






SQL for applications

PHP and MySQL exercises
Execution

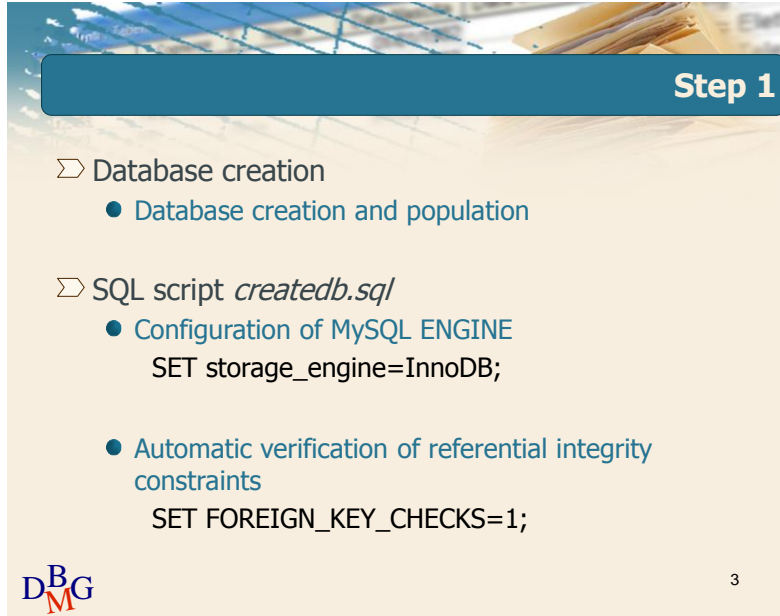


Resolution steps

- ▷ Step 1
 - Database creation and population
 - Creation of an SQL script
- ▷ Step 2
 - Creation of a select query
 - Creation of an update query
- ▷ Step 3
 - Creation of a static web page (client-side) with querying form
- ▷ Step 4
 - Insertion of the PHP script for dynamic interfacing to DB (server-side)




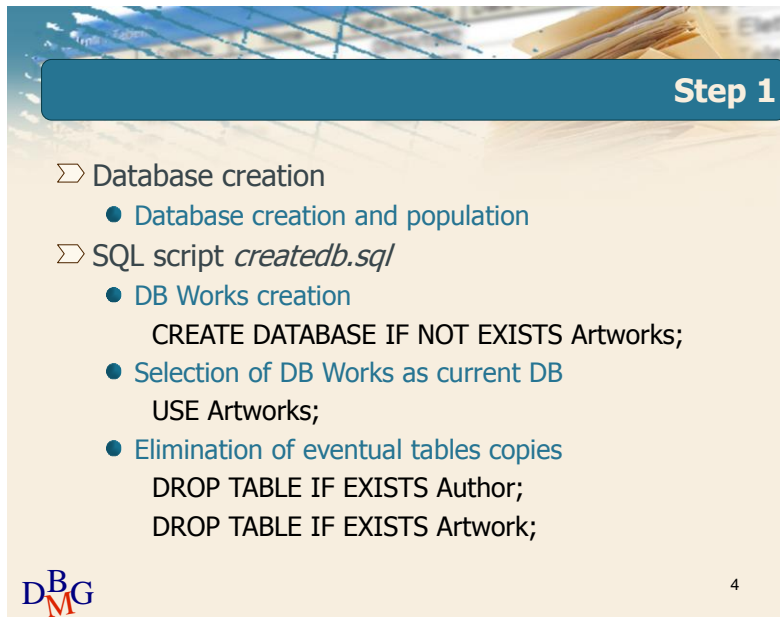
2



Step 1


- ▷ Database creation
 - Database creation and population
- ▷ SQL script *createdb.sql*
 - Configuration of MySQL ENGINE
SET storage_engine=InnoDB;
 - Automatic verification of referential integrity constraints
SET FOREIGN_KEY_CHECKS=1;

 3



Step 1

- ▷ Database creation
 - Database creation and population
- ▷ SQL script *createdb.sql*
 - DB Works creation
CREATE DATABASE IF NOT EXISTS Artworks;
 - Selection of DB Works as current DB
USE Artworks;
 - Elimination of eventual tables copies
DROP TABLE IF EXISTS Author;
DROP TABLE IF EXISTS Artwork;

 4

Step 1

- ▷ Database creation

- Database creation and population

- ▷ SQL script *createdb.sql*

- Disabling of auto-commit for each instruction

```
SET AUTOCOMMIT = 0;
```

- Creation of transaction of create tables

```
START TRANSACTION;
Tables creation instructions
COMMIT;
```

