

Memòria de la pràctica

“Introducció a la “connexió” PHP i Oracle”

Marc Bramon Tarrés

17/1/2019

1. Introducció	3
2. Pàgines php	4
2.1. Esquema de les pàgines	4
2.2. Consideracions generals	4
2.2.1. Estil	4
2.2.2. Patró comú de programació de les pàgines	5
2.2.3. Variables de sessió	6
2.2.4. Mòduls comuns a totes les pàgines	6
2.3. Descripció de les pàgines	6
3. Base de dades	13
3.1. Modificacions de l'esquema inicial	13
3.2. Funcions d'accés a la base de dades	15
3.2.1. Funcions d'utilitat	15
3.2.2. Funcions de consulta	15
3.2.3. Funcions d'actualització	17

1. Introducció

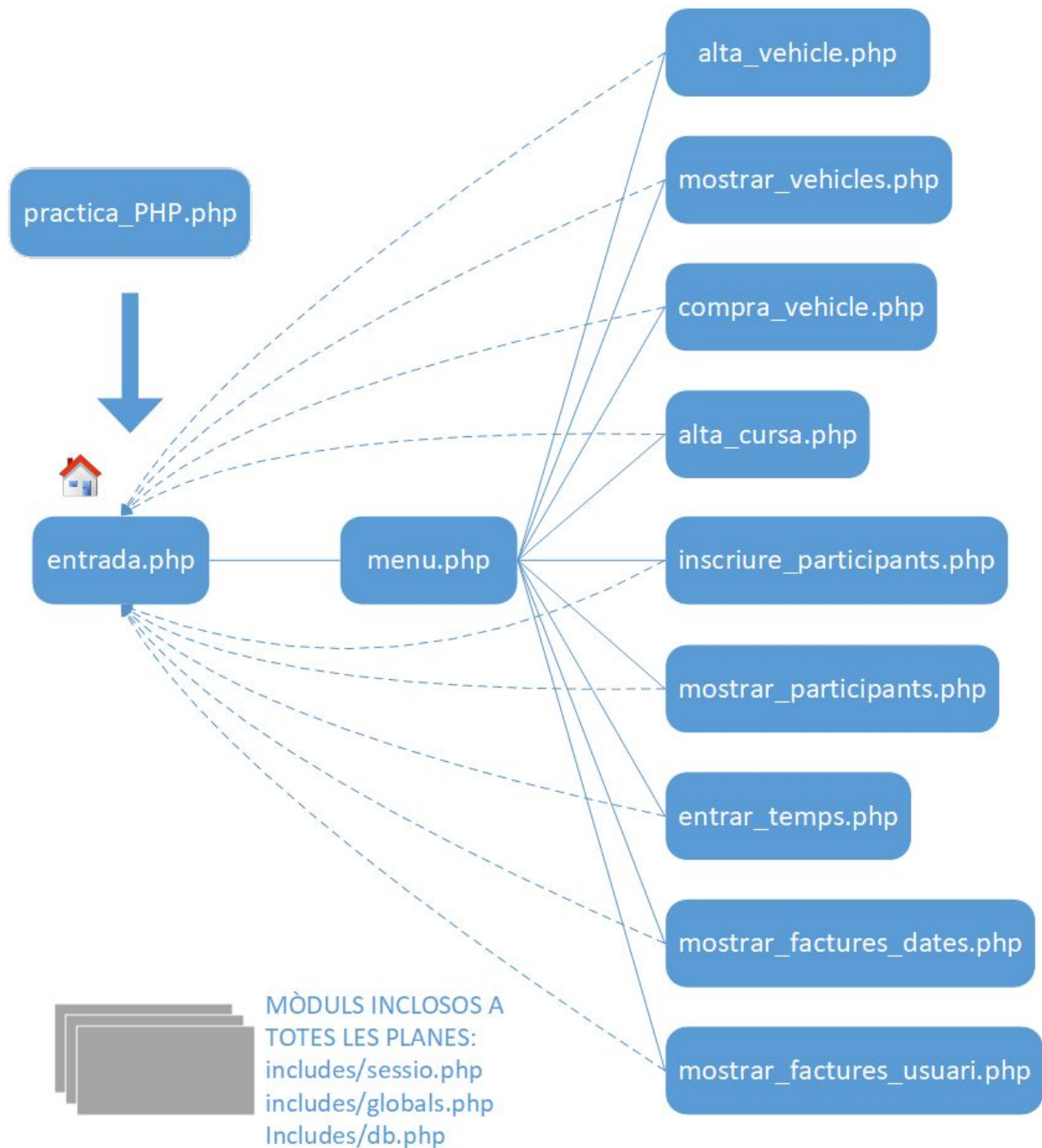
Aquest document descriu el desenvolupament que s'ha realitzat per fer la Pràctica d'introducció a la "connexió" PHP i Oracle.

