Daca avem identity, id se seteaza automat, incepand cu 1 si creste cu fiecare item;

**Sintaxa INSERT:**

INSERT INTO Sectiuni (nume, descriere) VALUES (‘sectiunea 1’, ‘cea mai mare sectiune’);

INSERT INTO Atractii (nume, descriere, varsta\_min, cod\_s) VALUES (‘rollercoaster’, ‘ok,’ 12, 1);

UPDATE face update la tabel

**Sintaxa UPDATE**

UPDATE Sectiuni SET descriere =’cea mai veche sectiune’

WHERE nume=”sectiunea 1”

Daca uitam WHERE se modifica tot (nu exista undo).

**Sintaxa DELETE**

DELETE FROM Sectiuni WHERE nume=’sectiunea 1’;

La fel ca la UPDATE, DELETE sterge tot daca uitam WHERE.

**Sintaxa SELECT**

SELECT \* FROM Sectiuni; //returneaza toate coloanele din table.

SELECT nume, descriere FROM Sectiuni //returneaza coloanele nume si descriere din table.

Pentru a extrage doar valorile distincte folosim DISTINCT.

SELECT DISTINCT varsta\_min FROM Atractii;

Pentru a returna doar inregistrari care indeplinesc anumite criterii, se foloseste WHERE

SELECT nume, descriere FROM Atractii WHERE varsta\_min=12;

SELECT \* FROM Atractii WHERE vasrta\_min>11 AND varsta\_min<16;

Strict mai mare decat 11 si strict mai mica decat 16 ^^^.

SELECT \* FROM Atractii WHERE varsta\_min BETWEEN 12 AND 15;

Cu BETWEEN e interval inchis;

SELECT \* FROM Atractii WHERE varsta\_min NOT BETWEEN 14 AND 18;

Returneaza toate atractiile care au varsta minima inafara intervalului inchis [14,18].

Returneaza toate atractiile care au varsta minima egala cu 12, 14 sau 16

SELECT \* FROM Atractii WHERE varsta\_min IN (12, 14, 16);

Diferita de 12, 14, 16

SELECT \* FROM Atractii WHERE varsta\_min NOT IN (12, 14, 16);

Pentru a specifica sabloane de cautare intr-o coloanal, vom folosi operatorul LIKE

SELECT \* FROM Vizitatori WHERE nume LIKE ‘A%’;

Returneaza vizitatorii al caror nume incepe cu A.

SELECT \* FROM Vizitatori WHERE nume LIKE ‘a%’;

Returneaza vizitatorii al caror nume se termina cu a;

SELECT \* FROM Vizitatori WHERE nume LIKE ‘%ana%’;

Returneaza toate numele care au ana in componenta : Ana, Iuliana etc.

SELECT \* FROM Vizitatori WHERE nume LIKE ‘\_na’;

Caracterul \_ inlocuieste un singur caracter, returneaza toate numele de 3 litere care se termina in -na;

SELECT \* FROM Vizitatori WHERE nume LIKE ‘[ABC]%’

Returneaza numele care incep cu A, B sau C;

SELECT \* FROM Vizitatori WHERE nume LIKE ‘[^ABC]%’;

Returneaza numele care nu incep cu A, B sau C (le ia pe restul);

SELECT \* FROM Vizitatori WHERE email IS NULL;

Returneaza toate inregistrarile care au email-ul cu valoarea NULL (initial este unknown)

SELECT \* FROM Vizitatori WHERE email IS NOT NULL; //cele care nu au email NULL

Daca dorim sa extragem numele categoriri numele vizitatorului si adresa de email pt toti vizitatorii care apartin unei categorii, vom executa urmatoare interogare:

SELECT C.nume AS categorie, V.nume, V.email FROM Categorii C, Vizitatori V WHERE C.cod\_c=V.cod\_c;

Interogarea poate fi rescrisa utilizand INNER JOIN

SELECT C.nume AS categorie, V.nume, V.email FROM Categorii C INNER JOIN Vizitatori V ON C.cod\_c=V.cod\_c;

Daca dorim sa afisam numele categoriilor si adresa de email a vizitatorilor, incluzand si categoriile care nu au vizitatori asociati (dar nu si vizitatorii care nu apartin unei categorii), vom executa urmatoare interogare:

SELECT C.nume, V.email FROM Categorii C LEFT JOIN Vizitatorii V ON C.cod\_c=V.cod\_c

Mai este RIGHT JOIN si FULL JOIN.

SELECT COUNT(\*) FROM Categorii;

Returneaza numarul total de inregistrari din tabelul Categorii

Exemple de functii de agregare: COUNT(), SUM(), AVG(), MIN(), MAX()…

Se folosesc de obicei impreuna cu clauzaele GROUP BY si HAVING

Ex. SELECT Localitate, COUNT(\*) [nr student] FROM Studenti GROUP BY localitate

Daca dorim sa afisam numele categoriilor si numarul de vizitatori pentru fiecare categorie care are cel putin un visitator, vom executa urmatoare interogare:

SELECT C.nume, COUNT(cod\_v) nr\_vizitatori FROM Categorii C INNER JOIN Vizitatori V ON C.cod\_c=V.cod\_c GROUP BY C.cod\_c, c.nume;

Daca dorim sa afisam numele categoriilor si numarul de vizitatorilor pt fiecare categorie (indifferent de nr de vizitatori)

SELECT C.nume, COUNT(cod\_v) nr\_vizitatori FROM Categorii C LEFT JOIN Vizitatori V ON C.cod\_c=V.cod\_c GROUP BY C.cod\_c, c.nume;