Connect to MySQL Server
And
View Data

Overview

- In this tutorial we will demonstrate:
  - Connecting to a MySQL server and connecting to a database using the NetBeans IDE
  - Viewing data using the NetBeans IDE
  - Running an SQL Query a Web-Based System with the output in a webpage
- Set a practical exercise to:
  - Improve the PHP script using `try…catch…finally` blocks

MySQL Server

- The following slides show the process to:
  - Connect to a MySQL server
  - Connect to a database
    - The example shows an example database embedded in the NetBeans IDE
  - Viewing data in the ‘Customers’ database table
  - The SQL statement used to create the ‘view’
    - Two SQL statements demonstrate alternative SQL methods

Starting the Apache Server and MySQL

NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

Projects | Services | Files

- Databases
  - MySQL Server at localhost:3306 [root]
    - information_schema
    - mysql
    - performance_schema
    - phpmyadmin
    - test
  - Java DB
  - Drivers
  - jdbc:derby://localhost:1527/sample [app on APP]
  - jdbc:mysql://localhost:3306/information_schema?zeroDateTimeBehavior=convertToNull [root on Default schema]
    - **information_schema**
      - Tables
      - Views
      - Procedures
    - Other databases
      - mysql
      - performance_schema
      - phpmyadmin
      - test
  - jdbc:mysql://localhost:3306/mysql?zeroDateTimeBehavior=convertToNull [root on Default schema]
  - jdbc:mysql://localhost:3306/personal?zeroDateTimeBehavior=convertToNull [root on Default schema]
- Web Services
- Servers
- Maven Repositories
- Cloud
- Hudson Builders
- Docker
- Task Repositories
- JS Test Driver
- Selenium Server

XAMPP Control Panel v3.2.2 [Compiled: Nov 12th 2015]

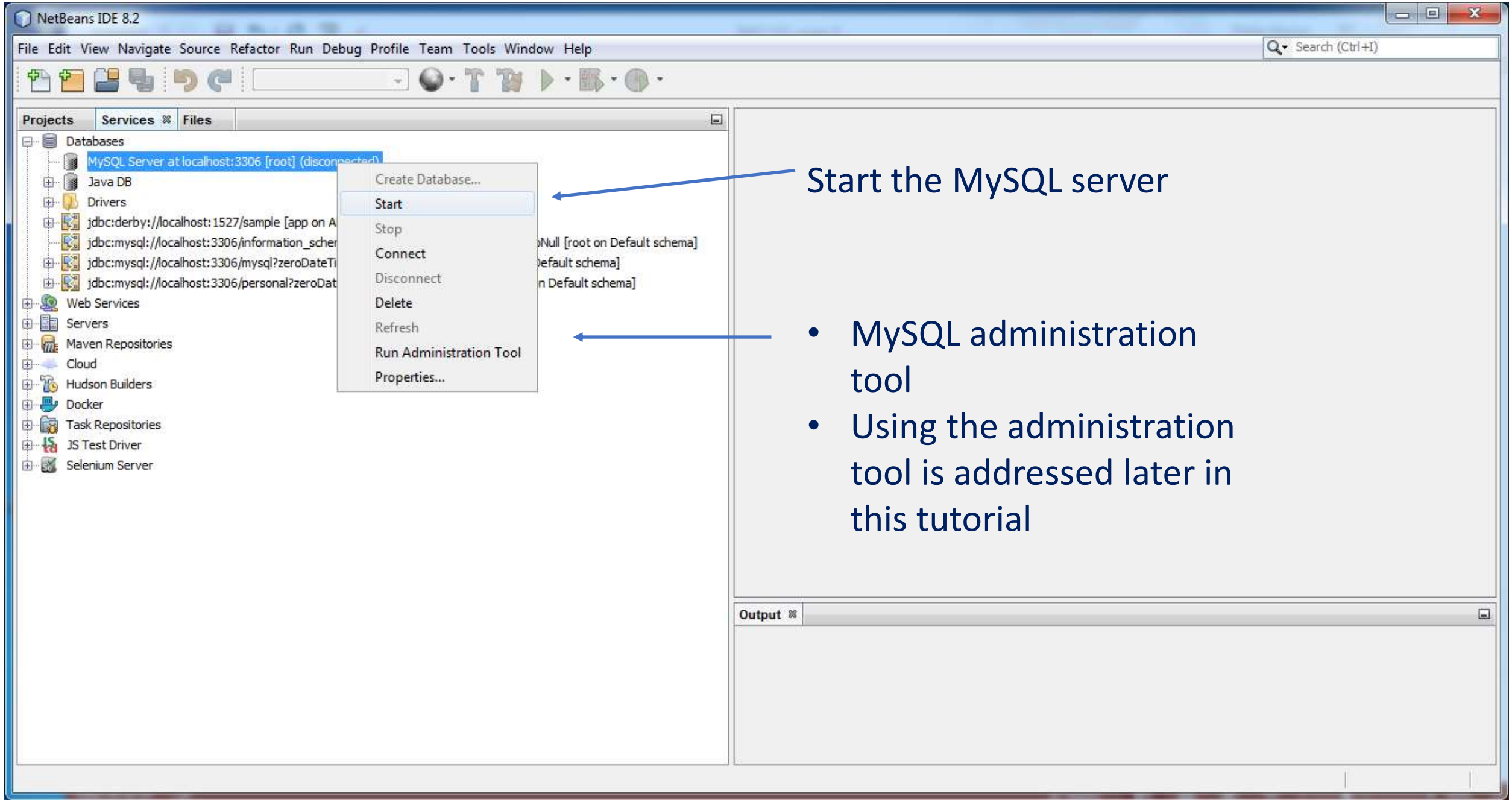
XAMPP Control Panel v3.2.2

Modules

Service	Module	PID(s)	Port(s)	Actions			
	Apache	6296 7696	80, 443	Stop	Admin	Config	Logs
	MySQL	7808	3306	Stop	Admin	Config	Logs
	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
	Tomcat			Start	Admin	Config	Logs

Config | Netstat | Shell | Explorer | Services | Help | Quit

```
09:33:24 [main]   All prerequisites found
09:33:24 [main]   Initializing Modules
09:33:24 [main]   Starting Check-Timer
09:33:24 [main]   Control Panel Ready
09:33:27 [Apache] Attempting to start Apache app...
09:33:28 [mysql]  Attempting to start MySQL app...
09:33:33 [Apache] Status change detected: running
09:33:34 [mysql]  Status change detected: running
```



Start the MySQL server

- MySQL administration tool
- Using the administration tool is addressed later in this tutorial

NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

Projects | Services | Files

- Databases
  - MySQL Server at localhost:3306 [root]
  - Java DB
  - Drivers
    - Java DB (Embedded)
    - Java DB (Network)
    - MySQL (Connector/J driver)
    - Oracle OCI
    - Oracle Thin
    - PostgreSQL
  - jdbc:derby://localhost:1527/sample [app on APP]
  - jdbc:mysql://localhost:3306/information_schema?zero
  - jdbc:mysql://localhost:3306/mysql?zeroDateTimeBeha
  - jdbc:mysql://localhost:3306/personal?zeroDateTimeBe
- Web Services
- Servers
- Maven Repositories
- Cloud
- Hudson Builders
- Docker
- Task Repositories
- JS Test Driver
- Selenium Server

Connected to MySQL server running on ‘root’
Running on port 3306 [root]

Output - Java DB Database Process

```
Thu Aug 16 10:13:58 BST 2018 : Security manager installed using the Basic server security policy.
Thu Aug 16 10:13:59 BST 2018 : Apache Derby Network Server - 10.11.1.2 - (1629631) started and ready to accept connections on port 1527
```

Connect to MySQL database

NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

Projects | Services | Files

- Databases
  - MySQL Server at localhost:3306 [root]
  - Java DB
  - Drivers
  - jdbc:derby://localhost:1527/sample [app on APP]
  - jdbc:mysql://localhost:3306/ ... vior=convertToNull [root on Default schema]
  - jdbc:mysql://localhost:3306/ ... oNull [root on Default schema]
  - jdbc:mysql://localhost:3306/ ... rtToNull [root on Default schema]
- Web Services
- Servers
- Maven Repositories
- Cloud
- Hudson Builders
- Docker
- Task Repositories
- JS Test Driver
- Selenium Server

Connect...
Disconnect
Execute Command...
Refresh
Delete Delete
Rename...
Properties

Output

NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

Projects | Services | Files

- Databases
  - MySQL Server at localhost:3306 [root]
  - Java DB
  - Drivers
  - jdbc:derby://localhost:1527/sample [app on APP]
    - APP
      - Tables
        - CUSTOMER
          - CUSTOMER_ID
          - DISCOUNT_CODE
          - ZIP
          - NAME
          - ADDRESSLINE1
          - ADDRESSLINE2
          - CITY
          - STATE
          - PHONE
          - FAX
          - EMAIL
          - CREDIT_LIMIT
          - Indexes
          - Foreign Keys
        - DISCOUNT_CODE
        - MANUFACTURER
        - MICRO_MARKET
        - PRODUCT
        - PRODUCT_CODE
        - PURCHASE_ORDER
      - Views
      - Procedures

