PDO Cheat Sheet

$result = $dbc->query($sql);	Performs a query on the database using the current database connection and SQL statement. Returns TRUE if the query runs, FALSE if not. (NOTE: TRUE is returned even if the number of rows in the result is zero.)
foreach($result as $row){ }	Iterate through the table, one row at a time, using associative arrays indexed from the database field names.
$numRows = $result->rowCount();	Returns the number of rows in the result set from a SELECT operation
$row = $result->fetch();	Fetches one row of the result as an associative array mapped to the database fields as keys.
$item = $result->fetchColumn();	Gets the first record in the first column of result. To get results from other columns, use column number (starting from 0) as argument.
$result->closeCursor();	An optional (but good practice) command to free the memory taken by $result. Needed if there are subsequent queries.
$affected = $dbc->exec($sql)	Submit non-SELECT queries (INSERT, UPDATE, DELETE). Although query() can be used, exec() returns the number of affected rows.
$id = $dbc->lastInsertId();	Get the ID generated from the previous INSERT operation

Prepared Statements

$stmt = $dbc->prepare($sql);	**Prepare the sql statement.**
$stmt->bindParam(num, $variable);** **$stmt->bindParam(':placeholder', $variable);	**Bind the parameters using one statement for each variable** **Do not include quotes around the variables, even it they represent strings.**
$stmt->execute();	**Execute the prepared statement**
$result = $stmt->fetch();	**Fetch one row of the result as an associative array.**
$result = $stmt->fetchAll();	**Fetch all rows of the query result.**
$numRows=$result->rowCount();	**Returns affected rows for UPDATE, DELETE, INSERT or number of rows for SELECT.**