

MySQL

- **Partea II. Crearea bazei de date MySQL în Workbench**

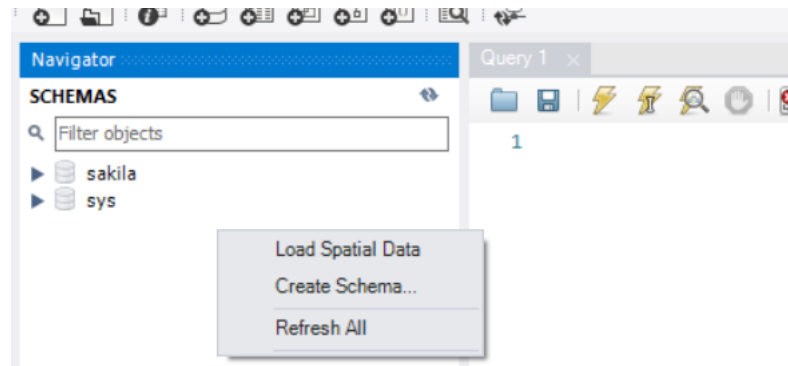
Ce ne așteaptă?

1. Crearea bazei de date
2. Crearea tabelelor
3. Tipuri de date în MySQL
4. Cheia primară
5. Cheia străină

1. Crearea bazei de date

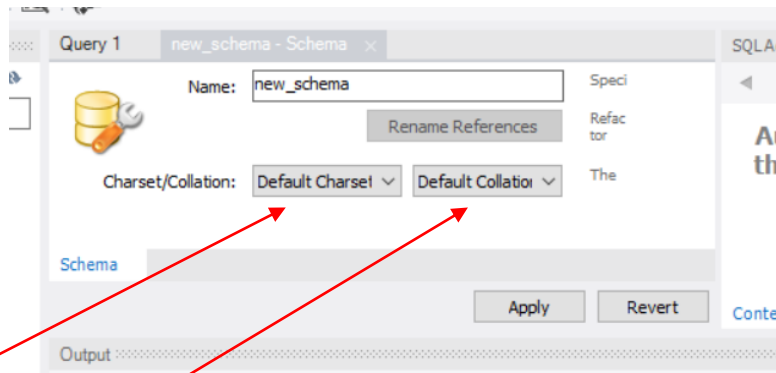
Crearea bazei fără componente

- MySQL Workbench permite crearea bazelor de date fără chiar a cunoaște limbajul SQL
- Inițial se creează o bază de date care nu va conține nici o componentă
- Se deschide Editorul SQL și în câmpul liber al submeniului Baze de date (Schemas) a câmpului Navigare se execută clic dreapta și se selectează opțiunea Create Schema...



Setul de caractere și regulile de corelare

- La crearea bazei se va specifica numele bazei, setul de caractere și regulile de corelare

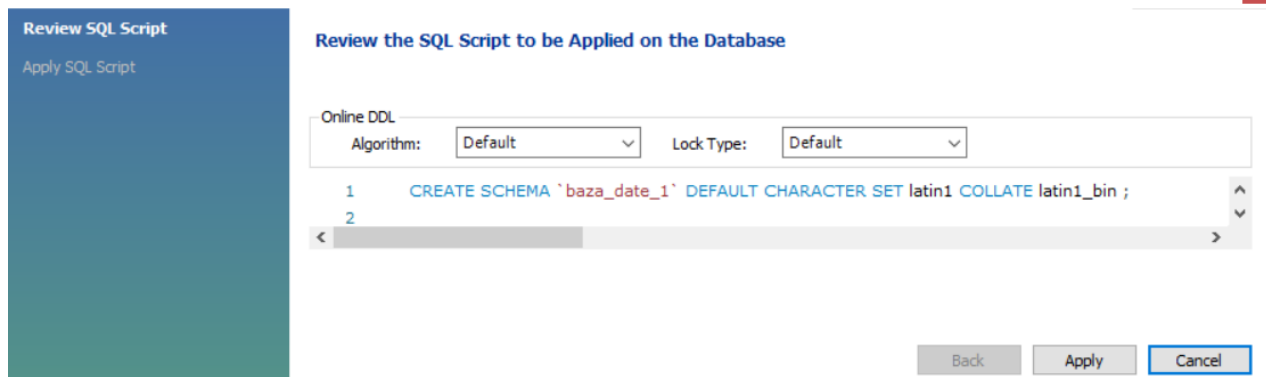


- Setul de caractere (Charset) – definește totalitatea caracterelor și valorilor lor codate (valorile în biți a caracterelor)
- Regulile de corelare (Collation) – definesc modalitatea de comparare a seturilor codate
- Setul de caractere și regulile de corelare sunt legate între ele

