

1.	Crearea bazei de date	<p><i>CREATE DATABASE nume_bd;</i></p> <p><i>ex: CREATE DATABASE `mylib`;</i> sau <i>CREATE DATABASE mylib;</i></p> <p>! Atentie la tipul ghilimelelor</p>
2.	Stergerea bazei de date	<p><i>DROP DATABASE nume_bd;</i></p> <p><i>ex: DROP DATABASE `mylib`;</i></p>
3.	Crearea unui tabel	<p><i>CREATE TABLE nume_tabel (nume_coloana1 tip_date1[(lungime)] [atribute_extra], nume_coloana2 tip_date2[(lungime)] [atribute_extra], [UNIQUE PRIMARY] KEY nume_index (coloana));</i></p> <p><i>Ex:</i> <i>CREATE TABLE utilizatori (</i> <i> utilizatorId int(11) unsigned NOT NULL auto_increment,</i> <i> username varchar(20) NOT NULL,</i> <i> email varchar(25) NOT NULL,</i> <i> varsta tinyint(4) unsigned default NULL,</i> <i> sex enum('m','f') NOT NULL,</i> <i> dataAdaugarii date NOT NULL,</i> <i> PRIMARY KEY (utilizatorId),</i> <i> UNIQUE KEY username (username),</i> <i> KEY email (email)</i> <i>);</i></p>
4.	Stergerea unui tabel	<p><i>DROP TABLE nume_tabel;</i></p> <p><i>Ex:</i> <i>DROP TABLE forum;</i></p>
5.	Redenumirea unui tabel	<p><i>ALTER TABLE nume_vechi RENAME nume_nou</i> sau <i>RENAME TABLE nume_vechi TO nume_nou;</i></p> <p><i>Ex:</i> <i>ALTER TABLE categorii RENAME domenii;</i> sau <i>RENAME TABLE domenii TO categorii;</i></p>
6.	Schimbarea definitiei unei coloane in tabel	<p><i>ALTER TABLE nume_tabel CHANGE nume_coloana_vechi nume_coloana_nou tip_date1[(lungime)] [atribute_extra]</i></p> <p><i>Ex:</i> <i>ALTER TABLE utilizatori CHANGE sex gen ENUM('m', 'f') DEFAULT NULL;</i></p>
7.	Adaugarea unei coloane in tabel	<p><i>ALTER TABLE nume_tabel ADD nume_coloana tip_date1[(lungime)] [atribute_extra] [AFTER nume_coloana_dupa_care_adaug]] [FIRST];</i></p> <p><i>Ex:</i></p>

		<p><i>ALTER TABLE utilizatori ADD telefon VARCHAR(20) NOT NULL AFTER email;</i></p> <p><i>ALTER TABLE utilizatori ADD telefon VARCHAR(20) NOT NULL FIRST;</i></p>
8.	Stergere unei coloane din tabel	<p><i>ALTER TABLE nume_tabel DROP COLUMN nume_coloana [. DROP COLUMN nume_coloana2, DROP COLUMN nume_coloana3..];</i></p> <p><i>Ex:</i> <i>ALTER TABLE utilizatori DROP COLUMN telefon;</i></p>
9.	Adaugarea unui index: INDEX, UNIQUE INDEX, PRIMARY KEY	<p><i>ALTER TABLE nume_tabel ADD UNIQUE (nume_coloana);</i></p> <p><i>Ex:</i> <i>ALTER TABLE `utilizatori` ADD UNIQUE (`utilizator`);</i></p>
10.	Inserarea datelor intr-un tabel	<p><i>INSERT INTO nume_tabel (coloana1, coloana2, ...) VALUES (valoare1, valoare2, ...);</i></p> <p>In cazul in care ometem anumite coloane de la insert, in ele se va adauga valoarea NULL daca attributele coloanei permit, sau valoarea DEFAULT implicita (specificata in definitia coloanei folosind atributul DEFAULT) sau cea explicita adica valoarea goala pentru acel tip de date.</p> <p>sau</p> <p><i>INSERT INTO nume_tabel VALUES (valoare1, valoare2, ...);</i></p> <p><i>! numarul de valori introduse trebuie sa fie egal cu numarul coloanelor tabelului</i></p> <p><i>Ex:</i> <i>INSERT INTO utilizatori (username, email, anNastere, gen, dataAdaugarii) VALUES ('vasile', 'vasile@yahoo.com', 1949, 'm', '2008-10-01');</i></p>
11.	Modificarea inregistrarilor tabel	<p><i>UPDATE tabel SET coloana1=valoare1, coloana2=valoare2... WHERE ...</i></p> <p><i>EX:</i> <i>UPDATE utilizatori SET username='test_modificat', email = 'test_modificat@yahoo.com' WHERE utilizatorId = 6;</i></p>
12.	Stergerea inregistrarilor din tabel	<p><i>DELETE FROM nume_tabel [WHERE conditie];</i></p> <p><i>Ex:</i> <i>DELETE FROM utilizatori WHERE utilizatorId = 6;</i> <i>DELETE FROM mesaje WHERE subiect LIKE '%test%';</i></p>

