What is a Database?

It is a collection of data in a format that can be easily accessed.

Why databases?

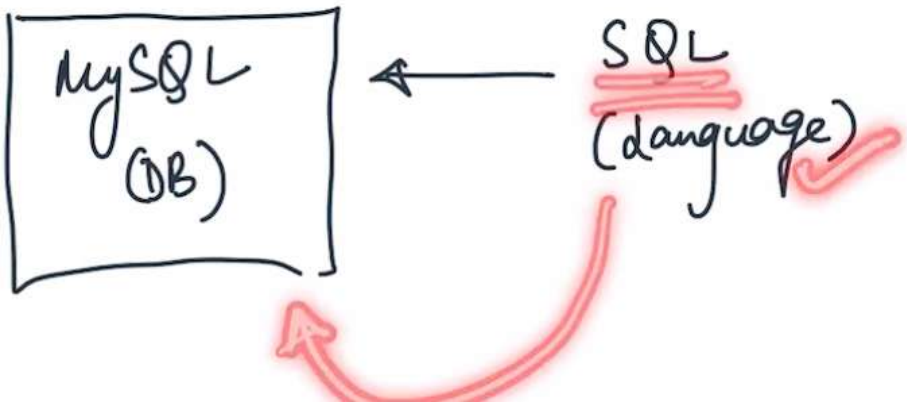
- can store large data
- features like security, scalability etc.
- Easier to insert, update or delete data

Our 1st Database : SQL

MySQL
(DB)

SQL
(language)

Our 1st Database : SQL



What is a Database?

It is a collection of data in a format that can be easily accessed.

Why databases?

- can store large data
- features like security, scalability etc.
- Easier to insert, update or delete data

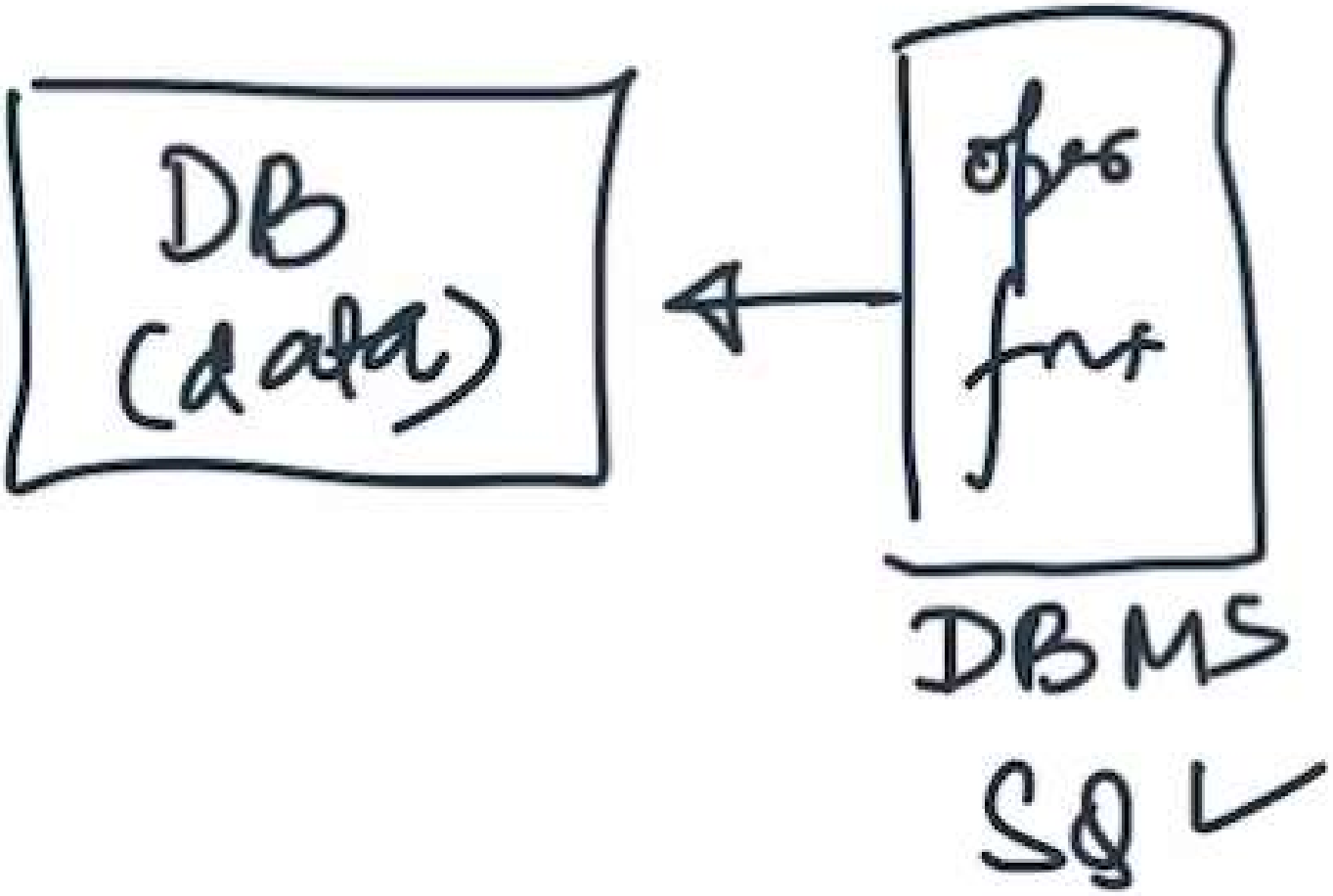
What is a Database?

It is a collection of data in a format that can be easily accessed.

Why databases?

- can store large data
- features like security, scalability etc.
- Easier to insert, update or delete data

DBMS



SQL

SQl v/s NoSQL

SQL

Relational Database
(data stored in Tables)

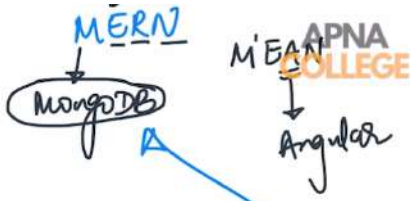
eg - MySQL, Oracle,
PostgreSQL etc.

NoSQL

Non Relational Database
(data stored in document/key-val/graphs etc.)

eg - MongoDb, Cassandra,
Neo4j etc.

SQl v/s NoSQL



SQL

Relational Database
(data stored in Tables)
Relation (table)

eg - MySQL, Oracle,
PostgreSQL etc.

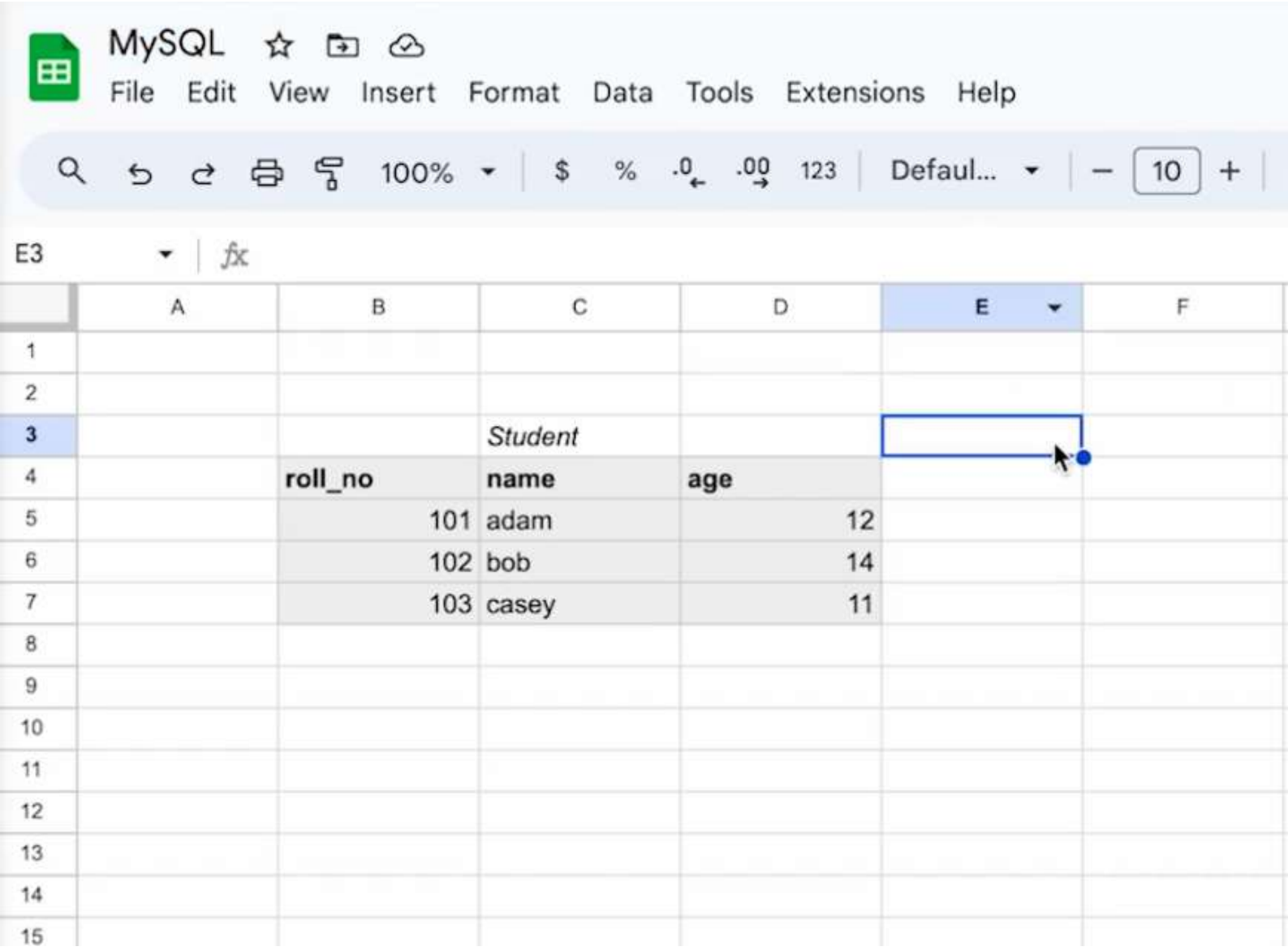
NoSQL

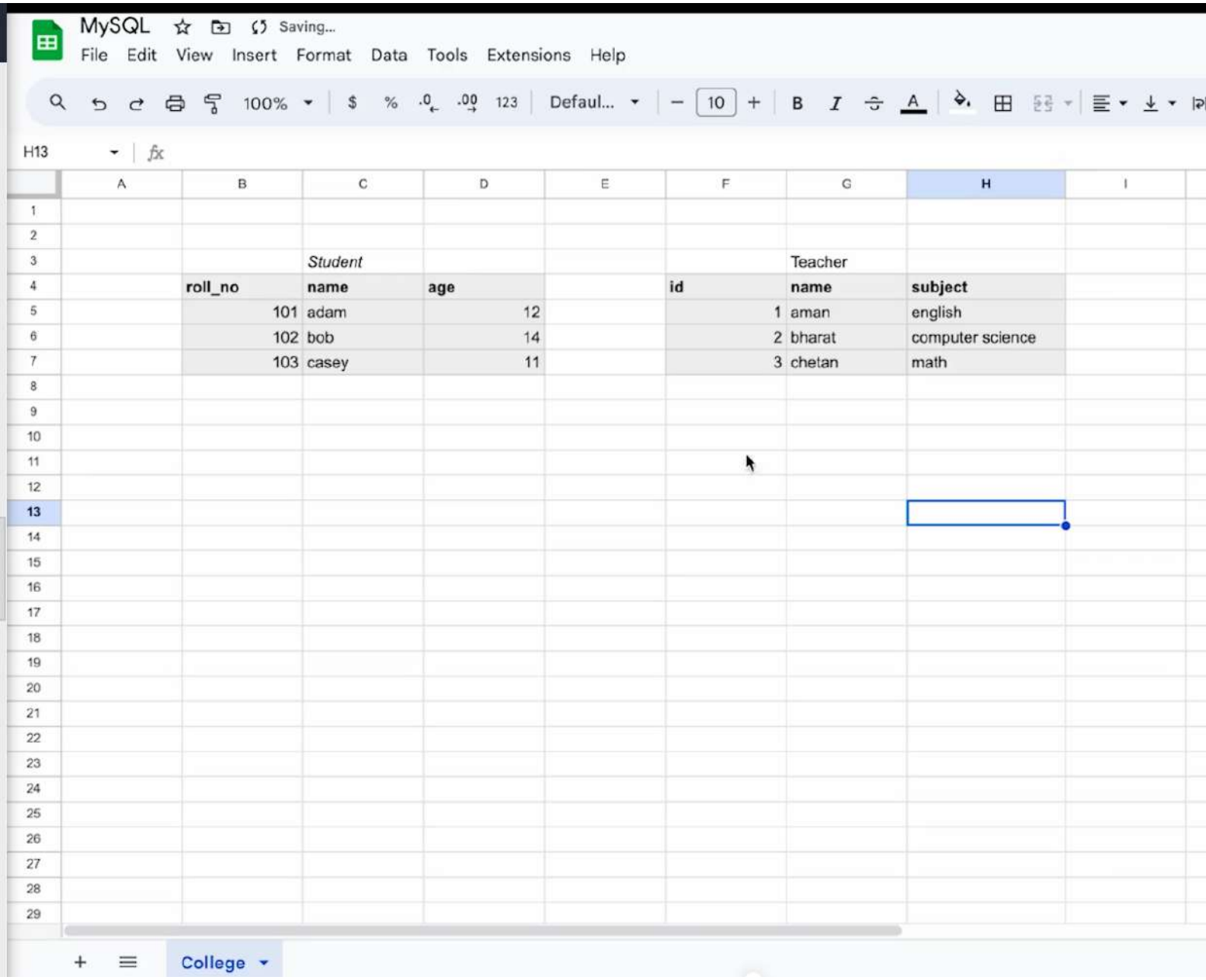
Non Relational Database
(data stored in document/key-val/graphs etc.)

eg - MongoDb, Cassandra,
Neo4j etc.

SQL (Structured Query Language)

SQL is a programming language used to interact with relational databases.


MySQL
File Edit View Insert Format Data Tools Extensions Help
100% $ % .0 .00 123 Defaul... 10
E3
A B C D E F
1
2
3 Student
4 roll_no name age
5 101 adam 12
6 102 bob 14
7 103 casey 11
8
9
10
11
12
13
14
15


MySQL
Saving...
File Edit View Insert Format Data Tools Extensions Help
100% $ % 123 Defaul... 10
H13
Student
Teacher
roll_no name age id name subject
101 adam 12 1 aman english
102 bob 14 2 bharat computer science
103 casey 11 3 chetan math
College

SQL (Structured Query Language)

SQL is a programming language used to interact with relational databases.

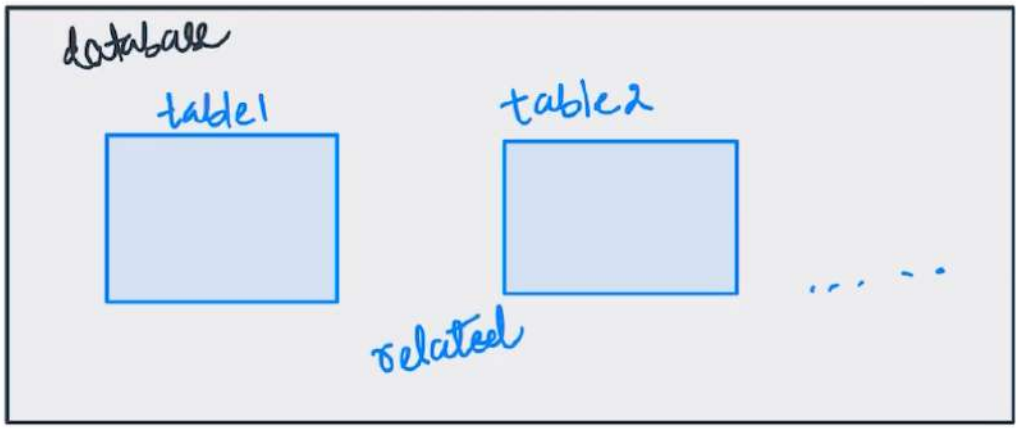


Table in SQL

	User			
Id	**Name**	**Email**	**Followers**	**Following**
1	Adam	adam@yahoo.in	123	145
2	Bob	bob123@gmail.com	200	200
3	Casey	casey@email.com	300	306
4	Donald	donald@gmail.com	200	105

Table in SQL

columns → design (schema)
[id, name, email, followers, following]
rows → tuple

	User			
Id	**Name**	**Email**	**Followers**	**Following**
1	Adam	adam@yahoo.in	123	145
2	Bob	bob123@gmail.com	200	200
3	Casey	casey@email.com	300	306
4	Donald	donald@gmail.com	200	105

col1 (id)
col2 (name)
col3 (email)

MySQL

1) MySQL Server

↓

DB
(tables)

2) MySQL Workbench

	Following

https://dev.mysql.com/downloads/file/?id=520406

MySQL Installer 8.0.34

Note: MySQL 8.0 is the final series with MySQL Installer. As of MySQL 8.1, use a MySQL product's MSI or Zip archive for installation. MySQL Server 8.1 and higher also bundle MySQL Configurator, a tool that helps configure MySQL Server.