- Creation of transaction of tables population

```
START TRANSACTION;
Tables population instructions
COMMIT;
```



5

Step 1

- ▷ Database creation

- Database creation and population

```
SET storage_engine=InnoDB;
SET FOREIGN_KEY_CHECKS=1;
CREATE DATABASE IF NOT EXISTS Artwork;
USE Artwork;
DROP TABLE IF EXISTS Author;
DROP TABLE IF EXISTS Artwork;
SET AUTOCOMMIT=0;
START TRANSACTION;
... table creation instructions...
COMMIT;
START TRANSACTION;
... table population instructions...
COMMIT;
```



6

Step 1

➤ Database creation

- Table creation instructions
 - Consistency with the logical schema
 - Selection of data types
 - Constraints specification



7

Step 1

➤ Database creation

- Table creation instructions

```
CREATE TABLE Author (
  `authorCode` varchar(10) PRIMARY KEY,
  `name` varchar(30) NOT NULL,
  `surname` varchar(30) NOT NULL,
  `birthYear` integer NOT NULL,
  `birthCity` varchar(20) NOT NULL
);
CREATE TABLE Artwork (
  `artworkCode` varchar(10) PRIMARY KEY,
  `name` varchar(30) NOT NULL,
  `category` varchar(20) NOT NULL,
  `city` varchar(20) NOT NULL,
  `country` varchar(20) NOT NULL,
  `author` varchar(10) NOT NULL REFERENCES Author(authorCode)
ON DELETE CASCADE;
);
```



8

Step 1

Database creation

● Tables population instructions

```
INSERT INTO Author VALUES ('1', 'Gian Lorenzo', 'Bernini',
1598, 'Napoli');
```

```
INSERT INTO Author VALUES ('2', 'Francesco', 'Borromini',
1599, 'Bissone');
```

```
INSERT INTO Artwork VALUES ('1', 'Apollo e Dafne',
'sculpture', 'Roma', 'Italia', '1');
```

```
INSERT INTO Artwork VALUES ('2', 'Baldacchino S.Pietro',
'architecture', 'Roma', 'Italia', '1');
```

```
INSERT INTO Artwork VALUES ('3', 'Fontana dei fiumi',
'architecture', 'Roma', 'Italia', '1');
```

```
INSERT INTO Artwork VALUES ('4', 'S.Ivo la Sapienza',
'architecture', 'Roma', 'Italia', '2');
```



9

Step 2

Interrogation query creation

- **Result to show:** name and category of artworks, author's surname
- **Constraints:** artwork's city, author's birth date
- **Order:** author's surname and artwork's name

```
SELECT author.surname AS author, artwork.name AS artwork,
category
```

```
FROM author, artwork
```

```
WHERE author.artworkCode = artwork.author
```

```
AND birthYear >= 1590
```

```
AND birthYear <= 1600
```

```
AND artwork.city = 'Roma'
```

```
ORDER BY surname, artwork
```



10

Step 3

➤ Access form to DB creation

- HTML page writing with editor text (e.g. notepad, HAPedit)

Set author's year of birth and interested city

Year of birth
between
year
and
year
city

Form contains:

- Two text fields (start and end year of birth)
- A drop down menu (artwork's city)
- A submit button
- A reset button



13

Step 3

➤ Static Webpage creation

- Method: GET
- Action: *test.php* (see Step 4)

```

1 <html>
2 <head>
3 <title>Artworks</title>
4 </head>
5 <body>
6 <h4> Set a range for author's birth year and the city </h4>
7
8 <form method="get" action="test.php">
9
10

```



14

Step 3

Creation of a static Webpage

```

10 <table>
11
12
13 <tr> Birth date between </tr>
14
15 <tr> <td> year </td>
16 <td> <input type="text" size="4" maxlength="4" name="year1"> </td> </tr>
17
18 <tr> <td> and year </td>
19 <td> <input type="text" size="4" maxlength="4" name="year2"> </td> </tr>
20
21 <tr> <td> City: </td>
22
23 <td>
24 <select name="city">
25 <option value="Torino">Torino</option>
26 <option value="Milano">Milano</option>
27 <option value="Roma">Roma</option>
28 <option value="Napoli">Napoli</option>
29 </select>
30 </td> </tr>
31
32 </table>
33

```

Step 3

Creation of a static Webpage

```

34
35 <br>
36
37
38 <input type="reset" value="Cancel">
39 <input type="submit" value="Send">
40
41 </form>
42
43
44 </body>
45 </html>
46

```

The **"submit"** triggers the transmission of parameters that will be processed by the application server for the creation of the dynamic content