Generarea scriptului SQL

- Avantajul utilizării MySQL Workbench e că acesta creează în mod automat scriptul tuturor acțiunilor realizate.
- Serverul MySQL acceptă doar comenzi SQL
- După activarea butonului Apply în fereastra de stabilirea a numelui bazei de date, Workbench va genera interogările SQL de crearea a bazei de date

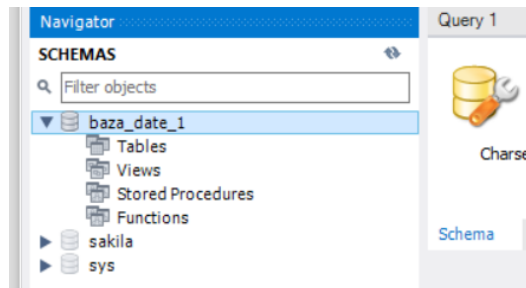
Apply SQL Script to Database



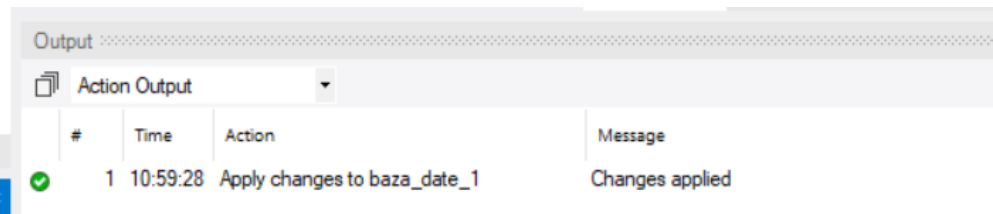
- Se activează butonul Apply pentru a transmite și executa interogările pe server

Vizualizarea componentelor bazei de date

- După crearea bazei de date în submeniul Baze de date (Schemas) a câmpului Navigare va apărea numele bazei de date.
- Pentru vizualizarea componentelor se va selecta semnul săgeții situat în fața numelui bazei



- În câmpul ieșiri (Outputs) se vor lista acțiunile întreprinse pe server



Stocarea fizică a bazei de date

- Pentru vizualizarea locației de stocare a bazei de date se va selecta opțiunea **Server Status** în submeniului **Administrare** a câmpului **Navigare** și în meniul **rulat** se va căuta **Data Directory**.

The screenshot displays the MySQL Workbench interface with the 'Server Status' window open. The window is titled 'MySQL Workbench' and shows the 'Server Status' tab selected in the left-hand navigation pane. The main content area displays the following information:

- Connection Name:** conex1
- Host:** LAPTOP-P63T64BB
- Socket:** MySQL
- Port:** 3306
- Version:** 8.0.29 (MySQL Community Server - GPL)
- Compiled For:** Win64 (x86_64)
- Configuration File:** unknown
- Running Since:** Sat Jul 2 09:21:18 2021 (3 days 1:46)

Below the connection details, the 'Available Server Features' section shows the status of various features:

Feature	Status
Performance Schema	On
Thread Pool	n/a
Memcached Plugin	n/a
Semisync Replication Plugin	n/a
SQL Availability	On
Windows Authentication	Off
Password Validation	n/a
Audit Log	n/a
Firewall	n/a
Firewall Trace	n/a

The 'Server Directories' section lists the following paths:

- Base Directory:** C:\Program Files\MySQL\MySQL Server 8.0\
- Data Directory:** C:\ProgramData\MySQL\MySQL Server 8.0\Data\
- Disk Space in Data Dir:** 34.80 GB of 146.05 GB available
- Plugins Directory:** C:\Program Files\MySQL\MySQL Server 8.0\lib\plugin\