Select Version:

8.0.34

Select Operating System:

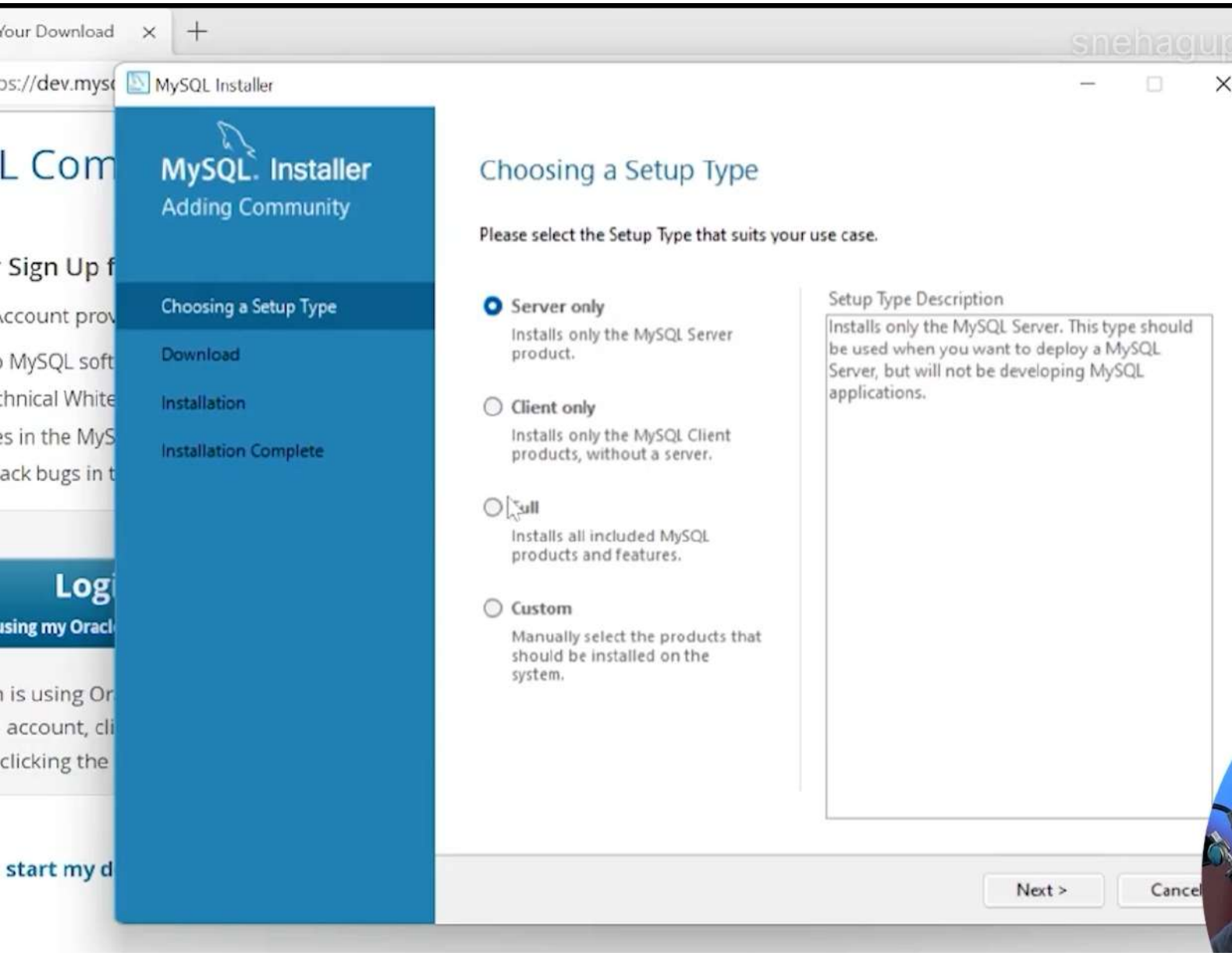
Microsoft Windows

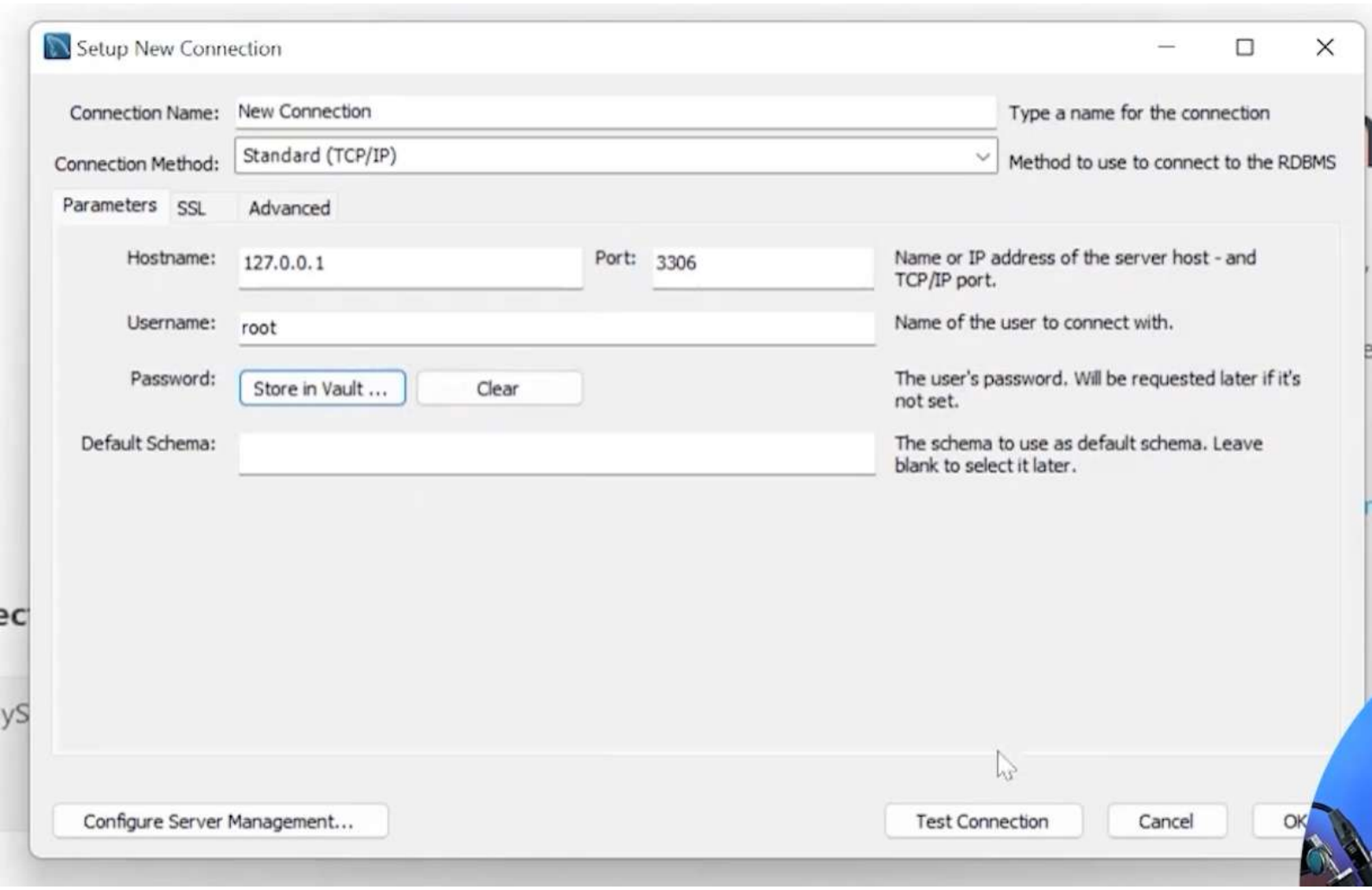
Windows (x86, 32-bit), MSI Installer	8.0.34	2.4M	**Download**
(mysql-installer-web-community-8.0.34.0.msi)	MD5: 01baf7b42e551d53efb557eed401ff91 | Signature		
Windows (x86, 32-bit), MSI Installer	8.0.34	331.3M	**Download**
(mysql-installer-community-8.0.34.0.msi)	MD5: 59eaa511c39011a2f0264311a80b0228 | Signature		

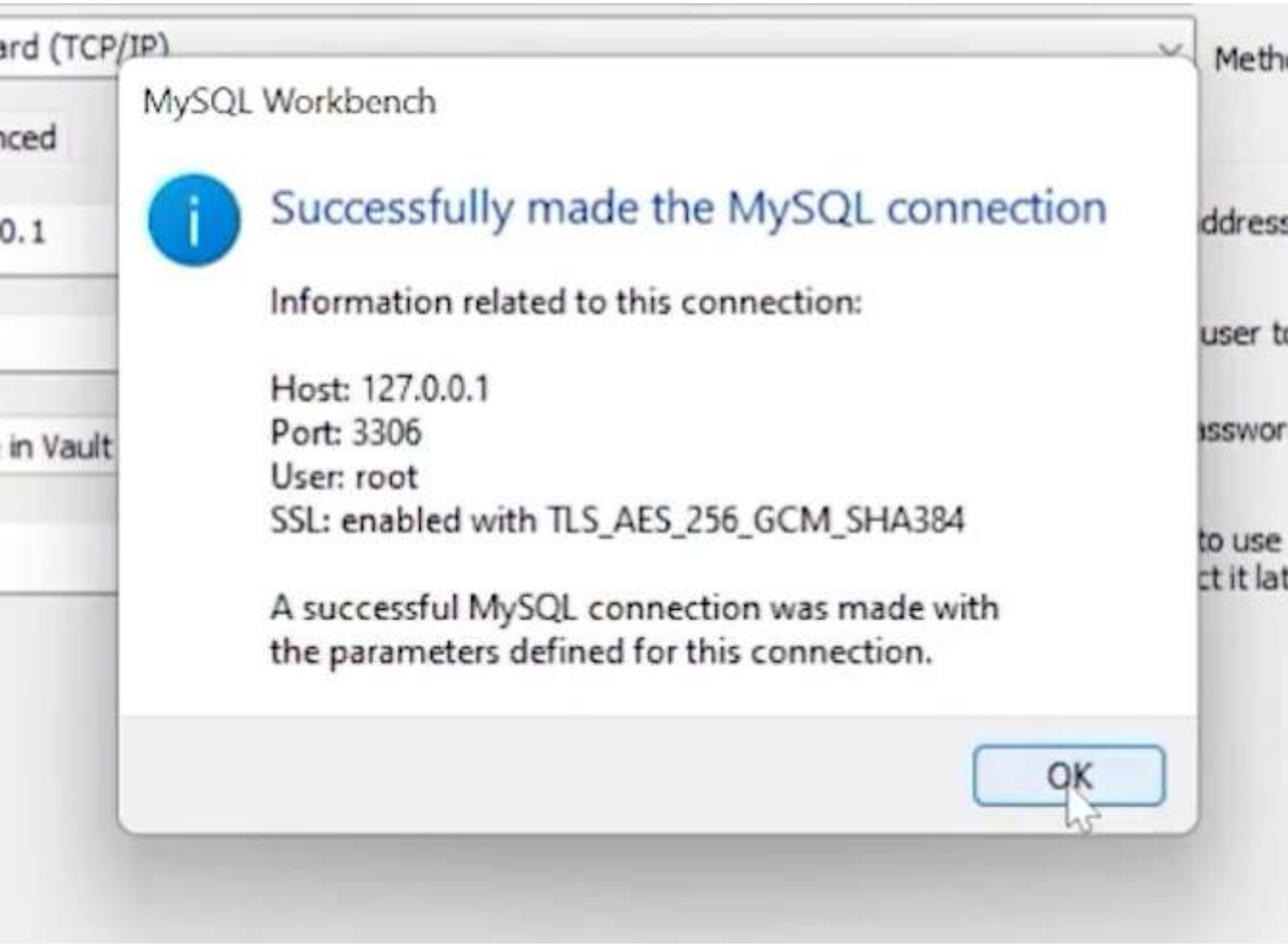
We suggest that you use the MD5 checksums and GnuPG signatures to verify the integrity of the packages you download.

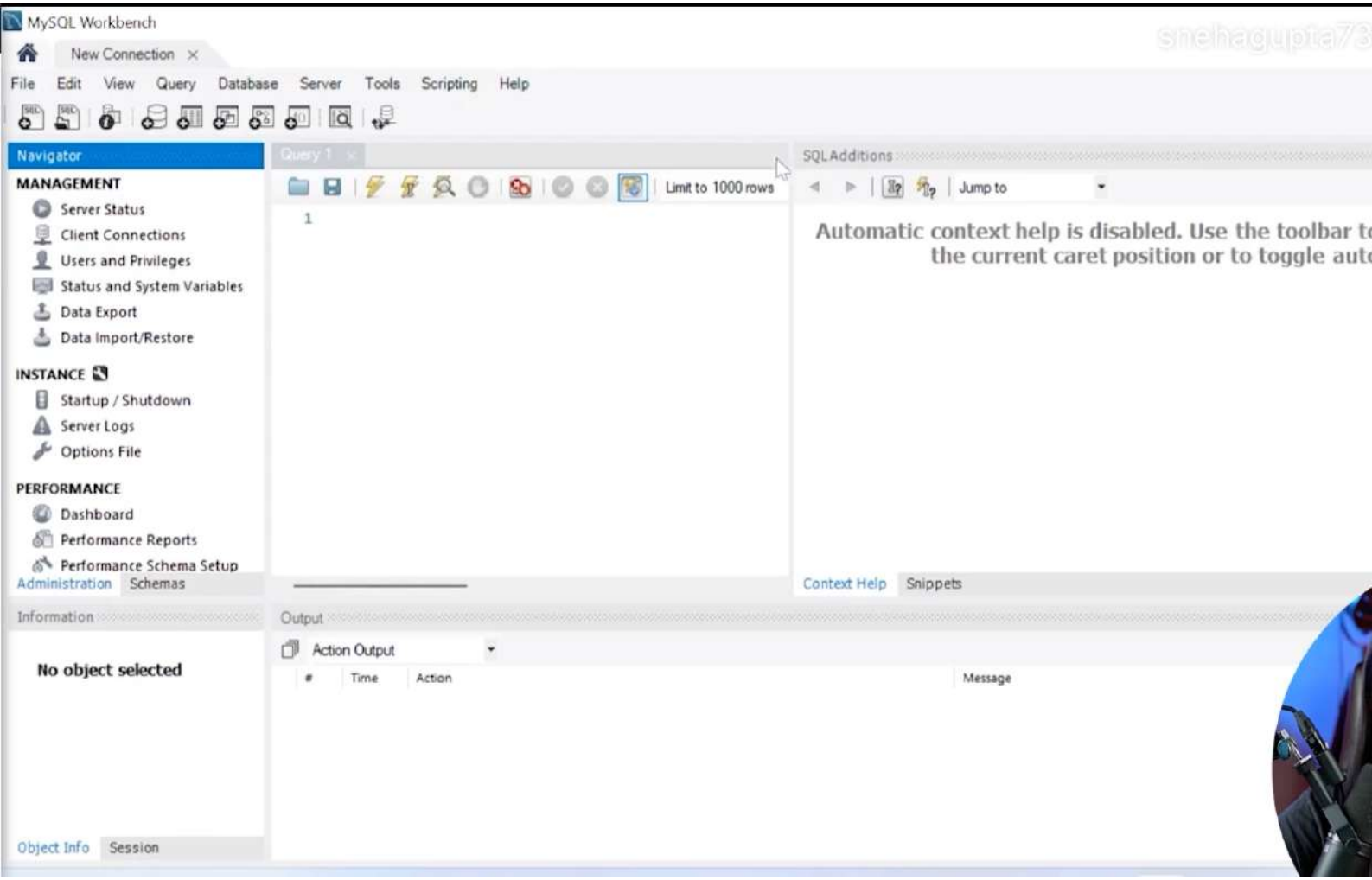
No thanks, just start my download.

for oracle.112.2o7.net...


