

10) Databáze a jejich implementace v MySQL-

-co to je MySQL a k čemu a kde se užívá, jaký je to model databáze, architektura jazyka, jak souvisí databázový systém se souborovým. (www.phpmyadmin.net)- prakt.cast?



-MySQL je implementace relační databáze*, která je plně kompatibilní s SQL standardem (standard = základy SQL jazyka, uplatnitelné v kterémkoliv systému založeném na SQL).- Švédská firma převzatá Oraclem [ˈprækəl]

*přesnější vysvětlení co jsou databáze obecně a relační model viz předchozí maturitní otázka

-multiplatformní databáze- funguje na více operač. systémech.

Každý klient má vlákno, přes které je připojen a klade své dotazy...důležité je i přihlášení a ověření autorizací k úpravám(v struktuře mají různí uživatelé různé možnosti a prvomoci dle nastavení správce)

-dvojitě licencování- my si ji můžeme stáhnout v omezenější verzi (GPL- general public license) a firma si jí musí koupit (komerční verze) a dostane k ní nějaký základní support atd.

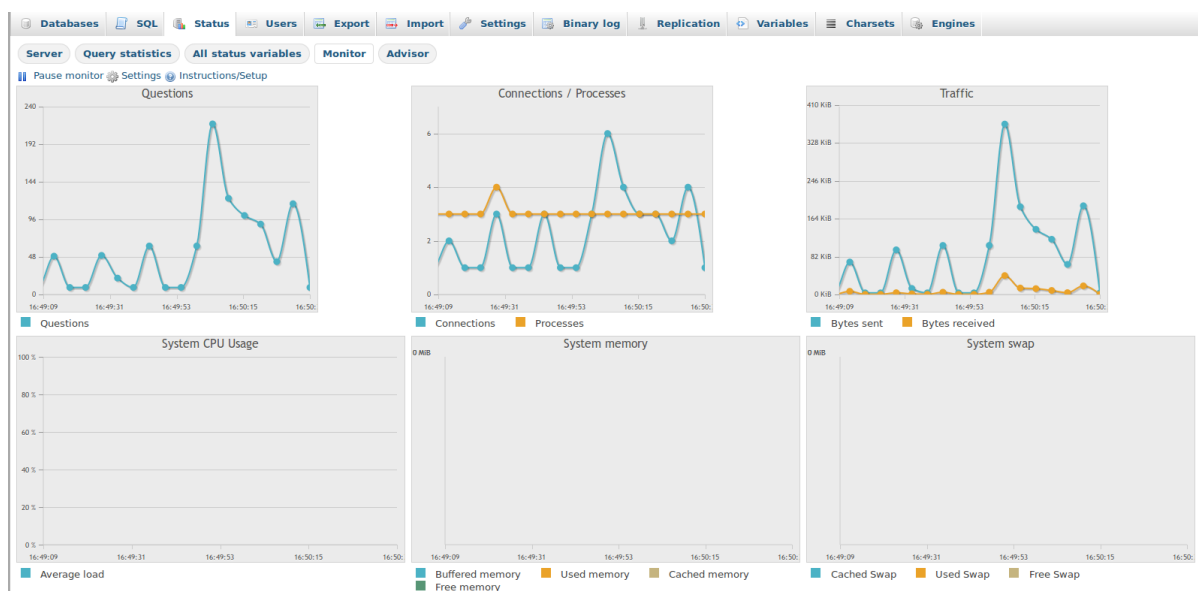
-MySQL databáze můžeme upravovat pomocí psaní příkazů například v příkazovém řádku nebo pokud neovládáme SQL programovací jazyk, tak použít například phpMyAdmin, což je uživatelské prostředí pro práci s prostředím, kde klikáme na potřebné možnosti a nemusíme psát příkazy, ty se vytvoří samy

-K ČEMU? – Jako všechny ostatní databáze slouží k ukládání, rychlému vyhledávání a využívání strukturovaných informací.

Výhody: - optimalizace datových struktur vzhledem k řešené problematice - menší systémové nároky

Nevýhody: - aplikační závislost, rozsah jejího využívání dokazuje převahu + nad -

-Kromě slučování a vybírání informací, MySQL také dělá statistiku (ke zvýšení efektivity, třídí hodnoty, dělá indexy), phpMyAdmin dokáže pro změnu například vytvořit i grafy zobrazující práci s daty v určitém období - viz ukázka z internetu↓



-KDE?- ve firmách, bankách, databáze zaměstnanců, zákazníků, katastry, e-shopy, fb, finance...prostě skoro všude-fenomén moderní doby...