Shows the MySQL database is connected

Output - Java DB Database Process

```
Thu Aug 16 10:13:58 BST 2018 : Security manager installed using the Basic server security policy.
Thu Aug 16 10:13:59 BST 2018 : Apache Derby Network Server - 10.11.1.2 - (1629631) started and ready to accept connections on port 1527
```

NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Projects | Services | Files

- Databases
  - MySQL Server at localhost:3306 [root]
  - Java DB
  - Drivers
  - jdbc:derby://localhost:1527/sample [app on APP]
    - APP
      - Tables
        - CUSTOMER
          - CUSTOMER_ID
          - DISCOUNT_CODE
          - ZIP
          - NAME
          - ADDRESSLINE1
          - ADDRESSLINE2
          - CITY
          - STATE
          - PHONE
          - FAX
          - EMAIL
          - CREDIT_LIMIT
          - Indexes
          - Foreign Keys
        - DISCOUNT_CODE
        - MANUFACTURER
        - MICRO_MARKET
        - PRODUCT
        - PRODUCT_CODE
        - PURCHASE_ORDER
      - Views
      - Procedures

View the database tables

Output - Java DB Database Process

```
Thu Aug 16 10:13:58 BST 2018 : Security manager installed using the Basic server security policy.
Thu Aug 16 10:13:59 BST 2018 : Apache Derby Network Server - 10.11.1.2 - (1629631) started and ready to accept connections on port 1527
```

NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

Projects | Services | Files

- Databases
  - MySQL Server at localhost:3306 [root]
  - Java DB
  - Drivers
  - jdbc:derby://localhost:1527/sample [app on APP]
    - APP
      - Tables
        - CUSTOMER
          - CREDIT_LIMIT
          - Indexes
          - Foreign Keys
        - DISCOUNT_CODE
        - MANUFACTURER
        - MICRO_MARKET
        - PRODUCT
        - PRODUCT_CODE
        - PURCHASE_ORDER
      - Views
      - Procedures

View Data...
Execute Command...
Add Column...
Refresh
Delete Delete
Grab Structure...
Recreate Table...
Properties

View data

Output - Java DB Database Process

```
Thu Aug 16 10:13:58 BST 2018 : Security manager installed using the Basic server security policy.
Thu Aug 16 10:13:59 BST 2018 : Apache Derby Network Server - 10.11.1.2 - (1629631) started and ready to accept connections on port 1527
```

NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

Projects | Services | Files

- Databases
  - MySQL Server at localhost:3306 [root]
  - Java DB
  - Drivers
  - jdbc:derby://localhost:1527/sample [app on APP]
    - APP
      - Tables
        - CUSTOMER
          - View Data...
          - Execute Command...
          - Add Column...
          - Refresh
          - Delete Delete
          - Grab Structure...
          - Recreate Table...
          - Properties
          - CREDIT_LIMIT
          - Indexes
          - Foreign Keys
        - DISCOUNT_CODE
        - MANUFACTURER
        - MICRO_MARKET
        - PRODUCT
        - PRODUCT_CODE
        - PURCHASE_ORDER
      - Views
      - Procedures

Write and execute SQL statement

Output - Java DB Database Process

```
Thu Aug 16 10:13:58 BST 2018 : Security manager installed using the Basic server security policy.
Thu Aug 16 10:13:59 BST 2018 : Apache Derby Network Server - 10.11.1.2 - (1629631) started and ready to accept connections on port 1527
```

NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Projects | Services | Files

- Databases
  - MySQL Server at localhost:3306 [root]
  - Java DB
  - Drivers
  - jdbc:derby://localhost:1527/sample [app on APP]
    - APP
      - Tables
        - CUSTOMER
          - CUSTOMER_ID
          - DISCOUNT_CODE
          - ZIP
          - NAME
          - ADDRESSLINE1
          - ADDRESSLINE2
          - CITY
          - STATE
          - PHONE
          - FAX
          - EMAIL
          - CREDIT_LIMIT
        - Indexes
        - Foreign Keys
        - DISCOUNT_CODE
        - MANUFACTURER
        - MICRO_MARKET
        - PRODUCT
        - PRODUCT_CODE
        - PURCHASE_ORDER
      - Views
      - Procedures

SQL 1 [jdbc:derby://localhost:15...] | SQL 2 [jdbc:derby://localhost:15...]

Connection: jdbc:derby://localhost:1527/sample [app on APP]

```
1   Select * FROM CUSTOMER
```

Run the SQL statement

Select * FROM CUSTOMER

Max. rows: 100 | Fetched Rows: 13 | Matching Rows:

#	CUSTOMER_ID	DISCOUNT_CODE	ZIP	NAME	ADDRESSLINE1	ADDRESSLINE2
1	1	N	95117	Jumbo Eagle Corp	111 E. Las Olivas Blvd	Suite 51
2	2	M	95035	New Enterprises	9754 Main Street	P.O. Box 567
3	25	M	85638	Wren Computers	8989 Red Albatross Drive	Suite 9897
4	3	L	12347	Small Bill Company	8585 South Upper Murray Drive	P.O. Box 456
5	36	H	94401	Bob Hosting Corp.	65653 Lake Road	Suite 2323
6	106	L	95035	Early CentralComp	829 E Flex Drive	Suite 853
7	149	L	95117	John Valley Computers	4381 Kelly Valley Ave	Suite 77
8	863	N	94401	Big Network Systems	456 444th Street	Suite 45
9	777	L	48128	West Valley Inc.	88 Northsouth Drive	Building C
10	753	H	48128	Zed Motor Co	2267 NE Michigan Ave	Building 21
11	722	N	48124	Big Car Parts	52963 Notouter Dr	Suite 35

Output

Java DB Database Process | SQL 1 execution | SQL 2 execution

```
Executed successfully in 0.01 s.
Fetching resultset took 0 s.
Line 1, column 1

Execution finished after 0.05 s, no errors occurred.
```

1:23 INS

NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Projects | Services | Files

- Databases
  - MySQL Server at localhost:3306 [root]
  - Java DB
  - Drivers
  - jdbc:derby://localhost:1527/sample [app on APP]
    - APP
      - Tables
        - CUSTOMER
          - CUSTOMER_ID
          - DISCOUNT_CODE
          - ZIP
          - NAME
          - ADDRESSLINE1
          - ADDRESSLINE2
          - CITY
          - STATE
          - PHONE
          - FAX
          - EMAIL
          - CREDIT_LIMIT
        - Indexes
        - Foreign Keys
        - DISCOUNT_CODE
        - MANUFACTURER
        - MICRO_MARKET
        - PRODUCT
        - PRODUCT_CODE
        - PURCHASE_ORDER
      - Views
      - Procedures

SQL 1 [jdbc:derby://localhost:15...] | SQL 2 [jdbc:derby://localhost:15...]

Connection: jdbc:derby://localhost:1527/sample [app on APP]

```
1  Select * FROM CUSTOMER
```

SQL statement

Select * FROM CUSTOMER

Max. rows: 100 | Fetched Rows: 13 | Matching Rows:

#	CUSTOMER_ID	DISCOUNT_CODE	ZIP	NAME	ADDRESSLINE1	ADDRESSLINE2
1	1	N	95117	Jumbo Eagle Corp	111 E. Las Olivas Blvd	Suite 51
2	2	M	95035	New Enterprises	9754 Main Street	P.O. Box 567
3	25	M	85638	Wren Computers	8989 Red Albatross Drive	Suite 9897
4	3	L	12347	Small Bill Company	8585 South Upper Murray Drive	P.O. Box 456
5	36	H	94401	Bob Hosting Corp.	65653 Lake Road	Suite 2323
6	106	L	95035	Early CentralComp	829 E Flex Drive	Suite 853
7	149	L	95117	John Valley Computers	4381 Kelly Valley Ave	Suite 77
8	863	N	94401	Big Network Systems	456 444th Street	Suite 45
9	777	L	48128	West Valley Inc.	88 Northsouth Drive	Building C
10	753	H	48128	Zed Motor Co	2267 NE Michigan Ave	Building 21
11	722	N	48124	Big Car Parts	52963 Notouter Dr	Suite 35

SQL results (a 'view')

Output

Java DB Database Process | SQL 1 execution | SQL 2 execution

```
Executed successfully in 0.01 s.
Fetching resultset took 0 s.
Line 1, column 1

Execution finished after 0.05 s, no errors occurred.
```

Output execution

1:23 INS

NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Projects | Services | Files

- Databases
  - MySQL Server at localhost:3306 [root]
  - Java DB
  - Drivers
  - jdbc:derby://localhost:1527/sample [app on APP]
    - APP
      - Tables
        - CUSTOMER
          - CUSTOMER_ID
          - DISCOUNT_CODE
          - ZIP
          - NAME
          - ADDRESSLINE1
          - ADDRESSLINE2
          - CITY
          - STATE
          - PHONE
          - FAX
          - EMAIL
          - CREDIT_LIMIT
        - Indexes
        - Foreign Keys
        - DISCOUNT_CODE
        - MANUFACTURER
        - MICRO_MARKET
        - PRODUCT
        - PRODUCT_CODE
        - PURCHASE_ORDER
      - Views
      - Procedures

SQL 1 [jdbc:derby://localhost:15...]

