Updating Existing Records

An update changes an existing row. The application must identify the row, validate changes and prevent unintended updates.

Load the selected record

```
$id = filter_input(INPUT_GET, "id", FILTER_VALIDATE_INT);
if (!$id) {
    exit("Invalid activity.");
}

$stmt = $pdo->prepare(
    "SELECT activity_id, activity_name, capacity
     FROM activities
     WHERE activity_id = :id"
);
$stmt->execute(["id" => $id]);
$activity = $stmt->fetch();

if (!$activity) {
    exit("Activity not found.");
}
```

Apply a valid change

```
$capacity = filter_input(INPUT_POST, "capacity", FILTER_VALIDATE_INT);

if ($capacity === false || $capacity < 1 || $capacity > 100) {
    $error = "Capacity must be from 1 to 100.";
} else {
    $stmt = $pdo->prepare(
        "UPDATE activities
```

```
        SET capacity = :capacity
        WHERE activity_id = :id"
    );
    $stmt->execute(["capacity" => $capacity, "id" => $id]);
}
```

Always include a precise `WHERE` condition. An update without it can change every row.

Authorisation

A valid login does not automatically grant editing permission. Check the session role before showing or processing administrative actions.

Check

- [ ] Identifier is validated.
- [ ] Missing records are handled.
- [ ] New values are validated.
- [ ] Prepared statements are used.
- [ ] The update has a precise `WHERE` clause.
- [ ] Authorisation is checked on the server.

Revision #1
Created 2026-08-18 23:33:48 UTC by Mr Napper
Updated 2026-08-18 23:33:52 UTC by Mr Napper