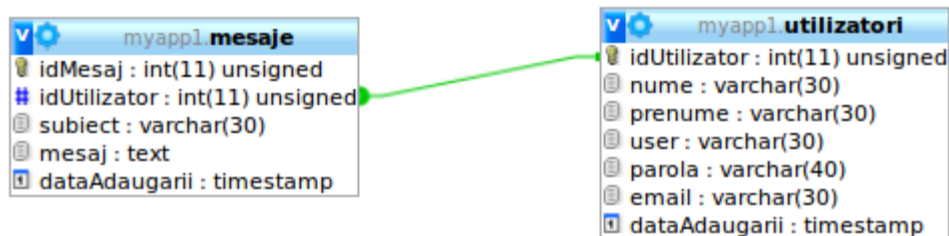


1. Creati designul unei baze de date pentru un forum de discutii. Parcurgeti urmatoarele etape:
 - 1.1. Schema conceptuala – stabilirea intrebarilor la care beneficiarul ar dori un raspuns.
Exemplu: Care sunt mesajele scrise de utilizatorul x? Cate mesaje sunt pe un anumit subiect?
Ce mesaje au fost postate in lna curenta? Etc.
 - 1.2. Entitati, relatii, attribute
 - 1.3. ERD-ul bazei de date (Diagrama Entitate – Relatie)
 - 1.4. Normaliza bazei de date: 1NF, 2NF, 3NF
 - 1.5. ERD-ul bazei de date normalizate
2. Implementati in MySQL ERD-ul din imagine. Pentru a putea stabili chei straine si restrictii de integritate, folositi tabele de tip InnoDB. Puteti folosi instrumentul Designer pentru stabilirea relatiilor.



3. Populati cu date tabelele. Testati respectarea integritatii referentiale impuse.
4. Pentru cerintele din tabel veti folosi SELECT si veti salva comenzile asociate cerintelor in fisierul select.txt.

Notiuni necesare:

Selectarea datelor din mai multe tabele

Produs cartezian

/*

*fara a folosi sintaxa JOIN, se extrag informatii din ambele tabele, fara nici o conditie.
fiecare inregistrare din primul tabel este combinata cu o inregistrare din cel de-al doilea*

*/

*SELECT * FROM mesaje, utilizatori*

INNER JOIN

Folosind sintaxa INNER JOIN, extragem fiecare inregistrare din primul tabel ce are ca pereche o inregistrare din cel de-al doilea tabel conform unei conditii.

De obicei, conditia folosita este: valoarea cheii primare din primul tabel (sa zicem coloana idUtilizator) este egala cu valoarea aceleiasi coloane (cheie straina) din al doilea tabel (coloana idUtilizator din al doilea tabel)

Exemplu:

```
SELECT *  
FROM mesaje m  
INNER JOIN utilizatori u  
ON m.idUtilizator = u.idUtilizator  
ORDER BY m.dataAdaugarii DESC  
LIMIT 0 , 30
```

LEFT JOIN

Sintaxa **LEFT JOIN** difera de **INNER JOIN** prin faptul ca selecteaza toate inregistrarile din primul tabel, indiferent ca au corespondent in cel de-al doilea sau nu.

Web2, curs05, Aplicatii

MySQL – DML.

Nota: Notiunile teoretice se regasesc si pe <https://www.invata-online.ro/php-mysql/baze-date-mysql/dml-data-manipulation-language>

Similar cu **INNER JOIN** se foloseste conditia de combinare a inregistrarilor din primul tabel cu cele din al doilea tabel, dar daca nu este gasita o inregistrare corespundenta in al doilea tabel se va afisa inregistrarea din primul tabel iar pe pozitiile campurilor din al doilea tabel se va afisa valoarea NULL.

Exemplu:

/*

adaug un utilizator cu idUtilizator = 999

*/

```
INSERT INTO `utilizatori` (`idUtilizator`, `nume`, `prenume`, `user`, `parola`, `email`, `dataAdaugarii`) VALUES ('999', 'cucu', 'mihai', 'cmihai', '1234', 'cmihai@gmail.com', now());
```

/*

adaug intai un mesaj ce nu are corespundenta in tabelul utilizatori pentru coloana idUtilizator = 999

*/

```
INSERT INTO mesaje (idMesaj, idUtilizator, subiect, mesaj, dataAdaugarii) VALUES (NULL, 999, 'subiect test', 'mesaj test', now());
```

SELECT *

FROM

mesaje m LEFT JOIN utilizatori u ON m.idUtilizator= u. idUtilizator

ORDER BY m.dataAdaugarii DESC

Alte exemple folosind INNER JOIN si LEFT JOIN:

idUtilizator

cu ajutorul LEFT JOIN vrem sa gasim DOAR inregistrarile din primul tabel ce nu au corespundenta in al doilea tabel folosind operatorul IS NULL

Putem folosi acest query pentru a identifica si a sterge mesajele utilizatorilor ce nu mai sunt in baza de date.