Connection: jdbc:derby://localhost:1527/sample [app on APP]

```
1  SELECT * FROM APP.CUSTOMER FETCH FIRST 100 ROWS ONLY;
2
```

SELECT * FROM APP.CUSTOME...

Max. rows: 100 | Fetched Rows: 13 | Matching Rows:

#	CUSTOMER_ID	DISCOUNT_CODE	ZIP	NAME	ADDRESSLINE1	ADDRESSLINE2
1	1	N	95117	Jumbo Eagle Corp	111 E. Las Olivas Blvd	Suite 51
2	2	M	95035	New Enterprises	9754 Main Street	P.O. Box 567
3	25	M	85638	Wren Computers	8989 Red Albatross Drive	Suite 9897
4	3	L	12347	Small Bill Company	8585 South Upper Murray Drive	P.O. Box 456
5	36	H	94401	Bob Hosting Corp.	65653 Lake Road	Suite 2323
6	106	L	95035	Early CentralComp	829 E Flex Drive	Suite 853
7	149	L	95117	John Valley Computers	4381 Kelly Valley Ave	Suite 77
8	863	N	94401	Big Network Systems	456 444th Street	Suite 45
9	777	L	48128	West Valley Inc.	88 Northsouth Drive	Building C
10	753	H	48128	Zed Motor Co	2267 NE Michigan Ave	Building 21
11	722	N	48124	Big Car Parts	52963 Notouter Dr	Suite 35

Output

Java DB Database Process | SQL 1 execution

```
Executed successfully in 0.367 s.
Fetching resultset took 0 s.
Line 1, column 1

Execution finished after 0.538 s, no errors occurred.
```

1:1 INS

SQL statement

SQL results (a 'view')

Output execution

Creating a MySQL Database

Overview

- We have introduced the basic database concepts with simple SQL statements and demonstrated how to create a database and populate it with records within the NetBeans IDE environment
- In this session we will show
  - The MySQL administration tool interface and how to create a MySQL user account
  - How to connect to a database and run MySQL (SQL) queries in a PHP script with the returned results presented in a web-browser
  - Note: use your own MySQL account details created in the previous tutorial session

NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Projects | Services | Files

- Databases
  - MySQL Server at localhost:3306 [root]
    - information_schema
    - mysql
    - performance_schema
    - phpmyadmin
    - test
  - Java DB
  - Drivers
    - Java DB (Embedded)
    - Java DB (Network)
    - MySQL (Connector/J driv
    - Oracle OCI
    - Oracle Thin
    - PostgreSQL
  - jdbc:derby://localhost:1527/sample [app on APP]
  - jdbc:mysql://localhost:3306/information_schema?zeroDateTimeBeha
  - jdbc:mysql://localhost:3306/mysql?zeroDateTimeBehavior=convert
  - jdbc:mysql://localhost:3306/personal?zeroDateTimeBehavior=conve
- Web Services
- Servers
- Maven Repositories
- Cloud
- Hudson Builders
- Docker
- Task Repositories
- JS Test Driver
- Selenium Server

Create Database...
Start
Stop
Connect
Disconnect
Delete
Refresh
Run Administration Tool
Properties...

Creating a database

Note: before creating a database, the MySQL server must be connected and running

Output - Java DB Database Process

```
Thu Aug 16 10:13:58 BST 2018 : Security manager installed using the Basic server security policy.
Thu Aug 16 10:13:59 BST 2018 : Apache Derby Network Server - 10.11.1.2 - (1629631) started and ready to accept connections on port
```

NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

Projects | Services | Files

- Databases
  - MySQL Server at localhost:3306 [root]
    - information_schema
    - mysql
    - newtest
    - performance_schema
    - phpmyadmin
    - test
  - Java DB
  - Drivers
    - Java DB (Embedded)
    - Java DB (Network)
    - MySQL (Connector/J driver)
    - Oracle OCI
    - Oracle Thin
    - PostgreSQL
  - jdbc:derby://localhost:1527/sample [app on APP]
  - jdbc:mysql://localhost:3306/information_schema?zeroDateTimeBeha
  - jdbc:mysql://localhost:3306/mysql?zeroDateTimeBehavior=convert
  - jdbc:mysql://localhost:3306/personal?zeroDateTimeBehavior=conve
- Web Services
- Servers
- Maven Repositories
- Cloud
- Hudson Builders
- Docker
- Task Repositories
- JS Test Driver
- Selenium Server

Create database name

Create MySQL Database

New Database Name: NewTest

Grant Full Access To: root@::1

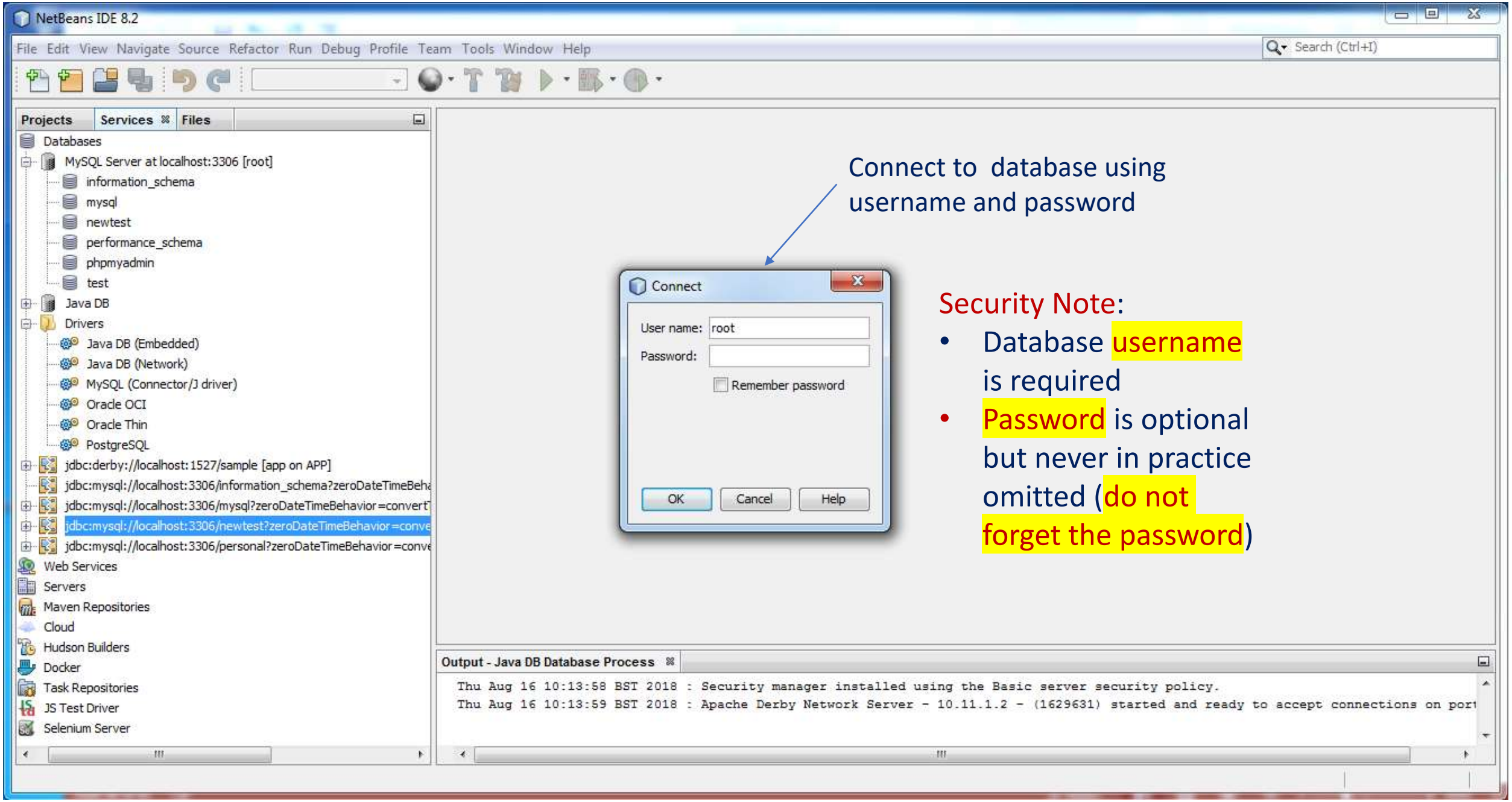
Creating the database...