Your Download
MySQL Installer
MySQL. Installer
Adding Community
Choosing a Setup Type
Download
Installation
Installation Complete
Choosing a Setup Type
Please select the Setup Type that suits your use case.
Server only
Installs only the MySQL Server product.
Client only
Installs only the MySQL Client products, without a server.
Full
Installs all included MySQL products and features.
Custom
Manually select the products that should be installed on the system.
Setup Type Description
Installs only the MySQL Server. This type should be used when you want to deploy a MySQL Server, but will not be developing MySQL applications.
Next >
Cancel


Setup New Connection
Connection Name:
New Connection
Type a name for the connection
Connection Method:
Standard (TCP/IP)
Method to use to connect to the RDBMS
Parameters
SSL
Advanced
Hostname:
127.0.0.1
Port:
3306
Name or IP address of the server host - and TCP/IP port.
Username:
root
Name of the user to connect with.
Password:
Store in Vault ...
Clear
The user's password. Will be requested later if it's not set.
Default Schema:
The schema to use as default schema. Leave blank to select it later.
Configure Server Management...
Test Connection
Cancel
OK


MySQL Workbench
Successfully made the MySQL connection
Information related to this connection:
Host: 127.0.0.1
Port: 3306
User: root
SSL: enabled with TLS_AES_256_GCM_SHA384
A successful MySQL connection was made with
the parameters defined for this connection.
OK


MySQL Workbench
New Connection
File Edit View Query Database Server Tools Scripting Help
Navigator
MANAGEMENT
Server Status
Client Connections
Users and Privileges
Status and System Variables
Data Export
Data Import/Restore
INSTANCE
Startup / Shutdown
Server Logs
Options File
PERFORMANCE
Dashboard
Performance Reports
Performance Schema Setup
Administration Schemas
Query 1
Limit to 1000 rows
1
SQL Additions
Jump to
Automatic context help is disabled. Use the toolbar to
the current caret position or to toggle auto
Context Help Snippets
Information
No object selected
Object Info Session
Output
Action Output
Time Action Message

General Availability (GA) Releases | Archives

MySQL Community Server 8.1.0 Innovation

Select Version:

8.1.0 Innovation

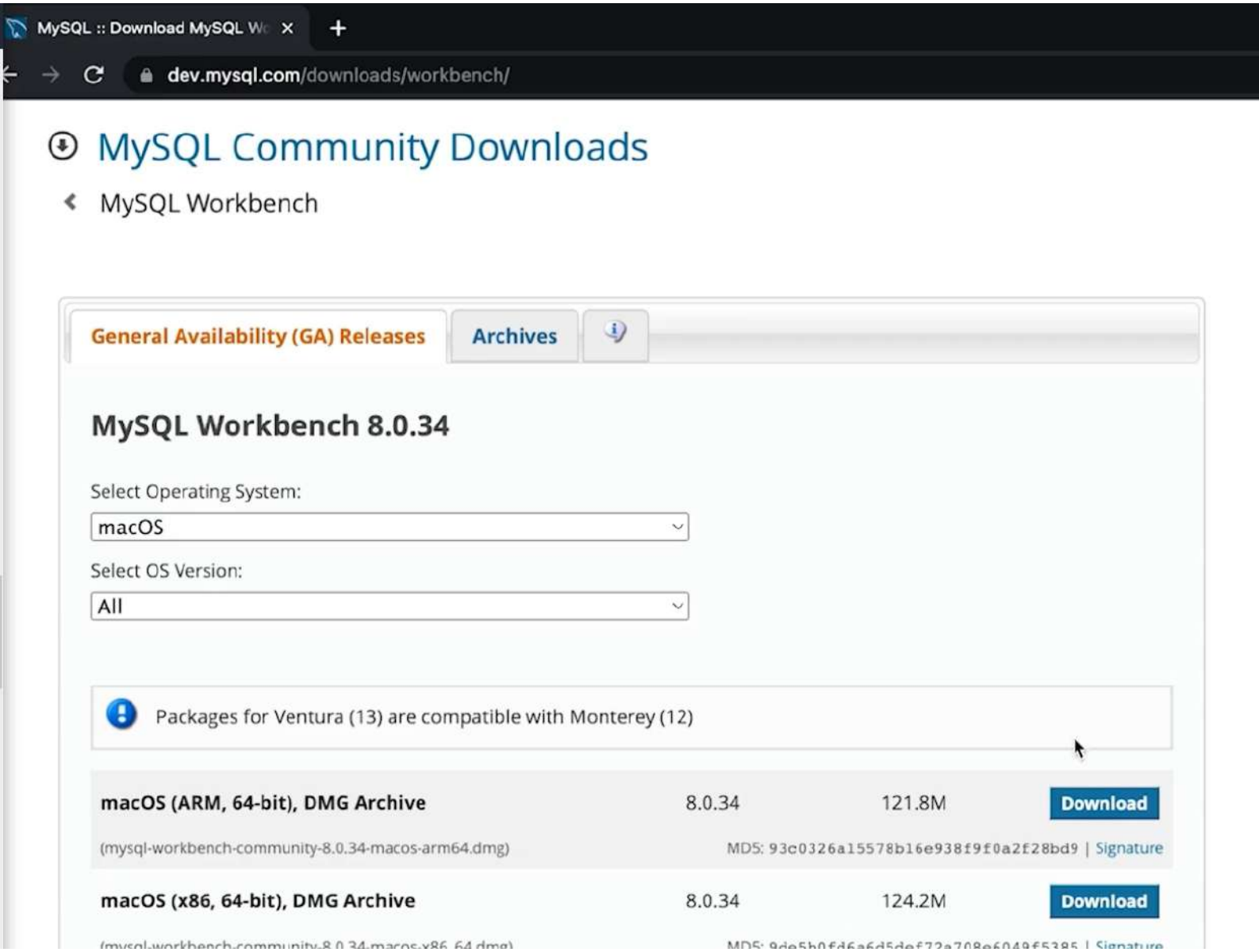
Select Operating System:

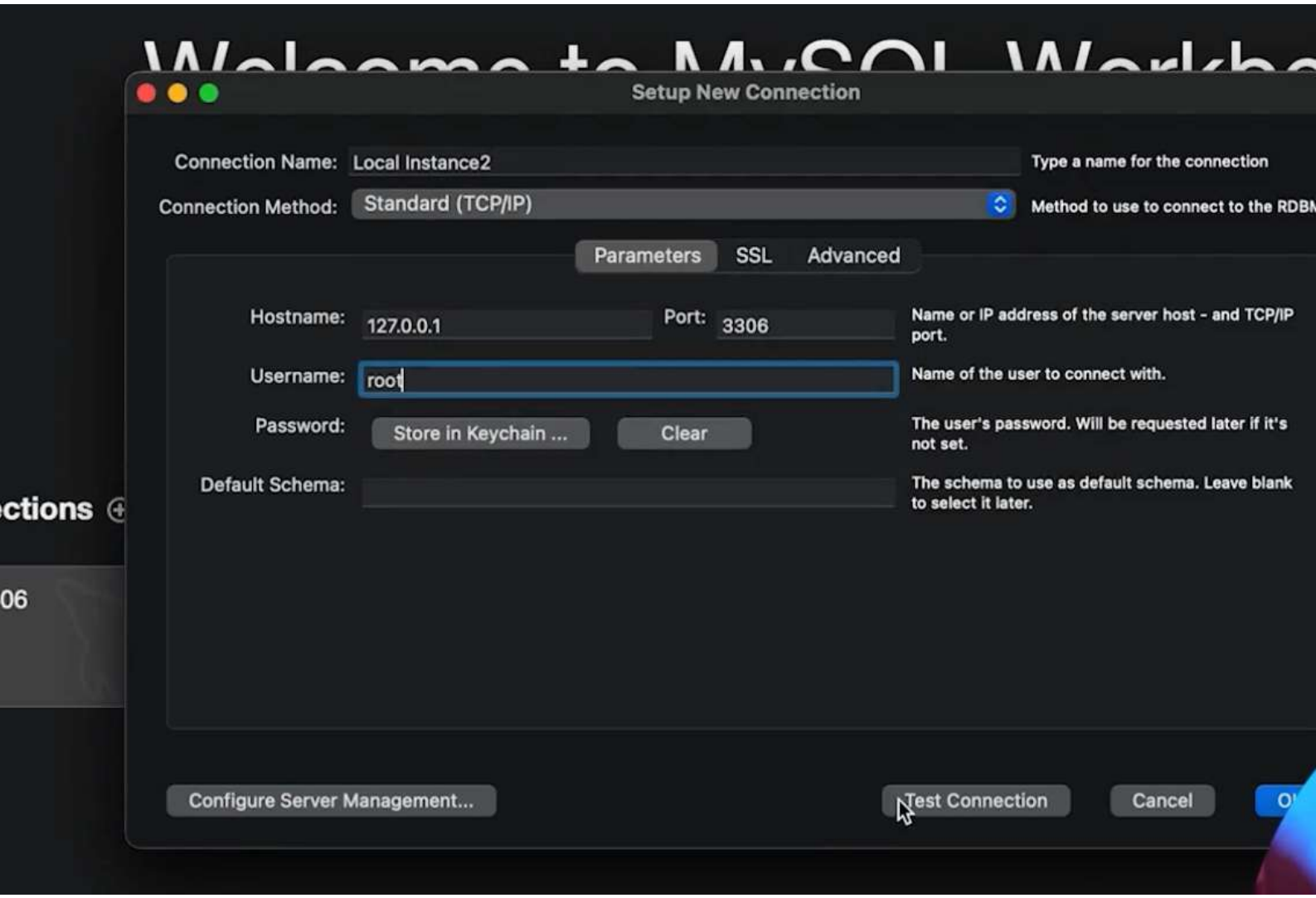
macOS

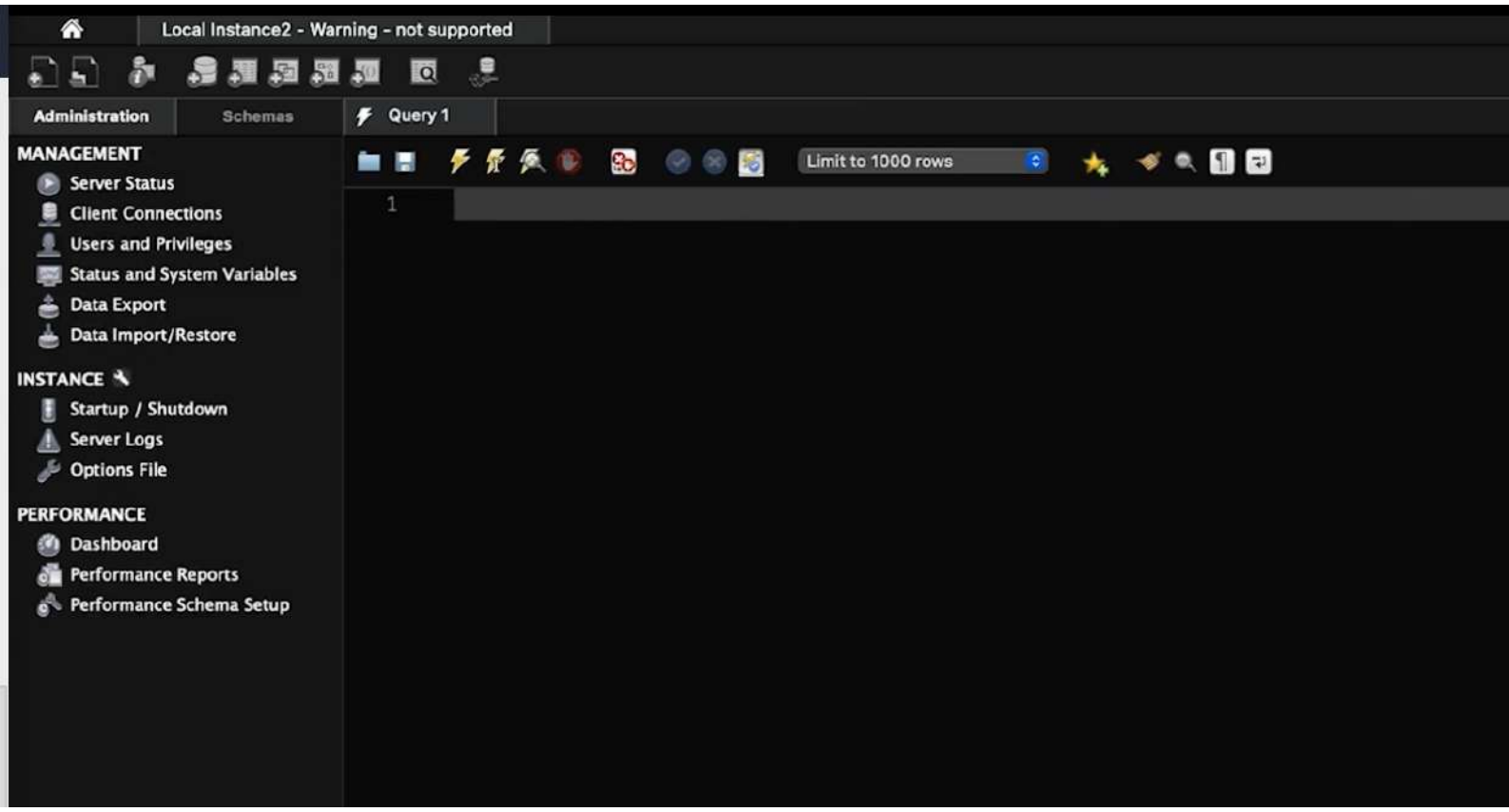
Select OS Version:

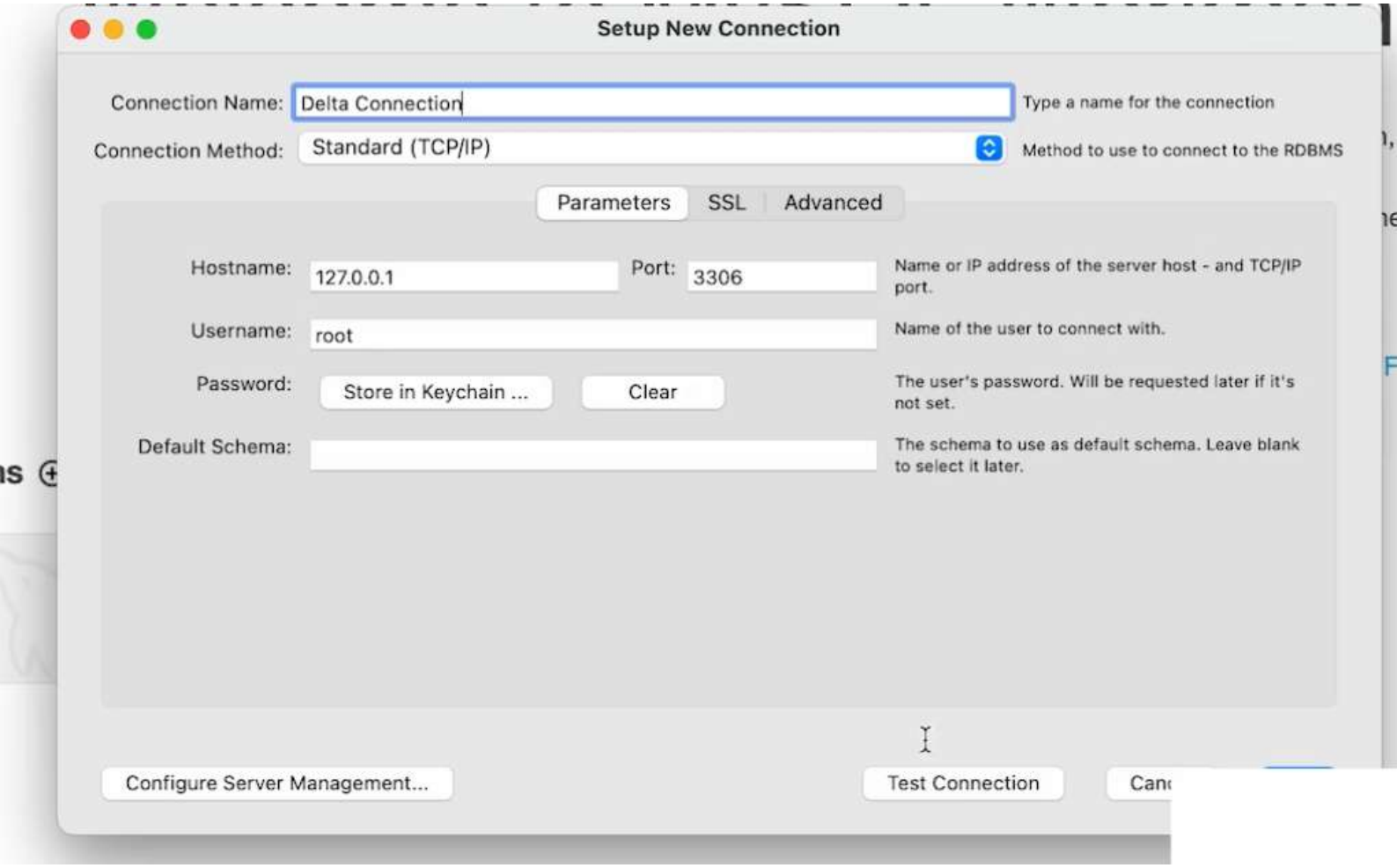
All

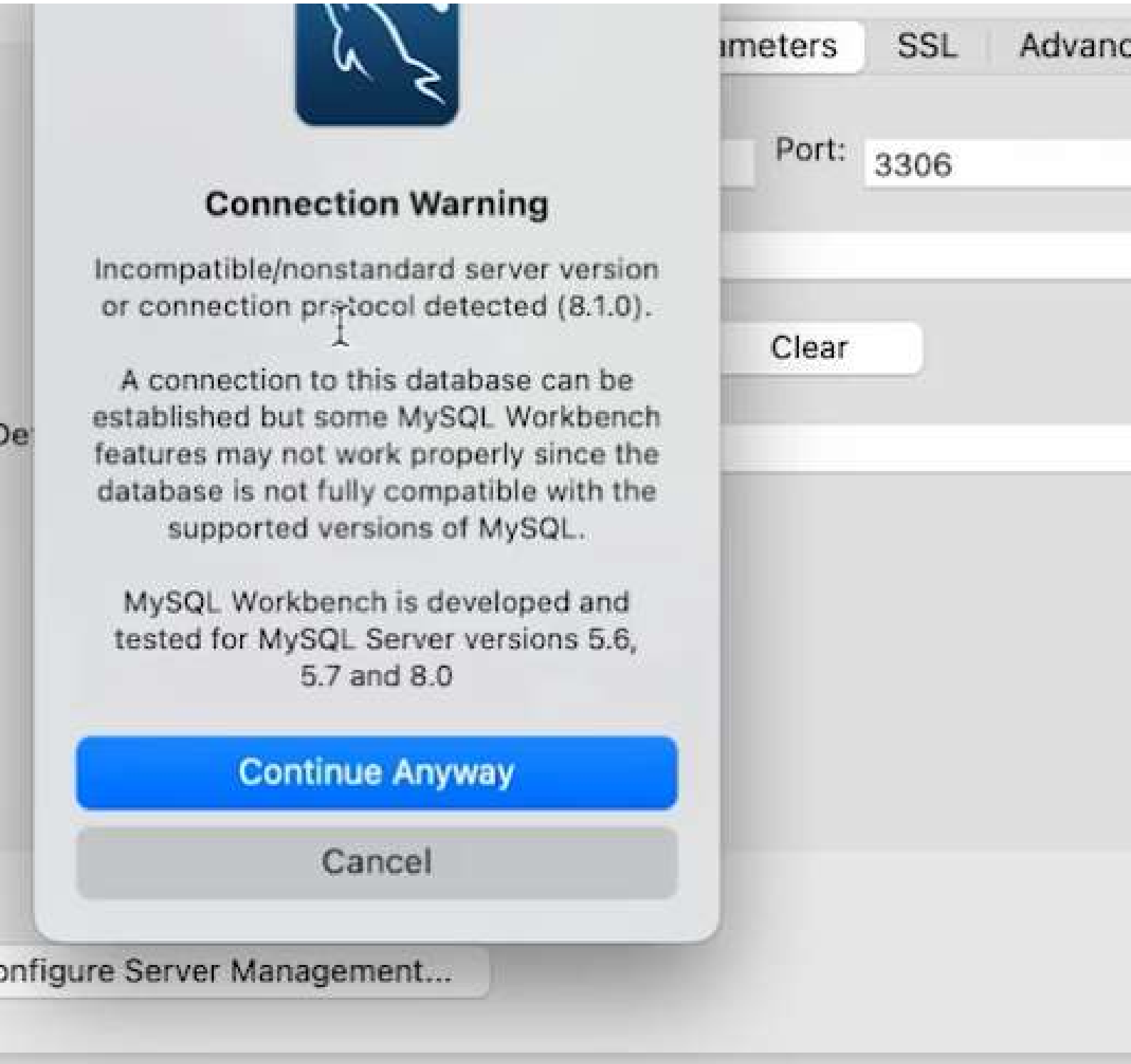
macOS 13 (ARM, 64-bit), DMG Archive	8.1.0	560.7M	Download
(mysql-8.1.0-macos13-arm64.dmg)	MD5: 0f0fc93dcdd165b714671e7fffe35594 | Signature		
macOS 13 (x86, 64-bit), DMG Archive	8.1.0	565.7M	Download
(mysql-8.1.0-macos13-x86_64.dmg)	MD5: d18aef83504da330fab3b880736b60a4 | Signature		
macOS 13 (ARM, 64-bit), Compressed TAR Archive	8.1.0	176.3M	Download
(mysql-8.1.0-macos13-arm64.tar.gz)	MD5: 420fe4563ff32ce50b841c8aab013e07 | Signature		
macOS 13 (x86, 64-bit), Compressed TAR Archive	8.1.0	180.7M	Download
(mysql-8.1.0-macos13-x86_64.tar.gz)	MD5: 71e17307da6cf992ae75931efe3e7069 | Signature		

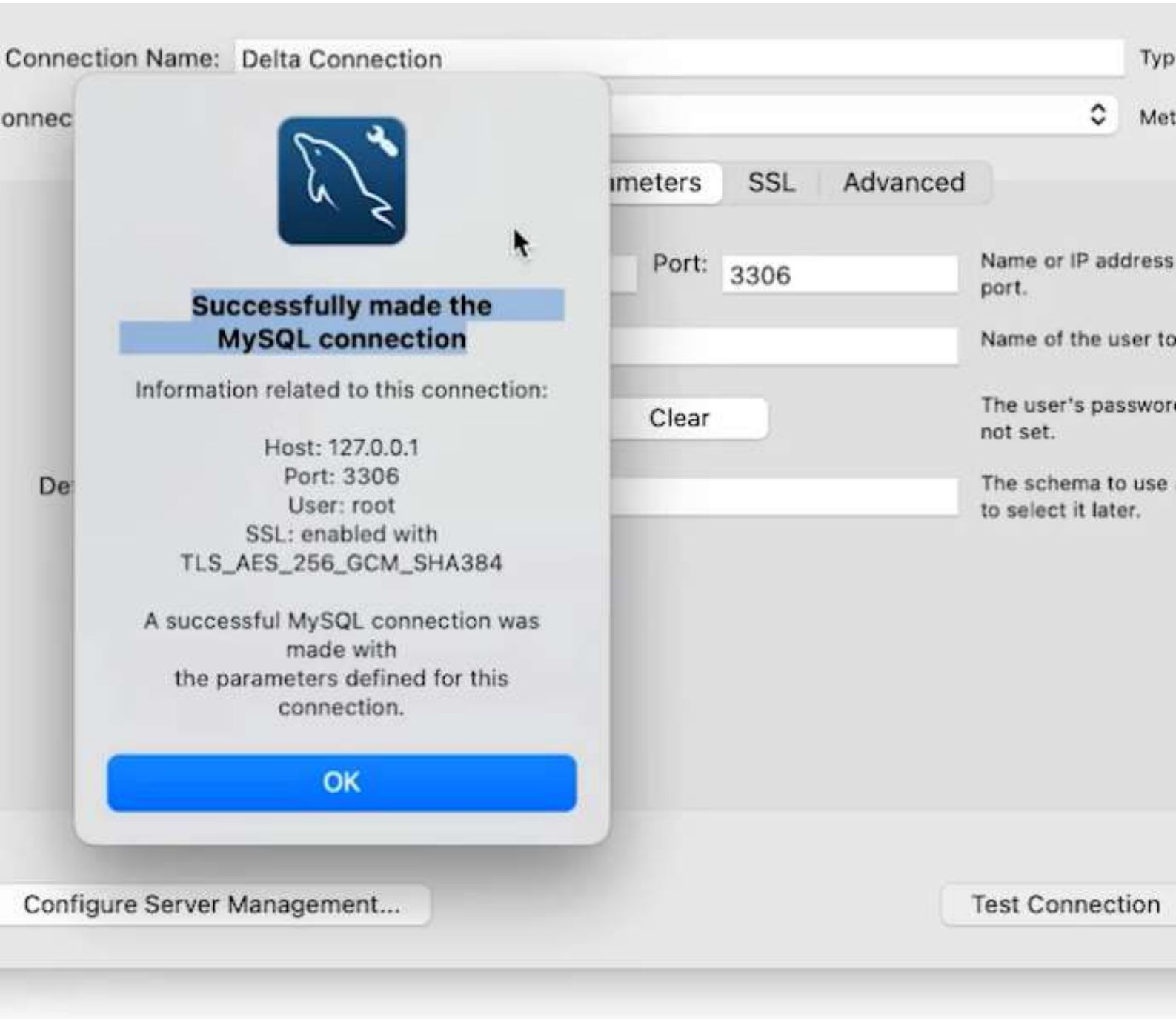

MySQL :: Download MySQL Wc
dev.mysql.com/downloads/workbench/
MySQL Community Downloads
MySQL Workbench
General Availability (GA) Releases
Archives
MySQL Workbench 8.0.34
Select Operating System:
macOS
Select OS Version:
All
Packages for Ventura (13) are compatible with Monterey (12)
macOS (ARM, 64-bit), DMG Archive
8.0.34
121.8M
Download
(mysql-workbench-community-8.0.34-macos-arm64.dmg)
MD5: 93c0326a15578b16e938f9f0a2f28bd9 | Signature
macOS (x86, 64-bit), DMG Archive
8.0.34
124.2M
Download


Setup New Connection
Connection Name:
Local Instance2
Type a name for the connection
Connection Method:
Standard (TCP/IP)
Method to use to connect to the RDBM
Parameters
SSL
Advanced
Hostname:
127.0.0.1
Port:
3306
Name or IP address of the server host - and TCP/IP port.
Username:
root
Name of the user to connect with.
Password:
Store in Keychain ...
Clear
The user's password. Will be requested later if it's not set.
Default Schema:
The schema to use as default schema. Leave blank to select it later.
Configure Server Management...
Test Connection
Cancel


Local Instance2 - Warning - not supported
Administration
Schemas
Query 1
MANAGEMENT
Server Status
Client Connections
Users and Privileges
Status and System Variables
Data Export
Data Import/Restore
INSTANCE
Startup / Shutdown
Server Logs
Options File
PERFORMANCE
Dashboard
Performance Reports
Performance Schema Setup
Limit to 1000 rows
1


Setup New Connection
Connection Name:
Delta Connection
Type a name for the connection
Connection Method:
Standard (TCP/IP)
Method to use to connect to the RDBMS
Parameters
SSL
Advanced
Hostname:
127.0.0.1
Port:
3306
Name or IP address of the server host - and TCP/IP port.
Username:
root
Name of the user to connect with.
Password:
Store in Keychain ...
Clear
The user's password. Will be requested later if it's not set.
Default Schema:
The schema to use as default schema. Leave blank to select it later.
Configure Server Management...
Test Connection


imeters
SSL
Advanc
Port:
3306
Clear
Connection Warning
Incompatible/nonstandard server version or connection protocol detected (8.1.0).
A connection to this database can be established but some MySQL Workbench features may not work properly since the database is not fully compatible with the supported versions of MySQL.
MySQL Workbench is developed and tested for MySQL Server versions 5.6, 5.7 and 8.0
Continue Anyway
Cancel
onfigure Server Management...


Connection Name: Delta Connection
Typ
Met
meters
SSL
Advanced
Port: 3306
Name or IP address
port.
Name of the user to
Clear
The user's passwor
not set.
The schema to use
to select it later.
Successfully made the MySQL connection
Information related to this connection:
Host: 127.0.0.1
Port: 3306
User: root
SSL: enabled with
TLS_AES_256_GCM_SHA384
A successful MySQL connection was made with
the parameters defined for this connection.
OK
Configure Server Management...
Test Connection