En el primer lliurament es van implementar tots els apartats obligatoris de la primera part de la pràctica. En aquest segon lliurament s'ha implementat l'opció B "Facturació de combustible" i s'ha modificat el codi del primer lliurament per fer el tractament dels temps de cursa en format MM:SS.

Conté dos grans apartats: pàgines php i base de dades. En el primer s'expliquen els detalls que tenen a veure amb el codi php i el funcionament de les pàgines. I en el segon es detallen els desenvolupaments que estan relacionats més directament amb la base de dades: tant les modificacions de l'esquema original com les funcionalitats de consulta i actualització d'aquesta des de php.

2. Pàgines php

2.1. Esquema de les pàgines



2.2. Consideracions generals

2.2.1. Estil

- S'ha utilitzat la fulla d'estil estàndard Bootstrap 4. Això ens ha permès donar-li a l'aplicació una aparença atractiva i professional sense "massa" esforç.

- A nivell estètic, totes les pàgines segueixen una distribució d'elements comuna:

2.2.2. Patró comú de programació de les pàgines

Després d'uns quants assajos, de mirar exemples per Internet i de consultar documentació de l'assignatura, s'ha definit un patró de programació que s'ha procurat seguir en totes les pàgines. L'esforç per fer les primeres pàgines ha estat considerable, però després ens ha fet més fàcil programar totes les altres. El patró és el següent i per aquest ordre:

CODI PHP
Inclusió dels fitxers comuns: sessio.php, globals.php i db.php
Declaració i inicialització de les constants i variables globals de pàgina: <ul style="list-style-type: none"> - Títol de la pàgina - Variables de presentació que utilitzarem a la part HTML: arrays per desplegable i llistes, per exemple. - Array \$data que es carregarà amb les dades que ens enviï el formulari i que ens serveix per mantenir l'estat de la pàgina
Declaració de funcions. Bàsicament: <ul style="list-style-type: none"> - Funcions de inicialització i càrrega de \$data - Funcions de càrrega de les variables de presentació - Funcions de validació - Funcions d'execució d'accions
Control del fluxe: <ul style="list-style-type: none"> - Es carrega de _POST la variable \$accio - Es carreguen les variables de presentació - En funció de la variable \$accio s'executa el codi corresponent. En molts casos, aquest codi inclou: càrrega de \$data amb les dades del formulari, validació de les dades entrades i, finalment, execució de l'acció
HTML

Aquesta part conté bàsicament HTML amb petits fragments de PHP incrustats que són d'algun dels següents tipus:

- Valors de variables de plana declarats a la part PHP
- Inici i finalització de bucles per a les presentacions iteratives
- Condicions per decidir si mostrar un fragment o altre d'HTML

2.2.3. Variables de sessió

Només s'utilitzen 2 variables de sessió que es recullen a la pàgina d'entrada a l'aplicació: usuari i password. Ambdues ens serveixen per autenticar-nos quan ens connectem a la base de dades. I la variable usuari, a més, ens serveix per controlar l'accés a les pàgines. Si no està informada, enviem l'usuari a la pàgina d'entrada.

2.2.4. Mòduls comuns a totes les pàgines


Totes les pàgines inclouen el codi de 3 fitxers php que contenen variables o funcionalitats comunes a tota l'aplicació:

db.php	Conté totes les funcions d'accés a la base de dades. La descriurem amb molt més detall a l'apartat base de dades, però si que caldria comentar els avantatges que té centralitzar tots els accessos a la base de dades en un sol fitxer: <ul style="list-style-type: none">- Es poden reaprofitar accessos que s'utilitzen en més d'una pàgina- Es poden reprogramar o refinar consultes sense haver de modificar les pàgines que les usen.
globals.php	Constants globals de l'aplicació: títol, nom del servidor de base de dades, expressions regulars de validació de camps, iva i preu del servei.
sessio.php	Conté un fragment de codi que s'executa a l'inici de cada plana i que permet controlar l'accés a la mateixa. Bàsicament, si l'usuari no està autenticat, el redirigeix a la pàgina d'entrada.