OK Cancel

Create database user

Output - Java DB Database Process

```
Thu Aug 16 10:13:58 BST 2018 : Security manager installed using the Basic server security policy.
Thu Aug 16 10:13:59 BST 2018 : Apache Derby Network Server - 10.11.1.2 - (1629631) started and ready to accept connections on port
```



Connect to database using username and password

Security Note:

- Database username is required
- Password is optional but never in practice omitted (do not forget the password)

NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

Projects | Services | Files

- Databases
  - MySQL Server at localhost:3306 [root]
    - information_schema
    - mysql
    - newtest
    - performance_schema
    - phpmyadmin
    - test
  - Java DB
  - Drivers
    - Java DB (Embedded)
    - Java DB (Network)
    - MySQL (Connector/J driver)
    - Oracle OCI
    - Oracle Thin
    - PostgreSQL
  - jdbc:derby://localhost:1527/sample [app on APP]
  - jdbc:mysql://localhost:3306/information_schema?zeroDateTimeBeha
  - jdbc:mysql://localhost:3306/mysql?zeroDateTimeBehavior=convert
  - jdbc:mysql://localhost:3306/newtest?zeroDateTimeBehavior=conve
  - jdbc:mysql://localhost:3306/personal?zeroDateTimeBehavior=conve
- Web Services
- Servers
- Maven Repositories
- Cloud
- Hudson Builders
- Docker
- Task Repositories
- JS Test Driver
- Selenium Server

Database created and connection achieved

Output - Java DB Database Process

```
Thu Aug 16 10:13:58 BST 2018 : Security manager installed using the Basic server security policy.
Thu Aug 16 10:13:59 BST 2018 : Apache Derby Network Server - 10.11.1.2 - (1629631) started and ready to accept connections on port
```

NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

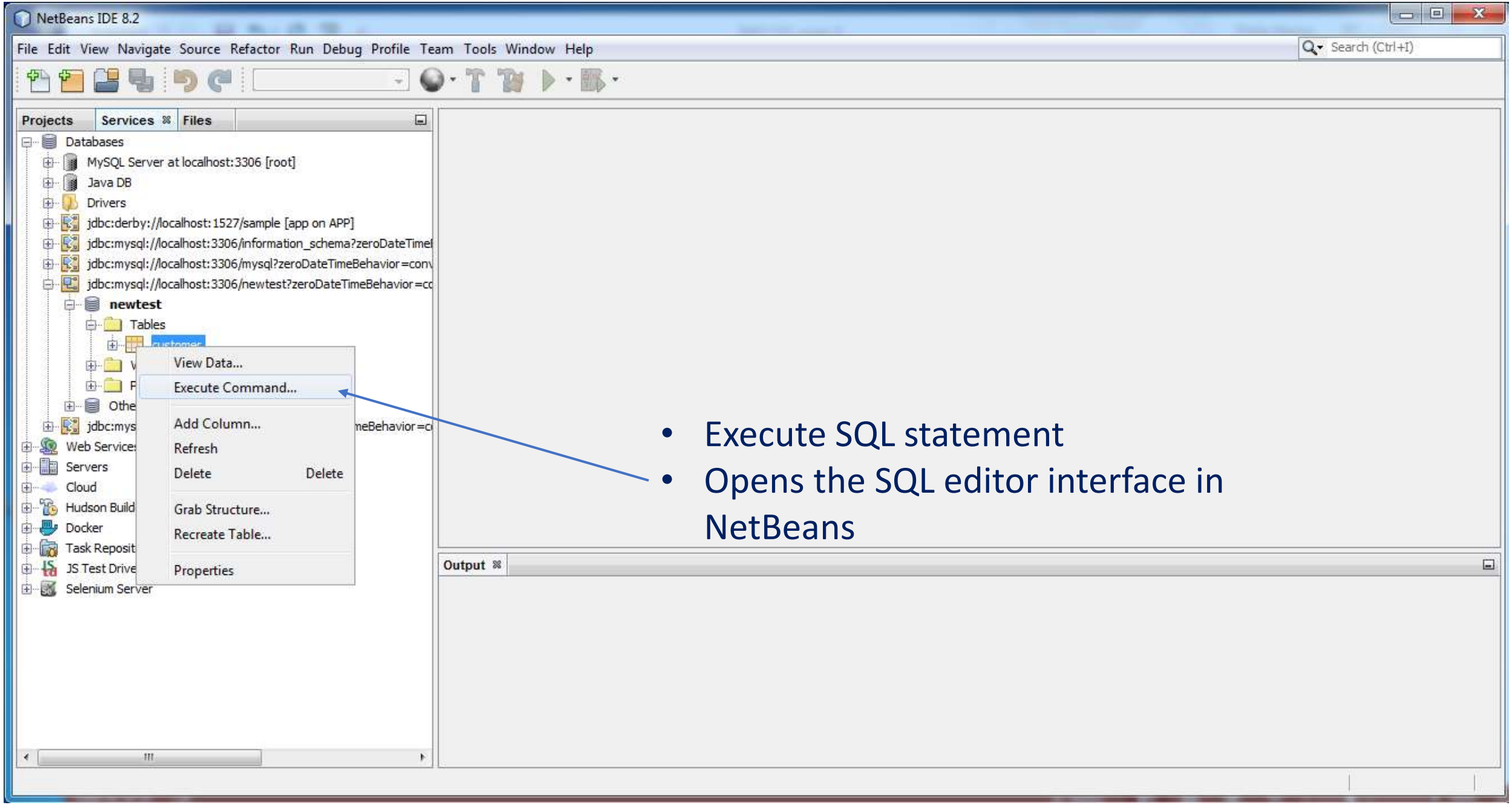
Projects | Services | Files

- Databases
  - MySQL Server at localhost:3306 [root]
  - Java DB
  - Drivers
  - jdbc:derby://localhost:1527/sample [app on APP]
  - jdbc:mysql://localhost:3306/information_schema?zeroDateTimeB
  - jdbc:mysql://localhost:3306/mysql?zeroDateTimeBehavior=conv
  - jdbc:mysql://localhost:3306/newtest?zeroDateTimeBehavior=co
    - **newtest**
      - Tables
        - customer
      - V
      - P
    - Othe
  - jdbc:mys ... meBehavior=co
- Web Services
- Servers
- Cloud
- Hudson Build
- Docker
- Task Reposit
- JS Test Drive
- Selenium Server

View Data...
Execute Command...
Add Column...
Refresh
Delete Delete
Grab Structure...
Recreate Table...
Properties

View data

Output



- Execute SQL statement
- Opens the SQL editor interface in NetBeans

NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Projects | Services | Files

- Databases
  - MySQL Server at localhost:3306 [root]
    - information_schema
    - mysql
    - newtest
    - performance_schema
    - phpmyadmin
    - test
  - Java DB
  - Drivers
    - Java DB (Embedded)
    - Java DB (Network)
    - MySQL (Connector/J driver)
    - Oracle OCI
    - Oracle Thin
    - PostgreSQL
  - jdbc:derby://localhost:1527/sample [app on APP]
  - jdbc:mysql://localhost:3306/information_schema?zeroDateTimeBeha
  - jdbc:mysql://localhost:3306/mysql?zeroDateTimeBehavior=convertT
  - jdbc:mysql:/localhost:3306/newtest?zeroDateTimeBehavior=conve
  - jdbc:mysql:/localhost:3306/personal?zeroDateTimeBehavior=conve
- Web Services
- Servers
- Maven Repositories
- Cloud
- Hudson Builders
- Docker
- Task Repositories
- JS Test Driver
- Selenium Server

SQL 1 [jdbc:mysql://localhost:33...]

Connection: jdbc:mysql://localhost:3306/newtest?zeroDateTimeBehavior=convertTo...

Enter SQL statement

Run SQL statement

- The NetBeans interface to write the SQL statement to create the required query (or view)
- Note: the database is empty – the data (records) must be inserted into the table(s)

Output - Java DB Database Process

```
Thu Aug 16 10:13:58 BST 2018 : Security manager installed using the Basic server security policy.
Thu Aug 16 10:13:59 BST 2018 : Apache Derby Network Server - 10.11.1.2 - (1629631) started and ready to accept connections on port
```

1:1 INS

SQL Statements

MySQL Server

- We have seen how to
  - *Connect* to and access a database and view data
  - *Create* a new MySQL database in NetBeans
- We now have a relational database ready to receive *actions*
- We can now:
  - Create tables
  - Populate the tables with attributes and data
  - Configure the tables with constraints
  - Set the primary key (and possibly a secondary key)
- To execute these actions we use *SQL statements*

Entering SQL Statements

- *Do not copy and paste the SQL statements into the editor*
- Copying and pasting may introduce unwanted formatting characters
  - The SQL statement may not work (*and you will wonder why?)*
- All the SQL statements (in both MySQL and PHP):
  - Must be typed directly into the editor
  - The syntax must be perfect
  - Any deviation from the syntax will result in failure
- Remember:
  - The database must be created
  - The MySQL server must be connected
  - The XAMPP server must be running (XAMPP and MySQL)