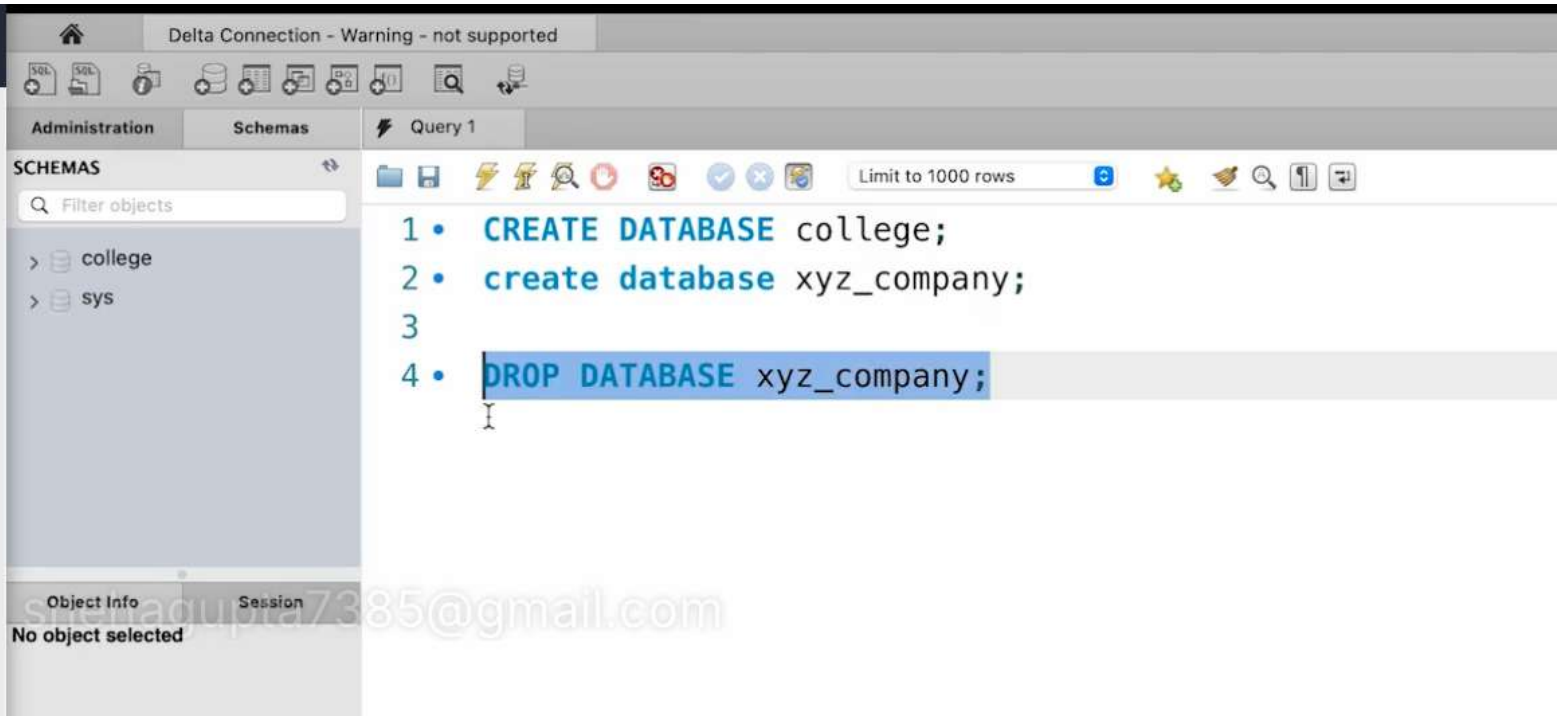
Our 1st Database

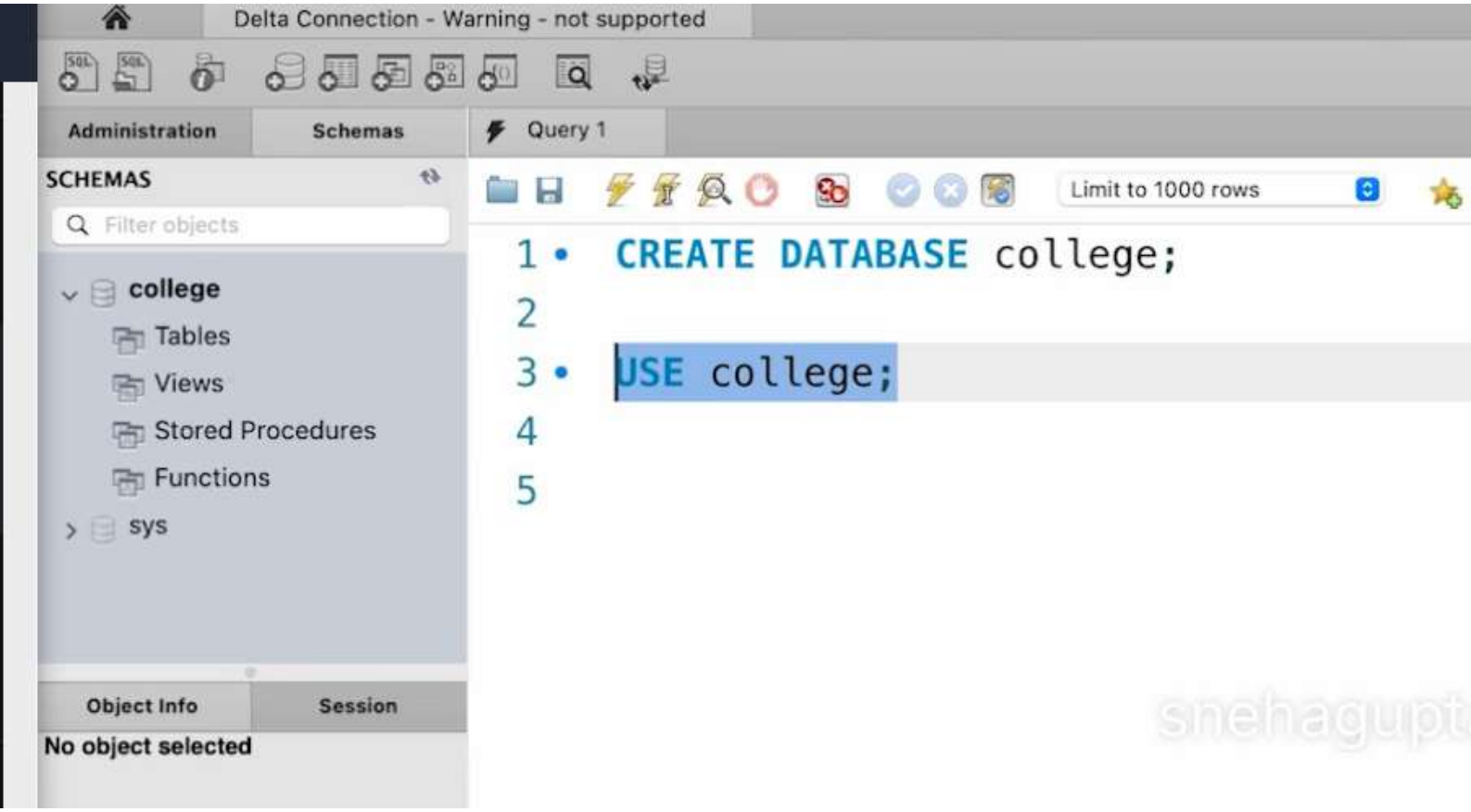
```
CREATE DATABASE db_name;
```

```
DROP DATABASE db_name;
```



```
USE db_name;
```

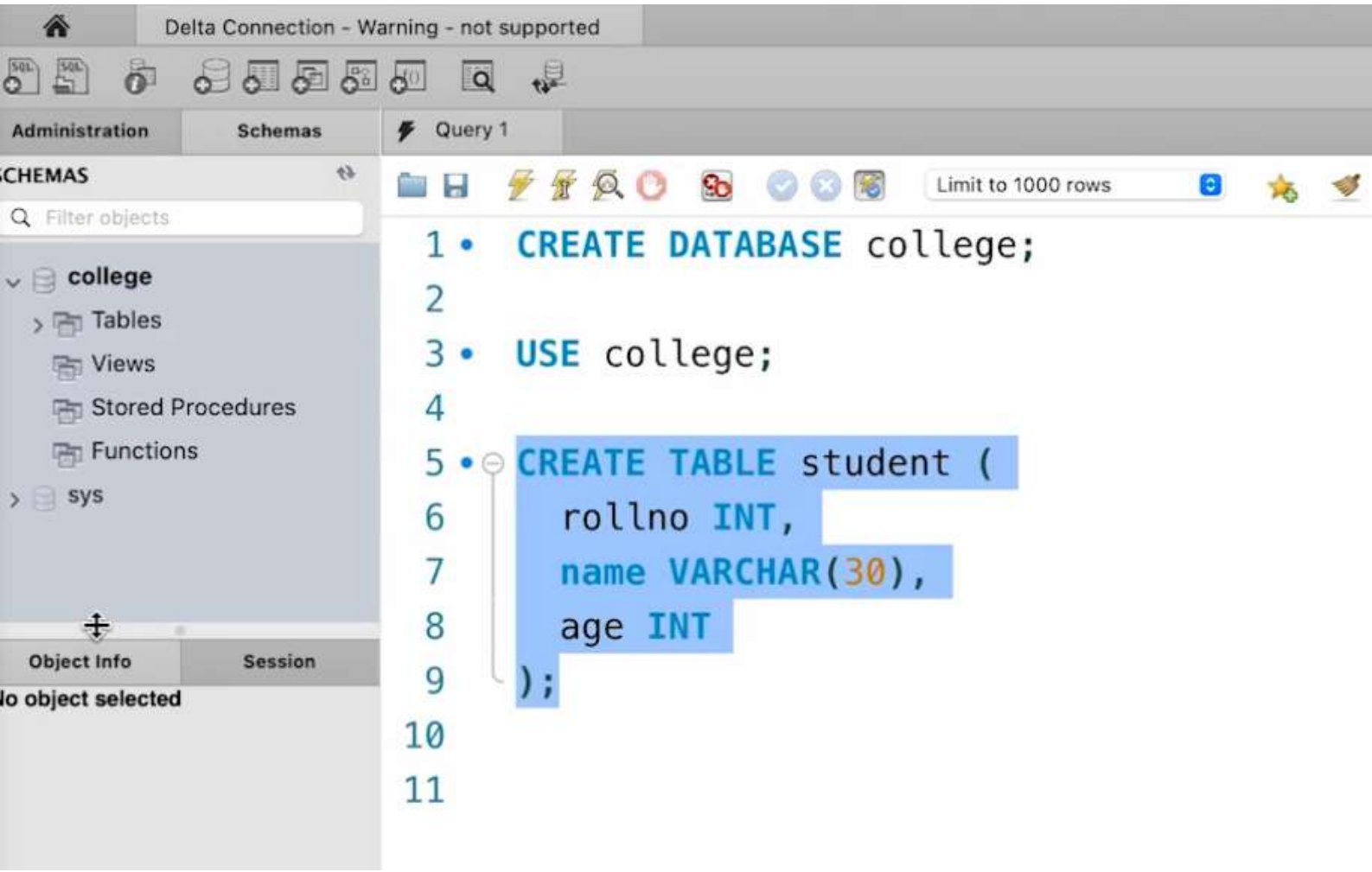


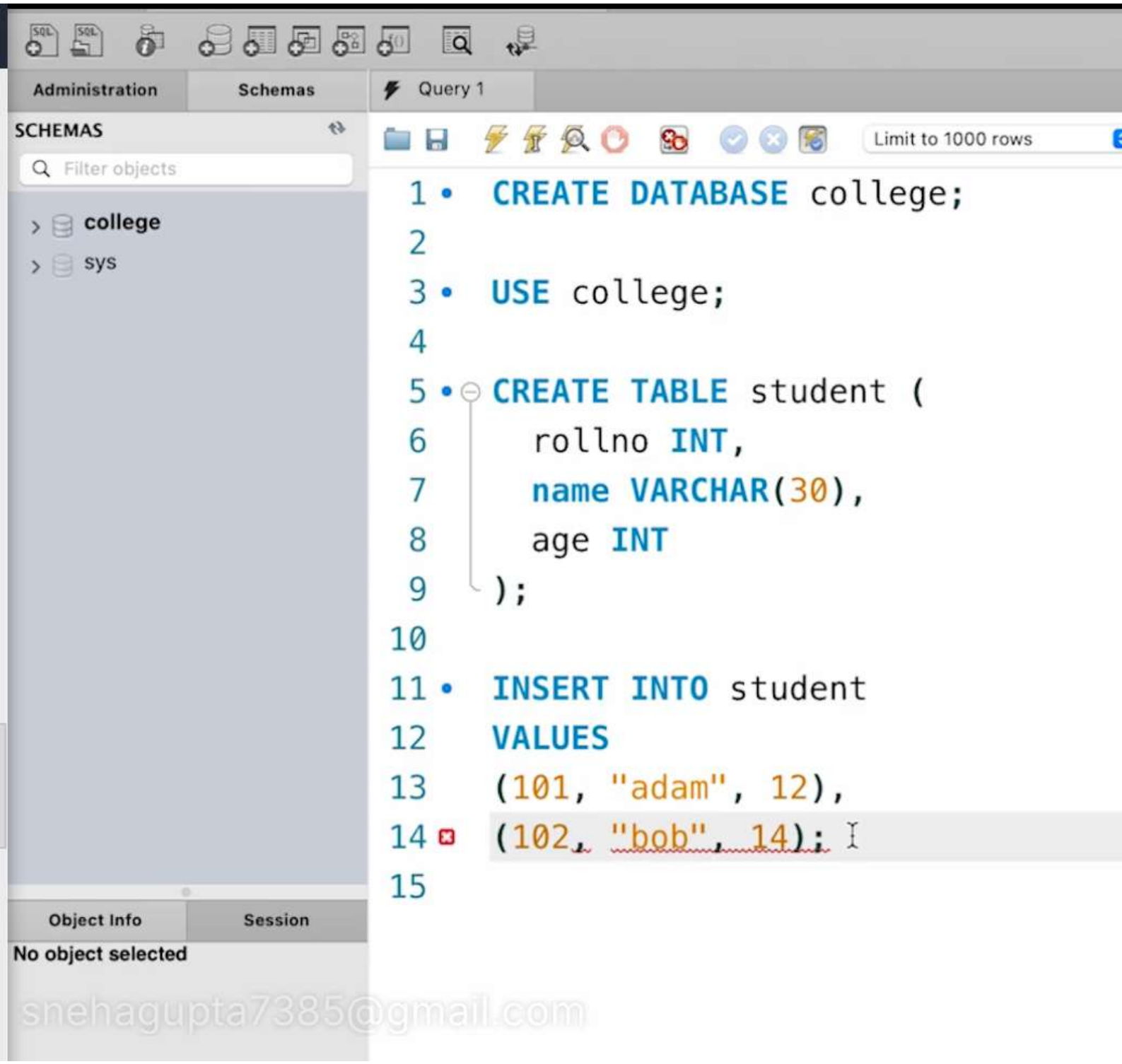

Delta Connection - Warning - not supported
Administration
Schemas
Query 1
SCHEMAS
Filter objects
college
Tables
Views
Stored Procedures
Functions
sys
Object Info
Session
No object selected
Limit to 1000 rows
1 • CREATE DATABASE college;
2
3 • USE college;
4
5
snehagupt

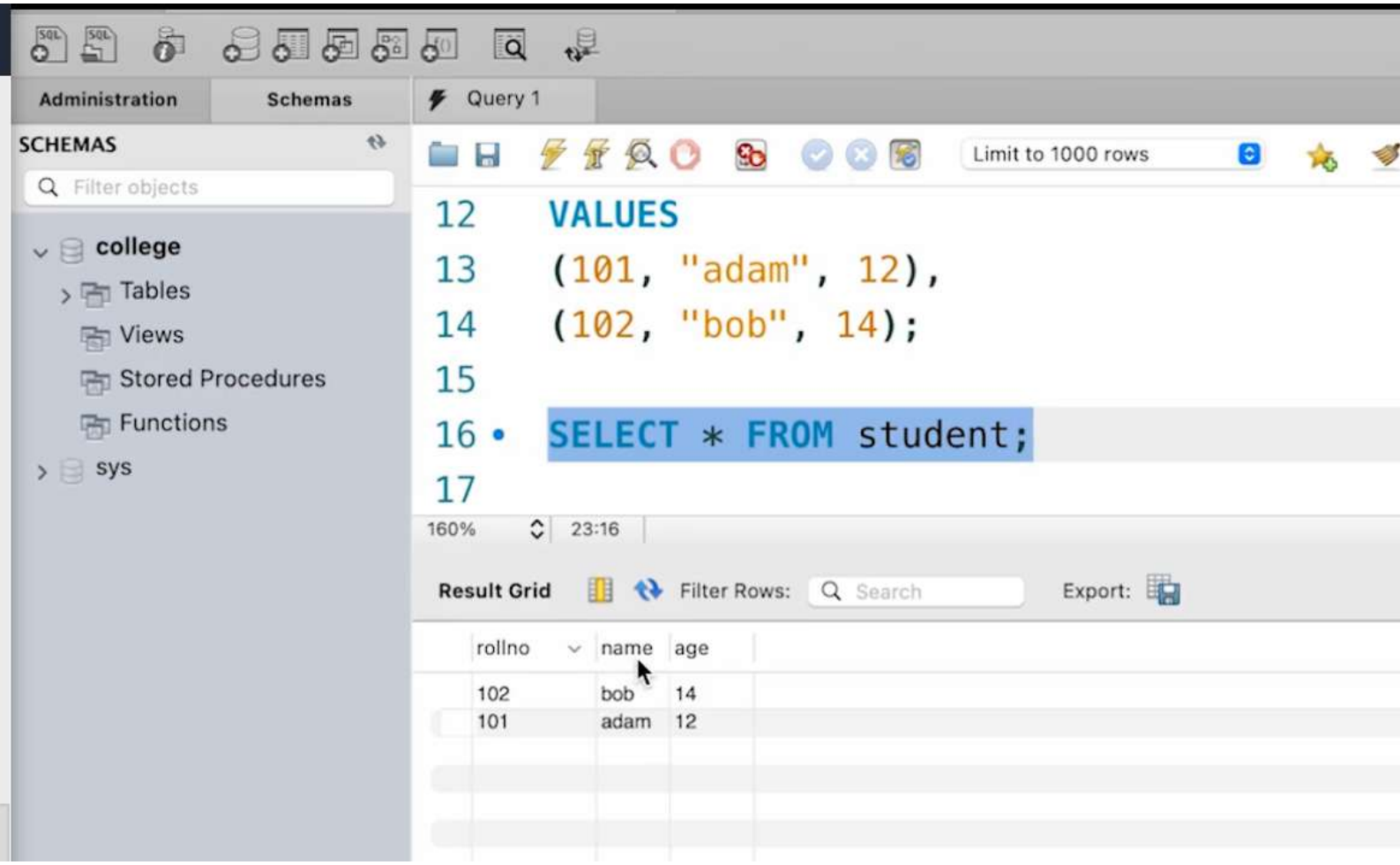
Our 1st Table

```
CREATE TABLE table_name (
    column_name1 datatype constraint,
    column_name2 datatype constraint,
    column_name2 datatype constraint
);
```

```
CREATE TABLE student (
    roll_no INT,
    name VARCHAR(30),
    age INT
);
```