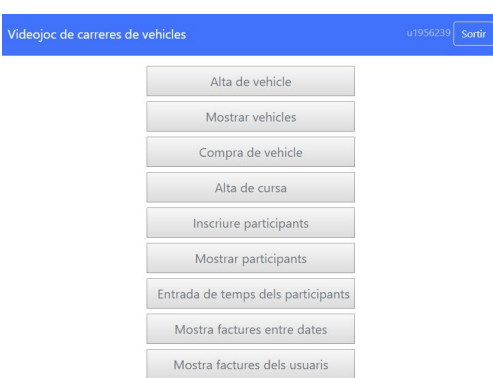
2.3. Descripció de les pàgines

Seguidament es descriuen totes les pàgines de l'aplicació. De cadascuna d'elles comentarem, si escau, quin és el flux principal, quines validacions fa i les funcions d'accés i actualització a la base de dades que utilitza. En aquest darrer cas només les esmentarem, donat que ja es descriuen amb més detall a l'apartat de base de dades.

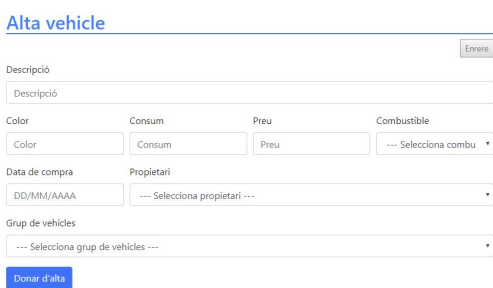
Entrada - entrada.php

	Flux principal
	<ul style="list-style-type: none"> - L'usuari entra usuari i password - Si tot és correcte, redireccionem a menu.php
	Validacions
	<ul style="list-style-type: none"> - Validem que es pugui fer una connexió a la bdd amb aquest usuari i password
	Funcions accés a la bdd
	<ul style="list-style-type: none"> - Només ens connectem des de la pròpia pantalla i, en aquest cas, no utilitzem cap funció de db.php

Menú - menu.php

	Flux principal
	<ul style="list-style-type: none"> - L'usuari selecciona una opció i l'aplicació el redirigeix a la pàgina corresponent
	Validacions
	-
	Funcions accés a la bdd

Alta vehicle - alta_vehicle.php







	Flux principal
	<ul style="list-style-type: none"> - L'usuari entra descripció, color, consum, preu, combustible, data de compra, propietari i grup de vehicles d'un vehicle i prem "Donar d'alta" - Si el formulari passa totes les validacions, es dona d'alta el vehicle a la bdd i es deixa la pantalla en disposició d'entra-ne un altre
	Validacions
	<ul style="list-style-type: none"> - Tots els camps han d'estar plens - Descripció i color → 45 caràcters màxim - Consum → num. 4 dígits i 2 decimals - Preu → num. 8 dígits i 2 decimals - Data de compra → Data DD/MM/AAAA - Propietari i grup de vehicles → Valors dels desplegable corresponents

	Funcions accés a la bdd
	llista_usuaris(), llista_grupsvehicles(), inserta_vehicle(\$a_dades_vehicle)

Mostrar vehicles - mostrar_vehicles.php

Mostrar vehicles

Entrar

Codi	Descripció	Color	Consum	Propietari	Fotografia
Toyota_04	Toyota Land Cruiser	blanc	13	Gemma Prats Pigem	
Toyota_01	Toyota Sequoia	gris	15	Pere Prats Pigem	
Toyota_02	Toyota Tacoma	verd	12	Rodrigo Benet Cantallops	
LRover_01	Land rover	festuc	2	Manel Perez Vidal	
Trek_01	Trek Domane	gris	0	Marta Peris Pujol	
Orbea_01	Orbea classic	blau	0	Jordi Perez Pals	

Flux principal

- Mostra una llista amb el codi, la descripció, el color, el consum, el propietari i la fotografia dels vehicles que estan habilitats

Validacions

Funcions accés a la bdd

llista_vehicles()