SQL to Create a Database Table

- The SQL shows the code to create a customer table
- The MySQL types shown are
  - char
  - varchar
  - Int
  - For other types see the course resources
- The (50) shows the length of the string allowed (50 characters)
- The int(4) specifies an int with 4 digits
- The PRIMARY KEY(cust_id) specifies the primary key for the customer table

```
CREATE TABLE customer (
cust_id int(5) NOT NULL,
surname varchar(50),
firstname varchar(50),
initial char(1),
title_id int(3),
address varchar(50),
city varchar(50),
state varchar(20),
zipcode varchar(10),
country_id int(4),
phone varchar(15),
birth_date char(10),
PRIMARY KEY (cust_id)
) ;
```

NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Projects | Services | Files

- MySQL (Connector/J driver)
- Oracle OCI
- Oracle Thin
- PostgreSQL
- jdbc:derby://localhost:1527/sample [app on APP]
- jdbc:mysql://localhost:3306/information_schema?zeroDateTimeE
- jdbc:mysql://localhost:3306/mysql?zeroDateTimeBehavior=conv
- jdbc:mysql://localhost:3306/newtest?zeroDateTimeBehavior=co
  - newtest
    - Tables
      - customer
        - cust_id
        - surname
        - firstname
        - initial
        - title_id
        - address
        - city
        - state
        - zipcode
        - country_id
        - phone
        - birth_date
        - Indexes
        - Foreign Keys
    - Views
    - Procedures
  - Other databases
- jdbc:mysql://localhost:3306/personal?zeroDateTimeBehavior=cc
- Web Services

SQL 1 [jdbc:mysql://localhost:33...] | SQL 2 [jdbc:mysql://localhost:33...]

Connection: jdbc:mysql://localhost:3306/newtest?zeroDateTimeBehavior=convertTo...

```
CREATE TABLE customer (
cust_id int(5) NOT NULL,
surname varchar(50),
firstname varchar(50),
initial char(1),
title_id int(3),
address varchar(50),
city varchar(50),
state varchar(20),
zipcode varchar(10),
country_id int(4),
phone varchar(15),
birth_date char(10),
PRIMARY KEY (cust_id)
);
```

Output

Java DB Database Process | SQL 2 execution

```
Executed successfully in 0.758 s.
no rows affected.
Line 1, column 1

Execution finished after 0.799 s, no errors occurred.
```

16:1 INS

NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Projects | Services | Files

- MySQL (Connector/J driver)
- Oracle OCI
- Oracle Thin
- PostgreSQL
- jdbc:derby://localhost:1527/sample [app on APP]
- jdbc:mysql://localhost:3306/information_schema?zeroDateTimeE
- jdbc:mysql://localhost:3306/mysql?zeroDateTimeBehavior=conv
- jdbc:mysql://localhost:3306/newtest?zeroDateTimeBehavior=co
  - **newtest**
    - Tables
      - customer
        - cust_id
        - surname
        - firstname
        - initial
        - title_id
        - address
        - city
        - state
        - zipcode
        - country_id
        - phone
        - birth_date
        - Indexes
        - Foreign Keys
    - Views
    - Procedures
  - Other databases
- jdbc:mysql://localhost:3306/personal?zeroDateTimeBehavior=co
- Web Services

SQL 1 [jdbc:mysql://localhost:33...] | SQL 2 [jdbc:mysql://localhost:33...] | SQL 3 [jdbc:mysql://localhost:33...]

Connection: jdbc:mysql://localhost:3306/newtest?zeroDateTimeBehavior=convertTo...

```
1   SELECT * FROM customer
```

SQL statement

Run SQL statement

SELECT * FROM customer

Max. rows: 100 | Fetched Rows: 0 | Matching Rows:

#	cust_id	surname	firstname	initial	title_id	address	city	state	zipcode	country_id	phone	birth_date

- database records
- The database is empty – the data (records) must be inserted into the table(s

SQL results (a 'view')

Output

Java DB Database Process | SQL 2 execution | SQL 3 execution

```
Executed successfully in 0.049 s.
Fetching resultset took 0 s.
Line 1, column 1

Execution finished after 0.1 s, no errors occurred.
```

Output execution

1:23 INS

SQL Statements

- The following slides demonstrate SQL statements:
  - INSERT INTO customer VALUES (1, 'Williams', 'Lucy', 'E', '2002-07-02’);
  - INSERT INTO customer VALUES (2, 'JOnes', 'Thomas', 'R', '1993-05-23’);
  - INSERT INTO customer VALUES (3, 'Thomas', 'Philip', 'M', '1997-11-17’);
  - SELECT * FROM customer LIMIT 100;
  - SELECT * FROM customer WHERE surname='Thomas’
  - UPDATE customer SET surname = 'Jones' WHERE cust_id = 2;
- These SQL statement demonstrate the INSERT / SELECT / UPDATE statements
  - The SELECT * FROM customer selects all the records in the table
- The other SQL statements follow this pattern
- Examples of all the SQL statements may be found in the course resources

NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Projects | Services | Files

- MySQL Server at localhost:3306 [root]
  - information_schema
  - mysql
  - newtest
  - performance_schema
  - phpmyadmin
  - test
- Java DB
- Drivers
- jdbc:derby://localhost:1527/sample [app on APP]
- jdbc:mysql://localhost:3306/information_schema?zeroDateTimeE
- jdbc:mysql://localhost:3306/mysql?zeroDateTimeBehavior=conv
- jdbc:mysql://localhost:3306/newtest?zeroDateTimeBehavior=co
  - **newtest**
    - Tables
      - customer
        - cust_id
        - surname
        - firstname
        - initial
        - birth_date
        - Indexes
        - Foreign Keys
    - Views
    - Procedures
  - Other databases
- jdbc:mysql://localhost:3306/personal?zeroDateTimeBehavior=cc
- Web Services
- Servers
- Cloud

SQL 1 | SQL 4 [jdbc:mysql://localhost:33...] | SQL 3 [jdbc:mysql://localhost:33...] | SQL 2 [jdbc:mysql://localhost:33...] | SQL 5 [jdbc:mysql://l...

Connection: jdbc:mysql://localhost:3306/newtest?zeroDateTimeBehavior=convertTo...

```
1  INSERT INTO customer VALUES (1, 'Williams', 'Lucy', 'E', '2002-07-02');
2
```

Output

SQL 5 execution | SQL 6 execution | SQL 7 execution | SQL 8 execution | SQL 9 execution | SQL 10 execution | SQL 11 execution | SQL 12 execution

```
Executed successfully in 0 s.
Fetching resultset took 0 s.
Line 1, column 1

Execution finished after 0.04 s, no errors occurred.
```

2:1 INS

NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Projects | Services | Files

- MySQL Server at localhost:3306 [root]
  - information_schema
  - mysql
  - newtest
  - performance_schema
  - phpmyadmin
  - test
- Java DB
- Drivers
- jdbc:derby://localhost:1527/sample [app on APP]
- jdbc:mysql://localhost:3306/information_schema?zeroDateTimeE
- jdbc:mysql://localhost:3306/mysql?zeroDateTimeBehavior=conv
- jdbc:mysql://localhost:3306/newtest?zeroDateTimeBehavior=co
  - **newtest**
    - Tables
      - customer
        - cust_id
        - surname
        - firstname
        - initial
        - birth_date
        - Indexes
        - Foreign Keys
    - Views
    - Procedures
  - Other databases
- jdbc:mysql://localhost:3306/personal?zeroDateTimeBehavior=cc
- Web Services
- Servers
- Cloud

SQL 1 | SQL 4 [jdbc:mysql://localhost:33...] | SQL 3 [jdbc:mysql://localhost:33...] | SQL 2 [jdbc:mysql://localhost:33...] | SQL 5 [jdbc:mysql://l...

Connection: jdbc:mysql://localhost:3306/newtest?zeroDateTimeBehavior=convertTo...

```
1  INSERT INTO customer VALUES (2, 'JOnes', 'Thomas', 'R', '1993-05-23');
```

Note:

- My deliberate error
- Corrected in a later update SQL statement

Output

SQL 5 execution | SQL 6 execution | SQL 7 execution | SQL 8 execution | SQL 9 execution | SQL 10 execution | SQL 11 execution | SQL 12 execution

```
Executed successfully in 0 s.
Fetching resultset took 0 s.
Line 1, column 1

Execution finished after 0.04 s, no errors occurred.
```

1:55 INS

NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Projects | Services | Files

- MySQL Server at localhost:3306 [root]
  - information_schema
  - mysql
  - newtest
  - performance_schema
  - phpmyadmin
  - test
- Java DB
- Drivers
- jdbc:derby://localhost:1527/sample [app on APP]
- jdbc:mysql://localhost:3306/information_schema?zeroDateTimeE
- jdbc:mysql://localhost:3306/mysql?zeroDateTimeBehavior=conv
- jdbc:mysql://localhost:3306/newtest?zeroDateTimeBehavior=co
  - **newtest**
    - Tables
      - customer
        - cust_id
        - surname
        - firstname
        - initial
        - birth_date
        - Indexes
        - Foreign Keys
    - Views
    - Procedures
  - Other databases
- jdbc:mysql://localhost:3306/personal?zeroDateTimeBehavior=cc
- Web Services
- Servers
- Cloud

...3... | SQL 5 [jdbc:mysql://localhost:33...] | SQL 6 [jdbc:mysql://localhost:33...] | SQL 7 [jdbc:mysql://localhost:33...] | SQL 8 [jdbc:mysql://localhost...