Delta Connection - Warning - not supported
Administration
Schemas
Query 1
SCHEMAS
Filter objects
college
Tables
Views
Stored Procedures
Functions
sys
Object Info
Session
No object selected
Limit to 1000 rows
1 • CREATE DATABASE college;
2
3 • USE college;
4
5 • CREATE TABLE student (
6 rollno INT,
7 name VARCHAR(30),
8 age INT
9);
10
11




Administration
Schemas
Query 1
SCHEMAS
Filter objects
college
Tables
Views
Stored Procedures
Functions
sys
Limit to 1000 rows
12 VALUES
13 (101, "adam", 12),
14 (102, "bob", 14);
15
16 SELECT * FROM student;
17
160%
23:16
Result Grid
Filter Rows:
Search
Export:
rollno
name
age
102
bob
14
101
adam
12

Database Queries

```
CREATE DATABASE db_name;

CREATE DATABASE IF NOT EXISTS db_name;


DROP DATABASE db_name;

DROP DATABASE IF EXISTS db_name;


SHOW DATABASES;

SHOW TABLES;
```

Table Queries

- Create
- Insert
- Update
- Alter
- Truncate
- Delete

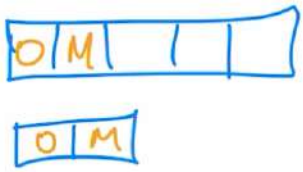
Table Queries

Create Table (schema / columns)

```
CREATE TABLE table_name (
    column_name1 datatype constraint,
    column_name2 datatype constraint,
);
```

User

Id	Name	Email	Followers	Following
1	Adam	adam@yahoo.in	123	145
2	Bob	bob123@gmail.com	200	200
3	Casey	casey@email.com	300	306
4	Donald	donald@gmail.com	200	105



name CHAR(5)
name VARCHAR(5)

Data Types

DATATYPE	DESCRIPTION	USAGE
CHAR	string(0-255), can store characters of fixed length	CHAR(50)
VARCHAR	string(0-255), can store characters up to given length	VARCHAR(50)
BLOB	string(0-65535), can store binary large object	BLOB(1000)
INT	integer(-2,147,483,648 to 2,147,483,647)	INT
TINYINT	integer(-128 to 127)	TINYINT
BIGINT	integer(-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807)	BIGINT
BIT	can store x-bit values. x can range from 1 to 64	BIT(2)
FLOAT	Decimal number - with precision to 23 digits	FLOAT
DOUBLE	Decimal number - with 24 to 53 digits	DOUBLE
BOOLEAN	Boolean values 0 or 1	BOOLEAN
DATE	date in format of YYYY-MM-DD ranging from 1000-01-01 to 9999-12-31	DATE
YEAR	year in 4 digits format ranging from 1901 to 2155	YEAR

Data Types

DATATYPE	DESCRIPTION	USAGE
CHAR	string(0-255), can store characters of fixed length	CHAR(50)
VARCHAR	string(0-255), can store characters up to given length	VARCHAR(50)
BLOB	string(0-65535), can store binary large object	BLOB(1000)
INT	integer(-2,147,483,648 to 2,147,483,647)	INT
TINYINT	integer(-128 to 127)	TINYINT
BIGINT	integer(-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807)	BIGINT
BIT	can store x-bit values. x can range from 1 to 64	BIT(2)
FLOAT	Decimal number - with precision to 23 digits	FLOAT
DOUBLE	Decimal number - with 24 to 53 digits	DOUBLE
BOOLEAN	Boolean values 0 or 1	BOOLEAN
DATE	date in format of YYYY-MM-DD ranging from 1000-01-01 to 9999-12-31	DATE
YEAR	year in 4 digits format ranging from 1901 to 2155	YEAR

Table Queries

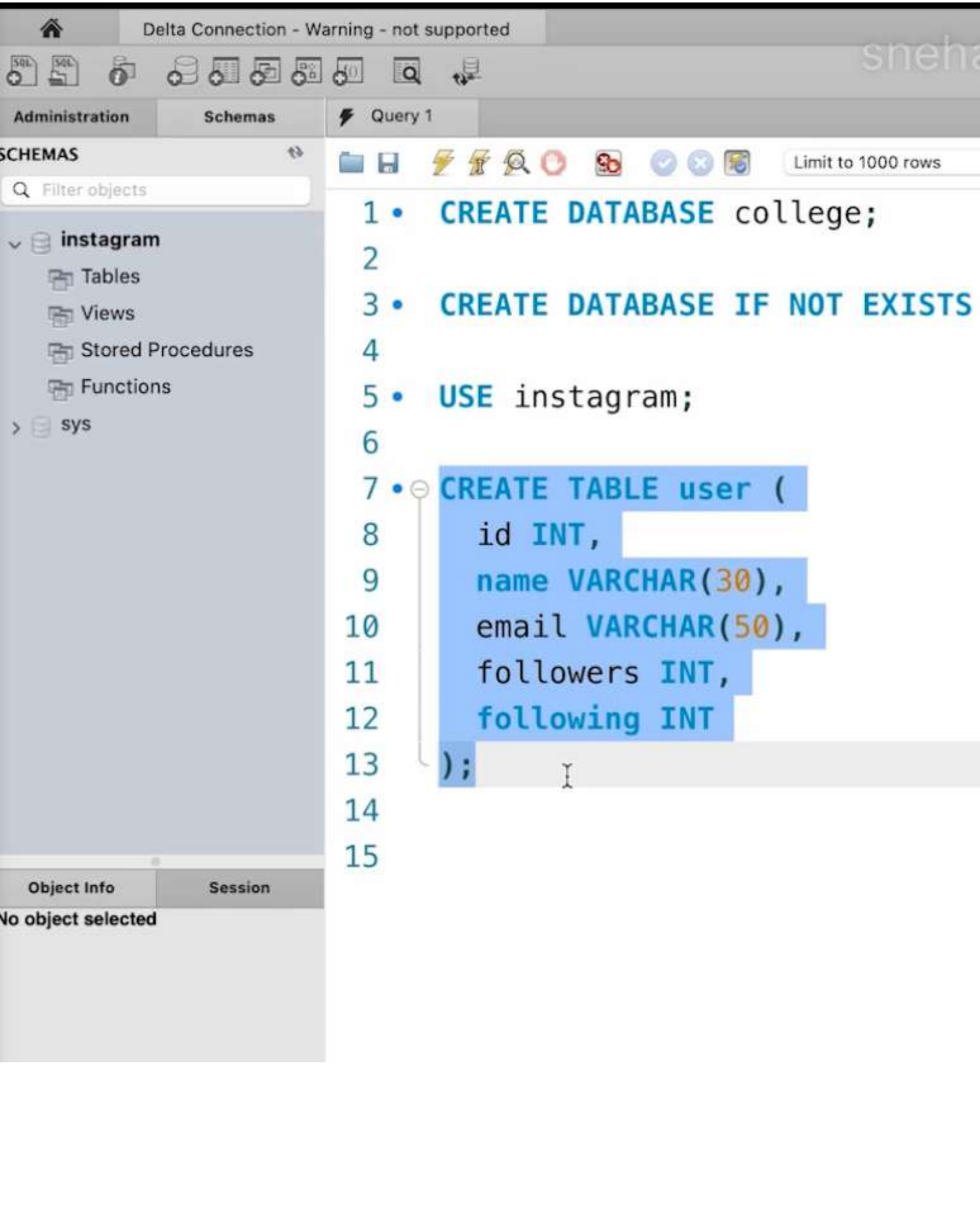
Create Table (schema / columns)

```
CREATE TABLE table_name (
    column_name1 datatype constraint,
    column_name2 datatype constraint,
);
```

INT VARCHAR VARCHAR INT INT

User

Id	Name	Email	Followers	Following
1	Adam	adam@yahoo.in	123	145
2	Bob	bob123@gmail.com	200	200
3	Casey	casey@email.com	300	306
4	Donald	donald@gmail.com	200	105


Delta Connection - Warning - not supported
Administration
Schemas
Query 1
SCHEMAS
Filter objects
instagram
Tables
Views
Stored Procedures
Functions
sys
Object Info
Session
No object selected
Limit to 1000 rows
1 • CREATE DATABASE college;
2
3 • CREATE DATABASE IF NOT EXISTS
4
5 • USE instagram;
6
7 • CREATE TABLE user (
8 id INT,
9 name VARCHAR(30),
10 email VARCHAR(50),
11 followers INT,
12 following INT
13);
14
15

Constraints

Rules for data in the table

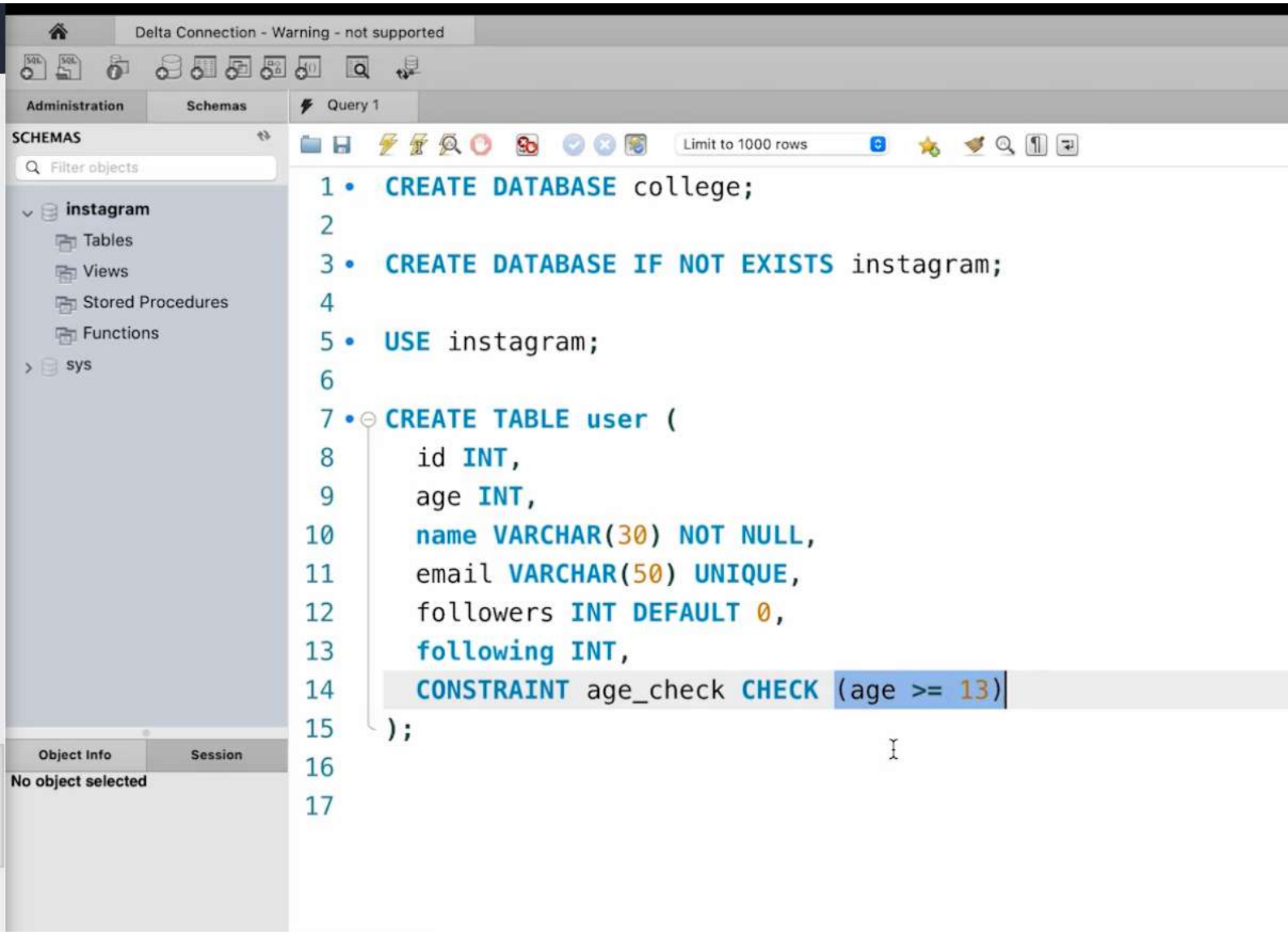
NOT NULL	columns cannot have a null value
UNIQUE	all values in column are different
DEFAULT	sets the default value of a column
CHECK	it can limit the values allowed in a column

```
salary INT DEFAULT 25000
```

```
CONSTRAINT age_check CHECK (age >= 18 AND city="Delhi")
```

User

id	name	email	followers	following
1	adam	adam@yahoo.in	123	145
2	bob	bob123@gmail.com	200	200
3	casey	casey@email.com	300	306
4	donald	donald@gmail.com	200	105


Delta Connection - Warning - not supported
Administration
Schemas
Query 1
SCHEMAS
Filter objects
instagram
Tables
Views
Stored Procedures
Functions
sys
Limit to 1000 rows
1 • CREATE DATABASE college;
2
3 • CREATE DATABASE IF NOT EXISTS instagram;
4
5 • USE instagram;
6
7 • CREATE TABLE user (
8 id INT,
9 age INT,
10 name VARCHAR(30) NOT NULL,
11 email VARCHAR(50) UNIQUE,
12 followers INT DEFAULT 0,
13 following INT,
14 CONSTRAINT age_check CHECK (age >= 13)
15);
16
17
Object Info
Session
No object selected

Constraints

PRIMARY KEY makes a column unique & not null but used only for one

```
CREATE TABLE temp (
  id int not null,
  PRIMARY KEY (id)
);
```

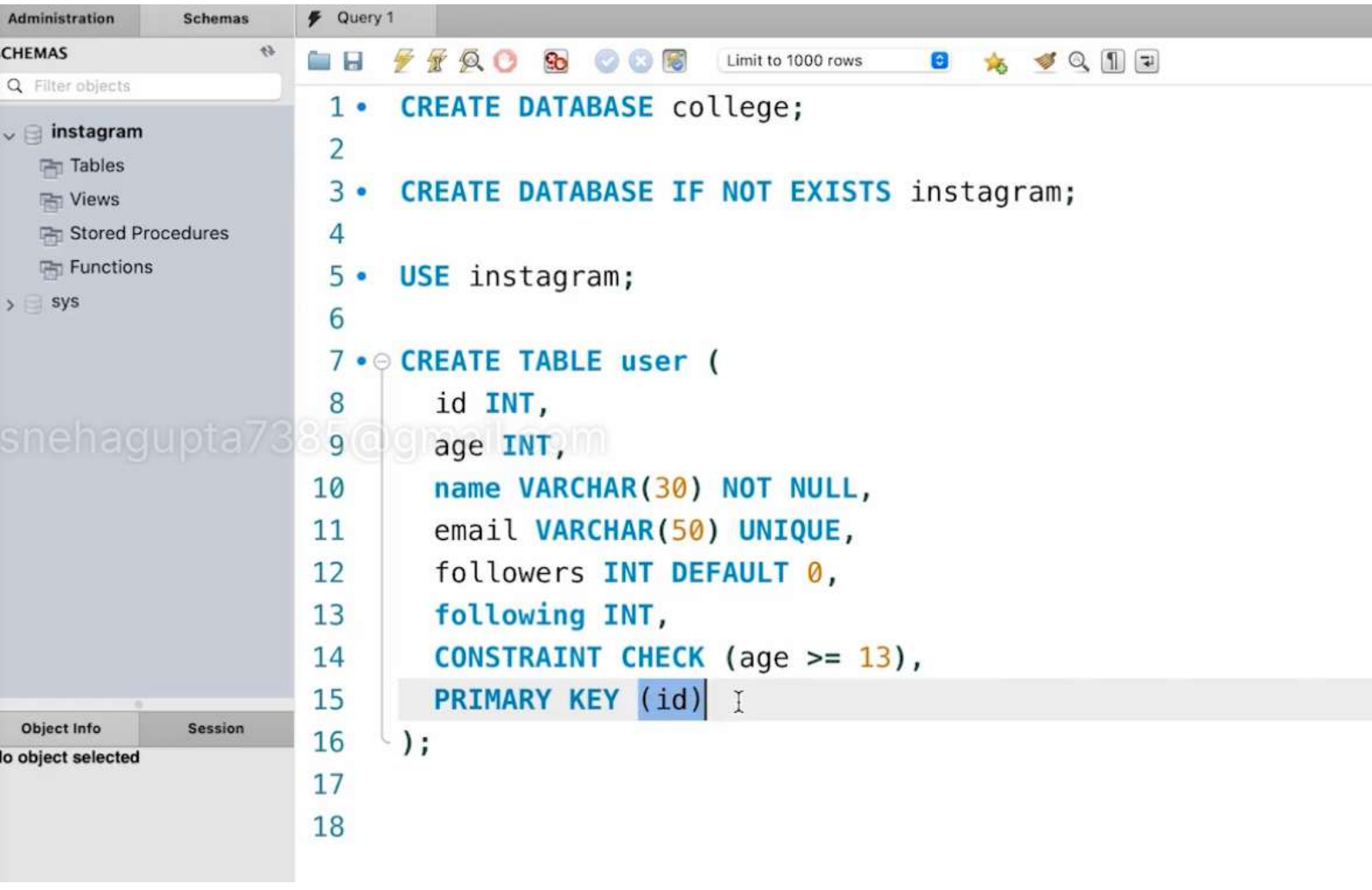
FOREIGN KEY prevent actions that would destroy links between tables