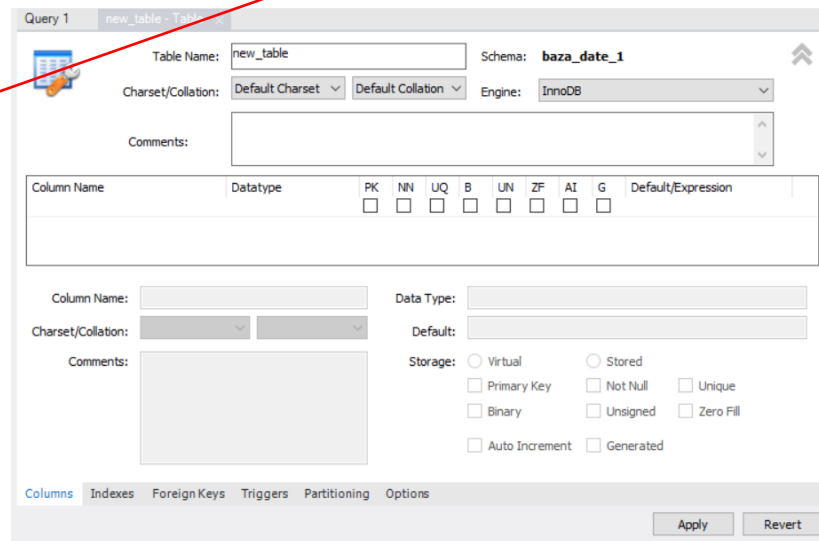
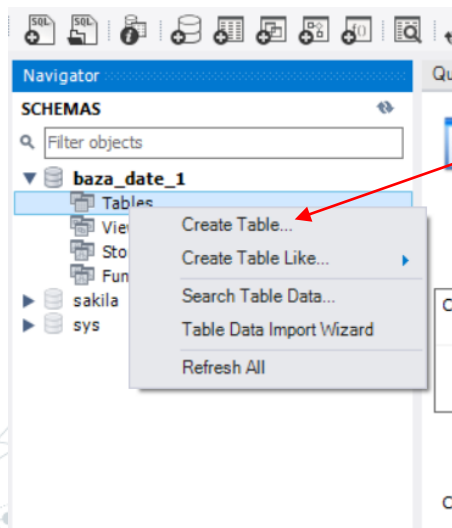
The right-hand pane of the 'Server Status' window displays various performance metrics, all of which are currently showing 'No Data' or '---':

- Server Status: Running
- CPU/Load: ---
- Connections: No Data
- Traffic: No Data
- Key Efficiency: No Data
- Selects per Second: No Data
- InnoDB Buffer Usage: No Data
- InnoDB Reads per Second: No Data
- InnoDB Writes per Second: No Data

2. Crearea tabelelor

Crearea tabelelor

- Pentru crearea tabelului se apasă clic drept pe componenta Tables a bazei de date dorite din submeniul Baze de date (Schemas) a câmpului Navigare și se selectează opțiunea Create Tables...
- Se deschide câmpul de creare a tabelului



Câmpul de creare a tabelului

- Se stabilește denumirea tabelului în câmpul Table Name
- Se verifică baza de date în care se va crea tabelul în câmpul Schema
- Se stabilesc setul de caractere și regulile de corelare
- Se selectează tipul tabelului din câmpul Engine

Table Name: new_table Schema: baza_data_1

Charset/Collation: Default Charset Default Collation Engine: InnoDB

Comments:

Column Name	Datatype	PK	NN	UQ	R	IN	ZF	AT	C	Default/Expression
-------------	----------	----	----	----	---	----	----	----	---	--------------------

- Cele mai importante tipuri de tabele:
 - InnoDB (implicit) - stochează datele folosind sistemul de paginare, unde principala unitate de măsură în baza de date o reprezintă pagina
 - MyISAM stochează datele folosind un simplu principiu al unui arbore binar.

Crearea coloanelor tabelului

- Se specifică denumirea coloane în câmpul Column Name
- Se specifică tipul de date a coloanei în câmpul Data type
- Se bifează opțiunile datelor din coloanele tabelului
- Se specifică o valoare/expresie ce se va considera în cazul neintroducerii datelor cu ajutorul câmpului Default/Expresion

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
idnew_table	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Column Name:	<input type="text" value="idnew_table"/>	Data Type:	<input type="text" value="INT"/>
Charset/Collation:	<input type="text" value="Default Charset"/> <input type="text" value="Default Collation"/>	Expression:	<input type="text"/>
Comments:	<input type="text"/>		
Storage:		<input type="radio"/> Virtual <input type="radio"/> Stored	
<input type="checkbox"/> Primary Key		<input type="checkbox"/> Not Null <input type="checkbox"/> Unique	
<input type="checkbox"/> Binary		<input type="checkbox"/> Unsigned <input type="checkbox"/> Zero Fill	
<input type="checkbox"/> Auto Increment		<input checked="" type="checkbox"/> Generated	

Opțiunile datelor coloanelor

- Coloana de identificare – cheie primară (PK)
- Coloana fără valori nule (NN)
- Coloana cu valori unice (UQ)
- Coloana cu valori în format binar (B)
- Coloana cu valori numerice fără semn (UN)
- Coloana cu umplere a spațiului celulelor cu zerouri (ZF)
- Coloana cu valori obținute prin auto-incrementare (AI)
- Coloana cu valori generate (G)

The screenshot shows the 'Column Options' dialog box in MySQL Workbench. At the top, there is a row of checkboxes for various constraints: PK, NN, UQ, B, UN, ZF, AI, and G. Below this, there is a section for 'Data Type' and 'Expression'. The 'Data Type' is set to 'INT'. The 'Expression' field is empty. Below these fields, there are several options for storage and constraints, each with a checkbox: 'Virtual' (selected), 'Stored', 'Primary Key', 'Not Null', 'Unique', 'Binary', 'Unsigned', 'Zero Fill', 'Auto Increment', and 'Generated' (checked).

PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Data Type:

Expression:

Storage: ☒ Virtual ☐ Stored

☐ Primary Key ☐ Not Null ☐ Unique

☐ Binary ☐ Unsigned ☐ Zero Fill

☐ Auto Increment ☒ Generated

3. Tipuri de date în MySQL

Tipuri de date în MySQL

- **Toate datele introduse într-o coloană a tabelului trebuie să fie de același tip**
- **MySQL acceptă trei categorii generale de date ce conțin mai multe subcategorii:**
 - **Tipuri numerice** – permit reprezentarea numerelor întregi și numerelor cu virgulă mobilă
 - **Tipuri string** – permit reprezentarea valorilor textuale
 - **Tipuri datetime** – permit reprezentarea valorilor de timp
- **În funcție de subcategoria selectată, se rezervă un anumit volum de memorie pentru păstrarea datelor**

Tipuri numerice întregi

Tip	Volum memorie	Valoarea cu semn	Valoarea fără semn
bit	1 bit	0 sau 1	-
tinyint	1 bait	-128 ...127	0...255
smallint	2 baiți	-32768...32767	0...65535
mediumint	3 baiți	-8388608...8388607	0...16777215
int	4 baiți	-2147483648...2147483647	0...4294967295
bigint	8 baiți	-9223372036854775808... 9223372036854775807	0...1844073709551615

- Pentru specificarea valorii fără semn se va bifa opțiunea UN la crearea coloanei
- Este posibilă specificarea numărului de cifre afișate, exemplu `int(2)` va afișa doar 2 cifre
- Pentru umplerea cu valori 0 a spațiilor cifrelor ne prezente se va bifa opțiunea ZF la crearea coloanei, de exemplu pentru tipul `in(5)` cifra 1 va fi afișată ca 00001

Tipuri numerice cu virgulă mobilă

- Tipurile numerice cu virgulă mobilă implică prezența a 2 parametri:
 - **n** – numărul total al cifrelor în număr
 - **p** – numărul cifrelor după virgulă
- MySQL acceptă următoarele tipuri de date cu virgulă mobilă:
 - **float** – volum = 4 baiți, valoarea maximă lui **n** =24, exemplu numărul 120 în float(7,4) va fi reprezentat ca 120,0000
 - **double** – volum = 8 baiți, valoarea maximă lui **n** =53, exemplu numărul 120 în double(10,5) va fi reprezentat ca 120,00000
 - **decimal** – cifrele din partea întreagă și din cel fracțională se păstrează separat în memorie fiind necesari cate 4 baiți pentru 9 cifre. valoarea maximă lui **n** =65, iar a lui **p**=30, exemplu numărul 120 în zecimal(10,6) va fi reprezentat ca 120,000000

Tipuri de date string (1)

- **char** – reprezintă un șir de caracter ce acceptă ca parametru numărul de caracter. Volumul se va determina considerându-se câte un bait pentru fiecare caracter declarat. Exemplu stringul “salut” în char(10) va ocupa $1 \times 10 = 10$ baiți
- **varchar** – reprezintă un șir de caracter ce acceptă ca parametru numărul de caracter. Volumul se va determina considerându-se câte un bait pentru fiecare caracter prezent. Exemplu stringul “salut” din 5 caractere în char(10) va ocupa $1 \times 5 = 5$ baiți
- **binary/varbinary** – identic cu char/varchar doar că datele nu vor fi tratate ca șir de caracterele ci ca șir de valori binare.

Tipuri de date string (2)

- **Blob/text** – identic binary/char doar că nu stochează valoarea în tabelele în care se găsesc, ci în depozite speciale de aceea se pot folosi dacă se manipulează valori cu adevărat mari
- Blob se folosește pentru stocarea datelor binare (de exemplu, fotografii), în timp ce textul se folosește pentru stocarea textului
- Subcategorii ale tipurilor blob și text.
 - tinytext / tinyblob – 255 caractere/baiți;
 - text / blob – 65535 caractere/baiți;
 - mediumtext / mediumblob – 16,777,215 caractere/baiți;
 - longtext / longblob – 4,294,967,295 caractere/baiți.