Step 4

- ▷ Creation of a dynamic Webpage
 - Dynamic creation of the response page in function of the DB content

- ▷ Page *test.PHP* called by the "submit" action of the form at step 3
 - Contains HTML static code for structuring the page content
 - Contains PHP code to generate the dynamic content



17

Step 4

- ▷ Creation of a dynamic Webpage
 - Static HTML page to display the results (*test.php*)

```
<html>
<head>
<title>Results</title>
</head>
<body>
...
</body>
</html>
```

In the **BODY** we insert the **PHP code** needed for the creation of the dynamic content

```
<?php
...
?>
```

The **PHP code** will be processed at server-side



18

Step 4

▷ Creation of a dynamic Webpage

- PHP code

▷ The PHP code executes the following steps

1. Reading of parameters from the form and verification of correctness and consistence
2. Storing the values of parameters in appropriate variables
3. Construction of the query using the aforementioned variables
4. Connection to the DB
5. Execution of the query
6. Generation of HTML code with the results (visible client-side through Web browser)
7. Closing the connection to the DB



19

Step 4

▷ Creation of a dynamic Webpage

- Reading and verification of parameters
 - Error signaling for missing or incostistent data (i.e. *end year < start year*)
 - Interrupt the execution flow

```
if( !isset($_REQUEST["year1"]) or
    !isset($_REQUEST["year2"]) or
    !isset($_REQUEST["city"])){
    die("Error: insert all requested data");
}
```

```
if( !is_numeric($_REQUEST["year1"]) or
    !is_numeric($_REQUEST["year2"]) or ($_REQUEST["year1"] >
    $_REQUEST["year2"])){
    die("Error: year values not correct");}
```



20

Step 4

▷ Creation of a dynamic Webpage

- Assignment of values to variables
 - variables will be useful to «compose» the query to send to the server
 - a variable for each constraint/input parameter
 - start year
 - end year
 - artwork city

```
$a1 = $_REQUEST["year1"];
$a2 = $_REQUEST["year2"];
$c = $_REQUEST["city"];
```



21

Step 4

▷ Creation of a dynamic Webpage

- Composition of the query
 - Storing of the support variable \$sql of the query built with parameters specified by the user

```
$sql = " SELECT surname AS author, Artwork.name AS artwork,
category
FROM Author, Artwork
WHERE Author.authorCode = Artwork.author
      AND year >= $a1
      AND year <= $a2
      AND artwork.city = '$c'
ORDER BY surname, artwork";
```



22

Step 4

▷ Creation of a dynamic Webpage

- Connection to the DB (immediate mode)
 - Connection to MySQL through the user joe (pwd: xyz)
 - *Note:* the user must exist in mySQL (create it through the Web interface)
 - Selection of the DB *Artworks* previously created

```
$con = mysqli_connect('localhost','joe','xyz','artworks');
if (mysqli_connect_errno())
{
  die ('Failed to connect to MySQL: ' . mysqli_connect_error());
}
```



23

Step 4

▷ Creation of a dynamic Webpage

- Execution of the query
 - The *result* variable allows to store the returned results and verify possible errors

```
$result = mysqli_query($con,$sql);
```

```
if( !$result )
  die('Query error: ' . mysqli_error($con))
```



24

Step 4

▷ Creation of a dynamic Webpage

- Generation of results
 - Reading of the content of *result* variable and generation of HTML code for the browser client to display the final result



25

Step 4

```

58
59 echo "<h1> Artworks in %c of authors born between $a1 and $a2 </h1>";
60
61
62 if( mysqli_num_rows($result) > 0 ){
63
64     echo "<table border=1 cellpadding=10>";
65
66     echo "<tr>";
67
68     for ($i = 0; $i < mysqli_num_fields($result); $i++){
69         $title = mysqli_fetch_field($result);
70         $name = $title->name;
71         echo "<th> $name </th>";
72     }
73
74     echo "</tr>";
75
76     while ($row = mysqli_fetch_row($result)) {
77
78         switch($row[2]){
79
80             case "architecture":
81                 $color = "yellow";
82                 break;
83             case "painting":
84                 $color = "orange";
85                 break;

```



26

Step 4

```
86     case "sculpture":
87         $color = "white";
88         break;
89     default:
90         $color = "blue";
91     }
92
93
94     echo "<tr bgcolor=$color>";
95
96     foreach ($row as $cell) {
97         echo "<td>$cell</td>";
98     }
99
100    echo "</tr>";
101 }
102
103 echo "</table>";
104
105 }else{
106     echo "<h4> No results </h4>";
107 }
108
```

DBG

27