Compra de vehicle - compra_vehicle.php

Compra de vehicle

Vehicle

JDeere_01 - John Deere 7000

Entrar

Dades del vehicle

Codi	Descripció	Color
JDeere_01	John Deere 7000	vermell
Consum	Propietari	Preu
15	Rodrigo Benet Cantallops	35000

Dades de la compra

Comprador	Data venda	Preu	Nou codi
--- Selecciona ---			

Comprar

Flux principal

- L'usuari selecciona el vehicle a comprar d'una llista
- Se li mostren les dades del vehicle i se li demana que entri les dades de la compra: comprador, data de venda, preu i nou codi
- Si prem "Comprar" i les dades introduïdes passen les validacions, s'efectua la compra i es deixa la pantalla en disposició de fer una nova compra

Validacions

- Totes les dades de la compra són obligatòries
- Comprador → Valor del desplegable corresponent
- Data de venda → Data DD/MM/AAAA
- Preu compra → num. 8 dígit i 2 decimals
- Saldo de l'usuari > Preu compra
- Nou codi → Que no estigui duplicat

Funcions accés a la bdd

```
llista_vehicles_per_propietari(), llista_usuaris(),  
compra_vehicle($codivehicle,$comprador,$datavenda,  
$preu,$codi)
```


Alta cursa - alta_cursa.php

Alta cursa

Flux principal

- L'usuari entra codi, nom, premi, inscripció i data prevista d'inici i prem "Donar d'alta"
- Si el formulari passa totes les validacions, es dona d'alta la cursa a la bdd i es deixa la pantalla en disposició d'entra-ne una altra

Validacions

- Tots els camps han d'estar plens
- Codi → 15 caràcters màxim
- Nom → 45 caràcters màxim
- Premi → num. sencer positiu
- Inscripció → num. 8 dígit i 2 decimals
- Data prevista → Data DD/MM/AAAA

Funcions accés a la bdd

llista_curses(), inserta_cursa(\$data)

Inscriure participants - inscriure_participants.php

Inscriure participants

Usuari	Vehicle	Personatge
Rodrigo Benet Cantalops	John Deere 7000	--- Selecció ---
Jordi Perez Pals	Orbea classic	--- Selecció ---

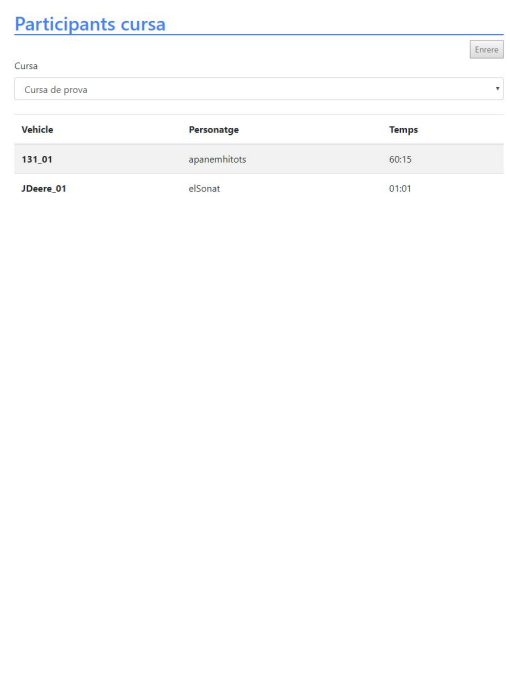
Flux principal

- L'usuari selecciona una cursa d'una llista que conté les curses que tenen inscripció oberta (data inici real a null)
- Omple la llista d'inscrits:
 - Selecciona el vehicle d'una llista de vehicles i prem "Afegir vehicle"
 - Selecciona el personatge de cada vehicle d'una llista que hi ha al costat del nom d'aquest i que conté els personatges del propietari del vehicle.
 - Repeteix l'operació anterior fins que ha afegit tots els vehicles
 - Sempre pot esborrar un vehicle de la llista prement el botó "Esborrar" que hi ha al costat de cada vehicle
- Finalment, l'usuari entra la data d'inici real de la cursa i prem el botó "Guardar participants i tancar inscripció" i, si les dades passen totes les validacions, s'enregistren els participants i s'actualitzen les taules corresponents a la bdd
- Es deixa la pantalla en disposició de fer les inscripcions en una altra cursa

