```php
<?php
/**
 * Database model for employee.
 * Inserting and Selecting based on controller statement
 */
include "env.php";

class DB {
  public $erMessage = "";
  private $pdo = null;
  private $stmt = null;

  // Establish connection to Database
  function __construct () {
    try {
      // Using PDO over Mysqli since PDO works with more than just mysql
databases
      // These same aspects can we rearranged for mysqli instances
      $this->pdo = new PDO(
        "mysql:host=".DB_HOST.";dbname=".DB_NAME,
        DB_USER, DB_PASSWORD, [
          PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION, // Error mode to
use
          PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC // Use
associative array instead of object
        ]
      );
    } catch (Exception $e) {
      die($e->getMessage());
    }
  }

  // Close connection with database
  function __destruct(){
    //Remove existing statements
    if ($this->stmt!==null) {
        $this->stmt = null;
    }
    //Remove pdo instance (close db connection)
    if ($this->pdo!==null) {
        $this->pdo = null;
    }
  }

   /**
   * Select Action
   * @return Array of objects or error statment
   */
  function select($sql, $cond = null){
    try {
      $this->stmt = $this->pdo->prepare($sql); // Prepare sql statemnt
      $this->stmt->execute($cond); // Execute sql statement with
interpolated data
      return $this->stmt->fetchAll(); //Get selected items
```

```
    } catch (Exception $ex) {
      $this->erMessage = $ex->getMessage();
      return false;
    }
  }

  /**
   * Insert Action
   * @return Integer of last added id or error message
   */
  function insert($sql, $cond = null){
    try {
        $this->stmt = $this->pdo->prepare($sql);
        return $this->stmt->execute($cond);
      } catch (Exception $ex) {
        $this->erMessage = $ex->getMessage();
        return false;
      }
  }
}
?>
```