hem trobat el conductor la variable $aleatori estarà buida
    if ($aleatori == "")
        $aleatori = "Estimat usuari, el conductor sol·licitat no es troba dins de la nostra base de dades." .
        "Alternativament, s'ha seleccionat un a l'atzar: <br><br>";
    $sentencia = "select c.* from conductor c, transport t where c.camión_matrícula = t.matrícula ".
    "order by rand() limit 1 ";
    // Ejecutem la consulta per trobar un conductor aleatoriament
    $resultat = mysqli_query ( $iden, $sentencia );
    $info_ruta->conductor = mysqli_fetch_assoc ( $resultat );
}
```



```
// Consulta per trobar el detall de la ruta del conductor, ciutats, dates, cost, etc.
```

```
$sentencia = "SELECT ci.*, r.codi_ruta, ru.nom as nom_ruta, ".
    "DATE_FORMAT(r.data_inici,'%d/%m/%Y') as data_inici, ".
    "DATE_FORMAT(r.data_finalització,'%d/%m/%Y') as data_finalització, ".
    "it.distància, ru.cost_peatge, r.incidències, c.consum " .
    " FROM ciutat ci, itinerari it, ruta ru, camió c, " .
    " (SELECT t.* FROM conductor c, transport t " .
    " WHERE t.matrícula = c.camión_matrícula AND c.DNI = '".
    $info_ruta->conductor['DNI'] . "' " .
    " ORDER BY t.data_finalització DESC LIMIT 1) r " .
    "WHERE it.codi_ruta = r.codi_ruta AND ci.id = it.ciutat_partida " .
    " and ru.codi_ruta = r.codi_ruta " .
    " and c.matrícula = r.matrícula ";
```

```
$resultat = mysqli_query ( $iden, $sentencia );
```

```
if ($resultat) {
    $bloc_li = "";
    $bloc_img = "";
    $n = 0;
    while($destins = mysqli_fetch_assoc ( $resultat )) {
        // Per el primer registre agafem les dades fixes
        if ($n == 0) {
            $info_ruta->data_inici = $destins['data_inici'];
            $info_ruta->nom_ruta = $destins['nom_ruta'];
            $info_ruta->inici = $destins['data_inici'];
            $info_ruta->fin = $destins['data_finalització'];
            $info_ruta->consum = $destins['consum'];
            $info_ruta->peatge = $destins['cost_peatge'];
            $info_ruta->incidencia = $destins['incidències'];
        } else {
            // Afegim a la variable de la ruta un guió excepte per el primer registre
            $info_ruta->destins = $info_ruta->destins . ' - ';
        }
        // Afegim les ciutats a la variable de la ruta i els km
        $info_ruta->destins = $info_ruta->destins . $destins['acrònim'];
        $info_ruta->km = $info_ruta->km + $destins['distància'];

        // Generem les línies <li>
        $bloc_li = $bloc_li . "<li data-target='#myCarousel' data-slide-to='{ $n }'";
        // Si és la primera la posem com activa
        if ($n == 0) $bloc_li = $bloc_li . " class='active'";
        // Tanquem el tag </li>
        $bloc_li = $bloc_li . "></li>\n";

        // Generem el bloc de les imatges, el nom del fitxer és l'acrònim de la ciutat
        $bloc_img = $bloc_img . "<div class='item' ". (($n == 0) ? "active">\n" : ">\n") .
            "<p>". $destins['nom'] . "</p>".
            "<img src='http://eimtdbd.uoc.edu/~groomete/imatges/'.
            $destins['acrònim']. ".png' alt='".
            $destins['nom']. "' width='709' height='276'>\n</div>";

        $n = $n+1;
    }
}
```

```
?>
```


pàgina 17

</div>

[DBD] RUTES - PRA1

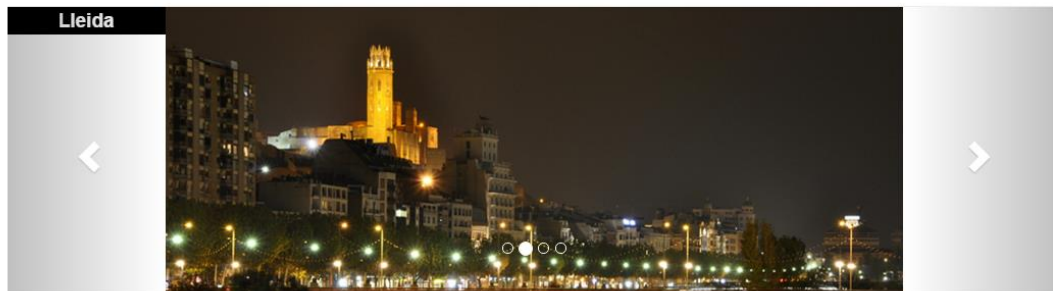
Pedro Puertas Estivill

Estimat usuari, el conductor sol·licitat Pere Font (52222222B) no te cap ruta. Hem triat un altre conductor a l'atzar:

Àlex Cabaner amb DNI : 52333333C

Última ruta : Ruta central (BCN - LLE - ZAR - MAD) 629 km Data : 10/02/2017 - 16/02/2017 Cost : 42.27€ (benzina) 30€ (peatge)

Incidències : **Hem patit una punxada a Saragossa**



Links Activitats:

Rutes

Rutes

Links Activitats:

Nota 1: Al servidor MySQL del vostre espai d'alumne no teniu permisos per a crear una nova base de dades. Haureu de restaurar la base de dades Rutes en la base de dades que ja teniu creada amb el vostre nom d'usuari. Un cop publiqueu el vostre *script*, haureu de tenir en compte de canviar les credencials d'accés.

Nota 2: Juntament amb l'enunciat se us passa un arxiu *rutes.php* que us servirà de plantilla per solucionar l'exercici. En aquest script trobareu indicats els espais on haureu de modificar el codi per tal de mostrar la informació recuperada de la base de dades.

A més, se us passa un arxiu comprimit amb les imatges associades a les ciutats registrades a la base de dades Rutes. Per referenciar-les a l'script PHP es pot generar el seu codi dinàmicament, passant el nom recuperat de la base de dades a l'etiqueta `` de HTML com per exemple:

```

```

Nota 3: L'*script* pujat al servidor web del vostre espai d'alumne haurà de poder-se visualitzar i tenir el nom *rutes.php*, mantenint el nom dels paràmetres d'entrada especificats a l'enunciat. No s'acceptaran solucions presentades amb noms diferents o scripts que no carreguin.