Connection: jdbc:mysql://localhost:3306/newtest?zeroDateTimeBehavior=convertTo...

```
1  INSERT INTO customer VALUES (3, 'Thomas', 'Philip', 'M', '1997-11-17');
2
```

Output

SQL 5 execution | SQL 6 execution | SQL 7 execution | SQL 8 execution | SQL 9 execution | SQL 10 execution | SQL 11 execution | SQL 12 execution

```
Executed successfully in 0 s.
Fetching resultset took 0 s.
Line 1, column 1

Execution finished after 0.04 s, no errors occurred.
```

2:1 INS

NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Projects | Services | Files

- MySQL Server at localhost:3306 [root]
  - information_schema
  - mysql
  - newtest
  - performance_schema
  - phpmyadmin
  - test
- Java DB
- Drivers
- jdbc:derby://localhost:1527/sample [app on APP]
- jdbc:mysql://localhost:3306/information_schema?zeroDateTimeE
- jdbc:mysql://localhost:3306/mysql?zeroDateTimeBehavior=conv
- jdbc:mysql://localhost:3306/newtest?zeroDateTimeBehavior=co
  - **newtest**
    - Tables
      - customer
        - cust_id
        - surname
        - firstname
        - initial
        - birth_date
        - Indexes
        - Foreign Keys
    - Views
    - Procedures
  - Other databases
- jdbc:mysql://localhost:3306/personal?zeroDateTimeBehavior=cc
- Web Services
- Servers
- Cloud

...3... | SQL 8 [jdbc:mysql://localhost:33...] | SQL 9 [jdbc:mysql://localhost:33...] | SQL 10 [jdbc:mysql://localhost:33...]

Connection: jdbc:mysql://localhost:3306/newtest?zeroDateTimeBehavior=convertTo...

```
1  SELECT * FROM customer LIMIT 100;
2
```

SELECT * FROM customer LI...

Max. rows: 100 | Fetched Rows: 3 | Matching Rows:

#	cust_id	surname	firstname	initial	birth_date
1	1	Williams	Lucy	E	2002-07-02
2	2	JOnes	Thomas	R	1993-05-23
3	3	Thomas	Philip	M	1997-11-17

Output

SQL 3 execution | SQL 2 execution | SQL 5 execution | SQL 6 execution | SQL 7 execution | SQL 8 execution | SQL 9 execution | SQL 10 execution

```
Executed successfully in 0 s.
Fetching resultset took 0 s.
Line 1, column 1

Execution finished after 0.11 s, no errors occurred.
```

1:1 INS

NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Projects | Services | Files

- MySQL Server at localhost:3306 [root]
  - information_schema
  - mysql
  - newtest
  - performance_schema
  - phpmyadmin
  - test
- Java DB
- Drivers
- jdbc:derby://localhost:1527/sample [app on APP]
- jdbc:mysql://localhost:3306/information_schema?zeroDateTimeE
- jdbc:mysql://localhost:3306/mysql?zeroDateTimeBehavior=conv
- jdbc:mysql://localhost:3306/newtest?zeroDateTimeBehavior=co
  - **newtest**
    - Tables
      - customer
        - cust_id
        - surname
        - firstname
        - initial
        - birth_date
        - Indexes
        - Foreign Keys
    - Views
    - Procedures
  - Other databases
- jdbc:mysql://localhost:3306/personal?zeroDateTimeBehavior=cc
- Web Services
- Servers
- Cloud

...3... | SQL 6 [jdbc:mysql://localhost:33...] | SQL 7 [jdbc:mysql://localhost:33...] | SQL 8 [jdbc:mysql://localhost:33...]

Connection: jdbc:mysql://localhost:3306/newtest?zeroDateTimeBehavior=convertTo...

```
1  SELECT * FROM customer WHERE surname='Thomas'
```

SELECT * FROM customer WH...

Max. rows: 100 | Fetched Rows: 1 | Matching Rows:

#	cust_id	surname	firstname	initial	birth_date
1	3	Thomas	Philip	M	1997-11-17

Output

SQL 4 execution | SQL 3 execution | SQL 2 execution | SQL 5 execution | SQL 6 execution | SQL 7 execution | SQL 8 execution

```
Executed successfully in 0.01 s.
Fetching resultset took 0 s.
Line 1, column 1

Execution finished after 0.06 s, no errors occurred.
```

1:46 INS

SQL `Update` Statements

MySQL UPDATE Statement in a PHP File

- We have seen how to:
  - Connect to a MySQL database
  - Use the SELECT SQL statement
  - Send the results (the output) to a web-page
- The following slides demonstrate how:
  - Connect to a MySQL database
  - Use the INSERT SQL statement
  - Send the results (the output) to a web-page
  - Use the SELECT SQL statement to check the new MySQL database record

The UPDATE SQL Statement

- To demonstrate the SQL syntax and working with a MySQL database a simple single table test database was created
- The test database has a contacts table with 5 rows (or records)
- Each record has 4 attributes:
  - `id ( int(5) )`
  - `f_name ( varchar (50) )`
  - `s_name ( varchar(50) )`
  - `email ( varchar(50) )`
- The initial table was populated with a for all records as shown in the previous slides
- The following slides show the UPDATE SQL statement and the result

SQL Statements

- SQL update statement:

```
UPDATE customer SET surname = 'Jones' WHERE cust_id = 2;
```

- The SQL statement corrects the error:
  - The text “JOnes” is corrected to “Jones”
- The SQL UPDATE SQL statement is shown in the following slides with the output in the NetBeans IDE

NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Projects | Services | Files

- MySQL Server at localhost:3306 [root]
  - information_schema
  - mysql
  - newtest
  - performance_schema
  - phpmyadmin
  - test
- Java DB
- Drivers
- jdbc:derby://localhost:1527/sample [app on APP]
- jdbc:mysql://localhost:3306/information_schema?zeroDateTimeE
- jdbc:mysql://localhost:3306/mysql?zeroDateTimeBehavior=conv
- jdbc:mysql://localhost:3306/newtest?zeroDateTimeBehavior=co
  - **newtest**
    - Tables
      - customer
        - cust_id
        - surname
        - firstname
        - initial
        - birth_date
        - Indexes
        - Foreign Keys
    - Views
    - Procedures
  - Other databases
- jdbc:mysql://localhost:3306/personal?zeroDateTimeBehavior=cc
- Web Services
- Servers
- Cloud

...3...] | SQL 9 [jdbc:mysql://localhost:33...] | SQL 10 [jdbc:mysql://localhost:33...] | SQL 11 [jdbc:mysql://localhost:33...]

Connection: jdbc:mysql://localhost:3306/newtest?zeroDateTimeBehavior=convertTo...

```
1   UPDATE customer SET surname = 'Jones' WHERE cust_id = 2;
```

UPDATE surname "Jones"

Output

SQL 2 execution | SQL 5 execution | SQL 6 execution | SQL 7 execution | SQL 8 execution | SQL 9 execution | SQL 10 execution | SQL 11 execution

```
Executed successfully in 0.03 s.
1 row affected.
Line 1, column 1

Execution finished after 0.04 s, no errors occurred.
```

1:38 INS

NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Projects | Services | Files

- MySQL Server at localhost:3306 [root]
  - information_schema
  - mysql
  - newtest
  - performance_schema
  - phpmyadmin
  - test
- Java DB
- Drivers
- jdbc:derby://localhost:1527/sample [app on APP]
- jdbc:mysql://localhost:3306/information_schema?zeroDateTimeE
- jdbc:mysql://localhost:3306/mysql?zeroDateTimeBehavior=conv
- jdbc:mysql://localhost:3306/newtest?zeroDateTimeBehavior=co
  - **newtest**
    - Tables
      - customer
        - cust_id
        - surname
        - firstname
        - initial
        - birth_date
        - Indexes
        - Foreign Keys
    - Views
    - Procedures
  - Other databases
- jdbc:mysql://localhost:3306/personal?zeroDateTimeBehavior=cc
- Web Services
- Servers
- Cloud

...3... | SQL 10 [jdbc:mysql://localhost:33...] | SQL 11 [jdbc:mysql://localhost:33...] | SQL 12 [jdbc:mysql://localhost:33...]

Connection: jdbc:mysql://localhost:3306/newtest?zeroDateTimeBehavior=convertTo...

```
1   SELECT * FROM customer LIMIT 100;
```

UPDATE surname "Jones"

SELECT * FROM customer LI...

