



Baze de date MySQL

Marin Bînzari
Pentalog



Cuprins

1. Ce este o bază de date ?
2. Proiectarea bazelor de date
3. Normalizarea
4. Relațiile : 1-1, 1-N, M-N, self-referencing
5. Tipuri de date
6. Index & Constrângeri de integritate
7. Tips

Ce este o bază de date ?





Baze de date

O bază de date este o colecție de date centralizate, creată și menținută computerizat, în scopul prelucrării datelor în contextul unui set de aplicații. Prelucrarea datelor se referă la operațiunile de introducere, ștergere, actualizare și interogare a datelor.

Ce oferă o bază de date :

1. Control centralizat al datelor
2. Viteză mare
3. Sunt compacte
4. Flexibilitate
5. Integritate





Tipuri de baze de date

Tipuri de baze de date :

1. Relaționale (MySQL, PostgreSQL)
2. Obiect orientate (MongoDB)
3. Graph (Neo4j, ArangoDB)
4. Key-Value (Redis, MemcacheDB)



Instalare MySQL

- `sudo apt-get update`
- `sudo apt-get install mysql-server`
- `mysql_secure_installation`

Proiectarea bazelor de date





Definiții

- Tabelă (entitate) este o colecție de informații logice relaționale tratată ca o unitate
- Înregistrare (n-uplu). O tabelă este compusă din înregistrări sau rânduri. Fiecare înregistrare este tratată ca o simplă unitate. Fiecare înregistrare este legată de înregistrări ale altei tabele.
- Câmpuri (atribute). Înregistrările sunt constituite din câmpuri (coloane) . Un câmp este o particulă atomică a bazei de date ce reprezintă cea mai mică cantitate de informație care poate fi manipulată. Toate înregistrările dintr-o tabelă au aceleași câmpuri.

Normalizarea



Date ne-normalizate

Următorul tabel conține date ne-normalizate. Coloanele 2, 3 și 4 conțin liste de valori, iar coloana 5 conține un atribut compus.

	Listă	Listă	Listă	Valori compuse				
Legitimație	Cod Calificări	Categorie Calificări	Profil	Nume	Vârstă	Birou nr.	Oraș	Superior
21	113	Sisteme	3	Mareș Ana	29	1	Iași	Ion
35	113 170 200	Sisteme Taxe Audit	5 7 4	Ionescu Dan	33	2	București	Damian
50	170	Taxe	3	Mircea Călin	35	2	București	Damian
77	150 200	Consultanță Audit	5 8	Traian Raluca	28	1	Iași	Ion

Prima formă normală (FN1)

O relație se prezintă în FN1 dacă valorile fiecărui câmp sunt ne-decompozabile (atomice) și fiecare tuplu este unic. Atributele care se pot partiționa în sub-atribute sau grupurile repetitive, care sunt foarte comune în bazele de date, nu sunt permise.

<u>Legitimatie</u>	<u>Cod Calificări</u>	Categorie Calificări	Profil	Nume	Prenume	Vârsta	Birou nr.	Oraș	Superior
21	113	Sisteme	3	Mareș	Ana	29	1	Iași	Ion
35	113	Sisteme	5	Ionescu	Dan	33	2	București	Damian
35	170	Taxe	7	Ionescu	Dan	33	2	București	Damian
35	200	Audit	4	Ionescu	Dan	33	2	București	Damian
50	170	Taxe	3	Mircea	Călin	35	2	București	Damian
77	150	Consultanță	5	Traian	Raluca	28	1	Iași	Ion
77	200	Audit	8	Traian	Raluca	28	1	Iași	Ion

A doua formă normală (FN2)

O relație este în FN2 dacă este în FN1 și toate atributele sale sunt dependente de întreaga cheie (adică, nici unul din atributele non-cheie nu este relaționat doar cu o parte a cheii). Tehnica de descompunere pentru obținerea FN2 este foarte simplă: presupune construirea unei relații separate care să înglobeze dependențele parțiale și să înlăture atributele dependente din relația originală.