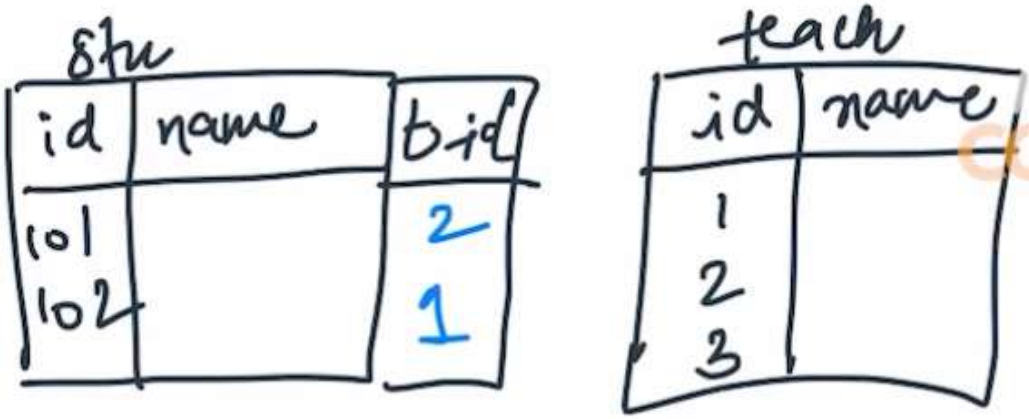
```
CREATE TABLE temp (
  cust_id int,
  FOREIGN KEY (cust_id) references customer(id)
);
```

User

id	age	name	email	followers	following
1		adam	adam@yahoo.in	123	145
2		bob	bob123@gmail.com	200	200
3		casey	casey@email.com	300	306
4		donald	donald@gmail.com	200	105

```sql

CREATE TABLE user (
    id INT PRIMARY KEY,
    age INT,
    name VARCHAR(30) NOT NULL,
    email VARCHAR(50) UNIQUE,
    followers INT DEFAULT 0,
    following INT,
    CONSTRAINT CHECK (age >= 13)
);


```


Administration
Schemas
Query 1
CHEMAS
Filter objects
instagram
Tables
Views
Stored Procedures
Functions
sys
Object Info
Session
lo object selected
Limit to 1000 rows
1 • CREATE DATABASE college;
2
3 • CREATE DATABASE IF NOT EXISTS instagram;
4
5 • USE instagram;
6
7 • CREATE TABLE user (
8 id INT,
9 age INT,
10 name VARCHAR(30) NOT NULL,
11 email VARCHAR(50) UNIQUE,
12 followers INT DEFAULT 0,
13 following INT,
14 CONSTRAINT CHECK (age >= 13),
15 PRIMARY KEY (id)
16);
17
18



ot null but used only for one

```
,,
FOREIGN KEY (t_id) REFERENCES teach(id)
```

User

id	age	name	email	followers	following
1		adam	adam@yahoo.in	123	145
2		bob	bob123@gmail.com	200	200
3		casey	casey@email.com	300	306
4		donald	donald@gmail.com	200	105

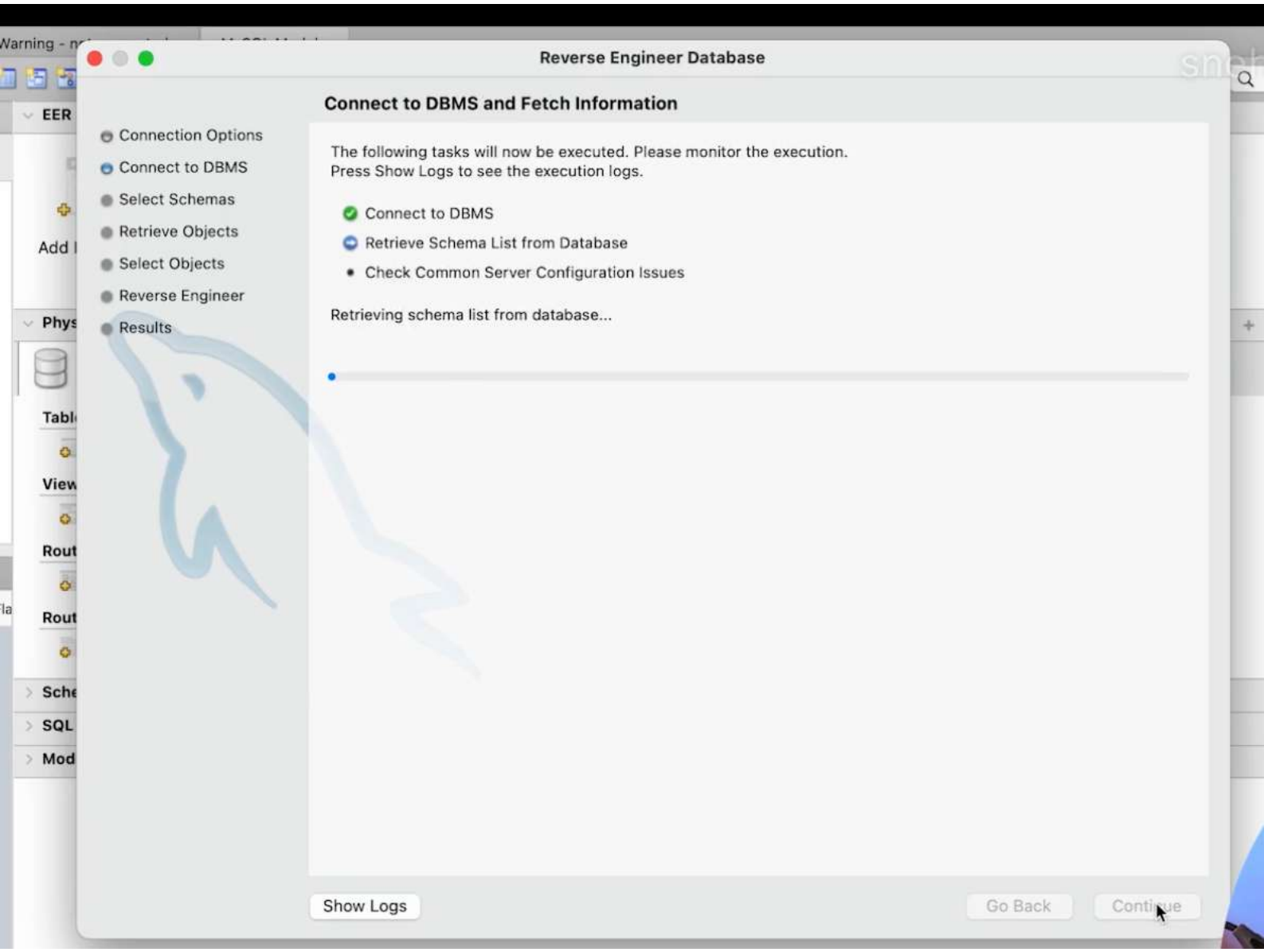
Posts

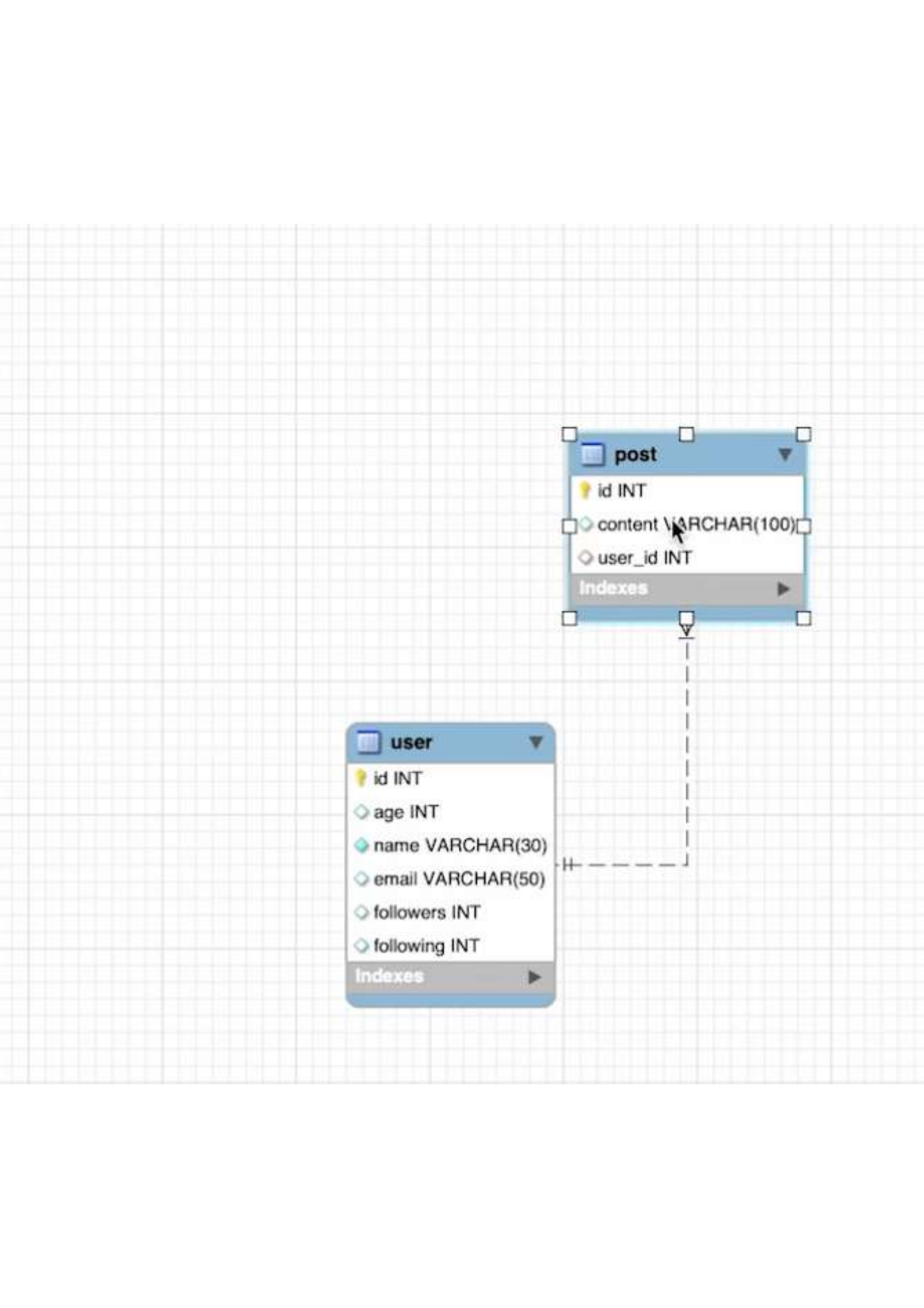
id	content	user_id
101	"Hello World"	3
102	"Bye Bye"	1

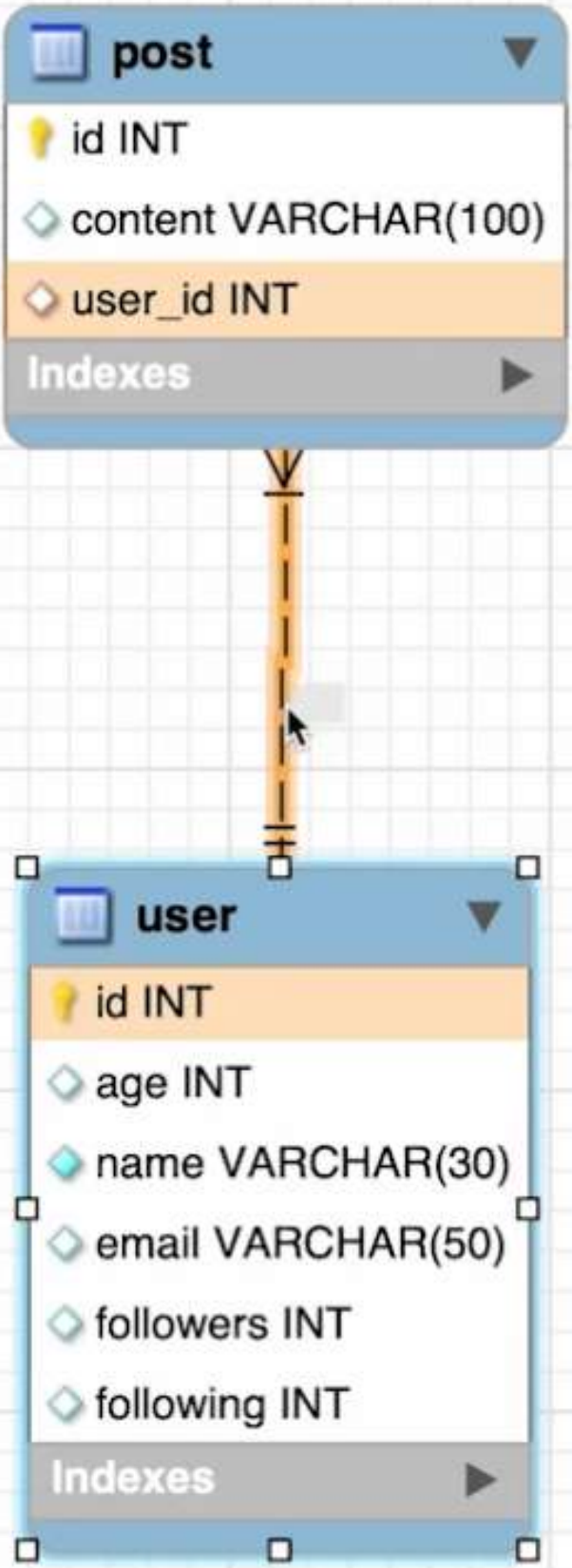
```
6
7  CREATE TABLE user (
8      id INT,
9      age INT,
10     name VARCHAR(30) NOT NULL,
11     email VARCHAR(50) UNIQUE,
12     followers INT DEFAULT 0,
13     following INT,
14     CONSTRAINT CHECK (age >= 13),
15     PRIMARY KEY (id)
16 );
17
18 CREATE TABLE post (
19     id INT PRIMARY KEY,
20     content VARCHAR(100),
21     user_id INT,
22     FOREIGN KEY user_id REFERENCES user(id)
23 );
24
25
```

```sql

CREATE TABLE user (
    id INT,
    age INT,
    name VARCHAR(30) NOT NULL,
    email VARCHAR(50) UNIQUE,
    followers INT DEFAULT 0,
    following INT,
    CONSTRAINT CHECK (age >= 13),
    PRIMARY KEY (id)
);

CREATE TABLE post (
    id INT PRIMARY KEY,
    content VARCHAR(100),
    user_id INT,
    FOREIGN KEY (user_id) REFERENCES user(id)
);

```


View
Query
Database
Server
Tools
Scripting
Help
Connect to Database... ⌘U
Manage Connections...
Reverse Engineer... ⌘R
Schema Transfer Wizard...
Migration Wizard...
Edit Type Mappings for Generic Migration...
Search Table Data...
Workbench
CREATE
id
age
name VARCHAR(30) NOT NULL,


Reverse Engineer Database
Connection Options
Connect to DBMS
Select Schemas
Retrieve Objects
Select Objects
Reverse Engineer
Results
Connect to DBMS and Fetch Information
The following tasks will now be executed. Please monitor the execution.
Press Show Logs to see the execution logs.
Connect to DBMS
Retrieve Schema List from Database
Check Common Server Configuration Issues
Retrieving schema list from database...
Show Logs
Go Back
Continue


post
id INT
content VARCHAR(100)
user_id INT
Indexes
user
id INT
age INT
name VARCHAR(30)
email VARCHAR(50)
followers INT
following INT
Indexes


post
id INT
content VARCHAR(100)
user_id INT
Indexes
user
id INT
age INT
name VARCHAR(30)
email VARCHAR(50)
followers INT
following INT
Indexes

ER Diagram

Relation

column unique & not null b

What are Keys?