Selectarea datelor din baza de date

1.	Interogari simple	<p><i>SELECT lista_coloane FROM tabel [WHERE conditie] [ORDER BY coloana1,...[ASC DESC]] [LIMIT a,b]</i></p> <p><i>Ex:</i> <i>SELECT * FROM utilizatori</i> <i>WHERE nume LIKE '%ion%' OR prenume LIKE '%ion%'</i> <i>ORDER BY nume;</i></p>
2.	Alias pentru tabele	<i>SELECT u.username, u.anNastere FROM utilizatori u;</i>
3.	Alias pentru coloane	<i>SELECT nume, prenume, YEAR(dataAdaugarii) AS anulAdaugarii FROM `utilizatori`;</i>
4.	Interogari de tip total	<p><i>SELECT lista_coloane FROM tabel [WHERE conditie] [GROUP BY coloana expresie] [HAVING conditie_includere_grup] [ORDER BY coloana1,...[ASC DESC]] [LIMIT a,b]</i></p> <p>Lista de campuri din comanda SELECT poate include urmatoarele functii de tip total:</p> <ul style="list-style-type: none"> • COUNT(*) - numara cate inregistrari sunt selectate in fiecare grup. • COUNT(DISTINCT expr) - numara cate inregistrari pentru care expresia are o valoare distincta sunt selectate in fiecare grup. • AVG(expr) - calculeaza media aritmetica a valorilor expresiei pentru inregistrarile din fiecare grup. • MIN(expr) - calculeaza valoarea minima a expresiei pentru toate inregistrarile din fiecare grup. • MAX(expr) - calculeaza valoarea maxima a expresiei pentru toate inregistrarile din fiecare grup. • SUM(expr) - calculeaza suma valorilor expresiei pentru toate inregistrarile din fiecare grup.
5.	Produs cartezian: fiecare inregistrare din primul tabel este combinata cu o inregistrare din cel de-al doilea	<i>SELECT * FROM mesaje, utilizatori</i>
6.	<p>INNER JOIN</p> <p>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.</p> <p>De obicei, conditia folosita este: valoarea cheii primare din primul tabel (sa zicem coloana utilizatorId) este egala cu valoarea</p>	<p><i>SELECT</i> <i>*</i> <i>FROM</i> <i>mesaje m</i> <i>INNER JOIN utilizatori u ON m.utilizatorId = u.utilizatorId</i> <i>ORDER BY m.dataMesaj DESC</i></p> <p>extragem doar numele de utilizator, subiectul si corpul</p>

	<p>aceleiasi coloane (cheie straina) din al doilea tabel (coloana utilizatorId din al doilea tabel)</p>	<p>mesajului vrem doar mesajele scrise dupa anumita data</p> <pre> SELECT u.username, m.subiect, m.mesaj FROM mesaje m INNER JOIN utilizatori u ON m.utilizatorId = u.utilizatorId WHERE m.dataMesaj > '2009-04-01' ORDER BY m.dataMesaj DESC </pre>
7.	<p>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.</p> <p>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 corespondenta in al doilea tabel se va afisa inregistrarea din primul tabel iar pe pozitiile campurilor din al doilea tabel se va afisa valoarea NULL.</p>	<pre> SELECT * FROM mesaje m LEFT JOIN utilizatori u ON m.utilizatorId = u.utilizatorId ORDER BY m.dataMesaj DESC </pre> <p>DOAR inregistrarile din primul tabel ce nu au corespondenta in al doilea tabel folosind operatorul IS NULL:</p> <pre> SELECT m.*, u.utilizatorId FROM mesaje m LEFT JOIN utilizatori u ON m.utilizatorId = u.utilizatorId WHERE u.utilizatorId IS NULL </pre>