-Databáze samotná a její servery jsou psány v programovacích jazycích C, C++. Pokud s databází pracujeme přes phpMyAdmin (což je uživatelské rozhraní k tomu určené), tak to je psáno v jazyce PHP (*Hypertext Preprocessor*), který je vhodný pro dynamické internetové stránky. S námi ale stále komunikuje na úrovni SQL (*Structured Query Language*), ani ten ale nutně nepotřebujeme, při základních příkazech si postačíme s klikáním a psáním do správných políček. Pokud píšeme příkazy/dotazy a nebo vkládáme data do databáze jen v příkazovém řádku, tak tam se žádné uživatelské rozhraní, které by nám pomohlo, nevyskytuje a ani PHP, tady už musíme znát základy jazyka SQL.

-Co je SQL???

(aj.- *Structured Query Language*)

-je to dotazovací jazyk, který využíváme v relačních databázích k práci s daty v tabulkách

-viz předchozí maturitní otázka: „(jedná se o dotazovací jazyk, tudíž když pak informace hledáme nebo chceme získat komplexnější data propojená z více tabulek, tak klademe dotazy v SQL a dostáváme na ně odpovědi)“

-*SEQUEL* (Structured English Query Language)-předchůdce SQL, 70. léta- výzkum relačních databází, snaha vytvořit programovací jazyk hodně podobný aj

-píšeme středníky na konci řádků, * označuje množinu všech položek třeba z určité tabulky atd.

-SQL používají např. databáze MySQL, ORACLE['brækəl], MS SQL, SERVER, SYBASE→ drobné odchylky syntaxu, drží ale stejný standard

-Základní architektura SQL!

-vše je v angličtině (...překvapivě)

Indexování tabulek příkaz - **CREATE** index

(Index je uspořádaný seznam všech hodnot jednoho nebo více sloupců: - rychlý přístup k datům tabulek - přístup do tabulek podle uspořádání - viz KLÍČE - primární, unikátní... opět viz předchozí maturitní otázka)

-Existuje pár základních příkazů, které se používají neustále (k manipulaci s daty - **SELECT**, **UPDATE**, **INSERT**, **DELETE**):

(dále existují příkazy pro definici dat: **CREATE** (vytvořit), **ALTER** (změna sloupců), **DROP** (zahodit tabulku) ; a další skupiny...)

--vytvoření báze dat (CREATE)

mysql> CREATE DATABASE `Nazev_databaze`; (tvorba databáze v SQL)

mysql> CREATE TABLE `tabulka1` (id int, jmeno varchar(222), prijmeni varchar(222));

(kód SQL pro vznik tabulky, viz následující obrázky postup v prostředí)

vidíme, že jsme v SQL, vytváříme tabulku, její název, sloupcečky a jejich formáty popřípadě jejich max. hodnotu

localhost / localhost / denisa | phpMyAdmin 4.6.6deb5 - Mozilla Firefox

localhost/phpmyadmin/db_structure.php?db=denisa&token=080c188c0d272f82fd9158579f79b0a3

Server: localhost:3306 » Database: denisa

Záložky: Structure SQL Search Query Export Import Operations Privileges Routines Events

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> firmy	★ Browse Structure Search Insert Empty Drop	5	InnoDB	latin1_swedish_ci	32 KiB	-
<input type="checkbox"/> jidelni_listek	★ Browse Structure Search Insert Empty Drop	7	InnoDB	latin1_swedish_ci	16 KiB	-
<input type="checkbox"/> obednavky	★ Browse Structure Search Insert Empty Drop	2	InnoDB	latin1_swedish_ci	16 KiB	-
<input type="checkbox"/> prirazeni_jedel	★ Browse Structure Search Insert Empty Drop	21	InnoDB	latin1_swedish_ci	16 KiB	-
<input type="checkbox"/> zamestnanci	★ Browse Structure Search Insert Empty Drop	3	InnoDB	latin1_swedish_ci	16 KiB	-
5 tables	Sum	38	InnoDB	latin1_swedish_ci	96 KiB	0 B

↑ ☐ Check all With selected: ▾

Print Data dictionary

Create table

Number of columns: 4

GO

localhost / localhost / denisa | phpMyAdmin 4.6.6deb5 - Mozilla Firefox

localhost/phpmyadmin/tbl_create.php?server=1&db=denisa&token=080c188c0d272f82fd9158579f79b0a3

Server: localhost:3306 » Database: denisa

Structure SQL Search Query Export Import Operations Privileges Routines Events

Table name: Add 1 column(s) Go

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index
id	INT		None			<input type="checkbox"/>	PRIMARY <input checked="" type="checkbox"/>
jmeno_prijmeni	VARCHAR	100	None			<input type="checkbox"/>	
handicap	TEXT		None			<input type="checkbox"/>	
zacatek_studia	DATE		None			<input type="checkbox"/>	