Tipuri de date datetime

- **Data** - implică toate datele, începând de la 1 ianuarie a anului 1000 până la 31 decembrie a anului 9999 și se poate introduce prin două tipuri diferite: data (de exemplu, prin funcția `curdate()`) sau stringul (2009-10-02)
- **Datetime** – implică datele și timpul pentru acestea și poate accepta valoarea prin tipul echivalent (de exemplu, prin funcția `now()`) sau prin stringul ('YYYY-MM-DD hh:mm:ss').
- **Timestamp** – reprezintă numărul de secunde trecute din 1 ianuarie 1970 și se folosește adesea pentru măsurarea timpului unei operații
- **Time** – acceptă timpul de la -838:59:59 până la 838:59:59, conform schemei HH:MM:SS și poate accepta o valoare „împachetată” în string
- **Year** – reprezintă anul în perioada dintre 1901 și 2155.

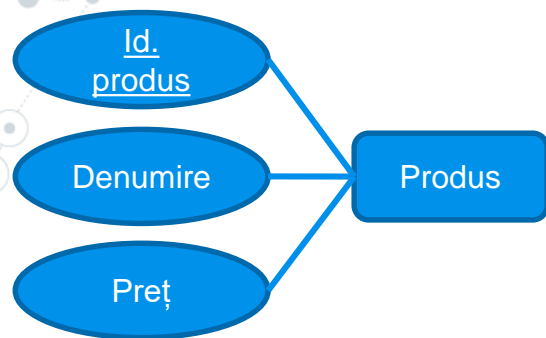
4. Cheia primară




Noțiune de cheie primară

- **Cheia primară** – o coloană a tabelului ce permite identificarea introducerilor în tabel și excluderea introducerilor identice.
- Cheia primară este de valoarea unică pentru fiecare introducere și nu acceptă valori nule
- În practică pentru definirea cheii primare, se mai adaugă un atribut entităților, respectiv o coloană tabelului cu rol de identificare
- De obicei datele coloanei cheii primare sunt de tip întreg și pentru asigurarea valorilor unice se utilizează mecanismul de auto-incrementare

Tabelul entității Produs

- La elaborarea conceptului logic fiecărei entități îi va corespunde un tabel, iar atributele acesteia vor reprezenta coloanele



Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
 id_produs	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
 Denumire	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 Preț	DECIMAL(10,2)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

- Coloana id_produs va fi de tip int și va reprezenta coloana de identificare a produselor (codul produselor) deci se va considera cheia primară bifându-se opțiunile PK (Primary Key), NN (Not Null), UQ (Unique) și AI (Auto Increment)
- Coloana Denumire va fi de tip varchar cu lungime maximă de 45 caractere și nu va accepta ca să lipsească date (toate produsele trebui să aibă denumire)
- Coloana Preț va fi de tip decimal cu lungime maximă de 10 cifre și 2 după virgulă și nu va accepta ca să lipsească date (toate produsele trebui să aibă preț)

Tabelul entității Client

- Coloana `Id_client` va fi cheia primară de tip `int`, coloanele `Nume`, `Prenume` și `Email` va fi de tip `varchar(45)` și coloana `Email` acceptă valori nule (nu toți clienții au email), iar coloana `Data_nașterii` va fi de tip `date`