Max. rows: 100 | Fetched Rows: 3 | Matching Rows:

#	cust_id	surname	firstname	initial	birth_date
1	1	Williams	Lucy	E	2002-07-02
2	2	Jones	Thomas	R	1993-05-23
3	3	Thomas	Philip	M	1997-11-17

Output

SQL 5 execution | SQL 6 execution | SQL 7 execution | SQL 8 execution | SQL 9 execution | SQL 10 execution | SQL 11 execution | SQL 12 execution

```
Executed successfully in 0 s.
Fetching resultset took 0 s.
Line 1, column 1

Execution finished after 0.04 s, no errors occurred.
```

1:34 INS

NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Connection: jdbc:mysql://localhost:3306/test?zeroDateTimeBehavior=convertToNull [p...

```
1  UPDATE contacts SET id = '3' WHERE f_name = 'Fred'
2
```

UPDATE id (3)

SELECT * FROM contacts LI...

Max. rows: 100 Fetched Rows: 5 Matching Rows:

#	id	f_name	s_name	email
1	1	Philip	Jones	pj@gmail.com
2	2	Tom	Mills	tm@outlook.com
3	1	Fred	Bloggs	fb@gmail.com
4	1	Alan	Smith	as@icloud.com
5	1	Steven	Nolan	sn@aol.com

Running an SQL Query a Web-Based System

SQL in Web-Based Systems

- Executing SQL in MySQL sever (in NetBeans) is interesting
  - There are cases where the data output within the MySQL server system is all that is required by an organization
  - However: the data would generally be presented to users over an *intranet* (an internal network)
- In this course we are working within web-systems using web services
  - We need a way to present the results of SQL queries in web-based systems
  - To achieve this we must integrate SQL queries into web programs
  - To do this we integrate SQL statements in PHP scripts in an HTML file

PHP MySQL Script

- We have seen how to access the MySQL database using NetBeans
- The following slides show a PHP script as used to access a MySQL database and process data
- It is a simple implementation of a database connection and access to the ‘test’ database with the results sent to a web browser
- From the slides we can see:
  - The PHP script embedded in a HTML file (index.php)
  - The SQL statements (in the correct PHP SQL format)
  - The output sent to the NetBeans IDE and a web browser

SQL **SELECT**

$sql

$sqla

NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Projects | Services | Files

- Databases
  - MySQL Server at localhost:3306 [root]
  - Java DB
  - Drivers
  - jdbc:derby://localhost:1527/sample [app on APP]
  - jdbc:mysql://localhost:3306/information_schema?zeroDa
  - jdbc:mysql://localhost:3306/mysql?zeroDateTimeBehavic
  - jdbc:mysql://localhost:3306/personal?zeroDateTimeBeha
  - jdbc:mysql://localhost:3306/test?zeroDateTimeBehavior
    - test
      - Tables
        - contacts
          - id
          - f_name
          - s_name
          - email
        - Indexes
        - Foreign Keys
      - Views
      - Procedures

The database used in the example PHP script

Navigator

- html
  - head
    - meta
    - title
  - body
    - pre

Filters:

index.php

Source | History

```
15  <?php
16  $dbhost = 'localhost:3306'; $dbuser = 'philip';
17  $dbpass = 'philip'; $db = 'test';
18  //SQL statements: $sql and $sqla
19  $sql = 'SELECT id, f_name, s_name, email FROM contacts';
20  $sqla = 'SELECT * FROM contacts';
21  //connect MySQL database
22  $conn = mysqli_connect($dbhost, $dbuser, $dbpass);
23  if(!$conn) {
24      Die_error('Connection failed!: '.mysqli_errno());
25  }
26  echo 'Database connected <br>';
27  mysqli_select_db($conn, $db);
28  echo 'database selected: ' .$db.'<br>';
29  echo ($sqla.'<br>'.'------------------<br>');
30  $retval = mysqli_query($conn, $sqla);
31  if(!$retval) {
32      die('Could not get data');
33  }
34  while($row = mysqli_fetch_assoc($retval)) {
35      echo "id: {$row['id']}<br>".
36           "f_name: {$row['f_name']}<br>".
37           "s_name: {$row['s_name']}<br>".
38           "email:  {$row['email']}<br>".
39           "-----------------------<br>";
40  }
41  mysqli_free_result($retval);
42  echo "fetched data successfully\n";
43  mysqli_close($conn);
44  ?>
```

36:50 INS

MySQL_Conn_Test - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Projects | Services | Files

- MySQL_Conn_Test
  - Source Files
    - index.php
  - Include Path
    - C:\xampp
- MySQL_INSERT_Test

Navigator

- html
  - head
    - meta
    - title
  - body
    - pre

index.php

Source | History

```php
<?php
$dbhost = 'localhost:3306'; $dbuser = 'philip';
$dbpass = 'philip'; $db = 'test';
//SQL statements: $sql and $sqla
$sql = 'SELECT id, f_name, s_name, email FROM contacts';
$sqla = 'SELECT * FROM contacts';
//connect MySQL database
$conn = mysqli_connect($dbhost, $dbuser, $dbpass);
if(!$conn) {
    Die_error('Connection failed!: '.mysqli_errno());
}
echo 'Database connected <br>';
mysqli_select_db($conn, $db);
echo 'database selected: ' .$db.'<br>';
echo ($sqla.'<br>'.'------------------<br>');
$retval = mysqli_query($conn, $sqla);
if(!$retval) {
    die('Could not get data');
}
while($row = mysqli_fetch_assoc($retval)) {
    echo "id: {$row['id']}<br>".
         "f_name: {$row['f_name']}<br>".
         "s_name: {$row['s_name']}<br>".
         "email:  {$row['email']}<br>".
         "-----------------------<br>";
}
mysqli_free_result($retval);
echo "fetched data successfully\n";
mysqli_close($conn);
?>
```

- SQL statements $sql and $sqla
- The SQL statements are assigned to a variable
- Multiple statements can be created and stored in the PHP script
- SQL statements can also be held in a separate file which the program calls

Connect MySQL Databas

localhost/MySQL_Conn_Test/index.php

```
        Database connected
database selected: test
SELECT id, f_name, s_name, email FROM contacts
------------------
id: 1
f_name: Philip
s_name: Jones
email:  pj@gmail.com
-----------------------
id: 2
f_name: Tom
s_name: Mills
email:  tm@outlook.com
-----------------------
id: 3
f_name: Fred
s_name: Bloggs
email:  fb@gmail.com
-----------------------
id: 4
f_name: Alan
s_name: Smith
email:  as@icloud.com
-----------------------
id: 5
f_name: Steven
s_name: Nolan
email:  sn@aol.com
-----------------------
fetched data successfully
```

SQL statement used

$sql

Connect MySQL Databas

localhost/MySQL_Conn_Test/index.php

Apps Bookmarks Outlook Google Google Scholar Google Sites BBC News Amazon Kindle Amazon CC EDAS EasyChair Other bookmarks

```
        Database connected
database selected: test
SELECT * FROM contacts
-----------------
id: 1
f_name: Philip
s_name: Jones
email:  pj@gmail.com
----------------------
id: 2
f_name: Tom
s_name: Mills
email:  tm@outlook.com
----------------------
id: 3
f_name: Fred
s_name: Bloggs
email:  fb@gmail.com
----------------------
id: 4
f_name: Alan
s_name: Smith
email:  as@icloud.com
----------------------
id: 5
f_name: Steven
s_name: Nolan
email:  sn@aol.com
----------------------
fetched data successfully
```

SQL statement used

$sqla

index.php | Output - Browser Log | SQL 1 [jdbc:mysql://localhost:33...] | SQL 2 [jdbc:mysql://localhost:33...]

Connection: jdbc:mysql://localhost:3306/test?zeroDateTimeBehavior=convertToNull [p...

```
1  SELECT * FROM contacts LIMIT 100;
2
```

- SQL statement with a constraint
- The SELECT statement will restrict the 'view' to a maximum of 100 records (rows in the database table)
- Confirms the web page data output

SELECT * FROM contacts LI...

Max. rows: 100 | Fetched Rows: 5 | Matching Rows:

#	id	f_name	s_name	email
1	1	Philip	Jones	pj@gmail.com
2	2	Tom	Mills	tm@outlook.com
3	3	Fred	Bloggs	fb@gmail.com
4	4	Alan	Smith	as@icloud.com
5	5	Steven	Nolan	sn@aol.com

1:1 INS

INSERT

$sqli

NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Projects | Services | Files

- Databases
  - MySQL Server at localhost:3306 [root]
  - Java DB
  - Drivers
  - jdbc:derby://localhost:1527/sample [app on APP]
  - jdbc:mysql://localhost:3306/information_schema?zeroDateTi
  - jdbc:mysql://localhost:3306/mysql?zeroDateTimeBehavior=c
  - jdbc:mysql://localhost:3306/personal?zeroDateTimeBehavior
  - jdbc:mysql://localhost:3306/test?zeroDateTimeBehavior=co
    - test
      - Tables
        - contacts
          - id
          - f_name
          - s_name
          - email
          - Indexes
          - Foreign Keys
      - Views
      - Procedures

Navigator

- html
  - head
    - meta
    - title
  - body

Output - SQL 1 execution | index.php | SQL 1 [jdbc:mysql://localhost:33...]

