

## 1.1. Adding Multiple Expenses with INSERT

To add multiple new expense records to your Expense Tracker, you can use a single INSERT statement with multiple VALUES clauses. This is efficient and allows you to insert several rows at once.

### Example SQL:

```
INSERT INTO expenses (category, amount, date)
```

```
VALUES
```

```
('Groceries', 150.00, '2024-11-10'),
```

```
('Office Supplies', 80.00, '2024-11-10'),
```

```
('Travel', 200.00, '2024-11-12'),
```

```
('Entertainment', 50.00, '2024-11-14'),
```

```
('Groceries', 120.00, '2024-11-15');
```

### Explanation:

- We are inserting multiple records into the expenses table at once.
  - Each set of values (category, amount, date) is inserted as a new row in the table.
- 

## 1.2. Updating Expense Information

To modify an existing record in your expenses table, you can use the UPDATE statement. The WHERE clause is essential to target the specific record you want to update.

### Example SQL:

```
UPDATE expenses
```

```
SET amount = 180.00, category = 'Groceries'
```

```
WHERE id = 3;
```

### Explanation:

- The UPDATE statement modifies the amount and category of the record where the id equals 3.

- Without the WHERE clause, all records in the table would be updated, so it's crucial to specify the exact record you want to update.
- 

### 1.3. Selective Data Deletion

You can delete records from a table using the DELETE statement. For safety, always create a copy of the table before performing deletions to avoid accidental data loss.

#### Copying the expenses Table:

```
CREATE TABLE expenses_copy AS SELECT * FROM expenses;
```

#### Delete Expense Entries Where the Amount is Zero:

```
DELETE FROM expenses_copy
```

```
WHERE amount = 0;
```

#### Delete Expense Entries Before a Specific Date:

```
DELETE FROM expenses_copy
```

```
WHERE date < '2023-07-01';
```

#### Explanation:

- The first DELETE statement removes all entries with an amount of zero from the expenses\_copy table.
  - The second DELETE statement removes all entries with a date before July 1st, 2023.
  - Always double-check the WHERE clause to ensure you are deleting the correct records.
- 

### Bonus Challenge: Handling NULL Values

To handle NULL values, you can check for them using IS NULL or IS NOT NULL in a query, and then decide how to handle them, such as updating or replacing them with default values.

#### Check for NULL values in the expenses table:

```
SELECT * FROM expenses
```

```
WHERE amount IS NULL;
```

**Update NULL values with a default value (e.g., set NULL amounts to 0):**

UPDATE expenses

SET amount = 0

WHERE amount IS NULL;

**Explanation:**

- The first query retrieves all records where the amount is NULL.
- The second query replaces any NULL value in the amount column with 0.