

# Admin Guide

---

## Admin Access

### Getting Admin Access

#### 1. Sign Up/In:

- Create account or sign in with existing account
- Default role is "user"

#### 2. Assign Admin Role:

- Contact system administrator
- Admin must run SQL to assign role:

#### 3. Access Dashboard:

- Sign in with admin account
- Navigate to </admin/dashboard>

## Admin Dashboard Overview

### Navigation

- **Import Data Tab:** CSV import functionality
- **View Data Tab:** Browse database tables
- **Sign Out:** Logout button in header

## CSV Import System

### Overview

The CSV import system allows admins to bulk import data into the database. It supports validation, error reporting, and dry-run mode.

### Supported Tables

#### 1. Lookup Tables:

- [lookups\\_categories](#) - Dish categories
- [lookups\\_units](#) - Measurement units

#### 2. Location Data:

- [chains](#) - Supermarket chains
- [ad\\_regions](#) - Advertising regions
- [stores](#) - Store locations
- [postal\\_codes](#) - Postal code mappings
- [store\\_region\\_map](#) - Store-region relationships

### 3. Product Data:

- **ingredients** - Ingredients
- **dishes** - Dishes/recipes
- **dish\_ingredients** - Dish-ingredient relationships

### 4. Offers:

- **offers** - Current supermarket offers

## Import Process

### Step 1: Prepare CSV File

#### Requirements:

- CSV format (comma-separated)
- First row must be headers (column names)
- Headers must match database column names exactly
- UTF-8 encoding recommended

#### Example CSV (offers):

```
region_id,ingredient_id,price_total,pack_size,unit_base,valid_from,valid_to,source
,source_ref_id
500,I001,2.99,500,g,2025-01-13,2025-01-19,aldi,OFFER123
500,I002,1.49,1,1,2025-01-13,2025-01-19,lidl,OFFER456
```

### Step 2: Upload CSV

1. Go to "Import Data" tab
2. Click "Choose File" or drag and drop
3. Select CSV file from your computer
4. Select table type from dropdown
5. Choose mode:
  - **Dry Run:** Validates without importing
  - **Import:** Validates and imports data

### Step 3: Review Results

#### Dry Run Results:

- Number of valid rows
- List of errors (if any)
- No data imported

#### Import Results:

- Number of valid rows

- Number of rows imported
- List of errors (if any)

## CSV Format Specifications

### Offers CSV

#### Required Columns:

- `region_id` - Integer (must exist in `ad_regions`)
- `ingredient_id` - Text (I001 format) or integer (auto-converted)
- `price_total` - Decimal (e.g., 2.99 or 2,99)
- `pack_size` - Decimal (e.g., 500.0)
- `unit_base` - Text (must exist in `lookups_units`)
- `valid_from` - Date (YYYY-MM-DD)
- `valid_to` - Date (YYYY-MM-DD)

#### Optional Columns:

- `source` - Text
- `source_ref_id` - Text

#### Example:

```
region_id,ingredient_id,price_total,pack_size,unit_base,valid_from,valid_to,source
500,I001,2.99,500,g,2025-01-13,2025-01-19,aldi
501,I002,1.49,1,1,2025-01-13,2025-01-19,lidl
```

### Dishes CSV

#### Required Columns:

- `dish_id` - Text (unique identifier)
- `name` - Text
- `category` - Text (must exist in `lookups_categories`)
- `is_quick` - Boolean (TRUE/FALSE)
- `is_meal_prep` - Boolean (TRUE/FALSE)

#### Optional Columns:

- `season` - Text
- `cuisine` - Text
- `notes` - Text

#### Example:

```
dish_id,name,category,is_quick,is_meal_prep,season,cuisine
D001,Spaghetti Carbonara,Main Course,FALSE,FALSE,,
```

```
D002,Quick Pasta,Main Course,TRUE,FALSE,,
```

## Ingredients CSV

### Required Columns:

- `ingredient_id` - Text (I001 format)
- `name_canonical` - Text
- `unit_default` - Text (must exist in lookups\_units)

### Optional Columns:

- `price_baseline_per_unit` - Decimal
- `allergen_tags` - Text (comma-separated)
- `notes` - Text

### Example:

```
ingredient_id,name_canonical,unit_default,price_baseline_per_unit,allergen_tags
I001,Tomatoes,kg,2.99,"gluten,soy"
I002,Milk,l,1.29,"dairy"
```

## Dish Ingredients CSV

### Required Columns:

- `dish_id` - Text (must exist in dishes)
- `ingredient_id` - Text (must exist in ingredients)
- `qty` - Decimal
- `unit` - Text (must exist in lookups\_units)
- `optional` - Boolean (TRUE/FALSE)