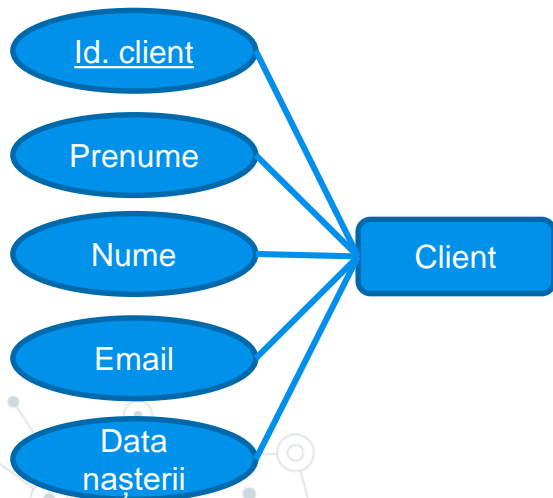


Table Name: `Client` Schema: `baza_date_1`

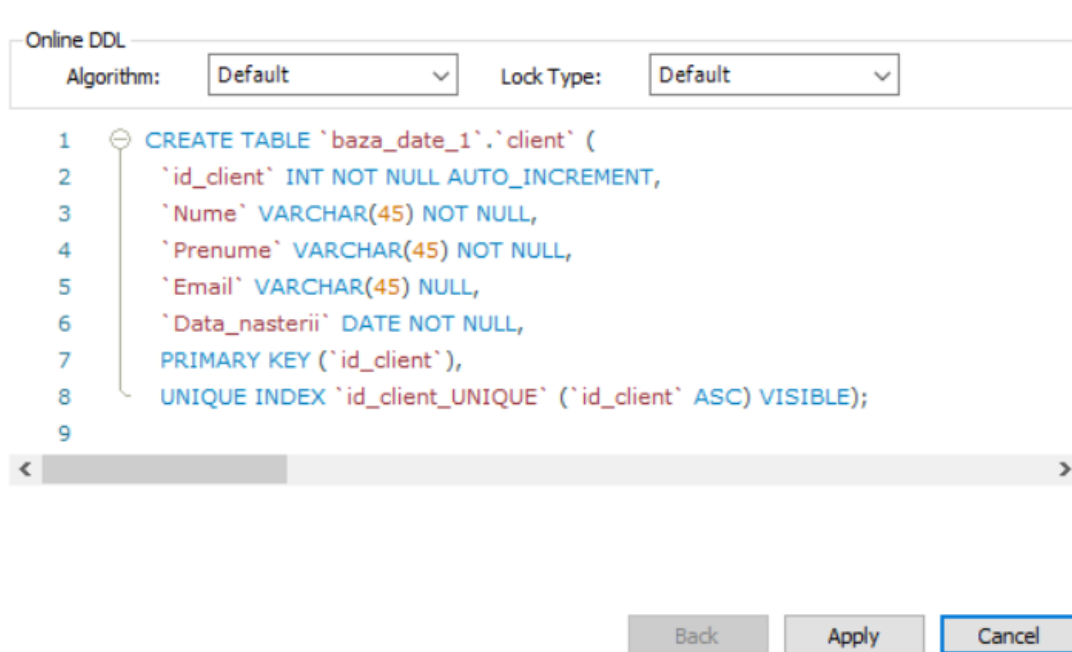
Charset/Collation: `Default Charset` `Default Collation` Engine: `InnoDB`

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
<code>id_client</code>	<code>INT</code>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
<code>Nume</code>	<code>VARCHAR(45)</code>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<code>Prenume</code>	<code>VARCHAR(45)</code>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<code>Email</code>	<code>VARCHAR(45)</code>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<code>Data_nașterii</code>	<code>DATE</code>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

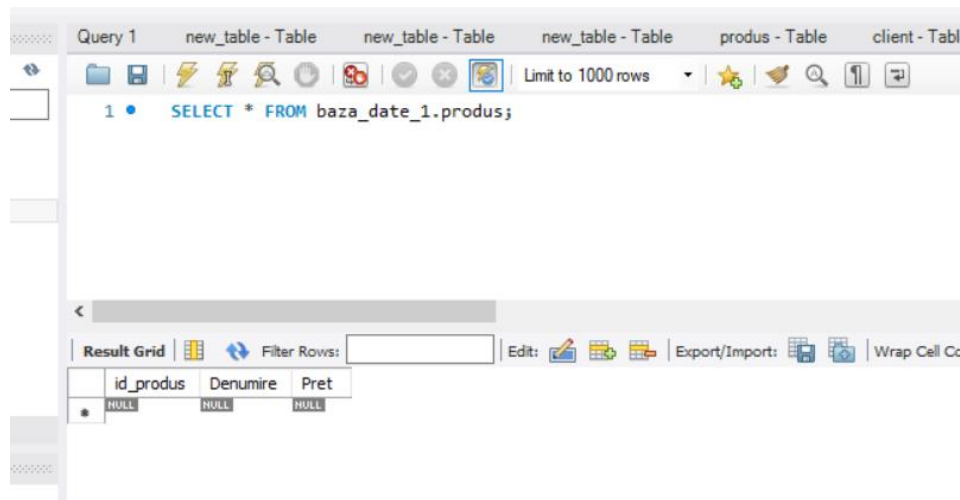
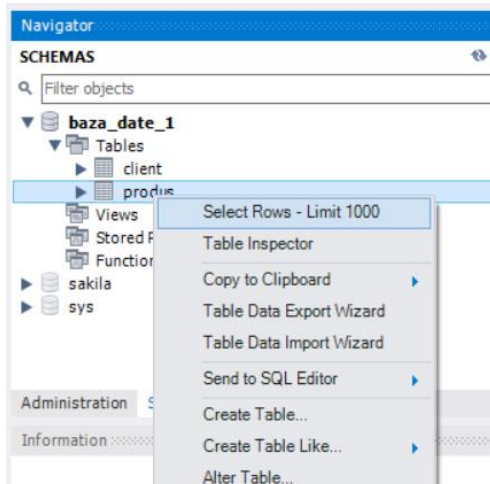
Scriptul SQL al tabelului

