

Appendix B

Jiun-Long Huang (黃俊龍)

Department of Computer Science

National Chiao Tung University

Materialized View

- ❑ MySQL does **not** support materialized view.
- ❑ PostgreSQL supports materialized view.
 - **CREATE MATERIALIZED VIEW** mymatview
AS SELECT * FROM mytab;

Transaction in MySQL

- ❑ To start a transaction, you use the `START TRANSACTION` statement.
 - The `BEGIN` or `BEGIN WORK` are the aliases of the `START TRANSACTION`.
- ❑ To commit the current transaction and make its changes permanent, you use the `COMMIT` statement.
- ❑ To roll back the current transaction and cancel its changes, you use the `ROLLBACK` statement.

-
- ❑ To disable or enable the auto-commit mode for the current transaction, you use the SET autocommit statement.
 - ❑ Not all engines of MySQL support transaction.
 - MyISAM (default engine of MySQL before version 5.5) does not.

Transaction in PHP

- `public PDO::beginTransaction () : bool`
 - Turns **off autocommit** mode.
 - While autocommit mode is turned off, changes made to the database via the PDO object instance are not committed until you end the transaction by calling `PDO::commit()`.
 - Calling `PDO::rollBack()` will roll back all changes to the database and return the connection to autocommit mode.

-
- `public PDO::commit () : bool`
 - Commits a transaction, returning the database connection to **autocommit** mode until the next call to `PDO::beginTransaction()` starts a new transaction.

-
- `public PDO::rollBack () : bool`
 - Rolls back the current transaction, as initiated by `PDO::beginTransaction()`.
 - If the database was set to **autocommit** mode, this function will restore autocommit mode after it has rolled back the transaction.

-
- public PDO::inTransaction () : bool
 - Checks if a transaction is currently active within the driver. This method only works for database drivers that support transactions.

□ Table accounts

- AccountName: char(10) not null
- Amount: int(11) not null

Transaction Test

```
<!DOCTYPE html>
<html>
<body>
<h1>Transaction Test</h1>
<form action="test.php" method="post">
  <input type="submit" value="Transfer">
</form>

</body>
</html>
```

Transfer

```
<?php
```

```
$dbservername='localhost';
```

```
$dbname='examdb';
```

```
$dbusername='examdb';
```

```
$dbpassword='examdb';
```

test.php

```
try
```

```
{
```

```
    $conn = new PDO("mysql:host=$dbservername;dbname=$dbname",  
$dbusername, $dbpassword);
```

```
    # set the PDO error mode to exception
```

```
    $conn->setAttribute(PDO::ATTR_ERRMODE,  
PDO::ERRMODE_EXCEPTION);
```

```
    $conn->beginTransaction();
```

```
    $stmt=$conn->prepare("insert into accounts (AccountName,  
Amount) values ('Alice', 100)");
```

```
    $stmt->execute();
```

```
    $stmt=$conn->prepare("insert into accounts (AccountName,  
Amount) values ('Bob', null)");
```

```
$stmt->execute();  
$conn->commit();  
    echo <<<EOT  
    <!DOCTYPE html>  
    <html>  
        <body>  
            <script>  
                alert("$msg");  
                window.location.replace("index.php");  
            </script>  
        </body>  
    </html>  
EOT;  
}
```

```
catch(Exception $e)
{
    if ($conn->inTransaction())
        $conn->rollBack();
    $msg=$e->getMessage();
    echo <<<EOT
        <!DOCTYPE html>
        <html>
            <body>
                <script>
                    alert("$msg");
                    window.location.replace("index.php");
                </script>
            </body>
        </html>
    EOT;
}
?>
```

Prepared Statement in PDO

❑ Without prepared statement

```
$uname=$_POST['uname'];  
$pwd=$_POST['pwd'];  
$conn = new PDO...  
...  
$stmt=$conn->prepare("select username, password,  
salt from users where username='" . $uname . "'");  
$stmt->execute();
```

□ With prepared statement

```
$uname=$_POST['uname'];  
$pwd=$_POST['pwd'];  
$conn = new PDO...  
...  
$stmt=$conn->prepare("select username, password,  
salt from users where username=:username");  
$stmt->execute(array('username' => $uname));
```

Reference

- ❑ <https://www.mysqltutorial.org/mysql-transaction.aspx/>
- ❑ <https://www.php.net/manual/en/pdo.transactions.php>
- ❑ <https://www.postgresql.org/docs/current/rules-materializedviews.html>