### Optional Columns:

- `role` - Text

### Example:

```
dish_id,ingredient_id,qty,unit,optional,role
D001,I001,500,g,FALSE,main
D001,I002,2,stück,FALSE,
D001,I003,1,TL,TRUE,
```

## Import Order

**Critical:** Import data in this order to avoid foreign key errors:

### 1. Lookup Tables:

- `lookups_categories`
- `lookups_units`

### 2. Location Data:

- `chains`
- `ad_regions`
- `stores`
- `postal_codes`
- `store_region_map`

### 3. Product Data:

- `ingredients`
- `dishes`
- `dish_ingredients`
- `product_map`

### 4. Offers:

- `offers` (last, depends on all above)

## Common Errors & Solutions

### "Foreign key constraint violation"

**Cause:** Referenced data doesn't exist

#### Solutions:

- Import referenced tables first
- Check that IDs match exactly
- Verify data exists in parent table

#### Example:

- Error: `ingredient_id "I999" not found`
- Solution: Import ingredients CSV first, ensure I999 exists

### "Invalid date format"

**Cause:** Date not in YYYY-MM-DD format

#### Solutions:

- Use format: `2025-01-13`
- No slashes or dots
- Include leading zeros

#### Example:

- Wrong: 1/13/2025, 13.01.2025
- Correct: 2025-01-13

### "Invalid number"

**Cause:** Non-numeric value in number field

#### **Solutions:**

- Use numbers only (decimals with dot or comma)
- No currency symbols
- No text in number fields

#### **Example:**

- Wrong: €2.99, 2,99 EUR
- Correct: 2.99 or 2,99

### "Unit not found"

**Cause:** Unit doesn't exist in lookups\_units

#### **Solutions:**

- Import units lookup first
- Check unit name matches exactly (case-sensitive)
- Common units: g, kg, ml, l, stück, st

### "Wrong number of columns"

**Cause:** Extra commas or missing values

#### **Solutions:**

- Check for trailing commas
- Ensure all rows have same column count
- Use empty values (not missing) for optional fields

### Best Practices

#### **1. Always Use Dry Run First:**

- Validate data before importing
- Fix errors before live import
- Saves time and prevents bad data

#### **2. Import in Order:**

- Follow the import order guide
- Don't skip steps
- Verify each step before proceeding

### 3. **Validate Data Quality:**

- Check for duplicates
- Ensure date ranges are valid

### 4. **Keep Backups:**

- Export data before major imports
- Keep CSV files for reference
- Document import dates

### 5. **Test with Small Files:**

- Start with 10-20 rows
- Verify results
- Scale up once confirmed

## Viewing Data

### Data Table Browser

#### 1. **Select Table:**

- Choose table from dropdown
- Click "Load Data"

#### 2. **View Data:**

- Table displays rows
- Pagination for large tables
- Search and filter (if implemented)

### Available Tables

All database tables are viewable:

- Product data (dishes, ingredients)
- Location data (chains, stores, regions)
- Offers data
- Lookup tables

## Data Management

### Updating Data

#### **Method 1: CSV Import (Upsert)**

- Import CSV with existing IDs
- System updates existing rows
- Adds new rows if ID doesn't exist

#### **Method 2: Direct Database Access**

- Use Supabase SQL Editor
- Run UPDATE statements
- More control but requires SQL knowledge

## Deleting Data

### **Cascade Deletes:**

- Deleting a dish deletes dish\_ingredients
- Deleting a chain deletes stores and regions
- Be careful with deletions

### **Safe Deletion:**

- Check dependencies first
- Use soft deletes (mark as inactive) if possible
- Keep backups

## Data Validation

### **Regular Checks:**

- Verify offer dates are current
- Check for orphaned records
- Validate price calculations
- Monitor error logs

## Troubleshooting

### Import Fails Completely

#### **Check:**

- File format (must be CSV)
- File encoding (UTF-8)
- File size (not too large)
- Network connection

### Partial Import Success

#### **Review Errors:**

- Some rows imported, some failed
- Check error list for patterns
- Fix errors and re-import failed rows

### Data Not Showing in App

#### **Possible Causes:**

- Offers expired (check dates)
- Missing region mapping



**Solutions:**

- Verify data in database
- Check RLS policies
- Test with admin account

**Performance Issues****Large Imports:**

- Split into smaller files
- Import in batches
- Monitor database performance

**Slow Queries:**

- Check database indexes
- Optimize queries
- Contact Supabase support

**Remember:** Always use dry-run mode first, and import data in the correct order!