Validacions

- Cal inscriure almenys un vehicle
- No es permeten vehicles duplicats ni es permet inscriure més d'un vehicle per

	propietari - El propietari del vehicle ha de tenir prou saldo per inscriure el vehicle a la cursa - Cada vehicle inscrit ha de tenir un personatge d'entre els personatges del propietari - Data d'inici real → Data DD/MM/AAAA
	Funcions accés a la bdd
	llista_curses_inscripcio_oberta(), llista_vehicles_per_propietari(), obtenir_vehicle(\$codi), obtenir_cursa(\$codi) tancar_inscripcio(\$cursa,\$inicireal,\$a_vehicles",\$a_personatges")

Participants cursa - mostrar_participants.php	
	Flux principal
	- L'usuari selecciona una cursa d'una llista - Se li mostren per pantalla el vehicle, el personatge i el temps dels participants en la cursa seleccionada - Cas que no hi hagi inscrits → missatge explicatiu - Cas que no s'hagin entrat els temps → Surten els participants amb temps="Pendent" - Cas que s'hagin entrat els temps → Els participants que no tenen temps surten amb temps="Abandonat" i la resta amb el temps de cursa
	Validacions
	Funcions accés a la bdd
	llista_curses(), llista_participants(\$cursa)

Entrar temps participants - entrar_temps.php

Entrar temps participants

Cursa

Vehicle	Personatge	Temps
307WRC_01	gustavo	29:05
500_01	jesuset	31:46
Civic_01	elQuiet	19:03
Clio_01	sargantana	27:18
Escort_01	elCec	15:24
Ibiza_01	bubu	22:48

Flux principal

- L'usuari selecciona una cursa d'una llista
- Si els temps ja han estat enregistrats, se li mostren per pantalla el vehicle, el personatge i el temps dels participants en la cursa seleccionada tal com es fa també a la pàgina mostrar_temps.php
- En cas contrari:
 - Surt la mateixa llista, però amb un camp de temps buit al costat de cada vehicle-personatge.
 - L'usuari omple els temps
 - Prem "Tanca taula de temps"
 - Si tot és correcte, s'enregistren els temps a la bdd* i es torna a mostrar la mateixa taula de temps, però ara sense poder-se modificar.

* Les accions addicionals que es desencadenen a l'enregistrar els temps es descriuen amb detall a l'apartat de base de dades.

Validacions

- Els camps de temps han d'estar o bé buits, o bé en format HH:MM

Funcions accés a la bdd

llista_curses_codi(), llista_participants(\$cursa),
actualitzar_temps(\$cursa,\$a_temps)

Factures entre dates - mostrar_factures_dates.php

Factures entre dates

Data inici Data fi

Codi	Cursa	Vehicle	Propietari	Data	Temps	Cost combustible	Servei	IVA	Total
1	ssdvsvsvsa	131_01	lolo92	30-DEC-18	10.01	1.15	10	21%	26.03
2	ssdvsvsvsa	JDeere_01	aaaauuuu	30-DEC-18	20.01	1.05	10	21%	37.52
3	ssdvsvsvsa	Tesla_01	granPili	30-DEC-18	20.01	.08	10	21%	14.04
4	prova	131_01	lolo92	15-JAN-19	55.95	1.15	10	21%	89.95

Flux principal

- L'usuari entra una data d'inici i una de fi
- Prem "Filtrar"
- Se li mostren per pantalla tots els camps de les factures entre aquestes dues dates (ambdues incloses)

Validacions

- Les dues dates amb format data DD/MM/AAAA
- La data d'inici ha de ser anterior o igual a la data de fi

Funcions accés a la bdd

llista_factures_entre_dates(\$data_inici,\$data_fi)

Facturació per usuari - mostrar_factures_usuari.php

Facturació per usuari

Usuaris

Enter

granPili - Pilar Mata Depera

Vehicle

Curses

Total

Tesla_01

Tesla S

1

14,04

Flux principal

- Es selecciona un usuari de la llista d'usuaris
- Es mostra per pantalla la facturació agrupada per vehicle amb els següents camps: vehicle, #curses, total facturació del vehicle.

