# MySQL

|  |  |
| --- | --- |
| **Befehl** | **Beschreibung** |
| mysql –u Benutzername –pPasswort | MySQL Konsolenprompt öffnen |
| USE Datenbank; | Datenbank benutzen |
| CREATE DATABASE Datenbank; | Datenbank erstellen |
| CREATE TABLE Tabelle (  id INT(11) UNSIGNED AUTO\_INCREMENT PRIMARY KEY,  name VARCHAR(30),  email VARCHAR(128) NOT NULL  ); | Tabelle erstellen |
| SELECT \* FROM Tabelle; | Daten herauslesen |
| INSERT INTO Tabelle (name, email)  VALUES(„test“, „test@mail.com“); | Daten einfügen |
| DELETE FROM Tabelle WHERE id=1; | Datensatz löschen |
| UPDATE Tabelle SET name=“John Doe“ WHERE name=“test“; | Datensatz updaten |
| CREATE USER 'user'@'localhost' IDENTIFIED BY 'Passwort'; | Benutzer erstellen |
| GRANT ALL PRIVILEGES ON Datenbank.Tabelle TO 'user'@'localhost';  FLUSH PRIVILEGES; | Benutzer alle Rechte auf Tabelle geben. |
| REVOKE ALL PRIVILEGES ON Datenbank.Tabelle FROM 'user'@'localhost';  FLUSH PRIVILEGES; | Benutzer die Rechte entziehen |

# MySQL-Verbindung in PHP

|  |  |
| --- | --- |
| **Mit mysqli** | **Mit PDO** |
| $mysqli = new mysqli(**"**localhost**"**, **"**root**"**, **"**Passwort**"**, **"**Datenbank**"**)**;**  if ($mysqli -> connect\_error) {  die(**"**Fehler bei Verbindung**"**)**;**  }  $stmt = $mysqli -> prepare(**"**SELECT name, email FROM Tabelle WHERE id=? AND name=?**"**)**;**  //i=integer, s=string, b=blob, d=double  $stmt -> bind\_param(**"**is**"**, 1, **"**John**"**)**;**  $stmt -> execute()**;**  $stmt -> bind\_result($name, $email)**;**  while ($stmt -> fetch()) {  echo **"**Name: **"** . $name . **"**, E-Mail: **"** . $email**;**  }  $stmt -> close()**;**  $mysqli -> close(); | $pdo = new PDO(**"**mysql:host=localhost;dbname=Datenbank**"**, **"**root**"**, **"**Passwort**"**)**;**  $stmt = $pdo -> prepare(**"**SELECT name, email FROM Tabelle WHEHRE id=:id AND name=:name**"**)**;**  $stmt -> execute([  **"**:id**"** => 1,  **"**:name**"** => **"**John**"**  ])**;**  $result = $stmt -> fetchAll()**;**  foreach($result as $row) {  echo **"**Name: **"** . $row->name . **"**, E-Mail: **"** . $row->email**;**  } |

# Passwort-Hashing

Mit bcrypt (Stand PHP7) und generiertem Salt:

$hashedPassword = password\_hash("Passwort", PASSWORD\_BCRYPT);

if(password\_verify("Passwort", $hashedPassword)) {

// Password is valid!

}

# Sicherheit

|  |  |
| --- | --- |
| **Angriffsart** | **Schutz** |
| SQL-Injection | Prepared Statements, Escaping |
| Cross-Site-Scripting (XSS) | Escaping von Output -> htmlspecialchars |
| Session Fixation / Session Hijacking | session.cookie\_httponly auf On stellen HTTPS-Verschlüsselung |
| Cross-Site-Request-Forgery (CSRF) | CSRF-Token |
| Directory-Traversal | Benutzerrechte anpassen |
| E-Mail-Injection | Benutzerdaten filtern, Mail-Bibliothek verwenden |
| Man-in-the-Middle / Man-in-the-Browser | SSL-Verschlüsselung / Antivirus |
| Denial of Service (DOS / DDoS) | IP-Sperrliste, Rate Limiting, Load Balancing |
| DNS-Spoofing / Cache-Poisoning | DNSSEC |
| Phising | Menschenverstand |