Relația CADRE:

<u>Legitimatie</u>	Nume	Prenume	Vârsta	Birou nr.	Oraș	Superior
21	Mareș	Ana	29	1	Iași	Ion
35	Ionescu	Dan	33	2	București	Damian
50	Mircea	Călin	35	2	București	Damian
77	Traian	Raluca	28	1	Iași	Ion

A doua formă normală (FN2)

Relația **CALIFICĂRI**:

<u>Cod Calificări</u>	Categorie Calificări
113	Sisteme
170	Taxe
200	Audit
150	Consultanța

Relația **COMPETENȚE**:

<u>Legiti matie</u>	<u>Cod Calificări</u>	Competențe
21	113	3
35	113	5
35	170	7
35	200	4
50	170	3
77	150	5
77	170	8

A treia formă normală (FN3)

O relație este în FN3 dacă este în FN2 și nu există nici un fel de dependențe tranzitive (adică, nici unul din atributele non-cheie nu este dependent de alt atribut, care la rândul său este dependent de cheia relației).

Relația **CADRE:**

<u>Legitimăție</u>	Nume	Prenume	Vârsta	Birou nr.
21	Mareș	Ana	29	1
35	Ionescu	Dan	33	2
50	Mircea	Călin	35	2
77	Traian	Raluca	28	1

Relația **BIROURI:**

<u>Birou nr.</u>	Oraș	Superior
1	Iași	Ion
2	București	Damian

A treia formă normală (FN3)

Relația **CALIFICĂRI**:

<u>Cod Calificări</u>	Categorie Calificări
113	Sisteme
170	Taxe
200	Audit
150	Consultanța

Relația **COMPETENȚE**:

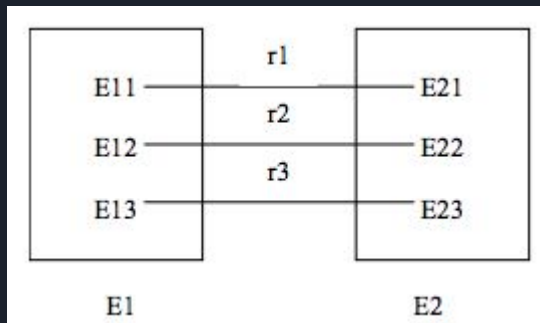
<u>Legiti matie</u>	<u>Cod Calificări</u>	Competențe
21	113	3
35	113	5
35	170	7
35	200	4
50	170	3
77	150	5
77	170	8

Relațiile : 1-1, 1-N, M-N,
unară



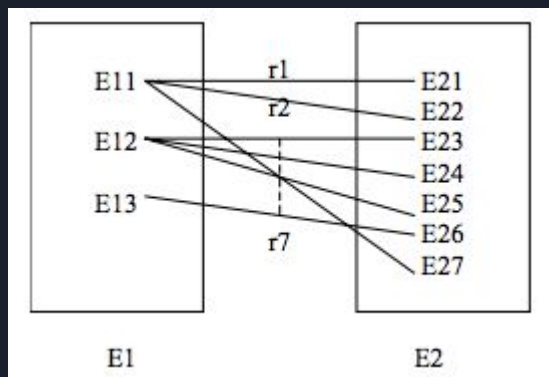
Relația „unul-la-unul” (1-1 sau one to one)

Relația „unul-la-unul” este cel mai simplu tip de relație. Ea este relația prin care unui element din mulțimea E1 îi corespunde un singur element din mulțimea E2 și reciproc.



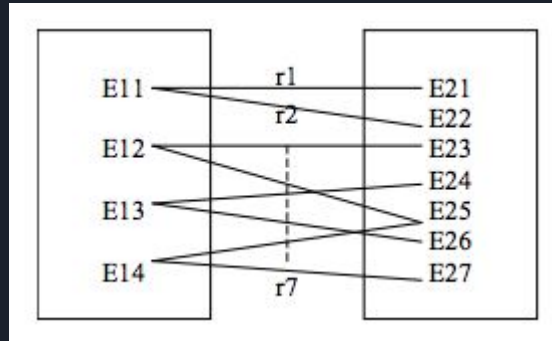
Relația „unul-la-multe” (1-N sau one to many)

Această relație este o relație prin care unui element din mulțimea E1 îi corespund unul sau mai multe elemente din mulțimea E2, dar unui element din mulțimea E2 îi corespunde un singur element din mulțimea E1.