- După setarea coloanelor tabelului se aplică butonul Apply și MySQL Workbench va genera în mod automat scriptul SQL corespunzător
- Exemplu script SQL generat pentru tabelul Client



Vizualizarea tabelelor create

- După executarea scriptului SQL pe server în submeniul Schemas a câmpului Navigare pentru baza de date corespunzătoare în componenta Tables numele tabelului creat
- Pentru vizualizarea tabelului se apăsă clic dreapta pe numele tabelului și se selectează Select Rows – Limit 1000



- În câmpul interogărilor va apărea interogarea SQL corespunzătoare și tabelul creat care la moment nu conține date

5. Cheia străină

Noțiune de cheie străină

- Pentru a crea relațiile dintre tabele se utilizează noțiunea de cheie străină
- **Cheia străină** reprezintă coloana cheii primare a unui tabel inclusă în tabelul curent
- Pentru reprezentarea relațiilor many-to-many este necesară existența unui tabel al acestora
- De exemplu pentru stabilirea relațiilor între tabelul produs și client se va crea un tabel numit vânzări ce va include o coloana de identificare a vânzării – cheie primare, două coloane ce vor include numerele de identificare a clientului și respectiv produsului – cheile străine și alte coloane-atribute ale vânzării (numărul de produse și data)

Tabelul relației Vânzări

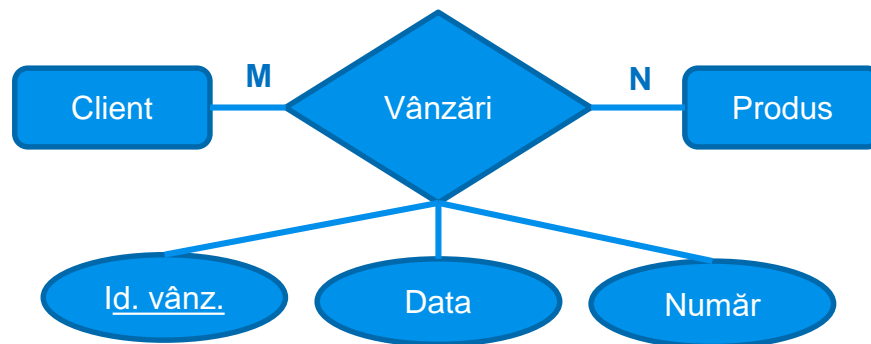


Table Name: Schema: **baza_date_1**

Charset/Collation: Engine:

Comments:

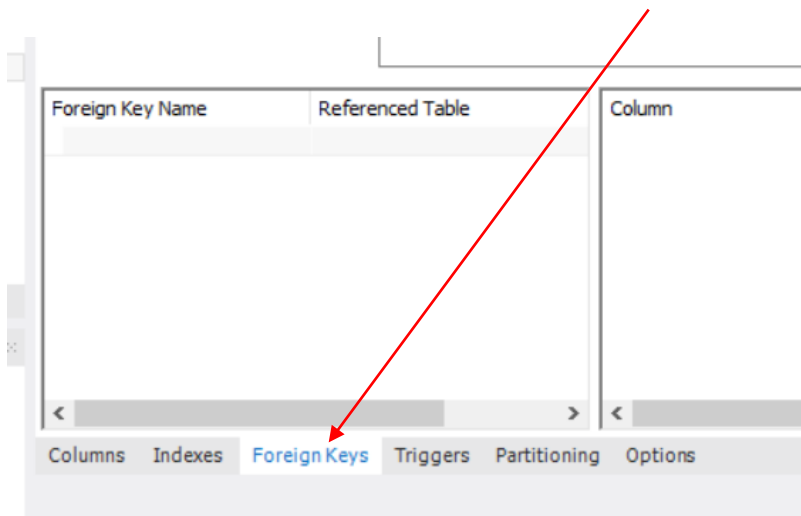
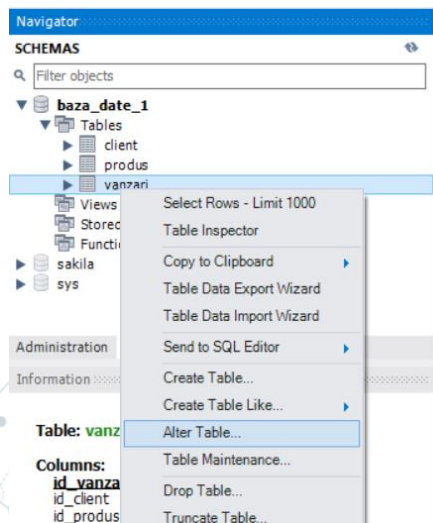
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
id_vanzari	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
id_client	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
id_produs	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
data_vanzarii	DATETIME	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	CURRENT_TIMESTAMP
numar_produce	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Integritatea referențială