SAVE

dole pod tabulkou, ať už zrovna zadáváme cokoliv se nám nabízí i možnost nahlédnout jak vypadá kód SQL samotný (některé náležitosti zápisu můžeme vynechat):

Preview SQL

```
CREATE TABLE `denisa`.`studenti_septima` (
  `id` INT NOT NULL AUTO INCREMENT ,
  `jmeno_prijmeni` VARCHAR(100) NOT NULL ,
  `handicap` TEXT NOT NULL ,
  `zacatek_studia` DATE NOT NULL ,
  PRIMARY KEY (`id`)
) ENGINE = InnoDB;
```

Close

--vkládání dat (INSERT)

-v phpMyAdmin jsem v jakékoliv tabulce, mohu jít do záložky „SQL“, tam si vyberu možnost INSERT a prostředí už mi předepíše podle existujících položek jakousi „formu“:

```
INSERT INTO `zamestnanci` (`id`, `prijmeni`, `jmeno`, `rok_narozeni`, `fid`) VALUES ([value-1], [value-2], [value-3], [value-4], [value-5]);
```

Doplníme si na toto (vpišeme naše hodnoty):

```
INSERT INTO `zamestnanci` (`id`, `prijmeni`, `jmeno`, `rok_narozeni`, `fid`) VALUES (4,'Novak','Martin', 1950, 2);
```

V této záložce se tímto způsobem také můžeme učit to, jak příkazy vypadají, abychom později byli schopni psát si je už sami.

Když se objeví okénko Error, tak se z něj snažíme vyčíst, co je špatně...což občas trvá, takže pozor na to, že názvy sloupečků a tabulek mohou a nemusí být v těchto obrácených uvozovkách/ zpětných čárkách...zatímco jméno či příjmení, které vpisujeme do tabulky jsou vždy v 'obyčejných jednoduchých uvozovkách'...středníky vám zde doplní automaticky, ale v příkazovém řádku ne!

--aktualizace dat (UPDATE)

Showing rows 0 - 4 (5 total, Query took 0.0005 seconds.)

SELECT * FROM `firmy`

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

	id	jmeno	adresa
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	HP	Praha 6
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	P	Praha 16
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	H	Praha7
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	4	mmP	Praha 26
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	5	Hekl	Rakovník

Check all | With selected: Edit Copy Delete Export

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Query results operations

Print Copy to clipboard Export Display chart Create view

Moje tabulka firem...budu chtít jednu přejmenovat

takže kliknu na edit a v kolonkách přepíši hodnoty, mohu změnit název firmy, její ID nebo adresu, vedle upravovaných položek vidím jejich datové typy (např. jméno firmy, které měním je **varchar**- kratší textový typ), vše je celkem intuitivní, stačí se trochu zorientovat

Příkaz pro upravení dat by vypadal nějak takhle:

```
UPDATE `firmy` SET `jmeno`='APPLE' WHERE `id` = 3;
```

=“upravte firmy, nastavte jméno Apple k radku s číslem 3“ - 3 je klíč díky kterému program ví, který řádek má ve sloupečku Jméno změnit

--rušení dat (DELETE)

Run SQL query/queries on table denisa.studenti_septima:

```
DROP table `studenti_septima`;
```

SELECT * SELECT INSERT **DELETE** Clear

DROP je speciální funkce, tou mažeme tabulky. Když mažeme položky, tak postupujeme stejně jako u předchozích funkcí...vždy můžeme jít do struktury, kde se položka nachází a zmáčknout delete, viz další obrázek;))

smazání položky příkazem - např.: **DELETE FROM `firmy` WHERE `id` = 4;**

localhost / localhost / denisa / prirazení jídel | phpMyAdmin

localhost/phpmyadmin/sql.php?server=1&db=denisa&table=prirazení_jidel

Server: localhost:3306 » Database: denisa » Table: prirazení jídel

Tabulka přiřazení jídel

	jidlo	id_jidla
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	babovka	1
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	ptacek	2
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	raut	3
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	ryze	4
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	salat	5
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	svickova	6

rajská.....13
kachna.....20

--výběr z báze dat (SELECT)

-tady se konečně plně projeví relační databáze, propojení tabulek
-máme více tabulek a informace v nich jsou propojeny...neopakují se, jen odkazují na už existující položky

Objednávky

	patek	sobota	nedele	pondeli	utery	streda	ctvrtetek	stravnik_id
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	19	13	21	1	2	10	4	1
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	5	20	7	8	16	17	4	2