Connection: jdbc:mysql://localhost:3306/test?zeroDateTimeBehavior=convertToNull [p...

```
1  SELECT * FROM contacts;
2
```

- SQL statement
- The database records **before** the **INSERT INTO** SQL statement for Jane Fox

SELECT * FROM contacts

Max. rows: 100 | Fetched Rows: 6 | Matching Rows:

#	id	f_name	s_name	email
1	1	Philip	Jones	pj@gmail.com
2	2	Tom	Mills	tm@outlook.com
3	3	Fred	Bloggs	fb@gmail.com
4	4	Alan	Smith	as@icloud.com
5	5	Steven	Nolan	sn@aol.com
6	6	John	Smith	js@icloud.com

2:1 INS

```
10      </head>
11      <body>
12          <?php
13          $dbhost = 'localhost:3306'; $dbuser = 'philip';
14          $dbpass = 'philip'; $db = 'test';
15          //SQL statement: $sqli
16          $sqli = "INSERT INTO contacts (id, f_name, s_name, email) "
17                  . "VALUES (7, 'Jane', 'Fox', 'jf@gmail.com')";
18          //connect MySQL database
19          $conn = mysqli_connect($dbhost, $dbuser, $dbpass);
20          if(!$conn) {
21              Die_error('Connection failed!: '.mysqli_errno());
22          }
23          //confirm database connection
24          echo 'Database connected <br>';
25          mysqli_select_db($conn, $db);
26          echo 'database selected: ' .$db.'<br>';
27          echo ($sqli.'<br>'.'------------------<br>');
28          // Attempt INSERT query execution
29          if(mysqli_query($conn, $sqli)) {
30              echo 'Records inserted successfully'.'<br>';
31          } else{
32          echo "ERROR: Could not execute $sqli." . mysqli_error($conn);
33          }
34          echo "INSERT successful\n";
35          mysqli_close($conn);
36          ?>
37      </body>
38  </html>
```

- We now have **3** SQL statements: $sql $sqla $sqli
- SQL statement $sqli is used in this PHP script
- The SQL statement ($sqli) is the INSERT INTO statement to add a new user record to the test database

MySQL INSERT Statemen

localhost/MySQL_INSERT_Test/index.php

Apps Bookmarks Outlook Google Google Scholar Google Sites BBC News Amazon Kindle Amazon CC EDAS EasyChair Other bookmarks

Database connected
database selected: test
INSERT INTO contacts (id, f_name, s_name, email) VALUES (7, 'Jane', 'Fox', 'jf@gmail.com')

Records inserted successfully
INSERT successful

SQL statement used
$sqli

Connect MySQL Databas

localhost/MySQL_Conn_Test/index.php

Apps Bookmarks Outlook Google Google Scholar Google Sites BBC News Amazon Kindle Amazon CC EDAS EasyChair Other bookmarks

```
        Database connected
database selected: test
SELECT * FROM contacts
------------------
id: 1
f_name: Philip
s_name: Jones
email:  pj@gmail.com
-----------------------
id: 2
f_name: Tom
s_name: Mills
email:  tm@outlook.com
-----------------------
id: 3
f_name: Fred
s_name: Bloggs
email:  fb@gmail.com
-----------------------
id: 4
f_name: Alan
s_name: Smith
email:  as@icloud.com
-----------------------
id: 5
f_name: Steven
s_name: Nolan
email:  sn@aol.com
-----------------------
id: 6
f_name: John
s_name: Smith
email:  js@icloud.com
-----------------------
id: 7
f_name: Jane
s_name: Fox
email:  jf@gmail.com
-----------------------
fetched data successfully
```

INSERT record added

NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Projects | Services | Files

- Databases
  - MySQL Server at localhost:3306 [root]
  - Java DB
  - Drivers
  - jdbc:derby://localhost:1527/sample [app on APP]
  - jdbc:mysql://localhost:3306/information_schema?zeroDateTi
  - jdbc:mysql://localhost:3306/mysql?zeroDateTimeBehavior=c
  - jdbc:mysql://localhost:3306/personal?zeroDateTimeBehavior
  - jdbc:mysql://localhost:3306/test?zeroDateTimeBehavior=co
    - test
      - Tables
        - contacts
          - id
          - f_name
          - s_name
          - email
          - Indexes
          - Foreign Keys
      - Views
      - Procedures

Navigator

- html
  - head
    - meta
    - title
  - body
    - pre

Output | index.php | SQL 1 [jdbc:mysql://localhost:33...]

Connection: jdbc:mysql://localhost:3306/test?zeroDateTimeBehavior=convertToNull [p...

```
1  SELECT * FROM contacts;
2
3
```

- SQL statement
- The output data (the 'view') **following** the **UPDATE INTO** SQL statement for Jane Fox

SELECT * FROM contacts

Max. rows: 100 | Fetched Rows: 7 | Matching Rows:

#	id	f_name	s_name	email
1	1	Philip	Jones	pj@gmail.com
2	2	Tom	Mills	tm@outlook.com
3	3	Fred	Bloggs	fb@gmail.com
4	4	Alan	Smith	as@icloud.com
5	5	Steven	Nolan	sn@aol.com
6	6	John	Smith	js@icloud.com
7	7	Jane	Fox	jf@gmail.com

2:1 INS

Improve the PHP MySQL Script

Improve the PHP Script

- The PHP script as used is a simple implementation of a database connection and access to my test database with the results sent to a web browser
- In practice the script would be improved by:
  - Using try…catch…finally blocks
  - These would be applied to (at least) lines 23 to 25 and 31 to 33 (see slide 30
- The motivation for using try…catch…finally blocks is:
  - To catch errors in connecting to and accessing the database
  - To catch attempted criminal access to the database
  - To catch any errors (possibly using regular expressions) in user input or structured SQL statements

Review

- In this tutorial we have shown how to:
  - Connect to a MySQL server
  - Connect to a database using the NetBeans IDE
  - Viewing data using the NetBeans IDE
  - Run an SQL Query a Web-Based System with the output in a web page
- The following slide sets the practical exercises to be completed:
  - Create and populate your own MySQL database
  - Run the original PHP script (with your own MySQL database
  - Improve the PHP script using `try…catch…finally` blocks
  - See the instruction in slides 63 and 64

Practical Exercises

Practical Exercise (1)

- Create your personal account in the MySQL Administration Tool:
  - Username
  - Password
  - Global privileges (in this case you will be the SysAdmin
- Create a database in NetBeans
  - Call it "test"
- Create a "customer" table with four attributes (columns) and four records (rows)
- Populate the table with data values
- Run your database with SQL queries using the PHP script

Practical Exercise (2)

- The PHP script as used and shown in the previous slide:
  - Is a simple solution to implement a connection to a MySQL (or other relational database systems) and run SQL queries on the database
- In practice the script would be improved using:
  - **`try`**…**`catch`**…**`finally`** block
  - Improve the PHP script shown by adding **`try`**…**`catch`**…**`finally`** blocks to:
    - The database connection: if(!$conn) {…} (with the error message)
    - The data retrieval : if(!retrieval) {…} (with the error message)
    - The closing statements (lines 40 – 43 in slide 64)

Source History

```php
<?php
$dbhost = 'localhost:3306'; $dbuser = 'philip';
$dbpass = 'philip'; $db = 'test';
//SQL statements: $sql and $sqla
$sql = 'SELECT id, f_name, s_name, email FROM contacts';
$sqla = 'SELECT * FROM contacts';
//connect MySQL database
$conn = mysqli_connect($dbhost, $dbuser, $dbpass);
if(!$conn) {
    Die_error('Connection failed!: '.mysqli_errno());
}
echo 'Database connected <br>';
mysqli_select_db($conn, $db);
echo 'database selected: ' .$db.'<br>';
echo ($sqla.'<br>'.'-------------------<br>');
$retval = mysqli_query($conn, $sqla);
if(!$retval) {
    die('Could not get data');
}
while($row = mysqli_fetch_assoc($retval)) {
    echo "id: {$row['id']}<br>".
         "f_name: {$row['f_name']}<br>".
         "s_name: {$row['s_name']}<br>".
         "email:  {$row['email']}<br>".
         "------------------------<br>";
}
mysqli_free_result($retval);
echo "fetched data successfully\n";
mysqli_close($conn);
?>
```

The test database used in the example PHP script

SQL statements $sql / $sqla

SQL statement used $sqla

PHP closing statements

Practical SQL Exercise (3)

- Implement the following SQL statements
- The following SQL statements to be written into your PHP script and run on the test database:
  - INSERT
  - ADD
  - SELECT (all values and selected values)
  - UPDATE records
  - REPLACE INTO table_name (column list) VALUES (column values);
  - DELETE FROM table_name [WHERE some_condition_is_true] [LIMIT rows]