- Cheile străine în tabelul relațiilor reprezintă simbolul relațiilor între tabelele entități și valorile coloanelor respective trebuie să corespundă cheilor primare în tabelele entității
- **Integritatea referențială (Foreign Key Constraint)** - existența unor restricții a cheilor străină ce nu ar permite introducerea unor date ce nu există în tabelul cu care se realizează legătura
- Integritatea referențială doar funcționează în tabelele de tip InnoDB
- Exemplu: coloanele `id_client` și `id_produs` au fost specificate ca fiind chei străine în tabelul `vânzări` pentru a stabili relația cu tabelele `client` și `produs`. Ele însă se comportă ca niște coloane obișnuite deoarece acceptă valori care nu sunt incluse în coloanele chei primare a tabelelor de legătură deci este necesară setarea lor ca chei străine,

Modificarea tabelului

- Dacă inițial nu au fost setate cheile străine la crearea tabelului, atunci este necesară modificarea setărilor acestuia
- Pentru modificarea tabelului se va selecta opțiunea **Alter Table...** După ce se aplică clic dreapta pe numele tabelului
- Pentru setarea cheii străine se selectează tab-ul **Foreign Keys**



Setarea cheii străine

1. Se introduce numele cheii străine pentru legătura dată

2. Se selectează tabelul cu care se stabilește legătura

3. Se stabilește cărei coloane a tabelului curent i se va seta cheia străină

4. Se stabilește cu care coloană a tabelului de legătură se va seta relația

Query 1 | vanzari - Table x

Table Name: Schema: **baza_date_1**

Charset/Collation: Engine:

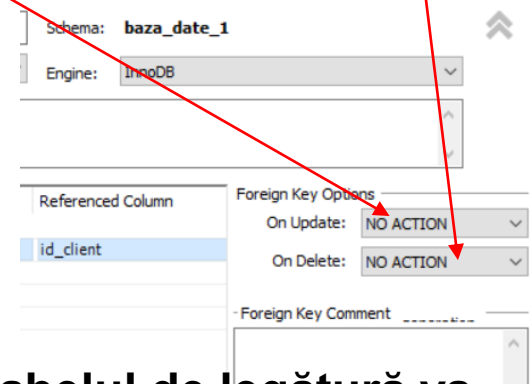
Comments:

Foreign Key Name	Referenced Table
fk_id_client	'baza_date_1'. 'client'

Column	Referenced Column
<input type="checkbox"/> id_vanzari	
<input checked="" type="checkbox"/> id_client	id_client
<input type="checkbox"/> id_produc	
<input type="checkbox"/> data_vanzarii	
<input type="checkbox"/> numar_produce	

Setări adiționale cheii străine

- La setarea cheilor străine se setează și comportamentul acestora în tabelul curent în cazul modificării (On update) sau ștergerii (On Delete) datelor corespunzătoare din tabelul de legătură
- Opțiuni de comportamente:
 - **NO ACTION** – nu se va permite modificarea sau ștergerea datelor în tabelele de legătură
 - **RESTRICT**– în MySQL are aceeași semnificație ca și **NO ACTION**
 - **CASCADE** – modificarea/ștergerea datelor în tabelul de legătură va modifica/ștergere și valorile cheii străine. Se recomandă doar la modificarea datelor și poate avea efecte ne dorite la ștergere
 - **SET NULL** – modificarea/ștergerea datelor în tabelul de legătură va seta valoarea nulă a cheii străine. Se recomandă doar la ștergere



Crearea cheii străine pentru tabelul produs

- Cheia straină `fk_id_produs` va stabili relația dintre colona `id_produs` din tabelul `vânzări` și coloana `id_produs` din tabelul `produs`
- Modificarea datelor în tabelul `produs` va produce modificarea în cascadă și a datelor corespunzătoare în tabelul `vânzări`, iar ștergerea datelor va fi imposibilă