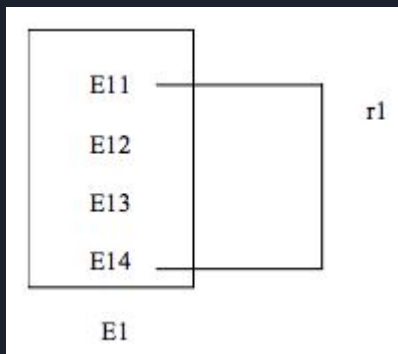
Relația „multe-la-multe” (M-N sau many to many)

Această relație este o relație prin care unui element din mulțimea E1 îi corespund unul sau mai multe elemente din mulțimea E2, și reciproc.



Relația unară (self-referencing)

Toate relațiile prezentate anterior sunt relații binare, având câte două relații implicate. Relațiile unare folosesc doar o singură relație, aceasta fiind asociată cu ea însăși.



Tipuri de date





Numere

Type	Length in Bytes	Minimum Value (Signed)	Maximum Value (Signed)	Minimum Value (Unsigned)	Maximum Value (Unsigned)
TINYINT	1	-128	127	0	255
SMALLINT	2	-32768	32767	0	65535
MEDIUMINT	3	-8388608	8388607 to	0	16777215
INT	4	-2147483648	2147483647	0	4294967295
BIGINT	8	-9223372036854775808	9223372036854775807	0	18446744073709551615
FLOAT	4	-3.402823466E+38	-1.175494351E-38	1.175494351E-38	3.402823466E+38
DOUBLE	8	-1.7976931348623 157E+ 308	-2.22507385850720 14E- 308	0, and 2.22507385850720 14E- 308	1.797693134862315 7E+ 308



Date

Types	Description	Display Format	Range
DATETIME	Use when you need values containing both date and time information.	YYYY-MM-DD HH:MM:SS	'1000-01-01 00:00:00' to '9999-12-31 23:59:59'.
DATE	Use when you need only date information.	YYYY-MM-DD	'1000-01-01' to '9999-12-31'.
TIMESTAMP	Values are converted from the current time zone to UTC while storing and converted back from UTC to the current time zone when retrieved.	YYYY-MM-DD HH:MM:SS	'1970-01-01 00:00:01' UTC to '2038-01-19 03:14:07' UTC



String

Types	Description	Range in characters
CHAR	Contains non-binary strings. Length is fixed as you declare while creating a table. When stored, they are right-padded with spaces to the specified length.	The length can be any value from 0 to 255.
VARCHAR	Contains non-binary strings. Columns are variable-length strings.	A value from 0 to 255 before MySQL 5.0.3, and 0 to 65,535 in 5.0.3 and later versions.
BINARY	Contains binary strings.	0 to 255
VARBINARY	Contains binary strings.	A value from 0 to 255 before MySQL 5.0.3, and 0 to 65,535 in 5.0.3 and later versions.



Text

Types	Description	Categories	Range in characters
BLOB	Large binary object that containing a variable amount of data. Values are treated as binary strings.You don't need to specify length while creating a column.	TINYBLOB	255 characters
		BLOB	65535 characters
		MEDIUMBLOB	16777215 characters
		LOB	4294967295 characters
TEXT	Values are treated as character strings having a character set.	TINYTEXT	255 characters (255 B)
		TEXT	65535 characters (64 KB)
		MEDIUMTEXT	16777215 characters (16 MB)
		LONGTEXT	4294967295 characters (4 GB)



ENUM

```
CREATE TABLE length ( length ENUM('small', 'medium', 'large') );
```

Max 64 choices.