Validacions

Funcions accés a la bdd

llista_vehicles_facturacio(\$usuari)

3. Base de dades

3.1. Modificacions de l'esquema inicial

El fitxer "pràctica_PHP.sql" conté les modificacions -amb comentaris que s'han d'efectuar sobre l'esquema inicial. Són:

Taula	Modificació
curses	<p>SQL:</p> <pre>alter table curses add tempsEnregistrats char(1) default 'N' NOT NULL;</pre> <p>update curses set tempsEnregistrats='S' where exists (select * from participantscurses pc where pc.cursa=curses.codi and not temps is null);</p> <p>Motivació:</p> <ul style="list-style-type: none">• Afegim la columna tempsEnregistrats• Aquesta columna ens permet saber si ja s'han entrat tots els temps dels participants d'una cursa.• Valors possibles: 'N'=No s'han entrat enregistrat els temps, 'S'=S'han enregistrat els temps• Tot i que els temps d'una cursa s'entren de cop, ens veiem obligats a afegir aquest camp per distingir les curses on tothom ha abandonat de les que no tenen cap temps entrat• Després d'afegir la columna, l'alimentem amb un valor calculat
vehicles	<p>SQL:</p> <pre>alter table vehicles add habilitat char(1) default 'S' not null;</pre> <p>Motivació:</p> <ul style="list-style-type: none">• Afegim la columna habilitat• Aquesta columna ens permet saber si un vehicle amb un propietari ha estat venut, tal com es demana en les especificacions de la pràctica.• Valors possibles: 'N'=Vehicle venut, 'S'=Vehicle no venut
factures	<p>SQL:</p> <pre>CREATE TABLE factures (codi NUMBER(8,0) CONSTRAINT cp_factures PRIMARY KEY, cursa VARCHAR2(15) NOT NULL, vehicle VARCHAR2(10) NOT NULL, propietari VARCHAR2(15) NOT NULL, data_factura DATE NOT NULL, temps NUMBER(6,3) NOT NULL, cost_combustible NUMBER(4,2) NOT NULL, preu_servei NUMBER(4,2) NOT NULL, iva NUMBER(3,0) NOT NULL, total NUMBER(8,2) NOT NULL, CONSTRAINT cf_factures_curses FOREIGN KEY (cursa) REFERENCES curses (codi), CONSTRAINT cf_factures_vehicle FOREIGN KEY (vehicle) REFERENCES vehicles (codi),</pre>

	<p>CONSTRAINT cf_factures_usuaris FOREIGN KEY (propietari) REFERENCES usuaris (alias));</p> <p><i>Motivació:</i></p> <ul style="list-style-type: none"> • Creem la taula factures • Inclou totes les columnes que es requereixen inicialment a l'enunciat de la pràctica. • S'han afegit, addicionalment, 2 columnes: codi i data_factura • El codi és de tipus sencer i actua com a clau primària. S'assigna automàtica i seqüencialment a partir de 1. • La data_factura és de tipus data i s'assigna automàticament amb la data actual quan es crea la factura • També afegim les claus foranes corresponents a les columnes: cursa, vehicle i propietari
--	--

3.2. Funcions d'accés a la base de dades

Com ja s'ha comentat, fitxer "db.php" conté totes les funcions d'accés a la base de dades que s'utilitzen a l'aplicació. Hi ha 3 grups de funcions: les funcions de consulta que retornen registres, les d'actualització de taules i un parell de funcions d'utilitat que són cridades internament per les dels altres 2 grups.

3.2.1. Funcions d'utilitat

<code>executa_query(\$sql)</code>	<p><i>PRE:</i> \$sql = string amb un comanda SELECT ben formatada</p> <p><i>POST:</i></p> <ul style="list-style-type: none">• Es connecta a la base de dades• Executa la comanda i carrega el resultat en un array• Tanca la connexió a la bdd• Retorna el resultat• En cas d'error, llança una excepció
<code>executa_cmd(\$conn,\$sql)</code>	<p><i>PRE:</i> \$conn = una connexió oberta a la base de dades \$sql = string amb un comanda SQL ben formatada</p> <p><i>POST:</i></p> <ul style="list-style-type: none">• Executa la comanda• En cas d'error, llança una excepció <p>Aquesta funció és la que utilitzem per executar instruccions d'actualització a la base de dades. Li passem la connexió perquè, en molts casos, llancem més d'una comanda d'actualització en el context d'una mateixa transacció</p>

3.2.2. Funcions de consulta

Tenen totes el mateix patró. Criden la funció "executa_query" i retornen el resultat d'aquesta. L'únic que les diferencia és la comanda sql que executen i els paràmetres que utilitzen per generar-la. Són:

Funció	\$sql
<code>llista_vehicles()</code>	<pre>select v.codi,v.descripcio,v.color,v.consum, u.nom ' ' u.cognoms as propietari, NVL(v.foto,g.foto) as foto from vehicles v join usuaris u on u.alias=v.propietari join grupsvehicles g on g.codi=v.grupvehicle where v.habilitat='S';</pre>
<code>llista_vehicles_per_propietari()</code>	<pre>select v.codi,v.descripcio,v.color,v.consum,v.propietari, u.nom ' ' u.cognoms as nompropietari, NVL(preu,0) as preu, NVL(v.foto,g.foto) as foto</pre>

	<pre> from vehicles v join usuaris u on u.alias=v.propietari join grupsvehicles g on g.codi=v.grupvehicle where v.habilitat='S' ". order by u.cognoms,u.nom,v.propietari,v.descripcio </pre>
obtenir_vehicle(\$codi)	<pre> select v.codi,v.descripcio,v.color,v.consum,v.propietari, u.nom ' ' u.cognoms as nompropietari, NVL(preu,0) as preu, NVL(u.saldo,0) as saldopropietari, NVL(v.foto,g.foto) as foto from vehicles v join usuaris u on u.alias=v.propietari join grupsvehicles g on g.codi=v.grupvehicle where v.codi='{\$codi}' </pre>
obtenir_cursa(\$codi)	<pre> select * from curses where codi='{\$codi}' </pre>
llista_personatges_usuari(\$usuari)	<pre> select * from personatges where usuari='{\$usuari}' </pre>
llista_usuaris()	<pre> select alias,nom ' ' cognoms as propietari,NVL(saldo,0) as saldo from usuaris order by cognoms,nom </pre>
llista_grupsvehicles()	<pre> select codi,descripcio from grupsVehicles order by descripcio </pre>
llista_combustibles()	<pre> select descripcio from combustibles order by descripcio </pre>
llista_curses()	<pre> select codi, nom, premi, inscripcio, iniciprevist from curses order by nom </pre>
llista_curses_inscripcio_oberta()	<pre> select codi, nom, premi, inscripcio, iniciprevist from curses where inicireal is null order by nom </pre>
llista_curses_codi()	<pre> select codi from curses order by codi </pre>
llista_participants(\$cursa)	<pre> select p.vehicle, p.personatge, (case when c.tempsenregistrats='N' then 'Pendent' when p.temps is null then 'Abandonat' else to_char(floor(temps),'FM00') ':' ". to_char(round((temps-floor(temps))*60),'FM00') end) as temps from curses c join participantscurses p on p.cursa=c.codi where c.codi='{\$cursa}' </pre> <p>En aquesta consulta cal destacar el càlcul de la columna temps en format HH:MM a partir dels minuts enregistrats a la base de dades</p>
llista_vehicles_facturacio(\$usuari)	<pre> select f.vehicle, min(v.descripcio) as nom_vehicle,count(*) as num_curses, sum(f.total) as total from factures f join vehicles v on v.codi=f.vehicle where f.propietari='{\$usuari}' group by f.vehicle order by sum(f.total) desc </pre>

llista_factures_entre_dates(\$data_inici,\$data_fi)	<pre>select codi,cursa,vehicle,propietari,to_char(data_factura,'DD/MM/YYYY') as data_factura, to_char(floor(temps),'FM00') ':' to_char(round((temps-floor(temps))*60),'FM00') as temps, cost_combustible,preu_servei,iva,total as data_factura_fmt from factures where data_factura>=to_date('{ \$data_inici}','dd/mm/yyyy') and data_factura<=to_date('{ \$data_fi } 23:59:59','dd/mm/yyyy HH24:MI:SS') order by data_factura, cursa,vehicle</pre> <p>En aquesta consulta cal destacar s'han de manipular les variables \$data_inici i \$data_fi, que són strings en format DD/MM/AAAA per aconseguir el filtre desitjat</p>
---	---

3.2.3. Funcions d'actualització

Totes elles segueixen també un patró comú: es connecten a la base de dades, executen una o n comandes de modificació i, si tot ha anat bé, fan commit. En cas contrari, fan rollback per no deixar les dades inconsistents.

