

Migration 015 Fix Summary

Date: December 1, 2025

Issue Report

A reported error indicated that migration 015 was failing with:

```
ERROR: 42703: column ak.customer_id does not exist
LINE 40: LEFT JOIN api_keys ak ON c.id = ak.customer_id
```

Analysis Results

1. Database Schema Investigation

After examining the database schema, I found that:

api_keys table structure:

- id (PRIMARY KEY)
- api_client_id (FOREIGN KEY → api_clients.id)
- key_hash, key_prefix, name
- scopes, rate_limit
- is_active, last_used_at, expires_at
- created_at, updated_at

✗ api_keys does NOT have a customer_id column

2. Correct Relationship Chain

The relationship between api_keys and customers is **indirect**:

```
api_keys
  → api_client_id
    → api_clients
      → customer_id
        → customers
```

3. Current Migration Status

✓ The migration 015 file is CORRECT and does NOT contain the reported error.

The current migration file (015_add_credits_to_customer_stats.sql) only joins with customer_credits table, which correctly has a customer_id column.

```
FROM customers c
LEFT JOIN users u ON c.id = u.customer_id
LEFT JOIN search_history sh ON c.id = sh.customer_id
LEFT JOIN customer_credits cc ON c.id = cc.customer_id -- ✓ CORRECT
-- No api_keys join present
```

Root Cause

The error message suggests someone may have:

1. **Tried to add an api_keys join manually** with incorrect syntax
2. **Used an old/incorrect version** of the migration file
3. **Misunderstood the error message** from a different query

Resolution

Files Created/Updated:

1. **SUPABASE_MIGRATION_015.sql** - Ready-to-use Supabase migration file
 - Contains the correct view definition
 - Includes verification queries
 - Documents the api_keys relationship
 - No api_keys join (correctly omitted)
2. **015_add_credits_to_customer_stats.sql** - Already correct
 - No changes needed
 - Verified to have correct joins only

How to Apply:

Option 1: Supabase SQL Editor

```
-- Copy and paste the contents of SUPABASE_MIGRATION_015.sql
-- into Supabase SQL Editor and execute
```

Option 2: Node.js Script

```
cd /home/ubuntu/roblox-tool
node scripts/run-migration-015.js
```

Option 3: Direct PostgreSQL

```
psql $DATABASE_URL < database/migrations/015_add_credits_to_customer_stats.sql
```

Verification

After running the migration, verify with:

```
-- Check new columns exist
SELECT column_name, data_type
FROM information_schema.columns
WHERE table_name = 'customer_stats'
AND column_name IN ('credit_balance', 'total_credits_purchased',
'total_credits_used', 'last_purchase_at');
```

Expected result: 4 rows showing the credit-related columns.

```
-- Test the view
SELECT id, name, credit_balance, total_credits_purchased
FROM customer_stats
LIMIT 5;
```

Important Notes

Why api_keys is NOT in the view:

1. **No direct relationship** - api_keys doesn't have customer_id
2. **Not needed for credit tracking** - Credits are tracked in customer_credits table
3. **Would require complex joins** - Would need to join through api_clients first
4. **Not relevant to customer stats** - API keys are separate from customer credit balance

If You Need API Key Count in Customer Stats:

If you need to show API key count per customer, you would need:

```
-- This is the CORRECT way to join api_keys to customers
LEFT JOIN (
  SELECT ac.customer_id, COUNT(ak.id) as api_key_count
  FROM api_clients ac
  LEFT JOIN api_keys ak ON ac.id = ak.api_client_id AND ak.is_active = true
  GROUP BY ac.customer_id
) api_key_stats ON c.id = api_key_stats.customer_id
```

But this is **NOT** needed for credit balance tracking, which is the purpose of migration 015.

Files Reference

- /database/migrations/015_add_credits_to_customer_stats.sql - Main migration file ✓
- /database/migrations/SUPABASE_MIGRATION_015.sql - Supabase-ready version ✓
- /database/migrations/APPLY_MIGRATION_015.md - Application instructions ✓
- /scripts/run-migration-015.js - Node.js runner script ✓
- /prisma/schema.prisma - Schema reference for table structures ✓

Next Steps

1. ✓ Migration file verified - no api_keys issues
2. ✓ SUPABASE_MIGRATION_015.sql created
3. ✓ Documentation updated
4. ⌚ Ready to apply migration to database
5. ⌚ Verify customer_stats view includes credit data
6. ⌚ Test admin dashboard displays credit balances

Conclusion

The migration 015 file is **already correct** and does not need fixing. The reported error likely came from a manual edit attempt or a misunderstanding. The file can be applied safely using any of the three methods above.

Status:  READY TO DEPLOY

Risk Level: LOW (No schema changes, only view recreation)

Rollback: Can revert to migration 013 if needed