Desi, in mod normal, cand stergem un utilizator, trebuie sa cautam sa stergem si mesajele aferente (de exemplu, stergem utilizatorul cu idUtilizator = 999, apoi stergem din tabelul mesaje toate mesajele ce au idUtilizator = 999, operatie ce se poate face si in mod automat inpunand conditiile de integritate referentiala (valabile numai in INNODB)

*/

SELECT

*m. *,*

u.idUtilizator

FROM

mesaje m

LEFT JOIN utilizatori u ON m.idUtilizator = u.idUtilizator

WHERE

u.idUtilizator IS NULL

/*

extragem doar numele de utilizator, subiectul si corpul mesajului

vrem doar mesajele scrise dupa anumita data

*/

SELECT

u.username,

m.subiect,

m.mesaj

FROM

mesaje m

INNER JOIN utilizatori u ON m.idUtilizator = u.idUtilizator

WHERE

m.dataAdaugarii > '2009-04-01'

ORDER BY m.dataAdaugarii DESC

Cerintele urmatoare se vor aplica pe baza de date myapp1. Salvati comenzile SQL in fisierul sql.txt.

1.	Adaugati 3 mesaje pentru primul utilizator si cate un mesaj pentru al doilea si al treilea utilizator. Folositi comanda INSERT.	
2.	Produs cartezian pe tabelele utilizatori si mesaje	<i>SELECT * FROM mesaje, utilizatori</i>
3.	INNER JOIN pe tabelele utilizatori si mesaje	<i>SELECT * FROM mesaje m INNER JOIN utilizatori u ON m.idUtilizator = u.idUtilizator ORDER BY m.dataAdaugarii DESC</i>
4.	Toate mesajele unui utilizator cu id-ul dat	<i>SELECT u.idUtilizator,u.nume, u.prenume, m.subiect,m.mesaj FROM utilizatori u INNER JOIN mesaje m ON u.idUtilizator=m.idUtilizator WHERE u.idUtilizator=5</i>
5.	Numarul de mesaje scrise de fiecare utilizator	<i>SELECT u.idUtilizator, u.nume, u.prenume, count(*) AS `Numar mesaje` FROM utilizatori u INNER JOIN mesaje m ON u.idUtilizator = m.idUtilizator GROUP BY u.idUtilizator LIMIT 0 , 30</i>
6.	Utilizatorul/Utilizatorii care au scris un singur mesaj	<i>SELECT u.idUtilizator, u.nume, u.prenume, COUNT(m.idMesaj) AS `Numar mesaje` FROM utilizatori u INNER JOIN mesaje m ON u.idUtilizator = m.idUtilizator GROUP BY u.idUtilizator HAVING `Numar mesaje` =1 LIMIT 0 , 30</i>
7.	Utilizatorul cu cel mai mare numar de mesaje	<i>SELECT u.idUtilizator, u.nume, u.prenume, COUNT(m.idMesaj) AS `Numar mesaje` FROM utilizatori u INNER JOIN mesaje m ON u.idUtilizator = m.idUtilizator GROUP BY u.idUtilizator ORDER BY `Numar mesaje` DESC LIMIT 1</i>
8.	Utilizatori LEFT JOIN cu mesaje => toti utilizatorii si pentru fiecare utilizator mesajele scrise de acesta; daca un utilizator nu are mesaje, campurile corespunzatoare sunt NULL	<i>SELECT * FROM utilizatori u LEFT JOIN mesaje m ON m.idUtilizator = u.idUtilizator ORDER BY m.dataAdaugarii DESC LIMIT 0 , 30</i>
9.	Mesaje LEFT JOIN cu utilizatori => toate mesajele si pentru fiecare mesaj datele utilizatorului care l-a scris; daca am mesaj fara utilizator asociat, campurile corespunzatoare sunt NULL	<i>SELECT * FROM mesaje m LEFT JOIN utilizatori u ON m.idUtilizator =</i>

MySQL – DML.

Nota: Notiunile teoretice se regasesc si pe <https://www.invata-online.ro/php-mysql/baze-date-mysql/dml-data-manipulation-language>

		<i>u.idUtilizator ORDER BY m.dataAdaugarii DESC</i>
10.	Pentru utilizatorii care au mesaje, sa se extraga username, email, subiect si mesaj in ordine invers cronologica dupa dataAdaugarii. (se va folosi INNER JOIN)	
11.	Sa se extraga pentru toti utilizatorii username, email, iar daca au mesaje, subiect si mesaj (se va folosi LEFT JOIN).	
12.	Sa se extraga pentru toti utilizatorii username, email, iar daca au mesaje, subiect si mesaj, dar numai daca au fost adaugati luna aceasta.	
13.	Sa se extraga utilizatorii care nu au mesaje.	<i>SELECT u . * , u.idUtilizator FROM utilizatori u LEFT JOIN mesaje m ON u.idUtilizator = m.idUtilizator WHERE m.idUtilizator IS NULL LIMIT 0 , 30</i>
14.	Sa se extraga utilizatorii care au scris mesaje in luna curenta.	
15.	Sa se extraga utilizatorii care isi aniverseaza ziua de nastere astazi, pentru fiecare specificandu-se si varsta.	