Zaměstnanci

	id	prijmeni	jmeno	rok_narozeni	fid
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	Novak	Petr	1960	5
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	Omacka	Karel	2000	1
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	Koblizek	Karel	1888	3
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	4	Novak	Martin	1950	2

+seznam přiřazení jídel je o 2 obrázky zpět

Nyní tyto 3 tabulky využijeme:

SELECT * FROM `obednavky` ; - tento příkaz nám ukáže co je v objednávkách, tedy tabulku kdo si co objednal

-vidíme, že v sobotu měl strážník s ID 1 jídlo 13 a strážník s ID 2 jídlo číslo 20

- teď si představíme kuchařku v jídelně, která chce vědět, kolik čeho má v sobotu vařit:

Showing rows 0 - 1 (2 total, Query took 0.0006 seconds.)

```
SELECT jidlo, COUNT(jidlo) FROM `obednavky`, `prirazeni_jidel` WHERE sobota=id_jidla GROUP by jidlo
```

Options: jidlo, COUNT(jidlo)

jidlo	COUNT(jidlo)
kachna	1
rajska	1

Příkaz, který vidíme výše, si můžeme přepsat do srozumitelných vět, abychom věděli, co se tam děje: VYBER(select) jídlo a ZPOČÍTEJ(count) ho. Potřebné informace si vezmi Z(from) tabulek „obednavky“ a „přiřazení jídel“, KDE(where) číslo v „objednávkách“ u soboty odpovídá číslu jídla v „přiřazení jídel“.

-obdobně mohu chtít vidět kdo si dal v sobotu tuhle kachnu a kdo rajsskou:

Showing rows 0 - 1 (2 total, Query took 0.0007 seconds.)

```
SELECT jidlo, jmeno, prijmeni FROM `obednavky`, `prirazeni_jidel`, `zamestnanci` WHERE sobota=id_jidla AND zamestnanci.id=stravnik_id
```

Options: jidlo, jmeno, prijmeni

jidlo	jmeno	prijmeni
rajska	Petr	Novak
kachna	Karel	Omacka

Opět: budeme chtít znát jídlo, jméno a příjmení - to jsou informace, které nalezneme v tabulkách ...xy... KDE(where) číslo v „objednávkách“ u soboty odpovídá číslu jídla v „přiřazení jídel“ AND (druhá doplňující informace) číslo strážníka v „objednávkách“ odpovídá id člověka v „zaměstnanci“ → taaa...Petr chce rajsskou a Karel kachnu:DD

BONUS: (--tvorba formulářů (FORMS)-nepatří do standardu)

Databázový systém se ukládá také do souborů, je souborově orientovaný. Když s ním uživatel pracuje, používané tabulky zůstávají v paměti, čímž se urychlí odezva systému, a když se systém vypíná, tak server uloží všechny informace na disk do souborů, ale ty nemají strukturu stejnou tomu, co vidíme v klientu či příkazovém řádku, existuje více způsobů u databází, jakými svá data ukládají, ale pro běžného uživatele to není podstatné. Složky jako takové mají hierarchickou/stromovou strukturu (ta už je v databázích dávno překonána), ale kromě toho, že si do nich databáze ukládá svá data, nemají moc společného. Složky neobsahují relace, ani funkce či software, který by kontroloval dodržování datových typů, zprostředkovával jejich úpravy, slučování a pokud nejsou na sdíleném disku, tak většinou nejsou ani přístupné komukoliv kdekoliv, popřípadě počítač nezjistí, jestli k nim máte povolený přístup.

A proč má MySQL ve znaku delfína?

Jeden ze spoluvůdců měl prostě delfíny rád, kvůli jejich inteligenci, rychlosti a dobré povaze. Původně měl skákat zleva doprava, stejně jako se píše, ale nakonec se rozhodli být více originální a inovativní a delfína otočit. Podle návrhů uživatelů bylo později vybráno pro delfína jméno Sakila.

Zdroje:

<http://www.adaptic.cz/webby/programovani-aplikaci/>

<http://maturita-vyt.buchtic.net/10.php>

<https://www.phpmyadmin.net/>

<https://cs.wikipedia.org/wiki/PHP>

<https://cs.wikipedia.org/wiki/PHP>

<https://www.youtube.com/watch?v=FR4QIeZaPeM>

<https://cs.wikipedia.org/wiki/SQL>

https://www.fi.muni.cz/usr/qdrasil/PV003_SLIDE.PDF

<https://infraadvisory.wordpress.com/2015/07/23/cpu-memory-stats-in-phpmyadmin/>