Keys are special columns in the table

Primary Key

It is a column (or set of columns) in a table that uniquely identifies each row. (a unique id)

There is only 1 PK & it should be NOT null.

Foreign Key

A foreign key is a column (or set of columns) in a table that refers to the primary key in

FKs can have duplicate & null values.

There can be multiple FKs.

Primary & Foreign Keys

User

Id	Name	Email	Followers	Following
1	Adam	adam@yahoo.in	123	145
2	Bob	bob123@gmail.com	200	200
3	Casey	casey@email.com	300	306
4	Donald	donald@gmail.com	200	105

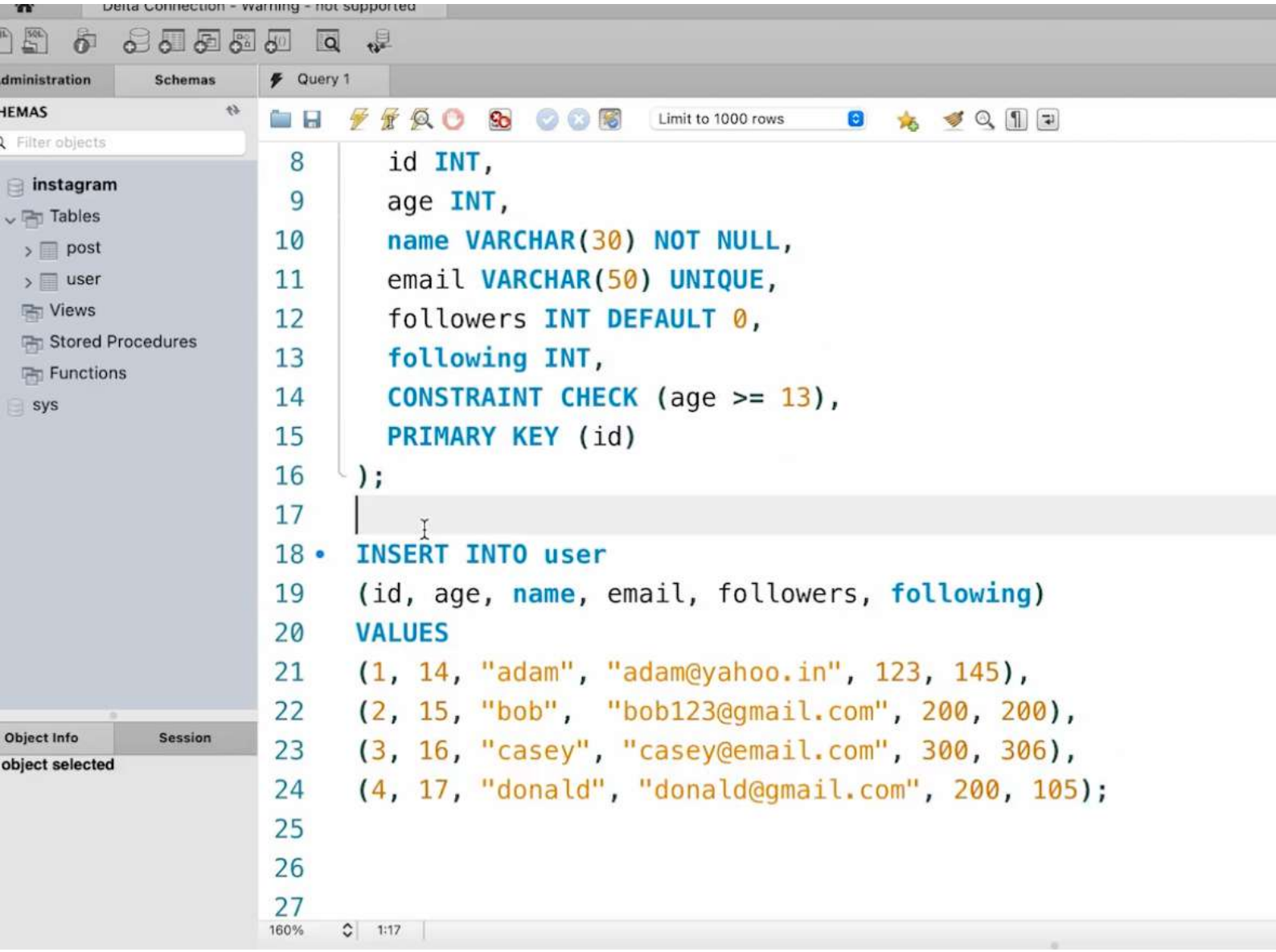
Posts

Id	Content	User_id
1	"Hello World"	3
2	"I am back!"	4
3	"Bye Bye"	1

Table Queries

Insert into Table

```
INSERT INTO table_name
(colname1, colname2);
VALUES
(col1_v1, col2_v1),
(col1_v2, col2_v2);
```


Delta Connection - Warning - not supported
Administration
Schemas
Query 1
HEMAS
Filter objects
instagram
Tables
post
user
Views
Stored Procedures
Functions
sys
Object Info
Session
object selected
Limit to 1000 rows
id INT,
age INT,
name VARCHAR(30) NOT NULL,
email VARCHAR(50) UNIQUE,
followers INT DEFAULT 0,
following INT,
CONSTRAINT CHECK (age >= 13),
PRIMARY KEY (id)
);
INSERT INTO user
(id, age, name, email, followers, following)
VALUES
(1, 14, "adam", "adam@yahoo.in", 123, 145),
(2, 15, "bob", "bob123@gmail.com", 200, 200),
(3, 16, "casey", "casey@email.com", 300, 306),
(4, 17, "donald", "donald@gmail.com", 200, 105);
160%
1:17

```
17
18 • INSERT INTO user
19   (id, age, name)
20   VALUES
21   (2, 20, "random");
22
23
24 • INSERT INTO user
25   (id, age, name, email, followers, following)
26   VALUES
27   (1, 14, "adam", "adam@yahoo.in", 123, 145),
28   (2, 15, "bob",  "bob123@gmail.com", 200, 200),
```

160% 1:18

Action Output

	Time	Action	Response
✓	17	15:08:51 CREATE TABLE user (id INT, age INT, name VARCHAR(30) NOT NULL, email VARCHAR(50) UNIQUE, followers IN...	0 row(s) affected
✓	18	15:10:44 CREATE TABLE post (id INT PRIMARY KEY, content VARCHAR(100), user_id INT, FOREIGN KEY (user_id) REFERE...	0 row(s) affected
✓	19	15:22:00 INSERT INTO user (id, age, name, email, followers, following) VALUES (1, 14, "adam", "adam@yahoo.in", 123, 145), (2,...	4 row(s) affected Records: 4 Dupli
✗	20	15:23:01 INSERT INTO user (id, age, name) VALUES (2, 20, "random")	Error Code: 1062. Duplicate entry '2'

Select Command

Selects & Show data from the DB

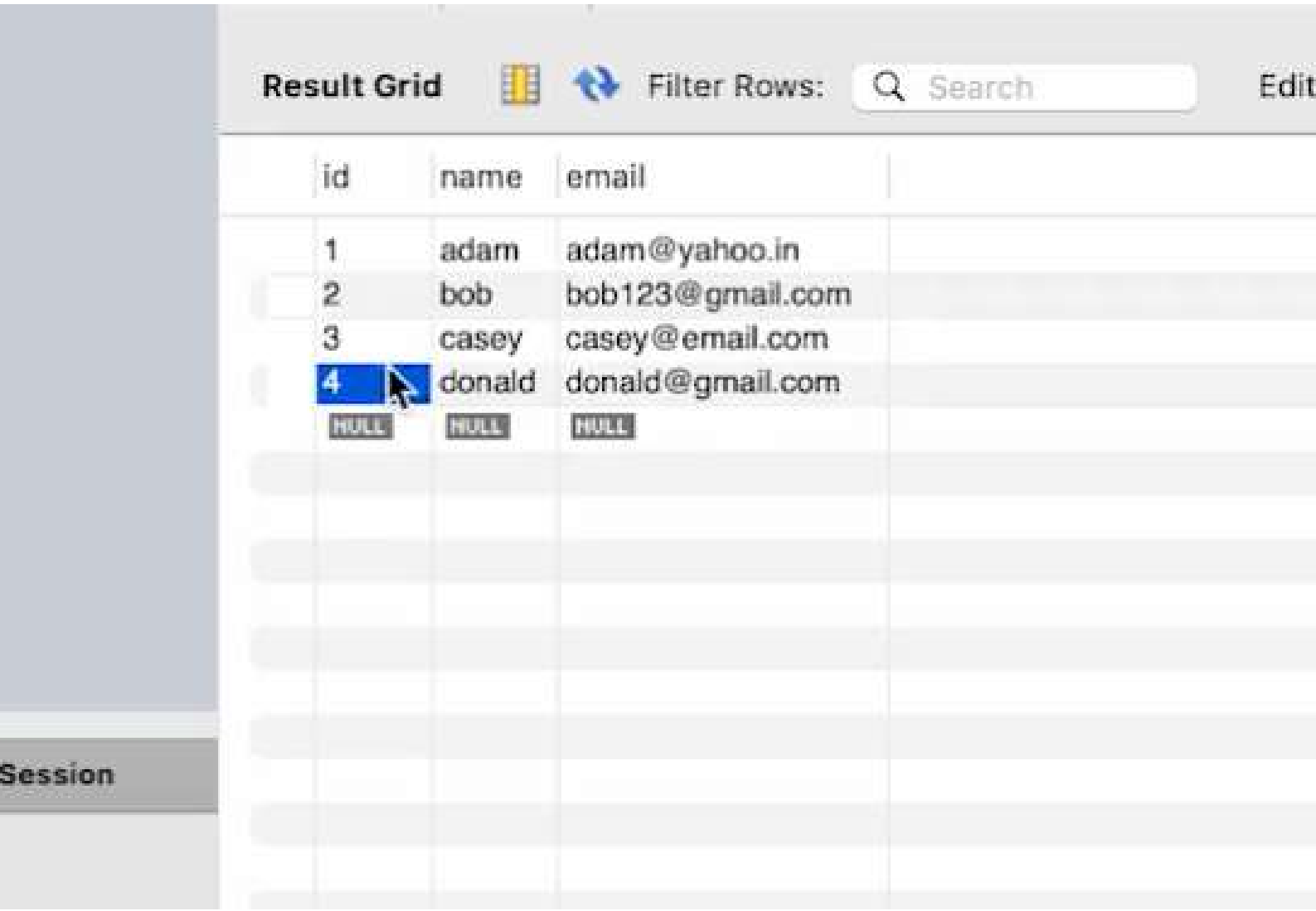
Syntax

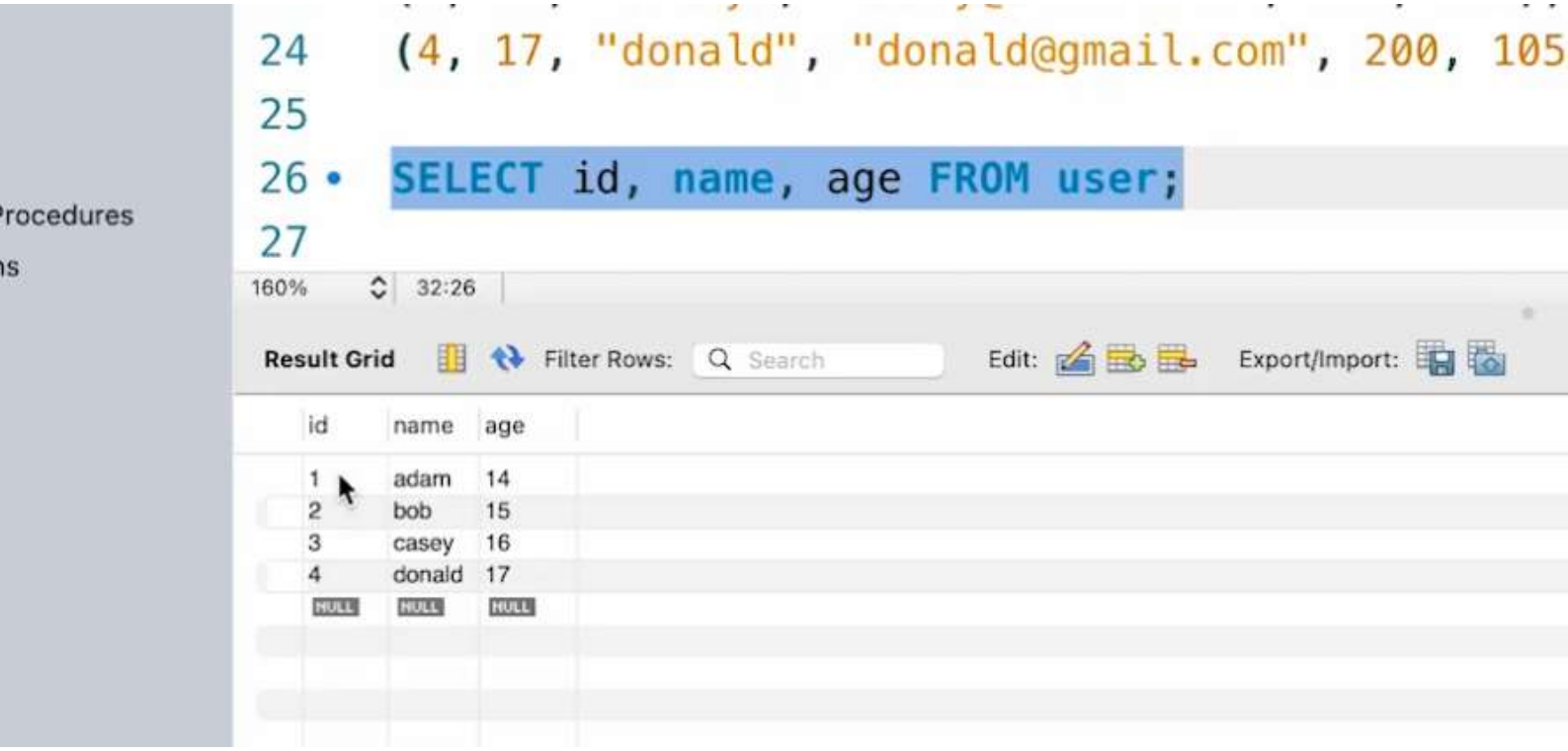
SELECT *col1, col2* **FROM** *table_name;*

Syntax **(to show all)**

SELECT * **FROM** *table_name;*

```
17
18 • INSERT INTO user
19   (id, age, name, email, followers, following)
20   VALUES
21   (1, 14, "adam", "adam@yahoo.in", 123, 145),
22   (2, 15, "bob",  "bob123@gmail.com", 200, 200),
23   (3, 16, "casey", "casey@email.com", 300, 306),
24   (4, 17, "donald", "donald@gmail.com", 200, 105);
25
26 • SELECT id, name, email FROM user;
27
28
```


Result Grid
Filter Rows:
Search
Edit
id
name
email
1
adam
adam@yahoo.in
2
bob
bob123@gmail.com
3
casey
casey@email.com
4
donald
donald@gmail.com
NULL
NULL
NULL
Session


Procedures
ns
24 (4, 17, "donald", "donald@gmail.com", 200, 105
25
26 • SELECT id, name, age FROM user;
27
160%
32:26
Result Grid
Filter Rows:
Search
Edit:
Export/Import:
id
name
age
1
adam
14
2
bob
15
3
casey
16
4
donald
17
NULL
NULL
NULL

```
25
26 •  SELECT * FROM user;
27
```

160% 20:26

Result Grid | Filter Rows: Search | Edit:

id	age	name	email	followers	following
1	14	adam	adam@yahoo.in	123	145
2	15	bob	bob123@gmail.com	200	200
3	16	casey	casey@email.com	300	306
4	17	donald	donald@gmail.com	200	105
NULL	NULL	NULL	NULL	NULL	NULL

```
25
26 •  SELECT DISTINCT age FROM user;
27
```

160% 31:26

Result Grid | Filter Rows: Search | Export:

age
14
15
16
17