Index & Constrîngeri





Index

```
CREATE INDEX [index name] ON [table name]([column name]);
```



Constrângeri de integritate

<https://www.w3resource.com/mysql/creating-table-advance/constraint.php>

```
CREATE TABLE [table name] ([column name] [data type]([size]) [column constraint] [table constraint] ([[column name].....]).....);
```

- NOT NULL
- UNIQUE
- PRIMARY KEY
- FOREIGN KEY
- CHECK
- DEFAULT



NULL CONSTRAINT

```
CREATE TABLE IF NOT EXISTS author
(
  id int(11) NOT NULL,
  name varchar(50) NOT NULL,
  email varchar(255) NOT NULL,
  country varchar(25) NOT NULL,
  home_city varchar(25) NOT NULL
);
```



UNIQUE

```
CREATE TABLE IF NOT EXISTS author  
(  
  id int(11) NOT NULL UNIQUE,  
  name varchar(50) NOT NULL,  
  email varchar(255) NOT NULL,  
  country varchar(25) NOT NULL,  
  home_city varchar(25) NOT NULL,  
  UNIQUE (email)  
);
```



PRIMARY KEY

```
CREATE TABLE IF NOT EXISTS author
(
  id int(11) NOT NULL AUTO_INCREMENT,
  name varchar(50) NOT NULL,
  email varchar(255) NOT NULL,
  country varchar(25) NOT NULL,
  home_city varchar(25) NOT NULL,
  PRIMARY KEY (id),
  UNIQUE (email)
);
```




FOREIGN KEY

FOREIGN KEY [column list] REFERENCES [primary key table] ([column list]);

```
CREATE TABLE IF NOT EXISTS book
(
  id int(11) NOT NULL PRIMARY KEY AUTO_INCREMENT,
  name varchar(50) NOT NULL,
  isbn_no varchar(15) NOT NULL UNIQUE,
  author_id int(11) NOT NULL,
  publication_date date,
  number_of_pages int(5) NOT NULL,
  FOREIGN KEY (author_id) REFERENCES author(id)
);
```



Check

```
CREATE TABLE IF NOT EXISTS book
(
  id int(11) NOT NULL PRIMARY KEY AUTO_INCREMENT,
  name varchar(50) NOT NULL,
  isbn_no varchar(15) NOT NULL UNIQUE,
  author_id int(11) NOT NULL,
  publication_date date,
  number_of_pages int(5) NOT NULL,
  FOREIGN KEY (author_id) REFERENCES author(id),
  CHECK(number_of_pages>0)
);
```



DEFAULT

```
CREATE TABLE IF NOT EXISTS author
(
  id int(11) NOT NULL AUTO_INCREMENT,
  name varchar(50) NOT NULL,
  email varchar(255) NOT NULL,
  country varchar(25) NOT NULL DEFAULT 'Moldova',
  home_city varchar(25) NOT NULL DEFAULT 'Chisinau',
  PRIMARY KEY (id),
  UNIQUE (email)
);
```



FOREIGN KEY - CASCADE & RESTRICT

NO ACTION === RESTRICT

```
CREATE TABLE IF NOT EXISTS book
(
  id int(11) NOT NULL PRIMARY KEY AUTO_INCREMENT,
  name varchar(50) NOT NULL,
  isbn_no varchar(15) NOT NULL UNIQUE,
  author_id int(11) NOT NULL,
  publication_date date,
  number_of_pages int(5) NOT NULL,
  FOREIGN KEY (author_id) REFERENCES author(id) ON UPDATE CASCADE ON
  DELETE RESTRICT,
  CHECK(number_of_pages>0)
);
```



FOREIGN KEY - SET NULL

```
CREATE TABLE IF NOT EXISTS book
(
  id int(11) NOT NULL PRIMARY KEY AUTO_INCREMENT,
  name varchar(50) NOT NULL,
  isbn_no varchar(15) NOT NULL UNIQUE,
  author_id int(11) NOT NULL,
  publication_date date,
  number_of_pages int(5) NOT NULL,
  FOREIGN KEY (author_id) REFERENCES author(id) ON UPDATE CASCADE ON
  DELETE SET NULL,
  CHECK(number_of_pages>0)
);
```



Database SQL

```
mysql> SHOW DATABASES;  
mysql> CREATE DATABASE web_academy;  
mysql> USE web_academy;  
mysql> SHOW TABLES;  
mysql> DROP DATABASE web_academy;
```

Tips





Export & Import

Dump the database

→ `mysqldump -h localhost -u username -p database_name -r "/home/mbinzari/dump.sql"`

Import database

→ `mysql -h localhost -u username -p database_name`

`mysql> SOURCE "/home/mbinzari/dump.sql";`