inserta_vehicle(\$data)	<p><i>PRE:</i> \$data =array que conté les dades del vehicle a insertar</p> <p><i>POST:</i></p> <ul style="list-style-type: none"> • Es connecta a la base de dades • Calcula el codi del vehicle tal com s'especifica als requeriments de la pràctica • Inserta a la taula vehicles un vehicle amb el codi calculat anteriorment i la resta de dades que ens han passat a \$data • Tanca la connexió a la base de dades • En cas d'error, llança una excepció
inserta_cursa(\$data)	<p><i>PRE:</i> \$data =array que conté les dades de la cursa a insertar</p> <p><i>POST:</i></p> <ul style="list-style-type: none"> • Es connecta a la base de dades • Inserta a la taula curses una cursa amb les dades que ens han passat a \$data • Tanca la connexió a la base de dades • En cas d'error, llança una excepció
compra_vehicle(\$codivehicle,\$comprador,\$datavenda,\$preu,\$codi)	<p><i>PRE:</i> \$codivehicle =codi del vehicle a comprar \$comprador=codi de l'usuari comprador \$datavenda=data de la venda \$preu=preu de venda \$codi=nou codi de vehicle</p>

	<p>POST:</p> <ul style="list-style-type: none"> • Calcula el codi del propietari actual del vehicle • Es connecta a la base de dades • Inserta a la taula vehicles un nou vehicle amb les mateixes dades que el vehicle comprat, a excepció de les columnes següents, que alimenta com segueix: <ul style="list-style-type: none"> ◦ codi = \$codi ◦ datavenda = \$datavenda ◦ preu = \$preu ◦ propietari = \$comprador • Actualitza la columna habilitat='N' del vehicle venut (taula vehicles) • Decrementa el saldo del comprador amb el preu de venda (taula usuaris) • Incrementa el saldo del venedor amb el preu de venda (taula usuaris) • Fa commit • En cas d'error en alguna de les actualitzacions, fa rollback i llança una excepció
tancar_inscripcio(\$codi,\$inicireal,\$vehicles,\$personatges)	<p>PRE:</p> <p>\$codi =codi de la cursa \$inicireal=data d'inici real de la cursa \$vehicles=array amb els vehicles inscrits \$personatges=array amb els personatges corresponents</p> <p>POST:</p> <ul style="list-style-type: none"> • Carrega en una variable les dades de la cursa • Es connecta a la base de dades • Per cada fila de \$vehicles: <ul style="list-style-type: none"> ◦ calcula el propietari del vehicle ◦ inscriu el vehicle i el personatge a la cursa (taula participantscurses) ◦ decrementa el saldo del propietari del vehicle amb el preu d'inscripció de la cursa (taula usuaris) • Actualitza la data d'inici real de la cursa (taula curses) • Fa commit • En cas d'error en alguna de les actualitzacions, fa rollback i llança una excepció
actualitzar_temps(\$cursa,\$temps)	<p>PRE:</p> <p>\$cursa =codi de la cursa \$temps=array que conté el codi de vehicle, l'alias del personatge i el temps de cada participant en format MM:SS</p> <p>POST:</p> <ul style="list-style-type: none"> • Es connecta a la base de dades • Per cada fila de \$temps que no conté un temps en blanc: <ul style="list-style-type: none"> ◦ Converteix el temps de MM:SS a decimal ◦ Actualitza amb aquest valor el temps del participant corresponent (taula participantscurses) • Per cada fila de \$temps, inserta una factura (taula factures) alimentant les dades de la següent forma: <ul style="list-style-type: none"> ◦ codi=max(codis de factura+1) ◦ cursa=\$cursa, ◦ vehicle=vehicle de la fila ◦ propietari=propietari del vehicle

	<ul style="list-style-type: none"> ○ data_factura=data actual ○ temps=si el temps de la fila és buit → max(temps) de la cursa, en cas contrari → temps de la fila ○ cost_combustible=preu combustible del vehicle ○ preu_servei=10.0, ○ iva=21.0, ○ total=(preu_servei+cost_combustible*temps)*(100+iva)/100 ● Actualitza els següents camps de la cursa (taula curses): <ul style="list-style-type: none"> ○ millortemps = mín(temps dels participants) ○ tempsEnregistrats = 'S' ● Incrementa el saldo de l'usuari corresponent al vehicle guanyador amb el premi de la cursa (taula usuaris) ● Fa commit ● En cas d'error en alguna de les actualitzacions, fa rollback i llança una excepció
--	---